2016 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2016, Yugawara, Japan, November 27th-30th, 2016

Multilayer Perceptron Including Different Amplitude Random Noise

Chihiro Ikuta† Yoko Uwate‡ and Yoshifumi Nishio‡

†Course of Electrical Engineering, Dept. of Creative Technology Engineering,
National Institute of Technology, Anan College
265 Aoki, Minobayashi, Anan, Tokushima 774-0017, Japan
‡Dept. Electrical and Electronics Engineering, Tokushima University
2–1 Minami-Josanjima, Tokushima 770-8506, Japan
Email: ikuta@anan-nct.ac.jp, {uwate, nishio}@ee.tokushima-u.ac.jp

Abstract—In this study, we introduce different amplitude random noise to a multilayer perceptron (MLP). We divide the neurons in hidden layer to some groups according to value of neuron output. Each neuron group is input the different amplitude random noise. When a neuron group has a large output, this neuron receives a large amplitude noise. When a neuron group has a small output, this neuro receives a small amplitude noise. The group member is dynamically changed during the learning because the neuron output changes with the learning. By simulations, we confirm that the proposed MLP performance is better than the standard MLP and MLP having simple noise input method. Moreover, we show the parameter dependency of the proposed MLP by changing the number of groups and the noise amplitude.

1. Introduction

Multilayer perceptron (MLP) is one of feed forward neural network which is applied to pattern recognition, data mining, and so on. In the MLP, neurons make some layers and connect with neurons in other layer. The MLP output is decided by the weight of connection between the neurons. In general, we use the back propagation algorithm (BP) [1] for a learning of weight of connection. The BP uses the steepest decent method, thus the network is often traps into local minimum. Then, the network cannot escape out from the local minimum. For a problem of local minimum, some methods are proposed such as a pre-training, a noise input, and an annealing [2][3].

In this study, we propose an MLP including different amplitude random noise for escaping out from the local minimum. We use the MLP of three layers and input a uniformed random noise to the inner state of the neurons in hidden layer. In general, the uniformed random noise is not efficiency to the escaping out from the local minimum. If amplitude of the random noise is larger, the network cannot learn the supervised signal, because many neuron outputs become random. On the other hand, if amplitude of the random noise is smaller, the network falls into local minimum. In our method, we divide some groups according to the value of neuron outputs. After that we input the different amplitude random noise into the neurons in descending order of output value. The neuron having large output value obtain the large amplitude random noise, and the neuron having small output value obtains small amplitude random noise. By this method, the network can learns the supervised signal under influence of the random noise and escaping out from the local minimum. By the simulation, we confirm that the different amplitude random noise is efficiency to the MLP learning, and dependency of the dividing number of neuron groups.

2. Proposed Method

The proposed MLP is shown in Fig. 1. We divide the neurons in hidden layer to some groups according to the output value of neuron. Each group has different amplitude random noise, and this noise is input into the neurons. In the same group. the neuron has same amplitude random noise. The group of the neuron having a large output obtains the large amplitude noise. On the other hand, the group of the neuron having a small output obtains the small amplitude noise.

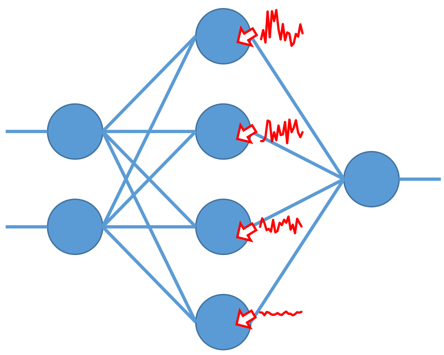

Figure 1: Proposed MLP.

2.1. Neuron Updating Rule

A standard neuron updating rule is described by Eq. (1).

$$y_i(t+1) = f\left(\sum_{j=1}^{n} w_{ij}(t)x_j(t) - \theta_i(t)\right), \tag{1}$$

where y is an output of the neuron, w is a weight of the connection, x is an input of the neuron, and θ is an excitation threshold of the neuron. In this equation, the weight of the connection and the threshold of the neuron are learned based on the BP algorithm. Next, we show the updating rule of the neuron with the uniformed random noise in Eq. 2. This updating rule is used for the neuron in the hidden layer.

$$y_i(t+1) = f\left(\sum_{j=1}^{n} w_{ij}(t)x_j(t) - \theta_i(t) + \alpha_n R(t)\right), \qquad (2)$$

where α is amplitude of the uniformed random noise, and R is a uniformed random noise. Here, the uniformed random noise is given by Mersenne Twister (MT) which was proposed by Matsumoto and Nishimura [4]. The uniformed random noise gives the energy to the network and helps escaping out from the local minimum.

