MODIFICATION OF GRADIENT COMPUTATION IN THE BACK-PROPAGATION ALGORITHM

BOGDAN M. WILAMOWSKI AND LISA M. TORVIK
Department of Electrical Engineering
University of Wyoming

ABSTRACT:
The training of multilayer neural networks using the standard back-propagation algorithm suffers from slow asymptotic convergence. In this paper, a method of modifying the back-propagation algorithm to improve convergence is proposed using a new technique for gradient computation. Two cases were examined using this modification, an "*EXCLUSIVE OR*" network and a four input, four hidden layer, three output network. The simulations using the standard method and the modified method were compared. The results showed significant improvement in convergence using the proposed modification.

INTRODUCTION

The back-propagation algorithm is known for its slow convergence ratio. To gain faster convergence and improve the learning procedure, several different approaches were investigated. Instead of using standard back propagation, which is based on gradient descent minimization of the total error function, more advanced optimization techniques were used. These techniques include the quasi-Newton optimization algorithm (Bello, 1992), algorithms using information about the second derivative of the total error (Stefanos and Anastassiou, 1988) and advanced filtering techniques (Singhal and Wu, 1989). Learning methods based on the optimization of network architecture during learning process were also researched. In some cases, nonrelevant weights or units are subsequently removed (Mozer and Smolensky, 1989; Karnin, 1990). In other cases, the training procedure starts with a simple network and additional connections and nodes are incorporated to improve classification by an iteration process (Barmann and Biegler-Konig, 1992). Advanced perceptron-based learning rules were examined to aid in the improvement of the learning procedure (Gallant, 1990). Another approach is to modify the back propagation algorithm by minimizing a different error function instead of the standard quadratic error function (Van Ooten and Nienhuis, 1992; Krogh, Thorbergsson and Hertz, 1989). Also considered was the method of dynamic adaptation of learning parameters (Eaton and Olivier, 1992; Miniani and Williams, 1990) including the variation of individual neuron gains as a learning procedure (Sperduti and Starita, 1993).

Convergence of back-propagation is usually very slow when the network has a high neuron gain, or when the neuron states are very well defined and the net values are far from threshold values. In these cases, the gradients computed for the

back propagation algorithm are very small and errors have a limited ability to propagate through the network. Under such conditions, "flat-spots" are encountered and the output can be maximally wrong without producing a large error signal. As a consequence, the learning process and weight adjustment can be very slow. This is a major limitation of the back-propagation algorithm. Other methods were investigated to eliminate these "flat spots". The activation function was modified by adding an offset, which resulted in a significant increase in the speed of convergence (Van Ooten and Nienhuis, 1992). The speed of convergence was also accelerated when the inverse sigmoid function was used or a scaled linear approximation of the sigmoid function for error calculation was used (Parekh, Balakrishnan and Honavar, 1992) during learning procedure.

In this paper, a new technique was developed for improving the convergence of the learning process by using a new method of calculating the "gradient" of the activation function. The method used to calculate the gradient depends on the error at the neuron output. If the error is large, the effective gradient is also large, and when the error is small, the effective gradient corresponds to the gradient calculated using the traditional method.

THEORY OF MODIFICATION

The back-propagation algorithm commonly employed for training of multilayer neural networks suffers from a slow asymptotic convergence rate. For the sigmoidal bipolar activation function:

$$f(net) = \frac{2}{1 + \exp(-\lambda\, net)} - 1 \qquad (1)$$

the gradient (slope) is computed as a derivative of (1):

$$g_1 = \frac{2\exp(-\lambda\, net)}{(1 + \exp(-\lambda\, net))} = 0.5(1 - f^2) \qquad (2)$$

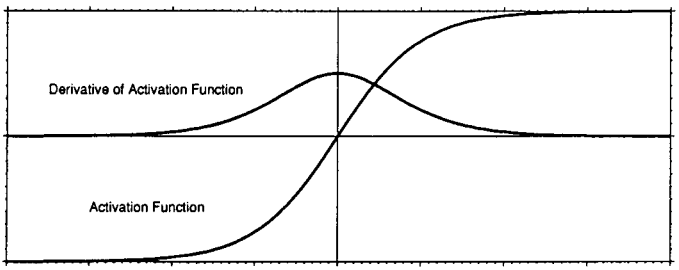

Figure 1: Standard sigmoidal activation function for bipolar neurons and its derivative.

