Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Neurocomputing 56 (2004) 455–460

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Letters

An improved backpropagation algorithm to avoid the local minima problem

X.G. Wang, Z. Tang*, H. Tamura, M. Ishii, W.D. Sun

Faculty of Engineering, Toyama University, 3190 Gofuku, 930-8555 Toyama, Japan

Received 10 May 2003; accepted 29 July 2003

Abstract

We propose an improved backpropagation algorithm intended to avoid the local minima problem caused by neuron saturation in the hidden layer. Each training pattern has its own activation functions of neurons in the hidden layer. When the network outputs have not got their desired signals, the activation functions are adapted so as to prevent neurons in the hidden layer from saturating. Simulations on some benchmark problems have been performed to demonstrate the validity of the proposed method.

Keywords: Backpropagation; Local minima; Saturation; Gain parameter

1. Introduction

The backpropagation algorithm [6] is well known to have difficulties with local minima. Most existing approaches [5,10] modify the learning model in order to add a random factor to the model, which overcomes the tendency to sink into local minima. However, the random perturbations of the search direction and various kinds of stochastic adjustment to the current set of weights are not effective at enabling a network to escape from local minima and they may make the network fail to converge to a global minimum within a reasonable number of iterations [11].

We have noted that many local minima difficulties are closely related to the neuron saturation in the hidden layer. Once such saturation occurs, neurons in the hidden layer will lose their sensitivity to input signals, and the propagation of information is blocked severely [3]. In some cases, the network can no longer learn [1]. The same

* Corresponding author. Tel.: +81-76-445-6752; fax: +81-76-445-6752.
E-mail addresses: wang@hi.iis.toyama-u.ac.jp (X.G. Wang), tang@iis.toyama-u.ac.jp (Z. Tang).

0925-2312/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2003.08.006

phenomenon is also observed and discussed by Andreas Hadjiprocopis [3], Christian Goerick [1] and Simon Haykin [4].

We propose an improved backpropagation algorithm to help the network avoid the local minima problem due to such neuron saturation in the hidden layer. This is accomplished without changing the network topology or consuming more computation time. Simulation experiments on several benchmark problems are presented and the validity of our method is substantiated.

2. Algorithm

Neuron saturation in the hidden layer here refers to situations where the input signals to the hidden layer nodes are so high (or so low) that all the neurons in the hidden layer are forced to produce an output response very close to the bound of the activation function, while the network outputs have not obtained their desired results yet. Hence we will adjust sigmoid activation functions in the hidden layer for each pattern. This is done in order to enable the weights connected to the hidden layer and the output layer to be modified harmoniously. Usually, the activation function of a neuron $f(x)$ is given by a sigmoid function with the 'gain' parameter:

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

where g is the 'gain' parameter. The gain parameters are usually set to 1.0 and not changed by the learning rule. Here, we define g_p as the gain parameter of the activation function in the hidden layer for pattern p. The parameter g_p should be adjusted according to the degree of approximation to the desired output of the output layer. To obtain such a value, we need to introduce two parameters. One is defined as

$$e_p = \max_j (|t_{pj} - o_{pj}|). \tag{2}$$

The other parameter H denotes the average value of the difference between teacher signals. It is a constant for a given learning task. For most classification problems whose teacher signals are either '1' or '0', it can be defined as

$$H = \frac{(1-0)}{2} = 0.5. \tag{3}$$

Then the approximation degree of the output layer A_p for pattern p is given by

$$A_p = \frac{e_p}{H}, \tag{4}$$

where A_p decreases as the network approximates to the teacher signal. Therefore, the gain parameters for pattern p are adapted as the following rule:

$$g_p = \begin{cases} \dfrac{1}{A_p} = \dfrac{H}{e_p} & \text{if } A_p > 1.0, \\ 1.0 & \text{else.} \end{cases} \tag{5}$$

Then, the new g_p is applied to all the hidden neurons during the training for pattern p. We can see from the above formulation that the gain parameters for all patterns will be adapted back to 1.0. This happens when the network starts to approximate to the teacher signal. Therefore, the network topology will not be changed. Moreover, not much additional computation will be needed.

