IWVCC'08
@ Xian
Nov. 1-3, 2008

An Analysis of Chaotic Noise Injected to Backpropagation Algorithm in Feedforward Neural Network

Azian Azamimi†, Yoko Uwate‡ and Yoshifumi Nishio†

†Department of Electrical and Electronic Engineering
Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
‡Institute of Neuroinfomatics,
University / ETH Zurich, Switzerland
Email: {azian, nishio}@ee.tokushima-u.ac.jp, yu001@ini.phys.ethz.ch

Abstract

There have been much interest in applying noise to neural networks in order to observe their effect on network performance. In our previous research, we have proposed a new modified backpropagation learning algorithm, in which chaotic noise is added into weight update process. By computer simulations, we confirmed that the presence of chaotic noise during weight update process in feedforward neural network can give a better convergence rate and can find a good solution in early time. Hence, in this study, we extend these results by introducing other difficult learning example and analyzing the effect of noise parameters such as noise amplitude and control parameter of chaos to the learning performance.

1. Introduction

Backpropagation (BP) learning is one of engineering applications of artificial neural networks. The BP learning operates with a feedforward neural network which is composed of an input layer, a single or more of hidden layers and an output layer. The effectiveness of BP learning has been recognized in many engineering applications especially in pattern recognition, system control and signal processing. Although the BP learning has been a significant research area of artificial neural network, it also has been known as an algorithm with a poor convergence rate. Many attempts have been made to the algorithm to improve the performance on convergence speed and learning efficiency. For example, there have been a lot of reports on changing the learning rate and also the number of neurons in the hidden layer but this will lead to slight improvement only [1]-[3]. Not many studies have been made on modifying the algorithm structure in order to improve the learning performance.

On the other hand, chaos has gained much attention and some applications in neural network over this recent years. There have been many reports on the good performance of Hopfield neural network when chaos is inputted to the neurons as noise [4],[5]. By computer simulations, it has been confirmed that chaotic noise is effective for solving quadratic assigment problem and gains better performance to escape out local minima than random noise. The authors [6] also have proposed the feedforward neural network with chaotically oscillating gradient of the sigmoid function and proved that the proposed network can find good solutions in early time. Hence, we consider that various features of chaos can give a good effect in neural network. Considering this, we conduct a further investigation of chaotic noise effect to the BP learning algorithm.

In this study, we analyze various methods of injecting chaotic noise into BP learning algorithm. In order to confirm the generality of this work, we simulate the results using more difficult functions as learning example. We also investigate various chaotic noise parameters such as noise amplitude and control parameter of chaos which might give different results to the learning performance.

2. BP learning algorithm

2.1. BP batch learning algorithm

In the standard backpropagation learning algorithm, the errors of output neurons are backpropagated through the network during training. This standard learning algorithm was introduced in [7]. The error signal of output neuron can be defined by taking the difference between the target output and the actual output. However, in this study, we use the batch BP learning algorithm. The batch BP learning algorithm is expressed by a formula similar to the standard BP learning algorithm 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 E of the network is defined as follows;

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

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

$$w_{i,j}^{k-1,k}(m+1) = w_{i,j}^{k-1,k}(m) + \sum_{p=1}^{P} \triangle_p w_{i,j}^{k-1,k}(m), \qquad (2)$$

$$\triangle_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}}, \qquad (3)$$

where $w_{i,j}^{k-1,k}$ is the weight between ith neuron of the layer $k-1$ and the jth 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.(3) where ζ denotes the inertia rate.

$$\triangle_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}} + \zeta \triangle_p w_{i,j}^{k-1,k}(m-1), \qquad (4)$$

2.2. Chaotic BP learning algorithm

In our previous research, we have proposed a new modified BP learning algorithm, namely chaotic BP learning algorithm. This new algorithm can be expressed by a similar formula with the standard BP algorithm but the different lies in weight update process. Chaotic noise is added into weight update process during error propagation. The weight update process for chaotic BP learning algorithm can be shown as follows, which β denotes the noise amplitude.

