A New Method of Learning for Multi-Layer Neural Network

Rong-Long Wang[†], Cui Zhang[††] and Kozo Okazaki[†]

[†]Faculty of Engineering, University of Fukui, Bunkyo 3-9-1, Fukui-shi, Japan 910-8507
[††]Department of Autocontrol, LiaoNing Institute of Science and Technology, Benxi, China 117022

Summary
Backpropagation (BP) is one of the most widely used algorithms for training feed-forward neural networks. One critical drawback is that the BP easily falls into local minima. In this paper, we propose a new method of learning for multi-layer neural network which is not only an efficient method of selecting reasonable parameter but also a supervised method of preventing the BP to be trapped at some local minima. The proposed method is tested through some benchmark problems. For all problems, the systems are shown to be trained efficiently by the proposed method.

Key words:
Multilayer neural network, Backpropagation, Local minima, Learning.

1. Introduction

Neural networks, viewed as adaptive nonlinear filters or nonlinear systems such as nonlinear autoregressive moving average with exogenous input model, have draw great interest [1][2][3]. The traditional method for training a multilayer perceptron is the standard backpropagation algorithm [4] (SBP). Although it is successfully used in many real-word applications [5], the SBP algorithm suffers from a number of shortcomings. One of which is the local minimum problem. As primarily deterministic algorithms, SBP will attempt to take the best path to the nearest minimum, whether global or local. If a local minimum is reached, the network will fail to learn. Local minima are known to be a serious obstacle to successful training when multilayer network are applied to practical task. Some existing approaches modify the SBP in order to help the network escape from local minima. Lehmen et al. [6] added noise to the weights during training for improved learning probability. Abunawass and Owen [7] also generalized the process of adding noise to weights. Hanson [8] based weight adjustment on a stochastic rule, with a weight represented as a mean of a probabilistic distribution. However, each of these solutions attempts to add a random factor to the model that will overcome the tendency to sink into local minima. The random perturbations of the search direction are not effective at enabling network to escape from local minima and make the network fail to converge to a global minimum within a reasonable number of iterations [9][10].

This paper proposes a new method of learning which is not only an efficient method of selecting reasonable parameter but also a supervised method of preventing the BP to be trapped at some local minima. In the new method, sigmoid function is used as the activation function of neuron. Neuron of each layer has associated temperature parameter in the activation function. The temperature parameters are determined through learning. We analyze the new method and find that the new method can overcome some local minima. The new method is applied to the parity problem and the numeric font recognition problem. For all problems, the systems are shown to be trained efficiently by the proposed new method. This paper is divided into five parts. Section II briefly presents the SBP algorithm. Section III introduces the new method including the learning method of temperature parameter in sigmoid function. In Section IV, experimental results and comparisons between the two methods are given. Finally, in Section V, we present the main conclusions.

2. Standard Backpropagation Algorithm

A multilayer feedforward neural network usually has one output layer and one input layer with one or more hidden layer. Each layer has a set of neuron. Each neuron has a threshold. It is usually assumed that each layer is fully connected with an adjacent layer without direct connections between layers that are not consecutive. Each connection has a weight. The input of *j*# neuron in the *s*# layer is given by

$$u_j^s = \sum_{i=0}^{n_{s-1}} w_{ji}^s y_i^{s-1} \tag{1}$$

where w_{ji}^s is the weight of the *i*th neuron in the (*s*-1)th layer to the *j*th neuron in the *s*th layer. Note that w_{j0}^s is the threshold of *j*# neuron in the *s*# layer. In general, sigmoid function is used as activation function of neuron. Thus, the output of neuron is specified by

$$o_j^s = \frac{1}{1+e^{-Tu_j^s}} \tag{2}$$

The SBP algorithm is widely used as method for training multilayer feedforward neural network. The SBP algorithm

attempts to find a set of weights that minimizes an overall error function E:

$$E = \sum E_p(\vec{W}) \tag{3}$$

where p indexes over all the patterns in the training set and $\vec{W}$ is a vector whose elements include all weights of neurons. E_p is defined by:

$$E_p = \frac{1}{2}\sum_{j=1}^{N}(t_j^p - o_j^p) \tag{4}$$

where N is the number of neuron in the output layer. t_j^p and o_j^p are, respectively, the desired and the current outputs for the jth neuron of output layer. The SBP algorithm is based on the following gradient descent rule:

$$\Delta w_{ji}^s = -\mu \frac{\partial E_p}{\partial w_{ji}^s} \tag{5}$$

where μ is a parameter called learning coefficient.

