Improving the Error Back-Propagation Algorithm with a Modified Error Function

Sang-Hoon Oh *

Abstract

This letter proposes a modified error function to improve the EBP(Error Back-Propagation) algorithm of MLPs(Multi-Layer Perceptrons) which suffers from slow learning speed. To accelerate the learning speed of the EBP algorithm, the proposed method reduces the probability that output nodes are near the wrong extreme value of sigmoid activation function. This is acquired through a strong error signal for the incorrectly saturated output node and a weak error signal for the correctly saturated output node. The weak error signal for the correctly saturated output node, also, prevents overspecialization of learning for training patterns. The effectiveness of the proposed method is demonstrated in a handwritten digit recognition task.

1 Introduction

Multi-Layer Perceptrons (MLPs) use the Error Back-Propagation (EBP) algorithm for training[1]. Training is usually done by iterative updating of weights according to the error signal, which is the negative gradient of the mean-squared error(m.s.e.) function. In the output layer, the error signal is the difference between desired and actual output values of MLP multiplied by the slope of sigmoid activation function. Then, the error signal is back-propagated to the lower layer. The EBP algorithm, however, has a drawback of slow learning. During learning, MLPs go through stages in which the reduction of the m.s.e. is extremely slow[2, 3]. These periods of stagnation dominate learning time.

In pattern recognition applications, the desired output value of MLP is one of the two extreme values of the sigmoid function. If the weighted sum to any output node is near the wrong extreme value, we say the node is "incorrectly saturated[3]."

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[3, 4, 5, 6]. In order to resolve this problem, the cross-entropy(CE) error function was proposed so that the error signal associated with the output layer is proportional to the difference between desired and actual output values[6, 7]. However, the CE method suffers from overspecialization for training patterns since the error signal for a correctly saturated output node is too strong[8]. Although many proposed other error functions to improve the generalization performance[9, 10, 11], none of these resolved the incorrect

*Research Department, Electronics and Telecommunications Research Institute, Yusong P.O. Box 106, Taejon, Korea

saturation problem of output nodes. The probability of incorrect saturation in the first epoch of learning showed that initialization of MLP with small random weights can avoid the stagnation period in the early stage of learning[3]. Even in this case, the possibility of incorrect saturation still exists through the learning process.

This letter proposes an improved EBP algorithm of MLPs to accelerate the learning speed by reducing the probability of incorrect saturation. Additionally, it intends to improve the generalization performance by preventing overspecialization for training patterns. In section 2, I describe the modified error function for EBP algorithm. In section 3, I demonstrate the effectiveness of the modified error function in the handwritten digit recognition task using the CEDAR database[12], and section 4 concludes this letter.

2 The Modified Error Function

Consider an MLP consisting of L layers in which each lth 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 $\mathbf{x}$ be $\mathbf{t} = [t_1, t_2, \ldots, t_{N_L}]$, where the class where $\mathbf{x}$ originates from is coded as follows

$$t_j = \begin{cases} +1, & \text{if } \mathbf{x} \text{ originates from class } j \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

When $\mathbf{x}$ is presented to the network and propagated forward to determine the output signal, the state $x_j^{(l)}$ in each lth layer is

$$x_j^{(l)} = \tanh(a_j^{(l)}/2), \quad l = 1, 2, \ldots, L \tag{2}$$

where

$$a_j^{(l)} = w_{j0}^{(l)} + \sum_{i=1}^{N_{l-1}} w_{ji}^{(l)} x_i^{(l-1)}. \tag{3}$$

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[1] is

$$E_m(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^{N_L} (t_j - x_j^{(L)})^2. \tag{4}$$

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

$$\Delta w_{ji}^{(l)} = -\eta \frac{\partial E_m(\mathbf{x})}{\partial a_j^{(l)}} \frac{\partial a_j^{(l)}}{\partial w_{ji}^{(l)}} = \eta \delta_j^{(l)} x_i^{(l-1)}. \tag{5}$$

