2013 International Workshop on Nonlinear Circuits,
Communications and Signal Processing
NCSP'13, Isrand of Hawaii, Hawaii, USA. March 4-7, 2013

Copying Weight Parameters in Back Propagation

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

Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
E-mail: {shibuya, yuta, s-tomoya, 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 dividing the hidden layer into two groups. The weight parameter of neurons in one group are copied into neurons in the other group. We confirm that the learning performance of the proposed network is better than the conventional network.

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 Back Propagation (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 study in the neural network. When we actualize the function of the cerebellum, we add two processing to conventional BP. First, the hidden layer in the Multi-Layer Perceptron (MLP) is divided into two groups. Second, the connection weight parameter of neurons in the one group are copied into neurons in the other group. We hope the contraction of the learning time, high efficiency and accuracy learning by applying this function. Because, the MLP learns by the cerebrum group. After that, the MLP learns particularly by the cerebellum group. 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. Back Propagation

The MLP is one of a feed-forward neural network. This network is used for the function approximation [1], pattern recognition, pattern classification and pattern learning. The MLP is composed some layers which are input layer, hidden layer and output layer.

The BP is the learning algorithm of the parameter study in the MLP [2]-[4]. 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. BP algorithm changes the value of weights to obtain smaller error than before.

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

$$x_i(t+1) = f\left(u_i(t+1)\right), \qquad (1)$$

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

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

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).

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

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

3. Structure and Algorithm

We consider a feed-forward neural network with three layers. The composition of the proposed network is shown in Fig. 1. The hidden layer is divided into two groups. The one group is the cerebrum group, and the other group is the cerebellum group. The cerebrum and the cerebellum group has 4 neurons. 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 parameter. Before copying, the weight parameters in the cerebellum group are shown in Fig. 2. Second, the connection weight parameter of neurons in the cerebrum group are copied into the cerebellum group. By copying, the weight parameters in the cerebellum group are shown in Fig. 3. Finally, the MLP learns by the cerebellum group in the hidden layer and updates the weight parameter.

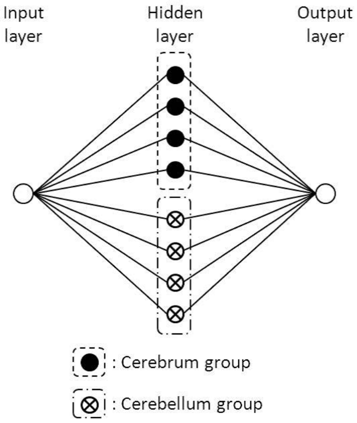

Figure 1: Composition of the proposed network.

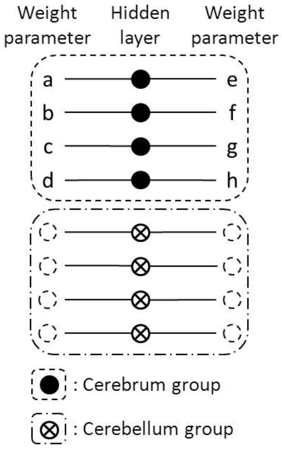

Figure 2: Before copying the weight parameter.

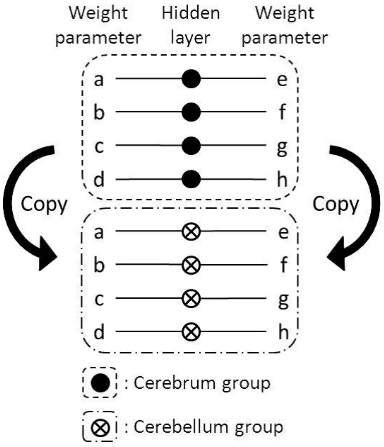

Figure 3: After copying the weight parameter.

4. Simulations

We apply the function approximation to the proposed network. We input the 2 dimensional and 3 dimensional Chebyshev polynomial to the network. We consider the 2 dimensional and 3 dimensional Chebyshev polynomial. The functions of the 2 dimensional and 3 dimensional Chebyshev polynomial are described by Eqs. (5) and (6).

$$T_2(x) = 2x^2 - 1, \tag{5}$$

$$T_3(x) = 4x^3 - 3x. \tag{6}$$

We compare the learning curve of the proposed network and the conventional network. In this study, we consider that the number of neurons in the hidden layer is 12. The number of neurons in the cerebrum group is set to 4. The number of neurons in the cerebellum group is set to 8. The number of neurons in the input layer and the output layer is 1. The learning time is 20000. The learning late is 0.005. The starting value of the weight parameter is randomly set for composing the network.

