OPTIMIZATION OF ACTIVATION FUNCTIONS IN MULTILAYER NEURAL NETWORK APPLIED TO PATTERN CLASSIFICATION

Kenji NAKAYAMA **Yoshinori KIMURA**

Dept. of Electrical and Computer Eng.,
Faculty of Technology, Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa 920 JAPAN
E-mail: nakayama@haspnn1.ec.t.kanazawa-u.ac.jp

ABSTRACT: An optimization method of activation functions is proposed. Three typical functions are combined in hidden layers. Contribution of the functions is evaluated using three criteria. The useful functions are selected or multiplied in the learning process. Problems of parity and of counting '1' in bit-patterns can be solved by the proposed method with the suitable functions and the minimum number of hidden units.

I INTRODUCTION

Neural networks (NNs) have powerful and flexible performances, such as self-organization and learning [1], [2]. However, many points still remain to be optimized by designers.

One of them is an activation function. From biological inspire, a squashing function including a threshold function and a sigmoid function, are widely used. Recently, radial basis functions have been discussed [2]. However, there are so many kinds of functions, which can be used for activation functions.

One method of optimizing activation functions is to choose an appropriate function before training the NNs. However, useful functions are highly dependent on distribution of the input patterns in an N-dimensional space. N is the number of dots used in the patterns. When N is relatively large, it is very difficult to estimate the distribution of N-dimensional vectors.

Another method is a self-optimization approach. Methods of self-adjusting the number of hidden units have been well discussed [3]-[6]. However, discussion on self-optimization of activation functions has not been well done.

This paper concerns the latter approach. An optimization method is proposed, which can optimize activaion functions and minimize the number of hidden units in multilayer neural networks applied to pattern classification. The backpropagation algorithm [1] is basically used. Efficiency of the proposed method is examined through computer simulation.

II MULTILAYER NEURAL NETWORK

2.1 Network Structure and Equations

A multilayer neural network is shown in Fig.1. A single hidden layer is used for simplicity in this paper.

Let $\mathbf{x}_p$ be a pth input pattern with N-dimensional to be classified.

$$\mathbf{x}_p=[x_{p1},x_{p2},\ldots,x_{pN}],\ 1\leqq p\leqq M \qquad (1)$$

The input and output of the jth hidden unit are given by

$$u_{pj} = \sum_{i=1}^{N} w_{ij}x_{pi} + \theta_j \qquad (2)$$

$$v_{pj} = f(u_{pj}) \qquad (3)$$

0-7803-1901-X/94 $4.00

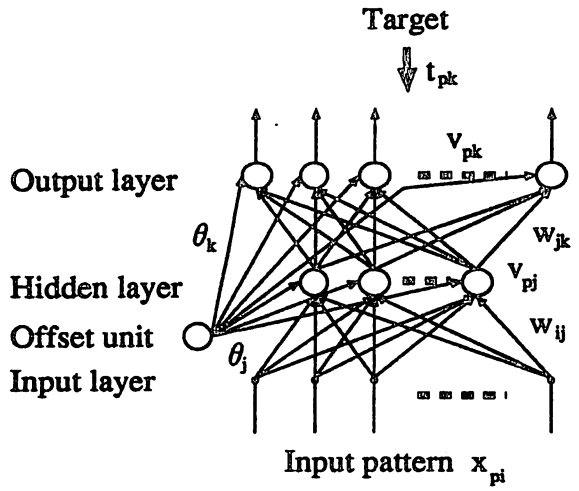

Fig.1 Block diagram of multilayer neural network.

w_{IJ} and θ_J are connection weights from the ith input unit and the offset unit to the jth hidden unit, respectively. The offset unit always outputs '1'. f() is an activation function. In the output layer, the same equations are held.

$$u_{pK} = \sum_{j=1}^{NH} w_{JK} v_{pJ} + \theta_K \qquad (4)$$

$$v_{pK} = f(u_{pK}) \qquad (5)$$

w_{JK} and θ_K are connection weights from the jth hidden unit and the offset unit to the kth output unit, respectively. N_H is the number of the hidden units.

