

Effect of Chaos Noise on the Learning Ability of Back Propagation Algorithm in Feed Forward Neural Network

Azian Azamimi^{#1}, Yoko Uwate^{#2}, Yoshifumi Nishio^{#3}

[#]*School of Mechatronic Engineering, Universiti Malaysia Perlis
Ulu Pauh Campus, 02600, Arau, Perlis, Malaysia*
¹azamimi@unimap.edu.my

[#]*Institute of Neuroinformatics
University / ETH Zurich, Switzerland*
²yu001@ini.phys.ethz.ch

[#]*Department of Electrical and Electronic Engineering
Tokushima University, 2-1 Minami Josanjima, Tokushima 770-8506, Japan*
³nishio@tokushima-u.ac.jp

Abstract— In the area of artificial neural networks, the Back Propagation (BP) learning algorithm has proved to be efficient in many engineering applications especially in pattern recognition, signal processing and system control. Although the BP learning has been a significant research area of neural network, it has also been known as an algorithm with a poor convergence rate. Many attempts have been made on the learning algorithm to improve the performance on convergence speed and learning efficiency. In this study, we propose a new modified BP learning algorithm by adding chaotic noise into weight update process during error propagation. The chaotic noise is generated using various chaotic maps such as Logistic map, Skew Tent map and Bernoulli Shift map. By computer simulations, we confirm that our proposed algorithm can give a better convergence rate and can find a good solution in early time compared to the conventional BP learning algorithm. Weight update position, noise amplitude and control parameter of chaos can give a big effect on the learning ability of feed forward neural network.

Keywords— Chaos Noise, Back Propagation, Neural Networks

I. INTRODUCTION

Recently, studies on the human brain have been carried out actively on various levels. On the system level, many researchers investigate how the brain operates to realize higher functions such as learning, memory, emotion and so on. However, much is still unknown about how the brain trains itself to process information, so theories abound. The learning process in human brain is modelled in the form of artificial neural networks. Just like human, neural networks learn from experience and example. In the area of artificial neural networks, the Back Propagation (BP) learning algorithm has proved to be efficient in many engineering applications. BP was introduced by Rumelhart in 1986 [1]. BP is used for learning algorithm of multilayer Perceptron and the error propagates backward in the network. The BP learning operates with a feed forward neural network which consists of an input layer, a single or more of hidden layers and an output layer. Although the BP learning has been a significant

research area of neural network, it also has been known as an algorithm with a poor convergence rate. In order to avoid this problem, some methods to improve the performance on convergence speed are required. Many researchers concentrated on changing the momentum and learning rate and also the number of hidden neurons in the hidden layer but this will lead to slight improvement only [2], [3]. Not many studies have been made on modifying the algorithm itself.

On the other hand, chaos has gained much attention and some applications in neural network during recent years. There have been many reports on the good performance Hopfield neural network when chaos is inputted to the neurons as noise [4], [5]. By computer simulations, it has been confirmed that the chaos noise is effective for solving quadratic assignment problem and gains better performance to escape out local minima than random noise. Hence, we believe that various features of chaos can give a good effect in neural network. Considering this, we investigate the effect of chaos on the learning ability of BP algorithm in order to improve its convergence rate and learning efficacy.

In this study, we propose a new modified BP learning algorithm by adding chaos noise into weight update process during error propagation. By computer simulations, we confirm that our proposed algorithm can give a better convergence rate and find a good solution in early time compared to the conventional BP algorithm. Weight update position, noise amplitude and chaos parameter can give a large impact on the BP learning performance.

II. BP LEARNING ALGORITHM

In this section, we explain the conventional BP algorithm and our new modified BP algorithm respectively.

A. Conventional BP Algorithm

In the standard BP learning algorithm, the errors of output neurons are back propagated through the network during training. The error signal of output neuron can be defined by

taking the difference between the target output and the actual output. BP algorithm contains one or more layers each of which are linked to the next layer. The first layer is called the input layer which meets the initial input and so does the last layer, output layer which usually holds the input identifier. The layers between input and output layers are called the hidden layers which only propagate the previous layer's output to the next layer and back propagates the following layer's error to the previous layer. These are the main operations of training a BP algorithm which can be summarized as follows:

- 1) Apply input to the network
- 2) Calculate the output
- 3) Compare the resulting output with target output for the given input, and then calculate the error.
- 4) Modify the weights for all neurons based on the error
- 5) Repeat the process until the error reaches an acceptable value, which means that the network was trained successfully.

