A learning method by stochastic connection weight update

A Learning Method by Stochastic Connection Weight Update

Kazuyuki Hara Yoshihisa Amakata∗ Ryohei Nukaga∗ Kenji Nakayama †
Tokyo Metropolitan College of Technology
1-10-40 Higashi-oi, Shinagawa, Tokyo 140-0011
hara@tokyo-tmct.ac.jp
∗ Tokyo University of Agriculture & Technology
† Graduate School of Natural Science and Technology, Kanazawa University

Abstract

In this paper, we propose a learning method that updates a synaptic weight in probability which is proportional to an output error. Proposed method can reduce computational complexity of learning and at the same time, it can improve the classification ability. We point out that an example produces small output error does not contribute to update of a synaptic weight. As learning progresses, the number of the small error examples will be increasing compared to the big one is decreasing. This unbalance will cause of difficulty of learning large error examples. Proposed method cancels this phenomenon and improve the learning ability. Validity of proposed method is confirmed through computer simulation.

1 Introduction

When a set of examples $\{(\mathbf{x_1}, \mathbf{t_1}), \ldots, (\mathbf{x_N}, \mathbf{t_N})\}$ is given, multilayer neural network (MLP) can acquire mapping of input-output automatically. Error back propagation algorithm [1] is one of the most popular learning algorithm for MLP. This scheme is effective to a problem of a classification rule which is not obvious. Acquired mapping depends on an learning example, and there is no guarantee which MLP gives correct mapping to the learning examples for newly generated examples by the same rule. However, even if a new example resembles the learning example, the learned network possibility classify the example correctly. Moreover, if the number of learning examples is increased, mapping of input-output becomes complicated, and the computation complexity of learning which with it becomes difficult and the learning time will be increased. From above reasons, to select the learning example from the given training examples using some criterion is required.

The mean squared error (MSE) distribution of the network output to each example changes with onward movement of learning. For example, on learning progress, the number of examples whereby MSE is small become a majority when MSE of a network becomes small, and large error examples becomes a minority. So that a update of a synaptic weight is proportional to MSE of the network output, a update of a synaptic weight to a small example of an error is small while. It is thought to cut a computation which needs quitting learning to an example whereby a update is small, for learning about. The effort of ignoring some number of small error example from learning example set must be considered. In this case, the distribution of learning examples will be different from the one for entire examples and this regime will possibly affect the learning results.

In the least, because variance is not considered BP method, we cannot learn the example which deviated from an average steeply. So, learning is not properly performed to large error example of small number. Therefore, MLP cannot respond to request of high classification efficiency on putting to practical use of MLP with BP method.

Cachin[2]has proposed a learning method of whereby MLP controlled learning definitely by using a repetition presentation scheme of the big error examples. However, the presentation scheme must be determined for each problem.

We will not update a synaptic weight frequently for a small error example, similar to Cachin's method. However, it does not make definite update, but it is performed in the probability which is proportional to an error. By replacing deterministic update to stochastic one, we can relax the constrain of the scheme. The update accepting function is introduced to determine the probability of updating of the connection weight propor-

0-7803-7044-9/01/$10.00

tional to largeness of the error. The distribution of the output error is considered so the output error is scaled so as to update the connection weight for the largest error example with probability of 1.

In this paper, the difficulty of learning a large error example is analytically explained. Then method of balancing the number of small error examples and the large error example is illustrated, and proposed method is given. The validity of propose algorithm is confirmed through computer simulation.

2 Structure of Network and Output Error

In this paper, we employ a two layer multilayer perceptron (MLP) consist of the input layer, one hidden layer and the output layer. We apply MLP for classification problem, so the number of output units is the same number as classes of the problem to be solved. The number of input units is as the same as the number of input dimension. The activation function in the hidden layer and the output layer is the sigmoid function. The sigmoid function $f(\cdot)$ is calculated by using input potential of the unit $net(\cdot)$ as follows.

$$y_j = f(net_j) = \frac{1}{1+\exp(-net_j)} \tag{1}$$

$$net_j = \sum_{i=0}^{N} x_{\mu i} w_{ij}^{(n)} \tag{2}$$

Here, w_{ij} is the synaptic weight between i-th input unit and j-th hidden unit at n-th iteration. $x_{\mu i}$ is i-th element of μ-th example. This can be apply for calculation at the output layer.

