Backpropagation with Two-Phase Magnified Gradient Function

Chi-Chung Cheung, *Member*, *IEEE* and Sin-Chun Ng, *Member*, *IEEE*

***Abstract*—Backpropagation (BP) learning algorithm is the most widely supervised learning technique which is extensively applied in the training of multi-layer feed-forward neural networks. Many modifications have been proposed to improve the performance of BP, and BP with Magnified Gradient Function (MGFPROP) is one of the fast learning algorithms which improve both the convergence rate and the global convergence capability of BP [19]. MGFPROP outperforms many benchmarking fast learning algorithms in different adaptive problems [19]. However, the performance of MGFPROP is limited due to the error overshooting problem. This paper presents a new approach called BP with Two-Phase Magnified Gradient Function (2P-MGFPROP) to overcome the error overshooting problem and hence speed up the convergence rate of MGFPROP. 2P-MGFPROP is modified from MGFPROP. It divides the learning process into two phases and adjusts the parameter setting of MGFPROP based on the nature of the phase of the learning process. Through simulation results in two different adaptive problems, 2P-MGFPROP outperforms MGFPROP with optimal parameter setting in terms of the convergence rate, and the improvement can be up to 50%.**

I. Introduction

BACKPROPAGATION (BP) learning algorithm [1] is the most widely supervised learning technique which is extensively applied in the training of multi-layer feed-forward neural networks. The success of BP is due to its simplicity and low computational complexity. However, the convergence rate of BP is slow and it is easily trapped in local minima, especially for non-linearly separable problems such as the exclusive OR (XOR) problem [2, 3]. Because of these weaknesses, many modifications have been proposed to improve the performance of BP. Among all modifications, many of them have focused on solving the "flat spot" problem to increase the convergence rate [12 – 14, 17, 19]. The "flat spot" problem is the occurrence of premature saturation in the derivative of the activation function [4 – 7]. If a learning algorithm is trapped into a "flat spot", the learning process and weight update will become very slow or even suppressed. Quickprop [12] predicts the location of a minimum and adjusts the weights to converge to the global optimal solution by considering successive values of the gradient of the error surface. RPROP [13] is a variable step size algorithm which varies the weight update to adapt the learning process. SARPROP [14] is modified from RPROP. It employs Simulated Annealing (SA) [15, 16] to enhance the global convergence capability. The integration of both SA and RPROP improves both the convergence rate and the global convergence capability of RPROP. The Levenberg-Marquardt algorithm (LMA) [17] is a hybrid of the method of gradient descent and the Guass-Newton algorithm (GNA) [18] to increase the convergence rate. The above modified BP algorithms increase the convergence rate of BP. However, they still suffer from the "flat spot" problem and sometimes trap in local minima.

MGFPROP (Backpropagation with Magnified Gradient Function) has recently been proposed [19]. It increases the convergence rate and the global convergence capability by magnifying the gradient function of the activation function without violating the gradient-descent property of the BP algorithm. The magnified gradient can escape from a "flat spot" effectively. Simulation results in [19] show that MGFPROP usually outperforms the above well-known modified BP methods in terms of the convergence rate and the global convergence capability. However, its performance is limited due to the error overshooting problem. When the parameter setting of MGFPROP is too conservative, its convergence rate is fast but limited. However, when the parameter setting of MGFPROP is too aggressive, the system error may be overshot and thus MGFPROP spends more epochs to converge to the global optimal solution, and sometimes even diverges.

Based on this observation, this paper proposes a new approach called BP with Two-Phase Magnified Gradient Function (2P-MGFPROP) to overcome the error overshooting problem and hence speed up the convergence rate of MGFPROP. 2P-MGFPROP is modified from MGFPROP. It divides the learning process into two different phases: the convergence-searching phase and the convergence-guarantee phase. In the convergence-searching phase, the effect of the error overshooting problem is significant; while in the convergence-guarantee phase, the effect of the error overshooting problem is minimal. Based on the different nature of the phase, the parameter setting of MGFPROP is adjusted accordingly to perform better. Through simulation results in two different adaptive problems, it is found the convergence rate of 2P-MGFPROP is faster than the optimal performance of MGFPROP, and the improvement can be up to 50%.