The SBP causes each iteration to modify weights in such way as to approximate the steepest descent. The parameter T in Eq.(2) is called temperature parameter. In SBP, it is often set to a constant value and is not changed by the learning algorithm. The solution quality of SBP is always influenced by the parameter [11]. However there is no efficient method to select reasonable value of parameter. Besides, the SBP will attempt to take the best path to the nearest minimum, whether global or local. If a local minimum is reached, the network will fail to learn. In the next section, we propose a method which is not only an efficient method of selecting reasonable temperature parameter but also a supervised method of preventing the backpropagation algorithm to be trapped at some local minima.

3. New Method of Learning

In this section, we propose a new learning method that adjusts temperature parameter T in the activation function of neuron by learning algorithm in order that a reasonable parameter can be selected and some local minima of the error function vanishes. The new learning method can be considered as a modified SBP which has following two new rules:

(1) Neuron of each layer has an associated temperature parameter in the activation function. We use T_k to denote the associated temperature parameter of neurons in k# layer.

(2) T_k can be changed by the following gradient descent rule:

$$\Delta T_k = -\mu \frac{\partial E_p}{\partial T_k} \tag{6}$$

where E_p is defined in Eq.(4) and μ is as the same as that in Eq.(5).

From rule (2), we can see that the value of temperature parameter can be trained by learning and the learning rule (Eq.(6)) means that adjusting temperature parameter can lead a descent of the error measure function. Besides, from the new rule (1), we know that neurons in different layer have different temperature parameter. For a K layer neural network, the temperature parameter has total K-1. Thus, in the new method the error function can be written as follow:

$$E_{new} = \sum E_p(\vec{W}, \vec{T}) \tag{7}$$

The E_{new} (Eq.(7)) can be considered as a modified error function by adding some dimensions to error function E (Eq.(3)). The new learning algorithm attempts to find a set of weights and temperature parameters that minimizes an overall error function E.

We analyze the difference between the SBP and the new method. In SBP, Eq.(3) is used as error function, thus once the network fall into a local minimum of E (Eq.(3)), the SBP will stop learning. On the other hand in the new method, E_{new} (Eq.(7)) is a modified error function that added some dimensions to error function E (Eq.(3)). For a multi-dimension function, a local minimum must be a minimum on every axis. Thus, a local minimum of E (Eq.(3)) is not always a local minimum of E_{new} (Eq.(7)). In other words, some sets of weight is a local minima in the error function E of SBP, however these sets may be not a local minimum in the new method. For an example, for a given set of temperature parameters ($\vec{T}_1$), set of weight $\vec{W}$ is a local minimum of error function. However temperature parameters can be changed according Eq.(6). After the temperature parameters changed, set of weight $\vec{W}$ may not be a local minimum of error function. Thus, we can say that in the proposed MBP, some local minima could be avoided.

We derive the explicit rule for the change in temperature parameter. For the sake of simplicity, the extension of the backpropagetion on a feedforward network with one hidden layer will be discussed. The extension to networks with any number of hidden layers is straightforward. Let N_h and N_o be the number of hidden and output neurons of network; o_j^p and y_j^p be the output of neurons in the output and hidden layers. Then we have the change in temperature parameters as follows:

$$\begin{aligned}\Delta T_o &= -\mu \frac{\partial E_p}{\partial T_o} = \mu \sum_{j=1}^{N_o}(t_j^p - o_j^p)\frac{\partial o_j^p}{\partial T_o} \\ &= \mu \sum_{j=1}^{N_o}(t_j^p - o_j^p)\cdot u_j \cdot o_j^p \cdot (1 - o_j^p)\end{aligned} \tag{8}$$

where u_j^p is the input of j# neuron in output layer. And for the hidden layer, we have the following derivation:

