A Modified Error Function to Improve the Error Back-Propagation Algorithm for Multi-Layer Perceptrons

Sang-Hoon Oh and Youngjik Lee

CONTENTS

ABSTRACT

This paper proposes a modified error function to improve the error back-propagation (EBP) algorithm for multi-Layer perceptrons (MLPs) which suffers from slow learning speed. It can also suppress over-specialization for training patterns that occurs in an algorithm based on a cross-entropy cost function which markedly reduces learning time. In the similar way as the cross-entropy function, our new function accelerates the learning speed of the EBP algorithm by allowing the output node of the MLP to generate a strong error signal when the output node is far from the desired value. Moreover, it prevents the overspecialization of learning for training patterns by letting the output node, whose value is close to the desired value, generate a weak error signal. In a simulation study to classify handwritten digits in the CEDAR [1] database, the proposed method attained 100% correct classification for the training patterns after only 50 sweeps of learning, while the original EBP attained only 98.8% after 500 sweeps. Also, our method shows mean-squared error of 0.627 for the test patterns, which is superior to the error 0.667 in the cross-entropy method. These results demonstrate that our new method excels others in learning speed as well as in generalization.

I. INTRODUCTION

The error back-propagation (EBP) algorithm [2] is widely used to train multi-layer perceptrons (MLPs) applied to many pattern classification problems. Training of MLPs is usually done by iterative updating of the weights to minimize the mean-squared error (m.s.e.) function. To update the weights of the output layer, one uses the error signal which is the difference between the desired and actual output values of MLP multiplied by the slope of the sigmoid activation function. The weights of the lower layer are updated based on the error signal back-propagated through the weights of the output or preceding layer. However, the EBP algorithm has a drawback of slow learning speed. During the learning process, the MLP goes through stages in which the reduction of the m.s.e. can be extremely slow [3], [4]. These periods of stagnation can influence learning times.

In pattern recognition applications, the desired output value of MLP is one of the two saturated values of the sigmoid function. When the weighted sum to any output node is in the saturation region which is opposite to the desired value, we say that the node is "incorrectly saturated."

Recently, there have been reports that the incorrect saturation of output nodes can cause the stagnation period [4]-[8]. When an output node is incorrectly saturated, the amount of weight change is small due to the small gradient of the sigmoid activation function, and the error remains nearly unchanged. In order to resolve this problem, Rezgui and Tepedelenlioglu attempted to adjust the slope of sigmoid activation function [6]. Also, Ooyen *et al*. proposed a cross-entropy error function in which the error signal associated with the output layer is proportional to the difference between the desired and actual output values [7]-[9]. However, we find that the simulation results of [8] still show the stagnation periods due to the saturation in the hidden layer. Furthermore, in our simulations, the cross-entropy method suffers from the over-specialization for training patterns since the error signal is too strong at the final stage of learning. Besides the cross-entropy error function, many other error functions were proposed to improve the generalization performance of MLPs [10]-[14]. However, none of these functions resolved the incorrect saturation problem of output nodes. The probability of the incorrect saturation of output nodes in the first sweep of learning shows that initialization of MLP with small random weights can avoid the stagnation period due to the saturation in the output layer as well as in the hidden layer [4]. Even in this case, the possibility of incorrect saturation still exists during the learning process.

In this paper, we address the problems of learning and generalization. Since the final goal of pattern recognition is to achieve good generalization capability, the learning should not be specialized too much for the training patterns. At the same time, the learning should be fast for practical implementation of pattern recognition systems. For this reason, we pro-

pose an improved EBP algorithm for MLPs by allowing the output nodes of the MLP to generate an appropriate error signal according to the situation of output nodes. When some output nodes of MLP are incorrectly saturated, the strong error signal of the output nodes updates the associated weights so that they can escape the incorrectly saturated state. This can accelerate the learning speed. For the correctly saturated output nodes, the weak error signal prevents the overspecialization of learning for training patterns. In Section II, we propose a modified error function for the EBP algorithm. In Section III, we show the effectiveness of the modified error function through the simulation of handwritten digit recognition using the CEDAR [1] database, and Section IV concludes this paper.