2.2. Different Amplitude Random Noise

The amplitude of the input random noise (α) decreases at an exponential rate. The α is described by Eq. (3).

$$\alpha_n = 0.8^{n-1} \quad (n = 1, 2, \cdots, N). \qquad (3)$$

Figure 2 is an overview of the proposed noise input method. We divide the neurons to some groups according to the value of neuron output. In this figure, six neurons are divided to three groups. L means that neuron has large output value. M means that the neuron has middle output value. S means that the neuron has small output value. Each group has different amplitude random noise. In the same layer, the same amplitude random noise. During the learning, the group is dynamically changed, because the value of neuron output changes with the learning.

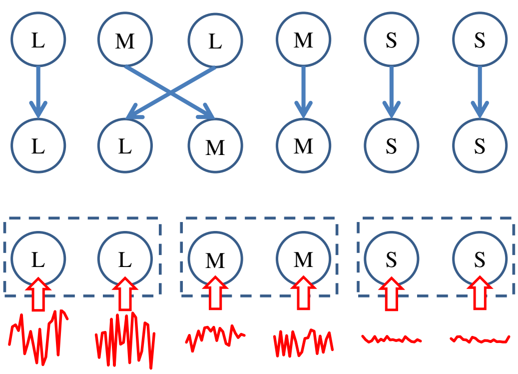

Figure 2: Input method of different amplitude random noise.

3. Simulation

In this section, we show the simulation results. We use four kinds of MLPs which are;

(1) Standard MLP.

(2) MLP with uniformed random noise.

(3) MLP including different amplitude random noise (the neuron group is fixed).

(4) MLP including different amplitude random noise (the neuron group is changed according to the value of neuron output).

The standard MLP (1) does not have the noise, thus this MLP reduces error faster, however it often falls into local minimum. The MLP with uniformed random noise (2) has the unformed random noise in the inner state of neurons in hidden layer. In the MLP including different amplitude random noise (3), we fix the neuron groups during the simulation, thus the neuron receives constant noise amplitude. The MLP including different amplitude random noise (4) is proposed MLP. The number of neurons in input layer, hidden layer and output layer is 2, 40 and 1, respectively. The number of iterations is 50000. We obtain the result from 100 trials from different initial weight of connection. We use the mean square error (MSE) to the error function which is described by Eq. (4).

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(T_n - O_n)^2, \qquad (4)$$

where N is the number of learning datum, T is a target value, and O is an output of MLP. We use the two spiral problem (TSP) for the task of the MLPs. The TSP is famous task for the ANN and has high nonlinearity, thus the standard MLP often falls into local minimum [5][6]. We give the coordinates of the each point to the neuron in input layer, and MLP learns the classification of the spiral point. In this simulation, the two spirals are constructed from 130 points.

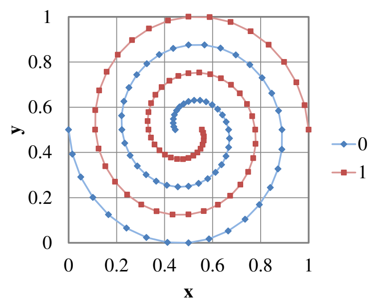

Figure 3: Two spiral problem.

3.1. Comparison learning performance of four MLPs

Firstly, we show the learning performance of each MLP. We compare the four measures which are average, minimum, maximum, and standard deviation (Std. Dev.). The

Table 1: Learning performance

	Average	Minimum	Maximum	Std. Dev.
(1)	0.082	0.000	0.262	0.071
(2)	0.076	0.000	0.257	0.069
(3)	0.076	0.000	0.245	0.073
(4)	0.061	0.000	0.206	0.064

learning performance is shown in Table 1. From this table, every MLP reached to 0.00 in minimum error, thus the MLP can solve this task when optimal initial weight is set. However, the MLP often falls into local minimum. Especially, the average error and the maximum error in the standard MLP (1) is the largest of all. Other MLPs reduce the error from the standard MLP. The MLP with uniformed random noise (2) is similar to the MLP with different amplitude random noise (3). From this result, the different amplitude is not efficiency to the learning performance when we simply introduce the different amplitude random noise to the MLP. The proposed MLP (4) reduces the error than the MLP with uniformed random noise (2) and MLP including different amplitude random noise (3). We divide the neurons in hidden layer to some groups according to the value of neuron output, thus the unadapted amplitude noise is not input to the neuron.

