

**Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596**

**PERCEPTUALLY GROUNDED LANGUAGE ACQUISITION:
A NEURAL/PROCEDURAL HYBRID MODEL**

V. Nenov

**December 1991
CSD-910083**

**Perceptually Grounded Language Acquisition:
A Neural/Procedural Hybrid Model**

Valeriy Iliev Nenov

December 1991

Technical Report UCLA-AI-91-07

Author's current address:

Valeriy Nenov, Ph.D., Ph.D.
Division of Neurosurgery
School of Medicine, 74-140 CHS
University of California,
Los Angeles, CA 90024-1301

UNIVERSITY OF CALIFORNIA

Los Angeles

**Perceptually Grounded Language Acquisition:
A Neural/Procedural Hybrid Model**

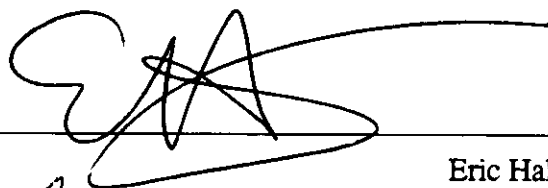
A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

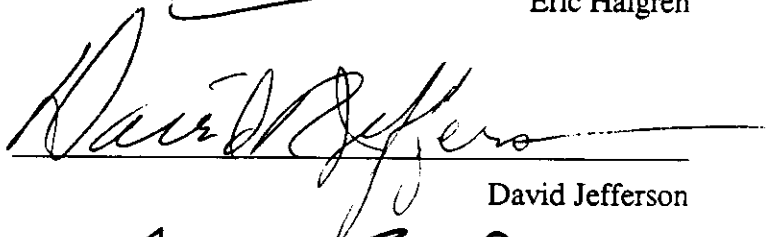
Valeriy Iliev Nenov, Ph.D.

Copyright © 1991 by Valeriy Iliev Nenov, Ph.D.
All Rights Reserved


The dissertation of Valeriy Iliev Nenov is approved.



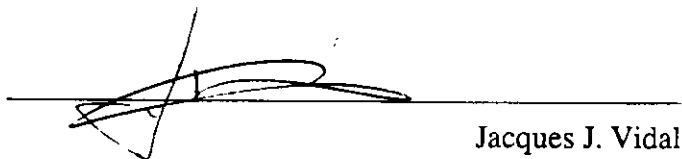
Eric Halgren



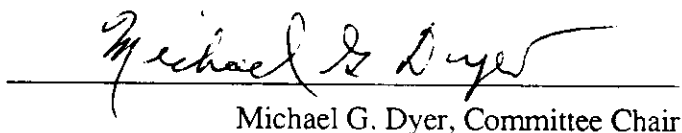
David Jefferson



Arnold Scheibel



Jacques J. Vidal



Michael G. Dyer, Committee Chair

University of California, Los Angeles

1991

*To my children,
Katarina and Michael Nenov*

TABLE OF CONTENTS

| CHAPTER | | PAGE |
|----------|--|-----------|
| PART I | An Overall View | 1 |
| 1 | Introduction | 2 |
| 1.1 | Task: Perceptually Grounded Language Acquisition..... | 2 |
| 1.1.1 | Semantics..... | 5 |
| 1.1.2 | Pragmatics:..... | 8 |
| 1.1.3 | Syntax | 9 |
| 1.2 | DETE: A Neural Architecture | 10 |
| 1.2.1 | Theoretical Issues | 10 |
| 1.2.2 | DETE in action..... | 11 |
| 1.2.3 | Overview of implementation | 12 |
| 1.3 | Motivations and goals..... | 12 |
| 1.4 | Background..... | 14 |
| 1.4.1 | Philosophy..... | 14 |
| 1.4.2 | Methodology..... | 16 |
| 1.5 | Guide to the reader | 17 |
| 2 | Scope of Task and Overall DETE Architecture | 20 |
| 2.1 | Input and Output Behavior | 20 |
| 2.1.1 | Input | 20 |
| 2.1.2 | Tasks..... | 20 |
| 2.2 | Learning tasks and relation of learning to performance | 23 |
| 2.3 | Architecture | 24 |
| 2.3.1 | Input devices..... | 25 |
| 2.3.2 | Selective Attention System..... | 27 |
| 2.3.3 | Visual Feature Extractors..... | 28 |
| 2.3.4 | Memories..... | 28 |
| 2.4 | DETE's Micro World | 30 |
| 2.4.1 | The Blobs world..... | 30 |
| 2.4.2 | The verbal modality | 32 |
| 2.4.3 | The motor modality | 33 |
| 2.4.4 | Consistency of inputs..... | 34 |
| 2.5 | DETE's World | 34 |

| | | |
|--|--|-----------|
| 2.5.1 | Internal World | 34 |
| 2.5.2 | Physics of External World | 35 |
| 2.6 | Modes of operation..... | 35 |
| 2.7 | Learning in DETE | 35 |
| PART II Modular structure of DETE | | 36 |
| 3 | The Visual system..... | 37 |
| 3.1 | The retina (EYE)..... | 37 |
| 3.2 | Visual feature extractors | 39 |
| 3.2.1 | Shape Feature Extractor | 39 |
| 3.2.2 | Size Feature Extractor..... | 41 |
| 3.2.3 | Color Feature Extractor..... | 43 |
| 3.2.4 | Location Feature Extractor | 44 |
| 3.2.5 | Motion Feature Extractor | 46 |
| 4 | The Verbal System..... | 49 |
| 4.1 | Verbal input segmentation..... | 55 |
| 4.2 | Verbal output..... | 55 |
| 5 | Temporal relations in DETE..... | 57 |
| 5.1 | Chunking the input..... | 58 |
| 5.1.1 | Recognition of synchrony/asynchrony (level-0-chunks) | 58 |
| 5.1.2 | Recognition of temporal order (level-1-chunks)..... | 59 |
| 5.1.3 | Level-2-chunks (words)..... | 59 |
| 5.1.4 | Recognition of subjective NOW (level-3-chunks)..... | 59 |
| 5.1.5 | Recognition of duration..... | 60 |
| 5.3 | Other processing issues..... | 60 |
| 6 | The Motor system..... | 61 |
| 6.1 | Representation of effector states (proprioception)..... | 62 |
| 6.1.1 | Representation of FINGER State..... | 62 |
| 6.1.2 | Representation of EYE state..... | 65 |
| 6.2 | Types and ranges of effector motions..... | 65 |
| 6.2.1 | EYE motions..... | 65 |
| 6.2.2 | FINGER motions | 66 |
| 6.3 | Motor memory..... | 67 |
| 7 | Selective Attention in DETE..... | 68 |
| 7.1 | Representation of attention in DETE | 68 |

| | | |
|-----------|--|------------|
| 7.2 | The Selective Attention Mechanism..... | 69 |
| 7.2.1 | Input Segmentation Mechanism..... | 71 |
| 7.2.2 | Focus of Attention Master..... | 71 |
| 7.3 | Control of selective attention..... | 72 |
| 8 | Basic memory mechanisms..... | 73 |
| 8.1 | Network architecture..... | 73 |
| 8.1.1 | Predictron..... | 74 |
| 8.1.2 | Recognitron..... | 76 |
| 8.1.3 | Bi-Stable Switch..... | 79 |
| 8.1.4 | A small scale example..... | 79 |
| 8.1.5 | Parameters & their values in a complete system..... | 82 |
| 8.2 | Network dynamics..... | 83 |
| 8.2.1 | The KATAMIC algorithm..... | 83 |
| 8.2.2 | Illustration of KATAMIC's dynamics..... | 88 |
| 8.2.3 | Signal Flow in the KATAMIC model..... | 100 |
| 8.3 | Implementation on the Connection Machine..... | 101 |
| 8.4 | Simulations..... | 101 |
| 8.4.1 | Performance dependence on the T_s & T_i decay constants..... | 101 |
| 8.4.2 | Effects of "1-bit-density" of the input sequences..... | 102 |
| 8.4.3 | Effect of noise in the patterns..... | 106 |
| 8.4.4 | Learning branching sequences..... | 108 |
| 8.4.5 | Memory capacity..... | 112 |
| 8.5 | Summary of KATAMIC characteristics..... | 117 |
| 9 | Taxonomy of memory in DETE..... | 119 |
| 9.1 | Short-Term Memory..... | 119 |
| 9.1.1 | STM implementation..... | 121 |
| 9.2 | Long-Term Memory (LTM)..... | 122 |
| 9.2.1 | Declarative Memory (DM)..... | 122 |
| 9.2.2 | The Procedural Memory (PM)..... | 125 |
| 9.3 | Representation of time in DETE..... | 132 |
| 10 | Putting it all together..... | 136 |
| 10.1 | Characteristics of the memory modules in DETE..... | 136 |
| 10.1.1 | Dimensionality of memory modules..... | 136 |
| 10.1.2 | 1-bit-density of inputs..... | 136 |
| 10.1.3 | Connection patterns and strengths within modules..... | 137 |

| | | |
|--|--|------------|
| 10.1.4 | Seed distribution..... | 138 |
| 10.1.5 | Winner Take All mechanism (WTA)..... | 143 |
| 10.2 | Interfacing the individual memory modules..... | 146 |
| 10.2.1 | Connectivity patterns between visual modules..... | 146 |
| 10.2.2 | Connectivity between verbal and visual memory modules..... | 147 |
| 10.2.3 | DETE's complete memory architecture..... | 148 |
| PART III Performance and Evaluation | | 150 |
| 11 | Incremental Language Acquisition..... | 151 |
| 11.1 | Experimental protocol..... | 151 |
| 11.2 | Learning single words..... | 152 |
| 11.2.1 | Learning words for objects..... | 152 |
| 11.2.2 | Learning words for events..... | 155 |
| 11.3 | Generalization..... | 163 |
| 11.3.1 | Verbal generalization..... | 163 |
| 11.3.2 | Visual Generalization..... | 164 |
| 11.4 | Learning Question/Answer Sequences..... | 167 |
| 11.5 | Learning spatial relations between two objects..... | 169 |
| 11.5.1 | Learning about size relations..... | 169 |
| 11.5.2 | Learning location relations between objects..... | 172 |
| 11.6 | Learning about motion relations..... | 177 |
| 11.7 | Learning temporal relations between events..... | 179 |
| 11.7.1 | Present tense..... | 181 |
| 11.7.2 | Future tense..... | 184 |
| 11.7.3 | Past tense..... | 186 |
| 11.8 | Learning Homonyms..... | 189 |
| 11.9 | Learning selected features of different languages..... | 191 |
| 11.10 | Discussion..... | 193 |
| PART IV Comparison | | 196 |
| 12 | Comparison to other work..... | 197 |
| 12.1 | Connectionist models of NLP..... | 197 |
| 12.1.1 | Localist connectionist models of NLP..... | 197 |
| 12.1.2 | Distributed connectionist models of NLP..... | 198 |
| 12.2 | Tough problems for neural network models of NLP..... | 201 |
| 12.2.1 | Formation of distributed representations..... | 201 |

| | | |
|-----------|---|------------|
| 12.2.2 | Types vs tokens | 202 |
| 12.2.3 | Role binding | 203 |
| 12.3 | Associative memory models | 204 |
| 12.4 | Self-organizing feature maps..... | 204 |
| 12.4.1 | Kohonen's feature maps..... | 204 |
| 12.4.2 | von der Malsburg's "dynamic link architecture"..... | 206 |
| 12.5 | Sequence processing..... | 207 |
| 12.5.1 | Comparison of KATAMIC and Kanerva's SDM models | 209 |
| 12.5.2 | Comparison of KATAMIC and Elman's SRN model | 211 |
| 12.5.3 | KATAMIC vs TDNN..... | 219 |
| 12.6 | Other models of selective attention | 220 |
| 12.6.1 | Fukushima's Neocognitron | 220 |
| 12.6.2 | Crick's "Searchlight of attention" hypothesis..... | 221 |
| 12.7 | Other Systems for Sensory-Motor integration | 222 |
| 12.7.1 | Darwin I, II, and III..... | 222 |
| 12.7.2 | Gary Drescher | 224 |
| 12.8 | Representation of space & time..... | 225 |
| 12.8.1 | George Lakoff..... | 225 |
| 12.8.2 | Leonard Talmy..... | 225 |
| 12.8.3 | Reichenbach..... | 226 |
| 12.9 | Other work in symbol grounding..... | 226 |
| 12.9.1 | Josep Maria Sopena..... | 226 |
| 12.9.2 | The LO miniature language acquisition project..... | 227 |
| 12.9.3 | MAIMRA & DAVRA..... | 228 |
| 12.9.4 | RobotWorld..... | 228 |
| 12.10 | Symbolic models of NLP | 229 |
| 12.11 | DETE versus language acquisition in children..... | 230 |
| 13 | Neuropsychological & Neurobiological Insights..... | 231 |
| 13.1 | Neural codes -- Discussion of representations..... | 231 |
| 13.2 | Visual Perception..... | 231 |
| 13.2.1 | The retina..... | 233 |
| 13.2.2 | Segmentation -- figure/ground separation | 233 |
| 13.2.3 | Motion representation..... | 234 |
| 13.2.4 | Shape representation..... | 235 |
| 13.2.5 | Location (position) variability representation | 235 |
| 13.2.6 | Visual associations | 236 |

| | | |
|-----------|---|------------|
| 13.2.7 | Imagery..... | 237 |
| 13.3 | The verbal subsystem..... | 237 |
| 13.3.1 | Input/Output and central processing of language | 238 |
| 13.3.2 | Language disorders (impairment by lesions)..... | 241 |
| 13.3.3 | Models of Language processing in the brain..... | 243 |
| 13.3.4 | Mapping of DETE's verbal modules to the brain..... | 245 |
| 13.3.5 | Hidden speech | 245 |
| 13.3.6 | Temporal aspects of language processing | 246 |
| 13.4 | Neural basis of Selective Attention..... | 246 |
| 13.4.1 | Control of the attentional focus | 247 |
| 13.4.2 | The neural plausibility of DETE's attentional mechanism | 249 |
| 13.4.3 | Anatomical localization of Selective Attention | 249 |
| 13.5 | Neural plausibility of the KATAMIC model | 253 |
| 13.6 | Neuropsychology of memory..... | 258 |
| 13.6.1 | Short-term memory (STM)..... | 258 |
| 13.6.2 | Long term memory (LTM) | 259 |
| 13.6.3 | The working memory..... | 262 |
| 14 | Current Status, Future Work and conclusions | 263 |
| 14.1 | Current status | 263 |
| 14.2 | Future work..... | 263 |
| 14.2.1 | Neurally realistic modules and connectivity..... | 263 |
| 14.2.2 | Additional language capacity..... | 264 |
| 14.2.3 | Additional basic cognitive capacities | 275 |
| 14.2.4 | Higher cognitive functions | 276 |
| 14.2.5 | Real time operation in real environments | 277 |
| 14.2.6 | Research on psychological validity..... | 277 |
| 14.3 | Conclusions..... | 278 |
| | Bibliography..... | 281 |
| | Appendix A: Implementation Details..... | 300 |
| A.1 | The CM-2 Connection Machine | 300 |
| A.2 | The CM-2 at UCLA..... | 300 |
| A.3 | The *LISP programming language..... | 300 |
| A.4 | Implementational strategy | 300 |
| A.5 | Summary of DETE's code and CM-2 usage | 301 |
| | Appendix B: *Lisp code for individual modules..... | 304 |

| | | |
|--------------------|--|------------|
| B.1 | Visual Feature Extractors..... | 304 |
| B.1.1 | Shape feature extractor..... | 304 |
| B.1.2 | Size Feature Extractor..... | 305 |
| B.1.3 | Location Feature Extractor | 306 |
| B.1.4 | Motion Feature Extractor | 306 |
| B.2 | Processing of visual features..... | 307 |
| B.3 | Simulator of blob's motions | 307 |
| B.3.1 | Generation of objects in the visual screen | 307 |
| B.3.2 | Moving Objects in the Visual Screen..... | 309 |
| B.3.2 | Generation of several objects on the Visual Screen | 312 |
| B.4 | The KATAMIC sequential associative memory | 313 |
| B.5 | Encoding & decoding of verbal I/O..... | 324 |
| B.5.1 | Word Encoding Mechanism..... | 324 |
| B.5.2 | Verbal Activity Decoder | 326 |
| Appendix C: | *Lisp code selected Experiments..... | 329 |
| C.1 | KATAMIC experiments | 329 |
| C.1.1 | Sequences used in the KATAMIC experiments | 329 |
| C.1.2 | Examples of KATAMIC experiments..... | 330 |
| C.2 | Experiments with DETE..... | 332 |
| C.2.1 | Examples of experiments with DETE..... | 332 |
| C.2.2 | I/O from experiments with DETE | 334 |
| Appendix D: | Monitoring the network's behavior..... | 337 |
| Appendix E: | Neural Nets for Procedural Modules..... | 339 |
| E.1 | A neural oscillator | 339 |
| E.2 | XOR network..... | 340 |

LIST OF FIGURES

| FIGURE | PAGE |
|---|------|
| Figure 1.1: DE TE’s Visual Screen | 2 |
| Figure 1.2: PGLA task 1: learning word meanings | 3 |
| Figure 1.3: PGLA task 2: learning word meanings | 3 |
| Figure 1.4: PGLA task 3: learning word meanings | 4 |
| Figure 1.5: PGLA task 4: learning word meanings | 4 |
| Figure 1.6: Generalizations | 6 |
| Figure 1.7: Homonyms..... | 7 |
| Figure 1.8: Synonyms..... | 7 |
| Figure 1.9: Word order..... | 10 |
| Figure 2.1: Tasks: Verbalization..... | 21 |
| Figure 2.2: Tasks: Imaging | 22 |
| Figure 2.3: Tasks: Motor actions..... | 23 |
| Figure 2.4: Block diagram of DE TE | 25 |
| Figure 2.5: Functions of DE TE’s retina | 25 |
| Figure 2.6: Retinal control | 26 |
| Figure 2.7: Block diagram of DE TE’s memory | 29 |
| Figure 2.8: The Blobs World | 31 |
| Figure 3.1: DE TE’s retinal structure | 37 |
| Figure 3.2: Retinal output | 38 |
| Figure 3.3: Shape Feature Plane (SFP)..... | 41 |
| Figure 3.4: siZe Feature Plane (ZFP)..... | 42 |
| Figure 3.5: Color Feature Plane (CFP)..... | 43 |
| Figure 3.6: Location Feature Plane (LFP)..... | 46 |
| Figure 3.7: Motion Feature Plane (MFP)..... | 48 |
| Figure 4.1: Formant frequencies of English vowels | 50 |
| Figure 4.2: Formant transitions..... | 51 |
| Figure 4.3: Gra-phonemes | 53 |
| Figure 4.4: Gra-phonemic encoding of a sentence | 54 |
| Figure 4.5: Decoding gra-phonemes -- verbal output..... | 56 |
| Figure 6.1: Motor interactions | 61 |

| | | |
|---------------|---|-----|
| Figure 6.2: | FINGER Position Plane (FPP)..... | 63 |
| Figure 6.3: | FINGER Motion Plane (FMP) | 64 |
| Figure 6.4: | EYE Diameter Plane (EDP) | 65 |
| Figure 7.1: | Input Segmentation Mechanism (ISM) | 69 |
| Figure 7.2: | Selective Attention Mechanism (SAM) | 70 |
| Figure 8.1: | Block diagram of the KATAMIC model..... | 74 |
| Figure 8.2: | Real & artificial neurons..... | 75 |
| Figure 8.3: | Predictron | 77 |
| Figure 8.4: | Recognitron & Bi-stable switch (BSS) | 79 |
| Figure 8.5: | Canonic circuit of the KATAMIC model | 80 |
| Figure 8.6: | The KATAMIC model..... | 81 |
| Figure 8.7.1: | KATAMIC dynamics: Plate 1..... | 91 |
| Figure 8.7.2: | KATAMIC dynamics: Plate 2..... | 93 |
| Figure 8.7.3: | KATAMIC dynamics: Plate 3..... | 95 |
| Figure 8.7.4: | KATAMIC dynamics: Plate 4..... | 97 |
| Figure 8.7.5: | KATAMIC dynamics: Plate 5..... | 99 |
| Figure 8.8: | Signal flow in the KATAMIC model | 101 |
| Figure 8.9a: | KATAMIC performance as a function of T_s & T_t (match/goal) | 103 |
| Figure 8.9b: | KATAMIC performance as a function of T_s & T_t (spur/goal)..... | 104 |
| Figure 8.10: | Various 1-bit-densities in different experiments | 105 |
| Figure 8.11: | Various 1-bit-densities in a single experiment | 107 |
| Figure 8.12: | Noise tolerance -- learning target sequences | 108 |
| Figure 8.13: | Noise tolerance -- processing of noisy sequences..... | 109 |
| Figure 8.14: | Learning branching sequences -- consecutive repetitions..... | 110 |
| Figure 8.15: | Learning branching sequences -- priming effects (100 reps)..... | 112 |
| Figure 8.16: | Learning branching sequences -- priming effects (1,000 reps) | 113 |
| Figure 8.17: | Memory capacity as a function of sequence length..... | 116 |
| Figure 9.1: | Taxonomy of DETE's memory | 120 |
| Figure 9.2: | Relation between DM & STM in DETE | 124 |
| Figure 9.3: | Block diagram of DETE's MSPM..... | 126 |
| Figure 9.4: | Neural circuitry of the MSPM | 127 |
| Figure 9.5: | Learning word order | 130 |
| Figure 9.6: | Temporal Memory (TM) -- connectivity and function | 134 |
| Figure 10.1: | Distribution of seed-DCps in SFM..... | 139 |

| | | |
|---------------|--|-----|
| Figure 10.2: | Details of the seed-DCps distribution in the SFM | 140 |
| Figure 10.3: | Distribution of seed-DCps in LFM..... | 141 |
| Figure 10.4: | Details of the seed-DCps distribution in the LFM | 142 |
| Figure 10.5: | Distribution of seed-DCps in MFM..... | 144 |
| Figure 10.6: | Distribution of seed-DCps in VM | 145 |
| Figure 10.7: | Inter-modular connectivity | 148 |
| Figure 10.8: | Detailed view of DETE's memory organization | 148 |
| Figure 11.1: | Learning words for objects & features..... | 153 |
| Figure 11.2: | Learning the meanings of "moves" & "stands"..... | 157 |
| Figure 11.3: | Learning about different movements..... | 160 |
| Figure 11.4: | Learning the meaning of "bounces"..... | 162 |
| Figure 11.5: | Verbal generalization..... | 164 |
| Figure 11.6: | Learning about size relations..... | 171 |
| Figure 11.7: | Description of spatial relations | 174 |
| Figure 11.8: | Descriptions of motion relations..... | 178 |
| Figure 11.9: | Learning present tense | 182 |
| Figure 11.10: | Imagining present events | 183 |
| Figure 11.11: | Learning future tense..... | 185 |
| Figure 11.12: | Imagining future events..... | 186 |
| Figure 11.13: | Learning past tense..... | 187 |
| Figure 11.14: | Imagining past events..... | 188 |
| Figure 11.15: | Learning curve of the word "stands"..... | 194 |
| Figure 11.16: | Example of a theoretical learning curve..... | 195 |
| Figure 12.1: | Kohonen's feature map..... | 205 |
| Figure 12.2: | Grossberg's avalanche model..... | 208 |
| Figure 12.3: | Jordan's network..... | 209 |
| Figure 12.4: | Kanerva's sequential SDM..... | 210 |
| Figure 12.5: | Elman's network | 212 |
| Figure 12.6: | Learning a single sequence of length 10..... | 216 |
| Figure 12.7: | Learning 10 random sequences of length 10..... | 217 |
| Figure 12.8: | Learning 10 correlated sequences of length 10..... | 218 |
| Figure 12.9: | Architecture of the TDNN..... | 220 |
| Figure 12.10: | The Neocognitron | 221 |
| Figure 12.11: | Crick's "Searchlight of attention" model | 222 |
| Figure 12.12: | Darwin III..... | 224 |

| | | |
|---------------|--|-----|
| Figure 13.1: | Functional architecture of the visual pathway..... | 232 |
| Figure 13.2: | Language-related areas in the brain..... | 239 |
| Figure 13.3: | Block diagram of language processing in the brain | 243 |
| Figure 13.4: | Coltheart's model..... | 244 |
| Figure 13.5: | Saccadic motions of the eye..... | 248 |
| Figure 13.6: | Brain areas involved in visual attention..... | 250 |
| Figure 13.7: | Attention-related areas in the cerebral cortex | 251 |
| Figure 13.8: | Attention-related areas in the brain stem..... | 252 |
| Figure 13.9: | Neural circuitry of the cerebellum..... | 254 |
| Figure 13.10: | Cerebellar connections (I/O) in the brain | 255 |
| Figure 14.1: | Learning pronoun resolution -- the meaning of "it" | 275 |
| Figure E.1: | Neural oscillator..... | 339 |
| Figure E.2: | XOR network..... | 340 |

LIST OF TABLES

| TABLE | PAGE |
|--|------|
| Table 2.1: Scope of the Blobs world..... | 31 |
| Table 2.2: A syntactic specification of FIRLAN..... | 32 |
| Table 2.3: A syntactic specification of SECLAN..... | 33 |
| Table 3.1: Mapping of shapes in the SFP | 40 |
| Table 4.1: Frequency range/loc mapping in the gra-phonemic representation | 52 |
| Table 4.2: Representations of the 26 gra-phonemes | 52 |
| Table 5.1: DETE's temporal hierarchy..... | 58 |
| Table 8.1: Typical values of KATAMIC parameters and variables..... | 82 |
| Table 9.1: Temporal XOR function | 128 |
| Table 11.1: Results of learning circle square triangle | 155 |
| Table 11.2: Results of learning W:moves & W:stands | 158 |
| Table 11.3: Results of learning W:moves_horizontally vertically diagonally | 159 |
| Table 11.4: Results of learning W:bounces..... | 163 |
| Table 11.5: Visual to verbal generalization | 166 |
| Table 11.6: Question answering -- slot-value retrieval | 168 |
| Table 11.7: Results of learning about size relations | 172 |
| Table 11.8: Results of learning closer / farther relations | 177 |
| Table 11.8: Verbal tenses and their S/E/R representations..... | 180 |
| Table 11.9: Results of learning a homonym..... | 190 |
| Table 12.1: Implementation details of the KATAMIC & SRN models..... | 214 |
| Table 13.1: Language dysfunctions | 242 |
| Table 14.1: Verb tenses of additional verbs | 273 |
| Table E.1: Computation of XOR | 341 |

ACKNOWLEDGMENTS

This thesis is dedicated to my children Katarina and Michael who are continuous sources of inspiration and joy for me.

First and foremost I wish to express my heartfelt appreciation to my academic advisor Professor Michael G. Dyer for introducing me to the fields of Artificial Intelligence and Natural Language Processing, for his friendship, financial and moral support, and unlimited research enthusiasm. He went through numerous drafts of my dissertation and helped me clarify the architecture, scope and limitations of the model.

The rest of my dissertation committee members: Professors Eric Halgren, David Jefferson, Arnold Scheibel, and Jacques J. Vidal deserve my deepest gratitude for their enthusiastic support of my work and for the numerous helpful comments and suggestions. In particular, I thank Professor Eric Halgren who helped clarify the neurophysiological basis of my model, but most of all for his friendship and willingness to serve for a second time on my Ph.D. committee; Professor Jacques J. Vidal who introduced me to the field of neural networks.

In addition, I would like to thank a number of past and present Airheads -- the nickname for graduate students at the AI lab, for creating an intellectually stimulating environment and for their help at various stages of my training in computer science. Among these people are Sergio Alvarado, Stephanie August, Charlie Dolan, Ric Feifer, Maria Fuenmayor, Michael Gasser, Seth Goldman, Jack Hodges, Adam King, Trent Lange, Gunbae Lee, Stuart Levine, Risto Miikkulainen, Eric Mueller, Alex Quilici, John Reeves, Ron Sumida, Scott Turner, Alan Wang, Greg Werner, and Uri Zernik. This environment was also significantly enriched by occasional visitors and users of the AI Lab including Edward Hoenkamp, Hideo Shimazu, Rob Collins and Craig Morioka.

Special thanks to Professor Walter Reed for bringing the joy of fruitful collaborative work to life, for his friendship, and numerous stimulating intellectual discussions.

My appreciation goes also to Professors Don Perlis and James Reggia from the University of Maryland, College Park; Mike Mozer and Paul Smolensky from the University of Colorado, Boulder; Don Alton and Robert Baron from the University of Iowa, Iowa City; George Lakoff and Terry Regier from ICSI at the University of Berkeley, and Steve Levinson from AT&T Bell Labs whose critical comments at various stages of DETE's development helped shape the project.

I am also indebted to Verra Morgan, Valerie Aylett and Anna Gibbons for making my navigation through with the academic administration less bumpy.

This research has been supported in part by an interdisciplinary research grant from the W. M. Keck Foundation to Professor Michael G. Dyer. The CM-2 Connection Machine, on which the model is implemented, was acquired through NSF equipment centers grant #BBS-87-14206 and maintained through both the Keck Foundation grant and NSF grant #DIR-90-24251, and managed by the UCLA Cognitive Science Research Program.

VITA

June 18, 1953 Born, Sofia, Bulgaria.

EDUCATION

- 1972 - 73 **Undergraduate**, Major: physics, Department of Physics, Sofia State University, Sofia, Bulgaria.
- 1974 - 79 **MS (Diplom Engineer)**, Major: Physical Electronics. Department of Nuclear Sci. and Engineering, CVUT, Prague, Czechoslovakia
- 1983 - 89 **Ph.D., Neuroscience**. Interdept. Neuroscience Ph.D. Program, Brain Research Institute, University of California, Los Angeles.
- 1986 - 91 **Ph.D., Computer Science**. Major: Artificial Intelligence. Computer Science Department, University of California, Los Angeles.

EXPERIENCE

- 1976 - 79 **Research Assistant**. Electrophysiology Lab, Brain Research Institute, Czechoslovakian Academy of Sciences, Prague, Czechoslovakia.
- 1979 - 83 **Service engineer and senior programmer**. Gas-Chromatography / Mass-Spectrometry Lab, Institute of Organic Chemistry, Bulgarian Acad. Sci.
- 1980 **Visiting Research Assistant**. Mass Spectrometry Lab, Department of Structural Chemistry, Ruhr University in Bochum, West Germany
- 1983 - 85 **Research Assistant and programmer**. Neurophysiology Lab, Neuropsychiatric Institute, University of California, Los Angeles.
- 1985 - 89 **EEG technologist**. Sleep Disorders Clinic, Neuropsychiatric Institute, University of California, Los Angeles.
- 1985 - 89 **Research Assistant**. PET Facility & Laboratory for Cognitive Neuroscience, VA Wadsworth Medical Center, West Los Angeles.
- 1986 - 90 **Research Assistant**. Artificial Intelligence Lab, Computer Science Department, University of California, Los Angeles.
- 1990 - 91 **Research Assistant Professor**. Department of Computer Science, University of Southern California, Los Angeles.
- 1991 **Instructor**. Department of Biomedical Engineering, University of Southern California, Los Angeles.
- 1991 - present **Senior Development Engineer**. Division of Neurosurgery, School of Medicine, University of California, Los Angeles.

PUBLICATIONS AND PRESENTATIONS

- Nenov, V. (1979). *An electronic system for recording and processing of some EMG manifestations of speech and thought*. Master's Thesis (Diplom Engineer), Department of Nuclear Science & Physical Engineering at CVUT and Brain Research Institute at Czechoslovakian Academy of Sciences, Prague, Czechoslovakia.
- Christov, V., Demirev, P., Mollova, N., and Nenov, V. (1981). Quantitative analysis of thalicarpine in *Thalictrum Minus L.* by mass fragmentography. In *Proceedings of the First International Conference on Biotechnology, Varna*. (pp. 367-371). Bulgarian Academy of Sciences, Bulgaria.
- Nenov, V., Mollova, N., Demirev, P., Bankova, V., and Popov, S. (1983). Computer assisted mass-spectral investigation of Flavonoid mixtures. *International Journal of Mass Spectroscopy and Ion Physics*, **47**, 333-336.
- Popov, S., Demirev, P., Nenov, V., Sidjimov, A., Marekov, N., Georgieva, R., and Ognyanova, V. (1983). Studies of Triquinol pharmacokinetics by mass fragmento-graphy. *International Journal of Mass Spectroscopy and Ion Physics*, **48**, 105-108.
- Woody, C.D., Oomura, Y., Gruen, E., Miyake, J., and Nenov, V. (1984). Attempts to rapidly condition increased activity to click in single cortical neurons of awake cats using glabella tap and iontophoretically applied glutamate. *Abstracts of the 14th Annual Meeting of the Society for Neuroscience*, (Abstract).
- Woody, C.D., Nenov, V., Gruen, E., Donley, P., Vivian, M., and Holmes, W. (1985). A voltage-dependent, 4-aminopyridine sensitive, outward current studied in vivo in cortical neurons of awake cats by voltage squeeze techniques. *Abstracts of the 15th Annual Meeting of the Society for Neuroscience*, 955 (Abstract).
- Hirano, T., Woody, C., Birt, D., Aou, S., Miyake, J., and Nenov, V. (1987). Pavlovian conditioning of discriminatively elicited eyeblink responses with short onset latency attributable to lengthened interstimulus intervals. *Brain Research*, **400**, 171-175.
- Nenov, V., Berka, C., Kamath, S., Halgren, E., Smith, M., and Mandelkern, M. (1987). Localization of metabolic and electrophysiologic activation during verbal tasks. *Abstracts of the 17th Annual Meeting of the Society for Neuroscience*, (Abstract).
- Nenov, V. and Dyer, M. (1987). *MOZAK: Computer comprehension of neuroscience abstracts*. Technical Report UCLA-AI-87-2, Department of Computer Science, University of California, Los Angeles, CA.
- Nenov, V.I., Feldstein, P., Halgren, E., Deutch, R., Mandelkern, M.A., Ropchan, J.R., and Bland, W.H. (1987). Correlations between local brain hypometabolism and neuro-

- psychological deficits in epileptic patients. *Abstracts of the Second World Congress of Neuroscience (IBRO), Budapest*, (Abstract).
- Nenov, V.I. and Dyer, M.G. (1988). *DETE: Connectionist/symbolic model of visual and verbal associations*. Technical Report UCLA-AI-88-6, Department of Computer Science, University of California, Los Angeles, CA.
- Schwartz, B., Nenov, V., Halgren, E., Rand, W., Delgado Esqueta, A.V., Mandelkern, M., Ropchan, J., and Bland, W. (1988). Pre and postoperative 18-FDG PET scans in temporal lobe epilepsy. In *Proceedings of the American Epilepsy Society Meeting, San Francisco, CA*.
- Nenov, V., Halgren, E., Mandelkern, M., Ropchan, J., and Bland, W. (1989). Metabolic and electrophysiologic manifestations of recognition memory in humans. In *Proceedings of 36th Annual Meeting of the Society of Nuclear Medicine, St. Louis*.
- Nenov, V.I. (1989). *Metabolic and Electrophysiologic Manifestations of Recognition Memory in Humans*. Ph.D. Thesis, Interdepartmental Neuroscience Program, University of California, Los Angeles, CA.
- Woody, C.D., Baranyi, A., Szenté, M.B., Gruen, E., Holmes, W., Nenov, V., and Strecker, G.J. (1989). An aminopyridine-sensitive, early outward current recorded in vivo in neurons of the precruciate cortex of cats using single-electrode voltage-clamp techniques. *Brain Research*, **480**, 72-81.
- Nenov, V.I. (1990). The cerebellar cortex as a sequential associative memory: A novel structural/functional interpretation. In *Analysis and Modeling of Neural Systems*. Kluwer Academic, Boston, MA. pp. 339-404.
- Nenov, V.I. (1990). Rapid learning of pattern sequences: A novel network model. In *Proceedings of the International Neural Networks Conference (INNC-90), Paris*.
- Nenov, V.I., Read, W., Halgren, E., and Dyer, M.G. (1990). The effects of threshold modulation on recall and recognition in a sparse auto-associative memory: implications for hippocampal physiology. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-90), Washington D.C.* pp. 225-235.
- Read, W., Nenov, V.I., and Halgren, E. (1990). Inhibition-controlled retrieval by an autoassociative model of hippocampal area CA3. *Hippocampus*, (to appear).
- Nenov, V.I. and Vidal, J.J. (1991). Modeling the cerebellar cortex as a sequential associative memory. In *Proceedings of 1991 Mathematica Conference, San Francisco*.
- Nenov, V.I., Halgren, E., Smith, M.E., Badier, J.-M., Ropchan, J., Bland, W.H., and Mandelkern, M. (1991). Localized brain metabolic response correlated with potentials evoked by words. *Behavioural Brain Research*, **44**, 101-104.

ABSTRACT OF THE DISSERTATION

**Perceptually Grounded Language Acquisition:
A Neural/Procedural Hybrid Model**

by

Valeriy Iliev Nenov, Ph.D.

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 1991

Professor Michael G. Dyer, Chair

Humans acquire natural languages such as English, Spanish or Japanese while immersed in an environment rich with visual, auditory and other sensory stimuli. It has been hypothesized that the meaning of words for some basic concepts such as shape, size, motion, and location of objects in space are grounded in perceptual experiences, whereas more abstract conceptualizations are constructed as metaphorical extensions of the primitive concepts of objects and events. This thesis describes the current status of an on-going, large-scale research project called DETE* whose objective is to explore how language semantics maps to sensory experiences -- a question known as the "Symbol Grounding Problem". DETE is a modular (neural/procedural) hybrid system whose design was inspired by the known structure/function of brain areas involved with vision, language processing, attention, and memory. It was developed as a test bed for computer simulations in language acquisition and is currently implemented in *LISP on the CM-2 Connection Machine. It accepts three kinds of input: (1) Visual -- a continuous sequence of visual scenes showing the behavior of simple 2D shaped objects in a square visual field; (2) Verbal -- occasional streams of English sentences describing the visual scenes; (3) Motor -- sequences of motor commands which instruct DETE to shift and/or resize its focus of attention (EYE), and to interact with objects by moving a simple simulated effector (FINGER). The output of the system includes language generation (word sequences), visual imagination, and simple motor performance (shift of visual attention and/or movement of the FINGER). Currently, the input-processing modules are

* DETE (pronounced "deetee") stands for child in Bulgarian -- the author's native language

procedural. These procedures form distributed representations of the visual and verbal inputs. The visual inputs are represented as localized firing patterns over a set of feature maps, which encode information concerning the shape, size, location, speed, and direction of motion of a few simple 2D objects (e.g., circular or square “blobs”). The representations of the verbal inputs capture some of the acoustical features of speech.

DETE’s ability to associate concurrent sequences of visual and verbal inputs is based on a novel neural network architecture called the KATAMIC sequential associative memory. This neural network model integrates learning with recognition, and with cue based recall of binary pattern sequences. It has a number of highly desirable features including: (1) *Extremely rapid learning*: Only a few exposures (on average 4 to 6) to a particular sequence are sufficient for learning. (2) *Flexible memory capacity*: Multiple sequences can be stored in the network, with a memory/processor ratio comparable to, if not better than that of other neural net, PDP or connectionist models. (3) *Sequence completion*: A short cue can retrieve the complete sequence. (4) *Sequence recognition*: A built-in mechanism allows sequence recognition on a pattern-by-pattern basis, which is used internally for switching from learning to performance mode. (5) *Fault and noise tolerance*: Missing elements (bits within patterns or whole patterns) within a reasonable amount (30% of the number of 1-bits) can be tolerated. (6) *Integrated processing*: The model is capable of concurrent learning, recognition, and recall of sequences, a significant improvement over most previously proposed models that focus only on specific aspects of processing at a time, e.g., the PDP class of models.

In a number of computer simulations DETE has proven its ability to acquire meanings of words in a basic lexicon for its task domain, comprehend simple syntactic constructs (word order, morphological inflections), comprehend verbal descriptions of spatial and temporal relations between objects, and answer questions. Currently DETE is being tested on subsets of English, Spanish, and Japanese. These languages differ greatly in inflectional properties, word order, syntactic structure, and in how they categorize or “carve up” perceptual reality.

In grounding its primitive symbols in sensory categories DETE differs from pure, autonomous, top-down symbol systems in which the primitive symbols are merely arbitrary, undefined atomic tokens. We believe that through computational modeling and simulations this research broadens the bridge between the current understanding of higher cognitive processes -- language, memory, attention, and vision on the one hand, and their physical embodiment in the neural systems of the human brain.

PART I

An Overall View

Part one of this dissertation provides a general overview. First, it introduces the task handled by the model -- Perceptually Grounded Language Acquisition (PGLA). This task involves learning to understand the meaning of relatively simple language constructs by association of simple visual scenes (involving moving objects) with accompanying short verbal descriptions of these scenes. Following the task description I introduce the model, called DE TE*, in general terms and outline its implementation. DE TE is a modular procedural / connectionist system capable of performing the PGLA task. A demonstration of the model in action is used in part I to illustrate some of its capabilities. The motivation and goals of this research project are also discussed in the light of related work on symbolic and neural network based systems for Natural Language Processing (NLP).

* DE TE stands for "child" in Bulgarian

1 INTRODUCTION

1.1 Task: Perceptually Grounded Language Acquisition

Imagine you are a child and you are about to start learning your FIRst LANGUAGE, FIRLAN, in the following way. You are placed in front of a Visual Screen with headphones on. You see objects moving on the screen while you hear your teacher's voice in the headphones. You are able to look at different locations on the screen by moving your eyes. You can also zoom in and out by accommodating your eyes so that you can see the whole screen or only a part of it in which one or more objects appear. There are two icons on the screen (Figure 1.1): a FINGER which the teacher or you can move around through a joystick, and an EYE -- your Visual Field (VF) shown as a shaded circle. The VF has the same dual type of control. The location of the VF on the screen represents the location at which you are looking. The diameter of the VF can vary and you can see things only through this aperture -- your window to the world (Visual Screen).

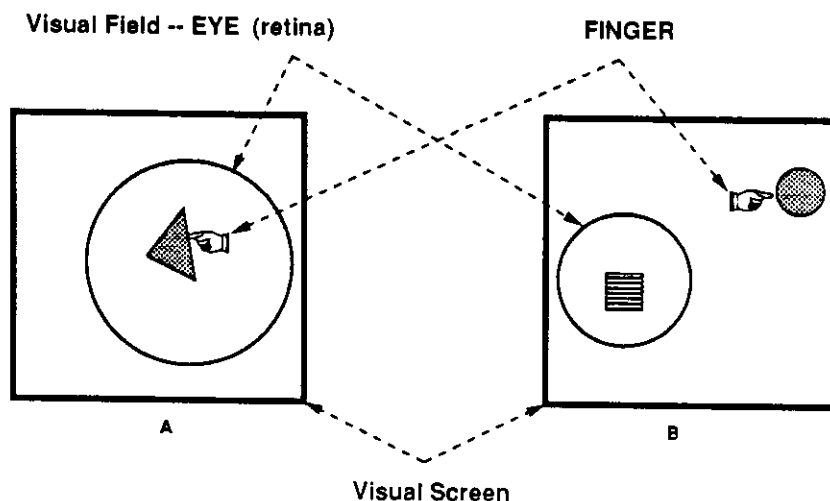


Figure 1.1: DETE's Visual Screen

Two different examples showing the position of the EYE (VF) & the FINGER on the Visual Screen (VS): (A) The FINGER is pushing a triangle which is within the VF. (B) The FINGER is pushing a circle which is out of the VF. There is a small square within the Visual Screen.

Your instruction begins by looking at a series of simple objects and hearing short (one or few words) descriptions of what you see. First, your teacher presents you with a number of circles with variable diameters, colors and locations and you hear in the headphones: "FOO" anytime a new circle appears (Figure 1.2). You begin to hypothesize that FOO means "circle".

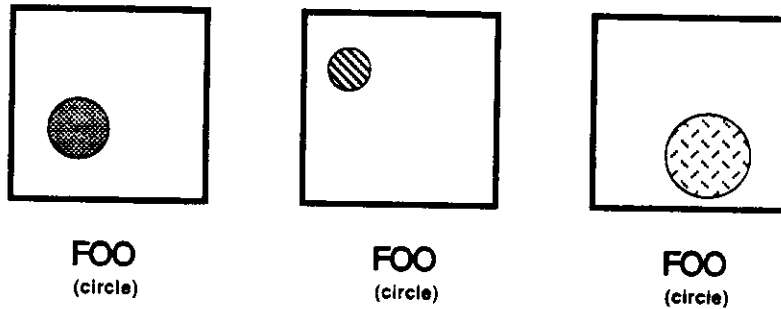


Figure 1.2: PGLA task 1: learning word meanings

In a sequence of visual frames you are shown circles with different diameters, colors* and locations while you hear FOO. You begin to hypothesize that FOO means “circle”.

But then you see a triangle and hear “FOO” again. Then a square and again “FOO”. Perhaps FOO means something like “object” (Figure 1.3).

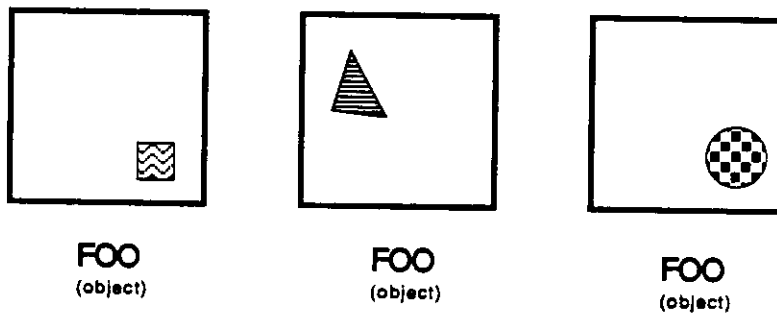
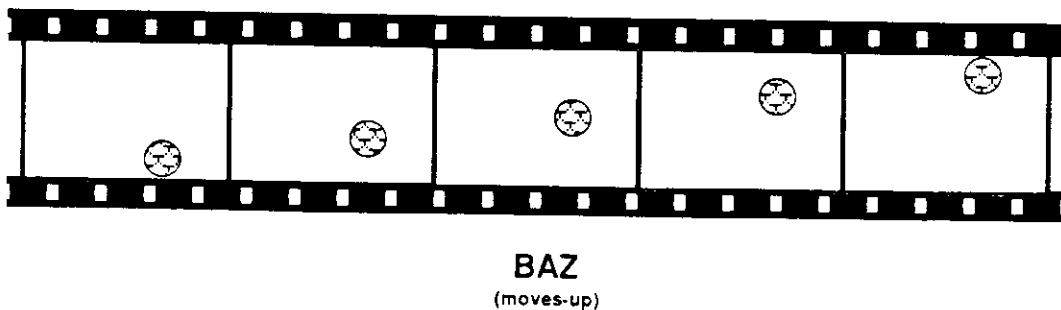


Figure 1.3: PGLA task 2: learning word meanings

You are shown a triangle and later a square and in both cases you hear FOO. Perhaps FOO means something like “object”.

The triangle begins to move up. You hear: “FOO BAZ”. “BAZ” could mean “moves”, “moving”, or it could mean “up” or “moving up”. If you see several objects moving up and hear always “BAZ” but never hear “BAZ” for moving objects in other directions, then it probably does not mean “moving”. It probably means “moving up” (Figure 1.4).



* Different colors are indicated in all figures by different texture fill-in patterns.

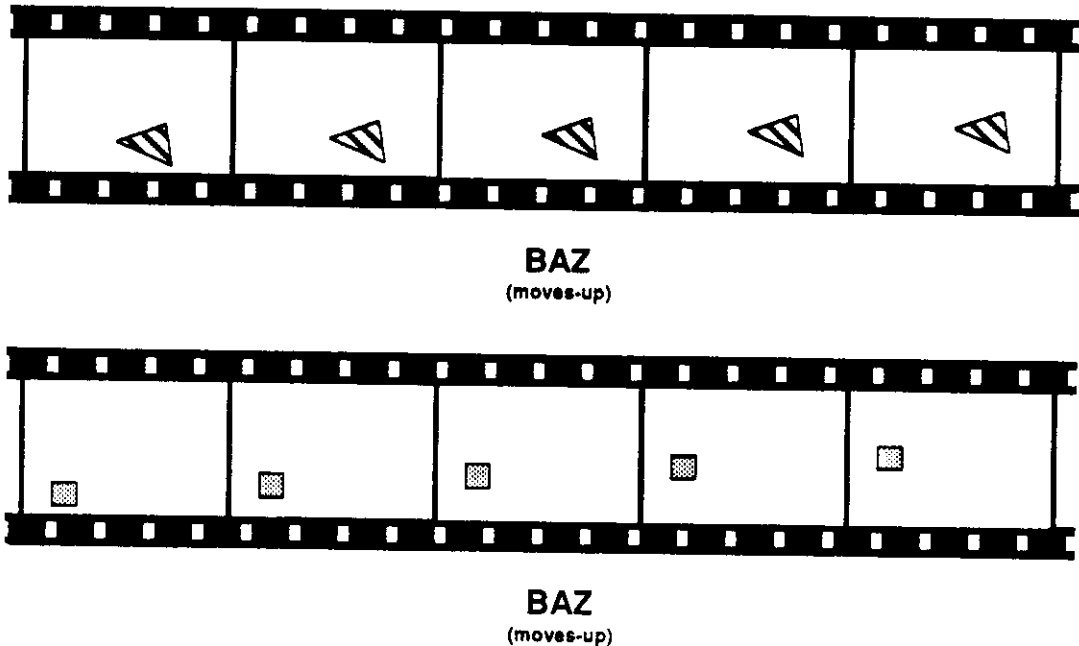


Figure 1.4: PGLA task 3: learning word meanings

Three sequences of visual frames showing successive states of a triangle, a circle, and a square. All of these objects are moving up.

If objects move in different directions when you hear, “BLITZ”, then it probably means something like “moves” (Figure 1.5), but it could also reflect some characteristic of the speed of motion, e.g., fast.

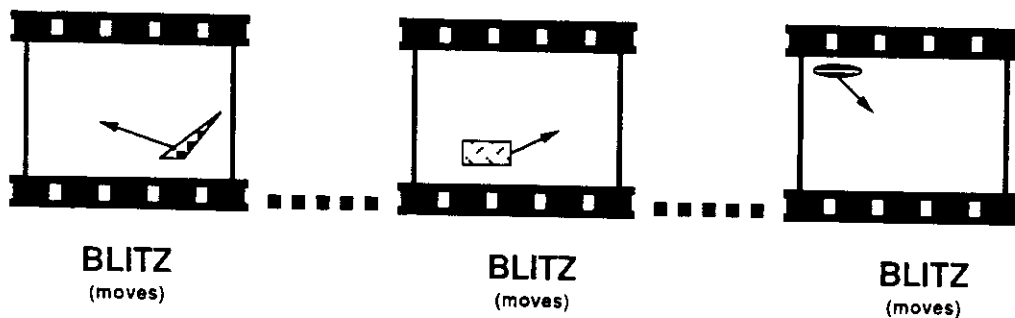


Figure 1.5: PGLA task 4: learning word meanings

Examples of objects moving in different directions. (The arrows do not appear on the screen. They are an abbreviation for a sequence of images and indicate the direction of motion.) You hear BLITZ as a description of each situation. BLITZ is interpreted as “moves”.

Later, in a more advanced stage of your training you are asked to describe what you see on the screen by simply having (stationary or moving) objects pointed to by the FINGER on the screen, or via a verbal request. The above task is termed: **Perceptually Grounded Language Acquisition (PGLA)** task.

A person (or system) capable of learning FIRLAN in the way described above must be able to perform a number of cognitive operations. These operations fall into three categories: (1) *Semantic* -- Operations that are semantic in nature include generalization/specialization, disambiguation, handling of synonyms, reference, and modifiers. (2) *Pragmatic* -- Such operations include temporal inference, attention, memory, action and their interactions. (3) *Syntactic* -- This category of operations includes the handling of morphology, word order, conjuncts, relative clauses, ellipses, and voice. Also, a system that can do the PGLA task must be able to learn a wide range of different languages, with different syntax, semantics and pragmatics; not just one single language.

1.1.1 Semantics

By “semantics” here I mean extension of objects and composition of meanings into larger conceptualizations.

• Generalization / specialization:

An essential characteristic of our language skill is the ability to generalize. There are different levels of generalization. A prerequisite for generalization is the ability to associate names with objects. Being able to use the same name for similar objects (e.g., “BOO” for all round objects) is a simple type of generalization. Having seen different objects (e.g., balls, triangles, and squares) and having learned their names, if later the same objects are referred to by one and the same name, e.g., “FOO”, then one should be able to infer that FOO is something like “object”. If a new but similar-looking object is introduced, the system should then be able call it FOO. This is a higher level of generalization. An example of possible generalizations (classifications) in the space of shapes is given in Figure 1.6.

Specialization is the opposite of generalization. If there are several objects that belong to the same category (e.g., a square and a triangle which are both polygons), in order to distinguish between the individual objects we need to know the individual names of the subcategories. Then, if we have a square and a triangle and want to ask someone to point to the triangle, we do not have to use a description such as “Point to the polygon that has three vertices”. Instead we can simply say “Point to the triangle”.

When the given language does not provide individual names for the subcategories and we need to distinguish between several instances of the same type, then we have to find the differences between the secondary features (the primary features define the membership in the particular class), and generate a composite verbal description. This is called specification. For instance, if there are two balls (e.g., red and blue) and we want somebody to bring to us a specific one of them, then we need to specify which one, e.g., “Bring the red ball”.

There is an important theoretical and practical issue here. Different languages carve up differently the same reality that people perceive through their senses. For instance, English distinguishes “crackers” and “cookies”, while Spanish has only the word “galleta” which indicates both crackers and cookies.

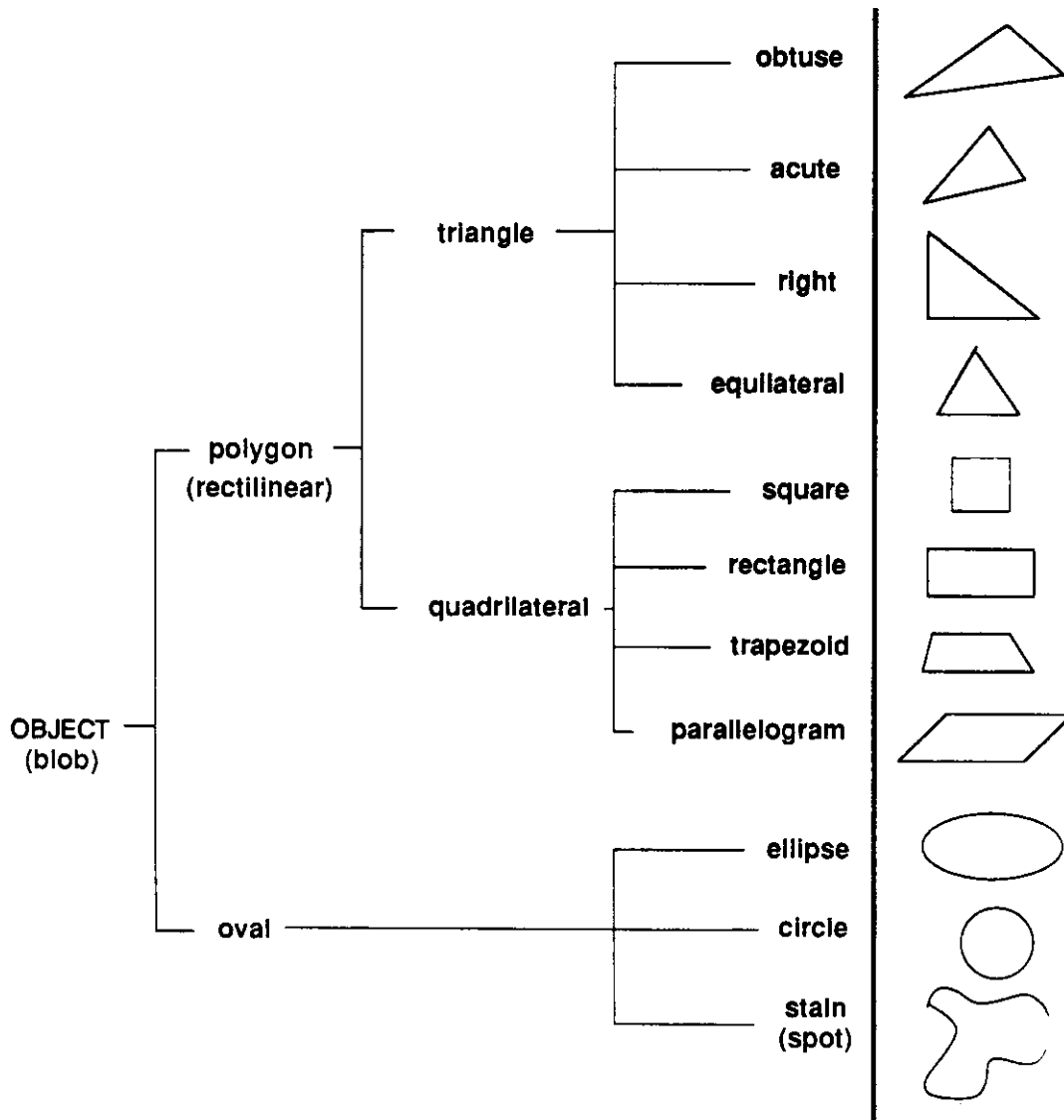


Figure 1.6: Generalizations

Some examples of the levels of generalization within the space of simple shapes. A hierarchical categorical structure formed by the verbal label (name) of the categories is presented to the left. Instances of objects belonging to the different categories are presented to the right. It is important to notice that this categorical “carving up” of the various visually perceived shapes is characteristic of the English language and is not necessarily the same in other languages.

• **Ambiguity (Homonyms):**

Many natural languages, (e.g., English, Spanish) often use one and the same word to refer to two or more things. For instance, in English the word “pot” has at least two meanings: (1) pot = cooking container (e.g., wash pot), (2) pot = marihuana (e.g., smoke pot). Word or phrase ambiguity can be also found in the domain of simple-shaped objects moving in a visual screen. For instance, the phrase “run over” can mean: (1) moving along a trajectory that is above some object

(Figure 1.7: A), or (2) hitting an object and continuing in the same direction of motion after the hit (Figure 1.7: B).

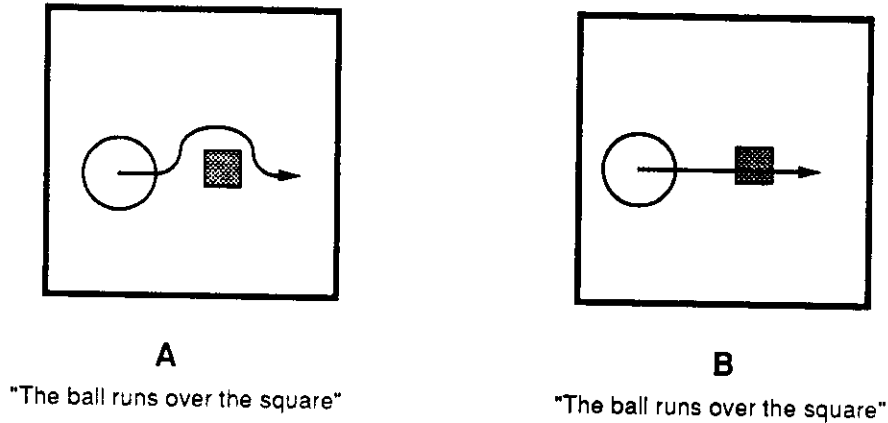


Figure 1.7: Homonyms

The sentence "The ball runs over the square" has two different meanings.

- **Synonyms:**

A common language task that children face early in life is the resolution of synonymous words. For instance, a rabbit is often called "bunny" and "hare"; a turtle is also referred to as "tortoise". Children must learn that different words can refer to the same object. Examples of synonyms in DETE's task domain are the words that refer to an object with circular shape, e.g., "circle", "ball", and "globe" (Figure 1.8).

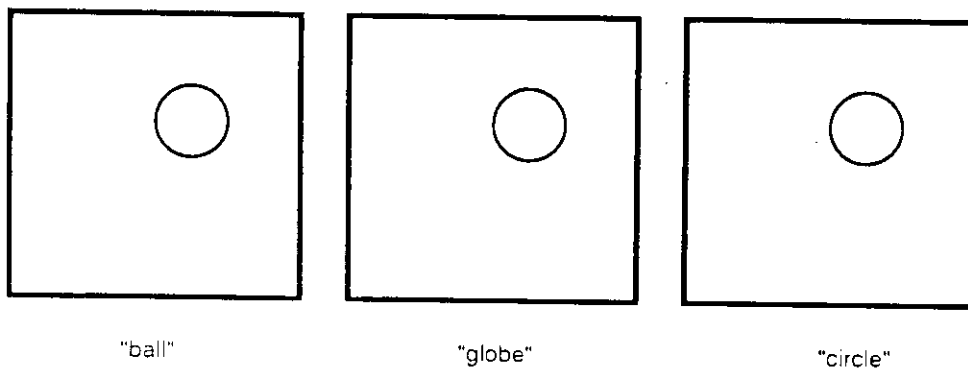


Figure 1.8: Synonyms

In English one can use several synonymous words to refer to the same object. For instance, "circle", "ball", and "globe" can be used to name an object with a circular shape

- **Temporal inference:**

You hear "BOO GLITCH" and see a ball repelled off the screen wall. You have already inferred that "BOO" means ball. Now you guess "GLITCH" means "bounce". In another situation you hear: "BOO WOO GLITCH" and the ball just moves toward the wall. A few moments later, when there

is no FIRLAN verbal input, the ball bounces. Now you must figure out that “WOO” refers to a future event like, for instance, the word “will”, as in “ball will bounce”.

- **Reference:**

You hear “FOO BAZ TI WOO GLITCH”. Assuming that you have learned the meanings of the individual words (FOO “object”, BAZ “moves”, WOO “will”, GLITCH “bounce”) except the word TI, can you figure out that TI is something like the pronoun “it” and refers to FOO?

- **Modifiers:**

You hear “GIB BOO” and see a large red ball. Knowing that “BOO” stands for ball, you figure out that “GIB” means something like “large” or “red”. You must see several large balls that are different colors before being able to factor color out and map GIB to size.

1.1.2 Pragmatics:

Pragmatics generally refers to the usage of language in social context. For example indexicals (“you” vs “me”), talking about the task, e.g., statements (“This is a red ball.”) vs questions (“Where is the red ball?”) vs commands (“Look at the red ball.”).

- **Attention, memory and interaction:**

There are two objects on the screen and you hear “PUM FOO ZI BAZO ... LOO TI” , which, let us assume, stands in English for “See the object that is moving ... push it” . To accomplish such a task requires that you already have learned the meanings of the individual words in terms of actions. For instance “PUM” means that you should search the visual world for a “FOO” (object) “ZI” (that) “BAZO” (moves) and if your search is successful, initiate the action “LOO” (push) on the object. In order for such a complex sequence of behaviors to be performed you need to have the ability to attend to different parts of the visual screen by moving your eyes and changing the size of your field of view. You also should be able to maintain a mental image of the target of your search in some type of a short-term memory until the task is completed.

- **Action:**

Performing actions upon the environment is an essential tool for acquiring experiences. To be able to perform actions such as pushing/pulling (dragging), hitting/blocking (stopping), or bouncing/catching objects in the environment one needs an effector -- a finger or a hand or something else. Another way of interacting with the environment is by means of selectively choosing the type and quantity of information which reaches our senses. This is done by controlling the state of our sensory organs (e.g., moving our eyes and focusing on different objects).

- **Memory:**

Memory is a critical prerequisite for a system to perform the PGLA task. Memory tasks include, for instance, the retrieval of the visual representation of a particular object in response to a verbal input. In other words, the ability to imagine a ball (i.e. bring to working memory the visual representation of a ball) when the word “ball” is heard. Whenever a sequence of learned words is given, then appropriate meaning of each word must be retrieved from memory.

- **Sensitivity to temporal dynamics of input:**

The duration of various chunks in the verbal input and the pauses between them, together with the position of stressed syllables in words (prosodic features) are critical for our comprehension. For

instance, the letter string “importantunderstanding” could be interpreted as “important understanding” or “import ant under standing”.

- **Ellipsis:**

Ellipsis is a linguistic phenomenon which occurs in a stream of sentences when a particular part of a given sentence (which can be considered to be a variation of the preceding sentence) is missing. If the given sentence were to be by itself it would not make sense without this part. Ellipsis can occur in a monolog or a dialogue. For instance: Question 1: “Where is the ball?” Answer 1: “In the middle.” Question 2: “And the triangle?” (must be understood as “Where is the triangle?”).

1.1.3 Syntax

The syntactic aspect of language concerns its constituent structure, e.g., morphology of individual words, word order, etc.

- **Morphology:**

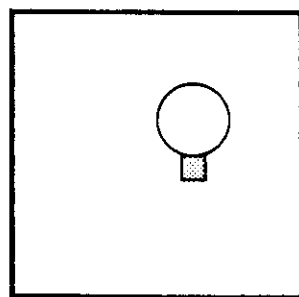
Natural languages often use suffixes or prefixes attached to the stem of a word to express new meanings. Different languages adopt various methods of modification. In some languages like Spanish, this is done by using suffixes, e.g., “pelota = ball”, “pelotoh = big ball”, “pelotita = small ball”, “rapido = fast”, “muy rapido = very fast”, “rapidisimo = extremely fast”. An example of the use of prefix in Spanish is: “bièn = good” vs “rebièn = very good”. Another Spanish example for inflected verbs is: “they walked” vs “andaron” -- here the “they” is encoded as a suffix and not a pronoun placed in front, as in English.

In the scenario set up above, if you hear “BOOL” and see a large ball after having learned that “BOO” stands for ball, you might want to “hypothesize” that the suffix -L in “BOOL” means “large”. In a similar situation, if you see a small ball and hear “BOOS”, and here you should be able to infer that -S stands for “small”. Then you should be able to generalize the suffix to other words, e.g., “FOOS” means “small object”.

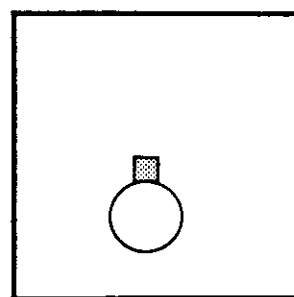
- **Word order:**

Word order varies between languages. For instance, in English we say “He did it” while a Spanish speaker would say “Lo hizo” (it he did).

To be able to understand the meaning of sentences one needs to learn how the word order in a sentence affects its meaning. For instance, “X on Y” is different from “Y on X” (Figure 1.9).



“ball on square”



“square on ball”

Figure 1.9: Word order

Two sentences “ball on square” and “square on ball”, which contain the same three words, have different meanings depending on their word order.

- **Conjuncts:**

To construct complex sentences, natural languages use conjuncts (e.g., *and*, *or*, *either*, *both*, etc.). For instance, in “The ball hit the square and the triangle.” both are hit, while in “The ball hit the square and the triangle did not.” the “and” groups actions instead of objects.

- **Relative clauses**

For instance, “The man hit the car” vs “The man (hit by the car) went to the hospital.” An example from our setup is: “The ball hit the triangle” vs “The ball (hit by the triangle) went up”.

1.2 DETE: A Neural Architecture

DETE is a modular connectionist system designed to perform the task of Perceptually Grounded Language Acquisition (PGLA). To perform this task DETE uses the interactions between several subsystems: visual, verbal, motor, memory, and attentional. DETE’s perceptual/motor modules are not intended to be neurally realistic models of the visual, the auditory, the attentional, or the motor systems. Instead DETE contains functional substitutes of these systems which produce outputs in which the information is encoded in a predefined, but distributed or quasi-distributed form. However, parts of DETE, and specifically the memory modules, are more neurally realistic than current parallel distributed processing (PDP) systems (McClelland et al., 1986). Our interest in perception & motor control is only to the extent that they serve as input to, and output from, the higher “cognitive” systems subserving language processing and acquisition.

As will be demonstrated in Chapter 11, DETE is able to perform several of the tasks described in the previous section (1.1) including: (1) *semantic*: generalization, disambiguation (homonyms), temporal inference (various verb tenses), and interpretation of modifiers; (2) *pragmatic*: attend to and act on objects in the visual screen, and learn (memorize) the meanings of words, and (3) *syntactic*: learn word order (simple syntactic rules), and morphological inflections of words.

1.2.1 Theoretical Issues

There are a number of theoretical issues which must be addressed in the process of constructing a system capable of performing the PGLA task.

(1) **Environment:**

- What are the essential characteristics of the environment (visual and verbal) which will allow a system, that is immersed in this environment, to develop language skills?

- Is there an optimal or preferred order of experiences needed for the learning of concepts, or can the system selectively pick up what it needs from the input, effectively creating its own learning protocol?

(2) **Innate Structure:**

- What innate capacities (in terms of available neural structures with their connectivity and function) are necessary for the emergence of complex behaviors such as language and perceptual reasoning?

(3) Modifiable Structure:

- How do dynamically modifiable associations of visual and verbal inputs aid in the formation of concepts and how are concepts represented in the system?

- How much and what can a system, that has such architecture and function, learn from its experiences?

(4) Interaction of visual/verbal modalities:

- What specifically visual and/or verbal structures/processes and interactions are needed to support various aspects (sub-tasks) of the PGLA task?

1.2.2 DE TE in action

In its current implementation DE TE is not an interactive system in the sense that a user cannot simply sit in front of the visual screen and chat with DE TE in real time. In practice, all of the experiments were run in a batch mode and the external user control was simulated. In other words, the training and testing sets were developed off-line and DE TE was left to run them overnight. Later, DE TE's performance was evaluated. However, if we imagine that the time-consuming computations could be compressed in time, then one could observe the following sequence of progressively more complex linguistic skills being acquired by DE TE.

(1) DE TE learns the meanings of words that name individual objects and their features, such as "ball" and "red". Its ability to generalize is tested on whether it can "imagine" a ball when it hears the word "ball" or whether it can "verbalize" the word "ball" when it is shown a ball.

(2) Having learned the meanings of words such as "ball", "triangle", "circle", DE TE proceeds to learn words that describe events in which such objects are involved. For example, DE TE learns the meaning of the words "moves" and "stands". Further, it learns about various types of motions (such as "moves horizontally", "moves vertically", and "moves diagonally"). The motions become more complex and start involving object interactions. As a result DE TE learns the meaning of the word "bounces".

(3) In the next stage of language skill development DE TE starts putting words together into short phrases. Here its ability to generalize is tested by forcing it to imagine (in response to verbal inputs) objects with their visual features (like "red square") which it has never seen before. Also, in the same set of experiments DE TE is asked to describe with a short noun phrase an object shown on the screen (e.g., "small red ball").

(4) The construction of longer word sequences requires that DE TE learn simple syntactic rules regarding word order. DE TE's ability to do that (based on the built-in Morphologic/Syntactic Procedural Memory) is tested on a simple rule which states that in a noun phrase which is formed by an adjective-for-size (adjZ), an adjective-for-color (adjC) and a noun, the correct word order is: adjZ adjC noun (e.g., "large blue square").

(5) An essential aspect of DE TE's language acquisition is its ability to interact with a user in the form of questions and answers. To illustrate this aspect, in a series of experiments DE TE is taught to answer simple questions about objects and their features like: *Q1*: "What is the color of the small

ball?" (while looking at a small ball) A1: "Red." Q2: "What is bigger?" (while looking at a small triangle and a large square) A2: "Square." In the process DETE learns to compare objects with respect to their size (e.g., bigger or smaller) and spatial relations (e.g., closer, farther, in-front, behind).

(6) Understanding propositions about the temporal relations between events is an essential linguistic ability which the majority of children acquire and bring to perfection by the time they reach four years of age. One of the several possible linguistic expressions of such relations is provided by the verb tense. Based on its unique neural architecture (which contains Temporal Memory Planes allowing for explicit representation of time) DETE learns the meaning (as opposed to the morphology) of a number of verb tenses including past, present, and future and their perfect forms for a small set of verbs (e.g., hits, hit, has hit, will hit).

(7) Other linguistic skills on which DETE has been tested include the learning of homonyms and the acquisition of selected grammatical features typical of different languages. For instance, DETE can learn gender agreement like in Spanish "la pelota roja" vs "el cuadro rojo".

1.2.3 Overview of implementation

Both the procedural and the neural part of the model are implemented in *Lisp (Thinking Machines Corporation, 1988) -- a data-parallel extension of Common Lisp (Steele, 1984) used for programming the CM-2 Connection Machine (Thinking Machines Corporation) (Hillis, 1985; Hillis and Steele, 1986). The Connection Machine CM-2 is a massively parallel computer with up to 65,536 individual processors (the version in which DETE was developed has 16K processors). Each processor contains 64K or 256K bits of local Random Access Memory (RAM) and a single-bit processing unit. The processors run in a Single Instruction Multiple Data (SIMD) mode. The communications between the processors are carried over an n -dimensional hypercube interconnection scheme which permits highly efficient n -dimensional grid communications. The system software provides a set of very efficient operations over "parallel variables" including SCAN and SPREAD operations. For instance, if $n*m$ processors are connected in a $m \times n$ 2-D grid, the summation (product, max, etc.) of a parallel variable value in all processors on a row of the grid (i.e. to add together one value from each processor on a row of the grid and distribute the sum into the rest of the processors on the same row) takes only $O(\log m)$ time. An important feature of the CM-2 is that any subset of its processors can be turned off so that the instructions are only performed by those processors that are currently active. Every 32 processors share a floating point processing unit which allows a 32-bit number to be stored across 32 processors (i.e., one bit per processor). These 32 processors can each access this 32-bit number as if it were stored in its own memory. The CM-2 uses a serial computer such as a VAX, Symbolics Lisp Machine or SUN-4 as a front-end machine. The front-end system is used to program the CM-2 using parallel extensions to the familiar programming languages LISP, C and FORTRAN. For more details concerning implementation of DETE, see appendices.

1.3 Motivations and goals

This research was motivated by a set of beliefs about the relationship between language and perception and the structure of systems capable of such cognitive functions.

1) *Interdependency of language and perception*: To understand how humans learn language we have to study the phenomenon of language as part of the complex interactions which we have with

the environment, including visual, auditory and motor. The idea that visual experiences play a significant role in the formation of the semantics of natural languages has been independently suggested by a number of linguists (Fauconnier, 1985; Jackendoff, 1983; Jackendoff, 1987; Lakoff, 1987; Langacker, 1987). The inverse relation, namely that languages structure (categorize) our perceptual experiences, has also been shown to exist (Talmy, 1983). It has been hypothesized that more abstract concepts about objects, events, and relations can be developed from more concrete, visually based ones through a process of analogy. While it is not clear yet what are the neural mechanisms involved in analogical reasoning, at this stage of our knowledge we can examine the possibility that basic language understanding is a result of interactions of high-level visual and verbal (speech) representations through complex mechanisms of memory and attention.

2) *Language and symbolic reasoning are emergent properties of a priori organized and sufficiently complex nervous systems*: Our human ability to reason symbolically is an emergent property of a highly structured and enormously complex processing system - the nervous system and more specifically the human brain. It is extremely unlikely that a complex cognitive system capable of performing multiple functions, including: recognition, recollection, learning, etc., could be built via learning from an initially random network. Thus, a great deal of innate structure is needed.

3) *Cognitive systems acquire their knowledge and skills through learning*: Humans acquire almost all of their skills by learning through experience (while in contrast most of the computer systems that exhibit some cognitive skills have been programmed to do so). Learning has definite advantages as compared to being programmed. For instance, being in contact with the perception of the physical world through sensory devices provides automatically information about all constraints and relations that exist in this world. This eliminates the need to construct by hand a model of the world -- an approach taken in most symbolic artificial intelligence (AI) systems. In other words, it is more natural to let the system itself discover the physical constraints and self-organize accordingly, instead of us (the programmers) having to (1) discover the relations in the physical world, (2) find a reasonable representation for these relations, and (3) program these into a computer.

The goal of this research is to construct a system capable of *learning* to perform the PGLA task - in other words, a system that can associate temporally related visual and verbal inputs, generate and manipulate internal representations of these inputs and initiate behaviors such as language output and mechanical interactions with the external environment. The system is a hybrid, composed of a number of functional modules, some of which are implemented as neural networks and others as procedures (non-neural implementation). The neural network modules form the core of the system -- i.e. the various types of memory. In these modules the information is stored in a distributed (vs. localist) manner. The peripheral sensory devices and preprocessors are designed as procedural modules. This choice of implementation is based on two factors: (1) The objective of this research is not to model the visual, the auditory, or the attentional systems in detail. Our main objective is to model, in a neurally realistic fashion, the memory mechanisms that underlie the cognitive processing that supports language. (2) Constructing an operational cognitive system exclusively out of neural modules is computationally very expensive and it was not practical in the framework of this research.

1.4 Background

1.4.1 Philosophy

Psychologists, linguists, cognitive scientists, neuroscientists and recently researchers in the field of Artificial Intelligence (AI) have been proposing and verifying models of systems underlying cognitive processes at various levels of detail. However, large gaps in knowledge are still present today. The main one is the lack of adequate knowledge of the relation between the *mind* and its abilities for language and thought as researched by philosophers, psychologists, and linguists, and the *brain* underlying the mind and viewed with its enormous complexity (neuroanatomical, biochemical and biophysical) by neuroscientists. In other words, there is still a missing bridge between *mind* and *brain*. There are strong camps of researchers that maintain opposite views of the relation between mind and brain. On the one hand, philosophers, psychologists, linguists, and symbolic AI researchers claim that their assumptions (i.e. the physical symbol system hypothesis) (Newell, 1980) are sufficient to explain the variety of human cognitive behavior, or at least language and thought. However, even their best attempts so far have proven to work only on toy-problems and are very fragile and unscalable when it comes to the explanation of human-level phenomena. On the other hand, neuroscientists and some cognitive scientists tend to view the complexity of the *mind* as an emergent behavior of the complex neural (connectionist) systems which underlie it. Recent developments in neural nets research have provided strong support for such a view (Smolensky, 1988).

The following sections provide brief descriptions of the symbolic and connectionist views and an outline of the contribution which this thesis offers to the problem of finding the relation between *mind* and *brain*.

The symbolic model

Theories of natural language processing (NLP) in Artificial Intelligence usually start with a set of primitives. For example, Conceptual Dependency (Schank, 1972) theory contains: **acts** -- (PTRANS, ATRANS, ...); **cases | slots | relations** -- (ACTOR, ABOVE, INSIDE, ...); **causality** -- (ENABLES, MOTIVATES, LEAD-TO, ...); **modalities** -- (TIME, DURATION, ...). Then these theories try to relate the meanings of words in terms of similar and varying configurations of the primitive elements and in this fashion to represent more complex sentence meanings. For example, the concept of "*John went home by car*" can be represented using the formalism of the Conceptual Dependence Theory (Schank, 1972; Schank and Abelson, 1977) as a PTRANS (physical transfer) from an unknown location to a new location -- home. Where the actor of the PTRANS is John and the car is the instrument of this act.

The proponents of the symbolic approach (Fodor, 1975; Fodor, 1987; Newell, 1980; Fodor and Pylyshyn, 1988) view the mind as a symbol system and the process of cognition as manipulation of symbols. Symbols capture mental phenomena such as thoughts and beliefs. Traditionally, the symbolic approach disregards the brain as the physical substrate that underlies the mind and postulates that the mind can be described completely in symbolic terms. While the symbolic view of mind is a useful formalism, we believe that it is only a formalism and it is not implemented in the brain the same way, for instance, as symbols are implemented in a conventional computer (memory locations with their addresses or values). A symbol system as defined by Harnad (Harnad, 1989) is:

"(1) a set of arbitrary "*physical tokens*" (scratches on paper, events in a digital computer, etc.) that are

- (2) manipulated on the basis of "*explicit rules*" that are
- (3) likewise physical tokens and *strings* of tokens. The rule-governed symbol-token manipulation is based
- (4) purely on the *shape* of the symbol tokens (not their "meaning"), i.e., it is purely *syntactic*, and consists of
- (5) "*rulefully combining*" and recombining symbol tokens. There are
- (6) primitive *atomic* symbol tokens and
- (7) *composite* symbol-token strings. The entire system and all its parts -- the atomic tokens, the composite tokens, the syntactic manipulations (both actual and possible) and the rules -- are all
- (8) "*semantically interpretable*." The syntax can be *systematically* assigned a meaning (e.g., as standing for objects, as describing states of affairs)."

The main thesis of the symbolic approach is that the symbolic level (by which the symbolists mean the mental level) has its own functionality and is independent of the specific physical realizations of the symbols. The concept of an autonomous symbolic level conforms to general foundational principles in the theory of computation and applies to all the work being done in symbolic AI. This is the branch of computer science that has so far been the most successful in generating (hence explaining) intelligent behavior. For a good example of the power of the symbolic approach, when applied to the understanding of natural language, see (Dyer, 1983).

The connectionist model

In parallel with the symbolic explanation of the *mind*, another approach exists, which has been at times more (and at times less) attractive. This approach was pioneered by Rosenblatt (Rosenblatt, 1962) and is presently known as "connectionist", "neural network", "dynamical systems" or "PDP" approach. It was almost forgotten for 15 years, mostly due to Minsky and Papert's negativistic prospect on the field (Minsky and Papert, 1969) and was recently re-born as a general theory of cognition and behavior (McClelland et al., 1986). According to connectionism, cognition is not symbol manipulation but processing of dynamic patterns of activity in a multilayered network of nodes or units with weighted positive and negative interconnections. The patterns change according to internal network constraints governing how the activations and connection strengths are adjusted on the basis of new inputs (e.g., the "delta rule" and "back-propagation") (McClelland et al., 1986). The result is a system that learns by experience, recognizes patterns, solves problems, and can even exhibit motor skills.

Scope and limits of the two approaches

There is considerable overlap in the scope of the symbolic and connectionist approaches; however, neither one has gone much beyond the stage of "toy" tasks toward life-size behavioral capacity. More specifically, the symbolic approach seems to be better at formal and language-like tasks. Our linguistic capacities are the primary examples here, but many of the other skills we have (e.g., logical reasoning, mathematics, chess-playing, perhaps even our higher-level perceptual and motor skills) also seem to be symbolic. However, there are some major unaddressed and unresolved issues in the symbolic area. These include: 1) What are the meanings of the primitive elements if any? 2) Why is a given set of primitives more appropriate than another? 3) How are the primitive elements themselves learned? These issues are part of a larger problem which has been the object of extensive discussions recently, namely the issue of the nature of the symbols in a symbolic system. The symbolic approach also suffers from a severe handicap, one that may be responsible for the limited nature of its success to date (especially in modeling human-scale capacities) as well as the ad

hoc nature of the symbolic knowledge it attributes to the “mind” of the symbol system. This handicap has been noticed in various forms since the advent of computing and one manifestation of it is termed the “symbol grounding problem” (Harnad, 1987).

Connectionist systems, on the other hand, are better at sensory, motor and learning tasks. However, they seem to be at a disadvantage in attempting to model higher cognitive functions (Pinker and Prince, 1988). Nevertheless, no connectionist system so far has been able to achieve the power of symbolic systems when it comes to handling language or thought (e.g., logical reasoning, planning, etc.). There are at least three main reasons for the current impotence of the connectionist approach.

1) **Structure:** Connectionism is still in its infancy. Connectionist models lack complex structure comparable to the structural architecture of the human brain. It is my belief that the brain circuitry (as described by neuroanatomists) is there with a purpose and that evolution has provided a good (if not optimal) solution to the efficiency problem. Thus, one would expect that behavior comparable to humans can be achieved with at least comparable complexity of the structures underlying it (especially if we are not only interested in systems that exhibit only a small set of behaviors but rather in systems having the large gamut displayed by humans).

2) **Common language:** There is no common language which can be used to map appropriately the behavior of the neural systems (usually described in mathematical and statistical terms) to the behavior of symbolic systems. This is basically an issue of *interpretability*. It is possible that such a scientific language will emerge gradually and in parallel with the successful charting of the range of behaviors observed in neural models and also due to the increasing interest from both camps to find such a communication medium.

3) **System Neuroscience:** Many connectionists are not interested in building a model of the *brain* as a system (i.e. try to explain its behavior using knowledge of its intrinsic subsystems). Instead, due to a lack or neglect of neuroscience knowledge, they build models of the *brain* as data (i.e. attempt to fit an explanation to the human’s intelligent behavior using that behavior as data that needs to be fitted -- explained by a model), and any model that fits the data is acceptable for them. While it is true that neuroscience knowledge may still be insufficient to construct a detailed model, an advantageous approach will be to at least incorporate as much as it is known and therefore work with a “grey-box” model instead of a completely “black-box”.

1.4.2 Methodology

There are three approaches to modeling complex systems. (1) “*Black box*” modeling, (2) *Isomorphic* modeling, and (3) “*Grey box*” modeling.

(1) “Black box” model: -- A “black box” model of a system is based only on observations of the system’s behavior in response to various inputs. This approach is also referred to as “functional” or “top-down” -- model the Input/Output behavior without regard to brain. In the terminology of dynamical systems theory, such approach is called “modeling of data”. The main problem with this approach is that, in disregarding completely the internal structure of the system, it does not provide unique solutions, since there are many ways to fit a curve to a given set of data. Despite this shortcoming, such models are often very helpful.

(2) *Isomorphic* modeling: -- The other extreme approach to system modeling is to take into account all structural and functional details of the system. In other words, to construct an isomorphic model of the system. This approach is also referred to as “structural” or “bottom-up” --

copy known neurocircuitry and see what it can do, without incorporating top-down constraints from the task/domain. Unfortunately, for complex systems such as the human brain, this approach is impractical since relatively little is known about the detailed neurocircuitry within and among various brain areas. Also, little is known about the neurochemistry underlying the generation, modulation and communication of signals in the brain.

(3) “*Grey box*” modeling: -- An alternative approach, which lies between the two previously mentioned, is to regard the system as a “grey box”, in other words to use not only the information about the input/output behavior of the system but also whatever details of the structure and function of the system are available to the extent that they can be incorporated in the model.

In DETE I have taken the third approach, motivated by the belief that complete understanding of the phenomena of language and perception cannot be achieved without taking into account the neural mechanisms that underlie such phenomena. I also believe that the temporal dynamics of the visual and verbal processes is important for understanding cognitive processes. DETE is sensitive to the duration of words (how long to say it), to pauses between words, and also to the temporal dynamics of the visual input. In contrast, both symbolic and PDP models usually apply a fixed amount of computation to each input before going onto the next input.

Any attempt to model the brain is constrained to its level of detail. Examples of such levels of detail are: (1) the behavioral level (language, reasoning, perception, etc.), (2) the neural systems level (visual, verbal, attentional, memory, and motor), (3) the cellular level (various types of neurons and the circuits in which they are involved), (4) the subcellular level (various proteins, receptors, membrane channels, neurotransmitters, etc.) (Sejnowski et al., 1988). Individual modules in DETE are modeled at different levels of detail. For instance, the sensory devices of DETE, which receive and preprocess the visual and verbal inputs, are modeled only at the behavioral (functional) and neural systems levels. The modeling of the memory modules in DETE is closer to the cellular level. Also, an attempt has been made to interpret some aspects of these neural networks at the subcellular level (see section 13.5).

A number of simplifications have been made even for the modules in which significant attention has been paid to the correctness of the functional and structural details. A basic simplification in the construction of our model is the discretization of the naturally analog (i.e. continuous) information processes in the brain. Also, a number of major characteristics of the visual and verbal systems are disregarded in the model. Some of them are: (1) In the visual system -- the existence of two retinas and two brain hemispheres, and thus, the ability for stereoscopic vision as well as the ability for perception of textures. (2) In the verbal system -- the information which is normally conveyed by the prosodic qualities of spoken language (e.g., stress, pitch, etc.) and which is missing in written language (the type of input to be used in DETE). This simplification prohibits us from studying such linguistic phenomena as prosody (which is important in early language acquisition).

1.5 Guide to the reader

The material covered in this dissertation is organized in four major parts. Each part consists of a number of chapters. There are altogether 14 chapters. Supporting data is given in 5 appendices. 112 figures and 23 tables have been used to illustrate important points and experimental results throughout the thesis.

Part one provides a general overview of the task and the system.

Chapter 1 introduces the task of Perceptually Grounded Language Acquisition. This is followed by a description of the model in general terms and an outline of its implementation. A demonstration of the model in action is used to illustrate some of its major capabilities. The motivation and goals of this research project are discussed in the latter part of the chapter in the light of related work on symbolic and neural network based systems for Natural Language Processing.

Chapter 2 gives an overview of DETE. The structure of the system is presented as a block-diagram and the architecture, function and information representation in each module are described in general terms. The visual and verbal modalities of the environment in which DETE operates are also described here.

Part two focuses on details of the architecture and functions of the major subsystems in DETE including the visual, verbal, motor, selective attention, and memory modules. Examples of the input/output behavior of each subsystem, the representations used and the mechanisms to construct these representations are also presented.

Chapter 3 gives details of the structure and function of DETE's visual system. The representations of five basic visual features of an object (shape, size, color, location, and motion) produced by visual feature extractors working in parallel on the visual input, are discussed and the algorithms for these procedurally implemented modules are presented.

Chapter 4 describes the verbal subsystem in DETE. It gives details on the representation of the verbal input (gra-phonemic representation), and presents the algorithms for the generation and decoding of such representations.

Chapter 5 outlines the temporal relations (dynamics) between the visual and verbal modules of DETE. A temporal hierarchy of processing in these subsystems is presented and supported by evidence from electrophysiology and psychophysics.

Chapter 6 focuses on the motor subsystem. It describes the choice of representations of the state of DETE's effectors -- the EYE and the FINGER. The variety of motions in which these effectors can be involved are also discussed here.

Chapter 7 is concerned with the mechanism for selective attention. The components of this neurally inspired (but procedurally implemented) system are described and the specific representation of the selective attention in DETE is discussed.

Chapter 8 presents the basic memory mechanism used in DETE -- the KATAMIC sequential associative memory. The results of a detailed numerical study of the dynamics of this memory mechanism and its limitations are summarized here.

Chapter 9 focuses on the specific types of memory in DETE. The choice of memory mechanisms was inspired by numerous neuropsychological and physiological studies that have characterized what memory types are necessary for supporting language and thought processes in the brain. In the context of this chapter memory is defined as the physical traces left from experiences in the brain and the mechanism for storing and retrieving of these traces (i.e. for "converting" of some activity into a trace and vice versa). Each of the memory types is discussed here in terms of its dynamics, including sensory modality, storage mechanism, capacity, recall and recognition, consolidation and forgetting.

Chapter 10 puts together the complete system of DETE. It provides details of the system architecture, a qualitative and quantitative (numbers of neural elements used, etc.) description of the individual modules and their connectivity patterns.

Part three provides an evaluation of the performance of the complete system.

Chapter 11 gives details of the various experiments performed with DETE to evaluate its ability to perform the PGLA task. It begins with learning of single words for objects, their features and various events in which these objects can be involved. Then DETE's ability to generalize its knowledge to new instances is tested. This is followed by a question answering session in which DETE describes in one or a few words visual features of individual objects. DETE's ability to learn spatial and motion relations between two objects is tested next. Finally, this chapter describes how DETE can acquire the meaning of various verb tenses including present, past, and future -- in their simple and perfect forms.

Part four compares DETE with other models and provides neuropsychological and neurobiological insights into the cognitive processes in humans.

Chapter 12 compares DETE along several functional dimensions to both classical symbolic models for natural language processing (NLP) and connectionist models (localist and distributed). A substantial part of this chapter is devoted to a comparison of the KATAMIC model to Kanerva's Sparse Distributed Memory (SDM) as well as to Elman's Simple Recurrent Network (SRN) -- two neural network models which are related to the KATAMIC model.

Chapter 13 provides a parallel between the various functional modules of DETE and various brain structures which are known to be involved in the language and visual processing of the PGLA task in humans. Possible brain counterparts of DETE's perceptual, attentional, and memory mechanisms are discussed. This chapter also makes an attempt to map the KATAMIC memory to the cerebellar cortex in structural and functional terms. As a result, a novel theory of cerebellar cortex is proposed and supported by substantial evidence from neurophysiology, neuroanatomy, and neurochemistry.

Chapter 14 describes the current status of this research project and outlines a number of possibilities for future extensions of the system. It also summarizes the whole project and discusses its major contributions.

Appendix A provides details of DETE's implementation on the CM-2 Connection Machine. Some of the advantage of using the CM-2 and the *LISP programming language are pointed out here. Also, provided is a summary of DETE's code and the usage of the CM-2 memory.

Appendix B gives examples of *LISP code for the basic modules of DETE including the Visual Feature Extractors, the KATAMIC sequential associative memory, and the verbal pre- and post-processing routines.

Appendix C contains code for selected experiments with the KATAMIC model and with DETE. I/O examples of some of the experiments are also given here.

Appendix D gives details of the routines used for monitoring of the models behavior.

Appendix E discusses possible neural network implementations of some of DETE's procedural modules.

2 SCOPE OF TASK AND OVERALL DETE ARCHITECTURE

This chapter presents an overview of DETE. The structure of the system is shown as a block-diagram and the architecture, function and information representation in each module are described in general terms. The visual and verbal modalities of the environment in which DETE operates are also described here.

2.1 Input and Output Behavior

2.1.1 Input

DETE accepts three types of inputs -- visual, verbal and motor.

The visual input is composed of series of visual scenes (movie frames) containing blobs -- simple-shaped objects such as circles, triangles or squares. These visual scenes are projected on a square visual screen of the size 64 by 64 pixels (i.e. a total of 4096 pixels). I will refer to this visual set-up as the Visual Screen (VS). DETE looks at the VS through a circular aperture -- retina (EYE or Visual Field) with a variable diameter. The center of the retina can be positioned anywhere on the screen and the diameter can vary from a few pixels to covering the whole VS. The circular retina is composed of neurons arranged in a square grid and each neuron corresponds to one pixel in the VS (i.e. retinotopic mapping).

The verbal input is composed of occasional utterances containing descriptions of current, past or future visual scenes (frame sequences), or questions demanding verbal responses (description of scenes), or requiring motor responses (manipulation of objects in the scene).

The motor input comes in the form of externally guided motions of DETE's motor devices -- an EYE and a FINGER. Two simulated joysticks are used. The first one controls the position of the EYE and the size of the retina by issuing commands such as: `move_eye` [from (x_0, y_0) to (x_1, y_1)] and `zoom_retina` [from d_0 to d_1]. The second joystick controls the position of the FINGER on the visual screen and the type of effect it has upon objects in the VF. Typical commands for controlling the finger are: `move_finger` [from (x_0, y_0) to (x_1, y_1)], `push_object` [object A from (x_0, y_0) to (x_1, y_1)], `hit_object` [object A at (x_0, y_0)].

2.1.2 Tasks

The model is tested in three different ways:

1) **Verbalization task:** A visual scene is shown to DETE and it is asked to generate a verbal description of the scene. For instance, DETE sees in the center of the Visual Screen a medium-sized red circle and a small blue square above and to the right of the circle. DETE is asked the question "What is this?" Depending on its prior learning and experiences in answering questions of this type it can produce several different answers (Figure 2.1). During the process of verbal answer generation DETE's attention (location of retina and its size) can follow different trajectories. During learning, DETE's attention may be controlled by the teacher; however, during performance, DETE's attention is under autonomous control. In the current implementation, the order in which attention is

switched from one object to another is determined by the objects size. First, DETE attends to the largest object on its retina, followed by the smaller in size, etc. This control is implemented procedurally.

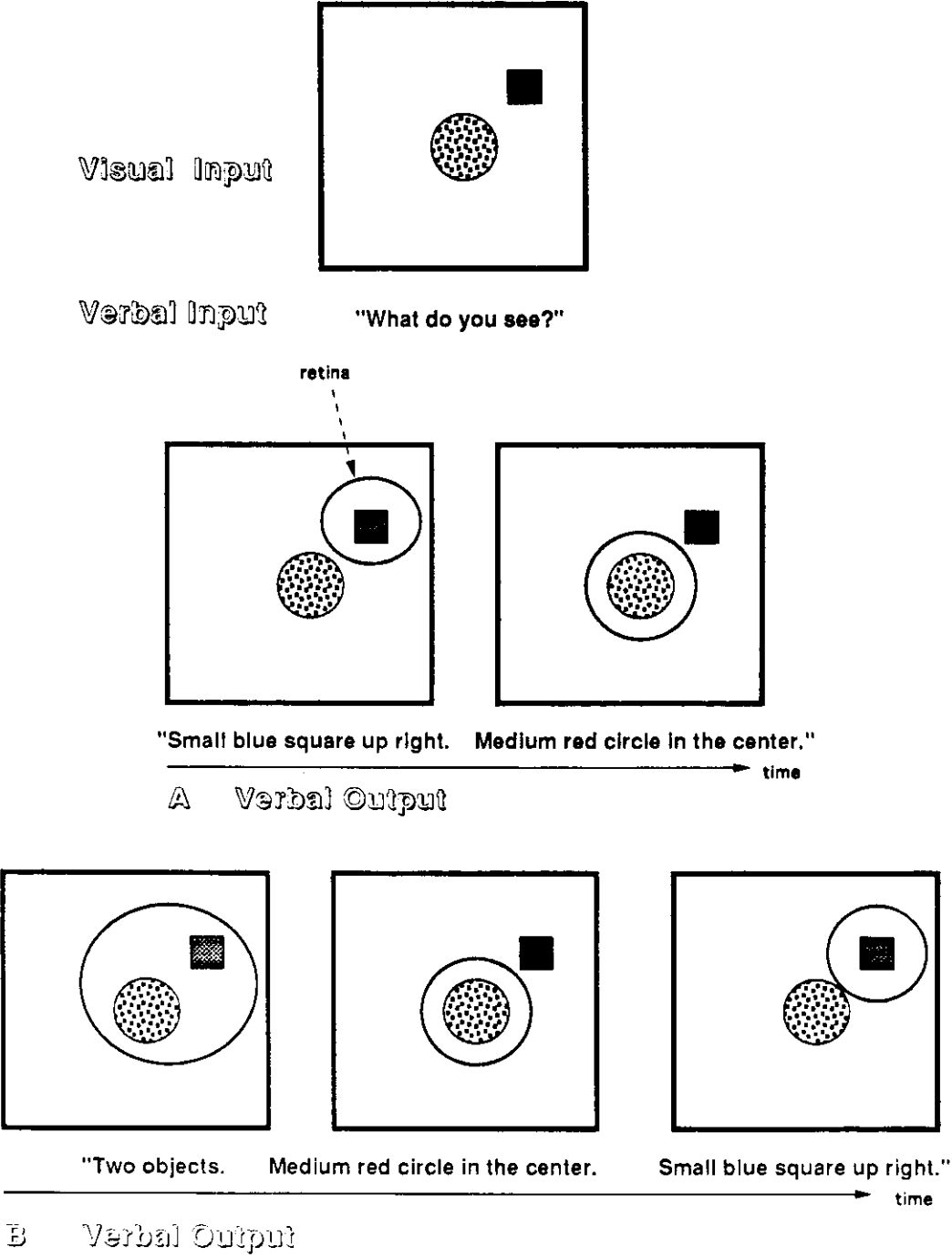


Figure 2.1: Tasks: Verbalization

Schematic drawing of the sequence of events involved in the generation of two different verbal descriptions of the same visual scene. **A:** DETE's attention is focused first to the square and then to the circle and as a result the following verbal response is produced "Small blue square up right. Medium red circle in the center". **B:** DETE focuses its attention to both objects at the same time after which it shifts it to the larger object (the circle) followed by the smaller (the square) and produces the following utterance "Two objects. Medium red circle in the center. Small blue square up right".

2) **Imaging task:** The user gives DETE a verbal input and observes the internal image (or sequence of images) generated in its "mind's eye" in response to this input. For instance, DETE hears the sentence "Red triangle moves up, hits blue circle and bounces". This verbal input causes a sequence of "mental images" to be generated in DETE's "minds eye" (Figure 2.2).

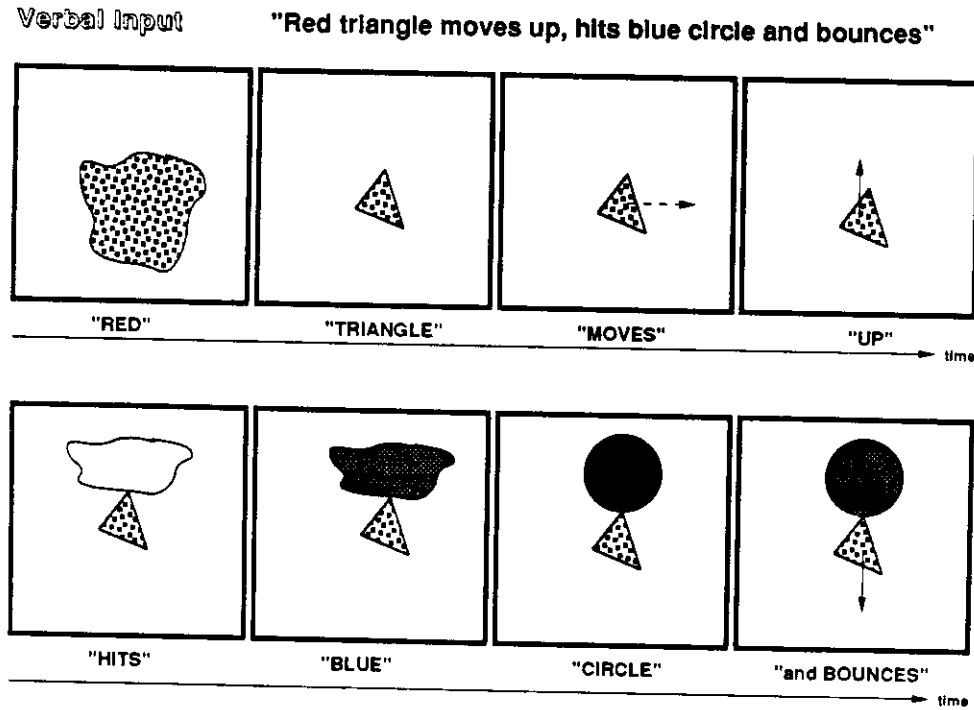


Figure 2.2: Tasks: Imaging

Schematic drawing of the sequence of "mental images" generated in DETE's "minds eye" in response to the verbal input "Red triangle moves up, hits blue circle and bounces". The word "RED" induces the image of a blob (of the shape most commonly seen by DETE) with red color. "TRIANGLE" specifies the shape (overrides the default shape). "MOVES" activates the representation of motion (shown here schematically by an arrow) with its most common (default) speed and direction (e.g., right). "UP" further specifies the direction of motion (overrides the default motion). "HITS" induces the representation of another blob located on the motion path of the triangle. The words "BLUE", "CIRCLE" further specify this blob. "BOUNCES" induces the representation of a bouncing event which is characterized by an abrupt change of motion direction of an object when it contacts another object (shown as the arrow pointing down).

3) **Motion task:** Generation of EYE and FINGER motions in response to verbal commands or internal (joystick) signals. For instance, suppose there are three objects on the screen, a blue circle, a red triangle and a red square. DETE hears "Push left the red square". The sequence of actions performed by DETE's EYE and FINGER are shown schematically in Figure 2.3.

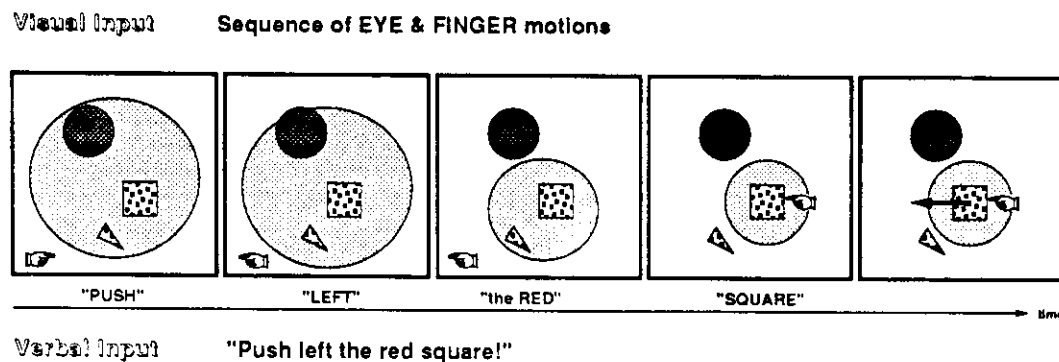


Figure 2.3: Tasks: Motor actions

The sequence of words causes the following to happen. The word "PUSH" focuses the attention on the FINGER (it is ready to move from its default position -- lower left-hand corner is the most common direction shown by the dimmed finger/hand icon, but does not move until there is something to push). Also, the visual attention initially encompasses all objects. "LEFT" causes an adjustment in the motion direction of the FINGER (i.e. overrides the default direction). "THE RED" reduces the focus of attention to the two red objects, the triangle and the square. "SQUARE" further focuses the attention. At this stage the verbal input is complete and a single object is being attended to. Once the object of the action is found the FINGER moves accordingly. This creates the necessary and sufficient conditions for the event of pushing to take place.

2.2 Learning tasks and relation of learning to performance

DETE's performance on specific tasks is not static -- it improves with experience and new more complex tasks can be learned on the basis of simpler, previously learned tasks. For instance, DETE does not know at first how to associate a word with an object, or how to follow a verbal command. These abilities must be learned.

In DETE there is not a fundamental separation between learning and performance (like in many PDP models). Instead, learning and performance are often interleaved. Also, learning simpler tasks helps DETE in learning more complicated tasks. For instance, As will be seen in Chapter 11, after DETE has learned the meanings of the words “circle”, “square”, “triangle”, “red”, “green”, “blue”, “small”, “medium”, and “large”, now it can further learn the correct word order in a noun phrase containing words for objects and adjectives for color and size.

2.3 Architecture

DETE’s architecture is shown as a block diagram in Figure 2.4. A list of the individual modules is presented below. For a detailed description of their function and connectivity see Chapters 3, 9, and 10.

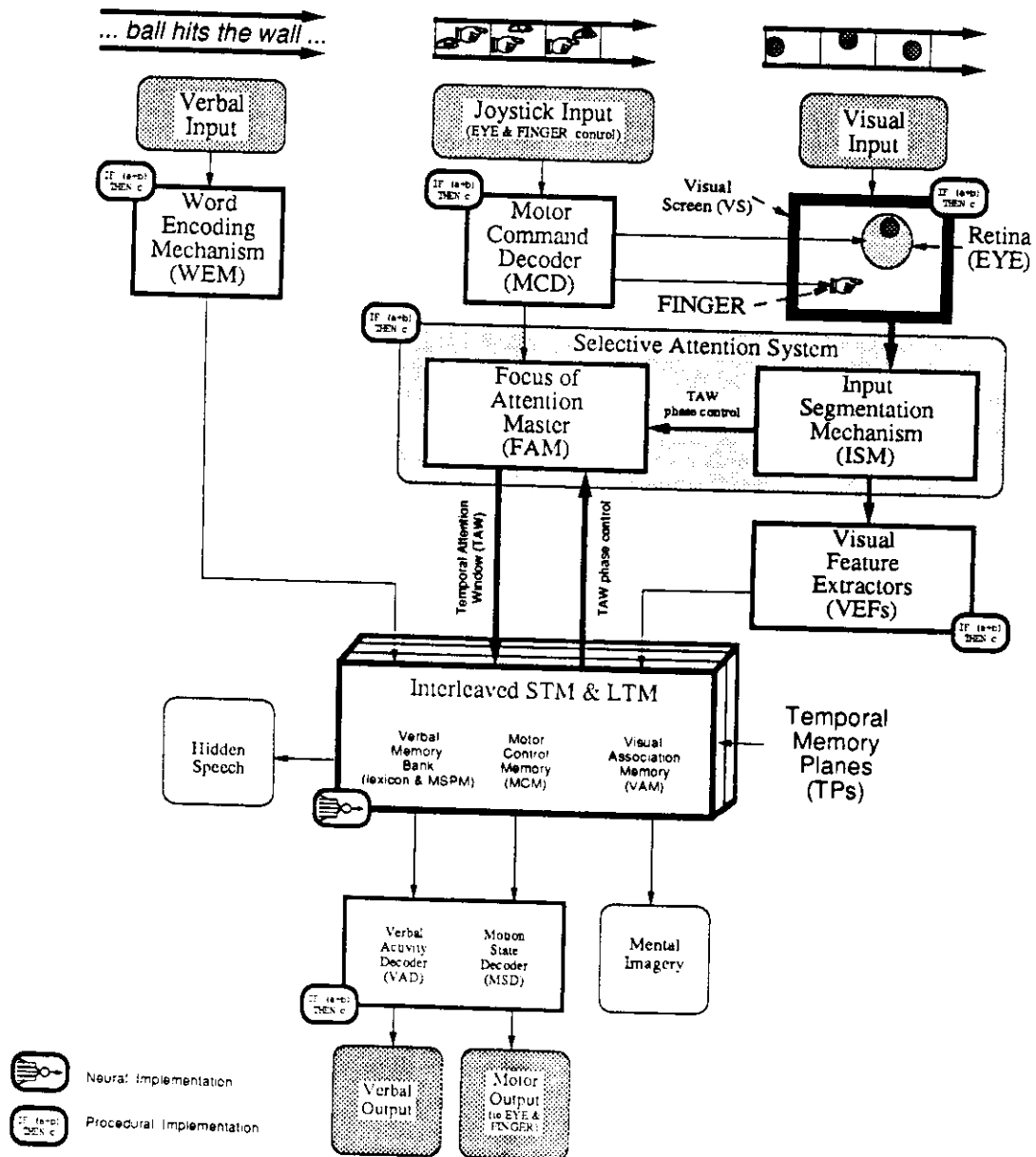


Figure 2.4: Block diagram of DETE

Two separate icons are used to show how each of DETE's modules has been implemented (procedurally or neurally).

2.3.1 Input devices

Information from the external world can enter DETE through three input devices (sensors): 1) retina -- a visual input device, 2) Word Encoding Mechanism (WEM) -- a verbal input device, 3) Motor Command Decoder (MCD) -- a motor input device.

1) *The retina (EYE)*. DETE looks at its visual world through a circular aperture, a retina (upper right corner of Figure 2.4). The size of this aperture is variable. It is usually smaller than the size of the visual world to which DETE is exposed, but can vary from a minimal size of a few pixels to a maximal size which is equal to the size of the visual world (Figure 2.5). The control over the size of the retina is done externally by a teacher during learning.

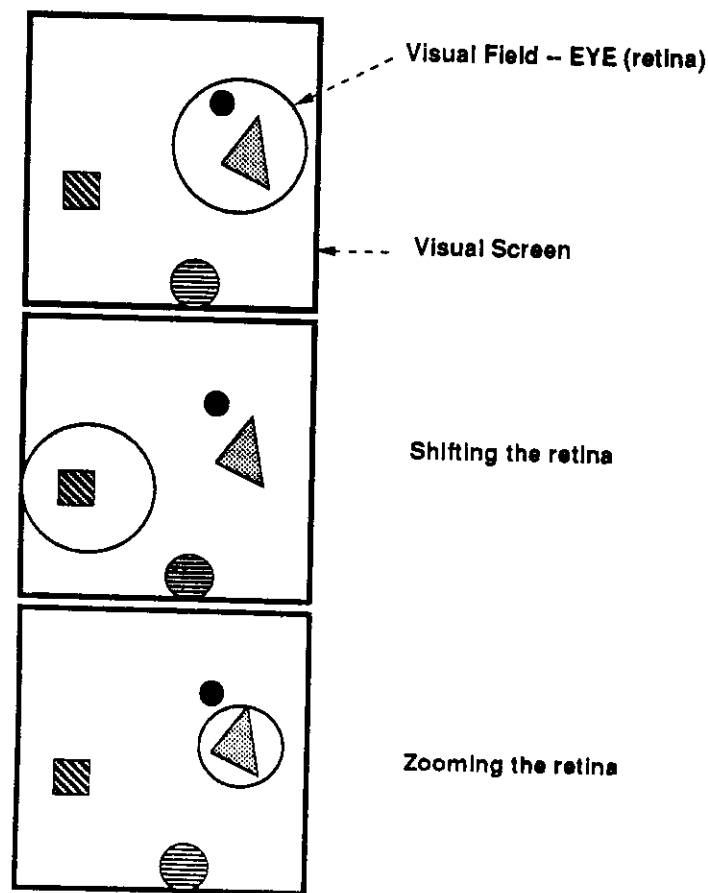


Figure 2.5: Functions of DETE's retina

Three scenes (visual frames) in which the retina has different positions and sizes.

The location of the retina on the screen is also directed externally by a joystick which is controlled by a teacher (Figure 2.6). In a natural setup this would correspond to a parent pointing something out in the environment while at the same time giving a verbal clue like "Look here". Children have the innate ability to move their gaze to different locations. In DETE, for the sake of simplicity, initially this action is performed by an external mechanism. In other words, initially the teacher moves DETE's retina and later DETE learns to move it on its own or in response to a verbal command in FIRLAN. If there are no verbal instructions, DETE's retina moves around from object to object at random exploring the Visual Screen.

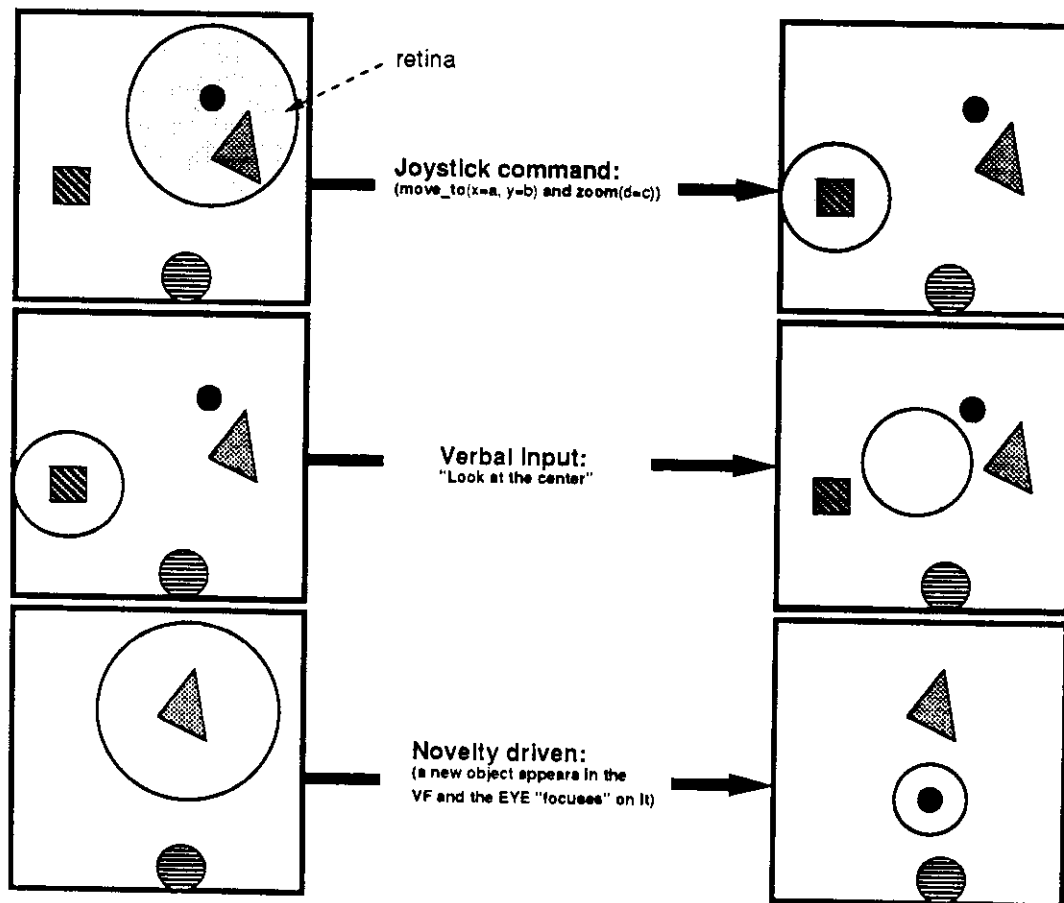


Figure 2.6: Retinal control

DETE can move its retina from point A to point B in response to: (1) external command provided by the teacher through the joystick, (2) verbal command given by the teacher, e.g., "Look at the center", (3) at random during the process of unsupervised exploration of the environment.

2) *The Word Encoding Mechanism (WEM)* is a procedural module, i.e. non-neural, (see Appendix B.5) which takes a string of words (sentences) entered through a keyboard or read in from a text file. It encodes the text in a sequence of binary patterns called **gra-phonemes**. There is one gra-phoneme for each letter of the English alphabet. Each gra-phoneme is a five-step long sequence of 64 bit binary patterns. In other words, a gra-phoneme has a spatial and a temporal dimension (64 bits and 5 steps respectively). The five patterns forming a particular gra-phoneme

are all the same while the patterns of different gra-phonemes differ. The representation is called gra-phonemic because some of its features (e.g., the number of elements -- gra-phonemes) correspond to the number of graphemes (letters) in the alphabet, whereas, other features encode phonemic aspects of language. For instance, in each binary pattern representing a gra-phoneme there are a number of 1 bits (the rest are 0) whose positions in the pattern in general encode frequency formant positions in corresponding phonemes. Words are represented as sequences of gra-phonemes separated by transition patterns. Sentences are represented as sequences of words separated by longer transition patterns (randomly generated patterns of 0s and 1s with 1-bit-density equal to the 1-bit-density of the gra-phonemes). The purpose of using a gra-phonemic representation of the verbal input is to provide DETE with a distributed representation of the verbal input which has the potential for encoding some prosodic features of speech. For instance, by having letter sequences, DETE can learn something like an intonation pattern, e.g., "BIIIG" means very big, "SMAAAL" means very small.

3) *The Motor Command Decoder (MCD)*. A procedural mechanism (upper middle of Figure 2.4) which allows the teacher to control the position of the retina on the screen and its diameter via a joystick. It also allows the teacher to control the position of the FINGER on the screen. This mechanism also conveys the state of the retina (i.e. location and aperture size) to the motor plane in DETE.

2.3.2 Selective Attention System

The Selective Attention System (SAS) (middle of Figure 2.4) consists of two components:

1) *Focus of Attention Master (FAM)*. The FAM contains a continuously running clock (an internal oscillator) which produces a spike (action potential) once every 5 time steps. The phase of this clock can be shifted back or forth. In other words, a given tick of the clock can be delayed by 1 to 4 steps (i.e. the phase of the FAM is shifted back) or it can come 1 to 4 steps sooner (i.e. the phase is shifted forward). After a phase shift, the clock continues to oscillate with the same frequency but with its new phase. Each tick of the FAM is used to open a functional time window -- the Temporal Attentional Window (TAW) which is one step long. The TAW controls a number of processes in the memory modules, including whether information should be stored in the short term memory (STM) or not. Under this scheme, DETE cannot handle more than 4 distinct objects at once on the retina (since there are only 4 possible delays of phase).

2) *Input Segmentation Mechanism (ISM)*. The ISM is an array of neural elements of the size 64*64. It takes the input from the retina and segments out the individual objects while preserving their topographical relations. At the output of the ISM each object is represented by a set of oscillating neurons. All neurons activated by a given object oscillate in phase, while the oscillations among different objects are out of phase. An object that is located in the middle of the retina is phase locked with the FAM clock while the farther away from the center of the retina an object is, the more it lags behind the FAM clock.

The SAS is implemented as a procedural module that takes three types of inputs: (1) Input from the motor control decoder -- this input carries information about the position and size of the retina. (2) Input from the retina itself -- this input carries information about the objects that appear on the retina. (3) Input from the memory system -- this input drives the state (phase) of the Focus of Attention Master. The function of the selective attention system is described in detail in section 7.2.

2.3.3 Visual Feature Extractors

This subsystem, (middle right of Figure 2.4) consists of a set of five feature extraction modules. The modules interact with the selective attention subsystem (that allows DETE to focus on one object at a time) and extract (in parallel) the visual features of shape, size, color, location and motion (represented as direction and speed). The outputs of these filters are further associated in the visual parts of DETE's memory. Detailed description of the visual feature extractors and the representations of the visual input that they generate is given in Chapter 3.

2.3.4 Memories

The memory modules of DETE (see Figure 2.4) are described in detail in Chapters 8, 9, and 10. All memory modules are implemented as neural networks based on a novel sequential associative memory architecture (see Chapter 8). They include:

- (1) Verbal Memory (VM). Contains memory traces of the verbal inputs.
- (2) Visual Feature Memories (VFM). Contain memory traces of the visual inputs.
- (3) Motion memory (MM). Contains memory traces of motion trajectories.

The Visual and Verbal Memories are further subdivided into:

a) *Short-term memory (STM)* (a.k.a. primary or iconic memory). It is used as a small capacity buffer for input information in all modalities. It is implemented in DETE as a type of sequential associative memory.

b) *Long-term memory (LTM)*. It contains traces of past experiences of DETE which can be unique episodes or well rehearsed items (e.g., the LEXICON in middle of figure 2.4). Like the STM it is also implemented as a related type of sequential associative memory. The LTM comes in several varieties which are discussed in section 9.2, e.g., (a) Declarative Memory (DM) specialized as Semantic Memory (SM) and Episodic Memory (EM); (b) Procedural Memory (PM). Both the STM and the LTM are partitioned into Verbal and Visual memories.

c) *Temporal Memory (TM)*. This is a special purpose neural network which consists of a set of Temporal Planes (a total of 8) and is used in the learning of a number of linguistic tasks such as verb tenses understanding. It contains visual and verbal components as well as STM and LTM components.

A block diagram of DETE's memory architecture is shown in Figure 2.7. For simplicity, only two out of the five Visual Feature Memories are shown, together with only two out of the eight Temporal Memory Planes. The STM and LTM components of each Feature Memory are not shown. The four motor memories are connected in a similar way and therefore they are not shown on the figure. The connectivity patterns are drawn schematically.

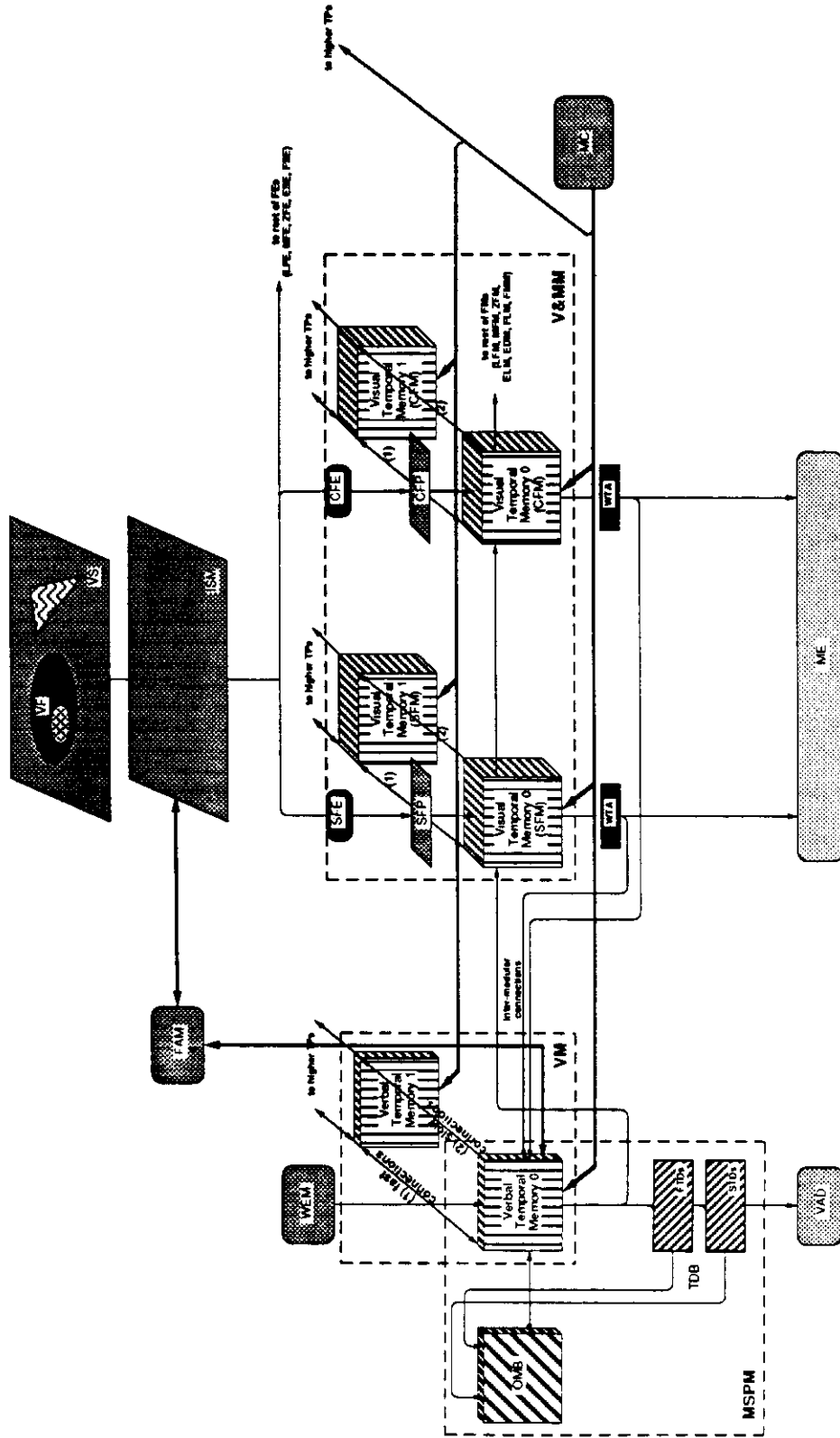


Figure 2.7: Block diagram of DETE's memory

Block diagram of DETE's memory architecture. The abbreviations in the figure refer to: VF(EYE) -- Visual Field; VS -- Visual Screen; FAM -- Focus of Attention Master; ISM -- Input Segmentation Mechanism; WEM -- Word Encoding Mechanism; SFE(P,M) -- Shape Feature Extractor (Plane, Memory); CFE(P,M) -- Color Feature Extractor (Plane, Memory); LFE(P,M) -- Location Feature Extractor (Plane, Memory); MFE(P,M) -- Motion Feature Extractor (Plane, Memory); ESE -- Eye State Extractor; ELM -- Eye Location Memory; EDM -- Eye Diameter Memory; FSE -- Finger State Extractor; FLM -- Finger Location Memory; FMM -- Finger Motion Memory; WTA -- Winner Take All mechanism; MSPM -- Morphologic/Syntactic Procedural Memory; VM -- Verbal Memory; TDB -- Transition Detectors Bank; TDs -- Transition Detectors; V&MM -- Visual & Motor Memories; VAD -- Verbal Activity Decoder; ME -- Mind's Eye; MC -- Moment Clock. Wires crossing each other at straight angles do not make contacts. Thin wires indicate data lines. Thick wires indicate control lines. Gray and black rectangles indicate peripheral (procedurally implemented) modules. Wires labeled by (1) are fast connections between Temporal Memory Planes. Slow connections are labeled by (2). The direction of data and control signal flow is indicated by arrows.

2.4 DETE's Micro World

Human vision and language are enormously complex phenomena. This complexity is reflected on the one hand in the variety of difficult visual and linguistic problems which the visual and verbal systems have to solve and on the other hand in the complexity of the brain structures and functions involved. While trying to remain realistic, for the purpose of this dissertation I have severely reduced the scope of the problem space. The external world, and specifically its two modalities (visual and verbal) which form DETE's operational task/domain, have been reduced to the "Blobs World" domain and the "FIRLAN language" task respectively.

2.4.1 The Blobs world

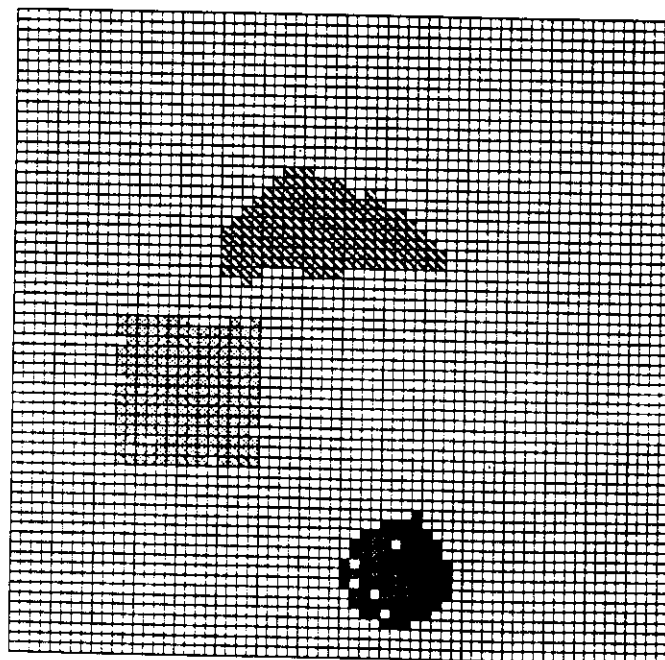


Figure 2.8: The Blobs World

Examples of blobs on DETE's Visual Screen -- a square, a circular, and a triangular shape. Note that the shapes are imperfect and noisy.