2.2 Activation Functions in Hidden Layers

Activation functions in the hidden layers are optimized. In the pattern classification, binary targets can be used. In this case, squashing functions can be used in the output layer. This does not limit the performance of NNs. If some kinds of activation functions are useful in the output layer, these functions can be implemented in the additional hidden layer instead of the output layer.

III CONDITIONS ON ACTIVATION FUNCTIONS

In order to guarantee stable convergence and calculation of differential, some conditions should be satisfied by activation functions.

3.1 Finite Value Functions

Let x and f() be an input and an activation function.

$$y = f(x) \qquad (6)$$

The activation functions should satisfy the following condition.

$$| f(x) | < +\infty \quad \text{for any } x \qquad (7)$$

This kind of function is called 'finite function' in this paper. For example, the following functions are not stable in learning.

$$y = f(x) = x^a, \; a>0 \qquad (8)$$

$$y = f(x) = e^{ax+b}, \; a>0 \qquad (9)$$

The reasons of instability caused by the above functions can be explained as follows: First, y can take a large value. In this situation, connections weights from the hidden units tend to become very small. This is unstable combination. Second, in the training data, the range of x is rather limited. Therefore, some parts of the functions are only used. However, if noise is added, x is easily expanded. In the expanded range, y will take a very large value, which cause unstable response.

3.2 Differential of Activation Functions

The backpropagation algorithm is a sort of the gradient methods, which requires differential of the activation functions. Therefore, a possibility of calculating differential is needed.

IV COMPOSITE ACTIVATION FUNCTIONS

4.1 Combination of Typical Activation Functions

In this paper, we propose composite activation functions, which combine typical functions in the hidden layer. One of the typical functions is assigned to a subset of hidden units.

4.2 Typical Activation Functions

Useful activation functions are highly dependent on distribution of the patterns to be classified in an N-dimensional space. However, it is difficult to estimate this distribution, when N is large.

Therefore, an optimization method, which selects or multiplies useful activation functions, is proposed. This method will be provided in **Sec. V**. For this method, typical functions should be discussed and determined in advance, taking the conditions given in **Sec. III** into account.

The following three functions are selected in this paper.

(1) Squashing Function
(2) Radial Basis Function
(3) Periodic Function

Using the first function, the space can be divided by hyperplanes. The second can form some isolated regions. The last can divide the space into periodical regions. Therefore, they can play an independent and important role for space division.

The above three categories include, for instance, **(1)** a threshold function and a sigmoid function, **(2)** a Gaussian distributed function and **(3)** a sinusoidal function.

4.3 Hyperspace Division by Three Functions

Examples for the three typical functions are shown here.

(1) Sigmoid function:

$$v = f_S(u) = \frac{a_1}{1 + e^{-(b_1 u + c_1)}} + d_1 \quad (10)$$

(2) Gaussian distribution function:

$$v = f_R(u) = a_2 e^{-b_2(u+c_2)^2} + d_2 \quad (11)$$

(3) Sinusoidal function:

$$v = f_P(u) = a_3 \sin(b_3 u + c_3) + d_3 \quad (12)$$

a_1, b_1, c_1 and d_1 are all constant. Another functions can be used. However, the ways of dividing the space are basically the same. For simplicity, the above concrete functions are taken into account.

The input of the jth hidden unit is given by Eq.(2). The hidden unit having one of three functions responds to the following input regions. Some basic functions are assumed.

$$v_{pj} = f_S(u_{pj}) \begin{cases} > \alpha_+, & u_{pj} > \theta_S \quad (13a) \\ < \alpha_-, & u_{pj} < \theta_S \quad (13b) \end{cases}$$

Thus, taking Eq.(2) into account, the space of the input patterns is divided by the hyperplanes.

$$v_{pj} = f_R(u_{pj}) \begin{cases} > \alpha_+, & |u_{pj} + c| < \theta_R \quad (14a) \\ < \alpha_-, & |u_{pj} + c| > \theta_R \quad (14b) \end{cases}$$

The space is divided into the belt region and the other. If several hidden units having $f_R()$ are combined, isolated regions can be formed.

$$v_{pj} = f_P(u_{pj}) \begin{cases} > \alpha_+, & |u_{pj} - (1/2+2n)\pi| < \theta_P \\ & \quad n: \text{integer} \quad (15a) \\ < \alpha_-, & |u_{pj} - (3/2+2n)\pi| < \theta_P \quad (15b) \end{cases}$$

A single $f_P()$ divides the space into the periodic belt region. By combining several $f_P()$, periodic and isolated regions are formed.

