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

# METHODOLOGY FOR CONSTRUCTING OPTIMAL MULTI-LAYERED NEURAL NETWORKS

M. Kayama

September 1991
CSD-910068

# Methodology for Constructing Optimal Multi-layered Neural Networks [1]

Masahiro Kayama

Hitachi Research Laboratory, Hitachi Ltd.

## ABSTRACT

The determination of the optimal number of hidden units, which gives the minimum structure and the maximum generalization ability to a multi-layered neural network, is discussed.

First the generalization ability of the network, trained by the back propagation algorithm (static back propagation, SBP), is simulated with various numbers of hidden units. With the aid of these results, the relationship between the generalization ability of the network and the number of hidden units is investigated and a simple representative model is established.

Then this relationship is carefully simulated again using networks trained by the back propagation algorithm, where suitable random numbers are added to original training data (dynamic back propagation, DBP). This suggests that the generalization ability of a network is determined by superposing two kinds of capability of the network, i.e., the essential capability which depends on just the number of hidden units and the sleeping capability, included in the essential capability, which does not contribute to mapping. By comparing the simulation results obtained by two different training methods, a more precise model of the relationship between the generalization ability and the number of hidden units of the network is developed.

Finally the determination of the optimal number of hidden units is investigated and a suitable algorithm is proposed. This algorithm is shown to estimate the optimal number of hidden units.

Key words : neural network, multi-layered model, hidden units, generalization ability, optimization

---

[1] This research was conducted based on the ideas come upon in Hitachi Research Laboratory

## 1. Introduction

One of the major problems of multi-layered neural networks is that an optimal construction of the network is unknown. If network size is too small, training does not converge because of insufficient network capacity. If it is too large, computational intensivity has to be increased, which causes either large scale or slow response of the network. This demerit is especially serious when the neural net algorithm is installed in certain machines or software products, because they become either large and expensive, or of low quality.

Accordingly, from a practical viewpoint, determining the optimal network structure is very important, however, unfortunately there are few algorithms for it. Thus the optimal network has often been determined by repeating simulation with various structures of networks, which is quite time consuming. Among several tasks included in designing neural network structures, such as numbers of layers and units of each layer, determining the optimal number of hidden units seems to be the hardest and the most interesting problem. So far, few methods have been reported for it.

Xue et al. [1] proposed a method using the rank of a weight matrix between input and hidden units. According to their paper, the rank calculated by a singular decomposition method (SVD) gives an optimal number of hidden units. But this method is effective only when the number of hidden units is smaller than that of the input units, that is, when input information is condensed at hidden units. For example, the method cannot be applied to a one input, one output network which approximates a non-linear function, because the rank is always 1 in this case. Ash [2] reported the dynamic creation of hidden units. In his method, starting from a small number of hidden units, other hidden units are dynamically created during learning until the network error converges to a desired value. But this process also takes much calculation time, particularly when the optimal network consists of many hidden units. Therefore a more general and deterministic algorithm is needed, which easily gives the optimal number of hidden units.

The author et al. also proposed how to determine the optimal number of hidden units by a linear regression analysis [3]. In our method, by using a trained network, the outputs of hidden units corresponding to the training inputs were analyzed statistically and separated into linear and non-linear components. The number of hidden units corresponding to the sum of non-

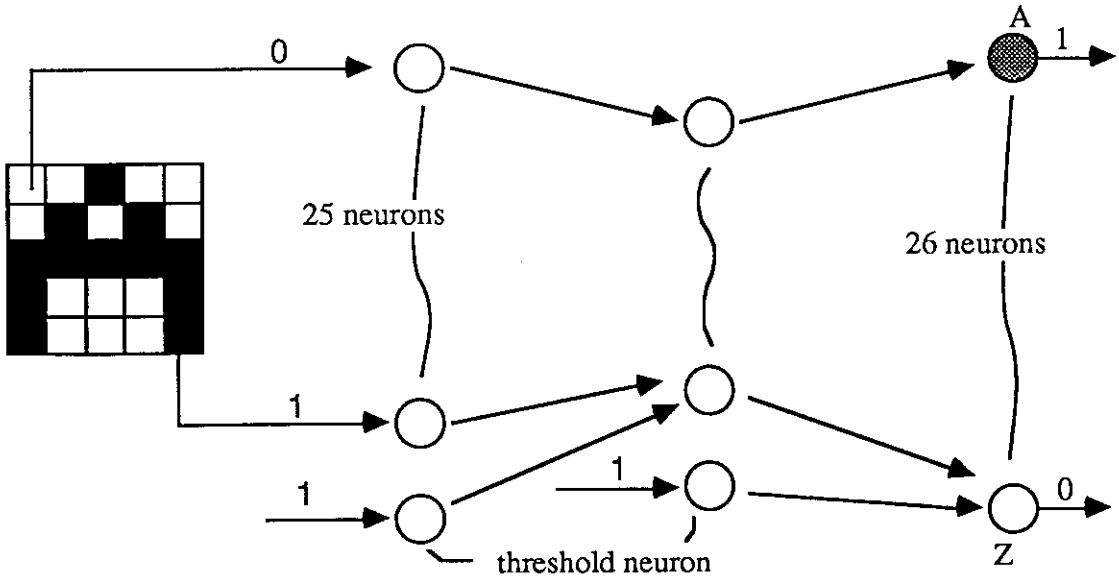
linear components was recognized as an optimal one. This method is applicable to any multi-layered neural network. Additionally, the number of hidden units can be deterministically obtained by just once of training. But, as mentioned in the following sections in detail, the number of hidden units obtained by this method gives the minimum number of hidden units for convergence of training. The generalization ability of the network with this number of hidden units is somewhat lower than the maximum ability. Therefore this result is not sufficient.

In this report, a methodology for constructing multi-layered neural networks with the minimum number of hidden units, which gives the maximum generalization ability is discussed. First the relationship between the number of hidden units and the generalization ability is carefully simulated with networks trained by the back propagation algorithm[4] (static back propagation, SBP), and a simple model describing the relation is established. Then improvements of the generalization ability by the back propagation algorithm, where suitable random numbers are added to the original training inputs[5] (dynamic back propagation, DBP), is also simulated to evaluate two effects on the generalization ability, i.e., the effect of the number of hidden units and the training method, independently. With these results, the developed model can be modified into a more accurate one. Finally an algorithm for determining the optimal number of hidden units is presented.

