An Adaptive Learning Rate with Limited Error Signals for Training of Multilayer Perceptrons

Sang-Hoon Oh and Soo-Young Lee

Although an *n*-th order cross-entropy (nCE) error function resolves the incorrect saturation problem of conventional error backpropagation (EBP) algorithm, performance of multilayer perceptrons (MLPs) trained using the nCE function depends heavily on the order of nCE. In this paper, we propose an adaptive learning rate to markedly reduce the sensitivity of MLP performance to the order of nCE. Additionally, we propose to limit error signal values at output nodes for stable learning with the adaptive learning rate. Through simulations of handwritten digit recognition and isolated-word recognition tasks, it was verified that the proposed method successfully reduced the performance dependency of MLPs on the nCE order while maintaining advantages of the nCE function.

Manuscript received October 8, 1999 ; revised July 8, 2000.

The authors are with the Department of Electrical Engineering, Brain Science Research Center, KAIST, Taejon, Korea.
Sang-Hoon Oh (phone: +82 42 869 5431, e-mail: shoh@neuron.kaist.ac.kr)
Soo-Young Lee (phone: +82 42 869 3431, e-mail: sylee@ee.kaist.ac.kr)

I. INTRODUCTION

Multilayer perceptron (MLP) is the most popular neural network model which has wide application areas such as mobile telecommunications [1], [2], ATM networks [3], [4], pattern recognition [5], speech recognition [6], time series prediction [7], and nonlinear control [8]. Especially theoretical analyses of MLPs in mathematical or statistical aspects support the applications and research efforts for MLPs [9]–[12].

Training of MLPs is usually done by the error backpropagation (EBP) algorithm [13], in which weights are iteratively updated according to the negative gradient of the mean-squared error (MSE) function, so called "error signal." In the output layer, the error signal is the difference between desired and actual output values multiplied by the gradient of sigmoid activation function. Then, the error signal is back-propagated to hidden layers.

The EBP algorithm, however, has a drawback with slow learning speed due to an incorrect saturation of output nodes. When an output node is incorrectly saturated during learning, the amount of weight change is small due to the small gradient of sigmoid activation function and the error remains nearly unchanged [14]–[16].

In order to resolve this problem, van Ooyen and Nienhuis proposed the cross-entropy (CE) error function which removed the gradient of the sigmoid function at the error signal of output nodes [17]. As an extended formulation of the CE function, the *n*-th order cross-entropy (nCE) error function was proposed to resolve the incorrect saturation problem as well as to prevent overfitting of MLPs for training patterns [18]. This was achieved by generating a strong error signal for incorrectly saturated output nodes and a weak error signal for correctly saturated output nodes. However, performance of the trained MLPs depends heavily on the order of nCE function and one should find an

optimum order of the nCE function to obtain good training results of MLPs with fast learning speed.

This paper proposes an adaptive learning rate to make the performance of MLPs insensitive to the order of nCE error. The proposed adaptive learning rate complements the variation of error signals on the order of nCE error. Additionally, it is proposed to limit error signal values of output nodes to prevent unstable characteristic of learning due to the adaptive learning rate. There are many techniques for adapting learning rates of EBP algorithm such as the bold driver [19], [20], delta-bar-delta [21], and optimum learning rates [22]. Although these accelerate the EBP algorithm, they do not have ability to reduce the performance dependency of MLPs on the order of nCE error functional.

This paper is organized as follows. Section II briefly reviews the nCE error function for EBP algorithm. Section III points out the performance variation of MLPs on the order of nCE and describes an adaptive learning rate with limited error signals to make MLPs insensitive to the order of nCE. In section IV, the effectiveness of the proposed method is demonstrated in handwritten digit recognition and isolated-word recognition tasks. Finally, Section V concludes this paper.

II. *n*-TH ORDER CROSS-ENTROPY ERROR

Consider an 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 $\mathbf{x}^{(l)} = [x_1^{(l)}, x_2^{(l)}, \cdots, x_{N_l}^{(l)}]$, and $\mathbf{x}^{(0)}$ and $\mathbf{x}^{(L)}$ be the input and output vectors, respectively. Here, $x_j^{(l)} (l \neq 0)$ has a value between -1 and 1. Also, let the desired output vector corresponding to a training pattern $\mathbf{x}$ be $\mathbf{t} = [t_1, t_2, \cdots, t_{N_L}]$. When $\mathbf{x}$ is presented to the network, the state $x_j^{(l)}$ in the l-th layer is

$$x_j^{(l)} = \tanh\left(\frac{w_{j0}^{(l)} + \sum_{i=1}^{N_{l-1}} w_{ji}^{(l)} x_i^{(l-1)}}{2}\right), \quad l = 1, 2, \cdots, L. \tag{1}$$

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 MSE function [13] is

$$E_m(\mathbf{x}) = \sum_{j=1}^{N_L} (t_j - x_j^{(L)})^2 / 2. \tag{2}$$

To minimize $E_m(\mathbf{x})$, each weight is updated by

$$\Delta w_{ji}^{(l)} = \eta \delta_j^{(l)} x_i^{(l-1)}. \tag{3}$$

Here,

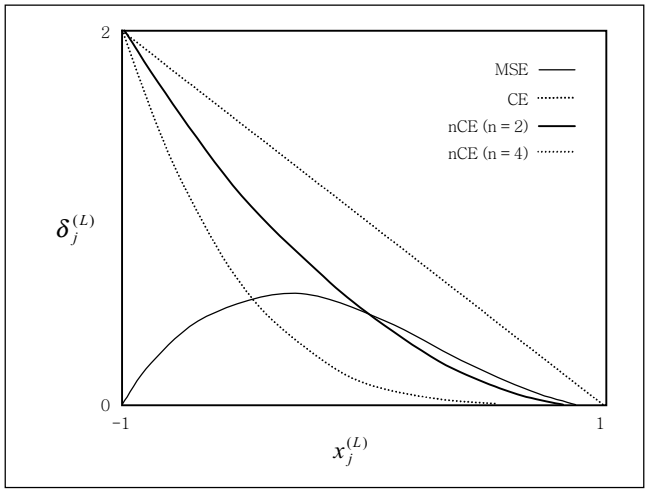


Fig. 1. The error signal of output node with $t_j = 1$. $x_j^{(L)}$ is the j-th output value and $\delta_j^{(L)}$ is the error signal of $x_j^{(L)}$.

$$\delta_j^{(l)} = \begin{cases} \dfrac{(1 - x_j^{(L)})(1 + x_j^{(L)})}{2}(t_j - x_j^{(L)}), \text{ where } l = L, \\ \dfrac{(1 - x_j^{(l)})(1 + x_j^{(l)})}{2} \sum_{k=1}^{N_{l+1}} w_{kj}^{(l+1)} \delta_k^{(l+1)}, \text{ where } 1 \le l \le L\text{-}1 \end{cases} \tag{4}$$

is the error signal and η is the learning rate.