Thus, these functions can be fundamental in dividing the space.

V OPTIMIZATION METHOD

The next step is how to optimize the activation functions. Selection and

multiplication of useful functions are employed. Some criteria are also proposed for evaluating usefulness.

5.1 Evaluation of Useful Functions

Contribution of the activaion functions, equivalently of the corresponding hidden units, are evaluated based on the following three criteria.

(A) Information from the jth hidden unit to the output layer $< I\hat{}_J >$.

$$I_J = \frac{1}{M} \sum_{p=1}^{M} \sum_{k=1}^{NO} | w_{Jk} v_{pJ} | \quad (16)$$

N_O is the number of the output units.

$$I\hat{}_J = \frac{I_J}{\max_j \{I_J\}} \quad (17)$$

(B) Variance of the jth unit output for all patterns $<V\hat{}_J>$.

$$V_J = \sum_{p=1}^{M} (v_{pJ} - \underline{v}_J)^2 \quad (18)$$

$$\underline{v}_J = \frac{1}{M} \sum_{p=1}^{M} v_{pJ} \quad (19)$$

$$V\hat{}_J = \frac{V_J}{\max_j \{V_J\}} \quad (20)$$

(C) Correlation between outputs of the jth and j'th hidden units $<R_{Jj'}>$.

$$A_{Jj'} = \sum_{p=1}^{M} (v_{pJ} - \underline{v}_J)(v_{pj'} - \underline{v}_{j'}) \quad (21)$$

$$B_{Jj'} = \sum_{p=1}^{M} (v_{pJ} - \underline{v}_J)^2 \sum_{p=1}^{M} (v_{pj'} - \underline{v}_{j'})^2 \quad (22)$$

$$\gamma_{Jj'} = A_{Jj'} / B_{Jj'}^{1/2} \quad (23)$$

$$R_{Jj'} = 1 - | \gamma_{Jj'} | \quad (24)$$

5.2 Selection Process of Useful Functions

Step1: Prepare several hidden units for three activation functions. The initial connection weights are set to small random numbers.

Step2: The NN is trained by a supervised learning, such as the backpropagation [1].

Step3: After the training converges to some extent, the contribution of each hidden unit is examined based on the three criteria. They are used in a multiplicative form. That is, if the following condition is held, then the jth hidden unit is considered to be removed or not to be.

$$C_J = I\hat{}_J V\hat{}_J R_{Jmin} \leqq \varepsilon \quad (25)$$

$$R_{Jmin} = \min_{j'} \{R_{Jj'}\} \quad (26)$$

Furthermore, the connection weights are modified as follows:

(A) If $I\hat{}_J$ is the minimum among three criteria, then the jth hidden unit is removed. The connection weights are not changed.

(B) If $V\hat{}_J$ is the minimum, then the jth hidden unit is deleted. The connection weight from the offset unit to the kth output unit is modified by taking effects of the removed hidden unit into account as follows:

$$\theta_k = \theta_k + w_{Jk} \underline{v}_J \quad (27)$$

(C) When R_{Jmin} is the minimum, the j'th hidden unit, which provides R_{Jmin}, is further investigated. If the j'th hidden unit is removed for the previous **(A)** or **(B)** reason, the jth hidden unit can still remain. Otherwise, the jth or j'th hidden unit is removed. If $V_J > V_{j'}$, then the j'th hidden unit is removed, and vice versa. In the former case, the connection weights from the jth hidden unit and the offset unit to the kth output unit are modified.

$$w_{Jk} = w_{Jk} + \alpha w_{j'k} \quad (28)$$

$$\theta_k = \theta_k + w_{j'k} (\underline{v}_{j'} - \alpha \underline{v}_J) \quad (29)$$

$$\gamma_{Jj'} \fallingdotseq 1, \quad \alpha = \{V_{j'} / V_J\}^{1/2} \quad (30)$$

$$\gamma_{Jj'} \fallingdotseq -1, \quad \alpha = -\{V_{j'} / V_J\}^{1/2} \quad (31)$$

Steps 2 and **3** are repeated, and the hidden units, whose contribution is low, are gradually decreased.