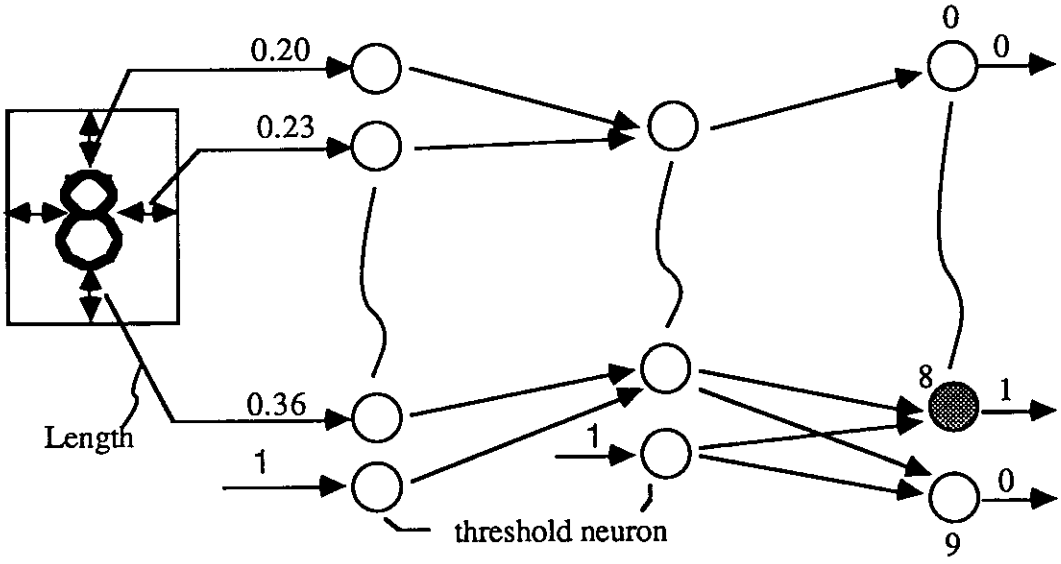
## 2. Networks for experiment

It has been proven that any non-linear function can be approximated by a one-hidden layered neural network[6][7]. Moreover, the method developed for a one-hidden layered network can be easily extended to a multi-hidden layered network. Therefore, in this paper, the investigations are limited to one-hidden layered networks.

Two feedforward neural networks are used in the following simulations as shown in Fig.1. The network of Fig.1(a) for character recognition is mainly used for simulation, and consists of 25 input units, receiving a value of each pixel (1 or 0), several hidden units, and 26 output units corresponding to clusters from A to Z. Fig.1(b) is the network for number recognition used in our previous research[3] at Hitachi Research Laboratory, whose results are compared with those of the character recognition network. The network has 12 inputs which receive each feature value obtained from a number image




(a) Neural Network for Character Recognition




(b) Neural Network for Number Recognition

Fig.1 Neural Network for Simulation

such as a length between each boundary and the nearest pixel from it, several hidden units, and 10 output units corresponding to 10 clusters from 0 to 9. In these two networks, the input and hidden layers have a bias unit.

## 3. Relationship between generalization ability and numbers of hidden units

First let's discuss the relationship between the generalization ability and the number of hidden units, using a network of Fig.1 (a) trained by the SBP. The database for training consists of 26 data sets (A ~ Z), while data for evaluating generalization ability are created by reversing several pixels of original training data as shown in Fig.2. The recognition rate (the generalization ability) were evaluated by using 50 testing data for each character, where reversed pixels were determined randomly, totally 1300 data for each network and Hamming distance. The convergence accuracy of training is within 1% error of full scale against each training output (desired output). That is, training is completed when the following equation is satisfied.

$$\sum_{i=1}^{n} (t_i - o_i)^2 < (1/2)*(0.01)^2*n \qquad (1)$$

$t_i$ : desired output of $i$-th output unit

$o_i$ : network output of $i$-th output unit
n : number of output units

Three networks with the same number of hidden units are used in the evaluation. The initial weight values of them are different.

Fig.3 shows the recognition rate plotted against the number of hidden units. The Hamming distance is the space distance between the testing data and the original training data given by the number of reversed pixels. In the figure, we observed two different regions. The recognition rate increases with the number of hidden units when the number of hidden units is small (region 1). Then it becomes almost saturated and increases only slightly with the increase of the number of hidden units (region 2). Though this behavior is independent of Hamming distances of the testing data, the number of hidden

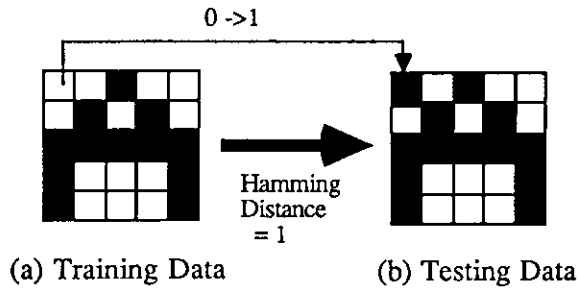



Fig.2 Testing Data Created from Training Data used in the Simulations

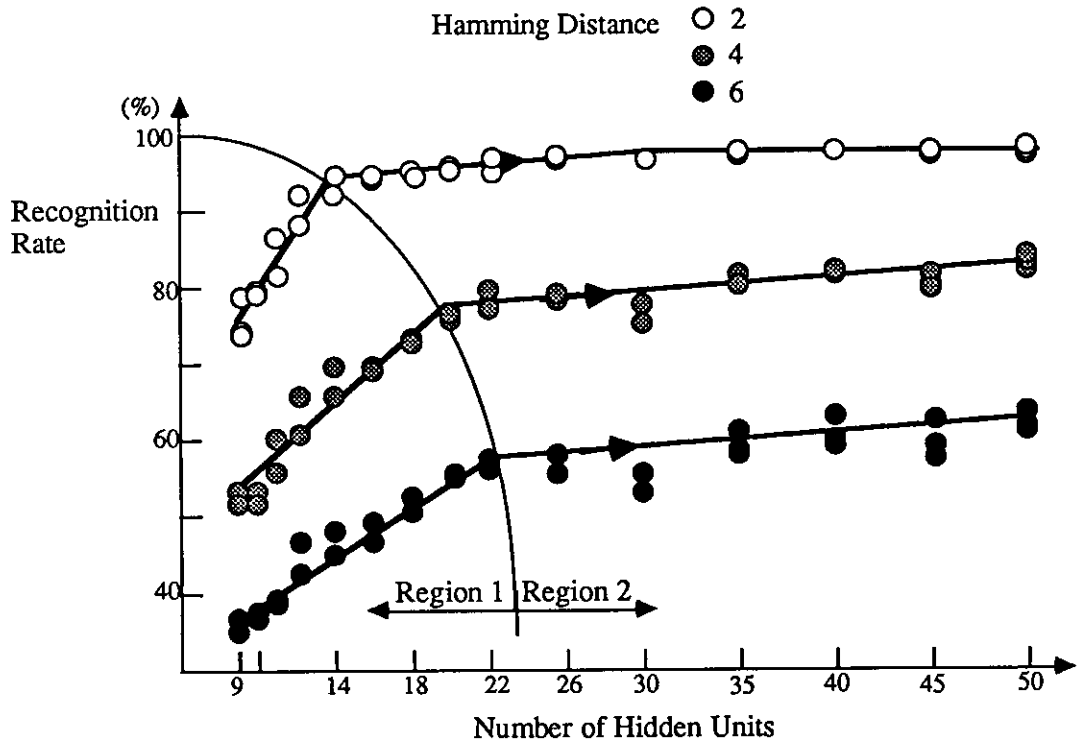



Fig.3 Recognition Rate Plotted against Number of Hidden Units
(Networks for Character Recognition Trained by SBP)

units at which saturation occurs is larger when the Hamming distance is large. That is, when the Hamming distance is 2, the recognition rate is saturated at almost 14 hidden units, while it is saturated at 22 hidden units when Hamming distance is 6. When the Hamming distance is 2, the recognition rate is completely saturated at 30 hidden units.

Fig.4 also shows the recognition rate plotted against the number of hidden units obtained from the network of Fig.1 (b). In this simulation, the database for training consists of 100 data sets (10 for each number), which were selected as representative ones for each number, while the database for evaluating the recognition rate consists of 1430 data sets collected from a practical number recognition system. In Fig.4, similar behavior is observed as the Fig.3, that is, the recognition rate of the networks with 4 hidden units is rather small, while it is almost constant in the case of more than 6 hidden units. These relationships between the recognition rate and the number of hidden units are found to be quite general and independent of the applications.

