NOLTA 2006
11-14 september
Bologna - ITALY

Chaotically Oscillating Sigmoid Function in Feedforward Neural Network

Yoko Uwate and Yoshifumi Nishio

Dept. of Electrical and Electronics Engineering,
Tokushima University, 2-1 Minami-Josanjima, Tokushima, Japan,
Phone: +81-88-656-7470, Fax: +81-88-656-7471,
Email: {uwate, nishio}@ee.tokushima-u.ac.jp.

Abstract—In this study, we propose the feedforward neural network with chaotically oscillating sigmoid function. By computer simulations, we confirm that the proposed neural network can find good solutions in early time of the back propagation learning process.

1. Introduction

Recently, studies on the human brain have been carried out actively on various levels. Many modelings of the human brain with the visual or the audio sensation are reported [1]-[3] due to development of the brain researches. However, the investigation of modeling of higher functions in the human brain is just getting started. We consider that it is very important to apply these high functional mechanisms of the human brain to novel artificial neural networks.

Back Propagation (BP) learning [4] 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 hidden layer and an output layer, and the effectiveness of the BP learning has been confirmed in pattern recognition, system control, signal processing, and so on [5]-[7]. The BP learning process requires that the input-output functions are bounded and differentiable functions. One of the most commonly used functions satisfying these requirements is the sigmoid function. The nonlinearity of the sigmoid function has an effect on modifying connection weights and it is very important for BP learning.

We consider that neurons in the human brain do not always output the same output for the same input. In order to reflect this idea to the feedforward neural networks, in this study, we propose the feedforward neural network with chaotically oscillating gradient of the sigmoid function. In order to confirm the effectiveness of the chaotically oscillating sigmoid function, we carry out computer simulations using other shaking methods. Further, we compare the proposed network to the Simulated Annealing (SA) method. By computer simulations, we confirm that the proposed network with chaotically oscillating sigmoid function can find good solutions in early time of the BP learning process.

2. Chaotically Oscillating Sigmoid Function

The BP learning process requires that the input-output functions are bounded and differentiable functions. One of the most commonly used functions satisfying these requirements is the sigmoid function. This function is an S shaped monotonic increasing function that has the general form as following equation:

$$f(x) = \frac{1}{1 + e^{-\varepsilon x}} \tag{1}$$

where ε is a constant that determines the steepness of the S shaped curve. Some curves of the function for different values of ε are illustrated in Fig. 1.

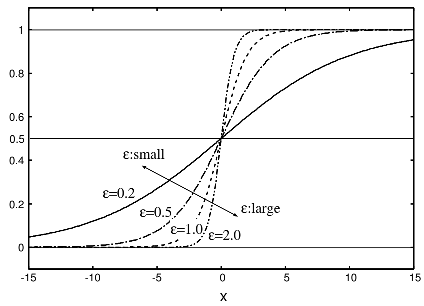


Figure 1: Sigmoid function.

The nonlinearity of the sigmoid function has an effect on modifying connection weights and it is very important for BP learning. We propose the feedforward neural network with chaotically oscillating gradient (ε) of the sigmoid function for BP learning.

The authors have investigated the performance of the Hopfield neural network solving combinatorial optimization problems when chaos is inputted to the neurons as noise [8]-[10]. By computer simulations, chaotic noise has been confirmed to gain better performance to escape out of local minima than random noise. Hence, we consider that various features of chaos are effective for neural networks.

The logistic map is used to shake ε of the sigmoid function chaotically:

$$\hat{\varepsilon}(t+1) = \alpha\hat{\varepsilon}(t)(1-\hat{\varepsilon}(t)). \qquad (2)$$

Varying the parameter α, Eq. (2) behaves chaotically via a period-doubling cascade. Further, it is well known that the map produces intermittent bursts just before periodic-windows appear. We apply the sequence generated by the logistic map to the sigmoid function after the following linear transform to set the standard as 1.0 and control the amplitude.

$$\varepsilon(t) = 2A(\hat{\varepsilon}(t)-1)+1 \qquad (3)$$

where A corresponds to the range of ε. One example of $A = 0.5$ is shown in Fig. 2.