This paper is organized as follows. Section II introduces the basic operations of the standard BP and MGFPROP algorithms. Section III describes the limitation of MGFPROP by investigating its performance in two different adaptive

Chi-Chung Cheung is with Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong, China (e-mail: encccl@polyu.edu.hk).

Sin-Chun Ng is with the School of Science and Technology , The Open University of Hong Kong, Hong Kong, China (email: scng@ouhk.edu.hk).

978-1-4244-1821-3/08/$25.00©2008 IEEE

problems. Section IV presents 2P-MGFPROP and compares its performance with the standard BP and MGFPROP algorithms. Finally, conclusions are drawn in Section V.

II. The Standard BP and MGFPROP

A. The Standard BP Algorithm

Consider the basic structure of a feed-forward network with a single hidden layer as shown in Fig. 1. The network consists of N input nodes, K hidden nodes and M output nodes. Sigmoidal functions are used as the activation functions for both the hidden and output layers. The notations used in the standard BP algorithm are listed in Table I.

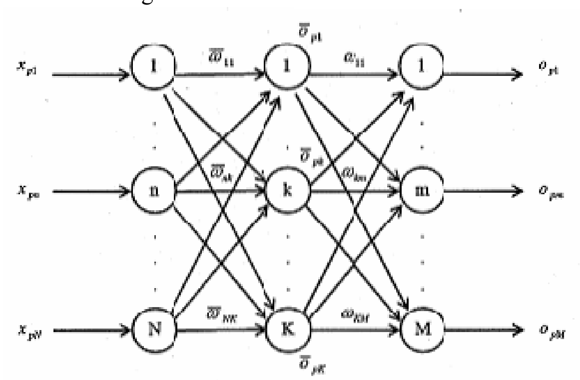


Fig. 1. The basic structure of a feed-forward network with a hidden layer.

TABLE I
NOTATIONS USED IN THE STANDARD BP ALGORITHM

Notation	Description
$\overline{o}_{pk}$	The output of the hidden node k from the input pattern p
o_{pm}	The output of the output node m from the input pattern p
$\overline{\omega}_{nk}$	The network weight for the input node n and the hidden node k
ω_{km}	The network weight for the hidden node k and the output node m
x_{pn}	The input value in the input node n for the input pattern p
Δ	The difference between the current and new value in the next iteration

The standard Backpropagation (BP) algorithm is shown below:

1) *Initialization*: Initialize all weights and refer to them as the current weights $\omega_{km}(0)$ and $\overline{\omega}_{nk}(0)$ for all k, m, and n. Set the learning rate μ and the momentum factor α to small positive values (e.g., 0.1). Set the error threshold to a very small positive value and the iteration number $i = 0$.
2) *Forward Pass*: Select an input pattern $\mathbf{x}_p = \{x_{p1}, \ldots, x_{pN}\}$ from the training set and compute $o_{pm}(i)$ and $\overline{o}_{pk}(i)$ by using the following equations:

$$o_{pm}(i) = f\left(\sum_{k=1}^{K} \omega_{km}(i)\overline{o}_{pk}(i)\right) \tag{1}$$

$$\overline{o}_{pk}(i) = f\left(\sum_{n=1}^{N} \overline{\omega}_{nk}(i)x_{pn}\right), \tag{2}$$

where

$$f(x) = \frac{1}{1+e^{-x}}. \tag{3}$$

Use the desired target $\mathbf{t}_p = \{t_{p1}, \ldots, t_{pM}\}$ associated with $\mathbf{x}_p$ to compute the squared error, $E(i)$, for all input patterns as follows:

$$E(i) = \frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M}[t_{pm} - o_{pm}(i)]^2 . \tag{4}$$

If $E(i)$ is not greater than the error threshold, then the algorithm is completed and the convergence is met; otherwise, go to step 3 (Backward Pass).