However, in this study, we use the batch BP learning algorithm. The batch BP learning is expressed by a formula similar to the conventional BP but the difference lies in timing of the weight update. The weight update of the standard BP is performed after each single input data while for the batch BP the weight update is performed after all input data has been processed. The total error of the network is defined as Eq. (1),

$$E = \sum_{p=1}^P E_p = \sum_{p=1}^P \left\{ \frac{1}{2} \sum_{i=1}^N (t_{pi} - o_{pi})^2 \right\}, \quad (1)$$

where P is the number of input data, N is the number of neurons in the output layer, t_{pi} denotes the value of desired target data for the p th input data and o_{pi} denotes the value of the output data for the p th input data. The goal of the learning is to set weights between all layers of the network so that the total E can be minimized. In order to minimize the total error E , the weights are updated according to the following equations:

$$w_{ij}^{k-1,k}(m+1) = w_{ij}^{k-1,k}(m) + \sum_{p=1}^P \Delta_p w_{ij}^{k-1,k}(m), \quad (2)$$

$$\sum_{p=1}^P \Delta_p w_{ij}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{ij}^{k-1,k}} \quad (3)$$

where w is the weight between i th neuron of the layer $k-1$ and the j th neuron of the layer k , m is the learning time and η is the learning rate. In this study, we add the inertia term to Eq. (4) where ζ denotes the inertia rate.

$$\Delta_p w_{ij}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{ij}^{k-1,k}} + \zeta \Delta_p w_{ij}^{k-1,k}(m-1) \quad (4)$$

B. Chaotic BP Algorithm

Slow convergence rate is the most common disadvantage of BP learning algorithm. The convergence rate of the learning process can be improved by changing how the error propagates back through the network. Considering this, we add chaos noise into weight update process during error propagation in order to see the effect of chaos noise to the learning ability. We name this proposed algorithm as "Chaotic BP Algorithm". We add chaos noise into three different position of weight update and propose three kinds of network;

- 1) Proposed-network 1 (from input layer to hidden layer)
- 2) Proposed-network 2 (from hidden to output layer)
- 3) Proposed-network 3 (both of them)

The new weight update process for chaotic BP can be shown as below equations where β denotes the noise amplitude.

$$\Delta_p w_{ij}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{ij}^{k-1,k}} + \zeta \Delta_p w_{ij}^{k-1,k}(m-1) + \text{noise}_{i,j}(m) \quad (5)$$

$$\text{noise}_{i,j}(m) = \beta_{i,j}(m)(x_{i,j}(m) - 0.5) \quad (6)$$

Chaos can be generated by various chaotic maps, such as Logistic map, Skew Tent map and Bernoulli Shift map. These maps can be shown in Fig.1, Fig. 2 and Fig.3 respectively.

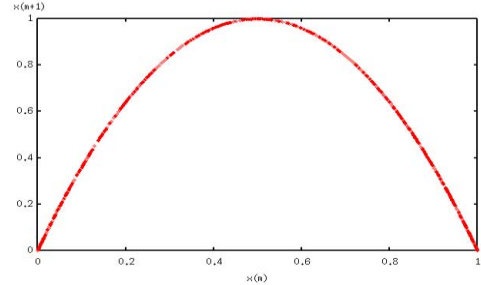


Fig. 1 Logistic map.

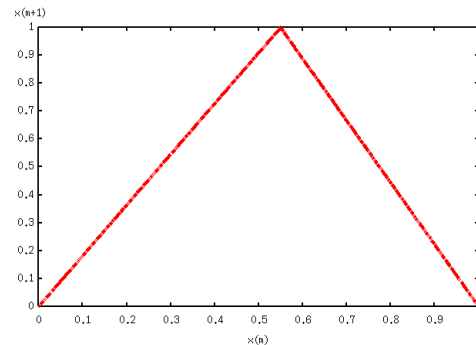


Fig. 2 Skew Tent map.

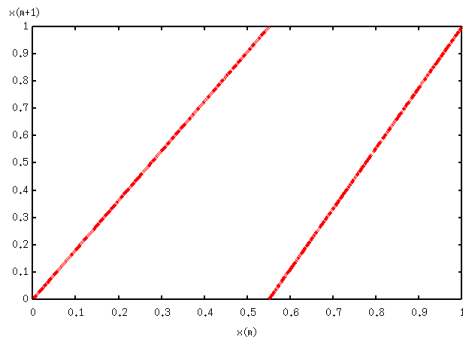


Fig. 3 Bernoulli Shift map.

III. SIMULATION RESULTS

In this section, we show the effectiveness of our proposed algorithm by simulating it to the learning examples. Two cases of learning example are used for this experiment.

A. Learning Example 1: x^2

Here, we consider the feed forward neural network produce output x^2 for input data x as one learning example. The sampling range of the input data is $[-1.0, 1.0]$. We carried out the chaotic BP learning algorithm by using the following parameters. The learning rate and the inertia rate are fixed as $\eta = 0.02$ and $\zeta = 0.002$ respectively. The initial values of the weights are given between -1.0 and 1.0 at random. The learning iteration is set to 10000 and 6 neurons are prepared in the hidden layer. The network structure and learning example are shown in Fig.4 and Fig. 5. For this learning example, we use logistic map to generate chaotic sequence.