II. MODIFIED ERROR FUNCTION

First, we describe the conventional EBP algorithm. Consider the MLP consisting of L layers in which each l-th layer has N_l nodes. Let the state vector of nodes in layer l be $\mathbf{x}^{(l)} = [x_1^{(l)}, x_2^{(l)}, \ldots, x_{N_l}^{(l)}]$, and $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(L)}$ be the input and output vectors, respectively. Here, $x_j^{(l)}, l \neq 0$, has value between -1 and $+1$. Also, let the desired output vector corresponding to a training pattern be $\mathbf{t} = [t_1, t_2, \ldots, t_{N_L}]$. When an input pattern $\mathbf{x}^p$ is presented to the network and propagated forward to determine the output signal, the state $x_j^{(l)}$ in each l-th layer is

$$x_j^{(l)} = f(a_j^{(l)}) = \frac{2}{1+\exp[-a_j^{(l)}]} - 1 \qquad (1)$$

where

$$a_j^{(l)} = w_{j0}^{(l)} + \sum_{i=1}^{N_{l-1}} w_{ji}^{(l)} x_i^{(l-1)}. \qquad (2)$$

Here, $w_{ji}^{(l)}$ denotes the weight connecting $x_i^{(l-1)}$ to $x_j^{(l)}$, and $w_{j0}^{(l)}$ denotes the bias to $x_j^{(l)}$.

The conventional m.s.e. function [2] is

$$E_m(\mathbf{x}^p) = \frac{1}{2}\sum_{k=1}^{N_L}(t_k - x_k^{(L)})^2. \qquad (3)$$

To minimize $E_m(\mathbf{x}^p)$, each weight is updated by an amount proportional to the partial derivative of $E_m(\mathbf{x}^p)$ with respect to the weight. Therefore, we update the weight $w_{kj}^{(L)}$ of the output layer using

$$\Delta w_{kj}^{(L)} = -\eta \frac{\partial E_m}{\partial a_k^{(L)}} \frac{\partial a_k^{(L)}}{\partial w_{kj}^{(L)}} = \eta \delta_k^{(L)} x_j^{(L-1)}, \qquad (4)$$

where

$$\delta_k^{(L)} = -\frac{\partial E_m}{\partial a_k^{(L)}} = (t_k - x_k^{(L)}) \frac{(1-x_k^{(L)})(1+x_k^{(L)})}{2} \qquad (5)$$

is the error signal and η is the learning rate. We also update the weight $w_{ji}^{(l)}$ below the output layer using

$$\Delta w_{ji}^{(l)} = \eta \delta_j^{(l)} x_i^{(l-1)} \qquad (6)$$

where

$$\delta_j^{(l)} = \frac{(1-x_j^{(l)})(1+x_j^{(l)})}{2} \sum_{k=1}^{N_{l+1}} w_{kj}^{(l+1)} \delta_k^{(l+1)}. \qquad (7)$$

Here, $\delta_j^{(l)}$ is the error signal back-propagated through the weights of preceding layer. Also,

the m.s.e. function for all P training patterns is

$$E_m = \frac{1}{P}\sum_{p=1}^{P} E_m(\mathbf{x}^p) \tag{8}$$

and we can minimize E_m through the iterative updates of weights for all training patterns.

In the above EBP algorithm, the error signal $\delta_k^{(L)}$ in (5) is the difference $(t_k - x_k^{(L)})$ multiplied by the gradient of the sigmoid function. If $x_k^{(L)}$ approaches one of the two extreme values, the gradient factor in (5) makes the error signal very small. Thus, the output node $x_k^{(L)}$ which has extreme value against t_k cannot produce a strong error signal [4], [8]. This incorrect saturation retards the search for a minimum in the error surface.

In order to accelerate the EBP algorithm, one can use the cross-entropy error function[7], [8]

$$E_c(\mathbf{x}^p) = -\sum_{k=1}^{N_L}[(1+t_k)\ln(1+x_k^{(L)}) + (1-t_k)\ln(1-x_k^{(L)})]. \tag{9}$$

Using the above error function, the error signal in (5) becomes

$$\delta_k^{(L)} = t_k - x_k^{(L)} \tag{10}$$

and the other equations for updating the weights are the same as the ones in the EBP algorithm. Thus, the output nodes can escape well from the state of incorrect saturation, since the associated weights of the output layer are adjusted proportional to the difference $(t_k - x_k^{(L)})$. However, the difference error signal will make the MLP specialized too much for training patterns, since the error signal is relatively strong when $x_k^{(L)}$ approaches t_k, as shown in Fig. 1(b).