We use the mean squared error (MSE) as the output error of the network. MSE is calculated by the followings:

$$E(\mathbf{W}^{(n)}) = \frac{1}{P}\sum_{\mu=1}^{P}\frac{1}{K}\sum_{k=1}^{K}(t_{\mu k} - y_k)^2 \ . \tag{3}$$

In this equation, $\mathbf{W}^{(n)}$ is a connection weight vector at the n-th iteration, , y_k is the k-th output unit, $t_{\mu k}$ is the k-th target for the μ-th example.

We employ the error back-propagation (BP) algorithm in batch mode as a learning rule of the network.

3 Error Distribution and Update of Synaptic Weight

In this section, we analyze relations of error distribution and the synaptic weight of update concerning to the error distribution.

3.1 Error distribution

The error distribution for each example is related to the differential of the activation function at the output unit. From equation (1), the differential of the activation function is shown in Fig. 1.

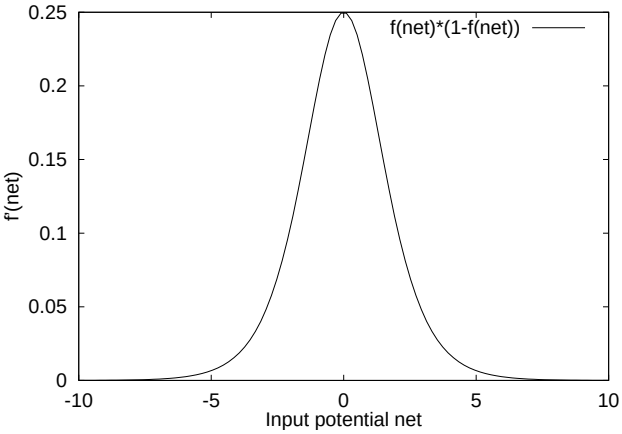

Figure 1: Differential of the activation function

We assume that the initial synaptic weights are set to small random value. In this case, correlation between each synaptic weight is small and input potential of the hidden units will be small. The same regime will occur in the output layer. So, for every unit, $net \sim 0$. In this case, the network error is large for every example. And from figure 1, because input potential for each example is distributed near the zero, then the error distribution will be wide.

When the learning is processed and the MSE becomes small, the network error for large number of examples will be small. In this case, input potential of the output unit will be large. So, from figure 1, error distribution will be narrow.

Therefore, the error distribution for large error is wide and the distribution for small error is narrow.

3.2 Batch Learning and Weight Update

In the BP algorithm, the update of the synaptic weight w_{ij} at n-th iteration, $\Delta w_{ij}^{(n)}$, is calculate by the next equation.

$$\Delta w_{ij}^{(n)} = \eta \delta_j^{(n)} \cdot f'(net^{(n)}) \cdot y_i^{(n)} \quad (4)$$

Here, $y_i(\cdot)$ is the i-th output unit in the hidden layer, $f'(\cdot)$ is differential of the activation function of an output unit, $net(\cdot)$ is a input of a output unit for example $\mathbf{x}_\mu$, η is the learning rate. From this equation, it is shown that for the same network, a update of a synaptic weight is proportional to an error, δ_j.

In batch learning, we calculate the update of the synaptic weight by average over the update for all learning examples. This mean that the connection weights are adapted to the example which error is equal to the MSE. Therefore, in batch learning, update of the synaptic weight is not proportional to the error of each examples, but the MSE. So, if there is an example of large error and MSE is small, the update is not performed for large error example properly.

3.3 Error Distribution and Weight Update

In this subsection, effect of change of the synaptic weight to the error distribution is discussed.