These simulations can be schematically illustrated as shown in Fig.5. Nc is the minimum number of hidden units which can lead to convergence subject to the given convergence accuracy. "Noise" indicates the space distance such as the Hamming distance between the training data and the data to be generalized. As mentioned above,

(1) The generalization ability increases with the increase of the number of hidden units, then almost saturates;
(2) The number of hidden units at saturation is larger when the noise is large, which is plotted on the critical curve against the amplitude of the noise.

The optimal number of hidden units, Nt, seems to be given by Nc plus Na, which is the additional number of hidden units determined by the amplitude of the noise to saturate generalization ability.

Then the mechanism for this interesting behavior is considered. Fig.6 illustrates the information flows in the network. The boxes show the hyperspace created by each layer, whose sizes indicate the number of its units. The maximum space distance between training data and data to be generalized is represented by the radius of a circle around the training point. This circle is designated the mapped circle. To obtain an accurate generalization, it is necessary to keep the mapped circle within the boundaries of its cluster.

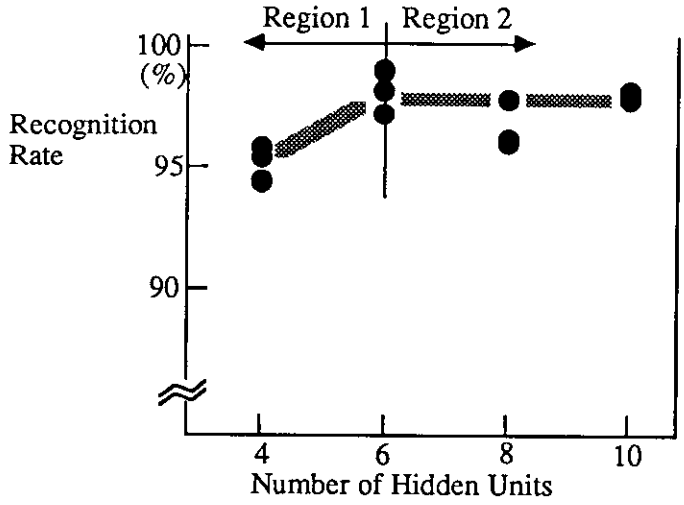
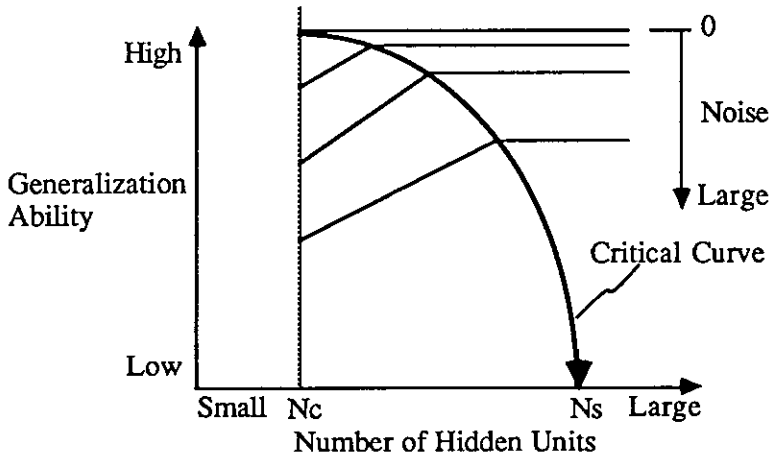



Fig.4 Recognition Rate Plotted against Number of Hidden Units
(Networks for Number Recognition Trained by SBP)