3.2. Dependency of the number of groups and amplitude of the random noise

Next, we change the number of groups in divided neurons in hidden layer and the maximum amplitude of the random noise. Firstly, we show the result of the MLP with uniformed random noise. In the MLP with uniformed random noise, we do not divide the groups, thereby we only show the result of changing maximum amplitude of random noise. In this case, the MSE decreases to the amplitude 0.01, after that the MSE increases with the maximum amplitude. From this result, the allowable noise amplitude is small in the MLP learning. If the noise amplitude increases, the noise disturbs the learning.

Secondly, we show the result of the MLP including different amplitude random noise (3) in Fig. 5. We show three graphs in this figure. This number means the number of groups of the neurons in hidden layer. In the MLP including different amplitude random noise (3), the change of MSE becomes small. The MSE becomes small when the noise amplitude is between 0.1 and 0.4. Moreover, we cannot see the high dependency of the number of groups. Each neuron have same amplitude from start of learning to end of learning, thereby this influence is almost same as the MLP with uniformed random noise (2).

Finally, we show the result of the proposed MLP (4) in Fig. 6. In this case, the dependency of the number of groups of neurons is large. The learning performance is high when we divide five groups. And also, it has high learning performance in wide range of the maximum amplitude of random

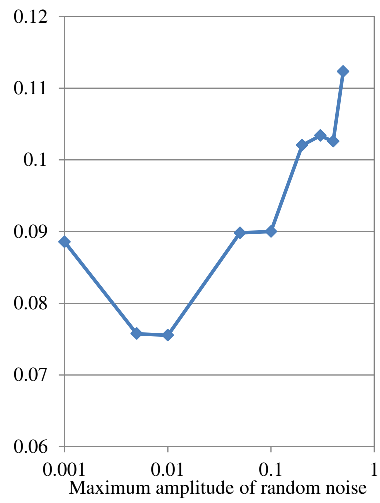

Figure 4: Parameter dependency of MLP with uniformed random noise.

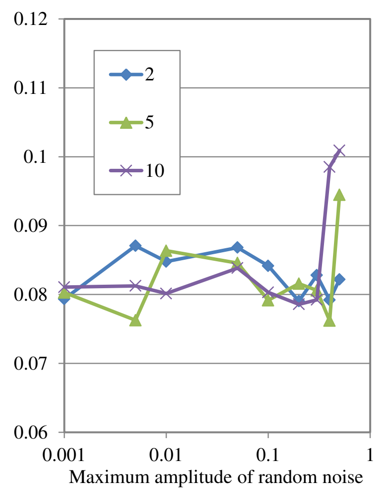

Figure 5: Parameter dependency of MLP including different amplitude random noise.

noise. The curve which we divide 2 groups, is similar to the result of the MLP including the different amplitude random noise (3). We consider that this group includes many neuron which has various value of output, thereby the influence of divide group becomes small.

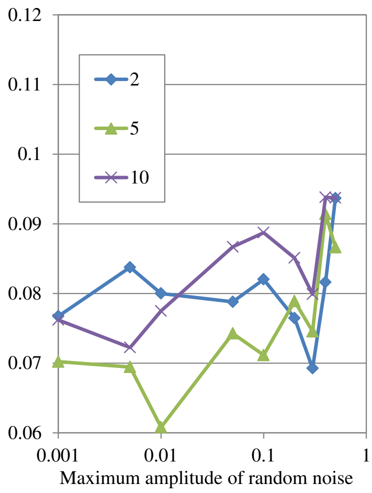

Figure 6: Parameter dependency of proposed MLP.

4. Conclusion

In this study, we have proposed the MLP including different amplitude noise. We input the uniformed random noise to the inner state of the neuron in hidden layer. Then, the neurons in hidden layer are divided to some groups according to the output value of neurons. The neurons in the same layer receives same amplitude noise. The amplitude of input noise becomes small when the neuron output becomes smaller. By the simulations, we confirmed that the proposed MLP has better learning performance than other MLPs. Moreover, we show the parameter dependency of the MLPs. From this result, the learning performance improves by dividing the groups of neurons in hidden layer according to the value of neuron output.

References

[1] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, vol.323-9, pp.533-536, 1986.

[2] M. Fernandez-Redondo, and C. Hernandez-Espinosa. "Weight initialization methods for multilayer feedforward," pp. 119-124, ESANN. 2001.

[3] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no.4598, pp. 671-680, 1983.

[4] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator," ACM Transactions on Modeling and Computer Simulation, vol.8 (1), pp.3-30, 1998.

[5] J.R. Alvarez-Sanchez, "Injecting Knowledge into the Solution of the Two-Spiral Problem," *Neural Computing & Applications*, vol.8, pp.265-272, 1999.

[6] H. Sasaki, T. Shiraishi and S. Morishita, "High Precision Learning for Neural Networks by Dynamic Modification of Their Network Structure," *Dynamics & Design Conference*, pp.411-1–411-6, 2004.