Instead of the complex world of visual scenes, DETE operates in the "Blobs World". In this world there are only simple objects of various shapes (blobs) moving on a simulated screen. I call this the Blobs World since the shapes do not have to be perfect and because they do not have internal structure -- objects composed of parts are not allowed (e.g., trees, bicycles, stick figures). DETE is designed to handle distorted or noisy shapes and also shapes that only resemble perfect shapes. For some examples of blobs see Figure 2.8.

Despite the fact that such a world may seem to be limited, it is actually quite rich. The scope of the possible verbal descriptions of the Blobs World is presented in Table 2.1.

| |
|--|
| <ul style="list-style-type: none">• Single objects and their features<ul style="list-style-type: none">- size (e.g., small, medium, large, ...)- shape (e.g., circle, triangle, same, similar, ...)- location (e.g., up, left, right, lateral, close, below, ...)- color (e.g., red, green, blue, ...)- movement (e.g., moving, standstill, ...)<ul style="list-style-type: none">- direction (e.g., north, east, parallel, along, ...)- speed (e.g., slow, fast, sluggish, ...)• Spatial relationships between & within objects<ul style="list-style-type: none">- location (e.g., above, inside, between, ...)- count (e.g., two, five, many, ...)- order (e.g., first, second, next, last, ...)• Temporal relationships<ul style="list-style-type: none">- present (e.g., now, ...)- future (e.g., next, after, later, will, ...)- past (e.g., before, long-ago, did, ...)- modifiers (e.g., soon, immediately, just, ...)• Object actions and interactions<ul style="list-style-type: none">- size (e.g., shrinks, expands, ...)- collisions (e.g., bounce, hit, stop, push, ...)- containment (e.g., enter, exit, ...)- relative movement (e.g., bypass, faster, slowest, ...)• DETE / Teacher interactions<ul style="list-style-type: none">- questions (e.g., "where is the ball", ...)- commands (e.g., "push the red ball", ...) |
|--|

Table 2.1: Scope of the Blobs world

Words that stand for various features are given in parentheses. Only English words are presented here. However, DETE is designed to learn any words and their syntax, semantics and pragmatics for other natural languages (specifically Spanish and Japanese in this dissertation).

2.4.2 The verbal modality

The natural languages which we hear and speak invoke (in the case of hearing) or express (in the case of speaking) "meanings" which are represented in our brains. These meanings are established through long-term, repeated personal experiences and depend on a number of factors, among which are the innate constraints of our perceptual systems. Since DETE is an artificial system, which does not possess elaborate analogues of all human perceptual systems, it is logical to infer that the language that it learns will depend mostly on its perceptual systems. As a result, the meanings of, say, English words describing the Blobs World can be somewhat different than the meanings that humans possess for the same words. To illustrate this and to avoid confusion regarding the interpretation of word meaning by the reader, DETE is taught an artificial language called FIRLAN. FIRLAN is based on a dictionary of English words. The actual purpose of using FIRLAN is to demonstrate that initially the utterances which we hear from our parents in our native language do not mean anything to us and only later with experience do we learn to attribute meanings to them.

Any natural language has a number of manifestations. It can be spoken -- a phenomenon that has various attributes (e.g., talking mouths, listening ears, sound waves in the air, systems for speech generation and audition in the brain, also prosody, temporal dynamics, etc.). It can be written -- a different phenomenon with its own attributes (e.g., writing or typing systems of the brain/body, reading abilities and physical storage media -- papers, monitors, etc). DETE receives its verbal input through a pseudo-auditory modality. In this modality words are represented as simulated sound streams (i.e. gra-phoneme sequences).

The grammatical structure of the FIRLAN language is specified in Table 2.2. (An alternative to a grammar is to use templates to specify the verbal input used.) FIRLAN's grammar is a very restricted subset of English grammar. Note that this grammar can generate some nonsensical sentences (e.g., S=WH(*when*) VI(*are*) NP(Det(*a*) NP1(OBJ(*circle*) LOC(*near*))))). For this reason all sentences generated were screened before they were used in DETE. That is, DETE is only given meaningful sentences -- ones that correspond completely to its perceptual inputs.

| | | |
|----------|---|---|
| S | = | NP NP VP WH VI NP |
| NP | = | Det NP1 Det NP1 and Det NP2 |
| VP | = | VI PP VT NP |
| NP1, NP2 | = | OBJ COL OBJ SIZ OBJ OBJ LOC OBJ MOT COMB |
| PP | = | LOC NP REL-SIZ NP |
| WH | = | When Where What How |
| VI | = | is are was were will be |
| VT | = | touch hit bounce enter shrink move |
| Det | = | a the |
| OBJ | = | circle square triangle |
| COL | = | white red orange yellow green blue purple black |
| SIZ | = | small medium large |
| REL-SIZ | = | smaller larger the smallest the largest |
| LOC | = | in_center above below left_of right_of far near |
| MOT | = | fast slow still north east west south |
| COMB | = | SIZ COL OBJ LOC |

Table 2.2: A syntactic specification of FIRLAN

S (Sentence); NP (Noun Phrase); VP (Verb Phrase); PP (Picture Phrase); WH (question word); VI (verb "auxiliary"); VT (verb "true"); Det (Determiner); OBJ (Object); COL (Color); SIZ (Size); REL-SIZ (Relative Size); LOC (Location); MOT (Motion); COMB (Combination)

While humans (due to the great similarity in their visual systems) perceive the visual world similarly, the different languages which they use "carve up" the visual reality in different ways. For instance, while some languages use a single word or phrase for a given object or event (e.g., piñata) other languages have to construct a whole phrase to express the same concept and some times only approximately (e.g., a party where children hit an object made of paper with candy inside).

Even between individuals that speak the same language, their perception of the visual reality is usually "carved up" differently depending on their experiences. For instance, a painter has a much richer vocabulary where colors are concerned (e.g., he/she can probably name a variety of blue colors like *aqua marine*, *cielo blue*, *navy blue*, etc.) than an auto mechanic who, on the other hand, has a richer vocabulary with respect to auto parts.

Languages differ also not only in term of their lexicons but also in terms of grammars.

DETE, by virtue of its memory mechanisms, can learn different languages. To demonstrate this ability, a second artificial language -- SECLAN (SECond LANguage) is introduced and used in parallel with FIRLAN. (See Table 2.3. for a specification of SECLAN.) At the current implementation, FIRLAN and SECLAN differ mostly in terms of their lexicons. Some of the lexical entries of these languages are the same while most are different. There are also some differences in the grammatical structures. For instance, the word order in the noun phrases containing adjectives for location and motion are reversed (**bold** in Tables 2.2 and 2.3).

| | |
|---------|--|
| S | = NP NP VP WH VI NP |
| NP | = Det NP1 Det NP1 and Det NP2 |
| VP | = VI PP VT NP |
| NP1,NP2 | = OBJ COL OBJ SIZ OBJ LOC OBJ MOT OBJ COMB |
| PP | = LOC NP REL-SIZE NP |
| WH | = When Where What How |
| VI | = is are was were will be |
| VT | = touch hit bounce enter shrink ... |
| Det | = a the |
| OBJ | = oval edged |
| COL | = warm_color cold_color |
| SIZ | = tiny small average big huge |
| REL-SIZ | = smaller bigger the smallest the biggest |
| LOC | = in_middle on_periphery |
| MOT | = stationary moves |
| COMB | = SIZ COL OBJ LOC |

Table 2.3: A syntactic specification of SECLAN

2.4.3 The motor modality

The motor modality in DETE is the most rudimentary portion of the system. As was mentioned before, DETE has only two effectors (articulators): a FINGER and an EYE. The states of the EYE

and the FINGER change with time. The FINGER moves *smoothly* within the Visual Screen with various speeds and directions while the EYE *jumps* from location to location and changes its diameter. The states of the effectors are continuously input to DETE's Motor Command Decoder (MCD) -- a procedural mechanism which generates a representation of these states in a set of motor planes. The MCD consists of two mechanisms, a Finger State Extractor (FSE) and an Eye State Extractor (ESE). The state of the FINGER is characterized by (1) its location in the Visual Screen, (2) its motion (speed and direction). The Finger State Extractor (FSE) represents these two state parameters in two separate state planes -- Finger Location Plane (FLP), and Finger Motion Plane (FMP). The EYE State Extractor (ESE) also generates two state planes -- Eye Location Plane (ELP), and EYE Diameter Plane (EDP) in which it represents the two state parameters of the EYE (location and diameter).

Both the FINGER and the EYE have default states (i.e. states in which they reside while they are not involved in any action). In its default state the EYE is centered in the middle of the Visual Screen with maximal diameter (looks at the whole Visual Screen), while the default state of the FINGER is stationary and positioned also in the middle of the Visual Screen.

The state planes of the effectors are passed as input to the motor memory (MM). The motor memory associates trajectories (sequences of states) of the effectors with activity within the visual and verbal memory modalities. The motor memory is composed of four parts -- one for each motor plane. These memories are: Finger State Memory (FSM); Finger Motion Memory (FMM); Eye Location Memory (ELM); Eye Diameter Memory (DM). The outputs of each part of the motor memory are passed through a Winner Take All (WTA) mechanism (which ensures that only one possible trajectory is produced) and through four Motor State Decoders (bottom of Figure 2.4) which in turn provide internal motor inputs to the EYE and the FINGER.

2.4.4 Consistency of inputs

The most important characteristic of the external inputs to DETE is their consistency. In other words, they obey a set of rules (e.g., physical laws within the visual modality and syntactic and semantic rules within the verbal modality). This characteristic of the external (verbal and visual) world is important if we want avoid creating a "schizophrenic" DETE.

1) *Visual consistency*. -- If all objects in the Blobs World are elastic then if an object hits a wall it should always bounce according to the laws of physics.

2) *Verbal consistency*. -- The language which DETE is taught has to be meaningful (i.e. properly structured both syntactically and semantically).

3) *Verbal & visual consistency*. -- If DETE is focusing on a yellow ball we should call it YELLOW during each learning trial, and not once RED, another time BLUE, etc., (unless of course RED, BLUE and YELLOW are meant to be synonymous in FIRLAN).

2.5 DETE's World

2.5.1 Internal World

While operating in the three modalities (visual, verbal, motor) of the external world as defined above, DETE maintains its own internal world. Its behavior in this internal world model is represented by the ability to have "mental imagery" and "hidden speech" (see very bottom of Figure 2.4). *Mental imagery* manifests itself as: (1) internal (mental) completion of noisy external-images;

(2) scene segmentation into objects based on selective attention; (3) dynamic synthesis of mental objects in the visual memory; (4) mental object manipulation in response to a verbal input. *Hidden speech* is manifested as spontaneous (usually fragmentary) utterances produced during recognition of external images or internal (mental) image manipulation.

The existence of mental imagery and hidden speech outputs allow the user to observe internal aspects of DETE's knowledge during intermediate stages of learning.

2.5.2 Physics of External World

To provide DETE with a visual field that contains a variety of moving objects, a simple simulator of the visual world was developed (see Appendix B.3). This simulator generates blobs of various shapes, colors, and sizes and allows their motion to be controlled in terms of direction and speed. The blobs can bounce off the walls of the Visual Screen or off any other blob (or the FINGER) that is placed on their path of motion. The bounces are elastic and obey the laws of solid-state physics.

2.6 Modes of operation

The system can be considered generally to work in three operational modes, which are not fully independent. These are learning, testing, and free-association. Actually, in the behavior of children there is often no clear-cut distinction among these three conceptually different modes. Children often switch rapidly from one mode to another and often three modes are interleaved (e.g., free-association can have a learning component and the same goes for testing). The same is true for DETE.

(1) *Learning mode* -- consists of multiple pairings of visual scenes, verbal description of the scenes, and interactions between DETE's EYE & FINGER and the objects in the scene.

(2) *Testing mode* -- the user feeds DETE verbal and/or visual and/or motor inputs and expects it to produce a verbal, visual (mental image) or motor response.

(3) *Free-association mode* -- the network generates mental images (in its mental imagery output or "mind's eye") as a result of interactions between activity patterns in the various memories.

2.7 Learning in DETE

In DETE there are two modes of learning:

(1) Without a teacher (unsupervised) -- happens continuously as a result of association of the extracted visual features in the visual association memory. This learning process is essential for building the representation of the physical constraints in the visual input. That is, DETE learns the physics of the world it is presented on its screen, as seen through its retina. For example, DETE learns that objects bounce off each other. However, if objects behaved differently, DETE could learn a different kind of visual physics. For instance, if at the contact with a stationary object, a moving object disappears, then DETE could learn the word "pierce" or "penetrate".

(2) With a teacher (supervised) -- during visual, verbal and motor association. Supervised learning involves all three memories -- visual, verbal, and motor.

PART II

Modular structure of DETE

Part II focuses on the detailed structure of the individual modules of DETE. The representations of visual, verbal and motor inputs are presented and the structure and function of the mechanisms that produce these representations are described.

3 THE VISUAL SYSTEM

DETE's visual system consists of a retina and five Visual Feature Extraction (VFE) modules. These modules extract location, size, color, motion, and shape. The retina looks at part of the Visual Screen and the retinal output is fed in parallel to the VFEs.

3.1 The retina (EYE)

DETE looks at its visual world through a circular aperture, a retina. The size of this aperture is variable. It is usually smaller than the size of the Visual Screen (64*64 pixels), and can vary from a minimal size of 7 pixels in diameter (total area of 29 pixels) to a maximal size which is equal to the size 64 pixels in diameter. Examples of the retina in different positions and with different diameter are given in Figure 3.1. Several "pure" and "noisy" objects are also shown on the same figure. The control over the size of the retina is provided externally by a teacher.

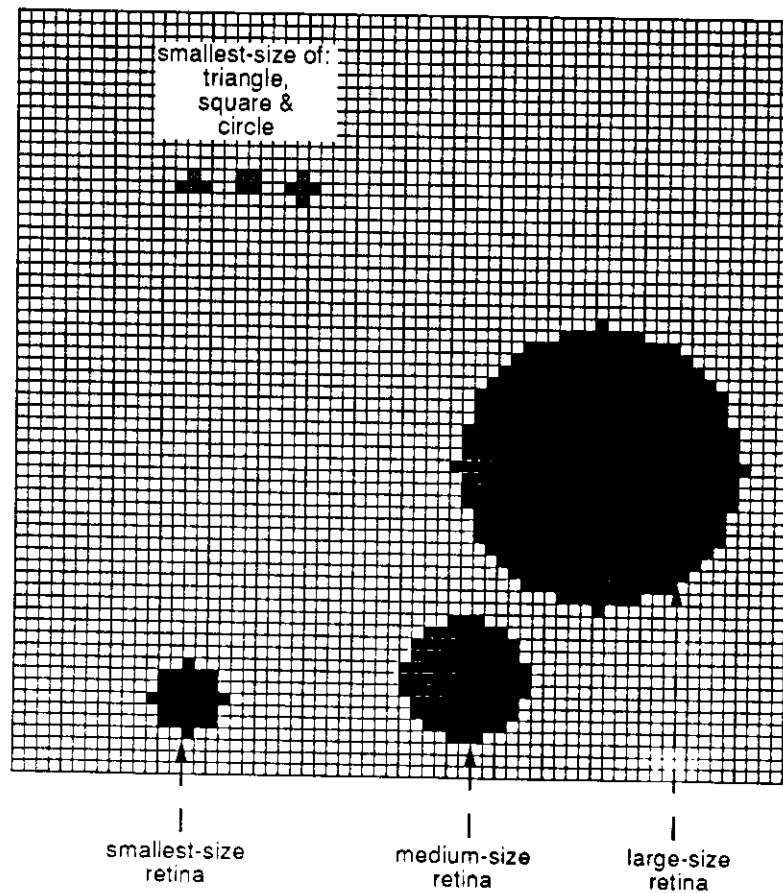


Figure 3.1: DETE's retinal structure

The retina is shown in three different positions on the Visual Screen and with different sizes (small, medium and large). Also, the smallest size triangle, square, and circle are shown.

The function of the retina is to pass the continuous visual input which it gets directly from the Visual Screen (unchanged) to the Input Segmentation Mechanism (ISM) of the Selective Attention System. The retina passes one 8-bit word per pixel at each time cycle from the Visual Screen to the ISM (Figure 3.2). Each 8-bit word encodes the color information for the corresponding pixel. Pixels which are part of an object send their color values (encoded in the 8 bits). Pixels that are part of the background are black (i.e. their color value is 0). The retina sends also another type of information to the ISM, namely, the location of its center on the Visual Screen and the size of its diameter.

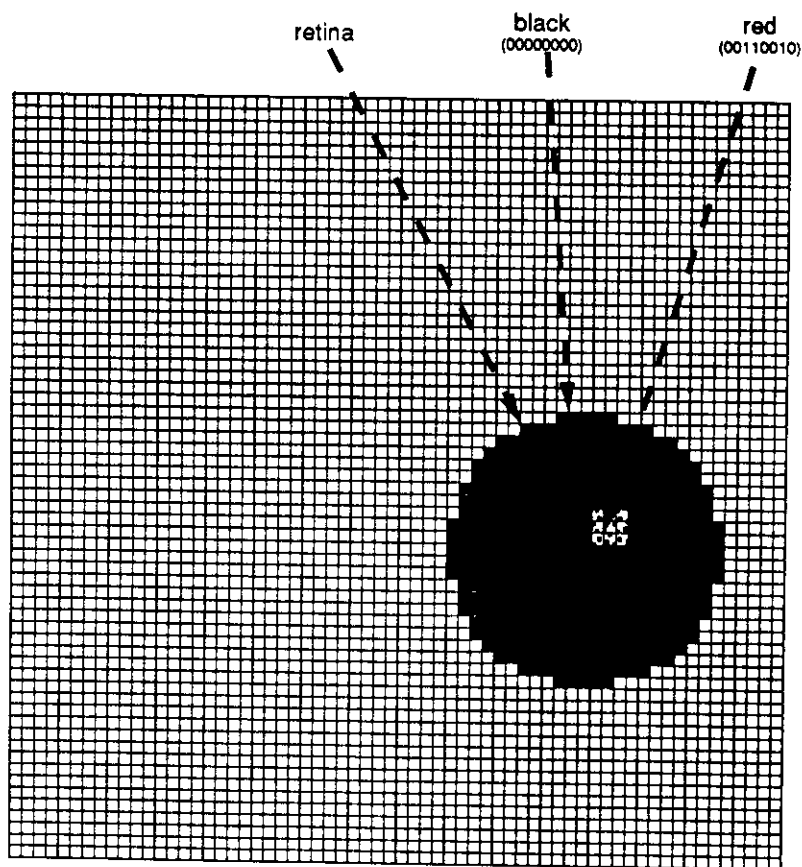


Figure 3.2: Retinal output

Encoding of visual input to the retina. At each time cycle all pixels in the retina send in parallel to the ISM (preserving the topology) an 8-bit word which encodes the color of the pixel. The figure shows the values of these words for two pixels, one that belongs to a red ball and the other to the retinal field (black). The rest of the visual field sends also 0s to the ISM (i.e. it is black), however on the figure it is shown white for distinction from the retina.

3.2 Visual feature extractors

The range of visual features accessible to DETE is determined by the Visual Feature Extractors (VFEs). It is important to note that the visual feature extractors *are not implemented* as neural networks. Each VFE is a simple algorithm (procedure) that takes the visual input from the retina and computes (extracts) the information concerning the particular feature dimension. I have chosen to use these algorithms instead of providing a special purpose neural network for each of the VFEs since the neural network approach is more computationally expensive. Also, the focus of this research is not on modeling the visual system in a neurally realistic fashion. The visual system in DETE is used only to provide appropriate representations of the visual features to the higher association areas. How these representations were initially generated is considered irrelevant to further processing.

Each feature extractor generates a *feature plane* which represents the space of possible values of the particular feature. All feature planes in DETE (i.e. Shape FP, Color FP, siZe FP, and Motion FP) are non-topographic in nature except for the location feature plane (LFP) which is topographic. Such non-topographic mapping leads to a reduction of the computational resources required. In other words, not every point in the visual screen is represented in every plane and the topology is not always preserved. However, this representational approach also substantially reduces the representational ability of the system. Every image projected through the retina to the input of a feature extractor generates a specific activity pattern at particular locations on a given feature plane. Each feature of an object is represented by 4 contiguous active neurons on the space of the plane. The main purpose for choosing more than one neuron to represent a single feature is that of redundancy of coding. The specific number 4 was chosen for reasons of computational resource availability and constraint by representational considerations. In each of the feature planes individual features are represented by a set of active neurons. An active neuron is a neuron that oscillates, i.e. it fires periodically (with output 1) and is silent the rest of the time (with output 0).

3.2.1 Shape Feature Extractor

The shape feature extractor (SFE) is a procedural module designed to recognize different shapes by selectively attending to one object at a time. The SFE recognizes simple shapes independently of size, location and noise/distortion. The SFE produces a Shape Feature Plane (SFP) onto which all shapes are mapped. To recognize individual shapes, the SFE uses a set of shape templates. Currently this set contains three templates, a circle, a square and a triangle. These shapes were chosen so that rotation is not needed for recognition. Each template has variable dimensions, i.e. it can grow or shrink in size while preserving its shape. The image of each individual object conveyed from the retina to the SFE is compared in parallel to each of the three templates in the following manner. (1) Starting with a maximal size (covering the whole Visual Screen) each template is fitted (in parallel) to the object, i.e. its size is decreased while at the same time its position is adjusted until a tight fit between the template and the object is achieved. (2) For each of the fitted templates T_i (where i is triangle, square or circle) a *Shape Difference* measure $SD(T_i, O)$ with respect to the object (O) is computed: $SD(T_i, O) = \frac{\text{pix-}T_i}{\text{pix-}O} - 1$. Where $\text{pix-}T_i$ and $\text{pix-}O$ are respectively the number of pixels in the fitted template i and the number of pixels in the object. The more similar an object is to a given template, the smaller the value of the shape difference measure. (3) Using the SDs of the previous computations, the representation of the object's shape is generated in the SFP (see Appendix B.1.1 for code).

The Shape Feature Plane is a square array of neurons with dimensions 16*16 pixels or a total of 256 pixels. All objects of the same shape are mapped to the same row of neurons on the SFP (Figure 3.3). The mapping has been designed such that it provides to some extent “smooth” transitions between shapes. For instance, triangular shapes are mapped onto the bottom few rows of units with “pure” triangles (i.e. $SD(T_{triangle}, O) = 0$) mapped to row 5). Above them are mapped rectangularly shaped objects with “pure” squares (i.e. $SD(T_{square}, O) = 0$) mapped onto row 10). Both triangles and rectangles have vertices and that is why they were placed closer together in the SFP. The circular shapes are mapped above the rectangular shapes (a circle can be regarded as a square transformed by smoothing its edges) with “pure” circles mapped onto row 15. Objects with shapes that do not fit perfectly against either of the templates are mapped onto the SFP depending on the relations between their SDs with respect to the three templates. For details of the mapping see Table 3.1.

| | | | | | | |
|---|---|---|---|---|-----|------------|
| S | > | T | > | C | ==> | rows 0,1,2 |
| S | > | C | > | T | ==> | rows 3,4 |
| C | > | S | > | T | ==> | rows 6,7 |
| C | > | T | > | S | ==> | rows 8,9 |
| T | > | S | > | C | ==> | rows 13,14 |
| T | > | C | > | S | ==> | rows 11,12 |

Table 3.1: Mapping of shapes in the SFP

The symbol “>” is used to represent the level of similarity between shapes: squares (S), triangles (T), and circles (C) are measured by means of SD (shape differences) as defined in the text. If an object’s SD measure with respect to the square template is bigger than that measured with respect to the triangle template which in turn is bigger than the SD measure with respect to the circle, then the object has a shape which is something similar to a circle and a triangle (e.g., a triangle with rounded edges) but not very similar to a square. The shape of such an object is represented as activation in rows 0, 1, or 2 of the SFP depending on the actual SD values (see row 1 of the table).

DETE does not have to actually recognize a square. Rather, it has to be able to discriminate different shapes and place them into different categories. Then different perceptual categories can be associated during learning with words in the verbal input. Theoretically, at least, the visual feature extractors could be improved and would supply a large set of possible discriminations (and hence verbal categorizations) of the input, which could then be verbally associated without changing the rest of DETE’s architecture. This would allow a greater vocabulary to be learned, e.g., if DETE could discriminate perceptually teardrop shape from a star shape then DETE could learn the “meaning” of the words “tear” and “star” used to describe these shapes.

Each instance of a particular shape is represented as a localized activation pattern of 4 neurons in a row. Again the choice of 4 neurons (instead of 1) is to provide some level of redundancy. Redundancy in the encoding of features is necessary for two reasons. First, to ensure that if the model is lesioned this will not affect the overall performance significantly. Second, since the dynamics of the model, as it will be demonstrated further, is to some degree stochastic, the redundancy of coding allows a feature to be present even if not the complete set of neurons (that are supposed to represent this feature) is active at a time. The SFP has the capacity to represent up to 4 individual objects of a particular shape-range that appear simultaneously in the retinal field. All

active neurons oscillate with a constant frequency. In other words, they fire once every 5 cycles and are silent the rest of the time. Neurons that represent the same object fire *in phase* while the oscillations of different objects are out of phase. As a result, at each time cycle the SFP represents at most one object since only the neurons that represent one object fire in phase.

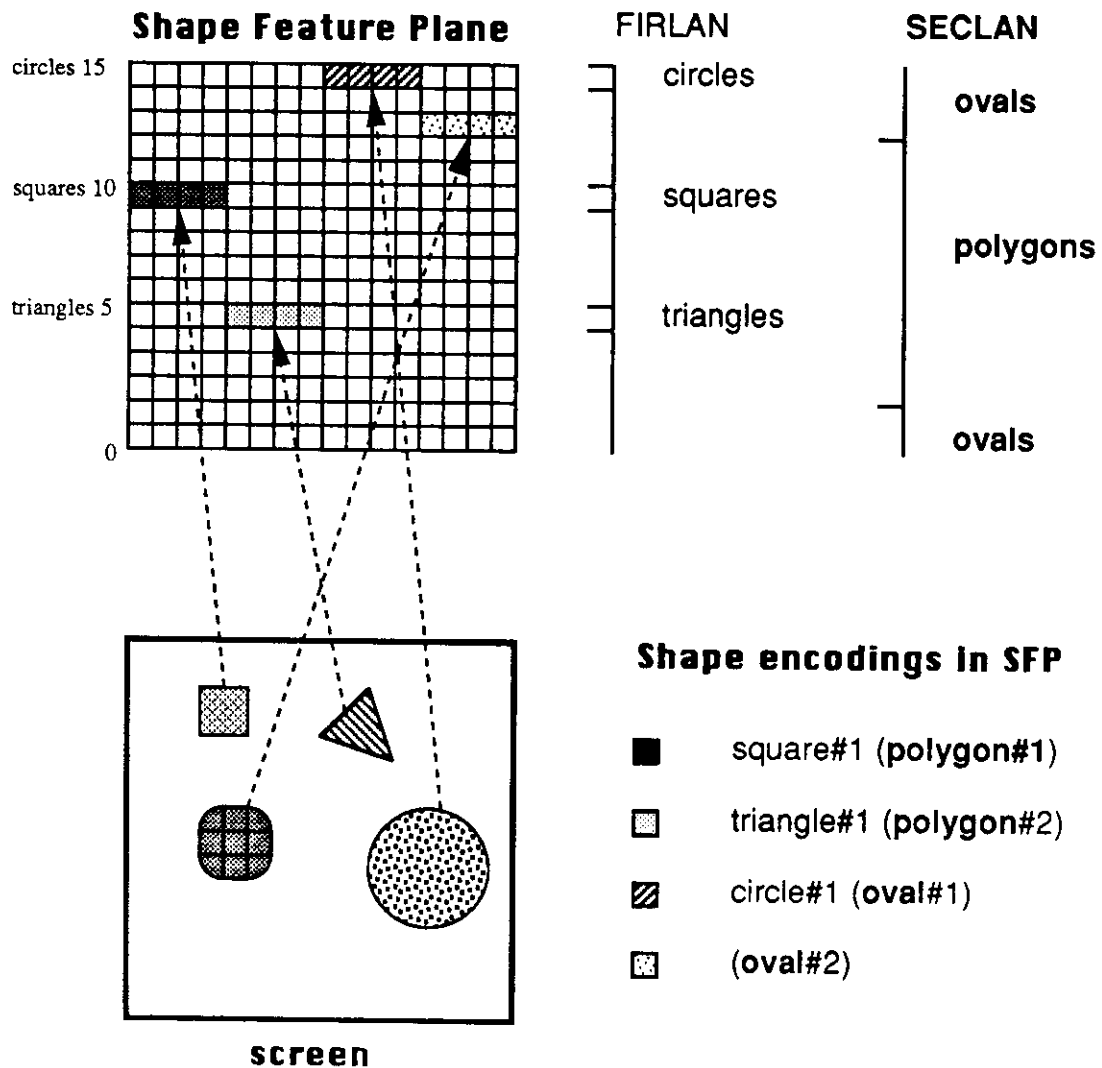


Figure 3.3: Shape Feature Plane (SFP)

Representations of three objects in the Shape Feature Plane: a circle, a square and an oval. The active neurons corresponding to each individual shape are represented by different levels of gray. The different grey levels represent the fact that the oscillatory activities of the neurons representing different shapes have different phases.

3.2.2 Size Feature Extractor

The siZe Feature Extractor (ZFE) in DETE is a procedural module which computes the absolute size of each object on the retina. The size is measured in number of pixels covered by the object. The sizes of the objects which DETE can recognize are in the range of 3 to 64 pixels in their longest dimension. Figure 3.1 shows examples of three objects, one of each kind (i.e. a circle, a square

and a triangle). Each of these objects has the smallest size which DETE can recognize. The size of each object is mapped onto a size Feature Plane (ZFP). The ZFP is a square array of dimensions 16*16. The ZFE was designed to categorize the size of each object into one of 16 different size groups. For instance, into size group 1 fall all objects of size 4 pixels to 256 pixels. All objects of sizes from 257 to 512 fall into size group 2, etc. The spatial organization of the different size groups in the ZFP is raster linear (Figure 3.4). There is one row of pixels in the ZFP for each size group. This choice of representation of the various sizes above, i.e. smaller sizes at the bottom rows of the ZFP changing gradually to larger sizes at the top rows, is important for the process of reasoning about sizes (as will be illustrated in section 11.5.1). The size of each individual object is represented as 4 contiguous active pixels (again to provide robustness) in the corresponding group. In other words, 4 objects belonging to the same size group can be represented simultaneously in the ZFP. All objects that fall in the same size group are considered to have the same size (see Appendix B.1.2 for code).

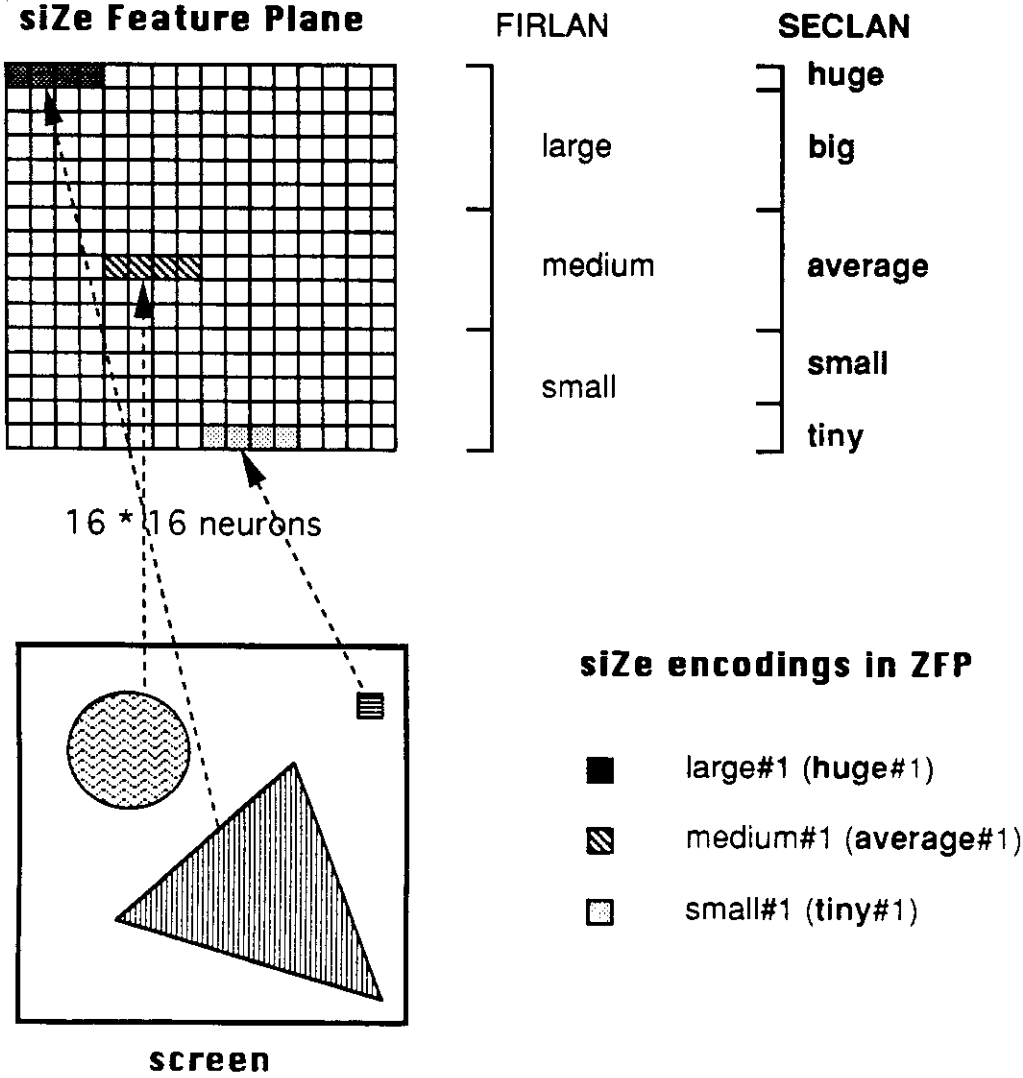


Figure 3.4: size Feature Plane (ZFP)

Representation of three different sizes in the Size Feature Plane. FIRLAN refers to them as "small", "medium", and "large", while SECLAN is more precise and calls the small object "tiny". The active neurons corresponding to each individual size are again represented by different levels of grey for each distinct phase.

While the mapping of the size of each object to the ZFP is done automatically by the ZFE, the verbal input in FIRLAN or SECLAN can provide a "meta-classification" of the objects that DETE attends to, in terms of their sizes. For instance, in FIRLAN all objects that fall into size groups 1 to 5 are called "small", within size groups 6 to 11 -- "medium size", and within size groups 12 to 16 -- "large". SECLAN, however, is more precise and instead of using only three words for the various sizes it supplies also "tiny" for objects with sizes in groups 1 and 2, and "huge" for objects in group 16 (see Figure 3.4).

3.2.3 Color Feature Extractor

The Color Feature Extractor (CFE) is a procedural module which does the following transformations of the visual input coming through the retina:

(1) Transforms each contiguous blob of pixels (contiguous = pixels that are adjacent and/or diagonal) that have the same color to a single active pixel with the same color located at the Center of Gravity (CG) of the blob. This transformation reduces each color blob to a single color pixel corresponding to that blob. Thus, DETE cannot currently recognize multi-colored objects. The CFE is designed to recognize 16 different colors.

(2) Maps each color pixel representing the CG of a blob (without preserving the info about the location of the CG) to 4 contiguous active pixels (redundancy of coding) in the Color Feature Plane (CFP). These pixels are randomly located in one of the sixteen color-banks (a row of neurons in the CFP). The mapping is such that neurons representing two different objects in the same color-bank do not overlap. Therefore, DETE can recognize up to 4 individual objects of the same color that appear simultaneously on the retina (see Appendix B.1.3 for code).

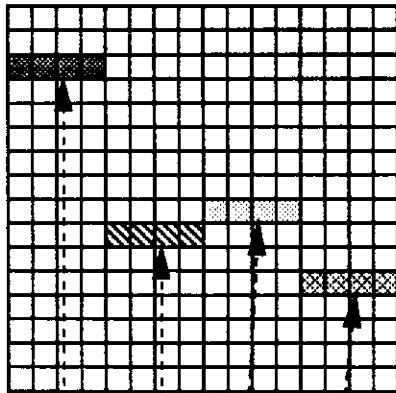
Encoding of the object's color is done in the CFP, and is similar to the encoding in the size feature plane. Here, instead of sizes as number of pixels, a rainbow of colors is encoded (Figure 3.5). Each of these colors is represented within one row of neurons (a color bank) in the CFP. In all of the visual scenes presented to DETE the black color is used as a background, and objects may be of any of the other available colors (only monochromatic objects).

Similarly, as in the cases of the SFP and ZFP, here the verbal input can also provide a "meta-classification", which is learned. FIRLAN, for instance, partitions the CFP into 8 areas corresponding to the colors white, red, orange, yellow, green, blue, purple, and black, while as a result of learning SECLAN, DETE divides all colors into only warm_colors and cold_colors.

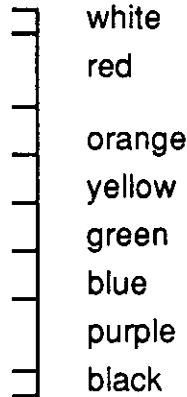
Figure 3.5: Color Feature Plane (CFP)

The colors of four different objects are mapped onto the in the Color Feature Plane. FIRLAN calls these objects: "red", "green", "green" and "blue", while SECLAN calls them "warm_color", "warm_color", "cold_color", and "cold_color". The active neurons (corresponding to each individual color) are illustrated by different patterns of gray on the figure.

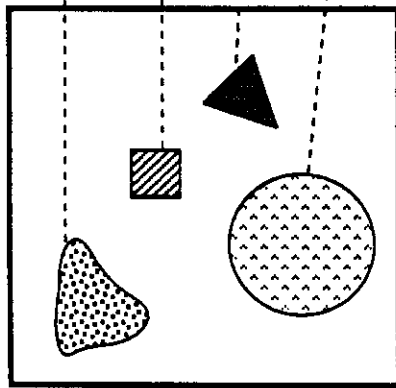
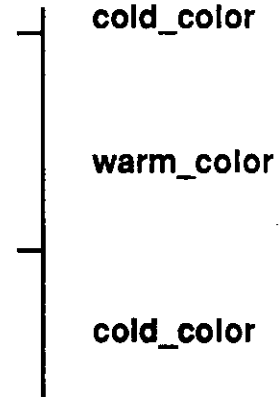
Color Feature Plane



FIRLAN



SECLAN



screen

Color encodings in CFP

- red#1 (warm_color#1)
- ▨ green#1 (warm_color#2)
- green#2 (warm_color#3)
- ▩ blue#1 (cold_color#1)

3.2.4 Location Feature Extractor

The representation of object location in DETE is straightforward. The location of each object on the retina is represented by the position of its center of gravity (CG) in a Location Feature Plane (LFP). The Location Feature Extractor (LFE) is a procedural module that calculates the CG of each object which is in the retina with respect to a coordinate system connected to the retina, (i.e. in retinal coordinates) (Figure 3.6). Then it calculates the absolute coordinates of the CG (i.e. its coordinates with respect to the Visual Screen) by a vector addition of the retinal coordinates of the object and the coordinates of the retina itself. The absolute location of the object is mapped onto the LFP with each object represented by 4 active pixels (redundancy of coding) and the CG is the lower left-hand pixel of these square pixels. The type of mapping is retinotopic, i.e. the topographic relation between objects in the VF is preserved when their locations are mapped on the LFP (see Appendix B.1.4 for code).

The verbal input can provide a meta-classification within this feature plane. For instance, FIRLAN partitions the feature space into 25 areas using words such as “center”, “near”, “far”, “above”, “below”, “left”, and “right” and phrases generated by combining these words (e.g., “far

left"). SECLAN, however, partitions it only into 2 areas using words such as "middle" and "periphery".

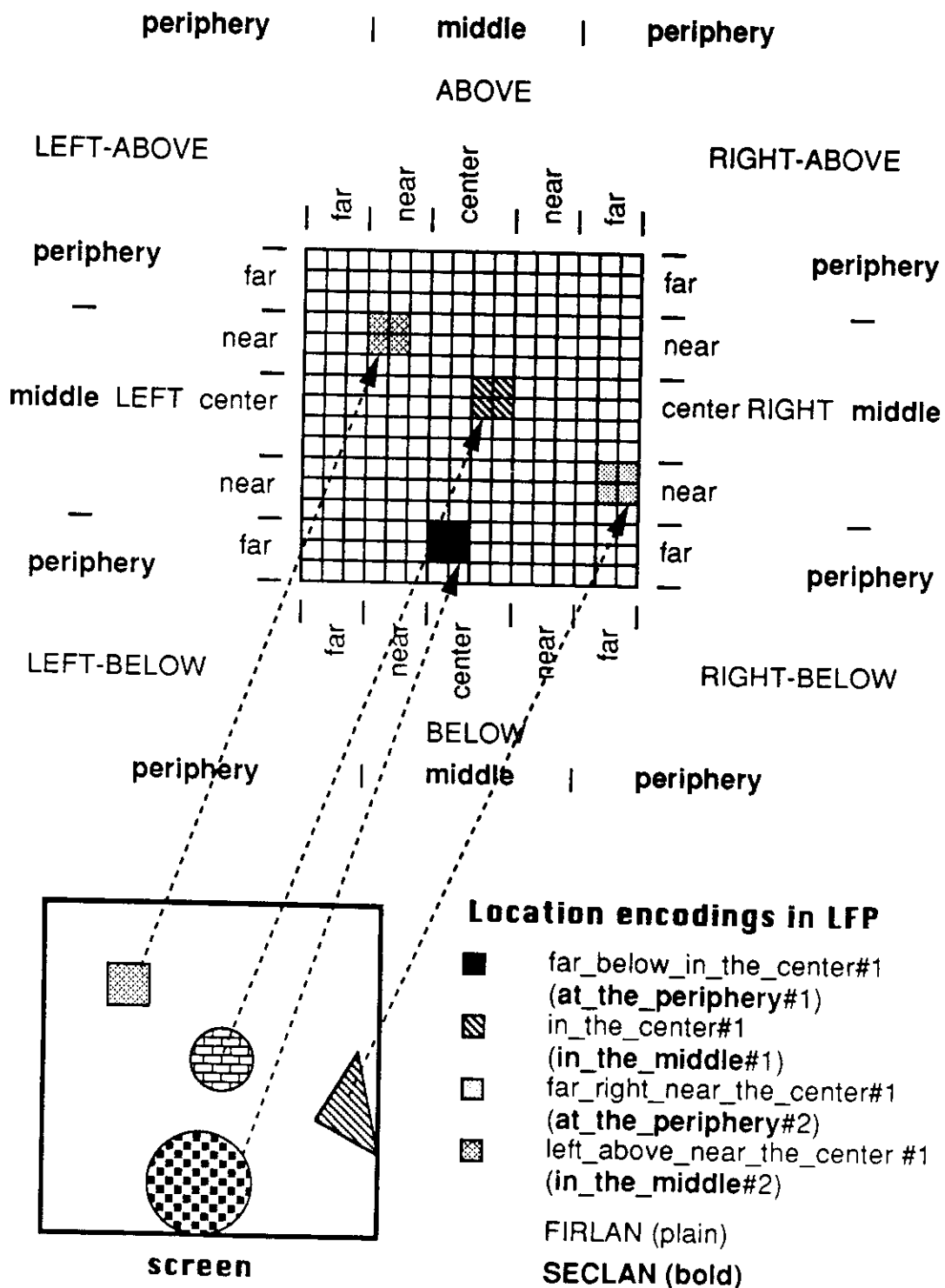


Figure 3.6: Location Feature Plane (LFP)

Representation of three objects that appear simultaneously on the retina in different locations. The active neurons corresponding to each individual location are again illustrated by different levels of gray. The gray levels also represent the fact that the oscillatory activities of the neurons representing the different sizes have different phases.

3.2.5 Motion Feature Extractor

DETE can analyze motion either with respect to the Visual Screen (I call this “absolute motion”), or with respect to the retina -- the Visual Field (i.e. “relative motion”). The Motion Feature Extractor (MFE) is implemented as a procedural module which takes input along two channels (see Appendix B.1.5 for code). The first channel carries information about the motion (in terms of velocity and direction) of the retina over the Visual Screen. The second channel carries information about the motion of the objects that are within the Visual Field. The MFE maps the motion of each object onto a Motion Feature Plane (MFP). To generate the motion representation of an object, the MFE uses information from both channels. The output representation of motion on the MFP is with respect to the Visual Screen (i.e. it is “absolute motion”).

The set-up described above allows for two extreme situations to occur if there is only one object on the retina. First, the retina can be stationary while the object moves within the retinal boundaries. Second, the retina can be tracking the object (i.e. they are both moving the same way). In this case the retinal image of the object is stationary. In both cases, however, the representation that the MFE forms on the MFP is the same.

The MFP is a square array of neurons with dimensions 16*16 (Figure 3.7). The motion of each object (which is within the Visual Field) is represented in this plane by a vector \mathbf{O} (Object motion, i.e. the motion of the object with respect to the VS or “absolute motion”). \mathbf{O} has origin in the center of the MFP, with its direction corresponding to the direction of motion of the object, and its length proportional to its velocity of movement. The Object motion is calculated at each time step as a vector sum of the Retinal motion (\mathbf{R}) (i.e. the motion of the object with respect to the retina) and the Visual Field Motion (\mathbf{V}) (i.e. the motion of the retina with respect to the VS). In other words, $\mathbf{O} = \mathbf{R} + \mathbf{V}$. Notice that for the purpose of calculating their motion, objects are regarded as points in space (i.e. represented by their center of gravity -- CG) and rotation is disregarded. \mathbf{V} is computed by the MFE at each time step as a difference between the current and the previous position of the center of the retina with respect to the VS. \mathbf{R} is computed similarly as the difference between the current and the previous position of the center of gravity of the object with respect to the retina -- VF.

The motion of an Object (\mathbf{O}) is represented on the MFP by 4 simultaneously active (oscillating) neurons which are clustered together and the phases of their oscillations are the same, i.e. their oscillations are phase locked.

As a result of the association of verbal descriptions of motion with the physical representation of motion in the MFP, the possible motions within the model become effectively clustered, i.e. a given language “carves up” the MFP into different categories or classes. FIRLAN, for instance, classifies motions (in terms of their velocity), into three groups, using words such as: “still”, “slow”, and “fast”. In terms of direction of motion FIRLAN classifies motions by using the words North,

East, West, and South and phrases such as North-East, North-West, South-East, and South-West. SECLAN, on the other hand, only classifies all objects as "still" or "moving".

It is important to note that the finer the classification, the smaller the number of objects that DETE can represent in each individual category simultaneously.

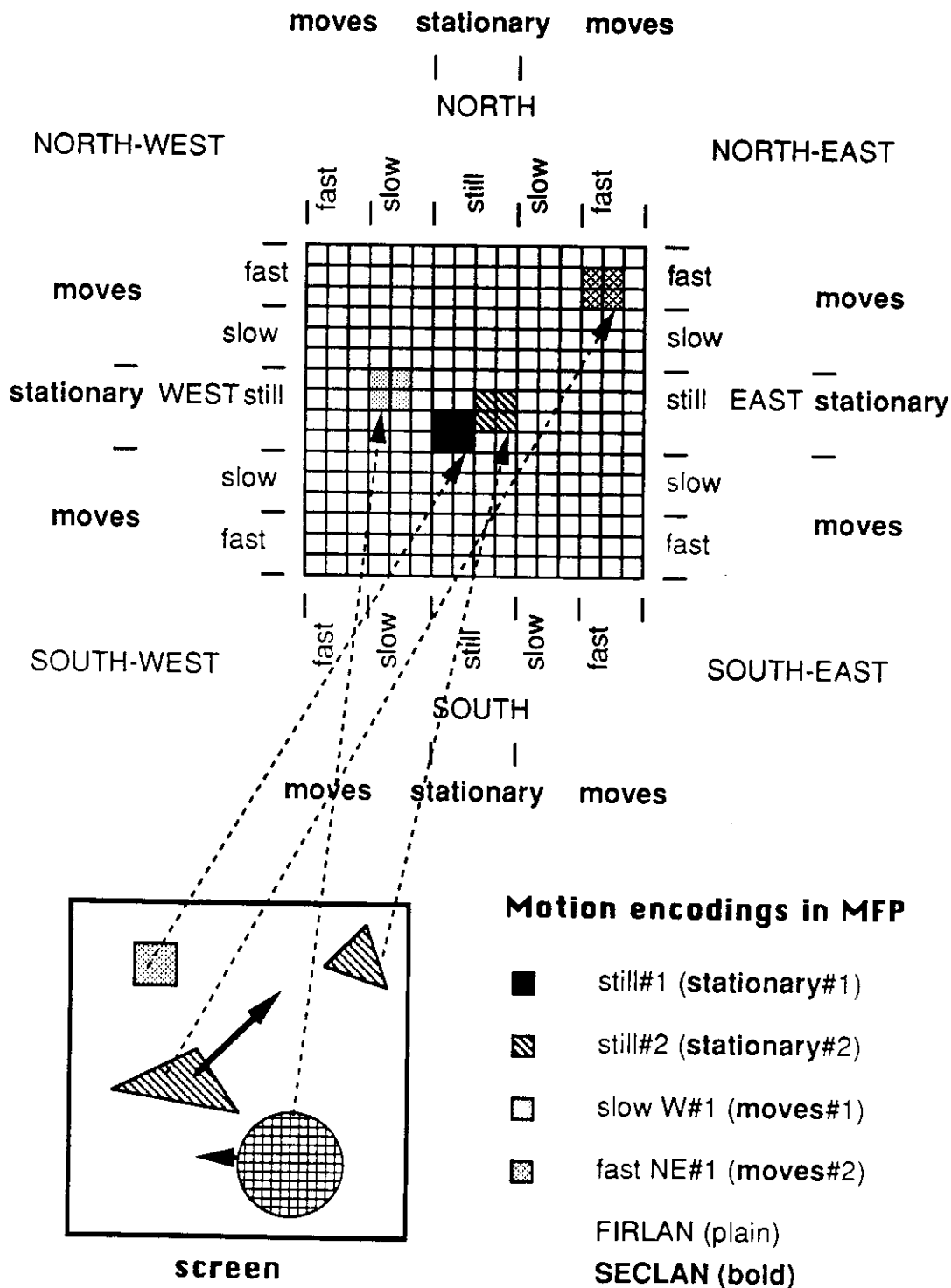


Figure 3.7: Motion Feature Plane (MFP)

Simultaneous representation of the motions of four different objects on the Motion Feature Plane. According to FIRLAN, there are two "still" objects, one moving slowly west, and another moving fast north-east. According to SECLAN, however, there are two "still" objects and two "moving" objects. The active neurons corresponding to each individual motion are again indicated by different levels of gray.

In summary, there are advantages and disadvantages associated with the use of maps. For instance, a disadvantage is that within each map the representations are localist (4 contiguous pixels) and are formed by heuristic procedures so only a finite number of feature values (e.g., shapes, colors, etc.) can be represented. However, the total representation of an object is distributed across 5 feature maps, thus DETE can currently discriminate # of objects (= size# x shape#..), and the number of concepts and events grows combinatorically.

4 THE VERBAL SYSTEM

The verbal input to DETE is provided as text that can be typed in directly via a keyboard or read in from a text file. A Text Encoder takes this input and encodes it into a sequence of **gra-phonemes**. Gra-phonemes are the basic representational units of DETE's verbal input. The name "gra-phoneme" was chosen because each of these representational units has both orthographic and phonemic features. There are 26 different gra-phonemes, one for each letter (grapheme) in the English alphabet. That is why "gra" appears in the name. The one-to-one correspondence between letters and gra-phonemes is what allows DETE to process textual input. However, the ability to read text is acquired by children after they learn spoken language. I consider this fact to be of significance and for this reason I designed DETE's representation of the verbal input to contain also some characteristics of spoken language. That is why "phonemic" appears in the name. A standard representation for spoken language is the phonetic representation. Only 44 phonemes are needed to code all words in the English language and 55 phonemes are sufficient to represent virtually all the words in all spoken languages (Marslen-Wilson, 1980). DETE does not use a complete phonemic representation but only a subset of 26 phonemes (i.e. the most frequently used phonemes corresponding to letters of the English alphabet. This set is sufficient also in representing a subset of Spanish, and Japanese textual input.)

Each gra-phoneme has a spatial and temporal structure. At each time step of DETE's mechanism, which is called a "B-cycle", a gra-phoneme is represented as a binary pattern over a bank of 64 verbal units. Each pattern has about 10% density (or 6 bits out of 64 are 1 while the others are 0). Each pattern is clamped over the verbal units for 5 B-cycles. This duration was chosen to correspond roughly to 50 ms -- the minimal time period within which a phoneme can be recognized by a human listener. This choice of duration is based on the observation that an average English speaker pronounces about 3 words per second (i.e. 330 ms/word) (Altmann and Shillcock, 1986). (A skilled reader on the other hand can recognize more than 5 words per second (Rayner and Pollatsek, 1987)). We can also assume that each word is composed on average of 7 phonemes (i.e. 50 ms/phoneme) (Altmann and Shillcock, 1986). Each word is represented as a sequence of gra-phonemes.

The choice of the binary patterns corresponding to the individual gra-phonemes is loosely based on the acoustic representation of various speech sounds of English. Speech research has revealed that specific aspects of the acoustic signal (acoustic cues) are relevant for the listener for perceiving the sounds of speech. To illustrate how such findings have influenced our choice of representation, I review briefly the acoustic correlates of English sounds. Vowels and consonants have different acoustic cues. For vowels, the first acoustic cue is the frequency positions of the first three formants. This cue is sufficient for most listeners to identify them. It is important to notice that the formant frequency relations that specify the vowels of English are relational rather than absolute. The mean values of the first three formants (F1, F2, and F3) of the vowels of American English are shown in Figure 4.1. The second acoustic cue is the difference in vowel duration (cf. Miller 1981, for a review). Typical duration of English vowels is between 180 to 330 ms (Peterson and Lehiste, 1960). Examples of some vowels with different durations are: (a) short vowels, e.g., [I] as in *hid* (180 to 200 ms), (b) medium duration vowels, e.g., [u] as in *who'd* (240 to 260 ms), (c) long

duration vowels, e.g., [æ] as in *had* (330 ms). Many of the longer vowels in English are diphthongized.

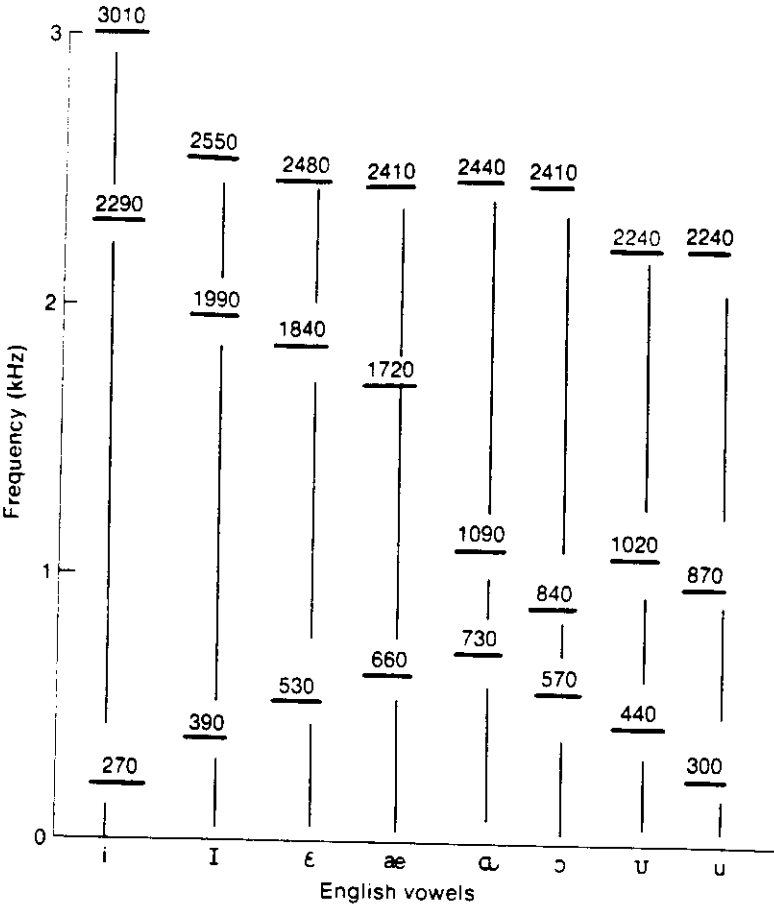


Figure 4.1: Formant frequencies of English vowels

Mean values of formant frequencies for adult males of vowels of American English measured by Petersen and Barney (1952). (Reproduced with permission from Lieberman and Blumstein, 1988).

The acoustic cues for consonants in English are more diverse than that for vowels. This is due to the diversity of the ways in which consonants are produced, e.g., stop consonants, nasal consonants, liquid consonants, glide consonants, and fricative consonants.

1) *Stop* [p t k b d g]. -- The basic acoustic cues for stop consonants are: (1) release burst (generated after a rapid release of a complete closure of the vocal tract). Its duration is typically 5-15 ms; (2) rate and duration of formant transitions. Their duration is in the order of 20-40 ms. The total duration of such consonants is between 25 and 55 ms. Figure 4.2. shows the formant transitions and bursts for the syllables [ba da ga pa ta ka].

2) *Nasal* [m n ŋ]. -- The basic acoustic cue for nasal consonants is a nasal "murmur" -- a low frequency sound (250 Hz) produced prior to release of oral closure. The average duration of nasal consonants is similar to that of stop consonants.

3) *Liquid* [l r] and *glide* [w y]. -- These consonants are characterized by their onset frequencies and duration of their formant transitions (about 40 ms). If their duration is smaller than 30 ms they are easily misinterpreted as stop consonants.

4) *Fricative* [f θ s ʃ δ z ʒ]. -- The main acoustic cue for the fricative consonants is the presence of aperiodic noise in their spectrum (Delattre et al., 1962) with a duration of 20 to 100 ms. The overall amplitude of this noise and the distribution of the spectral peaks contribute to the perception of different fricatives.

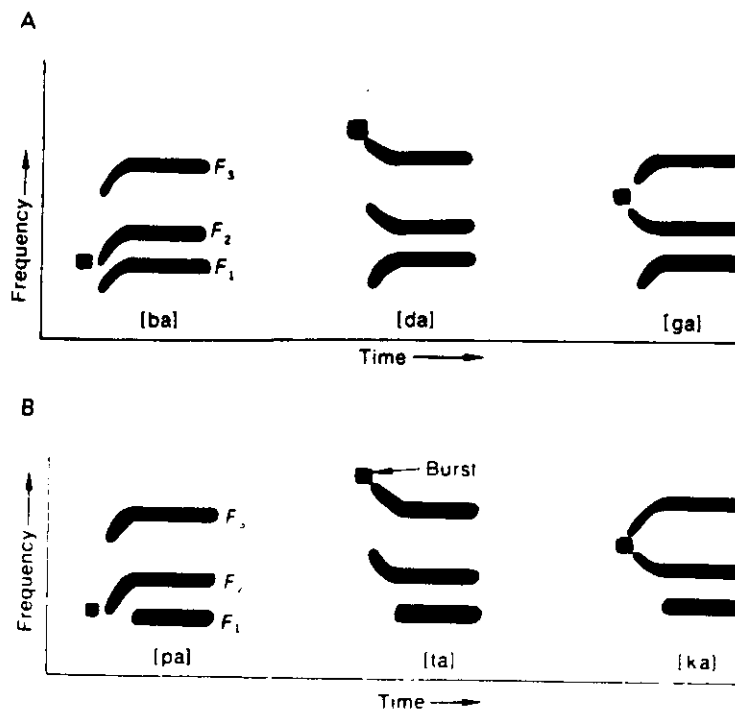


Figure 4.2: Formant transitions

Formant transitions and bursts for the syllables [ba da ga pa ta ka]. (Reproduced with permission from Lieberman and Blumstein, 1988).

The frequency range / bit mapping in the gra-phonemic representation is shown in Table 4.1. Each location (loc) in the verbal bank represents a sound frequency window of 40 Hz. For instance, loc 1 is set to 1 when in the frequency spectrum of a given phoneme there is a formant with an average frequency in the range of 270 to 310 Hz. To provide some robustness of representation of each formant, the next closest loc to the particular formant frequency is also set to 1. This encoding scheme also ensures that the total 1-bit-density of each pattern representing a gra-phoneme is about 10% (3 formants \times 2 = 6 locations).

A numeric representation of the 26 gra-phonemes in terms of the frequencies (in Hz) of the first three formants (F1, F2, F3), and the corresponding bits set to 1 in the 64 bit long vector corresponding to each gra-phoneme, are shown in Table 4.2. While the formant representation of the vowels was straightforward, the choice of formants to represent the consonants is somewhat

arbitrary. For instance, the formant frequencies of the stop consonants were chosen to be the initial values of the formant transitions (see Figure 4.2).

| loc | Hz | loc | Hz | loc | Hz | loc | Hz | loc | Hz | loc | Hz | loc | Hz |
|-----|-----|-----|------|-----|------|-----|------|-----|------|-----|------|-----|------|
| 1 | 270 | 11 | 670 | 21 | 1070 | 31 | 1470 | 41 | 1870 | 51 | 2270 | 61 | 2670 |
| 2 | 310 | 12 | 710 | 22 | 1110 | 32 | 1510 | 42 | 1910 | 52 | 2310 | 62 | 2710 |
| 3 | 350 | 13 | 750 | 23 | 1150 | 33 | 1550 | 43 | 1950 | 53 | 2350 | 63 | 2750 |
| 4 | 390 | 14 | 790 | 24 | 1190 | 34 | 1590 | 44 | 1990 | 54 | 2390 | 64 | 2790 |
| 5 | 430 | 15 | 830 | 25 | 1230 | 35 | 1630 | 45 | 2030 | 55 | 2430 | | |
| 6 | 470 | 16 | 870 | 26 | 1270 | 36 | 1670 | 46 | 2070 | 56 | 2470 | | |
| 7 | 510 | 17 | 910 | 27 | 1310 | 37 | 1710 | 47 | 2110 | 57 | 2510 | | |
| 8 | 550 | 18 | 950 | 28 | 1350 | 38 | 1750 | 48 | 2150 | 58 | 2550 | | |
| 9 | 590 | 19 | 990 | 29 | 1390 | 39 | 1790 | 49 | 2190 | 59 | 2590 | | |
| 10 | 630 | 20 | 1030 | 30 | 1430 | 40 | 1830 | 50 | 2230 | 60 | 2630 | | |

Table 4.1: Frequency range/loc mapping in the gra-phonemic representation

| #gra-pho | F1 | F2 | F3 | bit-1 | bit-2 | bit-3 | |
|----------|----|------|------|-------|-------|-------|----|
| 1 | a | 730 | 1090 | 2440 | 12 | 21 | 55 |
| 2 | b | 470 | 790 | 2030 | 6 | 14 | 45 |
| 3 | c | 990 | 1570 | 2160 | 19 | 33 | 48 |
| 4 | d | 670 | 1670 | 2790 | 11 | 36 | 64 |
| 5 | e | 530 | 1840 | 2480 | 7 | 40 | 56 |
| 6 | f | 370 | 1280 | 2630 | 3 | 26 | 60 |
| 7 | g | 630 | 1750 | 2350 | 10 | 38 | 53 |
| 8 | h | 780 | 1440 | 2730 | 13 | 30 | 62 |
| 9 | i | 390 | 1990 | 2550 | 4 | 44 | 58 |
| 10 | j | 1000 | 1480 | 2350 | 19 | 31 | 53 |
| 11 | k | 790 | 1790 | 2310 | 14 | 39 | 50 |
| 12 | l | 430 | 1090 | 2120 | 5 | 21 | 47 |
| 13 | m | 520 | 1280 | 2670 | 7 | 26 | 61 |
| 14 | n | 600 | 1050 | 2550 | 9 | 20 | 58 |
| 15 | o | 570 | 840 | 2410 | 8 | 15 | 54 |
| 16 | p | 750 | 870 | 2030 | 13 | 16 | 46 |
| 17 | q | 920 | 1390 | 1930 | 17 | 29 | 42 |
| 18 | r | 960 | 1600 | 2270 | 18 | 34 | 51 |
| 19 | s | 760 | 1800 | 2040 | 13 | 39 | 45 |
| 20 | t | 830 | 1710 | 2750 | 15 | 37 | 63 |
| 21 | u | 300 | 870 | 2240 | 2 | 16 | 50 |
| 22 | v | 550 | 1610 | 2550 | 8 | 34 | 58 |
| 23 | w | 770 | 1120 | 2430 | 13 | 22 | 55 |
| 24 | x | 390 | 1050 | 2260 | 4 | 20 | 50 |
| 25 | y | 680 | 1200 | 2560 | 11 | 24 | 58 |
| 26 | z | 780 | 1690 | 2070 | 14 | 36 | 46 |

Table 4.2: Representations of the 26 gra-phonemes

Formants frequencies and corresponding bits used in the representation of the 26 gra-phonemes used in DETE.

In developing DETE's gra-phonemic representation of the verbal input I have intentionally compromised a purely acoustic representation in four ways:

- 1) *Number of gra-phonemes.* -- I have reduced the number of representational units -- gra-phonemes from the number of phonemes (44) to the number of graphemes (26 in the English alphabet). The reason for this choice was to be able to handle textual input. A shortcoming is that for DETE the verbal input "sounds" as if it is read by a person who knows only one sound for each letter of the alphabet and is using only this knowledge to string sound together while reading words.
- 2) *Duration.* -- Instead of using different durations of phonemes (vowels and consonants) as revealed by acoustic research on speech, I have chosen to use a constant duration of 25 ms (5 B-cycles in DETE) for both vowels and consonants. As a result, each letter is represented by a sequence of 5 64-bit patterns, where each pattern is the same.
- 3) *Formants.* -- Instead of using the known dynamics of the formant changes for the individual consonants I use stationary patterns.

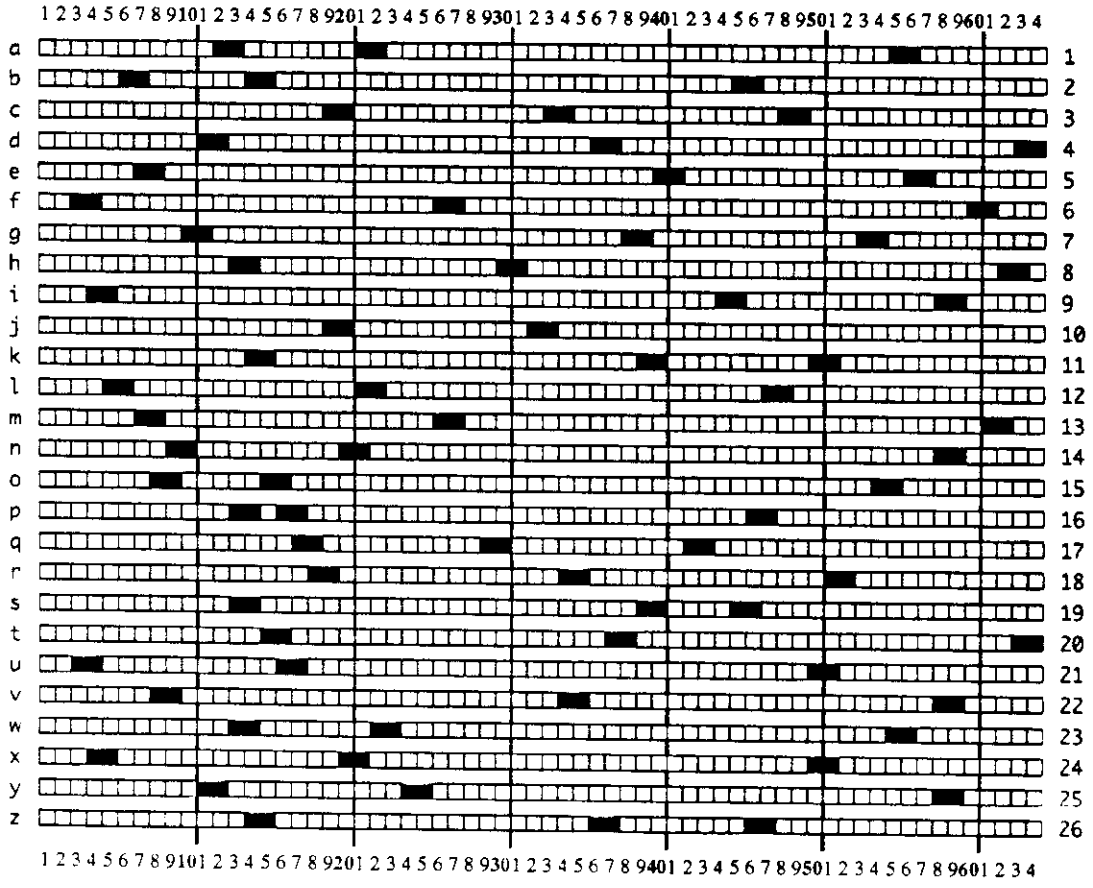


Figure 4.3: Gra-phonemes

Gra-phonemic representation of the letters of the English alphabet. To simplify the drawing, each letter of the alphabet (shown in the right column) corresponds to one 64-bit pattern. However, in the gra-phonemic representation, each gra-phoneme is encoded as a sequence of 5 64-bit patterns which are all the same.

4) *Segmentation.* -- In spoken language people usually do not make significant pauses between words. However, in the gra-phonemic representation, in order to make it easier for DETE to recognize the word boundaries (which are apparent in a textual input) I have introduced pauses between individual words. The duration of each pause is 5 time cycles (i.e. 5 64-bit patterns) and during that period the input to the verbal units is 0. In other words, the segmentation of the verbal input into words is provided externally. Figure 4.3 shows the gra-phonemic representation of the letters of the English alphabet.

As a result of this representational choice, words are not individual patterns, as they are usually treated in localist connectionist representations, but they are time sequences with a fully distributed representation in space (between the units of the verbal bank). An example of the gra-phonemic representation of the sentence "Red ball hits the left wall" is shown in Figure 4.4.

"Red ball hits the left wall"

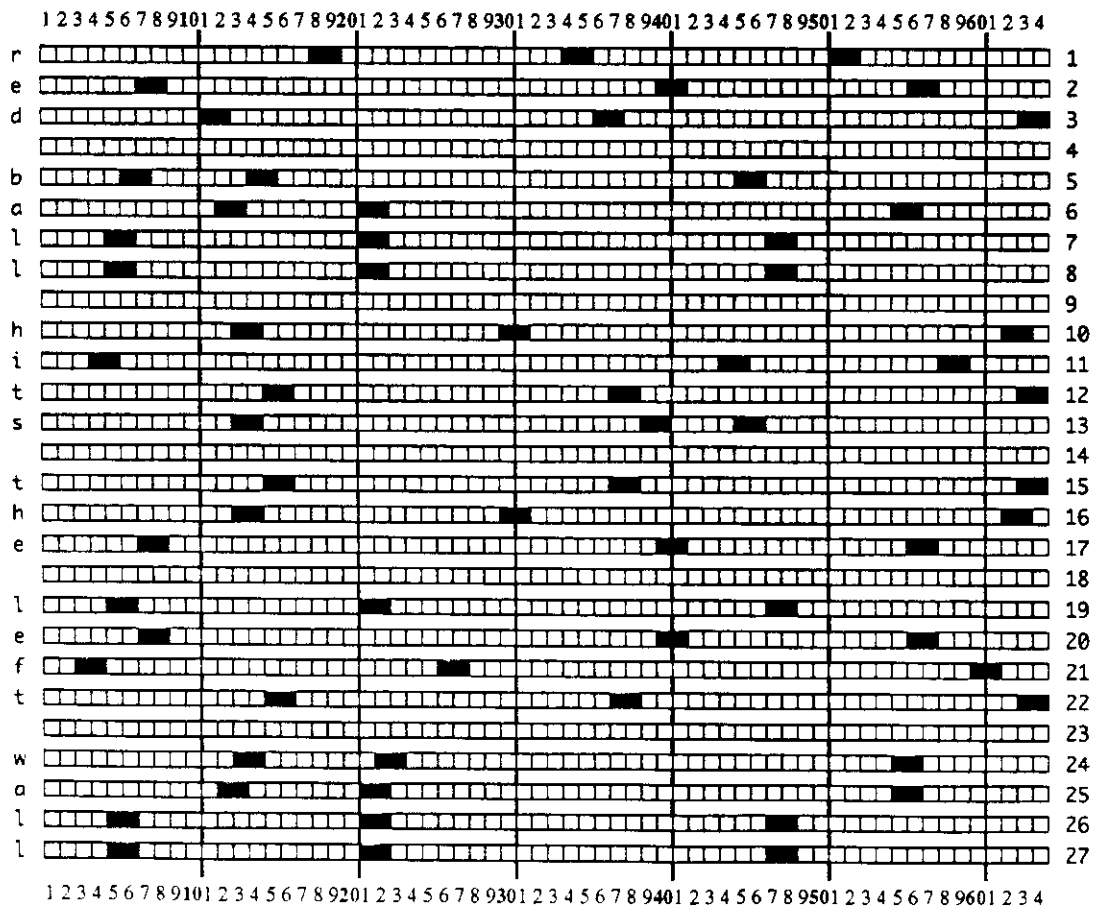


Figure 4.4: Gra-phonemic encoding of a sentence

Gra-phonemic representation of the sentence "Red ball hits the left wall". For simplicity, each gra-phoneme is shown as only one 64-bit pattern while in DETE it is composed by 5 such patterns.

For the range of experiments performed so far with DETE, the patterns representing the individual gra-phonemes could have been chosen at random and DETE's performance would not have suffered. While a simplification, the current gra-phonemic representation allows DETE to process the internal structure of words (e.g., inflections on verbs and suffixes on adjectives).

The acoustically based representation of the gra-phonemes is designed to allow a set of more sophisticated future experiments in which DETE can accept verbal input which contains prosodic inflections. Such inflections can be reflected in the "pitch" with which individual words are pronounced. A pitch change in the verbal input is represented as a shift of the gra-phonemic formants along the frequency scale. Prosodic inflections in the verbal input could be used by DETE for instance, to make a distinction between an interrogative and declarative verbal input (e.g., "Where is the red ball?" vs "The wall where the red ball bounced.")

4.1 Verbal input segmentation

To generate the gra-phonemic representation, a Word Encoding Mechanism (WEM) mechanism (upper left of Figure 2.4) goes through each sentence letter-by-letter and word-by-word (see Appendix B.5.1). Each letter is represented by the corresponding gra-phoneme. Words are sequences of gra-phonemes. Spaces (pauses) between words and sentences are represented as 0 patterns. The output of the WEM mechanism is further passed to the verbal component of DETE's memory which is called the "verbal bank".

4.2 Verbal output

DETE generates verbal output in the form of text. Internally the verbal output is represented as a stream of activity in the verbal bank. A Verbal Activity Decoder (VAD) (see Figure 2.4, and Appendix B.5.2) monitors the activity of the verbal memory bank and converts it into a gra-phonemic representation. This representation is further converted into a sequence of letters. To generate the gra-phonemic representation, the output activity at the verbal bank must be matched against all possible gra-phonemic representations during decoding. A simple solution to this pattern matching or classification problem is to bit-multiply the output pattern with each of the 26 gra-phonemic patterns (in-parallel); then sum along the width (64 bits) of the vectors and take the position of the maximum sum as an index into the alphabet. This procedure results in picking the gra-phonemic representation of a letter which is most similar to the output pattern. Thresholding of the maximum sum allows for the introduction of "silence" between words. An example of the generation of verbal output from a gra-phonemic representation is given in Figure 4.5.

Pattern "*" -- Output from the Verbal Memory Bank



Results of bit-wise multiplication of pattern * with the gra-phonemes Scores

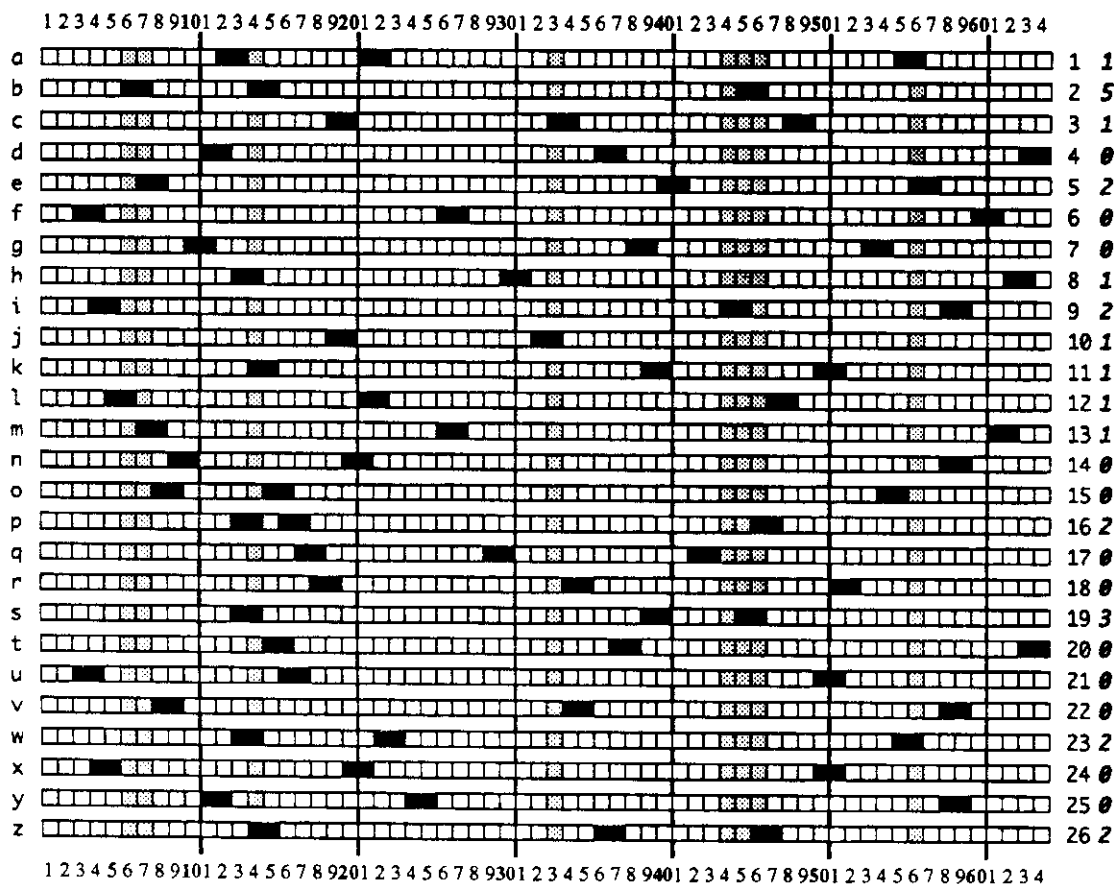


Figure 4.5: Decoding gra-phonemes -- verbal output

A pattern generated by the verbal memory bank is input to the VAD mechanism and is "tested" for similarity against the set of all 26 gra-phonemic patterns. The results of the bit-by-bit multiplication of the input pattern by each of the gra-phonemes are shown by two different shades of gray. The bits shaded light gray have value 0 while the bits shaded dark gray have values of 1. The score for each gra-phoneme is computed as the sum of the dark gray bits. The given pattern matches best (maximum score = 5) the "b" gra-phoneme.

5 TEMPORAL RELATIONS IN DETE

DETE's operation is synchronized by a clock. The basic cycle of this clock is called a B-cycle. The states of all neural elements in the system are updated synchronously at each B-cycle.

A major assumption in DETE is that the verbal and visual inputs come in time "chunks" of different duration. A B-cycle is the shortest chunk which I will also call a level-0-chunk. A sequence of level-0-chunks forms a level-1-chunk. Similarly, several level-1-chunks can be grouped to form a level-2-chunk. The highest level of this temporal hierarchy in DETE is a level-3-chunk which is a sequence of several level-2-chunks.

The choice of chunks at various levels is not arbitrary. The existence of a level-0-chunk corresponds to the minimal period of firing of neurons in the cortex (about 10 msec). A level-1-chunk corresponds to a phoneme in the verbal modality. I call this time duration a chunk since, in general, the characteristic features of the signal (e.g., frequency power spectrum) within this period remain unchanged, whereas the features of the signal in two different chunks are different. The shortest phoneme (e.g., a consonant such as "t" or "b") which humans can recognize is about 30 ms (Pöppel, 1988). A level-2-chunk in the verbal domain is a word, whereas in the visual domain this is a gaze (i.e. the duration of an eye fixation at a point of the visual scene between two consecutive saccades). A level-3-chunk in the verbal domain is a sentence. In the visual modality it corresponds to a "gestalt"?

While I have not attempted to map, in a one-to-one manner, the temporal characteristics of various processing stages in DETE to those observed in the human nervous system (i.e. DETE does not operate in real time), an effort has been made to preserve the ratios of these time characteristics between the visual, verbal, motor, attentional and memory modalities. The information about the time relations in humans have been compiled from various sources. Most of these data comes from psychophysical (reaction time) and evoked potential studies of the visual and auditory systems, from psycholinguistics, and from cellular electrophysiology. Table 5.1. summarizes the corresponding relations expressed in milliseconds or Hz in the human brain (left-hand side) and compares them with the basic temporal relations within DETE (right-hand side of table).

The temporal ratio of the verbal and visual input in DETE (i.e. how long a word persists in the input, as compared to the duration of a visual image) is set up such that it is close to the average ratio observed in humans. For instance, in humans the rate of reading and talking is about 3 to 5 words per second while in order to experience a smooth visual perception, the visual refreshment rate must be at least 25 frames per second. In other words, on average the ratio is 5 visual frames per word. Preserving such temporal relationships in a model is important because comprehension of language and visual images is a dynamic process and disambiguation of sentence meaning has a dynamic nature. For instance, the utterance "Give me the can ... opener" elicits different concepts in our mind depending on how close "opener" comes after "can" in the utterance. If the two words are uttered close to each other they are understood as "the instrument for opening cans". However, if there is a long pause between them, the first word "can" elicits the concept of container in our minds, and the second word elicits a separate concept.

| chunk level | period / frequency | | HUMAN | | comments | duration | DETE | | comments |
|-----------------|--------------------|---------------|--|--|---|----------|--------------------------------|--|--|
| | ms | Hz | auditory | visual | | | B-cycles | auditory | |
| | 1 | 1,000 | N/A | N/A | Action Potential | N/A | N/A | N/A | |
| level -0- chunk | 3-5 | 200-300 | fusion threshold | | maximal firing rate in cortex | 1 | N/A | N/A | attentional window (AW) |
| | 5-10 | 100-200 | | | | | | | |
| | 20-30 25±5 | 30-50 40±5 | | | | | | | |
| level -1- chunk | 30-50 40±10 | 20-30 25±5 | temporal order threshold -- (minimal time to recognize auditory event -- <u>phoneme</u> in a sequence) | temporal order threshold -- (minimal time to recognize visual event -- <u>image</u> in a sequence) | subjective perception of a single event ----- standard frame rate in movies | 5 | duration of a gra-phoneme | duration of a visual frame | maximal rate of sequence recognition in all modalities |
| level -2- chunk | 250-500 | 2-4 | average duration of a word | average duration of saccade | average duration of attention locked at object | 25-50 | average duration of a word | average duration of retinal fixation | |
| level -3- chunk | 3 sec | 0.3 | average duration of utterance 3*3=9 words | average duration of a scene in a TV ad | gestalt perception subjective NOW | 300 | average duration of a sentence | minimal duration of visual scene input | reset of STM & transfer between TPs |

Table 5.1: DETE's temporal hierarchy

5.1 Chunking the input

A more detailed examination of the behavioral and neurophysiological underpinnings of this segmentation hierarchy are presented below.

Psychophysical evidence suggests that the brain architecture imposes functional constraints for storage and retrieval of sequences of events. These constraints determine how humans perceive time. As it was pointed out by Pöppel (Pöppel, 1988), human temporal experiences are hierarchically organized. Five levels of temporal experiences (four according to Pöppel) can be distinguished on the basis of various experimental observations: (1) simultaneity/non-simultaneity (synchrony/asynchrony), (2) succession; (3) word-gaze duration, (4) duration of the subjective present (now), and (5) duration of an event (Pöppel, 1989). The following is a short description of each of these levels and how they fit in DETE.

5.1.1 Recognition of synchrony/asynchrony (level-0-chunks)

Experimental evidence has shown that objective asynchrony of two events is not sufficient for subjective asynchrony (Pöppel, 1988). The temporal characteristics of a subjective experience depend on the sensory modality through which these experiences are made available to our consciousness. For the auditory system, for instance, perception of asynchrony between two auditory stimuli (e.g., low and high pitch beeps of duration 1 msec each) is achieved only if the temporal difference between them (i.e. the Inter Stimulus Interval -- ISI) is longer than 3-5 msec.

In psychophysics, this time duration is known as the *auditory fusion threshold*. For the tactile system the *fusion threshold* is approximately 10 msec and in the visual system about 20-30 msec (see Table 5.1). On the basis of lesion studies it has been suggested that mechanisms in the peripheral nervous system (auditory, tactile, visual, etc.) are responsible for the existence of such thresholds, and the differences in fusion thresholds are probably due to the different transduction mechanisms in the different systems.

In the present version of DETE I do not wish to make a finer distinction between the level-0 chunking within the verbal and visual modalities and therefore I have chosen the time interval of 10 msec to correspond to one B-cycle -- a level-0-chunk. Our choice was based on the fact that the average firing rate of neurons in the cortex, where all sensory pathways lead, is about 100 Hz (i.e. about one action potential every 10 msec) (Kandel and Schwartz, 1985). One B-cycle has been chosen also to be the duration of the Temporal Attention Window (TAW) -- see section 7.1.

5.1.2 Recognition of temporal order (level-1-chunks)

Although the threshold for the perception of two auditory stimuli as asynchronous is 3 to 5 msec, on average an inter-stimulus interval of about 30 to 50 msec is necessary for recognition of temporal order (e.g., to be able to distinguish that the low pitch tone came first followed by the high pitch tone). It is a surprising and important fact, that this *temporal-order threshold* is essentially the same for all sensory modalities (Pöppel, 1988). The temporal order threshold provides the basis of level-1-chunking. In other words, in humans, the physiological limit of the sequence processing rate (i.e. how fast successive level-1-chunks, for instance, phonemes in an auditorily perceived word, can be processed/recognized) is 20 to 30 Hz. This suggests that only one (or very similar) cerebral mechanism(s) might be responsible for this phenomenon in the different sensory modalities. This possible mechanism might be responsible for establishing a temporal framework which enables inputs recorded by the various sensory systems to be correlated and subjectively perceived as a single event. Thus, the minimal time for a subjective discrimination of one level-1-chunk (an information processing step which I assume happens in the cortex) might be on the order of 30 to 50 msec. It is important, however, to clarify that the time for perceiving one event (from the stimulus onset to the moment when this stimulus is recognized) is certainly more than 50 msec. This is due to the fact that it takes time for the neural activation to travel and be processed along a particular sensory pathway (e.g., visual or auditory) before it is recognized in the cortex. This time may be on the order of 250 to 300 msec.

Consequently, in DETE I have chosen 5 B-cycles (corresponding to 50 ms) to be the length of a gra-phoneme (see section 2.3.1). 5 B-cycles have been chosen also to be the refreshment rate of the Visual Field.

5.1.3 Level-2-chunks (words)

Words in the auditory system and gazes in the visual system. The duration of the input is about 300 msec and it takes about 300 msec on average to process it. In the domain of Event Related Potentials (ERPs) this latency corresponds to a positivity labeled P3(P300) or Late Positive Component (LPC).

5.1.4 Recognition of subjective NOW (level-3-chunks)

From personal experiences we know that events are not perceived in isolation but usually several events are combined and perceived as (what psychologists call) *gestalts*. Evidently there is an integrative mechanism in the brain that is responsible for the *gestalt* formation (the subjective

experience of NOW), and the time constant of this system can be used as a measure of the duration of NOW. A number of experiments (Pöppel, 1982), concerned with the temporal organization of vision, suggest that the temporal limit of the mental availability of an experience (which is not forced to persist by external stimuli) is about 3 seconds. This 3-second segmentation of conscious experiences fits with the average time duration of verbal utterances in spontaneous speech, and with the average duration of spoken aloud verse lines. Also musical phrases have a temporal limit of about 3 seconds.

Is evident that in order to achieve behaviors such as comprehension/generation of sentences (which represent relations between concepts -- acts, events, etc.) DETE needs such an integrative mechanism with its appropriate time-constant (refreshment period). For this purpose, the update period of the Temporal Memory Planes (see sections 2.3.4 and 9.3) was chosen to be 300 B-cycles (see Table 5.1). This period is called one "*moment*". Notice that as a result of the chosen temporal hierarchy, everything that DETE learns has a temporal component. For instance, when DETE learns the meaning of the word "stands" it really learns <stands for n-moments>. When it learns "red" it really learns <red for n-moments>.

5.1.5 Recognition of duration

Our subjective experience of duration seems to be determined by the intensity of the mental experiences (i.e. content, rate or load). The bigger the mental content within a given unit of time, the longer the retrospective feeling of duration and vice versa. It seems that in humans some sort of memory must be the prerequisite for such experience of duration. Experiences must be stored cumulatively in this memory and should be accessible in terms of their duration. In general, such a memory mechanism should be responsible for our ability to perceive events as past and also as future. DETE contains a special neural network, the Temporal Memory Planes, which allow it to "perceive" events as present, past or future and to handle such linguistic categories as verb tenses (e.g., present, past, and future) (see section 11.7).

5.3 Other processing issues

There are a number of other issues regarding temporal processing in neural systems. The time available for processing places strict constraints on the types of algorithms that could be implemented in the cerebral cortex. For instance, it takes on average about half a second following a presentation of an image on the retina until we can recognize an object in the image. Considering the time taken for processing the information at any specific stage (e.g., 25-50 (100) msec for the signal to reach the cortex, a few hundred msec (vague) required for the motor system to produce a response) then the actual time left for visual processing is about 200-300 msec (Sejnowski, 1986). Cooperative algorithms, for instance, that require extensive exchange of information (e.g., more than 20 iterations -- like the requirements for back-prop (McClelland et al., 1986)) between local neurons (numerous iterations within some system of recurrent collaterals) do not seem likely. However, algorithms that take less than 20 iterations are quite plausible (cf. for example Wilson and Bower, 1989).

6 THE MOTOR SYSTEM

The motor system in DETE is very rudimentary. It consists of two motor effectors and a motor memory. The motor effectors are: (1) a FINGER which is used to act upon the blobs in the visual world, and (2) an EYE which can fixate at different locations of the visual screen and has a variable diameter. The motor memory is used for learning motor sequences of the FINGER and the EYE. The purpose of the motor system is twofold. First, to control the movement and diameter of the EYE (and thus the location of the focus of attention) with respect to the visual screen. Second, using the FINGER, to perform simple motor behaviors such as (a) *push* a blob in a given direction, (b) *drag* a blob from one location to another, and (c) *Deflect* a moving blob. Figure 6.1 illustrates these motor actions performed by the FINGER.

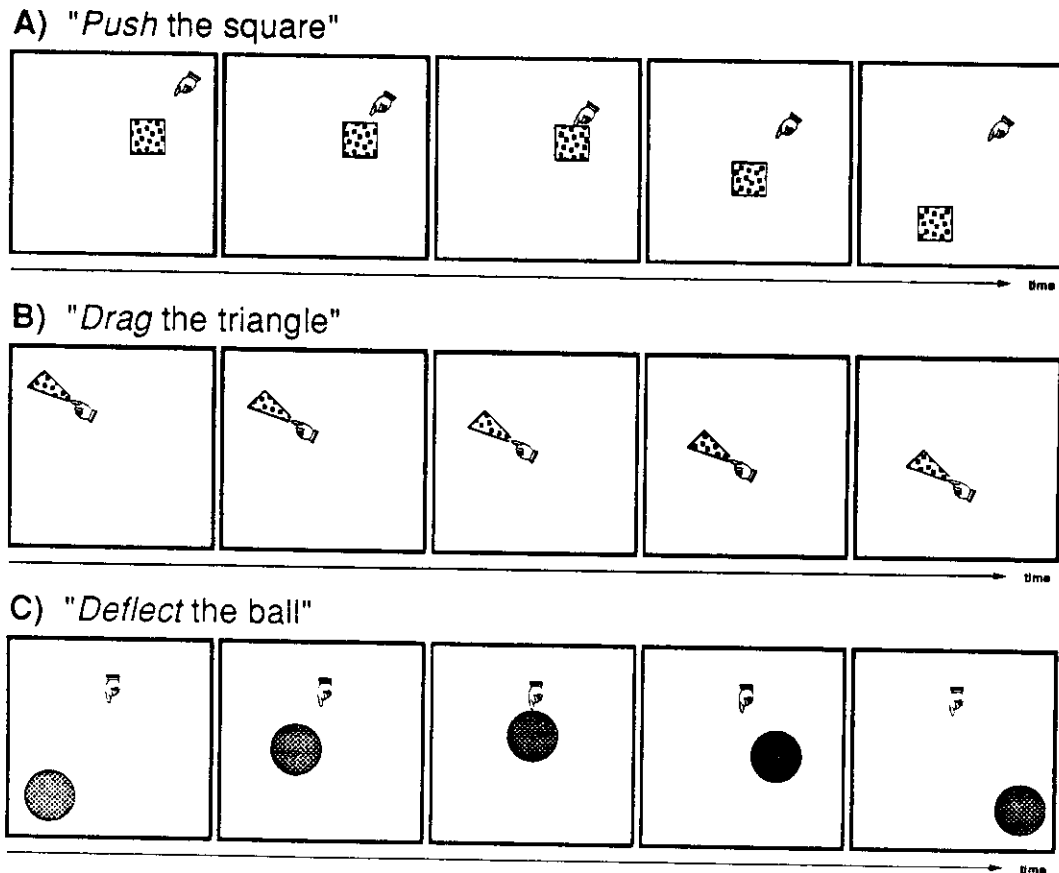


Figure 6.1: Motor interactions

Selected visual frames illustrating different interactions between DETE's FINGER and objects in the Visual Screen: **A)** *Push* the square; **B)** *Drag* the triangle; **C)** *Deflect* the circle.

DETE learns to perform motor sequences in response to a verbal command in the same way that it learns to generate verbal sequences. Namely, motor sequences are learned by motor examples associated with verbal inputs. For instance, DETE learns to move its EYE and/or FINGER from one object to another by being initially taken along this path (forced to) by the teacher while at the same time receiving the appropriate verbal command. To be able to learn such motor behaviors a system must maintain a "map" of the environment and to perform a search for particular features (objects) in the environment on demand. The motor map is maintained in a variation of the sequential associative memory which is described in a later chapter in this thesis. The search-for-feature strategy involves movements of the EYE which may be either random or memory driven and can involve resizing of the focus of attention field.

6.1 Representation of effector states (proprioception)

The states of DETE's two effectors, the FINGER and the EYE are represented in Effector State Planes (ESP). These planes are generated by the Effector State Extractors -- procedurally implemented modules. A relevant question regarding the choice of representations for objects is why DETE's effectors are represented in separate planes instead of being represented together with the rest of the objects that appear in the visual field in the five visual feature planes. In humans the position of their extremities (hands, legs, and even orientation of the eye-balls) in space is provided primarily by proprioceptive signals carried by the somatosensory system and visual feedback is used only during fine control. DETE does not possess a proprioceptive system (no muscle receptors or Golgi tendon organs), instead it uses visual information. However, for the purpose of compatibility with the human architecture, a decision was made in the design of DETE to process the visual information about effectors and objects in separate channels. Such a setup can be used as a basis for extending DETE by providing it with proprioception (substituting for the visual perception of its effectors).

6.1.1 Representation of FINGER State

The two state parameters of the FINGER (position and motion) are represented in two separate planes -- the FINGER Position Plane (FPP), and the FINGER Motion Plane (FMP). The FPP (Figure 6.2) is a topographic plane similar to the Location Feature Plane which is used to represent the location of objects in the Visual Screen (see section 3.2.4). The procedure which generates the FPP is the same as the procedure used to generate the LFP and therefore will not be described here in detail. The only functional difference is that the FPP maps the Visual Screen position of only one object -- the FINGER, whereas the LFP maps the positions of all other objects -- blobs which are not part of DETE. However, the FPP can potentially be used to represent different effectors (e.g., joints in an arm, multiple fingers, etc.).

The motion of the FINGER is represented in the FMP (Figure 6.3). Motion is represented similarly to the representation of object motion in the Motion Feature Plane (see section 3.2.5).

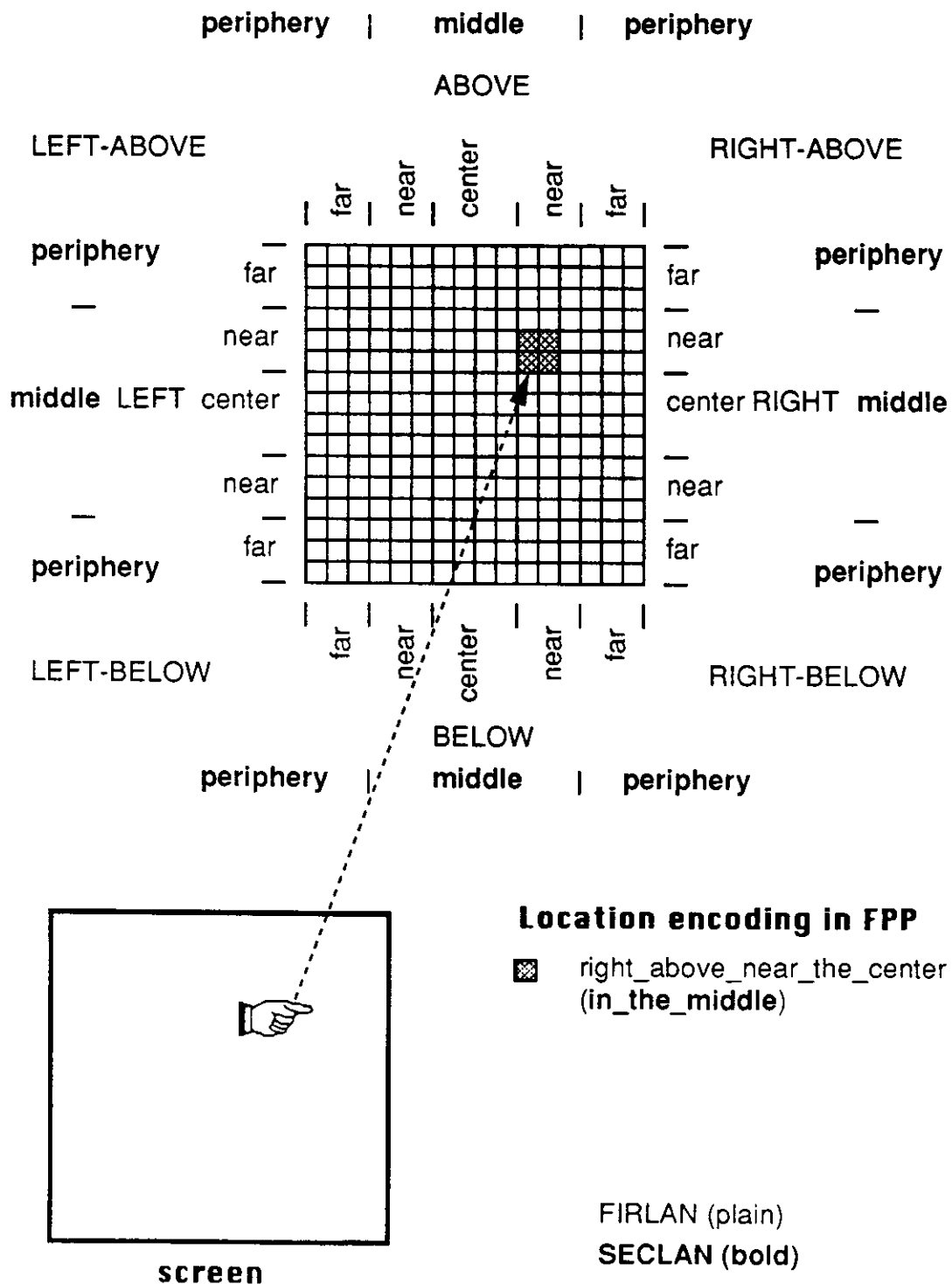


Figure 6.2: FINGER Position Plane (FPP)

Schematic drawing of the FPP. Notice that the lexical items used in FIRLAN or SECLAN to describe the location of an object in the LFP can also be used to describe the position of the FINGER in the FPP.

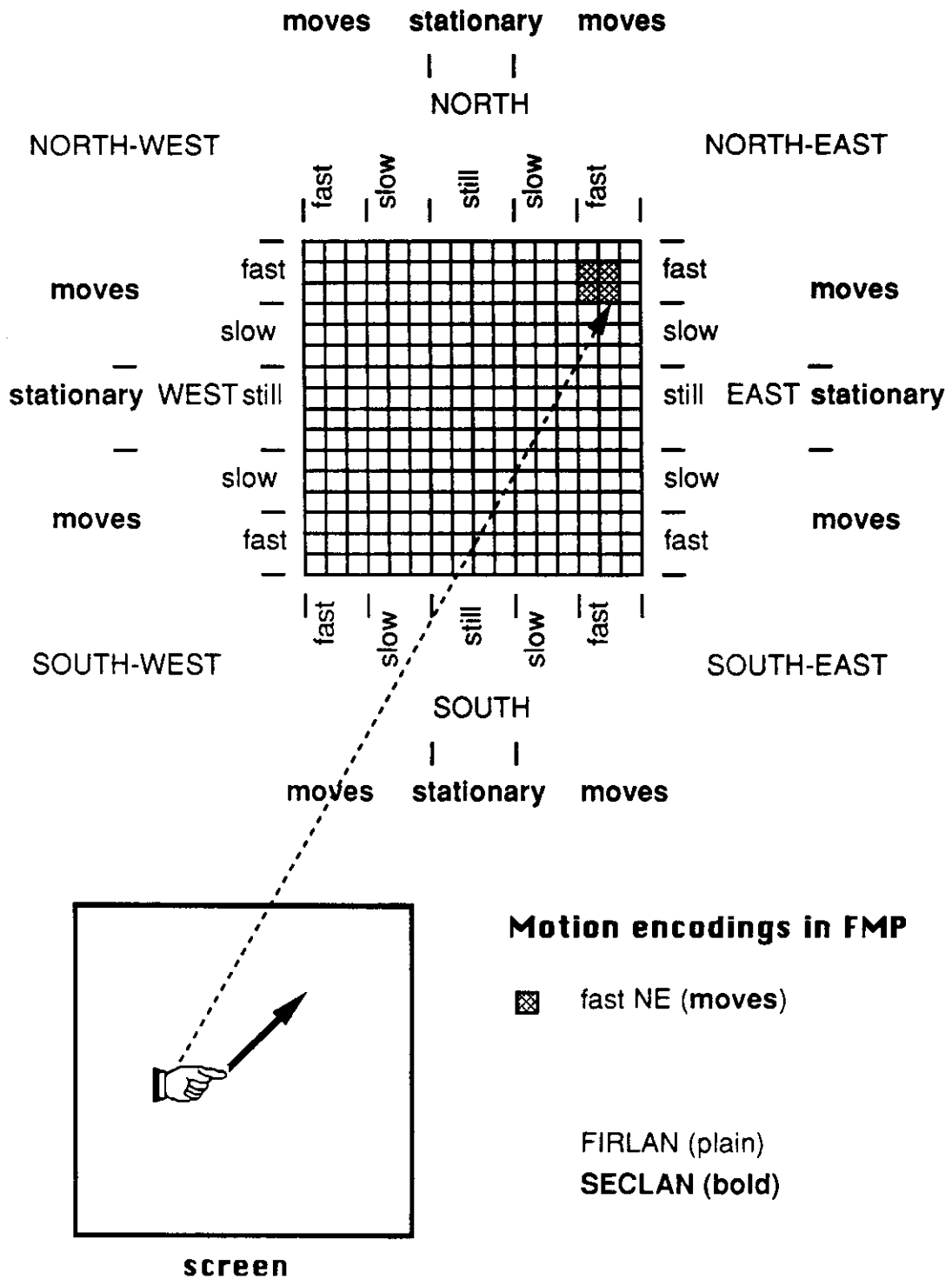


Figure 6.3: FINGER Motion Plane (FMP)

Representation of the FINGER motion (moving fast North-East) in the FINGER Motion Plane. As shown in the figure, a variety of words and phrases in FIRLAN and SECLAN can be used to describe the motion of the FINGER.

6.1.2 Representation of EYE state

The two state parameters of the EYE are its location on the visual screen and its diameter. The location of the EYE is represented similarly as the location of the FINGER in an EYE Location Plane (ELP) and will not be discussed here in detail. The diameter is represented in the EYE Diameter Plane (EDP). The EDP is a 64 bits long vector (Figure 6.4). Each value of the EYE's diameter is represented as a pair of four contiguous 1 bits. Four bits were chosen for redundancy of representation and also to provide bit-density comparable to that in the other modalities.

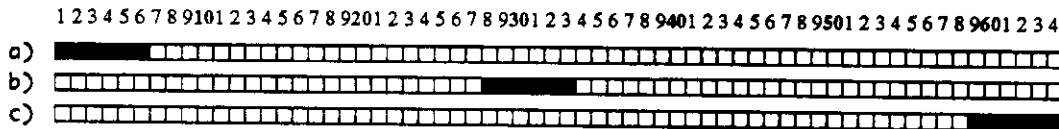


Figure 6.4: EYE Diameter Plane (EDP)

The representation is "linear", i.e. if the active bits are located at the beginning of the EDP vector (a), the diameter is minimal while if they are located at the end of the vector (c), the EYE diameter is maximal. Positions of the active bits between the beginning and the end of the EDP vector correspond to intermediate diameter sizes (b).

6.2 Types and ranges of effector motions

Due to their different purposes, the types of motions that DETE's effectors can accomplish are different and their range is limited. In general, however, they can be categorized as reflexive or voluntary.

6.2.1 EYE motions

DETE's EYE motions are of three general types: (1) *externally controlled motions* -- guided "mechanically" by the user, (2) *visual stimulus controlled motions* -- in response to stimuli in the visual scene, and (3) *verbally controlled motions* -- in response to verbal inputs which have been previously associated with externally controlled motions.

(1) *Externally controlled motions.* These motions of the EYE are controlled by the user via a joystick. Such motions are needed during the initial training period to move the retina over a particular object on which the teacher wants DETE to focus its attention (i.e. externally guided saccades) or to change the size of the EYE's diameter (i.e. externally guided EYE accommodations).

(2) *Visual stimulus controlled motions.* These motions, which can be either saccadic motions or EYE accommodation, are reflexive in nature. They are performed in tandem when DETE is left on its own to explore the Visual Screen, or when there are sudden changes in DETE's Visual Field that capture its attention.

(a) *Saccades* are simple linear (ballistic) motions of the EYE from an initial position to a target position that is off the fovea (the center of the retina). (For discussion of the neurobiology of saccadic motions see section 13.4). The saccadic motions of DETE's EYE have the following dynamics. Their duration is 10-50 msec (1-5 B-cycles) depending on how far from its original position the EYE is displaced. Each saccade is followed by a gaze -- a fixation of the EYE at a point

in the VS for 300 to 500 msec. These dynamics were chosen to roughly conform to those observed in humans.

The algorithm of the reflexive saccadic motion is simple. All pixel values in the Visual Field are examined in parallel for changes at each B-cycle. If one or few closely located pixels have changed their values (which correspond to the appearance/disappearance of an object in the VF, or to a motion of an object which is already in the VF), DETE does the following. First, it calculates the center of gravity (CG) of this cluster. This is done within the duration of the one B-cycle. Second, during the following 1-5 B-cycles the EYE is moved to its new position. If two or more clusters of changed pixels appear simultaneously, then the one with the maximal intensity (number of changed bits) is selected as a target of the saccadic motion and the rest are disregarded (this calculation is done procedurally). During the duration of a saccade, any additional changes in the visual field are disregarded -- DETE is not receptive.

(b) *Accommodation* is a reflexive change in the diameter of DETE's EYE, which is an analog of the eye accommodation reflex in humans. (For discussion of the neurobiological basis of eye accommodation in humans see section 13.4) This type of motion is necessary since DETE needs to focus on objects of various sizes or on whole scenes containing several spatially distributed objects.

The accommodation algorithm is also very simple. The driving signal is provided by the size Feature Plane (ZFP) which represents the size of an object or the cluster of the pixels that have changed their values. Similarly to the algorithm for saccadic motion, in the accommodation algorithm DETE first computes in parallel the diameter of the cluster of value-changed pixels. This is done by finding the min and max of the grid addresses of the value-changed pixels on the X and Y axis of the Visual Screen; subtraction of the min from the max for each axis, and assigning the value of the diameter to the max of the result. Then the current radius of the EYE is changed to the newly computed value.

(3) *Verbally controlled motions --voluntary* -- used in the performance of tasks such as "look up", "look left", "find", "zoom in", "zoom out". For instance the command "find the red ball" initiates a search over the objects within the Visual Screen which is expressed in saccadic motion of the EYE (VF) and zooming in and out -- accommodation. It results in DETE placing its EYE (and respectively finger tip) over the object (if such an objects exists in the Visual Screen), or in quitting after examining all objects (if the object is not in the VS).

6.2.2 FINGER motions

The FINGER is regarded (represented) as a solid object which can be moved via the joystick or "voluntary" within the boundaries of the Visual Screen and can interact with the rest of the objects (also solids) in the VS. The scope of possible FINGER-object interactions is very limited. Presently it includes only: push, follow, and deflect. Each of these interactions can be characterized by the initial and final states of the participating FINGER and object and a time interval during which the states of the object and the FINGER change.

Similar to the control of the EYE, the control of all FINGER motions is either via the joystick (i.e. external) or voluntary (i.e. internal). The external control is used during training of DETE to perform a particular motor task. The ability for voluntary control is acquired (learned) by giving DETE simple verbal instructions in association with externally controlled motions of the FINGER. The following verbal commands, (words or phrases) can be used to elicit the corresponding motor actions by DETE's FINGER:

1) "PUSH (HIT) the ball to the center". This command (which requires that DETE has information about the location of the ball, the location of its FINGER, and the desired direction of motion) causes DETE's FINGER to initiate the movement of the object in the given direction, while both the EYE and FINGER remain in place. Since this is an elastic hit, here we also worry about the conservation of the velocity moment (mxv , $m = \#pixels$).

2) "FOLLOW the ball". In this situation the ball is moving on its own. The command causes the EYE and the FINGER to move together with the ball.

3) "DEFLECT the ball". This is similar to PUSH but the object is moving. Depending on what are the initial locations of the ball and the FINGER, this command causes DETE's FINGER to interrupt the ball's motion by positioning itself in front of and on the path of the moving ball.

Variations on these commands/behaviors are also possible. For instance modifiers of the motions can be used (e.g., slow, fast, etc.).

6.3 Motor memory

To be able to perform the above-mentioned verbally-controlled motor behaviors, DETE needs to initially learn the meaning of the verbal instructions. The motor system contains a sequential associative memory which is used to learn and recall motor sequences. Part of this memory is used to control of the EYE and the other part to control the FINGER. The motor system takes two types of input: the position of the EYE on the screen and its diameter, and the position of the FINGER on the screen and its motion. At first, for the purpose of simplicity, the retina and the finger are tied together (i.e. wherever the finger points, that is where the retina is looking and vice versa).

The learning strategy which is adopted here is similar to the way parents teach their kids. For instance, first the attention of a child is somehow attracted to a given object (in DETE this is done externally through the joystick by moving the EYE-FINGER to the right location) and during the time when the child orients towards the object the parent issues the utterance "LOOK at the ball" (during the act of moving the EYE-FINGER from its initial location to the ball) or simply "BALL" (if the EYE is already on the ball). Of course, the same strategy is used while learning the meaning of "ball". From this it is evident that the motor behaviors have to be prerequisites for the language behavior. In similar situations the user says "HIT the ball" while holding DETE's EYE-FINGER on the ball and pushing it. Similarly, by pairing verbal and motor events (while using the visual capacity) DETE learns to actively interact with the blobs world.

7 SELECTIVE ATTENTION IN DETE

At any moment, we deal only with a small fraction of information that we perceive from the environment (visual, verbal, or through any other sensory modality). This faculty is attributed to our attentional mechanisms. Attention is a mental process through which we avoid distraction by irrelevant stimuli (external or internal) while seeking out and focusing on those stimuli that are behaviorally or task-wise important. This process allows us to access information selectively and sequentially. DETE has been provided with some functional capabilities that resemble those of humans. The attentional mechanism used in DETE is described in this chapter.

7.1 Representation of attention in DETE

Chapter 3 discussed how visual information is represented in the individual visual feature planes (VFPs). Each object on the retina is encoded by an assembly of synchronously oscillating units (object-assembly) at the VFP level. Subgroups of neurons in each object-assembly represent various object features. While the oscillations of the neurons within an object-assembly are phase-locked (have the same phase angles), the oscillations between object-assemblies are phase shifted. The phase shift is used to represent the segmentation of the visual input into objects in the temporal domain. This scheme allows several objects to be represented simultaneously. In order to learn about only one of several objects, there is a need for a mechanism that will make this object “special” for some time, i.e. focus attention on it. This mechanism is the Selective Attention Mechanism (SAM).

In DETE, selective attention is represented as a short time-window (Temporal Attention Window -- TAW) that opens and closes cyclically with the same frequency as that of the oscillating neurons in the object-assembly (Figure 7.1). Any object assembly that has the same phase as the TAW is considered to be “attended to”. All other objects are unattended. For example, in Figure 7.1., the TAW is open in phase with the phase of the circle within the retina. As a result, the circle (rather than the square or triangle) is in the focus of attention. All of the circle attributes (color, location, etc.) are also within the TAW -- i.e. have the same phase.

Notice that in the literature on attention, a “visual attentional window” is commonly understood as a small area of the visual field (i.e. its dimension is *space* rather than *time*). This area can be in the center of the retina (coinciding with the fovea) or out of it. In DETE, the TAW is in the temporal domain.

What makes the open state of the TAW special is that it is only during this time that the short term memory (STM) learns. The STM and the relation of its update to the TAW will be discussed in Chapter 9. Here I will mention only that as a result of this interaction, objects that are attended to leave stronger traces in the STM than those that are not. The bigger the phase gap between the TAW and the oscillations that represent a particular object, the weaker the memory trace left by this object in the STM.

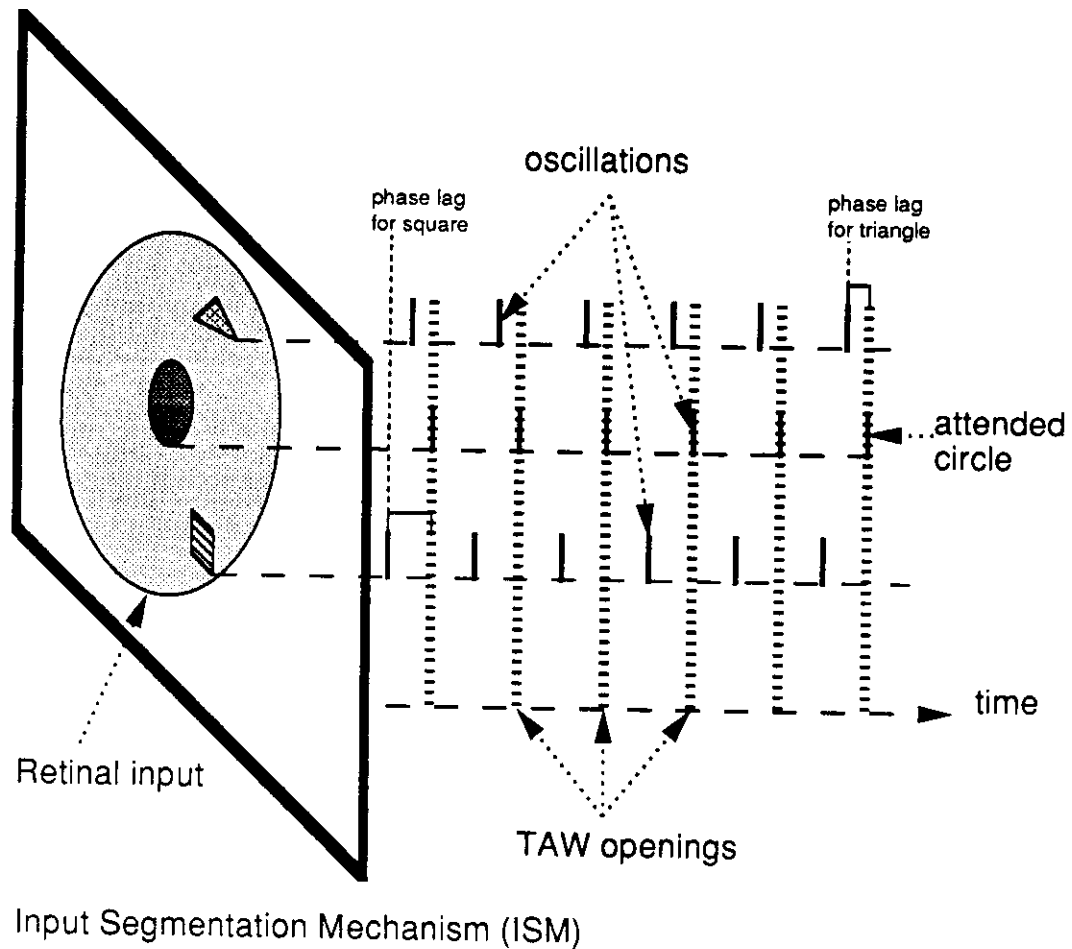


Figure 7.1: Input Segmentation Mechanism (ISM)

Three objects (a circle, a triangle and a square) with different locations on the retina are represented in the output of the ISM (see Figure 2.4). Shown are the oscillations of only one neuron per object which is representative for the oscillations of the whole object-assembly.

7.2 The Selective Attention Mechanism

The Selective Attention Mechanism (SAM) consists of two functional modules, the Input Segmentation Module (ISM), and the Focus of Attention Master (FAM). A block diagram of the Selective Attention Mechanism and its relation to the rest of DETE's modules is presented in Figure 7.2. This figure shows a retinal input received by the ISM. It consists of three objects (a circle in the center, a small triangle north-west of the center and close to it, and a square north-east of the center at the periphery of the retina). The ISM represents each object as an assembly of oscillating neurons with different phases between the assemblies and the same phase within the assemblies. The oscillations are passed to all feature extractors (FEs) in parallel. The mapping between the ISM and each of the FEs is topographic and one-to-one. Further, the individual FEs extract the relevant

features from the input signal and pass the outputs (feature planes) to the corresponding parts of the visual memory. Again the mapping is topographic. The FAM generates the Temporal Attention Window (TAW) and conveys it to the visual memory. The phase of the TAW by default is the same as the phase of the neurons oscillating in the center of the retina and can be controlled by the verbal input. The TAW determines when the STM and LTM are updated.

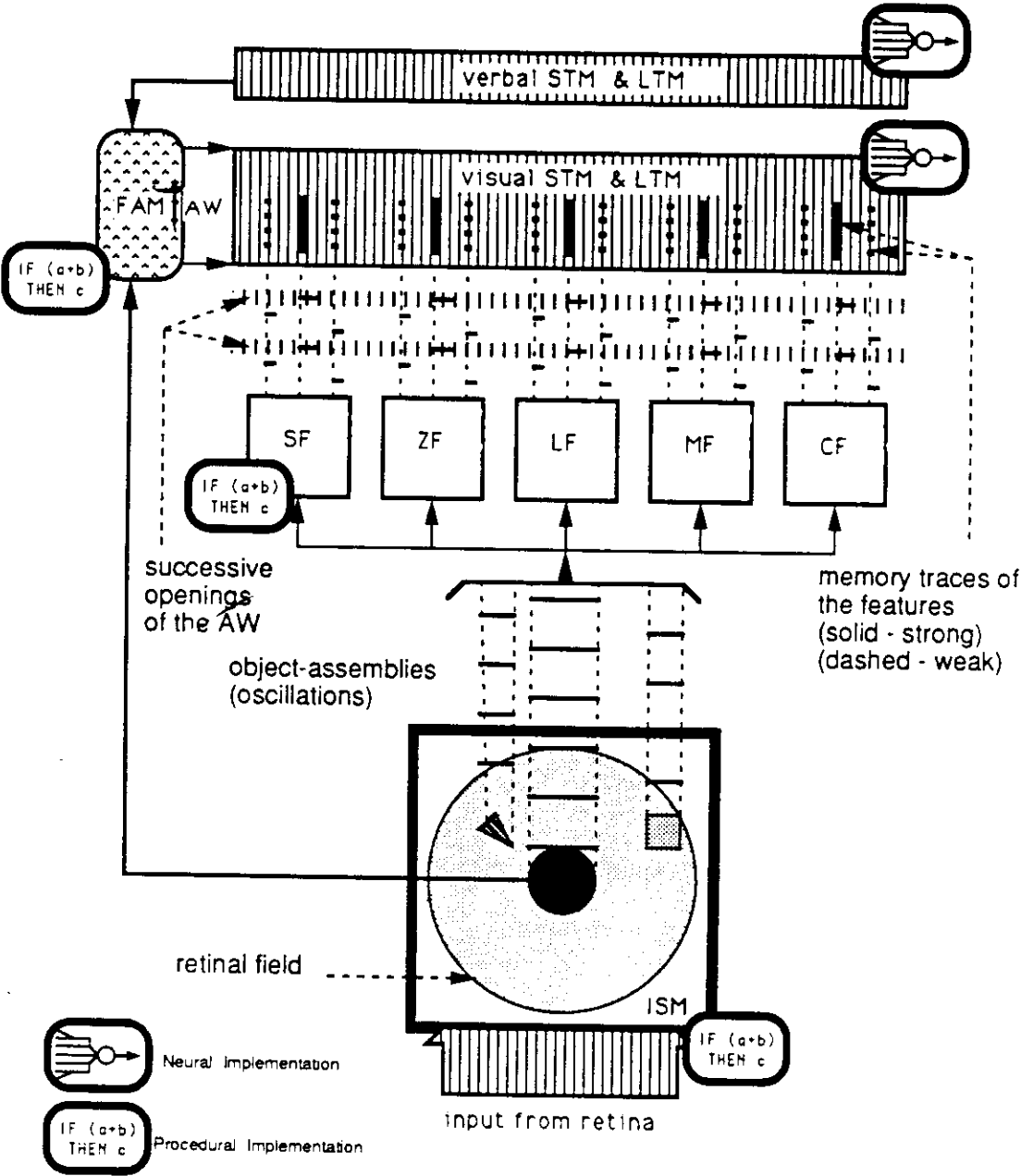


Figure 7.2: Selective Attention Mechanism (SAM)

Schematic drawing of the SAM function. The ISM (shown on the bottom) segments out in time the representations of the three objects and passes these representations in parallel to the five Feature Extractors (FEs: SFE, ZFE, LFE, MFE, and CFE). The FEs pass the generated feature maps to the corresponding parts of the visual memory. The FAM (shown up to the left) generates the TAW and conveys it to the visual memory (STM & LTM). The two icons (shown in the bottom left corner) denote the type of implementation of the individual modules (neural or procedural).

7.2.1 Input Segmentation Mechanism

The Input Segmentation Mechanism (ISM) performs initial segmentation of the visual input into objects. Segmentation is represented in terms of phase differences. This module is located between the retina and the visual feature extractors. Phase differences between the objects are created within the ISM. Remember that DETE looks at a visual scene through the retina (a circular aperture with a variable diameter, and center that can be targeted at any point of the Visual Screen). The representation of an object is such that the further away from the center of the retina it is (while still within the retinal field), the larger the phase shift. As emphasized in section 2.3.2, since the FAM clock ticks once every 5 B-cycles, in DETE there are only 5 different phases in which object representations can be. In other words, DETE can look at a maximum of 5 different objects simultaneously. The ISM is a procedural module which takes retinal input and for each object in the retina computes:

- (1) The location of its Center of Gravity (CG) with respect to the center of the retina.
- (2) A phase lag for each object such that, if the CG of the object coincides with the center of the retina, then the phase lag is zero. If the CG is at the periphery of the retina, the phase lag is 180° (i.e. delayed by 4 B-cycles after the TAW -- see Table 7.1). The magnitudes of the phase lags of objects located between the center of the retina and the periphery are proportional to the distance from the center.
- (3) A representation for each object as an assembly of neurons oscillating in phase with the CG of the object.

7.2.2 Focus of Attention Master

The Focus of Attention Master is the procedural mechanism that generates and controls the Temporal Attention Window. The FAM contains an internal clock which generates continuous oscillations. The frequency of the FAM is the same as that of the frequencies generated by the ISM. Its phase is always locked to the phase of the object that is closer to the center of the retina (the fovea). The FAM is connected to all memory modules, i.e. there is a connection from the FAM to every neural element in each memory bank. A tick of the FAM happens once every 5th B-cycles and is one B-cycle long. Each tick opens a short temporal window (one B-cycle long). The functional significance of this window is such that only in this time window can the memories learn. In other words, only the neural activity in the STM during this temporal window leaves a trace in the STM (the *l_{tm}* of the STM is modified). The length in terms of B-cycles and the phase relation of the TAW to the phases of the oscillating neural assemblies representing individual objects can be changed. The phase relation can be controlled by the verbal input. An example of how this happens is given in section 11.5.1. The fact that the phase of the TAW can be controlled verbally is especially important when visual images are generated mentally as a result of a verbal input. The fact that the Focus of Attention Master (FAM) is designed so that if there is not a visual input, the

first word of any verbal input is synchronized with the TAW. In other words, the TAW opening coincides with the first B-cycle of the first gra-phoneme.

There are various ways to construct an oscillating circuit which can serve as the FAM. One example of such oscillator which has a neural flavor (a simple neural network composed of an excitatory and an inhibitory elements connected in a feedback loop) is shown in Appendix E.

7.3 Control of selective attention

Once generated by the ISM, the phase differences that correspond to the different objects are maintained for an amount of time called an attention span. In humans, the attention span is about 300 to 500 msec. This period corresponds behaviorally to the duration of visual saccades *** CITATION EMPTY, OR MISSING CLOSURE *** (300 to 500 msec long fixation followed by a fast eye movement 10 to 20 msec long) or the average duration of auditorily presented words (Table 5.1).

To focus attention on different objects in the Visual Field DETE first moves the center of the retina to the object as a result of which the phase of the Focus of Attention Master changes. It is important to notice that DETE's visual memory can maintain activation (representing various objects) which is provided either through the visual input or is elicited through the verbal input. Attention can be switched between such differentially generated object representations. This is done by re-locking the phase of the FAM from the representation of one object to that of another. In humans such switches of attention correspond to the ability to switch from perception to thinking (mental imagery).

When an image of an object is elicited in the visual memory through a verbal input (e.g., "A ball"), new visual features can be added to the visual representation of the object through the verbal input. For instance, the verbal input "is red" elicits the representation of red color in the color bank of the visual memory. The new feature is phase locked automatically to the existing neural-assembly representing the ball and to the FAM respectively. If a new object is introduced through the verbal input (e.g., "a square") a new neural-assembly with a different phase is activated in the corresponding parts of the visual feature memories (e.g., some of its neural elements are in the area of the Shape Feature Memory which represents squares). The FAM is unlocked from the previous target -- the ball (i.e. the TAW phase is shifted) and the incoming information about the new object is automatically phase locked with this new phase of the TAW.

8 BASIC MEMORY MECHANISMS

The ability to learn, recall, and recognize sequences is fundamental for any system that is designed to handle dynamical tasks, e.g., language, vision, and motion. DETE's sequence processing ability is based on a unique, specially designed memory mechanism -- the KATAMIC* Sequential Associative Memory. This memory mechanism is used (with minor modifications) in all of DETE's memory modules. The KATAMIC neural network is a sequential associative memory that can rapidly learn multiple sequences of randomly generated or structured binary patterns, recognize them and recall them (i.e. do sequence completion) in response to cues (short sub-sequences of stored sequences). This section describes the architecture and dynamics of the KATAMIC model, and presents the results of basic simulation experiments that test its functional characteristics. The modifications introduced to the KATAMIC model which allow it to function as a Short Term Memory, or a Long Term Memory, or a Procedural Memory are described in detail in Chapter 9. The neural plausibility of the KATAMIC model is discussed in Chapter 13.