$$\begin{aligned}\triangle_p w_{i,j}^{k-1,k}(m) = & -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}} \\ & + \zeta \triangle_p w_{i,j}^{k-1,k}(m-1) + noise_{i,j}(m),\end{aligned} \qquad (5)$$

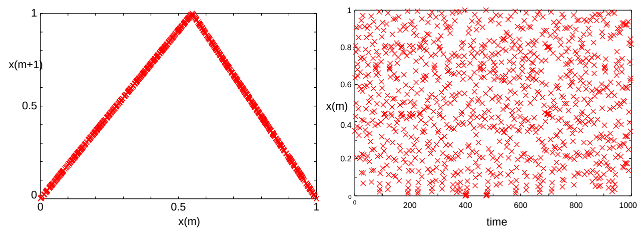
$$noise_{i,j}(m) = \beta_{i,j}(m)(x_{i,j}(m) - 0.5), \qquad (6)$$

Chaos, $x_{i,j}(m)$ is generated by skew tent map and α is the control parameter of chaos. The skew tent map and an example of time series obtained by Eq.(7) are shown in Fig. 1.

$$x_{i,j}(m+1) = \begin{cases} \frac{x_{i,j}(m)}{\alpha} & (0 \le x_{i,j}(m) \le \alpha) \\ \frac{x_{i,j}(m)-1}{\alpha-1} & (\alpha < x_{i,j}(m) \le 1) \end{cases} \qquad (7)$$

3. Simulation results

In order to confirm the generality of this chaotic BP learning algorithm, we simulate it to various sinwave functions with different amplitude as learning example. Here, we consider the feedforward neural network produce outputs $A\sin(\omega t - \varphi) + D$ 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 $\eta = 0.1$ and $\zeta = 0.001$ respectively. The initial values of the weights are given between -1.0 and 1.0 at random. The learning iterations is set to 20000 and 8 neurons are prepared in the hidden layer (Fig. 2). The learning example of sinwave functions with different amplitude are shown in Fig. 3.



(a) Skew tent map. (b) Time series.

Figure 1: Skew tent map.

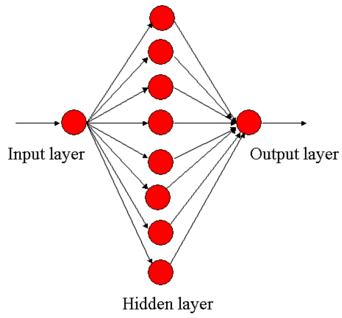


Figure 2: Network structure.

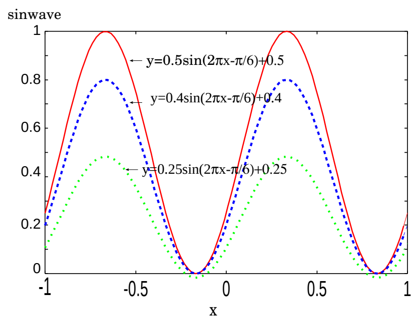


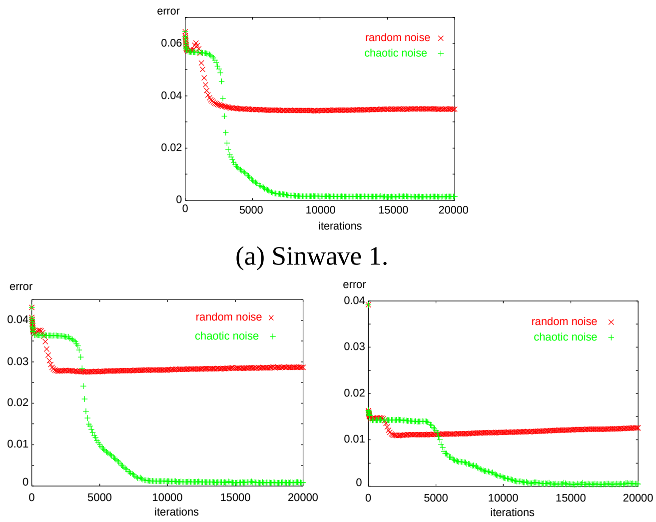
Figure 3: Learning example of sinwave with different amplitude.

