2013 International Symposium on Nonlinear Theory and its Applications
NOLTA2013, Santa Fe, USA, September 8-11, 2013

Back Propagation Learning of Neural Networks with Replicated Neurons

Naohiro Shibuya, Yuta Yokoyama, Chihiro Ikuta, Yoko Uwate and Yoshifumi Nishio

Dept. of E.E. Eng. Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, JAPAN
Email: {shibuya, yuta, ikuta, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract—The human brain is able to process the complex information. One of the reason is that the cerebellum has a particular function. This function is that the cerebellum copies information in the cerebrum. We focus on the function of the cerebellum.

In this study, we apply such function to the artificial neural network operating the Back Propagation (BP). We actualize the function of the cerebellum by some processing. We consider that the number of one neuron is doubled at arbitrary timing. The weight parameters of one neuron are taken over in the doubled neurons. The method of taking over the weight parameters is that the weight parameters are carved up and sort weight parameters into each neuron. When the weight parameters are taken over, the different errors are added to each weight parameter. The doubled neurons learn with small learning rate. We confirm that the learning performance of the proposed network is better than the other networks.

1. Introduction

The human brain is classified into the cerebrum, cerebellum and brain stem. It is able to process the complex information because different parts of the brain have various functions. One of the reason is that the cerebellum has a particular function. This function is that the cerebellum copies information in the cerebrum. For example, in the case of motion of human, the cerebellum copies rough information of motion in the cerebrum and learns more detail motion. Thereby, the human can do detailed motion.

We apply such function to the artificial neural network operating the BP. The neural network is the mathematical model and be able to actualize brain function by computation simulation. The BP is the technique of the parameter studying in the neural network. When we actualize the function of the cerebellum, we add some processing to conventional BP. First, the number of one neuron in the hidden layer is doubled at arbitrary timing. Second, the weight parameters of one neuron are taken over in the doubled neurons. When the weight parameters are taken over, the different errors are added to each weight parameter. The doubled neurons learn with small learning rate. We hope the contraction of the learning time, high efficiency and accuracy learning by applying this function. In this study, we prove that the learning performance of the proposed network is better than the conventional network by applying the cerebellum function.

2. Multi-Layer Perceptron

MLP is one of a feed-forward neural network and a most famous feed-forward neural network. This network is used for the function approximation [1]-[3], pattern recognition, pattern classification and pattern learning. MLP is composed some layers, it has input layer, hidden layer, and output layer. This network learns to the tasks by changing the weight parameters. Generally, the performance of the MLP is changed by the number of neurons. In this study, I consider which composed of three layers (one input, one hidden, and one output layer). Figure 1 shows the model of the MLP.

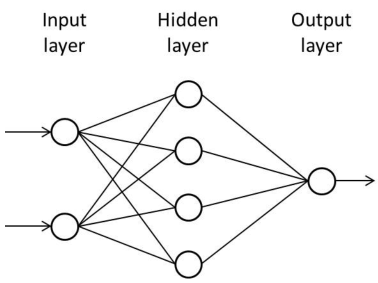

Figure 1: Multi-Layer Perceptron.

2.1. Back Propagation

The BP is the learning algorithm of the parameter study in the MLP [4]-[7]. The BP was introduced by D. J. Rumelhart in 1986. The algorithm of the BP is listed below. First, the teaching signal is provided to the neural network for learning. Second, the network calculates the error from the output and teaching signal. Finally, this error is propagating backward in the network. The network can learn to tasks by the repeating this process. The BP algorithm changes the value of weights to obtain smaller error than before.

The following are equations of the BP. The output function is described by Eq. (1). Moreover, the internal state and sigmoid function are described by Eqs. (2) and (3).

$$x_i(t+1) = f\left(u_i(t+1)\right), \tag{1}$$

$$u_i(t+1) = \sum_j w_{ij} x_j(t), \tag{2}$$

$$f(a) = \frac{1}{1+e^{-a}}, \tag{3}$$

where x is the input or output, u is the internal state and w is the weight parameter. The square error used for error evaluation is described by Eq. (4).

$$E = \frac{1}{2}\sum_{i=1}^{n}(t_i - O_i)^2, \tag{4}$$

where E is the error, n is the number of output, t is the target value and O is the output value.

