Performance Improvement of Backpropagation Algorithm by Automatic Tuning of Learning Rate using Fuzzy Logic System

Kyung-Kwon Jung, Joong-Kyu Lim, Sung-Boo Chung and Ki-Hwan Eom, *Member, KIMICS*

Abstract—We propose a learning method for improving the performance of the backpropagation algorithm. The proposed method is using a fuzzy logic system for automatic tuning of the learning rate of each weight. Instead of choosing a fixed learning rate, the fuzzy logic system is used to dynamically adjust the learning rate. The inputs of fuzzy logic system are delta and delta bar, and the output of fuzzy logic system is the learning rate. In order to verify the effectiveness of the proposed method, we performed simulations on the XOR problem, character classification, and function approximation. The results show that the proposed method considerably improves the performance compared to the general backpropagation, the backpropagation with momentum, and the Jacobs' delta-bar-delta algorithm.

Index Terms—Backpropagation; Fuzzy logic system; Automatic tuning of learning rate; Delta-bar-delta

I. INTRODUCTION

The backpropagation (BP) algorithm has been the most popular, most widely implemented of all neural network paradigms. It is based on a multilayer, feedforward topology, and supervised learning. The BP learning rule adjusts interconnection weights and biases of the network so as to minimize the sum squared error. This is done by iteratively changing the values of the weights and biases in the direction of steepest descent with respect to error, reducing the error by a small amount each time. Once the sum squared error of the network has reached an acceptably small value or a maximum number of iterations have been performed, the training is complete [1].

The backpropagation algorithm has other drawbacks. One is the long learning time requirement. The second problem is the construction of the proper architecture. Third is the tendency of the steepest descent technique, which is used in the training process to get stuck at local minima. A number of research studies have attempted to overcome these problems during recent years[2,3]. The research falls roughly into two categories. The first category of research has focused on the use of standard numerical optimization techniques. Charalambous explained how the conjugate gradient algorithm could be used to train multilayer networks, Hagan et al. described the use of the Levenberg-Marquardt algorithm for training multiplayer networks, and Sanossian et al. proposed the inverse gradient method [4-6]. However most of these methods involve several parameters, complicate the algorithm, and require excessive memory. The order category involves the development of heuristic techniques, based on studies of the distinctive performance properties of the general backpropagation algorithm.

These heuristic techniques include such ideas as varying the learning rate, using momentum rescaling variables, and topology optimization. We suggested that we might be able to speed up convergence if we increase the learning rate on flat surfaces and then decrease the learning rate when the slope increases. The maximum stable learning rate for the steepest descent algorithm is two divided by the maximum eigenvalue of the Hessian matrix. There are many different approaches for varying the learning rate. Jacobs[7] described here the delta-bar-delta learning rule, in which each network parameter has its own learning rate that varies at each iteration. Vogl et al. proposed techniques for accelerating convergence of the backpropagation algorithm [8]. They include batching, momentum and variable learning rate. T.Tollenaere presents a variable learning rate backpropagation algorithm in which different learning rates are used for each weight.

In order to improve the performance of the backpropagation algorithm, we propose an automatic tuning method of learning rate of each weight. Proposed method is used a fuzzy logic system for automatic tuning of learning rate. Instead of choosing a fixed learning rate, the fuzzy logic system is used to dynamically adjust learning rate. The inputs of the fuzzy logic system are the delta and delta bar, and the output is the learning rate. In order to verify the effectiveness of the proposed method, and to compare it with the general backpropagation, the backpropagation with momentum and the Jacobs'delta-bar-delta algorithm, we simulate the XOR problem, character classification, and function approximation.

II. PROPOSED LEARNING METHOD

The backpropagation algorithm has been used mainly for the multi layer neural network. The backpropagation

Manuscript received July 8, 2003.

K. K. Jung is with the Department of Electronic Engineering, Dongguk University, Seoul, 100-715, Korea (phone: +82-2-2260-3332; fax: +82-2-2279-1798; e-mail: kwon@dongguk.edu).

J. K. Lim is with the Department of Electronic Engineering, Dongguk University, Seoul, 100-715, Korea (phone: +82-2-2260-3332; fax: +82-2-2279-1798; e-mail: tekcom77@hotmail.com).