8.1 Network architecture

The KATAMIC model is a synchronously updating sequence processing network. The update cycle of this network will be called the BASIC-cycle (B-cycle for short). A block diagram of the KATAMIC memory is presented in Figure 8.1. The model has three modules. (1) *predictor* -- contains a set of predicting devices, (2) *recognizer* -- contains a set of recognition devices, and (3) *input-gator* -- contains a set of input-gating devices. At each B-cycle, the function of the *predictor* is to get one pattern from the input sequence and to generate an output pattern which is a prediction of the next pattern in the input sequence. The function of the *recognizer* is to compare the output pattern with the next input pattern and to generate a control signal for the *input-gator*. This signal reflects the quality of the prediction made by the *predictor*. The *input-gator*, in turn, decides which pattern should be used during the next B-cycle as input to the *predictor* -- the next (external) input pattern or the (internally generated) prediction. In other words, the *input-gator* sets the KATAMIC memory in a "receptive" or "generative" mode. In the "receptive" mode the memory "listens" to the input sequence, whereas in the "generative" mode it uses its knowledge of a particular sequence to recall (generate) this sequence in response of an initial cue (the first few patterns of the sequence).

The functional units which constitute the individual modules of the KATAMIC memory are called: *predictrons*, *recognitrons*, and *bi-stable switches* (BSSs). Each of these three types of units has different functional characteristics which are described in the following sections.

* Named after Katarina and Michael whose presence at the right place and time made this model possible.

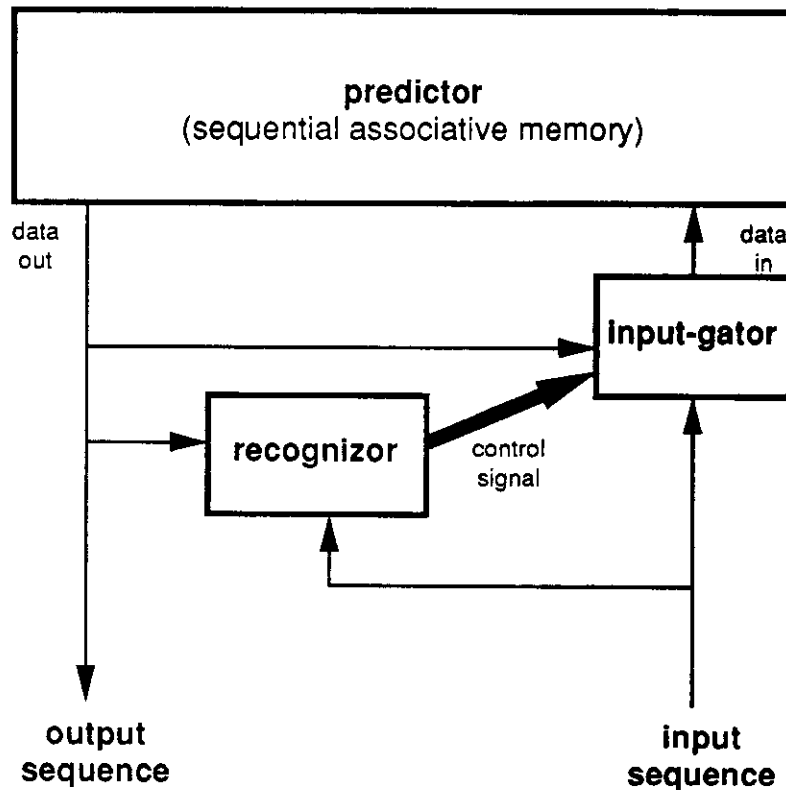


Figure 8.1: Block diagram of the KATAMIC model

A block-diagram of the KATAMIC model. The three basic components (predictor, recognizer, and input-gator) are shown as rectangles. The data flow paths (for input and output sequences) are shown as thin lines. The thick line shows the control signal provided from the recognizer to the input-gator. This signal instructs the input-gator to pass either the external signal -- "input sequence" or the internal signal -- "output sequence" to the predictor.

8.1.1 Predictron

The *predictron* (*predicting neuron*) is the basic neuron-like computing element of the KATAMIC model. A specific name was chosen for this unit since it is functionally different and structurally more complex than the classical McCulloch-Pitts neuron (McCulloch and Pitts, 1943) used in the majority of the connectionist models. However, the predictron is significantly simpler than any real neuron in the nervous system. Schematic drawings of a *neuron* (the basic processing element in the nervous system), a *predictron* (the basic processing element in the KATAMIC model), and a classical McCulloch-Pitts neuron are shown in Figure 8.2.

Let us first compare the predictron to a real neuron (e.g., a purkinje cell in the cerebellum). The neuron and predictron are both composed of soma, dendritic tree, and axon, whereas in the McCulloch-Pitts neuron the notion of dendritic tree is irrelevant and the soma maps to the neural element itself. However, while the dendritic tree of the neuron is composed of multiple branches, the predictron's dendritic tree has only a single branch. Also, both dendritic trees are composed of

dendritic compartments (DCs). In the neuron's dendritic tree, a compartment is defined as a functionally identifiable unit (specifically, it can be a synaptic button, or a synaptic stalk, or a patch of membrane, or the set of all channels of a particular type, or a part of the tree between two branching points). In the predictron, a dendritic compartment (DCP) is a part of the dendrite characterized by its own set of parameters. Also, unlike real neurons, the dendritic tree of the predictron consists only of one single branch.

Second, let us compare the predictron to the classical McCulloch-Pitts neuron. An important difference between the predictron and a McCulloch-Pitts neuron is that in the predictron the dendritic compartments are used to store several different types of memories and the position of the individual compartments in the tree is of importance whereas in the classical neuron the synapses store only one value -- a weight, and the positions of the synapses on the soma are irrelevant. Another significant difference is that for its operation a predictron needs on the order of hundreds dendritic compartments (similarly to real neurons) whereas a classical neuron can function with only a few synaptic inputs (weights).

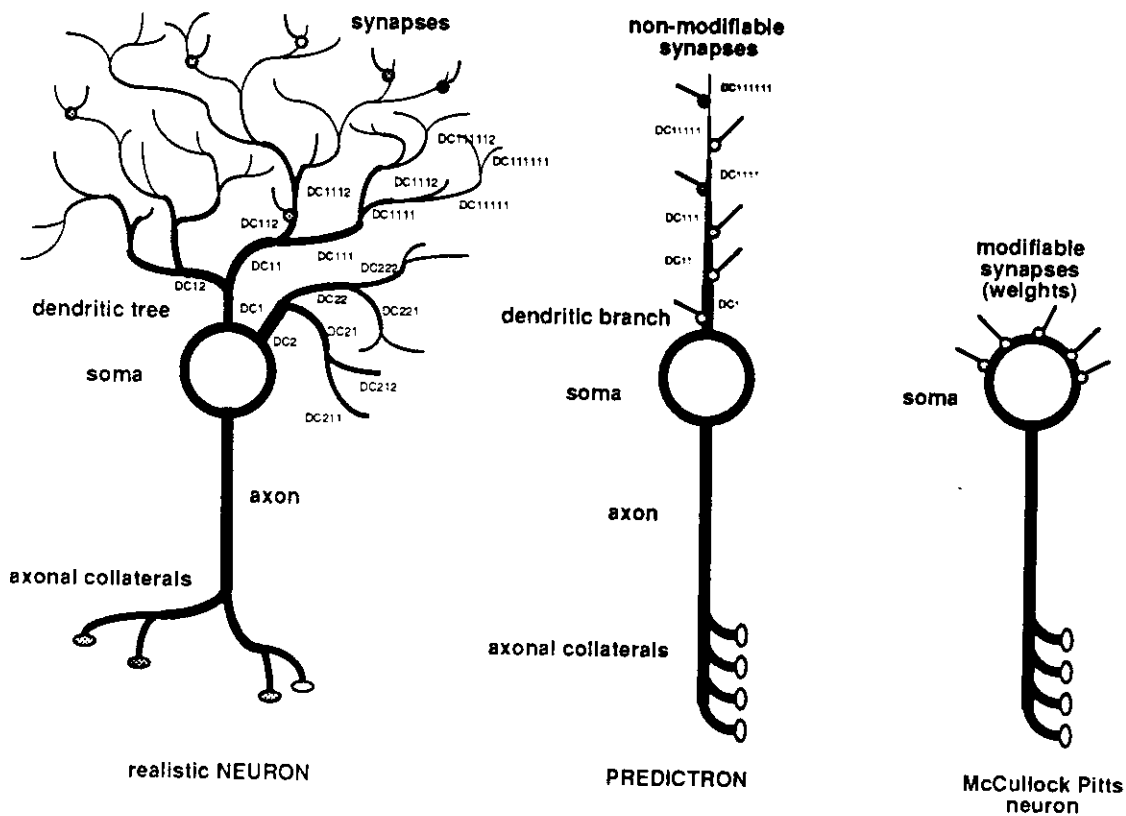


Figure 8.2: Real & artificial neurons

Several of the dendritic compartments are labeled according to their locations in the tree (e.g., DC212, DC111, etc.). Synapses made by other neurons onto the dendrites are shown as little circles. Different gray shadings of the circles are used to emphasize that the synapses have different weights.

Now let us consider the basic function of the predictron. At each B-cycle, a predictron takes inputs (specifically: one direct input from a particular location of the input pattern, and multiple indirect inputs from the rest of the bits forming the input pattern), changes its state, and generates an output. The function of the predictron is to learn to predict successive direct inputs. The number of *predictrons* in the model is P . Each predictron contains a soma (cell body) and a single dendritic branch composed of several Dendritic Compartments (DCPs) (Figure 8.3). The number of DCPs per predictron (DP) is the same for all predictrons. Having exactly the same number of DCPs in each predictron is not a theoretical limitation. The model operates even if the number of DCPs is only similar across predictrons. The choice of equality was based on implementational constraints. Each DCP is characterized by 3 variables:

- (1) a positive Long-Term Memory variable (*p-ltm*),
- (2) a negative Long-Term Memory variable (*n-ltm*),
- (3) a Short-Term Memory variable (*stm*).

Each of these variables is a real number between 0 and 1. The *ltm* variables are used to store information about the spatial relations (how far apart within a pattern) and temporal relations (how far apart along the time axis) of the active (i.e. ON or 1) bits in all sequences that were presented to the memory since its naive state (i.e. before it has learned anything). The *stm* variable is used to store information about the spatial and temporal relations only among the 1-bits of the most recently seen patterns in a given sequence. The *stm* value has a specific dynamic characteristic -- it "flows" towards the soma with a speed of one DCP per B-cycle. At the same time it decays with decay constant T_t . This dynamic allows the *stm* value to serve a dual purpose: (1) as an intracellular delay line, a feature which is used during learning, and (2) as a "look one step ahead" mechanism which is used for prediction generation. Another feature of the *stm* in each DCP is that it is reset to its initial value at the beginning of every new sequence presented to the network. The reset signal is provided externally and reaches each DCP through branches of a wire called the Climbing Fiber (CF).

The soma of the predictron is characterized by an activation value, $AP(t)$. This value is computed at each time cycle as the dot-product of two vectors -- the shifted-*stm*, and the difference of the *p-ltm* and *n-ltm* in the predictron's dendritic branch. The state of the predictron at each time cycle is either "fire = 1" or "silent = 0". If the somatic activation is larger than a threshold value Θ_P , then the predictron fires, otherwise it is silent.

8.1.2 Recognitron

The *recognitron* (*recognition neuron*) is a bi-stable neural element. Its function is to recognize the input sequence on a pattern-by-pattern basis. The number of recognitrons in the model is R . Like the predictron, the recognitron is composed of a soma and a dendritic tree. The soma is characterized by an activation value $AR(t)$ and a threshold Θ^R . There are two dendritic branches per recognitron extending horizontally in both directions for some distance. These dendritic branches are also composed of Dendritic Compartments (DC^rs) and the total number of compartments per recognitron is DR . In different modifications of the KATAMIC model this number can vary from 1 to P , depending on the choice of recognition criteria made by the designer. For instance, one recognition criterion is "recognize on a predictron-by-predictron basis", i.e. when each predictron generates a correct prediction independently of the others. Such a situation may occur when the information carried by each bit in the input pattern is not related to the information carried by the

neighboring bits. Another possible recognition criterion is “recognize on a global basis”, i.e. only when all predictrons generated correct predictions.

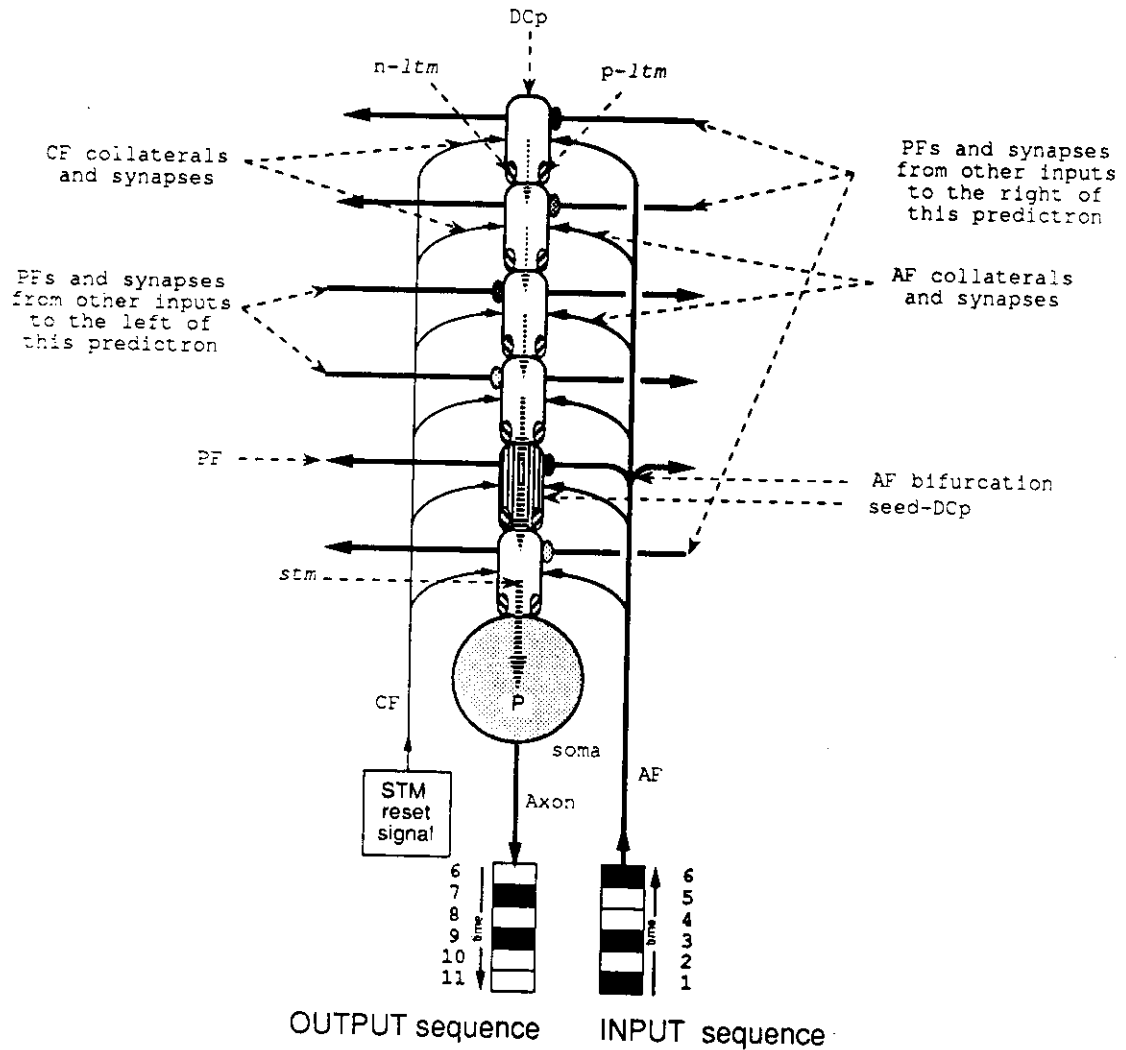


Figure 8.3: Predictron

The *p-ltm*, and *n-ltm* state variables for each DCP are shown as small ellipses on the bottom of each DCP. Their values can vary between 0 and 1. The *stm* state variables are shown as vertical arrows from one DCP to another. The various thicknesses of the arrows indicate different values of the *stm*. The arrows at the end of the climbing fiber (CF) branches and the ascending fiber (AF) branches that make contacts with individual compartments, represent non-modifiable synapses of weight 1. The small ellipses at the contact points between PFs and DCPs represent non-modifiable synapses with weights between 0 and 1. The darker the ellipse, the larger the synaptic weight.

The dynamics of the DCr's are different from those of the DCPs. Each DCr receives two inputs, (1) external -- a bit from the input pattern, and (2) internal -- a bit generated by a corresponding predictron (Figure 8.4). Each DCr computes a logical XOR of these inputs. The XOR function has

been chosen because it evaluates to 0 when the two inputs are the same and to 1 when the two inputs are different, and therefore it adequately distinguishes a successful prediction from a failed prediction. In our implementation of the KATAMIC model the computation of the XOR function of 2 arguments was done procedurally to save computational time. However, it is possible to design a simple neural network which can compute this function (see Appendix F).

At each B-cycle the results of the XOR computations are summed algebraically across the recognitron's DCr's to form the "somatic activation" $A^r(t)$ of the recognitron. The value of $A^r(t)$ is in the range of 0 to D^r . The somatic activation is further compared to the threshold of the recognitron Θ^r and if it is larger, then the output value passed along the recognitron's axon is 1, otherwise it is 0. The fact that a prediction has failed (represented by firing of the recognitron) or that it was correct (the recognitron is silent) is conveyed to the DCP's of the predictron via a separate wire called the Recognition Fiber (RF) -- a collateral of the recognitron's axon (Figure 8.4).

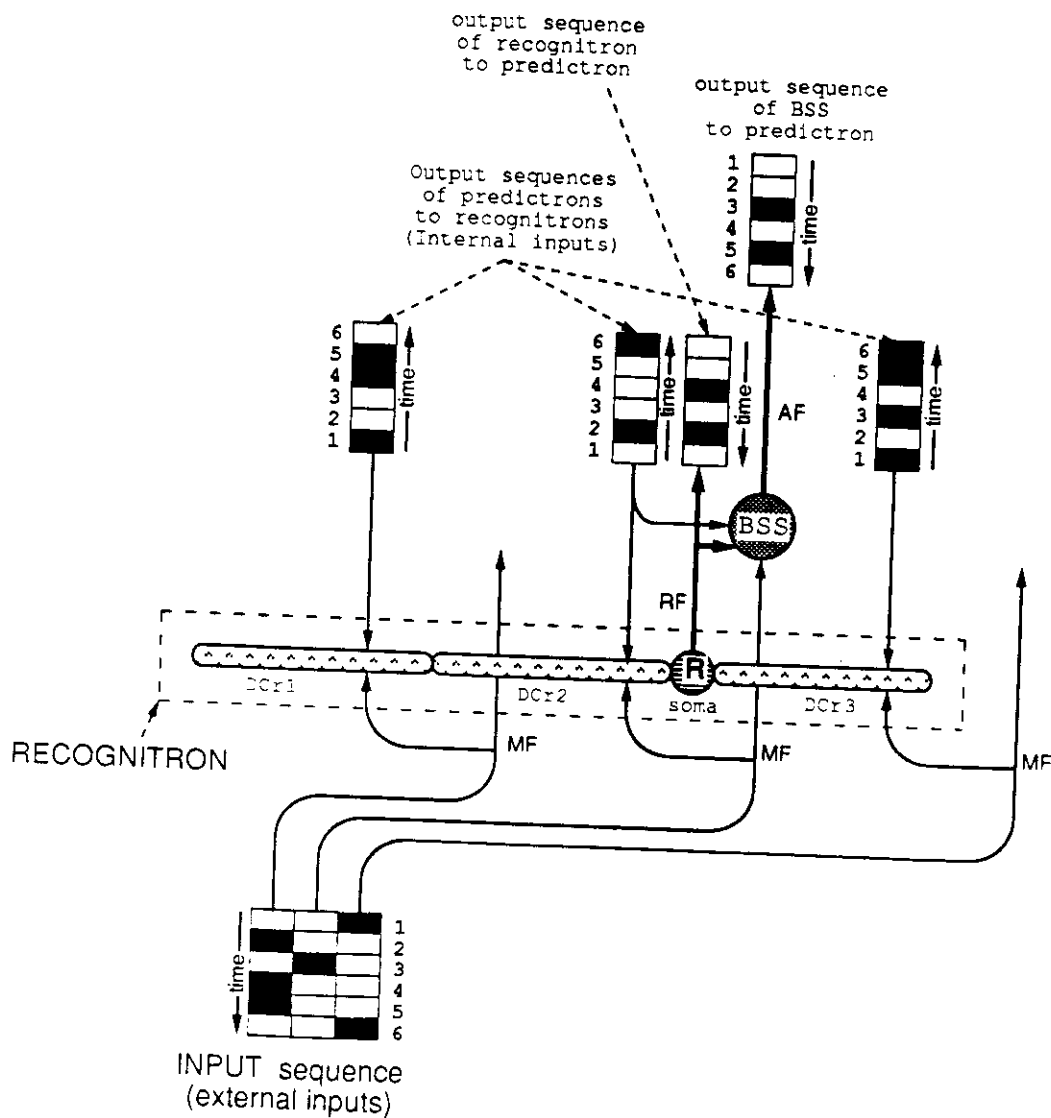


Figure 8.4: Recognitron & Bi-stable switch (BSS)

The recognitron and the BSS form a circuit. The BSS serves as a gate that passes either the external input bit (coming along a Mossy Fiber -- MF) or the internal input bit (coming along the predictron's axon) to its output. The recognitron functions as the controller of this gate. The control signal is passed to the BSS via a recognition fiber (RF). The state of the controller itself is a function of the similarity between an external input pattern (coming along MFs) and the internal input -- predicted pattern (coming via the predictrons' axons). The recognitron has a soma (R) and in the figure two dendritic branches, one to the left of the soma which contains 2 dendritic compartments (DC^rs), and one to the right with only one DC^r. The arrows at the end of the fibers (solid lines that make contacts with individual compartments), represent non-modifiable synapses of weight 1. Fibers that only by-pass the DC^rs without making synaptic contacts do not have arrows at the point of contact with the DC^rs.

8.1.3 Bi-Stable Switch

The model contains also a second type of *Bi-Stable Switch* (BSS) (Figure 8.4). There is one BSS per predictron. Each BSS gets three one-bit inputs and generates one output bit. Two of the inputs are "data" inputs, one bit from the external input sequence, and the other from the output of the corresponding predictron. The third input is a "control" bit which is the output of the corresponding recognitron coming along the RF. The function of a BSS is to select one of the two data inputs (i.e. the external input or the internal input) and to copy it to the output line. The input selection is controlled by the control bit from the recognitron. Effectively, the BSSs enable internal sequence completion. For instance, after a particular sequence has been learned, the BSSs turn off the external input and allow the outputs of the predictrons (i.e. the predictions made) to be used as inputs at the next B-cycle. The output of a BSS is a wire (axon). Each BSS's axon is partitioned into an ascending fiber (AF) and a parallel fiber (PF). The AF contacts all DCPs of the corresponding predictron via non-modifiable synapses with weights 1. The purpose of these synapses is to pass unchanged the value of the input bit coming along the AF to the DCPs. At a random location along the predictron's dendrite, each AF bifurcates to produce a PF which extends in the horizontal direction. The PF contacts the DCP of the corresponding predictron at the level of the bifurcation. This DCP is called a "seed-DCP" (Figure 8.5). There is one seed-DCP per predictron at a randomly selected location along its dendritic branch. (The KATAMIC model can function also with more than one seed-DCPs per predictron.)

8.1.4 A small scale example

A small scale example of the KATAMIC architecture is presented in Figure 8.6. The number of predictrons (**P**) in this network is four. The number of DCPs per predictron is six. The number of recognitrons is **R**, and there is one recognitron per predictron (i.e. **R = P = 4**). The predictrons are interconnected via the parallel fibers. Each PF contacts the same-level DCPs of all predictrons via non-modifiable weights (Figure 8.6). The purpose of this design is to distribute each bit of the input pattern across all predictrons in the network. The values of the weights of the PF synapses are set such that they decrease exponentially (decay constant T_s) with distance from the seed-DCP. This design ensures that the further away two predictrons are in the network, the less they influence each other.

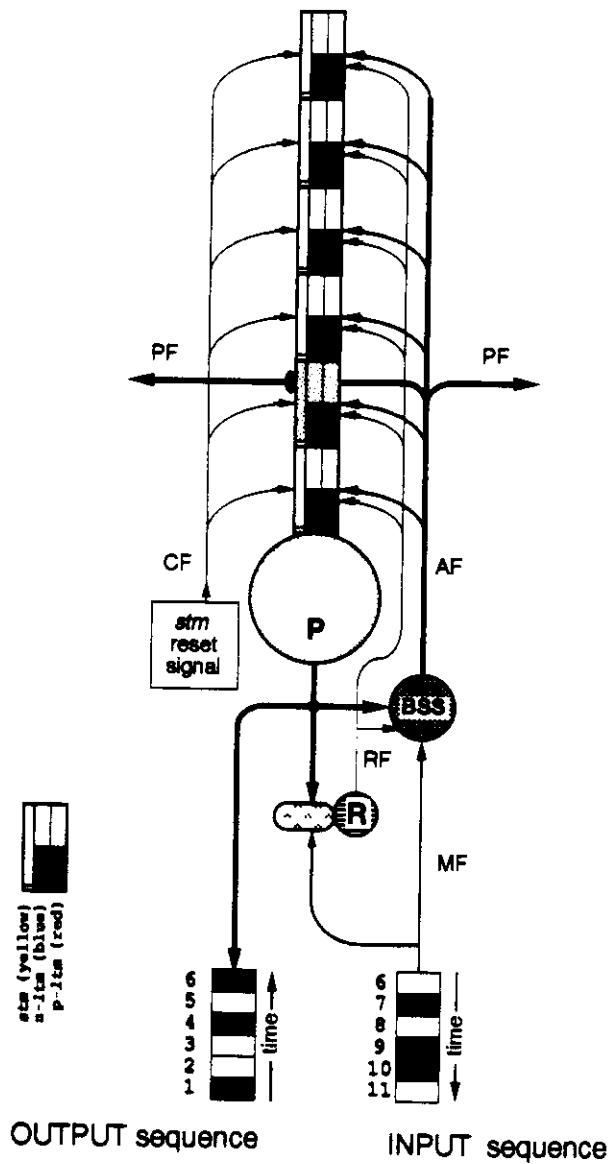


Figure 8.5: Canonic circuit of the KATAMIC model

The basic circuit of the KATAMIC network consists of one predictron, one BSS and one recognitron. A Parallel Fiber (PF) synapse on the seed-DCP is shown by a small dark oval. The *stm*, *n-ltm* and *p-ltm* within each DCP are shown as colored bars. The height of the bars relative to the height of the DCP indicate their values. The initial values of these variables are: *p-ltm* = 0.5 (medium-height red bars); *n-ltm* = 0.5 (medium-height blue bars); *stm* = 0.01 (tiny yellow bars).

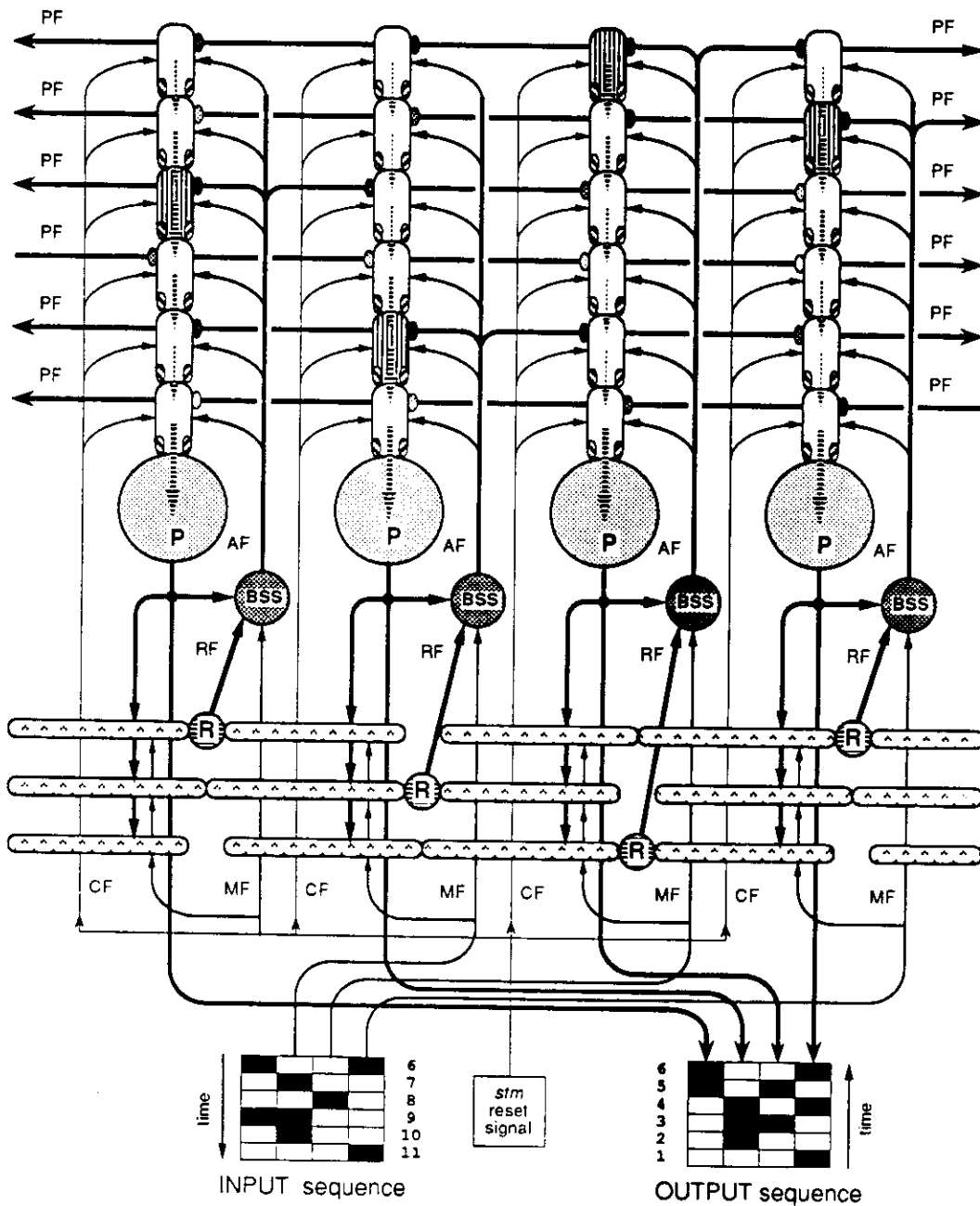


Figure 8.6: The KATAMIC model

A small scale example of the complete KATAMIC architecture. The network consists of four predictorons (corresponding number of BSSs, and recognitrons). Each predictor has six DCPs. Each recognitron has three DC's. Arrows at the contacting points between wires (fibers) and compartments represent non-modifiable synapses of weight 1. The rest of the arrows placed on wires are used to show the direction of signal flow.

8.1.5 Parameters & their values in a complete system

For a description of the parameters and variables used in the model and their typical values see Table 8.1. The predictrons are indexed by i , and the DCPs are indexed by j .

| symbol | description | typical values |
|--------------------|---|------------------------------------|
| parameters | | |
| P | number of <i>predictrons</i> | 64 |
| DP | number of DCPs per predictron | 512 |
| R | number of <i>recognitrons</i> | 64 |
| DR | number of DCPs per recognitron | 7 |
| T_s | time constant for <i>stm</i> spatial decay | -0.01 |
| T_t | time constant for <i>stm</i> temporal decay | -0.01 |
| b | <i>stm</i> update rate | 5.0 |
| c | <i>ltms</i> update (learning) rate | 1.0 |
| Θ^p | firing threshold of the predictrons | 0 |
| Θ^r | recognition threshold of the recognitrons | 5 |
| Θ^b | gating threshold of the BSS | 1 |
| variables | | |
| $p_{ij}(t)$ | positive <i>ltm</i> (<i>p-ltm</i>) | $p_{ij}(0) = 0.5 \in (0,1)$ |
| $n_{ij}(t)$ | negative <i>ltm</i> (<i>n-ltm</i>) | $n_{ij}(0) = 0.5 \in (0,1)$ |
| $s_{ij}(t)$ | <i>stm</i> | $s_{ij}(0) = 0.001 \in (0,1)$ |
| $\Delta s_{ij}(t)$ | <i>stm</i> injected | $\Delta s_{ij}(0) = 0.0 \in (0,1)$ |
| $s_{ij}^o(t)$ | <i>stm</i> normalization factor | $s_{ij}^o(0) = 0.001$ |
| $p_{ij}^o(t)$ | <i>p-ltm</i> normalization factor | $p_{ij}^o(0) = 0.001$ |
| $n_{ij}^o(t)$ | <i>n-ltm</i> normalization factor | $n_{ij}^o(0) = 0.001$ |
| $A_i^p(t)$ | Activation of predictron i | $\in (0,1)$ |
| $A_i^r(t)$ | Activation of recognitron i | $\in (0,DR)$ |
| $A_i^b(t)$ | Activation of BSS i | $\in (0,1)$ |
| $I_i(t)$ | Input to predictron i (= Output from BSS $_i$) | $\in (0,1)$ |
| $O_i^p(t)$ | Output of predictron i | $\in (0,1)$ |
| $O_i^r(t)$ | Output of recognitron i | $\in (0,1)$ |

Table 8.1: Typical values of KATAMIC parameters and variables

8.2 Network dynamics

8.2.1 The KATAMIC algorithm

During each B-cycle the model goes through one complete pass of the KATAMIC algorithm. This algorithm consists of the following 9 steps:

1. Get input:

Patterns of the input sequence are presented to the network in an orderly fashion -- one pattern per B-cycle. An input pattern $I(t)$ is passed to the predictrons (one bit per predictron -- $I_i(t)$). Each input bit reaches one BSS via the corresponding Mossy Fiber (MF) (Figure 8.6). After passing through the BSS each input bit is sent along an Ascending Fiber (AF) to the dendritic tree of the corresponding predictron. The connections made by the AF synapses are used to convey the value of the input (0 or 1) to the DCPs of the predictron. This information is used by the DCPs during learning.

2. Inject *stm* to DCPs:

The synaptic connections made by the PFs are used to "inject" *stm* ($\Delta s_{ij}(t)$) into the corresponding DCPs. The injected *stm* in a particular dendritic compartment is used to update the value of the *stm* in this compartment. At the seed dendritic compartment (seed-DCP) of each predictron which receives input 1 ($I_i(t) = 1$) the value of the injected *stm* is 1. In other words:

$$\forall i \in (1, P) \wedge j \in (1, DP) \text{ such that } \{ I_i(t) = 1 \wedge d_{ij} = 1 \} \text{ set } \Delta s_{ij}(t) = 1 \quad (8.1)$$

where: i is the i -th predictron; j is the j -th level DCP, $\Delta s_{ij}(0) = 0$, d_{ij} is the matrix of DCPs of all predictrons. For all seed-DCPs $d_{ij} = 1$, whereas for all non-seed-DCPs $d_{ij} = 0$.

In all DCPs of the neighboring predictrons that are at the same level as the seed-DCP of a given predictron, the values of the injected *stms* are equal to the synaptic weights made by the PF to these DCPs. Notice that the values of the PF synaptic weights are set such that they decay exponentially with a decay constant T_s . In other words:

$$\forall j \in (1, DP) \text{ such that } \Delta s_{i_0 j}(t) = 1 \text{ set } \Delta s_{ij}(t) = e^{-|i-i_0|T_s} \quad (8.2)$$

where: i_0 is the subset of predictrons which receive input 1 at B-cycle t .

3. Update *stm* at each DCP:

The increment of the *stm* value at each DCP carries information about the spatial relationships between the 1-bits in the input patterns. The further apart space-wise two 1-bits are, the smaller the value of the injected *stm*. The *stm* value in each DCP is updated using the injected-*stm* $\Delta s_{ij}(t)$ and the value of the *stm* at the previous B-cycle $s_{ij}(t-1)$ as inputs to a sigmoidal update function (8.3). This update function was chosen because it is monotonic and saturates. Also, if the value of the injected *stm* is zero, the current value of the *stm* remains the same as the previous value.

$$s_{ij}(t) = \sigma \left[s_{ij}^0(t) \hat{s}_{ij}(t-1) - \mathbf{b} \Delta \hat{s}_{ij}(t) \right] \quad (8.3)$$

where: $\sigma(x) = \frac{1}{1 + e^{-x}}$; \mathbf{b} is the *stm*-update rate which determines the effect of the injected *stm* in a given DCP on the previous *stm* value in this compartment; $s_{ij}^0(t)$ is a *stm* normalization factor which is chosen such that:

$$\text{IF } \Delta \hat{s}_{ij}(t) = 0 \text{ THEN } s_{ij}(t) = s_{ij}(t-1)$$

$$\text{This is ensured when: } s_{ij}^0(t) = \frac{1}{s_{ij}(t-1)} \ln \left(\frac{1}{s_{ij}(t-1)} - 1 \right) \quad (8.4)$$

4. Learning (Modify ltm):

Learning in the KATAMIC model means changing (i.e. updating) of the *p-ltm* or the *n-ltm* in each DCP as a result of the current value of the *stm* in this DCP and a number of conditions. The basic idea is to “imprint” the momentary pattern of the *stm* (over the DCPs) onto the pattern of one of the *ltms*. The *p-ltm* or the *n-ltm* of each predictron (but not both) is updated at each B-cycle using a two-stage update rule:

- **Stage 1: (Learning condition):** Predictrons learn only when their expectations (i.e. the prediction which they have generated) fail. In other words, it is not necessary for a system to over-learn a task if it already can perform the task correctly. This learning condition was proposed by Rosenblatt in his Perceptron model (Rosenblatt, 1958) and was used by Rescorla and Wagner (Rescorla and Wagner, 1972) as the basis of their learning model. The status of the recognition process is reflected in the activation value of each recognitron (0 -- correct prediction, 1 -- incorrect prediction) and is conveyed to the DCPs of each predictron via the collaterals of the Recognition Fibers (RFs). Each predictron (and respectively each DCP) uses the provided recognition signal to evaluate the learning condition. If the prediction is correct (recognition signal = 0), then no learning happens and stage 2 is bypassed. If, on the other hand, the prediction is incorrect then, depending on the type of prediction failure, the second stage of the learning rule is executed:

In the KATAMIC model there are only two possible ways for a prediction made by a predictron to fail:

(1) wrong-0-prediction occurs when the corresponding input bit from the currently processed input pattern is 1 and the prediction made by the predictron at the end of the previous B-cycle is 0.

(2) wrong-1-prediction occurs when corresponding input bit from the currently processed input pattern is 0 and the prediction made by the predictron at the end of the previous B-cycle is 1.

The particular type of prediction failure is mediated in each DCP via two wires:

(1) The RF signals the fact that a prediction has failed (value = 1)

(2) The AF signals 0 or 1 which specifies which kind of prediction failure has occurred.

- **Stage 2: (Modification function):** If the learning condition is met, and IF:

- **wrong-0-prediction:** Then the *p-ltm* value in each DCP of this predictron are modified (learning occurs) as follows:

$$\forall i \in (1,P) \wedge j \in (1,DP) \quad \text{IF } \left\{ (I_i(t) = 1) \wedge (O_i(t-1) = 0) \right\}$$

$$\text{THEN set } p_{ij}(t) = \sigma \left[p_{ij}^0(t) p_{ij}(t-1) - c s_{ij}(t) \right] \quad (8.5)$$

where: c is the learning rate (i.e. *ltms* update rate) and $p_{ij}^0(t)$ is the *p-ltm* normalization factor which is chosen such that:

$$\text{IF } s_{ij}(t) = 0 \quad \text{THEN } p_{ij}(t) = p_{ij}(t-1)$$

$$\text{This is ensured when: } p_{ij}^0(t) = \frac{1}{p_{ij}(t-1)} \ln \left(\frac{1}{p_{ij}(t-1)} - 1 \right) \quad (8.6)$$

• **wrong-1-prediction:** Then the *n-ltm* value in each DCP of this predictron are modified as follows:

$$\forall i \in (1,P) \wedge j \in (1,DP) \quad \text{IF } \left\{ (I_i(t) = 0) \wedge (O_i(t-1) = 1) \right\}$$

$$\text{THEN set } n_{ij}(t) = \sigma \left[n_{ij}^0(t) n_{ij}(t-1) - c s_{ij}(t) \right] \quad (8.7)$$

where: $n_{ij}^0(t)$ is the *n-ltm* normalization factor which is chosen such that:

$$\text{IF } s_{ij}(t) = 0 \quad \text{THEN } n_{ij}(t) = n_{ij}(t-1)$$

$$\text{This is ensured when: } n_{ij}^0(t) = \frac{1}{n_{ij}(t-1)} \ln \left(\frac{1}{n_{ij}(t-1)} - 1 \right) \quad (8.8)$$

5. *ltm* resource maintenance -- forgetting:

The purpose of this step is to assure that the total amount of *p-ltm* per predictron (and correspondingly that of *n-ltm*) is kept constant at each B-cycle. This is necessary since we do not want the amount of *ltm* to reach a saturation point after the memory has learned several sequences. The mechanism has two side effects: (1) It allows the KATAMIC memory to learn sequences only from positive examples. Each sequence (a positive example) is used effectively as a weak negative evidence for all other sequences, as it will be seen by the description of the resource maintenance / forgetting mechanism provided below. (2) It gives the memory the ability to forget previously learned sequences which are rarely used. The application of this mechanism results in weakening of the traces which old and rarely seen sequences left in the *ltm*.

The resource maintenance / forgetting mechanism functions in the following way. The sequence which is being processed is encoded within the dendritic trees of the predictrons as a set of *stm* vectors (one per predictron). Each of these *stm* vectors is used in the update of the *p-ltm* or the *n-ltm* depending on the type of prediction failure (see the previous step of the algorithm). The total amount of *p-ltm* and *n-ltm* values for each predictron i are respectively ($DCP \times p_{ij}(0)$) and ($DCP \times n_{ij}(0)$). The maintenance of *ltm* resources involves *p-ltm* and *n-ltm* redistribution among the dendritic compartments of each predictron. Intuitively speaking, the increase of *ltm* in the few dendritic compartments where there is a big amount of *stm* is compensated by a decrease of *ltm* in

the rest of the compartments of the predictron where there is a relatively low amount of *stm*. The actual resource redistribution is done in the following manner.

$$\forall i \in (1,P) \text{ compute } \dot{p}_{ij}(t) = p_{ij}(t) \frac{p\text{-}l\text{tmpp}}{DP} \sum_{k=1} p_{ik}(t) \quad (8.9a)$$

$$\forall i \in (1,P) \text{ compute } \dot{n}_{ij}(t) = n_{ij}(t) \frac{n\text{-}l\text{tmpp}}{DP} \sum_{k=1} n_{ik}(t) \quad (8.9b)$$

where:

p-ltmpp is a constant which specifies the total amount of positive long-term memory per predictron ($p\text{-}l\text{tmpp} = DP \times p_{ij}(0)$), and

n-ltmpp is a constant which specifies the total amount of negative long-term memory per predictron ($n\text{-}l\text{tmpp} = DP \times n_{ij}(0)$)

6. Temporal encoding:

This processing step produces an encoding of the temporal order of the 1-bits in the successive patterns by constructing a temporal history of the successive 1-bit inputs to a given predictron which is reflected in the *stm* pattern in the dendritic branch of each predictron. This is accomplished by shifting the value of the *stm* in each DCP to the next dendritic compartment towards the soma (replacing the previous *stm* value) and decaying it with a temporal time constant T_t . In a predictron, the later in time a 1-bit input arrives with respect to the previous 1-bit input, the smaller is the *stm* trace left by the previous input.

$$\dot{s}_{ij}(t) = s_{ij-1}(t) e^{-Tt} \quad (8.10)$$

The boundary condition is: for $j = 0$ set $\dot{s}_{i0}(t) = 0$ or $\dot{s}_{i0}(t) = s_{iD}(t)$ (used currently).

7. Predict next input:

At each B-cycle predictrons use the relation between the *ltms* and the *stm* in their DCPs to make predictions about the next pattern in the sequence. A prediction is generated by comparing the shifted (one DCP towards the soma) pattern of the *stm* values distributed in the DCPs of a predictron to the patterns of the *p-ltm* & *n-ltm*. For instance, if the *stm* pattern is more similar to the *p-ltm* than to the *n-ltm*, then the predictron fires, otherwise it is silent. In a given predictron, the comparative similarity of the *p-ltm* or the *n-ltm* dendritic distribution to the *stm* distribution is learned through experience.

The output pattern $O(t)$ of the network at each time cycle is actually a prediction of the input vector $I(t+1)$ at the next time cycle. $O(t)$ is generated as follows:

(1) Calculate the somatic activation as the dot-product of the *stm* and the per DCP difference of the *ltms* (*p* & *n*) for each of the predictrons. The result of this calculation shows whether the shifted *stm* pattern is more similar to the *p-ltm* pattern or to the *n-ltm* pattern.

(2) Make the decision to fire the predictron (1-prediction) or not (0-prediction) by comparing its activation to the threshold Θ^P .

$$\forall i \in (1,P) \text{ compute } A_i^P(t) = \sum_{j=1}^{DP} s_{ij}'(t) \left(p_{ij}'(t) - n_{ij}'(t) \right) \quad (8.11)$$

$$\text{IF } A_i^P(t) \begin{cases} \geq \Theta^P & \text{THEN set } O_i^P(t) = 1 \text{ (predictron fires)} \\ < \Theta^P & \text{THEN set } O_i^P(t) = 0 \text{ (predictron silent)} \end{cases} \quad (8.12)$$

8. Attempt sequence recognition:

At this step of the algorithm each recognitron computes its somatic activation as the sum of the results of an XOR function (FXOR) applied to the two inputs of each DC^r , the next "external" input ($I_{ext_i}(t+1)$) coming along the corresponding MF, and "internal" input $O_i^P(t)$ -- the output of the predictron at this B-cycle (8.13).

$$\forall i \in (1,R) \text{ compute } A_i^r(t) = \sum_{k=1}^{D^r} F_{\text{XOR}} \left(I_{ext_{i-k}}(t+1) \wedge O_{i-k}^P(t) \right) \quad (8.13)$$

The recognition mechanism is designed such that the decision about which input (external or internal) should be passed to each individual predictron at time $t+1$ does not depend only on the correctness of the prediction made by the predictron itself. It also depends on how good are the predictions which are made by the immediate neighbors of each predictron. The acceptable degree of performance of the neighbors is reflected in the magnitude of the recognitron's threshold -- a value which can be set by the user (8.14).

$$\text{IF } A_i^r(t) \begin{cases} \geq \Theta^r & \text{THEN set } O_i^r(t) = 1 \text{ (recognitron fires)} \\ < \Theta^r & \text{THEN set } O_i^r(t) = 0 \text{ (recognitron silent)} \end{cases} \quad (8.14)$$

The purpose of this design was to provide some flexibility for the process of recognition. The design also allows the system to operate (if set accordingly by the user) in two extreme modes of operation:

(1) *Local recognition* -- This is a case when the decision about which input (internal or external) should be used by a predictron depends only on its own performance at the previous cycle. In this extreme situation each recognitron needs to have only one dendritic compartment (DC^r).

(2) *Global recognition* -- This is a case when, unless all predictrons in the network have made correct predictions (i.e. a whole pattern in the sequence has been recognized bit by bit), none of the predictrons is allowed to use an internal input at the next time step. In this extreme situation we need as many DC^r 's per recognitron as there are predictrons. Actually, since the decision for all

prediction is the same, the system can operate with only one recognitron connected to all BSSs which has $DC^r = P$.

In general, however, the system will operate in between these two extreme cases. As a result the learning in individual predictiontrons will depend on the quality of predictions made in a small neighborhood around each predictiontron -- a measure provided by the recognitron.

9. Generate next input:

The input pattern for the next B-cycle ($I(t+1)$) is generated by the BSSs -- one bit per BSS. As was described above, each BSS takes 3 inputs, 2 "data" inputs (one "external" $I_{extj}(t+1)$ which comes along the MF, and one "internal" $I_{intj}(t+1)$ which is provided by the axon of the corresponding predictiontron -- the $O_j^p(t)$), and a "control" input from the corresponding recognitron $O_j^r(t)$. All of these inputs take values 0 or 1. The BSS functions as a gating device which passes either the external or the internal input to the Ascending Fiber. It has an activation value $A_j^b(t)$ which is updated (8.15) at each B-cycle by the control signal from the corresponding recognitron $O_j^r(t)$. The activation decays with a decay constant T_b .

$$\forall i \in (1, P) \text{ compute } A_i^b(t) = (A_i^b(t-1) + O_i^r(t)) e^{-T_b} \quad (8.15)$$

At each B-cycle the activation is compared to a threshold Θ^b and if the activation is larger, then the BSS passes the external input along the AFs whereas if it is smaller -- the internal input is passed through the gate (8.16).

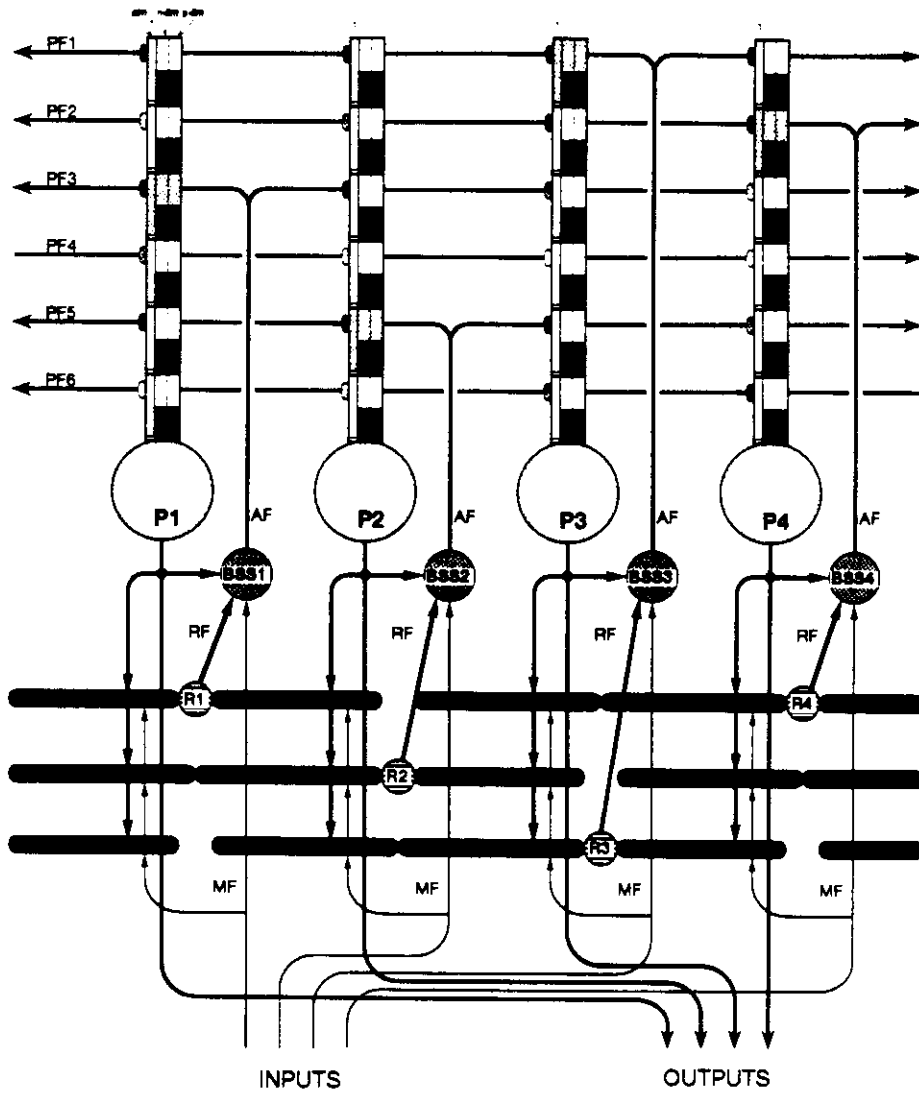
$$\forall i \in (1, R) \quad \text{IF } A_i^b(t) \begin{cases} < \Theta^b \\ > \Theta^b \end{cases} \text{ THEN set } I(t+1) = \begin{cases} I_{intj}(t+1) \\ I_{extj}(t+1) \end{cases} \quad (8.16)$$

Intuitively speaking, the dynamics of the BSSs were chosen such that they allow the network to start using internal inputs (predictions) as soon as the predictions become correct. Such inputs are being used as long as the predictions which they produce keep being correct. (Notice that if the predictions are correct, it is irrelevant which inputs are used -- internal or external, since they are the same.) However, the important feature which the BSSs provide is that if internal inputs are in use, the network does not switch back to using external inputs as soon as the predictions which the internal inputs generate become wrong (for whatever reason). Instead, the model keeps using internal inputs for a few more B-cycles. In other words, it keeps "singing its own song" despite the fact that it is false. The duration of the transition stage from using internal inputs to switching back to the use of external inputs depends on the decay constant T_b of the BSSs. Setting the value of T_b high establishes a network that is not very responsive to the external input and has the tendency to "sing its own song", whereas setting this value low results in a network that is "eager" to adapt its performance to the environment (the external input).

8.2.2 Illustration of KATAMIC's dynamics

To illustrate the dynamics of the KATAMIC model as a whole, this section provides a diagrammatic illustration in which the model goes through one B-cycle. A 4-bit wide pattern of an input sequence is being processed. To make the description more concise, some of the 9 steps of the KATAMIC algorithm have been combined and the complete B-cycle (i.e. 9 algorithmic steps) is shown

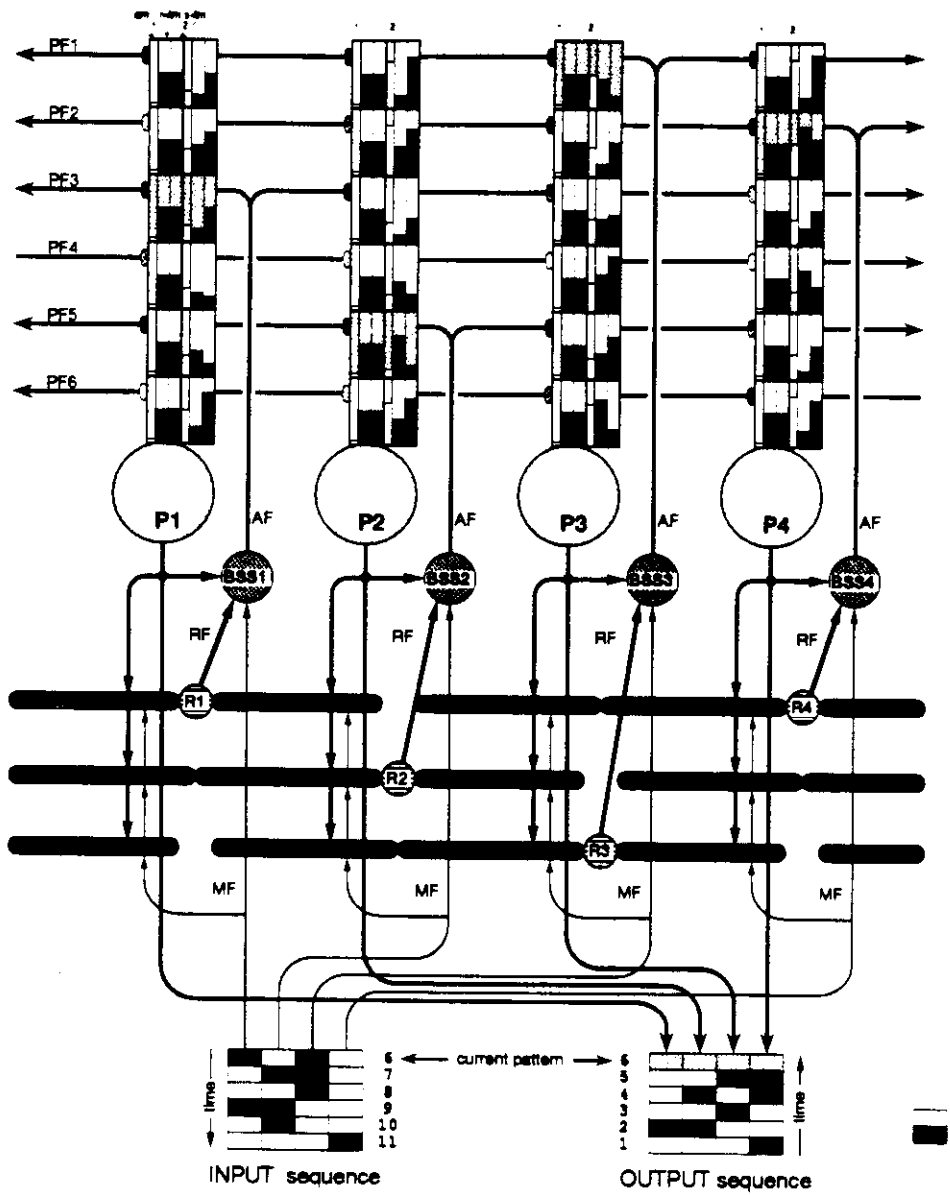
diagrammatically in 5 consecutive plates. In each plate, the values of the three state variables (*stm*, *n-ltm*, and *p-ltm*) are shown in color -- yellow, blue, and red respectively. Within each dendritic compartment of a predictron these variables are shown as color bars with heights encoding their values. A color bar with a height equal to the height of the dendritic compartment has a value 1. A color bar with half the height encodes a value of 0.5. In each consecutive plate a copy of the whole dendritic branch of each predictron is placed to the right. Color coded within this copy are the new values of the state variables which reflect the changes occurred during the corresponding algorithmic steps. These changes are also described in the accompanying figure captions.



1. Basic KATAMIC architecture (in naive state)

Figure 8.7.1: KATAMIC dynamics: Plate 1

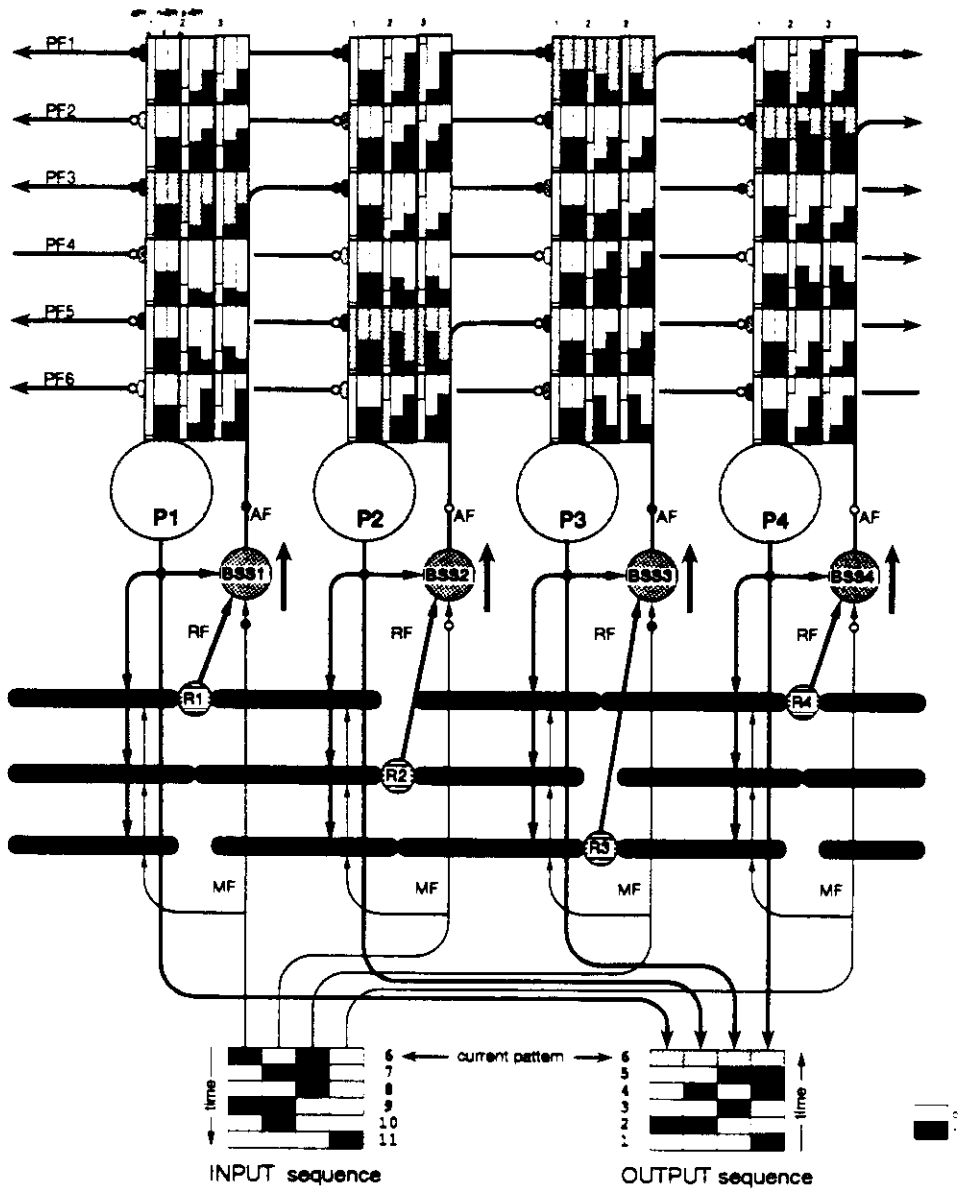
The KATAMIC network in its initial state. The values of the *n-ltm* and *p-ltm* within each DCP are 0.5 (medium-height blue and red bars) while the values of the *stm* is 0.01 (tiny yellow bars). The seed-DCPs are gray shaded. The values of the activations in the dendritic compartments of the recognitrons are set to 1 (black) which sets up the network in a receptive state (the BBSs can pass the external input to the AFs).



1. Basic KATAMIC architecture (in naive state)
2. Current state (after several sequences have been learned)

Figure 8.7.2: KATAMIC dynamics: Plate 2

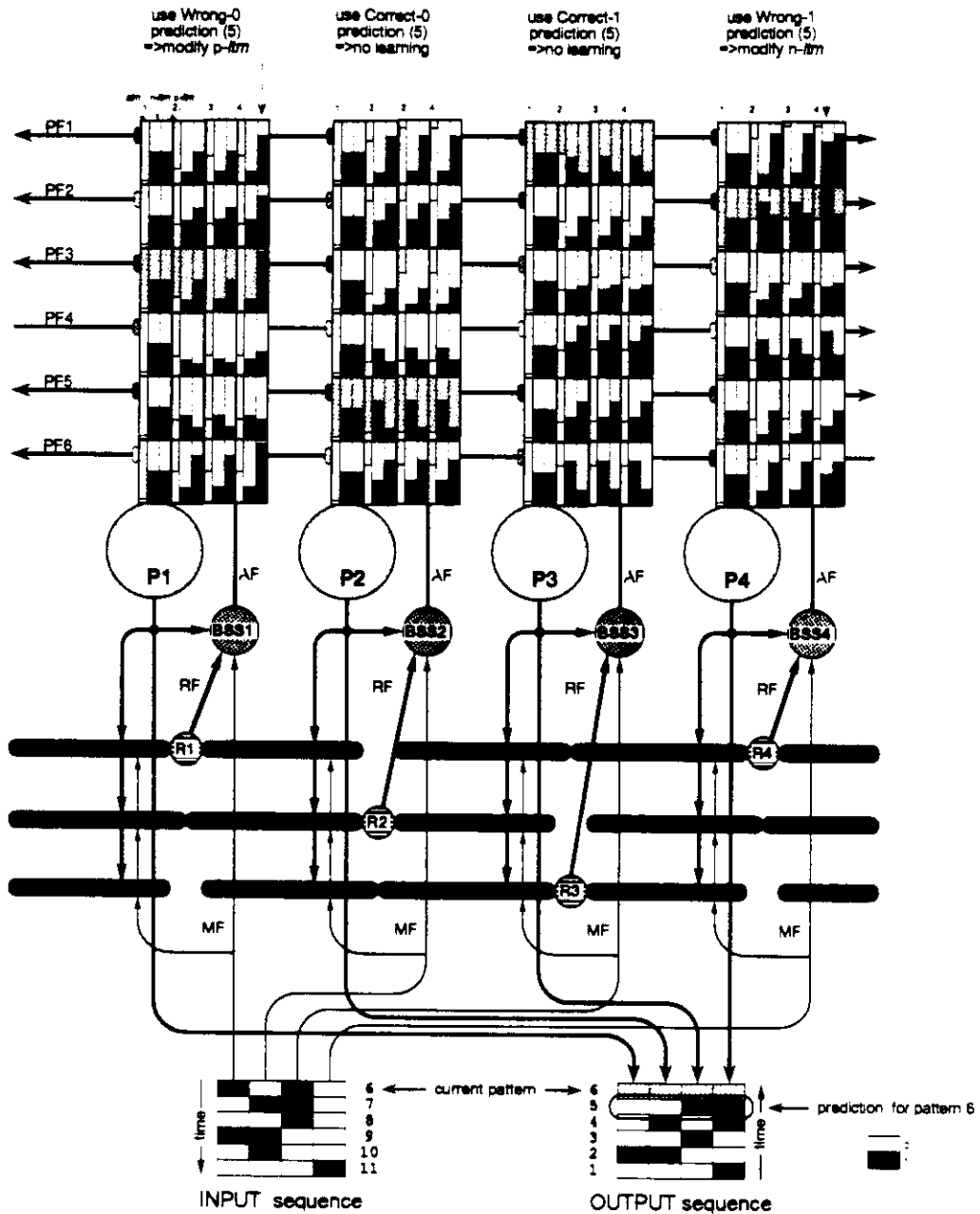
The dendritic trees drawn to the right of their original position show the KATAMIC network after it has learned few sequences and is currently processing of pattern 6 of a particular input sequence. The sequence is 11 patterns long and the remaining 6 patterns (from 6 to 11) are shown to the left whereas the outputs (predictions) of the network generated by the first 6 patterns are shown to the right. Black rectangles in the sequences encode 1-bits while white rectangles encode 0-bits. The output of the current B-cycle has not yet been generated (the corresponding bits are shaded gray). Notice that compared to the naive state the values of the *stm* & *ltm* state variables are different.



1. Basic KATAMIC architecture (*in naive state*)
2. Current state (*after several sequences have been learned*)
3. Get input; inject *stm*; Update *stm*

Figure 8.7.3: KATAMIC dynamics: Plate 3

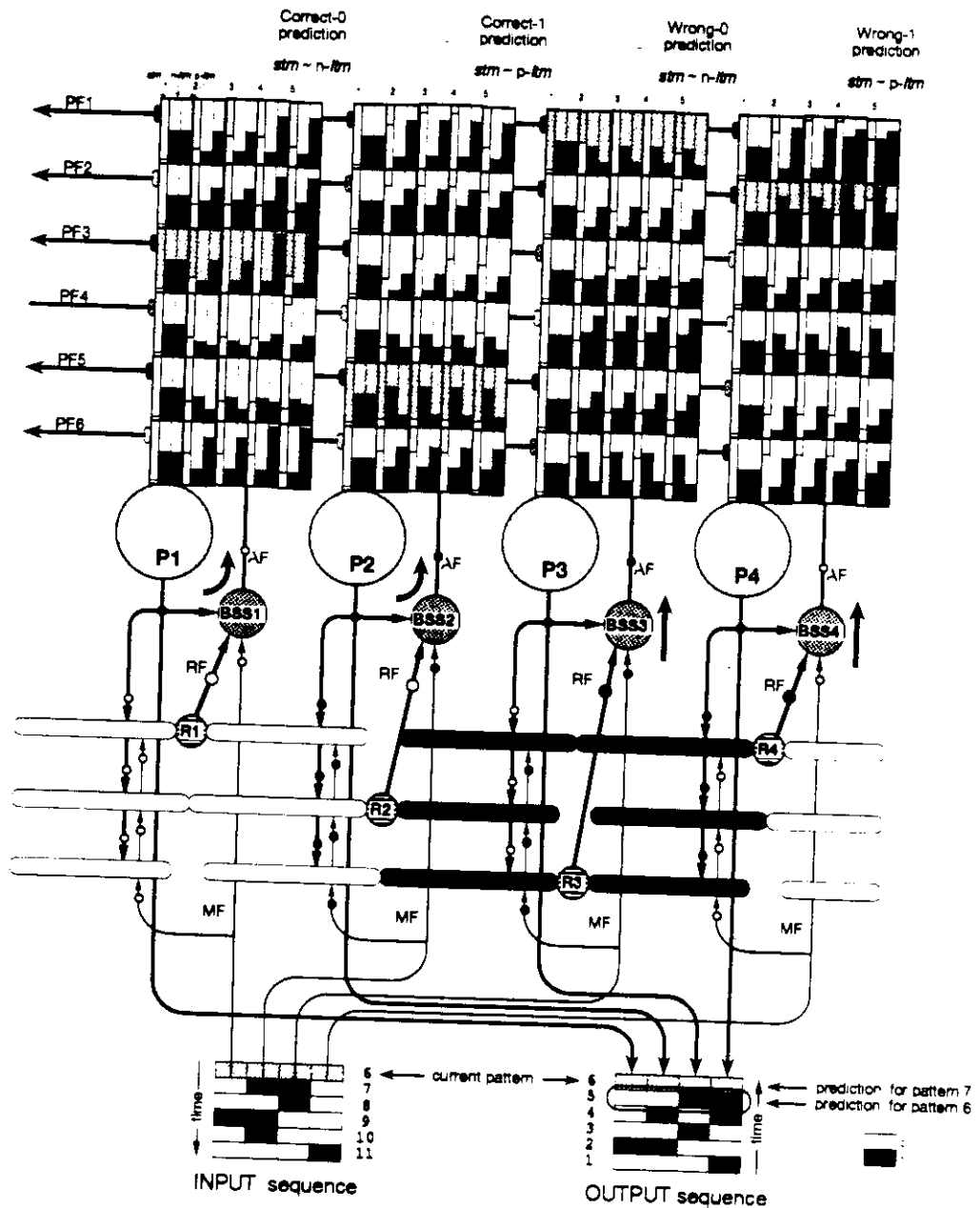
This plate illustrates steps 1, 2 & 3 of the KATMIC algorithm. (1) The network gets the input pattern (pattern 6) (formula 8.1). The network has not produced yet an output so pattern 6 in the output sequence is shaded gray. We assume that the states of the 4 BSSs are such that the input bits are passed directly (shown by the thick arrows to the right of the BSSs) along the AFs. The values of the individual bits are shown with black or white circles placed on the AFs and also next to the synaptic contacts made by the corresponding PFs. Notice that the values carried by PF4 and PF6 (which presumably originate from other BSSs not shown in the drawing) are assumed to be 0. (2) The input bits are injected in the corresponding DCPs (injected-*stm*). Each 1-bit (along PF1 & PF3) is multiplied by the value of the synaptic weight (encoded by different shades of gray -- black is 1 and lighter is smaller than 1) made by the corresponding PF to the 4 DCPs. This multiplication yields the values of the injected-*stm*. (formula 8.2). (3) The *stm* in each DCP is updated (formula 8.3). For instance, the *stm* value at the level of PF1 in P3, and at the level of PF3 at P1 have become 1 and the values of the *stm* in the neighboring DCPs at these levels (PF1 & PF3) have also increased. The values of the *stm* in the rest of the DCPs have remained the same due to 0 injected-*stm*.



1. Basic KATAMIC architecture (in naive state)
2. Current state (after several sequences have been learned)
3. Get input; Inject stm ; Update stm
4. Modify ltm ; Resource management (forgetting)

Figure 8.7.4: KATAMIC dynamics: Plate 4

This plate illustrates steps 4 & 5 of the KATAMIC algorithm. (4) The *ltms* within the DCPs are updated. To demonstrate how the *ltm*-update process works in the 4 possible situations (formulas 8.5-8) we assume that during processing of the previous step of the sequence (step 5) P1 has made a wrong-0 prediction ==> p-ltm is modified; P2 has made a correct-0 prediction ==> no learning; P3 has made a correct-1 prediction ==> no learning; and P4 has made a wrong-1 prediction ==> n-ltm is modified. The update of p-ltm in P1 and n-ltm in P4 has been done in all of their DCPs. (5) Resource management -- forgetting (formulas 8.9a,b).



1. Basic KATAMIC architecture (*in naive state*)
2. Current state (*after several sequences have been learned*)
3. Get input; Inject *stm*; Update *stm*
4. Modify *ltm*; Resource management (*forgetting*)
5. Temporal encoding (shift *stm*); Make a prediction; Attempt recognition (input 7); Generate next input

Figure 8.7.5: KATAMIC dynamics: Plate 5

This plate illustrates steps 6, 7, 8, and 9 of the KATAMIC algorithm. (6) Temporal encoding -- the *stm* values in each DCP are shifted to the next DCP towards the soma and decayed (formula 8.10). The *stms* at level PF6 are shifted/decayed to the DCPs at level PF1. (7) Make prediction -- each predictron computes its output (formulas 8.11-12). For illustrative purposes I assume that for P1 the *stm* vector is more similar to the *n-ltm* ==> 0-bit output; for P2 the *stm* vector is more similar to the *p-ltm* ==> 1-bit output; for P3 the *stm* vector is more similar to the *n-ltm* ==> 0-bit output; for P4 the *stm* vector is more similar to the *p-ltm* ==> 1-bit output. The values of the output bits are shown with small black or white circles placed at the synaptic contacts (small black arrows) made by the axons of the predictrons to the corresponding DCr's. (8) Attempt recognition -- at this step the network reads the next pattern from the input sequence (pattern 7). The values of input bits are shown by small black or white circles placed at the synaptic contacts made by the MFs to the corresponding DCr's and BBSs. Each DCr computes an XOR of its inputs. DCr's where the XOR yielded 1 are shaded black and the rest remain white -- value 0. The results of these computations are summed for each recognitron to yield its activation value (not shown graphically) (formula 8.13). These values are further thresholded ($\Theta^r = 1$) and the outputs of the recognitrons whose activations have exceeded the threshold (R3 & R4) have been set to 1 (formula 8.14). The outputs of the recognitrons are shown as black or white circles placed on the RF fibers. P1 has generated a correct-0 prediction; P2 has generated a correct-1 prediction; P3 has generated a wrong-0 prediction; and P4 has generated a wrong-1 prediction. (9) Generate next input -- each BSS reads its control signal from the corresponding RF and depending on its value resets the state of the associated gate (for simplicity we assume here that T_b is very low). BSS1 & BSS2 have received 0s and therefore they pass their "internal" inputs to the AF fibers (shown by the thick curved arrows and the small circles placed on the AF fibers (formula 8.16). BSS3 & BSS4 have received 1s and therefore they pass their "external" inputs to the AFs. At this point the network completes one full B-cycle.

8.2.3 Signal Flow in the KATAMIC model

To illustrate the dynamics of the KATAMIC network a signal-flow diagram portraying the network in several consecutive B-cycles is shown in Figure 8.8. It shows:

(1) The signals flowing along the major wires (MF -- mossy fiber, AF/PF -- ascending/parallel fiber, RF -- recognition fiber, axon of predictron -- prediction, CF -- climbing fiber). There is a set of these wires for each canonical circuit (predictron, recognitron, BSS). The values of the signals on the individual wires (0 or 1) are shown as low and highs states of the horizontal lines.

(2) Values of the *stm*, the *p-ltm*, and the *n-ltm* are shown for the seed-DCP of one predictron. They are encoded as different shades of gray such that white = 0, and black = 1.

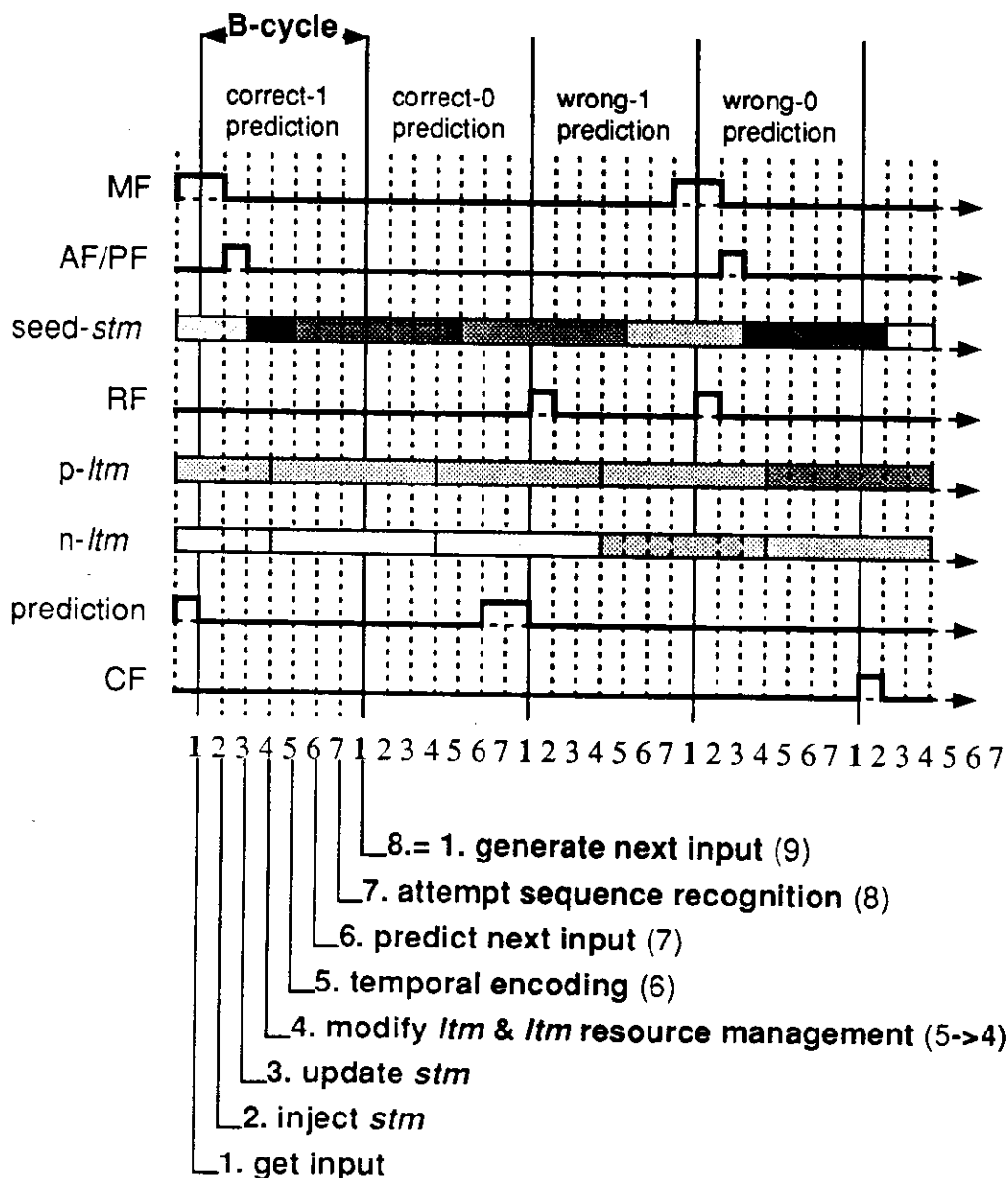


Figure 8.8: Signal flow in the KATAMIC model

The diagram illustrates the KATAMIC memory during processing of 4 consecutive steps of a sequence -- B-cycles. (Notice that these are not necessarily the first 4 steps of a sequence, that is why the *stm*, the *p-ltm*, and the *n-ltm* are shown to have some non-zero values at the beginning). The chart shows what happens during each of the four possible cases with respect to the types of predictions made (correct-1-prediction, correct-0-prediction, wrong-1-prediction, wrong-0-prediction -- see step 4 of the KATAMIC algorithm for definitions). The order in which these cases have been presented is for illustrative purposes only. Each B-cycle in the figure, during which a single pattern is processed, is divided into 7 steps (vertical dotted lines). These steps correspond to the steps of the KATAMIC algorithm. Steps 4 (modification of *ltm*) and 5 (*ltm* resource management) of the algorithm are lumped in step 4 of the flow-chart. Also, step 9 (generation of next input) is equated with step 1 (getting the next input).

8.3 Implementation on the Connection Machine

The *LISP code which implements the KATAMIC memory on the CM-2 Connection Machine is given in Appendix B.2.

8.4 Simulations

Mathematical analysis of the behavior of a multi-parametric non-linear dynamic system such as the one described here is not straightforward. Computer simulations provide a reasonable alternative. This is the approach that was taken to analyze the performance of the KATAMIC model. Of course, to examine the behavior of the system for all possible combinations of parameters is impractical and computationally expensive. Therefore, I focus only on a small set of simulations designed to test the most critical characteristics of the KATAMIC memory. These are: (1) speed and convergence of learning, (2) dependence of performance on network parameters, (3) memory capacity and interference between memory traces. The results of some of these experiments are described below.

Most of the experiments with the KATAMIC memory follow a common experimental design. A set of pattern-sequences of equal length (each pattern has the same width) is repeatedly presented to the network. The patterns forming the sequences are randomly generated and have, a priori, a specified "1-bit-density", (i.e. percentage of the randomly selected 1-bits of the total number of bits in a sequence). Therefore, for a given experiment, all sequences that have the same density are statistically equivalent and monitoring the network's performance on one of these sequences rather than on all of them is sufficient.

8.4.1 Performance dependence on the T_s & T_t decay constants

In a set of experiments I tested the dependence of the KATAMIC's performance on some network parameters; namely, the temporal and spatial decay constants (T_t and T_s). T_s specifies how individual patterns are distributed among the predictions, while T_t specifies the *stm* duration (i.e. for how many B-cycles its value persists in the dendritic tree before it decays to 0). In these experiments, ten sequences of length 10 (64 bits wide) and density 10% were presented to the memory. The set of all sequences was presented 10 times in a sequential order. The quality of the

predictions made for one (the first) of these sequences was monitored. For each repetition of this sequence the ratios **match/goal** and **spurious/goal** were recorded at each B-cycle. The measures: **goal**, **match** and **spurious** are defined as follows:

- **goal** is the total number of 1-bits in the input pattern at B-cycle (t).
- **match** is the number of 1-bits in the output pattern (i.e. prediction generated) at B-cycle (t-1) that match the 1-bits in the input pattern at B-cycle (t). From this definition it is evident that **match/goal** \leq 1.
- **spurious** is the number of 1-bits in the prediction that do not **match** the 1-bits in the input pattern at B-cycle t. Therefore the number of spurious is in the range (0, P - goal). The ratio of spurious/goal is in the range (0, (P-goal)/goal)

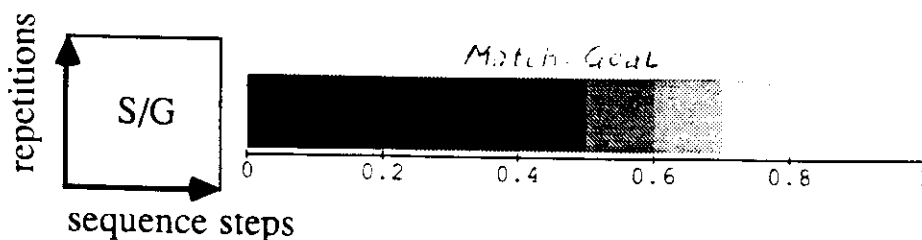
The criteria for correct (good) performance are: (a) **match/goal** equal or close to 1, i.e. none or very few misses, (b) **spurious** significantly lower than the value of **match** (e.g., 10% of the **match**).

The basic experiment was repeated for values for the spatial (Ts) & temporal (Tt) decay constants varying over 6 orders of magnitude (-10^{-5} , -10^{-4} , -10^{-3} , -10^{-2} , -10^{-1} , and -1), i.e. $6 \times 6 = 36$ experiments. The results are presented as sets of density-plots on Figure 8.9a (**match/goal**) and Figure 8.9b (**spurious/goal**). Within a wide range of values of the Ts & Tt constants (-10^{-5} , -10^{-2}), the quality of the predictions made improves rapidly during the first 3 to 5 repetitions (Figure 8.9a) while at the same time the **spurious** goes practically to zero (Figure 8.9b). Also, at each repetition the **spurious** decreases after the first few patterns of the sequence -- the time necessary for the memory to "recognize" the sequence (Figure 8.9b). Notice also that: (1) the pattern-by-pattern predictions made during the first exposure of the memory to a sequence are random, (2) the last (10th) prediction made for any of the sequences during any of the repetitions is irrelevant since the memory has not been exposed to an eleventh pattern. While any measure of the performance will depend on the set of performance criteria employed (e.g., speed of learning or quality of match), on the basis of these results it is safe to say that the memory operation is robust (i.e. good performance is maintained within a very wide range of Ts & Tt values).

8.4.2 Effects of "1-bit-density" of the input sequences

It is to be expected that the 1-bit-density of the processed sequences will have an effect on the performance of the network. To examine these effects, two sets of experiments were performed.

The first set was composed of eight separate experiments. It tested how performance is affected when the sequences stored are all of the same density within an experiment but of different 1-bit-densities (e.g., 10%, 20%, ..., 80%.) between experiments (1 to 8). In each of the eight experiments, a set of 10 sequences of the same 1-bit-density was presented 10 times to a naive system. For each experiment one arbitrarily chosen sequence from the 10 sequences in the set was monitored. Notice that the sequences used in each experiment are statistically equivalent. The results of these experiments are shown in Figure 8.10. The basic observations to be made here are:



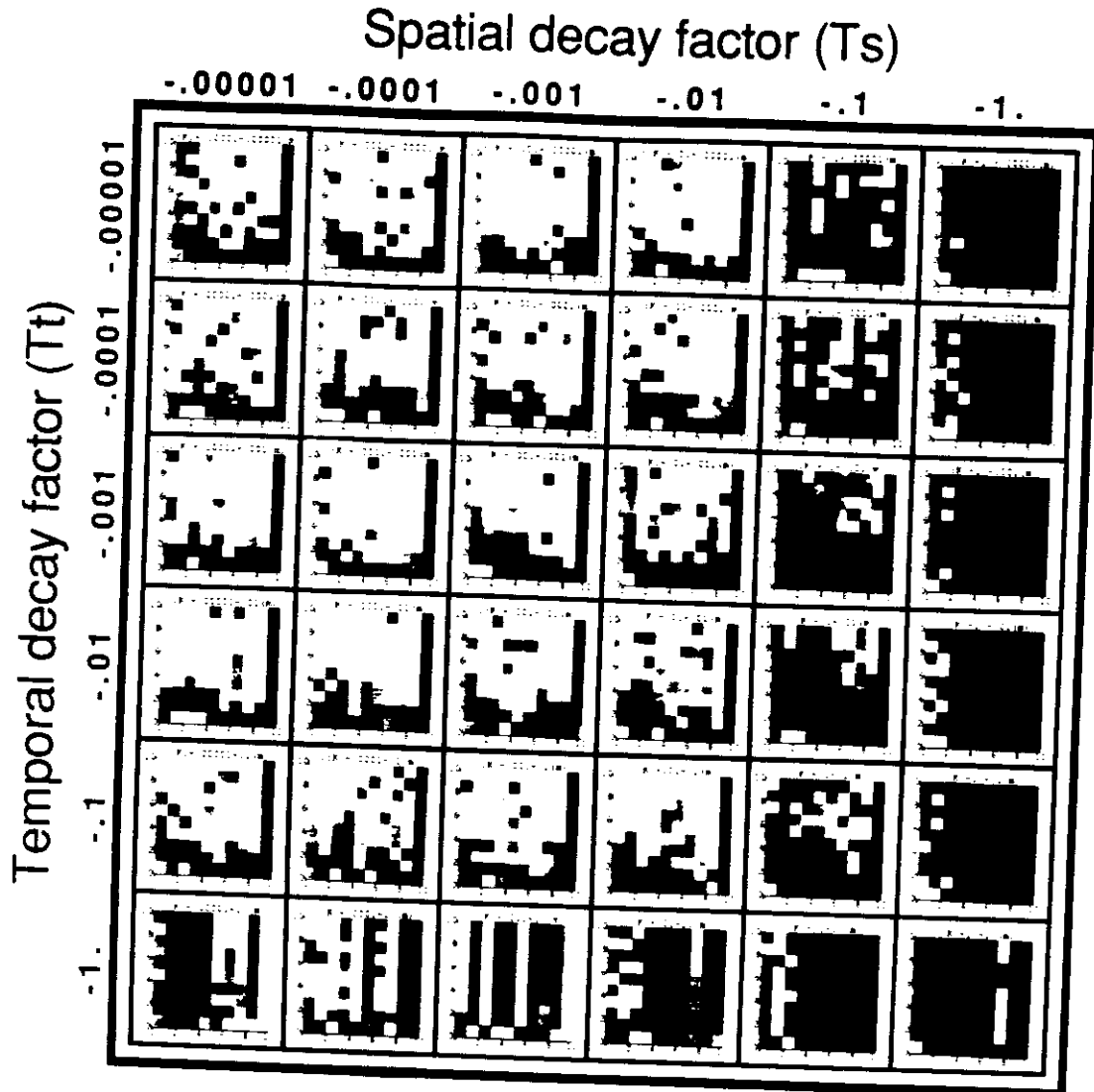


Figure 8.9a: KATAMIC performance as a function of T_s & T_t (match/goal)

The results (measurements of **match/goal**) of the 36 individual experiments are arranged in a 6*6 grid. In each experiment 10 different sequences were learned and the performance on only one of the 10 was monitored. The experiments are organized by increasing values of T_s & T_t with their smallest values (-.00001) in the upper left-hand corner. The x-axis for each of the experiments represents the pattern number within the monitored sequence (1 to 10), while the y-axis represents the number of repetitions of the monitored sequence. A gray scale encoding is used to represent the value of **match/goal** for each repetition and pattern (the small squares). In this encoding scheme, the bright end of the scale corresponds to good performance (correct recall) whereas the dark end indicates bad performance (poor recall).

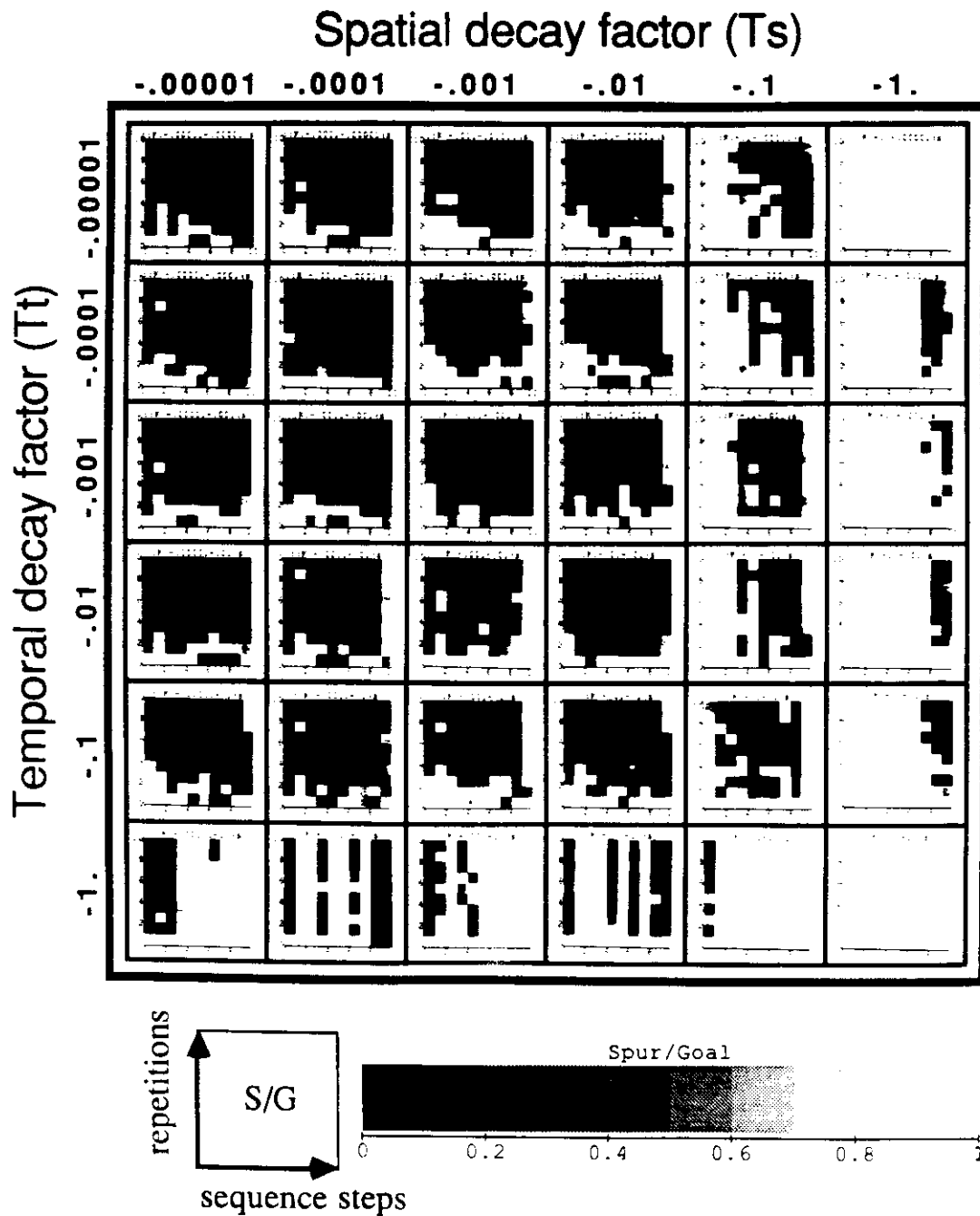


Figure 8.9b: KATAMIC performance as a function of T_s & T_t (spur/goal)

The layout of the results is the same as in Figure 8.9a. It is important to notice that the ratio **spurious/goal** can be bigger than 1. All such values are shown as white in the figure. In other words, the dark end of the scale means good performance (low **spurious**) while the white end means bad performance (high **spurious**).

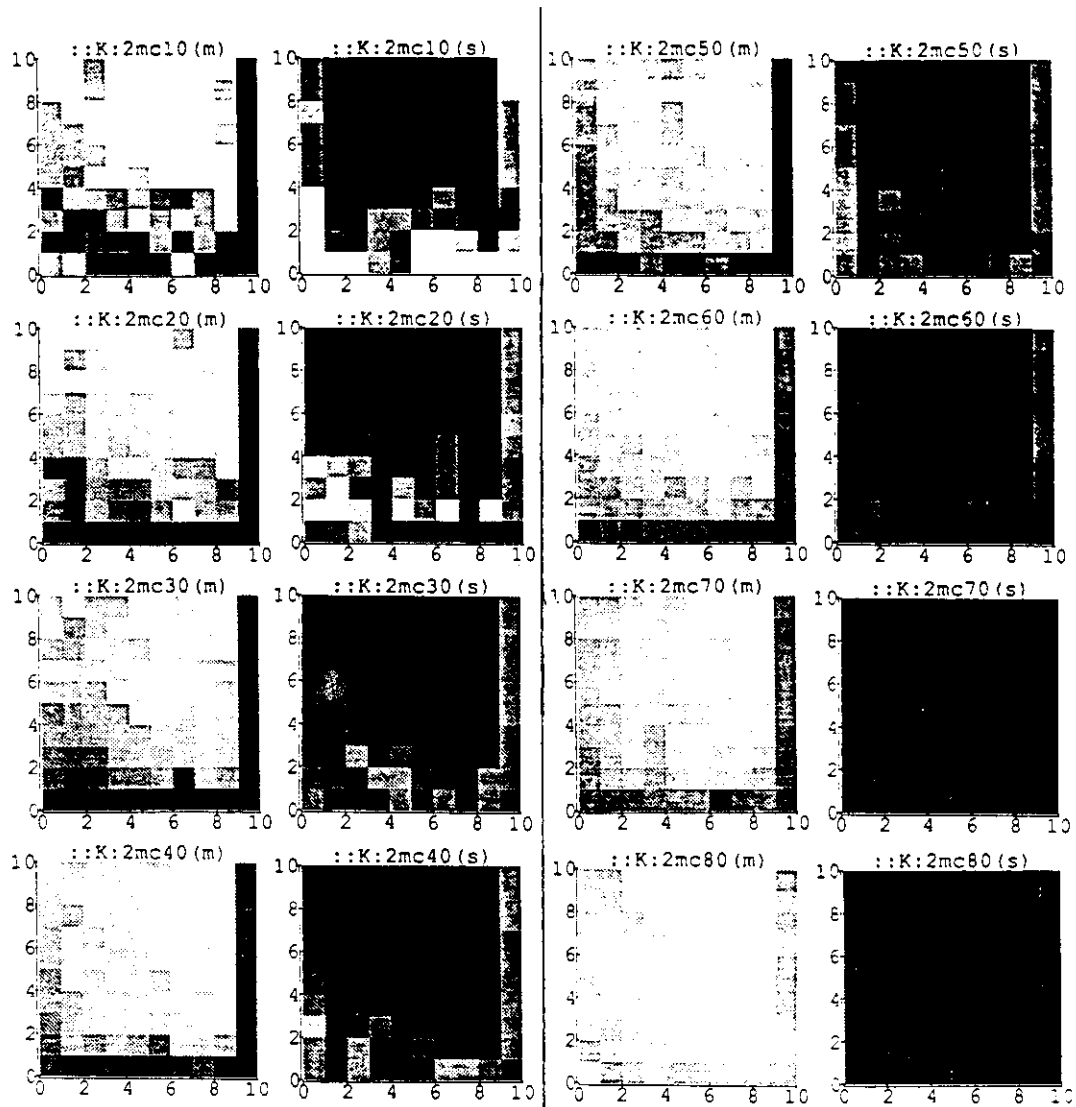


Figure 8.10: Various 1-bit-densities in different experiments

Each of the eight individual experiments are presented by two density plots, one for **match/goal** and the second for **spurious/goal**. Labels above the individual density plots (large squares) are interpreted as: the notations “:K:2mc” are irrelevant; the numbers “10 to 80” stand for the % 1-bit-density of the 64-bit-wide sequences learned in the individual experiment; “(m) or (s)” show what is being plotted (**match/goal** or **spurious/goal**). The interpretation of the gray scale code for the little squares within the large squares as well as the meanings of the X and Y axes are the same as in Figures 8.9ab.

(1) With a constant number of sequences stored (10), the smaller the 1-bit-density, the faster the learning. This is reflected in the fact that predictions improve (i.e. **match/goal** -> 1) while at the same time the **spurious** drops to 0. For sequences of density 10% and 20% it takes 4 to 8 repetitions for the predictions to become perfect (**match = goal**), while for sequences with more

than 40% density the quality of the predictions made improves with repetitions but perfect predictions are not achieved within the 10 repetitions.

(2) As it can be expected, the variance of the match is bigger for sequences with lower bit density.

The second experiment tested how performance is affected when the sequences learned within one and the same experiment have different densities. The model learned a set of 9 different sequences of 1-bit-densities 10%, 20%, ..., 90% (each pattern is 64 bits wide). This set of sequences was presented 10 times in a row. The results are shown in Figure 8.11. Several observations can be made here:

(1) Sequences of various 1-bit-densities can be stored in the network together (i.e. co-exist) and can be successfully recalled.

(2) Sequences with lower 1-bit-densities (10% to 40%) are learned faster and the quality of predictions made for these sequences undergoes a more significant change overall as compared to the sequences with higher 1-bit densities (50% to 90%). Also, as one might expect, for the sequences with lower densities (10% to 30%) the level of **spurious** during learning is very high because the possible **spurious** is very high. The results of the experiment suggest that if the model is used to store sequences in a broad range of 1-bit-densities, then it performs best for the sequences that have bit-densities in the middle of this range.

(3) Learning sequences of high 1-bit densities (> 50%) is not very "interesting" since the higher the density is, the less is the possible number of spurious and the less is the possible margin of difference between the sequences.

8.4.3 Effect of noise in the patterns

An important question is how noise affects the performance of the model. In other words, what happens when the network is presented with a noisy version of a previously learned sequence (e.g., with missing or added 1-bits in some or all of the individual patterns)? The expectation is that, within limits, the KATAMIC model will be able to tolerate the noise and the sequence of predictions it makes will be very similar to the learned sequence (the target sequence).

To systematically analyze the behavior under such conditions, the following experiment was designed. The KATAMIC model (configured as 128 predictrons with 256 DCPs per predictron) learned 10 different, randomly generated sequences (10% 1-bit-density for each sequence) of length 20 patterns. Using one of the learned sequences as a basis (target sequence), two sets of noisy sequences (5 sequences per set) were generated. The sequences in the first set were generated by reducing the number of 1-bits in the target. This was accomplished by turning them into 0-bits. This type of noise is called here "Deleted-noise". The resulting 5 sequence set had 10%, 20%, 30%, 40%, and 50% D-noise respectively. In other words, the total 1-bit-density of these sequences was 9%, 8%, 7%, 6%, and 5% respectively. The sequences in the second set were generated by increasing the number of 1-bits to be greater than in the original sequence. This was done by turning some of the original 0-bits to 1, i.e. "Added-noise". 10%, 20%, 30%, 40%, and 50% A-noise of the number of 1-bits in the target sequence were used. The total 1-bit-density of the resulting sequences was 11%, 12%, 13%, 14%, and 15% respectively.

Using these two sets of sequences, two experiments were performed -- one for each set. In each experiment, after the 10 original sequences were repeated 20 times and learned perfectly (see Figure 8.12 for the quality of learning of the target sequence), the learning was disabled and five

noisy sequences were presented in a row. Then each of the 5 predicted sequences (outputs) was compared with the target sequence in terms of match and spurious.

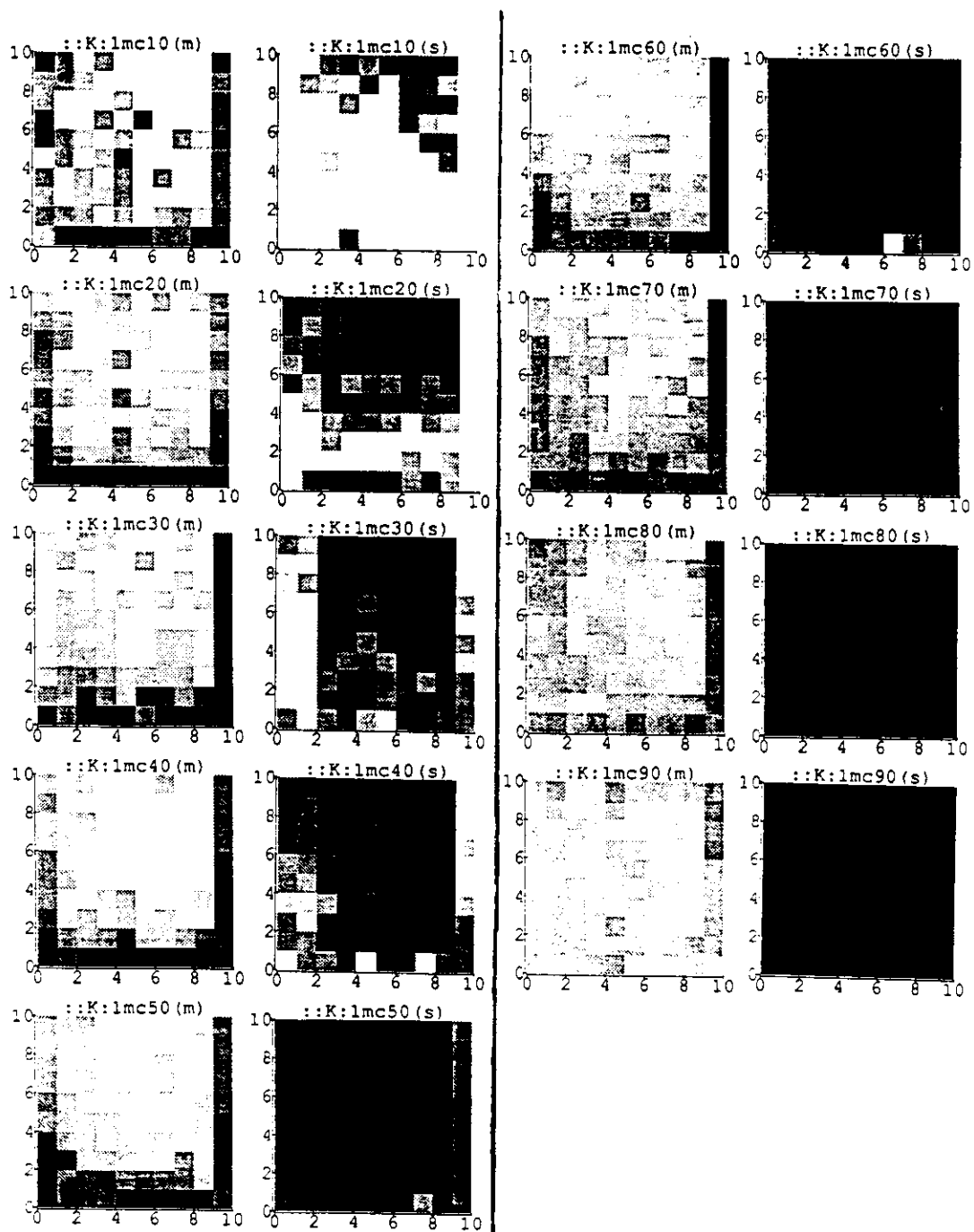


Figure 8.11: Various 1-bit-densities in a single experiment

The interpretation of the labels, the axes, and the gray-scale data encoding is the same as in Figure 8.10.

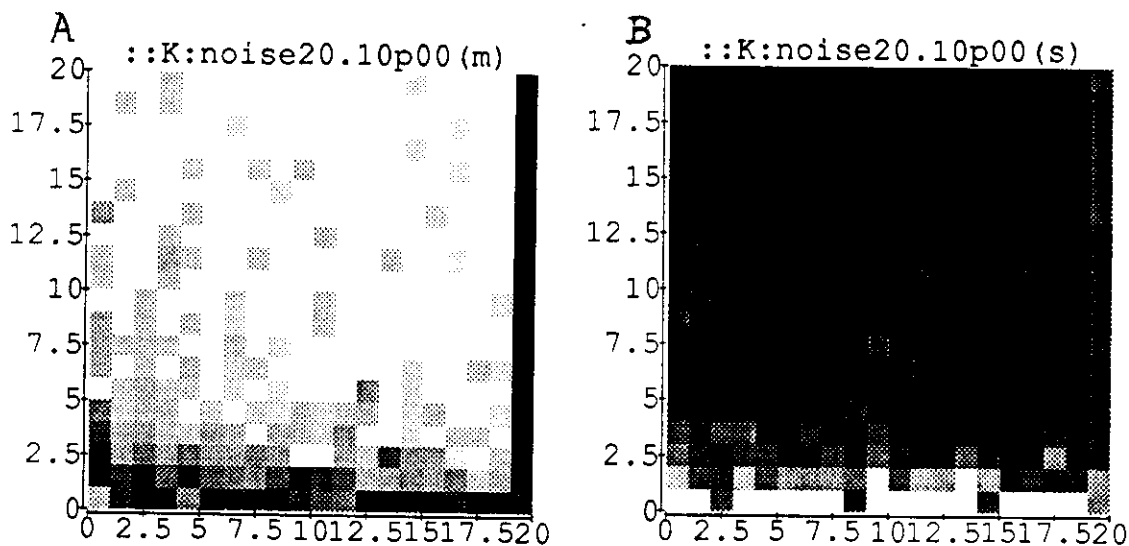


Figure 8.12: Noise tolerance -- learning target sequences

Density plots of **A) match/goal**, and **B) spurious/goal**. The pattern numbers (1 to 20) are shown on the X-axis, the repetitions (1 to 20) on the Y-axis. The gray-scale encoding of the measured values of **match/goal** and **spurious/goal** is the same as in all previous figures.

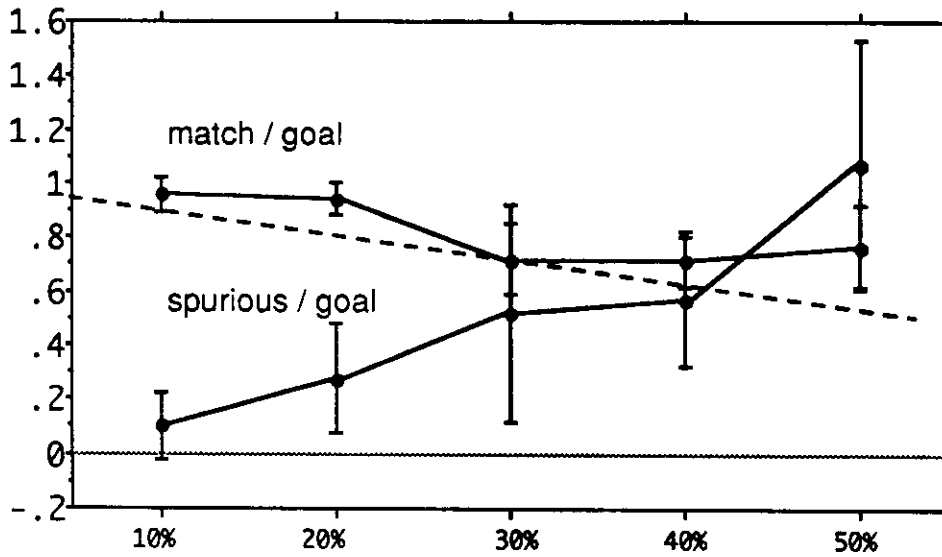
The results of the two experiments are presented in Figures 8.13a,b. As can be seen from graph (a), D-noise is tolerated well in terms of recognition of the noisy sequence (i.e. the predicted sequence matches the target sequence). The quality of match/goal gradually decreases when the percentage of D-noise is increased from 10% to 50%. At the same time the number of spurious bits generated increases. At a D-noise level of 30% the amount of spurious generated reaches the amount of correctly generated 1-bits -- the match (notice the overlapping error bars). If we take this D-noise level as a cut-off point, then we can say that within the paradigm of this experiment the model can tolerate about 30% D-noise. Another important thing to notice here is that within the same noise range the quality of the match is always better than its theoretical minimum (the case when the "noisy bits" are not corrected for). This theoretical minimum for the match is presented as a line going through $(x=10\%, y=0.9)$ and $(x=30\%, y=0.7)$

The effects of corresponding amounts of A-noise are significantly less severe than that of D-noise (Figure 8.13b) in terms of spurious generated. Overall, the results of these experiments show that the model can tolerate safely about 20% noise (of A or D type) in the patterns.

8.4.4 Learning branching sequences

An important question is how the network behaves if it has learned two or more sequences which have the same heads (the first few patterns) but different tails (the remaining patterns). To look at this issue, two sequences (S1 and S2) of length 10 (composed of 128 bits wide patterns) and 10% 1-bit-density were generated at random. The first three patterns of sequence S1 were copied over to the corresponding patterns (1,2,3) of S2 obtaining a new sequence S2'. As a result, S1 and S2' were the same from patterns 1 through 3 and different in patterns 4 through 10. Symbolically represented, the two sequences are: $S1 = \underline{ABC}DEFGHIJ$ $S2' = \underline{ABC}KLMNOPQ$. Sequences S1 and S2' were learned using two different learning protocols:

a) D-noise injected by "deleting" 1-bits



b) A-noise injected by "adding" 1-bits

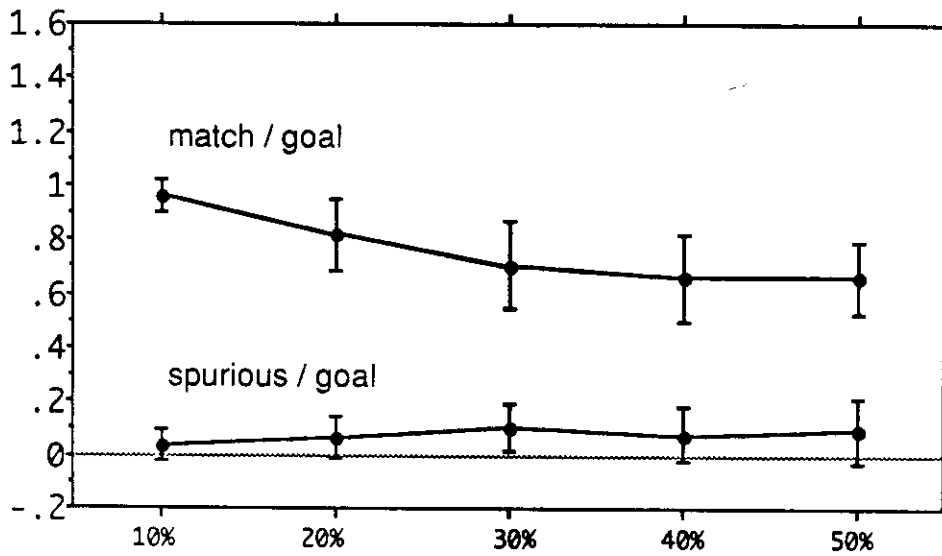


Figure 8.13: Noise tolerance -- processing of noisy sequences

Graph (a) shows the network's performance on sequences with D-noise. Graph (b) shows the performance when A-noise is added. Percentage of D- or A-noise are shown on the X-axis, the ratios **match/goal** and **spurious/goal** are plotted on the Y-axis. Error bars are used to show the magnitude of the Standard Deviation (SD).

(1) The sequence pairs S1 followed by S2' were repeated 100 times. Of interest here was the model's behavior for each sequence at and after the point of divergence (pattern #4 in each sequence).

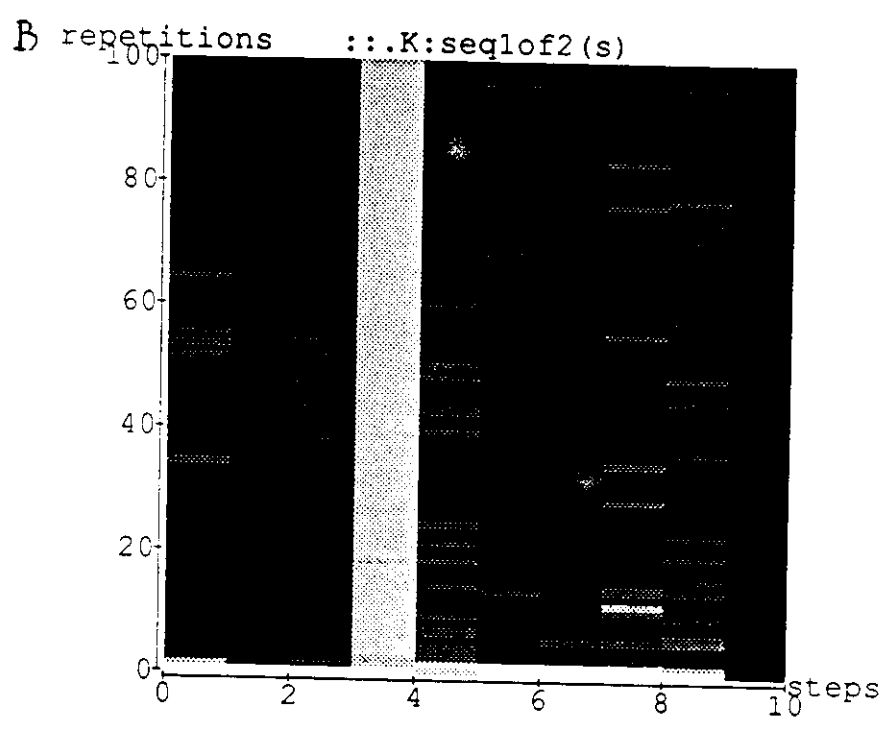
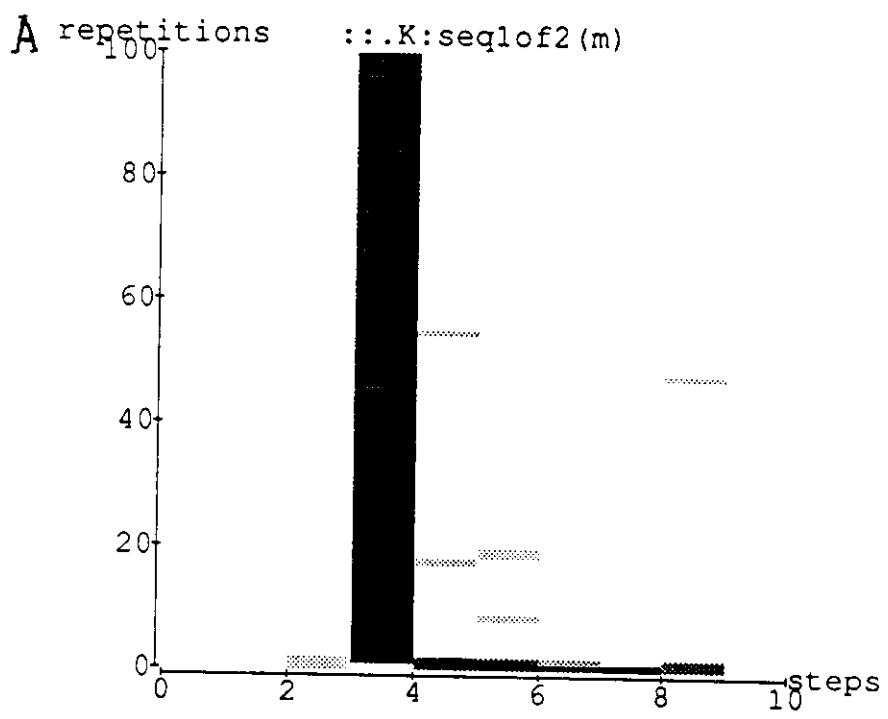


Figure 8.14: Learning branching sequences -- consecutive repetitions

Protocol 1: Performance results for only one of the two sequences (S1) are shown here. The results for the second sequence (S2') are similar. As in previous figures, the pattern numbers are plotted on the X-axis and the number of learning trials (i.e. repeated exposures of the network to the sequence) are shown on the Y-axis. Gray scale coding of the **A) match/goal** and **B) spurios/goal** are the same as in previous figures. The vertical columns at step 4 represent the values of the **match/goal** (plot A) and **spurios/goal** (plot B).

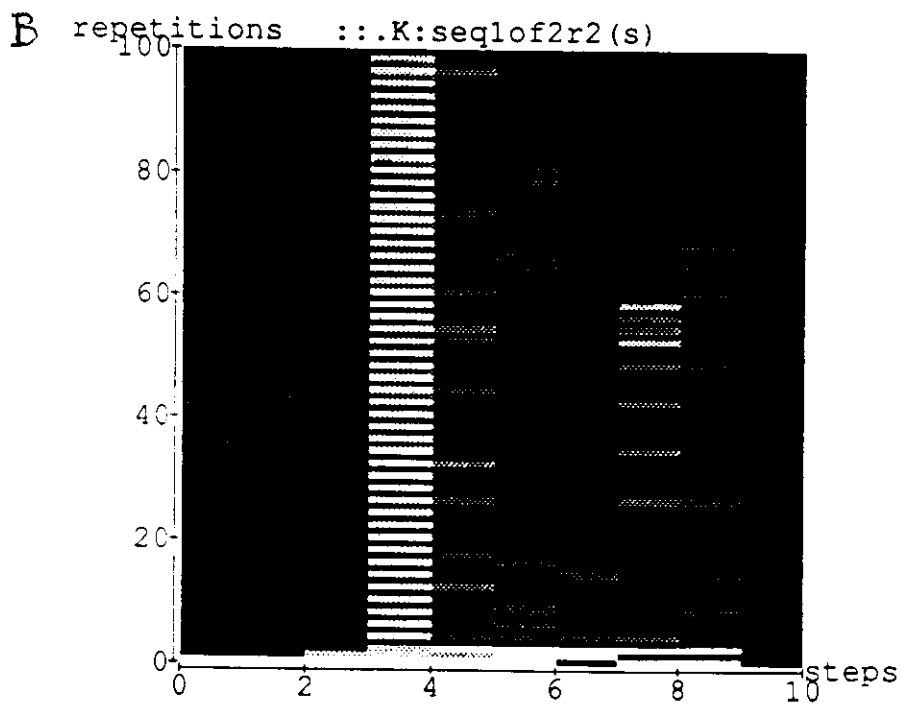
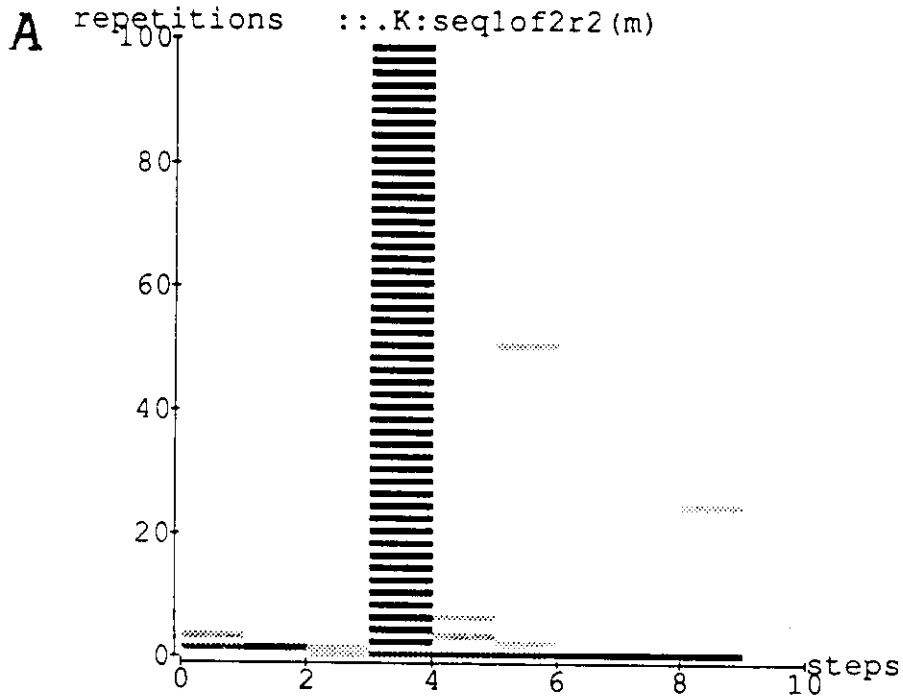


Figure 8.15: Learning branching sequences -- priming effects (100 repetitions)

Protocol 2: As in previous figures, the pattern numbers are plotted on the X-axis and the number of learning trials (i.e. repeated exposures of the network to the sequence) are shown on the Y-axis. Gray scale coding of the **A) match/goal** and **B) spurious/goal** are also the same as in previous figures.

(2) Sequence S1 was repeated 2 times in a row (a sequence-double) followed by 2 successive repetitions of sequence S2', followed by 2 repetitions of S1, and so on until 50 repetitions for each sequence-double were performed. The objective of this experiment was to test for "priming" effects. In other words, does the network's response differ between the first and the second exposure to the same sequence? It was expected that, while at the first exposure, the network will generate "noise" at step 4 (i.e. a mixture of predictions belonging to both sequences), at the second exposure it will predict the most recently seen sequence.

The results of the first experiment are presented as two density plots (**match/goal** and **spurious/goal**) in Figure 8.14. Learning of the two sequences took only few repetitions. As expected, the learning with protocol 1 resulted in noise at step 4. However, after step 4 (where the confusion occurs) the network unmistakably continued to generate correct predictions for the currently processed sequence (e.g., S1 or S2').

The performance under protocol 2 (Figure 8.15) confirmed our expectations. Namely, during the second repetition of each sequence (in a row) the model learned (in about 8 repetitions) to correctly predict step 4 and the spurious were minimal.

An unexpected network behavior was observed if protocol 2 was let to run longer (1000 repetitions). After about 200 repetitions the network managed to learn that the second repetition of each sequence (e.g., S1) is followed by the other sequence (e.g., S2') (Figure 8.16). Effectively the network exhibited a higher order learning. It learned not only the order of patterns in the sequences, but also the order of the sequences itself. This is an interesting effect.

At step 4 there were always some spurious bits observed. However, after about 400 repetitions they reached a steady state. At a closer examination, for each of the sequences, it was found that the spurious bits correspond to the active bits at the same step (4) in the alternate sequence. Effectively the prediction which the network learned to produce at this step represented an OR of the patterns at step 4 of sequences S1 & S2'.

8.4.5 Memory capacity

A variety of measures of memory capacity have been used in neural network research. One common measure is the number of traces that are stable under the recall operation. This is presumed to set an upper bound for the possible memory storage. However, with a set of random memory patterns to be learned, even a few stored traces might generate spurious associations and thus not be stable. I define the memory capacity of the network as the number of sequences that can be learned reasonably well without excessive spurious recall. In this sense, capacity is a function not only of the network size but also of the length of the sequences and 1-bit-densities, the length of the cues, and the relative importance of complete versus correct recall.

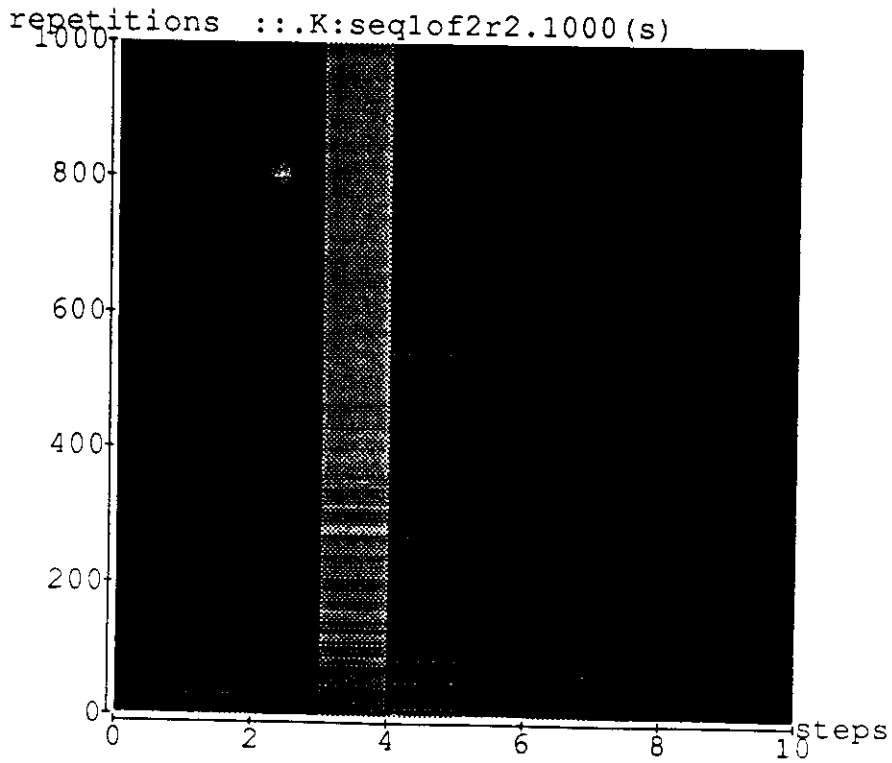
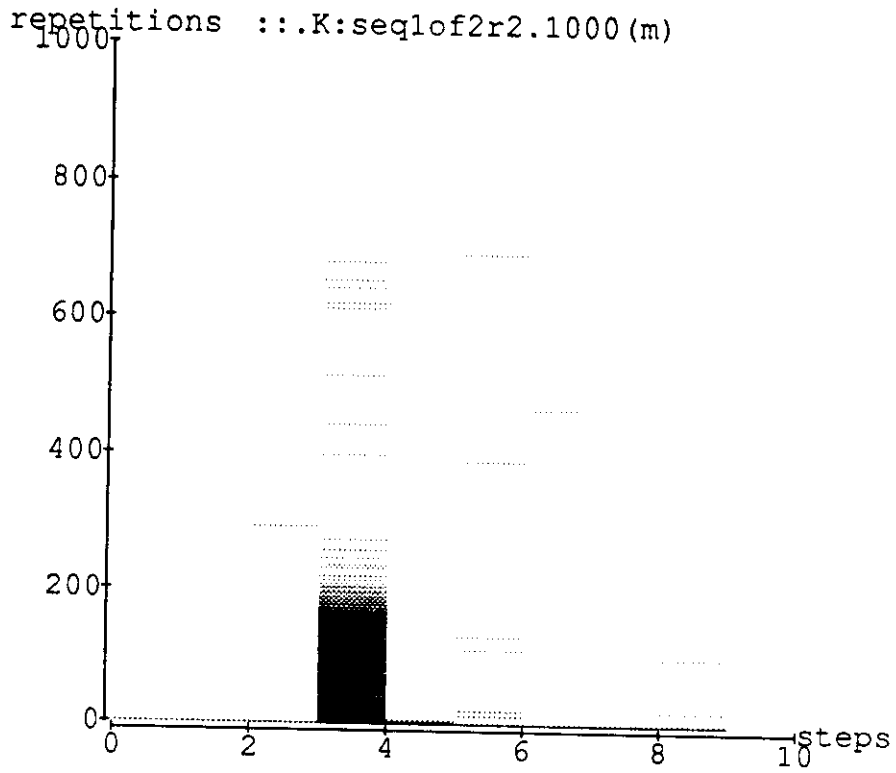


Figure 8.16: Learning branching sequences -- priming effects (1,000 repetitions)

The pattern numbers (10) are plotted on the X-axis and the number of learning trials (1000) are shown on the Y-axis. Gray scale coding of the **A) match/goal** and **B) spurious/goal** are the same as in previous figures. Notice that after about 200 repetitions the network learned the order of the sequences S1 and S2' as can be seen in the column representing **match/goal** at step 4 (plot a).

