LOCAL GRADIENT LEARNING RULES FOR NEURAL NETWORKS BASED ON MODERATISM

M. Tanvir Islam [1] , and Yoichi Okabe [2]

[1] Department of Electronic Engineering, University of Tokyo, Komaba 4-6-1, Tokyo 153-8904, Japan.
E-mail: tanvir@okabe.rcast.u-tokyo.ac.jp
[2] Department of Electronic Engineering, University of Tokyo, Komaba 4-6-1, Tokyo 153-8904, Japan.
E-mail: okabe@okabe.rcast.u-tokyo.ac.jp

ABSTRACT

Moderatism [1][2], which is a learning rule for ANNs, is based on some experimental results showing that neural networks tend to sustain a "moderate" level in their input and output signals so that the feedback signal from the surrounding environment is not too high or low. In this way, a close mutual relationship between the neural net and the outside environment is maintained. In this paper, the theoretical background of Moderatism is discussed. Then, two potential Moderatism-based local, gradient learning rules are proposed. Finally, a pattern learning experiment is performed to compare the learning performances of these two learning rules, the Error Based Weight Update (EBWU) rule [4][5], and Error Backpropagation [3].

1. INTRODUCTION

Artificial neural networks, especially the multi-layer neural networks have long been used in the field of pattern learning for their competence in learning. Among the learning algorithms for multi-layer neural networks, the most well known is Error Backpropagation (BP) [3]. Moderatism [1][2], on the other hand, is a learning model of neurons that models a biological learning characteristic of neither receiving nor transmitting too strong or weak signals. In our previous papers [4][5][6] showed that a learning rule called "Error Based Weight Update" (EBWU) performs better than BP in some pattern learning experiments. However, it was seen that when the number of outputs increase, the learning performance deteriorates. To solve this problem, gradient-based learning rules are a good solution. In [6] we showed that the inclusion of the cost of Moderatism in the error function of BP increases the average learning performance. In this paper, we propose two gradient learning rules that are based on Moderatism. The developments from [6] are the followings. Firstly, instead of using the gradient of the sum of the error of BP and the cost of Moderatism, only the gradient of the cost of Moderatism is used in these two learning rules, implementing Moderatism directly. Secondly, unlike BP, in the gradient descent local learning rules, each individual weight updates according to its influence on the AC cost of that particular weight, not the cost of global outputs. And finally, unlike EBWU rule, these two new rules can be applied to networks with multi outputs without performance deterioration. Also we shall show the comparison of performances of these learning rules with Error Backpropagation and EWBU rule.

2. MODERATISM

The theoretical background of Moderatism comes from the concept of handling feedback signals from the surrounding environment. In general, the feedback signal from the environment to the brain represents either the penalty or the reward for any action ordered by the brain. In either case, the amplitude of the signal can be equally high. Therefore, it is almost impossible to know beforehand what type of feedback signal will come next fro the environment. So, in an unknown environment, the best way to survive is to act in way so that the amplitude of the feedback signal from the environment is neither too big nor too small, and holds to a "Moderate" level. In this way it is possible to process signals of penalty and reward in the same manner.

About the output signal of the neural net to the environment, it can be assumed that if the output of the

brain (or neural network) is extremely big or small, then the feedback signal from the environment will not be at the moderate level. Therefore, there should be a moderate level in also the output signal from the network. The concept that moderate levels should be present in a neural network's input and output signal is called "Moderatism". A psycho-physics experiment done in [2] shows support for this concept. The experimental setup is showed in figure 1.

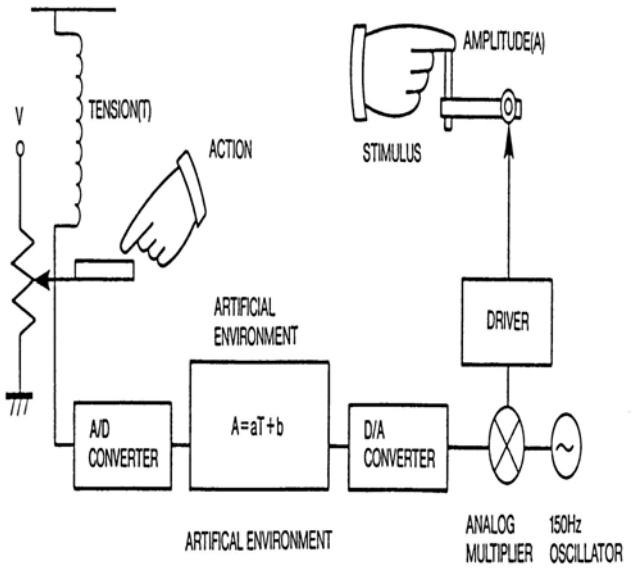

Figure 1: The experimental setup.