In the above EBP algorithm, the output node $x_j^{(L)}$ which has an extreme value opposite to t_j cannot make a strong error signal for adjusting the weights significantly [16], [17], as shown in Fig. 1. This incorrect saturation retards the search for a minimum in the error surface.

To resolve the incorrect saturation problem, the CE error function provides a strong error signal for the incorrectly saturated output nodes [17]. Additionally a weak error signal needs to be generated for correctly saturated output nodes so that the weight update associated with a training pattern scarcely perturbs the weights trained for all training patterns. The weak error signal is also necessary to prevent overfitting of learning for training patterns [6].

In this sense, an nCE error function [18] was proposed as

$$E_n(\mathbf{x}) = -\sum_{j=1}^{N_L} \int \frac{t_j^{n+1} (t_j - x_j^{(L)})^n}{2^{n-2} (1 - x_j^{(L)^2})} dx_j^{(L)}, \quad \text{where } t_j = \pm 1 \text{ and } n = 1, 2, \cdots \tag{5}$$

Using the above error function, the error signal of output layer becomes

$$\delta_j^{(L)} = \frac{t_j^{n+1} (t_j - x_j^{(L)})^n}{2^{n-1}}. \tag{6}$$

The nCE error function with $n = 1$ corresponds to the CE

error function. As shown in Fig. 1, the nCE error signal with $n \geq 2$ satisfies the above criterion, which provides a strong error signal for an incorrectly saturated output node and a weak error signal for a correctly saturated output node.

If the target values are not ± 1 but between -1 and $+1$, the slight modification of (6) as

$$\delta_j^{(L)} = \frac{[\mathrm{sgn}(t_j - x_j^{(L)})]^{n+1}(t_j - x_j^{(L)})^n}{2^{n-1}}. \quad (7)$$

will make the idea of nCE error work well. Here,

$$\mathrm{sgn}(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{otherwise.} \end{cases} \quad (8)$$

III. AN ADAPTIVE LEARNING RATE WITH LIMITED ERROR SIGNALS

If n of the nCE function increases from 2 to a higher value, the error signal will more effectively reduce the incorrect saturation of output nodes and prevent the overfitting for training patterns [18]. However, a very weak error signal for output nodes near desired values will delay learning since the associated weights are changed very small. And the training speed of MLPs as well as the classification performance for test patterns vary seriously on the order of nCE error. Therefore, an optimum order of the nCE function should be determined for good classification performance of MLPs with fast learning speed.

To remove the performance dependency of MLPs on the order of nCE, we propose an adaptive learning rate at each learning epoch s as

$$\eta(s) = \eta_0 \sqrt{\frac{E\{(t_j(s) - x_j^{(L)}(s))^2\}}{E\{\delta_j^{(L)^2}(s)\}}}. \quad (9)$$

Here, η_0 is an initial value of learning rate, and $E\{(t_j(s) - x_j^{(L)}(s))^2\}$ and $E\{\delta_j^{(L)^2}(s)\}$ are the expected values considering all output nodes in the s-th epoch. In the CE method, the learning rate is always η_0 since $\delta_j^{(L)}(s) = t_j(s) - x_j^{(L)}(s)$. Although $\eta(s)$ is dependent on $\delta_j^{(L)}(s)$, the expected intensity of $\eta(s)\delta_j^{(L)}(s)$ is

$$E\{\eta^2(s)\delta_j^{(L)^2}(s)\} = \eta_0^2 E\{(t_j(s) - x_j^{(L)}(s))^2\} \quad (10)$$

irrespective of the formulation of $\delta_j^{(L)}(s)$. Thus, the functional characteristics of $\delta_j^{(L)}$ can be used for updating weights while the expected intensity of $\eta(s)\delta_j^{(L)}(s)$ keeps the same value as that in the CE method. In simulations, $\eta(s)$ is calculated using $E\{(t_j - x_j^{(L)})^2\}$ and $E\{\delta_j^{(L)^2}\}$ estimated at the $(s-1)$-th epoch since they can not be derived at the beginning of the s-th epoch.

In the EBP algorithm using the nCE function, the adaptive learning rate may remove the dependency of learning on the order of nCE, while keeping the main effects of nCE function, i.e., reduction of incorrectly saturated output nodes and prevention of overfitting for training patterns. For a specific training pattern $\mathbf{x}$, however, the fraction term in the square-root of (9) is

$$\frac{(t_j(\mathbf{x}) - x_j^{(L)}(\mathbf{x}))^2}{\delta_j^{(L)^2}(\mathbf{x})} = \left(\frac{2}{t_j(\mathbf{x}) - x_j^{(L)}(\mathbf{x})}\right)^{2(n-1)} \quad (11)$$

by substitution of (6) and the fraction will increase when $x_j^{(L)}(\mathbf{x})$ converges to $t_j(\mathbf{x})$. That is, $\eta(s)$ may take a very large value when $E\{(t_j(s) - x_j^{(L)}(s))^2\}$ approaches to zero. Although $\eta(s)$ is very large, there are not oscillation or unstable problems of learning in the case that all output nodes approach to their target values. This can be easily verified from (10) which implies that the expected updating amount of weights is proportional to the distance between desired and actual output values multiplied by the initial learning rate. We can shortly prove this convergence as follows. When the weights approach to the minimum of error functional and all output nodes are near to their target values, it can be approximated that $\sqrt{E\{(t_j(s) - x_j^{(L)}(s))^2\}} \approx \left| t_j(s) - x_j^{(L)}(s) \right|$ and $\sqrt{E\{\delta_j^{(L)^2}(s)\}} \approx \left| \delta_j^{(L)^2}(s) \right|$. Therefore, the updating amount of $w_{ji}^{(L)}$ associated with a training pattern is