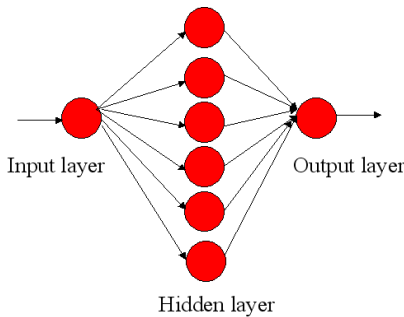


Fig. 4 Network structure.

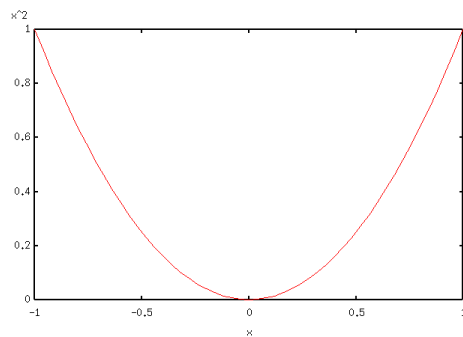


Fig. 5 Learning example 1: $y = x^2$

First, we investigate the learning efficacy by adding chaotic noise into different position of weight update. We compare the learning performance with the conventional method where there is no noise adding at all. Figure 6 shows the error curve of the learning performance for all proposed network. The horizontal axis is iteration time and the vertical axis shows the error value.

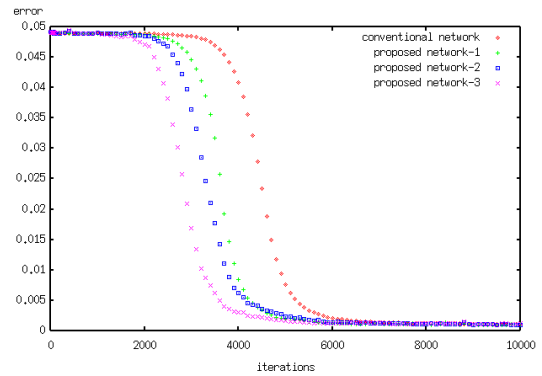


Fig. 6 Learning performance of proposed and conventional network.

From this figure, we can confirm that our proposed networks gain better performance than the conventional BP algorithm when the chaos parameter is set to 4.0 and noise amplitude is fixed as 0.01. We also confirm that the chaos noise applied to both sides of weight update (proposed network-3) requires less iteration time compared to others. Figure 7 shows the output results of all proposed and conventional networks.

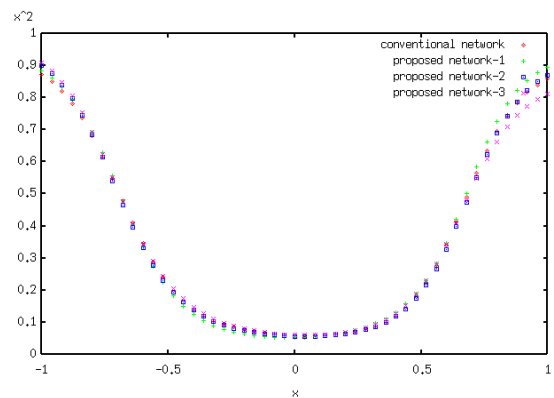


Fig. 7 Output results of learning example 1: $y = x^2$

B. Learning Example 2: Pattern recognition

Here, we consider pattern recognition task as one example of difficult problem. 10 patterns of numeric characters are fed into neural network for recognition as shown in Fig. 8. In this case, the number of input neurons is 35 and the number of output neurons is fixed to 10. We choose 16 neurons in the hidden layer. For recognition, a set of 10 patterns shifted 1 bit from each original pattern was prepared, leading a set of 100 patterns to be recognized. We use the same parameters as the

previous one. For this learning example, we use Bernoulli shift map to generate the chaotic sequence.

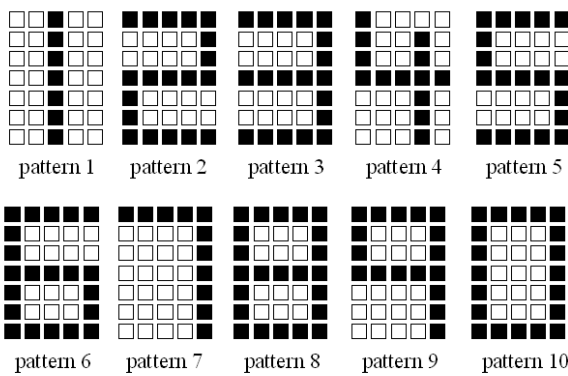


Fig. 8 Pattern recognition task.