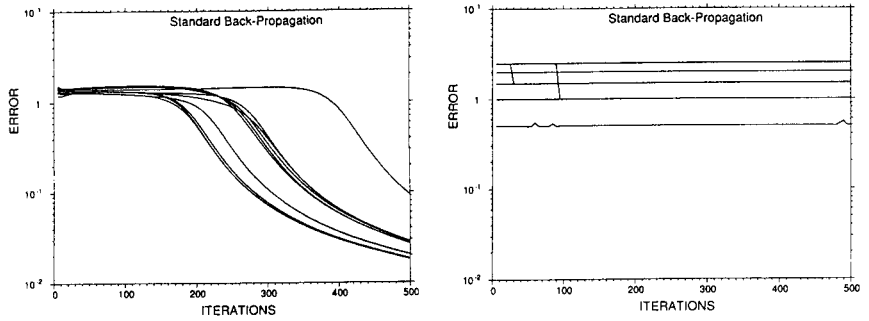

Figure 2: (a) Global error as a function of iterations for the *"Exclusive Or"* neural network using the standard back-propagation algorithm for ten randomly chosen initial weights with learning constant $\eta = 0.2$ and neuron gain $\lambda = 1.0$. (b) *"Exclusive Or"* network using the standard back-propagation algorithm with the same starting weights, but with $\lambda = 10$.

The activation function and it's gradient are illustrated in Figure 1. Typical cases for convergence using the "*EXCLUSIVE OR*" neural network with randomly chosen initial weights are shown in Figure 2a. For most cases, convergence was reached in less than 500 iterations. The neural network was then trained into saturation and the desired output were changed to their opposite values. Using these unfavorable weights as initial conditions, it is very difficult to recover the proper state during the learning procedure as shown in Figure 3a. In most cases, the standard back-propagation algorithm did not converge at all. Another example of poor convergence is shown in Figure 2b where the same set of initial weights were used as in Figure 2a, but the neuron gain was changed to $\lambda = 10$.

In the back-propagation algorithm, the weight changes are proportional to the error propagating from the output through the slopes of activation function and through the weights. This is a consequence of using the steepest gradient method for calculating the weight adjustments. Convergence of the learning process can be improved by changing how the error propagates back through the network.

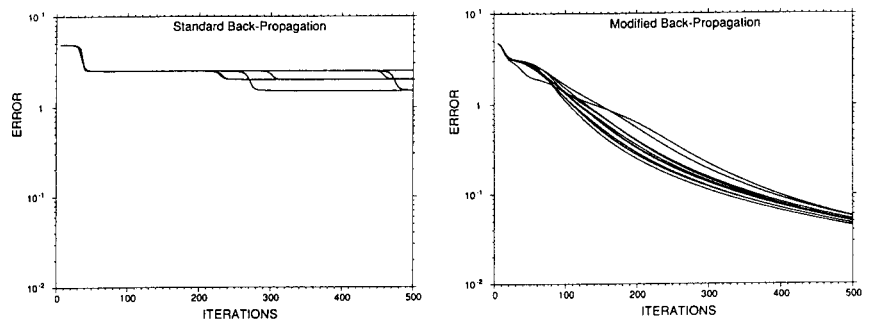

Figure 3: (a) Global error as a function of iterations for the *"Exclusive Or"* neural network using the standard back-propagation algorithm with unfavorable starting weights, $\eta = 0.2$ and $\lambda = 1.0$. (b) *"Exclusive Or"* network using the modified back-propagation algorithm with the same unfavorable starting weights.