5.3 Multiplication Process of Useful Functions

The selection method only removes

hidden units with less contribution. Therefore, a relatively large number of hidden units should be prepared in order to cover a wide range of pattern classification problems.

In order to save the hidden units, multiplication of useful hidden units is employed. A moderate number of hidden units are prepared. The NN is trained following the selection process described above. After the learning converges to some extent, the useful functions are investigated in the same ways. A hidden unit having the useful function is added. The connection weights to and from this unit are determined as average of the other hidden units in the same group. The training is continued until the NN reaches to an equilibrium point.

VI COMPUTER SIMULATION AND DISCUSSIONS

The proposed method is examined using binary pattern classification.

6.1 Parity Problem

The parity problem is difficult to be solved by the multilayer NNs with sigmoid functions trained by the backpropagation. The distribution of the even and odd bit-patterns are mixed together. Especially, when the number of bits is larger than 6, solving it is almost impossible from our experience.

The proposed method is applied to the 8-bit parity problem. Eight hidden units are assigned to each function. Thus, 24 hidden units are prepared in the initial NN. A sigmoid function is used in the output unit. The threshold in Eq.(25) ε is determined to 0.05.

After the first learning converges with 1355 iterations, five hidden units of the sinusoidal function, remain. After the second learning converges with 1543 iterations, two hidden units remain. Finally, a single hidden unit of the sinusoidal function is selected after 1544 iterations.

Figure 2 shows relation between the inputs and outputs of the hidden units. Table 1 shows relations between the numbers of '1' in the input patterns and the hidden unit outputs. This is a unique solution, that is a globally optimum solution with the minimum number of hidden unit.

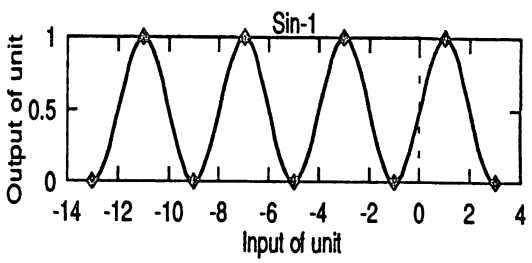

Fig.2 Relation between input and output of hidden unit, having sinusoidal function.

Table 1 Relations between numbers of '1' in bit-patterns and hidden unit outputs.

Unit No.	Output of hidden units	
	Upper	Lower
Sin-1	1, 3, 5, 7	0, 2, 4, 6, 8

6.2 Counting Number of '1' in Bit-Patterns

This is another interesting problem to examine activation functions. This problem is not linearly separable nor periodic. Several kinds of initial functions are examined.

(1) The initial hidden units and functions prepared are the same as in **Sec.6.1.** The proposed optimization was carried out. As a result, three sinusoidal functions and one Gaussian function are selected.

Relations between the number of '1' in the input bit-patterns and the hidden unit outputs are shown in Fig.3. These relations can be summarized in Table 2. Thus, roles of dividing the input patterns into the

categories are effectively shared by the selected functions.