3. Proposed Network

We consider a feed-forward neural network with three layers. The number of one neuron in the hidden layer is quadrupled at arbitrary timing. The biggest variation of weight parameter is chosen as one neuron. The weight parameters of one neuron are taken over in the quadrupled neurons. The method of taking over the weight parameters is that the weight parameters are carved up and sort weight parameters into each neuron. When the weight parameters are taken over, the different errors are added to each weight parameter. The quadrupled neurons learn with small learning rate. Here, the primordial neurons are called cerebrum group. Moreover, the first quadrupled neurons are classified as 1st cerebellum group and next quadrupled neurons are classified as 2nd cerebellum group.

The algorithm of the proposed network is listed below. First, the MLP learns several times by the cerebrum group in the hidden layer and updates the weight parameters. Second, the weight parameters of neurons in the cerebrum group are taken over neurons in the 1st cerebellum group. Here, the biggest variation of weight parameter is taken over the 1st cerebellum group. The method of taking over is that the weight parameters are divided and copied into the neurons. When the weight parameters are taken over, the different errors are added to each weight parameters. By taking over the weight parameters in the 1st cerebellum group, the weight parameters of neurons in the hidden layer are shown in Fig. 3 as an example. The parameter δ is uniform random numbers (-0.1 to 0.1). Third, the MLP learns by the cerebrum and 1st cerebellum groups in the hidden layer and updates the weight parameters. Fourth, the weight parameters of neurons in the cerebrum group are taken over neurons in the 2nd cerebellum group. Here, the second biggest variation of weight parameter is taken over the 2nd cerebellum group. The method of taking over is the same as last time. By taking over the weight parameters in the 2nd cerebellum group, the weight parameters of neurons in the hidden layer are shown in Fig. 4 as an example. Finally, the MLP learns by the cerebrum, 1st cerebellum and 2nd cerebellum groups in the hidden layer and updates the weight parameters.

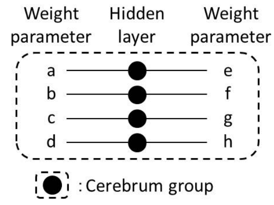

Figure 2: The primordial neurons.

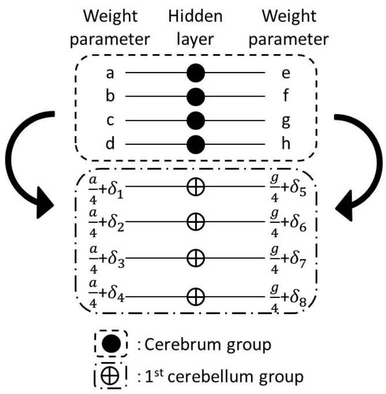

Figure 3: First quadrupled neurons.

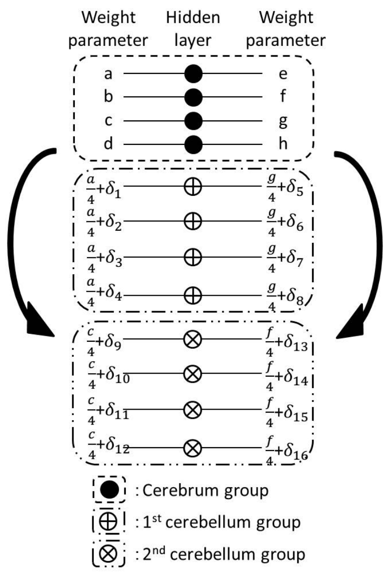

Figure 4: Second quadrupled neurons.

4. Simulations

We apply the proposed network to the function approximation. We input the 4 dimensional Chebyshev polynomial to the network. The functions of the 4 dimensional Chebyshev polynomial is described by Eq. (5) and shown in Fig. 5.

$$T_4(x) = 8x^4 - 8x^2 + 1, \tag{5}$$

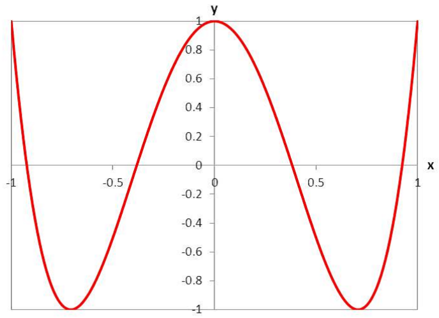

Figure 5: 4-dimensional Chebyshev polynomial.