$$\begin{aligned} \left|\Delta w_{ji}^{(L)}(s)\right| &= \eta(s) \times \left|\delta_j^{(L)}(s)\right| \times \left|x_i^{(L-1)}(s)\right| \\ &\approx \eta_0 \times \frac{\left|t_j(s) - x_j^{(L)}(s)\right|}{\left|\delta_j^{(L)}(s)\right|} \times \left|\delta_j^{(L)}(s)\right| \times \left|x_j^{(L-1)}(s)\right| \end{aligned} \quad (12)$$

by substitution of the approximations into (9). Thus,

$$\left|\Delta w_{ji}^{(L)}(s)\right| \approx \eta_0 \times \left|t_j(s) - x_j^{(L)}(s)\right| \times \left|x_i^{(L-1)}(s)\right| \approx 0 \quad (13)$$

when all output nodes approach to their target values.

If some output nodes are far from their target values and the associated error signals are large, however, the large $\eta(s)$ results in an unwanted phenomenon that the updating amount of weights associated with the large error signals is more than that needed to minimize the nCE error functional. Thus, the adaptive learning rate may make training of MLPs unstable.

In order to suppress the excessive change of weights due to the large learning rate, we propose to limit error signal values of output nodes as

$$\delta_j^{(L)} = \begin{cases} \delta_j^{(L)}, \text{ if } -3\sqrt{E[\delta_j^{(L)^2}(s)]} < \delta_j^{(L)} < 3\sqrt{E[\delta_j^{(L)^2}(s)]} \\ \mathrm{sgn}(\delta_j^{(L)}) \times 3\sqrt{E[\delta_j^{(L)^2}(s)]}, \text{ otherwise.} \end{cases} \quad (14)$$

Using the limited error signal, the possibility of unstable learning will be reduced since the maximum value of weight change in the output layer is

$$\left|\Delta w_{ji}^{(L)}(s)\right|_{\max} = \eta(s) \times \left|\delta_j^{(L)}(s)\right|_{\max} \times \left|x_i^{(L-1)}(s)\right| = 3\eta_0 \left|x_i^{(L-1)}(s)\right| \sqrt{E[(t_j(s) - x_j^{(L)}(s))^2]}. \quad (15)$$

Backpropagation of the limited error signals also prevents an excessive change of weights in hidden layers.

If $\delta_j^{(L)}(s)$ is Gaussian with zero mean, $\sigma \approx \sqrt{E[\delta_j^{(L)^2}(s)]}$ and 99.7 % of $\delta_j^{(L)}(s)$ will be in $\pm 3\sigma$. Thus, the limited error signal (14) will change only a small portion of $\delta_j^{(L)}$. Although $\delta_j^{(L)}(s)$ is not Gaussian in real problems, it will be shown in the simulation section that $E[\delta_j^{(L)}(s)] \approx 0$ and major portion of $\delta_j^{(L)}$ is within $\pm 3\sqrt{E[\delta_j^{(L)^2}(s)]}$.

It looks more reasonable than (10) that $E\{\eta(s)\delta_j^{(L)}(s)\} = \eta_0 E\{(t_j(s) - x_j^{(L)}(s))\}$. This is achievable if we use $E\{\delta_j^{(L)}(s)\}$ and $E\{(t_j(s) - x_j^{(L)}(s))\}$ instead of the expectations of squares in (9), respectively. If we adopt this strategy, however, $E\{\delta_j^{(L)}(s)\}$ is nearly zero although MLPs are not sufficiently trained and $\eta(s)$ will increase too much to minimize error. Thus, the proposed learning rate by (9) is better after all. Otherwise, we can use $E\{\left|\delta_j^{(L)}(s)\right|\}$ and $E\{\left|t_j(s) - x_j^{(L)}(s)\right|\}$ in (9) to achieve $E\{\left|\eta(s)\delta_j^{(L)}(s)\right|\} = \eta_0 E\{\left|t_j(s) - x_j^{(L)}(s)\right|\}$.

When comparing performance of various error functions which are proposed for the EBP algorithm [6], [13], [17], [18], [23], learning rates are very important for fair comparison. Due to the stochastic property of $\delta_j^{(L)}$, many usually used a heuristic rule or assumed a distribution of output nodes to allocate the learning rates [17], [18]. If the proposed adaptive learning rate is used for the comparison, the expected intensity of $\eta(s)\delta_j^{(L)}(s)$ will be same during the learning process although $\delta_j^{(L)}(s)$'s are different. Thus, the proposed learning rate is adequate for fairly comparing performance of EBP algorithms using various error functions.

There are the other techniques for adapting learning rates of EBP algorithm. In the bold driver technique, the learning rate is increased if the error has actually decreased after each epoch and decreased if not [19], [20]. The delta-bar-delta rule adopts one learning rate for each weight and adapts the learning rates according to the signs of gradients on consecutive epochs [21]. Contrary to the two heuristic methods, optimum learning rates were derived for each neuron and training pattern through linear approximation of hidden activation function and derivatives of error functional with respect to learning rates [22]. Since all these were proposed to accelerate the EBP algorithm, they can not remove the performance dependency of learning on the order of nCE function. Conversely, the above three methods may have better acceleration performance than the proposed adaptive learning rate.

IV. SIMULATION

A handwritten digit recognition problem was used to verify the effectiveness of the adaptive learning rate and limited error signals. A total of 18,468 handwritten digitized images from the CEDAR database [24] were used for training after size normalization. A digit image consisted of 12×12 pixels and each pixel took on integer values from 0 to 15. Figure 2 shows some examples of digit images. The MLP consisted of 144 inputs, 30 hidden nodes, and 10 output nodes. Initialized weights were drawn at random from a uniform distribution on $[-1 \times 10^{-4}, 1 \times 10^{-4}]$. Nine simulations were conducted using each order of the nCE error function and the results were averaged to draw figures.