Here,

$$\delta_j^{(l)} = -\frac{\partial E_m(\mathbf{x})}{\partial a_j^{(l)}} = \begin{cases} \frac{(1-x_j^{(L)})(1+x_j^{(L)})}{2} (t_j - x_j^{(L)}), & \text{where } l = L, \\ \frac{(1-x_j^{(l)})(1+x_j^{(l)})}{2} \sum_{k=1}^{N_{l+1}} w_{kj}^{(l+1)} \delta_k^{(l+1)}, & \text{where } 1 \leq l \leq L-1 \end{cases} \tag{6}$$

is the error signal and η is the learning rate. Also, the m.s.e. function for all P training patterns is

$$E_m = \frac{1}{P}\sum_{p=1}^{P} E_m(\mathbf{x}^p) \tag{7}$$

where $\mathbf{x}^p$ is the pth training pattern. We can minimize E_m through iterative updating of weights for all training patterns. In the limit $P \to \infty$, the minimizer of E_m converges (under certain regularity conditions, Theorem 1 in [13]) towards the minimizer of the function

$$\Delta_m = E\{\frac{1}{2}\sum_{j=1}^{N_L}(T_j - x_j^{(L)}(\mathbf{X}))^2\}, \tag{8}$$

where $E\{.\}$ is the expectation operator, $\mathbf{X}$ is the random vector denoting an input pattern, and T_j is the random variable denoting the target. Thus when network parameters are chosen to minimize Δ_m, $x_j^{(L)}(\mathbf{x})$ estimates $E\{T_j|\mathbf{x}\}$[7, 13]. Since the targets are coded as in (1),

$$E\{T_j|\mathbf{x}\} = 2Q_j(\mathbf{x}) - 1 \tag{9}$$

where I denote by $Q_j(\mathbf{x})$ the posterior probability

$$Q_j(\mathbf{x}) = Pr[\mathbf{X} \text{ originates from class } j|\mathbf{X} = \mathbf{x}]. \tag{10}$$

In the above EBP algorithm, $\delta_j^{(L)}$ in (6) is the difference $(t_j - x_j^{(L)})$ multiplied by the gradient of the sigmoid function. If $x_j^{(L)}$ approaches one of the two extreme values of sigmoid function, the gradient factor in (6) makes the error signal very small. Thus, the output node $x_j^{(L)}$ which has extreme value opposite t_j can not make a strong error signal for adjusting the weights significantly[3, 6], as shown in Figure 1. This incorrect saturation retards the search for a minimum in the error surface.

In order to accelerate the EBP algorithm, one can use the CE error function[6]

$$E_c(\mathbf{x}) = -\sum_{j=1}^{N_L}[(1+t_j)\ln(1+x_j^{(L)}) + (1-t_j)\ln(1-x_j^{(L)})]. \tag{11}$$

Using the above error function, the error signal $\delta_j^{(L)}$ in (6) becomes

$$\delta_j^{(L)} = t_j - x_j^{(L)}. \tag{12}$$

Thus, the output nodes can escape well from the state of incorrect saturation. When MLPs are trained to minimize $E\{E_c(\mathbf{X})\}$, $x_j^{(L)}(\mathbf{x})$ estimates $E\{T_j|\mathbf{x}\}$[7, 14]. The difference error signal, however, specializes the MLP too much for training patterns since the error signal is relatively strong when $x_j^{(L)}$ is near t_j, as shown in Figure 1.

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[2]. For instance, some of 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 nodes to minimize the competition, like the CE method. For correctly saturated output nodes, a weak error signal needs to be generated so that the weight update associated with a training pattern scarcely perturb the weights trained for all training patterns. The weak error signal is also necessary to prevent overspecialization of learning for training patterns, like the classification figure of merit(CFM) method[9].