S. B. Chung is with the Department of Electronic Engineering, Seoil College, Seoul, 131-702, Korea (phone: +82-2-490-7392; fax: +82-2-490-7387; e-mail: csbcsb@seoil.ac.kr).

K. H. Eom is with the Department of Electronic Engineering, Dongguk University, Seoul, 100-715, Korea (phone: +82-2-2260-3332; fax: +82-2-2279-1798; e-mail: kihwanum@dongguk.edu).

algorithm has other drawbacks, so Jacobs proposed a delta-bar-delta method for improving the performance of the backpropagation algorithm. Jacobs' delta-bar-delta learning rule is that each network parameter has its own learning rate that varies at each iteration[8]. The structure of multi layer neural network is shown in Fig.2-1. In the Fig. 2-1, i is input layer, j is hidden layer, and k is output layer.

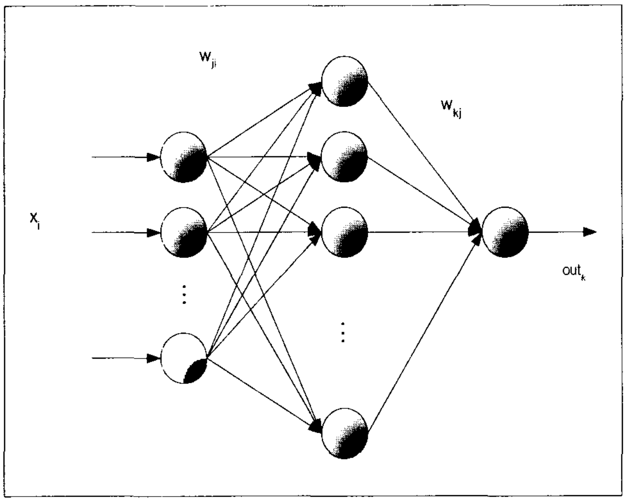


Fig. 2-1 The structure of multi-layer neural network.

Jacobs' delta-bar-delta algorithm get the change of weight as following formular[1,8].

$$\begin{aligned} w_{kj}(t+1) &= w_{kj}(t) - \alpha_{kj}(t+1)\frac{\partial E}{\partial w_{kj}} \\ &= w_{kj}(t) - \alpha_{kj}(t+1)\delta_k z_j \end{aligned} \quad (2\text{-}1)$$

where α_{kj} is the learning rate, δ_k is the error of output layer neuron, and z_j is the output of hidden layer neuron. The delta of each output layer neuron is equation (2-2), the delta of each hidden layer neuron is equation (2-3), and the delta bar of each output layer neuron is equation (2-4).

$$\Delta_{kj} = \frac{\partial E}{\partial w_{kj}} = -\delta_k z_j \quad (2\text{-}2)$$

$$\Delta_{ji} = \frac{\partial E}{\partial w_{ji}} = -\delta_j x_i \quad (2\text{-}3)$$

$$\overline{\Delta}_{kj}(t) = (1-\beta)\Delta_{kj}(t) + \beta\overline{\Delta}_{kj}(t-1) \quad (2\text{-}4)$$

where β is constant and has the value of $0 < \beta < 1$.

The learning rule of delta-bar-delta is varied a learning rate by the delta and delta bar. If the change of weight is the same directions while the learning process, the learning rate must be increased. It happens when $\overline{\Delta}_{kj}(t-1)$ and $\Delta_{kj}(t)$ is same sign. If the sign of $\overline{\Delta}_{kj}(t-1)$ and $\Delta_{KJ}(T)$ has opposition sign, the learning rate decreases by the ratio of the current value of $(1-\gamma)$. These variable learning rate are equation (2-5), (2-6) for each layer.