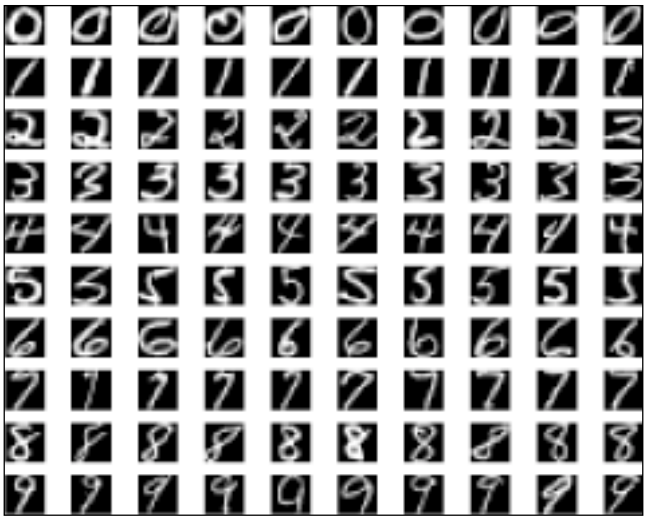


Fig. 2. Examples of handwritten digit images.

Firstly, MLPs were trained using the nCE error with a fixed learning rate. The learning rate for each order of nCE was derived as $\eta = 0.001 \times (n+1)$ so that $E\{\eta\delta_j^{(L)}\}$ had the same value under the assumption that $x_j^{(L)}$ had uniform distribution on $[-1, +1]$. The max rule also was used for classification of input patterns, that is, the index of maximum output node represented the classification result.

As shown in Fig. 3(a) which is the misclassification rates for the training patterns, the curves with $n = 2$ and 3 decrease more rapidly than one with $n = 1$ which corresponds to CE. Although

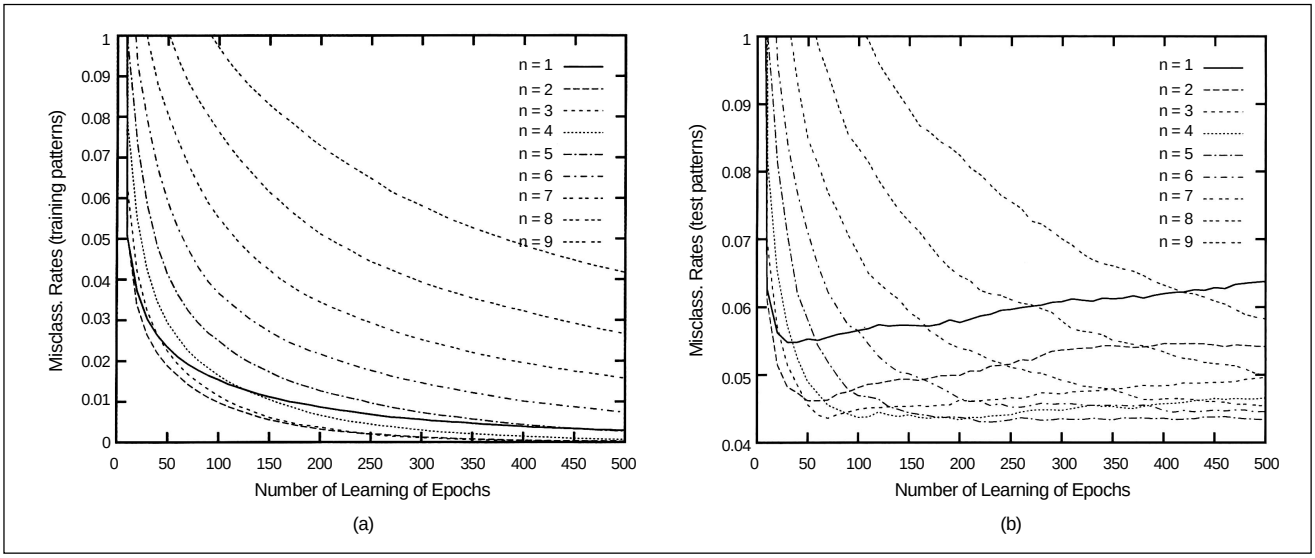


Fig. 3. Simulation results with the fixed learning rates for the handwritten digit recognition task.

it was shown that CE was better than MSE in the simulation of same problem using EBP algorithm [18], the poor performance of CE in this figure is mainly due to the competition during learning process. In the competition stage of learning, the direction of weight update for reducing total error and that for reducing error associated with a specific training pattern compete with each other [15]. To alleviate the competition, the weight update associated with a training pattern should scarely perturb the weights trained for all training patterns. In this sense, a weak error signal needs to be generated for output nodes near desired values. However, the error signal with $n = 1$ is relatively strong near a desired value as shown in Fig. 1 and this property induces more competitions than those with $n \geq 2$. When $n \geq 4$, the misclassification curves decrease more slowly since error signals are very small near desired values.

Figure 3(b) shows the misclassification rates for untrained 2,213 test patterns. The rate with $n = 1$ shows poor generalization since the CE error makes the MLP specialized too much for training patterns [18]. With increasing n until 5, we can get more improved results for the test patterns since weak error signals near desired values prevent overfitting for training patterns. With $n \geq 6$, however, very weak error signals near desired values retard learning and the misclassification curves decrease very slowly. From these results, $n = 3$ or 4 can be taken as an optimum order of the nCE in viewpoints of training speed and generalization performance.

To remove the performance variation on the order of nCE, the proposed method was adopted for training and the simulation results were drawn in Fig. 4(a) and (b). Comparing Fig. 4(a) with Fig. 3(a) which corresponds to the misclassification rates for the training patterns, it can be found that the proposed method successfully decreases the learning speed dependency on the order of nCE. Figure 4(b) shows the misclassification rates for the test patterns. The curve with $n = 1$ shows poor generalization performance since this curve corresponds to the CE method. With $n \geq 2$, the curves show better classification rates for the test patterns than that with the CE error. Comparing Fig. 4(b) with Fig. 3(b), it can be said that the proposed method reduces the variation of generalization performance on the order of nCE error. Thus, the proposed method alleviates the performance variation on the order of nCE, while it maintains the effect of nCE on preventing overfitting of MLP for training patterns. Naturally the proposed method maintains the effect of nCE on reducing incorrect saturation of output nodes.