The error distribution is a function of the synaptic weight, and the input vectors. The other hand, we focus our attention on the mean and the variance as main character of the error distribution. So, we assume that the error distribution is Gaussian distribution characterized by the synaptic weight w and the entire input vector $\mathbf{x}$.

$$y(net(w, \mathbf{x})) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(net(w, \mathbf{x}) - \hat{z})^2}{2\sigma^2}\right) \quad (5)$$

Here, $net(w, \mathbf{x})$ is the input potential of the output unit and $\hat{z}$ is the average. We denote output of the output unit y as $y(net)$ to explicit the output is a function of input potential of the unit. In this equation, only synaptic weight w is the parameter related to the synaptic weight update in every iteration, so the synaptic weight is concerned for the analysis. Then equation 5 is rewritten as

$$y(w) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(w - \hat{z})^2}{2\sigma^2}\right) \quad (6)$$

Assuming learning rate η is small, so update of the synaptic weight will be small and the difference of the distribution due to update of the synaptic weight vector can be considered as $\partial y/\partial w$. $\partial y/\partial w \propto \sigma$. From this fact, when variance σ is large, the difference of distribution for change of synaptic weight is large and for small distribution, the difference is also small. Therefore, it is confirmed that the MSE is calculated integrate the equation 6, so the contribution for the MSE of the example located far from the average $\hat{z}$ is small when the distribution is small then the example is not considered to update of the synaptic weight.

4 Non-Definite Update of Synaptic Weight

We propose method of learning whereby MLP updates a synaptic weight in probability which is proportional to the size of the error.

We employ batch learning to calculate error distribution of learning examples for the same network.

4.1 Update Accepting Function

As described in subsection 3.1, the error distribution is related to the MSE. When the MSE is large, the error distribution is wide, and the distribution is narrow when the MSE is small. The wideness of the error distribution is described by variance σ. It is also discussed that when the variance of the error distribution is small and there still remained some number of large error examples, these examples are not considered updating of the synaptic weight. Above fact affect the learning property, and examples which produce large error must be considered for updating of the synaptic weight to achieve higher accuracy.

To solve this problem, the number of examples of large error and small error must be balanced. To realize the balance, the following probability density function is introduced. Update probability of synaptic weight is denoted as the next equation.

$$Prob(q = 1) = \frac{\exp(m \cdot E'(\mathbf{w}^{(n)}; \mathbf{x}_\mu)) - 1}{\exp(m) - 1} \quad (7)$$

Here, a synaptic weight is updated when q=1, and it will not updated in the case of q=0. Therefore, we provide for probability whereby Prob(q=1) updates a synaptic weight to learning example $\mathbf{x}_\mu$.

$E'(\mathbf{w}^{(n)}; \mathbf{x}_\mu)$ is MSE whereby it was normalized by the following equation.

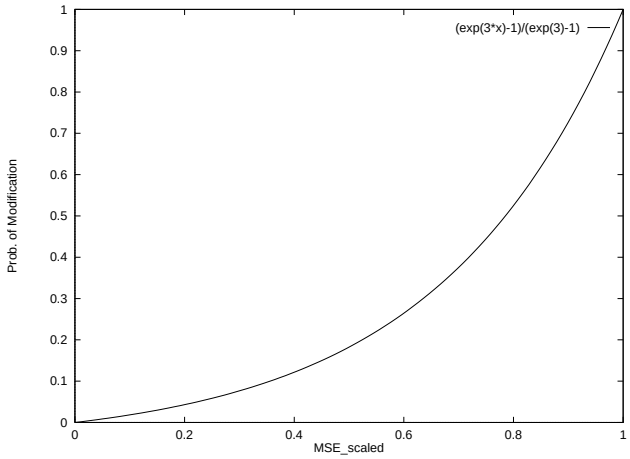

Figure 2: Update accepting function

$$E'(\mathbf{w}^{(n)};\mathbf{x}_\mu) = \frac{E(\mathbf{w}^{(n)};\mathbf{x}_\mu)}{\arg max_\mu\{E(\mathbf{w}^{(n)};\mathbf{x}_\mu), \mu = 1 \dots P\}} \tag{8}$$

By this operation, $E'(\mathbf{w}^{(n)};\mathbf{x}_\mu)$ is in a range of $0 \geq E'(\mathbf{w}^{(n)};\mathbf{x}_\mu) \geq 1$. Then the update probability of synaptic weight for maximum example of a error is 1. Beside, update is done with probability so, synaptic weight will adjusted for small example of a error.