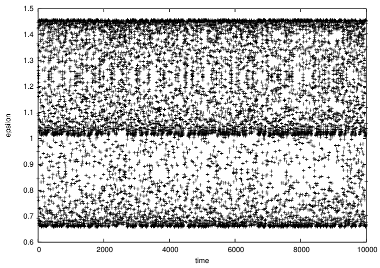


Figure 2: Oscillating ε by logistic map.

3. BP Learning Algorithm

The standard BP learning algorithm was introduced in [4]. The BP is the most common learning algorithm for feedforward neural networks. 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. The difference lies in the timing of the weight. The update of the standard BP is performed after each single input data, while for the batch BP the update is performed after all input data has been processed. The total error E of the network is defined as

$$E = \sum_{p=1}^{P} E_p = \sum_{p=1}^{P}\left\{\frac{1}{2}\sum_{i=1}^{N}(t_{pi}-o_{pi})^2)\right\}, \qquad (4)$$

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 as to minimize the total error E. In order to minimize E, the weights are adjusted according to the following equation:

$$w_{i,j}^{k-1,k}(m+1) = w_{i,j}^{k-1,k}(m) + \sum_{p=1}^{P}\Delta_p w_{i,j}^{k-1,k}(m),$$
$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta\frac{\partial E_p}{\partial w_{i,j}^{k-1,k}}, \qquad (5)$$

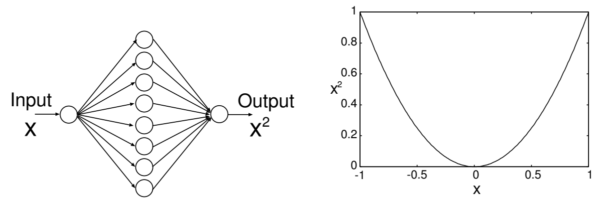
where $w_{i,j}^{k-1,k}$ is the weight between the ith neuron of the layer $k-1$ and the jth neuron of the layer k, m is the learning time, and η is a proportionality factor known as the learning rate. In this study, we add to the second line of Eq.(3) an inertia term, leading to

$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta\frac{\partial E_p}{\partial w_{i,j}^{k-1,k}} + \zeta\Delta_p w_{i,j}^{k-1,k}(m-1), \quad (6)$$

where ζ denotes the inertia rate.

4. Simulated Results

We consider the feedforward neural network producing outputs x^2 for inputs data x as one learning example. The sampling range of the input data is $[-1.0,\ 1.0]$ and the step size of the input data is set to be 0.01. We carried out the BP learning by using the following parameters. The learning rate and the inertia rate are fixed as $\eta = 0.2$ and $\zeta = 0.02$, respectively. The initial values of the weights are given between -1.0 and 1.0 at random. The learning time is set to 10000, and the 8 neurons are prepared in the hidden layer. The network structure using this study and learning example are shown in Fig. 3.



(a) Network structure. (b) Learning example.

Figure 3: Network structure and learning example.

4.1. Performance of Learning Process

We investigate the learning efficiency as the total error between the output and the desired target. We define "Average Error E_{ave}" by the following equation as mean square error.

$$E_{ave} = \frac{1}{P}\sum_{p=1}^{P}\left\{\frac{1}{2}(t_p - o_p)^2\right\} \qquad (7)$$

The bifurcation parameter of the logistic map is set to $\alpha = 3.8274$ generating intermittency chaos. The gradient of the sigmoid function of the conventional network is fixed as $\varepsilon = 1.0$. Figure 4 shows one example of the simulation results when the amplitude of ε of the sigmoid functions are changed. The horizontal axis is iteration time and the vertical axis is E_{ave}. This figure shows three learning curve of the proposed network and the conventional network, respectively, when the initial conditions of the connection weights are changed. From this figure, we can confirm that the proposed network with the chaotically oscillating sigmoid function gains better performance than the conventional network when the amplitude of ε is set to 0.5 (Fig. 4(a)). The learning curve of the proposed network converges oscillatory. On the other hand, the proposed network shows similar or weak performance to the conventional network when the amplitude of ε are set to 0.2 and 0.8 (Fig. 4(b) and (c)). We consider that the proposed network gains good performance by using appropriate amplitude of ε of the chaotically oscillating sigmoid function.