Next, the probability density function of $\delta_j^{(L)}(s)$ and its thresholding value given by (14) were estimated. For this estimation, the range $(-2, +2)$ of $\delta_j^{(L)}(s)$ was divided into 200 bins and the number of $\delta_j^{(L)}(s)$ which belonged to each bin was counted during an epoch. The counted results were normalized using the number of training patterns multiplied by the number of output nodes. Figure 5 (a) and (b) show the estimated results at the 150th and 450th epochs when an MLP is trained using nCE $(n = 4)$ with the proposed method. As shown in these figures, $\delta_j^{(L)}(s)$ is mainly in $\left[-3\sqrt{E[\delta_j^{(L)^2}(s)]},\ 3\sqrt{E[\delta_j^{(L)^2}(s)]}\right]$ with nearly zero mean. Thus, the limited error signal makes a little effect on the training of MLPs with the nCE error function.

Previously, it was suggested that thresholding of error signals was necessary to prevent unstable learning due to an excessive

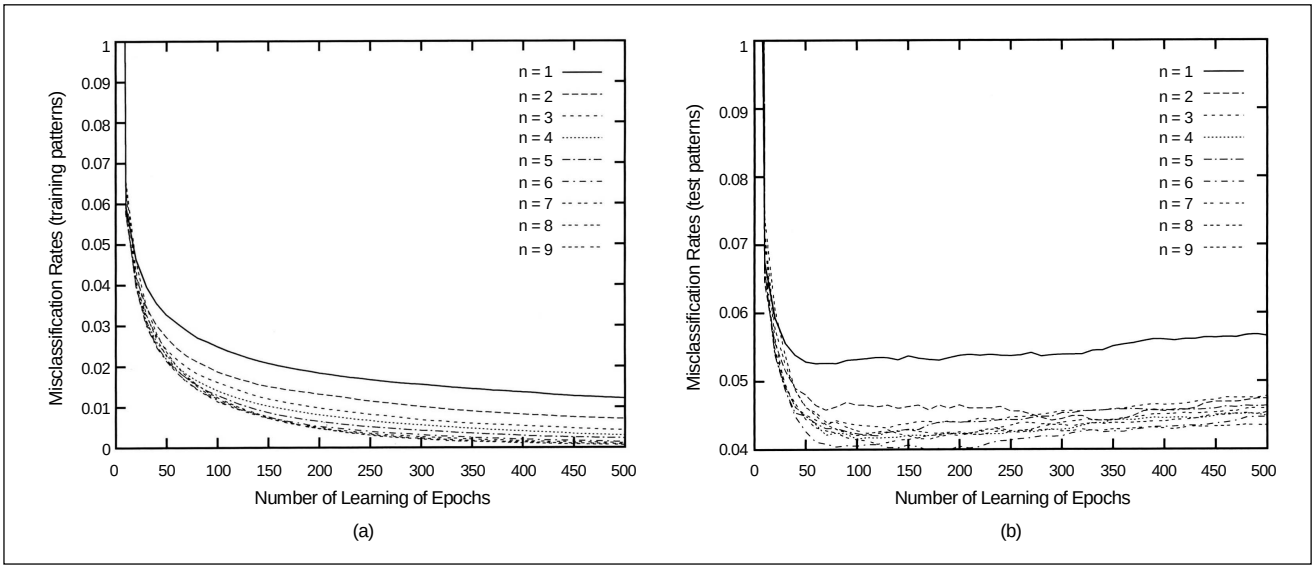


Fig. 4. Simulation results with the adaptive learning rate and limited error signals for the handwritten digit recognition task.

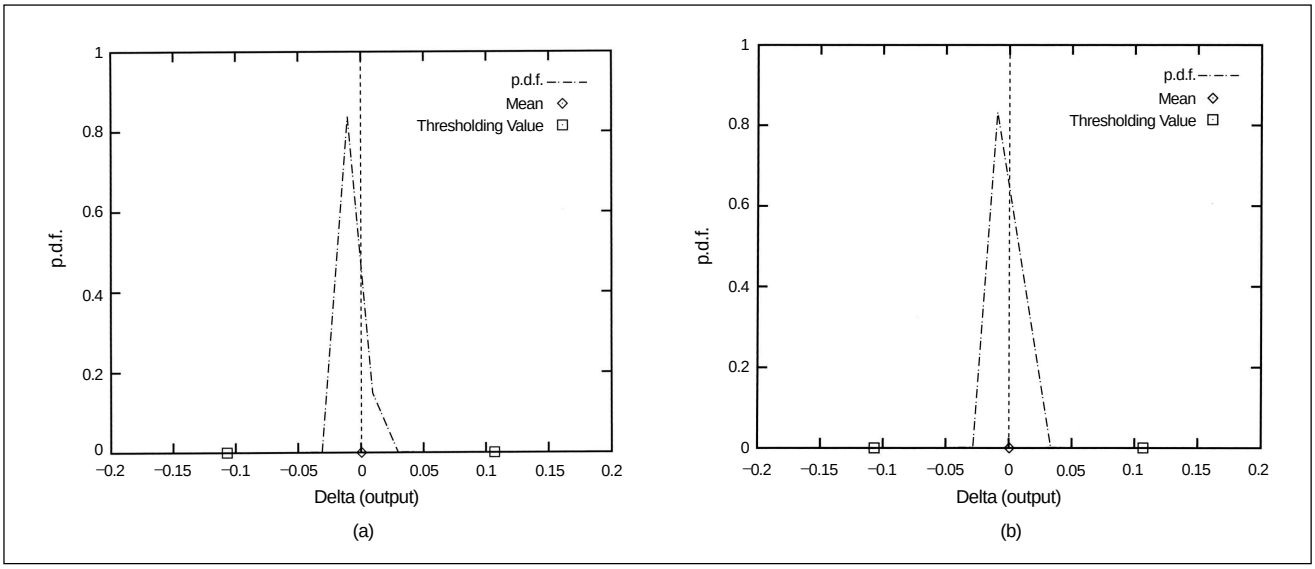


Fig. 5. Estimated results of probability density function and thresholding value for $\delta_j^{(L)}$ (a) at the 150th epoch and (b) at the 450th epoch.

change of weights with very large learning rates. To verify this argument, MLPs were trained only with the adaptive learning rate without thresholding of $\delta_j^{(L)}$. Figure 6 (a), (b), and (c) are the adaptive learning rate, the misclassification rate for the training patterns, and that for the test patterns, respectively, when an MLP initialized with a set of random weights was trained using the nCE error function with $n = 4$.