3) *Backward Pass*: Compute the changes of the weights for the next iteration $\Delta\omega_{km}(i+1)$ and $\Delta\overline{\omega}_{nk}(i+1)$ using the following equations:

$$\begin{aligned}\Delta\omega_{km}(i+1) &= -\mu\frac{\partial E(i)}{\partial \omega_{km}(i)} + \alpha\,\Delta\omega_{km}(i) \\ &= \mu\sum_{p=1}^{P}\delta_{pm}(i)\overline{o}_{pk}(i) + \alpha\,\Delta\omega_{km}(i)\end{aligned} \tag{5}$$

$$\begin{aligned}\Delta\overline{\omega}_{nk}(i+1) &= -\mu\frac{\partial E(i)}{\partial \overline{\omega}_{nk}(i)} + \alpha\,\Delta\overline{\omega}_{nk}(i) \\ &= \mu\sum_{p=1}^{P}\overline{\delta}_{pk}(i)x_{pn} + \alpha\,\Delta\overline{\omega}_{nk}(i)\end{aligned} \tag{6}$$

where

$$\delta_{pm}(i) = (t_{pm} - o_{pm}(i))o_{pm}(i)(1 - o_{pm}(i)) \tag{7}$$

$$\overline{\delta}_{pk}(i) = \overline{o}_{pk}(i)(1 - \overline{o}_{pk}(i))\sum_{m=1}^{M}\delta_{pm}(i)\omega_{km}(i) . \tag{8}$$

Update the weights $\omega_{km}(i+1)$ and $\overline{\omega}_{nk}(i+1)$ for the next iteration by using the equations

$$\omega_{km}(i+1) = \omega_{km}(i) + \Delta\omega_{km}(i+1) \tag{9}$$

and

$$\overline{\omega}_{nk}(i+1) = \overline{\omega}_{nk}(i) + \Delta\overline{\omega}_{nk}(i+1) . \tag{10}$$

Set $i = i + 1$ and go to step 2 (Forward Pass).

B. MGFPROP

The back-propagated error signals $\delta_{pm}(i)$ and $\overline{\delta}_{pk}(i)$ include the factors $o_{pm}(i)(1 - o_{pm}(i))$ and $\overline{o}_{pk}(i)(1 - \overline{o}_{pk}(i))$ respectively (as shown in Equation (7) and (8)). The characteristic of these factors causes BP to have a slow convergence rate. When the actual output $o_{pm}(i)$ or $\overline{o}_{pk}(i)$ approaches extreme values (i.e. 0 or 1), the error signals will become very small so that the effect of the true error signal $t_{pm} - o_{pm}(i)$ is weak. Thus, if the learning process diverges, the magnitude of $o_{pm}(i)(1-o_{pm}(i))$ or $\bar{o}_{pk}(i)(1-\bar{o}_{pk}(i))$ becomes too small so that the output cannot be effectively adjusted by the error signals and the learning process becomes very slow or is even suppressed. This leads to premature saturation — the "flat spot" problem — which is a phenomenon in which the error remains almost unchanged for some periods of time during learning. Moreover, even when the output approaches the target output value, the convergence rate in the final stage is hindered by the factors $o_{pm}(i)(1-o_{pm}(i))$ and $\bar{o}_{pk}(i)(1-\bar{o}_{pk}(i))$, which results in the standard BP and some modified BP algorithms being trapped into a "flat spot" and may not converge to the global optimal solution for some applications.

To overcome this problem, MGFPROP (BP with Magnified Gradient Function) magnifies the factors $o_{pm}(i)(1-o_{pm}(i))$ and $\bar{o}_{pk}(i)(1-\bar{o}_{pk}(i))$ by using a power factor $1/S$ where S is larger than or equal to 1 ($S \geq 1$), i.e., to replace $o_{pm}(i)(1-o_{pm}(i))$ and $\bar{o}_{pk}(i)(1-\bar{o}_{pk}(i))$ by $[o_{pm}(i)(1-o_{pm}(i))]^{1/S}$ and $[\bar{o}_{pk}(i)(1-\bar{o}_{pk}(i))]^{1/S}$ respectively. When compared with the standard BP algorithm, the gradient term of MGFPROP has a larger increment when $o_{pm}(i)(1-o_{pm}(i))$ or $\bar{o}_{pk}(i)(1-\bar{o}_{pk}(i))$ approach zero and the effect of the true error signal $t_{pm} - o_{pm}(i)$ is stronger. Note that the magnified gradient function of MGFPROP does not violate the gradient-descent property of the standard BP algorithm. Since the gradient term is always positive, the modified gradient term can also be kept positive to retain the gradient descent properties of BP, and the rate of change of error $E(i)$ with respect to time remains negative, i.e., the error is always decreasing. Fig. 2 shows the MGFPROP algorithm.