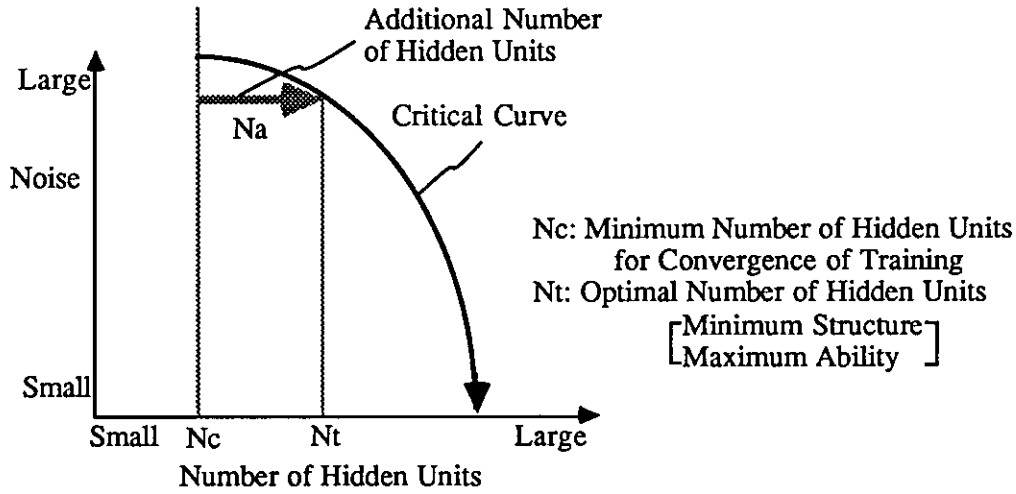




Fig.5 Relationships between Generalization Ability and Number of Hidden Units

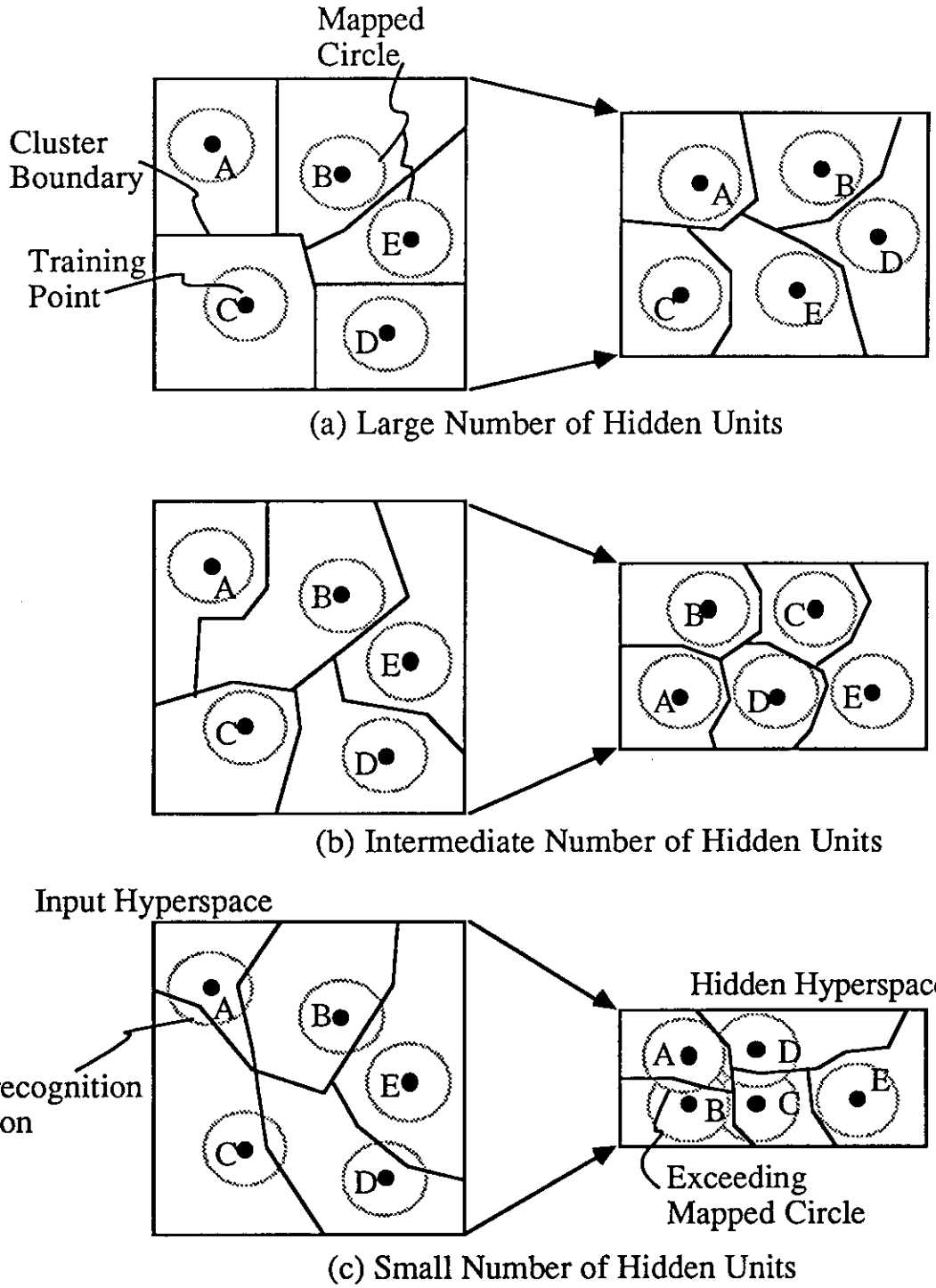



Fig.6 Information Flows in the Network

We now consider how the number of hidden units influences the relationship between the locations of the mapped circle and cluster boundary by information transformation through the input and hidden layers. Though the shape of each mapped circle is deformed by information transformation, all of them are described by circles schematically in Fig.6. In networks where the number of hidden units is smaller than the number of input units, information is condensed by the transformation from input to hidden units. In Figs.6 (a) and (b), in which information of the input layer is not condensed so much, the mapped circle can stay in its cluster, while in Fig.6 (c) with a high rate of information condensation, parts of the mapped circle cross the boundary and some information necessary for generalizing is lacking, which causes a misrecognition.

Fig.6 (b) has a slightly smaller number of hidden units than Fig.6 (a). However, it has enough of them to maintain the relationship between each mapped circle and corresponding cluster boundary. Therefore the generalization ability is not affected.

When the number of hidden units is too small, in Fig.6 (c) for example, several parts of the circle cross the boundaries with the decreasing number of hidden units. This results in misrecognition. The generalization ability, thus, decreases with the decrease of the number of hidden units in region 1.

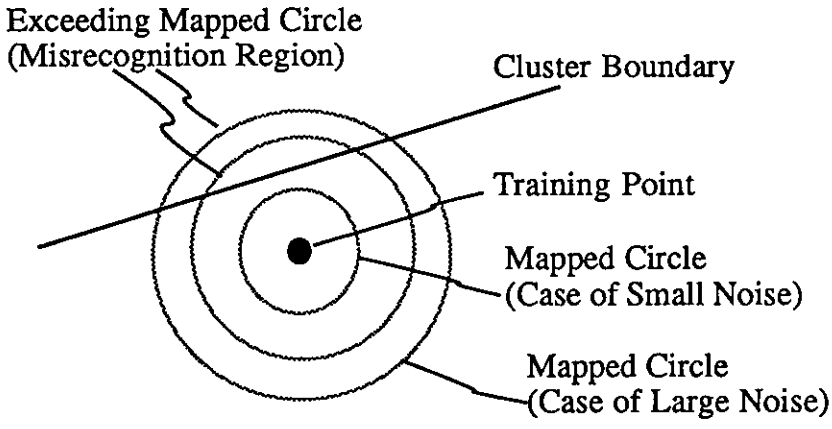



Fig.7 Relationship between Cluster Boundary and Mapped Circles

The amplitude of noise can also be represented by the radius of a circle in the input space. According to Fig.7, the smaller the radius, the less the mapped circles cross over the boundaries. So information condensation does not greatly deteriorate the generalization ability so much when the radius is small. This means that the number of hidden units at saturation is larger with more noise.

## 4. Improvement of generalization ability by dynamic back propagation

The adding of appropriate random numbers to the training inputs has been reported to be effective to improve the generalization ability of the network[4]. This training method contributes to preventing cluster boundaries from approaching each training point. Therefore the generalization ability is expected to be improved. In this section we discuss this method (DBP) to obtain a more accurate model describing the relationship the between generalization ability and the number of hidden units. When the training input vector is as in Fig.8 (a), the actual training input is modified as shown in Fig.8 (b) by adding random numbers of amplitude R. In this simulation, R is started at 2.0, and uniformly decreased to 0 with the number of training epochs. The total number of training epoch is 260000.

Fig.9 shows the recognition rate plotted against the number of hidden units of the network trained by the DBP. The dotted lines indicate the result for the network trained by the SBP, as shown in Fig.3. By this trained method, the recognition rate is improved, and three notable regions are found in it. The recognition rate increases with the number of hidden units (region 1); then it is almost saturated (region 2); and finally it is completely saturated (region 3). Compared with the SBP results, the recognition rate of the network trained by the DBP is uniformly improved, independent of the number of hidden units, when the number of hidden units is not too large.

This behavior can be explained by the simple model shown in Fig.10. In the figure, network ability for mapping is determined by superposing two factors; the essential capability and the sleeping capability. The essential capability is determined by the number of hidden units, while the sleeping capability is the measure of the amount of non-working capability included in the essential capability. The sleeping capability is caused by unbalanced mapping in the hidden layer for each cluster and reduces the total