As in the previous sections the method for evaluation of the memory capacity is based on simulations. Two separate sets of experiments were ran. In the first set (A) the dependence of the maximal length of a single memorized sequence on the number of DCPs per predictron was measured. The second set (B) examined how many short sequences (short with respect to the maximal sequence length obtained from the previous experiment) can be learned and recalled without much spurious recall.

(A) Maximal length of a single learned sequence

One estimate of the memory capacity of the model can be obtained by observing experimentally what is the maximal length of a single memorized sequence and how this length depends on the number of dendritic compartments per predictron. To obtain this estimate, a set of 10 experiments was performed. The network configured with 64 predictrons and 256 DCPs per predictron. In each experiment a single sequence of 15% 1-bit-density was presented 40 times in a row. In the first experiment the length of the sequence (i.e. the number of patterns it contained) was 10. This length was systematically increased to 20, 30, ..., 100 patterns in the 2nd to 10th experiment. The usual performance characteristics were measured at each B-cycle (**match/goal** and **spurious/goal**). The average value of the goal for this experiment was (15% of 64 = 9.6). By definition the ratio match/goal at each B-cycle is in the range 0 to 1. The maximal value of the spurious for these experiments is $(64 - 9.6) / 9.6 = 5.67$. For a given B-cycle, this maximum corresponds to the case when all "non-goal" predictrons fire.

The results of the experiments are summarized in Figures 8.17. As can be seen from Figure 8.17A, the quality of the predictions made improves rapidly during the first 5 to 10 repetitions and for all sequences after about 20 repetitions it becomes better than 95%. The speed of learning is somewhat non-monotonic for the longer sequences, e.g., the sequence with length 100 is learned faster than that of length 90. This can be explained with the fact that the sequences were randomly generated and have a limited length and width. These results are very satisfactory but they are not meaningful unless one takes into account the number of spurious generated in each experiment. As can be seen from Figure 8.17B, the number of spurious behaves very nicely, i.e. it drops rapidly to 0 for all sequences with lengths less than 60 patterns. For sequences of lengths more than 60 patterns the ratio of **spurious/goal** stays above the level of 1 throughout all repetitions (i.e. the number of spurious bits is larger than the number of 1 bits in the target pattern -- the goal). From observing both the matches and the spurious one can conclude that the maximal length of a single sequence (15% density) learned by the KATAMIC model (with the given network configuration -- 64 predictrons with 256 DCPs/predictron) is 60 patterns. In other words, the maximal length of the sequence is about 23 % of the number of DCPs per predictron. A sequence of length 60 patterns and width 64 bits contains $60 \cdot 64 = 3,840$ bits which are spatially (within patterns) and temporally (between patterns) ordered. If we assume that each DCP represents one memory location, then we have $256 \cdot 64 = 16,384$ memory locations. Therefore the ratio between the information stored in a sequence in terms of bits and the total number of memory locations in the model is $3,840 / 16,384 =$

23.4%. In other words, using the measure described above, the memory capacity of the KATAMIC model is about 23% of the total number of storage locations in the network.

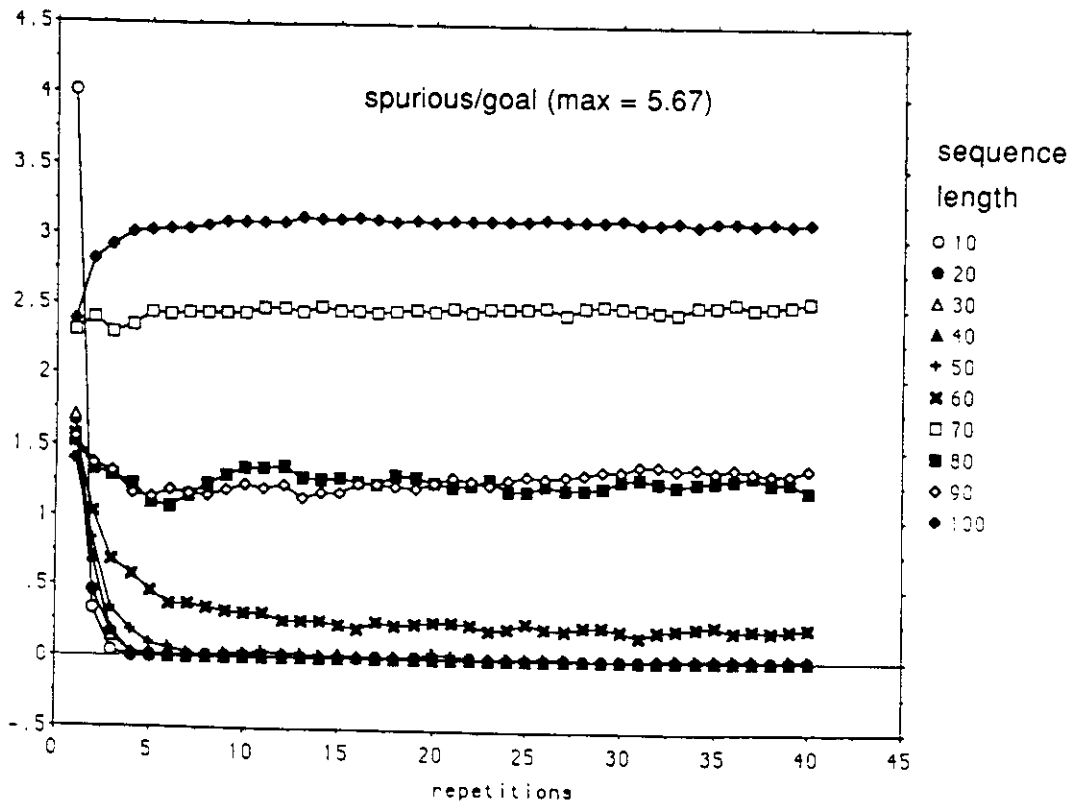
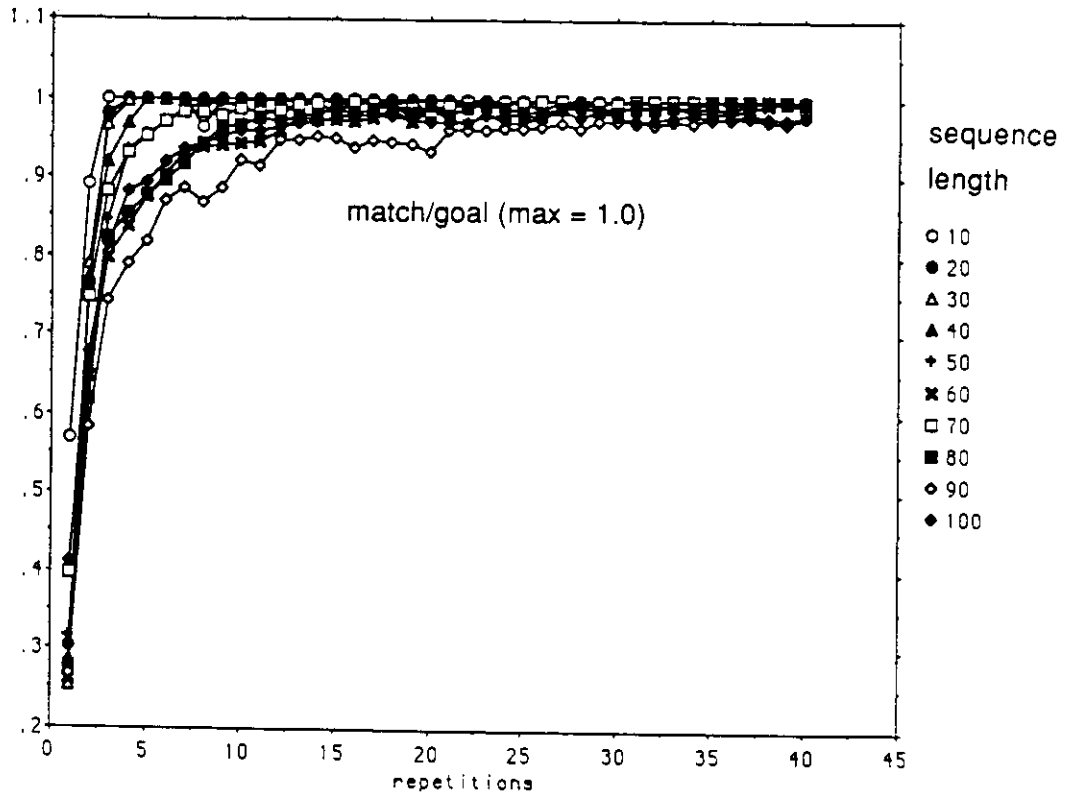


Figure 8.17: Memory capacity as a function of sequence length

Each of the 10 curves in the two plots represents a separate experiment. In successive experiments the sequence lengths were varied from 10 to 100 patterns labeled in the legend to the right. The repetitions of each sequence are shown on the X-axis while the measured ratios **A) match/goal** and **B) spurious/goal** are plotted on the Y-axis.

(B) Memory capacity for multiple short sequences

An alternative way of probing memory capacity is to run a set of experiments in which one gradually increases the total number of sequences to be stored is gradually increased. Of interest here is when the quality of the predictions starts to deteriorate significantly. Three different experiments were run. To probe the space of behaviors of the network, three experiments were run. The patterns used in these experiments have 10 % 1-bit density.

In the first experiment 5 sequences were learned. After about 4 presentations each of these sequences was learned almost perfectly (> 99%) and the level of spurious was minimal (<2%).

In the second experiment the network attempted to learn 10 sequences. The behavior was similar, i.e. each sequence was learned after about 5 repetitions. However, the overall quality of the predictions has decreased to about 97% and also the number of spurious has also increased to about 3%, especially within the first 3-4 patterns of each sequence. This is due to the fact that, with the chosen bit-density of the sequences (10% or about 6.4 bits per pattern set to 1), there is some overlap between the corresponding patterns in the 10 sequences.

In the third experiment the network tried to learn 20 different patterns. While the behavior with respect to the predictions was similar to the previous two experiments, the behavior with respect to spurious was poor. The number of spurious varied significantly and the average value of the spur/goal ratio was 0.6. The maximal value of this ratio for each B-cycle in this experiment is $(64 - 6.4) / 6.4 = 9$.

From these three experiments it is obvious that a KATAMIC model with dimensions 64×256 can safely learn about 10 sequences of length 10 patterns each. If we consider that the total bit content in these 10 sequences is $64 \times 10 \times 10 = 6,400$, then the estimate of the memory capacity using this second method is $(64 \times 10 \times 10) / (64 \times 256) = 39.0\%$. This number, which is higher than our previous (A) estimate, can be explained by the fact that the shorter the sequences, the smaller the number of transitions between patterns.

The estimates of the memory capacity of the KATAMIC model are only preliminary. They are significantly higher than estimates of memory capacity obtained in other models. For instance, the memory capacity of the Hopfield network for static patterns is about 14% of the number of storage locations. For Kanerva's Sparse Distributed Memory (SDM) this number is about 10% (Keeler, 1988).

The capacity of a generic associative memory and the SDM in particular (Kanerva, 1984; Kanerva, 1988) has been estimated analytically (Chou, 1988). Defining the capacity as the maximum number of words (patterns) that can be stored and retrieved reliably by an address within a given sphere of attraction (i.e. the area where the memory can correct delta factor errors), Chou demonstrated that the capacity of any associative memory is limited to an exponential growth rate of $1 - h_2(\delta)$ where $h_2(\delta)$ is the binary entropy function in bits, and δ is the radius of the

sphere of attraction. He also demonstrated that by choosing an optimal set of network parameters the SDM can actually achieve this exponential growth. As Chou pointed out, the exponential growth in capacity for the SDM is in sharp contrast with the sub-linear growth in capacity for the Hopfield associative memory (McEliece et al., 1988).

The growth of memory capacity of the KATAMIC model has not been estimated analytically as yet. However, as will be discussed in section 12.9.1, the KATAMIC architecture and dynamics can, to a first approximation, be mapped to the SDM model. Therefore, I suggest that the KATAMIC model will also have an exponential growth in memory capacity.

8.5 Summary of KATAMIC characteristics

The potential utility of the KATAMIC model is a result of its statistical properties, which lead to the following important and interdependent functional characteristics:

(1) Rapid learning: Few exposures (on average 4 to 6) of the network to a particular sequence are sufficient for learning (>90% correct). The speed of learning depends: (a) non-significantly on the length of the sequence. Learning of longer sequences is less accurate during the first 10 repetitions, (b) significantly on the 1-bit density of the sequences, (c) on the number of already learned sequences. This rapid learning is a major improvement over the simple error back-propagation recurrent networks (Jordan, 1986; Elman, 1988) which require hundreds/thousands of epochs to achieve reliable performance.

(2) Memory Capacity: Multiple sequences can be stored in the model. This is a significant improvement over oscillators/pacemaker based models (Torras, 1986; Miall, 1989). The memory capacity is comparable if not better than that of other models (e.g., Hopfield, Kanerva).

(3) Sequence completion: A short cue, which is sufficient to discriminate a particular previously stored sequence, can retrieve the complete sequence. This is an improvement over current models which allow the retrieval of only very few elements at the end of the sequence and then only after almost the whole sequence is presented as a cue.

(4) Sequence recognition: A built-in recognition mechanism allows flexible sequence recognition on a pattern-by-pattern basis. This mechanism is used internally for switching from learning to performance mode.

(5) Fault and noise tolerance: Missing elements (bits within patterns) within a reasonable range (up to 30% of the number of 1-bits) can be tolerated (i.e. substituted during recall). The memory can interpolate and extrapolate from existing data and is fault tolerant. With regard to noise and fault tolerance, the KATAMIC memory is comparable to other state-of-the-art models.

(6) Robustness of performance: The model operates within a wide range of values for the memory parameters. For instance, for the spatial and temporal decay constants (T_s , T_t) this range spans over several orders of magnitude (10^{-5} to 10^{-2}).

(7) Straightforward scalability: (a) Adding more predictiontrons, using the same inter-connection scheme, allows processing of correspondingly wider sequences (i.e. longer patterns). (b) Increasing the number of dendritic compartments per predictiontron allows storage of sequences with longer lengths. This feature of the KATAMIC memory is an improvement over the simple recurrent networks (SRN) (Elman, 1988) where there is not an obvious solution of how much to increase the number of hidden units and how to change the connectivity pattern if it should be less than fully

connected -- a structural constraint which causes significant technical problems for hardware implementation. In contrast, the KATAMIC memory can be made very large, and large amounts of information can be stored in it.

(8) Integrated processing: The model is capable of concurrent learning, recognition and recall of sequences. This is a significant improvement over the majority of previously proposed models which focus only on specific aspects of processing. Such models often must keep learning and performance stages separate.

9 TAXONOMY OF MEMORY IN DETE

There are two major types of memory mechanisms in DETE which are necessary for the development of its language and reasoning abilities (Figure 9.1). These are: (1) Short-term memory (STM) (a.k.a. immediate or primary memory), and (2) Long-term memory (LTM). This functional classification of DETE's memory corresponds to the memory classification scheme commonly used in cognitive psychology (Squire, 1987). An important difference, however, is that while the psychological classification is not clear on whether STM and LTM have different brain mechanisms, in DETE, there is a clear physical difference between them. More specifically, the neurons (predictrons) forming the STM and LTM are different from each other and interleaved. LTM is of two types: (a) Declarative memory (DM), and (b) Procedural memory (PM). While both the declarative and the procedural memories are intrinsically sequential, there is a fundamental difference between them. The Declarative memory stores sequences as a whole. For instance, it stores words as complete entities rather than as sequences of gra-phonemes. On the other hand, the procedural memory stores information about the order of segments within a sequence. For instance, the order of the gra-phonemes in a word or the order of words in a sentence. The DM is further divided into: Episodic Memory (EM), and Semantic Memory (SM). The PM is also subdivided into: Morphologic/Syntactic Procedural Memory (MSPM), and Motor Memory (MM). The MSPM is in turn subdivided into Verbal Memory Bank (VMB), Transition Detector Bank (TDB) and Order Memory Bank (OMB). The characteristics of each of these memories, their function and implementation in DETE are described below.

9.1 Short-Term Memory

The Short-Term memory (STM) is the basis of DETE's ability to repeat short sequences of items (e.g., several consecutive words or a short sequence of visual frames) immediately after presentation. It basically serves as a limited capacity buffer. DETE has separate short-term memories for the visual and the verbal modalities (Figure 9.1). Both, the visual and verbal STMs have been designed so that they exhibit the following functional characteristics:

(1) *One shot storage (memorization)*. A word or a sentence presented only once can be repeated right away. A simple visual sequence (e.g., of an attended object) can also be mentally recalled right away. As will be seen in the next section, this feature of the STM is a result of specific changes introduced in the dynamics of the KATAMIC model which serves as basis for the STM (and which normally needs 4 to 6 stimulus repetitions to learn).

(2) *Fast and complete reset*. When a new input (sentence or image) is presented (i.e. when the focus of attention is shifted), the STM content is reset (emptied) in response to a signal coming along the Climbing Fibers (CF) (Figure 8.2). For instance, the memory trace of the verbal input "The ball is left of the triangle" resides in the STM until the next verbal input is presented, e.g., "The triangle is moving up". The new input leaves a trace in the STM that overrides the old trace if the activation that produces the new trace is physically in the same area of the STM where the old trace resides. If the new trace is in a different part of the STM, then the old and the new traces can temporarily coexist. This feature allows DETE potentially to handle ellisions.

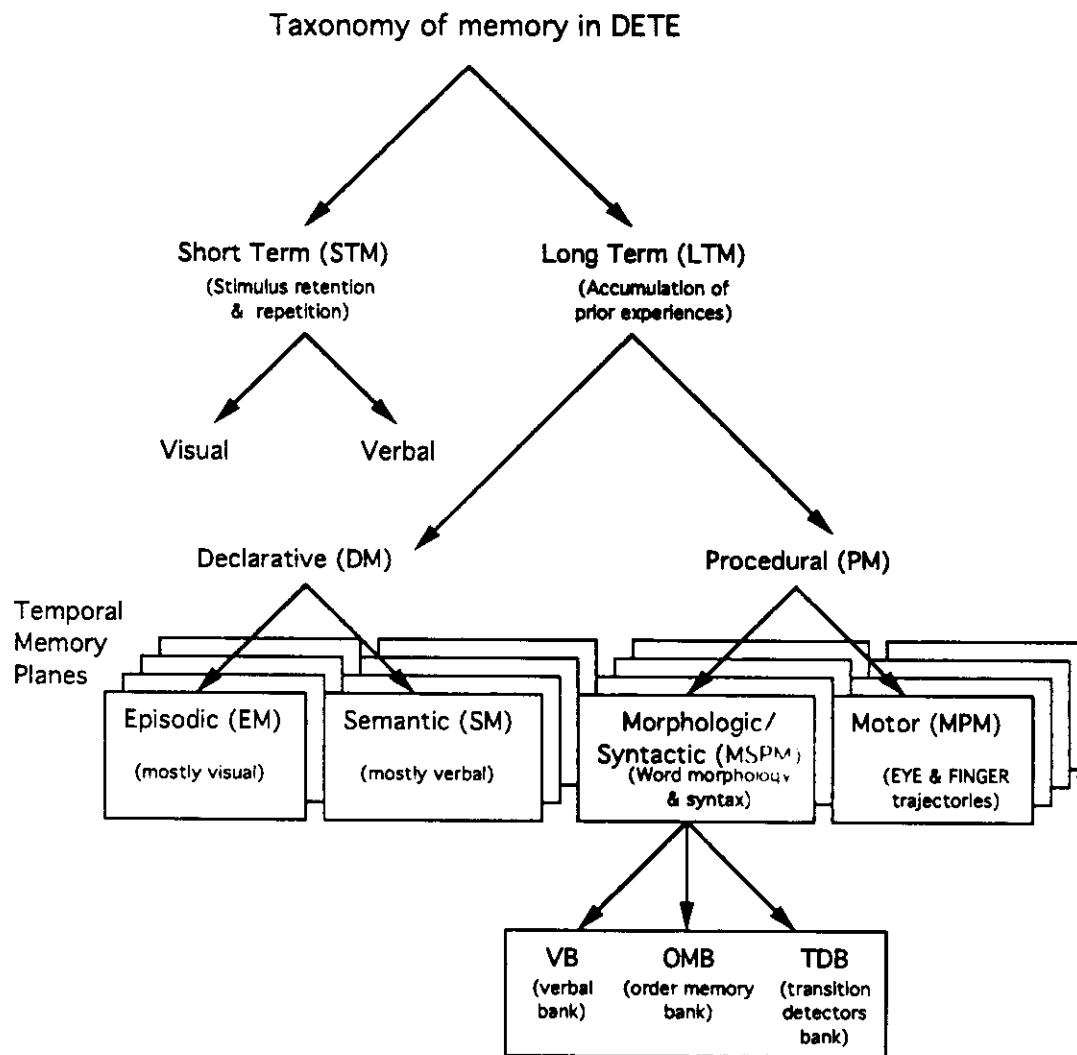


Figure 9.1: Taxonomy of DE TE's memory

Hierarchical organization of the memory types in DE TE. The leaves of this hierarchical structure correspond to individual, physically distinct memory mechanisms implemented in DE TE.

(3) *Recall triggered by a non-specific signal.* The content of the STM can be recalled (rehearsed) without the necessity of a *specific* external cue to trigger each iteration. An example of a specific cue for the trace left by the verbal input “The ball is left of the triangle” is for instance, “The ball is”. A *non-specific* signal (e.g., an external verbal request for repetition like “What?” or “Repeat” or “Say again”) can trigger the recall of the memory trace. In general, repetition (rehearsal) of the STM content is done by DE TE continuously. In other words, a sequence residing in the STM at a given *moment* is repeated again in the next *moment* if there is no external stimulus to interfere with this process. Behaviorally speaking, DE TE has an “intent” to repeat whatever it has heard or seen a *moment* ago. The trigger for this rehearsal is implemented as a non-specific cue to

the memory (see next section) generated at the end of each *moment*. Once the rehearsal is interrupted by another external stimulus the content is lost.

(4) *Limited capacity*. The recall performance deteriorates gracefully if the capacity is exceeded.

(5) *Fast decay*. As a result of the “drive” for internal repetition, the STM is refreshed at each moment. To match the characteristics of the human STM, DETE’s STM is set up so that if not refreshed through rehearsal its content is rapidly lost.

9.1.1 STM implementation

Both the verbal and the visual STMs are implemented as modifications of the basic KATAMIC memory model (see Chapter 8 for discussion of the KATAMIC model). To meet the above-mentioned specifications of the STM, the following three changes in the KATAMIC model were made:

(1) Modulation of the *stm* injection rate. In the KATAMIC model, the total amount of *stm* (not to be confused with STM in this section) in the dendritic branch of each predictron increases with the progress of the sequence. This is due to two factors; (1) more *stm* is injected every time cycle, and (2) the ratio of the *stm* injected into the seeds of the DCPs to the decaying *stm* in the rest of the DCPs is in favour of the injected. As a consequence, patterns which come later in the sequence leave stronger *ltm* traces, since *stm* accumulates in the dendritic tree with the progress of the sequence and the *stm* is an argument to the *ltm*-update function. Later, when a recall of a stored sequence is attempted, the stronger traces (i.e. those of patterns towards the end of the sequence) tend to express themselves easier. In other words, they have more weight when the dot-product is calculated and therefore they influence the predictions more strongly than do the older traces. In order to ensure that the earlier patterns leave stronger *ltm* traces, a mechanism that decreases the amount of injected *stm* during successive patterns was designed. This mechanism involves a modulation (exponential decay with time constant T_b) of the *stm*-injection-rate \mathbf{b} with time (for definition of \mathbf{b} see equation 3 in section 8.2).

$$\mathbf{b}'(t) = \mathbf{b}(t) e^{-T_b t} \quad (9.1)$$

(2) Reset of *ltm*. Another change introduced to the KATAMIC model is a reset of the *ltm* to its original value (see Table 8.2) at the beginning of every new input sequence. The *ltm* is always reset when a new sequence is input. However, if there is no input sequence, a “request” for repetition of the previous sequence is generated in the form of a non-specific cue. In this case the *ltm* is maintained unchanged to allow the sequence recall.

(3) Recall via a non-specific cue. A change in the way sequences are recalled is introduced in the STM (as compared to the recall in the KATAMIC memory). Instead of using a specific cue (i.e. the first few steps of a learned sequence), in the STM a non-specific “shock-input” is used for retrieval of a stored sequence. A “shock-input” is a pattern or a short sequence of patterns in which all bits are set to 1.

The STM mechanism described above was tested in simulations which demonstrated that a shock-input is capable of retrieving a complete trace (e.g., a whole sentence like “The red ball is in the center”) stored in the STM. The STM meets the design specifications because the non-specific cue triggers the recall of the learned sentence. After the initial few steps during which all input bits to the STM are set to 1, the STM starts to generate the correct sequence of predictions and uses these predictions to complete the recall. The switch from the external input (i.e. the non-specific

cue) to internal input (i.e. the output of the predictrons) is done by the recognitrons/BBSs after the end of the non-specific external cue.

9.2 Long-Term Memory (LTM)

The long-term memory in DETE is a type of memory the basic characteristic of which is a larger time span than the STM. In other words, the LTM is not reset any time a new sequence is input, but rather it stores information about multiple sequences. There are two kinds of LTM -- declarative and procedural, which reside in physically separate memory modules.

9.2.1 Declarative Memory (DM)

The declarative memory is memory for facts. There are two types of declarative memory, episodic memory (EM) and semantic memory (SM). In DETE, these two categories are subserved by one memory mechanism -- the basic KATAMIC memory. SM and EM can be regarded as the two poles of a continuum. This continuum is formed during DETE's training. Initially, while DETE is still naive, all experiences are stored as episodes. Later the ones that are repeated over and over again form the SM and the ones that are unique form the EM. The strength of the memory trace (encoded in the *ltm*) left by an input sequence depends on the magnitude of the *ltm* update rate -- a constant (**b** -- see table 8.2). At small values of **b**, (< 1) an input sequence leaves a weak *ltm* trace, whereas at larger values of **b** the trace which is left is stronger. The constant **b** can be regarded as a measure of the "emotional/alertness" state of DETE. Higher states of alertness (i.e. larger **b**) leave stronger traces in the DM. In DETE, the magnitude of **b** is controlled externally by the user. In other words, DETE currently does not contain an endogenous (internal) mechanism to control its "emotional/alertness" state.

The general characteristics of the DM are: (1) It is composed of the traces of the sequences of events that have been experienced, i.e. sequences of visual scenes or words in sentences. (2) The storage is sequential, i.e. new traces are added with new experiences. (3) Memories can be retrieved, i.e. "brought to mind" or instantiated through all modalities (e.g., verbally in the form of hidden articulation or non-verbally in the form of imagination). (4) A cue, by virtue of its content, can start a retrieval process at any point of a stored sequence and if left to itself the memory will complete the sequence. (5) The DM undergoes consolidation -- repeated input stimuli reinforce the existing *ltm* traces in the DM, and forgetting -- if particular trace is not reinforced through repetitions, it is gradually overwritten by other traces and can ultimately be lost.

Episodic Memory (EM)

The EM in DETE stores and recalls (re-experiences) specific episodes (events). All unique experiences (i.e. such that do not recur during its "life span") are treated as episodes while all recurring experiences form the semantic memory. To retrieve an episode (i.e. bringing it to WM) DETE requires specific cues which it further elaborates by using the outputs of the memory as inputs in consecutive B-cycles.

Since the EM is a part of the DM, it possesses all of DM's characteristics and also some specific ones such as: (1) Its content is formed by unusual or unique events. (2) Memories are stored one-shot as a result of unique experiences. For the one-shot storage of unique traces in the EM, DETE uses large values of the *ltm* update rate **b**. (3) Retrieval requires time-consuming elaboration of cues (i.e. several B-cycles are needed to recall a complete memory). (4) There is little cross-talk between

traces since they represent unique events and generally different events have different representations.

Semantic Memory (SM)

The SM extracts commonalities in and stores traces of multiple similar episodes or repetitions of one and the same episode. For instance, a ball always bounces when it hits a wall. Since the things that change in all cases of repetition are the times and locations where the events occur, repeated events are stored only as traces in which the temporal and location information are effectively lost (smeared in time).

Like the EM, the SM has all general characteristics of the DM and also some specific ones such as: (1) Content is formed by familiar items which have been experienced over and over again during the life-time (i.e. they have formed categories). For instance, words, recurring events, etc. Effectively it holds the invariant features of the perceptually experienced objects or events. (2) Memory traces are formed through multiple repetitions. Each individual repetition makes the particular trace stronger. Ultimately the traces left are permanent (i.e. they cannot be overridden by other traces) and strong (i.e. they need only a short cue with little or no elaboration to be retrieved). (3) Retrieval is done when a partial cue is presented (i.e. response-sequence retrieval depends on the cue and the memory content). Usually the context that is recalled together with the trace is either (a) the most recent context in which this concept was seen, or (b) the most frequent context (combination of contexts), or (c) an externally provided context.

DETE's SM is roughly equal to the verbal memory module -- the Lexicon. In other words, the traces in the SM are left by words which form "symbolic tokens". As will be seen further in this chapter, the Verbal Memory Bank together with the Order Memory form a functional unit which as a whole serves as the Lexicon. The Lexicon is commonly viewed as a repository of both grammatical and commonsense knowledge indexed by lexical items (Nakhimovsky, 1988). In DETE the lexicon has the same purpose. Namely, the grammatical (syntactic) knowledge is acquired with experience and involves knowledge about word order, e.g., the fact that (in English) modifiers (adjectives) are placed before the objects (nouns) which they modify, e.g., "red ball" or "large square". This knowledge is extracted from the verbal input and is encoded as the statistically most probable associations. The commonsense knowledge aspect of the lexicon is reflected in the association of the network's representation of the verbal tokens with the network representation of the visual reality to which the verbal tokens refer. It is "commonsense" in the sense that the representation of the visual world reflects the constraints which exist in the physical world. The network representation of each word itself in the verbal bank of the memory forms the lexical item index.

In DETE the lexical items are formed by memory traces of the individual words together with traces of the visual reality with which they were associated with. The memory trace of each word is associated with information about all contexts (visual and verbal) in which it was encountered. This representation is stored in the SM and evolves continuously with DETE's exposure to new experiences.

Implementation of the DM in DETE

The KATAMIC memory without modifications is used as the basis of the DM. This is possible since the characteristics of the KATAMIC memory (as described in Chapter 8) correspond well to the desired specifications of the DM outlined above.

Relation between the DM and the STM

The predictrons forming the DM component of the LTM are interleaved with predictrons that form the STM. As a result, both the STM and the DM have the same modality partitioning (i.e. visual and verbal) and also there is exchange of information between both memory categories.

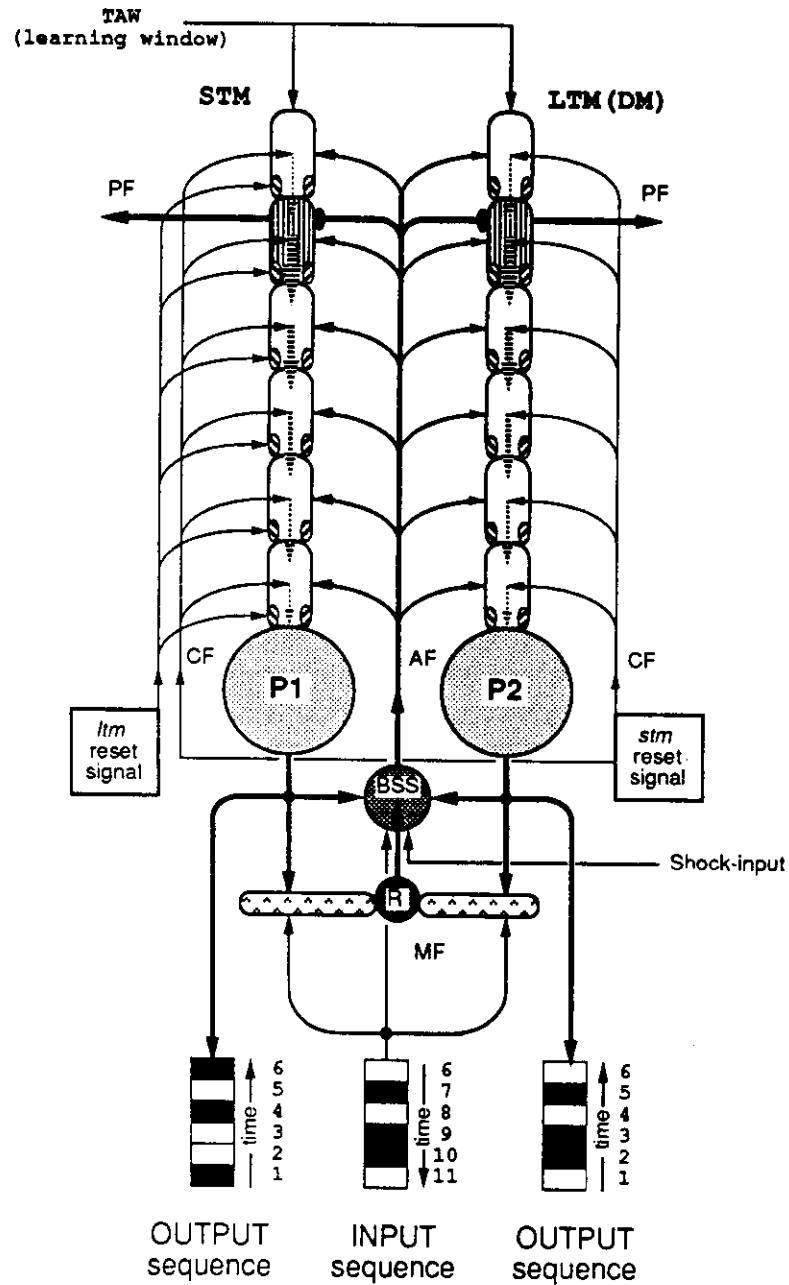


Figure 9.2: Relation between DM & STM in DEFE

The relation between the DM and the STM in DEFE is exemplified by the connectivity of two predictrons. To the left, P1 is a STM predictron while P2 is a LTM predictron. A single recognitron containing 2 DC's serves both predictrons. It controls the state of a single BSS.

The connectivity pattern between two neighboring predictrons, one of the DM type and the other of the STM type is shown in Figure 9.2. This STM/DM pair of predictrons is connected to other such pairs via parallel fibers (PFs). There are two important features of this wiring diagram: (1) The two predictrons share a common input, (2) They also share a common BSS (bi-stable switch). (3) The seeds of the two predictrons are located at the same level. (4) The DM has a special reset line for its *ltm* components. The purpose of this connectivity is to assure that, due to their physical proximity and the same seed location, the DM and the STM get almost the same PF inputs. The LTM influences the input to the STM because its output depends on prior experience. In other words, what the STM receives and stores depends on the expectations (predictions) developed by the LTM.

9.2.2 The Procedural Memory (PM)

Humans have the ability to remember the rhythm of a song without remembering the actual words. Actually, they can associate (learn) different words with one and the same rhythm. One can also remember the rhythm of the speech coming along a noisy phone line even when one cannot catch the actual words due to the noise. Knowing the context of the conversation, we can use this rhythm to partially reconstruct the possible content of the conversation. To explain this ability I postulate the existence of an unconscious memory mechanism which functions as a kind of Order Memory (OM) (learns order) and effectively counts the segments in the input stream and measures their duration. Such a mechanism can provide the basis of the human ability to learn to count successive events. For instance, this can be done simply by associating verbal labels (e.g., "one", "two", etc.) with the individual orders of events in the memory.

The PM module in DETE is used to store information about the relative positions and durations of the individual segments in any input sequence. Such segments are, for instance, the phonemes forming a word or the words forming a sentence. The part of the PM which stores such information is called the *Morphologic/Syntactic Procedural Memory* (MSPM) (Figure 9.1). The PM is also used to store information about the trajectories of the EYE and the FINGER. This part of the PM is called the *Motor Procedural Memory* (MPM) (Figure 9.1).

Architecture of the Morphologic/Syntactic PM (MSPM)

The morphologic/syntactic PM (MSPM) is also referred to as the Language Association Memory (see Figure 2.4) and is a part of the PM which is used to learn the order of phonemes in words (i.e. the morphology) and the order of words in sentences (i.e. syntax).

Components: The MSPM module consists of three basic components (Figure 9.3): (1) the *Verbal Bank* (VB) -- i.e. the Lexicon, (2) the *Transition Detectors Bank* (TDB), and (3) the *Order Memory Bank* (OMB). The OMB, the VB, and the TDB are essential components of the MSPM and therefore are all discussed together as an integrated mechanism.

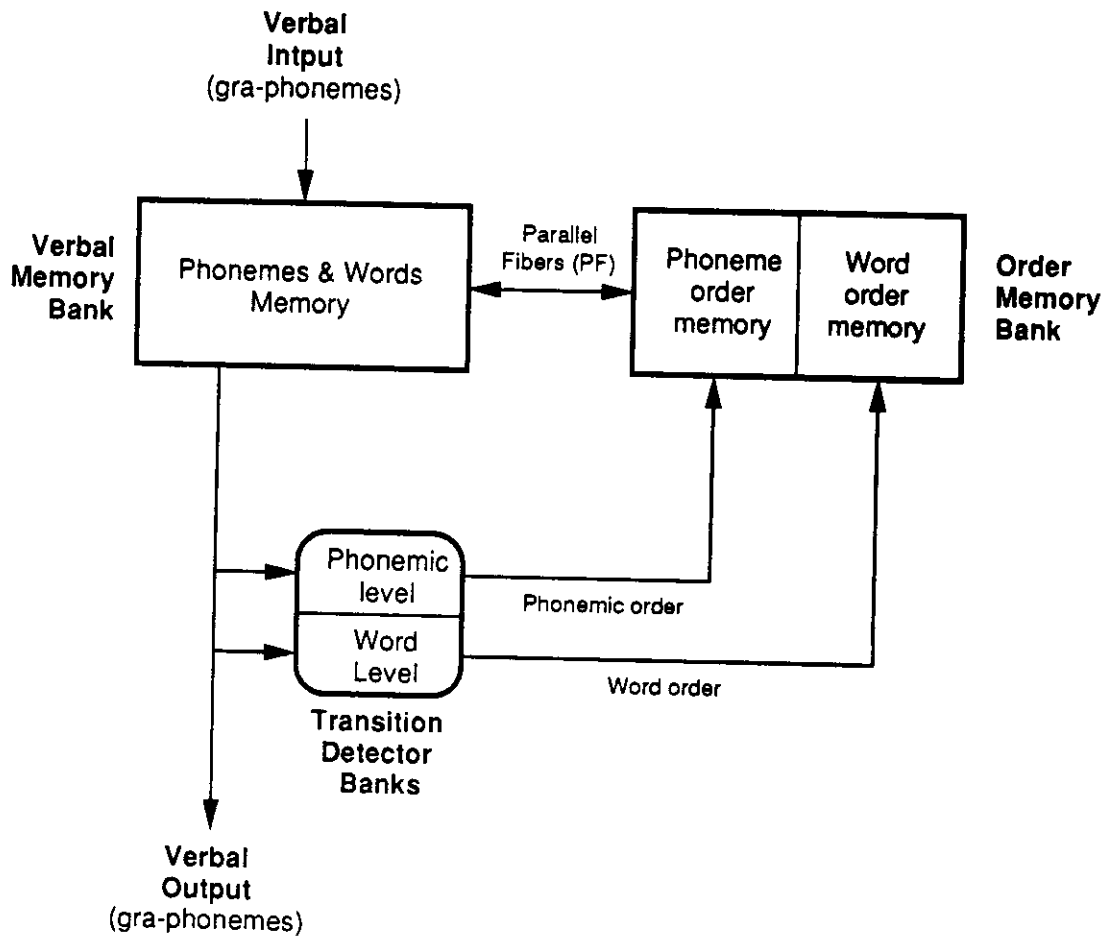


Figure 9.3: Block diagram of DETE's MSPM

Schematic drawing of the MSPM. MSPM takes verbal input in the form of gra-phonemes from the Word Encoding Mechanism (WEM) and produces verbal output (gra-phonemes) which is passed to the Verbal Activity Decoder (VAD) (see bottom left of Figure 2.4). For simplicity, the Phoneme Order Memory and the Word Order Memory are shown side by side in this figure while in practice the predictrons that form these memories are interleaved (see Figure 9.4).

Details of the neural circuitry of DETE's Morphologic/Syntactic Procedural Memory are provided in Figure 9.4.

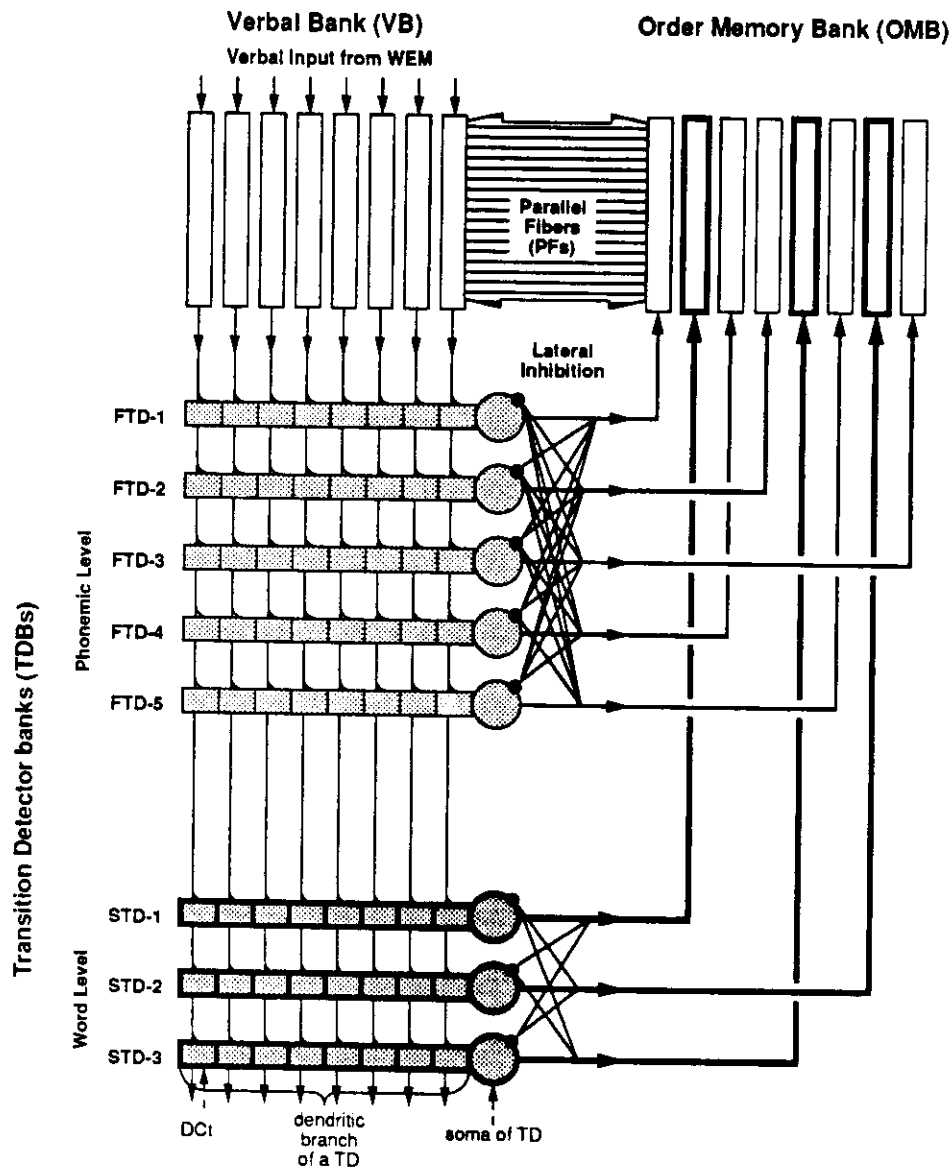


Figure 9.4: Neural circuitry of the MSPM

A small scale drawing of the MSPM. In the Verbal Bank (VB) only 8 of the 64 predictrons are shown. Each of these predictrons projects (makes a non-modifiable synapse of weight 1) to the corresponding Dendritic Compartments (DC^ts) of all Transition Detectors (TDs). The *Phonemic Level* of the Transition Detectors Bank has 5 Fast Transition Detectors (FTDs). The *Word Level* of the TDB has 3 Slow Transition Detectors (STDs). In the Order Memory Bank (OMB) only 8 predictrons are shown from the 64 in DEFE. The FTDs and STDs project one-to-one to randomly selected predictrons in the OMB.

The *Verbal Bank* is actually the set of predictrons that form the Verbal Memory in DEFE, i.e. the lexicon (8 out of the 64 VB predictrons are shown to the left in Figure 9.4).

The *Transition Detector Bank* (TDB) is a layer of neural elements (Transition Detectors -- TDs). TDs are not predictrons. Each TD has a soma characterized by an activation value. A single dendritic branch is also a part of each TD. This dendritic branch is composed by dendritic compartments DC^ts. The TDs are clock operated devices, i.e. at each time cycle each TD gets

inputs in parallel to their DC^ts, and computes a new state. The state of a DC^t is computed as a temporal XOR function of its most recent inputs (Table 9.1). Here I(t-1) is the input to a given DC^t at time (t-1), I(t) is the input to the same DC^t at time (t), and S(t) is its state at time (t).

| I(t-1) | I(t) | S(t) |
|--------|------|------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Table 9.1: Temporal XOR function

The Transition Detector Bank contains two types of TDs, *Fast* and *Slow* (see labels to the left in Figure 9.4). The Fast TDs (FTDs) are used to detect rapid transitions like those between consecutive phonemes in a word. In DETE the number of FTDs is 16, i.e. the maximal length of a word that DETE can process is 16 phonemes. The Slow TDs (STDs) are used to detect slower (i.e. longer lasting) transitions like the transitions between words. There are 16 STDs in DETE, i.e. the maximal length of a sentence that DETE can process is 16 words. The activation of a fast TDs is A^f and is computed as the sum of the activation values of the individual dendritic compartments. The activation of each DC^t is computed as a one-step-back temporal XOR function on its inputs (see formula 9.2). The activation of a slow TD is A^s and is computed similarly as that of a fast TD. The only difference is that the activity (states) over the dendritic branch is integrated over a longer period of time (e.g., 3 cycles) (see formula 9.3). This integrative mechanism ensures the slower response of the STDs to transitions in the external input.

$$A^f = \sum_{i=1}^{DC^t} \text{XOR}(I_i(t), I_i(t-1)) \quad (9.2)$$

$$A^s = \sum_{j=0}^3 \sum_{i=1}^{DC^t} \text{XOR}(I_i(t-j), I_i(t-j-1)) \quad (9.3)$$

All FTDs form a separate bank which is called the Phonemic Level since it is devoted to the recognition of the transitions between phonemes. The STDs form another bank -- the Word Level, which is used in the recognition between individual words.

The TDs within each Transition Detector bank have different thresholds ($th-1, th-2, \dots, th-n$). The relation between these thresholds is such that $th-1 < th-2 < \dots < th-n$. This ensures that one of the TDs responds first to the input. This TD is called a first order TD (TD-1). The TD which responds to the second segment of a sequence is respectively a second-order TD (TD-2) and so on. A segment is a subsequence of length 5 B-cycles which corresponds to a gra-phoneme. During the period while the activation value of a TD is above threshold, it fires continuously (i.e. bursts). If the activation value is below threshold, the TD is silent. Each TD, after it has stopped bursting, goes into a refractory period. If another segment of the input gets stabilized during this refractory period, the TD with the second lowest threshold picks up this input and gets into a bursting state. This is how successive phonemes are picked up in an orderly fashion by the TDs in the Phonemic Level. In other words, the main difference between the TDs in the different levels is in the sensitivity of their response to the length of the transition period. TDs in the Word Level bank are

less sensitive to fast transitions between segments than those in the Phonemic Level. This allows, for instance, the STD-1 in the Word Level bank to be continuously active all throughout the input of the phonemic sequence of the first word. It becomes silent only after the phonemic sequence representing the second word is input.

The *Order Memory Bank* is formed by a set of 64 predictrons (8 of them are shown as differently shaded bars to the right in Figure 9.4). The OMB stores information about the order and duration (number of patterns) of the individual segments that form the verbal sequences presented to the VB.

Connectivity: The outputs of the VB predictrons project topographically to the dendritic compartments of the TDs in both the Phonemic and Word Levels. Each axon of a VB predictron makes an excitatory connection (weight = 1) with the same level DC's of each of the TDs (shown as curved offsprings of the main connection lines in Figure 9.4). Within each level, the output of each TD projects back to all remaining TDs forming shunting inhibitory connections (lateral inhibition -- middle of Figure 9.4). The purpose of this lateral inhibition is to ensure that at any moment only one TD within each layer is active. Each TD also projects in a one-to-one way to a randomly selected subset of the predictrons forming the OMB. The VB and OMB are also interconnected via parallel fibers (PFs -- shown in Figure 9.4 as a bunch of horizontal lines connecting the two memory banks).

Note that this architecture (which in the current implementation of DETE is prewired) can instead be self-organized in the sense that the selection of TDs to represent the different orders is purely random depending on the values of their firing threshold which are provided at random.

Usage: The MSPM allows DETE to learn simple syntactic rules. For instance, it can effectively learn the rule "*In a NP the adjective comes before the noun*". Notice that DETE does not know anything about adjectives or nouns. DETE can learn that words associated with color and size feature maps precede words associated with shape feature map. So DETE can behave as if it has learned this rule. The MSPM stores associations between the verbal representation (in the VB) of any pair of words in the verbal input stream with a representation (in the OMB) of their relative sequential order in the sentence. For instance, the sentence "Small blue square moves up" leaves a trace in the VB of the word sequence as a whole and at the same time it leaves a trace in the OMB which contains information about the word order in this sentence. DETE's ability to learn such associations is due to the fact that during the training phase the verbal inputs have always a syntactic structure consistent with FIRLAN (or respectively SECLAN). During learning, each visual scene is associated with its verbal descriptions and an *order trace* left in the OMB. If at different occasions the same visual scene was described differently, then when the visual input is presented by itself, the strongest (most frequently heard) or the most recent (primed) verbal output is evoked in response in the MSPM.

Dynamics of the PM model

To illustrate the dynamics of the MSPM module at the word level, let us consider in detail the following experiment (Figure 9.5). DETE is taught that there is a proper order for placing of the different types of adjectives in front of a noun (the learning happens in the MSPM). For instance, in the sentence "small red ball", the adjective for size "small" comes before the adjective for color "red". At the same time we will see how the proper order of the phonemes forming the individual words is also learned. A step-by-step description of the processing in the MSPM is given below.

1
 gra-phonemically encoded verbal input of:
 "small red ball"

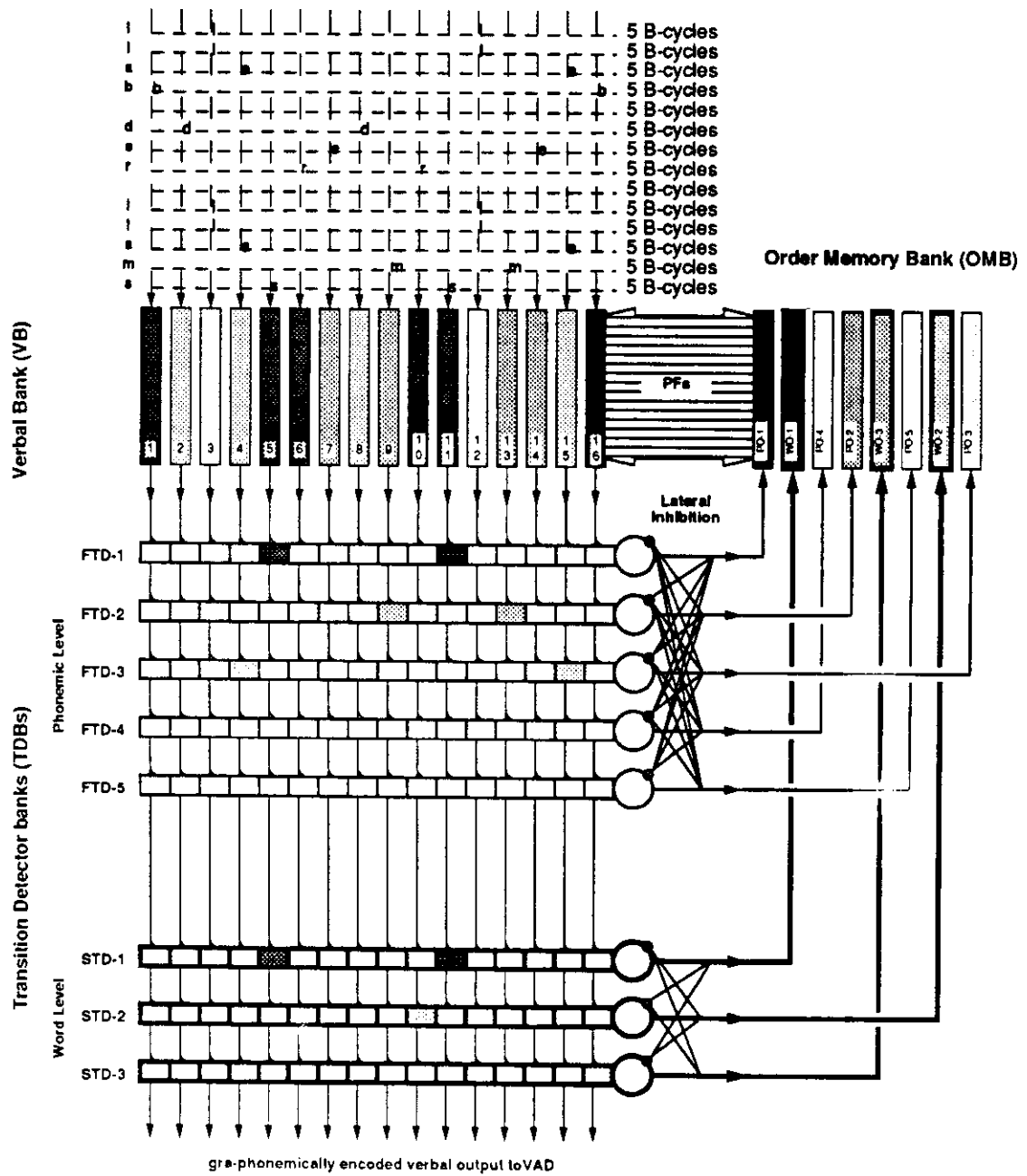


Figure 9.5: Learning word order

Schematic drawing of the MSPM dynamics during the learning of a syntactic rule. An abstract representation of the system dynamics during processing of the sentence "small red ball" is shown. The phonemic or word order is shown using a gray-scale coding (dark = beginning of word or sentence, light = end of word or sentence). The representation of the phonemes forming the words in the verbal input is also shown abstractly, i.e. only 2 bits (set to 1 and shown as the corresponding letters) are used instead of 6 bits used in DETE's verbal representation (see Chapter 4).

(1) The first gra-phoneme "s" in the word "small" is represented as a 5-step long temporal sequence (see section 2.3.1 and Chapter 4). It generates a burst of activity in the subset of the VB predictrons (5 & 11) which encode the gra-phoneme "s". This activity pattern is propagated to the corresponding DC's of the TDs in both Transition Detector Banks (Phonemic and Word Level). The FTD-1 in the Phonemic Level starts firing since it has the lowest threshold of all FTDs. It inhibits the rest of the FTDs. At the same time, it excites a subset of OMB predictrons (called phonemic-order-1 predictrons -- PO-1 shown as the first dark-shaded bar in the OMB in Figure 9.5). The bursting in the VB (representing "s") and in activity in the OMB (representing PO-1) are associated through the parallel fibers (PFs) connecting the two banks of predictrons. The onset of "s" also activates one of the STDs (STD-1) in the Word Level. STD-1 behaves similarly to FTD-1, i.e. inhibits the rest of the STDs and excites a different subset of the OMB predictrons (Word Order 1 predictron -- WO-1 -- the second dark-shaded bar in the OMB in Figure 9.5).

(2) When the first transition period comes (between "s" and "m") FTD-1 detects it (because its activation value A^f changes -- see formula 9.1) and stops firing. As a result the shunting inhibition output from FTD-1 to the rest of the FTDs is interrupted and during the transition all FTDs are silent. In other words, the MSPM does not learn how long the individual transitions are but does learn how to automatically detect transitions of different lengths (fast by the FTDs and slow by the STDs). At the same time (during the "s" to "m" transition) STD-1 in the Word Level continues to be active since the transition period is too short (1 cycle) to be detected. As a result, STD-1 maintains the information that the network is still processing the first word in the sentence.

(3) When the second gra-phoneme "m" is presented, FTD-2 detects it (starts firing) since it has the next lower threshold after the FTD-1, and also because FTD-1 is in a refractory period and therefore is unable to prevent FTD-2 from firing. FTD-2 in turn prevents all other FTDs from firing via its inhibitory influence. At the same time, it excites a different subset of the OMB predictrons (phonemic-order-2 predictrons -- PO-2 in Figure 9.5), and similarly as before, the activation of these OMB predictrons is associated with the activation representing "m" in the set of VB predictrons.

The process described above continues phoneme after phoneme until the end of the first word when a longer transition period is encountered. The effect of this transition period on the network dynamics is two-fold. (1) Within the Phonemic Level it causes a reset of the FTDs' states. While none of the FTDs are active during this long transition, all FTDs that have been in refractory period get out of this state and FTD-1 is again ready to fire first which will represent the first phoneme of the next word. (2) Within the Word Layer, STD-1 stops firing as a result of the long transition, goes in a refractory period and disinhibits the rest of the STDs. This allows STD-2 to respond selectively to the second word in the sentence.

The durations of the transition periods between the segments in the VB are critical for the type of reset which the TDBs perform. A sequence of short transitions (e.g., between the phonemes of a

word) causes sequential activation of FTDs (only one FTD is active at a time). Longer latency transitions (between words) trigger switching between the STDs in the Word Level of the hierarchy. At the same time, they reset all of the Phonemic Level FTDs (i.e. prepare them for a new word).

This MSPM described above segments out words and phonemes. It associates their verbal representations with the order information in the OMB only if the verbal input is in itself presegmented. In other words, if the phonemes forming a word in the verbal input are separated by short transition periods, while the words in the verbal input are separated by longer transitions. Notice that (in the current implementation) the proper segmentation of the external input is ensured by the design of the gra-phonemic representation scheme (see Chapter 4). However, pilot tests suggest that with a minor modification the KATAMIC model can be adapted to tolerate some degree of time-warping -- stretching or shrinking of the gra-phonemic input representations. The actual modification involves changing the way injected *stm* is delivered to the DCPs via the Parallel Fibers (see Formula 8.2). Instead of injecting *stm* only into the seed DCP, some *stm* is also injected in the nearby DCPs of the predictron. The distribution of the injected *stm* is Gaussian (centered at the seed DCP). The Full Width Half Maximum (FWHM) of the distribution defines the degree to which the model can correctly handle time-warped gra-phonemes. The bigger the FWHM, the higher the degree of time-warp tolerance and vice versa. Initial tests indicate that this design works as proposed. The reason is that instead of having only a discretized trace of *stm* in the dendritic branch of a predictron, we would have a more realistic (from the point of view of neurobiology) trace which is to some degree smeared along the length of the dendrite (Gaussian shape) which is equivalent to time smearing of the input. Using this modified KATAMIC architecture, DETE could successfully recognize gra-phonemes which are 4 or 6 B-cycles long (5 B-cycles is the standard duration).

9.3 Representation of time in DETE

DETE makes a distinction between modeling of dynamics of a sequence (via delay lines in the predictrons -- Table 5.1) and modeling time such as needed for representing verb tense in natural languages (e.g., moves, will move, moved). Time (in the latter meaning of the word) is represented in DETE as an additional dimension to the visual and verbal memories. DETE contains a neural structure called the Temporal Memory (TM). The purpose of the TM is to provide a time-buffer for consecutive sequences. This time buffer allows sequences occurring at different (but close to each other) *moments* to be associated. TM's connectivity and dynamics are chosen such that it can represent temporal characteristics of events (e.g., past, present or future) with respect to the present *moment*. The TM consists of 8 Temporal Planes connected in series. They are labeled from TP-0 to TP-7. Each TP is composed of a set of predictrons (one for each pixel in each feature map) and is divided into a verbal and a visual part. TP-0 represents the current moment in time (NOW). TP-1 represents the previous moment in time and so on. The sizes of the TPs map one-to-one to the sizes of the visual feature planes (16 x 16) and the verbal input vector (64 x 1). TP-0 gets information directly from the visual feature extractors and the verbal encoder. Each unit in TP-0 gets external input from one unit in the visual or verbal feature planes.

Activation is transferred between the predictrons of the TPs in temporal chunks (*moments*) -- i.e. it does not flow as if through a simple pipe-line. One *moment* is equivalent to 300 B-cycles and corresponds roughly to 3 seconds of real time. DETE contains a "moment clock" -- a procedure which controls the transfer schedule between TPs (see Figure 9.6). The "moment clock" generates a signal once every 300 B-cycles. The set of 8 TPs is divided dynamically into two interleaved

groups: Pumping TPs (PTPs) and Flushing TPs (FTPs). At each tick of the moment clock the first group “pumps in” information while the second group “flushes out” information (Figure 9.6A). The roles of pumping and flushing are reversed at the next tick of the moment clock (Figure 9.6B). At each *moment*, the pumping TPs get their input from their left-hand (flushing) neighbors with exception of TP-0 which gets external input. At the same time, (at the beginning of each *moment*) the flushing TPs get a “shock input”. A “shock input” is a short burst of activation passed to the inputs of the predictrons of all TPs which are in a flushing mode (FTPs). Each “shock-input” is procedurally generated by the “moment clock” and causes the memory content of the FTPs to be flushed-out. This same content is captured by the corresponding right-handed PTPs.

There are two types of connections between the TPs. The first type is between the corresponding neural elements in adjacent TPs. For reasons which will be explained later, these connections are called “slow” connections. Each predictron in a TP sends its output to the input of the corresponding predictron in the next TP. Therefore, each predictron serves both as a receiver of input from its left neighbor and a sender of output to its right-hand neighbor. In other words, the slow connections are used to pass a signal in a given direction (from TP-0 to TP-7) only between neighboring planes. The signal transmittal is initiated at each tick of the “moment clock”. The transfer line (wire) between two units in adjacent planes is gated externally. The gate is open for transfer only when the right-hand TP is in pumping mode (i.e. a receiver) and the left-hand TP is in flushing mode (i.e. a sender). At the beginning of each *moment* the receivers and senders reverse their roles.

The second type of connection is made between each predictron in TP-1 to TP-7 and the corresponding predictrons in TP-0 (Figure 9.6). Before making contacts with TP-0, these connections are also gated by the “moment clock”. These of connections, together with the connections made by the external input line to the TPs, are called “fast” connections because the information is reaches all connected predictrons within one B-cycle (rather than being transfered in *moment* chunks like the slow connections). In section 11.7 we show examples how this rapidly spreading information is used both during learning and during understanding of verb tenses.

There are also connections between the predictrons in each of the TPs itself. These connections are not shown in Figure 9.6, but are described in detail in section 10.1.3.

Each of the TP units is composed of a Short-Term Memory predictron and a Long-Term Memory predictron (see Figure 9.2). Both of these predictrons get the same input (i.e. they are connected in parallel) but they have different dynamics. The *stm* in all TPs (STM and LTM) is reset at the end of each *moment*. At the same time the *ltm* in all STM FTPs is reset and the *ltm* in the PTPs does not change. To ensure contiguity of perception and operation, during each *moment* only half of the predictrons in each TP are in the pumping mode whereas the other half are in the flushing mode.

The TM described above is quite rigid, since it assumes a constant duration of the temporal chunks. This limits the maximal length of a verbal sentence to 300 B-cycles (60 gra-phonemes or about 9 to 11 words). At the same time, short sentences are separated by a long pause since only one sentence is processed each *moment*.

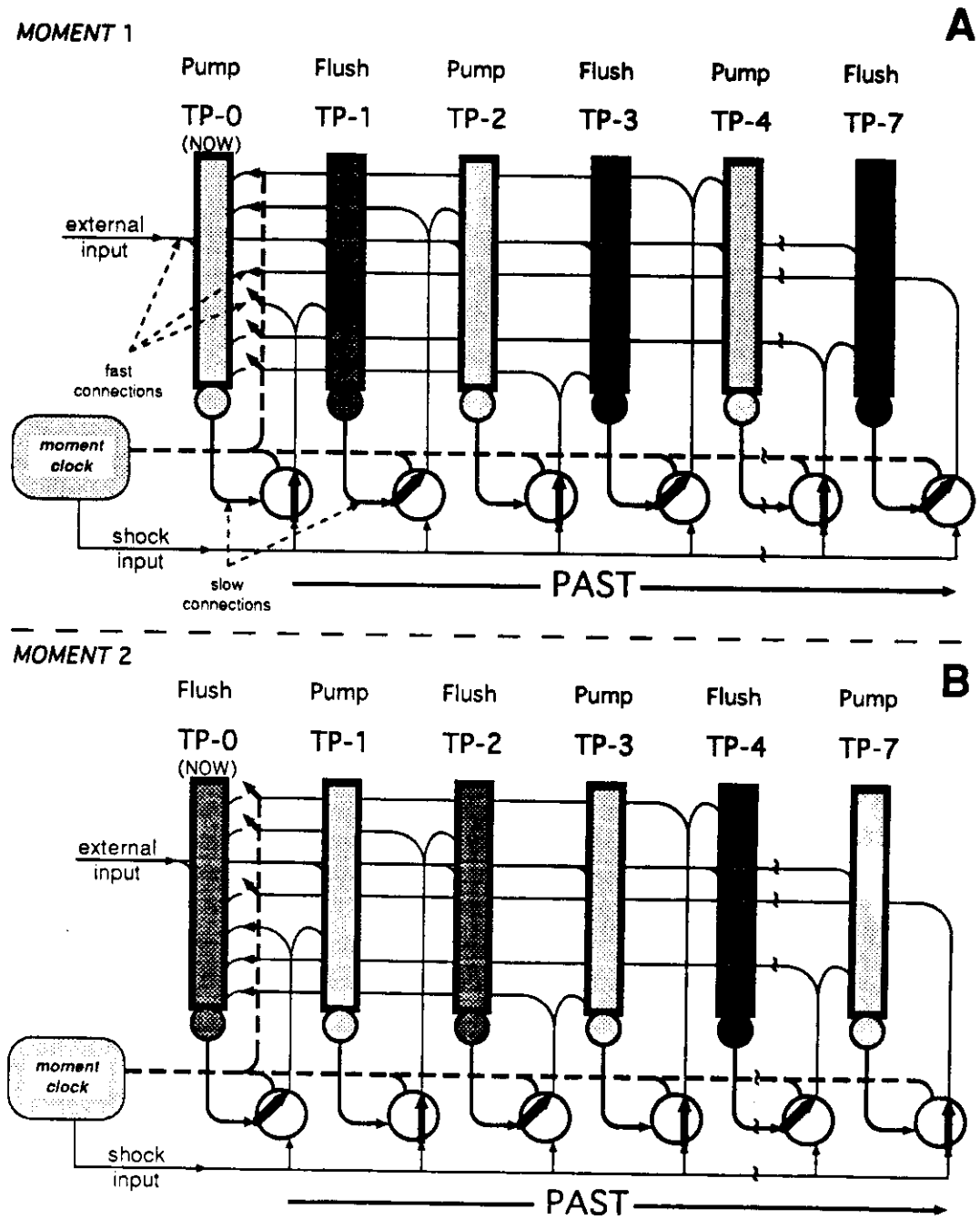


Figure 9.6: Temporal Memory (TM) -- connectivity and function

Schematic drawing of DETE's Temporal Memory (TM). The TM is shown in two consecutive *moments*. Only one predictron (the thermometer-shaped icons) per TP is shown. Slow connections are shown as thick lines with arrows at the end. Fast connections are shown as thin lines. Lines that cross at straight angles with other lines or with the predictrons do not make connections. The gates placed on the "slow" connections are shown as circles and thick arrows within them indicate their states (vertical arrows: -- the gates pass the shocking input through; oblique arrows: -- the gates pass through the inputs from the flushing predictrons). The dashed line originating at the moment clock shows the flow of control signal to the gates located **between the TPs** and the gates placed on the fast connections to TP-0.

The detailed dynamics of the TM will be illustrated in section 11.7 where we see on examples how DETE learns various verb tenses. Here is only a description of object representation in the TM. An object which is present for some time in the Visual Field (i.e. which has already some history) is represented by an activation (an oscillation over a set of TP-0 neurons). This activation is spread to the higher order TPs. It stays phase-locked to the visual features of the object since the predictrons which form the TPs do not introduce any phase shifts in the sequences which they process. This spreading activation leaves a slowly propagating (in chunks) time-trail of the history of the object in the TM. Notice that since each visual feature of an object can have different history, it is necessary that the TM forms a new dimension provided to all of the visual features. This architecture allows DETE to learn, for instance, the meaning of the word "transforms" while looking at objects that, as they move, their shape changes (e.g., from a circle to a square). At the same time DETE can learn the meaning of the word "shrinks" by looking at objects whose size decreases with time.

10 PUTTING IT ALL TOGETHER

This chapter focuses on how the individual modules of DETE were interfaced with each other to construct a complete functional system. Also discussed here are some modifications of the basic memory mechanism -- the KATAMIC sequential memory, which allow it to be used as Visual Feature Memories (VFMs) as well as Verbal and Motor Memories (VM & MM).

10.1 Characteristics of the memory modules in DETE

The KATAMIC architecture is the basis of all memory modules in DETE. Some of KATAMIC's parameters were tailored to support the necessary functional characteristics of the individual memory modules (visual, verbal, and motor) and to provide desired interfaces among these modules. The parameters which were varied include: (1) *Dimensionality* of the memory (1-D or 2-D arrangement of the predictrons); (2) *1-bit-density* of the inputs to the memories (the visual, the verbal, and the motor representations) and the temporal characteristics of the inputs to the different modalities; (3) *Internal partition of the memory modules* (stripes and columns -- see further discussion), and variations of the *connection strengths* (represented by the spatial decay constant T_s -- formula 8.2) of the synapses made by the parallel fibers within and between modules; (4) *Seed distributions* within each individual memory module.

10.1.1 Dimensionality of memory modules

The KATAMIC model described in Chapter 8 has a 1-D organization (the predictrons are arranged in one row -- form a vector). On the other hand, most of the memory modules in DETE require a 2-D organization, i.e. a square array of predictrons. Such spatial organization is necessary to support a direct mapping from the individual visual feature planes (VFps) to the corresponding visual feature memories (VFMs). For each VFp there is a corresponding VFM with the same dimensions (16 x 16 predictrons). Therefore, the number of predictrons forming each VFM is 256. Also, there are 8 temporal planes and therefore a total of $256 \times 8 = 2048$ predictrons. An increase in the number of predictrons results in a proportionate increase in storage locations in the memory.

10.1.2 1-bit-density of inputs

The combination of having a large number of pixels per VF plane and the choice of representation within the feature planes (see Chapter 3) results in a significantly lower 1-bit-density of the inputs to the VF memories as compared to the 1-bit-densities used to estimate the performance of the KATAMIC memory (see section 8.4). Since any given visual feature of an object is represented as 4 active pixels within the corresponding VF plane, and since this activity is represented as oscillations with a period of 5 B-cycles, the effective 1-bit-density of the inputs provided by the VF planes is $4/256/5 = 0.312\%$ (compared to 10 or 20% used typically in the KATAMIC memory experiments). This significantly lower 1-bit-density of the input patterns translates into an increase of the memory storage capacity of the individual VFMs (in terms of number of patterns stored) as compared to the memory capacity estimated in section 8.4.5 for denser patterns.

A similar assessment of the 1-bit-density can be done for the verbal input representation. While the activity in the Visual Field is encoded in oscillations, in the verbal representation the activity is encoded in "bursts". A given element of the 64 bit long verbal vector (see Chapter 4) is active for 5 consecutive B-cycles -- a *burst*, after which it becomes silent. Since each gra-phoneme is represented by $3 \times 2 = 6$ active bits corresponding to the three basic frequency formants, the 1-bit-density of the verbal input is $3/64 = 4.68\%$. On the basis of the number of predictrons allocated to the verbal memory and the 1-bit-density of the verbal input one would expect that the verbal memory would not be able to learn hundreds of sequences (words), since the KATAMIC's memory capacity was shown to be in the range of tens of words (see section 8.4.5). This, however, is not the case. It is important to notice that while the verbal representation is passed **directly** only to the verbal memory, information about the verbal input is spread **indirectly** to **all** Visual Feature Memories (see Figures 10.5 & 10.6). As a result, in effect the number of **storage** locations for verbal inputs is increased. This allows DETE to maintain a relatively large lexicon of about 100 different words where the length of an average word is about 35 B-cycles (7 gra-phonemes \times 5 B-cycles per gra-phoneme).

10.1.3 Connection patterns and strengths within modules

A common principle in the design of all memory modules in DETE was to provide sufficient hardware (predictrons) necessary to represent separately the features of several (up to 4) individual objects that appear simultaneously in the Visual Field. The object segmentation was done in tandem by two different encoding mechanisms: (1) segmentation in the temporal domain, and (2) separation in the spatial domain. The choice of such double encoding was inspired by our interpretation of recent neurophysiological data. More specifically, the temporal aspect of the representation, was suggested by the observation of relatively high frequency oscillations (about 40 Hz) in the visual cortex that can be correlated with features of objects in the Visual Field (Gray et al., 1989; Gray and Singer, 1989; Gray et al., 1990). The spatial aspect of the representation was suggested by the fact that in the neocortex the neuronal activity is very sparse and therefore it is reasonable to expect that there is no overlap between the neural assemblies representing the features of different objects. Moreover, it is quite unlikely that a particular neuron can double or triple its oscillation frequency from, say 100 to 200 or 300 Hz, which will be necessary if it were encoding features of two or three different objects. The spatial encoding of a particular feature of different objects (e.g., shape or color) can be realized in various ways. One possibility is to completely segregate in space the neural assemblies that represent different objects. For instance, they can be placed in separate *stripes* or *columns* of the memory. Such organization can be found in various parts of the neocortex and especially in the visual cortex (Hubel and Wiesel, 1962). Another possibility is to interleave in space the various assemblies. Such a type of representation is typical, for instance, for the olfactory cortex (Haberly, 1990). In our implementation, as discussed below, the former approach was chosen for the individual memory modules.

Each feature plane in DETE represents the features of different objects within separate areas of the plane. Such areas can take one of two different shapes called *stripes* and *columns*. *Stripe* is a group of predictrons organized in a 2-D array of dimensions 4 by 16. Each *stripe* represents only one feature of only one object. *Stripes* are used to represent object features in three out of the five visual feature memories -- the shape, the size, and the color memories (see Figure 10.1 for an example of a stripe in the Shape Feature Memory). The other two feature memories (the location and motion) use *columns* to represent the features of individual objects. *Columns* are groups of 4

predictrons arranged in a square (see Figures 10.2 & 10.3 for examples of columns in the Location and Motion Feature Memories).

10.1.4 Seed distribution

In the KATAMIC model, the choice of the dendritic compartments which serve as seeds (i.e. where the bifurcation of the ascending fibers occurs) was done at random with a priori defined density of the seeds (see Table 8.1). However, to allow for a gradual change of the representations of the visual features within individual modalities (e.g., in the siZe Feature Plane -- from small to large; in the Motion Feature Plane -- from slow to fast, etc) in DETE it was necessary to go from random to structured seed distribution. The following general principle of seed distribution was used: predictrons that are close to each other within a given feature memory have their seeds also close to each other along the dendritic branches. The most commonly used seed distribution was a diagonal distribution.

A detailed description of the seed distributions and connectivity patterns of the Parallel Fibers within each of the visual memory modules is provided below.

(1) *Shape Feature Memory (SFM)*. The SFM (a 16 x 16 set of predictrons) is divided into 4 x 16 = 64 *stripes* each of which contains 4 predictrons. Each *stripe* represents the shape of a single object. The locations of the seed-DCPs in the *stripes* of the SFM are shown in Figure 10.1.

A more detailed view of the connectivity pattern within the SFM is shown in Figure 10.2. In the X dimension (within a *stripe*) the connection strengths are all set to 1. This gives all predictrons within a *stripe* the same status (redundancy of coding). In the Y dimension (between *stripes*) the connection strengths decrease exponentially with distance in both directions from the seed-DCP. Such connectivity pattern allows for a smooth transition between neighboring *stripes*. There are no connections between the *stripes* along the X dimension. In other words, there is no interference between the shape representations of two or more objects which appear simultaneously in the Visual Field. This aspect of the current set-up might not be realistic, since there are well known perceptual illusions (e.g. based on line drawings) in which our perception of a particular geometrical feature (like line curvature) is distorted due to effects of perceptual interference with other shapes.

(2) *siZe Feature Memory (ZFM)*. The distribution of the seeds in the ZFM is the same as that in the SFM. The reason is that both the shape and the size representations are designed such that there is a gradual change of feature value along the feature dimension (the Y axis) of the feature plane. In the ZFM, the size dimension of an object varies monotonically from small to large along the Y axis.

(3) *Color Feature Memory (CFM)*. The distribution of the seeds in the CFM is the same as that in the SFM and ZFM. This choice is made for the same reasons described above. Here along the Y axis are encoded different color values ranging from black to white.

(4) *Location Feature Memory (LFM)*. The seed-DCPs in the Location Feature Memory are arranged in a spiral plane with an axis parallel to the Z axis and centered in the middle of the LFM (Figure 10.3). The projection of this axis onto the Visual Screen coincides with the center of the EYE (retina). The spiral plane makes one complete turn along its axis. The direction of the turn is clockwise but this is not critical. The reason for choosing this specific seed-DCP distribution is that it, together with the phase-lags based representation of the distance of the object to the center of the retina, provides a unique representation of the location of an object in the Visual Field. Effectively, the spiral angle and the distance from the axis provide a polar coordinate system in which the location of each individual object can be encoded in a unique way.

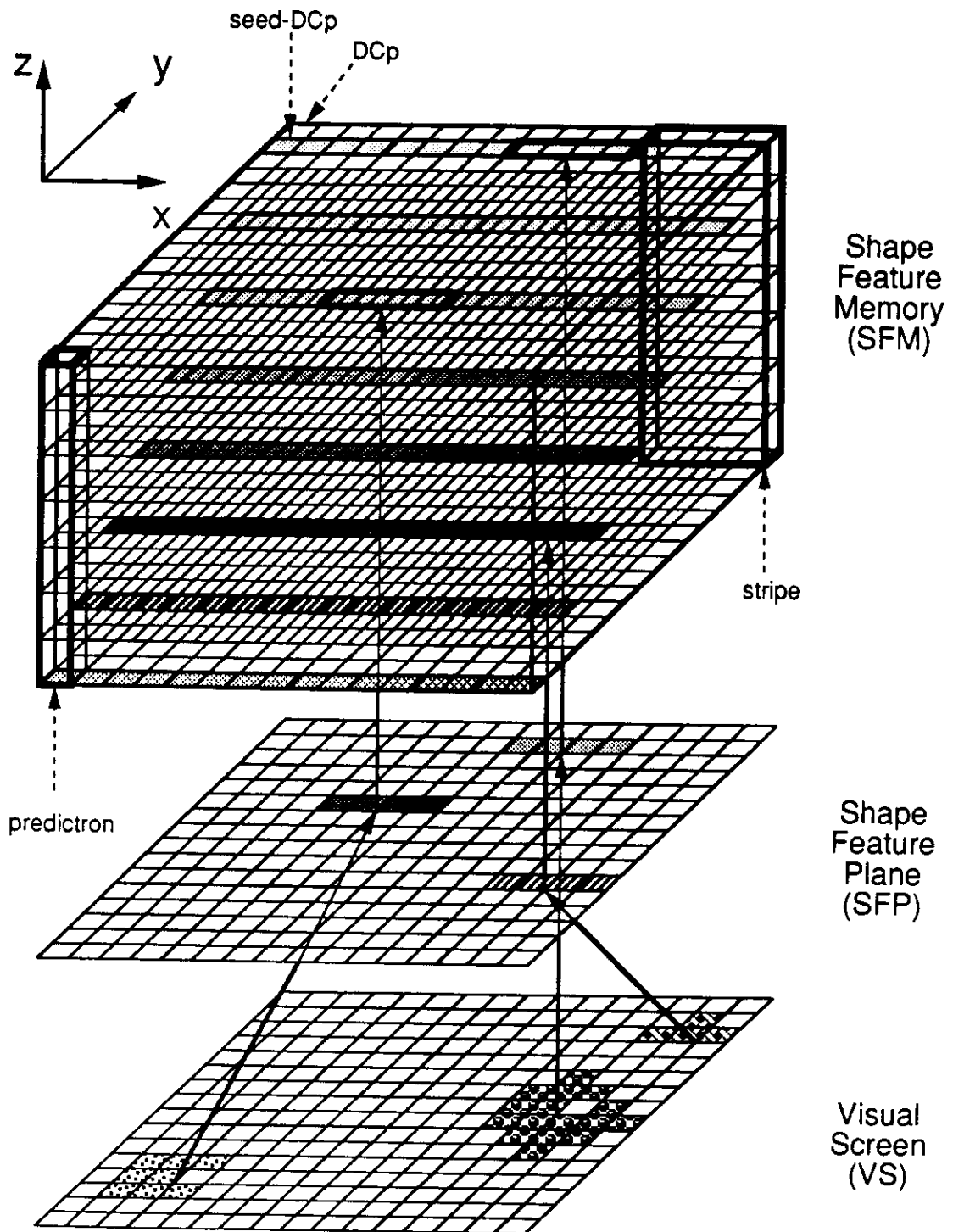


Figure 10.1: Distribution of seed-DCPs in SFM

Schematic drawing of the data flow between the Visual Screen (VS -- only 256 pixels are shown out of 4096 total), the Shape Feature Plane (SFP) and the Shape Feature Memory (SFM). Three noisy objects are shown on the VS (a square, a circle, and a triangle). The Shape Feature Extractor (not shown) has mapped the shapes of the objects onto the SFP (see Figure 3.3 for details). The mapping from the SFP to the SFM is topographic (one-to-one). Predictrons forming the SFM are shown as vertical bars. The 16 possible shapes that the SFP can represent are shown on the Y axis. Four objects having the same shape can be simultaneously represented along the X axis. The dendritic compartments of the SFM predictrons are organized along the Z axis. Of the 128 DCPs per predictron in the current implementation of DETE, only 8 are shown (white rhomboids). The *seed-DCPs* in all *stripes* of the SFM (in a given TP) are shown as shaded rhomboids. The organization of the *seed-DCPs* in consecutive predictrons along the Y axis is diagonal.

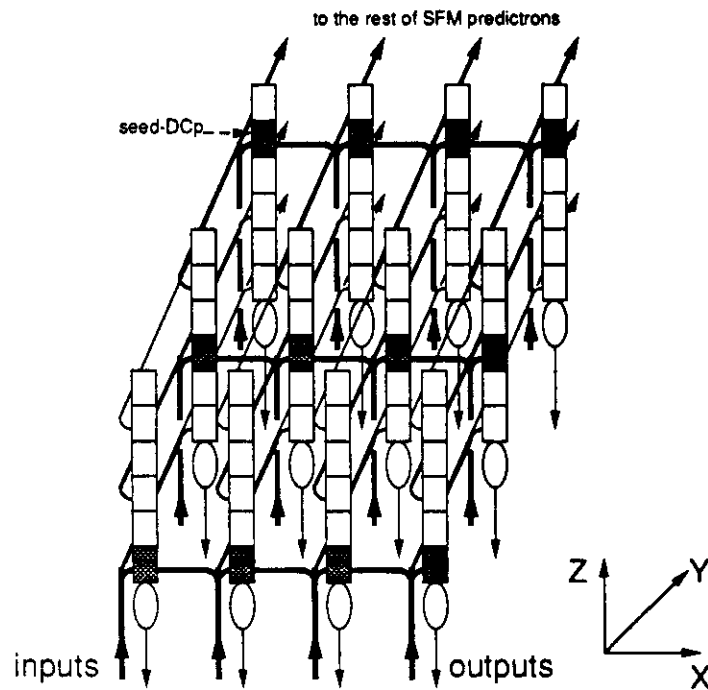


Figure 10.2: Details of the *seed-DCPs* distribution in the SFM

Schematic drawing of the connectivity pattern within the SFM. Out of the 256 predictrons forming the SFM only 12 are shown as thermometer-shaped icons. The *seed-DCPs* are shown as shaded rectangles. The organization of the *seed-DCPs* in consecutive predictrons along the Y axis is diagonal (in the YZ plane). The strength of the connections between predictrons is encoded by the thickness of the connecting lines. In the X dimension all connection strengths are set to 1 (shown as thick lines). In the Y dimension connection strengths decrease exponentially with distance in both directions from the *seed-DCP* (shown as gradually thinning lines).

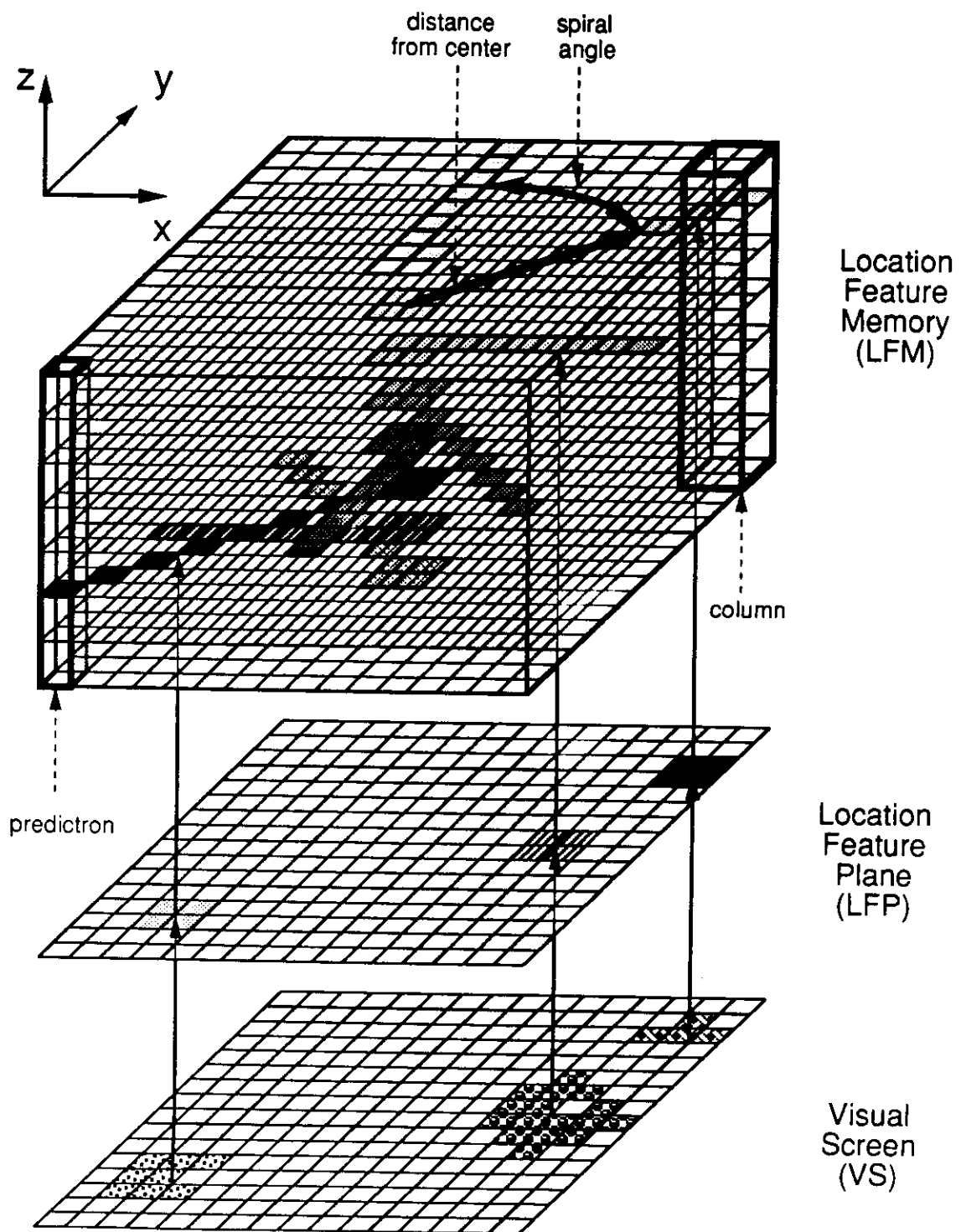


Figure 10.3: Distribution of seed-DCPs in LFM

Schematic drawing of the data flow between the VS (only 256 pixels are shown out of 4096 total), the Location Feature Plane (LFP) and the Location Feature Memory (LFM). The same objects as in Figure 10.1 are shown on the VS. The Location Feature Extractor (not shown) has mapped retinotopically the locations of the objects' centers of mass onto the LFP (see Figure 3.6 for details). The mapping from the LFP to the LFM is also one-to-one. Predictrons forming the LFM are shown as vertical bars. The DCPs of the LFM predictrons are arranged along the Z axis. Of the 128 DCPs per predictron in the current implementation, only 8 are shown (white rhomboids). A *column* is shown as 4 predictrons arranged in a square. The seed-DCPs are shown as shaded rhomboids. The arrangement of the seed-DCPs in the volume of the LFM is in a spiral plane (shown as a staircase).

A more detailed view of the connectivity pattern within the LFM is shown in Figure 10.4. In the radial dimension all connection strengths are set to 1. This allows all predictrons arranged along a radius to communicate more strongly among each other than predictrons that are not along the same radius. This feature is used for learning spatial relations like “in-front” and “behind” (see section 11.5.2). In the X and Y dimension the connection strengths decrease exponentially with distance in both directions from the seed-DCP. Such connectivity pattern allows for a smooth transition between neighboring *stripes*. It also allows for encoding of angular distance between predictrons.

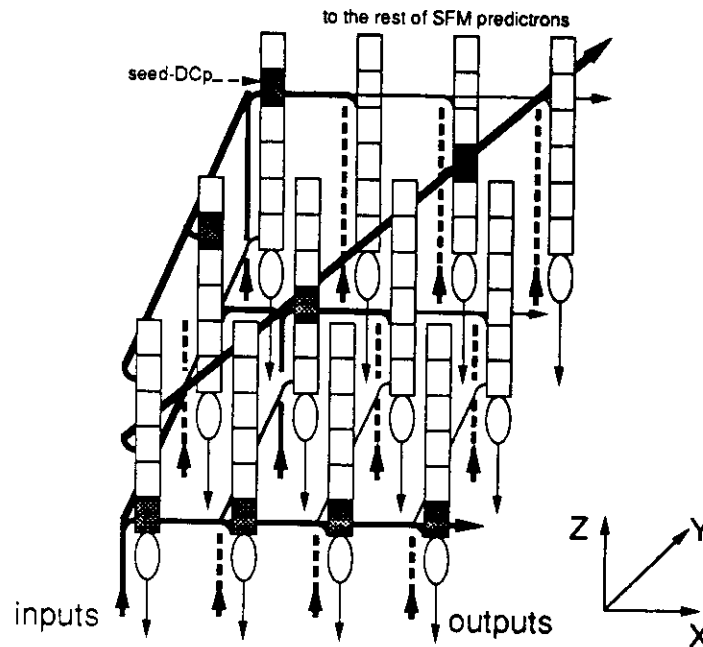


Figure 10.4: Details of the seed-DCPs distribution in the LFM

Schematic drawing of the connectivity pattern within the LFM. Out of the 256 predictrons forming the LFM only 48 are shown. Every 4 predictrons within a column are shown as a single thermometer-shaped icon. This piece of the LFM has been chosen such that the left-most predictron corresponds to a predictron in the center of the LFM (i.e. it coincides with the axis of the spiral plane). The seed-DCPs are shown as shaded rectangles. The organization of the seed-DCPs in consecutive predictrons within the 3-D volume of the LFM follows a spiral plane. The connections originating at only three of the predictrons are shown (solid thick lines used for inputs). The strength of the connections between predictrons is encoded by the thickness of the connecting lines. In the radial direction dimension all connection strengths are set to 1 (shown as thick lines). In the X and Y dimensions connection strengths decrease exponentially with distance in both directions from the seed-DCP (shown as gradually thinning lines).

(5) *Motion Feature Memory (MFM)*. The seed distribution in the MFM is the same as that in the LFM, however the mapping from the MFP and MFM (Figure 10.5) is different from that between the LFP and LFM. The choice of this seed-DCP distribution was made on the same basis as for the LFM, since here we need again something that amounts to a polar coordinate system representation. However, in the MFM the distance from the center corresponds to the speed of motion and the angle corresponds to the direction of motion.

(6) *Verbal Memory (VM)*. The predictrons in the verbal memory are arranged along a single dimension. There are 64 predictrons and each of them codes for a specific frequency range (see Chapter 4). The distribution of the seeds in this memory module is diagonal (Figure 10.6). Similarly to the cases of the shape, size, and color memories, this distribution ensures that frequencies that are close to each other are represented in predictrons that are near each other.

10.1.5 Winner Take All mechanism (WTA)

At each B-cycle, the output pattern of each of DETE's Visual Feature Memories (VFMs) is piped thru a Winner Take All (WTA) mechanism (see Figure 10.8). The WTA mechanism allows only one predictron in a given VFM (the one that has the strongest 1-bit prediction -- i.e. has the maximal activation level of all predictrons in the specific VFM -- see Formula 8.11) to pass its output to the Verbal Memory. The outputs of all other weaker predictions (i.e. those with smaller activations) are not passed thru. In other words, instead of 1s, 0s are passed along the axons of these predictrons. The WTA mechanisms are based on lateral inhibition between the predictrons from which each VFM is composed. Their function is to allow only one (the strongest) of all possible responses to be generated by the particular VFM (SFM, ZFM, LFM, CFM, and MFM). Since each of the predictrons has an activation threshold Θ^P (see Formula 8.12) effectively each WTA has the same generation threshold. At any B-cycle, depending on the magnitude of this threshold, a particular VFM generates an output or is silent. In the current implementation the thresholds of the WTA mechanisms are set externally by the user. Low settings of the thresholds (e.g., 0.05) result in a very "verbal" DETE while high settings (e.g., 0.1) correspond to a relatively quiet or, in the extreme case, a completely silent DETE (see "Learning word order" in section 11.3.2). As will be seen on examples in Chapter 11, the length of a visual scene description generated by DETE depends on (1) the thresholds of the WTA mechanisms, (2) all prior experiences, i.e. the distribution of utterances of various lengths in the training set (the most frequent length utterances have left the strongest trace in the memory), and (3) the most recent experiences (i.e. priming effects).

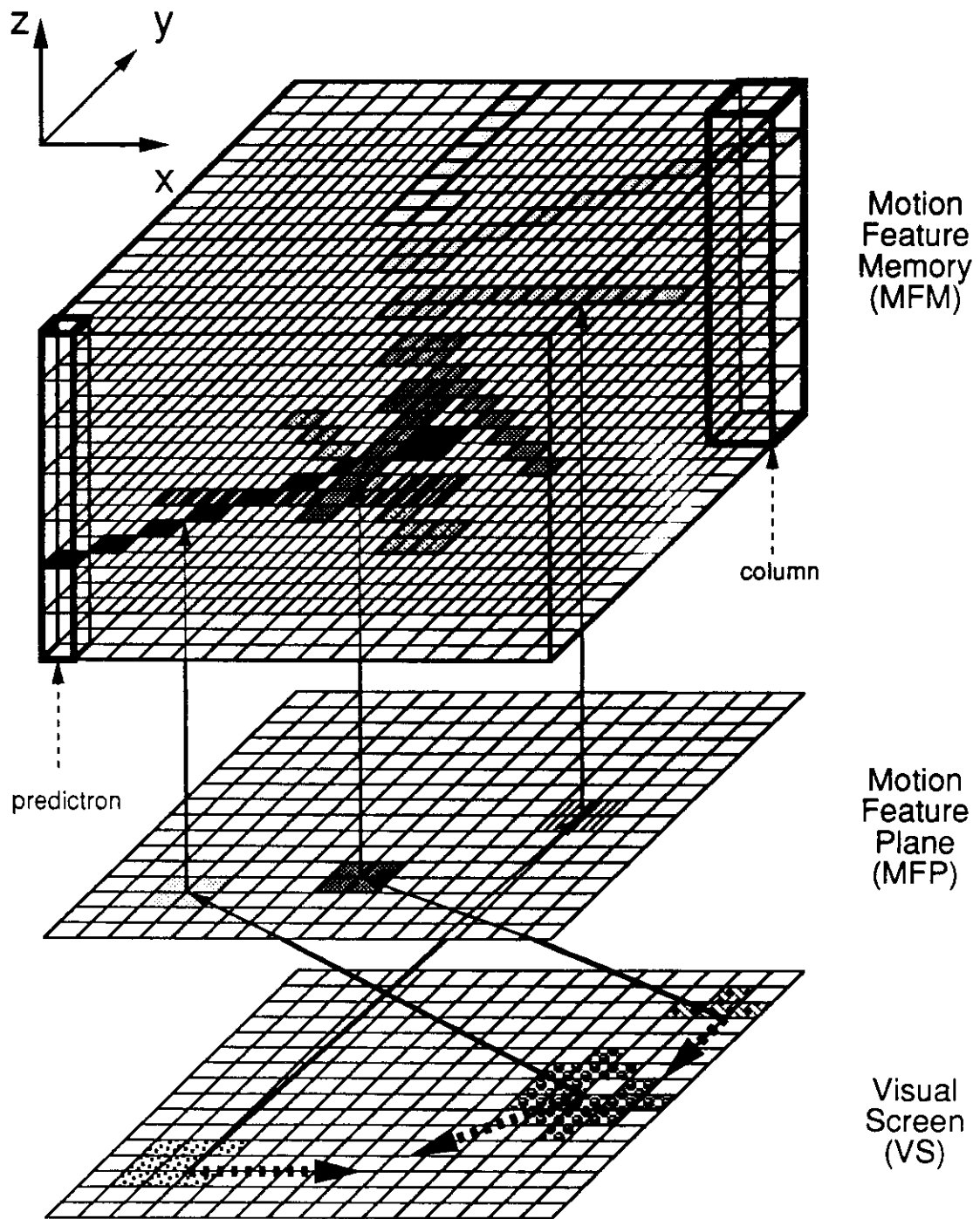


Figure 10.5: Distribution of seed-DCPs in MFM

Schematic drawing of the data flow between the VS, the Motion Feature Plane (MFP) and the Motion Feature Memory (MFM). The same objects as in Figure 10.1 are shown on the VS. The Motion Feature Extractor (not shown) has mapped the motions of the objects onto the MFP. Notice that the mapping is not topographic (see Figure 3.7 for details). The mapping from the MFP to the MFM is one-to-one. Predictrons forming the MFM are shown as vertical bars. The DCPs of the MFM predictrons are arranged along the Z axis. Of the 128 DCPs per predictron in the current implementation, only 8 are shown (white rhomboids). A *column* is shown as 4 predictrons arranged in a square. The seed-DCPs are shown as shaded rhomboids. The seed-DCPs are arranged in a spiral plane (shown as a staircase) within the volume of the MFM.

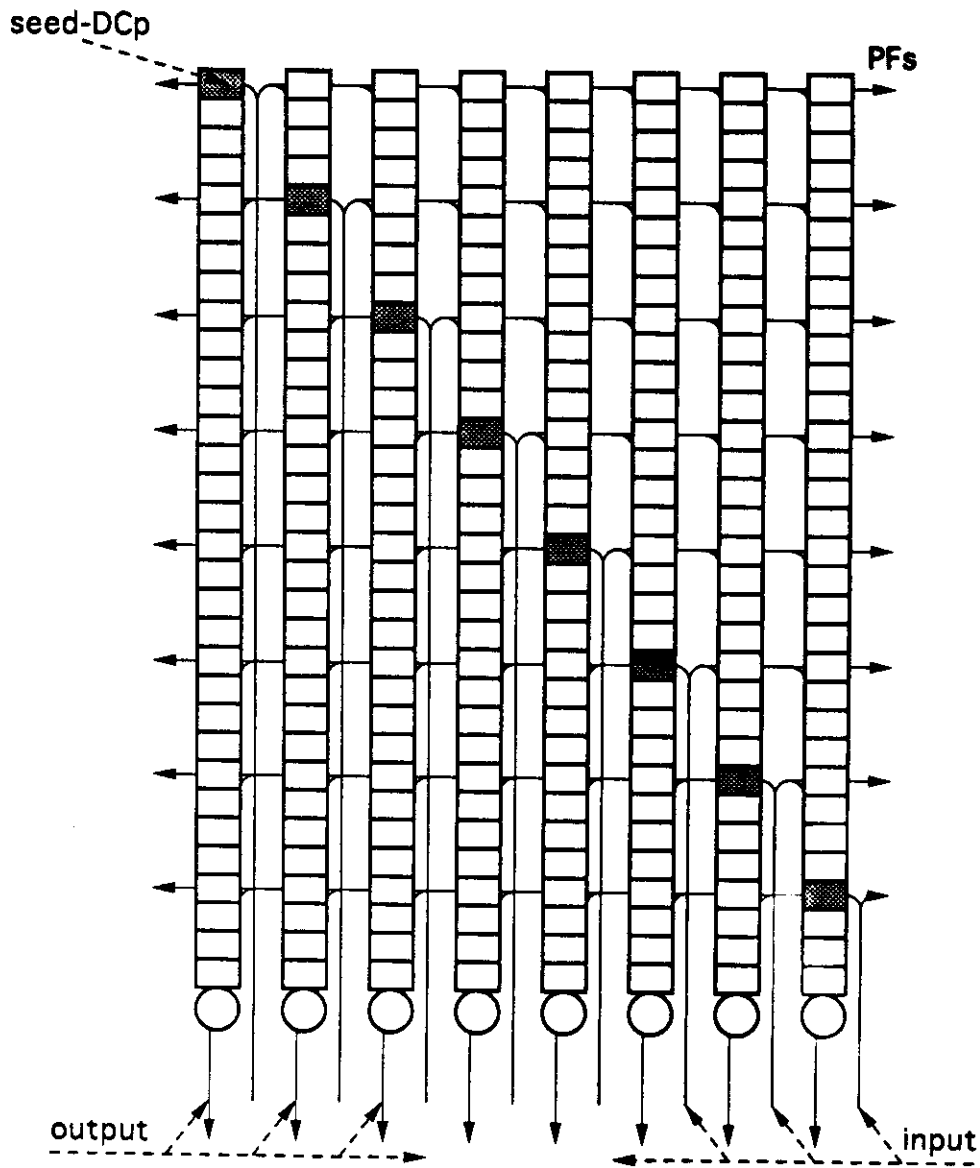


Figure 10.6: Distribution of seed-DCPs in VM

The verbal Memory is composed of 64 predictrons (only 16 are shown columns). There are 256 DCPs per predictron (only 64 are shown as small rectangles forming the columns). The seed-DCPs are placed on the diagonal and are shown as dark-shaded rectangles.

The WTA mechanism serves another purpose as well. Namely, the outputs the WTA are used to control the clock phase of the Focus of Attention Master (FAM), i.e. the position of the TAW in the space of possible phases of the FAM clock. The WTA output has a given phase. A procedural module then uses this phase to reset the FAM to be the same phase. Control of the FAM phase is important for learning of certain linguistic skills such as verbal description of size and location relations. An example of how the WTA mechanism and the FAM is used in learning of size relations between two objects is given in section 11.5.1.

In DETE, the pattern of activity generated in the Verbal Memory in response to a verbal input and filtered by the WTA mechanism can be used in a visual search task. For instance, DETE is shown several objects one of which is red. By a verbal command "Where is the red" or "Which one is red" or "Show me the red", DETE is also given the task to find the red object (i.e. focus the attention on it) and to verbalize "red". To accomplish this task DETE performs a sequential search in the Visual Screen. It moves its focus of attention randomly from one object to another (never to an empty space). To illustrate the function of the WTA, consider further how DETE accomplishes this task. The verbal input and specifically the word "red" generates a sustained expectation of a red object. This expectation is represented by activity in the red area of the Color Feature Memory. The threshold of the WTA is set such that this activity is subthreshold, i.e., it is not passed thru the WTA. When the object with the red color falls on the retina, the signal representing its color feature (red) is passed from the CFP to the CFM. Since the CFM is already biased towards red by the verbal input, the additional activation of the red area of the CFM (see Figure 3.5) exceeds the WTA threshold. As a result WTA mechanism passes out the signal from the red area back to the Verbal Memory. This activity, in turn, causes the Verbal Memory to verbalize the word "red". Notice that the WTA thresholds are set so that none of the Visual Features Memories by themselves (i.e. without the bias from the verbal memory activation) can produce a response (i.e. reach the WTA threshold).

10.2 Interfacing the individual memory modules

To be able to support the types of functionality which we expected from the system, the patterns of connectivity between the individual Visual Feature Memories on the one hand and between Visual and the Verbal Memories on the other hand were designed differently.

10.2.1 Connectivity patterns between visual modules

The different Visual Feature Memories are not directly interconnected. For this reason DETE does not learn by itself associations between visual features unless they are pooled together by the meaning of verbal inputs like: tomato = red circle; orange = orange circle; grapefruit = large yellow circle; comet = yellow circle moving down along the diagonal. This connectivity pattern was chosen for simplicity and does not reflect the neuropsychological and neurophysiological reality. It is well known that animals and pre-lingual children can easily associate various visual features. For instance, cats and dogs do not have verbal memory, and yet they manage to co-associate visual features. I.e. they learn that, say, rat-shapes are associated with grey color. Also, the connectivity

among various functional modules in the visual cortex suggests that often the same neural circuitry is shared by various features like color, motion, etc. (Van Essen and Maunsell, 1983).

10.2.2 Connectivity between verbal and visual memory modules

The outputs of all predictrons in the Verbal Memory are connected via *inter-modular* fibers making non modifiable synapses of weight 1 to all predictrons in the Visual Feature Memories (Figure 10.7). Such connectivity pattern is necessary so that any word can be equally easily associated with any visual representation, also because it is desirable that activity in any part of the verbal memory can affect the activity in any part of each VFM. The particular choice of using *output-to-input* connections (the outputs of the verbal predictrons spread to the inputs of the visual memory via *inter-modular* fibers) as opposed to *input-to-input* connections (the input lines to the verbal memory spread to the visual memories as parallel fibers and vice versa) is done because it is desirable that DETE can “imagine” a visual feature in response to a “hidden articulation” of the corresponding word. Such capability is only possible if an *output-to-input* connectivity is chosen and cannot be supported by an *input-to-input* connectivity pattern. The connectivity in the opposite directions -- from visual to verbal memory follow the same fully connected pattern and the synapses made by the visual to verbal inter-modular fibers are also non-modifiable and set to 1.

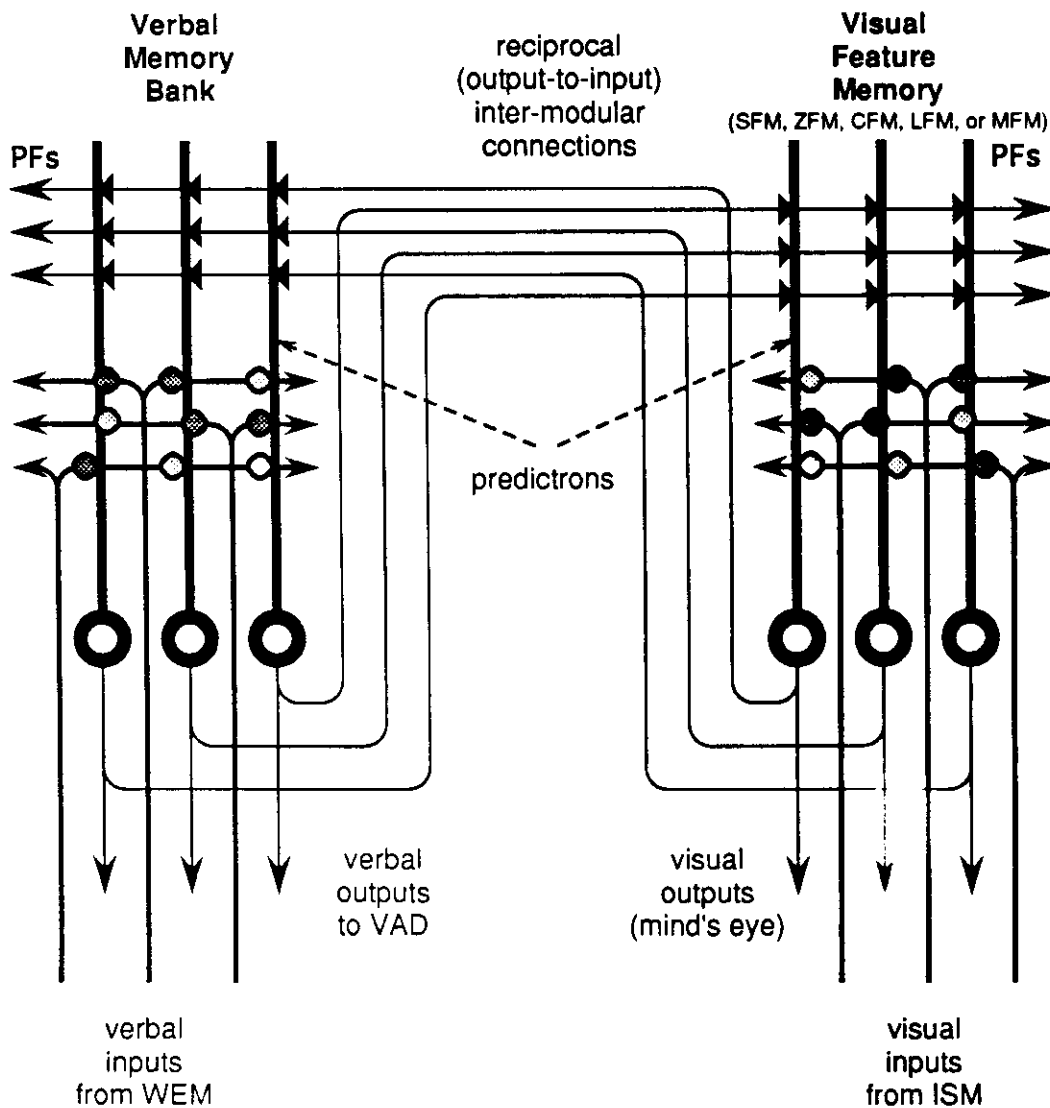


Figure 10.7: Inter-modular connectivity

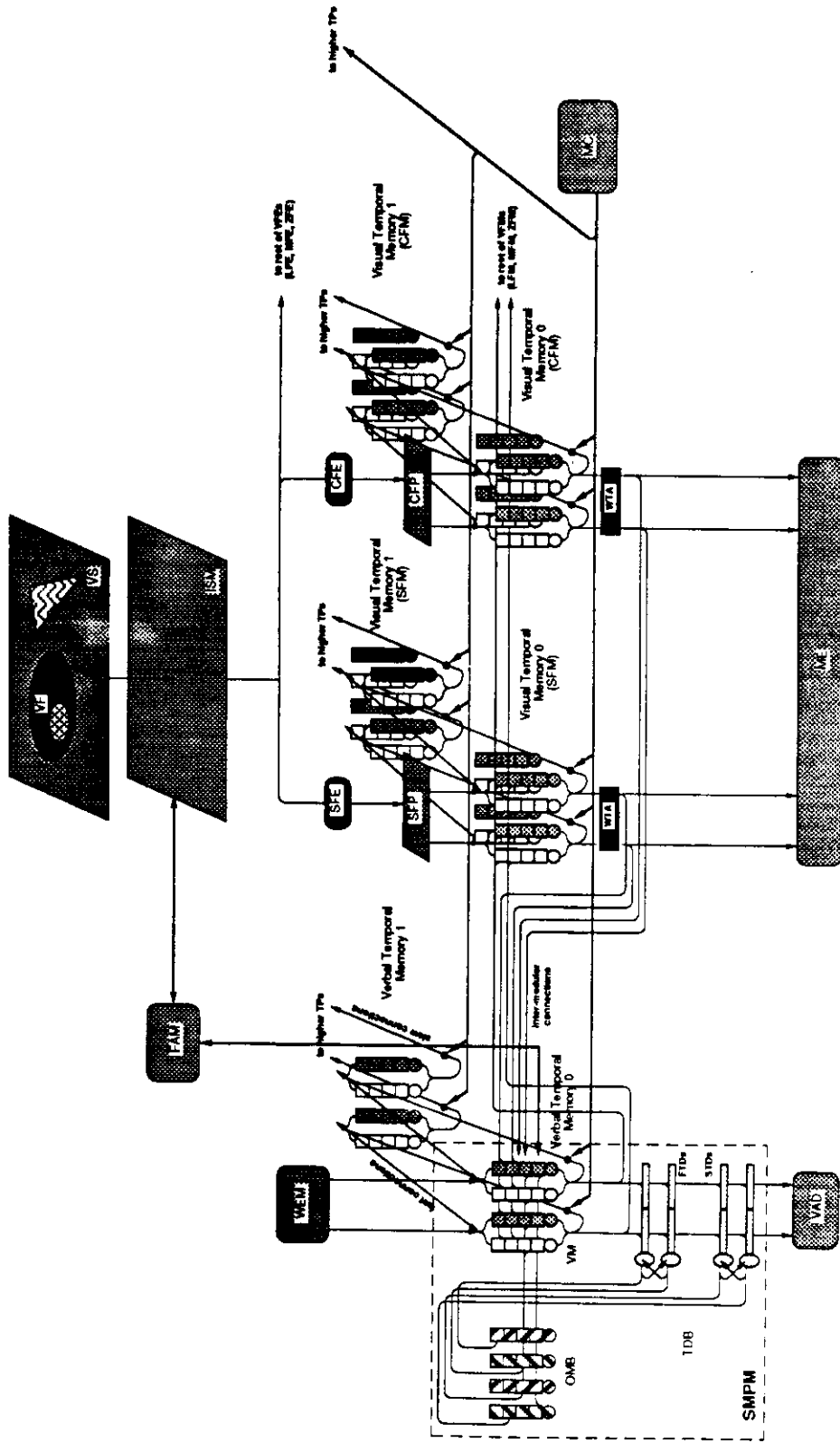
Schematic drawing of the connectivity pattern between the verbal memory module and any of the visual features memory modules. The inter-modular connectivity is achieved by reciprocal inter-modular output-to-input connections. Each predictron in a given module makes non-modifiable synapses with weights 1 to the dendritic branches of all predictrons in the other memory module. All synapses made by an inter-modular fiber are at the same "level" along the dendrites. This level is chosen at random for each fiber. Fibers crossing each other at straight angles are not connected.

10.2.3 DETE's complete memory architecture

A schematic view of DETE's memory architecture is shown in Figure 10.8. For the purpose of simplicity, only two out of the five Visual Feature Memories are shown, and also only two out of the eight Temporal Memory Planes are shown. Each of the Visual Feature Memories is reduced to an array of 2 x 2 pairs of predictrons (one STM and one LTM per pair). In the Verbal Memory only two out of the 64 pairs of predictrons are shown. The connectivity patterns are shown schematically.

Figure 10.8: Detailed view of DETE's memory organization

Down-scaled view of DETE's memory architecture. The vertical thermometer-shaped icons represent predictrons (4 dendritic compartments per predictron are shown as small squares stacked in a column; the circles underneath represent the somas). The STM predictrons are shaded whereas the LTM are non-shaded. The horizontal thermometer-shaped icons (bottom left) represent Transition Detectors (TDs). The connectivity patterns are shown schematically. Wires crossing each other at straight angles do not make contacts. The abbreviations refer to: VF(EYE) -- Visual Field; VS -- Visual Screen; FAM -- Focus of Attention Master; ISM -- Input Segmentation Mechanism; WEM -- Word Encoding Mechanism; SFE(P,M) -- Shape Feature Extractor (Plane, Memory); CFE(P,M) -- Color Feature Extractor (Plane, Memory); LFE(P,M) -- Location Feature Extractor (Plane, Memory); MFE(P,M) -- Motion Feature Extractor (Plane, Memory); ESE -- Eye State Extractor; ELM -- Eye Location Memory; EDM -- Eye Diameter Memory; FSE -- Finger State Extractor; FLM -- Finger Location Memory; FMM -- Finger Motion Memory; WTA -- Winner Take All mechanism; MSPM -- Morphologic/Syntactic Procedural Memory; VM -- Verbal Memory; TDB -- Transition Detectors Bank; VAD -- Verbal Activity Decoder; ME -- Mind's Eye; MC -- Moment Clock.



PART III

Performance and Evaluation

Part III of this thesis is concerned with the evaluation of DETE's performance. In a series of experiments, in which the task complexity is gradually increased, we demonstrate how DETE can learn meanings of words for objects and their features like "ball", "triangle", "red", "small"; also, words describing motion states such as "stands", "moves", "moves diagonally", "bounces", etc. Further, we show that DETE is capable of dynamically building generalizations from prior visual and verbal experiences by using an oscillation based role-binding mechanism. DETE's linguistic skills are probed further in the domain of syntax acquisition (e.g., word order). Finally, part III demonstrates how DETE, due to its unique architecture which contains explicit representation of time, can acquire meanings of verbs in several verb tenses (present, past, future, and their perfect forms).

11 INCREMENTAL LANGUAGE ACQUISITION

In a series of experiments this chapter demonstrates how DETE gradually acquires some basic language skills. In an increasing order of complexity these language skills are: (1) formation of simple concepts -- learning words for objects and their features, and learning words for events, (2) generalization within and between the visual and verbal modalities, (3) question answering, (4) learning more complex concepts for spatial and motion relations between objects, and (5) learning about temporal relations between events. Each experiment consists of a training (learning) phase during which DETE makes associations between the visual and verbal inputs, and a testing (performance) phase during which the quality of the associations made is tested. In some experiments the two phases are separate in time, i.e. testing is done after a whole block of learning trials is presented, while in other experiments the training and testing are interleaved. In a third group of experiments the testing is actually a part of the training.

11.1 Experimental protocol

The need of a well-designed learning protocol arises since DETE is a complex neural/procedural system. The more complex a system is, the more its degrees of freedom grow. Therefore, it is imperative that appropriate constraints are imposed on every single element of the system as well as on the inputs and expected behaviors. All experimental protocols reported in this chapter were designed using a common strategy which is characterized by a gradual increase of task complexity. In the beginning, DETE is taught the basic linguistic elements such as the meanings of single words for individual instances of objects or events and is trained to answer simple questions. Later, DETE learns to extend known words to novel instances. The next step is the learning of short phrases and sentences. It is important to point out that for the learning of more complex tasks DETE is required to have already mastered some simpler tasks. For instance, in order to be able to answer questions DETE had to first learn the names of individual objects. However, not all tasks require that DETE retains all the knowledge accumulated during prior experiences with simpler tasks. In such cases, to reduce the memory load, the number of training trials, (and the computational expenses necessary for the maintenance of a fully integrated system), I resorted to teaching DETE only the necessary prerequisites for the performance of the particular complex tasks. This strategy allows for obtaining a better understanding of the performance of the task at hand, but limits the possibility for exploration of the influences of more comprehensive prior knowledge on the performance of the particular task. In other words, the interference in the learning of different linguistic phenomena has not yet been tested. This makes the comparison of DETE's performance with the performance of human infants (which are exposed to the whole spectrum of linguistic phenomena and acquire the various linguistic skills on their own pace) somewhat difficult.

The designs of the experiments in this chapter are visualized in a number of figures and the results are shown in several tables. In general, the numerical values listed in the tables represent the number of learning trials it took for DETE to produce the first correct response for any particular task and also the number of trials necessary to achieve 90% correct response. This second measure is representative of DETE's behavior since the correctness of the responses was not sustained

through successive trials but in general improved with continuous training. This learning behavior was often a result of the choice of the particular training set. More specifically, depending on the ordering of pairs in the training set DETE experienced periods of non-monotonicity of learning. For instance, after it has learned the correct meaning of a particular word, DETE might unlearn it and associate another meaning with the same word. In general, however, with a sufficiently long training sequence DETE learns the correct meaning. The basic mechanism which allows DETE to accomplish this task is the built-in forgetting mechanism of the KATAMIC memory. This ability is a direct consequence of the continuous redistribution of *p-ltm* and *n-ltm* resources (see formulas 8.9a and 8.9b in section 8.2).

11.2 Learning single words

11.2.1 Learning words for objects

DETE's training started on a simple task of learning the meanings of words that name individual objects. For this purpose DETE was exposed to a number of pairs of the type (WORD:*x*, PICTURE:*x*). Here, *x* stands for any object, which has a verbal and visual representation denoted respectively with the prefixes WORD: (W:) and PICTURE: (P:). The pictorial representation of the object *x* (P:*x*) stands for a combination of the following visual features: color (C:*x*), shape (S:*x*), size (Z:*x*), location (L:*x*), and motion (M:*x,x*). Motion has two components, direction and velocity. The task is to prove that after learning a sufficient number of such pairs DETE can "imagine", for instance, red color (C:red) in response to the word "red" (W:red) and correspondingly it can utter (generate) W:red in response to a visual input of C:red (e.g., when the whole Visual Field is red). To learn this task, for a moment -- i.e. 300 B-cycles (see Table 7.1), DETE is simultaneously presented with a verbal input (e.g., W:ball) and a visual input (e.g., P:ball). Note that the visual input (P:ball) has five components, i.e. P:ball = [C:*, S:circle, Z:*, L:*, M:*,*]. The symbol "*" stands for any possible value of a given feature.

As described in section 3.2, each element of this set of components is represented as a small group of oscillating neurons in the corresponding visual feature plane. During the process of learning DETE captures the invariant features of the objects. Capturing feature invariance is possible because the training data set is constructed such that the characteristic features of each object are kept constant during the pairings, while at the same time the rest of the features are allowed to vary. For instance, the most characteristic visual feature of "ball" (W:ball) is its circular shape (i.e. P:ball = [C:*, S:circle, Z:*, L:*, M:*,*]), whereas for the word "red" (W:red) it is the color (C:red) that is most characteristic.

The verbal representation of each word is associated with a specific visual representation and this association constitutes the meaning that DETE has attached to the particular word (Figure 11.1). The learning of this kind of "meaning" (i.e. the grounding of a symbol) is achieved by forming a strong association between the word that names a set of objects (e.g., ball) and the most invariant of the visual features of the objects. For instance, W:ball is associated the strongest with the shape S:circle, while weaker traces are formed between the W:ball and colors like C:red, C:green, C:yellow, etc., or sizes as S:small, S:large, etc., and similarly for the rest of the features. Later, during testing, the presentation of a verbal input by itself like W:ball associatively activates (i.e. brings to the WM) the visual representations of S:circle, together with the statistically most frequent values of the other visual features (e.g., C:red, L:center, etc.).

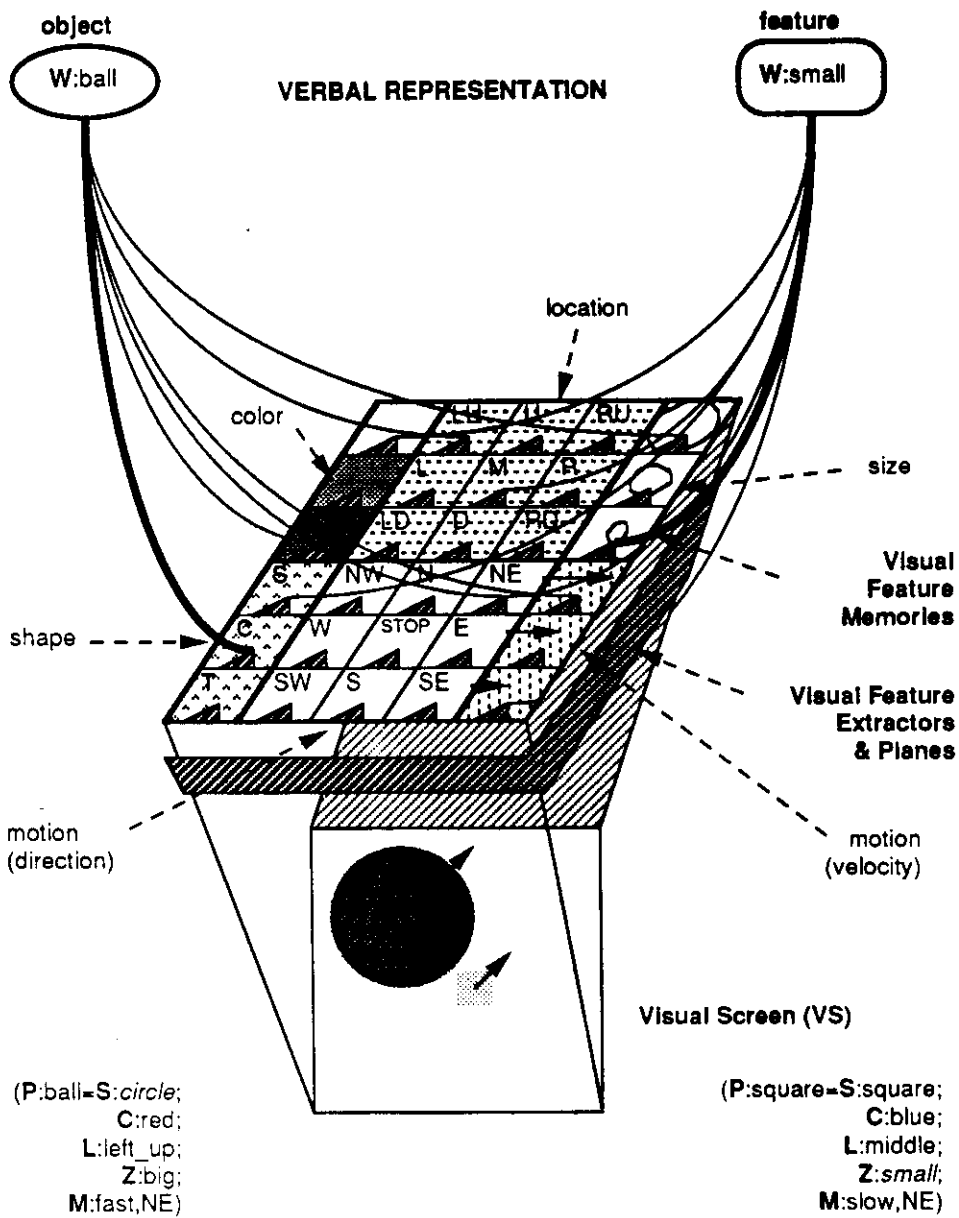


Figure 11.1: Learning words for objects & features

Schematic representations (abstracted and highly simplified) of the meanings of two words **W:ball** (an object) and **W:small** (a feature) are shown as ovals on the top. (The actual verbal representations are formed by the traces of the gra-phonemic sequences of the words which are stored in the verbal memory). The visual representations of two objects that appear in the Visual Field are shown in parentheses. The word **W:ball** names a large red ball located left above the center and moving fast in north-eastern direction (**P:ball** = **S:circle**; **C:red**; **L:up_left**; **Z:big**; **M:north_east**, fast). The word **W:small** names a small blue square located in the center moving slowly north_east (**P:square** = **S:square**; **C:blue**; **L:middle**; **Z:small**; **M:north_east**, slow). The five visual feature planes (**S,C,L,Z,M**) are shown schematically in a composit feature plane in the middle. The associations between the visual and verbal representations are shown as links between them. The meanings of the two words are represented by the thickness of the links that their verbal representations form with the individual features in the VFP. The word "ball" has its strongest link to the *circular* feature in the SFP, whereas the word "small" has its strongest link to the *small* feature in the ZFP.