Repeat

Execute normal BP forward pass operations

Update all weights and biases

$$\delta_{pm}^{(MGF)}(i) = (t_{pm} - o_{pm}(i))\left[o_{pm}(i)(1-o_{pm}(i))\right]^{1/S}$$

$$\bar{\delta}_{pk}^{(MGF)}(i) = \left[\bar{o}_{pk}(i)(1-\bar{o}_{pk}(i))\right]^{1/S}\sum_{m=1}^{M}\delta_{pm}^{(MGF)}(i)\omega_{km}(i)$$

$$\Delta\omega_{km}(i+1) = \mu\sum_{p=1}^{P}\delta_{pm}^{(MGF)}(i)\bar{o}_{pk}(i) + \alpha\,\Delta\omega_{km}(i)$$

$$\Delta\bar{\omega}_{nk}(i+1) = \mu\sum_{p=1}^{P}\bar{\delta}_{pk}^{(MGF)}(i)x_{pn} + \alpha\,\Delta\bar{\omega}_{nk}(i)$$

Until (converged)

Fig. 2. The MGFPROP algorithm.

The simulation results in [19] show that MGFPROP has a faster convergence rate and better global convergence capability when compared with the standard BP and some modified BP algorithms. In fact, the main advantage of this algorithm is that it does not have complicated additional operations. Thus, the computational complexity of the algorithm is almost the same as the standard BP algorithm and it is easy to implement in multi-layer feed-forward networks.

III. The Limitation of MGFPROP

MGFPROP is successful because it not only increases the convergence rate but also the global convergence capability with low implementation complexity [19]. However, when S is too large, the gradient function is over magnified so that the system error may be overshot. In this section, a number of experiments were conducted on two different adaptive problems — the exclusive OR (XOR) problem and the character recognition problem — to investigate the limitation of the performance of MGFPROP. The brief descriptions of the above problems are shown in Table II. Note that N, K, and M represent the number of input, hidden, and output nodes respectively. The input and the target patterns for these two problems consist of 0s and 1s only. All learning algorithms in this paper were terminated when the mean square error E reached 10^{-3} within 3×10^4 epochs, where

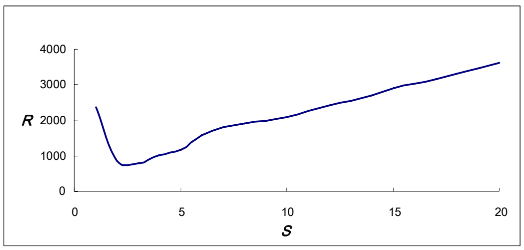
$$E = \sum_{p=1}^{P}\sum_{m=1}^{M}[t_{pm} - o_{pm}]^2 / PM\,. \qquad (11)$$

The initial weights were drawn at random from a uniform distribution between -0.3 and 0.3. Each experiment was performed 30 times for 30 different sets of initial weights. In the simulation results, R means the average number of epochs to converge (over 30 runs), which is inversely proportional to the convergence rate. % means the percentage of global convergence, which counts the percentage of runs (over 30 different runs) that successfully converged to the system error of less than 10^{-3}.

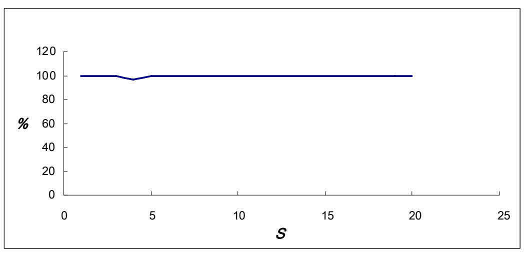
TABLE II
PROBLEM DESCRIPTIONS

Problem	Description	Network Architecture N-K-M	BP Parameter Setting (μ, α)
XOR	Give 2 binary inputs a and b, and output $a \oplus b$.	2 – 2 – 1	(0.5, 0.7)
Character Recognition	Recognize the inputs from 8 × 8 matrix to the lower case letter set [a, …, z].	64 – 20 – 26	(0.05, 0.7)