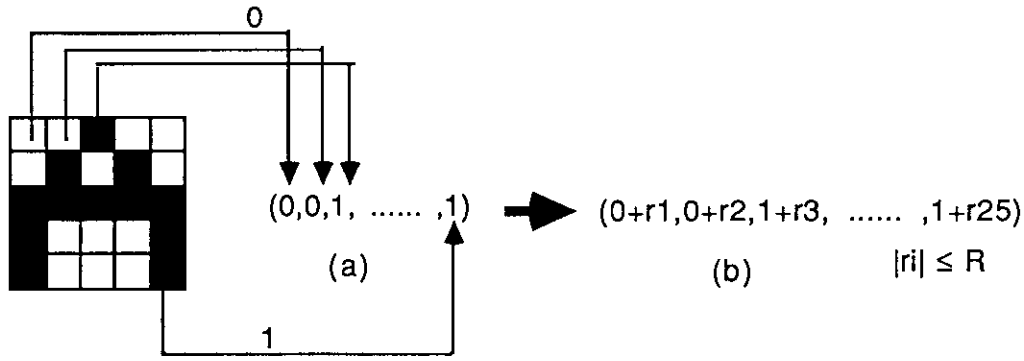



Fig.8 Original and Modified Training Input

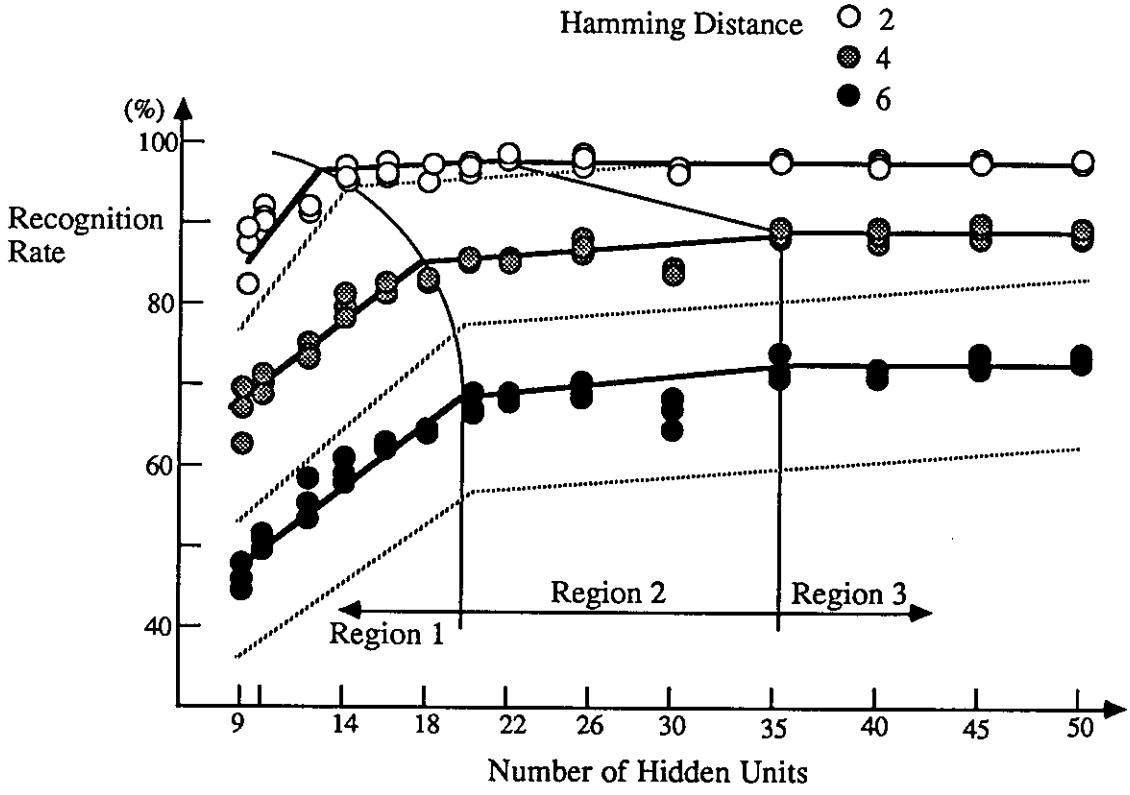



Fig.9 Recognition Rate Plotted Against Number of Hidden Units
(Network for Character Recognition Trained by DBP)

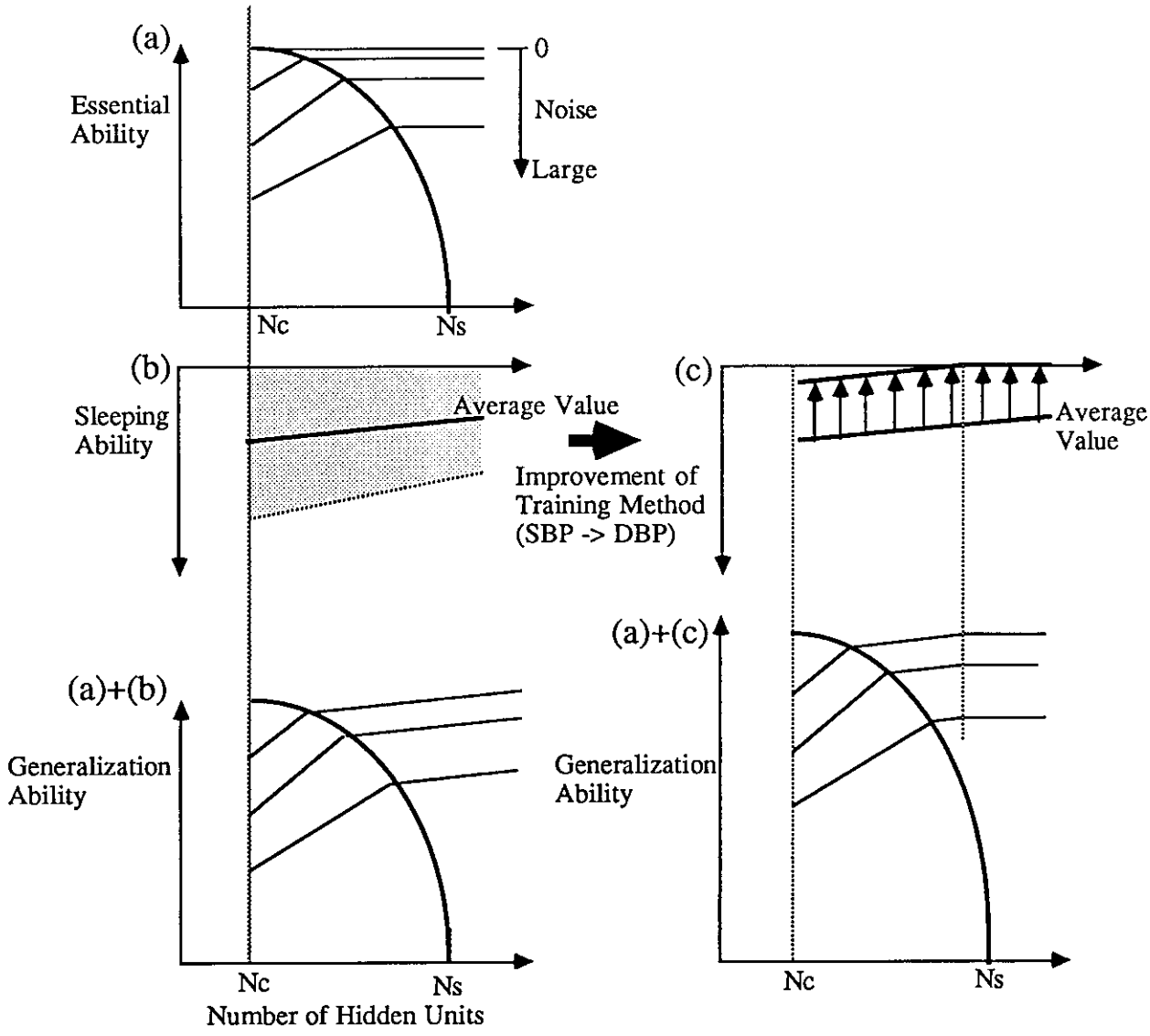
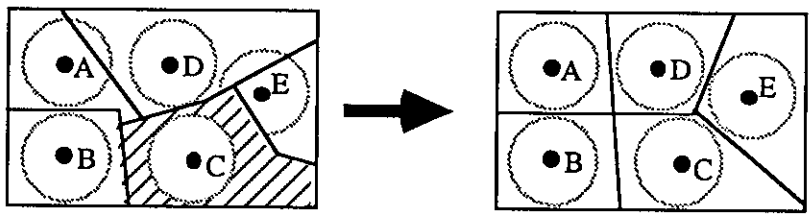



Fig.10 Schematic Illustration Describing the Effect of Two Kinds of Abilities




(a) Trained by SBP (b) Trained by DBP

Fig.11 Relationships between Cluster Boundaries and Mapped Circles in Hidden Space

generalization ability. The generalization ability of networks can be given by subtracting the sleeping capability from the essential capability. In Fig.11 (a), for example, there is extensive mapping for data around training point C, but mapping is insufficient for data around training points A and E. The hatched area around training point C is unnecessary for generalization. This makes some of the essential capability meaningless (sleeping), while necessary information for generalizing data close to A and E is missing in the hidden layer.