$$\begin{aligned} \alpha_{kj}(t+1) &= \alpha_{kj}(t) + \kappa && \text{if } \overline{\Delta}_{kj}(t-1)\cdot\Delta_{kj}(t) > 0 \\ &= (1-\gamma)\alpha_{kj}(t) && \text{if } \overline{\Delta}_{kj}(t-1)\cdot\Delta_{kj}(t) < 0 \\ &= \alpha_{kj}(t) && \text{if } \overline{\Delta}_{kj}(t-1)\cdot\Delta_{kj}(t) = 0 \end{aligned} \quad (2\text{-}5)$$

$$\begin{aligned} \alpha_{ji}(t+1) &= \alpha_{ji}(t) + \kappa && \text{if } \overline{\Delta}_{ji}(t-1)\cdot\Delta_{ji}(t) > 0 \\ &= (1-\gamma)\alpha_{ji}(t) && \text{if } \overline{\Delta}_{ji}(t-1)\cdot\Delta_{ji}(t) < 0 \\ &= \alpha_{ji}(t) && \text{if } \overline{\Delta}_{ji}(t-1)\cdot\Delta_{ji}(t) = 0 \end{aligned} \quad (2\text{-}6)$$

where κ, γ and β are constant.

The change of learning rate of output layer and hidden layer consists of fuzzy rule base of Table 2-1 by equation (2-5) and (2-6).

Table 2-1. Fuzzy rule base

$\overline{\Delta}$ \ Δ	N	Z	P
N	B	M	S
Z	M	M	M
P	S	M	B

Here N(Negative), Z(Zero), P(Positive), S(Small), M(Medium), and B(Big) are the linguistic variables [7]. The inputs are fuzzified according to the input membership functions shown in Fig. 2-2 and Fig. 2-3.

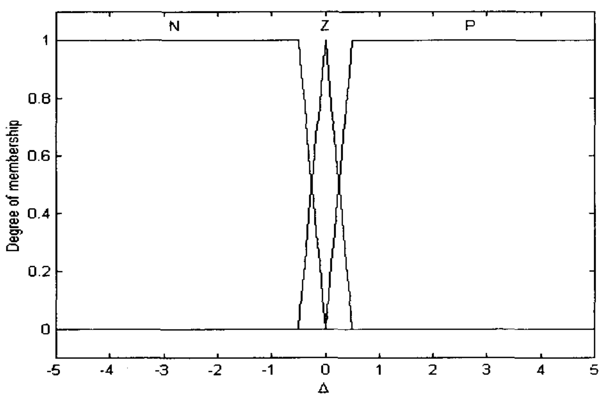


Fig. 2-2 The membership function of Δ.

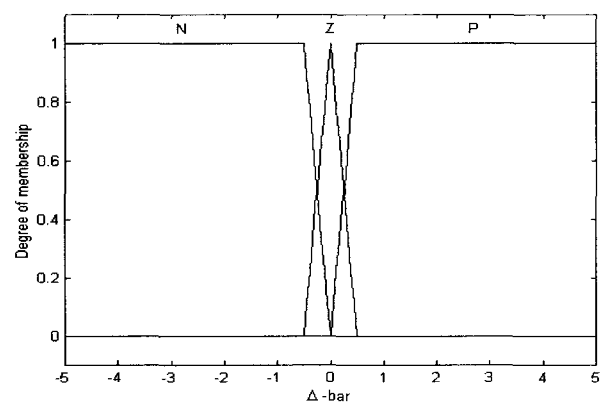


Fig. 2-3 The membership function of $\overline{\Delta}$.

The output of the fuzzy logic system is the learning rate, and shown in Fig. 2-4.

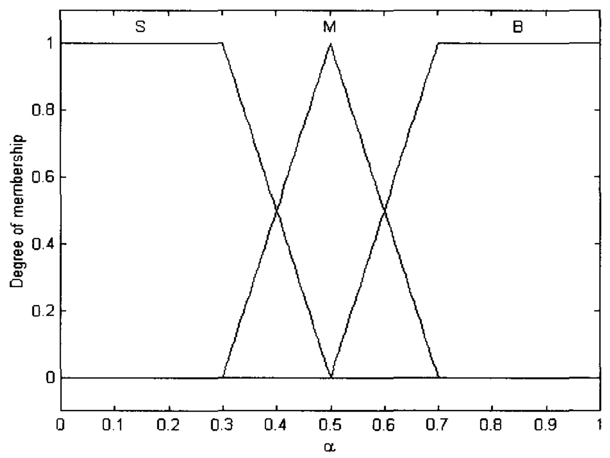


Fig. 2-4 The membership function of learning rate.