When output nodes converge to their target values during the progress of learning, the adaptive learning rate increases as shown in Fig. 6(a). It also accelerates training and the misclassification rate in Fig. 6(b) decreases more rapidly than those in Fig. 3(a) or 4(a). Although the training patterns are perfectly classified after the 150th training epoch, the misclassification rate for the test patterns in Fig. 6(c) is not satisfactory because the acceleration by the adaptive learning rate induces overfitting for training patterns. The positive feedback between the learning rate and classification rate ends up with drastic increasing of the misclassification at the 220th epoch. As described before, this is due to the adaptive learning rate which becomes larger than that needed to minimize the error function. After then, there is a progress of learning again with a decreased learning rate caused by increasing of the distance between desired and actual values of output nodes.

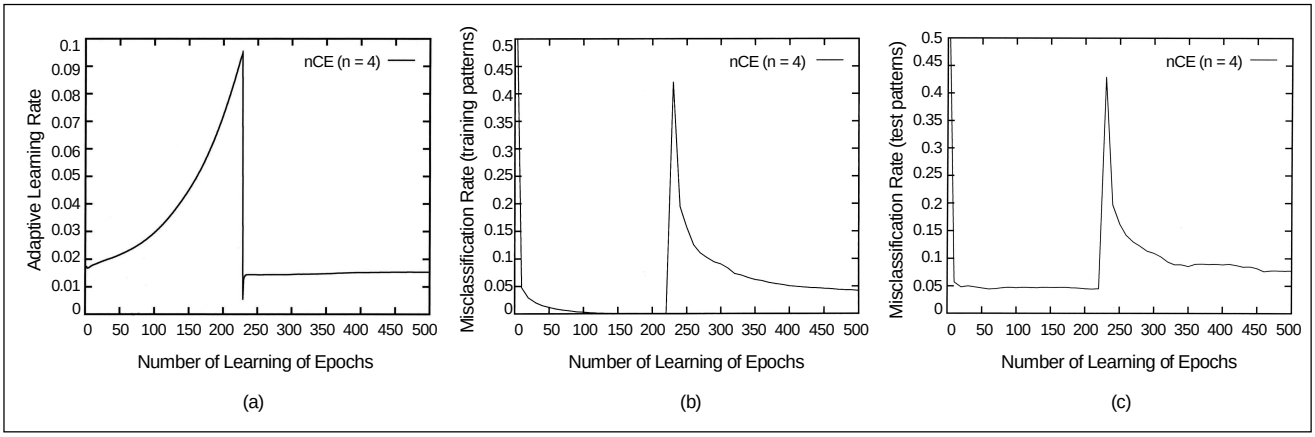


Fig. 6. A simulation result with the adaptive learning rate without thresholding error signals in the handwritten digit recognition task.

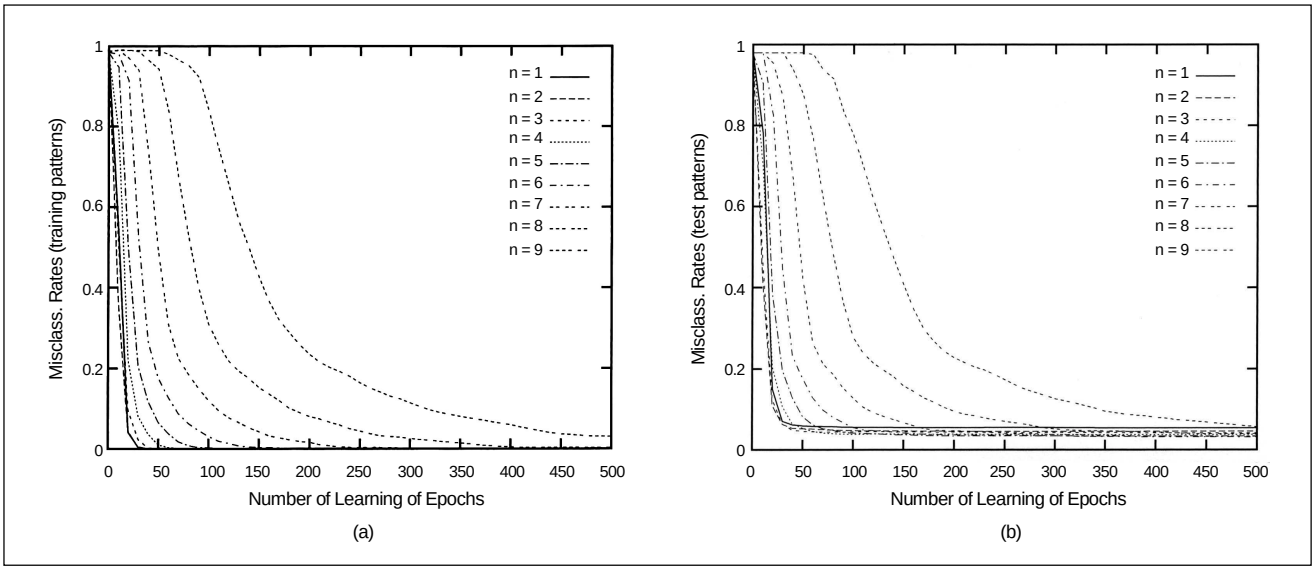


Fig. 7. Simulation results with the fixed learning rates for the isolated-word recognition task.

For more verification of the proposed method, an isolated-word recognition problem was used in which the vocabulary consisted of 50 words and each word was spoken two times by nine speakers. The 900 patterns were used for training after extracting the ZCPA feature of 1,024 dimensions [25]. The MLP consisted of 1,024 inputs, 50 hidden nodes, and 50 output nodes. Nine simulations were conducted with initial weights randomly drawn from a uniform distribution on $[-1\times10^{-4}, 1\times10^{-4}]$ and the results were averaged. Generalization performance for this task was evaluated using untrained 1050 test patterns, which were the 50 words spoken three times by seven speakers. Figure 7(a) and (b) are the simulation results with fixed learning rates $\eta = 0.01\times(n+1)$. In these figures, the misclassification rates for the training and test patterns vary seriously on the order of nCE error function. Figure 8 is the simulation results using the proposed adaptive learning rate and limited error signals. It is clear that the performance variation on the order of nCE error is dramatically reduced by the proposed method. This is consistent with the simulation results for the handwritten digit recognition task.