Fig. 3 and 4 show the effect of S on the performance of MGFPROP in the two different adaptive problems. In Fig. 3(a), when S increases from 1, R reduces quickly (i.e., the convergence rate increases quickly) because the gradient function is magnified gradually to speed up the learning process. However, if S is too large, R will increase (i.e., the convergence rate decreases) when S increases. This is because the effect of the magnification is too large so that MGFPROP overshoots the system error, and thus it takes more epochs to converge to the global optimal solution. In Fig. 3(b), the error overshooting problem almost does not affect the global convergence capability of MGFPROP, except the capability drops a little when S = 4. Fig. 4 shows that the effect of the error overshooting problem in the character recognition problem is more significant than that in the XOR problem. In Fig. 4(a), R reduces only when S increases from 1 to 2. When S increases further, R increases rapidly due to the effect of the error overshooting problem. Furthermore, in Fig. 4(b), the global convergence capability is highly affected by the value of S. When S increases from 2, the percentage of the global convergence drops very quickly and goes to 0 when S is larger than or equal to 7.

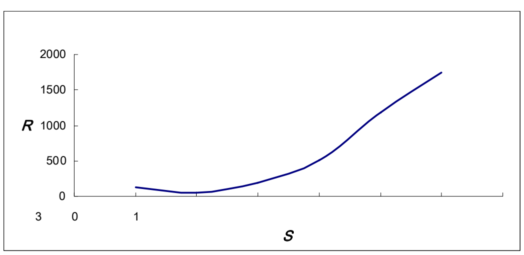


(a) R vs. S (*R is the number of epochs to converge*)

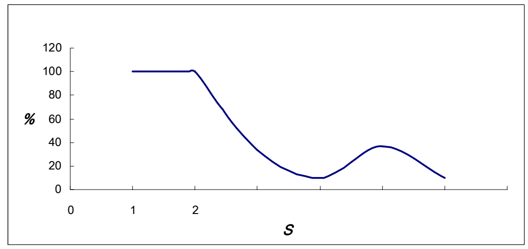


(b) % vs. S (*% is the percentage of global convergence*)

Fig. 3. The effect of S (the magnification factor) on the performance of MGFPROP in the XOR problem.



(a) R vs. S



(b) % vs. S

Fig. 4. The effect of S on the performance of MGFPROP in the character recognition problem.

IV. 2P-MGFPROP

This section presents a new approach called BP with Two-Phase Magnified Gradient Function (2P-MGFPROP) to overcome the error overshooting problem and hence speed up the convergence rate. 2P-MGFPROP is modified from MGFPROP. It divides the learning process into two phases: the convergence-searching phase (the first half-phase) and the convergence-guarantee phase (the second half-phase). In the convergence-searching phase, MGFPROP is far away from the minimum of the error surface. If the gradient function of MGFPROP is over magnified, it may overshoot the system error, and then the learning process may spend more epochs to converge to the global optimal solution and sometimes even diverge. Therefore, in this phase, S should be set to be sufficiently small to avoid the error overshooting problem and maintain the global convergence capability, although the improvement in the convergence rate is limited. In the convergence-guarantee phase, MGFPROP is close to the minimum of the error surface and it is highly likely to converge to the global optimal solution. Thus, the effect of the error overshooting problem is minimal and the global convergence capability can be maintained. Therefore, S should be set to be sufficiently large to magnify the gradient function significantly and hence speed up the learning process. Through the above arrangement, 2P-MGFPROP can speed up the overall convergence rate and, at the same time, avoid the error overshooting problem to maintain the global convergence capability.

To identify the region of two different phases, the maximum and minimum true error signals $t_{pm} - o_{pm}(i)$ are considered. Fig. 5 shows the relationship between the true error signals and the region of the two phases in the XOR problem. Note that the standard BP algorithm was used in this experiment and the figure shows the change of the maximum and minimum true error signals in one single run. The curves labeled “Max” and “Min” are the maximum and the minimum true error signals respectively. At the beginning, when the number of iterations is less than 400, the maximum and minimum error signals are almost unchanged and the learning process does not start to converge. When the number of iterations increases from 400 to 600, they move quickly towards zero which means the learning process starts to converge. When the number of iterations increases from 600, the learning process converges to the global optimal solution and the difference between the maximum and minimum true error signals becomes small. Based on the observation in Fig. 5, the boundary of the two phases can be found in the shaded region. Thus, a simple inequality with the measurement of these two terms is sufficient to identify the boundary of the two phases.