All rule base inference was accomplished using the Max-Min inference procedure. Defuzzification of the gain output was achieved through the center-of-gravity computation[9].

III. SIMULATION

Simulation studies were carried out for the XOR problem, function approximation, and character classification using the proposed method. The initial values for the weights and biases are 10 different values that are determined by the random numbers in [-0.1, 0.1]. The activation function uses the sigmoid function of equation (3-1).

$$f(net) = \frac{1}{1+e^{-net}} \tag{3-1}$$

A. XOR Problem

For the XOR problem is a popular benchmark for testing supervised learning algorithms. The network consists of two input neurons, two hidden neurons, and one output neuron. The training data are

$$\begin{aligned} &\{p_1 = \begin{bmatrix}0\\0\end{bmatrix}, t_1 = 0\}\{p_2 = \begin{bmatrix}0\\1\end{bmatrix}, t_2 = 1\} \\ &\{p_3 = \begin{bmatrix}1\\0\end{bmatrix}, t_3 = 1\}\{p_4 = \begin{bmatrix}1\\1\end{bmatrix}, t_4 = 0\} \end{aligned} \tag{3-2}$$

Learning is terminated when the sum of the squared errors between the outputs of the neural network and the target signals become smaller than 0.01. The number of learning epochs is limited to 5000, and 100 times simulation has been performed. Table 3-1 presents the parameters of the each learning method, which is selected by a trial and error method.

Table 3-1 Parameters

Learning Method \ Parameters	α	μ	κ	γ	β
BP	0.1				
BP with momentum	0.75	0.9			
Delta-bar-Delta	0.8		0.035	0.333	0.7
Proposed method					0.7

In the Table 3-1, BP is the general backpropagation algorithm, BP with momentum is the algorithm that used momentum to the BP, and Delta-bar-Delta is the Jacobs's delta-bar-delta algorithm. α is the learning rate, μ is the momentum coefficient, and κ, γ, β is the constant of delta-bar-delta.

Table 3-2 presents the simulation results.

Table 3-2 The simulation result of XOR problem

Learning Method	# Times Simulation	# Times Success	Average Epochs
BP	100	79	3873.23
BP with momentum	100	84	1297.35
Delta-bar-delta	100	87	271.02
Proposed method	100	100	147.34

The results show that the proposed method is 100% of the convergence success rate, and the average epochs decreased significantly. Fig. 3-1 shows the sum squared error curves of each learning method in 5000 epochs. We conformed that the proposed method is the fastest in the initial convergence speed, and is the least in the sum squared error compared to the other method.

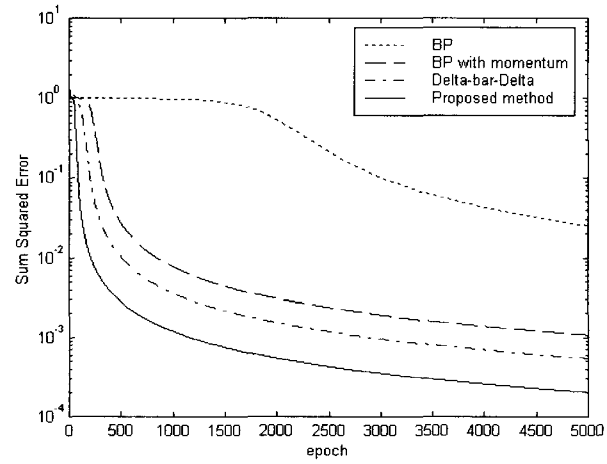


Fig. 3-1 The sum squared error curves.

B. Character classification

In this problem, the patterns of the letters shown in Fig. 3-2, were specified as the learning signals[10]. The input is used a row vector of 63 x 1 that has been converted from 9 x 7 matrix.

