Alternate Learning Algorithm on Multilayer Perceptrons

Bumghi Choi¹, Ju-Hong Lee^{2,*}, and Tae-Su Park³

¹ Dept. of Computer Science & Information Eng., Inha University, Korea
neural@inha.ac.kr

² School of Computer Science & Eng., Inha University, Korea
juhong@inha.ac.kr,

³ taesu@datamining.inha.ac.kr

Abstract. Multilayer perceptrons have been applied successfully to solve some difficult and diverse problems with the backpropagation learning algorithm. However, the algorithm is known to have slow and false convergence aroused from flat surface and local minima on the cost function. Many algorithms announced so far to accelerate convergence speed and avoid local minima appear to pay some trade-off for convergence speed and stability of convergence. Here, a new algorithm is proposed, which gives a novel learning strategy for avoiding local minima as well as providing relatively stable and fast convergence with low storage requirement. This is the alternate learning algorithm in which the upper connections, hidden-to-output, and the lower connections, input-to-hidden, alternately trained. This algorithm requires less computational time for learning than the backpropagation with momentum and is shown in a parity check problem to be relatively reliable on the overall performance.

1 Introduction

Backpropagation (BP) algorithm was developed by Rumelhart, Hinton, and Williams [1] as a learning algorithm for multilayered perceptrons. Though the BP algorithm has not yet been able to learn an arbitrary computational task in a network, it can solve many problems such as XOR, which the simple one-layer perceptrons can not solve. BP algorithm is described as follows in a two-layered network.

Our usual error measure or cost function is described as

$$E[w] = \frac{1}{2} \sum_{\mu i} \left[\zeta_i^\mu - g \left(\sum_j w_{ij}^2 g \left(\sum_k w_{jk}^1 \zeta_k^\mu \right) \right) \right]^2 \quad (1)$$

where ζ_i^μ is the ideal value of output unit i at pattern μ , g is an activation function, w_{ij}^2 is the weight from hidden unit j to output unit i , w_{jk}^1 is the weight from input unit k to hidden unit j .

For hidden-to-output connections the gradient-descent rule gives

$$\Delta w_{ij}^2 = -\eta \frac{\partial E}{\partial w_{ij}^2} = \eta \sum_{\mu} \left[\zeta_i^\mu - z(2)_i^\mu \right] g'(h(2)_i^\mu) z(1)_i^\mu \quad (2)$$

* Corresponding author.

From input-to-hidden connections,

$$\Delta w_{jk}^1 = -\eta \frac{\partial E}{\partial w_{jk}^1} = \eta \sum_{\mu i} [\zeta_i^\mu - z(2)_i^\mu] g'(h(2)_i^\mu) w_{ij}^2 g'(h(1)_j^\mu) \xi_k^\mu \quad (3)$$

BP is known to have convergence problems such as local minima or the plateau. The plateau causes the problem of very slow convergence. In the local minima, the gradient equals to zero. If the training process falls into a local minimum, the process of updating weight vectors stops.

Dynamic change of the learning rate and momentum[2,3,4] or the selection of a better function for activation or error evaluation followed by a new weight-updating rule have been proposed to avoid the problems. Quickpro[7], the resilient propagation(RPROP)[8], the genetic algorithm[5], the conjugate gradient[6], and the second-order method such as Newton's method[9,10] appear to pay some trade-off between the convergence speed and the stability of convergence avoiding the traps in wide range of parameters, or between the overall performance and storage requirement.

2 Alternate Learning Algorithm

In this section, a new learning strategy will be introduced, providing a conjecture why this can avoid local minima and plateaus. It just adopts the alternate learning strategy that combines the solvability of two-layered networks and the training power of the simple perceptrons. Based on that strategy, we develop a new learning algorithm. The new algorithm will be called *Alternate learning with the target values of hidden units*.

2.1 Analogy to Detour

In the proposed algorithm the learning process is separated into two components. First the upper connections are trained with the lower connections fixed. Whenever the training slows due to a local minima or plateau, the training is forcibly stopped. Then, the lower connections are trained with the upper connections fixed until the process meets the slow training time. This process is repeated until the error function reaches to the global minima.

At this point an analogy to traffic flow clarifies the philosophy of the learning technique. At a glance, BP algorithm and its variations seem to give the shortest way to the global minimum. But there are too many traffic jams like local minima or plateaus on the road of BP. When faced with a traffic jam the simple solution is to make a detour. The alternate learning method is based on this simple rule.

2.2 Approximation of the Target Values of Hidden Units

We need the target values of hidden units for the training of the lower connection. Here a target value of a hidden unit is intended to mean the value of a hidden unit which makes a selected output approximate to its ideal value as close as possible.