In this sense, I propose the modified error function

$$E_l(\mathbf{x}) = -\sum_{j=1}^{N_L} \int \frac{{t_j}^{n+1}(t_j - x_j^{(L)})^n}{2^{n-2}(1 - {x_j^{(L)}}^2)} dx_j^{(L)}, \quad \text{where} \quad t_j = \pm 1 \quad \text{and} \quad n = 1, 2, ... \tag{13}$$

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

$$\delta_j^{(L)} = \frac{{t_j}^{n+1}(t_j - x_j^{(L)})^n}{2^{n-1}}. \tag{14}$$

If one used a different activation function from (2), the error function (13) should be defined differently. The other equations in the EBP algorithm are the same. As shown in Figure 1, the proposed error signal with $n \geq 2$ can satisfy the above criterion, which requests a strong error signal for an incorrectly saturated output node and a weak error signal for a correctly saturated output node.

The error signal with $n = 1$ is one proposed by the CE method. If we increase n from 2 to higher value, the error signal will more effectively reduce the incorrect saturation of output nodes and prevent the overspecialization for training patterns. However, with increasing n, the very weak error signal for the output node near the desired value will delay learning. In simulations, I demonstrate the performance of the proposed error signals with $n = 2$ and 4.

Let us seek the function $\mathbf{b} = [b_1, b_2, \ldots, b_{N_L}]^T$ minimizing the criterion (in the space of all functions taking values in $(-1, 1)$)

$$E\{\sum_{j=1}^{N_L} \ell(T_j, x_j^{(L)}(\mathbf{X}))\} = \int \sum_{j=1}^{N_L} [Q_j(\mathbf{x})\ell(1, x_j^{(L)}(\mathbf{x})) + (1 - Q_j(\mathbf{x}))\ell(-1, x_j^{(L)}(\mathbf{x}))] f(\mathbf{x}) d\mathbf{x}, \tag{15}$$

where

$$\ell(t, y) = -\int \frac{t^{n+1}(t - y)^n}{2^{n-2}(1 - y^2)} dy \tag{16}$$

is the proposed error function for an output node. For a fixed $Q_j(\mathbf{x}), 0 < Q_j(\mathbf{x}) < 1$, we see that the optimal solution is given by $\mathbf{b}(\mathbf{X})$, where the components of $\mathbf{b}$ are given by

$$b_j(\mathbf{x}) = h_2(h_1(Q_j(\mathbf{x})). \tag{17}$$

Here, $h_1 : (0, 1) \to (0, \infty)$ and $h_2 : (0, \infty) \to (-1, 1)$ are given by

$$h_1(q) = \left(\frac{1 - q}{q} \right)^{1/n}, \tag{18}$$

$$h_2(u) = \frac{1-u}{1+u}. \tag{19}$$

Notice that both h_1 and h_2 are strictly decreasing functions, so $h_2 \circ h_1$ is strictly increasing. Figure 2 shows the optimal solutions with $n = 1, 2,$ and 4. Hence the Bayes classifier can be defined by

$$\text{decide } j \ , \quad \text{if } j = \max x_j^{(L)}(\mathbf{x}). \tag{20}$$

This argument shows that the proposed error function with $n \geq 2$ really is sensible in a classification task provided the targets are coded as in (1).

3 Simulation

A handwritten digit recognition problem is used to compare the proposed method with the conventional EBP and the CE methods. Besides the three methods, I simulate the CFM method which seeks to maximize the difference between the output value of node representing the correct classification, so called "target node", and all other nodes[9]. A total of 18,468 handwritten digitized images from the CEDAR database[12] are used for training after size normalization. A digit image consists of 12×12 pixels and each pixel takes on integer values from 0 to 15. The MLP consists of 144 inputs, 30 hidden nodes, and 10 output nodes.

Since no fair comparison is possible if the learning rate is kept the same for all methods[6], I derive the learning rates so that $E\{\eta\delta_j^{(L)}\}$ of the target node has the same value in each method. Here, I assume that $x_j^{(L)}$ has uniform distribution on $[-1, +1]$. As a result, the learning rates of 0.006, 0.002, 0.015, 0.003, and 0.005 are used for the conventional EBP method, the CE method, the CFM method, and the proposed method with n=2 and 4, respectively. Nine simulations are conducted using each method with same initializations and I average the results to draw figures. The initial weights were drawn at random from a uniform distribution on $[-1 \times 10^{-4}, 1 \times 10^{-4}]$.