We add chaotic noise into three different position of weight update. The learning performance of all proposed and conventional network is shown in Fig. 9. We conduct 100 trials by using different initial condition and we obtain reliable average of error rate. The final value of error rate at iteration time = 20000 can be summarized in Table 1.

TABLE I
FINAL VALUE OF ERROR RATE

Conventional	Network-1	Network-2	Network-3
0.5362454	0.2259438	0.3859357	0.1982767

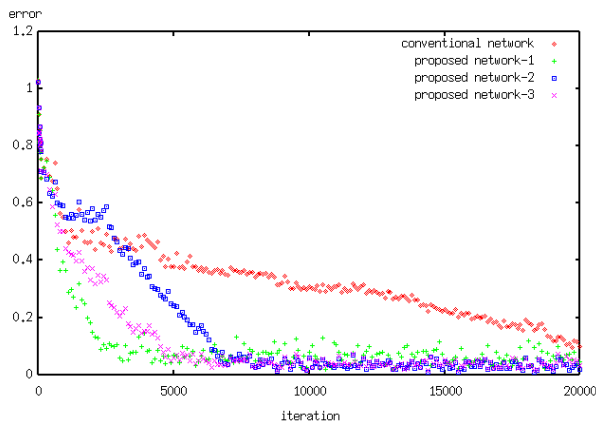


Fig. 9 Learning curve of all proposed and conventional network for pattern recognition.

During network training, the number of hidden neurons is critical. It is difficult to set an accurate number of hidden neurons. Therefore, we conduct further investigation on how to optimize the hidden layer neurons. Figure 10 shows the learning curve of proposed network-3 when the number of hidden neurons is changed. Upon increasing the number of hidden neurons, the learning ability also increases, showing that the hidden neurons play an important role during learning process. But if the number of hidden neurons is too large, it

will lead the learning performance to oscillate. Here, the optimum number of hidden neurons is 16.

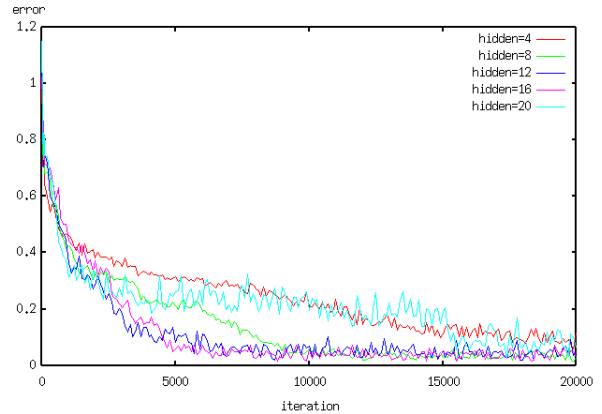


Fig. 10 Learning curve of proposed network-3 when the number of hidden neurons are changed.

From these results, we notice that our proposed network can give a better recognition rate compared with the conventional BP algorithm.

IV. CONCLUSIONS

In this study, we have investigated the effect of chaos noise on learning ability of Back Propagation learning algorithm in feed forward neural network. We have proposed a new modified BP learning algorithm by adding chaos noise into weight update process during error propagation. By computer simulations, we confirmed that the addition of chaos noise can give a better learning ability and faster convergence speed compared to the conventional BP learning algorithm. In order to confirm the generality of this work, we have presented many learning examples and used different function of chaos generator from easy problem to more complicated problem. All simulation results show that our proposed algorithm can give improvement of system performance. We discovered that the learning process will take a longer time if more difficult learning examples are used compared with the easy problem. We also discovered some factors that influence the learning performance such as noise amplitude, chaos parameter and the role of hidden neurons. Hence, we can conclude that various features of chaos can give a good effect in neural network.

REFERENCES

- [1] D.E. Rumelhart, G.E. Hinton and R.J. William, " Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol. 1, MIT Press, MA, pp. 318-362, 1986.
- [2] A. Sperduti and A. Starita, "Speed up learning and network optimization with extended back-propagation," *Neural Networks*, vol. 6, pp.365-383, 1993.
- [3] R.A. Jacobs, "Increased rates of convergence through learning rate adaption," *Neural Networks*, vol.1, pp.295-308, 1988.
- [4] Y.Tada, Y.Uwate and Y. Nishio, " Performance of Chaotic Switching Noise Injected to Hopfield NN for Quadratic Assignment Problem," *Proceedings of ISCAS'06*, pp. 5519-5522, 2006.
- [5] Y.Uwate and Y.Nishio, "Chaotically Oscillating Sigmoid Function in Feedforward Neural Network," *Proceedings of NOLTA'06*, pp.215-216.