In the first experiment DE TE learns the meanings of the words "circle", "square" and "triangle". Notice that in the English language these words commonly refer to shapes but also whole objects with circular, square or triangular shape are often named by these words. Also, objects named by these words can have different colors, locations, sizes, and motions. With this in mind, the experiment was designed in the following way. Pairs of visual and verbal inputs of the type (**[W:x]**, **[P:x]**), and specifically, (**[W:circle]**, **[C:x, S:circle, Z:x, L:x, M:x,x]**), (**[W:square]**, **[C:x, S:square, Z:x, L:x, M:x,x]**), and (**[W:triangle]**, **[C:x, S:triangle, Z:x, L:x, M:x,x]**) were presented in a sequence -- one pair per *moment*. DE TE made associations between the visual and verbal representations at each presentation. After the presentation of each pair the learning was disabled (i.e. no update of the *lrm* was done) and two tests were run: 1) visual-to-verbal test, 2) verbal-to-visual test. In the verbal-to-visual test DE TE was given a verbal input (e.g., **W:circle**) and the activity generated in the visual bank of the Long-Term declarative Memory was examined. The visual response was considered to be correct if, as a result of the verbal input, sustained oscillations were induced in at least one neuron located in the proper area (i.e. the area that represents circles) of the shape bank of the visual memory. In practice, however, we commonly observe several of the neurons exhibiting increased activation in the predefined area. In the visual-to-verbal test, a novel instance of a circle, a square, or a triangle was presented as visual input and the activity generated in the verbal bank was monitored. The response was considered correct when all gra-phonemes were generated in the correct order without intervening noise. Schematically, this experiment can be described as follows:

TRAINING: (**[W:circle|square|triangle]**, **[C:x, S:circle|square|triangle, Z:x, L:x, M:x,x]**)

TESTING (verbal -> visual): (**[W:circle|square|triangle]**, **[S: ?]**)

TESTING (visual -> verbal): (**[W: ?]**, **[C:x, S:circle|square|triangle, Z:x, L:x, M:x,x]**)

The results of the experiment are summarized in Table 11.1. As the table demonstrates, the learning of both tasks is quite fast but the learning of the visual-to-verbal task is somewhat slower than the learning of the verbal-to-visual task.

| Verbal input | Visual input | | | | | 1st(100%) correct after trial # | |
|--------------|--------------|-------|------|-------|--------|---------------------------------|----------|
| | Color | Shape | Size | Loctn | Motion | ver->vis | vis->ver |
| W:circle | * | ○ | * | * | * | 4(169) | 6(187) |
| W:square | * | □ | * | * | * | 5(183) | 8(192) |
| W:triangle | * | △ | * | * | * | 5(181) | 9(211) |

Table 11.1: Results of learning **circle | square | triangle**

Using the same experimental design DETE was trained to “understand” in separate experiments (starting from a naive system) a whole list of single words including:

- 1) *words for color*: white, red, orange, yellow, green, blue, and purple
- 2) *words for size*: small, medium, large
- 3) *words for location wrt center of VF*:
 - a) above, bellow
 - b) left, right
 - c) in_center, near, far
- 4) *words for motion in straight line and constant speed*:
 - a) still, slow, fast
 - b) north, east, west, south, north-east, north-west, south-east, south-west

In each of these experiments, the corresponding feature plane is carved up into mutually exclusive regions and each region is associated with only one word. In other words, in these simple sets of experiments I purposefully avoided teaching DETE about multiple verbal carvings of the same visual feature plane (categorization). How DETE learns different categorizations is discussed in the following sections.

An important characteristic of these sets of experiments is that all of the words that DETE learned map to features which are stationary in time. In other words, the location of the neural assembly that represents a particular object feature does not change in time. Even in the case of words for motion, the representation of motion is also static in the MFP (as long as the motion is in a straight line and with constant speed). However, when an object moves, its location changes in the LFP. For this reason, I will discuss the learning of the meaning of the word “moves” separately in the next section, where I focus on events.

A consistent observation on DETE’s performance for each of the experiments described above was that the learning of a particular word is fast. More significantly, the actual learning speed (number of trials) depends on how fine the carving of the feature plane is (i.e. into how many regions a particular feature plane is divided). In general, the larger an area of the FP (which is labeled by a particular word), the longer it takes for the word to be learned. However, the total number of trials to learn a complete carving of any of the five feature planes was approximately the same (remember that each feature plane is a square array with dimensions 16 x 16).

11.1.2 Learning words for events

The first series of experiments considered situations when the visual features do not vary in time (stationary events). In the following experiments DETE is taught the meaning of words for events.

By an **EVENT** happening to an object I mean a change in one or more of the visual features (that represent the object) from one visual frame to another (remember that a visual frame is 5 B-cycles long). Some examples of events are:

moves (change of location)

accelerates (change of motion speed)

turns (change of motion direction)

bounces (change of the direction of motion while in contact with another object)

shrinks (change in size relative to its previous size)

transforms (change in shape relative to its previous shape)

transcolors (change in color relative to its previous color)

disappears (change -- loss of all visual features)

In reality, most of the features of an object, such as shape or size, do not change with time. Actually, the time-invariant features of an object are those which we usually associate with the object. Other features, however, often change with time. For instance, if the object moves, its location changes. This change of location can be monotonic (e.g., in the case of linear motion with a constant speed), or non-monotonic (e.g., during jumping or bouncing).

The ability of DETE to learn the meaning of words that name events is illustrated in the following three experiments. First, DETE learns the meaning of the words "moves" and "stands". These words stand for events whose duration could be arbitrarily long (i.e. "enduring-events"). Next, DETE learns words that describe different types of motion with respect to the direction of motion (e.g., *moves_horizontally*, *moves_vertically*, *moves_diagonally*). Finally, it learns the meaning of the word "bounces". This is an event which occurs in a very short time -- the actual change of direction of motion happens within one B-cycle (i.e. a "momentary-event")

• Learning the meaning of "moves" and "stands"

To learn the meanings of the words "moves" (**W:moves**) and "stands" (**W:stands**) DETE uses the visual representation of object motion in the Motion Feature Plane (MFP). Details of DETE's motion representation are given in section 3.2.5. In accordance with this representation, an object is moving if the set of neurons that represent this object in the MFP is located anywhere outside of the central area of the MFP (Figure 11.2). The central area of the MFP represents stationary objects. The word **W:moves** does not specify a direction or speed of motion. Therefore, a moving object can be represented anywhere in the motion segment of the MFP. The actual location of the representation on the MFP depends on its speed and direction of motion.

In this experiment DETE is presented with multiple pairs containing the word **W:moves** and a sequence of visual frames showing a moving **P:object** (**M:*,(≠0)** -- i.e. the direction of motion is arbitrary but the speed is always non-zero). All other visual features are kept constant within each presentation but vary across pairs. Within the same experiment, the learning of the word "moves" is alternated with the learning of the word "stands". Notice that "moves" and "stands" are mutually exclusive in the sense that together their visual representations cover the whole MFP (Figure 11.2A). The learning of **W:stands** was achieved by presenting multiple pairs containing **W:stands** and **P:object** where **M:*,(=0)** (i.e. the direction of motion is irrelevant and the speed is zero). All other visual features were kept constant within each presentation but varied at random across pairs.

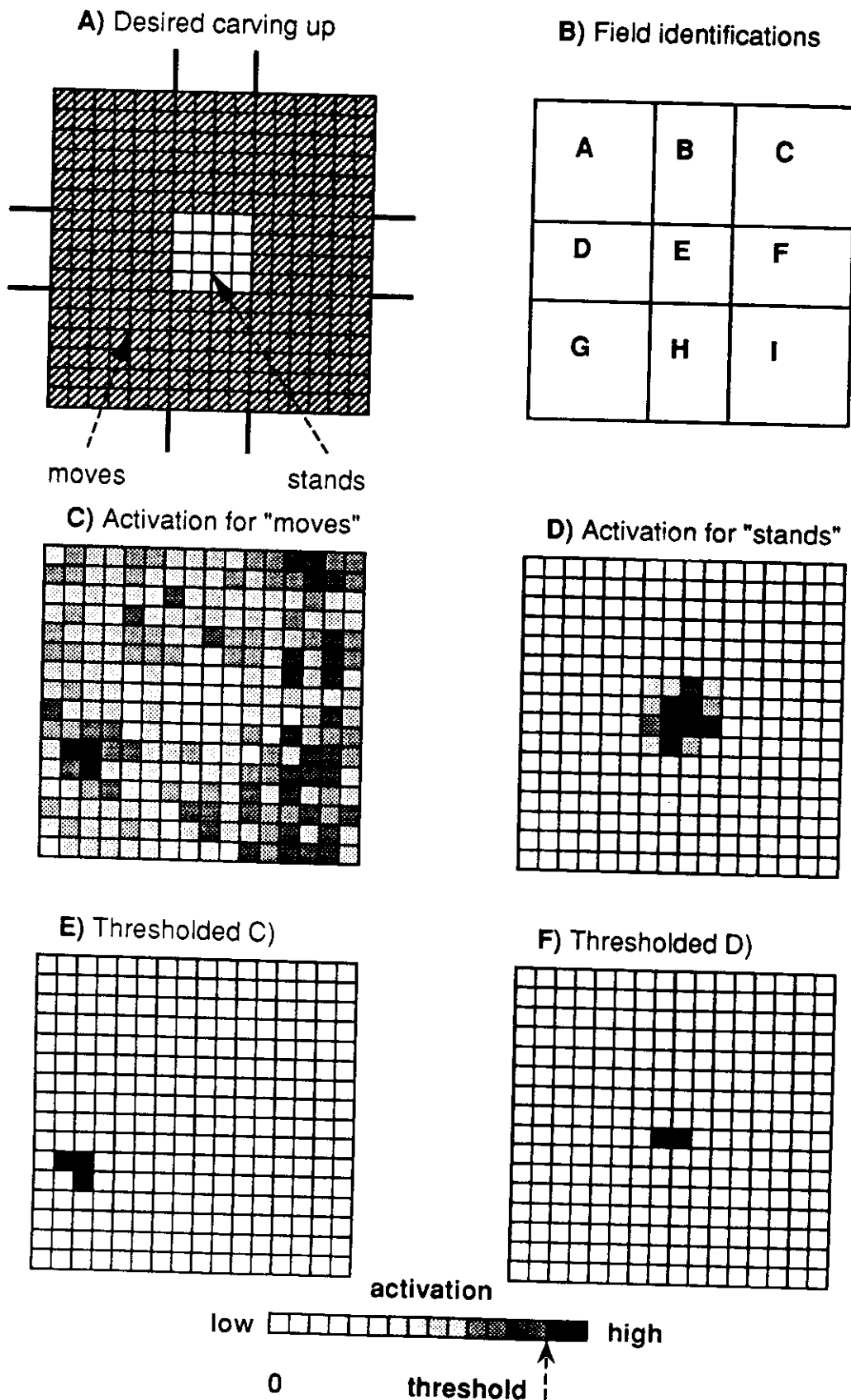


Figure 11.2: Learning the meanings of "moves" & "stands"

A) Theoretical carving up of the MFP. The shaded area around the center of the MFP is the segment to which the word "moves" is mapped. The white area in the center is the MFP segment to which the word "stands" is mapped. B) Labels of various segments in the MFP. C&D) Activation of the predictrons in the "moving" and "motionless" segments of the MFM in response to corresponding verbal inputs. Only positive activation is shown - see scale on the bottom. E&F) WTA thresholded activation. The threshold is such that only the highest activity (black) is passed through.

As a result of this training procedure DETE learns to generate (in response to the **W:moves** or **W:stands** presented without a corresponding visual input) activity in the motion or the motionless segment of the MFM (Figure 11.2C&D). As this figure demonstrates, the activation (see Formula 8.11) generated in the MFM in response to the verbal input itself is well localized in the corresponding areas -- in the motion segment when the verbal input is **W:moves**, and in the motionless segment in response to **W:stands**. However, the activation value is not the same for all predictrons in the segment. The pattern of activation generated in each particular instance depends on the training set to which DETE has been exposed. Additionally, the strongest activation is observed in those predictrons to which the most recent visual stimulation (extracted motion) has been mapped (i.e. priming effect) or to which the majority of the previous inputs have been mapped (i.e. frequency effect). The Winner Take All thresholded outputs of the MFM show which of the predictrons ultimately pass their activity to the rest of the memory modules in DETE (Figures 11.2E&F). The MF Memory neurons activated in response to **W:moves** oscillate in-phase. In other words, they represent one single moving object. Notice that if the oscillations of neurons were out of phase, such activity would represent multiple moving objects.

The results of the experiment are summarized in Table 11.2. They suggest that (1) DETE takes longer to learn **W:moves** than **W:stands**, and (2) it takes longer to learn the visual-to-verbal transformation than the verbal-to-visual transformation.

| Verbal input | Visual input | | | | | 1st(100%) correct after trial# | |
|-----------------|--------------|-------|------|------|---------|--------------------------------|----------|
| | Color | Shape | Size | Locn | MotnV,D | ver->vis | vis->ver |
| W:moves | * | * | * | * | ≠ 0,* | 48(1350) | 52(1420) |
| W:stands | * | * | * | * | = 0,* | 5(40) | 22(42) |

Table 11.2: Results of learning **W:moves** & **W:stands**

The observation that DETE takes longer to learn **W:moves** than **W:stands** can be attributed to the fact that the segment of the MFP where the representation of moving objects can be generated, encompasses a much larger area of the MFP (240 neurons) than the segment for the representation of objects that are standing-still (16 neurons). The ratio of the sizes of these segments is 15:1 and it was chosen somewhat arbitrarily when the Motion Feature Extractor was designed. However, the ratio of the number of learning trials it took, until the first correct responses were produced for the two words (**W:moves** than **W:stands**), is smaller than 15. In other words, the relation between the size of a particular segment of a plane (e.g., the "motion" segment of MFP) and the speed of learning is not linear in this example. If this relation was linear, one would expect that it will take about 15 times more trials to learn **W:moves** than **W:stands**. However, the time was only about 8 times longer. Notice that for all moving objects their location changed in time while the representation of the standing still objects was stationary in the LFP (but at different locations) giving another feature dimension along which "moves" and "stands" differ. Another observation is

that the achievement of 100% correct responses requires a significant number of trials. This phenomenon will be discussed later.

The table also shows that DETE takes longer to learn the visual-to-verbal transformation than the verbal-to-visual transformation. This is due to the different representations chosen for the visual and verbal features. The visual features are represented by simple oscillations (which evidently is easier to be learned by the KATAMIC model, whereas the verbal representation consists of complex pattern-sequences (sequences of gra-phonemes)).

• **Language carving (categorization) of motions**

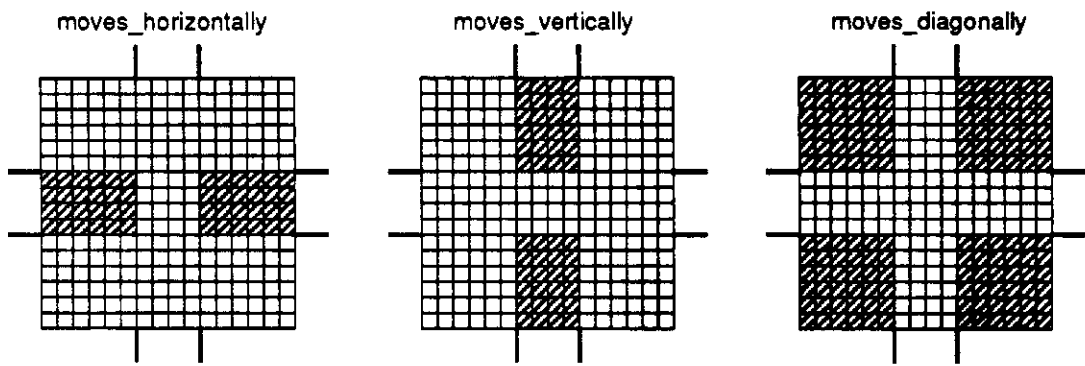
The relation of language to the perceptual categorization of various types of motion can be demonstrated in the following experiment. Starting from a naive state, DETE was taught the meanings of the phrases “moves horizontally”, “moves vertically”, and “moves diagonally”. As can be seen in Figure 11.3A, there is a direct and mutually exclusive mapping of the meaning of each of these phrases to specific segments of the Motion Feature Plane. The training protocol for this experiment was set up similarly to the protocol used in the previous experiment where DETE learned the meanings of the words “moves” and “stands”. In other words, the trials of horizontal motion were randomly mixed with trials of vertical motion and trials of diagonal motion. In each trial a single visual/verbal pair was shown to DETE. Testing was done by giving DETE only a verbal or a visual input. Figure 11.3B shows the response of the MFM when DETE hears the phrases “moves horizontally”, “moves vertically” or “moves diagonally” (after these phrases have been learned). These activations can be interpreted as if DETE is imagining objects moving in the corresponding directions. On the other hand, if DETE looks at an object that is moving horizontally (left or right) which will be represented by some localized activity in either one of the shaded segments in the left-hand drawing in Figure 11.3A, then this activation, if combined with an appropriate verbal question (e.g., “How does it move?”) can elicit by association activity in the verbal memory. When this verbal activity is decoded by the Verbal Activity Decoder (i.e. converted from gra-phonemic to graphemic -- alphabetic representation) it sounds as “moves horizontally”.

DETE's ability to learn such mappings from the visual representation of motion to verbal representation is based on the KATAMIC memory's ability to successfully associate a given sequence (e.g., the verbal representation of a word) running in the Verbal Memory with several different sequences representing various motions in different parts of the motion MFM segment .

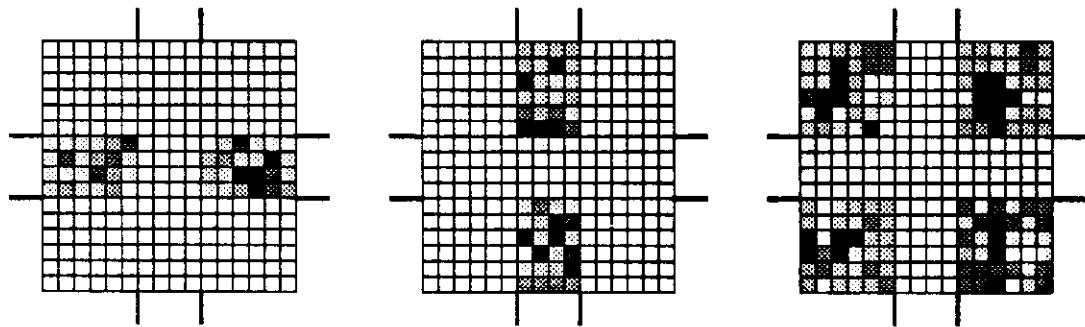
The results of this experiment are summarized in Table 11.3. As might be expected from the symmetry of the expressions “moves horizontally” and “moves vertically”, the number of trials it took DETE to learn each of them is about the same. The learning of “moves diagonally” was about two times slower, which can be explained by the larger area of the MFP which represents this type of motion.

| Verbal input | Visual input Motion direction | 1st (100%) correct after trial # | |
|----------------------|----------------------------------|----------------------------------|----------|
| | | ver->vis | vis->ver |
| W:moves_horizontally | ← or → | 17(170) | 19(200) |
| W:moves_vertically | ↓ or ↑ | 18(190) | 20(210) |
| W:moves_diagonally | ↗ or ↘ or ↙ or ↖ | 33(700) | 38(780) |

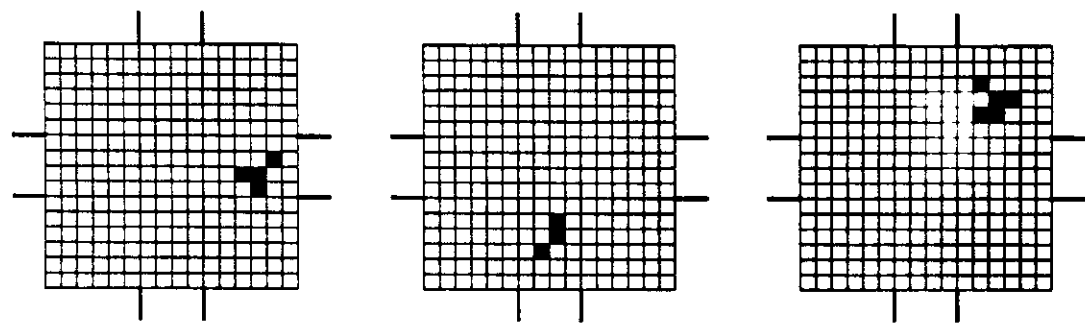
Table 11.3: Results of learning W:moves_horizontally | vertically | diagonally



A) Desired carvings of the Motion Feature Plane



B) Activations of the Motion Feature Memory



C) WTA thresholded outputs from the Motion Feature Memory

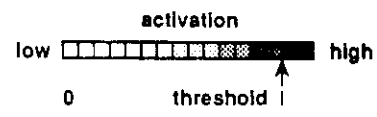


Figure 11.3: Learning about different movements

A) Desired representation of the meaning of the three types of motion (horizontal, vertical and diagonal) in the MFP: **B)** Activity generated in the MFP in response to the verbal input of the phrases without corresponding visual input. **C)** WTA thresholded activity patterns.

The Color, Shape, size and Location features of the objects used in this experiment were selected at random for each individual learning instance. The speed of the objects was also selected at random but in all cases it was non-zero. The directions of motion were represented by activity in the following areas of the MFP: (1) D or F for horizontal motion; (2) B or H for vertical motion; (3) A or C or G or I for diagonal motion (see Figures 11.2 & 11.3).

• **Learning the meaning of “bounces”**

DETE learns the meaning of the word **W:bounces** by watching objects bouncing at the walls of the Visual Screen. The objects used in this experiment moved along linear trajectories with constant speeds. The bounces were elastic.

An event of bouncing is represented in the Motion Feature Plane by an instantaneous change in the direction of motion (between two consecutive B-cycles). It is important to mention that change in direction, (which is independent of the speed and the location of the object immediately before and after the event), is what DETE associates with **W:bounces**. Actually, what DETE is learning about the word “bounces” is the temporal sequence of invariant features including: (1) near wall and moving towards the wall, (2) stationary for 1 B-cycle at the wall, (3) near the wall and moving away from the wall.

To learn the meaning of **W:bounces**, the following experiment was performed. A set of 150 pairs containing: 1) The same verbal component: **W:bounces**, and 2) Different visual components -- various objects (in terms of shapes, sizes, colors, speeds and directions of motion) were allowed to bounce at random locations off any of the four walls of the Visual Screen. Each pair was input during one *moment* (300 B-cycles) and the bounces were at random times during the *moment*. The experiment was composed of interleaved learning and testing trials. During the learning trials DETE was allowed to learn the visual and verbal associations (i.e. the update of the LTM was enabled). As in previous experiments, during the testing trials the learning was disabled (i.e. no new associations were learned). Two separate responses were tested: (1) Verbal-to-visual: In this case only the verbal input was given. To confirm whether or not DETE has learned the meaning of “bounces”, the activity in the motion segment (see Figure 11.3) of the MFP was monitored. When the verbal input associatively induces localized activity (oscillations) in the MFP and when the location of this activity flips over an axis of symmetry in the MFP (such behavior represents an event of bouncing within the *moment* when the verbal input is presented), then the interpretation of the response is that DETE has learned the meaning of the word “bounces” (Figure 11.4). (2) Visual-to-verbal: A novel instance of an object bouncing off the wall was presented and the output of the verbal memory was monitored to see if the visual representation of bouncing will trigger the generation of the word **W:bounces**.

The experimental protocol is summarized below. The notation “?” means that we are monitoring the activation in the particular Feature Plane. The notation “-><-” means a change in direction.

| | |
|-----------------------------|---|
| TRAINING: | ([W:bounces], [C:x, S:x, Z:x, L:x, M:x;-><-]) |
| TESTING (verbal -> visual): | ([W:bounces], [M: ?,?]) |
| TESTING (visual -> verbal): | ([W: ?], [C:x, S:x, Z:x, L:x, M:x;-><-]) |

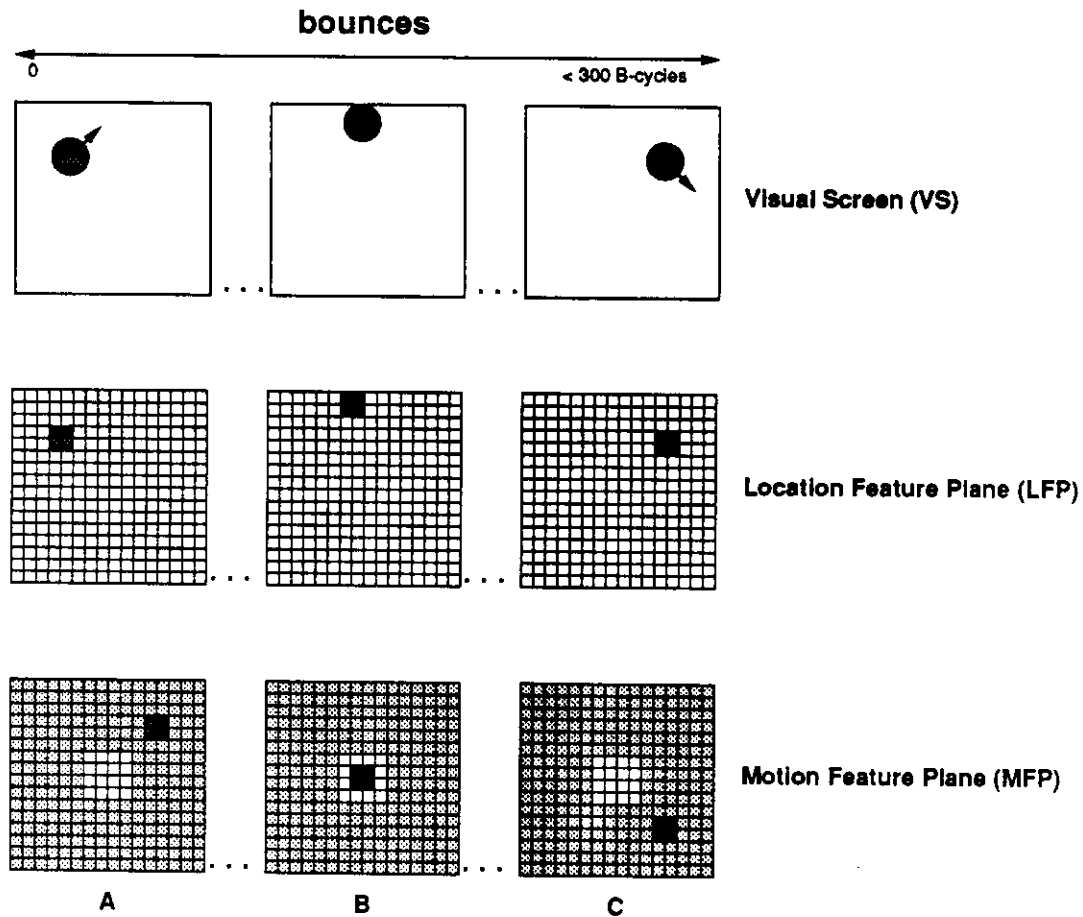


Figure 11.4: Learning the meaning of “bounces”

The verbal input, the visual input, and three states of the activation within the LFP and the MFP are shown to illustrate how the word “bounces” is represented and learned in DETE. **W**:bounces is input at random times during the same *moment* (300 B-cycles) when in the visual input DETE sees an object with an arbitrary shape, size, color, location and motion bouncing off one of the four walls of the visual screen. (A) Before the actual event, the representation of the object’s location in the LFP changes gradually from one visual frame to another (shown as an arrow in the figure). During the same interval the representation of the object’s motion is stationary. The representations of the rest of the features are also stationary since color, shape, and size are kept constant during this experiment. (B) When the object touches the wall it becomes stationary for one B-cycle. Therefore, the representation of its motion is in the “still” segment of the MFP since motion is computed as the difference of locations between two consecutive visual frames. (C) After the event has happened the representation of motion is again stationary within the MFP because the object is moving linearly with a constant speed. However, it has changed its position. In any instance of bouncing, this change is over an axis of symmetry which is parallel to the wall that the object bounces off (the bounce is elastic).

The results are presented in Table 11.4. DETE succeeded in “imagining” a bounce in response to **W**:bounces after 72 learning trails. However, it took 117 presentations of visual/verbal pairs for

DETE to generate the proper verbal response when it was presented with a new instance of bouncing. These results demonstrate that it is easier for DETE to learn to “imagine” things in response to verbal input, than to learn to “articulate” what it sees. The number of learning trials is also larger than in previous experiments which can be explained with the increased degree of freedom in the visual part of the data set (i.e. here we have left not only the size, location, shape and motion to vary as in the first experiment but we also vary the color).

| Verbal input | Visual input | | | | | 1(100%) correct after trial # | | |
|-----------------|--------------|-------|------|------|--------|-------------------------------|----------|----------|
| | Color | Shape | Size | Locn | Motn | V.D | ver->vis | vis->ver |
| W:bounces | * | * | * | * | *,-><- | | 72(400) | 117(512) |

Table 11.4: Results of learning W:bounces

11.3 Generalization

The ability of an information processing system to handle reasonably well inputs which it has never seen before is commonly known as generalization. In DETE we can look for a generalization ability within each of the input modalities: visual and verbal. We can test the generalization ability for a novel input within a given modality by monitoring the responses which the input produces in the other modality.

11.3.1 Verbal generalization

Verbal generalization is tested in the following manner. During the training phase DETE is presented with a set of input pairs containing corresponding propositions (i.e. visual scene and verbal description). For instance, the sentence “Red ball in the center.” is paired with a picture of a red ball which is in the center of the Visual Screen. Each pair is presented at least once.

A test of the verbal generalization ability is done when a novel sentence is presented, i.e. one that was not used in the training set. A successful generalization occurs when DETE generates a correct visual representation (image) in its “mind’s eye” -- the set of 5 activity patterns generated by the Visual Feature Memories. For instance, a presentation of the novel noun phrase “big triangle” leads to the construction of the corresponding image in the “mind’s eye” (Figure 11.5). Note that DETE must have already learned the meanings of the words W:big and W:triangle. The construction of the mental image is incremental. The first word, “big”, elicits activity in the segment of the ZFP where the size “big” is represented. Notice that this is a fairly large area in the ZFP (see Figure 3.4). In other words, this verbal-to-visual mapping is fuzzy. Exactly where in this segment the activity is induced depends on two factors: (1) the set of prior experiences, i.e. the content of the training set used to teach DETE the word “big”; (2) the most recent experience involving the word “big”, i.e. priming effects. The second word, “triangle”, elicits activity in the SFP in the area where triangles are represented. The activities elicited in the ZFP and SFP are in phase. This is due to the fact that the Focus of Attention Master (FAM) is designed so that if there is not a visual input, the first word of any verbal input is synchronized with the Temporal Attention Window (TAW). In other words, the TAW opening coincides with the first B-cycle of the first gra-phoneme. Also, since the duration of each gra-phoneme is the same as the duration of the pause between words and the same as the duration of the FAM oscillation period (5 B-cycles), the phase locking is automatically maintained for successive words (as long as nothing new appears in the VF and no saccades are done to other objects in the VF). Notice also that the verbal input “big triangle” does not elicit activation in the rest of the feature planes. In other words, DETE does not envision a

concrete triangle (e.g., one that is *big* but also *red*, located *in the center* and *stationary*). This is due to the fact that the visual feature memories were designed such that there is no direct interaction between them (see section 10.2.1). Therefore, due to this design constraint DETE cannot make direct associations between individual visual features. The interactions between the individual visual feature memories are indirect through the verbal memory.

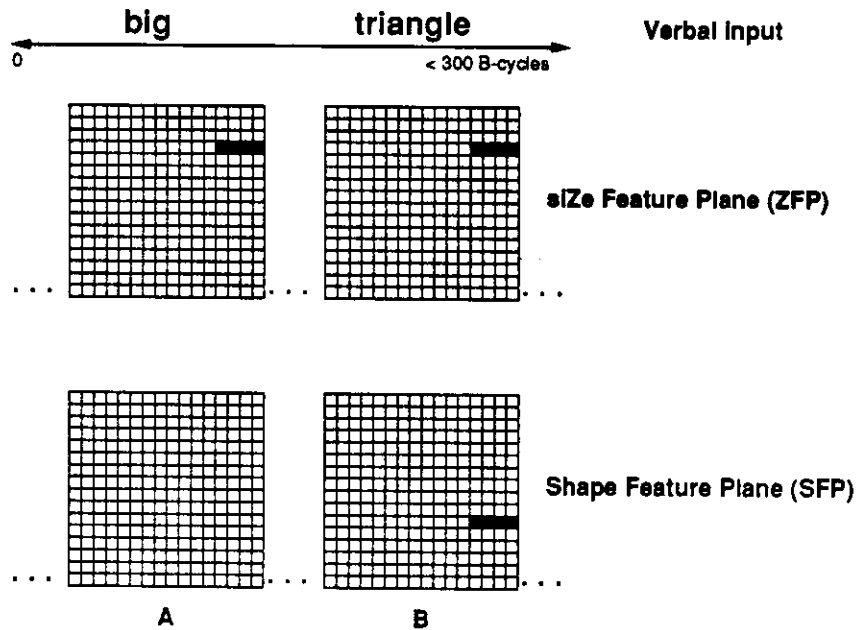


Figure 11.5: Verbal generalization

DETE has learned the words *W:big* and *W:triangle* separately but has never encountered the noun phrase “big triangle” paired with a visual input of a big triangle before. When DETE is presented with the verbal input “big triangle” without looking at one, it constructs incrementally in its “mind’s eye” an image of a big triangle which is represented as phase-locked oscillations in the appropriate areas of the ZFP and the SFP.

Our interpretation of DETE’s behavior in this case is that it has “understood” this novel verbal input. In other words, it has made a proper verbal generalization. The fact that the verbal input has evoked the visual representation of one object (and not just a set of disjoint visual features) is represented by the phase-locking of the oscillations. Notice that this ability is due to the fact that phase locking is pre-set up when there is only verbal input but no visual input.

11.3.2 Visual Generalization

The test of the visual generalization ability is done in a similar fashion. A novel visual scene is presented and DETE is expected to elicit a *proper* verbal response. For instance, if DETE has been taught to produce the verbal outputs “*small red ball*” and “*large blue square*” when it sees the appropriate pictures, then we can test if it can utter “*small blue ball*” at the sight of one *without a prior exposure* to the corresponding visual/verbal pair. There is a substantial difference between this type of generalization and the type of generalization described in the previous section. In the verbal-to-visual transformation DETE converts a sequential input (consecutive words) into a multi-featured

“image” in its “mind’s eye” (i.e. into activation patterns in the appropriate set of visual feature planes). In the case of visual-to-verbal transformation, however, DETE has to generate a single word sequence from a multi-featured image.

Of course, various verbal outputs that correspond to individual visual features or their combinations (a picture) can be generated. For instance, in response to a picture of a blue triangle, DETE can generate either the utterance “*blue triangle*” or “*triangle blue*”. The first utterance is linguistically more acceptable. Therefore, one of the issues that arises in verbal generation is the issue of proper word order. Another issue is that of the utterance content. In the previous example the blue triangle on the picture can also possess a number of other features (e.g., small, in the corner, moving fast up, etc.) and potentially DETE can generate a word sequence which includes words like “small”, “fast”, etc. DETE has to choose which of the visual features to describe (i.e. word selection). In the following section I discuss separately how DETE handles these two issues. First, I consider how DETE selects the content of a specific utterance, then I consider the order of the word in the generated sequence, given that its content is already selected.

Generation of multiple descriptions for the same scene

One and the same visual scene can be described differently, i.e. the visual reality can be carved up verbally according to different classification criteria. The question is on what basis can DETE select the content of its utterances. DETE’s selection of the set of visual features to be described verbally is based on the states of the Winner Take All (WTA) mechanisms which are coupled with each of the individual visual feature memories and placed on the pathways from these memories to the verbal memory. For descriptions of the functions of these mechanisms see section 10.1.5.

Learning word order (simple syntactic rules)

After the content of a possible utterance has been chosen (i.e. the information is available in parallel at the output of the visual memory -- WTA mechanism), the next issue is that of selecting the correct word order. What does a *proper verbal response* mean for us as teachers and judges of DETE’s performance? In the simple cases mentioned above (e.g., sentences of the type NP = ADJ NOUN), it is satisfactory if DETE generates the proper words in the correct order (i.e. words that correctly describe features of the object in an order consistent with FIRLAN’s or SECLAN’s syntax). For our example this translates to DETE generating size words before shape-words. This behavior might be interpreted as if DETE is obeying the syntactic rule: “*To make a noun phrase of an adjective and a noun, first generate the adjective, then the noun*”. The number of possible permutations in the word order increases with the number of words to be uttered. The ability to place words in a proper sequential order during generation is a task that DETE learns through experiences. English speakers have a feeling about the right order of words in verbal descriptions. Such a feeling of correctness is not innate but is learned and is a reflection of the predominant verbal experience (e.g., “Ladies and gentlemen” sounds OK vs “Gentlemen and ladies” sounds weird).

To test DETE’s ability to obey simple syntactic rules, a set of experiments was performed. A number of sentences containing a single noun phrase (NP) were presented to DETE after it had already learned the meanings of the individual words used to construct the sentences. All noun phrases were three words long and contained an adjective for size (e.g., “small”, “medium”, or “large”), an adjective for color (e.g., “red”, “green”, or “blue”), and a noun (e.g., “circle”, “square”, or “triangle”). In other words, the noun phrases had a particular word order: NP = adjective-for-size (adjZ) + adjective-for-color (adjC) + noun. The 27 possible phrases were divided into two groups. Eighteen phrases formed a training set and the remaining 9 formed a testing set.

The training set was presented in sessions containing the 18 phrases in a random order. Testing on the training set was done after each presentation of the complete training set. The training sessions continued until DETE was able to generate each of the sentences in the training set correctly. During testing, the learning was disabled and the visual component of the training set were presented in a new random order while the verbal output generated in response was monitored. After the whole training set was mastered, DETE was tested once on the testing set. During training, matching visual and verbal inputs were paired (presented simultaneously). During testing (both on the training set and on the testing set), only visual inputs were presented and DETE's verbal responses were monitored. Throughout the experiment, the locations and motions of the objects were chosen at random. The results of this experiment are presented in Table 11.5.

| Verbal & Visual input response after trial | 1st correct verbal to visual input | Verbal response |
|--|------------------------------------|-----------------------------------|
| Size Color Shape(Obj) | # on training set | from testing set($\Theta_P=.1$) |
| small red circle | 4 | |
| small red square | | small red square |
| small red triangle | 6 | |
| small green circle | 4 | |
| small green square | 5 | |
| small green triangle | | small green ... |
| small blue circle | | small blue circle |
| small blue square | 3 | |
| small blue triangle | 5 | |
| medium red circle | | ... red circle |
| medium red square | 5 | |
| medium red triangle | 6 | |
| medium green circle | | ... green circle |
| medium green square | 5 | |
| medium green triangle | 5 | |
| medium blue circle | 4 | |
| medium blue square | 6 | |
| medium blue triangle | | ... blue ... |
| large red circle | 3 | |
| large red square | 4 | |
| large red triangle | | large red ... |
| large green circle | 5 | |
| large green square | | large ... square |
| large green triangle | 6 | |
| large blue circle | 6 | |
| large blue square | | large blue square |
| large blue triangle | 5 | |

Table 11.5: Visual to verbal generalization

All 18 visual scenes used in the training set are given in plain font. Θ_P is the activation threshold of the predictrons /WTA mechanisms. The 9 visual scenes used in the testing set are in **bold**.

As the performance table indicates, DETE was able to generate the first correct word order for the phrases used in the training set within about 5 repetitions of each individual phrase (phrases with omissions were not considered correct). It started to give continuously correct responses after about 350 exposures to the entire training set. The fact that different phrases have different lengths in terms of gra-phonemes did not significantly change the number of presentations required for the learning of each phrase. This seems to be due to the fact that all 9 individual words forming the phrases have been already learned by DETE and in the present experiment DETE learned only the possible orderings in which they appear in the utterances (i.e. first, second, or third). This task is handled by the Morphologic/Syntactic Procedural Memory (MSPM) and is not very difficult since the sequences were only three words long.

DETE's performance *on the testing set* is more interesting. As can be seen from the table, in some cases DETE generated only a one-word response, in other cases it generated a 2-word response and only in 3 cases did it generate a complete three-word response. It is important to mention that the WTA generation threshold Θ_P for all visual feature memories was set to 0.1 (a relatively high threshold). This prevented activity from some visual memories reaching the verbal memory bank. With a lower setting of the WTA threshold (e.g., 0.05) all words were generated. Another observation is that in all cases the words appeared *in a correct order* and in the instances where the verbal output was incomplete (i.e. omitted words), there were pauses in the generated sequences in positions where the omitted words should have been generated. The omission of words can be explained by the relation between the chosen threshold and the extent to which the individual words were learned (i.e. the strengths of the traces which they have left in the memories).

11.4 Learning Question/Answer Sequences

A standard task for any Natural Language Processing (NLP) system is question answering. This task is often used to examine the system's understanding ability. The complexity of the question answering task has been studied in detail in (Lehnert, 1978). DETE was trained to answer simple questions such as:

Q1: "What is the color of the small ball?" (while looking at a small ball)

A1: "Red."

Q2: "What is bigger?" (while looking at a small triangle and a large square)

A2: "Square."

Q3: "What moves?" (looking at a stationary circle and a triangle moving to the left)

A3: "Triangle."

All of these questions contain a user-specified feature, e.g., color, size, etc. and a request that DETE returns the value of this feature for the attended object.

To be able to test DETE's ability to learn the meanings of words that stand for more abstract concepts such as color ($W:color$ -- a variable that can take different values like red, blue, green), and shape ($W:shape$ -- a variable that can take values like circle, square, triangle), we need to teach DETE to understand and answer questions. "Understanding" here is meant as the ability to generate appropriate verbal output (and maybe to do the appropriate imagination or motor response) in response to verbal queries. In symbolic terms, such behavior corresponds to the retrieval of slot-

fillers in a slot-filler representation (a.k.a. schema or frame representation). For instance, if DETE sees a small red ball in the left upper corner moving down, and is asked “What color?”, it should be able to generate the verbal response “red” rather than anything else like “ball” or “small”, etc. If it is asked “where” it should respond with “left” (or with a phrase like “up left”).

Such behavior can be learned using the following experimental protocol. During the learning phase DETE gets a verbal input (e.g., **W:what_color**) together with a visual input of an object with its five visual attributes. Then a second verbal input which corresponds to the desired answer is presented (e.g., **W:red**). Multiple sessions of this type cause DETE to learn to make the temporal-spatial association between (**W:what_color & C:red**)(t), and (**W:red**)(t+Δt) (or any other color). Later, during the testing, when prompted by the verbal input **W:what_color**, in the presence of the pictorial representation of the color (**C:red**) DETE produces the right utterance, namely **W:red**.

Table 11.6 shows the results of an experiment in which DETE is taught to answer the questions: “What_color” and “What_shape” while looking at various objects which are either balls or squares and are either blue or red. The sizes, locations, and motions of the objects are chosen at random (represented by “*” in the text and tables) during training and testing.

LEARNING protocol:

Input-1: ([**W1:what_color | what_shape**],[**C:red | blue, S:circle | square, Z:*, L:*, M:*,***])

Input-2: ([**W2:red | blue | circle | square**],[**C:red | blue, S:circle | square, Z:*, L:*, M:*,***])

TESTING protocol:

Input: ([**W1:what_color | what_shape**],[**C:red | blue, S:circle | square, Z:*, L:*, M:*,***])

Output: ([**W2:red | blue | circle | square**],[**C:red | blue, S:circle | square, Z:*, L:*, M:*,***])

| Verbal input | Visual input | | | | | 1st(100%)correct verbal output after trial # | |
|--------------|--------------|-------|------|------|--------|--|-------|
| | Color | Shape | Size | Locn | Motion | | |
| W:What_color | C:red | S:○ | * | * | * | red | 4(83) |
| W:What_shape | C:red | S:○ | * | * | * | circle | 6(89) |
| W:What_color | C:blue | S:○ | * | * | * | blue | 4(74) |
| W:What_shape | C:blue | S:○ | * | * | * | circle | 6(79) |
| W:What_color | C:red | S:□ | * | * | * | red | 4(69) |
| W:What_shape | C:red | S:□ | * | * | * | square | 5(78) |
| W:What_color | C:blue | S:□ | * | * | * | blue | 6(81) |
| W:What_shape | C:blue | S:□ | * | * | * | square | 7(91) |

Table 11.6: Question answering -- slot-value retrieval

As can be seen from this table, on average it took 5 trials before DETE generated the 1st correct response, and about 80 trials on average to start generating continuously correct responses.

DETE’s ability to answer questions of this type can be interpreted as an ability to learn the meaning of words such as “color” (i.e. a notion that encompasses all color values) or shape (i.e. a notion that encompasses all possible shapes). However, notice that DETE has not learned separately the meaning of the word “what”. In a similar way DETE can learn the meaning of the **W:what_location** (**W:where**), **W:what_size**, **W:what_speed** (**W:how_fast**), **W:what_direction**, etc.

11.5 Learning spatial relations between two objects

A variety of words can be used to describe spatial relations between two objects. Spatial relations are seemingly more complex than temporal relations because they can refer to several different spatially relevant features including size, distance, location and motion. Also, a variety of reference systems can be used within each of these modalities (e.g., the size of an object with respect to another object can be small, same, or large; the speed of an object with respect to (WRT) other objects can be slower, WRT a second object can be the the same, or WRT to a third object can be faster; etc.). Often the reference system is not mentioned explicitly (i.e. needs to be inferred). For instance, in the sentence “the ball moves left”, the direction “left” is with respect to the walls of the Visual Screen which is *not* mentioned in the sentence. This section describes separately how DETE learns the meanings of words that stand for relations within the size, location and motion feature spaces.

11.5.1 Learning about size relations

DETE’s representation of size was introduced in section 3.2.2. The size of an object is represented in the ZFP as a function of the total number of pixels of the Visual Screen which it covers. (Notice that this representation does not allow DETE to handle information about linear dimensions of an object, but rather it defines the size as the surface area covered by the object.)

The following set of experiments demonstrates how size relations between two objects are learned by DETE. DETE’s task was to learn to generate a one-word response -- an answer, (e.g., **W:circle**) while looking at a visual scene that contains two objects of different sizes (e.g., a circle and a square where **Z:circle** > **Z:square**). This verbal response is triggered by a verbal stimulus (input), e.g., **W:what_is_bigger**. In other words, DETE is trained to answer the question “What is bigger”, while looking at two objects of different sizes.

A set of 100 visual scenes was generated. Each of them contained two stationary objects of *different shapes* (a circle and a square; a square and a triangle; or a circle and a triangle) and different sizes -- six possibilities:

- 1) **Z:circle** > **Z:square** or 2) **Z:square** > **Z:circle**
- 3) **Z:circle** > **Z:triangle** or 4) **Z:triangle** > **Z:circle**
- 5) **Z:triangle** > **Z:square** or 6) **Z:square** > **Z:triangle**

The size of each object was chosen from within the whole range of possible sizes, but such that the two objects in a particular image fit together into the visual screen (i.e. no overlapping). The distribution of the sizes throughout the trials was random (multinomial). The rest of the objects’ features, i.e. color and location, were generated at random (the objects were stationary). For each of the 100 scenes, corresponding sentences of the type [**W1:what_is_bigger W2:(circle | square | triangle)**] were also generated. The phrase “what is bigger” was treated as one word. In other words, the teaching strategy was to give the question and to follow it by providing the correct verbal answer while at the same time pointing to the correct object itself (i.e. *focusing DETE’s attention externally*). The training consisted of presenting DETE with the corresponding visual/verbal pairs from the set. The testing was done by presenting novel images from the set and pairing them only with the verbal input **W:what_is_bigger**, after which DETE’s verbal response was observed.

During the testing, DETE's attention *was not forced externally* to the correct object. The response was considered correct if DETE was able to name the bigger object.

First, consider what happens during the learning (training) phase. To perform this task DETE uses the interactions between the visual and verbal memories and the focus of attention mechanism (Figure 11.6). When DETE looks at an image containing only two objects of different sizes, its attention at a given moment can be directed either to the smaller, or to the larger of them. (I do not consider here the case when the attention is directed to a location different from the location of the objects.) As was mentioned before, the fact that DETE is attending to an object is represented by phase-locking the Temporal Attention Window (TAW) to the object's features. (Notice that TAW-phase is always locked to some object but it has the ability to flip from one object to another from time to time.) The verbal and visual representations of each object are phase-locked. In other words, once the verbal-to-visual association of an object is learned, the phase relation between these two representations does not change. As a consequence, a particular verbal response to a visual input cannot be initiated at any B-cycle but only at particular cycles, which appear regularly and are dependent on the oscillations in the visual memory banks.

Immediately before the verbal input is provided, DETE can be attending visually to either one of the objects. A critical event during each training instance is that when the second word -- the name of the bigger object, is presented, DETE's attention is always shifted by the teacher to the bigger object or stays there if it was there to begin with. As a result of this, the trace in the *stm* left by *W:what_is_bigger* is always associated with the visual representation of the object that is bigger in size (more specifically with its size representation since all other features vary from trial to trial). Notice that at the same time there is also activity going on in another location of the ZFP, the one representing the size of the smaller object. While the absolute location of these two activities in the ZFP varies from trial to trial (since the objects in the images have various sizes), their topographic relation in the ZFP is maintained throughout the experiments. The activity representing the bigger object is always above the activity representing the smaller object in the ZFP (see Figure 3.2.2). Due to the fact that the *stm* update happens only during the time when the TAW is open, DETE associates the activity in the verbal memory stronger with the activity in the "bigger" area of the ZFP, rather than to that of the "smaller" area of the ZFP. The topographical relation between "bigger" and "smaller" is also reflected in the stored trace. The effect of the "smaller" is that there is also some moved-ahead previous firing activity in the smaller part of the memory with respect to the activity in the "bigger" part of the memory (Figure 11.6). The WTA mechanism which is coupled with the siZe Feature Memory (ZFM) selects the stronger activation and passes it to the verbal memory bank.

During testing, after the word "*W:what_is_bigger*" is presented, DETE switches and locks the phase of the FAM clock to the activity in the "bigger" segment of the ZFP independently of which object was initially in the focus of attention. This phase switch is due to the fact that the representation of *W:what_is_bigger* in the verbal memory and the input activity in the "bigger" area of the ZFP potentiate each other. As a result, the "bigger" activity in the ZFP prevails over the "smaller" activity while processed by the WTA mechanism. The WTA mechanism coupled with the siZe Feature Memory passes only the "bigger" activity out. This activity in term resynchronizes the FAM clock to the bigger object. This resynchronization is done procedurally. Namely, the WTA phase is used by a procedural module to reset the FAM to the same phase. Once the phase of the TAW is locked to the correct (bigger) object, then the verbal output corresponding to this object is

generated. This is due to the fact that the activity in the visual memory affects the activity in the verbal memory only during the time when the TAW is active.

To summarize, what DETE actually learns in these experiments is to switch appropriately its focus of attention in response to a given verbal input. Once the attention is directed to the correct object in the Visual Screen, the generation of the corresponding verbal response is facilitated by the relation between the opening of the TAW and the information transfer between the visual and the verbal memories.

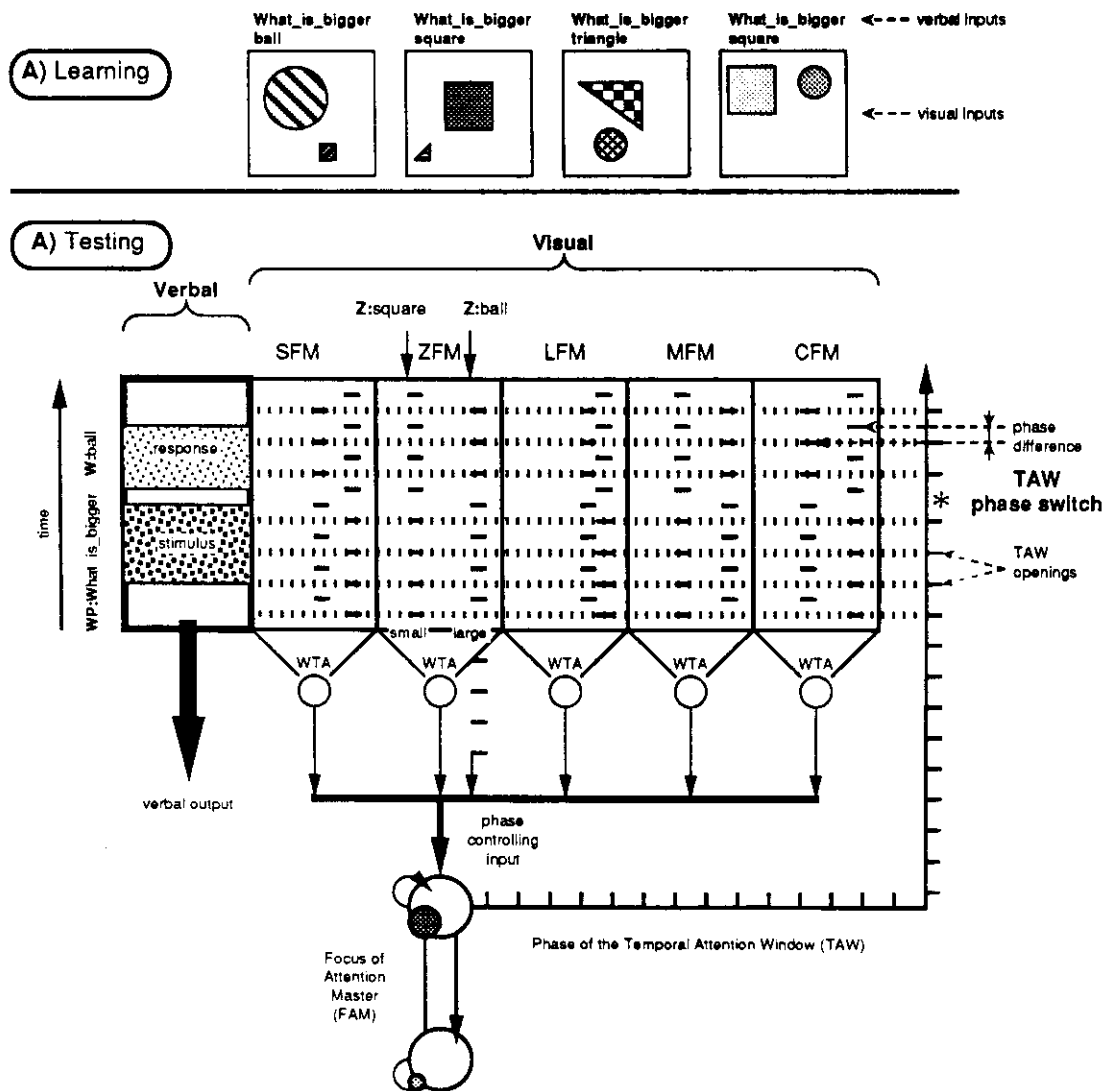


Figure 11.6: Learning about size relations

A) Several of the visual/verbal pairs used for learning. B) Schematic representation of the sequence of events in the system during a test with a square and a larger ball in the Visual Field. The 2-D Visual Feature Memories are shown as 1-D structures extended in time. Activations induced by the visual inputs in the VFMs are shown as sequences of black bars (oscillations). Notice that the oscillations representing the two objects have different phases. The dashed lines across the VF memories represent successive openings of the TAW. The * symbol to the right marks the instant when the initial TAW phase (equal to the phase of the square) is switched by the WTA signal to the phase of the ball.

The actual experiment was conducted by interleaving training and testing trials in the following way. After DETE has learned the words **W:circle**, **W:square** and **W:triangle**, it was exposed once to each of the six possible size relations, i.e. an instance of a visual/verbal pair from each of the 6 kinds was presented (see the first column of Table 11.7). Here, the verbal input contained the question followed by the answer. Testing began after this initial exposure. It was done by showing DETE (at the beginning of a new *moment*) a novel scene (not previously seen) and asking the question **W:what_is_bigger**. DETE's verbal response was then observed for the duration of the *moment*. If DETE did not generate a verbal response during this period, then at the beginning of the next *moment* the same visual scene was paired with the question and followed by the correct answer which was provided by the teacher verbally and visually by pointing at the object. The purpose of repeating the testing input in the form of training input (i.e. contains the answer) after a failure was to increase DETE's chances of learning it. No testing was done after this repetition and the training process continued with the presentation of another training/testing sequence. The frequency distribution of the six possible situations was equal throughout the whole training/testing session. The results of the experiment are shown in Table 11.7.

| size relations btwn. visually presented objects | correct verbal response | 1st correct verbal resp. at trial # | beginning of continuously correct resp. | total # of trials presented |
|---|-------------------------|-------------------------------------|---|-----------------------------|
| Z:circle > Z:square | W:circle | 15 | 176 | 300 |
| Z:square > Z:circle | W:square | 23 | 148 | 300 |
| Z:circle > Z:triangle | W:circle | 29 | 113 | 300 |
| Z:triangle > Z:circle | W:triangle | 18 | 205 | 300 |
| Z:square > Z:triangle | W:square | 26 | 184 | 300 |
| Z:triangle > Z:square | W:triangle | 21 | 152 | 300 |

Table 11.7: Results of learning about size relations

On average, the first correct response for each of the 6 possible size relations was about trial 22. As in all previous tasks, the first correct response was often followed by incorrect responses or no responses at all. With continued training, the trials when DETE generated correct responses became more and more frequent. Then, after a sufficient number of trials (this number was different for each of the six situations), DETE started to generate correct responses to all successive trials. On average, the "fusion" of the correct responses occurred after about 163 training/testing trials. The total number of trials presented was 1,800 or 300 for each situation (size relation).

11.5.2 Learning location relations between objects

First, consider how the descriptions of the spatial relations between two objects (events) by an observer (speaker) can be represented in a coherent way. I propose that such a representation

should be based on three separate notions (*spatial roles*); (1) Speaker location (S), (2) Event (object) location (E), (3) Reference location (R). These spatial roles provide a basis that can formally cover most of the spatial location relations. The S, E, and R are defined below:

1) *Speaker location* (S) is the physical location of the speaker -- the person who makes the observation and produces a statement about it. The speaker location is characterized by two features: (1) direction of gaze, and (2) speaker orientation. For instance, the speaker can be sitting in front of a TV monitor looking at the center of the screen. The direction of the gaze is a vector from the speaker's eye to the center of the screen. The speaker orientation is a vector originating also at the speaker's eye. This vector is perpendicular to the gaze vector and points up with respect to the speaker's body (mouth). Together with an assumption that the speaker (a human) is standing straight and still with his eyes fixed in their normal position in the orbits, these two features of the speaker location provide an unique coordinate system which can be used to describe the location of objects on the Visual Screen.

2) *Event location* (E) is the location of the object or event within the Visual Screen that the speaker is talking about. It can be the same or different from the S.

3) *Reference location* (R) is the location of a second object/event in the world with respect to which the first object/event is described. Again, R can be the same or different from S and/or E.

To illustrate the notions of S, E, and R, consider the sentence "The ball is behind the triangle". In this sentence, E is the location of the ball, R is the location of the triangle, and S is the location of the person (or DETE) with respect to the Visual Screen. Notice that S is not mentioned explicitly. From the sentence we can infer that S with respect to E and R is such that the three locations are aligned and R is between S and E. What distinguishes the roles which object A (the ball) and object B (the triangle) play (e.g., an E or a R) is the order in which they appear in the sentence relative to the content words (e.g., behind, in front, etc.).

In our set-up we have two speakers, the teacher who provides the verbal input to DETE and DETE itself which generated the verbal output. For simplicity, an assumption is made that the locations of these two speakers are the same at all times. Also, it is assumed that DETE and the teacher have the same orientation and look at the same object at all times (and specifically during training). What are the representations of the three spatial roles within the framework of DETE?

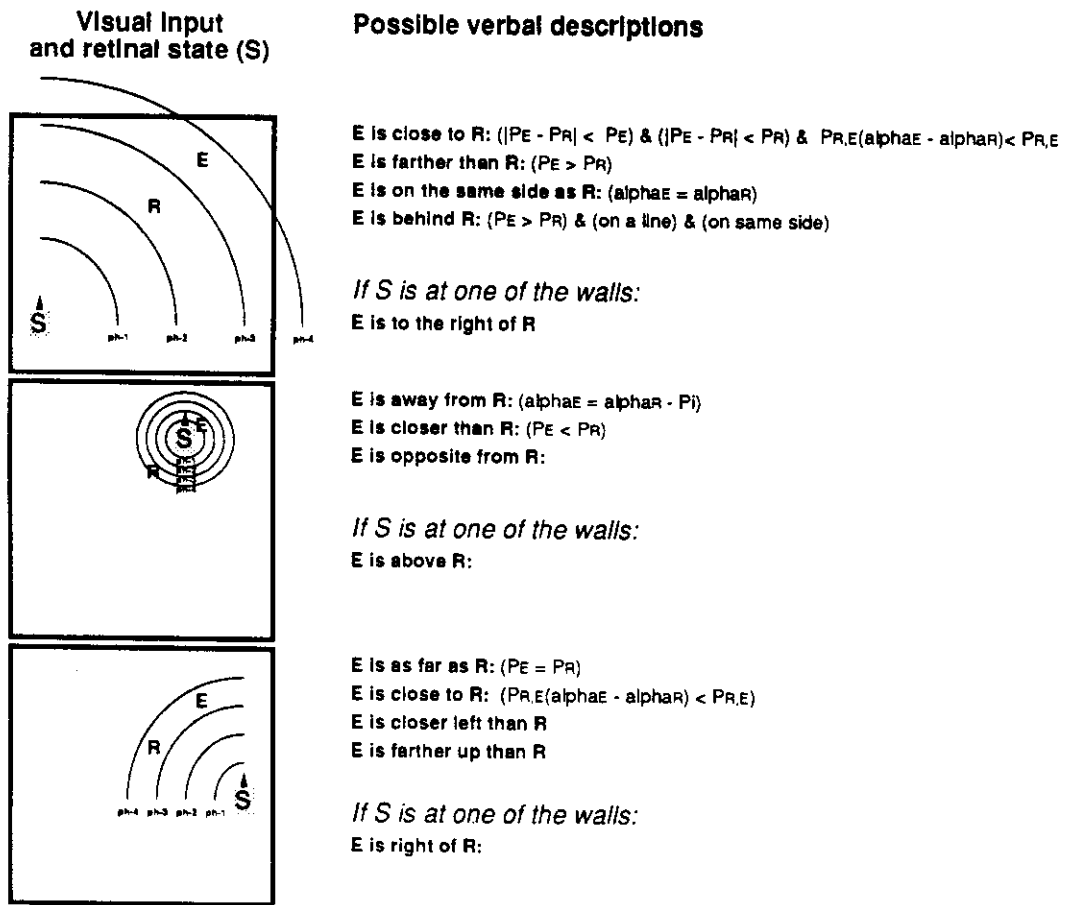
The *Speaker location* is not represented directly in DETE as some activity in the Location Feature Plane. It is represented indirectly by means of the location of the center of the retina (EYE) on the Visual Screen which in turn is reflected in the phase of the Temporal Attention Window opening. The farther away an object is from the center of the EYE, the larger the phase delay of the oscillations which represent this object. For simplicity, I assume that the speaker location within the Visual Screen can change (i.e. DETE's EYE can be at a different location on the VS), but the orientation is always the same -- pointing up.

The *Event location* is represented by activity in the LFP which corresponds to the object that is involved in the particular event. In the example given above, this is the activity in the LFP that represents the ball.

The *Reference location* is represented by a second patch of activity in the LFP. This activity can correspond to another object or to the particular part of the frame of the Visual Screen (e.g., the left wall, or the upper right corner, or the whole frame itself). In the example given above, this is the activity in the LFP that represents the triangle.

As will be seen in section 11.7 concerning temporal relations, the above-defined set of spatial roles (which is used to describe spatial relations between two objects with respect to an observer) corresponds to the set of roles proposed by Reichenbach to describe temporal relations (Raichenbach, 1947).

Since spatial relations are higher dimensional (e.g., 3-D in the real world and 2-D in the Blobs world) as compared to the temporal relations (1-D WRT the time axis), the set of possible verbal expressions that can be used to characterize spatial relations is larger than the set used for characterizing temporal expressions. Some examples of different verbal descriptions of a particular spatial arrangement of E and R while S varies are given in Figure 11.7.



In each image, E and R have the same locations on the screen but the location of S varies while its orientation is maintained

Figure 11.7: Description of spatial relations

Possible verbal descriptions of a fixed spatial relation between E and R when S varies -- in three different positions. The size of the retina corresponds to the diameter of the largest circle centered at S. The other three smaller, concentric circles demarcate the areas of the Visual Field in which objects are represented by oscillations with different phases (e.g., Ph-1, ..., -4). The sentences in **bold** to the right of each scene are possible verbal descriptions of the corresponding scenes. The text in plain font after each sentence is a list of discriminating conditions. P_E and P_R are the phases of the oscillations representing the E and R objects respectively. α_E and α_R correspond to the angles spanned between the speaker orientation (vertical direction) and the E and R objects respectively.

In a series of experiments DETE was taught to generate a verbal description of E with respect to R. The description was generated in response to the question "Where_is-E?". In separate experiments DETE learned the meaning of the word pairs: *closer / farther*; *in-front / behind*; *left_of / right_of*. In these experiments I consider only situations in which the Speaker location (i.e. DETE's observation point which is the same as the location of the center of the EYE on the Visual Screen) is always in the center of the Visual Screen. At the same time, the location of the objects/events which are being described (i.e. the Event locations), and the locations of the objects/events with respect to which the descriptions made (i.e. the Reference locations) vary. I also allow the size of the retina to vary depending on how far away from the center of the retina the two objects are. For the experiments described below, the size of the EYE is set large enough (but not larger) so that both objects are within the Visual Field.

Learning the meanings of *closer & farther*

The meanings of the words *closer & farther* were learned in a similar way as *smaller & bigger* (see section 11.5.1). DETE's task was to learn to answer the question "What_is_closer", while looking at two objects at different locations (E -- the closer object & R the farther object) and respectively at different distances from S.

A set of 100 visual scenes was generated, each of which contained two stationary circles of the same size but of *different colors* (a red and a green; a green and a blue; or a blue and a red) and different relative locations (L) with respect to S -- six possibilities (here L:color stands for the distance of the object with the particular color from S and the symbol ">" shows which distance is bigger):

- | | | |
|-----------------------|----|---------------------|
| 1) L:red > L:green | or | 2) L:green > L:red |
| 3) L:red > L:triangle | or | 4) L:blue > L:red |
| 5) L:blue > L:green | or | 6) L:green > L:blue |

Notice that for this experimental setup the only degree of freedom was within the Location Feature Plane. To simplify the experimental setup, the rest of the features were preset and maintained constant across all trials. The locations of the objects were chosen from within the whole range of possible locations. The distribution of the locations (distances) throughout the trials was random (multinomial). For each of the 100 scenes, corresponding sentences of the type [W1:what_is_closer W2:(red | green | blue)] were also generated. The phrase "what is closer" was treated as one word. The teaching strategy was to give the question and to follow it by providing the correct verbal answer while at the same time pointing to the correct object itself (i.e. *focusing DETE's attention externally*). The training consisted of presenting DETE with the corresponding visual/verbal pairs from the set. Similarly, as in the size learning experiment described above, the

testing was done by presenting novel images from the set and pairing them only with the verbal input **W:what_is_closer**. After this, DETE's verbal response was monitored. During the testing, DETE's attention *was not forced externally* to the correct (closer) circle. The response was considered correct if DETE was able to name the color of the closer circle.

How does DETE learn this task? At the beginning of each trial DETE's attention can be focused to either one or none of the circles. When the second word -- the color of the closer circle is presented, DETE's attention is always shifted by the teacher to that circle or stays there if it was there to begin with. As a result, the trace in the *stm* left by **W:what_is_closer** is always associated with the visual representation of the location of the circle that is closer. Notice that at the same time there is also ongoing activity in another part of the LFP -- at the location of the circle which is farther away. While the absolute positions of these two activities in the LFP vary from trial to trial (since the circles in the images have various locations), their topographic relations in the LFP are maintained throughout the experiments. The activity representing the closer circle is always closer to S in the LFP as compared to the activity representing the farther circle. Since the *stm* update happens only during the time when the TAW is open, DETE associates the activity in the verbal memory stronger with the activity in the "closer" area of the LFP, rather than to that of the "farther" area of the LFP. Therefore, the topographical relation between "closer" and "farther" is also reflected in the stored trace.

During testing, after the word "**W:what_is_closer**" is presented to DETE, it switches by itself the phase of the FAM clock and locks it to the activity in the "closer" segment of the LFP independently of which object was initially in the focus of attention. This phase switch is due to the fact that the representation of **W:what_is_closer** in the verbal memory and the input activity in the "closer" segment of the LFP potentiate each other by exchanging signals along the inter-modular fibers (see section 10.2.2) which results in injection of *stm* in the corresponding predictiontrons and increase of their activation levels. As a result, the "closer" activity in the LFP prevails over the "farther" activity while processed by the WTA mechanism. Consequently, the WTA mechanism coupled with the Location Feature Memory passes only the "closer" activity out. The output activity resynchronizes the FAM clock to the closer circle. Once the phase of the TAW is locked to the correct (closer) circle, then the verbal output corresponding to this object is generated since, as was mentioned before, the activity in the visual memory affects the activity in the verbal memory only during the Temporal Attention Window (TAW).

Similarly to the size learning experiment described in the previous section, in this experiment the training and testing trials were interleaved. After DETE has learned the words **W:red**, **W:green**, and **W:blue**, it was exposed once to each of the six possible location relations, i.e. an instance of a visual-verbal pair from each of the 6 kinds was presented (see Table 11.8, column 1). Here, the verbal input contained the question followed by the answer. Testing began after this initial exposure. It was done by showing DETE (at the beginning of a new *moment*) a novel scene (not previously seen) and asking the question **W:what_is_closer**. DETE's verbal response was then observed for the duration of the *moment*. If DETE did not generate a verbal response during this period, then at the beginning of the next *moment* the same visual scene was paired with the question and followed by the correct answer provided by the teacher verbally and visually (i.e. pointing at the object). The frequency distribution of the six possible situations was equal throughout the whole training/testing session. The results of the experiment are shown in Table 11.8.

| Location relations btwn. visually presented circles | correct verbal response | 1st correct verbal resp. at trial # | beginning of continuously correct resp. | total # of trials presented |
|---|-------------------------|-------------------------------------|---|-----------------------------|
| L:red > L:green | W:red | 9 | 77 | 300 |
| L:green > L:red | W:green | 12 | 84 | 300 |
| L:red > L:blue | W:red | 12 | 76 | 300 |
| L:blue > L:red | W:blue | 11 | 91 | 300 |
| L:green > L:blue | W:green | 8 | 87 | 300 |
| L:blue > L:green | W:blue | 11 | 69 | 300 |

Table 11.8: Results of learning closer / farther relations

On average, the first correct response for each of the 6 possible location relations was about trial 10. Compared to the results of the size learning experiment, here the learning is somewhat faster (10 vs 22). This can be explained by the fact that the degrees of freedom in this experiment are less than in the size learning experiment. Namely, in the present experiment only the color and the location of the objects (circles) were allowed to vary across trials whereas in the size-learning experiment the size was also allowed to vary. In general, the higher the degrees of freedom, the slower the learning. On average, DETE started to continuously produce correct responses after about 80 training/testing trials. The total number of trials presented was 1,800 or 300 for each situation (location relation).

Learning the meanings of *in-front & behind*

The *In-front/Behind* spatial relation between two objects can be viewed as the co-occurrence of two other relationships between the objects: (1) closer/farther, and (2) on the line that connects S, E and R. The learning of the first of these relationships was described above. The representation and learning of the second relationship takes advantage of the specific distribution pattern of the seeds in the Location Feature Memory (a spiral arrangement -- see section 10.1.4), and the radial connectivity pattern of the parallel fibers in this memory. This feature of the LFM ensures that, when E and R are on the same radial, the activity which represents them (i.e. the injected *stm*) is summed (added up) producing a strong radial activation pattern. As a result, any such pattern of activity generated in the Location Feature Memory can be interpreted as the presence of two objects on the same line as S (remember that the Location Feature Memory is rigidly coupled to the EYE).

In summary, DETE learns about relations of objects in space (the Visual Screen) similarly to the way size relations are learned. In both cases the choice of representations and specifically the functional topographical organizations of the LFP and ZFP on the one hand, and the dynamics of the KATAMIC memory, on the other hand, provide the physical basis for the learning.

11.6 Learning about motion relations

Propositions about motion carry information simultaneously about some or all of the following features: location (of speaker -- S, object -- E, and reference -- R) but also about the direction (of object and reference), and speed (of object, and reference).

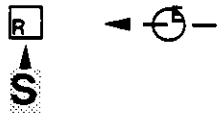

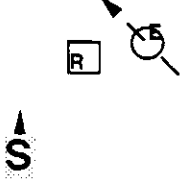
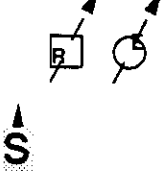
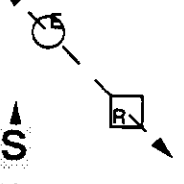
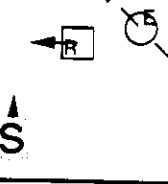
| Visual inputs | Verbal inputs | Relations between the representational elements |
|---|--|--|
|  | <p>The E is coming to the R</p> | <p>$(PE > PR) \text{ \& } (PE \text{ decreases})$</p> |
|  | <p>The E is going to the R</p> | <p>$(PE < PR) \text{ \& } (PE \text{ increases})$</p> |
|  | <p>The E is passing by the R</p> | <p>$(DE,R \text{ decreases...increases})$</p> |
|  | <p>The E is going along with the R</p> | <p>$(DE,R = \text{const.} \text{ \& } PE, PR \text{ increase})$</p> |
|  | <p>The E is going away from the R</p> | <p>$(DE,R \text{ increases})$</p> |
|  | <p>The E is faster than the R</p> | <p>$(ME > MR)$</p> |

Figure 11.8: Descriptions of motion relations

Possible verbal descriptions of various motion relations between E (a ball), R (a square), and S. Each sentence to the right of a scene is a possible verbal description of this scene. The relations between the representational elements characteristic for each scene are shown in parentheses. P_E and P_R are the phases of the oscillations representing the E and R objects respectively. $D_{E,R}$ is the distance between E and R, and M_E and M_R are the distances of the representations of the two objects from the center of the Motion Feature Plane (MFP).

Of importance is also that some temporal component of motion (the time when it occurs, its duration, etc.) is always associated with each verbal description of motion. This component is commonly associated with the current NOW window. In other words, to evaluate the direction and velocity of motion the evaluator (DETE or a human) needs time.

Another important observation is that the location relations between moving objects are typically not of the *order type* as in the case of temporal relations (e.g., before, after) but are *topographical* in nature (e.g., left of, above, close to, etc.). However, in sentences that describe the speeds of the objects contain information of the *order type* (e.g., faster, slower) there is a linear scale along which they can be described. Some visual examples and the corresponding verbal descriptions of motion relations between two objects in the Visual Screen at a given time are shown in Figure 11.8.

In learning the meanings of motion-related words, it is most important that each word have at least one distinctive representational feature in the Location and/or Motion Feature Memories. The representations of motion words/phrases can have spatial and/or temporal components. For instance, "passing by" has a temporal component (the distance between the objects in the Location Feature Plane initially decreases and then increases with time), whereas, "is faster than" also has a spatial component (the distance from S of the activity in the Motion Feature Memory that corresponds to E is larger than the distance to the cell assembly representing R).

11.7 Learning temporal relations between events

Recent studies in the field of computational linguistics have addressed many of the issues related to the understanding of temporal relations in narratives (Raichenbach, 1947; Hinrichs, 1988; Passonneau, 1988; Webber, 1988). Some linguistic forms that carry time information are: verb tenses (e.g., present, past, future, etc.), temporal adverbials (e.g., yesterday, morning, tonight), and temporal adjectives (e.g., slowly, fast). This section demonstrates how DETE learns to understand one of these linguistic forms, namely the verb tense.

The general approach taken in computational linguistics to the analysis of propositions about time is to represent them symbolically using structures such as histories (Forbus, 1985). Such structures are created during sentence parsing using analytically derived, hand-coded rules.

More specifically, the majority of the current theories of time representation build upon the work of Reichenbach (Raichenbach, 1947). According to this theory, all temporal relations encountered in narratives can be accounted for in a model which uses three different time measures:

(1) *Speech time* (S) -- a point in real-time at which the utterance is produced (read) or at which the question-answering session is taking place.

(2) *Event time* (E) -- the time when the event mentioned in the utterance occurs. The Event time can be before, at the same time as, or after the Speech time.

(3) *Reference time (R)* -- the temporal perspective (point in time) from which an event is viewed by the person who generates the utterance. It indicates where along the time axis the current focus of attention is located.

Some examples of the possible relations between the three time measures and the corresponding tenses in the English language are given in Table 11.8. In the last column the table shows the representations of the simple and perfect forms of the present, past and future tenses in terms of Reichenbach's Speech time (S), Event time (E), and Reference time (R). Here "<" indicates "temporary prior to", and "=", "at the same time as".

| | Tense | Example | S/E/R relations |
|----|-----------------|---------------------------------|------------------------|
| 1a | Simple present | The ball moves | $E=R=S$ |
| 1b | Present perfect | The ball has hit the square | $E<R=S$ |
| 2a | Simple past | The ball hit the wall | $E=R<S$ |
| 2b | Past perfect | The ball had hit the wall | $E<R<S$ |
| 3a | Simple future | The ball will hit the wall | $S<R=E$ |
| 3b | Future perfect | The ball will have hit the wall | $S<E<R$ |

Table 11.8: Verbal tenses and their S/E/R representations

It is important to notice that the relations between these three time measures are only with regard to their order, i.e. qualitative but not absolute or quantitative. Also each of these measures is actually a temporal window (has a real time duration) rather than a point in time.

In DETE, each of the temporal roles is represented as an activation of neural-assemblies in a specific plane of the Temporal Memory (see section 9.3). Sometimes, all three roles are represented in the same plane (e.g., in the case of present tense), in other cases two of them can share one and the same plane and the third is represented in a different plane (e.g., future tense, present perfect tense, etc.). In yet other cases, each of the roles is represented in a different plane (e.g., future perfect tense, past perfect tense, etc.). An important feature of these representations (based on the dynamics of the temporal planes) is that at each consecutive *moment* the representations of E, R, and S are shifted to the next temporal plane in ascending order. In other words, the temporal memory maintains traces of several (up to 8) consecutive *moments*.

Together with the aforementioned general features of the temporal role representations, each of these roles has its own specific representational characteristics. For instance, Speech time (S) is represented by the activation generated in the TP-0 of the verbal memory bank by the verbal input or by DETE during the generation of a verbal response. The representation of S also includes the activity generated in the visual bank of the same temporal plane. This activity can be produced either by direct visual input (e.g., during the learning of present tense) or it can be induced in the visual bank by the activity in the verbal bank (e.g., during the process of comprehending of verbal input). The Event time (E) is represented as activation generated in the TP-0 of the visual bank by the visual input -- a sequence of frames that capture the event. The Reference time (R) is also represented as an activation in the visual bank. This activation is induced by the "referent" visual event. The most important characteristic of this representation is that the phase of oscillations of this activation is always the same as the TAW. In other words, the temporal aspect or temporal focus is always directly related to the time of the Referent event (R).

11.7.1 Present tense

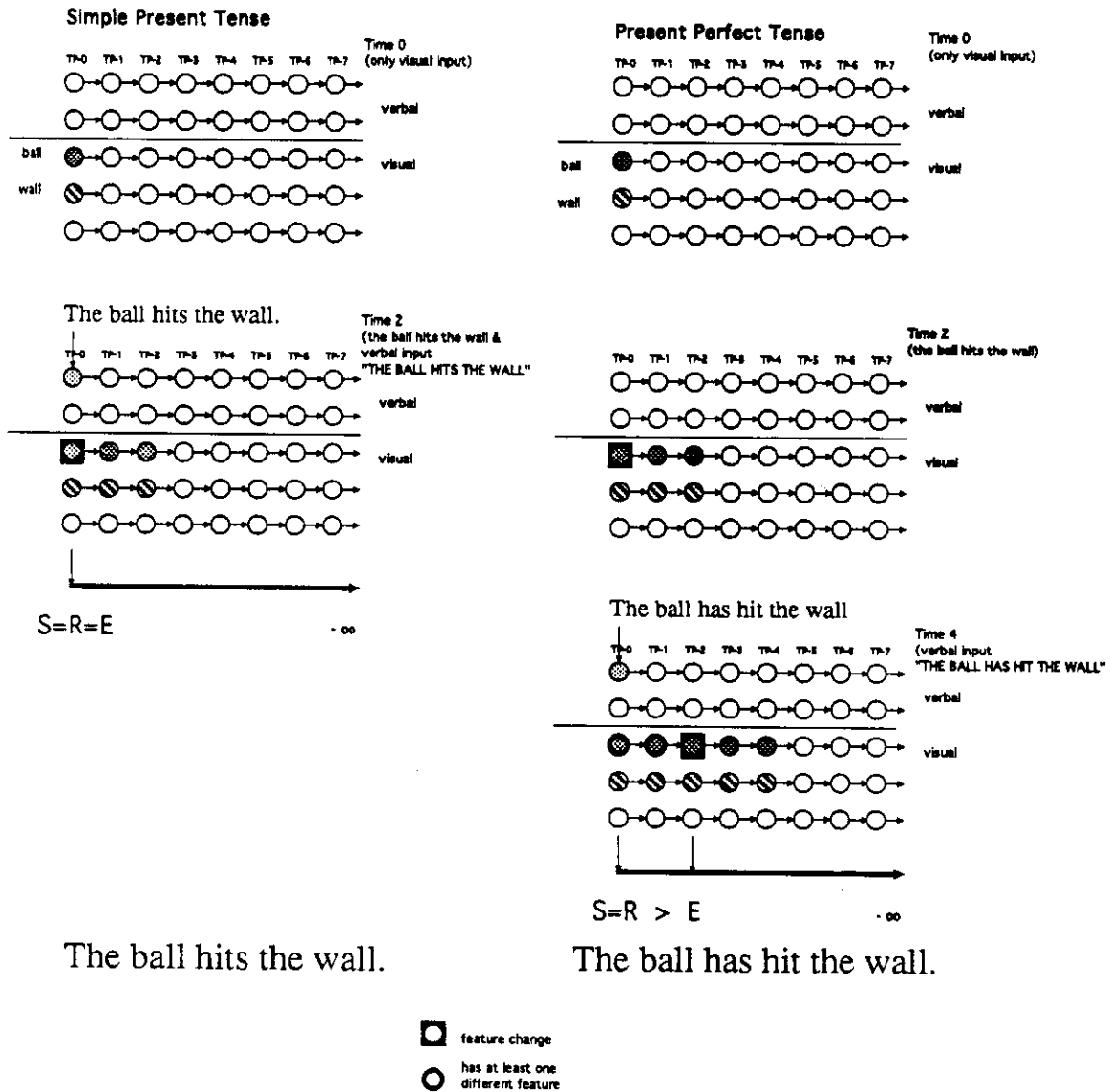
DETE learns the meaning of the verb “hits” in its present tense form in the following way. It observes a circle moving towards one of the walls of the screen and within a certain small time window around the event of hitting (i.e. within the same *moment* -- see Table 5.1) it is told: “The circle hits the wall”. This visual/verbal input pair is repeated multiple times while both the locations of the hits on the walls and the objects that hit the walls vary. As a result, DETE associates the whole sentence and in particular the word “hits” with an object that comes in contact with a wall independently of where along the wall such a contact is made.

Learning of Simple Present Tense and Present Perfect Tense in DETE is schematically represented in Figure 11.9. DETE’s Temporal Memory Planes (see section 9.3) are shown in this figure as a series of 8 unidirectionally connected columns of circles (e.g., TP-0, TP-1, ..., TP-7). There are two banks of circles in each Temporal Plane -- a visual and a verbal. The verbal bank contains 2 chains and the visual contains 3 chains of circles. Each circle in the visual bank symbolizes the distributed neural assembly (a subset of all predictrons of the visual memory) of oscillating predictrons which represent all features of an individual object at a given *moment*. A *moment* is defined as the interval between two resets of the *stm* in the STM. This interval is one sentence long. The left-most column of circles symbolizes the present *moment* (i.e. TP-0). Each successive column of circles represents the activity of its left-hand neighbor at the previous *moment*. Shading of circles indicates activation of the neural assembly -- a sustained oscillatory process persisting during that *moment*. Different shadings indicate different phases of the oscillations. Notice that all active circles in a particular chain have the same phase. The learning of tenses is presented schematically in the figure by a series of snapshots of the network’s activity at successive *moments*.

The learning of Simple Present Tense is presented in two snapshots (Figure 11.9A). At *moment 0* DETE is looking at a ball moving towards a wall. The motion of the ball continues during *moment 1* (not shown in the figure). During *moment 2* DETE sees the ball hitting the wall and at the same time hears the verbal input “The ball hits the wall”. The event of hitting is represented by the black box around the circle. This black box indicates that a change in some of the features of the ball has occurred (e.g., it has come in contact with the wall and as a result its direction of motion has changed).

After DETE has been exposed to multiple pairs of visual events and their verbal descriptions in present tense, we test how it understands a sentence which is a proposition about an event in present tense. The level of understanding can be examined by observing the activity that DETE generates in the visual memory banks (the image generated in its “mind’s eye”) when it hears a sentence describing an event in present tense without having any visual input.

When DETE gets a verbal input that contains a proposition in simple present tense, the following happens (Figure 11.10A). The verbal input associatively triggers activity in the visual memory. The pattern of activity in the visual memory is localized in the same TPs which were active when the particular visual / verbal association was learned. In the case of simple present tense, the induced activity is in TP-0. In successive *moments*, the activity in both the visual and the verbal banks is immediately transferred (during one B-cycle) via the “fast” connections (see section 9.3) to higher order TPs. These leaves lower order TPs prepared to get new inputs (e.g., successive sentences and/or visual inputs).

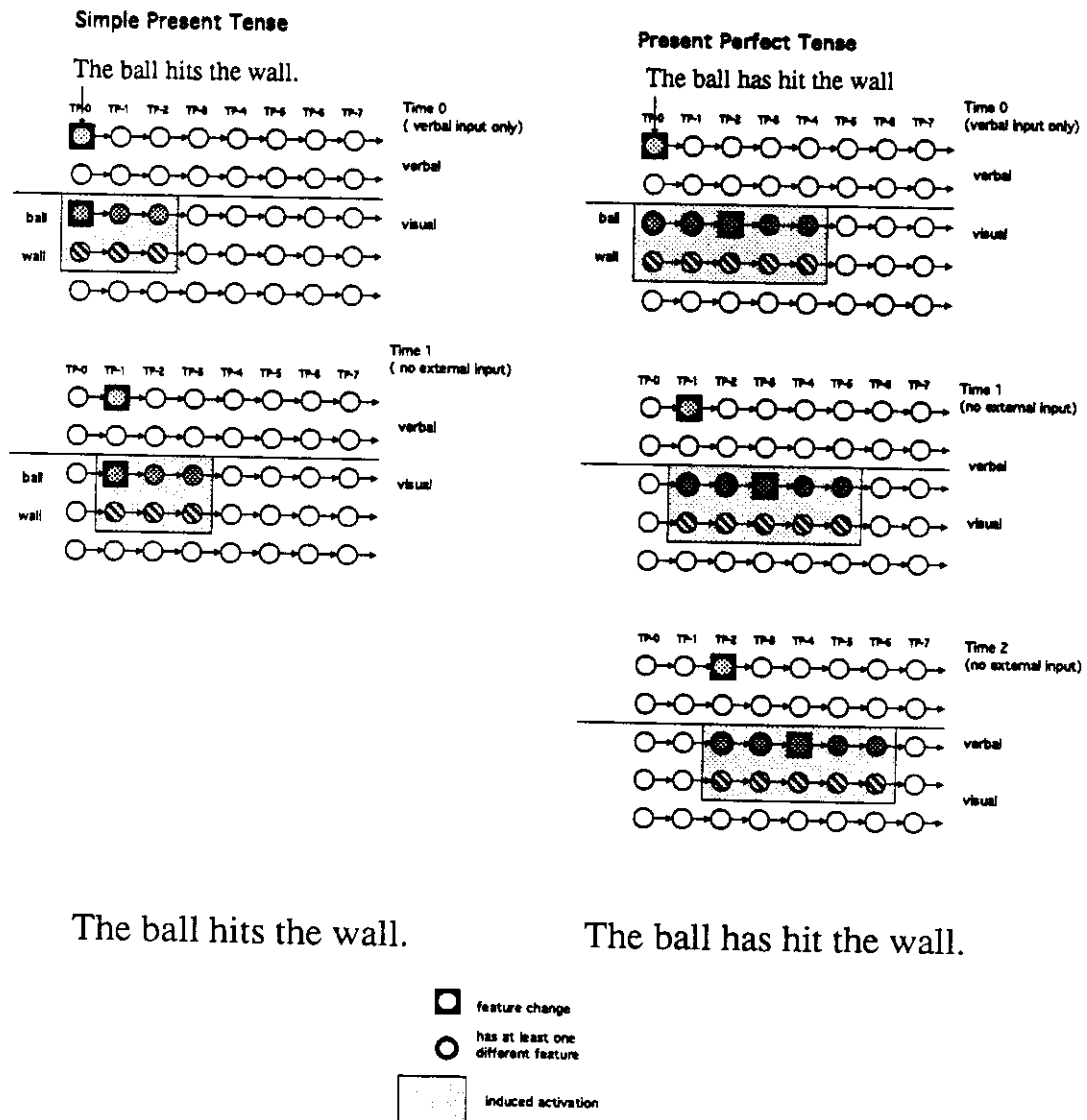


The ball hits the wall.

The ball has hit the wall.

Figure 11.9: Learning present tense

A schematic representation of the sequential stages of activation spread within the Temporal Feature Maps during learning of **A)** Simple Present Tense, and **B)** Present Perfect Tense. Empty circles indicate a lack of activation in that part of the plane. Shaded circles indicate activated neural assemblies. Different shades of gray represent different oscillation phases. Boxed circles represent assemblies of predictrons for which at least one element of the assembly has changed its activity status (i.e. stopped or started to oscillate) during the particular moment. Outlined and shaded circles represent assemblies of predictrons at least one of which is different as compared to the shaded circles (i.e. sustained change).



The ball hits the wall.

The ball has hit the wall.

Figure 11.10: Imagining present events

A schematic representation of the sequential stages of activation spread within the Temporal Feature Maps during understanding of **A)** Simple Present Tense, and **B)** Present Perfect Tense. The meanings of the symbols are the same as in Figure 11.9. Description of the figure is given in the text.

Learning of present perfect tense happens in a similar way as that of simple present tense. It requires that appropriate activation patterns are present in DETE's temporal memory during several (minimum two or three) consecutive *moments*. For instance, at time 0 DETE gets an activation in the visual memory bank by looking at a ball which is moving towards the wall (Figure 11.9B). A few *moments* later (e.g., at time 2) the ball hits the wall which produces activation in the appropriate visual feature memories (e.g., in the Motion Feature Memory -- change of direction of motion). This activation represents E -- the event time. In yet another few *moments* later (e.g., at time 4) DETE gets the verbal input "The ball has hit the wall". This input generates activation in the TP-0

of the verbal memory bank which represents S -- the speech time. Of importance here is that this activation is in phase with the Temporal Attention Window. In other words, it also represents R -- the reference time. Notice that R is not explicitly mentioned in this sentence. R is the same as the speaker time S. The activation generated in the verbal memory bank is associated with the activations currently present in various temporal planes of the visual memory bank and has left memory traces there.

What happens when DETE is given the verbal input "The ball has hit the wall" without any visual input, after it has learned the meaning of present perfect tense. The activation produced in the TP-0 plane of the verbal memory bank immediately (i.e. during the *same moment*) induces activation in a set of planes (e.g., TP-0,...,4) in the visual memory bank (Figure 11.10B). This activation is triggered by the signals going along the fast connections connecting the individual temporal memory planes. The induced activation pattern in the visual bank recreates the activation pattern which was present there (as a result of direct visual input) during the learning process.

11.7.2 Future tense

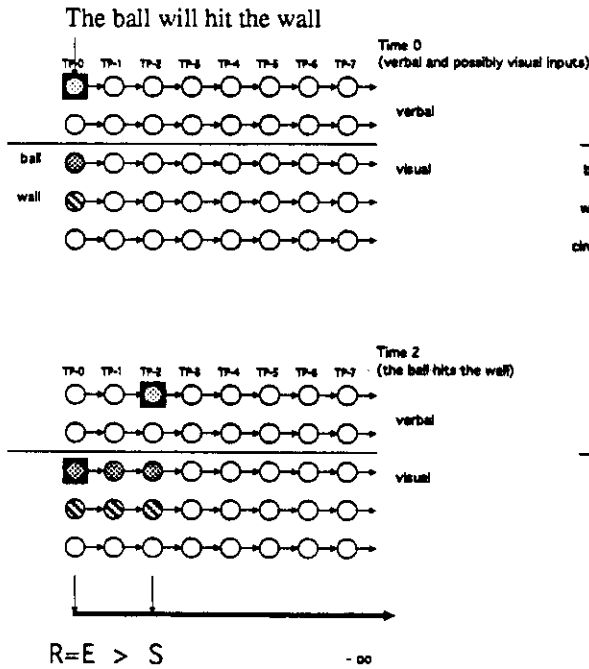
Observe a ball moving towards the wall of the Visual Screen and within a certain time period before it hits the wall tell DETE: "The ball will hit the wall". The representation of the phrase "will hit" induces activity in the verbal bank of TP-0 (Figure 11.11A). This activity is transferred to higher order TP-s in consecutive *moments*. When the actual event of hitting happens (e.g., two *moments* later), it induces a change of activity in the visual bank. The learning of simple future tense is done when the activity in the verbal bank of TP-2 is associated with the activity in the visual bank of TP-0. Note that depending on how late after the verbal input the visual input comes, the position of the activity representing the verbal input can be anywhere from TP-1 to TP-7.

In a much similar way DETE learns the meaning of future perfect tense (Figure 11.11B).

When DETE gets a sentence containing simple future tense, e.g., "The ball will hit the wall" without a corresponding visual input, the following happens (Figure 11.12A). At *moment 0* the representations of the ball and the wall are activated (by induction from the verbal memory) in the visual memory bank. At successive *moments* (1, 2, 3 ...) the representation of the verbal input moves from TP-0 to TP-1, TP-2, TP-3,... and in each case it keeps active (via the "fast" connections) the representation of the ball, the wall, and the event of hitting in TP-0. During learning, the verbal activity was at different TPs (1-7) when it was associated with the visual activity in TP-0, therefore during understanding the shifting of verbal activity will keep the corresponding units in the visual part of TP-0 active constantly at every successive *moment*. This maintained activity in TP-0 represents an expectation of the event of "hitting". Two extreme cases can occur. First, if during learning the visual input came always with the same delay after the verbal, then during understanding the expectations will not be generated continuously but will come only at the *a priori* learned *moment* of time. Second, if during learning the verbal and visual inputs were separated with a long temporal gap (e.g., 8 *moments*) then DETE will not be able to establish the association between both and future tense will not be learned.

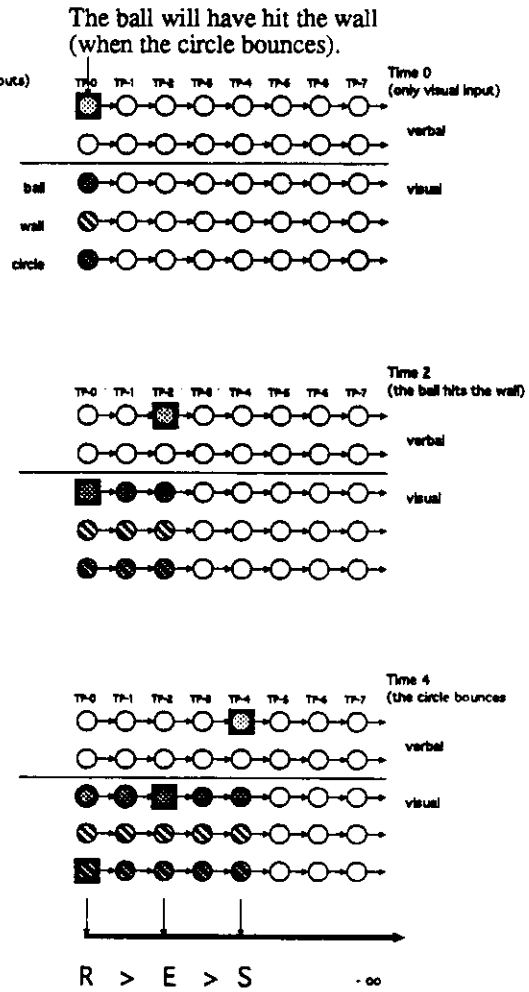
The sequence of visual representations (i.e. its understanding) which DETE generates when it gets a verbal input proposition in future perfect tense are shown in Figure 11.2B and their interpretation is similar.

Simple Future Tense



The ball will hit the wall.

Future Perfect Tense



The ball will have hit the wall
(when the circle bounces).

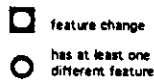
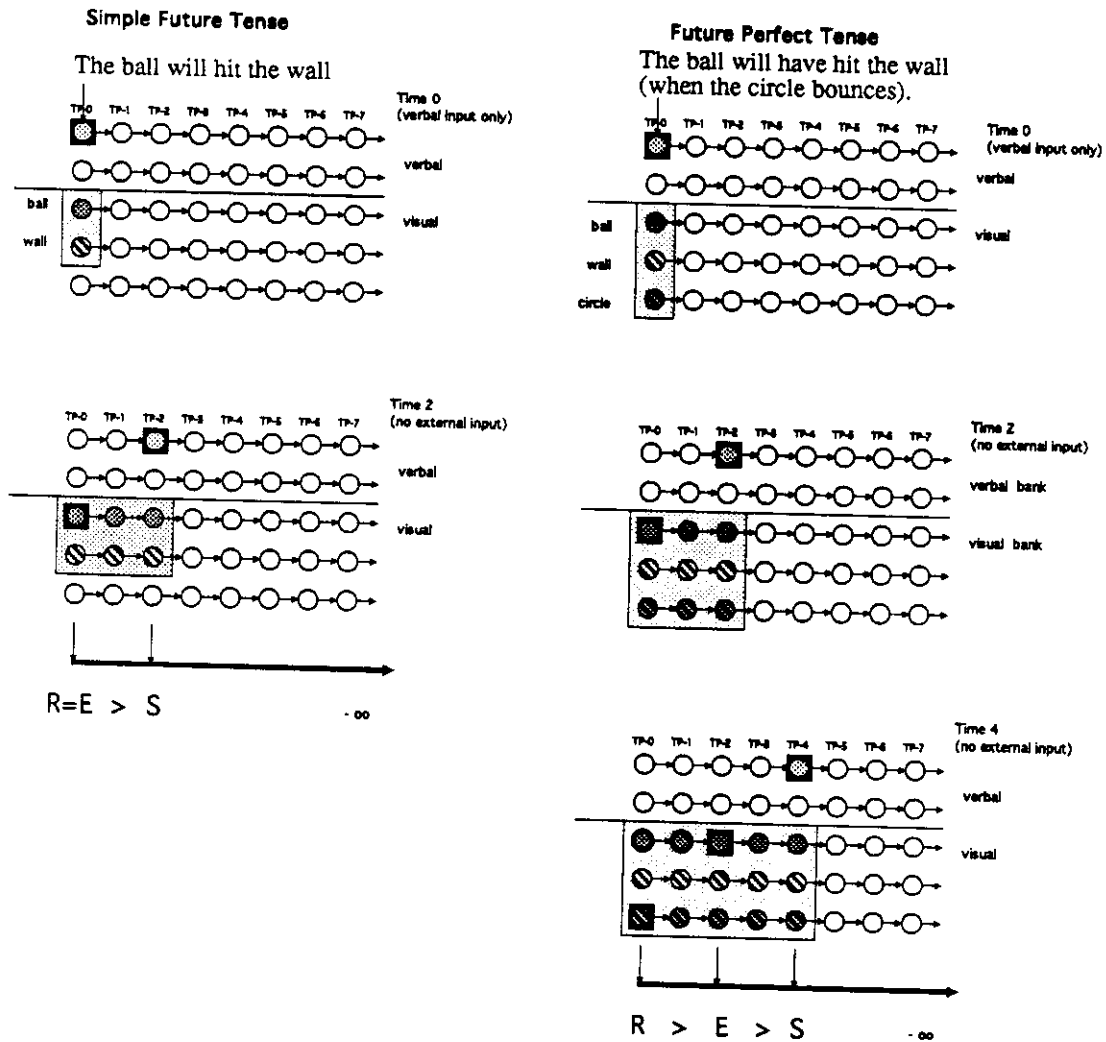


Figure 11.11: Learning future tense

A schematic representation of the sequential stages of activation spread within the Temporal Feature Maps during learning of **A)** Simple Future Tense, and **B)** Future Perfect Tense. The meanings of the symbols are the same as in figure 11.9. Description of the figure is given in the text.



The ball will hit the wall.

The ball will have hit the wall (when the circle bounces).

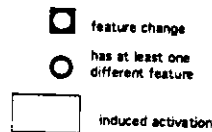
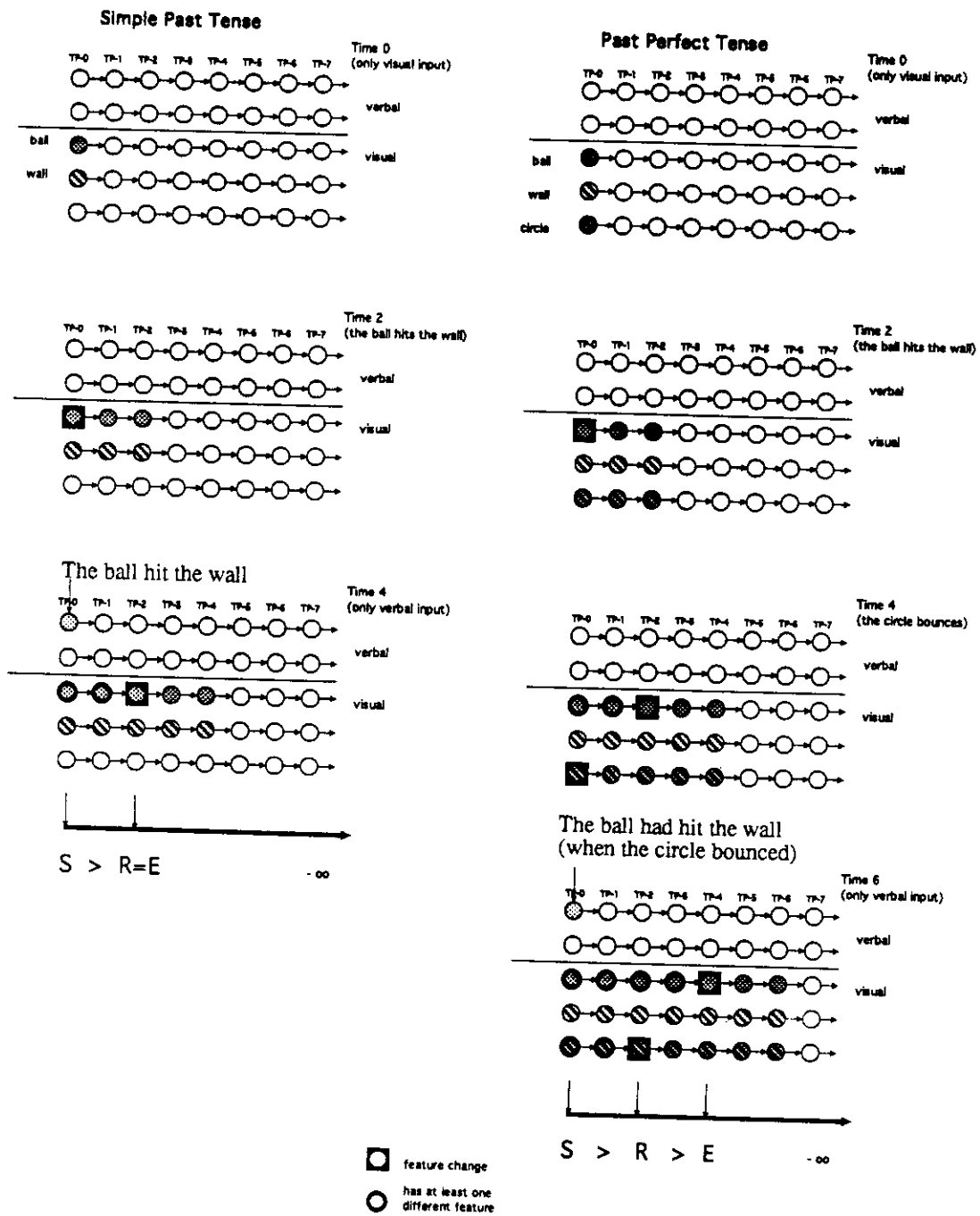


Figure 11.12: Imagining future events

A schematic representation of the sequential stages of activation spread within the Temporal Feature Maps during understanding of **A)** Simple Future Tense, and **B)** Future Perfect Tense. The meanings of the symbols are the same as in figure 11.9. Description of the figure is given in the text.

11.7.3 Past tense

DETE observes a ball hitting one of the walls of the Visual Screen and few *moments* later it hears: "The ball hit the wall" -- a simple past tense proposition.

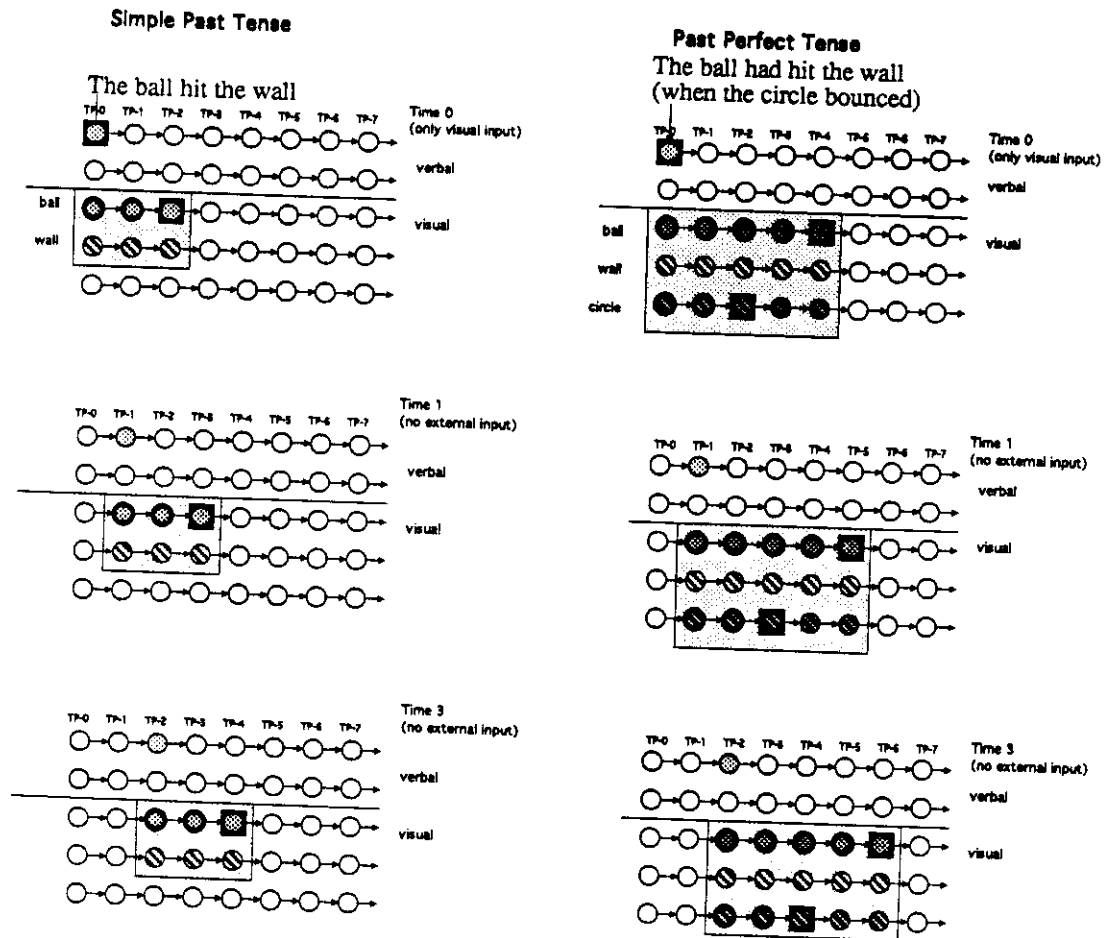


The ball hit the wall.

The ball had hit the wall (when the circle bounced).

Figure 11.13: Learning past tense

A schematic representation of the sequential stages of activation spread within the Temporal Feature Maps during learning of **A) Simple Past Tense**, and **B) Past Perfect Tense**. The meanings of the symbols are the same as in figure 11.9. Description of the figure is given in the text.



The ball hit the wall.

The ball had hit the wall
(when the circle bounced).

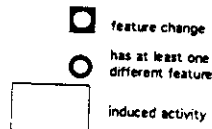


Figure 11.14: Imagining past events

A schematic representation of the sequential stages of activation spread within the Temporal Feature Maps during understanding of **A) Simple Past Tense**, and **B) Past Perfect Tense**. The meanings of the symbols are the same as in figure 11.9. Description of the figure is given in the text.

At the B-cycle when the ball hits the wall this event induces the corresponding activation in the visual bank of TP-0 (Figure 11.13A). During each successive *moment* before the verbal input is given, this visual activity is transferred to higher visual TPs. When the verbal input is presented, it induces activity in the verbal bank and this activity is associated with the activity in the visual bank. Therefore, the meaning of the word “hit” (past tense) is represented as an association of the corresponding verbal activity in the lower-order TP and the visual activity in the higher-order TP. During each instance of past tense learning, the stimuli (i.e. the verbal and visual inputs) can be separated by one or more *moments*. This period is called an Inter Stimulus Interval (ISI). The ISI in DETE can vary from 1 to 7 *moments*.

The learning of past perfect tense is done in a similar fashion as shown in Figure 11.13B.

What does DETE imagine when it hears the sentence “The ball hit the wall” without seeing the actual event. In other words, how does it “understand” this utterance? The verbal input induces activity in the TP-0 level of the verbal bank (Figure 11.14A). This activity immediately spreads to the visual bank along the “fast” connections and associatively activates the visual representation of a ball hitting the wall in the higher-order visual TPs. Depending on the temporal distribution of the Inter Stimulus Intervals (ISIs) between the visual and the verbal inputs during learning, the actual position of the visual representation of the event of hitting can vary.

An example of DETE’s processing dynamics during the understanding of past perfect tense is shown in Figure 11.14B.

11.8 Learning Homonyms

What happens when an alternate training regimen is used? Consider the example when during the learning of a word like “circle” (see section 11.2) the learning protocol is changed in the following way. Instead of using the training protocol described in Table 11.1. in which every visual/verbal training pair is of the type ([W:circle], [C:x, S:circle, Z:x, L:x, M:x,x]) -- call it type-1 pair, in this experiment, the type-1 pair was alternated with a pair of type-2. In a type-2 pair, the visual part instead of maintaining constant the circular shape feature and having different values for the rest of the feature dimensions (i.e. [C:x, S:circle, Z:x, L:x, M:x,x]), the red color feature is kept constant and the values of the remaining feature dimensions are varied (i.e. [C:red, S:x, Z:x, L:x, M:x,x]). This experiment effectively tests whether DETE can learn homonyms, since in some of the trials the word “circle” referred to an object with a circular shape independent of the rest of its visual feature, while in other trials the word “circle” referred to an object that was red and the rest of its features were irrelevant. Notice that, as expected, in some of the trials both the circular shape and the red color features were on.

As in the experiment in section 11.2., after the presentation of each pair (pairs of type-1 and type-2 were alternated) the learning was disabled, i.e. no update of the *ltm* was done and two tests were run: (1) *Verbal-to-visual test*: -- in this test DETE was given only the verbal input W:circle and the activity generated in the visual bank of the Long-Term declarative Memory was monitored (i.e. no external visual input was provided). The response was considered to be correct if, as a result of

the verbal input, sustained oscillations were induced in at least one neuron located in the proper areas, i.e. either in the area that represents circles in the shape memory bank, or in the area that represents red in the color visual memory bank. (2) *Visual-to-verbal test*: -- this test presented: (a) a novel instance of a circle with randomly chosen other features, followed by (b) a novel instance of a red object with all other features randomly chosen, and then (c) a red circle with the rest of the features randomly chosen. The activity generated in the verbal bank was monitored (i.e. no verbal input was provided). The response was considered correct when all gra-phonemes forming the word "circle" were generated in the correct order without intervening noise. Schematically this experiment can be described as follows:

TRAINING: ([W:circle], [C:x, S:circle, Z:x, L:x, M:x,x])
 ([W:circle], [C:red, S:x, Z:x, L:x, M:x,x])

Notice that one training trial consists of presentation of two pairs, one of type-1 and the other of type-2. Their order varied randomly from trial to trial. The total number of training trials was 100.

TESTING (verbal -> visual) ([W:circle], [C: ?,S: ?])

TESTING (visual -> verbal)

- a) ([W: ?], [C:x, S:circle, Z:x, L:x, M:x,x])
- b) ([W: ?], [C:red, S:x, Z:x, L:x, M:x,x])
- c) ([W: ?], [C:red, S:circle, Z:x, L:x, M:x,x])

The results are summarized in Table 11.9 which demonstrates that DETE can learn homonyms. In the given example, the word "circle" had two different meanings -- *circle* and *red*. The learning of both tasks was again rapid. In the *visual-to-verbal* task, when only one of the visual features (e.g., S:circle or C:red) was present in the visual input, DETE needed more trials (12 on average) to produce the verbal response (W:color) than if both visual features were present at the same time (1st correct response after trail 5). In the *verbal-to-visual* task which required that activity is generated in either the *red* or the *circle* field of the corresponding plane in order for the response to be considered correct, the learning was faster. Only after 4 to 5 trials DETE started to "envision" circular shape or a red color in response to the word "color". The number of traces to achieve a 100% correct performance are shown in parentheses.

| Verbal input | Visual input | | | | | 1st (100%)correct at trial # | |
|--------------|--------------|-------|------|-------|--------|------------------------------|----------|
| | Color | Shape | Size | Loctn | Motion | ver->vis | vis->ver |
| circle | * | ○ | * | * | * | 4(35) | 13(123) |
| circle | red | * | * | * | * | 5(41) | 11(112) |
| circle | red | ○ | * | * | * | not for 300 trials | 5(49) |

Table 11.9: Results of learning a homonym

An interesting observation on DETE's performance of the *verbal-to-visual* test was that during testing DETE exhibited priming effects. Namely, if the preceding visual/verbal training pair was of type-1, DETE responded with generating an activity pattern in the circle area of the shape memory bank, but did not induce activity in the color memory bank. Alternatively, if the last input pair was of type-2, DETE "imagined" red but did not "imagine" circle. For the total duration of the experiment (300 training trials) DETE never imagined both a circular shape and a red color feature

together in response to the verbal input "circle". Effectively, DETE disambiguated the meaning of this word on the basis of its immediate prior training experience.

11.9 Learning selected features of different languages

An interesting language feature which can demonstrate, on the one hand, DETE's ability to acquire more sophisticated grammatical rules (as compared to simple word order) and on the other hand to illustrate DETE's ability to handle grammatical features of languages different than English is *gender agreement*. English and Japanese do not have gender agreement, whereas Spanish does. For instance, in Spanish we say:

La pelota roja. (The red ball.)

El cuadro rojo. (The red square.)

To test DETE's ability to learn gender agreement we designed the following experiment. A set of 2-word noun phrases (NPs) was generated. Each NP had the form (Noun Adj). Two Spanish nouns were used (one feminine -- "pelota", and one masculine -- "cuadro") and a number of adjectives relevant to the blobs world. Only regular adjectives were chosen (i.e. use suffix -a for feminine, and suffix -o for masculine). Irregular adjectives were not used; e.g., verde (green). Most of the NPs were used as a training set and the rest in a testing set. Examples some of the adjectives used are:

| | | | | | |
|--------|---------------|----------------------|--------------------|------------------|-------------------|
| pelota | roja (red) | amarilla (yellow) | pequeña (small) | negra (black) | blanca (white) |
| cuadro | rojo | amarillo | pequeño | negro | blanco |

In humans, learning gender agreement in noun phrases is mostly based on the ability to make associations between words on the level of corresponding word fragments (e.g., between the endings -- suffixes, or the beginnings -- prefixes). Notice that for this particular task visual-to-verbal associations do not seem to be essential and are probably done only during the early stage of learning (when the individual words are learned). Later the mature speaker accomplishes this task as mostly a verbal-to-verbal task, i.e. making such associations becomes a kind of a verbal game. For instance, the child hears a new Adj associated with some Noun (e.g., "pelota negra") and without having even to know the meaning of the word negra(o) (black) it can successfully generate "cuadro negro". However, to produce this response, the child needs to be in the right context. First and foremost, before it can generate the response, it must be aware of the nature of the task (e.g., "combine Noun + Adj"). This can be done by receiving the verbal input "pelota negra" which serves as a context for the subsequent verbal input. Then it hears "cuadro" (or sees a square which can be just an outline without color). The representation of "cuadro" augments the context. It seems that humans have a built-in learning mechanism which has the natural tendency after a number of exposures to generate the response which contains the second word (negra) but modified by the context (cuadro) to (negro). Some of the possible ways to run such an experiment are presented below:

(1) Training: Learn individual NPs by associating each NP with its visual correspondence (e.g., P:red-ball & W-1:pelota, W-2:roja; or P:red-square & W-1:cuadro, W-2:rojo; etc.)

(2) Testing:

a) *Visual-to-verbal test*: Give verbal input **W-1:cuadro** and activate the black area of the Color Feature Plane (this represents the task or context). Observe if there is a verbal response (expect "negro"). Here the visual input provides the substance to the verbal response.

b) *Verbal-to-verbal test*: Immediately after an example has been given (e.g., **P:red-ball & W-1:pelota, W-2:roja**) (this represents the task or context), give only a novel Noun (e.g., **W-1:cuadro**). Observe if there is a verbal response (expect "rojo"). Here the previous verbal input provides the substance to the verbal response.

In this experiment we are interested in the verbal-to-verbal test. To be able to do this task the system must have the ability to associate two sequences (e.g., the phonemic sequence forming word-1 (**W-1**) with that forming **W-2** which are presented one after another (without intervening inputs). The first requirement for this to happen is that the representations of the two sequences must reside simultaneously in the memory for a while. Also, the sequences should be indexed in the memory as first and second (i.e. appropriate representations of word order should be generated). If these conditions are fulfilled, then the question is how the association of the corresponding suffixes can be done. One option is to align the corresponding parts of the representations of the two sequences in time (e.g., in this case their endings). Then they can be easily associated as co-occurring. Such alignment of parts could be done by giving a special status to the corresponding parts. Notice that the beginnings and the ends of words have by nature a special status since they are either preceded or followed by longer transition periods -- the pauses between words.

The critical module of DETE's architecture which allows it to learn this task is the Morphologic/Syntactic Procedural Memory (MSPM) and its integral part -- the Short Term Memory (STM) component of the Verbal Memory (VM) (see Figure 9.4). This memory module has almost all necessary functionality to accomplish this task; namely: a) the STM component of the VP serves as a temporal buffer for the first word (**W-1**) while the second word (**W-2**) is input. Since these words are different, they can coexist in memory which allows them to be associated. b) The input words are indexed by word order (**WO-1** and **WO-2**) in the Order Memory Bank (OMB) (see Figure 9.5). These indices are used during the testing mode. Specifically **WO-2** is used to generate the second word (**W-2**) by activating the most recently processed **W-2**. c) The representations of the words can effectively be aligned in time since the gra-phonemes of which they are composed are also ordered in the OMB (**PO-1, PO-2, ...** -- see section 9.2.2). d) The transition periods after each word give special status to the last gra-phoneme in each word. This is due to the fact that the representation of these gra-phonemic indices (and the gra-phonemes themselves) are active for the duration of the transition period between words. This feature allows the end-of-word gra-phonemes to be associated. This scheme will work if the lengths of **W-1** and **W-2** are the same in terms of number of gra-phonemes. However, in natural languages this is usually not the case. To overcome this difficulty the MSPM needs also another indexing component which marks the words not by the absolute order of the gra-phonemes (as it is currently done) but in addition marks the beginning-of-word (e.g., a prefix), word-stem, and end-of-word (e.g., a suffix). This can be done by normalizing the Phonemic Order (PO) representation of each word so that the first PO (**PO-1**) is always mapped to some unit (call it Word-Start -- **WS**) and the last PO (**PO-end**) of all words (notice that these differ in value) is mapped to another unit (call it Word-End -- **WE**).

11.10 Discussion

Our experience with DETE's performance on various language learning tasks so far has shown that it is a powerful and robust system. This is mostly due to its architectural design based on the idea of grounding symbols in perceptual experience.

From an engineering point of view, an important question (while designing a complex system such as DETE) is the complexity of the task which it is supposed to accomplish. This question is related to another question, namely what is the essential minimal architecture which can accomplish the task and what are the task-irrelevant components of the system? To estimate the complexity of the PGLA task from an information theoretic point of view is difficult, and while no attempt is made here to answer this question, one insight from the field of neurosciences might help to see it from another perspective. It seems that in the evolution of the brain, nature has allowed for almost any known "functional trick" to be incorporated in the nervous system. Such a "collection of tricks" provides the brain with flexibility and effectiveness in the processing of a variety of tasks. Therefore, looking for a minimally complex system to perform a given task seems not to be the turn that evolution has taken. Hence, it is worth separating the engineering aspects of designing an efficient system from the more basic questions of how does the brain do what it does.

There are several approaches to performance evaluation of a particular task. Sometimes it is sufficient to know that the system is able to perform the task. In other cases, especially when the system performance needs to be compared with the performance of other similar systems, it is important to design a set of performance criteria to be tested on the basis of particular measurements of system behavior. These criteria might be different for different tasks. In DETE such criteria were, for instance, speed of learning and accuracy of performance. While at this point no criteria have been designed to judge the performance of the system as a whole, specific criteria were used in the evaluation of the individual modules.

It is our observation that the overall performance of DETE during learning of various sub-tasks measured as speed and reliability of learning conforms well with a specific statistical model -- the model underlying the solution to the "occupancy problem" (Feller, 1957). The occupancy problem can be stated as follows: Consider n bins and a sequence of trials during each of which one and only one bin is visited. Assume also that the distribution of the visits over the bins is multinomial with equal probability for the visit of each bin. The question is: How does the probability that after trial x all bins have been visited at least once depends on the number of trials x ? The solution to this problem is given by the following formula (see Feller 1957, pp 91-92 for the derivation of this formula):

$$P(x) = \sum_{i=0}^n (-1)^i \frac{n!}{i! (n-i)!} \left(1 - \frac{i}{n}\right)^x \quad (11.1)$$

The bins in this statistical model correspond to the bits (predictrons) in the memory modules. The number of bins corresponds to the number of bits that represent a particular feature value. For instance, in the Motion Feature Plane, the number of bits that encode stationary (still) objects is $3 \times 3 = 9$. A "visit" of a bin corresponds to activation of the feature that this predictron encodes (e.g., red, square, etc.). The probability that after trial x all bins have been visited at least once gives us some estimate of the speed and reliability of learning.

For each of the experiments described in this chapter we know the number of bins (pixels) in the area of the particular feature plane to which a given concept (e.g., moves or stands) is mapped. For instance the “moves” area of the Motion Feature Plane is composed by $14 \times 14 - 3 \times 3 = 187$ bins (bits). The areas in the same feature plane where horizontal and vertical motions are mapped have equal sizes of $2 \times 5 \times 3 = 30$ bins (bits), etc. The results of each of the experiments described in this chapter were compared with the probability distribution obtained from the above-described statistical model in which the appropriate values for the number of bins were used. In general, the statistical model explains satisfactorily the speed of learning and the accuracy of the responses made by DETE. In other words, the learning curve for each experiment (after the necessary prerequisites have been learned) is similar to the curve that can be derived on the basis of this statistical model. To illustrate this claim, compare, for instance, the results of the experiment in which DETE learned the meaning of the word **W:stands** (see Table 11.2) with the predicted speed of learning on the basis of the statistical model. As shown in Figure 11.2, the area of the Motion Feature Plane (MFP) which represents stationary objects has the size $3 \times 3 = 9$ pixels (which map one-to-one to 9 predictrons in the Motion Feature Memory). Figure 11.15 shows the experimental learning curve (successful and non-successful trials) when DETE attempts to verbalize the word **W:stands** while looking at a stationary object. This data was obtained with the learning protocol described in section 11.1.2. As the plot shows, it took 22 trials for DETE to produce the first correct verbal response, and it took it 42 trials to start producing continuously correct responses.

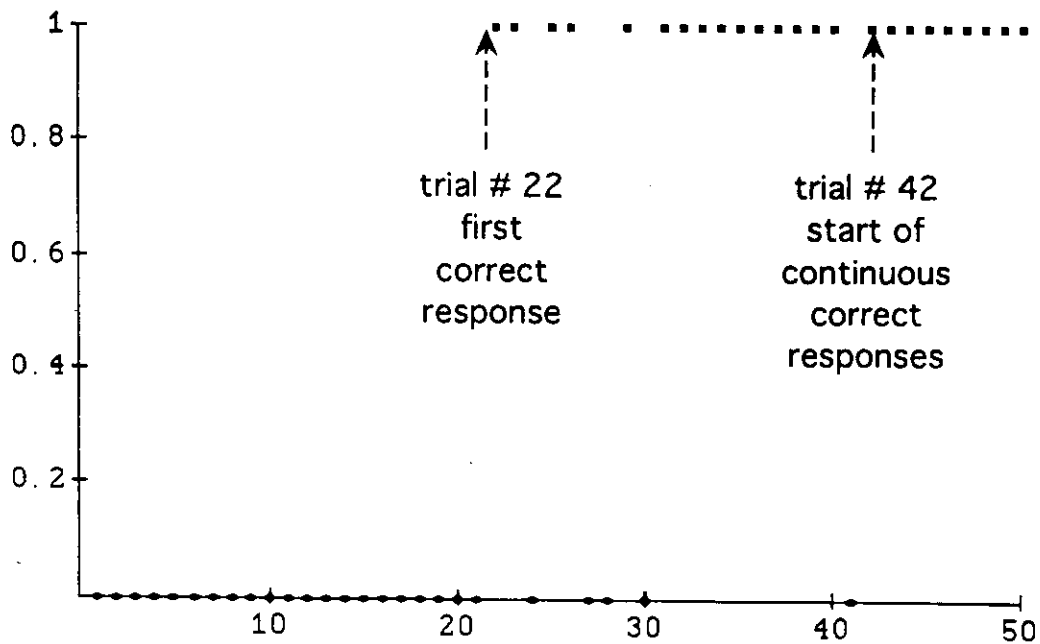


Figure 11.15: Learning curve of the word “stands”

The trial numbers are shown on the X-axis. Correct verbal responses are indicated by 1s on the Y-axis and incorrect responses by 0s (both 1s and 0s are shown as black dots at the corresponding Y level).

The theoretical curve (Figure 11.16) for this particular experiment was derived on the basis of formula (11.1) in which the value of n was set to 9. As can be seen even without applications of

any transformations (e.g., a cumulative sum) to the raw data (Figure 11.15), it is well fitted by the theoretical probability distribution shown in Figure 11.16. Similarly, good correspondences between the experimental observations and the theoretical predictions for the speed of learning were found for the most of the experiments. In the cases when the statistical model did not fit well the experimental data (e.g., learning of the word **W:moves** which was faster than predicted by the statistical model), the shapes of the curves (S-shape) were nevertheless the same. In the example of **W:moves** in particular, the disparity between the two curves can be explained by the influence of the remaining memory modules on the process of learning.

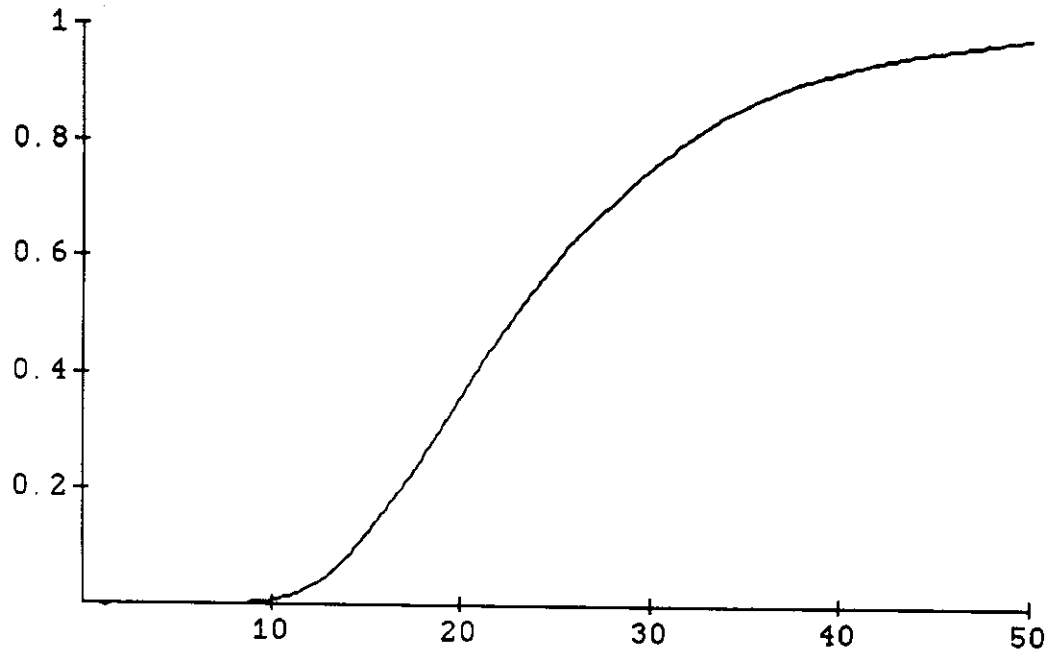


Figure 11.16: Example of a theoretical learning curve

The number of learning trials are shown on the X-axis. The probability for a correct response (a value between 0 and 1) is shown on the Y-axis.