Besides the proposed adaptive learning rate, there might be other methods to reduce the performance dependency of MLPs on the order of nCE. For example, we can adapt the order of nCE according to progress of learning. However, this method needs heuristics to select an order of nCE at any learning stage.

In application of MLPs to pattern recognition, it is important to choose the number of hidden neurons since the training speed and performance of MLPs depend strongly on the number of hidden neurons. Here, we used a pruning method to choose the number of hidden neurons in the two experiments of this paper. It is usually said that the hidden neurons of the single hidden layer perceptron are trained to extract features from input patterns. This process is done by a series of linear projections onto weight vectors and element-wise sigmoid transformations. After successful

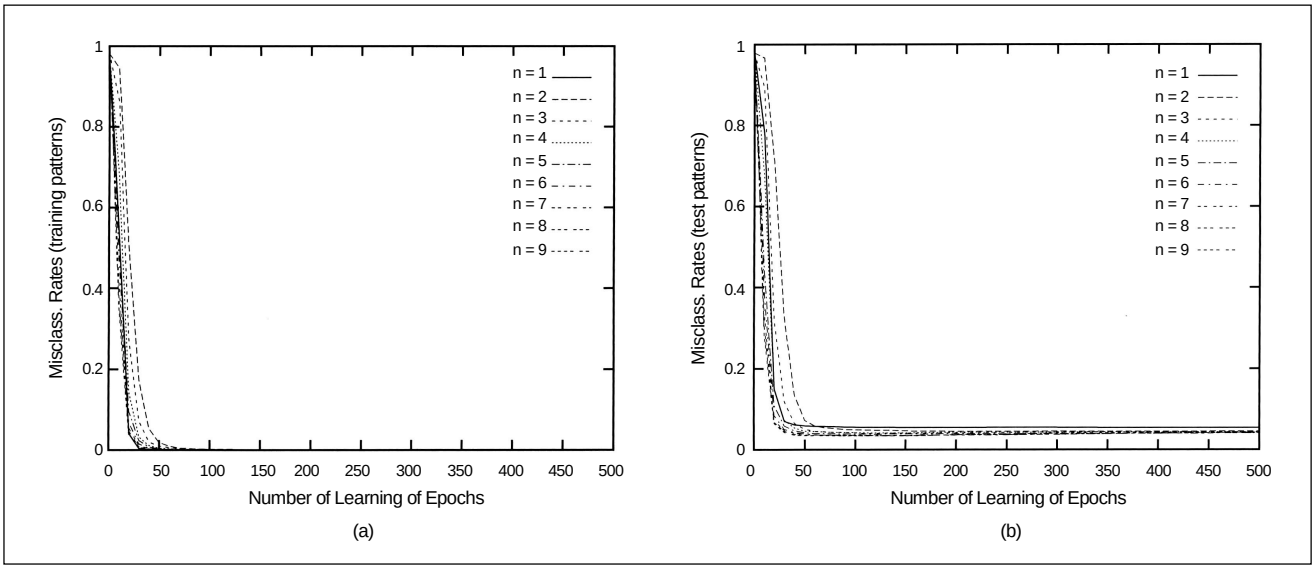


Fig. 8. Simulation results with the adaptive learning rate and limited error signals for the isolated-word recognition task.

training of the MLP, hidden weight vectors tend to be near-orthogonal to extract useful features [26]. If some hidden weight vectors are similar to each other, they extract similar features and we can remove one among them.

In this point of view, we trained an MLP with many hidden neurons and pruned hidden neurons one by one based on the pair-wise inner products of normalized hidden weight vectors when there was no progress of learning. After each pruning procedure of one hidden neuron, the MLP recovered the classification ability of patterns with additional learning. If the MLP could not recover the classification rate, the hidden neurons were pruned too many. Finally, we chose the number of hidden neurons in the two experiments of this paper based on the pruning results with some marginal number.

V. CONCLUSION

Although MLPs trained using the nCE error function show good performance, the training speed of MLPs as well as the classification performance for test patterns vary seriously on the nCE order. This paper proposed an adaptive learning rate to make performance of MLPs insensitive to the order of nCE error function. The proposed adaptive learning rate complemented the variation of error signals on the nCE order by regulating that the expected intensity of the error signal multiplied by the adaptive learning rate was the same for different orders of nCE. The adaptive learning rate has an weakness that it may take a very large value during a progress of learning. This results in an unstable characteristic of learning, i.e., the updating amount of weights is more than that needed to minimize the nCE error functional. A limited error signal of output node was additionally proposed to prevent the excessive change of weights due to the large learning rate.

The effectiveness of the proposed method was demonstrated through the simulation of handwritten digit and isolated-word recognition tasks. In the simulation, it was shown that the proposed method reduced the performance variation on the the order of nCE error, while maintaining the effect of nCE on preventing overspecialization of MLPs for training patterns. Also, it was verified that the limited error signals effectively removed the unstable learning caused by a large value of the adaptive learning rate. In addition to removing the performance dependency on the order of nCE function, the proposed method will be suitable for fair comparison of EBP algorithms using various error functions.

ACKNOWLEDGMENT

This research was partly supported as the Brain Science and Engineering Research Program by the Korean Ministry of Science and Technology. The authors wish to thank the reviewers for their helpful criticisms and suggestions in improving this paper.

REFERENCES

[1] J. Choi, S. H. Bang, and B. J. Sheu, "A Programmable Analog VLSI Neural Network Processor for Communication Receivers," *IEEE Trans. Neural Networks*, Vol. 4, 1993, pp. 484–495.

[2] B. Aazhang, B-P. Paris, and G. C. Orsak, "Neural Networks for Multiuser Detection in Code-division Multiple-access Communications," *IEEE Trans. Communications*, Vol. 40, 1992, pp. 1212–1222.

[3] E. Nordstrom, O. Gallmo, and L. Asplund, "Neural Networks for Adaptive Traffic Control in ATM Networks," *IEEE Communications Magazine*, Vol. 33, 1995, pp. 43–49.