3. Simulations

In order to test the effectiveness of the proposed method, we compare its performance with those of the backpropagation algorithm and a global search technology—simulated annealing method [5] on several benchmark problems. In our simulations, we use the modified backpropagation algorithm called the backpropagation with momentum algorithm [6]. The process of the simulated annealing method is based on [5] and the same parameters as in [5] are used in our simulations. For all methods, the learning rate $\eta = 0.1$ and the momentum term parameter $\alpha = 0.9$ are used in all experiments. The weights and thresholds are initialized randomly from $(-1.0, 1.0)$, and the gain parameters of all neurons are set to 1.0 initially.

Two aspects of training algorithm performance—'success rate' and 'training speed' are assessed for each algorithm. A training run is deemed to have been successful if the network successfully classified all patterns in the training set within a tolerance of 0.1 for each target element. Then, successful training runs are allowed to run on to lower error levels E ($E = 0.01$ and $E = 0.0001$), where E is the *sum-of-squares* error function for the full training set. The number of '*equivalent function evaluations*'—the mean number of the forward passes and backward passes (denoted by '*EFEs*' and defined in [7]) required to attain the given error level E is used as an appropriate measure of training speed. For all trials, the upper limit epochs are set to 50 000.

3.1. The modified XOR problem

The modified *XOR* problem is different from the classical *XOR* problem because one more pattern is included (that is, inputs = (0.5, 0.5) and teacher signal = 1.0) such that a unique global minimum exists. Furthermore, several local minima exist simultaneously in this problem [2]. We used a 2-2-1 neural network to solve this problem.

Fig. 1 shows the typical error curves of these algorithms. From this figure, we can see that the training process of the backpropagation algorithm is stopped because of the local minimum problem. The simulated annealing method could run successfully after 48 050 epochs by adding stochastic noise to the weights; however, it involved too much computation. Meanwhile, the proposed method can avoid such a local minimum problem and train the network successfully and rapidly within 1400 epochs.

Table 1 shows the experiment results of the three methods based on 100 runs of this problem. It can be seen from the table that the proposed method can obtain successful solutions for almost every run, while the backpropagation algorithm and the simulated annealing method show many failures in convergence to the global solution. Meanwhile, the mean of *EFEs* of the proposed method is almost the same as that of the backpropagation in every case. But it is much less than that of the simulated annealing

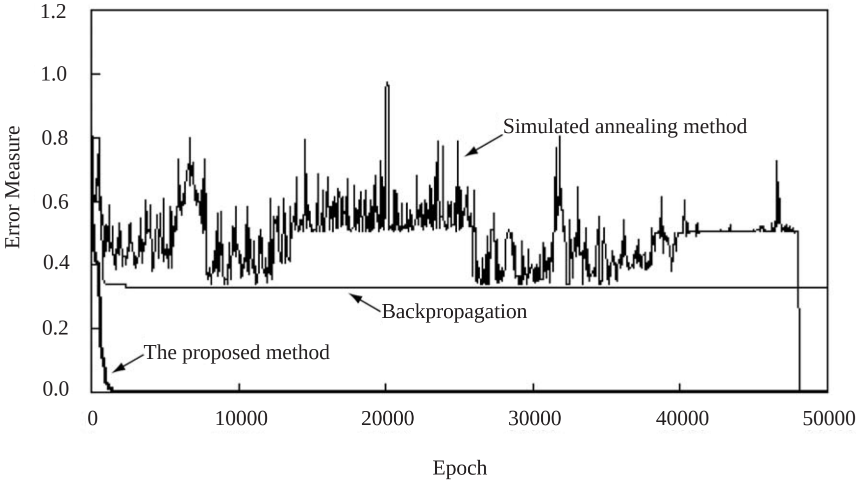

Fig. 1. Comparison of the typical error curves for each training algorithm.

Table 1
Experiment results for the modified *XOR* problem

Methods	Success rate	Mean of *EFEs*	
		$E = 0.01$	$E = 0.0001$
Backpropagation	71%	1024	28 776
Simulated annealing	42%	10 325	16 291
Our proposed method	98%	1410	29 223

method when $E = 0.01$ is used. These results indicate that the local minima problem can be avoided by the proposed method efficiently.

3.2. Parity problem

The parity problem is also 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. We have tried a number of parity problems with input patterns ranging from size two to four. An N-N-1 (N-input, N-hidden neurons and 1-output) architecture is used for the N-bit parity problem. Ida G. Sprinkhuizen-Kuyper et al. [8,9] have described the local minima problem in the 2-bit parity (XOR) network. He showed that these regions of local minima occurred for combinations of the weights from the inputs to the hidden nodes such that one or both the hidden nodes were saturated (given output 0 or 1) for at least two patterns [8]. Thus, the XOR problem is a very good example for testing our local minima avoidance method.