PART IV

Comparison

Part IV compares DETE to other connectionist and symbolic models designed to perform tasks of language acquisition and sensory-motor integration. The KATAMIC sequential model is compared in terms of architecture and performance to other state-of-the-art connectionist models for sequence processing. Also, described are some neuropsychological aspects of humans' ability to acquire language. A review of the brain structures underlying such ability is presented. Parallels with DETE's architecture are drawn along the way. Finally, a summary of the current status of this research and directions for future work are outlined.

12 COMPARISON TO OTHER WORK

DETE is a multifaceted project. On the one hand, it is a novel attempt to handle a version of the blocks world task, as described by Winograd (Winograd, 1972; Winograd, 1973), and therefore its performance can be compared to other systems designed to handle this task. On the other hand, DETE is built from connectionist modules, some of which (e.g., the KATAMIC memory) are unique, while others have been already described in the literature (e.g., Winner Take All network, lateral inhibition). The newly invented modules can be compared to existing ones in terms of functionality and architecture, while the modules borrowed from the literature can be discussed in implementational terms. DETE is also capable of language processing up to a certain level of complexity and as such can be compared to other natural language processing systems.

12.1 Connectionist models of NLP

Some of the recently proposed connectionist models of NLP can be seen as an alternative or complement to symbolic models. However, PDP models of NLP fall short of being able to understand stories at the level achieved by the symbolic models. While it is possible to construct connectionist models that can effectively handle stories with multiple scripts (see e.g., Sharkey et al., 1986; Miikkulainen, 1990; Miikkulainen and Dyer, 1991) such systems have difficulties in handling unexpected or unusual situations. Such ability requires higher-level monitoring and control mechanisms. Translated to the traditional AI terminology, these systems lack the ability to do dynamic inferencing (i.e. construction of novel information structures from seemingly disjoint pieces of available information) (Touretzky, 1989). Dynamic inferencing is the basis of planning (construction, analysis, and execution) (Dolan, 1989; Lee, 1991; Dyer, 1990).

12.1.1 Localist connectionist models of NLP

The approach to representation taken in the localist models is to assign single nodes in the network to individual items (e.g., words or concepts) (van Gelder, 1989). The idea of building localist representations is not new. A somewhat more elaborated version of this idea is embodied in semantic networks (Quillian, 1967), which are widely used in classical symbolic systems. The difference between semantic and localist networks is that while the former have labeled arcs between the nodes, the latter have activation values which can be spread along weighted connections. There are two basic classes of localist networks. On the one hand are those that spread continuous values between nodes, i.e. "pure spreading activation networks" (Anderson, 1983; Cottrell and Small, 1983; Waltz and Pollack, 1985) to achieve retrieval, variable binding (Lange and Dyer, 1989; Shastri and Ajjanagadde, 1989b; Shastri and Ajjanagadde, 1989a) and inferencing. On the other hand, there are those that propagate discrete symbols, i.e. "marker passing networks" (Charniak, 1986; Fahlman, 1977; Hendler, 1988) to achieve the same functionality. Recently, attempts have been made to implement markers just in terms of activation (Lange and Dyer, 1989). Unlike DETE, the ROBIN system developed by Lange and Dyer is able to do goal/plan analysis by spreading activation (e.g., to select the correct meaning of the word "pot" in the sentence "Hid pot in dishwasher when saw the police coming."). However, ROBIN is not capable of learning.

Another attempt was made recently to replace individual nodes with patterns over ensembles of nodes (Sumida and Dyer, 1989).

The main advantages of the localist networks is that they offer knowledge-level parallelism (Sumida and Dyer, 1989). Since the activation values on the nodes are allowed to relax over time, which gives the user a handle over the temporal dynamics of these networks, one can interpret their behavior with respect to timing effects in cognitive tasks and semantic priming. The basic problem with such networks is that their structures are ad hoc and programming of such networks is tedious. They cannot learn from experiences and localist models are not robust with respect to network damage or choice of task. As it has been suggested (Sumida and Dyer, 1989; Dyer, 1990; Dyer, 1991) that the structure of the localist networks can be regarded as a metaphor for a system composed of numerous functionally differentiated distributed networks.

The `single_node_per_concept` approach taken in some localist networks seems absurd. However, the majority of Localist Connectionism (LC) is involved in constructing networks which replace single concept nodes with distributed assemblies over ensembles. These ensembles are further hooked up and trained -- intra- and internodal weights are modified. DETE takes basically the same approach within its feature memories, except that DETE's representation of individual concepts (e.g., "ball", or "hit") are spread over several feature memories.

12.1.2 Distributed connectionist models of NLP

Language processing tasks of various degrees of complexity have been explored by a number of researchers. Simple recurrent networks (or similar models) are used in most of these research efforts. Below is a short summary of the most prominent attempts made in this direction. Included are descriptions of the specific tasks, summary of the results and discussion of the shortcomings of these studies.

Jeffrey Elman

Elman (Elman, 1988; Elman, 1989b) applied his simple recurrent network (SRN) -- an extension of the model of Jordan (Jordan, 1986) to a number of linguistic tasks. In his earlier work he demonstrated that a SRN can learn the lexical category structure (e.g., the order of verbs, nouns, adjectives, etc.) which is implicit in a large corpus of language (Elman, 1989b). The success of his modeling effort was based on the fact that in natural languages there is a clear correlation between lexical category structure and word order. In other words, as Elman points out, not all classes of words may appear in any position -- certain classes tend to co-occur with other classes.

Elman trained his network on a large set (10,000) of two- and three-word sentences. The input stream did not have indications of where one sentence ends and the next begins. During testing on novel sentences, the network was not capable of predicting exactly the second and third word after it was given the first word of a sentence. Nevertheless, it had a tendency to activate output nodes that correspond to words belonging only to the plausible classes that can follow the class to which the first word in the sentence belongs. By examining the weights in the hidden layer (using hierarchical clustering analysis), Elman demonstrated that as a result of the training, the weight space had evolved clusters that correspond to different lexical categories (e.g., nouns and verbs). Furthermore, the verbs were subdivided by their argument requirements and the nouns were divided into animates and inanimates. The knowledge of class behavior was shown to be quite detailed.

DETE was tested on a similar task -- learning some simple syntactic rules like word order (see section 11.3.2). However, there are several differences between the tasks used by Elman's

network and DETE. (1) The body of language (three word sentences) was smaller (27 sentences). (2) During learning DETE did not associate consecutive pairs of words in the sentences, but associated visual images with verbal sequences. (3) The sentence boundaries were provided to DETE. (4) The testing was done not by giving the first word of a new sentence and expecting the system to complete it but instead a novel visual image was presented and DETE's verbal description of this image (object) was observed. (5) The most important difference perhaps is that Elman's model (as he acknowledges himself -- Elman 90 (Elman, 1990)) lacks semantic information. In other words, its structural information is not grounded in the real world.

Robert Allen

Sequential back-propagation networks (Allen, 1988; Allen and Riecken, 1988; Allen, 1987) (called *connectionist language users* -- *CLUEs*) were used to answer simple questions about objects in a microworld. In these experiments the input to the network is composed from both a sequential verbal part (words forming a sentence which demanded a verbal response about the visual scene to be generated) and a semantic representation part (the visual world which was kept static during the sentence processing). The network was taught to generate a simple yes/no response (output) regarding the correctness of the linguistic description with respect to the simultaneously presented semantic representation. Allen tested the network performance on a number of tasks: (1) generalization (i.e. transfer of the ability to generate answers to patterns which have not been seen before) (Allen, 1988), (2) learning of pronoun reference for objects in some simple cases (recency and semantic priming) (Allen and Riecken, 1988). In the most complex experiment Allen's system performs disambiguation of anaphora (i.e. finding the correct referent for a pronoun which can have several different referents in the sentence). While Allen demonstrates that his system is capable of generalization and can learn some basic linguistic skills such as simple pronoun reference, one problem with the chosen network architecture is its slow learning rate (about 1.5 million epochs are necessary before convergence is reached). This slow learning process is typical for networks using the error back-propagation learning algorithm and justifiably raises the question of neural plausibility. Another shortcoming of this system is that it cannot be easily scaled up. This limits the robustness of the system and does not allow the integration of multiple simple tasks.

Mark St. John and James McClelland

St. John and McClelland (St. John, 1990; St. John and McClelland, 1989) used a cue-based constraint satisfaction algorithm to process a number of prototypical stories. The model takes a story as input and it learned to answer questions. Story comprehension was learned by experience. Some of the specific tasks on which this model was tested include pronoun resolution, inferencing, revision of on-going interpretation, and learning. The network used in these experiments was also a variation of Elman's model. The major shortcomings of this model were: (1) The predicate roles were chosen arbitrary, i.e. they were not grounded in any physical reality. (2) The representation of each proposition in terms of different roles such as "agent", "predicate", "patient", or "recipient" are hand-coded. In other words, there is no mechanism for parsing the sentences into the chosen representation. In essence, this system faces the same problem as that faced by classical symbolic systems, namely that of building the representation for each entry. (3) The learning was extremely slow. This is a problem with all systems that use simple recurrent networks with a back-propagation learning mechanism.

In its current implementation DETE is not capable of dealing with verbal input of the same level of complexity as St. John's model. The main reason is that DETE does not explicitly allocate sets

of processing units for the various roles such as “agent”, “predicate”, “patient”, or “recipient”. There are at least two possible ways how to bring DETE to this level of performance. The straightforward way is to allow for such type of unit allocation. Actually, this solution might not be completely at hoc. (There is anecdotal neurophysiological evidence that selective ablation of parietal association cortex can lead to very specific impairments of the semantic processing abilities.) The second way is to invent a completely new representation which can be based for instance on the temporal properties of the memory.

Risto Miikkulainen

A large scale natural language processing system called DISCERN was constructed by Miikkulainen (Miikkulainen, 1990). DISCERN uses a recurrent back-propagation mechanism (FGREP) to form distributed representations. These representations are processed by a hierarchy of memory modules implemented as Kohonen type feature maps. DISCERN reads short stereotypical stories (e.g., about restaurant visits), generates expanded paraphrases of these narratives and answers questions about them. The properties of the neural networks used in this system allowed for the emergence of abilities such as inferring of unmentioned events and unspecified role fillers. While DISCERN does not have the ability to do one-shot learning, which is necessary to store an event in a Short-Term Memory (e.g., a phone number), it can (after some iterations) store a declarative representation of some episode such as a specific visit to a restaurant. (Notice that such an episode has a much longer duration which allows for an incremental construction of its representation in memory.) In other words, DISCERN can store: AGENT = John, SCRIPT = restaurant, LOCATION = Malibu seafood, etc. Thus, DISCERN has an event memory that it can refer to. It can also answer questions about its content.

Currently DETE cannot answer questions about past events, such as “What green object went north and hit the wall ?” where the event was something way back in the past. To be able to do this task DETE needs additional architectural modules to represent a hierarchy of temporal units. For instance, seconds, minutes, hours, days, months, years, centuries, etc. Also, it needs structures that represent parts of temporal units, e.g., morning, noon, evening, January, February, ..., quarter, semester, etc.

Irving Biederman and John Hummel

Biederman and Hummel have developed a theory and a computational model of human image understanding. Their system represents objects as a spatial arrangement of a limited number of volumetric primitives (e.g., block, cylinder, etc.) (Biederman, 1987; Hummel and Biederman, 1990). It does not have any language processing ability. Like DETE, Hummel and Biederman’s model uses phase differences of oscillations to accomplish visual binding. However, unlike DETE, their system is also capable of representing complex objects.

Rumelhart & McClelland

Several connectionist models have been proposed during the past few years that simulate different aspects of the surface dynamics (morphology and phonology) of past tense acquisition as observed empirically in psychological studies of children. In a widely publicized paper Rumelhart and McClelland have shown that a simple neural network can exhibit to a remarkable degree the characteristics of young children learning the morphology of the past tense in English (Rumelhart and McClelland, 1987).

Unlike Rumelhart & McClelland's model, DETE addresses specifically this issue of the acquisition of the semantics of past tense in some simple and non-exhaustive cases. For instance, DETE assumes that there are different verbal tokens for the different tenses of each verb. (In the current implementation DETE is not concerned with morphological issues of verb tenses.) An important question is whether DETE needs to have understood (learned) the present tense in order to be able to learn the past or future tense, or can they be learned independently? At first glance it seems that morphology helps (e.g., knowing the meaning of "bounce" may help understanding "bounced"). However, the verb stem is not always preserved in the past tense form (e.g., "go" and "went" refer to the present and past tense of one and the same action but have quite different morphology). Also, there are verbs that have the same morphological form in the present and past tenses (e.g., hit, hit). Here some contextual information should help.

12.2 Tough problems for neural network models of NLP

The non-symbolic nature of neural network technology raises a number of issues when used for construction of Natural Language Processing systems. Some of the most difficult are the choice of representation and the related problems of modularity, role binding, and sequence processing. In this section, the methods employed by DETE for dealing with these problems are described and discussed in the light of related research.

12.2.1 Formation of distributed representations

The use of distributed representations, instead of localist representations, has a number of advantages -- the most significant of which are robustness and graceful degradation of performance in the face of damage. However, a major issue is that of choice of specific representations. In other words, where do distributed representations come from and what do they stand for, i.e. do they have some internal meaning?

One of the most widely used approaches for forming distributed representations is to encode semantic features. This "semantic microfeature" approach has been used, for instance, in the case-role assignment task (Hinton, 1981; McClelland and Kawamoto, 1986). In this approach concepts are classified along a set of dimensions predetermined by the designer (e.g., animate/inanimate, male/female, etc.) and one or more units are assigned to each feature. During processing, a particular classification of an input corresponds to a pattern of activity over a subset of the units. This approach is analogous to the way the nervous system processes information by means of dedicated (labeled) lines. Such labelled lines can be found in almost all sensory systems (Kandel and Schwartz, 1985) and form the basis of topography-preserving maps (e.g., color is encoded in the retina using three different types of labeled lines, for red, green, and blue). However, such labeled lines are typical for the lower-level sensory processing and functional analogues of such lines in higher cognitive processing (while theoretically possible) have not been demonstrated experimentally.

Another approach is to develop internal representations in intermediate layers in the neural network hierarchy. A good example of this approach is the network used by Hinton to learn family-tree relations (Hinton, 1986). Another example is the simple recurrent network used by Elman to predict next word in a sequence of words (Elman, 1989a; Elman, 1990). This approach, however, does not address the issue of encoding input/output representations.

Another approach to the development of representations was proposed by Miikkulainen (Miikkulainen and Dyer, 1987; Miikkulainen and Dyer, 1989; Miikkulainen and Dyer, 1988; Miikkulainen and Dyer, 1991). His FGREP network develops representations for the symbols automatically while the network is learning the processing task. These representations are global (both input & output) and are stored in a separate network (called the lexicon). The main advantage of such representations is that they encode the properties of the input that are most critical for the task since they are adapted according to the back-propagation error signal.

Yet another method -- "symbol recirculation" (Dyer, 1990) was used by Lee et al. (Lee et al., 1990) for forming lexical patterns (XRAAMs). Lee stored hidden layers of Pollack's RAAM (Pollack, 1990) into a lexicon formed by a recirculation.

In DETE, the representations of the words in the language which are stored in the memory have two components: (1) a gra-phonemic representation of the word itself which is associated with (2) a distributed representation of the corresponding visual experience which the particular word refers to. In this sense, the language which DETE acquires is completely grounded in visual experiences. One might argue that the approach taken by DETE suffers from the same problems as the other PDP approaches to representation, namely choosing the set of the visual features is equivalent to making a choice of semantic features and such a choice is arbitrary. While on the surface such an argument appears to be justified, the substantial contribution which DETE makes in choosing the visual features comes from the fact that: (1) They are natural and there is abundant evidence that, namely these features are used for grounding of most of our early language (for references on the acquisition of early language in blind and sighted children see (Dunlea, 1989)). (2) These representations provide the basis for the formation of higher-order mental representations (Lakoff and Johnson, 1980; Lakoff, 1987; Lakoff, 1989).

12.2.2 Types vs tokens

Symbolic models of cognition in general keep separate the information about symbols (e.g. their properties and relations) from the information about the individual instances of a specific symbol. In other words, they keep the *types* separate from the *tokens*. In the field of neural networks, various approaches have been proposed to merge *types* and *tokens*. For instance, in analyzing his experiments on learning lexical categories, Elman found that the network had encoded every *token* by a distinct representation which reflects the context in which the *token* appears. The representation of *type* was done by clustering all *tokens* of a particular class closer in space while keeping the distances between the different types large. It was also observed that the *tokens* of a particular *type* were not randomly distributed but there were sub-clusters which correspond to the different linguistic contexts (e.g. "boy" in a subject position vs "boy" in an object position) in which a *token* may appear. This feature of the representation corresponds to a grammatical-role distinction which cuts across lexical items.

DETE introduces a novel approach to the representation of *types* and *tokens*. A *type* in DETE is represented as a memory trace (*lm* pattern in the LTM) while a *token* is represented as an activity (i.e. an oscillation pattern) in the network which has been generated as a result of the memory trace that represents the *type*. Several tokens can be instantiated in DETE at the same time. They are represented as phase shifted oscillations over the same (if the *tokens* are identical) or overlapping (if the *tokens* are only similar) parts of network.

12.2.3 Role binding

One of the hardest problems for connectionist models is that of role binding, or more generally, variable binding. One approach to the role-binding problem is the use of conjunctive coding (Touretzky, 1987). In this approach the role filler pairs are represented conjunctively in a matrix of units. A generalization of conjunctive coding is to represent bindings as a tensor product of the representation vectors of the role and the filler (Smolensky, 1987; Dolan, 1989; Dolan and Smolensky, 1989). Other solutions to the binding problem are based on building dynamic connections (Feldman, 1982), application of parallel constraint satisfaction (Touretzky and Hinton, 1988), signatures (Lange and Dyer, 1989), and position specific encoding (Barnden, 1989).

An approach in some ways similar to ours was taken by Shastri and Ajjanagadde (Shastri and Ajjanagadde, 1989b; Shastri and Ajjanagadde, 1989a). The authors designed a continuous rule-based reasoning system based on a localist connectionist model. Similarly to DETE they use phase-locking of oscillations. However, while DETE uses phase-locking to assemble objects from their features, their system uses phase-locking of oscillations to encode fillers and arguments and to propagate bindings. Shastri's system can represent predicates such as "give", "own" and "can-sell" as collections of arguments (or roles). For instance, "give" is represented as a collection of argument bindings which include: *giver*, *recipient*, and *give-object*. The argument bindings are established dynamically in the system (e.g., *giver* = John, *recipient* = Mary, *give-object* = book) when it is 'told' that "John gave Mary Book1". Given this type of representation and dynamics the system can do dynamic inferencing. For instance, from the input "John gave Mary Book1" it can infer that Mary owns the book.

Currently DETE cannot do inferencing of the type done by Shastri's system since it does not represent causal relations of the sort that Shastri can encode in his network. However, unlike DETE this system is not capable of acquisition of language. Its representations (connectivity patterns and synaptic weights) were hand-crafted and fixed. Also, it does not discriminate between memory types (e.g., STM vs LTM). The architecture uses several types of special purpose nodes to perform basic logical functions (e.g., τ -and nodes, τ -or nodes, ρ -btu nodes, etc.) for which there is no evidence that they can be directly mapped to neurons in the nervous system.

Currently DETE does not face the role binding problem at the same level as the script-based processors since it operates in a simpler world (the Blobs world). DETE does not represent explicitly case roles such as agent, act, recipient, patient, location (e.g., "The ball (**agent**) hit (**act**) the triangle (**recipient**) in the corner (**location**) hard (**modifier**)."). However, to be able to answer questions of the sort "Who left a big tip at Malibu seafood?" DETE would need special visual modules that deal with images of moving, complex agents (humans/robots), actions (e.g., ungrasping of round object above a table = leaving a tip), and abstract concepts (e.g., transfer of possession of the coin by voluntary agreement). These visual modules might be implemented as some sort of higher order maps. They would serve the purpose of case roles but will be more robust than case roles in AI and other connectionist systems.

While currently DETE does not have case roles, it has structures that function as "visual roles". These visual roles are represented by the topographic maps of predictrons to which various visual features are mapped. For instance there is a "shape role", a "color role", etc. The fillers of the visual roles (e.g., "circle", "red", etc.) vary in time. DETE can answer questions about the fillers. For instance: **Q1:** What color is the ball? **A1:** Red. **Q2:** What size is the triangle? **A2:** Small.

The visual role binding in DETE is done in the temporal domain and the focus of attention (FA) mechanism operating over the current active context is used to keep things together.

12.3 Associative memory models

Associative models of memory have been studied in psychology, neuroscience and computer science in many forms (Anderson and Bower, 1973). Linear matrix memory models are described by Kohonen (Kohonen, 1972; Kohonen, 1977), Anderson (Anderson, 1970; Anderson, 1972), and Gardner-Medwin (Gardner-Medwin, 1976).

As was pointed out by von der Malsburg (von der Malsburg, 1981; von der Malsburg, 1983; von der Malsburg, 1987) and others, a major weakness of simple associative memories as models for cognitive processing is their low power of generalization. This is a result of the fact that associative memories treat patterns as monolithic wholes. In other words, they glue all pairs of features in a pattern together and during recall, in general, they recover either the whole pattern or nothing of it. In order to successfully generalize, a system must be able to decompose complex patterns into functional components that can be later used in new combinations (Feldman, 1988). The introduction of a selective attention mechanism in DETE, which is coupled with a sequential associative memory, allows DETE to overcome the limitations of simple associative memories by being able to segment out one object from another (or background) in a scene containing several objects.

12.4 Self-organizing feature maps

12.4.1 Kohonen's feature maps

Self-organizing feature maps (Kohonen, 1982b; Kohonen, 1984) is a neural mechanism for clustering high-dimensional data into a lower dimensional space. For instance, a *2-D topological feature map* can implement a topology-preserving mapping from a higher dimensional input space (e.g., 3-D) onto a 2-D output space. The map consists of an array of neural elements. Each data point (e.g., a 3-D vector) in the input space is projected via a set of three weights (one for each dimension of the input) to each unit in the map (Figure 12.1). The map responds to any input with a localized pattern of activity over the units. The response of each unit in the map is proportional to the similarity of the input vector and the unit's weight vector. The unit with the largest output value is usually considered as the image of the input vector on the map.

During the process of self organization, the weight vectors are tuned to specific items of the input space so that topological relations are retained. This means roughly that nearby vectors in the input space are mapped onto nearby units in the map. This is a very useful property, since the complex similarity relationships of the high-dimensional input space become visible on the map (Miikkulainen, 1990).

A significant property of Kohonen's mechanism is that the whole area of the map is eventually covered by data. For example, a map of hierarchical data essentially displays the minimal spanning tree of the data items, curved to fill in the whole area of the map (Kohonen, 1982b). The map consists of subareas, which are continuous and where nearby units stand for nearby items in the input space. The most frequent areas of the input space are represented in greater detail, i.e. more units are allocated to represent these inputs. A neurobiological analogue of this characteristics of the map can be easily demonstrated. For instance, in the developing motor and somatosensory cortices,

areas that correspond to body parts that are used more often (e.g., the hand and the tongue) have larger representations (Kandel and Schwartz, 1985). However, the boundaries of these continuous areas are not marked on the map.

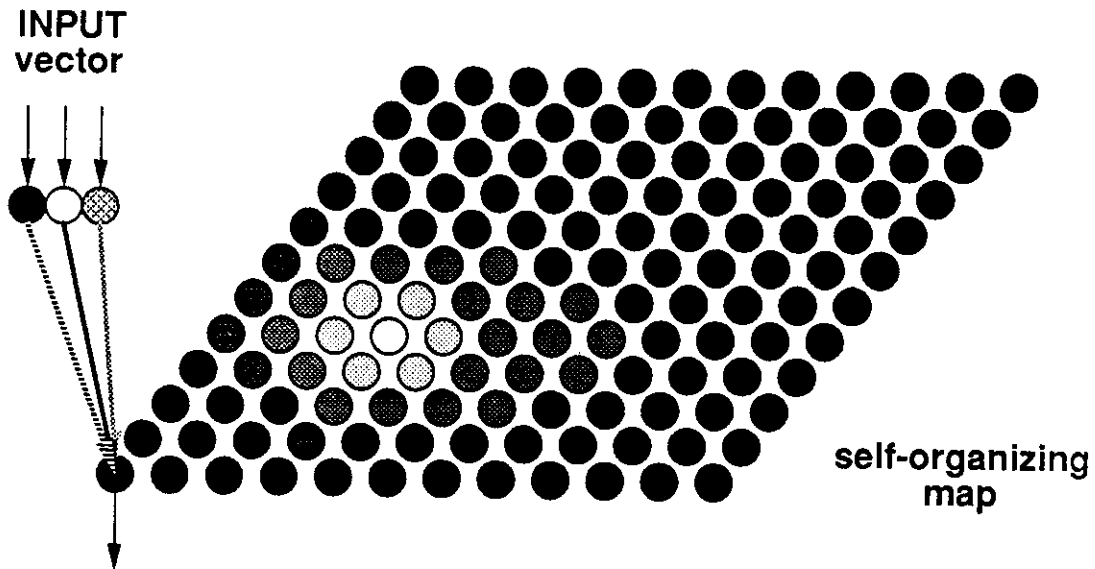


Figure 12.1: Kohonen's feature map

A data point in a 3-D input space is projected via modifiable weights (differently shaded arrows) to each of the processing units in the 2-D feature map. (The connections of the input vector to only one of the units are shown, and similar connections to the rest of the units are assumed.) Different levels of gray on the map are used to represent the response of the units to this input. The bright end of the scale corresponds to strong responses while the dark end stands for weak responses. The response of one of the units (white) is strongest -- this is the image of the input vector in the feature map. Neighboring units also show strong responses which diminish with distance. The lateral connectivity of the processing elements is not shown on this figure.

The organization of the map, i.e. the development of the weight vectors, is formed in an unsupervised learning process (Kohonen, 1982a; Ritter and Schulten, 1988). During each step of this process the map adapts in two ways: (1) the weight vectors become better approximations of the input vectors, and (2) neighboring weight vectors become more similar. Together these two adaptation processes eventually force the weight vectors to become an ordered map of the input space.

Each adaptation step consists of three tasks:

(1) Measuring the similarity of the input vector and the unit's weight vector. One method for measuring the similarity is with a scalar product of the input vector and the weight vector, i.e. by computing a weighted sum of the input components (Kohonen, 1982a; Miikkulainen, 1987).

(2) Determining the adapting neighborhood. This can be achieved by focusing the initial response of the map through lateral inhibition spread via non-modifiable lateral connections with pre-determined weight distribution. A weight distribution with the form of a "Mexican hat", i.e.

difference of Gaussians (DOG) has been successfully used for this purpose. It also has the advantage of being biologically plausible (Hartline, 1949; Ratliff et al., 1966).

(3) Changing the weights within this neighborhood. Conventionally, the weights change in proportion to the Euclidian distance of the input vector and the weight vector.

The visual feature maps (VFMs) used in DETE are not of the same type as Kohonen's feature maps. DETE's VFMs do not self-organize with experience but are artificially designed. Also the features which are mapped in these VFMs are extracted by procedural modules from a simulated visual world (observed via a retina). The particular feature choices were not arbitrary but were intended to capture the choices made in the human brain during evolution.

12.4.2 von der Malsburg's "dynamic link architecture"

A "dynamic link architecture" for graph matching in two layer neural networks (containing feature layer and a pattern layer) was developed by von der Malsburg and his co-workers (von der Malsburg 81, 88a; Bienenstock and von der Malsburg 87) (von der Malsburg, 1981; von der Malsburg and Singer, 1988; Bienenstock and von der Malsburg, 1987). This neural network architecture augments traditional neural nets by the ability to flexibly encode syntactic bindings between data atoms (neurons) and to organize complex binding structures efficiently. The basic approach to the representation of bindings is to have neurons synchronize their temporal activity to express binding between them. This synchronization is produced by excitatory connections between neurons in the two layers. The organizational process that produces appropriate binding structures consists of a feed-back loop between signal correlation and rapidly modifying connections. This feed-back loop is positive, in the sense that strong correlations lead to connections of increased strength, and strong connections lead to correlations. This process of self-organization naturally favors certain ordered connectivity patterns. One particular kind of organized connectivity pattern consists of two-dimensional locally connected networks. These are ideal for the representation of objects.

An important natural organization process in the dynamic link architecture is the storage and retrieval of network patterns. Another useful process natural to the architecture is labeled graph matching: The ability of the system to "discover" that for an active connectivity pattern there exists another, stored connectivity pattern which is label-isomorphic to the first, to activate that isomorphic connectivity pattern and to realize the isomorphism by an activated linkage between the two. Discussions on deriving excitation and inhibition dynamics within the feature and pattern layers and between these layers for the labeled graph matching problem are given in (von der Malsburg 88b) (von der Malsburg, 1988).

The main difference between DETE's feature planes and the feature maps proposed by von der Malsburg as well as those proposed by Kohonen is that while the latter are dynamical self-organizing systems, the ones used in DETE are hard-wired representations generated by procedural feature extractors. This approach has been sufficient for the current stage of DETE's development. It is foreseeable, however, that in a more complex visual world it will be necessary to incorporate some sort of self-organizing feature maps, such as those described above, that will be capable of learning to automatically extract task-relevant features. Such maps might be used for creation of both the visual and verbal representations.

12.5 Sequence processing

Human information interactions with the environment -- such as perception of spoken language and generation of speech, writing and reading text, singing as well as visual perception of ever changing scenes in the world -- are sequential in nature. On the other hand, the brain mechanisms involved in the processing of these phenomena are intrinsically parallel and distributed. The design of systems with externally sequential behavior produced by parallel architectures is a challenging problem. Models of sequential memories have been proposed by a number of investigators. The simplest way to model sequential recall of patterns is to store the patterns sequentially and to recall them sequentially. This method is, unfortunately, not sufficient for recall of sequences, such as songs, as was shown by Lashley in a paper on serial order (Lashley, 1951).

One of the first attempts to demonstrate a neural mechanism capable of storing and retrieval of sequences was done by David Marr (Marr, 1969). He suggested that the cerebellum learns to perform motor skills. In his model the information about movements and the contexts in which particular movements should occur was learned by cerebellar Purkinje cells. Marr proposed a detailed mathematical model of the cerebellum and used it to explain how conditional reflexes can be learned in this structure as well as how it can serve in the initiation of movements.

More recently, with the new wave of interest in neural networks, recurrent PDP networks capable of performing the task of sequence completion are gaining much attention (Rumelhart et al., 1986). The general procedure for a recurrent network is that a sequence is presented to a running system while it performs a number of iterations. The output of certain units are compared to the target for that unit at a priori specified time points and error signals are generated. Each of these error signals is then passed back through the network for the same number of iterations as in the forward pass and weight changes are computed at each iteration. The sum of all such changes for any given weight is saved. This procedure poses some memory problems since weights on the hidden and output layer synapses must encode information about all patterns in the sequence through which the network has processed during the forward pass. Rumelhart et al. used a simple fully connected PDP network (5 input, 30 hidden, and 3 output nodes) to store a set of 25 sequences of 6 steps each. The network was supposed to learn to predict the last 4 steps of each sequence on the basis of the first two which uniquely identified the rest. For this task the network required thousands of sweeps through the training examples and the performance was not perfect.

Grossberg and Stone proposed a hierarchical matched filter avalanche structure to store time-delayed patterns (Grossberg, 1969). The *avalanche* network was capable of generating an arbitrary space-time pattern (Figure 12.2). It can be viewed as composed of a series of outstars. An outstar is a hetero-associative memory module designed to carry out spatial pattern learning. The *avalanche* model did not have autonomous learning abilities. In general the order of activation of the outstars, as well as the spatial patterns themselves, needed to be learned. This task was left to other networks, like auto-associative memories, as was described in the theory of serial learning (Grossberg and Pepe, 1970).

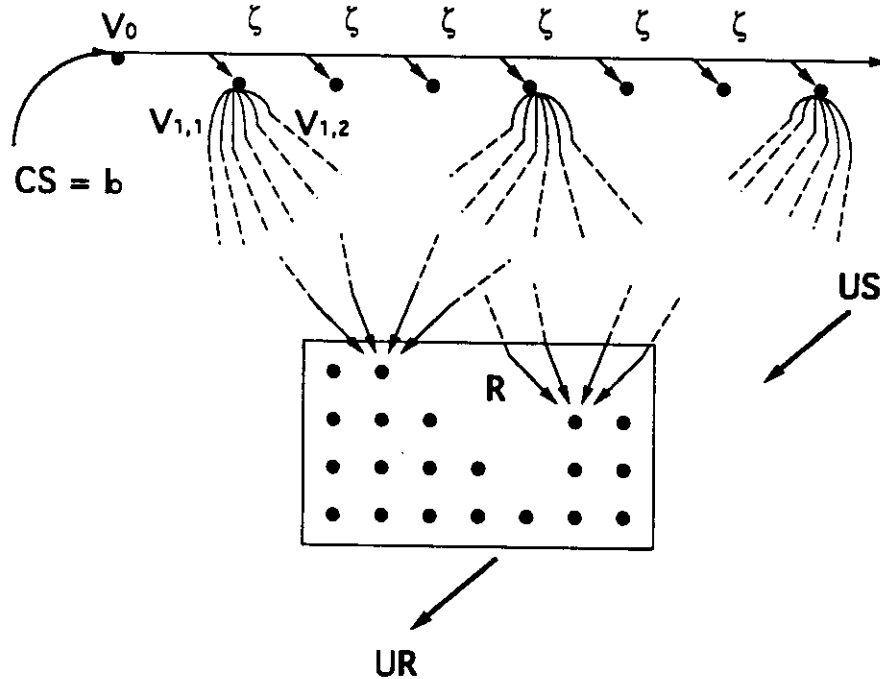


Figure 12.2: Grossberg's avalanche model

A) a probabilistic graph of an outstar. v_1 is a *source* vertex, $v_i, i \neq 1$ are *sink* vertices. B) an *outstar avalanche* -- series of outstars $v_{k1} \dots v_{kK}, k = 1, 2, \dots, K(\xi, T)$ whose source vertices are excited successively every ξ time units via axon collaterals from v_0 . If the k^{th} outstar could learn the k^{th} spatial approximation to a space-time pattern, then a single control vertex v_0 could activate an arbitrary complicated space-time pattern by successively activating each source v_{k1} . (Adapted from Grossberg, 1969.)

Another architecture for a connectionist sequential machine was proposed by Jordan (Jordan, 1986). This was a three layer recurrent network which can be regarded as a generalization of a content-addressable memory (Hopfield, 1982) in which the memories correspond to cycles or other dynamic trajectories rather than static points. The input to this network is composed of a "plan" vector which is kept constant during the retrieval of a given sequence and a state vector which changes as a function of the previous state and output (Figure 12.3). The network learns a sequence of output vectors using a back-propagation (BP) learning rule. The performance of such a network was analyzed in comparison with a sparse distributed memory (SDM) by Keeler (Keeler, 1988) who found that its memory capacity is lower for similar sized networks. Another weakness is that learning and sequence production are done during separate phases. Such a setup does not allow the network to be used in a situation when a real time adaptation to the environment is required. Also, the use of a BP learning mechanism requires hundreds of epochs before the error correction procedure leads to convergence. Finally, the possibility that such learning rule as error back-propagation can be found in the brain is very slim.

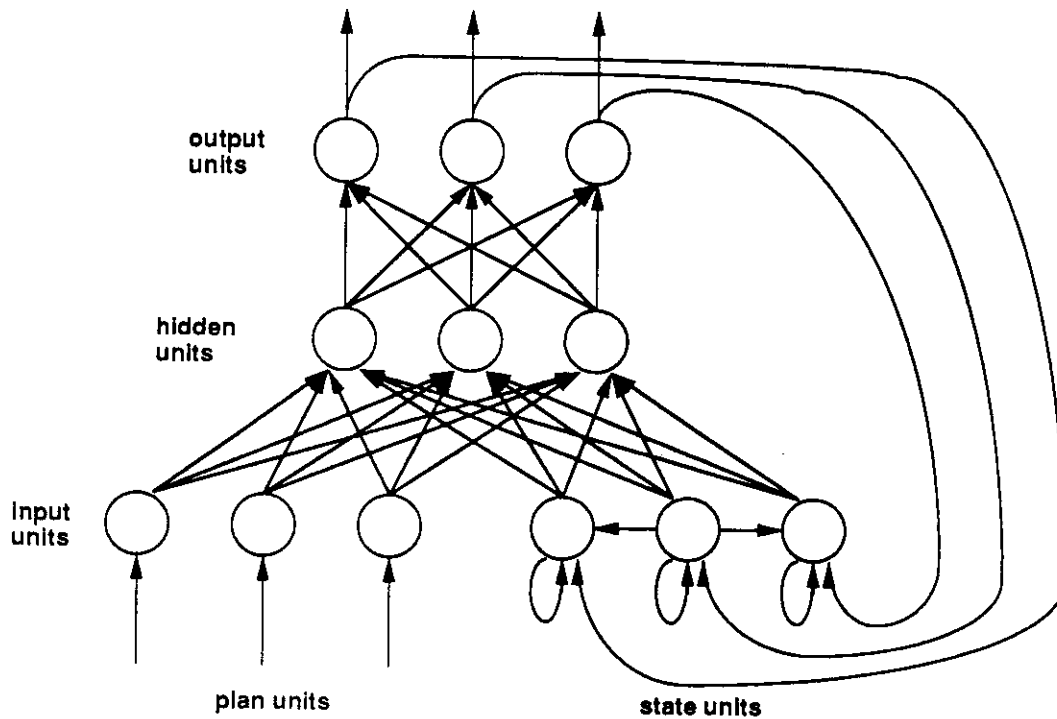


Figure 12.3: Jordan's network

Schematic drawing of Jordan's three layer recurrent network. Three input and three state units are shown in the input layer. All of these units are fully connected via modifiable synapses to three hidden units which in turn are fully connected again via modifiable synapses to the three output units. The output layer units feed back (in a one-to-one mapping) via non-modifiable synapses to the state units which are auto and mutually connected via inhibitory connections. (Adapted from Jordan, 1986.)

12.5.1 Comparison of KATAMIC and Kanerva's SDM models

A very interesting and promising approach to the problem of sequence storage and retrieval was taken by Pentti Kanerva (Kanerva, 1984; Kanerva, 1988) in his work on Sparse Distributed Memory (SDM). SDM is a massively parallel architecture. The model is very similar in mathematical terms to the models of the cerebellum proposed by Marr (Marr, 1969) and Albus (Albus, 1971). In brief, SDM is an associative, random-access memory that uses very large patterns (hundreds to thousands of bits long) as both addresses and data. When writing a pattern (n-bit vector of 1 & -1 elements) at an address in the memory, the pattern is added to existing information at each of many nearby memory locations within a given Hamming distance. Each storage location in the SDM is a set of n counters. When reading from an address in the memory, information stored at nearby memory locations (within the same Hamming sphere) is pooled (the n bits of each location added in parallel) and thresholded for output (output in the i-th bit = 1 if the sum of i-th bits > 0, and = -1 otherwise). The SDM is of interest due to its inherent ability to store sequences of patterns and to "predict" (recall) the remaining portion of a sequence when prompted by an earlier segment of the sequence. To store such sequences, SDM uses information about its

immediate past. This information is stored in SDM using “folds”. A k-fold system contain the time history of the k-1 previous time steps (Figure 12.4).

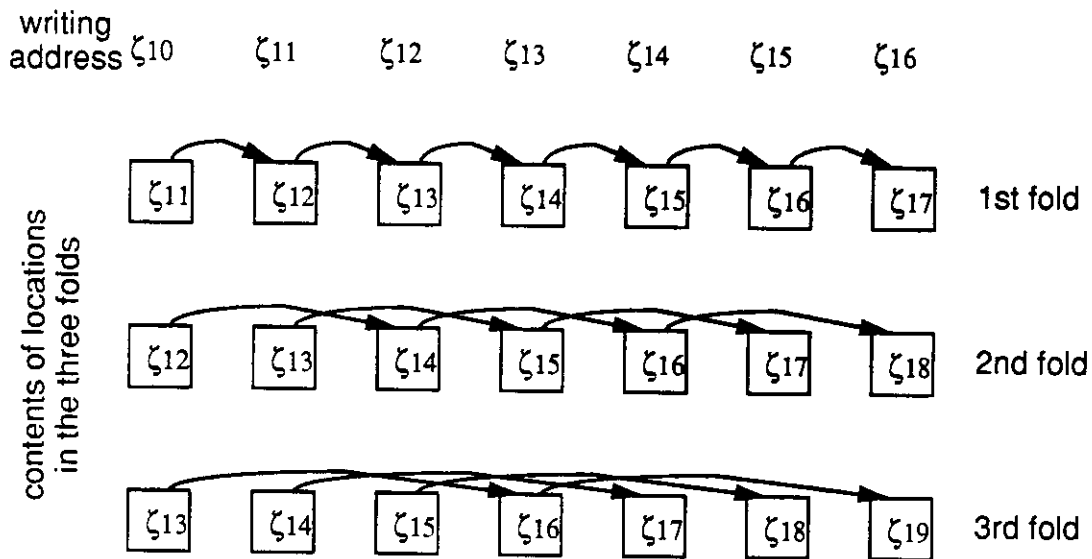


Figure 12.4: Kanerva's sequential SDM

Transitions in a modified Sparse Distributed Memory -- a three-fold memory for sequences.
(Adapted from Kanerva, 1988.)

While the SDM is mathematically very elegant and computationally powerful, the proposed neural plausibility of the model is questionable. Kanerva has investigated a possible mapping of the SDM on the mammalian cerebellum. If adequate, such a mapping would be very important and can lead to useful results for controlling robots because the cerebellum coordinates a myriad of sensory inputs and motor outputs with far more sophistication than is possible with present-day man-made computers. However, the assumption that Purkinje cells in the cerebellum can serve as address decoders is highly questionable, since the information which they receive through mossi fibers and climbing fibers can hardly be encoding addresses.

A limitation of Kanerva's algorithm is that it does not allow retrieval of sequences at different rates and sequences with missing steps. In a simple extension on Kanerva's model, Keeler has proposed the addition of time-delay terms with weights which are “smeared out” in time to achieve this functionality (Keeler, 1988).

The KATAMIC memory, which was conceived independently of Kanerva's SDM, has a number of similarities to it. Here the former is compared to the latter in terms of architecture and the essential differences are pointed out. The mathematical tools used to assess SMD's capacity can be applied to the KATAMIC model.

- (1) The number of predictrons corresponds to the dimensionality of the n-bit vector stored in each location of the SDM.
- (2) The spatial time constant T_s corresponds to the radius of the Hamming sphere in the SDM.

(3) The values of the fields in the n-bit input vector (0,1) in the KATAMIC memory map to (-1,1) in SDM.

(4) The process of computing the difference between the *p-ltm* & *n-ltm* is analogous to the process of reading from SDM (thresholding of the pooled vector). Notice, however, that there is no analog of the dot-product between the *stm* and the (p-n)-*ltm* in SDM.

(5) The *stm* in the predictrons does not have a direct analog in SDM but functionally can be mapped to the folds. The temporal decay constant T_t of the *stm* maps to the number of folds in SDM. Keeler's "time-smearing" enhancement of the SDM brings it even closer to the KATAMIC memory.

12.5.2 Comparison of KATAMIC and Elman's SRN model

While the efforts in inventing novel and more powerful neural network models have been extensive, little has been done in terms of comparative assessment of their strengths and weaknesses. For instance, only few studies have been done to compare AI and connectionist approaches with learning from examples (Fisher and McKusick, 1989; Mooney et al., 1989). The number of comparative studies between neural net models is also limited (Keeler, 1988).

Sequence processing networks form a specific class of connectionist models. In general these networks fall into two categories -- synchronous updating networks (SUNs) and asynchronous updating networks (AUNs) (D'Autrechy and Reggia, 1989). SUNs are characterized by the fact that activation values on the nodes are updated at each clock cycle (Kohonen et al., 1989; Kohonen, 1984; Kanerva, 1984; Kanerva, 1988; Rumelhart et al., 1986; Jordan, 1986; Pollack, 1986; Elman, 1988; Shimohara et al., 1988; Nenov, 1990). The majority of these nets use error back-propagation as a learning mechanism (Rumelhart et al., 1986). Exceptions are the Kohonen's model which uses a Hebbian learning mechanism (Hebb, 1949) and the recently proposed KATAMIC model (Nenov, 1990) in which a novel, more complex, neurally inspired learning mechanism is adopted. AUNs, on the other hand, are characterized by the fact that the node activations are not updated synchronously (Buhmann and Schulten, 1987; Buhmann and Schulten, 1988; Amit, 1988; Kleinfeld, 1986; Sompolinsky and Kanter, 1986; Tank and Hopfield, 1987; Bell, 1988; Peretto and Niez, 1986; Willwacher, 1982). These networks can be regarded as modifications of the classical Hopfield-Little-network models (Hopfield, 1982; Hopfield, 1984; Little, 1974; Little and Shaw, 1975). The classical models which have symmetric weights and operate within a noise-free space *cannot* be used as content addressable memories for *temporal* patterns. Such spin-like networks, however, can realize storage of temporal patterns if the synaptic weights are sufficiently non-symmetric and noise is introduced in the system. The sequential order of stored patterns in these models is a consequence of the asymmetry of the synapses, which provide direct projections between equilibrium states of the network. The transitions between the states are noise triggered.

Both the SUNs and the AUNs have problems along different functional dimensions (D'Autrechy and Reggia, 1989). Recently, several attempts have been made to bring these two classes closer together (Park, 1988). A type of hybrid architecture, as it was pointed out by D'Autrechy and Reggia, may incorporate the best features of each of them.

Methods

Our choice was based on the relative simplicity and popularity of this SRN model as well as on the availability of results from several experimental studies. While the KATAMIC memory has not yet been used extensively by the neural network community, and therefore only variations of the basic

architecture is at hand, the SRN model has different variants (cf. Amit, 1988; Elman, 1988; Mozer, 1988, Pearlmutter, 1988, Pineda, 1987a; Pineda, 1987b, Rowher and Forrest, 1987; Servan-Schreiber et al., 1988; Sompolinsky and Kanter, 1986; Stornetta et al., 1987; Williams and Zipser, 1988).

Comparisons of KATAMIC and SRN models were done along a number of functionally relevant axes including prediction accuracy, length of training, and memory capacity. Network implementations with comparable structural complexity and resource utilization (# of links & # of state variables, nodes) were used. The implementation details of both the KATAMIC and the SRN used in this study are described farther in this section. The data sets, the design, the motivation, and the results of the individual experiments are discussed below.

Elman's simple recurrent network (SRN)

Elman's SRN neural net has a three-layer feedforward architecture (Figure 12.5) and is a modification of the model originally proposed by Michael Jordan (Jordan, 1986). The input layer is augmented by additional units called *context units*. Their number is equal to the number of *hidden units*. The context units get one to one projections from the hidden layer units via non-modifiable weights of value 1. They project back to the hidden layer in a fully connected fashion via trainable weights. The activation of the context units is set initially to 0.5 which represents a "don't know" state, since the activation values vary in the range of 0.0 to 1.0. The weights between the input and hidden, context and hidden and hidden and output layers are trained using the error back-propagation method. The objective of the network training is such that each pattern in the input sequence can produce an output pattern equal to the next pattern in the sequence. This is achieved by propagating back to the network an error signal which at each time cycle is calculated as the difference between the next pattern in the input and the activation of the output layer obtained through the forward propagation of the present input pattern.

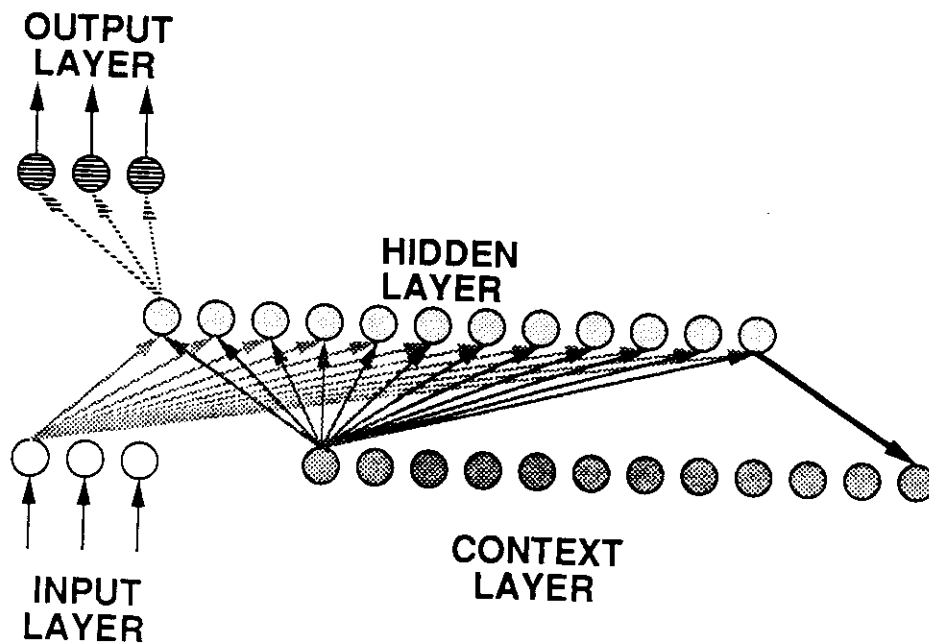


Figure 12.5: Elman's network

A schematic diagram of the simple recurrent network (SRN) model (Elman 88). Four layers of neurons labeled: input; context; hidden; and output are presented as shaded circles. Different shades of gray are used for units that belong to different layers. In the forward path (from input to hidden; from context to hidden; and from hidden to output layers) the connectivity is full (i.e. each neuron in the layer of origin projects to all neurons in the destination layer via modifiable synaptic weights). The projections from the units in the hidden layer to the context layer are topology-preserving, i.e. one to one. In the drawing, for the purpose of clarity, connections from only one representative neuron for each layer are shown.

Implementation of KATAMIC and SRN on the CM-2

The KATAMIC memory which was discussed in Chapter 8 has 64 predictrons with 512 dendritic compartments (i.e. a total of 32,768 compartments). The *p-ltm* & *n-ltm* were initially set to 0.5 and were continuously updated but not reset during the learning process. The initial value of the *stm* in each DCP was 0.001. It was updated at each time cycle and reset at the end of each sequence. At each time step the following performance features were recorded:

- (1) **goal** -- number of ON bits in the next input pattern
- (2) **match** -- number of ON bits in the present output pattern which match the ON bits in the next input pattern
- (3) **spurious** -- number of ON bits in the present output which do not match the ON bits in the next input pattern

In our implementation the SRN* model had 64 input nodes, 256 context nodes, 256 hidden nodes, and 64 output nodes. All weights were randomly initialized in the range of 0.0 to 1.0 except the recurrent weights between the hidden and context layers which were set to 1.0. All weights except the recurrent ones were updated at each cycle. At the beginning of each sequence the activation of all nodes was reset to 0.5. The learning rate was set to 0.25. At each time step the following performance features were recorded:

- (1) **goal** -- number of ON bits in the next input pattern
- (2) **match** -- sum of output unit activation in the network-generated present output pattern for those output units which match the ON bits in the next input pattern
- (3) **spurious** -- sum of activation in the network-generated present output pattern for those output units which match the OFF bits in the next input pattern

Detailed information about the CM implementation of the KATAMIC and the SRN networks is presented in Table 12.1.

* The code for the RBP model was written by Walter Read.

| SRN | |
|---|-------------------|
| #links (weights) | |
| input-> hidden | 64*256 =16,384 |
| context->hidden | 256*256 =65,536 |
| hidden->context (set to 1) | 256*1 = 256 |
| hidden->output | 256*64 =16,384 |
| Total (each link has also a delta value) | = 98,560*2 |
| #state variables (activation) | |
| input nodes | = 64 |
| hidden nodes | = 256 |
| context nodes | = 256 |
| output nodes | = 64 |
| Total | = 640 |
| ===== | |
| KATAMIC | |
| # links | |
| input->AF synapses (non-modif. set to 1) | 64*256 = 16,384 |
| input->PF synapses (Ts non-modif) (1% of (64*256) x 64) | = 10,496 |
| Total | =26,880 |
| # state variables | |
| p-ltm | 64*256 = 16,384 |
| n-ltm | 64*256 = 16,384 |
| stm | 64*256 = 16,384 |
| Total | = 49,152 |

Table 12.1: Implementation details of the KATAMIC & SRN models

Data sets

The sequences used in the experiments were composed of binary vectors (0 & 1 code). Various structured and random distributions of the active (ON or 1) bits were used. A random number generator was used to produce the sequences with 10% 1-bit-density for experiments 1 and 2. The 10 sequences used in experiment 3 were generated by permuting 10 random patterns of 10% 1-bit-density.

Experiments

Essential requirements for any kind of experimental comparison is that the system-dependent parameter settings are justified, and that the experimental paradigms and data encoding chosen for both systems are fair. We must be also aware of the fact that each system may be superior at different performance characteristics (e.g., correctness vs. cost). To compare the speed and accuracy of sequence learning in the two models, three separate experiments were run. In the first experiment both models learned a single, 10 steps long sequence composed of randomly generated binary patterns of 1-bit-density 10%. The second experiment tested both models in learning of 10 randomly generated sequences of 10% 1-bit-density. In experiment three the models again attempted to learn 10 sequences, however this time sequences 2 through 10 were generated by

random permutation of the patterns in sequence 1. Hence, this experiment was designed to test the performance on correlated sequences.

Experiment 1: Learning a single sequence

A single randomly generated pattern sequence of 1-bit-density 10% and 10 steps length was presented to the models multiple times (10 times for the KATAMIC model and 100 times to the SRN model). During the learning, the accuracy of the predictions made by both models was recorded.

Experiment 2: Learning a set of randomly generated sequences

Ten sequences of length 10 and 1-bit-density 10% were presented to each of the models 10 times in sequential order. The quality of the predictions made for one (the first) of these sequences was monitored. NOTE: From a statistical point of view all sequences were equivalent and it was sufficient to monitor the behavior of the models for only one of the sequences. The criteria for correct (good) performance were: a) none or very few misses, b) low (how low) number of spurious.

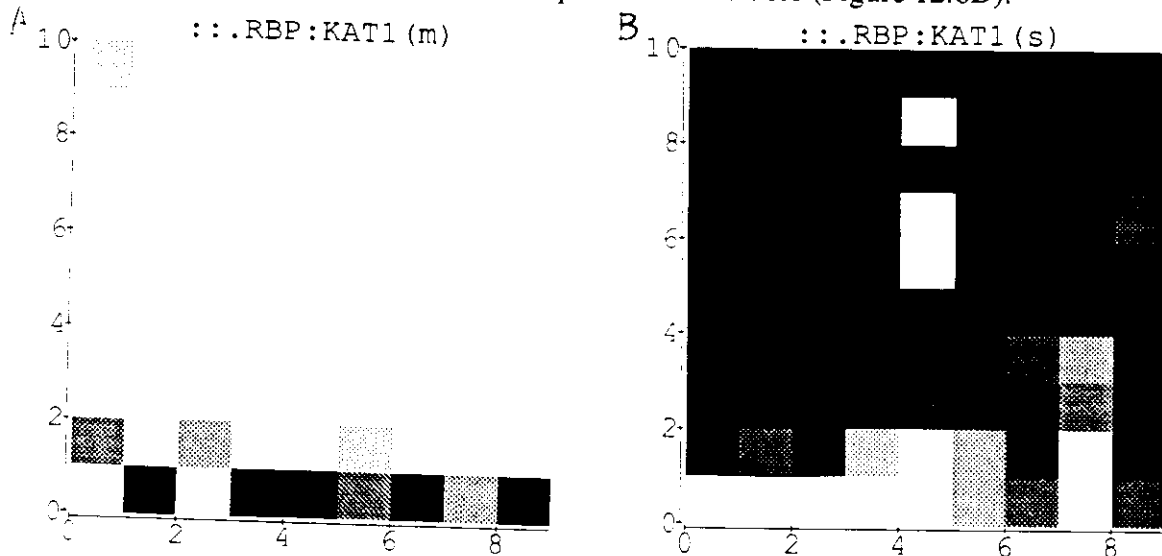
Experiment 3: Learning a set of correlated sequences

In this experiment again ten sequences were learned but they were generated by permuting ten randomly generated patterns of 1-bit-density 10%. Both network models were exposed to 100 repetitions of the whole set of sequences.

Results

Experiment 1: Learning a single sequence of length 10.

The results of this experiment from the KATAMIC and SRN runs are presented as sets of density-plots on Figure 12.6A (KATAMIC: match/goal), Figure 12.6B (KATAMIC: spur/goal), Figure 12.6C (SRN: match/goal), Figure 12.6D (SRN: spur/goal). For the KATAMIC model the predictions become effectively perfect (1.0) and stable after only 2 repetitions (Figure 12.6A). At the same time the number of spurious goes practically to zero (Figure 12.6B). On the other hand, after about 20 repetitions the SRN model reaches an average accuracy level of only 0.7. This accuracy does not improve significantly during the next 80 repetitions and in fact has a tendency towards deterioration (Figure 12.6C). Also the predictions made by the SRN model are very "choppy" (Figure 12.6C). The amount of spurious is tolerable (Figure 12.6D).



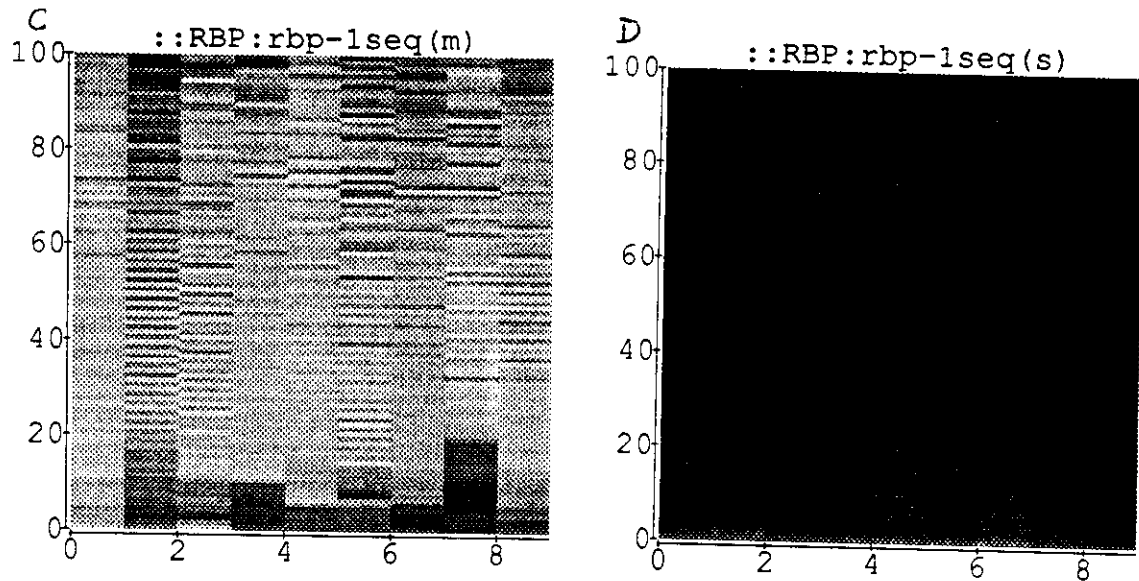
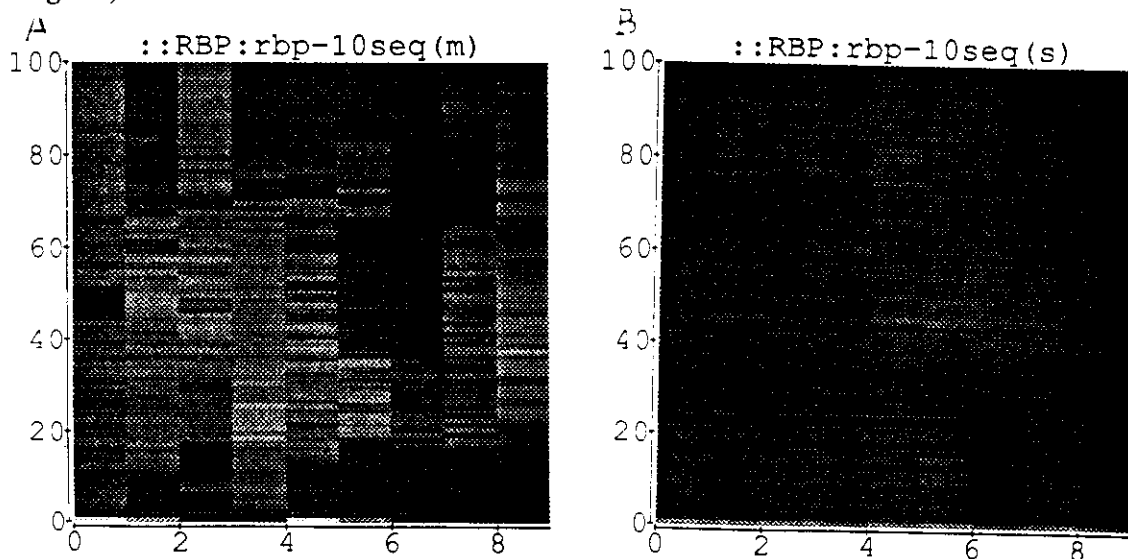


Figure 12.6: Learning a single sequence of length 10

On the X-axis we plot the sequential steps in any particular sequence and on the Y-axis the number of learning trials (i.e. repeated exposures of the network to the sequence). Two separate measures are plotted for each experiment. (1) The ratio of **match** to **goal**, and (2) the ratio of **spurious** to the difference of the total number of I/O bits and the **goal**. Both of these measures take values between 0 and 1. In the plots of the **match/goal**, the bright end of the gray-scale represents accurate predictions (i.e. the **match** is close to the **goal**), while the dark end represents poor prediction quality (i.e. the **match** is small as compared to the **goal**). The opposite is true for the plots of the **spurious/goal**. Here the bright end of the scale shows multiple **spurious** -- poor performance.

Experiment 2: Learning 10 *random* sequences of length 10.

The results of this experiment are presented on Figure 12.7A (KATAMIC: **match/goal**), Figure 12.7B (KATAMIC: **spur/goal**), Figure 12.7C (SRN: **match/goal**), Figure 12.7D (SRN: **spur/goal**).



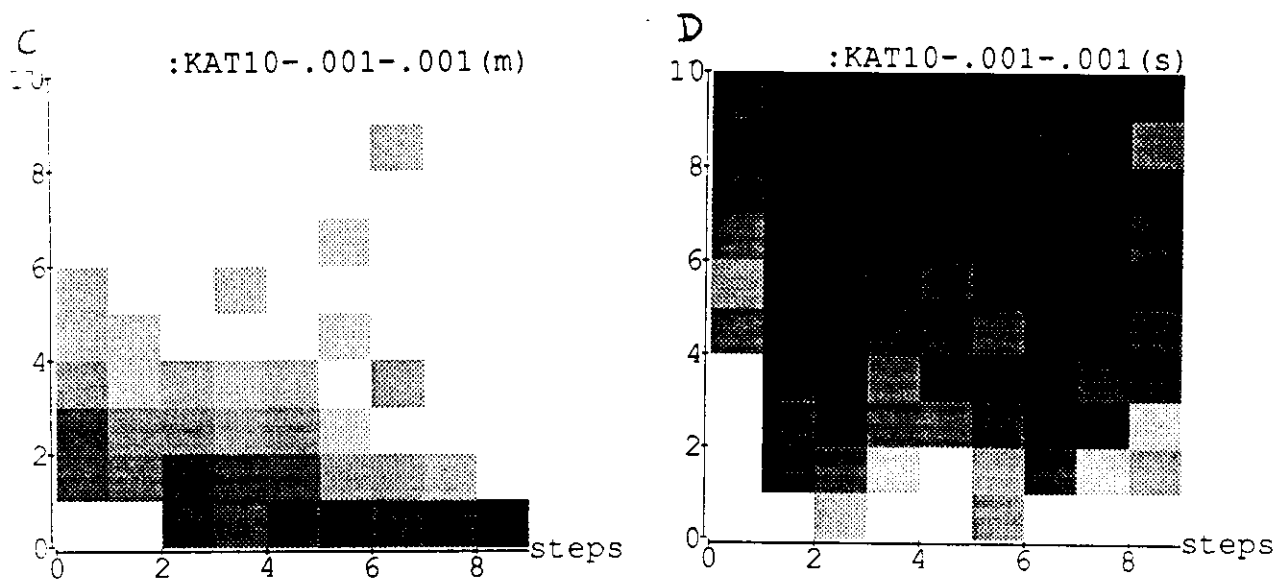


Figure 12.7: Learning 10 *random* sequences of length 10

The interpretations of the gray-scale and axes labels are the same as in Figure 12.6.

For the KATAMIC model the predictions effectively become perfect (1.0) and stable after 6 repetitions (Figure 12.7A). Also, at each repetition the number of spurious decreases after the first few steps of the sequence -- the time necessary for the memory to “recognize” the sequence (Figure 12.7B). The SRN model reaches a peak accuracy of only 0.35 within the first 100 repetitions and the predictions are very “choppy” (Figure 12.7C).

Experiment 3: Learning 10 *correlated* sequences of length 10.

As expected, both models exhibited worse performance on correlated sequences. However, while the SRN model did not show *significant* learning during 100 repetitions of the sequences, after 15-20 repetitions the KATAMIC memory learned the sequences with 0.96 accuracy (Figure 12.8A) and about 0.02 spurious (Figure 12.8B).

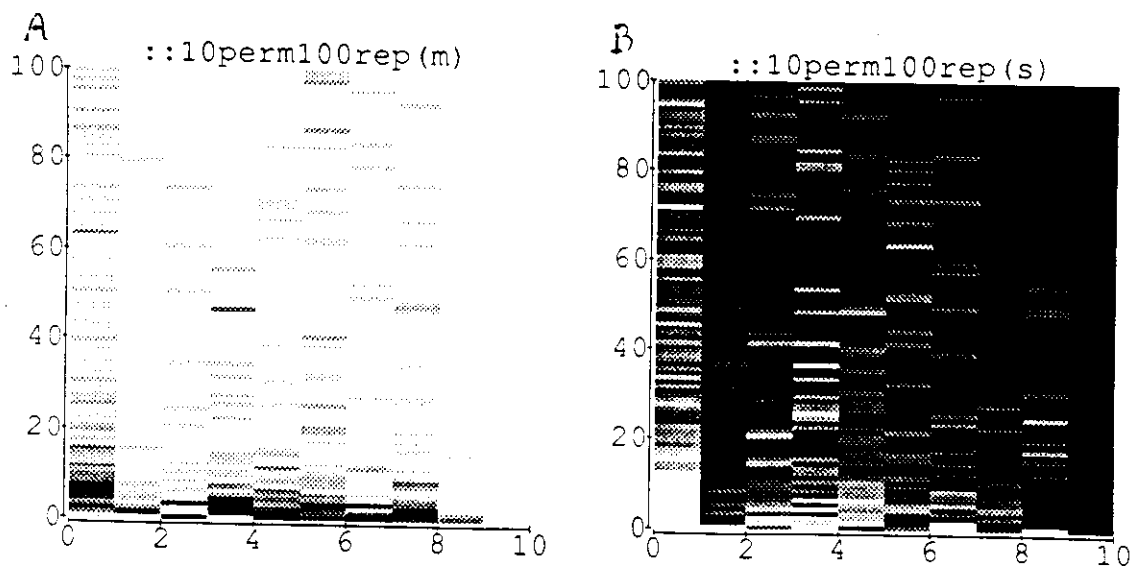


Figure 12.8: Learning 10 *correlated* sequences of length 10

A) match; B) spurious. The interpretations of the gray-scale and axes labels are the same as in Figure 12.6.

NOTE, that for any sequence in this set some steps are harder to learn than others. For the different sequences in the “permuted” set the hard-to-learn steps are different. This, of course, is a direct reflection of the relations between the different sequences (i.e. that they are correlated). Therefore, if a given step contains the *same* pattern in several different sequences which is followed by different patterns in the next step, then such a step will be more difficult to learn than if the same step contained a *different* pattern in the different sequences.

Discussion

Several system dependent characteristics may account for the observed performance differences. Notice that with the SRN network parameters which were chosen (see Table 12.1), the network performance in experiments 1 and 2 reached a plateau after the first few dozen repetitions. For this reason, experiments with thousands of learning trials (as commonly done in SRN network training) were not run.

It seems that the superior performance of the KATAMIC memory can be explained by the fact that during sequence processing the network maintains information about the spatial-temporal relations of the ON bits within several previous input patterns. The closer time-wise a previous input is to the present input the more it affects the current prediction. This short-term information is maintained within the *stm*. In contrast, the SRN model maintains information in the context units only about the input pattern which immediately precedes the current input pattern. This major functional difference between the two models can explain the vastly different performance on the data set of experiment 3 (correlated sequences). For successful performance of this test a model actually needs to maintain information on several previous time steps.

Another major advantage of the KATAMIC model is that the predictions which it makes are “absolute” in the sense that they are either 1 or 0. In the case of the SRN model, on the other hand, a threshold needs to be chosen beyond which an activation value at an output node can be considered as a positive prediction (sometimes two thresholds: high for 1 and low for 0, can be used). The choice of threshold is arbitrary.

It is important to notice that for both models the experiments were designed so that the actual predictions made were never used as consecutive input to the memories. Using the actual predictions made is of importance if we want to test the sequence completion ability of the models. As a direct consequence of the far better level of prediction accuracy which can be reached by the KATAMIC model, this model can be successfully used for learning and recall of actual *individual* sequences. The SRN model, on the other hand, even if we accept a generous 0.5 threshold level for the output unit activation, has such poor performance that it seems impossible to be used for recall of longer sequences from initial segments (cues).

Conclusions

The results of this comparative study (which uses systems of similar order of complexity and same problem sizes) indicate that the KATAMIC memory performs significantly better than the SRN model along all functional dimensions.

In terms of accuracy of predictions made, as it can be seen from experiment 1, the SRN model never reaches the performance level of the KATAMIC memory. On average, within 100 repetitions the SRN model reaches an accuracy level of 0.7 and the predictions are very "choppy", while the accuracy of the predictions made by the KATAMIC memory in the same task are almost perfect (0.98) and very stable.

In terms of speed of learning the KATAMIC model is also better. As it can be seen also in experiment 1. While the SRN model took about 20 repetitions to achieve its asymptotic level of accuracy (0.7), the KATAMIC model required only 2 steps. While from a computational point of view, with the introduction of fast parallel computers, the actual learning speed may seem not to be of importance, for real-time applications such as adaptive speech recognition and robotics the advantages which a fast learning algorithms offers are still very valuable.

In terms of memory storage capacity, measured here as the number of sequences which can be learned and retrieved satisfactory (with acceptable accuracy and a tolerable amount of noise), the KATAMIC model is again superior. While extensive testing of the memory capacity of both models was not performed, the results of experiment 2 are quite revealing. While the KATAMIC memory managed to learn 10 sequences after about 6 repetitions and was able to recall each one of them perfectly after the second step on average, the SRN model performed very poorly with an average prediction accuracy of 0.35 which did not improve substantially during the first 100 repetitions.

12.5.3 KATAMIC vs TDNN

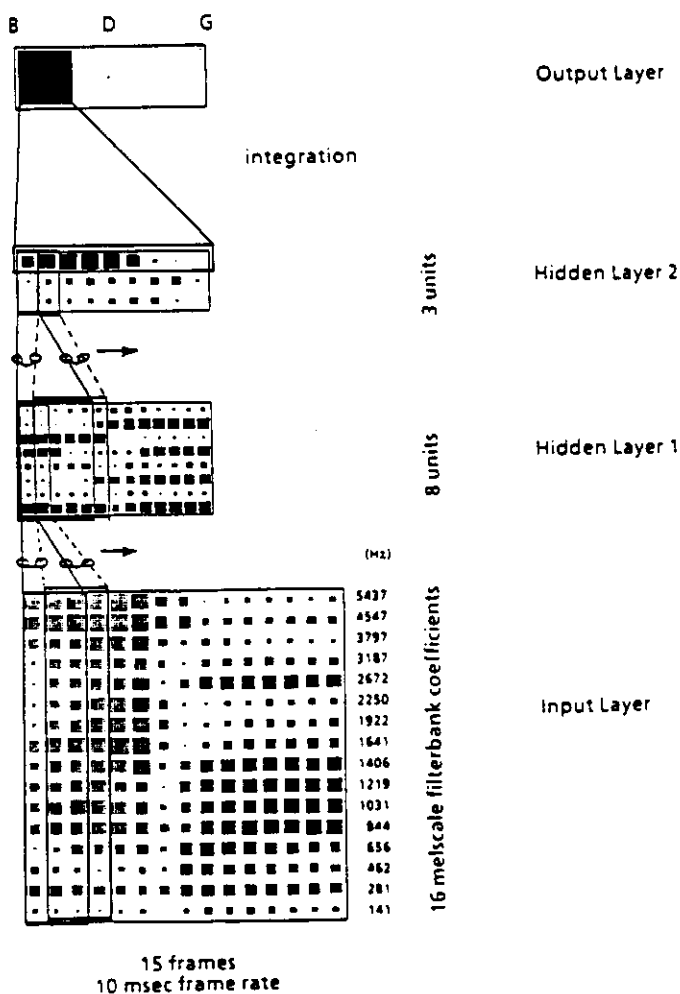


Figure 12.9: Architecture of the TDNN

Dynamics of the four layer TDNN architecture in the process of recognition of the phonemes "B", "D", and "G". The input layer (containing 16 units encoding consecutive values of 16 melscale spectral coefficients computed at 10 msec rate) is fully connected to the 8 time delay units in the 1st hidden layer. These units are connected to 3 TDNN units in the 2nd hidden layer which in turn are connected to the 3 output units -- one for each phoneme. The sizes of the black and gray squares for each unit encode their activation values. (Reproduced with permission from Waibel et al., 1987.)

Another neural network for sequence processing, called the Time Delay Neural Network (TDNN), was proposed by Alex Waibel (Waibel et al., 1988a; Waibel et al., 1988b; Waibel, 1989) and is successfully used for speech processing at Advanced Telephony Research (ATR), Osaka and Carnegie Mellon University (CMU). TDNN is a multi-layer perceptron type of device which uses modified McCulloch & Pitts neurons as basic processing units. The modification consists of allowing each unit to receive inputs measured at several consecutive time steps (usually 2 to 5) via separate, modifiable weighted connections (Figure 12.9). The layers are fully interconnected. This design allows the TDNN to relate and compare current input with the past history of the events. Another important feature of this model is that its hierarchical structure allows higher layers to attend to larger time spans which leads to a division of labour between the layers -- local short duration features of the input are formed in the lower layer while more complex longer duration features are learned at the higher layer. A major disadvantage of the model is its learning algorithm -- back-propagation (Rumelhart et al., 1986). Consequently, learning of a relatively simple task (as compared to the general speech recognition task) -- e.g, speaker-dependent recognition of the phonemes "B", "D", and "G" in varying phonetic contexts, takes 20,000 to 50,000 iterations on a supercomputer -- several days (Waibel et al., 1987).

12.6 Other models of selective attention

12.6.1 Fukushima's Neocognitron

A mechanism for control of visual attention was proposed by Fukushima in his "Neocognitron" model (Fukushima, 1980; Fukushima, 1987a; Fukushima, 1987b) -- an extension of his earlier "Cognitron" model (Fukushima, 1975). The Cognitron as well as the Neocognitron are multilayered neural networks with strong self-organizing capability modeled after the anatomy and physiology of the visual system. They consist of several modules [U_0 , U_1 , U_2 , U_3] connected in series (Figure 12.10). In accordance with Hubel and Wiesel's view (1962) on the visual system, the models have a number of "s-cells" (organized in layers U_{s1} , U_{s2} , U_{s3}) which correspond to simple cells in the primary visual system, and "c-cells" (organized in layers U_{c0} , U_{c1} , U_{c2} , U_{c3}) having the properties of complex cells. The Neocognitron has been used for recognition of handwritten characters independently of their location in the visual field. In its most advanced version, the system is able to sequentially recognize individual characters from a set of overlapping and noise-distorted characters appearing simultaneously in the visual field. The ability to sequentially refocus its attention was accomplished by temporary interruption of the feedback from the higher to the lower layers in the network (the X line in Figure 12.10).

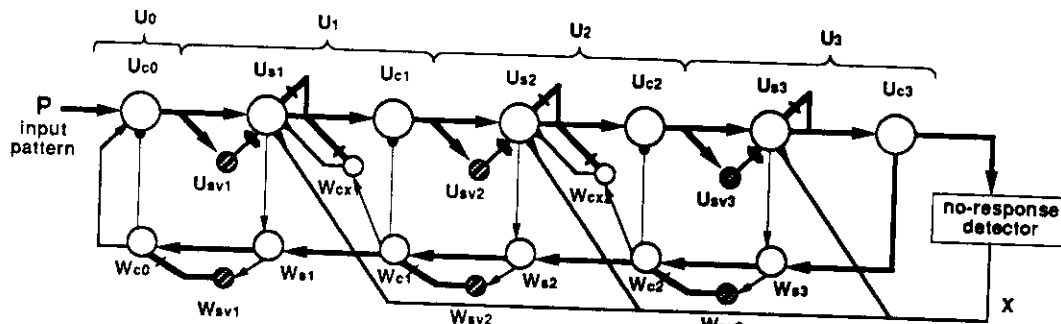


Figure 12.10: The Neocognitron

Hierarchical organization of the Neocognitron architecture. The white circles represent neural elements or groups of neural elements. Small shaded circles represent inhibitory interneurons. Thick wires indicate converging and diverging connections between two groups of cells; thin wires indicate one-to-one connections between two corresponding cells; excitatory synapses are indicated by arrows (small for fixed weights and large for variable weights); inhibitory synapses are indicated by T-shaped connections (thin for fixed and thick for modifiable); connections used for gain control are indicated by small black circles; connections indicating threshold control are indicated by small black triangles. (Adapted from Fukushima, 1987.)

12.6.2 Crick's "Searchlight of attention" hypothesis

A neurally realistic model of selective attention -- "the Searchlight of Attention" was proposed by Francis Crick (Crick, 1984). This is a multi-layer neural network which models part of the visual pathway in the brain including the Lateral Geniculate Nucleus of the Thalamus (LGN), the Perigeniculate Nucleus (PGN), and the neocortex (Figure 12.11). The connectivity and dynamics of the neural elements in this architecture model closely their counterparts in the brain. In this model attention is defined as increased thalamic input to a patch of neocortex -- a "searchlight beam". The searchlight is expressed in rapid firing in a subset of principal thalamic neurons controlled by a negative feedback from the reticular formation. Due to the specific firing properties of the thalamic neurons (ability to generate a burst of action potentials as a result of hyperpolarization by reticular inputs, followed by a 100 msec long refractory period), the searchlight can be turned off and moved to the next place in the stimulus space demanding attention. The function of the searchlight in the neocortex is to form temporary neuronal assemblies. Crick suggested that visual binding in the neocortex occurs in longer intervals of about 50 ms during which bursts of impulses, produced by "searchlights" of attention in the thalamus, provide the signal for rapid synaptic changes. This theory seems to have electrophysiological support, for instance Dempsey and Morison (Dempsey and Morison, 1942) observed "augmenting" and "recruiting" waves spreading to the cortex after thalamic stimulation.

The "searchlight of attention" model was implemented by Nenov and Read and tested on the CM-2 Connection Machine (Read and Nenov, 1991). Based on our experiments, we can confirm that the model is capable of generating and maintaining a searchlight for a short period of time. It is also capable of switching the beam from the most intensely firing patch of neurons in the visual input to the next lower intensity patch. However, it was not able to continue switching to further

less intense patches in the input and the temporal discrimination between the patches became blurred with time. For this reason, and because of its demand of computational time, this model was not incorporated in DETE (despite our original intention).

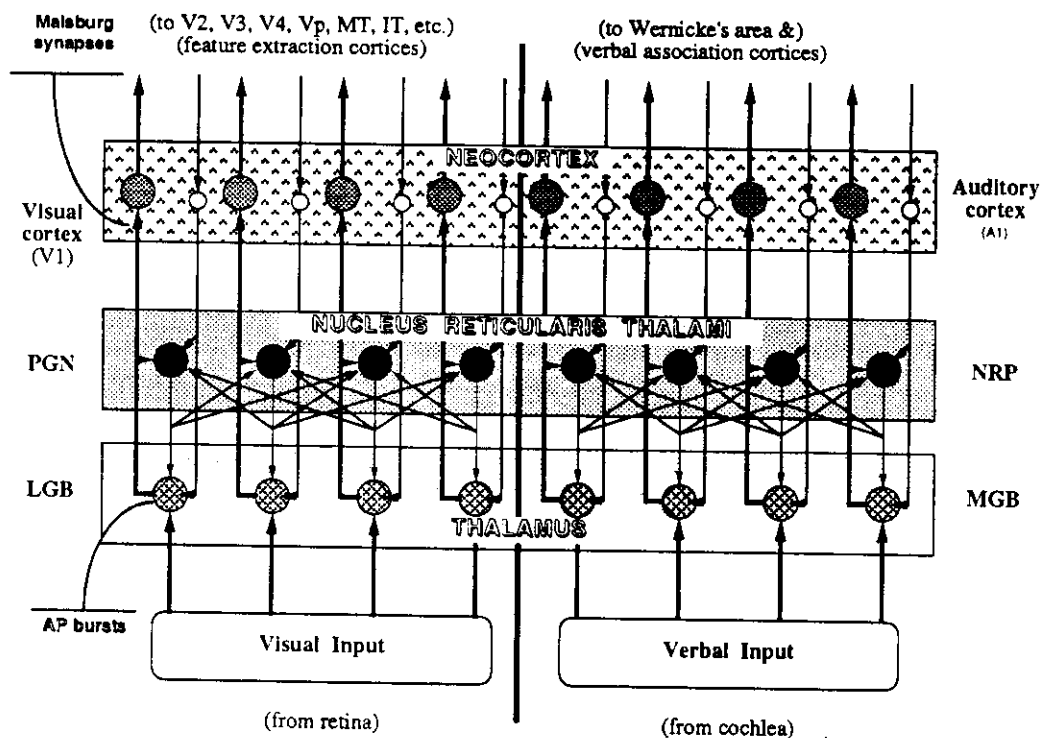


Figure 12.11: Crick's "Searchlight of attention" model

Schematic drawing of the neural architecture supporting Crick's "searchlight of attention" hypothesis. A three layer network containing idealized neural elements which correspond to real neurons in various brain cortices and nuclei is depicted. Cell bodies (shaded circles) are located in the corresponding nuclei/cortices (shaded rectangles). Thick arrows represent excitatory synapses whereas thin arrows represent inhibitory synapses. The brain structures belonging to two distinct sensory processing pathways are shown: (1) the visual pathway (to the left) includes: LGB -- lateral geniculate body, PGN -- perigeniculate nucleus, V1 -- primary visual cortex, and (2) the auditory pathway (to the right) includes: MGB -- medial geniculate body, NRP -- nucleus reticularis proprius, and A1 -- primary auditory cortex. (Adapted and elaborated from Crick, 1984.)

12.7 Other Systems for Sensory-Motor integration

12.7.1 Darwin I, II, and III

A selective recognition automaton based on the principles of neuronal group selection (Edelman, 1987) was constructed by Gerald Edelman, George Reeke, and their colleagues. The neuronal group selection principle is based on the premiss that the maturation of neural circuits in the developing brain follows a Darwinian selection process. This process is a sort of competition in

which a set of random variables are modified by experience and due to selection few of them win and are potentiated whereas the rest degenerate. As a result of this process the system adapts to the conditions imposed by environment in which the contest takes place.

The earlier bare-bone versions of the system called Darwin I and II (Edelman and Reeke, 1982; Reeke and Edelman, 1984) engaged in recognition and classification of patterns. Darwin III (Figure 12.12) (Reeke et al., 1989) is provided with three senses: (1) *vision* -- through a simple square-shaped retina which has a central (higher acuity) and peripheral (lower acuity) areas; (2) *touch* -- mediated by a four-joint arm that had a single pressure sensing device at the end (the finger); and (3) *kinesthesia* -- measuring the position of the arm joints in space. Darwin III, whose basic design principles were *re-entrant mappings and neuronal selection* contains about 46 networks (repertoires), and about 6,000 neurons with 150,000 connections. These networks are organized in several modules: (1) Oculomotor (OM) system with a main purpose of learning to track moving objects. In its naive state the motion of the eye is random. After about 2,000 trials the OM system learns to follow an object that appears in the visual field. The learning of this behavior is based on lateral inhibition between neural groups (a proto-attention mechanism) and depression of activation of selected neuronal groups. (2) Motor system capable of performing several tasks: (a) Reaching task -- the arm can move around the visual field until it can touch objects that are in this field. Initially the motions are random but with multiple trials during which Darwin is looking at an object while at the same time executing an arm motion, it learns to reach to an object as soon as one appears in the visual field. The learning of this task is controlled by a cerebellum-like network that inhibits neural activity going from a motor cortex-like network to a network that models the spinal cord. The cerebellar network implements the goal to reach by filtering out gestures of the arm that are inappropriate for reaching. (b) Sensing task -- touching and feeling the object that the arm has just reached for. (c) Tracing task -- exploration of the object.

Categorization of objects is done by associating the visual representation of object with its tactile representation (obtained via tracing). After successful recognition, a reflexive motion is generated in response (e.g., swat away of a bumpy object -- rejection response).

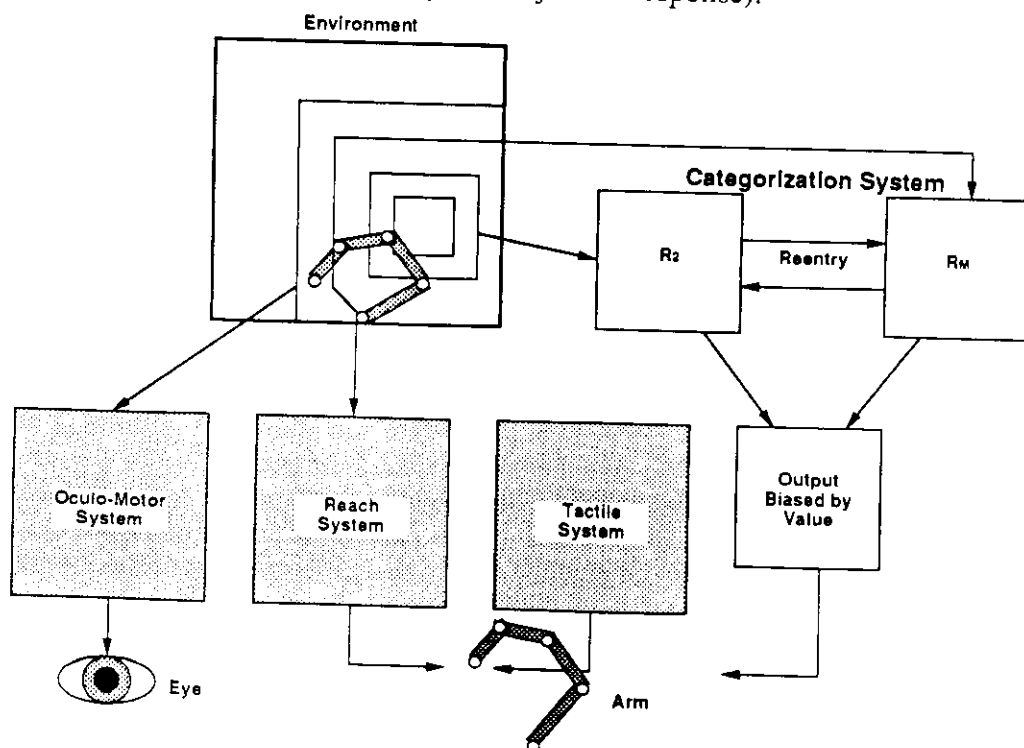


Figure 12.12: Darwin III

The principle functional subsystems of Darwin III. For explanation, see text. (Adapted from Reeke, Sporns, and Edelman, 1989.)

Darwin III can be compared to DETE along several dimensions. (1) *Language*: DETE is capable of handling language whereas Darwin does not take language input. (2) *Vision*: While Darwin learns to recognize objects by modifying the strengths of the connections between maps, DETE is provided with hard-wired feature extraction modules which automatically categorize the features of objects in various feature maps. A meta-level classification of these maps is done in DETE by means of the verbal input. (3) *Goal-driven behavior*: Darwin III was provided with innate goals. For instance: (a) "seeing is better than not seeing" (i.e. if you are moving from a bright area to a dark area, go back); (b) "it is better to be close to an object than far from it" (i.e. if you see an object, reach for it); (c) "it is better to touch than not to touch" (i.e. if you are touching an object, keep exploring it). These innate goals were implemented by value-sensitive cells which monitor the environment and evaluate the consequences of Darwin's behavior. DETE, on the other hand, in its current version lacks completely an intentional (motivation or goal /plan) component and operates in a purely reflexive or filter-like fashion. (4) *Motor behaviors*: Darwin is capable of actions in response to external stimuli. For instance it can reject a rough object when it touches one. DETE does not possess this functionality. (5) *Associative memories*: Both systems use the association between two modalities to achieve recognition: in DETE, visual & verbal; in Darwin III visual and somesthetic (tactile/ kinesthetic).

12.7.2 Gary Drescher

In his Ph.D. thesis from M.I.T. Gary Drescher (Drescher, 1991) describes a schema-based symbolic processing mechanism which operates in a 2-D microworld containing uniform-sized objects which are able to move but cannot rotate. Drescher's model, unlike DETE which is concerned primarily with early language acquisition in humans, focuses on the early stages of human sensori-motor integration. The model is an implementation of the cognitive development theory of Jean Piaget (Piaget, 1952; Piaget, 1954). In computer simulations it addresses the questions of sensori-motor learning and concept formation. The program, like DETE is implemented in *Lisp on the CM-2 Connection Machine. It controls a simulated robot that has a body, a hand, and a visual system. The system is able to shift its eye forward, backward, left or right. The visual input (a cross-shaped visual field composed of 5 foveal regions) of the simulated robot is designed to provide a bird's-eye view of its visual world (an array of $5 \times 5 = 25$ regions). Each region of the visual world can either contain or not contain an object. Each region of the visual field (retina) contains information about 16 features (items) of an object present in it (e.g., shape, texture, color, taste, touch, etc.). The robot's hand can touch and grasp objects and move them around by performing sequences of primitive actions including: a) move hand forward, backward, left or right; b) grasp and ungrasp. There is an initial, bare schema for each primitive action. For instance, the "grasping" schema asserts that the result of a grasp action is a tactile sensation of the object in the hand. Learning in this model corresponds to incremental construction of a chain of interconnected schemas from the 10 initially supplied bare schemas. An example of such learned schema is "shift the eye from a given orientation to any other orientation". While Drescher's model can perform a broader spectrum of sensory-motor interactions than DETE, it lacks the ability to represent events happening in the past or events that are expected to happen in the future. Also,

since it is not a language acquisition and processing system, it cannot answer questions about the states of the objects that it can see and manipulate.

12.8 Representation of space & time

12.8.1 George Lakoff

In his book “Women, Fire, and Dangerous Things” George Lakoff presents a theory of cognitive linguistics (Lakoff, 1987). One of the major conjectures of this theory is that humans reason by using some of the same mechanisms that are involved in perception. Lakoff provides linguistic evidence that the humans’ reasoning mechanism can be seen as growing out of perceptual and motor mechanisms. In a later paper Lakoff argues that cognitive linguistics converges with connectionist cognitive science in a variety of ways (Lakoff, 1989). More specifically he emphasizes the fact that the brain uses patterns of activation over topographic maps of the sensory space to represent the meanings of sensori experiences. He points out that a particular activation pattern is meaningful only with respect to the particular map in which it appears. In fact, two activations patterns which are the same (or very similar) will have completely different meanings if they appear in two different maps. Their meaning will depend on the sensori features that each of the maps encodes and the sensory system from which it gets its input. While Lakoff does not suggest a concrete neural architecture and implementation of these topographic maps, DETE’s Visual Feature Planes are in fact an embodiment of Lakoff’s theory. Indeed, if we compare the organization of the Location Feature Plane (LFP) to that of the Motion Feature Plane (MFP) (or the organization of the siZe Feature Plane with that of the Color Feature Plane) we can see that similar (or even exactly the same) patterns of activation in the two Feature Planes will have completely different meanings. For instance, an activation pattern that has four active pixels in the lower left-hand corner of the LFP will represent an object located in the lower left-hand corner. However, in the MFP the same pattern will represent an object moving fast South-West. Notice, however, that DETE’s feature maps are not just an implementation of Lakoff’s theory. We have augmented Lakoff’s theory by proposing that the maps need not be necessarily topological in nature (e.g., the LFP). Instead, they can be based on any mapping which is adequate for solving of particular task as long as the implementation of this mapping is neurally plausible (e.g., MFP). Also, DETE’s maps have a complex temporal dimension.

12.8.2 Leonard Talmy

In his theory of how language structures space (Talmy, 1983) Talmy focuses on the “fine structure” that language ascribes to space. Talmy postulates the existence of such “fine structure” arguing that two sentences such as “The bike is near the house.” and “The house is near the bike.” (which one would expect to be synonymous on the grounds that they represent two inverse forms of a symmetric spatial relations) obviously do not have the same meanings. These sentences, Talmy argues, would be synonymous if they specified only this symmetric relation (i.e. the quantity of distance between the objects). However, in addition to this the first sentence makes the non-symmetric specification that the house is to be used as a reference point for the bike’s location. The second sentence makes the opposite non-symmetric specification. In other words, Talmy assigns different spatial roles to the objects in space depending on the syntactic structure of the utterance. In his theory Talmy provides a detailed analysis of a variety of examples of spatial relations between actual physical objects and also of the metaphorical extensions of such spatial relations.

Like Talmy's representation, DETE's representation of space uses spatial roles such as "Speaker location", "Event (object) location" and "Reference location" (see section 11.5.2) to capture the "fine structure" of the spatial relations between objects. The scope of the spatial relations that DETE deals with is much smaller than that covered by Talmy and DETE does not deal with metaphorical extensions. However, unlike Talmy, who proposes a symbolic representation without testing it experimentally, in DETE we propose and implement a detailed neural representation of the theory and demonstrate its potential in computer simulations.

12.8.3 Reichenbach

Together with a representation of space, the representation of time is the other most important component of a cognitive theory. As discussed in section 11.7, most of the current theories of time representation are extensions of the work of Reichenbach (Reichenbach, 1947). According to this theory, all temporal relations encountered in narratives can be accounted for in a model which uses three basic temporal roles: (1) *Speech time* (S); (2) *Event time* (E); (3) *Reference time* (R). DETE provides a connectionist implementation of Reichenbach's abstract theory. It represents each of the temporal roles as an activation of neural-assemblies in a specific plane of the Temporal Memory (see section 9.3). Specifically, Speech time (S) is represented by the activation generated in the TP-0 of the Verbal Memory by the verbal input or by DETE itself during the generation of a verbal response. The Event time (E) is represented as a pattern of activation generated in the TP-0 of the Visual Feature Memories by the visual input -- a sequence of frames that capture the event. The Reference time (R) is also represented as a pattern of activation in the Visual Feature Memories. However, this activation is induced by the "referent" visual event. In DETE's implementation the temporal aspect (or temporal focus) is always directly related to the time of the Referent event (R). Sometimes, all three roles are represented in the same Temporal Plane (e.g., in the case of present tense), in other cases two of them can share one and the same TP while the third is represented in a different TP (e.g., future tense, present perfect tense, etc.). In yet other cases, each of the roles is represented in a different TP (e.g., future perfect tense, past perfect tense, etc.). This connectionist representation of time allows the model to learn the meanings of verb tenses.

12.9 Other work in symbol grounding

12.9.1 Josep Maria Sopena

Josep Maria Sopena (Sopena, 1988) used a modification of Servan-Schreiber's (Servan-Schreiber et al., 1988), and Elman's (Elman, 1988) architectures (3-layer neural network made up of 72 input, 35 hidden, and 8 output nodes) to associate visual patterns to their verbal descriptions. The visual patterns consist of objects (pyramids, cubes, etc.) having simple features (red, green, etc.) and arranged in a variety of spatial relations (behind, on, next to, etc.). Each visual pattern is associated with a verbal description composed of a five word long sequence of the form: ADJ NOUN VERB ADJ NOUN (e.g., red pyramid is_on_a green block). Sopena used a small set (22) of training visual/verbal pairs and tested the model on a larger set (42 pairs). The experiments were focused on the generalization capacity of the network, the learning of new concepts, and on the kind of grammar which the network acquired. In comparison to DETE, Sopena's architecture is much simpler. For instance, it does not have separate memory modules (e.g., verbal, visual, short-term, long-term, temporal memory, etc.). As a result, it cannot handle most of the tasks on which DETE was tested such as learning about motion, learning the meanings of various verb tenses, etc.

12.9.2 The L₀ miniature language acquisition project

This research project, which is currently underway at the International Computer Science Institute (ICSI Berkeley, CA), is lead by Jerome Feldman and George Lakoff (Feldman et al., 1990; Hollbach Weber and Stolcke, 1990; Regier, 1991; Stolcke, 1990). The L₀ project seeks to develop computational models of language acquisition in the semantic domain of spatial relations between geometrical objects in two-dimensional scenes. The major achievements of this project are discussed below.

Andreas Stolcke

Learning feature-based semantics using Simple Recurrent Networks (SRNs) was the objective of Stolcke's studies (Stolcke, 1990). In these studies, sequential natural language input was mapped into static feature-based semantic output. The language taught to the network is a restricted version of the L₀ language -- an artificial language specified by Feldman and his colleagues (Fahlman, 1988; Feldman et al., 1990) which is a subset of English. Stolcke's networks are simple extensions to the original Elman's SRN model. They are used in two sets of experiments exploring the task of extracting semantic features from sequential word input: (1) learning of sentences containing a single predicate applied to multiple-feature objects (e.g., *a light circle touches a small square*); (2) learning of sentences with embedded structures (e.g., *a triangle touches a square above a circle*). The results of these studies show that SRN can indeed learn (with fair amount of robustness) to incrementally assemble complete (static) semantic representations (feature vectors) from word-by-word presented simple declarative sentences. However, the results of the second set of experiments (with sentences containing embedded structures) were less satisfactory. SRNs were unable to correctly process center-embedded PPs (e.g., *a triangle touches a circle below a square*). This inability was due to the fact that SRNs cannot produce a hierarchical representation of the semantic space (which seems to be necessary for the performance of this task). The network was able, however, to handle multi-level sentence-final embeddings (e.g., *a triangle to the left of a circle touches a square*) because it was able to maintain information about the immediate past.

A general problem with the choice of representations taken in this study is that unlike DETE the semantic representations in Stolcke's system are not generated during the learning process (i.e. these symbols are not mapped to any perceptual information from the visual world) but are generated a priori in a random fashion. Such an approach does not allow capturing of real meanings of the words since the randomly generated representations do not reflect the actual relations between the objects and features in the physical world. Another weakness of this approach is that the output (semantic) representation is static and there are no temporal cues concerning the correlation between the input elements (words) and the target (semantic) representations. As a result Stolcke's system cannot learn about a world in which there is motion, interaction among objects and feature changes.

Terry Regier

As a part of the same L₀ research effort (Fahlman, 1988), Regier developed a neural network model based on a quickprop architecture (Fahlman, 1988) which learns to associate scenes containing several simple objects with terms that describe the spatial relations among the objects in the scenes (e.g., above/below, on/off, inside/outside, to_the_left_of/to_the_right_of) (Regier, 1991). As in our initial experiments on learning spatial lexemes (see section 11.5.2), Regier considers the situation when the feature space is partitioned by the words into mutually exclusive regions. In a number of experiments Regier demonstrates that the system can generalize the learned words so that they can be applied to new situations. In order to avoid overgeneralization or

“roughness” in the categorization of the feature space, Regier uses each positive evidence for a particular spatial arrangement as an explicit, weak negative evidence for all other spatial arrangements. The choice of this approach was based on observations that children acquire language apparently without the benefit of negative evidence (Braine, 1971; Bowerman, 1983; Pinker, 1989).

Like Regier’s system, DETE uses each positive evidence (e.g., an instance of a particular object with its features or a relation between two objects) as a weak negative evidence for the rest of possible objects or relations. This functionality is accomplished by the “resource management” step of the learning algorithm for the KATAMIC sequential associative memory (see Formulae 8.9a&b in section 8.2.1(5)). A significant difference between DETE and Regier’s model (with respect to the representations of spatial relations) is that while DETE learns the meanings of words like “above” with respect to a horizontal line passing thru the center of the Visual Field, Regier’s system is more sophisticated in categorizing spatial relations. For example, by using actual 2-D objects as landmarks (instead of an imaginary line) it manages to partition the visual field in a more adequate way.

Another parallel between DETE and the L_0 project is that the grammatical structure of the FIRLAN language used in DETE is similar but more complex (has a larger set of grammatical rules and a larger lexicon) than the L_0 language used in the domain of simple two-dimensional static scenes considered by Feldman and co-workers (Feldman et al., 1990).

12.9.3 MAIMRA & DAVRA

In an attempt to address some of the issues related to language acquisitions in children, Jeffrey Mark Siskind at M.I.T. developed a system called MAIMRA (Siskind, 1990). MAIMRA is a symbol manipulation system composed of the traditional parser, linker and inferencer. It receives two types of inputs: (1) descriptions of sequences of scenes depicted via a conjunction of true and negated atomic formulas, and (2) time ordered sequence of sentences which describe the events taking place in the scenes. The output of this system is a lexicon consisting of the category and the meaning of each word in the verbal input. This output is generated by a process of reverse application of a compositional semantics linking rule followed by constraint satisfaction.

Recently MAIMRA was extended to DAVRA (Siskind, 1991b; Siskind, 1991a). DAVRA relies on a collection of syntactic and semantic principles, collectively termed *Universal Grammar* to determine the syntactic category and meaning of the words. DAVRA is more flexible than MAIMRA in the sense that it does not assume a fixed, built in grammar prior to lexical acquisition, but rather it uses a parameterized variant of Siskind’s theory on which MAIMRA is based and acquires the parameter values during the training sessions. The system was tested on sets of few English and few Japanese sentences.

Unlike DETE, both MAIMRA and DAVRA do not have a perceptual mechanism; both the linguistic and non-linguistic input are presented in symbolic form to these systems. For instance, the symbolic description of a scene in the form $(BE(cup, AT(John)) \wedge \neg BE(cup, AT(Mary)))$; $(BE(cup, AT(Mary)) \wedge \neg BE(cup, AT(John)))$ is paired with the sentence “The cup slid from John to Mary”.

12.9.4 RobotWorld

Patrick Suppes and his graduate students at Stanford University have developed a natural language acquisition system implemented in RobotWorld (Suppes et al., 1991). Like DETE this system

focuses on learning natural language about spatial relations in a dynamically changing visual environment. Like Siskind's system, this system is symbolic in nature (implemented in Lisp) and uses extensively the inherent power of Lisp for symbol manipulation. Each experimental trial with this model consists of a verbal input and a corresponding action performed by a simulated robot. During each trial the system's memory is modified -- the system learns. The learning consists of: (1) Formation of associations between words (for objects, properties and relations) and internal symbols (which represent various categories). (2) Derivation of the grammatical form (e.g., "ACTION the OBJECT") from the verbal instance (e.g., "Get the screw!"). The acquisition of new grammatical forms is done by making probabilistic associations between words in the utterance and internal symbols. (3) Storage of the verbal input in a short-term memory for the duration of the trial. Suppes' system is currently capable of learning about 50 words in 360 three-word long sentences in English, German, Chinese, and Russian.

Despite its symbolic basis, Suppes' system and DETE share certain important commonalities, e.g., no prior linguistic knowledge is assumed. Both systems learn the words for specific objects or spatial relations. The set of possible mappings of the words are given a priori to the systems. However, DETE assumes prewired features in visual feature memories -- distributed representations, whereas Suppes' system assumes internal language -- symbolic tokens.

Both systems associate verbal and non-verbal inputs in the process of language acquisition. DETE associates visual scenes and verbal descriptions whereas Suppes' system associates verbal stimuli and (user-coerced) actions which are used to correct mistaken actions. Also, both systems are able to acquire simple grammars, which mostly contain grammatical forms defining word order in short sentences. Unlike DETE, however, Suppes' system acquires only a "comprehension grammar" whereas DETE also acquires a "production grammar". In other words, DETE is capable not only of "understanding" grammatically correct verbal inputs but it can also generate grammatically correct verbal outputs in response to questions. Suppes' system, on the other hand, is mute and can only learn to execute verbal commands.

12.10 Symbolic models of NLP

A major approach taken by symbolic models of natural language processing (NLP) is based on Script Theory (Cullingford, 1978). (Other approaches include: goal/plans, beliefs and attack/support). The Script Theory is based on the construction of a Conceptual Dependency (CD) representations of natural language (Schank and Abelson, 1977; Schank et al., 1981). A NLP system based on this theory usually contains in memory a number of scripts and the language input to a script application system is matched to the known scripts after it has been parsed into a conceptual representation.

Among the most comprehensive models which used the script technology for NLP are *SAM* (Script Applier Mechanism) (Cullingford, 1978), *FRUMP* (Fast Reading Understanding and Memory Program) (DeJong, 1979), *BORIS* (Dyer, 1983). The major shortcomings of the traditional symbolic NLP models is that they are very fragile and knowledge-limited. In other words, the data processing mechanisms which they involve are usually very specific to domain of knowledge for which a particular system has been designed. Also, these mechanisms have to be hand-crafted and with some exceptions (e.g., *IPP* which learned to generalize terrorism stories (Lebowitz, 1980)) the models cannot learn, i.e. modify their behavior depending on their experience. The representations of new concepts have to be hand-crafted by the programmer. The

performance spectrum of such rule-based systems is limited by the set of rules which the system designer has provided. Also, their ability to generalize is often based on complex rules which require substantial knowledge engineering.

DETE shares some of the limitations of symbolic models. For instance, it operates in a limited domain -- the Blobs World. In fact DETE is even more domain-limited than the symbolic NLP systems since it cannot operate in the task domains that symbolic systems typically can (e.g., shopping, restaurants, castles and dragons, visiting dignitaries, legal cases, etc.). However, a significant advantage of DETE is that it can learn from experience. Once its architecture is constructed, the representations of new instances are automatically extracted and new concepts are learned by experience. Also, as DETE's visual architecture is expanded, it will be able to deal with structured objects and more complex interactions between them. Then it could learn about restaurant scripts, knights fighting dragons, etc. simply by viewing visual scenes of such event sequences and associating them with verbal descriptions.

12.11 DETE versus language acquisition in children

Children do not start to learn language all at once (Bruner, 1983). A number of psychological experiments concerned with language development in children have shown that there is a phase in early development, when the child communicates with its parents by means of gestures such as reaching, smiling, pointing and babbling vocalizations. This type of communication occurs in the form of a dialog. An important and necessary characteristic of this pre-language type of communicative behavior is the innate ability of the child to direct and maintain attention to objects in the environment to which the parent points (i.e. shared object of attention) (Bruner, 1975; Ninio and Bruner, 1977). This behavioral characteristic is fundamental also for the DETE model.

In the field of theoretical linguistics there have been several theories concerning the relationship between language and cognition in general and visual perception in specific. A group of researchers lead by Chomsky (Chomsky, 1986) have advanced the theory that language is an independent aspect of intelligence and relies only on language-specific mechanisms. This theory is founded largely on the belief that humans possess an innate grammatical ability which is universal. A second group of researchers including Piaget (Piaget, 1951) and Bates (Bates, 1976) take an opposite stance by suggesting that cognitive processes are primary and that language is fully dependent and emerging from already developed cognitive concepts. Middle-ground hypotheses, that argue for the significance of the interaction between language and non-linguistic cognition, have also been advanced. While it is difficult to take a definite position with regard to any of these theories (since none of them has actually proposed a realistic functional model of language acquisition), and since there is a possibility that all of them are to some extent correct or can even be regarded as alternative and mutually non-exclusive interpretations of the same phenomenon, in the development of DETE we have leaned towards the latter ones because they seem to have much stronger support from experimental studies (Andersen et al., 1984; Dunlea, 1989).

13 NEUROPSYCHOLOGICAL & NEUROBIOLOGICAL INSIGHTS

DETE's structure and functions were inspired by a number of general principles underlying the functional organization of the language, vision, attention, memory, and motion related areas in the brain. However, with the exception of the memory mechanisms (which are based on neural networks interpretable from a physiological perspective), none of DETE's peripheral modules have been implemented as realistic neural networks (of the types found in the cortical and subcortical areas of the brain). In other words, all modules except the memory modules are implemented as procedures and therefore the model as a whole is a hybrid one. This chapter describes in some detail the currently available neuroscientific knowledge about the brain systems involved in vision, language processing and attention and relates it to DETE's architecture and function. It also outlines future directions of DETE's development in terms of a more realistic neural architecture and function. Such an effort is relevant and necessary since the ultimate question of our research is how does the human child (its nervous system) acquires language and not just how can we build any device that mimics human language processing abilities.

13.1 Neural codes -- Discussion of representations

The question of what constitutes meaningful signals in different parts of the brain, or in other words, what is the physical embodiment of the units of information, is still open. Most neural network models assume that the average frequency of action potentials is the carrier of information between neurons. This "frequency coded" representation of the information flow ignores the exact timings of the individual firings of the neurons. While frequency coding seems to be a good representation of information for some parts of the nervous system (e.g., at the neuro-muscular junction), for other areas of the brain precise time of spike arrival can make a crucial difference. Since Action Potentials (APs) are always carried along bundles of nerve fibers (axons), the AP state of a given nerve at a particular time and location along the nerve (cross section) can be regarded as a pattern of 0s and 1s (i.e. silence or APs) and such representation can be called "pattern coding". A classical example of pattern coding is that of pre-synaptic inhibition, a widespread mechanism in the brain (Kandel and Schwartz, 1985). The importance of precise timing has been demonstrated in several studies (Segundo et al., 1963; Segundo and Kohn, 1981; Carr and Konishi, 1988). Recent evidence of phase-locked activity in the visual cortex is also very suggestive (Gray et al., 1989; Gray and Singer, 1989; Gray et al., 1990; Eckhorn et al., 1988).

DETE uses this second type of representation -- pattern coding. Henceforth, what is important for DETE is not the average frequency of APs along a wire but the relative timing of the APs.

13.2 Visual Perception

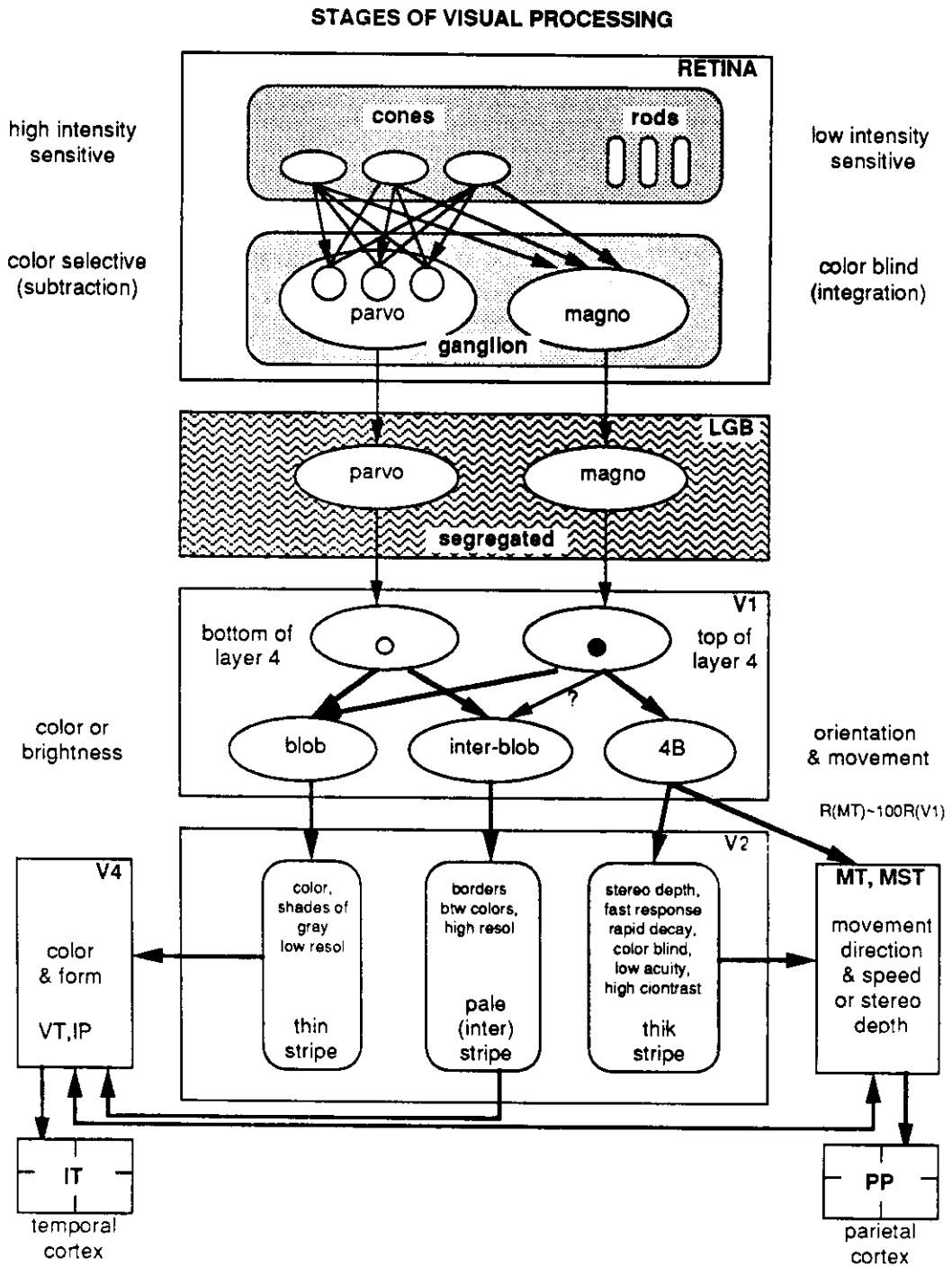


Figure 13.1: Functional architecture of the visual pathway

The retinal ganglion cells are the ultimate source of all information flow from the retina to the lateral geniculate bodies (LGBs). The cells in the LGBs as well as the ganglion cells are of two types which differ in size, connectivity patterns and function -- small cells (parvocellular division) and large cells (magnocellular division). The LGBs project to the primary visual cortex (area V1). The parvocellular division projects to "blob" and "inter-blob" areas while the magnocellular division projects primarily to area 4B. These three cellular divisions in turn project to the "thin stripe", "pale(inter)stripe", and the "thick stripe" areas of the secondary visual cortex (V2). Here functions are segregated as shown in the figure. Further projections lead to higher associational visual cortices in the occipital, temporal and parietal lobes (IT -- inferotemporal cortex, PP -- posterior parietal cortex).

The human ability to see is made possible by the brain's capacity to process enormous quantities of information simultaneously. In the already classical papers of Hubel and Wiesel (Hubel and Wiesel, 1974; Hubel and Wiesel, 1977), and more recently Livingstone (Livingstone, 1988) it was suggested that visual signals are segregated and processed by at least three separate processing systems in the brain, each with its own distinct function. One system processes information about shape; a second, information about color; a third, information about movement, location and spatial organization (Figure 13.1). The three systems can be physiologically segregated at the level of the lateral geniculate nuclei (LGN). Within each of these tracts the visual information goes through a hierarchy of semi-independent processing stages (Van Essen and Maunsell, 1983). There are significant back-projections between different levels of this hierarchy as well as some cross-connections between stages whose functions are not well known (Koch, 1987). These observations triggers an important question about the nature of object representation as a whole in the visual system.

The visual features listed above are extracted at different levels of the visual processing system in the brain. In its present version, DETE is not concerned with the complexities involved with the extraction of each feature in the brain. Only the magnocellular system and especially its encoding of shape, motion and location properties are modeled. However, of importance is the structure of the final representations produced by each visual feature extraction module.

13.2.1 The retina

DETE's simulation of the visual system can be assumed to start at the level of the ganglion cells of the retina whose axons form the optic nerve. The rest of the retinal processing stages are skipped, assuming that the form in which the visual input is represented corresponds more or less to the type of information flow generated by the ganglion cells and transmitted along the optic tract. At this level, an assumption is made that there is a retinotopic pattern encoding of the input image. In other words, pixels of the VF (which correspond to ganglion cells) carry trains of spikes depending on the existence of an object projection at this location of the "retina".

13.2.2 Segmentation -- figure/ground separation

Before we can recognize an object, "figure" must be segregated from "ground" -- one must somehow pick out regions that are likely to correspond to distinct objects. A figure must be selected on the basis of physical properties of the visual input, such as regions of homogeneous color or texture, or contiguous zero-crossings in the second derivative of the function relating intensity to position (which occur at the edges of objects). That is, because one has not yet identified the object (segregating its form from the background is a logical prerequisite to recognition), one can only use physical parameters to parse figure from ground. There are numerous proposals in the computer

vision literature for ways of organizing input into regions likely to correspond to figures (Ballard and Brown, 1982).

A theory that provides a physiologically plausible answer to the segmentation problem was proposed by Ungerleider and Mishkin (Ungerleider and Mishkin, 1982). According to this theory the visual system has two separate mechanisms: a “ventral” and a “dorsal system”. The “ventral system” forms representations of an individual part or the overall shape envelope of an object. Because the receptive fields of the cells in this system are large, it does not register the locations of parts; hence, only one part can be processed at a time (otherwise it would not be able to tell when two objects of the same type are present (i.e. the front and back wheels of a car). The representation of shape used in the ventral system should be concrete, capturing the precise shape and surface details of the part or object. This system does not process relations among parts, except insofar as they are implicit in a low resolution representation of the entire object (a kind of blurry blob-like form). In contrast, the “dorsal system” seems to be able to derive abstract representations of the spatial relations among parts or objects. The use of categorical representations of spatial relations (e.g., “connected to”) is especially appropriate for classes of objects whose members are subject to non-rigid transformations. In such cases, the parts can be arranged in a large number of topographical configurations.

The most straightforward solution to the segmentation problem requires a minor revision to the Ungerleider and Mishkin theory. It seems clear that “what” and “where” are not so distinct conceptually: sometimes the spatial relations among the parts are critical for identifying the form. Rather than “where,” the dorsal system seems specialized for representing spatial relations, including those among parts of a single object. The relations among high-resolution representations of parts presumably are represented the same way as are the spatial relations among separate objects in a scene.

In DETE the problem of visual segmentation is much simpler since DETE looks only at simple shaped objects that are non-overlapping and well defined in terms of boundaries and colors. The segmentation is done procedurally by the Input Segmentation Mechanism (ISM) (see section 7.2.1).

13.2.3 Motion representation

The perception and processing of motion in the brain has been studied extensively in the past few decades (Koch and Poggio, 1985). Reichardt’s original work (Reichardt, 1970) on motion detection in the fly retinae looked at the timing difference between two adjacent detectors to determine direction of motion. The information about visual motion in primates at the level of the retina is represented as time differences in the firing pattern of axons in the optic nerve. At the level of the visual cortex, the relative timing information is used to drive cells that respond best to edges that are moving in particular direction. Semir Zeki (Zeki, 1976) has found an area labeled MT located in the middle temporal lobe (MTL) in monkeys (see bottom right corner of Figure 13.1), which has a high proportion of cells sensitive to movement or stereoscopic depth.

DETE’s motion feature extractor is very simple in comparison to the known brain circuitry involved in the perception of motion. This procedural module calculates the difference between location of the center of gravity of an object between successive visual frames. The system handles only solid objects. A more realistic neural implementation is desirable.

13.2.4 Shape representation

Psychological experiments on object recognition have shown that the shape of an object is the most significant of all visual features (Biederman, 1987). The visual system processes images at different spatial frequency bandwidths. Higher spatial frequencies correspond to more light/dark alternations per degree of visual angle; thus, higher resolution is required to detect higher spatial frequencies. The shape-extracting module of the human visual system can be described as having a number of different "channels," each differing in resolution. At average viewing distances, the lowest spatial frequency channel produces an output that will often correspond to the general shape envelope of an object. Neurons in area V4 (see bottom left corner of Figure 13.1) have been found to be sensitive to shape and color.

DETE's shape extraction module is extremely primitive. As it was mentioned before it was implemented procedurally, i.e. not as a neural network. Therefore, a meaningful comparison of this module's functionality to the known brain mechanisms for shape processing is not possible. However, in the process of further development of the system it is very desirable to redesign this module so that it is more neurally realistic.

13.2.5 Location (position) variability representation

One and the same object often occurs at various locations in the Visual Field. Nevertheless, once we have seen an object, we can recognize it just as easily when it subsequently is in a different position in the Visual Field. A number of mechanisms have been proposed to solve the problem of position variability. Marr suggested that the appearance of objects is stored in object-centered representations (Marr, 1982). In such representations, the locations of parts of objects are specified relative to other parts, not positions in space. The solution to the problem of position variability adopted by the primate visual system is evident in the neurophysiological literature. Namely, in primates the visual cells in area TE (near the anterior end of the inferior temporal lobe) have very large receptive fields, and respond when patterns are present over a large range of positions (the receptive field size is usually larger than 20 x 20 degrees of visual angle). This area of the brain is crucial in recognition per se (Mishkin, 1982). Therefore, primates rely on not representing the position of a pattern in the high-level shape representation system. One implication of this solution is that only one shape can be recognized at a time although we can rapidly switch back and forth between stimuli. If multiple stimuli are processed simultaneously, the large receptive fields would result in the system's inability to tell whether there is one or two stimuli of the same kind present in different locations. Hence, figure/ground segregation is necessary to isolate individual patterns before they can be processed further. If such segregation is done, then duplicate patterns can be isolated and processed separately, preventing confusions about how many instances of a pattern are present in the field.

However, when we see an object, we automatically know where it is with respect to other objects in the scene. Therefore, there must be a separate representation of the object's location, which implies two separate mechanisms -- one to represent the object's shape independently of its position and one to represent its position. Ungerleider and Mishkin have shown evidence for "two cortical visual systems" (Ungerleider and Mishkin, 1982). They claim that the ventral system, running from area OC (primary visual cortex) through area TEO down to area TE, is concerned with analyzing *what* an object is, whereas the dorsal system, running almost directly from circumstriate area OB to area OA and then to area PG (in the parietal lobe) is concerned with analyzing *where* an

object is. There are well-known neural connections running along both pathways, and the visual properties of these areas have been well documented (Mishkin and Ungerleider, 1982).

The approach taken in DETE is consistent with the above mentioned observations. Namely, the representation/recognition of object's shape is done separately from the representation/recognition of its location. For this purpose two different visual feature memories are used.

13.2.6 Visual associations

The assembly of an object representation from its extracted features is performed by the visual association cortex. Perception requires that the various features extracted by the different subsystems are further organized in such a way that related ones (those belonging to a single object) are grouped together. The Gestalt psychologists (Wertheimer, 1923; Koffka, 1935) suggest that visual perceptions are possible because the brain uses certain visual properties to group the parts of an image together and also to separate objects from one another and from their background. Such organizing properties are, for instance: location and velocity of motion (e.g., elements that move together probably belong to the same object); co-linearity (e.g., a house is not perceptually split in two when a telephone wire crosses in front of it); depth, lightness and texture. The fact that these functions all fail at equiluminance (when different objects have equal luminance) suggests that the ability to link parts of scenes together, to discriminate figure from ground and to perceive the correct spatial relationship of objects might be carried by the magno system (Livingstone, 1988) (see right-hand side of Figure 13.1) which is performed at the level of the visual association cortex. The main function of the visual association cortex is to learn (cluster and categorize) the relations between the visual features of the objects such as shape, size, motion, location, and color. It is evident that such a learning process takes place independently of any verbal processes. Such functional independence is advantageous for any language learning, since it allows for *a single trial* (or only a few trials) verbal/visual association which speeds up the verbal learning.

Neuroscience research has revealed many details of the neural mechanisms involved in low-level vision (e.g., stereopsis, clustering of visual features, segmentation, feature detection). However, nearly nothing is known about high-level (model-level) vision in humans including recognition and learning of different images and extraction of their meaning. For instance, it is not known where in the brain the image of a chair (objects in general) are represented and how. Also it is not known where in the brain visual actions (e.g., grasping, or eating, or bending) are represented. When a child reads a picture book like "Sleeping Beauty", it uses the (static) images to help learn the meanings of the words. The child sees a prince swinging a sword at a dragon and hears "fight the dragon". The child needs a system to recognize and learn general as well as story features that are representationally more abstract.

The neural substrate involved in object assembly may consist of convergence zones in the brain (Damasio, 1989). These convergence zones could be neurally represented by subsets of neurons located in the neocortex, the thalamus or the claustrum which project back from higher to lower level structures (Crick and Koch, 1990).

In DETE, the various memory modules serve as associative mechanisms in which the "gestalts" are formed. As discussed before, the essential mechanism which accomplishes this gestalt formation is based on phase locking of oscillating neural assemblies each of which represents individual components of the whole gestalt (mental image).

13.2.7 Imagery

Visual mental imagery is a kind of “seeing” in the absence of the appropriate immediate sensory input; it is “re-seeing” and “re-creating” of information previously seen and stored in the memory (not necessarily exactly in the same way as it was experienced initially). Imagery is an important component of the process of thinking. For instance, one can anticipate the trajectory of a moving object by mentally projecting its path. This is a type of visual inference-making process. We can also discover an efficient way for packing irregularly-shaped objects into a container by imagining them arranged in different ways. In other words, we are able to voluntarily manipulate an internal visual scene in time and space. Imagery is also often used when one tries to answer questions from memory about subtle visual properties of objects, resulting in one’s imaging an object and “recognizing” previously-unrecognized (i.e. internally non-verbalized) features. For example, we tend to use imagery when we try to answer questions like: “*What shape are a beagle’s ears?*” (Kosslyn, 1985) or “*Which is larger, a goat or a hog?*”. It seems that in such cases we are recycling visual memories, internally “looking” at the object again and recognizing (verbalizing for ourselves) previously-unrecognized properties.

In DETE, visual imagery is an inherent property of the system. Imagery occurs always when a meaningful (i.e. previously learned) verbal input is presented. For instance, DETE “imagines” a red ball above a green triangle when it hears the verbal input “Red ball is above a green triangle”. While in humans we do not have technical means to observe the image generated in somebody’s “mind’s eye”, we can easily monitor the activity generated in DETE’s visual feature memories and effectively have an access to DETE’s “mind’s eye”. (Notice that DETE does not have a separate “mind’s eye”.) Knowing how information is encoded in these visual feature memories, we can interpret such activity in terms of various internal images.

For the image generation module (in DETE this is the visual memory) to be useful, some other part of the system needs to receive it as input. Neuroscience research tells us that there are direct connections from the high-level visual cortices to polymodal association cortices (e.g., the angular and supramarginal gyri) where visual information is associated with auditory and somatosensory information. In DETE the output of the visual memory is fed directly to the verbal memory bank.

Humans can combine symbolic thoughts in novel ways (e.g., “the giant hamburger from outer space ate UCLA”). This combinatorial ability is essential for language (animals do not seem to have it). To be able to do that, a language processing system needs to go beyond the level of simple symbol grounding. Essentially it needs to be able to generate novel visual representations on the basis of verbal input for which there is already grounding in some sensory experiences. At this stage of development, DETE is not capable of doing that (has not been tested yet). Unfortunately, the neurolinguistic research has not yet found satisfactory answers about how such verbal to verbal associations might come about in the brain.

13.3 The verbal subsystem

Looking at neuroanatomical, electrophysiological, neuropsychological and lesion data one can easily get the impression that almost every part of the brain is involved in processing of one aspect of language or another. The reason for this is that language has various components (e.g., perceptual vs. expressional or cognitive vs. affective, etc.) the processing of which is done by various systems of the brain such as the sensory systems (language inputs), the motor systems (language production), the limbic system (motivation for language performance -- perception/production), and

the higher associational areas (language comprehension and planning). This section discusses briefly: (1) The modules that compose the language system in the brain and describes the information flow through the network which they form. (2) The impairment of various language functions as a result of localized brain lesions. (3) Some representative models which explain these data. (4) A mapping between language-related brain areas and functionally corresponding modules in DETE.