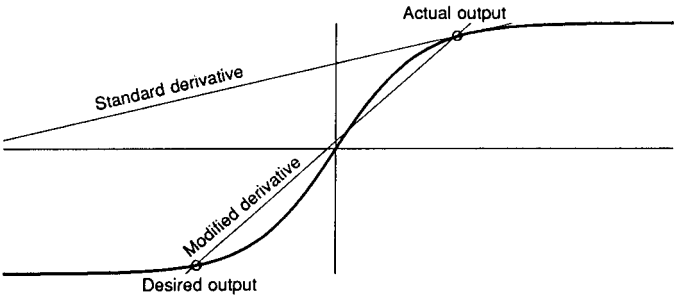

Figure 4: Illustration of the modified derivative computation using the slope of the line connecting the points of actual output and desired output.

It is proposed in this paper that for the purpose of error propagation, the slope (gradient) of the activation function is calculated as the slope of the line connecting the output value with the desired value, rather than the derivative of the activation function at the output value. This is illustrated in Figure 4.

Note that if the output value is close to the desired value, the calculated slope corresponds to the derivative of the activation function, and the algorithm is identical to the standard back-propagation formula. Therefore, the "derivative" is calculated differently only for large errors, when the classical approach significantly limits error propagation.

In a second example, a four input, four hidden layer and three output neuron was used. All possible 16 binary combinations of input patterns were classified into five randomly chosen categories as shown in Table 1. This network was trained with unfavorable initial weights using both the standard back-propagation and the modified back-propagation method.

INPUT	OUTPUT	CATEGORY
1 1 1 1 1 -1 1 1 1 -1 -1 1	1 -1 1	I
1 1 1 -1 1 -1 -1 -1 -1 1 -1 1	1 1 -1	II
1 1 -1 1 1 -1 1 -1 -1 1 1 1	1 1 1	III
-1 1 1 -1 -1 -1 1 1 -1 -1 1 -1	-1 1 1	IV
1 1 -1 -1 -1 1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1	-1 1 -1	V

Table 1: Training set for four input, four hidden neurons and three output neurons.

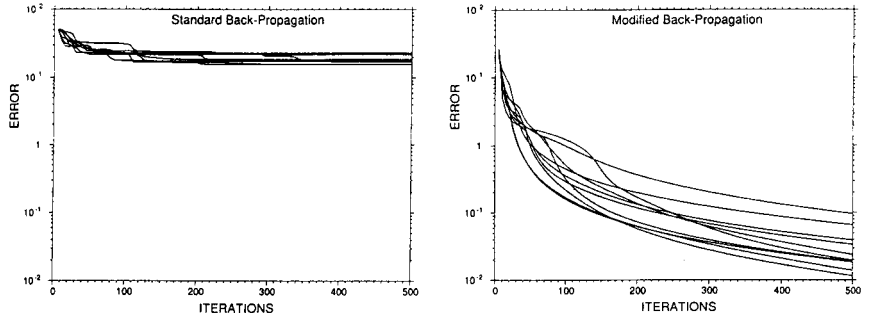

Figure 5: (a) Global error as a function of iterations for a four input, four hidden layer and three output network using the standard back-propagation algorithm and unfavorable starting weights, $\eta = 0.2$ and $\lambda = 1.0$. (b) Four input, four hidden layer and three output network using the modified back-propagation algorithm with the same unfavorable starting weights.

Figures 5a and 5b show results for both methods. Significant improvement was made when the modified method of gradient computation was used. Figures 6a and 6b illustrate results for the same network, but with $\lambda = 5$.

Similar results were obtained with various neural network structures such as binary number classifiers, parity bit finders and others. In most cases, significant improvement of convergence was noted, especially in networks with large neural gains and with unfavorable chosen starting weights.

CONCLUSION

Experiments using the three layer feedforward network (one hidden layer) have shown that substantial improvements can be obtained if the slopes in the last layer are computed using the modified algorithm. Improvement was especially significant when the output neurons were initially set to maximally wrong states (the system was initially trained for maximally wrong values). It is illustrated that the standard back-propagation algorithm does not converge at all while the modified back-propagation algorithm does converge and gives good results. Also, for efficient classification of patterns, the neural networks with high gain values (λ) have to be used. In such practical cases, the standard back-propagation algorithm has difficulty converging.