During the learning process, the direction of weight update for reducing error associated with a specific training pattern will assist or compete with that for reducing total error [3], [8]. For instance, some of the output nodes are pushed towards the wrong extreme value by competition in the network. In this case, a strong error signal is necessary for the incorrectly saturated output node to escape the wrong extreme value, like the cross-entropy method. For the correctly saturated output node, a weak error signal needs to be generated so that the weight update associated with the training pattern can scarcely perturb the weights trained for all training patterns [15]. These strong and weak error signals according to the situation of output nodes can minimize the competition. The weak error signal is also necessary to prevent the overspecialization of learning for the training patterns.

In this sense, we propose the modified error function

$$E_l(\mathbf{x}^p) = -\sum_{k=1}^{N_L} t_k[-x_k^{(L)} + \frac{1+t_k}{2}\ln\frac{1+x_k^{(L)}}{1-x_k^{(L)}} + t_k\ln(1-x_k^{(L)})(1+x_k^{(L)})]. \tag{11}$$

Using the above error function, the error signal is

$$\delta_k^{(L)} = \frac{t_k(t_k - x_k^{(L)})^2}{2} \tag{12}$$

and the other equations for updating weights are the same. Figure 1 shows the error function and error signal in each method. As shown

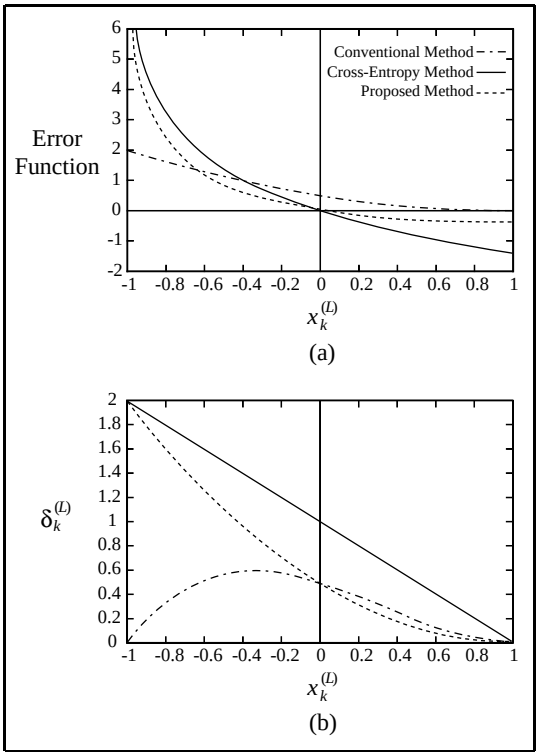

Fig. 1. The error functions and error signals in the three learning methods when $t_k = 1$.

in Fig. 1(b), the modified error signal can satisfy the above criteria which request a strong error signal for the incorrectly saturated output node and a weak error signal for the correctly saturated output node. After all, we can reduce the incorrect saturation of output nodes due to the competition, and prevent the overspecialization for training patterns.

When the MLP parameters are chosen to minimize the conventional m.s.e. or cross-entropy error function, the outputs estimate the conditional expectations of the desired outputs [9]. Using the same method as in [9], we can show that the modified error function has the same property. We verify the efficiency of the proposed method through a handwritten digit recognition, which is described in the next section.

III. SIMULATION RESULTS

A handwritten digit recognition problem is used to compare the conventional EBP method, the cross-entropy method, and our proposed method. A total of 2,308 handwritten digitized images from the CEDAR database [1] are used for training after size normalization. A digit image consists of 12×12 pixels and each pixel is represented by one of 16 grey levels. We use an MLP architecture that consists of 144 inputs, 30 hidden nodes, and 10 output nodes for training. We use local coding for the target pattern.

Since no fair comparison is possible if the learning rate is kept the same for all three methods [8], we derive the learning rates so that the expectation value of $\eta\delta_k^{(L)}$ has the same value in each method. Here, we assume that $x_k^{(L)}$ has uniform distribution between -1 and $+1$. As a result, the learning rates of 0.06, 0.02, and 0.03 are used for the conventional EBP method, the cross-entropy method, and the proposed method, respectively. Nine simulations are conducted using each method with the same initializations and the results are averaged. The initial weights were selected at random from an uniform distribution between -1×10^{-4} and 1×10^{-4}. Figure 2 shows the misclassification ratio and m.s.e. for the train-