Figure 3 shows the misclassification ratio and the incorrect saturation ratio(ISR) for the training patterns. Here, I use the max. rule for classification which means that the index of maximum output node represents the classification result. The ISR is defined by the number of output nodes−which are farther from the desired value than 1.8 in each epoch−divided by $N_L \times P$. As shown in Figure 3, the misclassification ratio for the training patterns is highly correlated with the ISR. The conventional method shows the worst results. Although the CE and the CFM methods reduce the ISR, they still attain high ISR. On the contrary, the proposed method with n=2 and 4 shows much smaller ISR than those of the others. Thus, the proposed method attains nearly perfect classification for training patterns with faster learning speed than the others. Although the proposed method with n=4 shows negligible ISR, the very weak error signal of output node near the desired value retards the learning speed. As a result, the proposed method with n=2 is faster than one with n=4.

We can find the generalization performance of each method in Figure 4 which shows the misclassification ratio of 2,213 test patterns. The generalization performance of the conventional method is poor since the ISR is the worst. In the CE method, despite the reduced ISR, the generalization performance goes worse after some period of better result than that of the conventional method. This indirectly supports the previous argument that the difference error signal in the CE method overspecializes the MLP for training patterns. The CFM method generates a very weak error signal for a correctly saturated output node to prevent overspecialization for training patterns[9]. Thus, as shown in figure 4, the generalization performance of the CFM method is better than that of the CE method although the CFM method shows higher ISR than that of the CE method.

Because of the reduced ISR and the weak error signal near the desired value, the proposed method with n=2 and 4 shows good generalization performance. Among the simulation methods, the proposed method with n=4 shows the best generalization performance since it attains the least ISR and generates the weak error signal to prevent the overspecialization. Although the proposed method generates the weak error signal for a correctly saturated output node, iterative updating of weights overspecializes the MLP for training patterns. Therefore, the generalization performance using the proposed method goes worse after the best result when the learning epoch increases as shown in Figure 4.

Figure 5 shows the generalization performance using each method when the number of training patterns has various values. For this figure, I randomly select nine sets of training patterns corresponding to the number of training patterns from the 18,468 patterns and conduct nine times of simulations with each set. In each simulation, I measure the best classification ratio for the test patterns within 500 training epochs and average the results to draw Figure 5. In the case that training patterns are less sampled compared to the complexity of recognition problem, each method shows similar performance of generalization. With enough training patterns, each method shows different generalization performance and the proposed method with n=4 shows the best result.

We can compare Figure 5 with Table 1 in [15] which shows the performance of handwritten digit recognition using k nearest-neighbor, radial-basis function network, and MLP.

4 Conclusion

This letter proposed a modified error function to accelerate the learning speed of the EBP algorithm by reducing the probability that output nodes are near the wrong extreme values. The modified error function, also, prevented the overspecialization of learning for training patterns by generating the weak error signal for the output node near the desired value. Using the modified error function, the error signal of output layer could be represented by an n-th order function of the difference between desired and actual output values.

I have compared the proposed method with the conventional EBP algorithm, the cross-entropy method, and the classification figure of merit method through the simulation of classifying handwritten digits in the CEDAR database[12]. The conventional EBP algorithm did not show good performance because of the incorrect saturation. Although the cross-entropy method reduced the incorrect saturation, it still attained high ratio of incorrect saturation and poor generalization performance. Despite the good generalization performance, the classification figure of merit method showed slow learning speed with high ratio of incorrect saturation. Compared to these methods, the proposed method showed the best performance for both the training and test patterns with accelerated learning speed, since it effectively eliminated the incorrect saturation and prevented the overspecialization during learning.