4.2 Learning Algorithm

We show learning algorithm. This is repeated until (iii) is hold.

1. calculate $\{E(\mathbf{w}^{(n)};\mathbf{x}_\mu), \mu = 1 \cdots P\}$ for all the examples.
2. Calculate classification rate.
3. Stop the learning when one of the next condition is hold. (i)Classification rate is 1.0 (ii) MSE<0.01 (iii)Iteration $n > 10000$.
4. Normalize MSE for each example by equation 8, then calculate update probability of a synaptic weight. Update a synaptic weight if $q = 1$.

5 Computer Simulation

5.1 Problem

We employ a two-dimensional Gaussian distribution of two class. Class regions are overlapped each other. Haykin compared classification performance between Bayesian decision boundary and BP method by using this problem[3]. Theoretical bound of the classification rate of Bayesian Decision Boundary is 81.5%. We show distribution of examples in Fig. 3.

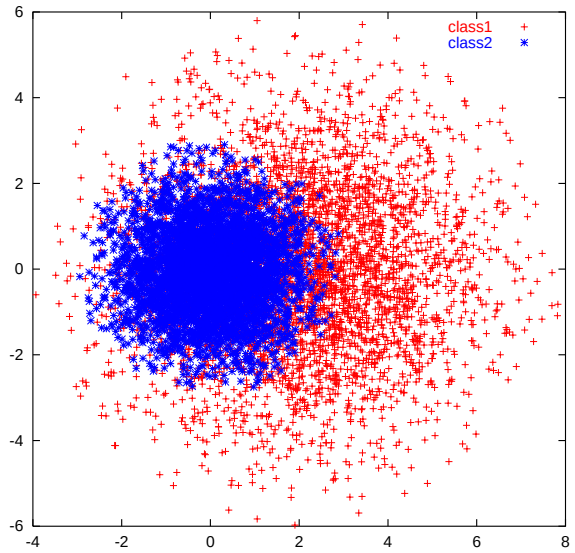

Figure 3: Distribution of data of problem.

The average vector of class 1 is $\mu_\mathbf{1} = [\mathbf{0}, \mathbf{0}]^\mathbf{T}$, the variance is $\sigma_1^2 = 1$. The average vector of class 2 is $\mu_\mathbf{2} = [\mathbf{2}, \mathbf{0}]^\mathbf{T}$, the variance $\sigma_2^2 = 4$. The number of class 1 and class 2 are 2000. To evaluate generalization, we used examples of the same number of unlike the example which we used for learning.

For proposed method and BP method, we used two layer multilayer perceptron consist of input layer, one hidden layer and output layer. Learning rate $\eta = 0.5$, momentum term $\alpha = 0.98$. m of equation 7 is set to 3.

5.2 Result

Proposed method achieved 81.5% classification rates to a non-learning examples while BP method achieved 81.1%. This is the same as the bound of classification of 81.5%. Computational complexity for proposed method is 2388 while BP method needs 10000. From this result, proposed method cut a computational complexity of BP method substantially.

In the figure 6 we show error distribution of after learning of BP method and proposed method. From the figure, error is widely distributed up to 0.8 for BP method. Error distribution of proposed method is mostly near the error of 0.2 and up to 0.4. From the result, proposed method reduced big example of an error.