In this experiment, a human represents the neural net or the biological body. The subjects of the experiment receive environmental feedback in the form of the amplitude of the oscillation (150 Hz) of a pin with one finger. With the other finger, the subjects can change the tension of the spring (k = 140 gw/cm), which is linearly related to the feedback signal from the artificial environment. In the experiment, for a particular value of *a*, the value of *b* is changed once in every 30 seconds, and the tension of the spring and the amplitude of the pin are measured. This process is done for *a* = -0.5, -0.25, 0.25, 0.5. It was observed that the resulting points lie on a straight line for every value of *a* (for example, figure 2 shows the case of *a* = -0.5), for one particular subject. When these straight lines are plotted together (figure 3), they intersect in one point. This point represents the Moderate level of input and output signal for a particular subject for this experiment. Therefore, the difference between the input value and the respected moderate value is the cost function for input, while in the same manner we can define the cost function for output. The total cost is the sum of these two cost functions.

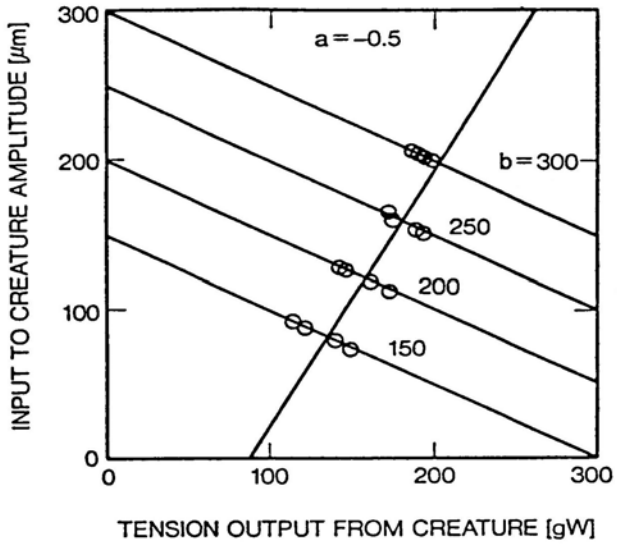

Figure 2: Tension output from one subject for *a* = -0.5

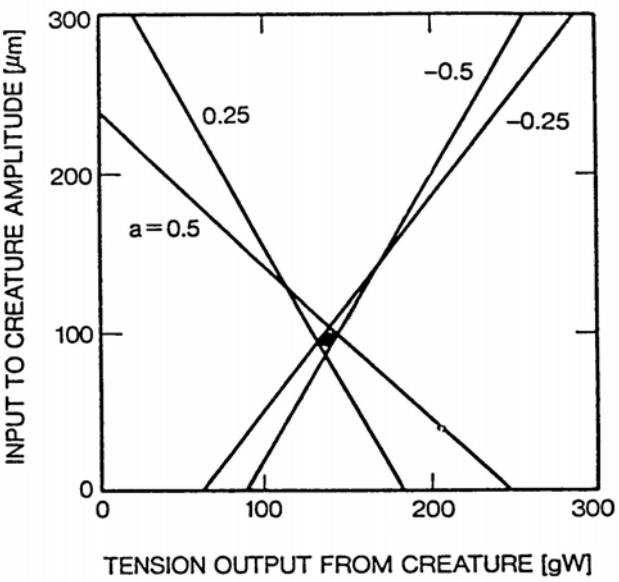

Figure 3: Moderate input and output for one subject.

From here we shall go through the mathematical model of Moderatism. It has been supposed by the results of this experiment that biological neurons neither prefer to receive too strong signals from environment nor they want to be in a totally separated environment. This idea is the base of Moderatism*: Neurons continuously change their internal state in order to preserve a suitable input-output relationship with the environment*. All the neurons try to minimize a cost function that expresses the deviation of the inputs and outputs from the suitable moderate level. The dynamics is showed in figure 4.

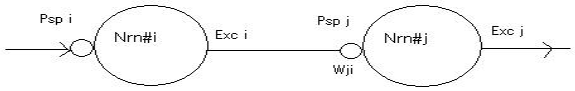

Figure 4: Neuron-Synapse model.