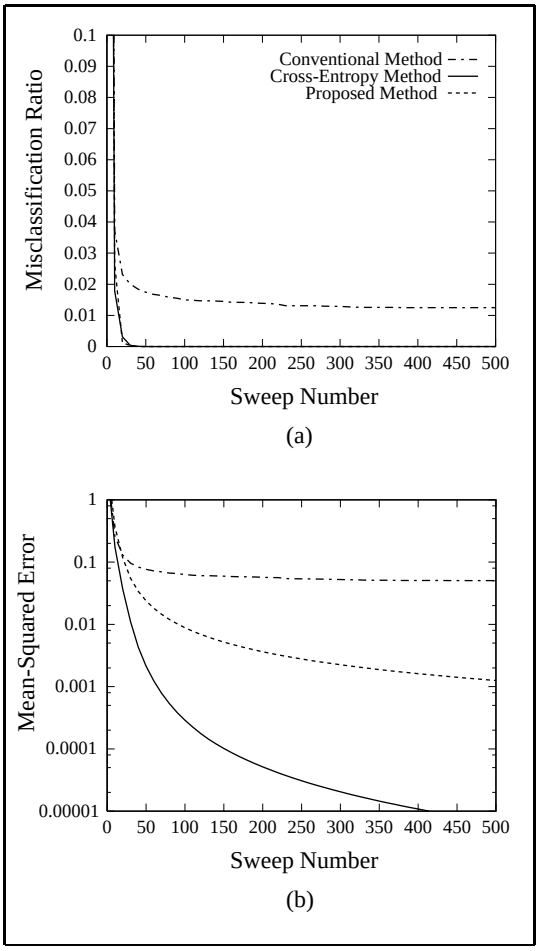

Fig. 2. The misclassification ratio and the mean-squared error for training patterns using the conventional EBP method, the cross-entropy method, and the proposed method.

ing patterns at each learning sweep using the three methods. The cross-entropy method accelerates the reduction of m.s.e. and achieves the perfect classification for the training patterns, since the output nodes escape the incorrect saturation during learning as shown in Fig. 3. The proposed method performs the same during learning. Furthermore, our method has a very small incorrect saturation ratio in the initial stage of learning (see Fig. 3, dashed line), as expected. On the contrary, the conventional EBP algorithm cannot achieve good performance for the training patterns, since some output nodes cannot escape the incorrect saturation through the conventional EBP learning, as shown in Fig. 3.

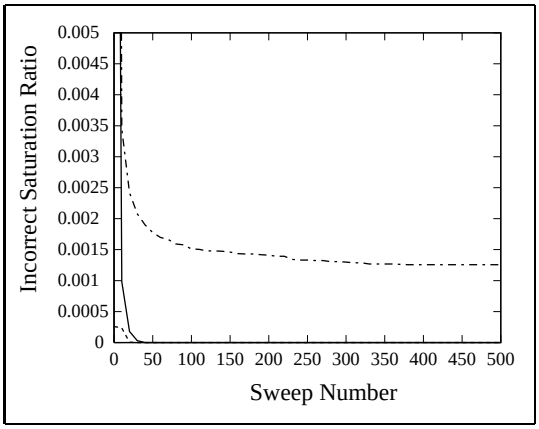

Fig. 3. The incorrect saturation ratio of output nodes for training patterns using the three learning methods.

In the cross-entropy method, the m.s.e. for the training patterns decreases very sharply in the final stage of learning. On the contrary, the simulation results for 2,213 test patterns are the worst as shown in Fig. 4. This implies that the MLP is specialized too much for the training patterns. In the proposed method, the error signals become small when the output values are near the target values and the overspecialization as in the cross-entropy method does not occur. Thus, the simulation results for the test patterns using the proposed method are superior to those using the cross-entropy method.

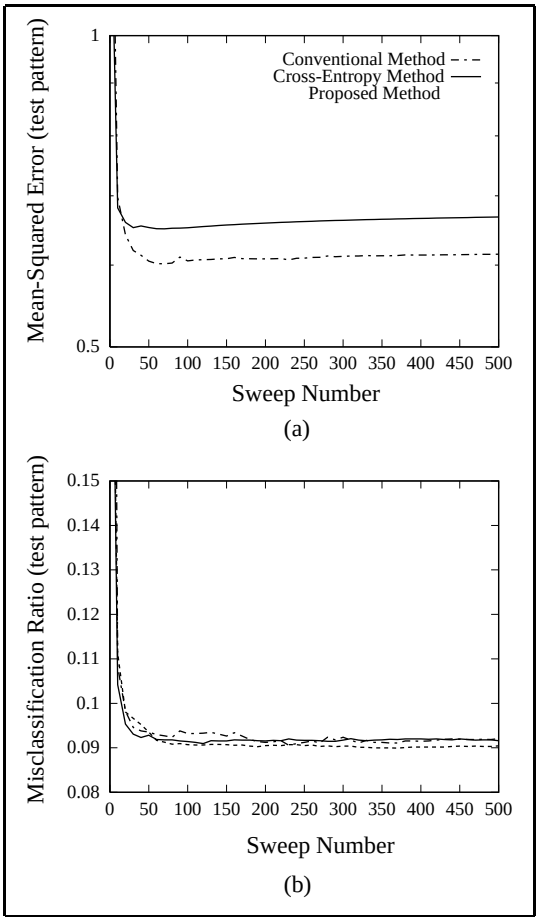

Fig. 4. The misclassification ratio and m.s.e. for test patterns using the three learning methods.