5 Acknowledgment

This research was partly funded by Korea Telecom. and the Ministry of Information and Communication, Korea. The author wishes to thank Dr. El Hang Lee for his continuing support and guidance. Also, the author wishes to thank Dr. Youngjik Lee for his helpful discussions and proofreadings and the reviewer for his helpful criticisms, who suggested the theoretical basis of the proposed learning algorithm.

References

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

[2] 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, 1990.

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

[4] 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, 1990.

[5] 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, 1990.

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

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

[8] S.-H. Oh and Y. Lee, "A modified error function to improve the error back-propagation algorithm for Multi-Layer Perceptrons," *ETRI Journal*, vol. 17, no. 1, pp. 11–22, April 1995.

[9] 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, pp. 216–228, June 1990.

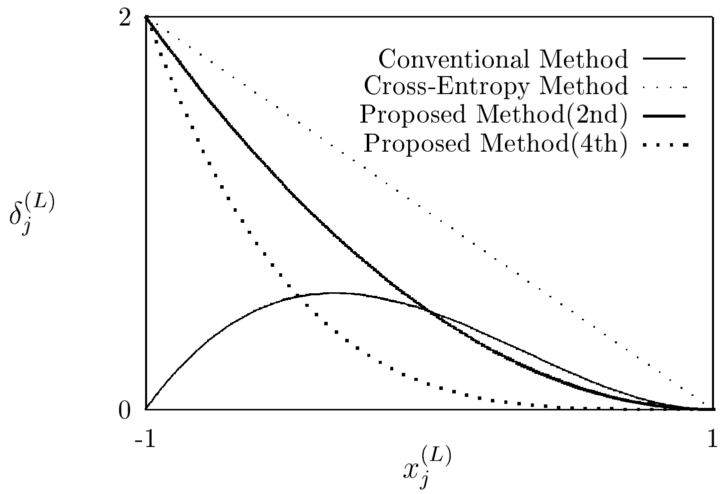


Figure 1: The error signal of output node with $t_j = 1$. $x_j^{(L)}$ is the jth output value and $\delta_j^{(L)}$ is the error signal of $x_j^{(L)}$.

[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] D.-G. Jeong and S.-Y. Lee, "Merging back-propagation and Hebbian learning rules for robust classifications," *Neural Networks*, vol. 9, no. 7, pp. 1213–1222, 1996.

[12] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pat. Ana. Mach. Int.*, vol. 16, no. 5, pp. 550–554, May 1994.

[13] H. White, "Learning in artificial neural networks: a statistical perspective," *Neural Computation*, vol. 1, pp. 425–464, 1989.

[14] J. B. Hampshire II and B. Pearlmutter, "Equivalence proofs for multi-layer perceptron classifier and the bayesian discriminant function," *Proc. 1990 Connectionist Model Summer School*, pp. 159–172, 1990.

[15] Y. Lee, "Handwritten digit recognition using k nearest-neighber, radial-basis function, and backpropagation neural networks," *Neural Computation*, vol. 3, pp. 440–449, 1991.