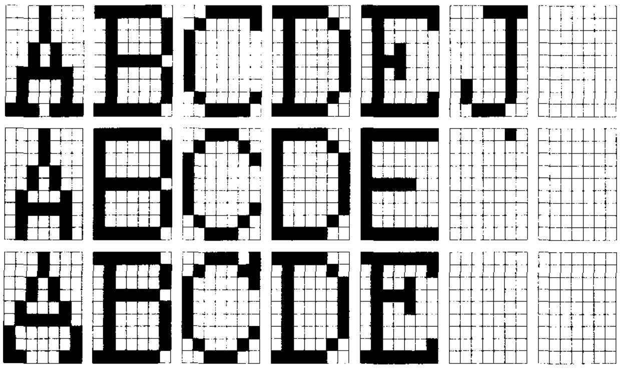


Fig. 3-2 Alphabet letters.

The target signals for the outputs were determined as in Table 3-3.

Table 3-3 The target signals

Letters	A	B	C	D	E	J	K
Binary Coding	0	0	0	0	1	1	1
	0	0	1	1	0	0	1
	0	1	0	1	0	1	0

The network consists of 63 input neurons, 10 hidden neurons, and 3 output neurons. Learning is terminated when the sum of the squared errors between the outputs of the neural network and the target signals become smaller than 0.01.

The number of learning epochs is limited to 5000, the sum of the squared errors become smaller than 0.01, and 100 times simulation has been performed. Table 3-4 presents the parameters of the each learning method which is selected by a trial and error, and Table 3-5 presents the simulation results.

Table 3-4 Parameters

Learning Method \ Parameter	α	μ	κ	γ	β
BP	0.5				
BP with momentum	0.75	0.9			
Delta-bar-Delta	0.8		0.035	0.333	0.7
Proposed method					0.7

Table 3-5 Simulation results

Learning Method	# Times Simulations	# Times Successes	Average Epochs
BP	100	0	>5000
BP with momentum	100	74	2839.10
Delta-bar-delta	100	85	833.33
Proposed method	100	100	592.81

As seen in the results, the proposed method was 100% successful, and the average epochs decreased significantly. On the other hand, there were failures using the BP method where attempts became trapped in local minimum. Fig. 3-3 is the sum squared error curves, and the proposed method shows that the initial convergence speed and the accuracy are superior.

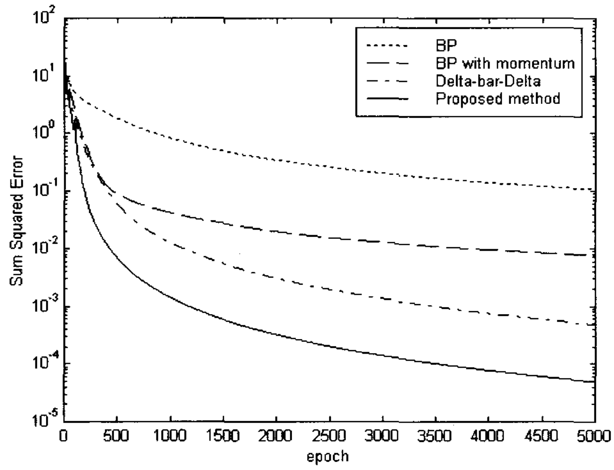


Fig. 3-3 The sum squared error curves.

C. Function Approximation

The network of the function approximation problem consists of one input neuron, ten hidden neurons, and one output neuron. The target function is a fig. 3-4[11].

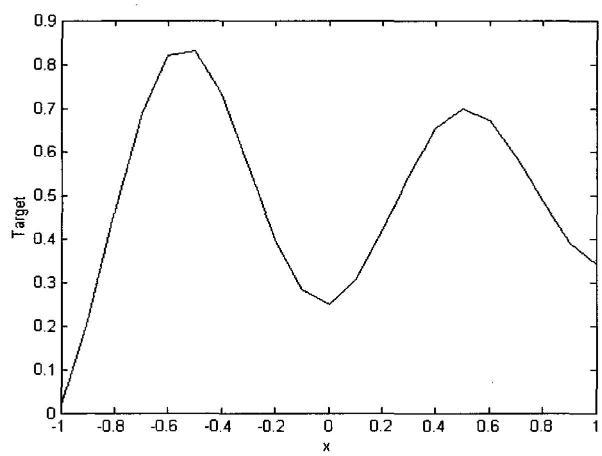


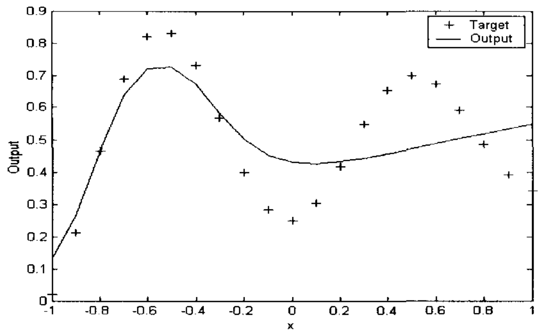
Fig. 3-4 Target function.