The synaptic weight between neuron $\#i$ and $\#j$ is w_{ji}, the thresholds of the two neurons are θ_i, θ_j, inputs to the neurons are Psp_i, Psp_j and the outputs are Exc_i, Exc_j respectively. The input signal coming to neuron $\#j$ from neuron $\#i$ is:

$$Psp_j = w_{ji} \times Exc_i \tag{1}$$

The output of neuron $\#j$ is

$$Exc_j = Sigmoid(Psp_j - \theta_j) \tag{2}$$

Where

$$Sigmoid(x) = \frac{1}{(1 + e^{-4(x-0.5)})} \tag{3}$$

The variable parameters in this case are the synaptic weights (w_{ji}) of the network. The calculation of signals and cost are given in equation (4).

$$\begin{aligned}
&Exc = Sigmoid(Input - \theta)\\
&DcPsp = mean(Input)\\
&DcExc = mean(Exc)\\
&AcPsp = |Input - DcPsp|\\
&AcExc = |Exc - DcExc|\\
&CostPsp = |AcPsp - AcMod|\\
&CostExc = |AcExc - AcMod|\\
&CostPsp(w_{ji}) = CostPsp_i + CostPsp_j\\
&CostExc(w_{ji}) = CostExc_i + CostExc_j\\
&AcCost(w_{ji}) = CostPsp(w_{ji}) + CostExc(w_{ji})
\end{aligned} \tag{4}$$

Here, $mean(x)$ is the average of x over a certain period of time, $AcCost(w_{ji})$ is the cost of weight w_{ji} that needs to be lessened through weight update, and $AcMod$ is the moderate value of the network.

3. ERROR BASED WEIGHT UPDATE

To make the neural networks moderate, only the weights with high *AcCost* will have to be modified. Also, like Backpropagation, the expected outputs are known beforehand and the difference between the expected output and network output is used to update the weights. Again, the quantity to which a particular weight will update is determined by the importance of that weight. However, unlike the gradients used in BP, in this algorithm the importance is calculated by considering the number of the layer in which the weight exists in the weight update equation. The weight update rule is given in equation (5).

$$\begin{aligned}
&\Delta w_{ji} = -\alpha[i][j] \times AcCost(w_{ji}) \times (Exc_{out} - Expected)\\
&\alpha[i][j] = (p+1) \times \frac{1}{q} \times error_rate \times A(const)
\end{aligned} \tag{5}$$

Here, $error\_rate$ is the percentage of wrong outputs in a period of time, Exc_{out} is the network output, and $Expected$ is expected output for the input. Also, p is the layer number of the weight w_{ji} and q is the number of weights in that layer. It is clear from the equations that with training going on, the $error\_rate$ decreases and as a result the weight update rate also decreases to zero. Again, the cost terms are used so that the value of Δw_{ji} is smaller for already moderate synaptic weights.

4. ERROR BACKPROPAGATION

Error Backpropagation is a gradient descent method for training the weights of a multi-layer NN. For a given problem, there is a set of training vectors X such that for every vector $x \in X$, there is an associated desired output vector $d \in D$, where D is the set of desired outputs associated with the training vectors in X. Let the instantaneous error E_p be defined as:

$$E_p = \frac{1}{2}(d_p - z_p)^T (d_p - z_p) = \frac{1}{2}\sum_{k=1}^{N} (d_{k,p} - z_{k,p})^2 \tag{6}$$

Where $d_{k,p}$ is the k_{th} component of the p_{th} desired output z_p when the p_{th} training exemplar x_p is input to the multi-layer perceptron. In BP, the change of weight is proportional to the gradient of this error:

$$\Delta w_{ji}^{t} = -\alpha \frac{\partial E_p}{\partial w} + \eta \Delta w_{ji}^{t-1} \tag{7}$$

Where α, which is the learning rate, is some small positive number between 0 and 1. η, the momentum factor is also a small positive number (between 0 and 1), and w_{ji} represents any single weight in the network. In the above equation, Δw_{ji}^{t} is the change in the weight computed at time t.