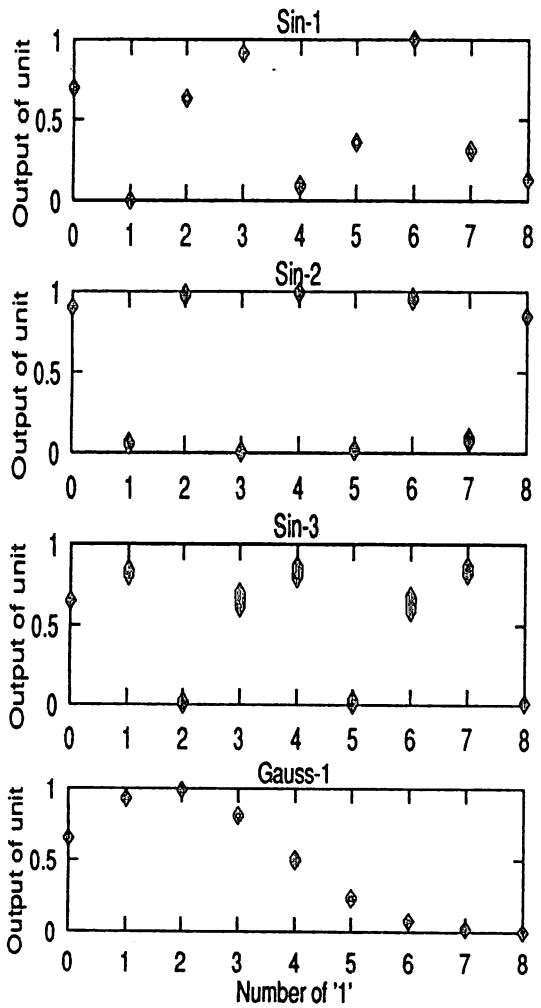

Fig.3 Relation between number of '1' in bit-patterns and hidden unit outputs.

Table 2 Relations between numbers of '1' in input bit-patterns and hidden unit outputs.

Unit No.	Output of hidden units	
	Upper	Lower
Sin-1	0, 2, 3, 6	1, 4, 5, 7, 8
Sin-2	0, 2, 4, 6	1, 3, 5, 7
Sin-3	0, 1, 3, 4, 6, 7	2, 5, 8
Gauss-1	0, 1, 2, 3	5, 6, 7, 8

(2) One of three functions is used. Ten hidden units, having the same function, are initially set. From the simulation results, the learning using the sigmoid function did not converge. The Gaussian function requires 6 hidden units. The sinusoidal function requires only four hidden units. Exactly saying, this problem is not periodic. However, the periodic function is still useful.

Relations between the number of '1' in the bit-patterns and the outputs of the hidden units are listed in Table 3.

Table 3 Relations between numbers of '1' in input bit-patterns and hidden unit outputs.

Unit No.	Output of hidden units	
	Upper	Lower
Sin-1	0, 2, 3, 6, 7	1, 4, 5, 8
Sin-2	0, 1, 3, 4, 7	2, 5, 6
Sin-3	0, 4, 5,	1, 2, 3, 7, 8
Sin-4	0, 1, 5, 6, 7	3, 4, 8

Ⅷ CONCLUSIONS

An optimization method of activation functions has been proposed. Three typical functions are combined in the hidden layers. Useful functions are evaluated using three criteria, which has been also proposed. The useful functions are automatically selected and multiplied in the learning process. The parity problem and the problem of counting '1' in bit-patterns have been effectively solved by the proposed method with the suitable functions and the minimum number of hidden units.

REFERENCES

[1]D.E.Rumelhart and J.L.McClelland, Parallel and Distributed Processing, MIT Press, 1986.
[2]P.D.Wasserman, Advanced Methods in Neural Computing, Van Nostrand Reinhold, 1993.
[3]M.Hagiwara,"Novel back propagation algorithm for reduction of hidden units and acceleration of～", Proc., IJCNN'90 San Diego, pp.I-625-630, June 1990.
[4]J.Sietsma and R.J.F.Dow,"Creating artificial neural networks that generalize", Neural Networks, vol.4, pp.67-79, 1991.
[5]E.Watanabe and H.Shimizu,"Algorithm for pruning hidden units in multi layered neural network for binary～", IJCNN'93 Nagoya, pp.327-330, Oct. 1993.
[6]T.Oshini et al,"Method for gradually reducing a number of hidden units on back propagation～", IEICE Trans., vol.J76-D-II, no.7, pp.1414-1424, July 1993.