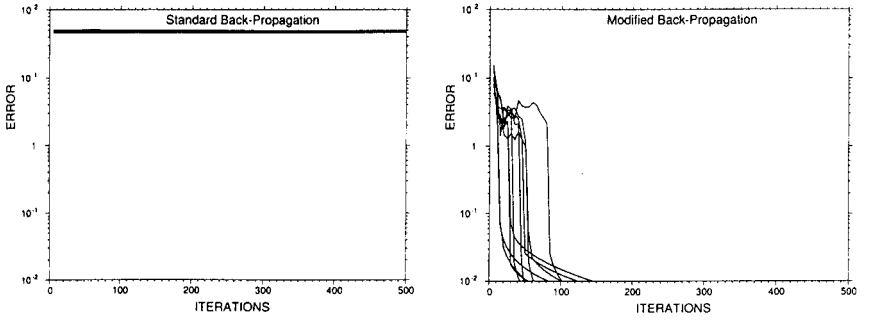

Figure 6: (a) Global error as a function of iterations for a four input, four hidden layer and three output network using the standard back-propagation algorithm with the same unfavorable starting weights as in Figure 5, but with $\lambda = 5.0$. (b) Four input, four hidden layer and three output network using the modified back-propagation algorithm with unfavorable starting weights and $\lambda = 5.0$.

It was also observed that in cases with small gains $\lambda \approx 1$, modification of the gradient computation did not improve the convergence. This is due to the fact that in cases where the output values are far from saturation, the slope computed using the modified method (Figure 4) is smaller than the slope computed as a simple derivative which causes the error to propagate slowly. This corresponds to the lowering of the effective learning constant.

REFERENCES

Barmann F., Biegler-Konig, F. (1992). On class of efficient learning algorithms for neural networks, *Neural Networks*, (5), 139-144.

Bello, M. G. (1992). Enhanced training algorithms, and integrated training/architecture selection for multilayer perceptron networks, *IEEE Trans. on Neural Networks*, (3), Nov, 864-875.

Eaton, H. A. C., Olivier, T. L. (1992). Learning coefficient dependence on training set size, *Neural Networks*, (5), 283-288.

Gallant, S. I. (1990) Perceptron-based learning algorithms, *IEEE Trans. on Neural Networks*, (1), June, 179-191.

Karnin, E. D. (1990). A simple procedure for pruning back-propagation trained neural networks, *IEEE Trans. on Neural Networks*, (1), 239-244.

Krogh, A., Thorbergsson, G. I., Hertz, J. A. (1989). A cost function for internal representations, In D. Touretzky (Eds.), *Advances in neural information processing systems II*, 733-740.

Miniani, A. A., Williams, R. D. (1990). Acceleration of back-propagation through learning rate and momentum adaptation, *Proceedings of International Joint Conference on Neural Networks*, (1), 676-679.

Mozer, M.C., Smolensky, P. (1989). Using relevance to reduce network size automatically, *Connection Science*, (1), 3-16.

Parekh, R., Balakrishnan, K., Honavar, V. (1992). An empirical comparison of flat-spot elimination techniques in back-propagation networks, *Proceedings of Third Workshop on Neural Networks - WNN'92*, 55-60.

Singhal, S., Wu, L. (1989). Training feed-forward networks with the extended Kalman filter, *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, 1187-1190.

Sperduti, A., Starita, A. (1993). Speed up learning and network optimization with extended back-propagation, *Neural Networks*, (6), 365-383.

Stefanos, K., Anastassiou, D. (1988). Adaptive training of multilayer neural networks using a least-squares estimation technique, *Proceedings of IEEE International Conference on Neural Networks*, (1), 383-390.

Van Ooten A., Nienhuis B. (1992). Improving the convergence of the back-propagation algorithm, *Neural Networks*, (5), 465-471.

ASME PRESS

INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS

VOLUME 3

Editors:
Cihan H. Dagli
Laura I. Burke
Benito R. Fernández
Joydeep Ghosh