It should be noted that although there maybe more suitable values of these parameters, it is diificult to set them theoretically. Hence, we use the same parameters during all process to make it easier to analyze the simulation results.

3.1. Random noise vs. chaotic noise

In order to see the effectiveness of chaotic noise to the learning performance, we make a comparison between random noise and chaotic noise. We observe their performance by simulating it to various sinwave functions with different amplitude. The noise amplitude, β is fixed as 0.01. The learning performance for both random noise and chaotic noise are shown in Fig. 4.

The results show that the chaotic noise has much better performance than random noise. The random noise cannot escape local minima while the chaotic noise can give a faster convergence rate.



(a) Sinwave 1.

(b) Sinwave 2.

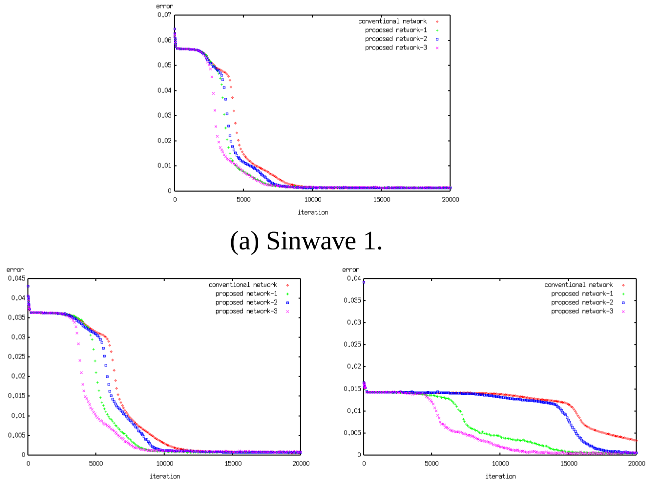
(c) Sinwave 3.

Figure 4: Comparison of learning result between random noise and chaotic noise

3.2. Noise added to different weight update positions

We also investigate the learning efficiency by adding chaotic noise into different position of weight update. As we may know, in backpropagation learning algorithm, the main purpose of weight adjustment is to reduce the error value betweet output and desired target. Thus, we apply the chaotic noise to different position of weight update in order to see the effectiveness of noise adding. We add chaotic noise to three different positions of weight update ; a) Proposed network-1 (from input layer to hidden layer only), b) Proposed network-2 (from hidden layer to output layer only) and c) Proposed network-3 (both of them). We also compare the learning performance with the conventional method which there is no chaotic noise application at all.

Figure 5 show the learning performance of all proposed and conventional network for different sinwave functions.. The horizontal axis is iteration time and the vertical axis is error value.



(a) Sinwave 1.

(b) Sinwave 2.

(c) Sinwave 3.

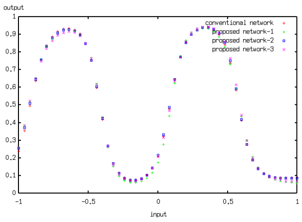
Figure 5: Learning performance of proposed and conventional network for different sinwave example .

From this figure, we confirm that our proposed network which chaotic noise is added into the weight update gains better performance than the conventional method when the chaos parameter of α is set to 0.55 and noise amplitude β is fixed as 0.01. The addition of chaotic noise during weight update can help the learning process to find a good solution in early time compared to the conventional backpropagation algorithm. Futhermore, the different position of weight update also give the different performance. From the simulation results, we confirm that when the chaotic noise is applied to both sides of weight update (from input layer to hidden layer and from hidden layer to output layer), the learning process gain better performance compared to others. Figure 6 also show the output results for all proposed and conventional method.

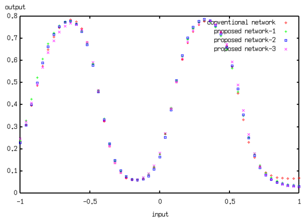
3.3. Noise amplitude and control parameter of chaos

We also study the effect of noise amplitude and chaos parameter to the learning performance. The different value of noise amplitude give the different learning results. Here, we only choose one sinwave example to compare the different value of noise amplitude. Figure 7 show the learning performance when the noise amplitude β is set to 0.001, 0.01 and 0.1 respectively.