Table 3-6 presents the parameters of the each learning method which is selected by a trial and error.

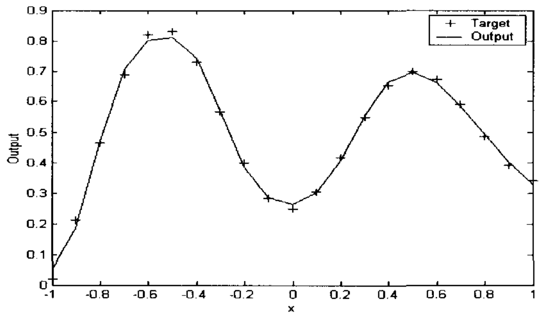
Table 3-6 Parameters

Learning Method \ Parameter	α	μ	κ	γ	β
BP	0.1				
BP with momentum	0.5	0.9			
Delta-bar-Delta	0.8		0.035	0.333	0.7
Proposed method					0.7

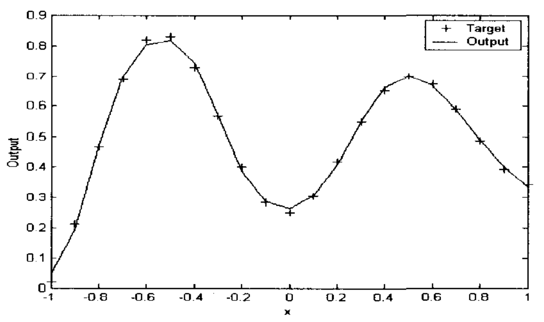
Fig. 3-5 show the simulation results for each learning method respectively.



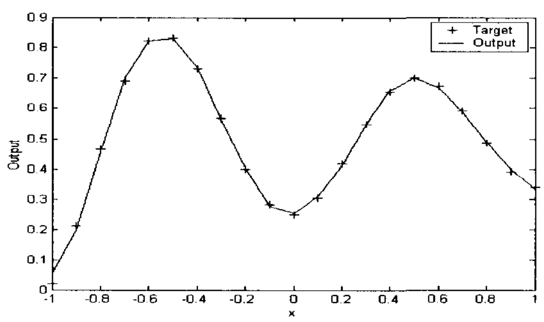
(a) BP



(b) BP with momentum method



(c) Delta-bar-delta method



(d) Proposed method

Fig. 3-5 Function approximation results.

The figure shows that the proposed method improves considerably on the accuracy of any other learning method. Fig. 3-6 is the sum squared error curves, and the proposed method shows that the initial convergence speed and the accuracy are superior.

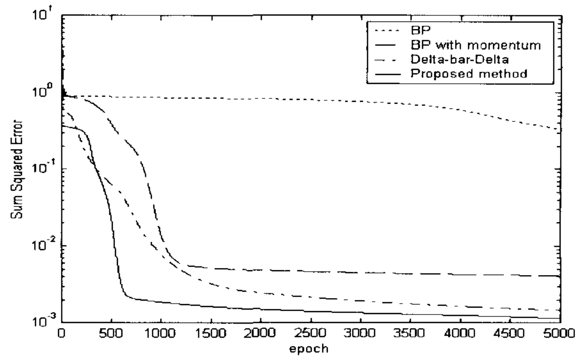


Fig. 3-6 The sum squared error curves.

IV. CONCLUSIONS

We proposed a learning method for improving the performance of the backpropagation algorithm using a fuzzy logic system for automatic tuning of the learning rate of each weight. Instead of choosing a fixed learning rate, the fuzzy logic system is used to dynamically adjust the learning rate. The inputs of fuzzy logic system are delta and delta bar, and the output of fuzzy logic system is the learning rate.