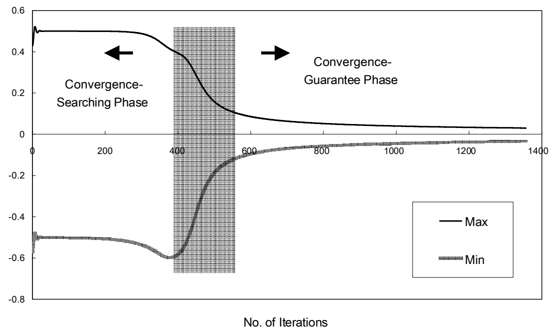


Fig. 5. The relationship between the true error signal and the two phases in the XOR problem.

The algorithm of 2P-MGFPROP is shown in Fig. 6. Note that *upper_err* and *lower_err* are the upper and lower bounds of the true error signals respectively. Additionally, *upper_S* and *lower_S* are the values of *S* used in the convergence-guarantee and convergence-searching processes respectively. Note that these four parameters may be different in different adaptive problems. The advantage of 2P-MGFPROP is not only that its performance is better than the optimal performance of MGFPROP in terms of the convergence rate (which will be shown later), but its implementation complexity is minimal. From Fig. 6, the additional implementation includes some simple comparisons and a simple inequality only.

Repeat
  Execute normal BP forward pass operations
  For all p
    Max = **maximum** ($t_{pm} - o_{pm}(i)$)
    Min = **minimum** ($t_{pm} - o_{pm}(i)$)
  End for loop
  if ((Max < *upper_err*) **and** (Min > *lower_err*))
    Set S **to** *upper_S* // convergence-guarantee process
  else
    Set S **to** *lower_S* // convergence-searching process
  Update all weights and biases by using MGFPROP
Until (converged)

Fig. 6. The 2P-MGFPROP algorithm.

The performance of 2P-MGFPROP was compared with the standard BP and MGFPROP algorithms in the XOR and character recognition problems. Table III shows the parameter settings of 2P-MGFPROP and Table IV presents the simulation results of the three algorithms. Note that the percentage of the global convergence of the three algorithms in the two adaptive problems is 100%.

In Table IV, the performance of MGFPROP is optimized in different adaptive problems. However, the performance of 2P-MGFPROP is still better in terms of the convergence rate. Additionally, it is found that the improvement of 2P-MGFPROP over MGFPROP in the character recognition problem is better than that in the XOR problem. This is because the number of epochs in the convergence-searching phase of the character recognition problem is smaller than that in the XOR problem and thus 2P-MGFPROP can speed up the learning process relatively earlier in the character recognition problem. For the character recognition problem, the improvement of 2P-MGFPROP is up to 50%. Note that the improvement is

$$\frac{C_{2P-MGFPROP} - C_{MGFPROP}}{C_{MGFPROP}}, \tag{11}$$

where $C_{2P\text{-}MGFPROP}$ and $C_{MGFPROP}$ are the converge rates of 2P-MGFPROP and MGFPROP respectively, and the converge rate is equal to the inverse of the number of epochs to converge, i.e., R.

TABLE III
PARAMETER SETTING OF 2P-MGFPORP

Problem	*upper_err*	*lower_err*	*upper_S*	*lower_S*
XOR	0.5	0.5	12	3
Character Recognition	0.7	0.7	3	2

TABLE IV
PERFORMANCE COMPARISONS

Problem	Standard BP	MGFPROP (optimum *S*)	2P-MGFPROP
XOR	2376.0	793.3 (S = 3)	743.6
Character Recognition	121.1	50.2 (S = 2)	33.0

V. CONCLUSION

This paper investigates the performance of MGFPROP by conducting two different adaptive problems, the XOR problem and the character recognition problem. It is found that the performance of MGFPROP is limited by the error overshooting problem. To overcome this problem, this paper proposes a new approach called Backpropagation with Two-Phase Magnified Gradient Function (2P-MGFPROP). 2P-MGFPROP is modified from MGFPROP. It divides the learning process into two different phases: the convergence-searching process and the convergence-guarantee process. In the different phases, different parameter settings of MGFPROP are assigned to maintain the global convergence capability and speed up the convergence rate. The identification of the phase of the learning process is based on the true error signals with a simple inequality. Thus, the implementation complexity of 2P-MGFPROP is minimal. The performance of 2P-MGFPROP in the above two adaptive problems outperforms MGFPROP and the standard BP algorithms in terms of the convergence rate, and the improvement is up to 50% compared with the optimal performance of MGFPROP.