5. LEARNING PROBLEM AND NETWORK STRUCTURE

The network structure is based on "watcher-environment" model given in figure 5. To learning problem used in this paper is a 20 (5×4) pixel digit recognition problem. The numbers 0-9 are used as digit patterns. The pixel values used are 0.2 (low), 0.5 (half occupied), and 0.8 (high). For each digit, four variations are made by changing the pixel values by a small value, thus creating 40 patterns as the training set. The neural network that was used for training has 20 input neurons, 10 hidden neurons and one output neuron. The expected output for digit pattern 0 is 0.2, for digit pattern 1 is 0.25 and so on. The output of the network is then sent to the "environment", where the learning error is determined. For simplicity, the environment is set to a function that calculates the sum of squares of the learning errors for the training patterns. Then this environment sends back its response (in this case the sum of the squared errors) to a "watcher" neuron, which symbolizes the sensory input from the environment. The output of the watcher neuron is used as the amplitude of a sine curve, which is given as feedback to the input neurons, along with the input patterns. As the network tries to keep the AC signals to a very small moderate value, the weights are updated in a way so that the sine wave deteriorates, thus minimizing the amplitude of the watcher, and as a result minimizing the training error.

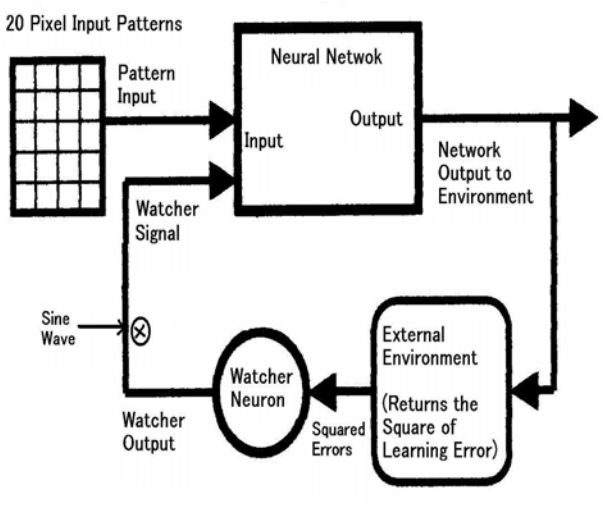

Figure 5: The "watcher-environment" model.

6. TWO MODERATISM-BASED GRADIENT LEARNING RULES

In this section we propose two local learning rules based on Moderatism. If we take the gradient of *AcCost* in equation (4) with respect to weight w_{ji}, we find:

$$\frac{\partial AcCost(w_{ji})}{\partial w_{ji}} = (exc_i - DcExc_i) + \frac{exc_i \times \exp(IN_j)}{(1+\exp(IN_j))^2} - \frac{DcExc_i \times \exp(DcIN_j)}{(1+\exp(DcIN_j))^2}$$
$$IN_j = \sum_k w_{jk} \times exc_k \qquad (8)$$
$$DcIN_j = \sum_k w_{jk} \times DcExc_k$$

The first gradient learning rule is found like equation (9).

$$\Delta w_{ji}^t = -\alpha \times \frac{\partial AcCost(w_{ji})^t}{\partial w_{ji}} \qquad (9)$$

Where α is the learning rate. We can see from equation (9) that the learning rule is local, unlike BP. Each weight is updated according to the cost that particular weight bears, thus following the principles of connectionism. It is expected that the network will minimize the learning cost as individual weights minimize their own costs, resulting the minimization of the output signal of the watcher neuron.

We thought that the weight update rule should be dependent on the *AcCost* itself, as was the case in the EBWU rule. So, as the second gradient rule, we included *AcCost* in the right side of the equation (9):

$$\Delta w_{ji}^t = -\alpha' \times AcCost(w_{ji})^t \times \frac{\partial AcCost(w_{ji})^t}{\partial w_{ji}} \qquad (10)$$

Where $\alpha' \ll \alpha$. It should be noticed that the performances of the two learning rules stated in equations (9) and (10) depend on the learning rates. Learning is faster if the learning rate is gradually increased up to a limit, after which the learning rules do not converge. As we shall see in the following figures 6-8, in the experiments used in this paper, the achieved upper limits of α and α' are 0.002 and 0.0001 respectively. However, for the EBWU rule, the upper limit of learning rate (constant A) was 0.00005.

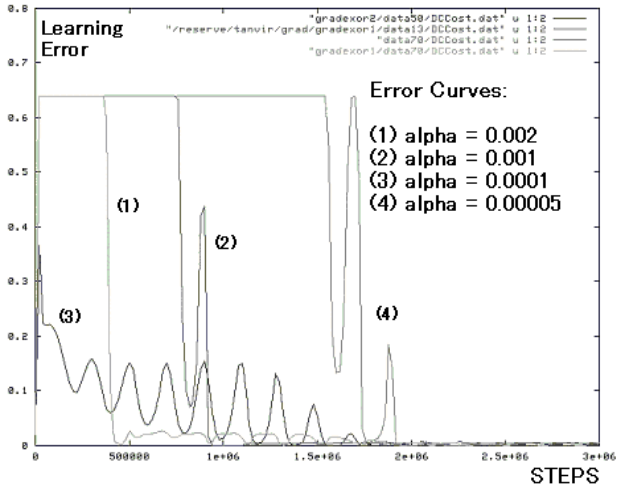

Figure 6: The change of learning performance with changing α for the rule of equation (9).