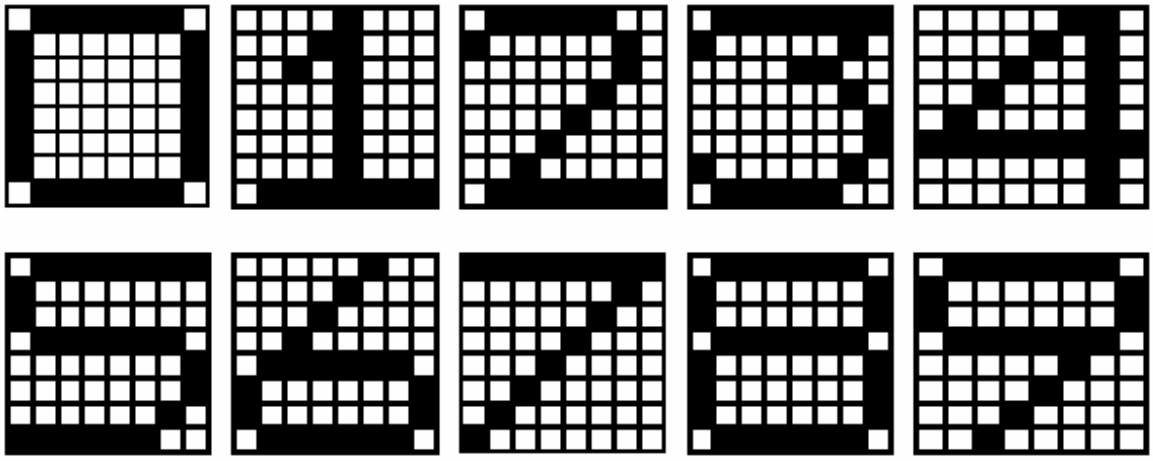

Fig. 2 Training set for the 8×8 dot numeric font recognition

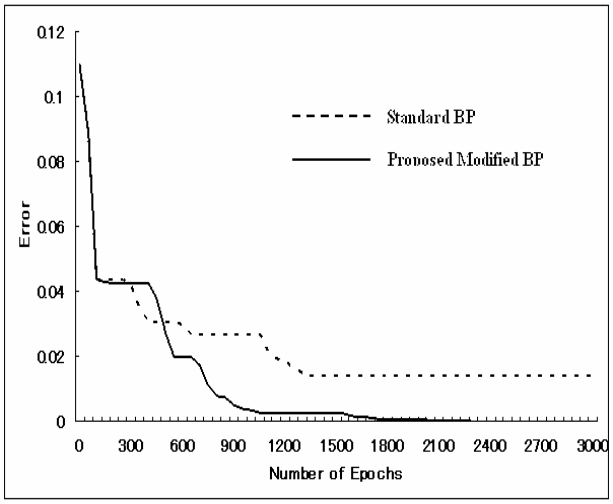

Fig. 1 Learning curve for the proposed MBP and the SBP on 2-bit parity problem.

$$
\begin{aligned}
\Delta T_h &= -\mu \frac{\partial E_p}{\partial T_h} \\
&= \mu \sum_{j=1}^{N_o} (t_j^p - o_j^p) \frac{\partial o_j^p}{\partial T_h} \\
&= \mu \sum_{j=1}^{N_o} (t_j^p - o_j^p) \cdot o_j^{p'} \cdot \frac{\partial (\sum_{i=0}^{N_h} w_{ji} \cdot y_i^h)}{\partial T_h} \\
&= \mu \sum_{j=1}^{N_o} (t_j^p - o_j^p) \cdot o_j^{p'} \cdot \sum_{i=0}^{N_h} w_{ji} \cdot \frac{\partial y_i^h}{\partial T_h} \\
&= \mu \sum_{j=1}^{N} (t_j^p - o_j^p) \cdot o_j^{p'} \cdot \sum_{i=0}^{N_h} w_{ji} \cdot u_i^h \cdot y_i^h \cdot (1 - y_i^h)
\end{aligned}
\tag{9}
$$

Note that w_{j0} represents the threshold of output neurons.

4. Simulation Results

In order to test the effectiveness of the proposed learning algorithm, two examples – the parity problem and the numeric font recognition problem were used in simulations for experimental purposes. We compare the performance of the proposed method with that of SBP. For all methods, the weights and thresholds were initialized randomly from -1.0 to 1.0 and the learning rate $\mu = 0.5$. The temperature parameter of the activation function in SBP is set at 1.0. In order to give a fair comparison, the initial temperature parameter in the proposed method is also set at 1.0.