4.1 2 dimensional Chebyshev Polynomial

4.1.1 Comparison between Proposed and Conventional Networks

In this section, we compare the proposed network and the conventional network. We investigate effects on the error of the network if the weight parameter of the neurons in the cerebrum group are copied into the cerebellum group. Here, the error of the network is average of 10 random initial values.

We input the 2 dimensional Chebyshev polynomial. The algorithm of the proposed network is listed below. From 0 until 4000 learning times, the MLP learns by the cerebrum group in the hidden layer and updates the weight parameter. At 400 learning times, the weight parameter of the neurons in the cerebrum group are copied into neurons in the cerebellum group. From 4001 learning times, the MLP learns by the cerebellum group in the hidden layer and updates the weight parameter. The simulation result is shown in Fig. 4.

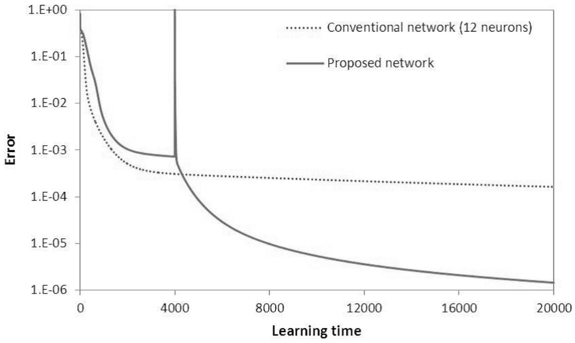

Figure 4: Comparison between proposed and conventional network.

From 0 until 4000 learning times, the conventional network is better than the proposed network. At 4000 learning times, the error of the proposed network increases because the weight parameter of neurons in the cerebrum group are copied into neurons in the cerebellum group. From 4001 learning times, suddenly the error decreases. Finally, the learning performance of the proposed network is better than the conventional network. However, the convergence speed of the proposed network is slower than the conventional network.

4.1.2 Comparison with Three Proposed Networks

In this section, we compare three proposed networks. The timing of copy of the weight parameter is set to 500, 4000 and 8000 learning times. We investigate effects on the error of network if the timing of copy of the weight parameter is changed. The error of network is average of 10 random initial values. The simulation result is shown in Fig. 5.

From Fig. 5, we can see that the error of the proposed network (500) is the best. Moreover, the proposed network (8000) is the worst. We consider that the timing of copy of the weight parameter relates to convergence of the error curve. Therefore, we assume that there is the optimal timing of copy of the weight parameter at before convergence of the error curve.

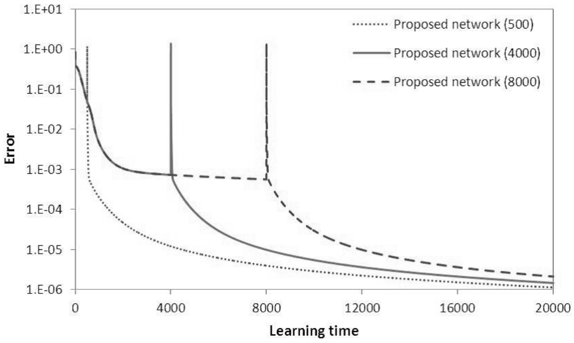

Figure 5: Comparison with three proposed networks.

4.1.3 Average of Results

We give 100 random initial values. The results are average of error and minimum value in the final learning time. The results are shown in Table 1.

Table 1: Average of results

	Conventional network		
Number of neurons	4	8	12
Average of error	$8.02E-03$	$1.78E-04$	$1.64E-04$
Minimum value	$8.27E-03$	$5.42E-05$	$2.29E-05$

	Proposed network		
Timing of copying	500	4000	8000
Average of error	$1.42E-06$	$1.54E-06$	$2.55E-06$
Minimum value	$1.82E-09$	$2.12E-08$	$5.08E-09$

From Table 1, we can see that the error of the conventional network is the worst. The learning performance of the copy in 500 learning times is the best of all.