The essential capability increases with the number of hidden units, which is consistent regardless of training methods, SBP, DBP, or others. However, too much capability is unnecessary when the distance between training data and data to be generalized is small. Therefore it is saturated at some number of hidden units which is determined by this distance.

The sleeping capability cannot easily be evaluated quantitatively and can take on various values depending on the network. According to the simulations of Figs.3 and 9, it appears to be as shown in Fig.10 (b). Increasing the number of hidden units slightly and probablistically contributes to moving the cluster boundaries far from training points. Therefore the average of the sleeping capability increases gradually with the number of hidden units. The sleeping capability can be 0, if the unbalanced mapping can be completely excluded and hidden units fit each training input safely. DBP contributes to excluding this unbalanced fitting, and makes the fitting area of each training point as in Fig.9 (b). But actually it can not work perfectly unless the amplitude of the random numbers added to the training inputs is decreased during infinite training time. The sleeping capability is supposed to be excluded like Fig.10 (c). Consequently the results of Fig.3 can be obtained by superposing by Figs.10 (a) and (b), while Fig.9 can be obtained by superposing Figs.10 (a) and (c). In both cases, these models can explain the existence of different regions of the relationship between the generalization ability and number of hidden units without contradiction.

When the distance between training and testing data is small, the generalization ability may be saturated at the maximum value. When the Hamming distance is 2 in Figs.3 and 9, for example, it is completely saturated at a small number of hidden units.

## 5. How to obtain the optimal number of hidden units

In this section, the methodology for determining the optimal number of hidden units is discussed. The above investigation suggests, from a view point of generalization ability of networks, that the larger the number of hidden units, the better, unless the sleeping capability of the network is not completely removed. But in the region 2, the improvement of the generalization ability is very small compared with the increase in the network scale.

On the other hand, if it is possible to exclude the sleeping capability perfectly by improving training method, the boundary number of hidden units between regions 1 and 2 (Np) is recognized as optimal, where the network has the minimal structure and has the maximum generalization ability.

Accordingly, evaluating Np is meaningful and important when determining the number of hidden units. We now consider how to obtain Np.

From Figs.3, 4, and 10, Np is given by Nc plus Na. We already suggested an effective method for obtaining Nc using a linear regression analysis [3]. This method consists of following general steps;

(1) Create a trained network which has a sufficient number (initial number, m) of hidden units.
(2) Input the training data into the network again and detect the output of the hidden units.
(3) 1 -> $i$.
(4) Approximate the output of $i$-th hidden unit by a linear combination of other outputs, Y$i$+1 ~ Ym, which is Y$i$*, using linear regression analysis.
(5) The ratio of Y$i$ represented by the other outputs, C$i$, is calculated using a multiple regression coefficient between original output Y$i$ and estimated output Y$i$*.
(6) $i$+1 -> $i$.
(7) If ($i \neq$ m), then go to (4).
(8) Nc = m - (C1+C2+ ..... +Cn-1).

m ; initial number of hidden units except a bias unit

In the network of Fig.1 (b), training dose not converge within 1% of full scale of the desired output unless the network has at least 4 hidden units. So 4

is the minimum number needed for convergence. Fig.12 shows the Nc estimated in the network of Fig.1 (b) with 4, 6, 8 and 10 initial hidden units. In the case of 4 hidden units, Nc's are from 3.0 to 3.4, while in the case of more than 6 hidden units, they are around 4, and almost independent of the initial numbers of hidden units. This suggests that at least 4 hidden units are necessary for convergence of the network, which is in accordance with the above simulation results.

By this method, Nc is estimated quantitatively, however, evaluating Na can be difficult. In the following investigation, we suggest a simple method to obtain Np, which is applicable to a network whose output consists of binary signals (binary network) for classification.

In the network of Fig.1 (a), the input and output signals are either 1 or 0. Therefore the outputs of the input and output layers are binary. Also, in a classification problem, where training data are binary, most hidden units work in their saturation regions, whose outputs corresponding to the training inputs

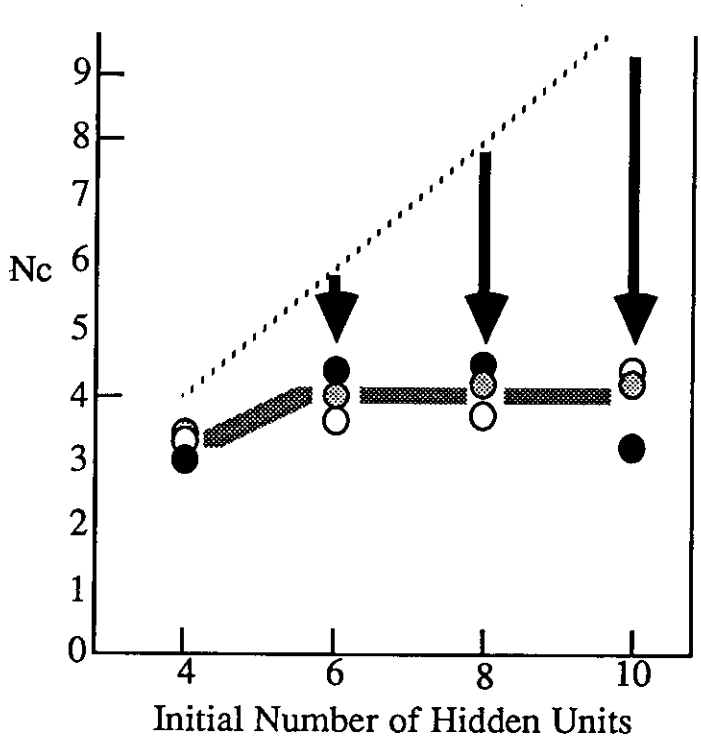



Fig.12 Minimum Number of Hidden Units for Convergence of Training

are about 0 or 1. Tables 1 and 2 show the output of the hidden units against each training input from A to Z, in cases of 10 and 25 hidden units. In the tables, "I" , "O", and "." mean that the output of the hidden unit is larger than 0.7, smaller than 0.3, and between 0.3 and 0.7, respectively. When output is "I" or "O", the corresponding hidden unit is assumed to move into its saturated region. According to both table, 88.5% and 86.6% hidden units move into their saturation regions in the cases of 10 and 25 hidden units, respectively. Therefore the network of Fig.1(a) can be recognized to be approximately a binary network.