5.3 Analysis of Result

Update of a synaptic weight by learning using an error of a network. Therefore, we often use an error as a criterion whereby we judge efficiency of a network. For classification problem, we want to classify examples into several classes. For this, a class boundary formed by hyperplane composed by synaptic weights is used. If target t_μ is 1 for example $\mathbf{x}_\mu$, following inequalities can perform classification.

$$\begin{cases} y(\mathbf{x}_\mu) > \mathbf{0.5} & \mathbf{x}_\mu \in \mathbf{X_1} \\ y(\mathbf{x}_\mu) < \mathbf{0.5} & \mathbf{x}_\mu \in \mathbf{X_2} \end{cases} \tag{9}$$

Therefore, for classification problem, it does not need to be a small MSE if example is in a correct class area. From the figure 6, examples are distributed near the class boundary, that is MSE of 0.25 by proposed method. However, most of the examples are in correct class area. By BP method, most of the examples are in small error area, but remained examples are still in big error area. This produces small MSE but there still have missclassifications. Figure 4 shows MSE curve of BP method and proposed method. Horizontal axis is computation complexity and Vertical axis is MSE. From the figure, we can see that the MSE of proposed method is larger than BP.

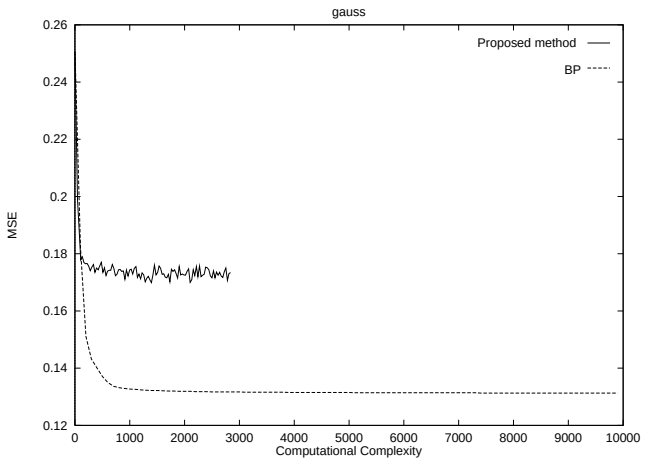

Figure 4: MSE Curve. BP and proposed method

5.4 Biased Classification Problem

In this subsection, the proposed method is applied to the biased classification problem. The problem is depicted in figure 5.

In this problem, number of the examples included in class 1 is 5000, and the class 2 is 500. So, examples of class 2 is minority in the learning.

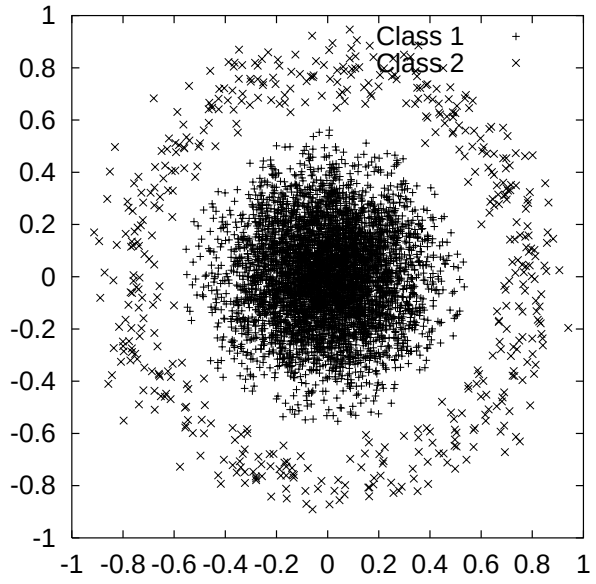

Figure 5: Biased classification problem

For this problem, the proposed method achieved high accuracy. The classification rate is 99.96%. BP achieved 93.33% classification rate. We investigated the classification result by BP and observed that non of class 2 data is classified by BP correctly. This result presents that the proposed method can learn biased classification problem by balancing the number of examples to be learn.

6 Conclusions

We have proposed learning method whereby it updates a synaptic weight in probability which is proportional to the error. We have pointed out that a example whereby an error is small does not contribute to update of a synaptic weight. As learning progresses, the number of small examples of an error increasing compared to the big ones decreasing. By using these two phenomena, proposed method can drastically reduced computational complexity of learning, and at the same time, classification performance is improved.