In order to verify the effectiveness of the proposed method, we performed simulations on the XOR problem, function approximation, and Arabic numerals classification. The results show that the proposed method has some advantages compared to the general backpropagation, the backpropagation with momentum, and the Jacobs'delta-bar-delta:

- The fuzzy logic system is simple because the fuzzy rules number just 9, and there is only one output of the system.
- The convergence of the XOR problem, character classification, and function approximation was 100% successful.
- The average epochs for the XOR problem, character classification, and function approximation were reduced greatly; this means significant reductions in computation time.
- The accuracy is improved notably in the function approximation problem.

REFERENCES

[1] Martin T. Hagan, Howard B. Demuth, Mark Beale, *Neural Network Design*, PWS Publishing, Boston, 1995.

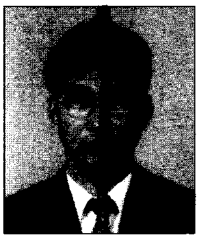
[2] Peiman G. Maghami and Dean W. Sparks, "Design of Neural Networks for Fast Convergence and Accuracy: Dynamics and Control", *IEEE Transactions on Neural Networks*, vol. 11, no. 11, pp. 113-123, 2000.

[3] Tokumitsu Fujita, Takao Watanebe and Keiichiro Yasuda, "A Study on Improvement in Learning Efficiency of Multilayered Neural Networks based on Dynamical System," *T.IEE Japan*, vol. 117-C, no. 12, pp. 1848-1855, 1997.

[4] C. Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks," *IEE Proceedings*, vol. 139, no. 3, pp. 301-310, 1992.
[5] M. T. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 187-199, 1994.
[6] H. Y. Y. Sanossian and D. J. Evans, "Gradient Range-Based Heuristic Method for Accelerating Neural Network Convergence," *Integrated Computer-Aided Engineering*, vol. 2, pp. 147-152, 1995.
[7] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Networks*, vol. 1, no. 4, pp. 295-308,1988.
[8] T.P. Vogl, J.K. Mangis, A.K. Zigler, W.T.Zink and D.L.Alkon, "Accelerating the convergence of the backpropagation method", *Biological Cybernetics.*, vol. 59, pp. 256-264, 1988.
[9] W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, John Wiley & Sons, Inc., 1992.
[10] Laurene Fausett, *Fundamentals of Neural Networks*, Prentice Hall, 1994.
[11] Howard Hemuth, Mark Beale, *Neural Network Toolbox User's Guide*, The Math Works, 1994.

Kyung-Kwon Jung
He received the B.S., M.S., and Ph.D. degrees in Electronic Engineering from Dongguk University, Seoul, Korea in 1998, 2000 and 2003 respectively. He is currently a research engineer in Digital Media Technology Research Center at Dongguk University. His research interests are in intelligent systems and digital signal processing.

Joong-Kyu Lim
He received the B.S. and M.S. degrees in Electronic Engineering from Dongguk University, Seoul, Korea in 1984 and 1999, respectively. He is currently working towards the Ph.D. degree in Electronic Engineering at Dongguk University. From 1983 to 1986, he joined at Samsung Electronic co. ltd. computer division R&D part. His research interests are in intelligent systems and image processing.



Sung-Boo Chung
He received the B.S., M.S., and Ph.D. degrees in Electronic Engineering from Dongguk University, Seoul, Korea in 1979, 1981 and 2002 respectively. Since 1987, he has been with Seoil College, where he is currently a Professor in the Department of Electronic Engineering. His research interests are in fuzzy, neural network and adaptive control.

Ki-Hwan Eom
He received the B.S. and M.S. and Ph.D. degree in electronic engineering from Dongguk University, Korea in 1972, 1975 and 1986, respectively. He was a visiting Professor from 1989 to 1990 and from 2000 to 2001 at Toho University and University of Cantervery. From 1978 to 1993, he was a Professor at Yuhan College. Since 1994, he has been with Dongguk University, where he is currently a Professor in the Department of Electronic Engineering. His research interests are in system application, digital signal processing, and biomedical signal processing.