ACKNOWLEDGMENT

The authors would like to thank Dr. R. G. Sharman for her proof-reading. The authors would also like to thank the anonymous reviewers for their valuable suggestions and comments.

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation", in *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, vol. 1. MIT Press, Cambridge, Mass, 1986.

[2] E. K. Blum, and L. K. Li, "Approximation theory and feedforward networks", *Neural Networks*, vol. 4, pp. 511 – 515, 1991.

[3] M. Gori, and A. Tesi, "On the problem of local minima in back-propagation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 76 – 86, 1992.

[4] Y. Lee, S. H. Oh, and M. W. Kim, "An Analysis of Premature Saturation in Back Propagation Learning", *Neural Networks*, vol. 6, pp. 719 – 728, 1993.

[5] F. Stager, and M. Agarwal, "Three methods to speed up the training of feedforward and feedback perceptrons", *Neural Networks*, vol. 10, no. 8, pp. 1435 – 1443, 1997.

[6] A. Van Ooyen, and B. Nienhuis, "Improving the convergence of the back-propagation algorithm", *Neural Networks*, vol. 5, pp. 465 – 471, 1992.

[7] J. E. Vitela, and J. Reifman, "Premature Saturation in Backpropagation Networks: Mechanism and Necessary Conditions", *Neural Networks*, Vol. 10, no. 4, pp. 721 – 735, 1997.

[8] M. Ahmad and F. M. A. Salam, "Supervised learning using the cauchy energy function", *International Conference on Fuzzy Logic and Neural Networks*, 1992.

[9] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation", *Neural Networks*, Vol. 1, pp. 295 – 307, 1988.

[10] C. C. Yu and B. D. Liu, "A Backpropagation Algorithm with Adaptive Learning Rate and Momentum Coefficient", *Proceedings of IJCNN'02*, vol. 2, pp. 1218 – 1223, 2002.

[11] X. H. Yu, G. A. Chen and S. X. Cheng, "Acceleration of backpropagation learning using optimised learning rate and momentum", *Electronics Letters*, Vol. 29, No. 14, pp. 1288-1289, 8th July 1993.

[12] S. E. Fahlman, "Fast learning variations on back-propagation: An empirical study", *Proceedings of the 1988 Connectionist Models Summer School* (Pittsburgh, 1988), D. Touretzky, G. Hinton, and T. Sejnowski, eds., pp. 38 – 51, Morgan Kaufmann, San Mateo, California, 1989.

[13] M. Riedmiller, and H. Braun, "A direct adaptive method for faster back-propagation learning: The RPROP Algorithm", *Proceedings of International Conference on Neural Networks*, vol. 1, pp. 586 – 591, 1993.

[14] N. K. Treadgold, and T. D. Gedeon, "Simulated Annealing and Weight Decay in Adaptive Learning: The SARPROP Algorithm", *IEEE Trans. on Neural Networks*, vol. 9, no. 4, pp. 662 – 668, July 1998.

[15] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies", *Journal of Statistical Physics*, vol. 34, pp. 975 – 986, 1984.

[16] S. Kirkpatrick, C. D. Gilatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science*, vol. 220, pp. 671 – 680, 1983.

[17] Y. Bard, *Non-Linear Parameter Estimation*, New York: Academic Press, 1974.

[18] S. Haykin, "Single-layer perceptrons", *Neural Networks: A Comprehensive Foundation*, 2nd ed., London: Prentice Hall, 1999, Ch.3, pp. 117 – 155.

[19] S. C. Ng, Chi-Chung Cheung and S. H. Leung, "Magnified Gradient Function with Deterministic Weight Evolution in Adaptive Learning", page 1411 – 1423, *IEEE Transactions in Neural Networks*, vol. 15, No. 6, November 2004.