This work partly supported by Grant-in-Aid for Scientific Research Japan Society for the promotion of science #13680472.

References

[1] D. E. Rumelhart and J. L. McCelland. : Parallel Distributed Processing. Cambridge, MA: MIT Press. (1986)

[2] C. Cachin. : Pedagogical pattern selection strategies, *Neural Networks* **7**(1) 175-181. (1994)

[3] S. Haykin. : Neural Networks–A Comprehensive Foundation, pp. 165-176. *Macmillan College Publishing Company* (1994)

[4] K. Hara, K. Nakayama, A.A.M.Ashraf. : A Data Selection Training in On-Line for Multilayer Neural Networks, Proc. *IJCNN* pp. 2247-2252. (1998)

[5] K. Hara, K. Nakayama. : A Training Method with Small Computation for Classification, *IJCNN* pp. III-543-548 (2000)

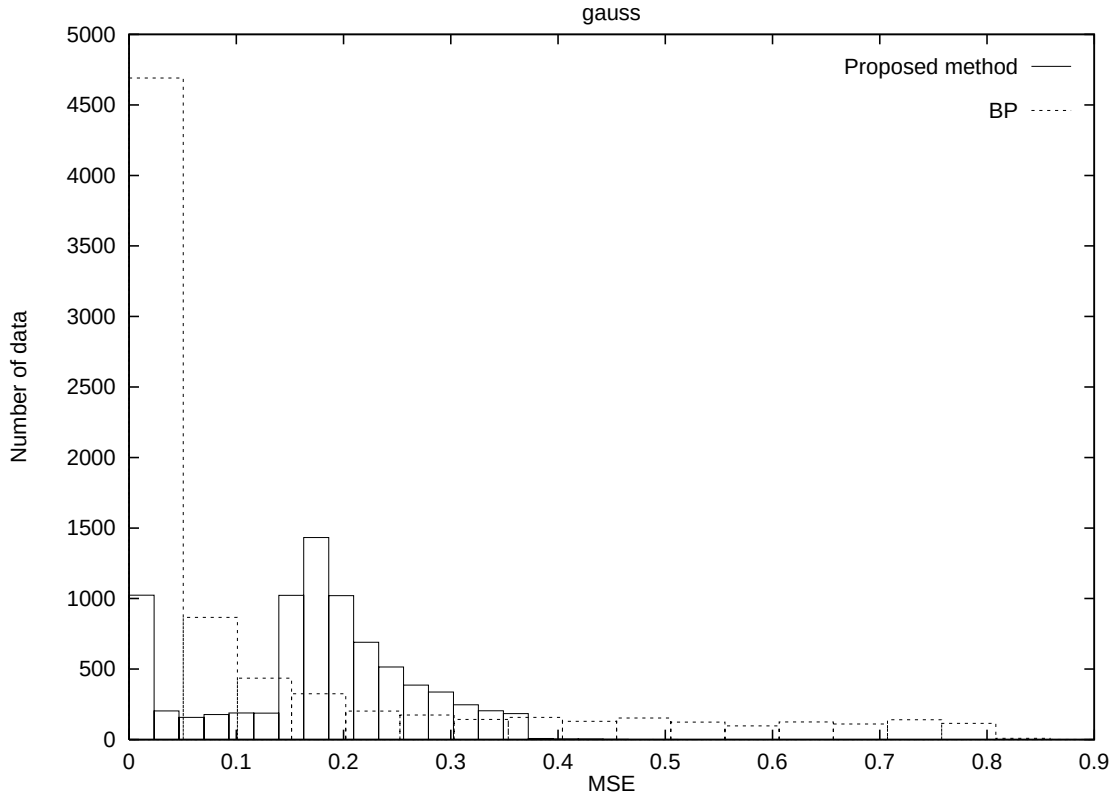

Figure 6: Distribution of data of problem.