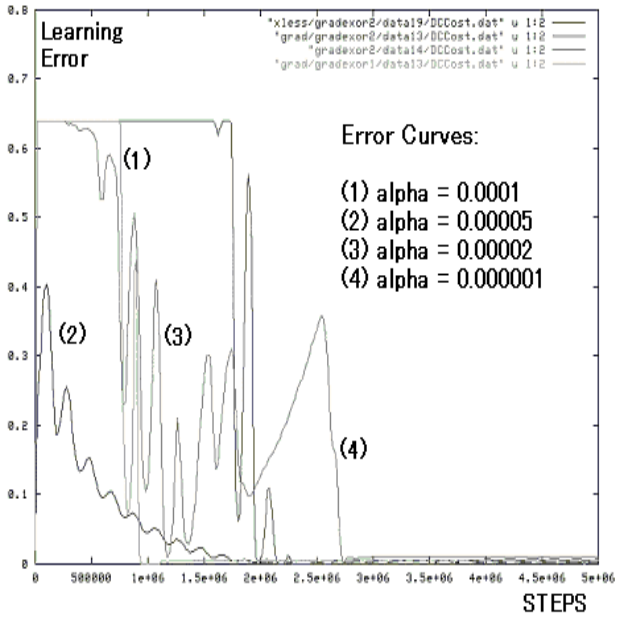

Figure 7: The change of learning performance with changing α' for the rule of equation (10).

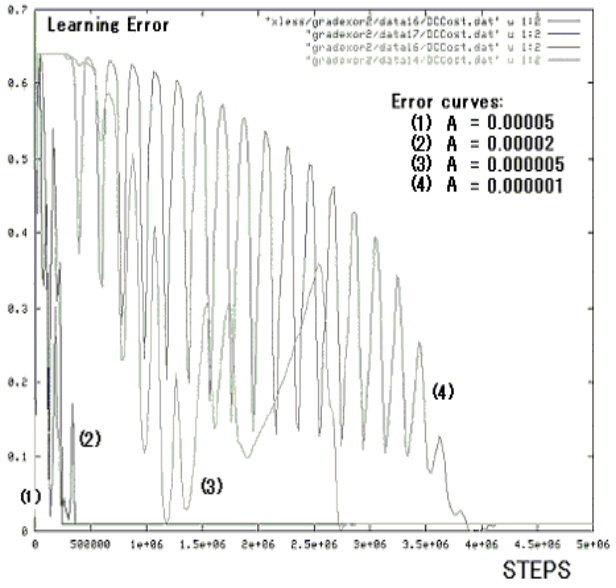

Figure 8: The change of learning performance with changing A for the EBWU rule.

Figure 9 shows that the neural AC excitations of the neurons reach the AC moderate level during simulation, and maintain it. The ACMOD value used here is 0.2 .

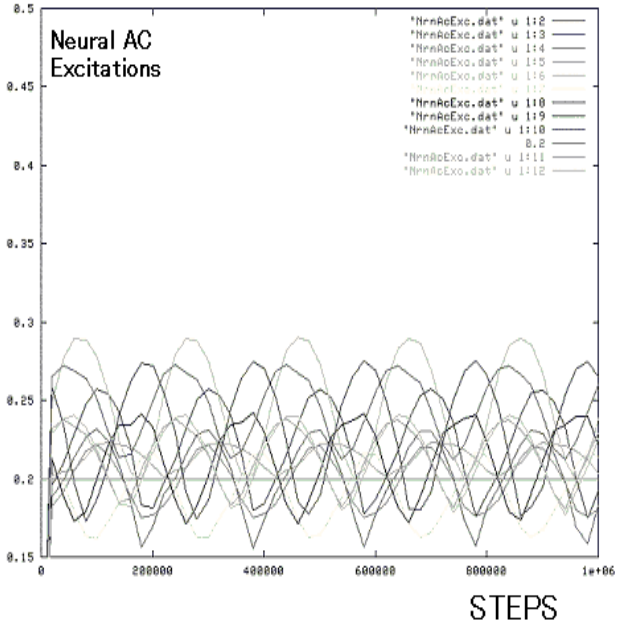

Figure 9: The neural AC excitations of the network when ACMOD is set to 0.2 .