In this simulation, the number of neurons in the input layer and the output layer is set to 1. The learning rate of cerebrum group is set to 0.01. The learning rate of 1st cerebellum and 2nd cerebellum groups are set to 0.001. The learning time is 50000. The timing of taking over the weight parameters are set to 10000 and 20000 learning time. The starting value of the weight parameter is randomly set for composing the network. The error of the network is average of 10 random initial values.

4.1. Comparison between Proposed and Two Conventional Networks

In this section, we compare the proposed network and two conventional networks. We investigate effects on the error of the network if the weight parameter of the neurons in the cerebrum group are taken over the cerebellum group. The one conventional network is that the number of neurons is set to 12, and the learning rate is set to 0.01. The other conventional network is that the noise is added to weight parameters at 10000 and 20000 learning times.

The algorithm of the proposed network is listed below. From 0 until 9999 learning times, the MLP learns by the cerebrum group in the hidden layer and updates the weight parameter. At 10000 learning times, the weight parameter of the neurons in the cerebrum group are taken over neurons in the 1st cerebellum group. Here, the biggest variation of weight parameter is taken over the 1st cerebellum group. From 10001 until 19999 learning times, the MLP learns by the cerebrum and 1st cerebrum group in the hidden layer and updates the weight parameter. At 20000 learning times, the weight parameters of the neurons in the cerebrum group are taken over neurons in the 2nd cerebellum group. Here, the second biggest variation of weight parameter is taken over the 2nd cerebellum group. From 20001 until 30000 learning times, the MLP learns by the cerebrum, 1st cerebrum and 2nd cerebellum groups in the hidden layer and updates the weight parameter. The simulation result is shown in Fig. 6.

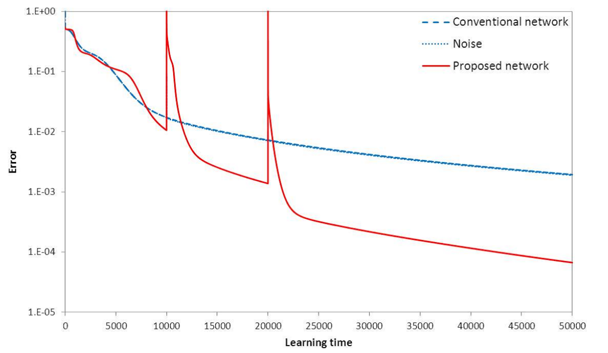

Figure 6: Comparison between the proposed and two conventional networks.

Table 1: Simulation results

	Conventional	Noise	Proposed
Average of error	$1.94E-03$	$1.89E-04$	$6.67E-05$
Minimum value	$7.19E-04$	$7.10E-04$	$6.06E-05$

From 0 until 10000 learning times, the error of proposed network is differ little from the conventional network. However, after 10001 learning times, the error of proposed network has decreased. Finally, the error of proposed network is the best of all. From these results, we have confirmed that taking over the weight parameters is effective.

4.2. Comparison between Proposed and Two Other Networks

In this section, we compare the proposed network and two the other networks. The two other networks are that the weight parameter of the neurons in the cerebrum group are copied into neurons in the cerebellum group at 10000 and 20000 learning time. The algorithm of the proposed network is the same as last time. The simulation result is shown in Fig. 7.

From 0 until 10000 learning times, the error of "copy(20000)" is better than the error of proposed network. However, from 10001 until 20000 learning times, the error of "copy(20000)" is same as the error of proposed network. Finally, the error of proposed network is the best of all.