13.3.1 Input/Output and central processing of language

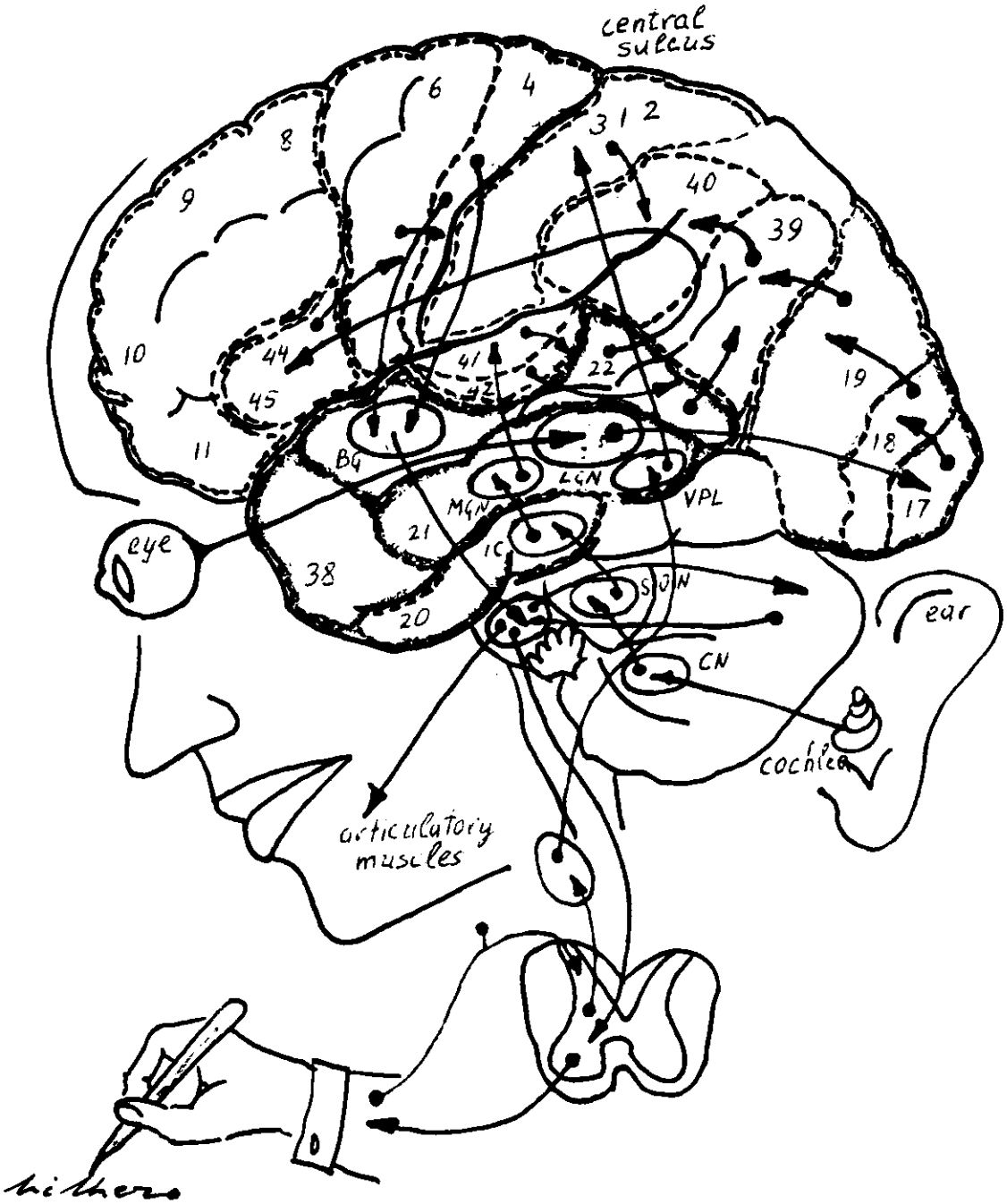


Figure 13.2: Language-related areas in the brain

The neocortex is functionally divided (Brodmann's classification) into: visual areas -- 17 (V1), 18,19 (V2); auditory areas -- 41 (A1), 42 (A2); somatosensory areas -- 1,2,3,5 (S1,S2); motor and premotor areas -- 4,6,8; posterior association areas -- 39 (angular gyrus), 40 (supramarginal gyrus); language-specific areas -- 22 (Wernicke's area), 44,45 (Broca's area). subcortical relay nuclei along various neural pathways are shown as ovals and labeled by their standard abbreviations.

Reviews of brain areas related to speech and language can be found in (Penfield, 1966; Millikan et al., 1967; Zurif, 1990; Caplan, 1980; Studdert-Kennedy and (Ed.), 1983; Caplan et al., 1984; Ojemann, 1991). While some of these regions have been investigated in great detail, the actual circuitry in these cortical and subcortical areas is mainly unknown. Figure 13.2 shows in a schematic way the major brain areas involved in language processing. Details are given in the sections that follow.

Language Input systems

Information which is verbal in nature can enter the brain via at least three distinctive sensory channels: (1) The auditory channel for spoken language. (2) The visual channel for written or signed language. (3) The somatosensory channel which allows blind people to read brail using the tactile receptors in the skin of their fingers. These sensory systems are not dedicated uniquely to language but are used for a variety of sensory inputs.

Language Output systems

Along with the existence of a variety of ways to get language input to the brain, there are several ways how language can be expressed. (1) Spoken language. (2) Written or typed language (3) Signed language. Similarly to the language input systems, the language output systems are not used uniquely for language.

Language-specific systems

On the basis of numerous clinical studies of various forms of aphasia (motor and semantic language dysfunctions caused by stroke, brain trauma, tumor, Alzheimer's disease, etc.), dyslexia (reading impairment), and aprosodias (impairment of the affective components of language), a number of functionally unique and locationally different areas in the brain have been identified. The major language-specific cortical areas and subcortical gray matter nuclei are described below. They can be broadly grouped in areas involved in processing of the cognitive aspects of language (e.g., semantics, syntax, morphology, etc.), and areas subserving the affective aspects of language (e.g., prosody, emotional gesturing, etc.).

Cortical Areas:

Areas subserving the cognitive components of language:

The major cortical areas which are involved in processing of the *cognitive* aspects of language are shown in (Figure 13.2). They include: (1) The *motor speech center* of Broca (Brodmann's areas 44, 45); (2) The *auditory speech center* of Wernicke (posterior part of Brodmann's area 22); (3) The *lexical storage centers* (Brodmann's areas 20,21,37,38); and (4) The *center for reading* (Brodmann's areas 39,40). All of these centers are highly interconnected and to a great extent their

functions can be overlapping, i.e. verbal processes are dynamically localized. Following are brief descriptions of the basic language specific cortices:

(1) Broca's area: is located in the posterior part of the inferior frontal gyrus (Brodmann's area 44, 45) (Broca, 1865). It is adjacent to the motor cortex which controls the movements of the lips, tongue, jaw, palate, vocal cords, and diaphragm. This area seems to be a syntactic center for language comprehension and production (Caramazza and Zurif, 1976; Zurif, 1990). It is also possible that Broca's area contains the neural network which functions as an Order Memory (Sabouraud, 1981).

(2) Wernicke's area: is located posterior to the auditory cortex in the superior temporal lobe (posterior part of Brodmann's area 22) (Wernicke, 1908). It is responsible for the generation of the underlying structure of an utterance. This structure is then transmitted through the arcuate fasciculus to Broca's area, where it evokes a detailed and coordinated program for vocalization. The program is passed to the adjacent face area of the motor cortex, which activates the appropriate muscles of the mouth, the lips, the tongue, the larynx and so on. Wernicke's area is important not only in speaking but also plays a major role in the comprehension of spoken word and in reading and writing. This area might be responsible for sustaining of semantic inference (Zurif, 1990).

(3) Lexical storage centers: are located in the anterior inferotemporal cortex (areas 20,21) and in the posterior temporal region (areas 37, 38). These cortical areas have been implicated in anomia (difficulty in finding words as a result of lesions -- naming problem) (Damasio, 1990).

(4) Reading center: is located in the angular and supramarginal gyri in the occipito-parietal junction of the left hemisphere. Lesions in these areas can cause profound alexia and agraphia (lost of ability to read and write) (Dejerine, 1891).

Areas subserving the affective components of language:

In addition to the propositional (cognitive) aspects of language represented in the left hemisphere (i.e. in right-handed people in which the left hemisphere is language dominant), homologous areas to Broca's and Wernicke's areas were found in the right hemisphere (Ross, 1981). These right-hemispheric areas support the *affective* components of language including musical intonation of speech (prosody), and emotional gesturing (right Broca's area), as well as prosodic comprehension and comprehension of emotional gesturing (right Wernicke's area). These two areas are connected via the right arcuate fasciculus. The cognitive language areas located in the dominant hemisphere are vastly interconnected with the affective language areas in the non-dominant hemisphere via the corpus callosum.

Subcortical and other brain areas

The neocortical areas in the left and right cerebral hemispheres are not the only areas in the brain involved in language. Yet in the early days of neurophysiology of higher cognitive functions it had been recognized that a number of extra-cortical brain structures such as the thalamus, the basal ganglia, and the reticular formation are heavily involved in different aspects of language processing (Penfield and Roberts, 1959). According to Penfield, the thalamus, and the basal ganglia (caudate nucleus, putamen, globus pallidus) are the main subcortical centers responsible for the integrative processes underlying language generation. The basal ganglia, for instance, receive inputs from a wide-spread areas of the postcentral neocortex (sensory and associational) and project to large areas in the frontal lobes (specifically motor, premotor, and supplementary motor areas) (Côté and Crutcher, 1985).

13.3.2 Language disorders (impairment by lesions)

Aphasias are various language disorders which arise as a consequence of focal brain lesions. Aphasias have been studied extensively by neurologists, linguists, and cognitive scientists. A brief summary of the major types and their characteristics is given below. A more detailed account of the major language and speech disorders along with their characteristics and implicated brain areas is presented in Table 13.1. For reviews on aphasias and related disorders see (Benson, 1979; Schwartz, 1985).

Broca's aphasia: A lesion in Broca's area disturbs the production of speech but has a much smaller effect on comprehension. In Broca's aphasia, speech is labored and slow and articulation is impaired. Construction of more complex grammatical structures is impaired (agrammatic) and as a result the speech has a telegraphic style.

Wernicke's aphasia: Damage to Wernicke's area on the other hand, disrupts all aspects of language usage. In Wernicke's aphasia speech is phonetically and even grammatically normal but it is semantically deviant. Words are often strung together with considerable facility and with the proper inflections. The utterances have the recognizable structure of sentences, however, the choice of words is often inappropriate. Usually an utterance as a whole may express its meaning in a remarkably round-about way.

Conduction aphasia: Destruction of the arcuate fasciculus, disconnecting Wernicke's area from Broca's area, leaves speech fluent and well articulated but semantically aberrant. Lesions to the angular gyrus have the effect of disconnecting the system involved in auditory language and written language. This results in an almost intact ability to speak and understand spoken language but impaired ability to read.

In table 13.1, the phenomenon of language has been divided into *comprehension* (from input into internal representation), *production* (from internal representation into output), and *language tasks* (from input, via internal representation to output). The comprehension and production are further subdivided according to the sensory modalities involved. X-ed fields mark the language features which are impaired in a particular disorder, whereas blanks mark the language features that are intact. The numbers in brackets are pointers to the following references: [1] (Broca, 1861), [2] (Wernicke, 1874), [3] (Dejerine, 1891), [4] (Ross, 1981), [5] (Heilman, 1975), [6] (Liepmann, 1914), [7] (Damasio, 1990), [8] (Goodglass and Kaplan, 1972), [9] (Lecours et al., 1983), [10] (Caramazza and Zurif, 1976), [11] (Heilman and Scholes, 1976). The question mark (?) means "not known".

| disorder | Comprehension (input → level representation) | | | | Production (internal representation → output) | | | | | | | | | | Team (input → output) | | | | areas of disorder | brain location |
|---|--|-------------------|-----------|-------------------|---|-----------------|-----------------------|--------------------|--------------|--------------|--------------|--------------------|--------------------|-----------------------------|--------------------------|--------------------------|-----------------------------------|--------------------------------|-------------------|---|
| | reading fluently (number) | content (English) | | paralel (picture) | writing (content) | writing (rhyme) | writing (high vowels) | phonological level | | | | orthographic level | phonological level | repetition (input → verbal) | reading (input → verbal) | copying (input → verbal) | reading fluently (input → verbal) | comprehending (input → verbal) | | |
| | | word & semantic | syntactic | | | | | semantic | orthographic | phonological | orthographic | | | | | | | | | |
| Wernicke's (R) | X | X | | | X | | | | | | | | | | X | X | X | | NO | Left posterior superior temporal lobe (Wernicke's area) (22) |
| Broca's (L) | | | X | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | YES | Left motor association cortex in frontal lateral Broca's area (44, 45) |
| Conduction (B) | | | | | | X | | | | | | | | | | | | | YES | arcuate fasciculus supramarginal gyrus (46) |
| Anomic (L) | | | | | | | | | | | | | | | | | | | | posterior left inferior temporal lobe |
| Global | X | X | X | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | NO | Broca's, Wernicke's & arcuate fasciculus perisylvian region |
| Transcortical motor sensory | X | | | | X | | | | | | | | | | | | | | | perisylvian region |
| border zone (watershed area) Left anterior Broca | | | | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | | border zone (watershed area) Left anterior Broca |
| border zone & parietal, occipital temporal junction | X | X | X | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | ? | border zone & parietal, occipital temporal junction |
| Right posterior hemisphere | | | | | | | | | | | | | | | | | | | | Right posterior hemisphere |
| Right anterior hemisphere | | | | | | X | | | | | | | | | | | | | | Right anterior hemisphere |
| Left inferior cortex & subcortical (angular gyrus or supramarginal gyrus) | X | | | | | | | | | | | | | | | | | | | Left inferior cortex & subcortical (angular gyrus or supramarginal gyrus) |
| angular gyrus (39) | | | | | X | | | | | | | | | | | | | | | angular gyrus (39) |
| Parkinson's disease (basal ganglia) | | | | | | | | | | | | | | | | | | | | Parkinson's disease (basal ganglia) |
| cardiolum | | | | | | X | | | | | | | | | | | | | | cardiolum |
| corpus callosum (left right asymmetry) | | | | | | | | | | | | | | | | | | | | corpus callosum (left right asymmetry) |

Table 13.1: Language dysfunctions

13.3.3 Models of Language processing in the brain

With the advances of knowledge about the relation between brain and language complexity, sophistication and the diversity of theoretical models has increased. To give a flavor of the modeling efforts, two of them are discussed below.

The Wernicke-Geschwind model

From the analysis of different language-related deficits, Karl Wernicke formulated the first model of language processing in the brain (Wernicke, 1906; Wernicke, 1874/1977) (Figure 13.3). Wernicke hypothesized that the language faculty was dependent upon the ability to manipulate sensory and motor images of words. In this model the storage of the sensory images of words was attributed to Wernicke's area whereas the motor representations of words were held in Broca's area. When a word is read, the visual pattern (from the primary visual cortex) is transmitted to the angular gyrus, which applies a transformation that activates the auditory representation of the word in Wernicke's area. There is a substantial neurophysiological evidence that the comprehension of a written word seems to require that the auditory representation of the word be evoked in Wernicke's area. Damage to the angular gyrus seems to interrupt communication between the visual cortex and Wernicke's area, so that the comprehension of written word is impaired. Writing a word in response to an oral instruction requires information to be passed along the same pathways in the opposite direction: from the auditory cortex to Wernicke's area to the angular gyrus.

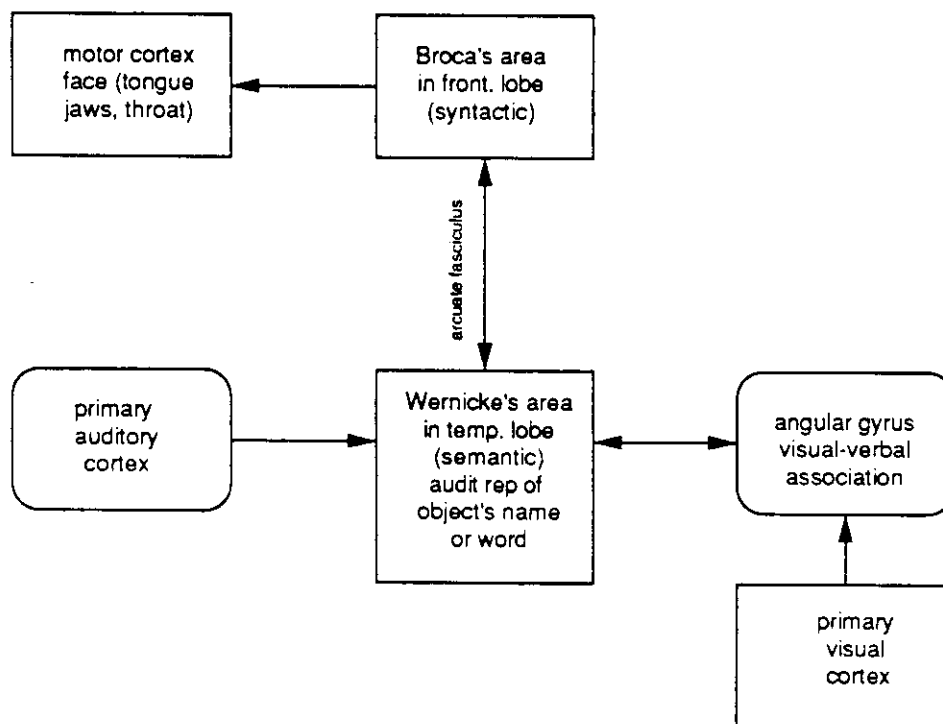


Figure 13.3: Block diagram of language processing in the brain

Language specific areas in the left hemisphere. The main cortical areas involved in language processing according to this model are: 1) the primary auditory cortex; 2) Wernicke's area; 3) Broca's area, and 4) the angular gyrus.

Wernicke's model was further elaborated by Norman Geschwind (Geschwind, 1970) and is still fairly adequate today. However, as was noticed by numerous researchers, this model fails to account for various aspects of language disturbances like the tendency for omission of function words and the tendency to nominalize verbs in Broca's aphasics (Goodglass and Kaplan, 1972; Lecours et al., 1983).

Coltheart's model

An example of a more advanced process model of language recognition, comprehension, and production which explains most of the data from various language deficits is the model proposed by Coltheart (Coltheart, 1987) (Figure 13.4). This model of the functional architecture of the language processing system includes processing modules and connectivity patterns that can collectively account for observed aphasia, alexia, agraphia, and other language deficits. The explanations which the model supports are in terms of decouplings between modules or disruption (lesioning) of modules.

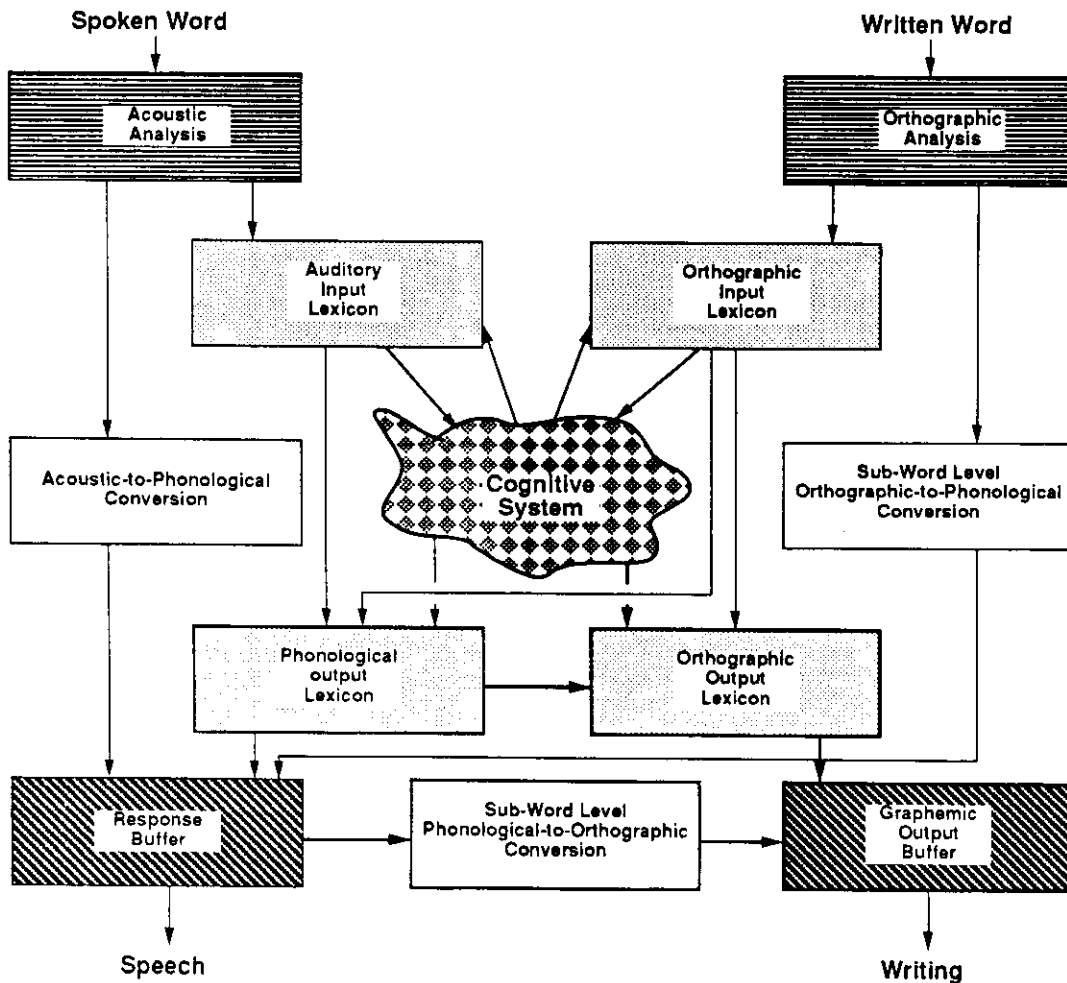


Figure 13.4: Coltheart's model

A simple process model for the recognition, comprehension and production of spoken and written words and non-words. Contains: (1) Preprocessors for acoustic and orthographic analysis (top left and right); (2) Input memories -- auditory and orthographic lexicons (immediately below); (3) Data conversion modules -- orthographic-to-phonological & acoustic-to-phonological (middle left and right); (4) Output memories -- phonological & orthographic lexicons and buffers; (5) Cognitive system (in the middle) -- coupled reciprocally to most of the memory modules. (Adapted from Coltheart, 1987.)

Some of the major shortcomings of this and similar models are: (1) They are not computational. In other words, there is no choice of specific representations of the processed data, no concrete mechanisms proposed for the individual modules, etc. Therefore, testing the validity of these models is difficult and their predictive power is limited. (2) The putative functional components are not mapped to actual neural circuits in the brain that subserve the proposed functions. (3) The levels of specificity in the functional description of the individual modules are not comparable (e.g., "cognitive system" vs "response buffer" -- Figure 13.4).

13.3.4 Mapping of DETE's verbal modules to the brain

A coarse correspondence can be established between DETE's modules and the language specific areas in the neocortex of the left hemisphere: (1) DETE's Word Encoding Mechanism (WEM) corresponds collectively to the auditory pathway which includes: ear, cochlea, cochlear nucleus, superior olivary nucleus, inferior colliculus, medial geniculate nucleus, primary and secondary auditory cortices (Heschl's gyrus -- A1, A2). (2) The lexical (verbal) memory corresponds to the Wernicke's area and the anterior temporal areas. (3) The Order memory for morphologic and syntactic processing corresponds to Broca's area. (4) The EYE followed by the visual feature extractors and the visual memory modules correspond to the visual pathways which include: retina, lateral geniculate nucleus, primary visual cortex (V1), secondary visual cortex (V2), etc.

13.3.5 Hidden speech

One of the very interesting but not extensively explored forms of interaction between language and thought is the so-called "inner speech phenomenon". Psychologists usually characterize "inner speech" as soundless mental verbalization, which occurs at instances when we are thinking about something, planning or solving, reading or writing (Sokolov, 1972). In all these cases, part of our mental processing is done in the form of hidden articulation -- we talk to ourselves. One of the first attempts to study this phenomenon was made by L.S. Vigotskij in 1934 (Vigotskij, 1970). A possible reason for leaving this phenomenon out of the main track of language research seems to be the common belief that "hidden articulation" is simply a side effect of the language generator rehearsing (anything we think we may have to put into words at some point), so hidden articulation is not important in thinking, but is mainly important for expressing thoughts that come from elsewhere.

Hidden articulation is manifested in DETE in situations when DETE receives a visual input without verbal input. In such cases, if we observe the activity in the verbal memory bank generated by association with the activity in the visual memory banks, we often can interpret this "hidden" verbal activity as individual words or phrases. While in DETE I have not explored the effect of this "hidden" verbal activity on the ongoing activity in the visual bank, I expect that it might be significant. In other words, the "hidden articulation" may influence the way DETE perceives the visual world. In humans such interactions may be significant during reading of text. If a linear string of letters (e.g., words forming a sentence) falls on our retina (Visual Field), maybe the first

internally verbalized word in the sentence drives our oculomotor apparatus to initiate a specific sequence of saccadic motions from one word to another. I would suggest, that if the hidden articulation mechanism is not engaged when we have a text in front of our eyes, then the mere process of moving the eyes over the text does not result in reading it (i.e. transforming of the visual input into a meaningful internal representation). One may argue, however, that hidden articulation is not commonly present during visual perception. In other words, we do not constantly articulate for ourselves the names of the multiple familiar objects (whose names we know) whenever they fall in our visual field. However, interestingly enough, children in the early stages of their language development seem to do that spontaneously. For instance, at the sight of a familiar object in a picture book, a child will often spontaneously name the object. While this is not hidden but rather overt verbalization, one may expect that in the rest of the situations, when the child sees the same familiar object but does not name it allowed (maybe the child has a pacifier in its mouth), it nevertheless articulates the name in its mind.

13.3.6 Temporal aspects of language processing

Text reading and writing are sequential processes (at least during the input and output stages). Traditional AI natural language processing (NLP) programs also exhibit some sequentiality in the processing of language, e.g., they read and generate text from left to right. After each word or phrase is read by the system, it is immediately subjected to some lexical analysis. At a first glance, it may appear that humans are functioning similarly. However, one important difference between humans and NLP AI programs is that people perform text reading and generation in real time and the speed of text reading affects the level of comprehension. For instance, the faster we read, the less we usually manage to understand. On the other hand, if reading is done very slowly comprehension also deteriorates and can cause ambiguities (e.g., "I want a can ... opener"). Traditionally, computer programs modeling human language comprehension do not operate in real time. In other words, there is no time limit for the processing of each word. Such programs usually process each word to the extent that is desired by the programmer before the processing of the next word begins. Feeding words faster or slower to such a system does not affect the performance (words are usually stored in some buffer and the processing speed depends on the speed of the computer and the complexity of the lexical analysis).

Compared to such traditional NLP systems, DETE depends critically on the time allocated for the processing of each word or phrase relative to the time allocated for input and output processing or for processing in the other modules (e.g., visual, attentional, and motor). Description and physiological/psychological justification of the hierarchy of temporal relations in DETE is given in Chapter 5.

13.4 Neural basis of Selective Attention

Multiple objects are usually present in the Visual Field when our eyes are open. However, we deal only with a fraction of them at any instant (usually one or two). For instance, psychophysical studies of the eye movements during the visual exploration of a scene have revealed that the eyes are directed sequentially to various locations of the scene. The shift from one location of eye fixation to another (saccadic motion) happens very rapidly (3-5 saccades per second). As a result, the information contained in the scene is passed to our visual processing systems selectively and sequentially. This ability is due to an attentional mechanism whose function is to control the quantity and the temporal order of information chunks through the system.

There are two main issues in constructing a visual selective attention mechanism:

(1) *Where to focus* -- which of the many objects in the Visual Screen should the system attend to and what mechanisms control the movement of attention from one object to another, also how long should the attention be kept at a given object?

(2) *How to focus* -- this is a question of what is the “meaning” (i.e. neural representation) of the attentive process with respect to the other components of the system.

These two issues, as they relate to DETE, will be discussed separately in light of the currently available models in psychology and neuroscience that address them.

13.4.1 Control of the attentional focus

The focus of attention is often referred to as the “spotlight” of attention (e.g., Eriksen & Hoffman 74; Crick 84; Koch & Ullman 85; LaBerge & Brown 89; Posner 80) (Eriksen and Hoffman, 1974; Crick, 1984; Koch and Ullman, 1985; LaBerge and Brown, 1989; LaBerge and Brown, 1989; Posner, 1980). The focus of attention can be driven either by the available data in the external environment (*data driven*) or by the state of the internal environment of the system (*conceptually driven*). The neural mechanisms underlying (a) the data driven and (b) the conceptually driven attentional control are different. This dichotomy of focus control has a long history in the psychological literature. For instance, Milner (1974) distinguishes *extrinsic* and *intrinsic* control of attention; Butter (1987) distinguishes *reflexive* and *voluntary* control; LaBerge and Brown (1989) use the terms *bottom-up* and *top-down* control (Mozer, 1991). Both the data driven and the conceptually driven control of attention have two different components: (1) control of location of the attentional focus, and (2) control of its size.

(1) *Control of location*: -- The location of the visual attention focus is closely associated with the location of the eye(s) (more specifically the foveal part of the retina) with respect to the Visual Screen. In other words, most of the time we are attending to whatever we are looking at (i.e. whatever part of the Visual Screen is projected to the foveal part of the retina). Chapman (Chapman, 1990a; Chapman, 1990b) and Weismeyer and Laird (Weismeyer and Laird, 1990) made major steps towards a concrete model of voluntary attentional control by taking a computational approach in describing some attentional strategies and control primitives for visually guided behavior.

(a) *Bottom-up control is reflexive in nature*. It is closely related to the types of reflexive motions in which the eyes are involved. On the basis of neurophysiological and psychophysical studies one can discriminate five neural control systems that keep the fovea on the target (Gouras, 1985b). The eye movements which these systems subservise are: saccadic eye movement; smooth pursuit movement; optokinetic movement; vestibulo-oculomotor reflex; and vergence movement. The eye movements which DETE performs can be categorized as saccadic eye movements.

Saccades are rapid re-directions of the eye(s) to targets of interest in the Visual Field which result in foveation of the targets. Making saccadic motions is the natural state of the human eye during awake state. For instance, it has been demonstrated that in order to fixate the eye on a stationary object for some time, one must voluntarily inhibit the saccadic motions. Saccades are reflexive and ballistic movements of the eyes (Figure 13.5).

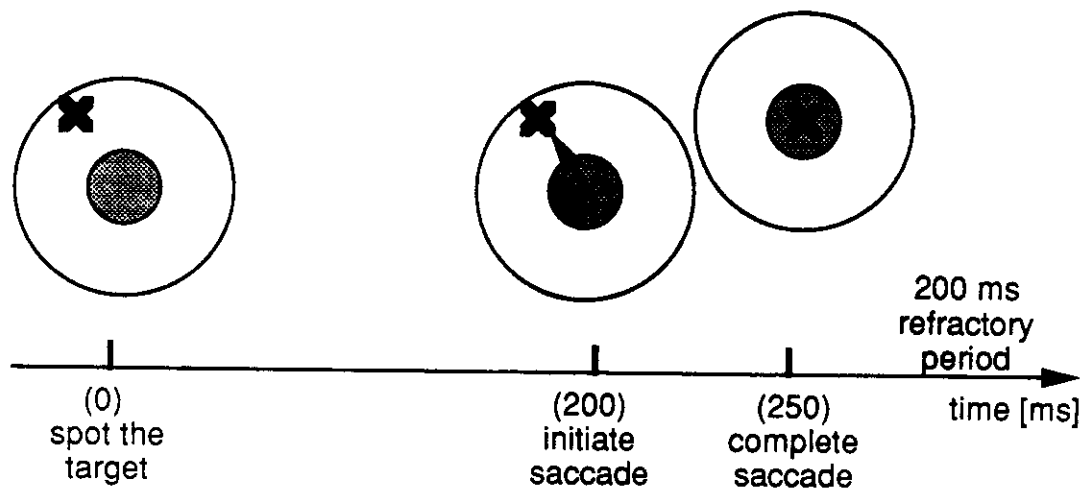


Figure 13.5: Saccadic motions of the eye

Dynamics of a saccade: A target (x) appears in the periphery (upper left corner) of the Visual Field (the large circle) at moment $t=0$ and is detected by the visual system. About 200 msec later the oculomotor system initiates a saccade (rapid ballistic movement towards the target) which is completed within 50 ms on average -- the target is now projected on the fovea (the small gray circle in the middle of the Visual Field). Following is a 200 msec refractory period during which the oculomotor system is unable to generate another saccade.

The control of saccades involves complex neural circuitry which includes two distinct cortical areas -- the *frontal eye fields* and the *occipital eye fields*, and a number of premotor nuclei located in the midbrain -- *vestibular nuclei*, in the reticular formation -- *pontine gaze center*, and *superior colliculus*. Several neural models have attempted to account for various aspects of saccadic control (Hikosaka, 1989; Desimone et al., 1989). In the present version of DETE, the saccadic system has not been modeled in detail. Instead the voluntary and reflexive components of saccadic control have been implemented as follows.

The reflexive component which is active during free exploration of the visual space or during the execution of a visual search task "Where is the red ball" has been implemented as a procedural module.

In humans, a variety of stimuli from the environment can trigger a switch in the bottom-up attentional mechanism (i.e. capture the attention). Some examples are intense or flashing light, a sudden appearance of a new object in the Visual Field, an object motion, etc. In DETE examples of such attention capturing stimuli are appearance/disappearance of an object in the Visual Field or sudden change of one or more visual features of an object, e.g., its location (during motion) or its size, etc.

(b) *Top-down control is voluntary*. In humans it depends on the current expectations triggered by the task at hand. In DETE, the basic principle underlying such mechanism is that the neural representation of the task enables only some of the features -- those which are of interest for the task to capture the attention (similar ideas were suggested by LaBerge (LaBerge and Brown,

1989), and Mozer (Mozer, 1991). During initial learning, the voluntary control is substituted by external (joystick guided) control of the retinal position on the Visual Screen. Later this control is active during the execution of a task such as the direction of the eye to a particular location in response to a verbal command.

(2) *Control of size*: -- Intuitively, it is easy to see that if the objects on the Visual Screen vary in size, so must the size of the attentional spotlight. Empirical evidence for this was provided by Eriksen and Yeh (Eriksen and Yeh, 1985) and LaBerge (LaBerge, 1983). In physiology, this process is known as accommodation of the eye. The accommodation (focusing reflex) involves three separate processes: a) increase of the lense curvature by contraction of ciliary muscles innervated by the parasympathetic part of the autonomic nervous system, b) constriction of the pupils by the pupillary sphincter muscles -- parasympathetic control, c) convergence of the eyes by the medial rectus muscles innervated by the 3rd cranial nerve (Gouras, 1985a). This reflex can be modulated by voluntary control. The neural pathways for this complex reflex arch involve a complex neural circuit that includes: the fovea, LGN, visual cortex, pretectal region, Edinger Westphal nucleus, the occipital eye fields, the superior colliculus, and other structures. The major control systems for eye movements are discussed in more detail in (Julesz, 1991; Eriksen and Murphy, 1987; Arbib, 1989).

DETE's algorithm for accommodation in DETE is described in section 6.2.1. This accommodation mechanism is different from the eye accommodation reflex in humans which is driven by detection of blurriness in the image on the retina. DETE is not provided with blurred images.

13.4.2 The neural plausibility of DETE's attentional mechanism

The visual selective attention mechanism used in DETE is based on the idea that the segmentation of visual patterns can take place in the temporal domain. This hypothesis was first suggested by von der Malsburg (von der Malsburg, 1981; von der Malsburg, 1983; von der Malsburg, 1987). According to this idea, components of each pattern become temporary synchronized while the activity between patterns is desynchronized (phase shifted). From the various type of oscillations observed in the brain such as the α -waves in the visual cortex or the θ -waves in the hippocampus, it seems that the oscillations on which attention is based are the γ -oscillations (Gray et al., 1989; Gray and Singer, 1989; Gray et al., 1990; Eckhorn et al., 1988) (40-70 Hz) typical for some complex cells in the cortex.

The idea, that coherent oscillations of different sets of stimulus-specific neurons is the neural representation of selective attention, was also suggested by Crick and Koch (Crick and Koch, 1990). According to these authors, such an attentional mechanism can serve as a foundation of a neurobiological theory of consciousness. They also suggested that a similar mechanism is involved in auditory, olfactory, as well as tactile attention.

13.4.3 Anatomical localization of Selective Attention

In humans the ability to direct attention towards sensory events within the extrapersonal space is mediated by a complex cerebral network which includes cortical and subcortical components. A schematic diagram of the brain structures subserving visual attention and their relation to ocular motion and fixation is shown in Figure 13.6.

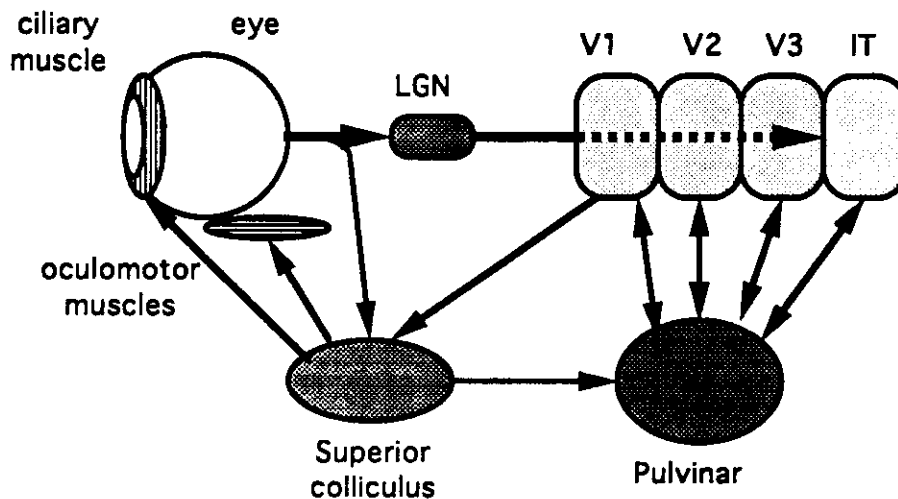


Figure 13.6: Brain areas involved in visual attention

The eye (retina) is the source of the signals which provide the basis for attentional and cognitive processing. A number of brain nuclei (Lateral Geniculate Nucleus -- LGN, Superior Colliculus, and Pulvinar) serve as relay and control stations along the signal path and during its processing in the various areas of the cortex (V1, V2, V3, IT). Feedback loops at different levels provide signals which reach the oculomotor muscles and the ciliary muscle via the superior colliculus. The muscles, in turn, accomplish the various types of eye movements and eye accommodation.

The function of the attentional mechanism is based on two major assumptions: (1) Attention is expressed in the cortex. (2) There is a subcortical control mechanism of the attention window.

(1) *Cortical expression:* -- The function of the association cortex is to bind together distributed circuits which by virtue of their simultaneous activation represent a set of facts. During sustained attention, neurons located in the inferotemporal, inferior parietal, and frontal cortices have been demonstrated to fire in response to the salient aspect of the stimulus (e.g., to redness if the task is to recognize the color of an object) (Fuster, 1980). The posterior parietal cortex of the inferior parietal lobule, the periarculate region (including the frontal eye fields), and the cingulate cortex constitute the three major components of the association cortex that contain separate representations of the extrapersonal world (Figure 13.7). The first representation, centered around the posterior parietal cortex (area PG) and perhaps extending into the high-order sensory association cortex of the superior temporal sulcus, may contain a sensory template of the extrapersonal world. The second representation, centered around the frontal eye fields (Brodmann's area 8), contains a motor map for the distribution of scanning, orienting and exploration (Schiller et al., 1979). A third representation, centered around the cingulate cortex, contains a motivational map for the distribution of interest and expectancy.

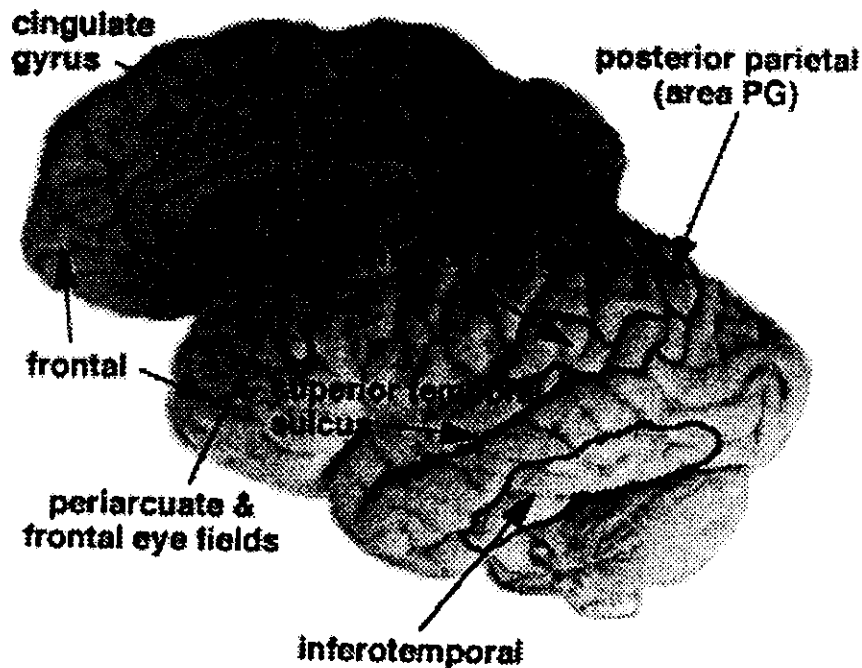


Figure 13.7: Attention-related areas in the cerebral cortex

Location of the major cortical areas where the focus of attention is expressed.

The correspondence of these three attentional representations to DETE's modules is as follows: The posterior parietal cortex (area PG) contains the focus of attention towards the external world (the visual and verbal input) -- *sensory attention*. This corresponds to the visual and verbal memory modules in DETE. The frontal eye fields correspond to part of the motor memory module involved in EYE control. The cingulate cortex could be the area where the focus of attention towards the internal world (internally generated visual images and verbal activity) is expressed -- *mental attention*. In DETE this area is also mapped to the visual and verbal memory modules.

(2) *Subcortical control*: -- It is provided collectively by a subcortical neural circuit which includes: (a) the superior colliculus, (b) the thalamus -- distributor of the activation from the reticulum, and (c) the pulvinar (Figure 13.8).

(a) The superior colliculus: -- Plays important role in the control of eye movements via the oculomotor neurons located in the oculomotor nuclei of the medulla.

(b) The thalamus: -- Abundant neuroanatomical evidence suggests that all sensory impulses, except the olfactory ones, terminate in the gray masses of the thalamus (Carpenter and Sutin, 1983). The thalamus serves as a chief sensory integrating mechanism which maintains/regulates the states of consciousness and attention. It is involved not only in the transmission of sensory inputs and tuning of the cortical output, but also in the synchronization and desynchronization of cortical activity (Purpura, 1970). Specific areas in the thalamus such as nucleus reticularis thalami (Rafal and Posner, 1987; Crick, 1984) probably play a significant role in the control of attention.

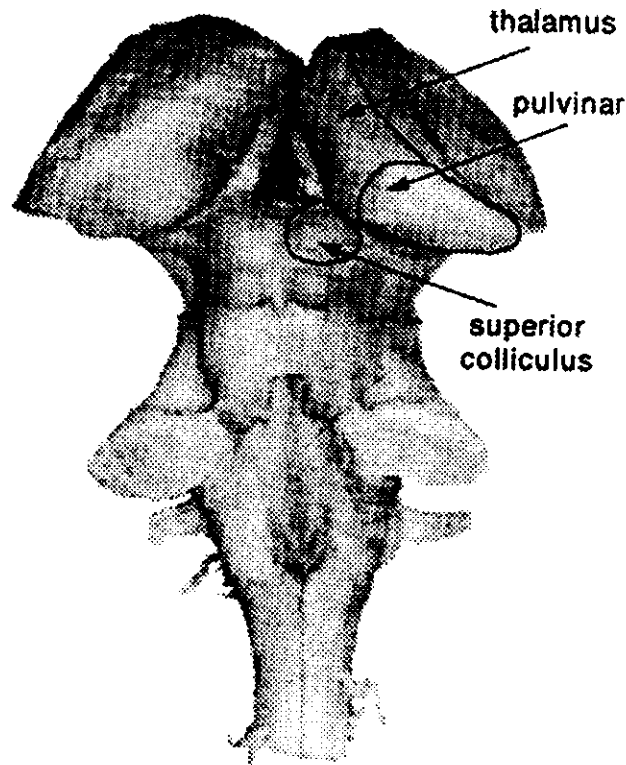


Figure 13.8: Attention-related areas in the brain stem

Dorsal view of the brain stem showing the the major subcortical areas involved in the control of visual attention. Labeled the thalamus, the pulvinar and the superior colliculus.

If we postulate the existence of Focus of Attention Master (FAM) type circuits in the brain functionally similar to that in DETE, then it is possible that the FA-master for the frontal lobe is located in the Nucleus Medialis Dorsalis (NMD) of the thalamus -- the thalamic nucleus which is most distinctly and directly connected to it (Fuster, 1985). The FA-master for the posterior (sensory) lobe might be distributed in the other sensory specific nuclei of the thalamus. For instance, the visual-FA-master is perhaps in the Lateral Geniculate Body (LGB) (Crick, 1984), which receives fibers from the optic tract and projects via the optic radiation to the calcarine cortex (V1 -- the primary visual area). The neurons in the LGB are retinotopically organized. Some internal oscillatory mechanism in the LGB/PGN/V1 can serve as a phase filter for the "clock" located in these structures which allows only the firing of neurons which are in phase with this clock to reach the cortex.

(c) The pulvinar: -- Another possible location of the centralized clocking mechanism for control of the attentional window is the pulvinar (Rafal and Posner, 1987; Petersen et al., 1987) or the claustrum. The pulvinar has reciprocal connections to all visual cortical areas. This connectivity pattern is exactly what was postulated for the Focus of Attention Master in DETE.

Since DETE's visual world is composed of simple objects (e.g., no shades of gray or various color mixtures, no recurring patterns, etc.), DETE does not have mechanisms for driving attention on the basis of Gestalt grouping principles such as *proximity* or *similarity* (Koch and Ullman, 1985). Such principles are evidently very important in human attentive processing. Also, since

DETE is exposed only to simple-shaped objects which are perceived as a whole rather than a collection of parts or regions of various textures, colors, and shades, the modeling has not focussed on how the visual system integrates multiple shifts of attention in order to achieve a coherent perception of more complex objects (Baron, 1987).

So far, our emphasis has been on visual attention. However, attention is also a property of other sensory systems. The auditory system is the other relevant sensory system within DETE's framework that exhibits attentive properties. However, while it is known that humans can handle, for instance, the cocktail party effect (see von der Malsburg for possible neural mechanisms) (von der Malsburg and Schneider, 1986), this is out of the scope of DETE's "auditory" system, and therefore will not be discussed in detail. I will only point out that the Auditory-FA-master is possibly located in the Medial Geniculate Body (MGB) of the thalamus which receives fibers from the Inferior Colliculus (IC) and projects via the geniculotemporal radiation to the transverse temporal gyrus of Heschl (the primary auditory cortex -- Brodmann's area 41). The response properties of the MGB neurons are tonotopically organized (high frequencies are represented medially and low frequencies laterally).

13.5 Neural plausibility of the KATAMIC model

As was mentioned before, the KATAMIC model is the basis of all memory modules in DETE's architecture. While it has been used as a general and special purpose-sequential memory in DETE, a separate research line was developed in which an attempt was made to demonstrate that the KATAMIC model can be regarded as a physiologically plausible model of the cerebellar cortex. It is important to stress that no claim and not even a suggestion is made here that the cerebellum is crucial for language processing in the brain. However, it is plausible that a sequential associative memory mechanism functionally similar to the KATAMIC memory could be found in the archicortex (e.g., the hippocampus) as well as the neocortex.

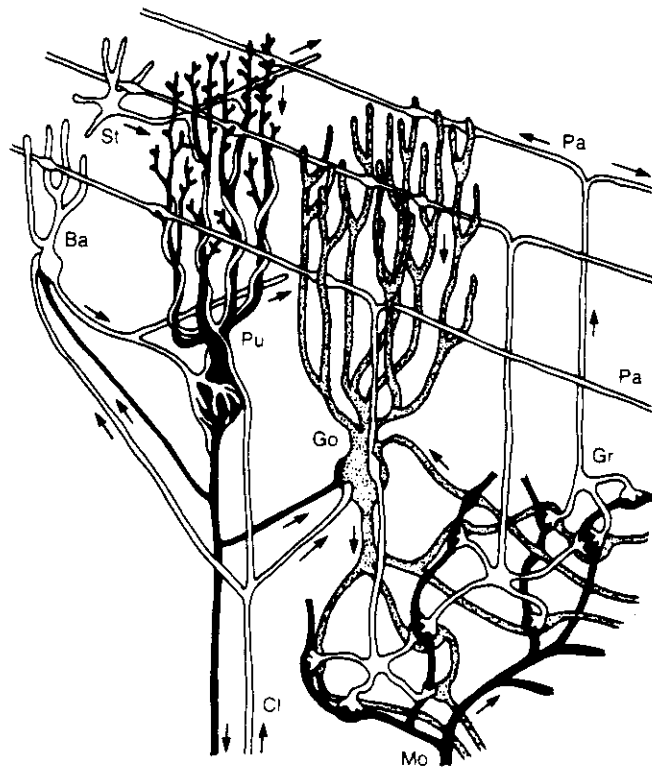
In the following sections, a mapping between the KATAMIC model and the cerebellar cortex (CC) is presented at the levels of cerebellar cortical cytoarchitecture and cellular neurochemistry. Supporting experimental evidence is provided. Since detailed descriptions of cerebellar cytoarchitecture can be found elsewhere, here only the aspects critical for the KATAMIC model are pointed out (Figure 13.9). From the 5 major cellular types in the CC: Purkinje cells (PC), Golgi cells (GoC), Granule cells (GrC), Basket cells (BC), and superficial Stellate cells (SC) the focus in the present model is only on the first three. The model incorporates several specific cytoarchitectural features:

(1) Each GrC axon has two functionally distinct components: (a) ascending fiber (AF) which makes multiple excitatory synapses with the Purkinje dendritic tree (Llinas, 1982; Bower and Woolston, 1983) before it bifurcates and turns into (b) parallel fiber (PF) which passes through the dendritic trees of 200-450 PCs.

(2) While most drawings of the cerebellar circuitry suggest that the GoCs are sampling the PFs, in the present model these synaptic connections are not considered and instead we incorporate the fact that many of the deep GoCs which have dense arborization in the granule layer (Eccles et al., 1967) are actually sampling mossy fibers via *en marron* synapses (observed at least in the rat brain) (Chan-Palay and Palay, 1971).

(3) In the KATAMIC model, the BSSs (GrCs) get direct projections from the prediction neurons (PCs), whereas, in the cerebellar circuitry, PCs give recurrent collaterals to the GoCs (Hamori and Szentagothai, 1966) instead of to the GrCs. However, these collaterals make synapses at the cell bodies (axo-somatic) of the GoCs and I hypothesize that functionally the two circuits (the KATAMIC, on the one hand, and cerebellar on the other hand) are equivalent. This assumption is based on the fact that the two negative synapses within the cerebellar circuit -- from PC axon to GoC soma, and from GoC to the glomerulus (a complex synapse containing mossy fiber, GoC axon, and GrC dendrite) yield a positive copy of the PC output to the GrC.

(4) Another crucial cerebellar feature concerns the climbing fibers (CF). They originate in the inferior olivary nucleus and pair off one-for-one with the PCs. The olivary projections reach the CC in synchronously firing patches (zones of Oskarsson) (Ito, 1979) (see Figure 13.10). I hypothesize that each zone is processing a different sequence and the size of a zone maps to the pattern width. An action potential (AP) coming along a CF is guaranteed to trigger a complex spike in the corresponding PC.



- | | |
|----------------------------|--------------------------|
| Pu = Purkinje cell (black) | St = stellate cell |
| Go = Golgi cell (dotted) | Ba = basket cell |
| Gr = granule cell | Cl = climbing fiber |
| Pa = parallel fiber | Mo = mossy fiber (black) |

Figure 13.9: Neural circuitry of the cerebellum

A simplified drawing of the basic cerebellar circuit. (Reproduced with permission from Llinas, 1975. Copyright © 1975 by Scientific American, Inc. All rights reserved).

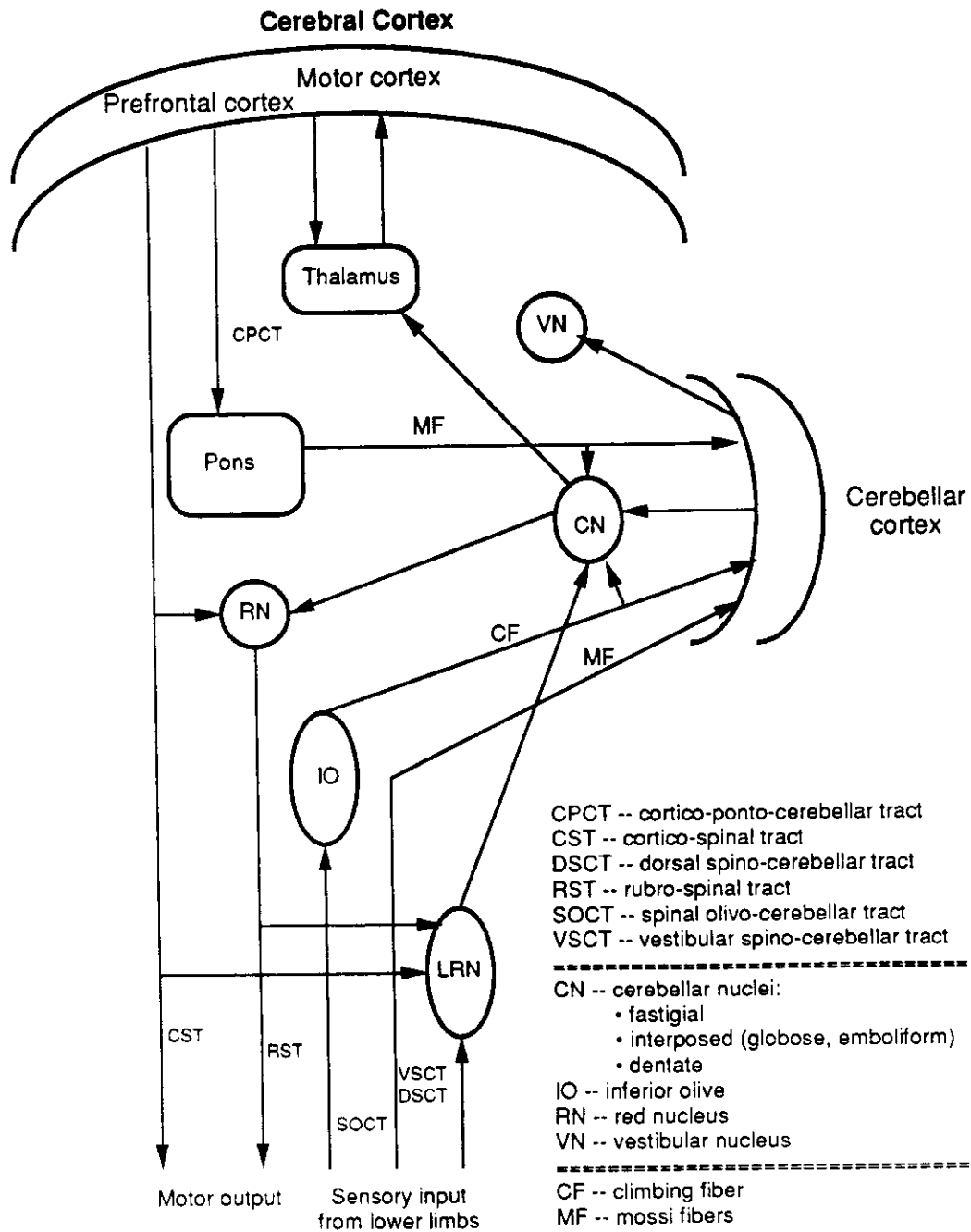


Figure 13.10: Cerebellar connections (I/O) in the brain

The set of nuclei and the positive feed-back loops involved in the cerebellar control of voluntary movements.

Two basic assumptions are made in the KATAMIC model of the cerebellar cortex. First, it is postulated that at different stages of processing in the cerebellum, the information code (representation) is different. While I agree with the widely accepted belief that the important factor in the CC control of the intra-cerebellar nuclei is the average firing frequency of the PCs (i.e. a "unit of information code" is a time segment during which a particular PC maintains a stable frequency (Eccles, 1977)), I hypothesize that within the CC of significance are the spatial-temporal relations between the APs arriving along MFs. The PCs serve as code transformation devices effectively time-multiplexing the predictions (made by their primary dendritic branches) of the incoming spike trains along the ascending parts of the GrC's axons. Second, while the relay stations involved in the actual transformation of a motor program into motor action are represented by a multitude of brain-stem nuclei interconnected by relatively long (Houk, 1987) positive feedback loops (see Figure 13.10), the actual motor programs for automated motions are processed by a short loop located within the cerebellar cortex which involves the PCs, GoCs & GrCs, and these motor programs are represented as pattern sequences in a KATAMIC like sequential associative memory in the CC.

KATAMIC-to-cerebellum mapping -- the network level

The following paragraphs compare a set of different hypotheses about the functional significance of the CC neurons to the hypotheses made by the KATAMIC model.

Various hypotheses

KATAMIC hypotheses

Purkinje Cells

Control the firing of the intra-cerebellar nuclear cells which serve as comparators for the signals arriving from motor cortex to those from muscles and proprioceptors, and correct the motion (Eccles, 1973).

Function as **PREDICTRONS** -- neural elements capable of learning to generate predictions (APs) in a time-multiplexed form concerning the input which they will receive along ascending fibers (AFs).

Golgi Cells

(1) Maintain GrC (hence PFs') activity fixed at a relatively constant rate (i.e. automatic gain control) (Albus, 1971).

(2) Act as phase lag elements (e.g., leaky integrators with time constants of several seconds) (Fujita, 1982).

Serve as **RECOGNITRONS** - sequence recognition devices which generate a recognition signal when part of a familiar sequence is processed by them and the neighboring PCs. They control the state of the GrCs.

Granule Cells

Ensure that little correlation occurs between the activities of individual PFs (Sabah, 1971).

Input gating devices which pass either the external input (from MF) or the internal input (from PC collaterals) to the parallel fibers.

Climbing Fibers

Provide a teaching signal for the modification of the synaptic properties of the PF->PCs junctions (Brindley, 1964).

sequences cannot be recalled (Demer and Robinson, 1981).

Reset sequence generation (Boylls, 1975). Without them, previously learned

Granule Cells' axons (AFs & PFs)

Provide the context for learning and the synapses they make with PCs are modifiable (Marr, 1969).

Carry the external input or the prediction, both of which modulate the *stm* in the DCPs.

KATAMIC-to-cerebellum mapping -- the sub-cellular level

The mapping at this level is based on three hypotheses:

Hypothesis 1:

There is a short-term memory storage mechanism in the dendritic tree cytoplasm of PC represented by the intracellular $[Ca^{++}]$.

Facts: Inputs (i.e. action potentials -- APs) arriving along the PFs cause infusions of Ca^{++} ions at the synaptic junctions (Ito, 1987; Kano and Kato, 1987). Also, dendritic spike bursts (d.s.b.) are interposed between the somatic spikes and patches of PC dendritic membrane have active electrogenic properties (Llinas and Sugimori, 1980).

Interpretation: The spontaneous firing in the PC dendrites is generated along multiple sites of the dendritic tree and these sites are associated with the synapses made by the PF.

Fact: The time course of the dendritic action potentials (for PF-induced APs) is much longer than that of the soma generated Na^+ AP (Eccles, 1977).

Interpretation: The existence of such persistent dendritic APs fits with the KATAMIC model according to which the *stm* is not being reset (d.s.b. persists) until they die out or a CF input indicates the end of one sequence and the beginning of another.

Hypothesis 2:

There is a long-term memory storage mechanism in the membrane/cytoplasm of the PCs dendrites represented by the $[PKC]$.

Facts: During cerebellar learning, long-lasting changes in the number of membrane associated K^+ channels occur in the dendritic tree (Alkon, 1984; Jaeger et al., 1988). These changes are due to the movement of calcium-sensitive enzyme Protein Kinase C (PKC) from the cytoplasm to the dendritic membrane. The PKC movement is in response to changes in $[Ca^{++}]$ and another second messenger, diacylglycerol, that accompany the association of temporally related sensory stimuli (Alkon, 1989). Also, at the cell membrane the PKC causes a decrease in potassium conductance (gK^+) which reduces the potassium-ion flow and makes the cell more excitable. The critical factor for reduction of this flow is not the input stimuli themselves but their orderly temporal relation.

Hypothesis refinement: Cerebellar "learning" occurs not exclusively (or even not at all) at the parallel fiber synaptic sites, but is a feature of the PC's dendritic membranes and specifically of the membrane patches that contain K^+ channels whose concentration can be affected through Ca^{++} dependent transportation of PKC from the cytoplasm to the membrane.

Hypothesis 3:

The firing state of the PCs depends on the interaction of the *stm* & *ltm* in each dendritic compartment and in the cell as a whole.

Facts: K^+ conductances (*ltm*) together with the cytoplasmic Ca^{++} distribution (*stm*) are critical in the establishment of the PC activation level (i.e. membrane potential) which further determines the response state of the cell. The somatic activation (membrane potential) is effectively computed as the dot-product of *stm* and *ltm*.

Interpretation: Physiologically the **dot-product** can be viewed as a multiplication of the *ltm* (resistance) and the *stm* (current) at each DCP which yields a local membrane potential, followed by an addition of all these potentials (i.e. batteries connected in parallel) to produce the somatic potential.

13.6 Neuropsychology of memory

There are at least two possible levels of comparison between DETE's memory mechanisms and the memory systems in humans. (1) Taxonomy: DETE's memory taxonomy is discussed in Chapter 9. In general it is the same as the taxonomy of human memory as revealed by neuropsychological and neurobiological studies. (2) Circuitry: This is the level of the neural circuits and their dynamics which serve as memory modules. In the following sections I discuss this second level, making comparisons between DETE and human's memory mechanisms and their dynamics.

13.6.1 Short-term memory (STM)

The short-term memory (a.k.a. primary memory (James, 1890; Halgren, 1989), immediate memory (Squire, 1986), or iconic memory (Coltheart, 1983)) is the basis of our ability to repeat short sequences of items. For example, one can repeat several consecutive digits of a phone number or a sentence immediately after it has been read or heard. Its neural substrate is in general a short-lasting change in the neuronal membrane properties (most likely at the synapses). The STM is not impaired in amnesia.

Modalities of the STM

STM can be observed in all sensory modalities. Its characteristics, however, vary between modalities. These variations seem to be due to the dynamics of the inputs to each of the individual modalities. For instance, in the auditory modality, meaningful inputs usually come sporadically (e.g., single words or sentences) and the auditory STM is specialized in maintaining of such well-defined temporary chunks. In the visual modality, the input seems to enter the system continuously. However, there is also a degree of chunking which corresponds to saccades. The duration of a fixation between saccades is on average few hundred milliseconds.

Functional characteristics of the STM

The visual and verbal STM have the following characteristics: (1) One-shot storage (memorization) - a sentence presented only once can be repeated right away. A simple visual sequence (e.g., of an attended object) can also be mentally recalled right away. (2) Fast and almost complete reset of its content when a new input (sentence or image) is presented (i.e. when the focus of attention is shifted). (3) Possibility of rehearsal of content without the necessity of a *specific* external cue to trigger each iteration. A *non-specific* signal (e.g., external verbal request for repetition) can trigger

the recall of the memory trace. Once the rehearsal is interrupted the content is lost. (4) Limited capacity, recall performance deteriorates if the capacity is exceeded. (5) Fast decay if not refreshed through rehearsal. If rehearsed it can be kept in mind for few minutes. All of the above functional characteristics of the STM have been successfully incorporated in the Short Term Memory used in DETE.

One possibility for the anatomical localization of the STM is that it is an intrinsic capacity of each cortical processing system (Monsell, 1984). Thus, each unimodal sensory cortex, like the visual and the auditory cortices, has the ability to store short spatial-temporal sequences (i.e. serve as a small capacity memory buffer).

The "localization" of the STM in DETE in general corresponds to the physiological observations. Namely, the STM is interleaved with the LTM and is part of each of the visual feature memories.

13.6.2 Long term memory (LTM)

The long-term memory is a type of memory which is characterized by a much larger time span. It comes in two modalities: declarative and procedural.

Declarative Memory (DM)

The declarative memory is a memory for facts. There are two types of declarative memory, episodic memory (EM) and semantic memory (SM). These two categories are not completely separable and in fact can be regarded as the two poles of a continuum. This continuum is formed during our lifetime. Initially (while we are young) all experiences are stored as episodes, later the ones that are repeated over and over again form the SM and the ones that are more unique form the EM. The DM (and especially its episodic component) is profoundly affected in amnesia in the sense that new episodes cannot be stored (anterograde amnesia) and some episodes experienced prior to the onset of amnesia cannot be recalled (retrograde amnesia) (Milner and Teuber, 1968).

The general characteristics of the DM are: (1) It is open to introspection (i.e. accessible to conscious awareness). (2) It is composed by the traces of the sequences of events (their representations) that have been experienced (i.e. seen or heard) for instance, facts, episodes, lists, routes, maps, thoughts, words, and images of everyday life. (3) The storage is sequential (with experience). (4) Memories can be declared (i.e. brought to mind or instantiated) through all modalities (e.g., verbally in the form of hidden articulation or non-verbally in the form of imagination). (5) A cue, by virtue of its content, can start a retrieval process at any point of a stored sequence and if left to itself the memory will complete the sequence. (6) The efficiency of its encoding and retrieval can be enhanced voluntarily (e.g., through repetition, emotional state, drugs, etc.). (7) Undergoes consolidation and forgetting.

Considering these points one by one we can say: (1) While the content of DETE's DM is available to inspection it is not open to introspection. (2) The content of DETE's DM is also stored in traces of the sequences of events that DETE has experienced. (3) Storage is sequential. (4) DETE contains all necessary mechanisms for "declaration" of memories, i.e. bringing them up to the working memory in the form of patterns of activity. (5) The KATAMIC model (which is the basis of the DM) allows complete sequences to be recalled by a cue. (6) While the efficiency of encoding can be enhanced through repetition, the repetitions are not a result of a voluntary process, since DETE currently does not have internal motivation. (7) A process of consolidation and forgetting is

also typical for DE TE's memory. It is based on a mechanism for competition for *stm* resources in the dendritic trees of the predictrons that form the memories (see section 8.2.1 for details).

Episodic Memory (EM)

Episodic memory in humans is the ability to store and recall (re-experience) specific episodes (events) (Tulving, 1985). Examples of such episodes are for instance, the day I got my diploma, the birth of my baby, etc. A trace of each episode is stored together with the information of the specific place and time in which it was experienced. For the retrieval of an episode (bringing it to WM) one usually requires a conscious effort to elaborate specific cues (Kolodner, 1984). From the above definitions it is evident that unique episodes form the content of the episodic memory while multiple re-occurrences of similar episodes form the semantic memory. Since DE TE operates in a very impoverished sensory world it is important to distinguish what constitutes a unique episode for DE TE. In DE TE all experiences that do not recur are treated as episodes while all recurring experiences form the semantic memory.

The EM has all general characteristics of the DM and also some specific ones such as: (1) Content is formed by unusual or unique events in the personal history (autobiographic events). (2) Memories are stored one-shot as a result of unique experiences. (3) Retrieval requires time-consuming elaboration of cues. (4) Cross-talk between traces rarely occurs. There is little cross-talk between traces since they represent unique events and generally different events have different representations. (5) Emotional states affects the strength of the created traces.

To the extent that each unique experience that DE TE has had (i.e. one that has not been repeated many times) is an episode by definition and since such episodes leave traces in DE TE's long term memory, DE TE has an episodic memory. However, DE TE lacks a major component of the episodic memory system -- the ability of this memory to be influenced by the emotional state. Humans do not remember equally well all unique episodes that they have ever experienced. In general, episodes that have a stronger emotional context are remembered better. To match DE TE's behavior better to that of humans we need to incorporate emotional and motivational modules.

Semantic Memory (SM)

Semantic memory is the ability to extract the commonalties in and store traces of multiple similar episodes or repetitions of one and the same episode. Such examples in DE TE are, for instance, a ball bounces when it hits a wall, or a balloon explodes when poked with a pin. Since the things that change in all cases of repetition are the times and locations where the events occur, the temporal and location information is effectively lost (smeared in time).

Like the EM, the SM has all general characteristics of the DM and also some specific ones such as: (1) Content is formed by familiar items which have been experienced over and over again during the life-time (i.e. they have formed categories). For instance: words, scripts, stable facts, familiar faces, etc. (2) Memory traces are stored through multiple repetitions. Each individual repetition makes the particular trace stronger. Ultimately the traces left are more permanent and strong. (3) Retrieval is done when a partial cue is presented (response-sequence retrieval depends on the cue and the memory content). Usually the context that is recalled together with the trace is either a) the most recent context in which this concept was seen, or b) the most frequent context (combination of contexts), or c) an externally provided context. (4) Cross-talk between traces: similar sequences create similar traces and different sequences create different traces. This ability is natural and desirable for the establishment of semantic memory.

The lexicon is one type of semantic memory. It is a repository of both grammatical and commonsense knowledge indexed by lexical items (Nakhimovsky, 1988). The grammatical knowledge of DETE which is acquired with experience involves knowledge about word order and syntactic relations (e.g., gender agreement in SPANLAN). It is extracted from the verbal input and is encoded as the statistically most probable associations. The commonsense knowledge aspect of the lexicon is represented in the association of the verbal tokens with their visual representations. It is commonsense in the sense that these representations reflect the constraints which exist in the physical world. The network representation of each word itself forms the lexical item index.

In DETE the LEXICON is formed by the representations of the individual words together with the conceptual structures which they were associated with. The memory trace of each word is associated with information about all contexts (visual and verbal) in which it was encountered. This representation evolves continuously with DETE's exposure to new experiences. In other words, DETE extracts the meaning of each word from its syntactic (verbal) and semantic (visual) use and stores it as a distributed memory trace.

The Declarative Memory system resides partly in the hippocampal formation (HCF) and partly in the neocortex. One theory is that a short-living memory trace is established in the HCF which is later consolidated (transferred) in the neocortex (Halgren, 1984). Other theories, however, suggest that the HCF is not involved specifically in the storage of declarative memory traces but rather in the formation of configural associations (Sutherland and Rudy, 1989).

In DETE, the declarative memory is distributed among the verbal and visual long term memory modules. DETE does not possess modules which resemble functionally the HCF.

The Procedural Memory (PM)

The procedural memory (PM) is a type of LTM which is not affected by amnesia. The PM is responsible for such highly automated behaviors as reading, typing, swimming, diving, etc. (Tulving, 1985). It has the following characteristics: (1) It is unconscious (i.e. accessible only through performance). (2) It is modality specific. (3) It contains skills (motor -- e.g., diving; perceptual -- e.g., reading; cognitive -- mental arithmetic; and their combinations). (4) Encoding and retrieval can be enhanced voluntarily (e.g., through repetition, emotional state or drugs). (5) Exhibits priming effects (Warrington, 1970; Graf et al., 1984). (6) It is amenable to consolidation and forgetting (competition effects).

Humans have the ability to remember the rhythm of a song without remembering the actual words. Actually they can associate (learn) different words with one and the same rhythm. We can also remember the rhythm of the words coming on a busy phone line even when we cannot catch the actual words due to the noise. Knowing the context of the conversation we can use this rhythm to reconstruct the possible content of the conversation. To explain this ability I postulate the existence of an unconscious memory mechanism. This mechanism functions as a kind of an Order Memory which effectively counts the segments in the input stream and measures their duration. Such a mechanism can be later used as the basis of our ability to learn to count successive events. This can be simply done by associating verbal labels (e.g., **W**:one, **W**:two, etc.) to the individual orders in the memory.

The PM is used in DETE to control the location and size of the visual focus of attention to the external world; to learn motion trajectories of the finger, and to do segmentation of the visual and verbal input. In other words it is used as a motor memory.

The extrapyramidal motor system seems to be the anatomical substrate subserving the Procedural Memory (Mishkin et al., 1984).

13.6.3 The working memory

The working memory (WM) (Weiskrantz, 1982; Phillips, 1983) is in general the collection of activity patterns in all memory modules at a given moment of time. These patterns change with time as a result of external and internal influences. The patterns that form the working memory can be generally classified as: (1) *Current attention* -- patterns that are in the focus of attention, and (2) *Current context* -- patterns that are out of the focus of attention. Brain areas that have been implicated in the localization of the WM (also known as mind or consciousness) are the neocortex, paleocortex also some midbrain structures such as the thalamus, basal ganglia and the claustrum. The hippocampal formation (HCF) does not seem to be directly involved (but certainly contributes) since complete bilateral damage of the HCF leaves most aspects of consciousness intact (Damasio et al., 1985). The WM is impaired in amnesia (Weiskrantz, 1982; Baddeley, 1982; Baddeley, 1986).

Anatomical localization of the Current Context

The current (mental) context is stored in the frontal cortex (FC) (specifically in the prefrontal cortex -- the association area of the frontal lobe) but is instantiated in the HCF. The contribution of the frontal cortex to contextual memory can be inferred from clinical observations of patients with damage to the FC. In such cases experimentalists have observed: source amnesia, (Halgren, 1989), confabulation, defective recency judgements (i.e. incapability to inhibit the interference of an old habit with respect to current behavior (Mishkin, 1964)), defective veracity judgements, and failure to release from proactive interference.

One electrophysiological manifestation of the mental context is the Contingent Negative Variation (CNV), also called the *expectancy wave*. CNV is a slow electrical potential which can be recorded from the frontal scalp region of subjects engaged in expectation of a future event (Walter et al., 1964).

Anatomical localization of the temporal representations

It is plausible that the FC contains only the representations of time (Fuster, 1985; Milner, 1982; Milner et al., 1985) and allows for planning. An interesting question is, what happens to a person without the FC? Does he/she live in an eternal NOW and cannot understand propositions concerning time?

14 CURRENT STATUS, FUTURE WORK AND CONCLUSIONS

14.1 Current status

DETE was conceived and developed as a test-bed for neurally-based computational modeling of language acquisition through associations of visual and verbal experiences. This thesis reports on the completion of the first phase of the project. This phase included the following stages: (1) All individual components of DETE's architecture were implemented and tested separately, and special attention was paid to the performance of the KATAMIC memory which is the heart of DETE. (2) The individual components were assembled in a complete functional architecture. (3) A series of basic experiments with increasing level of complexity (described in Chapter 11) were performed. In the beginning of each experiment DETE was either in its naive state or it has already learned the prerequisites for the particular experiment. However, due to limited availability of computational time on the CM-2 Connection Machine and for the purpose of speeding up of the developmental process, most of the experiments were performed using "canned" representations of the visual input. (4) A mature (educated) DETE -- one that was trained on all experimental situations to the point that it can perform any of the learned behaviors on demand, was not tested since the computational resources available to us at present (a 16K processors CM-2) were not sufficient for this task. As a result, a number of interesting questions such as what are the possible interferences between the different acquired abilities, has not yet been looked at.

14.2 Future work

By its nature, the DETE project allows extensions in a number of directions. At present, however, we are interested in extending the work on DETE in only few directions which we consider of significant interest.

14.2.1 Neurally realistic modules and connectivity

As described in Chapter 13, DETE incorporates only some general organizational and functional principles which are characteristic to the neural circuits in the brain that subserve the tasks at hand. This is especially true for the peripheral modules in DETE which were implemented procedurally. For this reason, I have no strong commitment to the nuts and bolts of the system. In fact, from a perspective of neural realism many of them are wrong.

In the current implementation of DETE only the memory modules are neural networks and a specific attempt was made to map the KATAMIC architecture to the neural architecture and function of the cerebellum (see section 13.5). However, the rest of DETE's modules were implemented procedurally. In our future work we intend to substitute the procedural modules with neural networks based on neural architectures of the corresponding functional systems in the brain.

Similarly to the way in which lesion studies of language-related areas provide important data about their functional significance and the significance of their interconnections, DETE's language abilities can also be examined by introducing artificial lesions of individual modules or their

connections. Such experiments can throw light on a number of questions: (1) How good is the correspondence between the model and the brain? This can be assessed on the basis of similarity or dissimilarity of behavioral abnormalities when corresponding areas or connections are lesioned. (2) Lesions in the model which lead to specific functional disorders can have predictive value for the explanation of similar behavioral disorders in human language.

14.2.2 Additional language capacity

Currently, as demonstrated by the experiments described in Chapter 11, DETE's linguistic abilities are very limited. For instance, DETE's lexicon is very small. Also, we have not demonstrated convincingly that DETE can learn various languages with their grammatical idiosyncrasies. DETE is not capable yet to make verbal-to-verbal associations. It cannot handle prosodic inflections of the language, etc. In the remaining part of this section we focus on some of these limitations and outline future experiments. When necessary we suggest modifications and additions to DETE's architecture.

(1) *Expanding the lexicon:* The design of the visual feature planes used in DETE allow it to represent the meaning of a larger number of individual words than demonstrated in Chapter 11. For instance, DETE has a unique visual representation for non-linear and/or accelerated motions and therefore it can potentially learn the meanings of words such as "turns" or "speeds-up". However, a number of word categories cannot be currently handled by DETE. For instance: (a) Words concerning structured objects and their motion -- e.g., "cat" (structured object), "walks" (involves coordinated motion of pieces of structured object). To be able to handle such words DETE needs a more sophisticated visual system which is capable of processing of part/whole relations. (b) Words concerning abstractions -- e.g., "power", "organization", "process", "responsibility". (c) Words involving emotions, goals, plans, mental states -- e.g., "sad", "wants", "decides", "sleepy".

(2) *Learning multiple languages/grammars:* All natural languages have a common inherent feature -- they possess a syntactic structure. The syntactic structure of a language can be partially expressed as a set of recursive rules about the acceptable ordering of words in sentences, about word classes, and about word relations (i.e. constituent structure). Spanish, for example, inverts the common English noun phrase order ADJ NOUN to NOUN ADJ and sometimes uses suffixes in place of explicit pronouns (e.g., "andaron" vs "they walked").

The development/acquisition of each grammatical system can be viewed from two different perspectives: 1) Ontogenetic: In other words, how did different grammatical rules emerge in the process of evolution of a particular language, and are these rules (or their essence) universal across languages? These questions (related to the ontogenesis of language) were not in the focus of this thesis. 2) Phylogenetic: How do we acquire grammatical skills being exposed to language inputs which are already grammatically structured? Is there some innate universal grammar or at least some predisposition for grammatical order in our minds? It is evident that people are able to acquire languages that have very different grammars and this means that we as human language learners create unconsciously in our minds (during the early years of our development) functional representations of the grammatical constraints in a particular language. This second perspective on language acquisition is the focus of this thesis. We attempted to examine the neural mechanisms that are involved in the process of "shaping" of our minds for language understanding and production.

While the experiments described in this thesis demonstrate explicitly that DETE is capable of learning only small subsets of different languages (e.g. a number of lexical items and simple

syntactic structures), we believe that the current architecture can support more robust language capabilities. In its current version DETE has been tested on examples from two artificial languages (FIRLAN and SECLAN) which were both subsets of English (see section 2.4.2). For our future work we believe that instead of these two languages we should use another set of artificial languages -- ENGLAN, SPANLAN and JAPNLAN (subsets of English, Spanish and Japanese). These languages will have separate lexical items, e.g., “pelota” or “tama” (Japanese for “ball”) instead of “ball” and “mover” or “ugoku” instead of “move”. In some cases the words are totally different, (e.g., “pelota” vs “ball”), but in other cases, the words are close (e.g., “move” vs “mover”). In our future experiments with DETE we envision two different scenarios: (a) Learning the three languages separately. In this case we can compare the rates of acquisition for corresponding words or grammatical structures. (b) Learning two (or three) languages simultaneously or one after another. In this case we can study the types of interferences between languages. We can test also whether it takes longer for DETE to learn SPANLAN after ENGLAN has been learned than if SPANLAN is the first language? Concretely, does the ADJ NOUN (in English) vs NOUN ADJ (in Spanish) word order difference impair DETE in some way when learning SPANLAN? (For instance, it is known that English speakers have trouble with such thing when learning Spanish, due to, ostensibly, interference from English forms).

Some interesting features of English (vs Spanish and Japanese), limited, as much as possible, to the blobs task/domain which can be used for testing DETE are listed below. (We use words like “book” in the Japanese, but they can later be replaced with Blobs’ World words, e.g. “ball” = tama, “moves (intrans)” = ugokimasu)

1. Demonstrative adjectives vs demonstrative pronouns

- English: the same words are used:

Look at this (that) ball. vs Look at this (that).

- Spanish: there is a single demonstrative pronoun:

Mira esta pelota. (Look at this ball.)

Mira este cuadro. (Look at this square.)

vs.

Mira esto. (Look at this.)

(one can say “Mira esta” but it really is an ellision, as in “Mira esta (pelota)”)

Empuja esa pelota. (Push that ball.)

Empuja esa. (Push that (one).) (ellision)

- Japanese: one MUST use different words for demonstrative adjectives vs demonstrative pronouns:

adjectives:

kono isu (this chair) vs

sono isu (that chair) vs

pronouns:

kore (this)

sore (that)

e.g. Kono hako mite kudasai. (This box, look at it please.)

Kore wa, mite kudasai. (Look at this please.)

2. Verbs “to be”

- English: same verb is used for:

- equals (The ball is a ball)
- property assignment (The ball is broken)
- existence (There is a ball in the corner)

- Spanish:

“ser” is used for equivalence and fundamental properties

“esta” is used for non-essential properties

La pelota es una pelota. (The ball is a ball)

La pelota esta rota. (Ball is broken <temporarily>)

“hacer” is used for existence

Hay una pelota abajo. (There is (exists) a ball below.)

Habia una pelota arriba. (There was a ball above.)

- Japanese: breaks up “to be” in a different way:

“desu” is used for equivalence

“arimasu” for existence (and properties of) inanimate objects

“imasu” for existence (and properties of) animate objects

e.g. watashi-wa Nenov desu (I am Nenov)

koko-ni-wa shinbum arimasu (here-at-topic magazine is) i.e. here is a magazine

soko-ni-wa Dyer-san imasu (There is Mr. Dyer.)

To test DETE on a subset of the Japanese “to be”, one could say, always use “arimasu” for rounded objects and “imasu” for objects with corners. This would not be “correct” but could show how DETE could learn to associate correct restrictions with use of these verbs. Since in the blobs world there are no “animate” objects, to test the ability of DETE to make such a distinction is currently impossible. (DETE can’t even recognize an animate figure, which would be complex). However, one CAN test DETE’s ability to associate different verbs with different visual objects, based on some OTHER features (i.e. other than animate/inanimate distinction). For instance, in JAPNLAN DETE could be trained to distinguish arimasu (i.e. with inanimates) with, say, large objects and imasu (i.e. with animates) with small objects.

e.g. soko ni chisai tama-ga arimasu (there-at small ball-subj is) There is a small ball.

soko ni ookii tama-ga imasu (there is a large ball)

To test GENERALIZATION, DETE could be trained on small/large circles and triangles, and then be tested on small/large squares. If DETE has associated “imasu” with large in the size map, then the generalization should work for any shape.

3. Spatial relations

- English and Spanish: use prepositions:

The book is on the table.

El libro está encima de la mesa.

- Japanese: does NOT allow this construction. One must say something more like, “The table’s top at, the book is.”

e.g. tsukue-no ue-ni hon-ga arimasu (desk-’s top-at book-<subj> is).

i.e. the book is on the top of the desk

4. Case indication

- English and Spanish: indicate cases by word order:

Juan toma leche.

John drinks milk.

- Japanese: uses particles in post-fix position:

John-san-wa mizu-o nomimasu. (John-Mr.-topic water-obj drinks.)

5. Number agreement

- English & Spanish: indicate number by a postfix “s” inflection and matching verb inflection:

The ball moves. vs The balls move.

La pelota mueve. vs Las pelotas mueven.

- Japanese: does not directly indicate number, nor does the verb get inflected due to number:

... kono hana-wa sakura-desu (this/these flower(s)-topic cherry blossom(s) is/are)

6. Negation

- English and Spanish: use a marker to negate positive sentences:

The ball is broken. vs The ball is not broken.

La pelota está rota. vs La pelota no está rota.

- Japanese inflects each verb tense differently to negate it:

watashi-wa mizu-o nomisu (I water drink.)

watashi-wa mizu-o nomasen (I water not-drink.)

watashi-wa mizu-o nomashita (I water drank.)

watashi-wa mizu-o nomasen deshita (I water drank not.)

Since one of the first words that children learn is “no”, an interesting research question is what would be necessary for DETE to be able to learn negations. For instance, as we demonstrated in section 11.1.2 DETE can learn the words “stands” and “moves”. However, another way of expressing the meaning of “stands” is to say “does not move”. Can DETE learn the meaning of “not” -- i.e. generalize “not” to other verbs (or nouns) it hears? A major problem is that unlike “stands” the phrase “not moves” contains the word “moves” which will cause DETE to imagine movement. It is possible that something similar occurs in people. For instance, if we one says “John is not hitting Mary” we first imagine John hitting her and then in some way mark it as not so.

One possible experiment would be to (1) teach DETE the word “moves” on moving ball, (2) teach it “not moves” on a stationary ball, (3) test it on other objects. Another experiment would be to (1) teach DETE the phrases “not red”, “not green”, “not blue”, “not large”, and (2) test it on “not white” and “not small”.

7. Causality & desire

- English and Spanish: indicate causality/desire by clausal/phrasal constructions:

The ball hit the square and caused it move. (Made it move.)

- English: uses infinitival construction for desire:

I want the ball to move.

- Spanish: uses relative clause + subjunctive form of verb:

La pelota choquó con el quadro y lo hizo mover.

Quiero que mueva la pelota. (subj of mover)

- Japanese: indicates both causality and desire via inflections on the verb:

e.g. nomu (to drink), nomimasu (drinks),

nomaseru (cause to drink),

nomitai (want to drink)

e.g. mizu-o nomitai (I want to drink water)

John-san ni ano mizu-o nomasaremashita ((I) John-by that water was-caused-to-drink)

John made me drink that water

In the blobs world, this could be tested by sentences like:

“The ball hit the wall and that made it go south.”

“The ball hit the square and made the square move.”

To test a limited meaning of the word “want” in the blobs world would require making up some distinction in a dialect for this task/domain. For instance, we can interpret “Want red ball” as DETE putting the red ball in the upper right corner. However, to acquire the real meaning of this word DETE needs additional modules which represent internal states such as desire.

8. Subject

- English: (usually) must mention the subject of an active verb:

I drink water.

- Spanish and Japanese: drop the subject if it can be inferred from context:

Tomo agua. ((I) drink water)

mizu-o nomimasu ((I) water-obj drink)

9. Topic indicators

- English & Spanish: have direct/indirect articles to indicate first (vs subsequent) mention of an object in the discourse:

I see A ball. (Veo UNA pelota)

Where is THE ball? (Donde está LA pelota¿)

- Japanese: does not have this distinction. Instead, particles (-wa, -ga) are used:

“-wa” is a topic marker, about something the speakers both know about.

6:30-ni-wa mizu-o nomimasu (at 6:30 topic, water (I) drink)

“-wa” is used in certain negations:

asoko-ni-WA hon-wa arimasen (over there-at-topic book-topic is-not)

There is not a book over there.

“-ga” is used to introduce new things, as in statements of existence.

isu-ga arimasu (There is (exists) a chair.)

Figuring out when to use “-ga” vs “-wa” seems as impossible for non-Japanese as is the correct use of “the” and “a” for non-native English speakers. Currently DETE cannot track conversational topics and cannot engage in real dialogs. To do that it needs to be able to maintain a separate conversational context in memory. Also, in addition to its current representational capacity it needs some way of representing its partner in the dialog, i.e. a sufficiently complex model of a human or another robot. Such model is necessary so that DETE can predict, anticipate and elicit specific responses from its partner which would be impossible unless DETE knows what this dialog partner is capable of.

10. Speaker-hearer distinctions

Example of such type of distinction can be found in statements about distance of objects wrt both speaker and hearer:

- English: “this” and “that” indicate distance of object from speaker
- Spanish: has 3: esta, esa, aquel (this, that, that (over there))
- Japanese: (likewise) has kono, sono, ano, where:

“kono” is near speaker

“sono” is far from speaker but near hearer

“ano” is far from both speaker and hearer

To add this to DETE would require for the system to track and maintain topics of conversation. Also, DETE would need to be able to “see” an object in visual field that represents the teacher or other person. This would greatly expand the conversations between DETE and the teacher(s). For example, we can have a special 2-D shape in the visual field that represents NENOV and another which represents DYER. Then we can teach DETE:

-- point to shape, say “this is ME”

-- point to other shape, say “that is DYER”

One possibility for achieving such functionality is by encoding of VOICE TONE. However, it could be also done by setting a master “who’s talking” switch.

- English, Spanish and Japanese: have indexicals, such as

| | | |
|--------------|----|------------|
| “I” | vs | “you” |
| “yo” | vs | “tu” |
| “watashi-wa” | vs | “anata-wa” |

For instance I could say to DETE: “I am to the right of DYER.” and DYER could say: “I am to the left of NENOV”. From such verbal statements DETE would have to learn that “I” is used by the speaker to refer to WHOEVER is the one who happens to be speaking, and “you” is whoever happens to be listening. As discussed above, to be able to learn the meanings of “I”, “you”, “they” etc. DETE needs to have a model of itself as well as a model of the partner(s) in the dialog.

11. Modifiers

- English: puts adjective before noun, with referred order of <number> <size> <attribute> <color> NOUN:

The 3 big expensive red balls (vs The red 3 expensive big balls)

- Spanish: puts adjectives after noun (usually), with number before:

Los tres pelotas rojas y caras

(the 3 balls red and expensive)

- Japanese: puts “real” adjectives in front of NOUN (like English)

chisai hon (small book)

shiroi kami (white paper)

Japanese has na-adjectives. They normally function like nouns but can modify other nouns (like noun-grouping in English). There is no way (except for memorizing) to tell if an adjective is the -na type and therefore requires a -na inbetween. (DETE should be able to memorize this from experience.)

kirei-na tatemono (pretty-na building)

As demonstrated in Chapter 11 DETE is capable of learning the meaning of some modifiers as well as the order in which various classes of modifiers are arranged when constructing noun phrases. While the number of examples on which DETE was tested was very small, we believe that DETE’s current memory modules (and specifically the Morphologic/Syntactic Procedural Module) are sufficient for successfully learning of a large number of syntactic forms containing various modifiers in a number of languages.

12. Transitive, intransitive & reflexive verbs

- English: has verbs that can act as either transitive or intransitive:

John went away.

John went home.

- Japanese: verbs must be one or the other:

intransitive:

John-san uchi-e ikimasu

(need particle “-e” for destination)

(John-Mr. house-to goes)

- Spanish: has reflexive:

Juan se fue. (John went himself)

Juan se lavo las manos. (John himself washed the (his) hands.)

13. Possessive

- English: has two forms:

(a) John's book's pages.

(b) The pages of the book of John.

- Spanish: has only form (b)

(b) Las paginas del libro de Juan.

- Japanese: has only form (a)

John-san-no hon-no kami

14. Conjuncts

- English: uses "and" for conjuncts

John and Mary went home.

- Spanish: uses "y" in much the same way

Juan y Maria se fueron a casa.

- Japanese: has several different conjuncts for "and"

John-san-TO Mary-san-wa uchi-e ikimashita

(John-Mr. AND Mary-Miss-topic house-to went)

the particle -to is not used for clauses, or implied clauses)

(e.g. "John is an american and a student" canNOT use -to

i.e. John-san-wa america-jin-to gakusei desu (NOT allowed)

instead must do:

John-san-wa america-jin de gakusei desu

("de" is a shortened form of "desu")

other conjuncts are -mo and -ya

A-mo B-mo verb = Both A and B verb

A-ya B-ya verb = A and B (from among others).

15. Spatial/temporal relations

Particles/prepositions in English, Spanish and Japanese vary widely.

e.g. English: We think OF Mary.

Spanish: Pensamos EN Maria. (i.e. We think IN Maria.)

e.g. Japanese: uses -ni for time and -de for place
particle -ni indicates a STATE at a location

6:30-ni = "at 6:30"

particle -de indicates an ACTION at a location

uchi-de = "at home"

e.g., Spanish: a las 6:30 (at the 6:30)

Location

particle -e indicates location as a DESTINATION

e.g. uchi-NI hon-ga arimasu (the book is at home)

(home-at book-subj is)

uchi-DE mizu- nomimasu ((I) drink water at home)

uchi-E ikimashita ((I) went home)

(home-to going)

This could be tested in Blobs World by sentences such as:

The ball is at the north. (in Japanese, use -ni)

The ball is moving in the north (area). (in Japanese, use -de)

The ball is moving north. (in Japanese, use -e)

16. Expressions of power/authority/politeness/status

- English: has one "you"
- Japanese: has polite inflections on verbs. The Japanese "you" (anata) is only good among friends who are equals (it has no other "you" form).
- Spanish: as an informal and formal "you" (i.e. tu and Usted)

17. Embedded grammatical constructs

Languages have the ability to embed grammatical constructions and (at some point) children have the ability to do this embedding ad nauseum. This is basically an ability to generalize, not at the category level, but at the level of recursive structure. For instance:

I want to go home.

I want to ask John to go home.

I want John to ask Mary to go home.

I want John to want to ask Mary to want to ask Fred to go home.

I want ...

or

The small ball.

The small small small small ball.

or

The man who knows the boy is named Fred.

The man who knows the boy who shot the deer is named Fred.

• Japanese: appears to NOT have anything directly equivalent to either relative clauses or infinitival forms. It uses a kind of adjectival clause, for instance, the Japanese for “The man who shot the deer went home.” would be something like: “The deer-shooting man went home.” Currently DETE is not capable of dealing with such embedded structures..

18. Learning verb tenses of other verbs

Besides the verb “hit” (the learning of its tenses was described in section 11.7), we can use a number of other verbs in the blobs world and as part of our future work we intend to test DETE on learning such verb tenses. Some examples of such verbs are given in Table 14.1.

| Feature | Past Tense | Present Tense | Future Tense |
|-----------|---------------|---------------|-------------------|
| location | moved | moves | will move |
| size | shrunk | shrinks | will shrink |
| velocity | stopped | stops | will stop |
| shape | transformed | transforms | will transform |
| color | changed color | changes color | will change color |
| existence | disappeared | disappears | will disappear |

Table 14.1: Verb tenses of additional verbs

19. Pronoun reference

One of the classical problems discussed in linguistic literature is that of reference. The problem of reference has various manifestations and numerous linguistic theories have attempted to explain and model them (Lees and Klima, 1963; Langacker, 1969; Winograd, 1972; Charniak, 1974; Hobbs, 1986). I believe that DETE’s dynamics can allow it to learn correct reference in a relatively simple case, the meaning of the word “it”. “It” can refer in different contexts to different things. For instance, in one context “it” can refer to a ball, in another to a circle or any object, or even to an event.

The following experimental design can be used. After DETE learns the meanings of a set of individual words, including “triangle”, “ball”, “wall”, “hits”, “explodes”, and “bounces”, DETE will be exposed to a series of visual scenes of two kinds: (1) Objects hitting a wall and bouncing. (2) Objects hitting another objects which explode as a result of the hit. After these visual and verbal experiences, DETE will be given two sentences with the same syntax but without any visual input: (1) “Triangle hits wall. It bounces.” (2) “Triangle hits circle. It explodes.”

To find out how DETE interprets the word “it” in the two sentences we can look at the image generated in DETE’s “mind’s eye” in each case. In the first case, I expect that the context established by the rest of the words in the sentence will generate an image of a bouncing triangle. Notice that since “it” might refer to the wall instead of the triangle, it is possible also that DETE imagines the wall moving back after the hit. In other words, DETE may get confused by the two possibilities. However, since in all of DETE’s prior experiences with objects hitting the wall, the objects were those that bounced, DETE should not get confused and should generate the correct

visual image (a bouncing triangle). In other words, it will interpret "it" as being the triangle -- the first noun in the sentence (left of Figure 14.1).

Assume that in all of DETE's prior experiences with sharp objects (e.g., triangles, squares, etc.) hitting circles, the circles exploded (like balloons). Then when DETE gets the second sentence, due to its prior experience, it should imagine an exploding circle rather than an exploding triangle. In other words, in this case DETE interprets the word "it" as referring to the circle -- the second noun in the sentence (right of Figure 14.1).

**VERBAL
INPUT**

TRIANGLE HITS WALL.
IT BOUNCES.

TRIANGLE HITS BALL.
IT EXPLODES.

(BALL HITS TRIANGLE.
IT EXPLODES).

VISUAL IMAGINATION

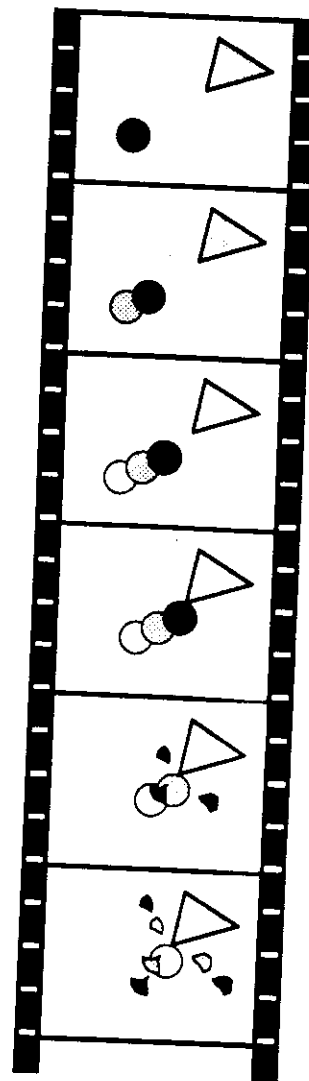
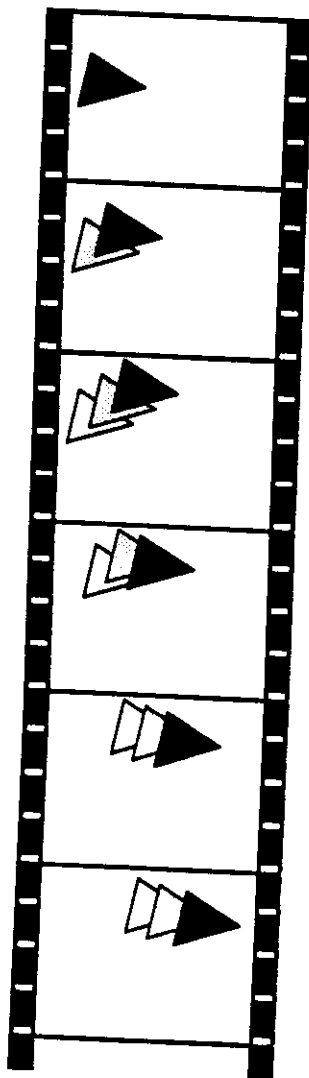


Figure 14.1: Learning pronoun resolution -- the meaning of "it"

Schematic view of the expected sequences of visual images generated in DETE's "mind's eye" by the two verbal inputs. For clarity, the motion of the objects is indicated by brighter "motion traces" left behind the dark objects.

(3) *Learning verbal-to-verbal associations*: DETE has not made yet the major leap of being able to learn new concepts by associating verbal input with prior visual and verbal experiences. For instance, it is not capable of understanding the meaning of the word "apple" by mentally processing the verbal input "Apple is a small red circle" without associating directly the word apple with a small red circle in the Visual Field. In other words, DETE lacks the ability to do verbal-to-verbal association. As part of our future work we intend to train DETE on examples such as: "Red is a color." or "Circle is a shape." Notice that in these cases the system is expected to have learned in advance the meanings of some or all of the individual words. By processing the verbal input it will acquire some additional aspects of the meaning of the individual words (if all of them have been learned before) or one specific meaning of the novel words in the sentence. For instance, if in the first sentence "Red is a color." the system has already learned the meaning of the word "red" and has also learned that "is a" means in general an association of the two words between which this phrase appears, then by processing the verbal input "Red is a color." the system should make the specific association. (In theoretical terms, it will build an "IS-A hierarchy"). As a system which possesses a declarative memory (DM), DETE should also be able to answer questions such as: Q: "What is red?" A: "A color." or Q: "What is circle?" A: "A shape (or an object)."

(4) *Prosodic processing of language*: It has been established empirically that in the early stages of language development children rely heavily upon prosodic cues and prosody is essential for language bootstrapping. If DETE is to model the development of language acquisition in humans, it is essential that it is sensitive to prosody. In its current implementation DETE has the potential of limited prosodic processing. The choice of verbal representation described in Chapter 4 is such that it can be used in a natural way to encode prosodic inflections in language which can be used in experiments. For instance, DETE should be able to learn words that differ only in the position of the accent such as the Spanish words "háblo" (= a talk) and "habló" (he talked).

14.2.3 Additional basic cognitive capacities

(1) *Mental representation of the teacher or other agents*: The ability of maintain and manipulate an internal model of its instructor or parent is evident in children. During social interactions, kids are aware when a teacher or a parent is attending to them and soon in their cognitive development they learn to detect when their own attempts to communicate something are being understood by the teacher. When children realize that they are not understood, they usually elaborate a new plan which often involves language creativity. Incorporating the ability of creative language usage in the model is of great interest for us. Empirical observations tell us that an interaction with a knowledgeable and capable person is necessary for the acquisition of language. Also kids, and for that matter an artificial system like DETE, must have (needs) a model of the partner, e.g. it has to know what this partner can do and how he/she does it. This knowledge must be acquired through interactive experiences. To be able to communicate with a teacher DETE needs to expand its vocabulary to include talking about self and teacher. At minimum it has to learn the meanings of the words "I" and "you".

(2) *Internal motivation to communicate*: Motivation or drive to communicate is the most essential component of human language development. The reason kids learn to communicate initially by gesticulations (point to something, orient the body to something) and later verbally is to achieve some goal (e.g., get attention or toy or food). It is interesting to observe that a developing child would keep asking (repeating a demand) until the response it expects is given by its partner in the dialog -- D-partner (this can be a parent, another child, etc.). If the D-partner misunderstands the child, it keeps repeating the demand in the same verbal form (few seconds apart with maybe changing intonation) or tries to elaborate it. The internal urge to keep asking and the mental ability to elaborate the demand so that it can be understood comes natural to the child. Notice also that the child is *aware* (has a belief or a model) whether he/she is being understood or not. What clues does a child use to establish such belief? Most often the cue is a response of a kind that the child expected generated by the D-partner. Such response can take many forms, e.g., the D-partner does the thing the kid wants; or the D-partner makes a gesture of understanding (ok, yes + repeat demand, or just with an appropriate intonation). The set of satisfactory responses are learned by the child through multiple experiences.

(3) *Imposed and self control*: Another very important observation, which is so trivial that we tend to overlook it, is that parents and kids share a common world and both can perform a variety of behaviors, however, they encourage each other to do only some of these behaviors and discourage each other from doing others. The reason for this is mainly to keep some personal/social homeostasis (set-point).

Unfortunately, at the current stage of the project, DETE does not possess neither internal motivation to acquire language (no set of goals to satisfy), nor does it have a model of its teacher. So DETE does not know if the response that it generates is understood by the teacher. Also, it does not have the necessary mechanisms to elaborate its response or demand so that it can be understood. Effectively, in its current version, DETE is a system that generates complex but only reflex-like verbal responses and does not have sophisticated mechanisms to elaborate its responses in accordance with feedback from its teacher. In other words, it cannot get into a meaningful dialog. Development of a computational model of parent and self imposed control during verbal or motor interaction with the environment is essential.

14.2.4 Higher cognitive functions

A natural path for DETE to follow is the path already taken by symbol manipulating systems for Natural Language Processing. Some prominent cornerstones along this path include: (1) SHRDLU (Winograd, 1972) a semantics-based NLP program which by representing language in terms of procedures for actions was capable of conducting dialog about a world of blocks, and was also capable of carrying out simple commands, such as "Put the blue pyramid on the green block". (2) MARGIE (Schank, 1975; Schank, 1981) a program based on a domain independent representational system (Conceptual Dependency theory) which was capable of making inferences. (3) SAM (Cullingford, 1978) -- a script based system capable of paraphrasing and learning stereotypic image sequences on the screen. (4) PAM (Wilensky, 1978; Wilensky, 1983) -- explanation based story understanding system endowed with goal/plan manipulation abilities. It was able to learn and process stories and infer unstated goals & plans of the actors (e.g., knight, princess, and dragon). (5) BORIS (Dyer, 1983) -- a system capable of reading text and reasoning about goals, plans, emotions, interpersonal relations. (6) OpEd (Alvarado, 1990) -- a system that reads editorials about economic protectionism and uses Argument Units to abstractly represent the beliefs of the participants in an argument.

An interesting research question is what needs to be added to DE TE to be able to go along this path of development? One reason for augmenting DE TE with additional functional modules is that then the above-mentioned higher cognitive abilities can be learned (vs. hand coded) as a result of which the system will be more robust.

14.2.5 Real time operation in real environments

In the current implementation DE TE operates in a simulated visual environment (the Blobs World). In itself this world is very impoverished and in many ways is not an adequate model of the real world. This adds additional limitations on DE TE's ability to acquire concepts that adequately correspond to the words which it hears. To be useful as a research tool for computational modeling of early child language acquisition, it is imperative that DE TE be placed in a real environment. As part of our future work we intend to outfit a mobile robot with video and speech processing equipment and link it (by cable via a workstation front-end) to the CM-2 Connection Machine which will be running a DE TE-style software system. We intend to teach this "sensory-grounded" robot simple verbal commands for manipulation of the objects in the environment.

In order to perform FINGER-object interactions the system will require additional features: a) To be able to accomplish the command "DRAG the ball from A to B." The FINGER must be capable of connecting itself (via a hook, by glue or by friction) to the object. b) In order to execute the command "STOP the ball." DE TE requires either the ability to control the final state of the objects (in case of elastic objects) by the FINGER or the objects should be able to stick to the FINGER (i.e. non-elastic interactions allowed).

In a real world scenario DE TE needs to learn spatial relations that involved composite rather than simple objects. For instance, it needs to learn the meaning of the sentence "The ball is between the hind legs of the chair". Correspondingly it should also learn actions involving composite motions.

In a real world DE TE will require 3-D capabilities. It should learn to navigate in space, approach objects, and sense the changes in their visual appearance (e.g., DE TE would *move* and an object on the retina would grow in size; or DE TE would *grasp* and see its finger on retina touching an object; DE TE would *turn* and a new object would appear to move on retina -- but it would be DE TE moving, not the object being seen). The language which DE TE is taught needs also to be augmented to be able to describe the real world.

Ultimately we envision two or more robots talking and interacting with each other. With built-in motivational mechanisms the robots will engage in solving tasks such as stacking blocks in cooperation with each other. We will explore the issues in teaching one robot to give/receive commands from another robot. With the help of a human teacher the robots should learn to produce utterances such as "It is your turn." (notice that the concept of "turn taking" cannot currently be represented in DE TE) or "You can put that block on after I put this one on." or "Bring me the red block from over there." or " Let's start a stack here."

14.2.6 Research on psychological validity

The ultimate test of any model is how well it fits the data. In the last few years, powerful computational tools have been developed which allow a standardized approach to the analysis of early language acquisition in children (MacWhinney, 1991). Using these tools and protocols from the extensive database of child's talk collected as a part of the CHILDES project we can compare the stages through which DE TE goes (in learning its language) with the stages of human child's language development. It is important to state here that such a comparison is not necessarily fair

(legitimate) since the two systems (DETE and a human child) differ immensely in the levels of their complexity, in the content of their environments and the length of time during which they are exposed to these environments. In other words, such comparisons can be at the most only suggestive and are a useful exercise which can be expanded to a full-blown project once more sophisticated computational models of early language acquisition are developed.

14.3 Conclusions

The following major contributions of the research described in this thesis can be stated:

(1) *Natural language acquisition*: DETE demonstrates that a large-scale neural/procedural system capable of acquisition of basic linguistic skills through associations of visual and verbal inputs can be developed and successfully tested. The experimental results show that:

- Symbols that refer directly to objects or features in the visual world like “ball” “red”, etc., (but also more abstract words like “shape”, “color”, etc.) are grounded in sensory experiences. While NLP systems, e.g., Conceptual Dependency, assume this grounding by creating ungrounded symbols and then relate them causally, DETE actually does the grounding and proposes explicit representations of physical objects and actions.
- Subsets of language semantics and syntax (word order and morphological inflections) need not have explicit rules. Instead they can be represented and processed as temporal/spatial correlations in sequence memories. Also, they can be learned dynamically through interactions of verbal and sensory experiences.
- A visual binding mechanism based on phase-locked oscillations of neural assemblies can successfully be used for processing associations of visual and verbal inputs. It allows the system to perform visual-to-verbal generalizations and vice versa.
- A class of visual feature planes (and the corresponding memories) connected to a verbal memory module can be successfully used in the process of word meaning acquisition. Specifically, we demonstrate how some linguistic theories for the representation of space and time and notably those of George Lakoff, Leon Talmy, and Reichenbach (Lakoff, 1989; Talmy, 1983; Raichenbach, 1947) can be implemented and tested computationally.
- A pseudo-acoustic (gra-phonemic) representation of the verbal inputs is proposed. It allows the linguistic processing in DETE to be linked to acoustic processing of speech rather than textual input. This representation can be used to handle prosodic inflections in the verbal input which will allow future versions of DETE to distinguish between various language accents of speakers.
- A novel and substantially more robust representation (as compared to symbolic and other connectionist models) of lexical items in the lexicon is proposed. In this representation the verbal tokens for the individual concepts are stored as distributed *ltm* patterns in the Verbal Memory, whereas the meanings of the words are represented as distributed patterns across a set of Visual Feature Memories and Temporal Memory Planes. The representations of the lexical items are not hand-coded (as in symbolic systems) or generated as random patterns (as in the majority of the connectionist systems) but are acquired in the process of associations between visual and verbal experiences.

(2) *Connectionism & Neural Networks*:

a) The KATAMIC model: As an essential part of this thesis, a novel neural architecture for rapid learning, recognition, and recall of pattern sequences, called the KATAMIC sequential associative memory, was designed, implemented and evaluated. The KATAMIC memory uses a novel type of neural element (not classical McCulloch-Pitts neurons) called predictrons. Like real neurons, predictrons have dendritic branches formed by dendritic compartments with complex dynamics and built-in short-term and long-term storage capacity. Our experiments demonstrate that the KATAMIC memory exhibits:

- *Extremely rapid learning*: Only a few exposures (on average 4 to 6) to a particular sequence are sufficient for learning.
- *Flexible memory capacity*: Multiple sequences can be stored in the network, with a memory/processor ratio comparable to, if not better than that of other neural net, PDP or connectionist models.
- *Sequence completion*: A short cue can retrieve the complete sequence.
- *Sequence recognition*: A built-in mechanism allows sequence recognition on a pattern-by-pattern basis, which is used internally for switching from learning to performance mode.
- *Fault and noise tolerance*: Missing elements (bits within patterns or whole patterns missing within sequences can be tolerated) within a reasonable range (30% of the number of 1-bits).
- *Integrated processing*: The model is capable of concurrent learning, recognition, and recall of sequences. This is a significant improvement over most previously proposed models that focus only on one specific aspect of processing at a time, e.g., the PDP class of models. Also, the KATAMIC model has the built-in ability to monitor the quality of its performance and automatically to switch (during the processing of a particular sequence) from accepting input to producing output. This ability is critical in the sense that it allows DETE to learn while exploring the environment.

b) DETE's architecture: In terms of its architectural design DETE falls midway between "minimalist" PDP architectures and "completely innate" architectures. The former use minimal complexity -- e.g., three layers of nodes and a simple learning rule and attempt to discover regularities in the input space through thousands (and often hundreds of thousands) of training trials. The latter are completely pre-wired, do minimal learning if at all and accomplish pre-specified tasks in an almost reflexive manner. DETE's neural architecture, which combines localist representation (within the Visual Feature Planes) with distributed (space and time smeared) representation (within the Visual Feature Memories) makes the complex task of perceptually grounded language acquisition (PGLA) learnable within a reasonable amount of time and with substantially smaller number of training trials.

(3) *Computational Technology*: The implementation of DETE on the CM-2 Connection Machine demonstrates that this fine-grain SIMD massively parallel computer is an adequate platform for implementation of large-scale neural architectures.

- Unlike most of the connectionist/PDP/neural network models, which contain from few dozens to few hundreds of computational units, DETE is a large-scale model which contains over a million neural elements. To achieve this, DETE took advantage of the virtual processing capability of the CM-2.

- DETE's memory architecture, which contains several highly interconnected modules, is significantly more complex than all current connectionist models used for language acquisition tasks. Despite its complex connectivity, the model rendered itself to a straightforward implementation on the CM-2 since the 2-D and 3-D topography of the modules was easily mapped to the hypercube architecture of the CM-2.

- Taking advantage of the supercomputer power of the CM-2 we were able to run numerous computationally intensive experiments with DETE in a reasonable amount of time. Such a volume and complexity of experiments would be impractical to do with a general purpose serial processor (e.g., a workstation) and would tax even a supercomputer like the Cray.

- Taking advantage of *LISP -- this high level language is designed for parallel programming of the CM-2 and made DETE's code relatively compact. In contrast, an implementation of the same system on a serial computer would require substantially more code.

(4) *Computational Neuroscience*: An attempt was made to establish mappings between DETE's structure and function and the connectivity and function of brain areas involved in vision, attention, memory, and the integrated processing of language.

- A novel theory of cerebellar cortical function is proposed. Founded on the well established fact that the cerebellar cortex serves as a sequential associative memory for storage of motor programs, this theory suggests an actual neural representation of these motor programs and the cellular mechanisms involved in their storage and recall. Based on neurophysiological, neuroanatomical, and neurochemical data about the structure and function of this cortex, the theory suggests specific functionality for the major neuronal types in the cerebellum. Purkinje cells are viewed as neural elements (predictrons) which learn to predict consecutive inputs (Action Potentials) coming along parallel fibers. Golgi cells (recognitrons) are viewed as monitoring the correctness of the predictions made by the Purkinje cells with respect to the actual inputs coming along the mossy fibers. Cerebellar granule cells (BSSs) are involved in switching of the cerebellar cortical function from "attending" to "performing" mode.

- A model of selective attention processing, based on phase-locking of oscillating neural assemblies, was developed. Its major characteristic is the ability to simultaneously represent several (up to 4) objects that appear in the visual field and to keep one of them in the focus of attention while treating the rest of them as a context.

- Possible mappings between the individual components of DETE's architecture and brain structures implicated in carrying out corresponding functions are proposed and discussed.

BIBLIOGRAPHY

- Albus, J.S. (1971). A theory of cerebellar function. *Mathematical Biosciences*, **10**, 25-61.
- Alkon, D.L. (1984). Calcium-mediated reduction of ionic currents: a biophysical memory trace. *Science*, **226**, 1037-1045.
- Alkon, D.L. (1989). Memory Storage and Neural Systems. *Scientific American*, **261**, 42-50.
- Allen, R.B. (1987). Several studies on natural language and back-propagation. *Proceedings of the International Conference on Neural Networks, San Diego, CA*, **2**, 335-341, (Abstract).
- Allen, R.B. (1988). Sequential connectionist networks for answering simple questions about a microworld. *Proceedings of the Cognitive Science Society, (Montreal, August 1988)*, 489-495, (Abstract).
- Allen, R.B. and Riecken, M.E. (1988). Reference in Connectionist Language Users. *Connectionism in Perspective (Zurich, October 1988)*, (Abstract).
- Altmann, G. and Shillcock, R.C. (1986). Statistical studies of the lexicon. *Association Européenne de psycholinguistique*, newsletter no. **13**.
- Alvarado, S.J. (1990). *Understanding Editorial Text: A Computer Model of Argument Comprehension*. Kluwer Academic, Norwell, MA.
- Amit, D.J. (1988). Neural networks counting chimes. *Proceedings of the National Academy of Sciences*, **85**, 2141-2145.
- Andersen, E.S., Dunlea, A., and Kekelis, L.S. (1984). Blind children's language: resolving some differences. *Journal of Child Language*, **11**, 645-664.
- Anderson, J.A. (1970). Two models for memory organization using interacting traces. *Mathematical Biosciences*, **8**, 137-160.
- Anderson, J.A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, **14**, 197-220.
- Anderson, J.R. (1983). *The Architecture of Cognition*. Harvard University Press, Cambridge, MA.
- Anderson, J.R. and Bower, G.H. (1973). *Human associative memory*. Winston, Washington D.C.
- Arbib, M.A. (1989). Neural control of movement. In *The Metaphorical Brain 2: Neural Networks and Beyond*. (pp. 277-319). John Willey & Sons, New York, NY.
- Baddeley, A.D. (1982). Implications of neuropsychological evidence for theories of normal memory. *Philosophical Transactions of the Royal Society London [Biology]*, **298**, 59-72.
- Baddeley, A.D. (1986). *Working Memory*. Oxford University Press, New York, NY.
- Ballard, D.H. and Brown, C.M. (1982). *Computer Vision*. Prentice Hall, New York, NY.
- Barnden, J. (1989). Neural-net implementation of complex symbol-processing in a mental model approach to syllogistic reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-89)*. (pp. 568-573).
- Baron, R.J. (1987). *The Cerebral Computer*. Lawrence Erlbaum Associates, Hillsdale, NJ.

- Bates, E. (1976). *Language and Context: The Acquisition of Pragmatics*. Academic Press, New York, NY.
- Bell, T. (1988). Sequential processing using attractor transitions. In D. Touretzky, G. Hinton, and T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. (pp. 93-102). Morgan Kaufman, San Mateo, CA.
- Benson, D.F. (1979). *Aphasia, Alexia, and Agraphia*. Churchill Livingstone, New York, NY.
- Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, **94**(2), 115-147.
- Bienenstock, E. and von der Malsburg, C. (1987). A neural network for invariant pattern recognition. *Europhysics Letters*, **4**(1), 121-126.
- Bower, J.M. and Woolston, D.C. (1983). Congruence of spatial organization of tactile projections to granule cell and Purkinje cell layers of cerebellar hemispheres of the albino rat: vertical organization of cerebellar cortex. *Journal of Neurophysiology*, **49**, 745-766.
- Bowerman, M. (1983). How do children avoid constructing an overly general grammar in the absence of feedback about what is not a sentence. In *Papers and Reports on Child Language Development*. Stanford University, Palo Alto, CA.
- Boylls, C.C. (1975). *A Theory of Cerebellar Function with Applications to Locomotion*. COINS Tech. Rept. Vol 76-1, Amherst, MA.
- Braine, M. (1971). On two types of models of the internalization of grammars. In D. Slobin (Ed.), *The Ontogenesis of Grammar*. Academic Press, London.
- Brindley, G.S. (1964). The use made by the cerebellum of the information that it receives from the sense organs. *International Brain Research Organization Bulletin*, **3**, 80.
- Broca, P. (1861). Nouvelle observation d'aphémie produite par une lésion de la moitié postérieure des deuxième et troisième circonvolutions frontales. *Bulletin de la Société Anatomique*, **6**, 398-407.
- Broca, P. (1865). Sur le siège de la faculté du langage articulé. *Bulletin de la Société Anthropologie*, **6**, 377-393.
- Bruner, J. (1975). The ontogenesis of speech acts. *Journal of Child Language*, **2**, 1-19.
- Bruner, J. (1983). *Child's Talk*. W. W. Norton, New York, NY.
- Buhmann, J. (1989). Oscillations and low firing rates in associative memory neural networks. *Physical Review*, **40**(7), 4145-4148.
- Buhmann, J. and Schulten, K. (1988). Sorting sequences of biased patterns in neural networks. In R. Eckmiller and C. von der Malsburg (Eds.), *Neural Computers*. NATO ASI series Vol. F41.
- Buhmann, J. and Schulten, K. (1987). Noise-driven temporal association in neural networks. *Europhysics Letters*, **4**(10), 1205-1209.
- Buhmann, J. and von der Malsburg, C. (1990). Sensory segmentation in oscillatory neural networks, (submitted).
- Butter, C.M. (1987). Varieties of attention and disturbances of attention: A neuropsychological analysis. In M. Jeannerod (Ed.), *Neurophysiological and Neuropsychological Aspects of Spatial Neglect*. (pp. 1-24). North Holland, Amsterdam.
- Caplan, D. (1980). *Biological Studies of Mental Processes*. The MIT Press, Cambridge, MA.

- Eccles, J.C. (1973). The cerebellum as a computer: Patterns in space and time. *Journal of Physiology (London)*, **229**, 1-32.
- Eccles, J.C. (1977). An instruction-selection theory of learning in the cerebellar cortex. *Brain Research*, **127**, 327-352.
- Eccles, J.C., Ito, M., and Szentagothai, J. (1967). *The Cerebellum as a Neuronal Machine*. Springer Verlag, Berlin.
- Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, W., Munk, M., and Reitboeck, H.J. (1988). Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biological Cybernetics*, **60**, 121-130.
- Edelman, G.M. (1987). *Neural Darwinism: The Theory of Neuronal Group Selection*. Basic Books, Inc., New York.
- Edelman, G.M. and Reeke, G.N. (1982). Selective networks capable of representative transformations, limited generalizations, and associative memory. *Proceedings of the National Academy of Sciences, USA [Neurobiology]*, **79**, 2091-2093.
- Elman, J.L. (1988). *Finding Structure in Time*. Technical Report 8801, Center for Research in Language, University of California, San Diego.
- Elman, J.L. (1989a). Structured representations and connectionist models. In *Proceedings of the Eleventh Annual Cognitive Science Society Conference*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Elman, J.L. (1989b). *Representation of Structure in Connectionist Models*. Technical Report 8903, Center for Research in Language, University of California, San Diego.
- Elman, J.L. (1990). Finding structure in time. *Cognitive Science*, **14**, 179-211.
- Eriksen, C.W. and Hoffman, J.E. (1974). Selective attention: Noise suppression or signal enhancement? *Bulletin of the Psychonomic Society*, **4**, 587-589.
- Eriksen, C.W. and Murphy, T.D. (1987). Movement of attentional focus across the visual field: A critical look at the evidence. *Perception and Psychophysics*, **42**, 299-305.
- Eriksen, C.W. and Yeh, Y.-Y. (1985). Allocation of attention in the visual field. *Journal of Experimental Psychology: Human Perception and Performance*, **11**, 583-597.
- Fahlman, S. (1988). Faster-learning variations on back propagation: An empirical study. In *Proceedings of the 1988 Connectionist Summer School*. Morgan Kaufman, San Mateo.
- Fahlman, S.E. (1977). *NETL: A System for Representing and Using Real-World Knowledge*. The MIT Press, Cambridge, MA.
- Fauconnier, G. (1985). *Mental Spaces*. MIT Press, Cambridge, MA.
- Feldman, J.A. (1982). Dynamic connections in neural networks. *Biological Cybernetics*, **46**, 27-39.
- Feldman, J.A. (1988). *Time, Space and Form in Vision*. Technical Report TR-88-011, ICSI, Berkeley, CA.
- Feldman, J.A., Lakoff, G., Stolcke, A., and Hollbach Weber, S. (1990). *Miniature Language Acquisition: A touchstone for cognitive science*. Technical Report TR-90-009, ICSI, Berkeley, CA.

- Feller, W. (1957). *Introduction to Probability Theory and its Applications (Vol. 1)*. John Wiley, New York.
- Fisher, D.H. and McKusick, K.B. (1989). An empirical comparison of ID3 and back-propagation. In N.S. Sridharan (Ed.), *Proceedings of IJCAI-90, Detroit, MI, August 20-25*. (pp. 788-793). Morgan Kaufmann, San Mateo, CA.
- Fodor, J.A. (1975). *The Language of Thought*. Thomas Y. Crowell, New York.
- Fodor, J.A. (1987). *Psychosemantics*. The MIT Press/Bradford, Cambridge, MA.
- Fodor, J.A. and Pylyshyn, Z.W. (1988). Connectionism and cognitive architecture: a critical analysis. *Cognition*, **28**, 3-71.
- Forbus, K.D. (1985). Qualitative process theory. In D.G. Bobrow (Ed.), *Qualitative Reasoning about Physical Systems*. The MIT Press/Bradford, Cambridge, MA.
- Fujita, M. (1982). Adaptive filter model of the cerebellum. *Biological Cybernetics*, **45**, 195-206.
- Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Journal of Biological Cybernetics*, **20**, 121-136.
- Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Journal of Biological Cybernetics*, **36**, 193-202.
- Fukushima, K. (1987a). A neural network model for selective attention. *Proceedings of the IEEE First International Conference on Neural Networks*, **2**, 11-18, (Abstract).
- Fukushima, K. (1987b). Neural network model for selective attention in visual pattern recognition and associative recall. *Applied Optics*, **26(23)**, 4985-4992.
- Fuster, J.M. (1980). *The prefrontal cortex: anatomy, physiology and neuropsychology of the frontal lobe*. Raven Press, New York.
- Fuster, J.M. (1985). The prefrontal cortex, mediator of cross-temporal contingencies. *Human Neurobiology*, **4**, 169-179.
- Gardner-Medwin, A.R. (1976). The recall of events through the learning of associations between their parts. *Proceedings of the Royal Society London [Biology]*, **194**, 375-402.
- Geschwind, N. (1970). The organization of language in the brain. *Science*, **170**, 940-944.
- Goodglass, H. and Kaplan, E. (1972). *The assessment of aphasia and related disorders*. Lea and Febiger, Philadelphia.
- Gouras, P. (1985a). Physiological optics, accommodation, and stereopsis. In E.R. Kandel and J.H. Schwartz (Eds.), *Principles of Neural Science*. (pp. 866-875). Elsevier, New York, NY.
- Gouras, P. (1985b). Oculomotor system. In E.R. Kandel and J.H. Schwartz (Eds.), *Principles of Neural Science*. (pp. 571-583). Elsevier, New York, NY.
- Graf, P., Squire, L.R., and Mandler, G. (1984). The information that amnesic patients do not forget. *Journal of Experimental Psychology [Learning, Memory and Cognition]*, **10**, 164-178.
- Gray, C.M., Engel, A.K., Konig, P., and Singer, W. (1990). Stimulus-dependent neuronal oscillations in cat visual cortex: Receptive field properties and feature dependence. *Journal of Neuroscience*, **2**, 607.

- Caplan, D., Lecours, A.R., and Smith, A. (Eds.) (1984). *Biological Perspectives on Language*. The MIT Press, Cambridge, MA.
- Caramazza, A. and Zurif, E.B. (1976). Dissociation of algorithmic and heuristic processes in sentence comprehension. *Brain and Language*, **3**, 572-582.
- Carpenter, M.B. and Sutin, J. (1983). *Human Neuroanatomy*. Williams & Wilkins, Baltimore, MD.
- Carr, C.E. and Konishi, M. (1988). Axonal delay lines for time measurement in the owl's brainstem. *Proceedings of the National Academy of Sciences, USA [Biophysics]*, **85**, 8311-8316.
- Chan-Palay, V. and Palay, S.L. (1971). The synapse *en marron* between Golgi II neurons and mossy fibers in the rat cerebellar cortex. *Z. Anat. Entwickl. -Gesch.*, **133**, 274-287.
- Chapman, D. (1990a). *Vision, instruction, and action*. Technical Report 1204, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
- Chapman, D. (1990b). *Intermediate vision: Architecture, implementation, and use*. Technical Report TR-90-6, Teleos Research, Palo Alto, CA.
- Charniak, E. (1974). *'He will make you take it back': A study in the pragmatics of language*. Working Paper 5. Institute for Semantic and Cognitive Studies.
- Charniak, E. (1986). A neat theory of marker passing. In *Proceedings of the Sixth National Conference on Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., Los Altos, CA.
- Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origine, and Use*. Praeger, New York, NY.
- Chou, P.A. (1988). The capacity of the Kanerva associative memory is exponential. In D.Z. Anderson (Ed.), *Neural Information Processing Systems*. (pp. 184-191). American Institute of Physics, New York.
- Coltheart, M. (1983). Iconic memory. *Philosophical Transactions of the Royal Society London [Biology]*, **302**, 283-294.
- Coltheart, M. (1987). Functional architecture of the language processing system. In M. Coltheart, G. Sartori, and R. Job (Eds.), *The Cognitive Neuropsychology of Language*. (pp. 1-25). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Cottrell, G.W. and Small, S.L. (1983). A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory*, **6(1)**, 89-120.
- Crick, F. (1984). Function of the thalamic reticular complex: the searchlight hypothesis. *Proceedings of the National Academy of Sciences, USA*, **81**, 4586-4590.
- Crick, F. and Koch, C. (1990). Towards a neurobiological theory of consciousness. *Seminars in the Neurosciences*, **2(4)**, 263-275.
- Cullingford, R.E. (1978). *Script Application: Computer Understanding of Newspaper Stories*. Ph.D. Thesis, Department of Computer Science, Yale University, Technical Report 116.
- Côté, L. and Crutcher, M.D. (1985). Motor functions of the basal ganglia and diseases of transmitter metabolism. In E.R. Kandel and J.H. Schwartz (Eds.), *Principles of Neural Science*. (pp. 523-536). Elsevier, New York, NY.