Though the exact values may not be figured out directly, the Newton-like approximation of the inverse function can be possible. We can draw the errors of hidden units from a selected output error by using the inverse function. To explain this process in terms of mathematical formulae, consider the expected error of output $z(2)_i$ from a point, $z(1)_j$, at the hidden units space

$$= \frac{(\gamma_i^\mu - z(2)_i^\mu)}{|\nabla z(2)_i^\mu(z(1)^\mu)|} \quad (4)$$

The component in the direction of $z(1)_j$ of $\nabla z(2)_i$

$$= \frac{\frac{\partial z(2)_i^\mu}{\partial z(1)_j^\mu}}{|\nabla z(2)_i^\mu(z(1)^\mu)|} = \frac{w_{ij}^2 g'(h(2)_i^\mu)}{|\nabla z(2)_i^\mu(z(1)^\mu)|} \quad (5)$$

By multiplying two factors, the expected error is

$$\gamma_j^\mu - z(1)_j^\mu = \frac{(\gamma_i^\mu - z(2)_i^\mu) w_{ij}^2}{g'(h(2)_i^\mu) |w_i^2|} \quad (6)$$

where γ_j is the target value, $z(1)_j$ is the current value of j 'th unit of the hidden layer, $w_i^w = (w_{i1}^2, w_{i2}^2, \dots, w_{in}^2)$, n is the dimension of hidden space.

The algorithm is summarized as follows

1. Train upper-connections with the ordinary gradient-descent rule until the process converges or meets a slow training point.
2. If the process converges then stop the program, otherwise propagate the error and target value of each hidden unit using formula (6)
3. Train the lower-connections from input units to the hidden units with produced target values of the hidden units until the training is slow or converges.
4. Go to 1.

3 Test and Evaluation

3.1 Test Environment

In order to verify the effectiveness of the alternate learning, 4-2-1 parity problem is used. For comparison, we used the BP with momentum 0.1 and 0.9.

3.2 Test Results

The convergence rate is defined as the inverse of the averaged convergence epochs. Fig. 1. shows the comparison of the stability between the alternate learning and the BP with momentum.

Fig. 1. The comparison of the alternate learning, the left, and BP with momentum, the right

Table 1. shows the experimental results of the three methods based on 100 runs randomly chosen from learning rate [0.1, 0.5], and initial weight [-1, 1] at 4-2-1 parity problem. The case of epochs over 1000 is treated as failed.

Table 1. Experimental results from 4-2-1 parity problem

Method	Average epochs	Minimum epochs	Success rate
BP with momentum 0.1	155	3	72%
Bp with momentum 0.9	150	2	54%
The alternate learning	78	3	100%

4 Conclusion

Thus far, the alternate learning algorithm on multilayer perceptrons have been derived, tested and compared with BP. From logical conjecture and the experimental results, the alternate learning algorithm is more stable in convergence than BP. Furthermore, since it is a kind of learning strategy, combined with existing methods announced as substitutes for BP, the additional effectiveness can be achieved.

Acknowledgments. This research was supported by the Ministry of Information and Communication, Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment.

References

1. Rumelhart, D.E., G.E. Hinton, and R.J. Williams.: Learning Internal Representations by Error propagation. In Parallel Distributed Processing, vol. 1, chap8 (1986)
2. Jacobs, R. A.: Increased Rates of Convergence Through Learning Rate Adaptation. Neural Networks 1 (1988) 293-280
3. Vogl, T. P., J.K. Magis, A.K. Rigler, W.T. Zink, and D.L. Alkon.: Accelerating the Convergence of the Back-Propagation Method. Biological Cybernetics 59 (1988) 257-263
4. Allred, L. G., Kelly, G. E.: Supervised learning techniques for backpropagation networks. In Proc. of IJCNN, vol. 1 (1990) 702-709

5. Montana D. J., Davis L.: Training feedforward neural networks using genetic algorithms, in Proc. Int. Joint Conf. Artificial Intelligence, Detroit (1989) 762-767
6. Moller, M. S.: A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks, vol. 6 (1993) 525-534
7. Fahlman, S. E.: Fast learning variations on backpropagation: An empirical study, in Proc. Connectionist Models Summer School (1989)
8. Riedmiller, M. and Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm, in Pro. Int. Conf. Neural Networks, vol. 1 (1993) 586-591
9. Ricoti, L. P., Ragazzini, S. and Martinelli, G.: Learning of word stress in a suboptimal second order back-propagation neural networks, in Proc. 1st Int. Conf. Neural Networks, vol. I (1988) 355-361
10. Watrous, R. L.: Learning algorithms for connectionist network: applied gradient methods of nonlinear optimization, in Proc. 1st. Int . Conf. Neural Networks, vol. II (1987) 619-628