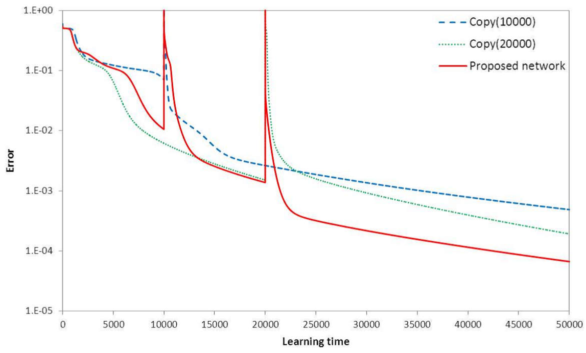

Figure 7: Comparison between proposed and two other networks.

Table 2: Simulation results

	Copy(10000)	Copy(20000)	Proposed
Average of error	$4.88E-04$	$1.93E-04$	$6.67E-05$
Minimum value	$2.78E-04$	$1.33E-04$	$6.06E-05$

From these results, we have confirmation that taking over the weight parameters is effective.

4.3. Comparison between Two Types Proposed Networks

In this section, we compare the two types proposed networks. The algorithm of the two types proposed networks is same as proposed network of last section. However, when the value of neuron is taken over, the chosen neuron of the proposed network (biggest variation) is different from the proposed network (smallest variation). In the case of the proposed network (biggest variation), the chosen neuron is the biggest variation and second biggest variation of weight parameter. On the other hand, I chose the neuron that the smallest variation and second smallest variation of weight parameter. The simulation result is shown in Fig. 8.

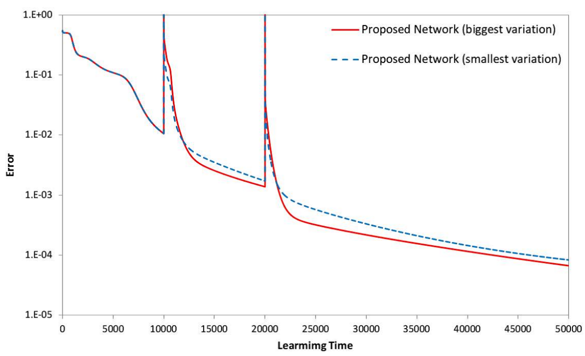

Figure 8: Comparison between two types proposed networks.

From this result, we consider that the proposed network (biggest variation) is better than the proposed network (smallest variation). Namely, we have confirmed that we are able to search the optimized solution by using the proposed network (biggest variation).

5. Conclusion

In this study, we applied the function of the cerebellum to the MLP. When we actualize the function of the cerebellum, we add some processing to the conventional MLP. The number of one neuron in the hidden layer is quadrupled at arbitrary timing. The weight parameters of one neuron are taken over in the quadrupled neurons. When the weight parameters are taken over, the different errors are added to each weight parameters. We applied the proposed network to the function approximation. From these results, the learning performance of the proposed network is better than the conventional network. We considered that there are optimal learning rate combination, timing of taking over the weight parameters and other parameters.

Acknowledgment

This work was partly supported by MEXT/JSPS Grant-in-Aid for JSPS Fellows (24·10018).

References

[1] T. Poggio and F. Girosi, “Networks for approximation and learning”, Proc. IEEE, vol.78, no.9, pp.1481-1497, 1990.

[2] K. Hornik , M. Stinchcombe and H. White, “Multilayer feedforward networks are universal approximators”, Neural Networks, vol.2, pp.359-366, 1989.

[3] Y. Uwate and Y. Nishio, “Durability of Affordable Neural Networks against Damages”, International Joint Conference on Neural Networks (IJCNN’06), pp.8365-8370, July, 2006.

[4] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, “Learning representations by back-propagating errors”, Nature, vol.323-9, pp.533-536, 1986.

[5] Takeshi Agui, Hiroshi Nagahashi, Hiroki Takahashi, “Neural Program”, Shokoudou corp, pp.11-18, 20-42, 1995.

[6] Jain. Rarnesh C, “A robust backpropagation learning algorithm for function approximation”, Journals, vol.5, pp.467-479, May, 1994.

[7] Horng-Lin Shieh, Ying-Kuei Yang, Po-Lun Chang, Jin-Tsong Jeng, “Robust neural-fuzzy method for function approximation”, Expert Systems with Applications, vol.36, pp.6903-6913, April, 2009.