When the output of each unit is either 0 or 1, the layer consisting of n units can represent $2^n$ different states. Assume that in the case of a binary network, the amount of information represented by each layer including n units is also $2^n$. According to this assumption, the input space of Fig.1(a) can represent $2^{25}$ (=33554432) states. Also, if the Hamming distance of the test data is $i$, the number of states represented by this distance from each training input, $Si$, without considering duplicated regions, is

$$Si = N * {}_nC_i \tag{2}$$

N: number of training data (number of clusters)
n : number of input units

When the Hamming distance of the test data is $h$, the total number of states corresponding to the entire mapped circles, $Stotal$, is

$$Stotal = \sum_{i=0}^{h} Si \tag{3}$$

The minimum number of units which can represent all the mapped circles, m, is

$$2^m \geq Stotal$$
$$\geq N * \sum_{i=0}^{h} {}_nC_i \tag{4}$$

Table 1 Output of Hidden Units (Case of 10 Hidden Units)

| | Number of Hidden Unit | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A | I | I | I | O | O | I | I | I | O | O |
| B | O | . | I | O | I | O | I | I | O | . |
| C | I | O | I | . | I | O | O | I | O | I |
| D | I | I | O | O | I | I | O | I | I | O |
| E | O | . | O | O | I | I | I | I | O | O |
| F | O | . | O | . | I | I | I | O | I | O |
| G | . | I | . | O | I | I | O | I | O | I |
| H | O | I | I | I | . | I | I | O | O | O |
| I | . | O | I | O | I | O | O | I | I | I |
| J | I | O | . | O | . | I | I | I | I | O |
| K | O | O | I | I | I | O | I | O | I | I |
| L | O | I | I | I | I | O | O | I | . | O |
| M | I | O | O | O | I | I | . | . | O | I |
| N | I | I | O | I | . | I | I | O | O | O |
| O | I | O | O | O | I | O | I | I | O | I |
| P | O | I | O | I | I | . | O | . | I | I |
| Q | I | I | O | O | I | O | . | I | O | I |
| R | O | I | O | I | . | I | . | I | O | I |
| S | I | O | I | I | I | . | I | I | O | O |
| T | I | . | O | O | I | . | I | O | I | I |
| U | I | . | I | O | I | O | I | O | O | O |
| V | O | I | O | O | . | I | I | O | O | I |
| W | O | I | I | I | I | I | O | O | O | I |
| X | I | I | . | I | O | I | O | I | I | . |
| Y | O | I | I | O | I | O | I | O | I | . |
| Z | I | O | O | I | I | . | O | I | I | O |

Table 2 Output of Hidden Units (Case of 25 Hidden Units)

| | Number of Hidden Unit | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| A | . | I | O | I | I | O | I | I | O | O | O | O | I | O | I | I | I | O | . | I | I | I | I | I | O |
| B | I | O | O | I | I | . | I | I | O | I | O | O | O | I | I | O | . | O | I | O | O | . | O | I | O |
| C | O | I | . | O | I | I | I | . | I | I | I | O | O | I | O | O | . | I | . | . | I | O | O | I | I |
| D | . | O | I | . | O | I | O | I | O | I | O | O | O | . | O | I | O | I | O | I | I | . | O | . | I |
| E | I | I | O | . | I | O | O | I | O | O | O | O | O | . | I | O | O | O | I | O | O | I | I | I | I |
| F | I | I | O | I | O | O | O | I | O | O | O | O | I | I | I | I | . | I | I | O | O | I | I | . | O |
| G | O | I | I | I | I | I | . | I | O | I | O | O | O | . | O | O | O | O | . | . | I | I | O | I | . |
| H | I | I | I | I | I | . | O | I | O | O | O | O | I | I | I | . | I | O | O | O | O | I | I | I | O |
| I | I | O | O | I | O | I | I | I | I | I | I | O | O | I | O | O | O | O | I | . | I | O | O | O | I |
| J | I | I | O | I | . | . | O | I | I | I | O | O | O | O | O | I | O | . | O | O | I | O | I | O | I |
| K | I | . | O | O | O | I | O | O | O | O | O | O | I | I | . | O | I | O | I | I | . | I | I | I | I |
| L | I | O | O | O | I | I | O | . | O | I | I | O | O | I | I | . | O | O | I | . | . | I | O | I | I |
| M | I | I | I | O | O | . | I | I | O | O | O | O | I | I | . | I | O | I | O | I | I | I | O | I | O |
| N | . | I | I | I | I | I | O | . | O | O | O | O | I | I | . | I | I | O | O | I | I | I | O | O | I |
| O | O | I | I | O | I | O | I | I | O | I | I | O | O | I | I | O | I | . | . | . | I | O | O | I | O |
| P | I | O | O | . | O | I | I | I | O | I | O | O | O | I | . | I | O | O | I | O | O | I | I | . | O |
| Q | O | O | I | I | I | O | . | . | O | I | I | O | O | I | O | O | I | O | I | I | I | O | O | I | I |
| R | O | . | I | I | O | O | O | I | O | I | O | O | O | . | I | I | I | O | I | . | I | I | I | I | O |
| S | I | I | . | I | I | O | . | . | O | . | O | O | O | I | . | O | I | I | I | O | . | O | I | I | O |
| T | I | I | O | O | O | . | I | I | I | O | O | O | I | I | O | . | O | . | I | I | I | O | I | O | I |
| U | I | . | I | O | I | I | O | I | . | . | O | O | O | I | I | O | I | O | O | O | I | O | O | I | I |
| V | . | O | I | O | . | O | O | . | I | O | O | . | O | I | I | I | I | I | O | I | I | I | I | . | I |
| W | . | O | I | I | I | I | O | O | I | . | O | I | . | I | O | I | . | O | O | O | I | I | O | I | I |
| X | I | I | I | I | O | I | O | . | O | O | O | I | I | O | O | I | O | . | I | I | I | I | I | O | I |
| Y | I | O | . | . | I | I | O | . | I | O | O | O | I | I | I | . | . | O | . | O | O | I | I | O | I |
| Z | I | I | O | O | O | . | I | I | O | I | O | O | O | O | O | I | O | I | I | O | I | . | O | O | I |

$$m \geq \log_2(N * \sum_{i=0}^{h} n\,C_i) \qquad (5)$$

Fig.13 shows the critical curve calculated by eq.(5) where Nt was obtained by Fig.9. We found, in Fig.9, that Np was 13, 18, and 20, in the case of 2, 4, 6 hidden units, respectively. When the Hamming distance *h* is relatively small, they are in accordance with each other, however, when *h* is large, the number of hidden units obtained with the critical curve become large. The reason of this difference seems to be the duplicated regions; that is, regions included in several mapped circles are counted twice or more. Therefore *Stotal* evaluated by eq.(3) seems to be much larger than the actual value, *Sact*, when *h* is large.

*Sact* can be estimated accurately by the following algorithm, but with an enormous CPU time;

(1) i -> 1, 0 -> *Sact*.
(2) if i exists within the *h* Hamming distance for at least one among all training inputs,
*Sact*+1 -> *Sact*.
(3) i+1 -> i.
(4) $i \leq 2^{25}$ go to (2).
(5) *Sacc* is the total number of states.

The obtained *Sact* values are shown in Fig.14 with the results of Fig.13 (dotted line). When the number of hidden units is less than 5, the effect of the duplicate regions on the critical curve is found to be very small. But when it is larger than 5 or 6, the effect cannot be neglected. In this case *Sact* gives a more appropriate number of hidden units. From Figs.13 and 14, this simple method is confirmed to estimate the optimal number of hidden units.

## 6. Conclusion

The quantitative relationship between generalization ability and the number of hidden units of the multi-layered neural networks was modeled, and the determination of the optimal number of hidden units was investigated. The following conclusions were reached.