The parity problem is one of the most popular tasks given a good deal of discussion. In this problem, the output required is 1 if the input pattern contains an odd number of 1's and 0 otherwise. The parity problem is a very demanding classification task for neural network to solve, because the target-output changes whenever a single bit in the input vector change and *N*-parity training set consists of 2^N training pairs. Our first simulation was performed on the 2-bit parity problem. Both the proposed method and the SBP used a simple architecture with one hidden layer and two hidden neurons. Figure 1 show a typical learning curve for the proposed method and the SBP for the 2-bit parity problem. From the figure, we note that the proposed method finally reached 0.000001 after 2530 iterations as opposed to the SBP remain 0.014 after 1350 iterations which can be considered that the network fell into a local minimum. It is clear that the proposed method can find better solution than the SBP. We have tried a number of parity problems with input patterns ranging from size two to four. We use an *N*-*N*-1 (*N*-input, *N*-hidden neurons and

1-output) architecture network to solve the *N*-bit parity problem. Our simulation found that the proposed method significantly outperformed the SBP in success rate and training speed.

To show the effectiveness of the proposed method for high-dimensional problem, we applied the proposed method to a 8×8 dot numeric font recognition problem which is a classical pattern classification problem. For this problem, we used a 64-6-10 network where each output neuron is associated with an input pattern by which it is activated and each input neuron corresponds to a dot in the 8×8 pattern grid. Figure 2 shows the train set. If the dot is white (black), zero (one) is input to the corresponding input neuron. Using the proposed method and the SBP, we trained the problem respectively. The convergence rate of the proposed method and the SBP are 73% and 32% respectively. It is clear that the proposed method has better performance than the SBP on the numeric font recognition problem.

5. Conclusions

This paper proposed a new learning method for multi-layer neural network. The proposed method was designed to be of higher convergence to global minimum than the standard backpropagation algorithm. The proposed method is applied to parity problem and the numeric font recognition problem. The simulation results on the two problems show that the proposed method significantly outperformed the SBP in 'success rate' and 'training speed'. Thus, it could say that the proposed method is not only an efficient method of selecting reasonable parameter but also a supervised method of preventing the BP to be trapped at some local minima.

References

[1] A. Cichocki and R. Unbehauen, "Neural Network for Optimization and Signal Processing," Chichester, U. K. Wiley 1993.

[2] S. Haykin, Adaptive Filter Theory, Englewood Cliffs, NJ: Prentice Hall, 1986.

[3] R. P. Lippman, "An introduction to computing with neural neyworks," IEEE ASSP. Mag., Vol.4, No.2, Apr., 1987.

[4] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by back-propagating errors, in: D.E. Rumelhart, J.L. McClelland, the PDP Research Group (Eds.), Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge, MA, 1986, pp. 318_362.

[5] J. Alirezaie, M. E. Jernigan and C. Nahmias, "Neural network based segmentation of magnetic resonance images of the brain," IEEE Nuclear Science Symposium and Medical Imaging Conference Record, Vol.1, pp.1397-1401, 1995.

[6] A. Von Lehmen, E.G. Paek, P.F. Liao, A. Marrakchi, J.S. Patel, "Factors influencing learning by backpropagation," Proceedings of the IEEE International Conference on Neural Networks, Vol.I, 988, pp. 335-341, 1988.

[7] A. M. Abunawass and C. B. Owen, "A statistical analysis of the effect of noise injection during neural network training," SPIE Proceedings, Vol.1966, pp.362-371, 1993.

[8] S. J. Hanson, "Behavioral Diversity, Search & Stochastic Connectionist System," In Quantitative Analysis of Behavior: Neural Network Model of Conditioning and Action, pp.295-344, Cambridge, MA: Harvard Press.

[9] D. Ingman and Y. Merlis, "Local minimization escape using thermodynamic properties of neural networks," Neural Networks, Vol.4, No.3, pp.395-404, 1991.

[10] C. Wang and J. C. Principe, "Training neural networks with additive noise in the desired signal," IEEE Transactions on Neural Networks, Vol.10, No.6, pp.1511-1517, 1999.

[11] M.R. Meybodi., H. Beigy, "A note on learning automata-based schemes for adaptation of BP parameters," Neurocomputing, Vol.48, pp.954-974, 2002.