In order to show the effectiveness of our method for complex problems, we simulate the handwritten digit recognition problem with a total of 18,468 training images after size normalization. Increasing the number of training patterns may make the classification problem more complex. Therefore, very small learning rates of 0.006, 0.002, and 0.003 are used for the conventional EBP method, the cross-entropy method, and the proposed method, respectively. As shown in Figure 5, no method can attain 0% incorrect saturation ratio during learning process. However, the proposed method shows the lowest ratio of incorrect saturation. Also, it attains 99.8% correct classification ratio for the training patterns and 94.7% for the 2,213 test patterns at 300 sweep, which are superior to those of the others. Thus, our new method excels the others in incorrect saturation of output nodes as well as in generalization, especially for complex problems.

For a further comparison, we propose an adaptive learning rate at each sweep n as

$$\eta(n) = \eta_o \sqrt{\frac{E[(t_k(n) - x_k^{(L)}(n))^2]}{E[\delta_k^{(L)\,2}(n)]}}. \qquad (13)$$

Here, η_o is the learning rate at the first sweep, and $E[(t_k(n) - x_k^{(L)}(n))^2]$ and $E[\delta_k^{(L)\,2}(n)]$ are the expectation values considering all of the output nodes and training patterns in the n-th sweep. Then, the expected intensity of $\eta(n)\delta_k^{(L)}(n)$ which determines the amount of weight update is

$$E[\eta^2(n)\delta_k^{(L)\,2}(n)] = \eta_o^2 E[(t_k(n) - x_k^{(L)}(n))^2]. \qquad (14)$$

Thus, we can use the functional characteristic of $\delta_k^{(L)}$ for updating weights while the expected intensity of $\eta(n)\delta_k^{(L)}$ has the same value as that in the cross-entropy method. Here, the adaptation of learning rate does not affect the cross-entropy learning method because the error signal $\delta_k^{(L)}$ generated by the cross-entropy error function is $(t_k - x_k^{(L)})$. However, in simulation of real problems, we cannot estimate $E[(t_k(n) - x_k^{(L)}(n))^2]$ and $E[\delta_k^{(L)\,2}(n)]$ at