- D'Autrechy, C.L. and Reggia, J.A. (1989). *An overview of sequence processing by connectionist models*. Technical Report UMIACS-TR-89-82 (CS-TR-2301), University of Maryland, College Park, MD.
- Damasio, A.R. (1989). The brain binds entities and events by multiregional activation from convergence zones. *Neural Computation*, **1**, 123-132.
- Damasio, A.R. (1990). Synchronous activation in multiple cortical regions: a mechanism for recall. *Seminars in the Neurosciences*, **2**, 287-296.
- Damasio, A.R., Eslinger, P.J., Damasio, H., Van Hoesen, G.W., and Cornell, S. (1985). Multimodal amnesic syndrome following bilateral temporal and basal forebrain damage. *Archives of Neurology*, **42**, 252-259.
- DeJong, G.F. (1979). *Skimming Stories in Real Time: An Experiment in Integrated Understanding*. Ph.D. Thesis, Department of Computer Science, Yale University, Research Report 158.
- Dejerine, J. (1891). Sur un cas de cécité verbale avec agraphie, suivi d'autopsie. *C. R. Séances Mem. Soc. Biol.*, **43**, 197-201.
- Delattre, P.C., Liberman, A.M., and Cooper, F.S. (1962). Formant transitions and loci as acoustic correlates of place of articulation in American fricatives. *Studies Linguistica*, **16**, 104-21.
- Demer, J.L. and Robinson, D.A. (1981). Effect of reversible lesions of the inferior olive (IO) on gain of the vestibulo-ocular reflex (VOR). *Investigative Ophthalmology and Visual Science, Supplement*, **20(3)**, 57.
- Dempsey, E.W. and Morison, R.S. (1942). The production of rhythmically recurrent cortical potentials after localized thalamic stimulation. *American Journal of Physiology*, **135**, 293-300.
- Desimone, R., Moran, J., and Spitzer, H. (1989). Neural mechanisms of attention in extrastriate cortex of monkeys. In M.A. Arbib and S-i. Amari (Eds.), *Dynamic Interactions in Neural Networks: Models and Data*. (pp. 169-182). Springer-Verlag, New York, NY.
- Dolan, C.P. (1989). *Tensor Manipulation Networks: Connectionist and Symbolic Approaches to Comprehension, Learning and Planning*. Ph.D. Thesis, Computer Science Department, University of California, Los Angeles.
- Dolan, C.P. and Smolensky, P. (1989). Implementing a connectionist production system using tensor products. In D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kaufmann, Los Altos, CA.
- Drescher, G.L. (1991). *Made-up Minds: A Constructivist Approach to Artificial Intelligence*. The MIT Press, Cambridge, MA.
- Dunlea, A. (1989). *Vision and the Emergence of Meaning*. Oxford University Press, New York, NY.
- Dyer, M.G. (1983). *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension*. The MIT Press, Cambridge, MA.
- Dyer, M.G. (1990). Distributed symbol formation and processing in connectionist networks. *Journal of Experimental and Theoretical Artificial Intelligence*, **2**, 215-239.
- Dyer, M.G. (1991). Symbolic NeuroEngineering for natural language processing: A multi-level research approach. In J. Barnden and J. Pollack (Eds.), *High-Level Connectionist Models*. (pp. 32-86). Ablex Publishers, New York, NY.

- Gray, C.M., Konig, P., Engel, A.K., and Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, **338**, 334-337.
- Gray, C.M. and Singer, W. (1989). Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Sciences, USA [Biophysics]*, **85**, 1698-1702.
- Grossberg, S. (1969). Some networks that can learn, remember, and reproduce any number of complicated space-time patterns. *Journal of Mathematics and Mechanics*, **19**, 53-91.
- Grossberg, S. and Pepe, J. (1970). Schizophrenia: Possible dependance of association span, bowing, and priimacy vs. recency on spiking threshold. *Behavioral Science*, **15**, 359-362.
- Haberly, L.B. (1990). Olfactory cortex. In G.M. Shepherd (Ed.), *The Synaptic Organization of the Brain*. (pp. 317-345). Oxford University Press, New York.
- Halgren, E. (1984). Human hippocampal and amygdala recording and stimulation: Evidence for a neural model of recent memory. In N. Butters and L. Squire (Eds.), *The neuropsychology of memory*. (pp. 165-181). Guilford, New York.
- Halgren, E. (1989). Physiological integration of the declarative memory system. In R.P. Kesner and D.S. Olton (Eds.), *Neurobiology of Comparative Cognition*. (in-press) Erlbaum, Hillsdale, NJ.
- Hamori, J. and Szentagothai, J. (1966). Identification under the electron microscope of climbing fibers and their synaptic contacts. *Experimental Brain Research*, **1**, 65-81.
- Harnad, S. (1987). Category induction and representation. In S. Harnad (Ed.), *Categorical perception: The groundwork of cognition*. Cambridge University Press, New York.
- Harnad, S. (1989). The symbol grounding problem. *Physica D*.
- Hartline, H.K. (1949). Inhibition of activity of visual receptors by illuminating nearby retinal areas in the limulus eye. *Federation Proceedings*, **8(1)**, 69.
- Hebb, D.O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York.
- Heilman, K.M. (1975). Auditory affective agnosia. Disturbed comprehension of affective speech. *Journal of Neurology Neurosurgery and Psychiatry*, **38**, 69-72.
- Heilman, K.M. and Scholes, R.J. (1976). The nature of comprehension errors in Broca's, conduction, and Wernicke's aphasics. *Cortex*, **12**, 258-265.
- Hendler, J. (1988). *Integrating Marker Passing and Problem Solving: A Spreading Activation Approach to Improved Choice in Planning*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Hikosaka, O. (1989). Role of basal ganglia in initiation of voluntary movements. In M.A. Arbib and S-i. Amari (Eds.), *Dynamic Interactions in Neural Networks: Models and Data*. (pp. 153-167). Springer Verlag, New York.
- Hillis, D.W. (1985). *The Connection Machine*. The MIT Press, Cambridge, MA.
- Hillis, D.W. and Steele, G.L. (1986). Data parallel algorithms. *Communications of the ACM*, **29**, 1170-1183.
- Hinrichs, E.W. (1988). Tense, quantifiers, and contexts. *Computational Linguistics*, **14(2)**, 3-14.

- Hinton, G.E. (1981). Implementing semantic networks in parallel hardware. In G.E. Hinton and J.A. Anderson (Eds.), *Parallel Models of Associative Memory*. (pp. 161-188). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Hinton, G.E. (1986). Learning distributed representations of concepts. In *Proceedings of the Eight Annual Cognitive Science Society Conference*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Hobbs, J.R. (1986). Resolving pronoun references. In B.J. Grosz, K.S. Jones, and B.L. Webber (Eds.), *Readings in Natural Language Processing*. (pp. 339-352). Morgan Kaufman, Los Altos, CA.
- Hollbach Weber, S. and Stolcke, A. (1990). *L0: A Testbed for Miniature Language Acquisition*. TR-90-010, International Computer Science Institute, Berkeley, CA.
- Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA [Biophysics]*, **79**, 2554-2558.
- Hopfield, J.J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA [Biophysics]*, **81**, 3088-3092.
- Houk, J.C. (1987). Model of the cerebellum as an array of adjustable pattern generators. In M. Glickstein, C. Yeo, and J. Stein (Eds.), *Cerebellum and Neuronal Plasticity*. (pp. 249-260). Plenum Press, New York, NY.
- Hubel, D.H. and Wiesel, T.N. (1962). Receptive fields, binocular interaction and functional architecture of the cat's visual cortex. *Journal of Physiology (London)*, **160**, 106-154.
- Hubel, D.H. and Wiesel, T.N. (1974). Sequence regularity and geometry of orientation columns in monkey striate cortex. *Journal of Comparative Neurology*, **158**, 267-294.
- Hubel, D.H. and Wiesel, T.N. (1977). Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society London [Biology]*, **198**, 1-59.
- Hummel, J.E. and Biederman, I. (1990). *Dynamic binding in a neural network for shape recognition*. Technical Report NO. 90-5, Department of Psychology, University of Minnesota, Minneapolis, MN.
- Ito, M. (1979). Neuroplasticity. Is the cerebellum really a computer? *Trends in Neuroscience*, **2**, 122-126.
- Ito, M. (1987). Characterization of synaptic plasticity in the cerebellar and cerebral neocortex. In J.P. Changeux and M. Nonishi (Eds.), *The Neural and Molecular Basis of Learning*. (pp. 276-279). John Wiley, New York, NY.
- Jackendoff, R. (1983). *Semantics and Cognition*. MIT Press, Cambridge, MA.
- Jackendoff, R. (1987). *Consciousness and the Computational Mind*. The MIT Press, Cambridge, MA.
- Jaeger, C.B., Kapoor, R., and Llinas, R. (1988). Cytology and organization of rat cerebellar organ cultures. *Neuroscience*, **26**, 509-538.
- James, W. (1890). Association. In *Psychology (Briefer Course)*. (pp. 253-279). Holt, New York, NY.

- Jordan, M.I. (1986). Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. In *Program of the Eighth Annual Conference of the Cognitive Science Society*. (pp. 531-546). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Julesz, B. (1991). Early vision and focal attention. *Review of Modern Physics*, (in press),
- Kandel, E.R. and Schwartz, J.H. (1985). *Principles of Neural Science*. Elsevier Science Publishing Co., New York, NY.
- Kanerva, P. (1984). *Self-propagating search: A unified theory of memory*. Ph.D. Thesis, Stanford University, Department of Computer Science.
- Kanerva, P. (1988). *Sparse Distributed Memory*. The MIT Press, Cambridge, MA.
- Kano, M. and Kato, M. (1987). Quisqualate receptors are specifically involved in cerebellar synaptic plasticity. *Nature*, **325**, 276-279.
- Keeler, J.D. (1988). Comparison between Kanerva's SDM and Hopfield-type Neural Networks. *Cognitive Science*, **12**, 299-329.
- Kleinfeld, D. (1986). Sequential state generation by model neural networks. *Proceedings of the National Academy of Sciences, USA [Biophysics]*, **83**, 9469-9473.
- Koch, C. (1987). The action of the corticofugal pathway on sensory thalamic nuclei: a hypothesis. *Neuroscience*, **23**(2), 399-406.
- Koch, C. and Poggio, T. (1985). Biophysics of computation: nerves, synapses, and membranes. In G.M. Edelman, W.E. Gall, and W.M. Cowan (Eds.), *New Insights into Synaptic Function*. (pp. 616-637). Neuroscience Research Foundation, New York.
- Koch, C. and Ullman, S. (1985). Shifts in selective attention: towards the underlying neural circuitry. *Human Neurobiology*, **4**, 219-227.
- Koffka, K. (1935). *Principles of Gestalt Psychology*. Harcourt, New York.
- Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers*, **C-21**(4), 353-359.
- Kohonen, T. (1977). *Associative Memory: A System-theoretical Approach*. Springer Verlag, Berlin.
- Kohonen, T. (1982a). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 59-69.
- Kohonen, T. (1982b). Clustering, taxonomy, and topological maps of patterns. In *Proceedings of the Sixth International Conference on Pattern Recognition*. IEEE Computer Society Press.
- Kohonen, T. (1984). *Self-Organization and Associative Memory*. Springer-Verlag, New York.
- Kohonen, T., Oja, E., and Lehtiö, P. (1989). Storage and processing of information in distributed associative memory systems. In G.E. Hinton and J.A. Anderson (Eds.), *Parallel Models of Associative Memory*. (pp. 129-170). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Kolodner, J.L. (1984). *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*. Lawrence Erlbaum and Associates, Hillsdale, NJ.
- Kosslyn, S.M. (1985). Computational neuropsychology: A new perspective on mental imagery. *Naval Research Reviews*, **37**(4), 30-40.
- LaBerge, D. (1983). Spatial extent of attention to letters and words. *Journal of Experimental Psychology: Human Perception and Performance*, **9**, 371-379.

- LaBerge, D. and Brown, V. (1989). Theory of attentional operations in shape identification. *Psychological Review*, **96**, 101-124.
- Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. University Press, Chicago, IL.
- Lakoff, G. (1989). A suggestion for a linguistics with connectionist foundations. In D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. (pp. 301-314). Morgan Kaufmann, San Mateo, CA.
- Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By*. The University of Chicago Press, Chicago, IL.
- Langacker, R. (1969). On pronomialization and the chain of command. In D. Reibel and S. Schane (Eds.), *Modern Studies in English*. (pp. 160-186). Prentice Hall, Englewood Cliffs.
- Langacker, R. (1987). *Foundations of Cognitive Grammar. Vol. 2*. Stanford University Press, Stanford, CA.
- Lange, T.E. and Dyer, M.G. (1989). High-level inferencing in a connectionist network. *Connection Science*, **1**(2), 181-217.
- Lashley, K.S. (1951). On serial order. In L. Jeffress (Ed.), *Cerebral mechanisms of behavior*. John Wiley, New York, NY.
- Lebowitz, M. (1980). *Generalization and Memory in an Integrated Understanding System*. Ph.D. Thesis, Department of Computer Science, Yale University.
- Lecours, A.-R., Lhermitte, F., and Bryans, B. (1983). Aphasiology. London: Bailliere Tindall.
- Lichtheim, K. (1885). On aphasia. *Brain*, **7**, 433-484.
- Lee, G. (1991). *Distributed semantic representation for goalplan analysis of narratives in a connectionist architecture*. Ph.D. Thesis, Computer Science Department, University of California, Los Angeles, CA.
- Lee, G., Flowers, M., and Dyer, M.G. (1990). Learning distributed representations for conceptual knowledge and their application to script-based story processing. *Connection Science*, **2**(4), 313-345.
- Lees, R. and Klima, E. (1963). Rules for English pronomialization. *Language*, **39**, 17-28.
- Lehnert, W.G. (1978). *The Process of Question Answering*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Lieberman, P. and Blumstein, S.E. (1988). *Speech physiology, speech perception, and acoustic phonetics*. Cambridge University Press, Cambridge, GB.
- Liepmann, H. (1914). Bemerkungen zu v. Monakows Kapitel "Die Lokalisation der Apraxie". *Monatsschrift für Psychiatrie und Neurologie*, **35**, 490-516.
- Little, W.A. (1974). The existence of persistent states in the brain. *Mathematical Biosciences*, **19**, 101-120.
- Little, W.A. and Shaw, G.L. (1975). A statistical theory of short and long term memory. *Behavioral Biology*, **14**, 115-133.
- Livingstone, M.S. (1988). Art, illusion and the visual system. *Scientific American*, **258**(1), 78-85.

- Llinas, R. (1982). Radial connectivity in the cerebellar cortex: a novel view regarding the functional organization of the molecular layer. In S.L. Palay and V. Chan-Palay (Eds.), *The Cerebellum, New Vistas*. (pp. 189-192). Springer Verlag, New York.
- Llinas, R. and Sugimori, M. (1980). Electrophysiological properties of in vitro Purkinje cell dendrites in mammalian cerebellar slices. *Journal of Physiology (London)*, **305**, 197-213.
- MacWhinney, B. (1991). *The CHILDES Project: Computational Tools for Analyzing Talk*. Carnegie Mellon University, Pittsburg, PA.
- Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology*, **202**, 437-470.
- Marr, D. (1982). *Vision*. W. H. Freeman and Co., New York.
- Marslen-Wilson, W. (1980). *Optimal efficiency in human speech processing*. Unpublished manuscript, Max-Planck-Institut fur Psycholinguistik, Nijmegen, The Netherlands.
- McClelland, J.L. and Kawamoto, A.H. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In J.L. McClelland and D.E. Rumelhart (Eds.), *Parallel Distributed Processing: Explorations of the Microstructure of Cognition. Volume II: Psychological and Biological Models*. The MIT Press, Cambridge, MA.
- McClelland, J.L., Rumelhart, D.E., and the PDP Research Group, (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Volume I*. The MIT Press/Bradford, Cambridge, MA.
- McCulloch, W.S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin Mathematical Biophysics*, **5**, 115-133.
- McEliece, R.J., Posner, E.C., Rodemich, E.R., and Venkatesh, S.S. (1988). The capacity of the Hopfield associative memory. *IEEE Transactions on Information Theory*.
- Miall, C. (1989). The storage of time intervals using oscillating neurons. *Neural Computation*, **1**(3), 359-371.
- Miikkulainen, R. (1987). *Self-organizing process based on lateral inhibition and weight redistribution*. Technical Report UCLA-AI-87-16, Computer Science Department, University of Calif, Los Angeles, CA.
- Miikkulainen, R. (1990). *DISCERN: A Distributed Artificial Neural Network Model of Script Processing and Memory*. Ph.D. Thesis, Computer Science Department, University of California, Los Angeles, CA.
- Miikkulainen, R. and Dyer, M.G. (1987). *Building distributed representations without microfeatures*. Technical Report UCLA-AI-87-17, Computer Science Department, University of Calif, Los Angeles, CA.
- Miikkulainen, R. and Dyer, M.G. (1988). Forming global representations with extended backpropagation. In *Proceedings of the Second IEEE Annual Conference on Neural Networks, Vol II*. (pp. 17-24).
- Miikkulainen, R. and Dyer, M.G. (1989). Encoding input/output representations in connectionist cognitive systems. In D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kaufmann Publishers, Inc., Los Altos, CA.
- Miikkulainen, R. and Dyer, M.G. (1991). Natural language processing with modular PDP networks and distributed lexicon. *Cognitive Science*, **15**(3), 343-399.

- Millikan, C.H. and Darley, F.L. (Eds.), (1967). *Brain Mechanisms Underlying Speech and Language*. Grune and Stratton, New York.
- Milner, B. (1982). Some cognitive effects of frontal-lobe lesions in man. *Philosophical Transactions of the Royal Society London [Biology]*, **298**, 211-226.
- Milner, B., Petrides, M., and Smith, M.L. (1985). Frontal lobes and the temporal organization of memory. *Human Neurobiology*, **4**, 137-142.
- Milner, B. and Teuber, H.L. (1968). Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of H.M. *Neuropsychologia*, **6**, 213-234.
- Milner, P.M. (1974). A model for visual shape recognition. *Psychological Review*, **81**, 521-535.
- Minsky, M. and Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. The MIT Press (Reissued in an Expanded Edition, 1988), Cambridge, MA.
- Mishkin, M. (1964). Perseveration of central sets after frontal lesions in monkeys. In J.M. Warren and K. Akert (Eds.), *The Frontal Granular Cortex and Behavior*. (pp. 219-241). McGraw-Hill, New York, NY.
- Mishkin, M. (1982). A memory system in the monkey. *Philosophical Transactions of the Royal Society London [Biology]*, **298**, 83-95.
- Mishkin, M., Malamut, B., and Bachevalier, J. (1984). Memories and habits: Two neural systems. In G. Lynch, J.L. McGaugh, and N.M. Weinberger (Eds.), *Neurobiology of Learning and Memory*. (pp. 65-77). Guilford, New York, NY.
- Mishkin, M. and Ungerleider, L.G. (1982). Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural Brain Research*, **6**, 57-77.
- Monsell, S. (1984). Components of working memory underlying verbal skills: A "distributed capacities" view (A tutorial review). In H. Bouma and D. Bouwhuis (Eds.), *International Symposium on Attention and Performance. Vol. 10*. (pp. 327-350). Laurence Erlbaum, Hillsdale, NJ.
- Mooney, R., Shavlik, J., Towell, G., and Gove, A. (1989). An experimental comparison of symbolic and connectionist learning algorithms. In N.S. Sridharan (Ed.), *Proceedings of IJCAI-89 Detroit, MI, August 20-25*. (pp. 775-780). Morgan Kaufmann, San Mateo, CA.
- Mozer, M. (1988). *A focused back-propagation algorithm for temporal pattern recognition*. Technical Report CRG-TR-88-3, Departments of Psychology and Computer Science, University of Toronto.
- Mozer, M.C. (1991). *The Perception of Multiple Objects: A Connectionist Approach*. The MIT Press/Bradford, Cambridge, MA.
- Nakhimovsky, A. (1988). Aspect, aspectual class, and the temporal structure of narrative. *Computational Linguistics*, **14**(2), 29-43.
- Nelson, M.E., Furmanski, W., and Bower, J.M. (1989). Simulating neurons and networks on parallel computers. In C. Koch and I. Segev (Eds.), *Methods in Neuronal Modeling: From Synapses to Networks*. (pp. 397-437). The MIT Press/Bradford, Cambridge, MA.
- Nenov, V.I. (1990). Rapid learning of pattern sequences: A novel network model. In *Proceedings of the International Neural Networks Conference (INNC-90), Paris*.
- Newell, A. (1980). Physical symbol systems. *Cognitive Science*, **4**, 135-183.

- Ninio, A. and Bruner, J. (1977). The achievement and antecedents of labelling. *Journal of Child Language*, **5**, 1-15.
- Ojemann, G.A. (1991). Cortical organization of language. *Journal of Neuroscience*, **11**(8), 2281-2287.
- Park, K. (1988). Sequential Learning. In D. Touretzky, G. Hinton, and T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. (pp. 85-92). Morgan Kaufman, San Mateo, CA.
- Passonneau, R.J. (1988). A computational model of the semantics of tense and aspect. *Computational Linguistics*, **14**(2), 44-60.
- Pearlmutter, B.A. (1988). *Learning state space trajectories in recurrent neural networks: A preliminary report*. Technical Report AIP-54, Computer Science Department, Carnegie Mellon University.
- Penfield, W. (1966). Speech, perception and the cortex. In J.C. Eccles (Ed.), *Brain and Conscious Experience*. (pp. 217-237). Springer, Berlin.
- Penfield, W. and Roberts, L. (1959). *Speech and Brain Mechanisms*. Princeton University Press, Princeton, NJ.
- Peretto, P. and Niez, J.J. (1986). Collective properties of neural networks. In E. Bienenstock and et al. (Eds.), *Disordered Systems and Biological Organization*. Springer Verlag, Berlin.
- Petersen, S.E., Fox, P.T., Posner, M.I., Mintun, M., and Raichle, M.E. (1988). Positron emission tomographic studies of the cortical anatomy of single-word processing. *Nature*, **331**, 585-589.
- Petersen, S.E., Robinson, D.L., and Morris, J.D. (1987). Contributions of the pulvinar to visual spatial attention. *Neuropsychologia*, **25**, 97-105.
- Peterson, G.E. and Lehiste, I. (1960). Duration of syllable nuclei in English. *Journal of the Acoustical Society of America*, **32**, 693-703.
- Phillips, W.A. (1983). Short-term visual memory. *Philosophical Transactions of the Royal Society London [Biology]*, **302**, 295-309.
- Piaget, J. (1951). *Play, Dreams and Imitation*. Routledge and Kegan-Paul, London.
- Piaget, J. (1952). *The Origins of Intelligence in Children*. Nonon, New York, NY.
- Piaget, J. (1954). *The Construction of Reality in the Child*. Ballentine, New York, NY.
- Pineda, F.J. (1987a). Generalization of back propagation to recurrent and higher order neural networks. In *Proceedings of IEEE Conference on Neural Information Processing Systems*. SOS Printing, San Diego, CA.
- Pineda, F.J. (1987b). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, **59**, 2229-2232.
- Pinker, S. (1989). *Learnability and Cognition: The acquisition of Argument Structure*. The MIT Press, Cambridge, MA.
- Pinker, S. and Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, **28**, 73-193.
- Pollack, J. (1986). *Cascaded Back Propagation on Dynamic Connectionist Networks*. Technical Report MCCS-86-67, CRL, New Mexico State University, NM.

- Pollack, J.B. (1990). Recursive distributed representations. *Artificial Intelligence*, **46(1-2)**, 77-105.
- Posner, M.I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, **32**, 3-25.
- Purpura, D.P. (1970). Operations and processes in thalamic and synaptically related neural subsystems. In F.O. Schmitt (Ed.), *The Neurosciences. Second Study Program*. (pp. 458-470). Rockefeller University Press, New York, NY.
- Pöppel, E. (1982). *Lust und Schmerz. Grundlagen Menschlichen Erlebens und Verhaltens*. Severin und Siedler, Berlin.
- Pöppel, E. (1988). *Mindworks: Time and Conscious Experience*. Harcourt Brace Jovanovich, Boston.
- Pöppel, E. (1989). The measurement of music and the cerebral clock: A new theory. *Leonardo*, **22(1)**, 83-89.
- Quillian, M.R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, **12**, 410-430.
- Rafal, R.D. and Posner, M.I. (1987). Deficits in human visual spatial attention following thalamic lesions. *Proceedings of the National Academy of Sciences, USA [Biophysics]*, **84**, 7349-7353.
- Raichenbach, H. (1947). *The Elements of Symbolic Logic*. The Free Press, New York.
- Ratliff, F., Hartline, H.K., and Lange, D. (1966). The dynamics of lateral inhibition in the compound eye of limulus. In *Proceedings of the International Symposium on the Functional Organization of the Compound Eye*.
- Rayner, K. and Pollatsek, A. (1987). Eye movements in reading: A tutorial review. In M. Coltheart (Ed.), *Attention and Performance XII: Reading*. (pp. 327-362). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Read, W. and Nenov, V.I. (1991). *The searchlight of attention revisited: insights from computer simulations*. (UnPubl)
- Reeke, G.N., Sporns, O., and Edelman, G.M. (1989). Synthetic neural modelling: comparisons of population and connectionist approaches. In R. Pfeifer, Z. Schreter, F. Fogelman-Soulie, and L. Steels (Eds.), *Connectionism in Perspective*. (pp. 113-139). Elsevier, Amsterdam, The Netherlands.
- Reeke, G.N.Jr. and Edelman, G.M. (1984). Selective networks and recognition automata. *Annals of the New York Academy of Sciences*, **426**, 181-201.
- Regier, T. (1991). *Learning Perceptually-Grounded Semantics in the L0 Project*. International Computer Science Institute Tech Report, Berkeley, CA.
- Reichardt, W. (1970). The insect eye as a model for analysis of uptake, transduction, and processing of optical data in the nervous system. In Schmitt (Ed.), *The neurosciences. Volume 2*. (pp. 494-511). Rockefeller University Press, New York.
- Rescorla, R.A. and Wagner, A.R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. In A.H. Black and W.F. Prokasy (Eds.), *Classical Conditioning II: Current research and theory*. (pp. 64-99). Appleton-Century-Crofts, New York.

- Ritter, H.J. and Schulten, K.J. (1988). Convergency properties of Kohonen's topology conserving maps: Fluctuations, stability and dimension selection. *Biological Cybernetics*, **60**, 59-71.
- Rosenblatt, F. (1958). The Perceptron, a probabilistic model for information storage and organization in the brain. *Psychological Review*, **62**, 386-408.
- Rosenblatt, F. (1962). *Principles of neurodynamics*. Spartan, New York.
- Ross, E.D. (1981). The aprosodias: Functional-anatomical organization of the affective components of language in the right hemisphere. *Archives of Neurology*, **38(9)**, 561-569.
- Rowher, R. and Forrest, B. (1987). Training time-dependence in neural networks. In M. Caudill and C. Butler (Eds.), *Proceedings of the First International Conference on Neural Networks*. (pp. II:701-708). SOS Printing, San Diego, CA.
- Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition (Volume 1)*. (pp. 318-362). MIT Press, Cambridge, MA.
- Rumelhart, D.E. and McClelland, J.L. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processing. In B. MacWhinney (Ed.), *Mechanisms of Language Acquisition*. Lawrence Erlbaum, Hillsdale, NJ.
- Sabah, N.H. (1971). Reliability of computation in the cerebellum. *Biophys. J.*, **11**, 429-445.
- Schank, R.C. (1972). Conceptual Dependency: A theory of natural language understanding. *Cognitive Psychology*, **3**, 552-631.
- Schank, R.C. (1975). Conceptual information processing. In *Fundamental Studies in Computer Science, Volume 3*. American Elsevier, New York, NY.
- Schank, R.C. (1981). *Inside Computer Understanding: Five Programs Plus Miniatures*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Schank, R.C. and Abelson, R.P. (1977). *Scripts, Plans, Goals, and Understanding - An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Schank, R.C. and Riesbeck, C.K. (Eds.), (1981). *Inside Computer Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Schiller, P.H., True, S.D., and Conway, J.L. (1979). Effects of frontal eye field and superior colliculus ablations on eye movements. *Science*, **206**, 590-592.
- Schwartz, M.F. (1985). Classification of language disorders from a psycholinguistic viewpoint. In J. Oxbury, R. Whurr, M. Coltheart, and M. Wyke (Eds.), *Aphasia*. Butterworth, London, UK.
- Segundo, J.P. and Kohn, A.F. (1981). A model of excitatory synaptic interaction between pacemakers. Its reality, its generality, and the principles involved. *Biological Cybernetics*, **40**, 113-126.
- Segundo, J.P., Moore, G.P., Stensaas, L.J., and Bullock, T.H. (1963). Sensitivity of neurons in Aplysia to temporal pattern of arriving impulses. *Journal of Experimental Biology*, **40**, 643-667.
- Sejnowski, T.J. (1986). Open questions about computation in cerebral cortex. In J.L. McClelland and D.E. Rumelhart (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. (pp. 372-389). The MIT Press/Bradford, Cambridge, MA.

- Sejnowski, T.J., Koch, C., and Churchland, P.S. (1988). Computational Neuroscience. *Science*, **241**, 1299-1306.
- Servan-Schreiber, D., Cleeremans, A., and McClelland, J.L. (1988). *Encoding sequential structure in simple recurrent networks*. Technical report CMU-CS-88-183, Computer Science Department, Carnegie Mellon University.
- Sharkey, N.E., Sutcliffe, R.F.E., and Wobcke, W.R. (1986). Mixing binary and continuous connection schemes for knowledge access. In *Proceedings of the Sixth National Conference on Artificial Intelligence*. Morgan Kaufmann, Los Altos, CA.
- Shastri, L. and Ajjanagadde, V. (1989a). *A connectionist system for rule based reasoning with multiple-place predicates and variables*. Technical Report, MS-CIS-89-06, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA.
- Shastri, L. and Ajjanagadde, V. (1989b). *From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings*. Technical Report MS-CIS-90-05, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA.
- Shimohara, K., Uchiyama, T., and Tokunaga, Y. (1988). Back-propagation networks for event-driven temporal sequence processing. In *Proceedings of the IEEE International Conference on Neural Networks*. (pp. 665-672). San Diego, CA.
- Siskind, J.M. (1990). Acquiring core meanings of words, represented as Jackendoff-style conceptual structures, from correlated streams of linguistic and non-linguistic input. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*. (pp. 143-156).
- Siskind, J.M. (1991a). Naive physics, event perception, lexical semantics and language acquisition. In *AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*. Stanford, CA.
- Siskind, J.M. (1991b). Dispelling myths about language bootstrapping. In *AAAI Spring Symposium Workshop on Machine Learning of Natural Language and Ontology*. Stanford, CA.
- Smolensky, P. (1987). *A method for connectionist variable binding*. Technical Report CU-CS-356-87, Dept. Comp. Sci. and Inst. Cogn. Sci., University of Colorado, Boulder, CO.
- Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences*, **11**, 1-74.
- Sokolov, A.N. (1972). *Inner Speech and Thought*. Plenum Press, New York.
- Sompolinsky, H. and Kanter, I. (1986). Temporal association in asymmetric neural networks. *Physical Review Letters*, **57**, 2861-2864.
- Sopena, J.M. (1988). *Verbal description of visual blocks world using neural networks*. Technical Report UB-DPB-88-10, Departament de Psicologia Basica, Universitat de Barcelona, Spain.
- Squire, L.R. (1986). Mechanisms of memory. *Science*, **232**, 1612-1619.
- Squire, L.R. (1987). *Memory and Brain*. Oxford University Press, New York.
- St.John, M.F. (1990). *The Story Gestalt -- Text Comprehension by Cue-Based Constraint Satisfaction*. Ph.D. thesis, Department of Psychology, Carnegie Mellon University.

- St. John, M.F. and McClelland, J.L. (1989). Applying contextual constraints in sentence comprehension. In D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kaufmann publishers, Inc., Los Altos, CA.
- Steele, G.L. (1984). *Common LISP: The Language*. Digital Press, Burlington, MA.
- Stolcke, A. (1990). *Learning Feature-based Semantics with Simple Recurrent Networks*. TR-90-015, International Computer Science Institute, Berkeley, CA.
- Stornetta, W.S., Hogg, T., and Huberman, B.A. (1987). A dynamical approach to temporal pattern processing. In *Proceedings of the IEEE Conference on Neural Information Processing Systems, Denver, CO*. SOS Printing, San Diego, CA.
- Studdert-Kennedy, M. and (Ed.), (1983). *Psychobiology of Language*. The MIT Press, Cambridge, MA.
- Sumida, R.A. and Dyer, M.G. (1989). Storing and generalizing multiple instances while maintaining knowledge-level parallelism. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., Los Altos, CA.
- Suppes, P., Liang, L., and Bottner, M. (1991). Complexity issues in robotic machine learning of natural language. In L. Lam and V. Naroditsky (Eds.), *Modeling Complex Phenomena*. Springer-Verlag, New York, NY.
- Sutherland, R.J. and Rudy, J.W. (1989). Configural association theory: The role of the hippocampal formation in learning, memory, and amnesia. *Psychobiology*, **17**(2), 129-144.
- Talmy, L. (1983). How languages structure space. In H. Pick and L. Acredolo (Eds.), *Spatial Orientation: Theory, Research, and Application*. Plenum Press, London.
- Tank, D.W. and Hopfield, J.J. (1987). Neural computation by concentrating information in time. In *Proceedings of the 1st IEEE International Conference on Neural Networks*. (pp. IV:455-468). San Diego, CA.
- Thinking Machines Corporation, (1988). **Lisp Reference Manual*. Cambridge, MA.
- Torras, C.I.G. (1986). Neural network model with rhythm-assimilation capacity. *IEEE Syst.Man,Cybernet.*, **16**, 680-693.
- Touretzky, D.S. (1987). Representing conceptual structures in a neural network. In *Proceedings of the IEEE First Annual Conference on Neural Networks*. IEEE,
- Touretzky, D.S. (1989). *Connectionism and compositional semantics*. Technical Report CMU-CS-89-147, Computer Science Department, Carnegie Mellon University.
- Touretzky, D.S. and Hinton, G.E. (1988). A distributed connectionist production system. *Cognitive Science*, **12**(3), 423-466.
- Tulving, E. (1985). Memory and consciousness. *Canadian Psychology*, **26**, 1-12.
- Ungerleider, L.G. and Mishkin, M. (1982). Two cortical visual systems. In D.J. Ingle, M.A. Goodale, and R.J.W. Mansfield (Eds.), *Analysis of visual behavior*. (pp. 549-586). The MIT Press, Cambridge, MA.
- Van Essen, D.C. and Maunsell, J.H.R. (1983). Hierarchical organization and functional systems in the visual cortex. *Trends in Neuroscience*, **4**(September), 370-375.

- van Gelder, T. (1989). *Distributed Representation*. Ph.D. Thesis, Department of Philosophy, University of Pittsburgh.
- Vigotskij, L.S. (1970). *Thinking and Speech*. SPN, Prague.
- von der Malsburg, C. (1981). *The correlation theory of brain function*. Internal Report 81-2, Department of Neurobiology, Max-Planck-Institute for Bioph, Gottingen, West Germany.
- von der Malsburg, C. (1983). how are nervous structures organized? In E. Basar, H. Flohr, H. Haken, and A.J. Mandel (Eds.), *Synergetics of the Brain. Proceedings of the International Symposium on Synergetics, May 1983*. (pp. 238-249). Springer Verlag, Berlin, Heidelberg.
- von der Malsburg, C. (1987). Synaptic plasticity as basis of brain organization. In J.-P. Changeux and M. Konishi (Eds.), *The Neural and Molecular Bases of Learning*. (pp. 411-432). John Wiley & Sons Ltd., London.
- von der Malsburg, C. (1988). Pattern recognition by labeled graph matching. *Neural Networks*, **1**, 141-148.
- von der Malsburg, C. and Schneider, W. (1986). A neural cocktail-party processor. *Biological Cybernetics*, **56**, 29.
- von der Malsburg, C. and Singer, W. (1988). Principles of cortical network organization. In P. Rakic and W. Singer (Eds.), *Neurobiology of Neocortex*. (pp. 69-99). John Wiley & Sons Ltd., London.
- Waibel, A. (1989). Consonant recognition by modular construction of large phonemic time-delay neural networks. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems I*. (pp. 215-223). Morgan Kaufman, San Mateo, CA.
- Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1987). *Phoneme recognition using time-delay neural networks*. ATR Technical Report TR-I-0006, ATR Interpreting Telephony Research Laboratories, Japan.
- Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1988a). Phoneme recognition: Neural Networks vs. Hidden Markov Models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. (pp. 8.S3.3).
- Waibel, A., Sawai, H., and Shikano, K. (1988b). *Modularity and scaling in large phonemic neural networks*. Technical Report TR-I-0034, ATR Interpreting Telephony Research Laboratories, Japan.
- Walter, W., Cooper, R., Aldridge, V., McCallum, W., and Winter, A. (1964). Contingent negative variation: an electric sign of sensori-motor association and expectancy in the human brain. *Nature*, **203**, 380-384.
- Waltz, D.L. and Pollack, J.B. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, **9**, 51-74.
- Warrington, E.K. (1970). *Nature*, **228**, 628.
- Webber, B.L. (1988). Tense as discourse anaphor. *Computational Linguistics*, **14**(2), 61-73.
- Weiskrantz, L. (1982). Comparative aspects of studies of amnesia. *Philosophical Transactions of the Royal Society London [Biology]*, **298**, 97-109.
- Wernicke, C. (1874). *Die Aphasische Symptomencomplex*. Breslau.
- Wernicke, C. (1906). *Grundriss der Psychiatrie*. G. Thieme Verlag, Leipzig.

- Wernicke, C. (1977). The aphasia symptom complex: A psychological study on an anatomical basis. In G.H. Eggert (Ed.), *Wernicke's works on aphasia (1984)*. Mouton, The Hague.
- Wertheimer, M. (1923). *Psychologische Forschung*, 4, 301-350.
- Wiesmeyer, M. and Laird, J. (1990). A computer model of 2D visual attention. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*. (pp. 582-589). Lawrence Erlbaum, Hillsdale, NJ.
- Wilensky, R. (1978). *Understanding Goal-Based Stories*. Ph.D. Thesis, Technical Report I40, Department of Computer Science, Yale University.
- Wilensky, R. (1983). *Planning and Understanding: A Computational Approach to Human Reasoning*. Addison-Wesley, Reading, MA.
- Williams, R.J. and Zipser, D. (1988). *A learning algorithm for continually running fully recurrent neural networks*. Technical Report ICS 8805, University of California at San Diego, Institute of Cognitive Science.
- Willwacher, G. (1982). Storage of a temporal pattern sequence in a network. *Biological Cybernetics*, 43, 115-126.
- Wilson, M.A. and Bower, J.M. (1989). The simulation of large-scale neural networks. In C. Koch and I. Segev (Eds.), *Methods in Neural Modeling*. (pp. 291-333). The MIT Press/Bradford, Cambridge, MA.
- Winograd, T. (1972). *Understanding Natural Language*. Academic Press, New York, NY.
- Winograd, T. (1973). A procedural model of language understanding. In R. Schank and K. Colby (Eds.), *Computer Models of Thought and Language*. W.H. Freeman, New York.
- Zeki, S. (1976). The functional organization of projections from striate to prestriate visual cortex in the rhesus monkey. *Cold Spring Harbor Symposium on Quantitative Biology*, 40, 591-600.
- Zurif, E.B. (1990). Language and the Brain. In D.N. Osherson and H. Lasnik (Eds.), *Language: An Invitation to Cognitive Science, Volume 1*. (pp. 177-198). The MIT Press / Bradford, Cambridge, MA.

APPENDIX A: IMPLEMENTATION DETAILS

A.1 The CM-2 Connection Machine

In recent years powerful new computers with massively parallel processing architectures have made possible the implementation of large scale neural network models and the running of complex and computationally intensive simulations in reasonable time (Nelson et al., 1989). DETE is implemented on one of the most powerful parallel processors available at present -- the CM-2 Connection Machine from Thinking Machines Corporation. The CM-2 is a 64K processor Single-Instruction-Multiple-Data (SIMD) computer (Hillis, 1985). A high-resolution 21" color monitor (known as the "framebuffer") is interfaced to the back-plane of the CM-2. It allows the memory content of multiple fields defined across all processors of the CM-2 (parallel variables, "pvars") to be displayed simultaneously and rapidly refreshed.

A.2 The CM-2 at UCLA

DETE was implemented on the UCLA Connection Machine CM-2. Currently this is only a "quarter" machine. In other words, instead of 64K processors it has only 16K processors. It uses a Sun 4/380 computer as a front end. The CM-2 has proven to be an effective tool for implementing DETE. The configuration used in DETE allowed each dendritic compartment of the KATAMIC sequential associative memory to have its own processor (actually *virtual processor*). The SCAN instruction on the CM-2 enabled an efficient parallel spread of activation from the input to all DCPs at a given DCP level. This configuration allowed the execution of the KATAMIC algorithm to be done simultaneously on all processors. The SEND-WITH-ADD instruction, which is implemented efficiently in the router in the CM-2, allowed us to compute the total incoming activation to any given predicon in parallel.

A.3 The *LISP programming language

The CM-2 Connection Machine can be programmed in three high-level parallel languages: C*, *LISP, and *FORTRAN. The code for DETE was written in *LISP (Version 5.2) -- a parallel programming language which forms a super-set of Common Lisp. The initial development stages were done using a *LISP simulator running under Lucid Common Lisp on Apollo Domain Workstations and Macintosh II. The *LISP language provides a powerful programming interface to the parallel hardware and the interactive environment permitted us to rapidly debug the code.

A.4 Implementational strategy

Natural neural systems function in real time. They are composed of a huge number of vastly interconnected complex neurons, each of which has its own temporal behavior. If we assume that the most important behavior of a neuron (from the view point of its communication with other neurons) is its spiking activity, then to a great extent we can regard each of these elements as an

asynchronously functioning device. Often small time differences in neural processing are of importance. Thus, real neural systems can be regarded as Multiple-Instructions-Multiple-Data (MIMD) types of computing devices (with memory) where each processor (neuron) is semi-independent from other processors. Neurons "issue" their own commands (when to fire) depending on the input data arriving via multiple channels and the current state of the neurons. In an attempt to implement a neurally realistic model of a natural neural system on a SIMD machine, we had to face the problem of how to model a MIMD system on a SIMD machine. In other words, the problem is that digital computers have a central clock that synchronizes the processing of signals throughout the CPU, whereas no such clock has been found in the central nervous system. This problem, of course, can be solved theoretically and the main issue here is the implementation of a reasonable solution.

The operation of our model is sequential in the sense that each processing cycle consists of (1) getting an input, and (2) calculating a state and passing an output. This is repeated until we freeze the system and save its final state for usage as an initial state in further performance testing. In general, each cycle of the network consists of two sub-cycles (Wilson and Bower, 1989):

(1) *Calculate state* sub-cycle: -- during this sub-cycle each neuron is subjected to three processing steps: a) read all inputs; b) calculate the state; c) place the result to an output buffer. This sub-cycle has to be executed separately for all neuronal subsets (selected sets) because the CM-2 is a SIMD machine. The execution order of the subsets should not matter.

(2) *Propagate outputs* sub-cycle: -- during this sub-cycle: a) neurons do any necessary learning (modulations to the state variables $p-ltm$ or $n-ltm$) as a result of a comparison between their current state and the saved history, b) they copy all outputs to the corresponding inputs.

At each time cycle DETE also listens to the visual as well as to the verbal input. If inputs are present (the visual input is present at most all times while the verbal input is occasional), they are incorporated in the operation of the system. If the system initiates any verbal or motor behavior (e.g., describes a visual scene, answers a question, or manipulates objects in its internal or internal image-field) these events are also spread along the appropriate number of cycles.

A.5 Summary of DETE's code and CM-2 usage

Approximately 60 training, testing and analysis programs were developed in the DETE project. These programs contain a total of about 7,500 lines of *LISP code. The code for constructing DETE's architecture, inter-modular communications and control mechanisms constitutes approximately 20%. The performance analysis routines constitute approximately 65%. A control program allowed us to reuse some of the subroutines for different experiments which reduces the total amount of code lines. Routines for running comparative studies (e.g., with Elman's SRN) constitute another 10%. The framebuffer graphics routines take up the additional 5%. The basic code in DETE, which contains the core KATAMIC model (Appendix B.4), was only about 700 lines (including miscellaneous utilities). The amount of code that each procedural module takes up is summarized below:

| <u>Procedural modules (total of 12)</u> | <u>Lines of *LISP code</u> |
|---|----------------------------|
| 1) Word Encoding Mechanism (WEM) | 160 |
| 2) Motor Command Decoder | 40 |

| | |
|---------------------------------------|-----------|
| 3) Selective Attention System | |
| a) Focus of Attention Master (FAM) | 20 |
| b) Input Segmentation Mechanism (ISM) | 50 |
| 4) Visual Feature Extractors | |
| a) Shape Feature Extractor (SFE) | 40 |
| b) siZe Feature Extractor (ZFE) | 30 |
| c) Color Feature Extractor (CFE) | 20 |
| d) Location Feature Extractor (LFE) | 30 |
| e) Motion Feature Extractor (MFE) | 40 |
| 5) Verbal Activity Decoder (VAD) | 60 |
| 6) Motion State Decoder | 40 |
| 7) Blob's Simulator | 350 |
| <hr/> | |
| Total of about | 880 lines |

DETE's current implementation (including all essential and support (dummy) parallel variables (pvars)) uses about 1 million virtual processors (vps). The loaded data structures take up about 7/8 of the usable memory on the CM-2. The distribution of memory among DETE's Neural Network modules (based on the KATAMIC memory) is as follows:

| | |
|------------------------------------|---------------------------------|
| (1) <u>Visual Feature Memories</u> | <u>virtual processors (vps)</u> |
| Shape Feature Memory (SFM) | 16x16x64 = 16,384 |
| siZe Feature Memory (ZFM) | 16x16x64 = 16,384 |
| Color Feature Memory (CFM) | 16x16x64 = 16,384 |
| Location Feature Memory (LFM) | 16x16x64 = 16,384 |
| Motion Feature Memory (MFM) | 16x16x64 = 16,384 |
| (2) <u>Verbal Memory</u> | 64x256 = 16,384 vps |
| (3) <u>Motor Memories</u> | <u>virtual processors (vps)</u> |
| Eye Location Memory (ELM) | 16x16x64 = 16,384 |
| Eye Diameter Memory (EDM) | 16x16x64 = 16,384 |
| Finger Location Memory (FEM) | 16x16x64 = 16,384 |
| Finger Motion Memory (FMM) | 16x16x64 = 16,384 |

A brief description of how DETE's architecture maps onto the CM-2 architecture is given below:

(1) Each of the 5 Visual Feature Memories was mapped onto a 3-D data structure (using a 3-D geometry) with dimensions 16x16 predictrons (the size of the feature map) x 64 (number of dendritic compartments per predictron), or a total of $5 \times 256 \times 64 = 81,920$ vps. For each feature memory there are $16 \times 16 = 256$ recognitrons mapped to vps. Also, each memory contains a STM and an LTM component which results in doubling the number of vps ($2 \times 81,920 = 163,840$ vps). Also, each memory is copied in 8 temporal memory planes ($8 \times 163,840 = 1,310,720$ vps). For

convenience (to save memory and to improve performance), during some of the experiments, only the memory modules necessary for the particular experiment were loaded in the CM-2.

(2) The Verbal Memory was mapped to a 2-D data structure of dimensions 64 predictrons x 256 dendritic compartments.

The two different geometries (a 2-D used for the Verbal Memory and a 3-D used for the Visual and Motor Memories) co-exist in the CM-2.

APPENDIX B: *LISP CODE FOR INDIVIDUAL MODULES

B.1 Visual Feature Extractors

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-  
(in-package 'dete :use '(lisp *lisp))
```

B.1.1 Shape feature extractor

```
(*defvar SFP (!! 0))      ;; Shape Feature Plane  
(*proclaim '(type (pvar (unsigned-byte 1))  
                circle-template square-template triangle-template))  
  
(*defvar circle-template (!! 0))  
(*defvar square-template (!! 0))  
(*defvar triangle-template (!! 0))  
  
(*set circle-template  
      (make-circle!! (!! 32) (!! 32) (!! 64) (!! 0) (!! 100)))  
(*set square-template  
      (make-rectangle!! (!! 0) (!! 0) (!! 64) (!! 64) (!! 0) (!! 100)))  
(*set triangle-template  
      (make-triangle!! (!! 3) (!! 20) (!! 20) (!! 21) (!! 10) (!! 5)  
                      (!! 0) (!! 100)))  
  
(*defun fit-circle!! (object))  
(*defun fit-square!! (object))  
(*defun fit-triangle!! (object))  
  
(defvar object-size 0)  
(defvar circle-diff 0)  
(defvar square-diff 0)  
(defvar triangle-diff 0)  
  
(*defun calc-shape-diff!! (object)  
  (*all  
    (setq object-size (*sum object))  
    (setq circle-size (*sum circle-template))  
    (setq circle-diff (1- (/ object-size circle-size)))  
    (setq square-size (*sum square-template))  
    (setq square-diff (1- (/ object-size square-size)))  
    (setq triangle-size (*sum triangle-template))  
    (setq triangle-diff (1- (/ object-size triangle-size))))))  
  
;;  
;; extract-shape!!  
;;  
;; uses three shape templates (circle, square & triangle)  
;; compares in parallel each individual object to each template by:  
;; (1) Start from maximal template size of each template  
;; adjusts template position and size until a tightly fit object  
;; (2) calculates Shape Difference (SD) measures for each template
```

```

;; (3) maps the object's shape to the SFP using the SDs
;;
(*defun extract-shape!! (object)
  (declare (type (pvar (unsigned-byte 2)) image))
  (*all
  ;; (1) do the shape-fitting
    (fit-circle!! object)
    (fit-square!! object)
    (fit-triangle!! object)
  ;; (2) calculate the shape differences
    (calc-shape-diff!! object)
  ;; (3) do the mapping to the SFP
    (cond ((= triangle-diff 0) (map-to-SFP-rows '(5)))
          ((= square-diff 0) (map-to-SFP-rows '(10)))
          ((= circle-diff 0) (map-to-SFP-rows '(15)))
          ((and (> square-diff triangle-diff)
                (> triangle-diff circle-diff))
           (map-to-SFP-rows '(0 1 2)))
          ((and (> square-diff circle-diff)
                (> circle-diff triangle-diff))
           (map-to-SFP-rows '(3 4)))
          ((and (> circle-diff square-diff)
                (> square-diff triangle-diff))
           (map-to-SFP-rows '(6 7)))
          ((and (> circle-diff triangle-diff)
                (> triangle-diff square-diff))
           (map-to-SFP-rows '(8 9)))
          ((and (> triangle-diff square-diff)
                (> square-diff circle-diff))
           (map-to-SFP-rows '(13 14)))
          ((and (> triangle-diff circle-diff)
                (> circle-diff square-diff))
           (map-to-SFP-rows '(11 12)))))))

```

B.1.2 Size Feature Extractor

```

(*defvar ZFP (!! 0))      ;; size Feature Plane

;;
;; extract-size!!
;;
;; counts all active pixels in the VF containing a single object
;; maps the object-size to the ZFP
;;
(*defun extract-size!! (object)
  (declare (type (pvar (unsigned-byte 2)) object))
  (*all
    (*let ((tmp (!! 0))
          (*when (\=!! object (!! 0)) (*set tmp (!! 1)))
          (setq object-size (*sum tmp))
          (cond ((< object-size (* 1 256)) (map-to-ZFP-row 0))
                ((< object-size (* 2 256)) (map-to-ZFP-row 1))
                ((< object-size (* 3 256)) (map-to-ZFP-row 2))
                ((< object-size (* 4 256)) (map-to-ZFP-row 3))
                ((< object-size (* 5 256)) (map-to-ZFP-row 4))
                ((< object-size (* 6 256)) (map-to-ZFP-row 5))
                ((< object-size (* 7 256)) (map-to-ZFP-row 6))
                ((< object-size (* 8 256)) (map-to-ZFP-row 7))
                ((< object-size (* 9 256)) (map-to-ZFP-row 8))

```

```

      ((< object-size (* 10 256)) (map-to-ZFP-row 9))
      ((< object-size (* 11 256)) (map-to-ZFP-row 10))
      ((< object-size (* 12 256)) (map-to-ZFP-row 11))
      ((< object-size (* 13 256)) (map-to-ZFP-row 12))
      ((< object-size (* 14 256)) (map-to-ZFP-row 13))
      ((< object-size (* 15 256)) (map-to-ZFP-row 14))
      ((< object-size (* 16 256)) (map-to-ZFP-row 15))))))

```

B.1.3 Location Feature Extractor

```

(*defvar LFP (!! 0))      ;; Location Feature Plane
(defvar xc)
(defvar yc)
;;
;;  extract-location!!
;;
;;  calculates the x & y coordinats of the center of mass (CM)
;;  by projecting the object up and left the visual screen
;;  and calculating the center of the distribution of each projection
;;  mappes the absolute location of the object onto the LFP
;;
(*defun extract-location!! (object)
  (declare (type (pvar (unsigned-byte 2)) object))
  (*all
    (*let ((up (!! 0))
           (left (!! 0))
           (center-mass (!! 0)))
      (declare (type (pvar (unsigned-byte 14)) up left center-mass))
      ;;  project the object "up" and then "left"
      (*set up (scan-up!! object))
      (*set left (scan-left!! object))
      ;;  calculate the center-of-mass x & y coordinats
      (setq xc (truncate (/
                          (*sum (*!! up (1+!! (self-address-grid!! (!! 1))))
                          (*sum up))))
      (setq yc (truncate (/
                          (*sum (*!! left (1+!! (self-address-grid!! (!! 1))))
                          (*sum left))))
      (setf (pref-grid center-mass (1- xc) (1- yc)) 1)
      (map-to-LFP center-mass)
      center-mass))

```

B.1.4 Motion Feature Extractor

```

(*defvar MFP (!! 0))      ;; Motion Feature Plane
(*proclaim '(type (pvar (unsigned-byte 2)) *previous-location*))
(*defvar *previous-location* (!! 0))
(defvar dx)
(defvar dy)
;;
;;  extract-motion!!
;;
;;  calculates the CM at each time cycle
;;  uses the value of the previous cycle to calculate dx & dy
;;  returns a pvar with only one field ON (1) displaced
;;  from the center of the grid (7 7) by dx & dy
;;  processor with address x=7 & y=7 represents "no-motion" (stop)
;;  7 is half-size of the 16*16 pixels MFP

```

```

;;
(*defun motion-filter!! (object)
  (declare (type (pvar (unsigned-byte 2)) object))
  (declare fixnum dx dy)
  (*all
    (*let ((current-location (the field-pvar
                              (extract-location!! object)))
          (x-temp (!! 0))
          (y-temp (!! 0))
          (out (!! 0)))
      (declare (type (pvar (unsigned-byte 14))
                    current-location x-temp y-temp out))
      ;; find the coordinats of the prev-loc & calculate dx & dy
      (*when (==!! *previous-location* (!! 1))
        (*set x-temp (self-address-grid!! (!! 1)))
        (*set y-temp (self-address-grid!! (!! 1))))
      (setq dx (*sum x-temp))
      (setq dy (*sum y-temp))
      ;(*set out (+!! current-location *previous-location*))
      ;; shift the previous location to center of grid
      (*set out (pref-grid-relative!!
                current-location
                (!! (the fixnum (- dx half-size)))
                (!! (the fixnum (- dy half-size)))
                :border-pvar (!! 1)))
      ;; memorize the current location
      (*set *previous-location* current-location)
      (map-to-MFP out))))

```

B.2 Processing of visual features

```

(*defvar object *current-image*)
;;
;; process-image!!
;;
;; takes an image and passes it to the feature extractors
;; extracting object's SHAPE, SIZE, COLOR, LOCATION and MOTION
;;
(*defun process-image!! (image)
  (*all
    (declare (type (pvar (unsigned-byte 2)) image))
    (extract-shape!! image))
    (extract-size!! image))
    (extract-color!! image))
    (extract-location!! image))
    (extract-motion!! image)))

```

B.3 Simulator of blob's motions

B.3.1 Generation of objects in the visual screen

```

;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-
(in-package 'dete :use '(lisp *lisp))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; UTILITIES
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

;;
;;   square!!
;;
(*defun square!! (npvar)
  (declare (type (pvar (unsigned-byte 14)) npvar))
  (*all (*!! npvar npvar)))

;;
;;   random-rate!!
;;
;;   generates a boolean pvar with 1-bit-density equal to the rate
;;
(*defun random-rate!! (rate)
  (declare (type (field-pvar *) rate))
  (*all (*let ((tmp (!! 0)))
    (declare (type (field-pvar 1) tmp))
    (*when (<!! (random!! (!! 100)) rate)
      (*set tmp (!! 1)))
    tmp)))

;;
;;   random-rate-per-100 (!! 1)
;;
(*defun rrpm!! (rate)
  (declare (type (field-pvar *) rate))
  (*all (*let ((tmp (!! 0)))
    (declare (type (field-pvar 1) tmp))
    (*when (<!! (random!! (!! 1000)) rate)
      (*set tmp (!! 1)))
    tmp)))

(*proclaim '(type (pvar (unsigned-byte 1)) *current-image*))
(*defvar *current-image* (!! 1))
;;
;;   make-circle!!
;;
(*defun make-circle!! (cx cy radius back-rate for-rate)
  (declare (type (pvar (unsigned-byte 14))
    cx cy radius back-rate for-rate))
  (*all
    (*let ((i (self-address-grid!! (!! 1)))
      (j (self-address-grid!! (!! 1))))
      (declare (type (pvar (unsigned-byte 14)) i j))
      (*set *current-image* (!! 1))
      (*if (<!! (+!! (square!! (-!! i cx))
        (square!! (-!! j cy)))
        (square!! radius))
        (*if (==!! (random-rate!! for-rate) (!! 1))
          (*set *current-image* (!! 1)) t!!)
        (*if (==!! (random-rate!! back-rate) (!! 1))
          (*set *current-image* (!! 1)) t!!))
      (*when (==!! (!! 255) *current-image*)
        (*set *current-image* (!! 1))) ;red
      (*show *current-image*))
    *current-image*))

;(make-circle!! (!! 40) (!! 17) (!! 8) (!! 0) (!! 95)) ;for 64*64 retina

```



```

;;
;;  make-rectangle!!
;;
(*defun make-rectangle!! (x1 y1 x2 y2 back-rate for-rate)
  (declare (type (pvar (unsigned-byte 14))
                x1 y1 x2 y2 back-rate for-rate))
  (*all
    (*let ((i (self-address-grid!! (!! 1)))
           (j (self-address-grid!! (!! 1))))
      (declare (type (pvar (unsigned-byte 14)) i j))
      (*set *current-image* (!! 1))
      (*if (and!! (>!! i x1) (<!! i x2)
            (>!! j y1) (<!! j y2))
          (*if (==!! (random-rate!! for-rate) (!! 1))
              (*set *current-image* (!! 1)) t!!)
          (*if (==!! (random-rate!! back-rate) (!! 1))
              (*set *current-image* (!! 1)) t!!))
      (*when (==!! (!! 255) *current-image*)
              (*set *current-image* (!! 200))) ;blue
      (*show *current-image*))
    *current-image*))

;(make-rectangle!! (!! 4) (!! 4) (!! 10) (!! 12) (!! 0) (!! 100))

;;
;;  make-triangle!!
;;
(*defun make-triangle!! (x1 y1 x2 y2 x3 y3 back-rate for-rate)
  (declare (type (pvar (unsigned-byte 14))
                x1 y1 x2 y2 x3 y3 back-rate for-rate))
  (*all
    (*let ((x (self-address-grid!! (!! 0)))
           (y (self-address-grid!! (!! 1)))
           (k12 (/!! (-!! x1 x2) (-!! y1 y2)))
           (k23 (/!! (-!! x2 x3) (-!! y3 y2)))
           (k31 (/!! (-!! x3 x1) (-!! y1 y3))))
      (declare (type (pvar (unsigned-byte 14)) x y))
      (declare (type (pvar (single-float 32)) k12 k23 k31))
      (*set *current-image* (!! 1))
      (*if (and!! (>!! (-!! x1 x) (*!! k12 (-!! y y1)))
                (>!! (-!! x2 x) (*!! k23 (-!! y y2)))
                (<!! (-!! x3 x) (*!! k31 (-!! y y3))))
          (*if (==!! (random-rate!! for-rate) (!! 1))
              (*set *current-image* (!! 1)) T!!)
          (*if (==!! (random-rate!! back-rate) (!! 1))
              (*set *current-image* (!! 1)) T!!))
      (*when (==!! (!! 255) *current-image*)
              (*set *current-image* (!! 100))) ;green
      (*show *current-image*))
    *current-image*))

;(make-triangle!! (!! 3) (!! 20) (!! 20) (!! 21)
;                  (!! 10) (!! 5) (!! 0) (!! 100))

```

B.3.2 Moving Objects in the Visual Screen

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;  UTILITIES

```

```

;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defvar xi 0)
(defvar x-count 0)
;;
;; flip-x-counter
;;
(defun flip-x-counter ()
  (incf xi)
  (if (oddp xi)
      (progn (defmacro x-direction (count) `(decf ,count))
              (decf x-count) (decf x-count))
      (progn (defmacro x-direction (count) `(incf ,count))
              (incf x-count) (incf x-count))))

(defvar yi 0)
(defvar y-count 0)
;;
;; flip-y-counter
;;
(defun flip-y-counter ()
  (incf yi)
  (if (oddp yi)
      (progn (defmacro y-direction (count) `(decf ,count))
              (decf y-count) (decf y-count))
      (progn (defmacro y-direction (count) `(incf ,count))
              (incf y-count) (incf y-count))))

(defvar touch-top)
(defvar touch-bottom)
(defvar touch-left)
(defvar touch-right)
;;
;; touch-wall-&-bounce!!
;;
;;
(*defun touch-wall-&-bounce!! (image)
  (declare (type (pvar (unsigned-byte 1)) image))
  (*all
   ;;
   ;; touch-wall
   ;;
   (*when (== (self-address-grid!! (! 1)) (! 1))
           (if (= (*sum image) 0)
               (setq touch-top nil)
               (setq touch-top t))
           (format t "touch-top ~d~%" touch-top))
   (*when (== (self-address-grid!! (! 1)) (! 1))
           (1-!! (! (dimension-size 1))))
           (if (= (*sum image) 0)
               (setq touch-bottom nil)
               (setq touch-bottom t))
           (format t "touch-bottom ~d~%" touch-bottom))
   (*when (== (self-address-grid!! (! 1)) (! 1))
           (if (= (*sum image) 0)
               (setq touch-left nil)
               (setq touch-left t))
           (format t "touch-left ~d~%" touch-left))

```

```

(*when (=!! (self-address-grid!! (!! 1))
         (1-!! (!! (dimension-size 0))))
  (if (= (*sum image) 0)
      (setq touch-right nil)
      (setq touch-right t))
      (format t "touch-right ~d~%" touch-right)
  ;;
  ;;   bounce!!
  ;;
  ;;   inverts the value of dx and/or dy when it touches a border
  ;;   to be run before/after each call to MOVE-object!!
  ;;
  (when (or touch-top touch-bottom) (flip-y-counter))
  (when (or touch-left touch-right) (flip-x-counter))
  (x-direction x-count)
  (y-direction y-count))

;(touch-wall-&-bounce!! *current-image*)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   initialize the state
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun init-state ()
  (setq x-count 1)
  (setq y-count 1)
  (setq xi 0)
  (setq yi 0)
  (flip-x-counter)
  (flip-y-counter))

;;
;;   move-triangle!!
;;
;;
(*defun move-triangle!! (dx dy count)
  (declare (type (pvar (unsigned-byte 14)) dx dy count))
  ;; initialize state
  (init-state)
  ;;
  ;;   start the moving loop
  ;;
  (dotimes (i count)
    (format t "~*~----- cycle -----> ~d~%" i)
    (touch-wall-&-bounce!! *current-image*)
    (format t "x-count ~d~%" x-count)
    (format t "y-count ~d~%" y-count)
    (make-triangle!! (+!! (!! 30) (*!! (!! x-count) (!! dx))) ; x1
                     (+!! (!! 100) (*!! (!! y-count) (!! dy))) ; y1
                     (+!! (!! 90) (*!! (!! x-count) (!! dx))) ; x2
                     (+!! (!! 110) (*!! (!! y-count) (!! dy))) ; y2
                     (+!! (!! 60) (*!! (!! x-count) (!! dx))) ; x3
                     (+!! (!! 50) (*!! (!! y-count) (!! dy))) ; y3
                     (!! 0) ; back-rate
                     (!! 100) ; for-rate
    )))

```

```

;to be incorporated in the learning procedure
;(dotimes (i 10) (move-triangle!! i)(pg16 *current-image*))
;(move-triangle!! 3 3 12)

;;
;;  move-circle!!
;;
;;
(*defun move-circle!! (dx dy count)
  (declare (type (pvar (unsigned-byte 14)) dx dy count))
  ;; initialize state
  (init-state)
  ;;
  ;;  start the moving loop
  ;;
  (dotimes (i count)
    (format t "~%*----- cycle -----> ~d~%" i)
    (touch-wall-&-bounce!! *current-image*)
    (format t "x-count ~d~%" x-count)
    (format t "y-count ~d~%" y-count)
    (make-circle!! (+!! (!! 100) (*!! (!! x-count) (!! dx))) ; cx
                   (+!! (!! 50) (*!! (!! y-count) (!! dy))) ; cy
                   (!! 20) ; radius
                   (!! 0) ; foreground
                   (!! 100) ; background
    )))

;;
;;  move-rectangle!!
;;
;;  takes an object by name (rec cir tri) NOOooo
;;  and attaches to it a speed vector (dx dy)
;;  such that at any successive input load the
;;  particular object moves in the given direction
;;
(*defun move-rectangle!! (dx dy count)
  (declare (type (pvar (unsigned-byte 14)) dx dy count))
  ;; initialize state
  (init-state)
  ;;
  ;;  start the moving loop
  ;;
  (dotimes (i count)
    (format t "~%*----- cycle -----> ~. " i)
    (touch-wall-&-bounce!! *current-image*)
    (format t "x-count ~d~%" x-count)
    (format t "y-count ~d~%" y-count)
    (make-rectangle!! (+!! (!! 40) (*!! (!! x-count) (!! dx))) ; x1
                      (+!! (!! 30) (*!! (!! y-count) (!! dy))) ; y1
                      (+!! (!! 80) (*!! (!! x-count) (!! dx))) ; x2
                      (+!! (!! 70) (*!! (!! y-count) (!! dy))) ; y2
                      (!! 0) ; back-rate
                      (!! 100) ; for-rate
    )))

```

B.3.2 Generation of several objects on the Visual Screen

```

;;
;;  generate a combined image containing multiple different objects

```

```

;;
;; (defparameter *list-of-objects* (list c1 r1 t1))

(defun generate-input ()
  (*all
    (*let ((temp (!! 0)))
      (dolist (x *list-of-objects*)
        (*set temp (+!! x temp))))))

:(generate-input)

```

B.4 The KATAMIC sequential associative memory

The KATAMIC sequential associative memory has been tested in various 2-D configurations up to the size of 1024 predictrons with 1024 dendritic compartments per predictron (mapped to 1 million virtual processors). This configuration had also 1024 recognitrons with 8 dendritic compartments each, and 1024 bi-stable switches. Notice that the various pvars (parallel variables) used to implement the different state and dummy variables of the predictrons, recognitrons, and BSS were defined within the same set of 1 million virtual processors (vps) (i.e. several pvars share the memory of each vp.). Notice also that the widths of the input and output bit patterns are the same and are equal to the number of predictrons.

```

;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-
(in-package 'dete :use '(lisp *lisp))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;   KATAMIC.lisp
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(in-package 'katamic :use '(lisp *lisp))
;; recommended CM configuration '(128 256)

(*proclaim '(type single-float-pvar
  stm                ;; Short Term Memory (stm)
  inj-stm            ;; Injected stm
  all-inj-stm
  p-ltm              ;; positive ltm
  n-ltm              ;; negative ltm
  threshold
  match
  miss
  spur
  fit
  Ts                 ;; time constant -- spatial
  Tt                 ;; time constant -- temporal
  Tf                 ;; time constant -- forgetting
  ltm-per-predictron ;; ltm normalization factor (0,1)
  ltm-update-rate    ;; ltm update rate
  stm-update-rate    ;; stm update rate
  show-switch-r      ;; fb color display of recognition
))

(*defvar stm (!! 0.0))
(*defvar inj-stm (!! 0.0))
(*defvar all-inj-stm (!! 0.0))
(*defvar p-ltm (!! 0.0))

```

```

(*defvar n-ltm (!! 0.0))
(*defvar threshold (!! 0.0))
(*defvar match (!! 0.0))
(*defvar miss (!! 0.0))
(*defvar spur (!! 0.0))
(*defvar fit (!! 0.0))
(*defvar Ts (!! 0.0))
(*defvar Tt (!! 0.0))
(*defvar Tf (!! 0.0))
(*defvar ltm-per-predictron (!! 0.0))
(*defvar ltm-update-rate (!! 1.0))
(*defvar stm-update-rate (!! 1.0))
(*defvar show-switch-r (!! 0.0))

(*proclaim '(type (pvar (unsigned-byte 1))
  seeds
  spread-seeds
  output
  !recog-th!
  *pho-gras*
  pho-gra-input
  !lexicon!
  !order!
  fir-part
  sec-part
  third-part
  ord-part
  pho-n-ord
  gra-n-ord
  recall-cue))
(*defvar seeds (!! 0))
(*defvar spread-seeds-in-x (!! 0))
(*defvar output (!! 0))
(*defvar !recog-th! (!! 1) "the recognition threshold")
(*defvar *pho-gras* (!! 0))
(*defvar pho-gra-input (!! 0))
(*defvar !lexicon! (the (field-pvar 1) (random-rate!! (!! 10))))
(*defvar !order! (!! 0))
(*defvar fir-part (!! 0))
(*defvar sec-part (!! 0))
(*defvar third-part (!! 0))
(*defvar ord-part (!! 0))
(*defvar pho-n-ord (!! 0))
(*defvar gra-n-ord (!! 0))
(*defvar recall-cue (!! 0))

(*proclaim '(type (pvar (unsigned-byte 1))
  phoneme
  grapheme))
(*defvar phoneme (!! 0))
(*defvar grapheme (!! 0))

(proclaim '(type integer y-max-pho y-max-gra
  utter-threshold last-lex-position))

(defvar y-max-pho 0)
(defvar y-max-gra 0)
(defvar utter-threshold 0)
(defvar last-lex-position 0)

```

```

(*proclaim '(type boolean-pvar
  segment-x segment-y
  !switch!
  speech-p text-p
  context-p
  first-order-p second-order-p third-order-p
  t-seeds
  basic-katamic-p working-memory-p episodic-memory-p))
(*defvar segment-x nil!!)
(*defvar segment-y nil!!)
(*defvar !switch! nil!! "if = nil, use extinp, if = t, then use intinp")
(*defvar speech-p nil!!)
(*defvar text-p nil!!)
(*defvar context-p nil!!)
(*defvar first-order-p nil!!)
(*defvar second-order-p nil!!)
(*defvar third-order-p nil!!)
(*defvar t-seeds nil!!)
(*defvar basic-katamic-p nil!!)
(*defvar working-memory-p nil!!)
(*defvar episodic-memory-p nil!!)

(proclaim '(type integer
  dx dy sentence-length
  first-order-counter
  second-order-counter
  noise-steps-counter
  max-first
  max-second))
(defvar dx (dimension-size 0))
(defvar dy (dimension-size 1))
(defvar sentence-length 0)
(defvar first-order-counter 0) ;; counts the phoneme order in a word
(defvar second-order-counter 0) ;; counts the word order in a sentence
(defvar noise-steps-counter 0)
(defvar max-first 16) ;; maximum number of letters in a word
(defvar max-second 16) ;; maximum number of words in a sentence

(proclaim '(type integer
  *goal* *match* *miss* *spur* *recog-window* stm-inj-decay))
(defvar *goal* 0)
(defvar *match* 0)
(defvar *miss* 0)
(defvar *spur* 0)
(defvar *recog-window* 0) ;; looks only at the corresponding prediction
(defvar stm-inj-decay 0) ;; efficacy of the inj-stm with time

(proclaim '(type boolean stats-p show-p
  learn-p bk-p wm-p wem-p sto-p so-p))
(defvar stats-p nil)
(defvar show-p nil)
(defvar learn-p t)
(defvar so-p nil)
(defvar sto-p nil)
(defvar bk-p nil)
(defvar wm-p nil)
(defvar wem-p nil)

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;      utilities
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;
;;      rand-seeds
;;
;;      generates random seeds
;;
(*defun rand-seeds (promile)
  (declare (type fixnum promile))
  (*all
    (*set seeds (the (field-pvar 1) (rrpm!! (!! promile))))))

;;
;;      swex
;;
;;      SPREAD THE WAVE si FROM THE SEEDS sb IN :X WITH EXP DECAY tau
;;      (spread_with_exp_in_x)
;;      this is the slowest function. it can be speaded by running it only
;;      once at the beginning to establish all possible linear spreads
;;      of the injected stm and later at each step we need to select out
;;      only the ones that receive a 1 input and mask the rest out
;;
(*defun swex (si sb tau)
  (declare (type (field-pvar 1) si)) ;;
  (declare (type boolean-pvar sb)) ;; all seeds
  (declare (type single-float-pvar tau))
  (*all
    (*let ((tmp (!! 0))
           (tmp1 (!! 0))
           (sum (!! 0))
           (res (!! 0.0)))
      (declare (type (field-pvar 28) tmp tmp1 sum))
      (declare (type single-float-pvar res))
      (*set tmp (scan!! si 'copy!!
                    :direction :forward
                    :segment-pvar sb
                    :include-self nil
                    :dimension :x))
      (*set tmp (scan!! tmp '+!!
                    :direction :forward
                    :segment-pvar nil!!
                    :include-self t
                    :dimension :x))
      (*set tmp1 (scan!! si 'copy!!
                    :direction :backward
                    :segment-pvar sb
                    :include-self nil
                    :dimension :x))
      (*set tmp1 (scan!! tmp1 '+!!
                    :direction :backward
                    :segment-pvar nil!!
                    :include-self t
                    :dimension :x))
      (*set sum (+!! tmp tmp1))
    ))

```



```

(*when (/=!! sum (!! 0))
  (*set res (the single-float-pvar (exp!! (*!! tau sum))))
res))

;;
;;   update-stm
;;
(*defun update-stm (prev inj)
  (declare (type single-float-pvar prev inj))
  (*all
    (*let ((res (!! 0.0)))
      (declare (type single-float-pvar res))
      (*set res
        (/!! prev
          (+!! prev
            (*!! (-!! (!! 1.0) prev)
              (exp!! (-!! (!! 0.0) (*!! stm-update-rate inj)))))))
      res)))

;;
;;   update-ltm
;;
(*defun update-ltm (ltm stm)
  (declare (type single-float-pvar ltm stm))
  (*all
    (*let ((res (!! 0.0)))
      (declare (type single-float-pvar res))
      (*set res
        (/!! ltm
          (+!! ltm
            (*!! (-!! (!! 1.0) ltm)
              (exp!! (-!! (!! 0.0) (*!! ltm-update-rate stm)))))))
      res)))

;;
;;   testing the learning efficacy
;;
(defun update (lt st)
  (let ((res lt))
    (dotimes (i 20)
      (setq res (/ res (+ res (* (- 1 res) (exp (- 0 (* rate st)))))))
      (format t "~d      ~d~%" i res))))

;;(update 0.5 .01)

;;
;;   normalize-ltm
;;
;;   dynamic normalization of ltm
;;
(*defun normalize-ltm (ltm)
  "the sum of the ltm's of any katon is a constant"
  (declare (type single-float-pvar ltm))
  (*all
    (*let ((tmp (!! 0.0)))
      (declare (type single-float-pvar tmp))
      (*set tmp
        (/!!
          (reduce-and-spread!! ltm '!!! 1) ;; :dimension 1

```

```

        (!! (the single-float (float dy))))))
    (*set tmp (*!! ltm (/!! ltm-per-predictron tmp)))
tmp)))

;;
;; dot-product
;;
(*defun dot-prod (pvar1 pvar2)
  (declare (type single-float-pvar pvar1 pvar2))
  (*all
    (*let ((tmp (!! 0.0)))
      (declare (type single-float-pvar tmp))
      (*set tmp
        (/!!
          (reduce-and-spread!! (the single-float-pvar (*!! pvar1 pvar2))
                                '!! 1)
          (!! (the single-float (float dy))))))
      tmp)))

;;
;; calc-fit
;;
;; monitoring of KATAMIC's performance
;;
(*defun calc-fit (sequence step out-stream)
  (declare (type (field-pvar 1) sequence))
  (declare (type fixnum step))
  (*all
    (*let ((tmp-sequence (!! 0)))
      (declare (type (field-pvar 1) tmp-sequence))
      (*set tmp-sequence (spread!! sequence 1 (mod step dy)))
      (*set match (!! 0.0))
      (*set miss (!! 0.0))
      (*set spur (!! 0.0))
      (*when (==! (self-address-grid!! (!! 1)) (!! 0))
        (*when (and!! (==! tmp-sequence (!! 1))
                      (==! output (!! 1)))
          (*set match (!! 1)))
        (*when (and!! (==! tmp-sequence (!! 1))
                      (==! output (!! 0)))
          (*set miss (!! 1)))
        (*when (and!! (==! tmp-sequence (!! 0))
                      (==! output (!! 1)))
          (*set spur (!! 1)))
      (*set fit (+!! (*!! match (!! 0.95)) ;RED
                  (*!! miss (!! 0.7)) ;GREEN
                  (*!! spur (!! 0.3))) ;BLUE
      (*set show-switch-r (if!! !switch!
                          (!! 0.1) ;PURPLE use intinp
                          (!! 0.8) ;YELLOW use extinp
                          ))
      );end-*when
      (*show fit 0 1)
      (*set fit (news!! fit 0 -1))
      (*show show-switch-r 2 1)
      (*set show-switch-r (news!! show-switch-r 0 -1))
      (setq *goal* (/ (*sum tmp-sequence) dy))
      (setq *match* (truncate (*sum match)))
      (setq *miss* (truncate (*sum miss))))
  )

```

```

        (setq *spur*      (truncate (*sum spur)))
;; SPEACH-TEXT output
  (setq pho-letter ".")
  (setq gra-letter ".")
  (decode-row output)
    (setq pho-predicted pho-letter)
    (setq gra-predicted gra-letter)
  (setq pho-input ".")
  (setq gra-input ".")
  (decode-row tmp-sequence)
    (setq pho-input pho-letter)
    (setq gra-input gra-letter)
  (when sto-p
    (format t "~d ~4,2f  ~4,2f  ~d      ~d      ~d      ~d~%"
      step
      (if (= *goal* 0) 1 (float (/ *match* *goal*)))
      (if (= *goal* 0) 0 (float (/ *spur* *goal*)))
      pho-input
      pho-predicted
      gra-input
      gra-predicted))
  (when (or wm-p so-p wem-p)
    (format t "~d ~4,2f  ~4,2f  ~d      ~d~%"
      step
      (if (= *goal* 0) 1 (float (/ *match* *goal*)))
      (if (= *goal* 0) 0 (float (/ *spur* *goal*)))
      pho-input
      pho-predicted))
  (when bk-p
    (format out-stream "~d      ~4,2f  ~4,2f~%"
      step
      (if (= *goal* 0) 1 (float (/ *match* *goal*)))
      (if (= *goal* 0) 0 (float (/ *spur* *goal*))))))

;;
;;   KATAMIC
;;
;;   the main learn/recall routine
;;
(*defun katamic (xt-sequence xt-target times length out-stream)
  (declare (type (field-pvar 1) xt-sequence xt-target))
  (declare (type fixnum times length))
  (*all
    (*let ((tmp-fire (!! 0))
          (tmp-stm (!! 0))
          (fire-seeds nil!!)
          (tmp-sq xt-sequence) ; used only for display purposes
          (tmp-tg xt-target)   ; used to display the target sequence
          (p (!! 0.0))
          (n (!! 0.0)))
      (declare (type (field-pvar 1) tmp-fire tmp-sq))
      ; (declare (type (field-pvar 2) tmp-stm))
      (declare (type boolean-pvar fire-seeds))
      (declare (type single-float-pvar p n))
      ;;
      ;;   The repetition loop
      ;;
      (when stats-p
        (format t "SEQ#   MATCH/g SPUR/g~%"

```

```

(dotimes (j times)
  (*set stm (!! 0.01)) ;; reset stm at beginning of each sequence
  ;;
  ;; the main learn/recall loop
  ;;
  (dotimes (i length)
    ;; 1. Get input (next pattern from the sequence)
    ;; 2. Compose the next input from extinp (xt-sequence) & intinp (predictions)
    (*if !switch! (*set tmp-fire (spread!! output 1 0))
      (*set tmp-fire (spread!! xt-sequence 1 (mod i dy))))
    ;;..... make sequence scroll in y direction
    (*set tmp-sq (news!! tmp-sq 0 -1))
    (*set tmp-tg (news!! tmp-tg 0 -1))
    (*show (float!! tmp-sq) 0 0)
    (*show (float!! tmp-fire) 1 0)
    ;(*show p-ltm 3 1)
    ;(*show n-ltm 4 1)
    ;; 3. Set the tmp-stm to 1.0 at the seeds which are under fire
    (*set tmp-stm (!! 0))
    (*set fire-seeds nil!!)
    (*when (and!! (==!! tmp-fire (!! 1))
      (==!! seeds (!! 1)))
      (*set tmp-stm (!! 1))
      (*set fire-seeds t!!))
    ;; spread the tmp-stm in x
    (*set tmp-stm (reduce-and-spread!! tmp-stm '+!! 0))
    ; (*show (float!! tmp-stm) 2 0)
    ;; 4. Spread tmp-stm in x with exp decay Ts
    ;; (*set inj-stm (the single-float-pvar
      (swex tmp-stm fire-seeds Ts)))
    (*set inj-stm (!! 0))
    (*when (/=! tmp-stm (!! 0)) (*set inj-stm all-inj-stm))
    (*show inj-stm 3 0)
    ;; 5. update the stm
    ;; the amount of injected STM decays in time which allows
    ;; the beginning of the sequence to leave a stronger trace
    (when bk-p
      (*set stm (the single-float-pvar (update-stm stm inj-stm)))
      (when wm-p
        (*set stm (the single-float-pvar (update-stm stm (/!! inj-stm
          (!! (if (= i 0)
            1.0
            (* stm-inj-decay i))))))))
      (*show stm 4 0)
    ;; 6. learn using co-incidence & counter-incidence
    (when learn-p
      (*when (and!! (==!! tmp-fire (!! 1)) (==!! output (!! 0)))
        (*set p-ltm (the single-float-pvar (update-ltm p-ltm stm))))
      (*when (and!! (==!! tmp-fire (!! 0)) (==!! output (!! 1)))
        (*set n-ltm (the single-float-pvar (update-ltm n-ltm stm))))
      )
      (*show p-ltm 3 1)
      (*show n-ltm 4 1)
    ;; 7. Dynamic renormalization of weights
    (*set p-ltm (the single-float-pvar (normalize-ltm p-ltm)))
    (*set n-ltm (the single-float-pvar (normalize-ltm n-ltm)))
    ;; 8. advance the stm one DCp towards the soma and decay it with Tt
    (*set stm (news!! stm 0 -1))
    (*set stm (*!! stm (exp!! Tt)))

```

```

;; 9. predict next step by calculating the dot-product of stm & weights
    (*set p (the single-float-pvar (dot-prod p-ltm stm)))
    (*set n (the single-float-pvar (dot-prod n-ltm stm)))
;; 10. Compare both distances to the threshold & fire
    (*set output (!! 0))
    (*set output (if!! (>=!! p n) (!! 1) (!! 0)))
    (*show (float!! output) 1 1)
;; 11. Calculate match, miss & spur from the predicted & next step
    (if stats-p (calc-fit xt-target (1+ i) out-stream) nil)
;; 12. Recognition-driven switch from extinp to intinp
    (if learn-p
      (recognize-and-switch (news!! xt-sequence 0 (1+ i)) output)
      (if (> i 100) ;20 used for e9
        (*set !switch! t!!)
        (*set !switch! nil!!)))
) ;end-do i
;; make a demarcation line on fit
    (*when (==! (self-address-grid!! (!! 1)) (!! 1))
      (*set fit (!! 0.01))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; Recognition routines
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;
;; sum-LR-neighbours
;;
(*defun sum-LR-neighbours (base n)
  "n is the size of the recognition window --
  2n+1 is the real size measured in predictron numbers"
  (declare (type (field-pvar 1) base))
  (declare (type fixnum n))
  (*all
    (*let ((tmp-L base)
          (tmp-R base)
          (tmp-LR (!! 0)))
      (declare (type (field-pvar 5) tmp-L tmp-R tmp-LR))
      (dotimes (i n)
        (*set tmp-L (news!! tmp-L -1 0))
        (*set tmp-R (news!! tmp-R 1 0))
        (*set tmp-LR (+!! tmp-L tmp-R tmp-LR))
      )
      (*set tmp-LR (+!! base tmp-LR)
        tmp-LR)))
)

;;
;; recognize-and-switch
;;
(*defun recognize-and-switch (extinp intinp)
  (declare (type (field-pvar 1) extinp intinp))
  (*all
    (*let ((tmp (!! 0))
          (tmpsum (!! 0)))
      (declare (type (field-pvar 1) tmp))
      (declare (type (field-pvar 5) tmpsum))
      (*when (==! (self-address-grid!! (!! 1)) (!! 1)) (!! 0))
        (*set tmp (if!! (==! extinp intinp) (!! 1) (!! 0)))
    )
  )

```

```

    (*set tmpsum (the (field-pvar 5)
                    (sum-LR-neighbours tmp *recog-window*)))
    (*set !switch! (if!! (>=!! tmpsum !recog-th!) t!! nil!))
  ) ;end-*when
)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   setup the environment
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(*defun set-fb ()
  (fb-init)
  (da-color)
  (color-scale)
  (zoom 1))

(*defun make-all-inj-stm ()
  (*all
   (*set spread-seeds-in-x (reduce-and-spread!! seeds '+!! 0))
   (*when (/=!! spread-seeds-in-x (!! 0))
    (*set spread-seeds-in-x (!! 1)))
   (*when (=!! seeds (!! 1)) (*set t-seeds t!))
   (*set all-inj-stm (the single-float-pvar (swex seeds t-seeds Ts)))
  ;; to fill in the unintended 0 at the seeds with 1
  (*when t-seeds (*set all-inj-stm (!! 0.9))))))

(*defun all-init ()
  (*all
   (setq dx (dimension-size 0))
   (setq dy (dimension-size 1))
   (setq stats-p t)
   (setq show-p nil)
   (setq learn-p t)
   (setq so-p nil)
   (setq sto-p nil)
   (setq wm-p nil)           ;working memory
   (setq wem-p nil)         ;working-and-episodic memory
   (setq bk-p nil)          ;basic-katamic
   (setq sequence-oscillation-p nil)
   (setq last-lex-position 0)
   (setq *lexicon* nil)
   (setq number-of-sentences 0)
   (*when (=!! (self-address-grid!! (!! 1)) (!! 0))
    (*set segment-x t!))
   (*when (=!! (self-address-grid!! (!! 0)) (!! 0))
    (*set segment-y t!))
   (*set seeds (!! 0))
   (rand-seeds 2);; 2 normally
   (*set *pho-gras* (random-rate!! (!! 10))))))

(*defun run-init ()
  (*all
  ;; KATAMIC parameters
   (*set p-ltm (!! 0.5))
   (*set n-ltm (!! 0.5))
   (*set stm (!! 0.01));;0.001
   (*set threshold (!! 0.0))

```

```

(*set Ts (*!! (!! -0.01) (!! 1)))
(*set Tt (*!! (!! -0.01) (!! 1)))
(*set Tf (!! -0.05))
(*set match (!! 0.0))
(*set miss (!! 0.0))
(*set spur (!! 0.0))
(*set fit (!! 0.0))
(*set ltm-per-predictron (!! 0.5))
(*set ltm-update-rate (!! 0.1))      ;; c=0.1
(*set stm-update-rate (!! 5.0))     ;; b=10.0
(setq stm-inj-decay 0.5)
(setq *recog-window* 0)
(*set show-switch-r (!! 0.0))
;; Order Memory set-up
(*set fir-part (!! 0))
(*set sec-part (!! 0))
(*set third-part (!! 0))
(*set ord-part (!! 0))
(*set pho-n-ord (!! 0))
(*set gra-n-ord (!! 0))
(*set recall-cue (!! 0))
(*set pho-gra-input (!! 0))
(*set speech-p nil!!)
(*set text-p nil!!)
(setq dete-p nil)
;; different experimental setups
(*set basic-katamic-p nil!!)
(*set working-memory-p nil!!)
(*set episodic-memory-p nil!!))

```

```

(*defun set-up ()
  (*all
    (all-init)
    (run-init)
    (make-all-inj-stm)))

```

```

(*defun print-params ()
  (format t "~%

```

Parameters of the KATAMIC memory:

```

=====
Param  description                               value
-----
N      # of predictrons                          ~d
M      # of D-coms per predictron                ~d
Ts     stm decay constant in :x                  ~d
Tt     stm decay constant in :y                  ~d
Tf     ltm forgetting constant                   ~d
Pij(0) p-ltm at time 0                          *min = ~d; *max = ~d
Nij(0) n-ltm at time 0                          *min = ~d; *max = ~d
Wij(0) stm at time 0                             *min = ~d; *max = ~d
ltm-per-predictron                              ~d
stm-inj-decay                                    ~d
stm-update-rate (b)                              ~d
ltm-update-rate (c)                              ~d
*recog-window*                                  ~d
!recog-th! recognition threshold                  ~d
=====

```

```

dx
dy

```

```

(*max Ts)
(*max Tt)
(*max Tf)
(*min p-ltm) (*max p-ltm)
(*min n-ltm) (*max n-ltm)
(*min stm) (*max stm)
(*max ltm-per-predictron)
stm-inj-decay
(*max stm-update-rate)
(*max ltm-update-rate)
*recog-window*
(*max !recog-th!)))

```

B.5 Encoding & decoding of verbal I/O

B.5.1 Word Encoding Mechanism

```

;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-
(in-package 'dete :use '(lisp *lisp))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; word.lisp
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; Word Encoding Mechanism & Sentence Parser
;;
;; takes a sentence and generates a text-pvar (encoding of pho-gras)
;; the position of the sentence representation in the text-pvar
;; is offset from the begining

(proclaim '(type cons *s*))
(defvar *s*)

(proclaim '(type string *curr-w*))
(defvar *curr-w* nil)

(proclaim '(type integer *l*))
(defvar *l* 0)

(proclaim '(type integer
             number-of-letters
             number-of-words
             number-of-sentences))
(defvar number-of-letters)
(defvar number-of-words)
(defvar number-of-sentences 0)

;;
;; PARSE a sentence
;;
(*defun parse (sentence offset)
  (declare (type cons sentence))
  (declare (type integer offset))
  (*all
   (let ((count 0)
         (word-position 0)

```



```

        (word-length 0))
    (declare (type integer count word-position word-length))
(declare (type cons *s*))
(declare (type string *curr-w*))
(declare (type integer *l*))
    (setq *s* sentence)
    (setq *curr-w* nil)
    (setq *l* 0)
    (*set pho-gra-input (!! 0))
;; go through the sentence word by word
    (setq number-of-words (length *s*))
    (dotimes (i number-of-words)
        ;; get a word
        (setq *curr-w* (string (pop sentence)))
        (format t "~%~s~% " *curr-w*)
        (if (check-lexicon *curr-w*)
            (progn (setq word-position (get-position *curr-w*))
                    (setq word-length (get-length (quote *curr-w*))))
            (progn (add-lexicon (list *curr-w* last-lex-position
                                     (length *curr-w*)))
                    (setq last-lex-position (+ last-lex-position
                                                (length *curr-w*))))))
;; go through the word letter by letter
        (setq number-of-letters (length *curr-w*))
        (dotimes (j number-of-letters)
            (setq *l* (- (char-code (char *curr-w* j)) 65))
;; copy the appropriate raw of the *pho-gras*
;; to the proper raw in the pho-gra-input
            (when sto-p
                (*when (== (self-address-grid!! (!! 1))
                           (!! (+ count j offset)))
                    (*when speech-p
                        (*set pho-gra-input (pref!! *pho-gras*
                                                  (grid!! (self-address-grid!! (!! 0)) (!! *l*))))))
                    (*when text-p
                        (*set pho-gra-input (pref!! *pho-gras*
                                                  (grid!! (self-address-grid!! (!! 0)) (!! *l*))))))
                    (*when first-order-p
                        (*set pho-gra-input (pref!! *pho-gras*
                                                  (grid!! (self-address-grid!! (!! 0)) (!! j))))))
                    (*when second-order-p
                        (*set pho-gra-input (pref!! *pho-gras*
                                                  (grid!! (self-address-grid!! (!! 0)) (!! i))))))
                    (*when third-order-p
                        (*set pho-gra-input (pref!! *pho-gras*
                                                  (grid!! (self-address-grid!! (!! 0))
                                                           (!! number-of-sentences))))))
                (when so-p
                    (*when (== (self-address-grid!! (!! 1))
                               (!! (+ count j offset)))
                        (*when speech-p
                            (*set pho-gra-input (pref!! *pho-gras*
                                                      (grid!! (self-address-grid!! (!! 0)) (!! *l*))))))
                        (*when first-order-p
                            (*set pho-gra-input (pref!! *pho-gras*
                                                      (grid!! (self-address-grid!! (!! 0)) (!! j))))))
                        (*when second-order-p
                            (*set pho-gra-input (pref!! *pho-gras*
                                                      (grid!! (self-address-grid!! (!! 0)) (!! i))))))
                    ))
            ))
    ))

```

```

      (*when third-order-p
        (*set pho-gra-input (pref!! *pho-gras*
          (grid!! (self-address-grid!! (!! 0))
            (!! number-of-sentences))))))
    (when (or wm-p wem-p)
      (*when (==!! (self-address-grid!! (!! 1))
        (!! (+ count j offset)))
        (*when speech-p
          (*set pho-gra-input (pref!! *pho-gras*
            (grid!! (self-address-grid!! (!! 0)) (!! *1*))))))
      ) ; end dotimes
    (setq count (+ count (length *curr-w*) 1))
    (setq sentence-length (1- (+ count offset)))
    (format t "The length of this sentence is ~d~%" sentence-length)
    ;; stretch the text pvar 5 times in the y direction
    (*set pho-gra-input (zoom!! pho-gra-input (!! 1) (!! 5)))
    ;; smear every 5th row
    (smear-5th-row pho-gra-input)
    ;; generate the testing pvars
    (when sto-p
      (*when speech-p (*set pho-part pho-gra-input))
      (*when text-p (*set gra-part pho-gra-input))
      (*when first-order-p (*set fir-part pho-gra-input))
      (*when second-order-p (*set sec-part pho-gra-input))
      (*when third-order-p (*set third-part pho-gra-input))
      (*set ord-part (+!! fir-part sec-part third-part))
      (*set pho-n-ord (+!! pho-part ord-part))
      (*set gra-n-ord (+!! gra-part ord-part)))
    (when so-p
      (*when speech-p (*set pho-part pho-gra-input))
      (*when first-order-p (*set fir-part pho-gra-input))
      (*when second-order-p (*set sec-part pho-gra-input))
      (*when third-order-p (*set third-part pho-gra-input))
      (*set ord-part (+!! fir-part sec-part third-part))
      (*set pho-n-ord (+!! pho-part ord-part)))
    (when (or wm-p wem-p)
      (*when speech-p (*set pho-part pho-gra-input)))
    ;; include the order info in the pho-gra-input
    (dotimes (i sentence-length)
      (update-noise-step-counter i)
      (provide-order-input i)
      (incf number-of-sentences)))

;;
;; smear-5th-row
;;
(*defun smear-5th-row (pvar)
  (declare (type (field-pvar 1) pvar))
  (*all
    (*when (==!! (mod!! (self-address-grid!! (!! 1)) (!! 5)) (!! 4))
      (*set pvar (the (field-pvar 1) (random-rate!! (!! 20))))))

;; Example: (parse '(this is a test) 0)

```

B.5.2 Verbal Activity Decoder

```

;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-
(in-package 'dete :use '(lisp *lisp))
;;

```

```

;; Verbal Activity Decoder
;;
;; during decoding, pho/gra-patterns must be converted to pho/gras.
;; This is a problem of pattern matching or clasification.
;; A simple solution is to bit-multiply the pho/gra-OUT pattern
;; with each of the 26 pho/gra-IN patterns (in-parallel).
;; Then sum along the width (64) of the vectors and
;; take the position of the max-sum as an index into the pho/gra set.
;; This procedure results in picking the most similar pho/gra-IN
;; pattern to the pho/gra-OUT.
;; Thresholding of the max-sum allows for the introduction of
;; "silence" between pho/gras and between words.
;;
;; Decode-raw
;;
(*defun decode-raw (raw)
  (declare (type (field-pvar 1) raw))
  (*all
    (*let* ((tmp-raw (spread!! raw 1 0)) ;spread from 0 in y:
            (tmp-phoneme (!! 0))
            (tmp-grapheme (!! 0)))
      (declare (type (field-pvar 1) tmp-raw))
      (declare (type (field-pvar 12) tmp-phoneme tmp-grapheme))
      (*set phoneme (!! 0))
      (*set grapheme (!! 0))
      (*when speech-p (*set phoneme tmp-raw))
      (*when text-p (*set grapheme tmp-raw))
      (*set tmp-phoneme (reduce-and-spread!!
                        (*!! phoneme *pho-gras*) '+!! 0))
      (*set tmp-grapheme (reduce-and-spread!!
                        (*!! grapheme *pho-gras*) '+!! 0))
      ;; find the Y position of the max for the phoneme
      ;; threshold it and convert it to pho-letter
      (setq y-max-pho (compute-Y-of-max tmp-phoneme))
      (setq pho-letter (number-to-letter tmp-phoneme y-max-pho))
      ;; find the Y position of the max for the grapheme
      ;; threshold it and convert it to gra-letter
      (setq y-max-gra (compute-Y-of-max tmp-grapheme))
      (setq gra-letter (number-to-letter tmp-grapheme y-max-gra))))))

;;
;; compute-Y-of-max
;;
(*defun compute-Y-of-max (pvar)
  (declare (type (field-pvar 12) pvar))
  "assumes that there is a single maximum field in the pvar"
  (*all
    (let ((y-max 0))
      (declare (type fixnum y-max))
      (*let ((tmp-max (!! 0)))
        (declare (type (field-pvar *) tmp-max))
        (if (= (*max pvar) 0)
            (*set tmp-max (!! 0))
            (*set tmp-max (truncate!! (/!! pvar (!! (*max pvar))))))
        ;; if there is more than one maximum
        ;; we find the coordinates of the one farthest from the origine
        (setq y-max (*max (*!! tmp-max (self-address-grid!! (!! 1))))))
      (if (> y-max 26) (setq y-max -19)) ;; set it to #\
      y-max))))

```

```
::  
::   number-to-letter  
::  
(*defun number-to-letter (pvar number)  
  (declare (type (field-pvar 12) pvar))  
  (declare (type fixnum number))  
  (*all  
    (let ((letter "."))  
      (declare (type string letter))  
      (when (> (*max pvar) utter-threshold)  
        (setq letter (string (code-char (+ number 65))))))  
    letter)))
```

APPENDIX C: *LISP CODE SELECTED EXPERIMENTS

C.1 KATAMIC experiments

C.1.1 Patterns (sequences) used in the KATAMIC experiments

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-
(in-package 'dete :use '(lisp *lisp))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   katamic-patterns.lisp
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(in-package '*lisp)

;;
;;   refile#
;;
;;   send the output to various files
;;
(defun refile0 (str)
  (setq out0 (open str :direction :output)))

(defun refile1 (str)
  (setq out1 (open str :direction :output)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   generate patterns (sequences) of various % 1-bit-density
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(*defvar 10p00 (random-rate!! (!! 10)))
(*defvar 10p01 (random-rate!! (!! 10)))

(*defvar 20p02 (random-rate!! (!! 20)))
(*defvar 20p03 (random-rate!! (!! 20)))

(*defvar 30p04 (random-rate!! (!! 30)))
(*defvar 30p05 (random-rate!! (!! 30)))

(*defvar 40p06 (random-rate!! (!! 40)))
(*defvar 40p07 (random-rate!! (!! 40)))

(*defvar 50p08 (random-rate!! (!! 50)))
(*defvar 50p09 (random-rate!! (!! 60)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   patterns (sequences) used in the noise experiments
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```

;;
;;   delete 10% of the 1-bits
;;
(*defvar 10p00d1 10p00)
(*when (== 10p00 1) (*set 10p00d1 (random-rate!! (1 90))))

(*defvar 10p00d2 10p00)
(*when (== 10p00 1) (*set 10p00d2 (random-rate!! (1 80))))

(*defvar 10p00d3 10p00)
(*when (== 10p00 1) (*set 10p00d3 (random-rate!! (1 70))))

(*defvar 10p00d4 10p00)
(*when (== 10p00 1) (*set 10p00d4 (random-rate!! (1 60))))

(*defvar 10p00d5 10p00)
(*when (== 10p00 1) (*set 10p00d5 (random-rate!! (1 50))))

(*defvar 10p00d6 10p00)
(*when (== 10p00 1) (*set 10p00d6 (random-rate!! (1 40))))

;;   insert 1% noise
;;
;;   convert 1% of the 0-bits to 1-bits
;;
(*defvar 10p00i1 10p00)
(*when (== 10p00 0) (*set 10p00i1 (random-rate!! (1 1))))

(*defvar 10p00i2 10p00)
(*when (== 10p00 0) (*set 10p00i2 (random-rate!! (1 2))))

(*defvar 10p00i3 10p00)
(*when (== 10p00 0) (*set 10p00i3 (random-rate!! (1 3))))

(*defvar 10p00i4 10p00)
(*when (== 10p00 0) (*set 10p00i4 (random-rate!! (1 4))))

(*defvar 10p00i5 10p00)
(*when (== 10p00 0) (*set 10p00i5 (random-rate!! (1 5))))

```

C.1.2 Examples of KATAMIC experiments

```

;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-
(in-package 'dete :use '(lisp *lisp))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   KATAMIC-experiments.lisp
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(*defun b0 ()
(format t "RUNNING b0 -- the basic experiment:
-----
1 sequence, 10% density, 10 length, 10 repetitions
-----~%~%"
(print-params)
(dotimes (i 10)
  (format t "cycle ~d~%" (1+ i))

```

```

(katamic 10p00 1 10))

(*defun b1 ()
(format t "RUNNING b1 -- 3 sequences experiment:
-----
3 sequences, 10% density, 30 length, 10 cycles
-----~%~%")
(print-params)
(dotimes (i 10)
  (format t "cycle ~d~%" (1+ i))
  (katamic 10p00 1 30)
  (katamic 10p01 1 30)
  (katamic 10p02 1 30))

(*defun b2 ()
(format t "RUNNING b2 -- variable 1-bit-density experiment
-----
9 sequences, 10-90% density, 10 length, 10 repetitions
-----~%~%")
(print-params)
(refile1 "lmc10")
(refile2 "lmc20")
(refile3 "lmc30")
(refile4 "lmc40")
(refile5 "lmc50")
(refile6 "lmc60")
(refile7 "lmc70")
(refile8 "lmc80")
(refile9 "lmc90")
(dotimes (i 10)
  (format t "cycle ~d~%" (1+ i))
  (katamic 10p00 1 10 out1)
  (katamic 20p00 1 10 out2)
  (katamic 30p00 1 10 out3)
  (katamic 40p00 1 10 out4)
  (katamic 50p00 1 10 out5)
  (katamic 60p00 1 10 out6)
  (katamic 70p00 1 10 out7)
  (katamic 80p00 1 10 out8)
  (katamic 90p00 1 10 out9))
(close out1)
(close out2)
(close out3)
(close out4)
(close out5)
(close out6)
(close out7)
(close out8)
(close out9))

(*defun b4 ()
(refile1 "noise20")
(format out1 "RUNNING b4 -- sequences with noise (128 512)
-----
learn 10 sequences of 10% density, 20 length, 20 repetitions
setq learn-p nil
present 10 noisy patterns variation of 10p00
-----~%~%")
(print-params)

```

```

(format out1 "~%learinng the prototype 10p00 ~%")
(dotimes (i 20)
  (katamic 10p00 10p00 1 20 out1)
  (katamic 10p01 10p01 1 20 t)
  (katamic 10p02 10p02 1 20 t)
  (katamic 10p03 10p03 1 20 t)
  (katamic 10p04 10p04 1 20 t)
  (katamic 10p05 10p05 1 20 t)
  (katamic 10p06 10p06 1 20 t)
  (katamic 10p07 10p07 1 20 t)
  (katamic 10p08 10p08 1 20 t)
  (katamic 10p09 10p09 1 20 t))
(format out1 "~%testing with a noisy pattern 10p00d1~%")
(setq learn-p nil)
(katamic 10p00d1 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00d2~%")
(katamic 10p00d2 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00d3~%")
(katamic 10p00d3 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00d4~%")
(katamic 10p00d4 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00d5~%")
(katamic 10p00d5 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00i1~%")
(katamic 10p00i1 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00i2~%")
(katamic 10p00i2 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00i3~%")
(katamic 10p00i3 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00i4~%")
(katamic 10p00i4 10p00 1 20 out1)
(format out1 "~%testing with a noisy pattern 10p00i5~%")
(katamic 10p00i5 10p00 1 20 out1)
(close out1)

```

C.2 Experiments with DETE

C.2.1 Examples of experiments with DETE

For illustrative purposes the code for only two of the number of experiments performed with DETE is given bellow.

```

;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10.-*-
(in-package 'dete :use '(lisp *lisp))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   DETE-experiments.lisp
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;
;;   EXPERIMENT 1 (learning the meanings of words)
;;
(*defun ex1 (times)
  (format t "~%RUNNING EXPERIMENT 1~%")
  (da-init)
  ;; learn word-picture (W-P) pairs
  (format t "~%1. Learning~%")

```



```

dotimes (i times)
  (ex red-ball-wp 1 15)
  (ex blue-square-wp 1 15)
  (ex red-square-wp 5 15)
  (ex blue-ball-wp 5 15)
:: retry learning to test for savings
format t "~%2. retry learning to test for savings~%"
  (ex red-ball-wp 3 15)
  (ex blue-square-wp 3 15)
  (ex red-square-wp 3 15)
  (ex blue-ball-wp 3 15)
:: test verbalization (p->w)
(format t "~%3. test for verbalization P->W~%"
  (ex red-ball-p 5 15)
  (ex blue-square-p 5 15)
  (ex red-square-p 5 15)
  (ex blue-ball-p 5 15)
:: test imagination (w->p)
(format t "~%4. test for imagination W->P~%"
  (ex red-ball-w 5 15)
  (ex blue-square-w 5 15)
  (ex red-square-w 5 15)
  (ex blue-ball-w 5 15))

::
:: EXPERIMENT 2
::
:: Learning to answer questions about the color and shape of object
::
:: DETE looks at an object (which has the 5 visual features -- clmsz
:: the objective of the experiment is to teach DETE to make the
:: correct slot-type selection (e.g. color) and when asked
:: to verbalize the value of the selected slot
::
:: Learning protocol:
:: wI1 -- color-w & vI -- small-red-ball-up-fastN (zcslm)
:: wI2 -- red-w & vI -- continuous (later test for memorizations
:: by stoping vI before starting wI2)
:: 1. Repeat the above until wI2 --> wO (i.e. DETE utters red-w)
:: 2. Repeat while changing the z,s,l,m values in vI (c=red-c)
:: 3. Repeat 1&2 using blue-c/w instead of red-c/w
::
:: The same experimental design can be used
:: to learn about shape, size, etc.
::
(*defun ex2 (times)
  (format t "~%RUNNING EXPERIMENT 2~%"
  (da-init)
  (dotimes (i times)
    (format t "recycle ~d~%" (1+ i))
    (ex (+!! color-red-w--red-ball-p (rz) (rl) (rm)) 1 15)
    (ex (+!! shape-ball-w--red-ball-p (rz) (rl) (rm)) 1 15)
    (ex (+!! color-blue-w--blue-ball-p (rz) (rl) (rm)) 1 15)
    (ex (+!! shape-ball-w--blue-ball-p (rz) (rl) (rm)) 1 15)
    (ex (+!! color-red-w--red-square-p (rz) (rl) (rm)) 1 15)
    (ex (+!! shape-square-w--red-square-p (rz) (rl) (rm)) 1 15)
    (ex (+!! color-blue-w--blue-square-p (rz) (rl) (rm)) 1 15)
    (ex (+!! shape-square-w--blue-square-p (rz) (rl) (rm)) 1 15))
  :: testing

```

```
(format t "testing ~%")
(ex color-w--red-ball-p 1 15)
(ex shape-w--red-ball-p 1 15)
(ex color-w--blue-ball-p 1 15)
(ex shape-w--blue-ball-p 1 15))
```

C.2.2 I/O from experiments with DETE

Excerpts of DETE's output while running EXPERIMENT 1 are given below. The decoded output from the Verbal Memory is shown (as generated letter by letter by the Verbal Activity Decoder -- VAD) in column "LETTERS". The symbol "zzz...*" indicates failure of the VAD to map the verbal output to a single letter. The outputs from the five Visual Feature Memories (post-processed by Winner Take ALL mechanisms) are shown in the columns labeled "COLOR", "SHAPE", "SIZE", "LOC" and "MOTN". The location of the maximal activation pattern in each of the Visual Feature Planes is also decoded and labeled appropriately (e.g., red, ball, fastS = moving fast South, etc.). The symbol "*" which appears in some of the VFM columns indicates failure of the decoder to generate a unique mapping. The symbol "." in the same columns indicate subthreshold response. Only the processing results of selected learning trials (repetitions) are shown.

1. learning

```
#<Structure PVAR 21004F6> "RED BALL" (verbal/visual pair)
```

repetition 1

| SEQ# | GOAL | MISS | SPUR | LETTERS | COLOR | SHAPE | SIZE | LOC | MOTN |
|------------------------------|------|------|------|---------|--------|---------|------|-----|-----------|
| 1 | 2 | 2 | 62 | zzz...* | purpl* | tringl* | z6 | | dr*fastS* |
| 2 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 3 | 3 | 1 | 0 | . | red | ball | . | . | . |
| 4 | 3 | 1 | 1 | .R. | red | ball | . | . | . |
| 5 | 3 | 1 | 1 | .E. | red | ball | . | . | . |
| 6 | 2 | 0 | 1 | .D. | red | ball | . | . | . |
| 7 | 3 | 1 | 0 | . | red | ball | . | . | . |
| 8 | 3 | 1 | 1 | .B. | red | ball | . | . | . |
| ;;; Expanding Dynamic Memory | | | | | | | | | |
| 9 | 3 | 1 | 1 | .A. | red | ball | . | . | . |
| 10 | 3 | 0 | 0 | .L. | red | ball | . | . | . |
| 11 | 2 | 0 | 1 | .L. | red | ball | . | . | . |
| 12 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 13 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 14 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 15 | 2 | 0 | 0 | . | red | ball | . | . | . |

repetition 5

| SEQ# | GOAL | MISS | SPUR | LETTERS | COLOR | SHAPE | SIZE | LOC | MOTN |
|------|------|------|------|---------|-------|-------|------|-----|------|
| 1 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 2 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 3 | 3 | 0 | 0 | .R. | red | ball | . | . | . |
| 4 | 3 | 0 | 0 | .E. | red | ball | . | . | . |
| 5 | 3 | 0 | 0 | .D. | red | ball | . | . | . |
| 6 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 7 | 3 | 0 | 0 | .B. | red | ball | . | . | . |
| 8 | 3 | 0 | 0 | .A. | red | ball | . | . | . |
| 9 | 3 | 0 | 0 | .L. | red | ball | . | . | . |
| 10 | 3 | 0 | 0 | .L. | red | ball | . | . | . |
| 11 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 12 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 13 | 2 | 0 | 0 | . | red | ball | . | . | . |

| | | | | | | | | | |
|----|---|---|---|---|-----|------|---|---|---|
| 14 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 15 | 2 | 0 | 0 | . | red | ball | . | . | . |

*<Structure PVAR 24ECF5E> "BLUE SQUARE" (verbal/visual pair)

repetition 1

| SEQ# | GOAL | MISS | SPUR | LETTERS | COLOR | SHAPE | SIZE | LOC | MOTN |
|------|------|------|------|---------|--------|---------|------|-----|--------|
| 1 | 2 | 0 | 59 | zzz...* | purpl* | tringl* | z6 | dr* | fastS* |
| 2 | 2 | 0 | 0 | . | blue | square | . | . | . |
| 3 | 3 | 1 | 1 | .L. | blue | square | . | . | . |
| 4 | 3 | 1 | 1 | .B. | blue | square | . | . | . |
| 5 | 3 | 1 | 1 | .L. | blue | square | . | . | . |
| 6 | 3 | 1 | 1 | .U. | blue | square | . | . | . |
| 7 | 2 | 0 | 2 | .D.E. | blue | square | . | . | . |
| 8 | 3 | 1 | 0 | . | blue | square | . | . | . |
| 9 | 3 | 1 | 2 | .B.S. | blue | square | . | . | . |
| 10 | 3 | 1 | 2 | .A.Q. | blue | square | . | . | . |
| 11 | 3 | 1 | 1 | .U. | blue | square | . | . | . |
| 12 | 3 | 1 | 1 | .A. | blue | square | . | . | . |
| 13 | 3 | 0 | 1 | .E.R. | blue | square | . | . | . |
| 14 | 2 | 2 | 2 | . | red | ball | . | . | . |
| 15 | 2 | 0 | 0 | . | blue | square | . | . | . |

repetition 5

| SEQ# | GOAL | MISS | SPUR | LETTERS | COLOR | SHAPE | SIZE | LOC | MOTN |
|------|------|------|------|---------|-------|--------|------|-----|------|
| 1 | 2 | 0 | 1 | .E. | blue | square | . | . | . |
| 2 | 2 | 0 | 1 | .E. | blue | square | . | . | . |
| 3 | 3 | 0 | 1 | .B.E. | blue | square | . | . | . |
| 4 | 3 | 0 | 1 | .E.L. | blue | square | . | . | . |
| 5 | 3 | 0 | 1 | .E.U. | blue | square | . | . | . |
| 6 | 3 | 0 | 0 | .E. | blue | square | . | . | . |
| 7 | 2 | 0 | 0 | . | blue | square | . | . | . |
| 8 | 3 | 0 | 0 | .S. | blue | square | . | . | . |
| 9 | 3 | 0 | 0 | .Q. | blue | square | . | . | . |
| 10 | 3 | 0 | 0 | .U. | blue | square | . | . | . |
| 11 | 3 | 0 | 0 | .A. | blue | square | . | . | . |
| 12 | 3 | 0 | 0 | .R. | blue | square | . | . | . |
| 13 | 3 | 0 | 0 | .E. | blue | square | . | . | . |
| 14 | 2 | 0 | 0 | . | blue | square | . | . | . |
| 15 | 2 | 0 | 0 | . | blue | square | . | . | . |

3. test for verbalization P->W

*<Structure PVAR 24ED09E> "RED BALL" (visual input only)

repetition 1

| SEQ# | GOAL | MISS | SPUR | LETTERS | COLOR | SHAPE | SIZE | LOC | MOTN |
|------|------|------|------|---------|-------|-------|------|-----|------|
| 1 | 2 | 0 | 1 | .L. | red | ball | . | . | . |
| 2 | 2 | 0 | 1 | .L. | red | ball | . | . | . |
| 3 | 2 | 0 | 1 | .R. | red | ball | . | . | . |
| 4 | 2 | 0 | 2 | .E.L. | red | ball | . | . | . |
| 5 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 6 | 2 | 0 | 1 | .A. | red | ball | . | . | . |
| 7 | 2 | 0 | 2 | .B.L. | red | ball | . | . | . |
| 8 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 9 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 10 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 11 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 12 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 13 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 14 | 2 | 0 | 0 | . | red | ball | . | . | . |
| 15 | 2 | 0 | 0 | . | red | ball | . | . | . |

#<Structure PVAR 24ECDCE> "BLUE BALL" (visual input only)
 repetition 1

| SEQ# | GOAL | MISS | SPUR | LETTERS | COLOR | SHAPE | SIZE | LOC | MOTN |
|------|------|------|------|---------|-------|-------|------|-----|------|
| 1 | 2 | 0 | 2 | .B.L. | blue | ball | . | . | . |
| 2 | 2 | 0 | 2 | .B.L. | blue | ball | . | . | . |
| 3 | 2 | 0 | 2 | .B.L. | blue | ball | . | . | . |
| 4 | 2 | 0 | 1 | .L. | blue | ball | . | . | . |
| 5 | 2 | 0 | 0 | . | blue | ball | . | . | . |
| 6 | 2 | 0 | 1 | .E. | blue | ball | . | . | . |
| 7 | 2 | 0 | 0 | . | blue | ball | . | . | . |
| 8 | 2 | 0 | 1 | .B. | blue | ball | . | . | . |
| 9 | 2 | 0 | 0 | . | blue | ball | . | . | . |
| 10 | 2 | 0 | 0 | . | blue | ball | . | . | . |
| 11 | 2 | 0 | 1 | .L. | blue | ball | . | . | . |
| 12 | 2 | 0 | 0 | . | blue | ball | . | . | . |
| 13 | 2 | 0 | 0 | . | blue | ball | . | . | . |
| 14 | 2 | 0 | 0 | . | blue | ball | . | . | . |
| 15 | 2 | 0 | 0 | . | blue | ball | . | . | . |

4. test for imagination W->P

#<Structure PVAR 24ED0EE> "BLUE SQUARE" (verbal input only)
 repetition 3

| SEQ# | GOAL | MISS | SPUR | LETTERS | COLOR | SHAPE | SIZE | LOC | MOTN |
|------|------|------|------|---------|-------|--------|--------|-----|-------|
| 1 | 0 | 0 | 24 | zzz...* | green | square | medium | dr* | fastE |
| 2 | 0 | 0 | 25 | zzz...* | green | square | z6 | dr* | fastW |
| 3 | 1 | 0 | 14 | zzz...* | blue | square | z6 | . | slowS |
| 4 | 1 | 0 | 14 | zzz...* | blue | square | . | dc | fastW |
| 5 | 1 | 0 | 16 | zzz...* | . | square | . | dc | slowS |
| 6 | 1 | 0 | 13 | zzz...* | . | square | . | dc | fastW |
| 7 | 0 | 0 | 0 | . | . | . | . | . | . |
| 8 | 1 | 0 | 1 | .B.S. | . | . | . | . | . |
| 9 | 1 | 0 | 0 | .Q. | . | . | . | . | . |
| 10 | 1 | 0 | 0 | .U. | . | . | . | . | . |
| 11 | 1 | 0 | 0 | .A. | . | . | . | . | . |
| 12 | 1 | 0 | 0 | .R. | . | . | . | . | . |
| 13 | 1 | 0 | 0 | .E. | . | . | . | . | . |
| 14 | 0 | 0 | 0 | . | . | . | . | . | . |
| 15 | 0 | 0 | 0 | . | . | . | . | . | . |

APPENDIX D: MONITORING THE NETWORK'S BEHAVIOR

DETE is a complex system and therefore, for its development, debugging, and testing it is quite desirable to have means for monitoring its behavior at various levels. The state of each sub-network of neurons or weight matrix at any time cycle can be monitored graphically through the frame buffer which is interfaced in parallel to the CM-2 Connection Machine. A flexible graphics display was designed which allows us swapping of full screen displays of particular internal states: visual input or outputs, or displaying them all at once. It is still an open question which states of the system are appropriate to be monitored and how should the visual output be arranged.

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: (dete); Base: 10. -*-
(in-package 'dete :use '(lisp *lisp))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;   framebuffer-utils.lisp
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(proclaim '(special *fb*))

(defun fb-init ()
  "A simple initialization procedure."
  (cmfb::initialize-display
   (setq *fb* (cmfb::attach-display)))
  (cmfb::set-zoom *fb* 0 0))

(defvar *color-scale-x* 400 "X origin of color scale")
(defvar *color-scale-y* 0 "Y origin of color scale")

;;
;;   color-scale
;;
;;   draws a color scale on the framebuffer
;;
(defun color-scale ()
  "Draws a color scale on the framebuffer."
  (let ((colors 256)
        (width 64)
        (buffer (cmfb::current-buffer *fb*))
        (x-origin 1216)
        )
    (when (not (zerop (mod width 32))) (error "lose"))
    (dotimes (color colors)
      (cmfb::write-rectangle-constant
       *fb*
       x-origin
       (- 1024 (* 4 color))
       width 4 buffer
       color))))

;;
```

```

;;      *show
;;
;;      shows pvars with values between 0.0 and 1.0
;;;
(*defun *show (image x-offset y-offset)
  (declare (type (pvar (defined-float * *)) image))
  (declare (type fixnum x-offset y-offset))
  "transforms a pvar to (0,255) and shows it with x,y offset"
  (*let ((8-bit-pvar (coerce!! (round!! (*!! image (!! 255.0)))
                              '(field-pvar 8))))
    (declare (type (field-pvar 8) 8-bit-pvar))
    (if show-p
        (cmfb:write-always *fb* :green
                           (pvar-location 8-bit-pvar)
                           (* (dimension-size 0) x-offset)
                           (* (dimension-size 1) y-offset)) nil)))

;;
;;      zoom
;;
;;      a simple version of CMFB::SET-ZOOM.
;;      (zoom 4) enlarges the display by a factor of 4 in each direction.
;;
(*defun zoom (n)
  "Simple zoom function."
  (declare (type fixnum n))
  (cmfb::set-zoom *fb* (1- n)))

;;
;;      pan
;;
;;      a simple version of CMFB::SET-PAN
;;
(*defun pan (x y)
  (declare (type fixnum x y))
  (cmfb::set-pan *fb* (* (dimension-size 0) x) (* (dimension-size 1) y))
)

```

APPENDIX E: NEURAL NETS FOR PROCEDURAL MODULES

E.1 A neural oscillator

A simple neural oscillator composed of an excitatory and an inhibitory elements connected in a feedback loop is shown in (Figure E.1).

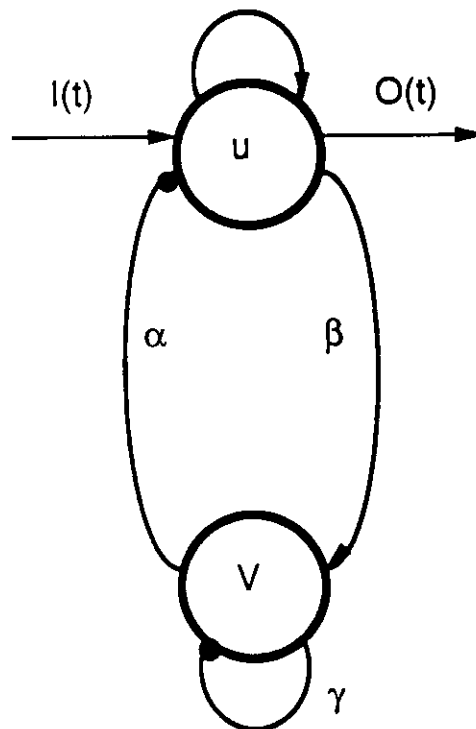


Figure E.1: Neural oscillator

Schematic drawing of a neural oscillator. Arrow represent excitatory synapses and small black circles -- inhibitory synapses.

The activity of the excitatory element u and the activity of the inhibitory element v evolve in time according to the kinetic equations (1) and (2) which can be derived as a meanfield approximation of a network of threshold neurons with stochastic dynamics (Buhmann, 1989).

$$\frac{du}{dt} = -u + \sigma_{\lambda}(u - \beta v - \Theta_u + I) \quad (\text{E.1})$$

$$\tau \frac{dv}{dt} = -v + \sigma_{\lambda}(\alpha u - \gamma v - \Theta_v) \quad (\text{E.2})$$

where τ defines the time scale of the inhibitory element relative to the excitatory element. $\sigma_\lambda(x)$ is a sigmoid gain function with gain parameter $1/\lambda$ (3).

$$\sigma_\lambda(x) = \frac{1}{1 + e^{-x/\lambda}} \quad (\text{E.3})$$

The two elements have different firing thresholds Θ_u and Θ_v . The synaptic strengths of the inhibitory feedback loop responsible for the generation of the oscillatory behavior are specified by the parameters α (excitation of the inhibitory element v), β (strength of the feedback inhibition to element u) and γ (defines the self-inhibition of v). The self-excitation of u is set to 1 to normalize the scale for synaptic weights. The neural oscillator receives an excitatory input $I(t)$ from the retina and the verbal memory (Figure 7.1) and produces an oscillatory output $O(t)$ which is set diffusely to all memory neurons. The behavior of such an oscillator was studied analytically (Buhmann and von der Malsburg, 1990). Buhmann has shown that under certain conditions which restrict the parameter range of the different parameters (e.g., $\alpha=1$, $\beta=2.5$, $\gamma=0.5$, $\Theta_u=-0.275$, $\Theta_v=0.15$, $\lambda=0.025$), the oscillator exhibits a limit cycle behavior.

E.2 XOR network

The XOR function of 2 arguments can be computed by a simple neurally realistic network that contains a simple summation unit -- the DCr (Figure E.2). This unit receives two inputs via excitatory synapses. Two axonal collaterals (one from each input) make reciprocal inhibitory axo-axonal synapses. The function of these synapses is to exert presynaptic inhibition. This kind of inhibition is ubiquitous throughout the nervous system. Its basic function is to block the transmission of the input signal through the synapse that is modulated.

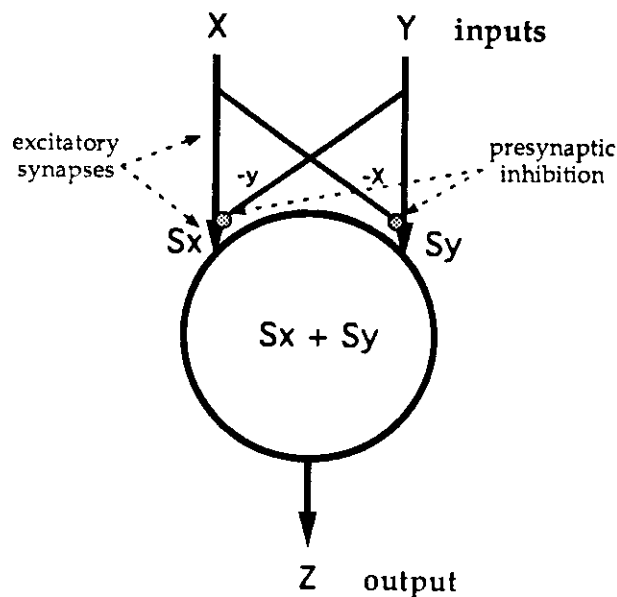


Figure E.2: XOR network

The DC^r is represented by a circle. The external input X and the input coming via the predictron's axon Y are passed to the DC^r via non-modifiable but modulated synapses of value 1 (the thick arrows). Two collaterals (one from each axon) make modulatory inhibitory synapses (the small gray circles) to the presynaptic sites of the excitatory inputs. The circuit computes the XOR of the inputs and passes it to the output Z.

The details of the computations performed by this circuit for each of the four possible input patterns are summarized in Table E.1.

| X | Y | S _x | S _y | S _x +S _y =XOR(X,Y) |
|---|---|----------------|----------------|--|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |

Table E.1: Computation of XOR