[4] I. W. Habib, A. A. Tarraf, and T. N. Saadawi, "Intelligent Traffic Control for ATM Broadband Networks," *IEEE Communications Magazine*, Vol. 33, 1995, pp. 76–85.

[5] R. P. Lippmann, "Pattern Classification Using Neural Networks," *IEEE Communications Magazine*, Nov. 1989, pp.47–64.

[6] J. B. Hampshire II and A. H. Waibel, "A Novel Objective Function for Improved Phoneme Recognition Using Time-delay Neural Networks," *IEEE Trans. Neural Networks*, Vol. 1, June 1990, pp. 216–228.

[7] A. S. Weigend and N. A. Gershenfeld, *Time Series Prediction: Forecasting the future and understanding the past*, Addison-Wesley Publishing Co., 1994.

[8] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamic System Using Neural Networks," *IEEE Trans. Neural Networks*, Vol. 1, 1990, pp. 4–27.

[9] K. Hornik, M. Stincombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, 1989, pp. 359–366.

[10] H. White, "Learning in Artificial Neural Networks: a Statistical Perspective," *Neural Computation*, Vol. 1, 1989, pp. 425–464.

[11] M. D. Richard and R. P. Lippmann, "Neural Network Classifier Estimate Bayesian a Posteriori Probabilities," *Neural Computation*, Vol. 3, 1991, pp. 461–483.

[12] Y. Lee and S.-H. Oh, "Input Noise Immunity of Multilayer Perceptrons," *ETRI Journal,* Vol. 16, April 1994, pp. 35–43.

[13] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 1986, pp. 318–362.

[14] J. R. Chen and P. Mars, "Stepsize Variation Methods for Accelerating the Backpropagation Algorithm," *Proc. IJCNN Jan. 15-19, 1990, Washington D.C. USA*, Vol. I, pp. 601–604.

[15] 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 D.C. USA*, Vol. I, pp. 707–710.

[16] Y. Lee and S.-H. Oh and M. W. Kim, "An Analysis of Premature Saturation in Back-propagation Learning," *Neural Networks*, Vol.6, 1993, pp. 719–728.

[17] A. van Ooyen and B. Nienhuis, "Improving the Convergence of the Back-propagation Algorithm," *Neural Networks*, Vol. 5, 1992, pp. 465–471.

[18] S.-H. Oh, "Improving the Error Backpropagation Algorithm with a Modified Error Function, *IEEE Trans. Neural Networks*, Vol. 8, No. 3, 1997, pp.799–803.

[19] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon, "Accelerating the Convergence of the Back-propagation Method," *Biol. Cybern*, Vol. 59, 1988, pp.257–263.

[20] R. Battiti, "Accelerated Backpropagation Learning: Two Optimization Methods," *Complex Systems*, Vol. 3, 1989, pp. 331–342.

[21] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford Univ. Press, New York, 1995, pp. 270–271.

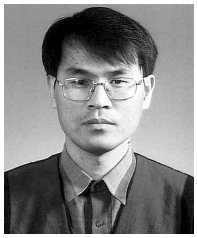
[22] S.-H. Oh and Soo-Young Lee, "A New Error Function at Hidden Layers for Fast Training of Multilayer Perecptrons," *IEEE Trans. Neural Networks*, Vol. 10, 1999, pp. 960–964.

[23] B. A. Telfer and H. H. Szu, "Energy Functions for Minimizing Misclassification Error with Minimum-complexity Networks," *Neural Networks*, Vol. 7, 1994, pp. 809–818.

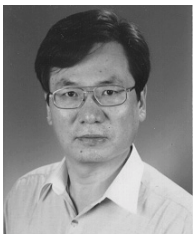
[24] J. J. Hull, "A Database for Handwritten Text Recognition Research," *IEEE Trans. Pat. Ana. Mach. Int.*, Vol. 16, No. 5, May 1994, pp. 550–554.

[25] D.-S. Kim, S.-Y. Lee, and R. M. Kil, "Auditory Processing of Speech Signals for Robust Speech Recogtion in Real-world Noisy Environments," *IEEE Trans. Speech and Audio Processing*, Vol. 7, 1999, pp. 55–69.

[26] S.-H. Oh and Y. Lee, "Effect of Nonlinear Transformations on Correlation between Weighted Sums in Multilayer Perceptrons," *IEEE Trans. Neural Networks*, Vol. 5, 1994, pp. 508–510.



Sang-Hoon Oh received the B.S. and M.S. degrees in electronics engineering from Pusan National University, Korea, in 1986 and 1988, respectively, and the Ph.D. degree in electrical engineering from Korea Advanced Institute of Science and Technology, Korea, in 1999. From 1988 to 1989, he was with Goldstar Semiconductor, Ltd., Korea, where he was a quality control engineer of MOS FAB. He was with ETRI from 1990 to 1998, pursuing interests in analyses of supervised neural network algorithms and developing new supervised learning algorithms. Also he was related to implementation of channel fading emulators for mobile telecommunications. He was with Brain Science Research Center, KAIST, from Sept. 1999 to March 2000. Since April 2000, he has been a staff of research at Lab. for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Japan, doing researches in ICA (Independent Component Analysis)/BSS (Blind Source Separation) algorithms and their applications to real problems. His research interests include pattern recognition and biometrics.



Soo-Young Lee is a director of Brain Science Research Center, and also professor at Department of Electrical Engineering, Korea Advanced Institute of Science and Technology. He received B.S. degree in Electronics at Seoul National University in 1975, M.S. degree in Electrical Engineering at Korea Advanced Institute of Science in 1997, and Ph.D. degree in Electrophysics at Polytechnic Institute of New York, 1984. He has published over 70 journal papers and presented over 130 conference papers in the area of neural networks, speech recognition, VLSI implementations, optical signal processing, and computational physics. Currently his main research interests reside in robust speech recognition, auditory models, selective attention, independent component analysis, and self-learning neuro-chips. He is now the project manager of Korean Brain Science and Engineering Research Program, which consists of over 120 professors and 340 graduate researchers. He has been invited speakers, session organizers, and program committee members for over 30 international conferences. He is now president-elect of Asia-Pacific Neural Network Assembly, and conference chair of the ICONIP2000.