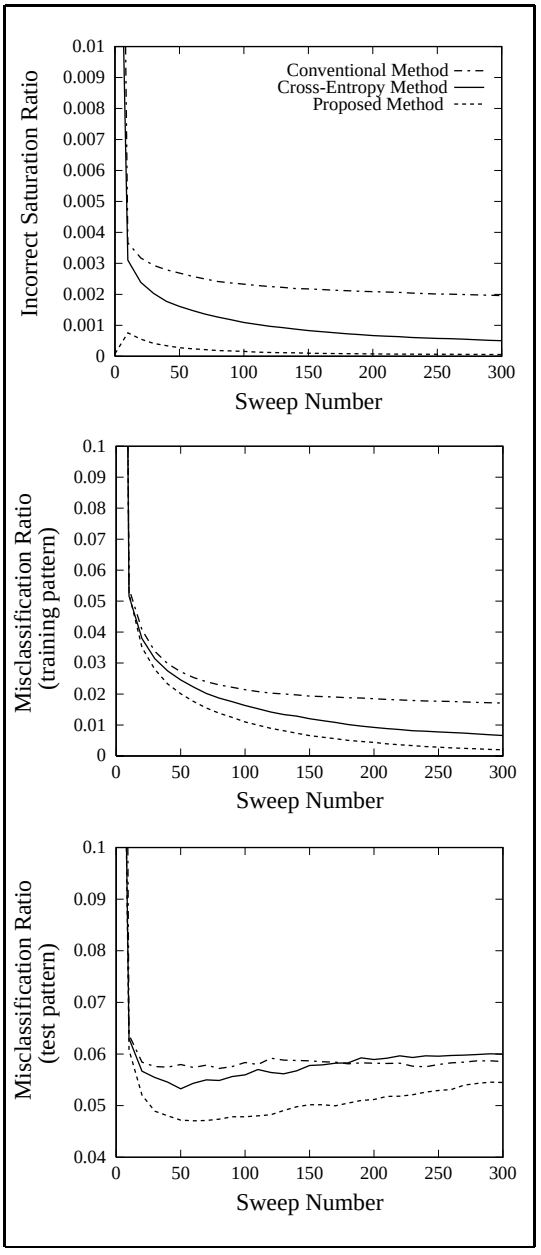

Fig. 5. The incorrect saturation ratio, misclassification ratio for training patterns, and misclassification ratio for test patterns using the three learning methods with 18,468 training patterns.

the beginning of sweep n. Therefore, we use the two expected values estimated through the $(n-1)$-th sweep to calculate $\eta(n)$.

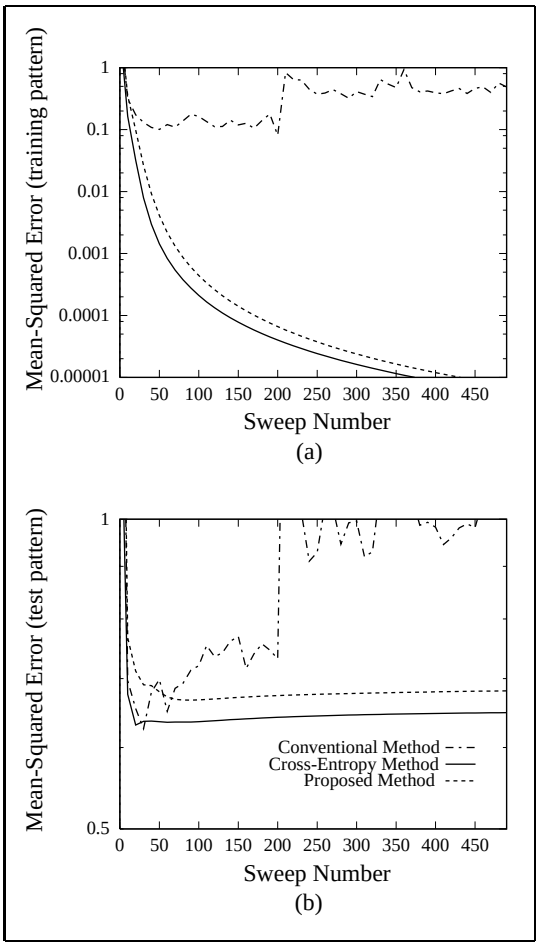

Fig. 6. The m.s.e. for training and test patterns of the three EBP algorithms with adaptive learning rates.

Figure 6 shows the averaged results of four simulations using the three learning methods with the adaptive learning rate. Here, the initial learning rate η_o is 0.02 and the initial weights are drawn uniformly in $[-1 \times$

$10^{-4}, +1\times10^{-4}]$. Also, the 2,308 training patterns and 2,213 test patterns are used for evaluation of learning. In the proposed method, the adaptive learning rate accelerates the final convergence of m.s.e. for the training patterns (Fig. 6(a)). However, in Fig. 6(b), the proposed method with the adaptive learning rate shows worse generalization capacity than that with the fixed learning rate 0.03 (Fig. 4(a), dashed line), since the adaptive learning rate makes the error signal strong when $x_k^{(L)}$ approaches t_k. This fact indirectly supports the previous argument that the cross-entropy method overspecializes the MLP for the training patterns because of the strong error signal, while the proposed method does not.

Although the adaptive learning rate accelerates the decrement of the error, it makes variation in the error curve. The variation appears especially severe in the conventional EBP algorithm. For explanation of the variation, Fig. 7 shows the m.s.e. per output node, the adaptive learning rate, and the incorrect saturation ratio for the training patterns in a simulation using the conventional EBP algorithm. When the m.s.e. decreases, the adaptive learning rate $\eta(n)$ increases according to (13). The enlarged $\eta(n)$ accelerates the decrement of the error. However, the enlarged $\eta(n)$ amplifies the possibility of competition. If some output nodes are saturated incorrectly by the competition, the m.s.e. increases and $\eta(n)$ decreases. Thus, the positive feedback between the error decreasing and the learning rate, and the negative feedback from the incorrect saturation to the learning rate make variation in the error curve.