4.2. Comparison of Shaking Methods

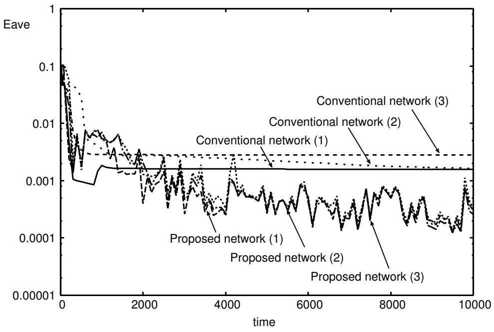
In this section, in order to confirm the effectiveness of the chaotically oscillating sigmoid function, we carry out the computer simulations by other shaking methods; fully developed chaos and at random. The fully developed chaos was realized by setting parameter of the logistic map (Eq. (2)) as $\alpha = 4.0000$. The simulation result is shown in Fig. 5. In this figure, the results of the four cases "intermittency chaos," "fully developed chaos", "random" and "conventional network" are shown.

We can confirm that the learning curve of the random method is more oscillatory than the intermittency chaos and the fully developed chaos. We consider that the chaotically oscillating ε is important to find good solution for BP learning.

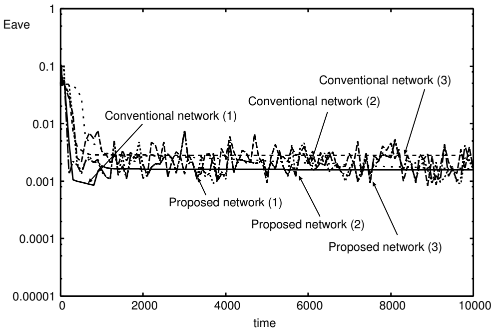
4.3. Efficient Learning

The proposed network with chaotically oscillating sigmoid function finds better solutions than the conventional network. However, the learning curve of the proposed network is oscillatory and do not converge. We consider that the proposed network can converge effectively, if oscillating ε stops on BP learning.

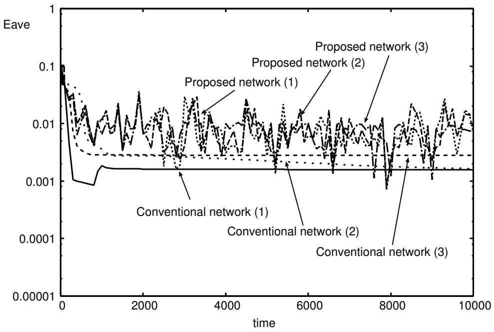
Figure 6 shows the simulated result when the stopped iteration time is set to 3000, 4000 and 5000. After oscillating stop, the gradient is fixed as $\varepsilon = 1.0$. From this figure, we can see that the proposed network converges to good solution by oscillating stop. We consider that it is important to oscillate the gradient of the sigmoid function in early time of learning process to find good solution.



(a) A=0.5.



(b) A=0.2.



(c) A=0.8.

Figure 4: Learning curve for difference range of ε.

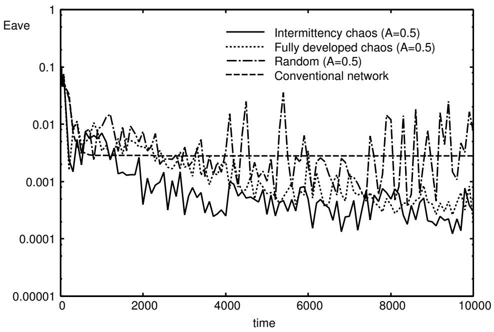


Figure 5: Learning curve for difference shaking methods.

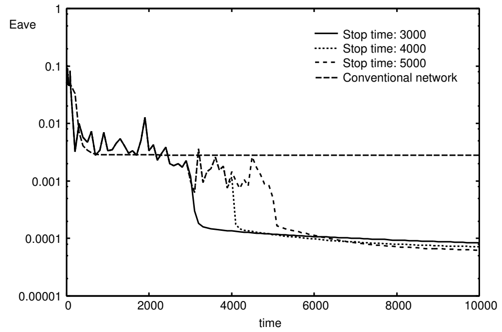
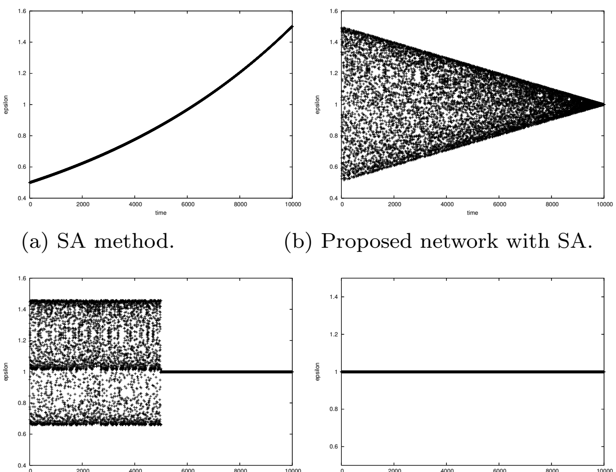


Figure 6: Learning curve for oscillating stop.