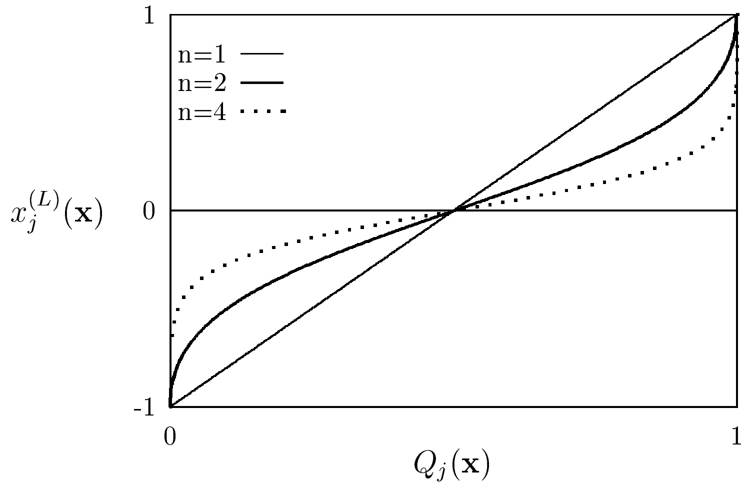
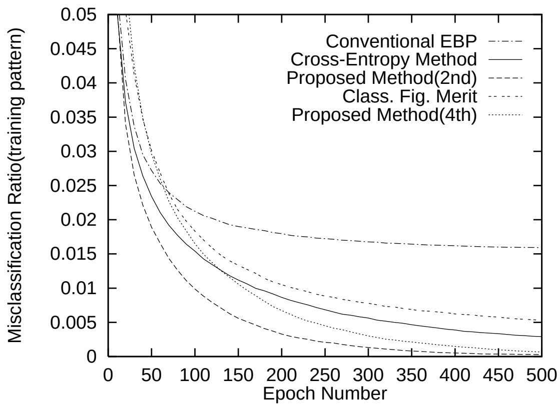
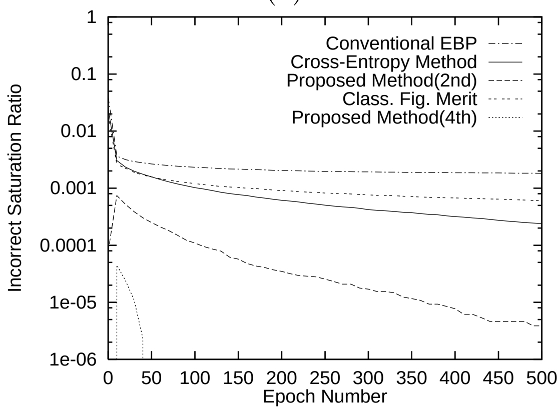


Figure 2: The optimal solutions of $x_j^{(L)}(\mathbf{X})$ for minimizing $E\{E_l(\mathbf{X})\}$ with n=1, 2, and 4. $E\{.\}$ denotes the expectation operator and $E_l(\mathbf{X})$ is the proposed error function when a random vector $\mathbf{X}$ is presented to an MLP as an input pattern. Also, $Q_j(\mathbf{x})$ is the posterior probability $Pr[\mathbf{X}$ originates from class $j|\mathbf{X} = \mathbf{x}]$.



(a)



(b)

Figure 3: (a) The misclassification ratio and (b) the incorrect saturation ratio of output nodes for the training patterns.

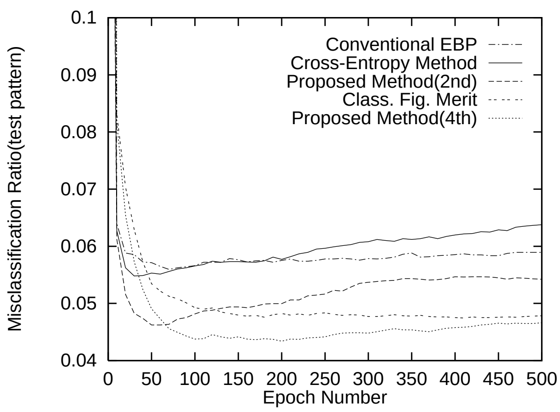


Figure 4: The misclassification ratio for the test patterns.

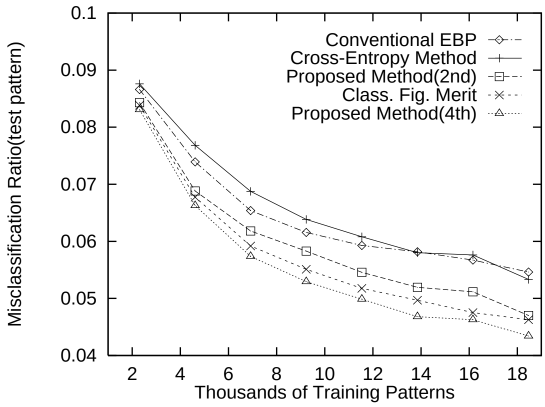


Figure 5: The misclassification ratio for the test patterns vs. the number of training patterns.