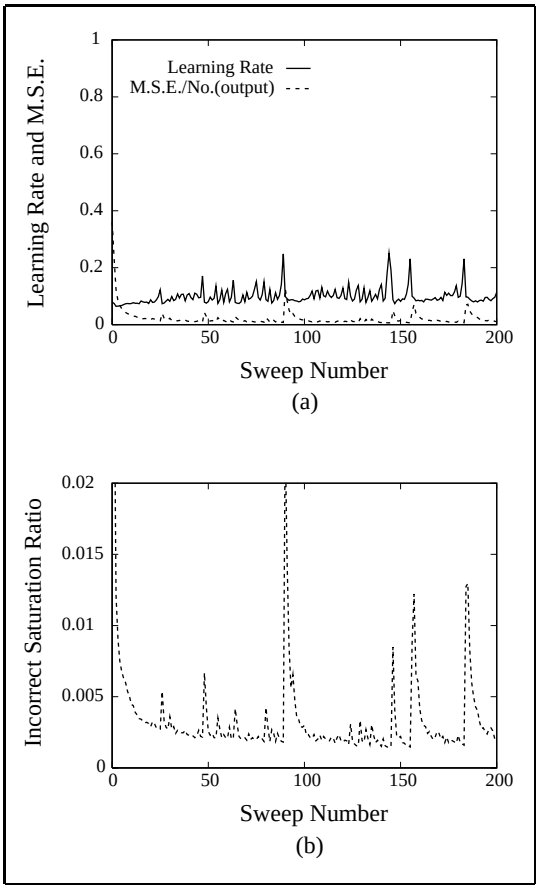

Fig. 7. A simulation result for training patterns using the conventional EBP algorithm with adaptive learning rate.

Figure 8 shows a similar simulation result with the proposed method. Here, the competition is minimized and the output node can escape the incorrect saturation well. As a result, the variation is terminated after some periods of learning and the positive feedback is permanent.

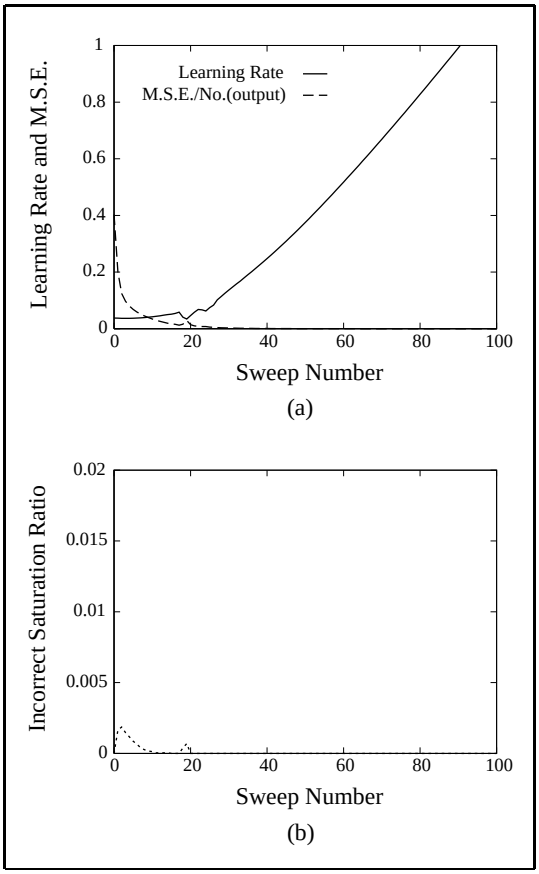

Fig. 8. A simulation result for training patterns using the proposed method with adaptive learning rate.

IV. CONCLUSION

In this paper, we proposed a modified error function of EBP algorithm to resolve the slow learning and specialization problem in pattern recognition applications. Using the modified error function, the error signal can be represented by a square function of the difference between the desired and actual output values. This accelerates the learning speed of EBP algorithm through the effective elimination of the incorrect saturation, and prevents the overspecialization of learning for the training patterns.

We have compared the proposed method with the conventional EBP algorithm and cross-entropy method through the simulation of classifying handwritten digits in the CEDAR [1] database. The simulation results showed that, although the cross-entropy method markedly reduced learning time, it tended to overspecialize the MLP for the training patterns, resulting poor generalization capability for the test patterns. On the contrary, the conventional EBP algorithm did not show good performance for the training patterns although it showed better generalization capacity for the test patterns. Compared to these methods, our approach showed good performance for both the training and test patterns especially in complex problems, since it effectively eliminated the incorrect saturation and prevented the overspecialization during learning.