7. COMPARISON OF LEARNING PERFORMANCES

In this section we show the simulations and results of the learning experiment. The training and test sets of patterns described in section 5 are used in the network structure from figure 5. The network is trained with EBWU rule, the two rules of equation (9) and (10), and finally BP. For each learning rule, the network is trained 10 times. Then an average of those 10 trainings was taken for comparison. As for learning rates, the learning rate for which a learning rule performs best is chosen. For BP, the values of α and η was 0.5 and 0.9 respectively. Figure 10 shows the average training performances of the four learning rules.

We can see from figure 10 that although BP takes the shortest time for learning, there is not a big difference among them. The EBWU rule performs almost as fast as BP, however the precision of learning was less than BP. On the other hand, the gradient learning rules of equation (9) and (10) show similar precision as BP, with learning time of the same order. However, we think the two gradient learning rules can achieve better performance in future with optimization and inclusion of a “momentum” term like BP.

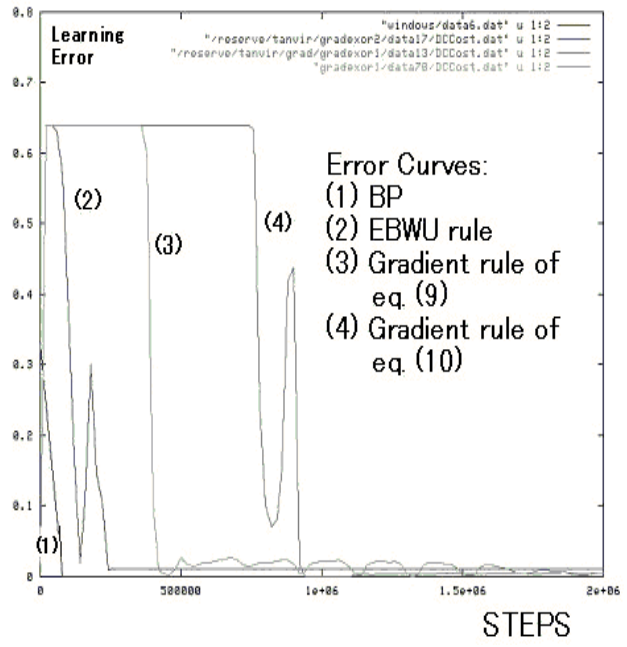

Figure 10: The comparison among learning performances of the four learning rules.

8. CONCLUSION

In this paper we have discussed the theoretical background of Moderatism, and explained the concept by earlier experiment. We have formed a mathematical model of Moderatism to conduct learning experiments for artificial neural networks. Then we proposed two local gradient learning rules that are theoretically based on Moderatism. These learning rules are implemented on a neural network model that follows the principles of connectionism. We found that these two learning rules show good performance in the learning experiments. Although they require more convergence time than BP and EBWU rule, they show learning precision as good as BP and better than EBWU rule. In future we want to optimize these learning rules and perform further learning experiments with them. For now we consider these learning rules as potentially effective Moderatism-based local learning rules.

9. REFERENCES

[1] Y. Okabe, "Moderationism: Feedback Learning of Neural Networks", *Proceedings of 1988 Intn'1 Industrial Electronics Conference (IECON'99)*, IEEE, pp. 1028-1033, 1988.

[2] T. Kouhara, and Y. Okabe, "Learning algorithm based on Moderationism for multilayer neural networks", *Proc. of 1993 Intn'l Joint Conf. On Neural Networks*, vol. 1, pp. 487-490, 1993.

[3] Cristopher M. Bishop, "Neural Networks for pattern recognition", *Oxford University Press*, 1995.

[4] M. Tanvir Islam, and Y. Okabe, "A New Pattern Learning Algorithm For Multilayer Feedforward Neural Networks", *In Proceedings of International Conference on Computer and Information Technology (ICCIT2001)*, pp. 251-255, December 2001.

[5] M. Tanvir Islam, and Y. Okabe, "Pattern Learning by Neural Networks Based on Moderatism", *11th Annual Conference of Japanese Neural Network Society, JNNS 2001*, pp. 129-130, September 2001.

[6] M. Tanvir Islam, and Y. Okabe, "Pattern Learning by Multilayer Neural Networks Trained by A Moderatism-Based New Algorithm", *In Proceedings of the International Conference On Neural Information Processing (ICONIP02)*, (CDROM), November 2002.