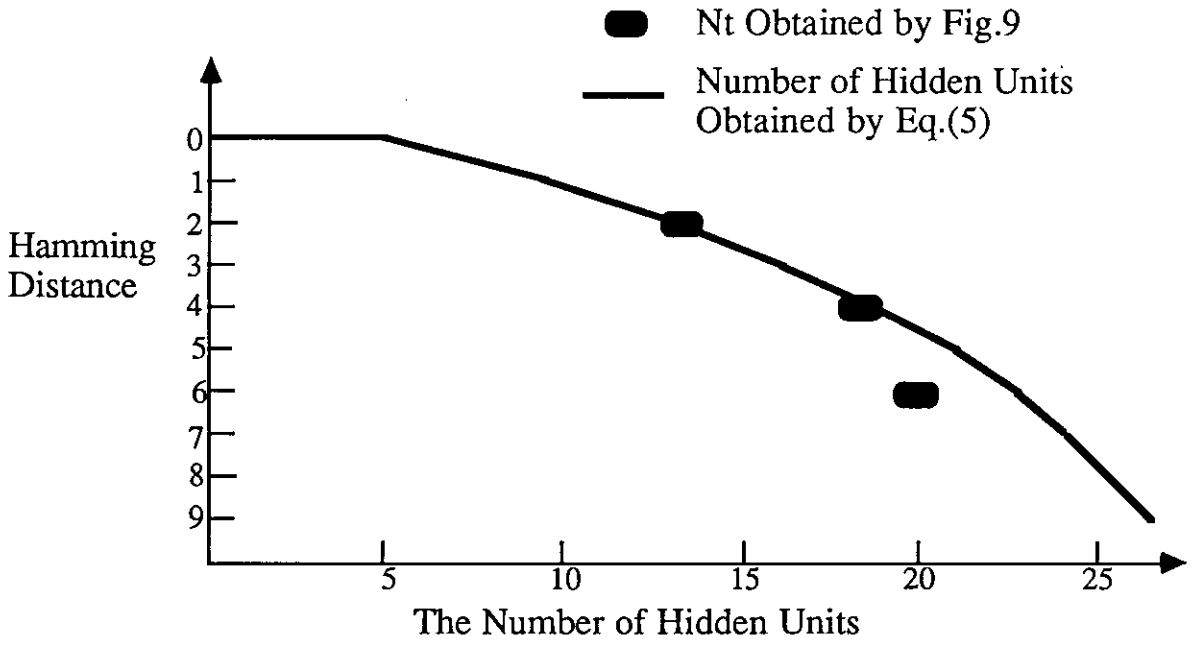



Fig.13 Critical Curve Obtained by eq.(5)

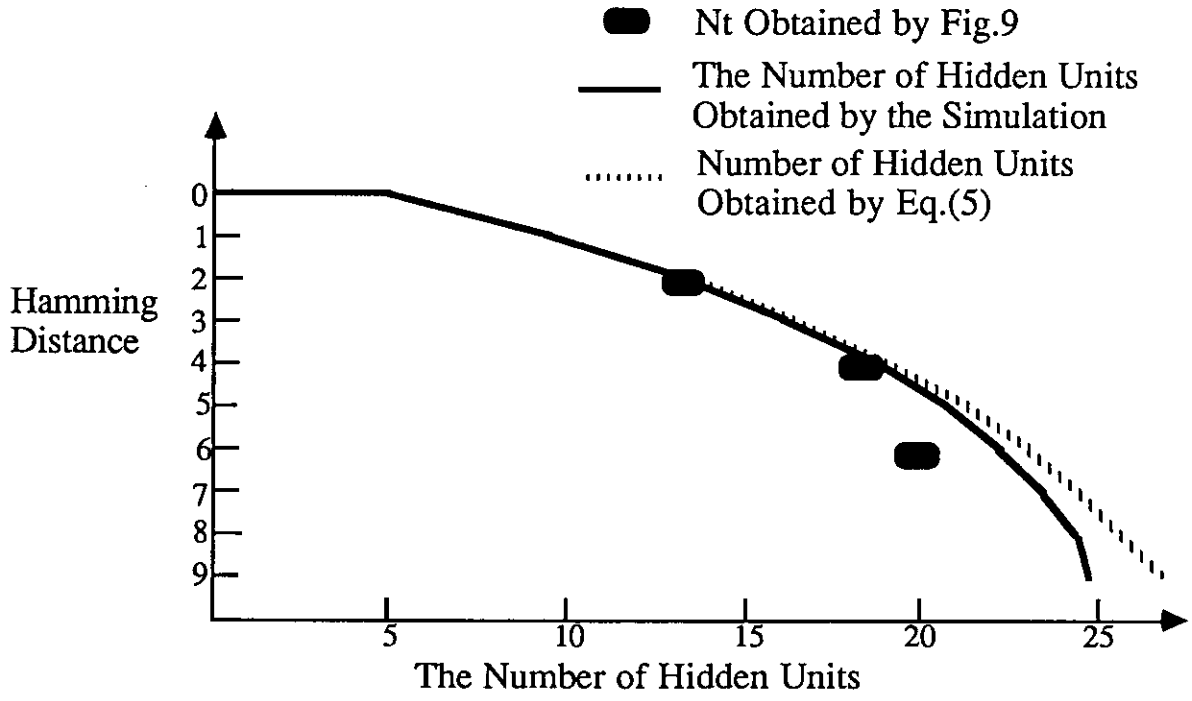



Fig.14 Critical Curve Obtained by the Simulations

1. The relationship between generalization ability and the number of hidden units can be represented by a simple model using two kinds of network capabilities: the essential and the sleeping capabilities. The generalization ability of the network is determined by subtracting the sleeping capability, which does not contribute to the mapping, from the essential capability, which depended on just the number of hidden units.

2. In the network trained by SBP, where the sleeping capability is large, the generalization ability increases with the number of hidden units (region 1), and then almost saturates (region 2). On the other hand, in the network trained by DBP, where most of the sleeping capability is removed, the generalization ability is improved. It increases with the number of hidden units (region 1), then almost saturates (region 2), and finally, completely saturates (region 3). In both cases, the number of hidden units at saturation became larger with the distance between the training data and the data to be generalized.

3. To evaluate the optimal number of hidden units which provides the minimum network structure and the maximum generalization ability, estimation of the boundary number of hidden units between regions 1 and 2 is important. For this, estimating the number of states which should be mapped, out of the entire input space was found to be effective. This idea was applied to a binary network, which handles binary inputs and outputs, and its effectiveness was confirmed.

In future studies, we will extend this idea to general networks.

## ACKNOWLEDGEMENTS

The author gratefully acknowledges the help of Dr. Walter J. Karplus, Mr. Edwin Robert Tisdale, and Mr. Han-sen Dai.

## References

[1] Q. Xue et al., "Analyses of the Hidden Units of Back Propagation Model by Singular Value Decomposition SVD", IJCNN'90-WASH-DC, January 1990

[2] T. Ash, "Dynamical Node Creation in Back Propagation Networks", IJCNN'89-WASH-DC, June 1989

[3] M. Kayama et al., "Constructing Optimal Neural Networks by Linear Regression Analysis", Neuro-Nimes '90, November 1990

[4] D. E. Rumelhart et al., "Parallel Distributed Processing", Vol. 1,2 MIT Press, 1986

[5] J. Sietsma et al., "Creating Artificial Neural Networks That Generalize", Neural Networks , Vol.4, No.1 (1991)

[6] K. Funahashi, "On the Approximate Realization of Continuous Mapping By Neural Networks", Neural Networks, Vol.2, No.3 (1989)

[7] Z. Wei, et al., "Approximation Property of Multi-Layer Neural Net and its Application in Non-linear Simulation", IJCNN-Seattle, 1991