ACKNOWLEDGMENT

This research has been partly funded by Korea Telecom and the Ministry of Information and Communications, Korea. The authors wish to thank Dr. El-Hang Lee for his continuing support and guidance, and Dr. Rhee Man Kil for his helpful discussions. Also, the authors wish to thank the reviewers of this paper for their helpful criticisms.

REFERENCES

[1] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pat. Ana. March. Int.*, accepted to appear as a correspondance.

[2] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing.* Cambridge, MA : MIT Press, 1986.

[3] R. K. Cheung, I. Lustig, and A. L. Kornhauser, "Relative effectiveness of training set patterns for back propagation," *Proc. IJCNN*, Jan. 15-19, 1990, Washington, DC, USA, vol. I, pp. 673-678.

[4] Y. Lee, S.-H. Oh, and M. W. Kim, "An analysis of premature saturation in backpropagation learning," *Neural Networks*, vol. 6, pp. 719-728, 1993.

[5] J. R. Chen and P. Mars, "Stepsize variation methods for accelerating the backpropagation algorithm," *Proc. IJCNN*, Jan. 15-19, 1990, Washington, DC, USA, vol. I, pp. 601-604.

[6] A. Rezgui and N. Tepedelenlioglu, "The effect of the slope of the activation function on the back propagation algorithm," *Proc. IJCNN*, Jan. 15-19, 1990, Washington, DC, USA, vol. I, pp. 707-710.

[7] A. Krzyzak, W. Dai, and C. Y. Suen, "Classification of large set of handwritten characters using modified back propagation model," *Proc. IJCNN*, June 17-21, 1990, San Diego, USA, vol. III, pp. 225-232.

[8] A. van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, vol. 5, pp. 465-471, 1992.

[9] M. D. Richard and R. P. Lippmann, "Neural network classifier estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, pp. 461-483, 1991.

[10] B. A. Telfer and H. H. Szu, "Energy functions for minimizing misclassification error with minimum-complexity networks," *Neural Networks*, vol. 7, pp. 809-818, 1994.

[11] H. H. Szu and F. Lu, "A double-well potentional energy function resolving bistable ambiguity figures by neural networks," *Proc. WCNN*, June 5-9, 1994, San Diego, USA, vol. III, pp. 426-431, 1994.

[12] R. Kamimura and S. Nakanishi, "Maximum entropy principle: Improving generalization performance by maximizing the number of internal representations," *Proc. ICONIP*, Oct. 17-20, 1994, Seoul, Korea, vol. I, pp. 207-212.

[13] J. S. N. Jean and J. Wang, "Weight smoothing to improve network generalization," *IEEE Trans. Neural Networks*, vol. 5, pp. 752-763, Sept. 1994.

[14] S.-Y. Lee and D.-G. Jeong, "Hybrid Hebbian/back-propagation learning rule for improved generalization of multilayer feed-forward neural networks," *Proc. ICONIP*, Oct. 17-20, 1994, Seoul, Korea, vol. I, pp. 189-194.

[15] Y. Lee and S.-H Oh, "Improving the error back-propagation algorithm," *Proc. ICONIP*, Oct. 17-20, 1994, Seoul, Korea, vol. II, pp. 772-777.

Sang-Hoon Oh received the B. S. and M. S. degrees in electronics engineering from Pusan Nationual University, Pusan, Korea in 1986 and 1988, respectively. From 1988 to 1989, he was with Goldstar Semiconductor, Ltd., Korea where he was involved in quality control of MOS FAB. Since 1990, he has been with Research Department of ETRI, Taejon, Korea, pursuing interests in theories, implementations, and applications of neural networks.

Youngjik Lee received the B. S. degree in electronics engineering from Seoul National University, Seoul, Korea in 1979, the M. S. degree in electrical engineering from Korea Advanced Institute of Science, Seoul, Korea in 1981, and the Ph.D. degree in electrical engineering from the Polytechnic University, Brooklyn, New York, U.S.A.

From 1981 to 1985 he was with Samsung Electronics Company, Suwon, Korea where he was involved in the development of video display terminal. From 1985 to 1988 his research topic was concentrated on the theories and applications of sensor array signal processing. His dissertation was on the direction finding from first order statistics and spectrum estimation. From 1989 to 1993, he was with Research Department of ETRI, Taejon, Korea pursuing interests in theories, implementations, and applications of neural networks, digital signal processing, and pattern recognition. Since 1994, he has been with Spoken Language Processing Section of ETRI, doing researches in speech recognition, speech synthesis, and spoken language translation.