Table 2
Experimental results for parity problems

Methods	Success rate	Mean of EFEs	
		$E = 0.01$	$E = 0.0001$
(a) *The 2-bit parity (XOR) problem*			
Backpropagation	79%	1112	29 781
Simulated annealing	84%	12 801	15 829
Our proposed method	97%	895	26 078
(b) *The 3-bit parity problem*			
Backpropagation	100%	1355	35 537
Simulated annealing	26%	19 307	30 683
Our proposed method	100%	1064	35 611
(c) *The 4-bit parity problem*			
Backpropagation	56%	13195	$>$50 000
Simulated annealing	0%	$>$50 000	$>$50 000
Our proposed method	97%	2571	47 885

Table 2 gives the statistics based on 100 trials of every parity problem. As noted in the parity problems, there are many learning failures almost all of which can be avoid by our proposed method. As a result, the proposed method greatly outperforms both the backpropagation algorithm and the simulated annealing method in terms of success rate, while still maintaining the same speed of learning as the backpropagation algorithm for two- and three-bit parity problems. It has also been shown that the mean *EFEs* of our proposed method is much less than that of the backpropagation algorithm for the 4-bit parity problem, which implied that our method might also accelerate the learning by eliminating any standstill state during learning. In theory, the simulated annealing method can converge to a global minimum successfully. In practice, however, it is so slow that it cannot converge within the given upper limit epochs. Thus the low training speed resulted in a high failure rate for this method.

4. Conclusions

We have proposed an improved learning method for multilayer feedforward neural networks. In this method, each training pattern has its own activation functions of neurons in the hidden layer. The activation functions are adjusted by the adaptation of gain parameters during the learning process. These adjustments are made in order to prevent the network from trapping into a local minimum caused by the neuron saturation in the hidden layer. When the network starts to approximate to the teacher signals, the gain parameters of all patterns will be adapted back to their original values. Thus, the proposed method does not change the network topology and does not require additional computation. Finally, our proposed method has been indicated to be very effective in avoiding the local minima by testing it and comparing the results with those of the backpropagation algorithm and the simulated annealing method on several benchmark problems. More analyses on large problems and practical applications are required.

References

[1] C. Goerick, W.V. Seelen, On unlearnable problems or A model for premature saturation in back-propagation learning, in: Proceedings of the European Symposium on Artificial Neural Networks ’96, Brugge, Belgium, 24–26 April 1996, pp.13–18.

[2] M. Gori, A. Tesi, On the problem of local minima in backpropagation, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1) (1992) 76–86.

[3] A. Hadjiprocopis, Feed forward neural network entities, Ph. D. Thesis, Department of Computer Science, City University, London, UK, 2000.

[4] S. Haykin, Neural Networks, a Comprehensive Foundation, Macmillan Publishing, New York, 1994.

[5] C.B. Owen, A.M. Abunawass, Application of simulated annealing to the backpropagation model improves convergence, in: Proceedings of the SPIE Conference on the Science of Artificial Neural Networks II, Vol. 1966, Orlando, FL, USA, 13–16 April 1993, pp. 269–276.

[6] 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.

[7] A.J. Shepherd, Second-order Methods for Neural Networks: Fast and Reliable Training Methods for Multi-layer Perceptrons, Springer, London, 1997.

[8] I.G. Sprinkhuizen-Kuyper, E.J.W. Boers, The error surface of the 2-2-1 XOR Network: Stationary Points with infinite Weights, Technical Report 96-10, Dept. of Computer Science, Leiden University, Leiden, The Netherlands, 1996.

[9] I.G. Sprinkhuizen-Kuyper, E.J.W. Boers, The error surface of the 2-2-1 XOR network: the finite stationary points, Neural Networks 11 (1998) 683–690.

[10] A. Von Lehmen, E.G. Paek, P.F. Liao, A. Marrakchi, J.S. Patel, Factors influencing learning by backpropagation, in: Proceedings of the IEEE International Conference on Neural Networks, Vol. I, San Diego, California, 1988, pp. 335–341.

[11] C. Wang, J.C. Principe, Training neural networks with additive noise in the desired signal, IEEE Trans. Neural Networks 10 (6) (1999) 1511–1517.