4.2 3 dimensional Chebyshev Polynomial

4.2.1 Comparison between Proposed and Conventional Networks

In this section, we compare the proposed network and the conventional network. Here, the error of network is the average of 10 random initial values. We input the 3 dimensional Chebyshev polynomial. The simulation result is shown in Fig. 6.

From this result, the learning performance of the proposed network is better than the conventional network in the final learning time. If we give the complex input, the learning performance is better by copying the weight parameter.

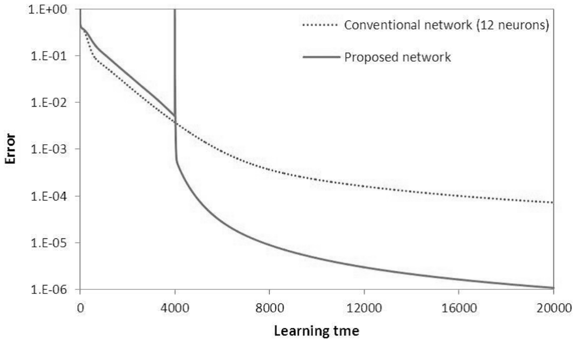

Figure 6: Comparison between proposed and conventional network.

4.2.2 Comparison with Three Proposed Networks

In this section, we compare the three types proposed network. The timing of copy of the weight parameter is set to 500, 4000 and 8000 learning times. The error of network is average of 10 random initial values. The simulation result is shown in Fig. 7.

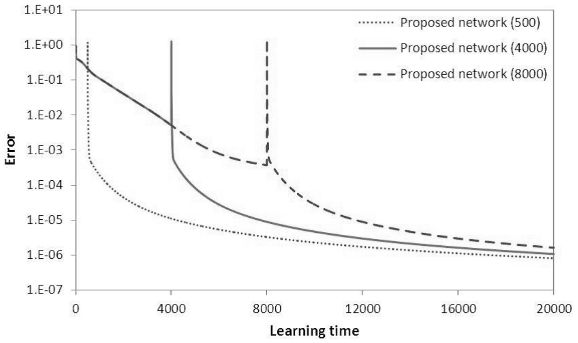

Figure 7: Comparison with three proposed networks.

From this result, if we give the complex input, the error of the proposed network (500) is the best. Moreover, the proposed network (8000) is the worst. Because, the timing of copy of the weight parameter of proposed network (8000) is the slowest of all. In the case of 3 dimensional Chebyshev polynomial, the timing of copy of the weight parameter relates to convergence of the error curve .

4.2.3 Average of Results

We give 100 random initial values. The results are average of error and minimum value in the final learning time. The results are shown in Table 2.

From Table 2, we can see that the error of the conventional network is the worst. The learning performance of the copying at 500 learning times is the best of all.

Table 2: Average of result

	Conventional network		
Number of neurons	4	8	12
Average of error	$1.58E-04$	$1.17E-04$	$2.67E-04$
Minimum value	$6.60E-05$	$4.29E-05$	$5.91E-05$

	Proposed network		
Timing of copying	500	4000	8000
Average of error	$1.34E-06$	$1.63E-06$	$2.15E-06$
Minimum value	$2.52E-09$	$5.30E-08$	$1.58E-08$

5. Conclusions

In this study, we applied the function of the cerebellum to the BP. When we actualize the function of the cerebellum, we add two processing to the conventional BP. First, the hidden layer in the MLP is divided into two groups. The one group is the cerebrum group, and the other group is the cerebellum group. Second, the connection weight parameter of neurons in the cerebrum group are copied into neurons in the cerebellum group.

We applied the function approximation to the proposed network. From these results, the learning performance of the proposed network is better than the conventional network. However the convergence speed of the proposed network is slower than the conventional network. Moreover, we considered that the timing of copy of the weight parameter relate to the convergence of error curve.

Acknowledgment

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

References

[1] 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.

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

[3] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group: ”Parallel distributed processing”, MIT Press, 1986.

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