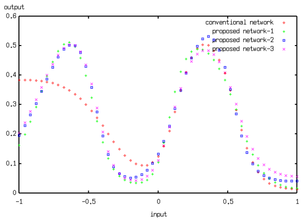
Choosing the approriate noise amplitude value is important for this work because it will effect the learning performance. From this results, we can see that when β is fixed as 0.001, there will be no improvement in learning performance because the noise amplitude is too small. On the



(a) Sinwave 1.

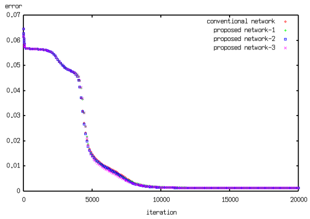


(b) Sinwave 2.

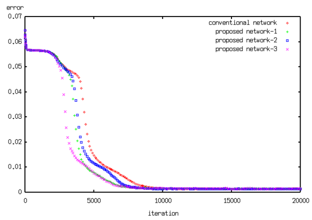


(c) Sinwave 3.

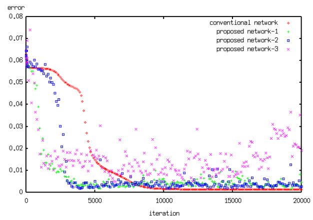
Figure 6: Output results of proposed and conventional network for different sinwave example.



(a) β=0.001.



(b) β=0.01.

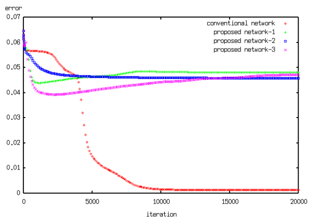


(c) β=0.1.

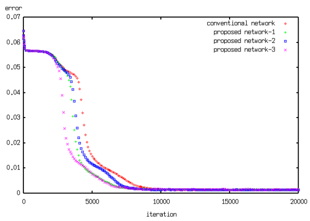
Figure 7: Learning performance of proposed and conventional network with different value of noise amplitude .

other hand, when β value is set to 0.1, the learning performance become not stable due to the large value of noise amplitude. Hence, we understand that the proposed network give good results if noise amplitude is set to 0.01.

Furthermore, we also investigate the effect of chaos parameter α to the learning performance. Figure 8 show the learning performance when chaos parameter α is set to 0.50 and 0.55 respectively. From skew tent map bifurcation, we know that chaos parameter α=0.55 show the fully-developed chaos. We confirm that the fully-developed chaos gains better learning performance.



(a) α=0.50.



(b) α=0.55.

Figure 8: Learning performance of proposed and conventional network with different value of chaos parameter.

4. Conclusions

In this study, we have conducted a further analysis of chaotic noise to the feedforward neural network. We have simulated the proposed method by introducing other difficult learning example and confirmed that the addition of chaotic noise into weight update process in backpropagation learning algorithm can gives a better convergence rate compared to standard backpropagation algorithm. We also found that different weight update position, noise amplitude and control parameter of chaos can give a big effect on the backpropagation learning performance.

References

[1] A. Sperduti and A. Starita, "Speed up learning and network optimization with extended back-propagation," *Neural Networks*, vol.6, pp.365-383, 1993.

[2] A. Miniani and R.D. Williams, "Acceleration of backpropagation through learning rate and momentum adaption," *Proceedings of IJCNN'90*,San Diego,CA, pp.676-679, 1990.

[3] R.A. Jacobs, "Increased rates of convergence thorugh learning rate adaption," *Neural Networks*, vol.1, pp.295-308, 1988.

[4] T. Ueta, Y. Nishio and T. Kawabe, "Comparison between Chaotic Noise and Burst Noise on Solving Ability of Hopfield Neural Networks," *Proceedings of NOLTA'97*, vol.1, pp.409-412, 1997.

[5] 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.

[6] Y. Uwate and Y. Nishio, "Chaotically Oscillating Sigmoid Function in Feedforward Neural Network," *Proceedings of NOLTA'06*, pp.215-218, 2006.

[7] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol.1, MIT Press, MA, pp.318-362, 1986.