4.4. SA Method

Simulated annealing (SA) is a generic probabilistic meta-algorithm for global optimization problems, namely locating a good approximation to the global optimum of a given function in a large search space. SA can find a good solution by decreasing the gradient of the sigmoid function gradually. In this section, we investigate the performance of the SA method and the proposed network with the concept of SA. For comparison, the learning ability of the proposed network and the conventional network are investigated. The changing ε in these networks are shown in Fig. 7. The horizontal axis is time and the vertical axis is ε.



(a) SA method. (b) Proposed network with SA.

(c) Proposed network. (d) Conventional network.

Figure 7: Changing ε of the sigmoid function.

The simulated result is shown in Fig. 8. From this figure, the SA method and the proposed network with SA do not escape from local minima as well as the conventional network. We consider that the network needs some irregular change of neurons themselves in early learning time to find a good solution.

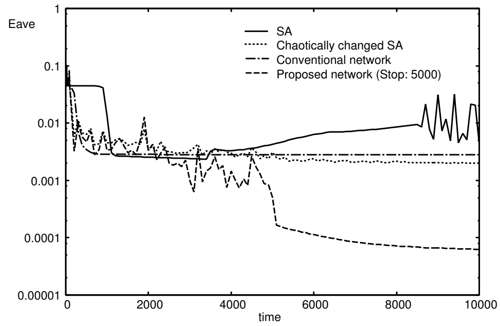


Figure 8: Learning curve of SA method.

5. Conclusions

In this study, we proposed the feedforward neural network with chaotically oscillating gradient of the sigmoid function. By computer simulations, we confirmed that the feedforward neural network with chaotically oscillating sigmoid function can find good solutions in early time of the BP learning process.

References

[1] R. P. Broussard, S. K. Rogers, M. E. Oxley and G. L. Tarr, “Physiologically Motivated Image Fusion for Object Detection Using a Pulse Coupled Neural Network,” IEEE Transactions on Neural Networks, 10 (3), pp. 554-563, 1999.

[2] R. Stoop and A. Kern, “Two-Tone Suppression and Combination Tone Generation as Computations Performed by the Hopf Cochlea,” Phys. Rev. Lett., 93 (26) 268103, 2004.

[3] R. Stoop, W.-H. Steeb, J.C. Gallas and A. Kern, “Auditory Two-Tone Suppression from a Subcritical Hopf Cochlea,” Physica A, 351: pp. 175-183, 2005.

[4] 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.

[5] Y. LeCun, B. Boser, J.S. Denker, et al, “Handwritten Digit Recognition with a Back-Propagation Network,” *Advances in Neural Information Processing Systems*, vol. 2, MIT Press, MA, 1990.

[6] M. Kawano, Y. Uno, M. Isobe and R. Suzuki, “A Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics,” *IEEE Control Systems Magazine*, vol. 8, pp. 8-16, 1988.

[7] T.J. Sejnowski and C.R. Rosenberg, “Parallel Networks that Learn to Pronounce English Text,” *Complex Systems*, vol. 1, pp. 145-168, 1987.

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

[9] Y. Uwate, Y. Nishio and A. Ushida, "Markov Chain Modeling of Intermittency Chaos and its Application to Hopfield NN,” *IEICE Transactions on Fundamentals*, vol. E87-A, no. 4, pp. 774-779, Apr. 2004.

[10] Y. Uwate, Y. Nishio, T. Ueta, T. Kawabe and T. Ikeguchi, “Performance of Chaos and Burst Noise Injected to the Hopfield NN for Quadratic Assignment Problems,” *IEICE Transactions on Fundamentals*, vol. E87-A, no. 4, pp. 937-943, Apr. 2004.