A Differential Adaptive Learning Rate Method for Back-Propagation Neural Networks

Saeid Iranmanesh[(1)], M. Amin Mahdavi[(2)]
(1) MSc Candidate, Department of Computer Engineering,
Azad University of Qazvin, IRAN
Iranmanesh.s@gmail.com
(2) Ph.D., Lecturer, Department of Computer Engineering,
Imam Khomeini International University, Qazvin, IRAN
mahdavi@researchattic.ca

***Abstract*—**In this paper a high speed learning method using differential adaptive learning rate (DALRM) is proposed. Comparison of this method with other methods such as standard BP, Nguyen-Widrow weight Initialization and Optical BP shows that the network's learning speed has highly increased. Learning often takes a long time to converge and it may fall into local minimas. One way of escaping from local minima is to use a large learning rate at first and then to gradually reduce this learning rate. In this method which is used in multi-layer networks using back-propagation learning algorithm, network error is reduced in a short time using differential adaptive learning rate.

***Keywords*—**standard BP, optical-BP, adaptive learning rate, Nguyen-Widrow method, activation function.

I. INTRODUCTION

BACK-PROPAGATION was created by generalizing the Widrow-Hoff learning rule to multi-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in a predefined manner. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Standard back-propagation is a gradient descent algorithm -- as is the Widrow-Hoff learning rule -- in which the network weights are moved along the negative of the gradient of the performance function. The term back-propagation refers to the manner in which the gradient is computed for nonlinear multilayer networks. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods. The optical back-propagation (OBP) algorithm [1] is designed to overcome some of the problems associated with standard BP training using nonlinear function, which is applied on the output units. One of the important aspects of this algorithm is its ability to escape from local minima with high speed of convergence during the training period. A simple modification of random initialization, developed by Nguyen and Widrow [1990], is based on a geometrical analysis of the response of the hidden neurons to a single input; the analysis is extended to the case with several inputs by means of Fourier transformation. The analysis is based on the activation function.

II. BACK-PROPAGATION ALGORITHM

The Back-propagation (BP) learns a predefined set of output example pairs, after an input pattern has been applied as a stimulus to first layer of network units. The propagation occurs through each upper layer until an output is generated. This output pattern is then compared to the desired output, and an error signal is computed for each output unit. The signals are then transmitted backward from the output layer to each unit in the intermediate layer that contributes directly to the output. However, each unit in the intermediate layer receives only a portion of the total error signal. This process repeats, layer by layer, until each unit in the network has received an error signal that describes its relative contribution to the total error. An epoch is one cycle through the entire set of training vectors. Typically, many epochs are required for training a back-propagation neural net. The foregoing algorithm updates the weights after each training pattern is presented. An activation function for a back-propagation net should have several important characteristics: It should be continuous, differentiable, and monotonically non-decreasing.

III. OPTICAL BACK-PROPAGATION

The convergence speed of the learning process can be improved significantly by OBP through adjusting the error,

which will be transmitted backward from the output layer to each unit in the intermediate layers. In BP, the error at a single output unit is defined as:

$$\delta°pk = (Ypk - O\ pk) \quad (1)$$

Where the subscript "*P"* refers to the pth training vector, and *"K "*refers to the *kth* output unit. In this case, *Ypk* is the desired output value, and *Opk* is the actual output from *kth* unit, then *δpk* will propagate backward to update the output-layer weights and the hidden-layer weights. The error at a single output unit in adjusted *OBP* will be computed as follows*:*

$$New\,\delta°pk = (1+e^{(Ypk - O\ pk)^2})$$

$$, \text{if } (Ypk - O\ pk) >= zero$$

$$New\,\delta°pk = -(1+e^{(Ypk - O\ pk)^2}) \quad (2)$$

$$, \text{if } (Ypk - O\ pk) < zero$$

Where *Newδ°pk* is the new proposed function in the OBP algorithm.

An *OBP* uses two forms of *Newδ°pk,* because the exp-function always returns to zero or positive value (and the adapts operation for many output units need to decrease the actual outputs rather than increasing it). This *Newδ°pk* will minimize the errors of each output unit more quickly than the old δ°pk, and the weights on certain units experience very large changes from their initial values. There are no huge differences between the two algorithms in adapting the weights from the hidden layer to the output one. The surprising results are the number of epochs using the two algorithms, which proves that the *OBP* is an *optical algorithm* if it is compared with the standard BP. the results of *OBP* are much better and faster than the BP for simple training processes with different learning rates. The disadvantage of this method is that the error will never be close to zero therefore there will always be update weights and the convergence of network is dependent on input set because the more input variance, there is less weight change. For this algorithm we never have an error of zero and always have little change for weights. Hence, mean square error will not be zero.

IV. PROPOSED METHODOLOGY (DALRM)

As explained previously the problem with OBP algorithm is that it does not guarantee a convergence to absolute minimum error, despite improving the convergence speed in standard BP. This is because if the error is close to maximum, the OBP error grows gradually. This paper presents a computational technique to work out network error so that the learning rate can be computed. This is achieved by computing the error for each layer based on which the differential of the error for each layer may be determined. The result obtained is the learning rate for that layer. Note that if the activation function is linear, differentiation may not be used because for vertical and horizontal lines the differential is zero so there will be no weights update. In fact instead of changing the delta error, the learning rate for the whole layer for which the error was computed is changed. In each iteration, the learning rate is updated according to the error of the layers. The assumption is that each layer has a varying learning rate which needs to be updated in each iteration. In BP method, error is calculated as follows:

$$Error = target - output$$

While the error in adjusted (DALRM) will be as*:*

$$\alpha = \acute{f}(|target - output|) \quad (3)$$

Where *f* is an activation function that can be different in each layer

The new algorithm is as follows:

Input unit: $x_i = (i = 1,2,3,\ldots,n)$

Hidden unit: $z_j = (j = 1,2,3,\ldots,p)$

Step 1: initialize weights, for each training pair compute the network output (forward)

$$z\_in_j = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$

$$z_j = f(z\_in_j)$$

$$y\_in_k = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$$

$$y_k = f(y\_in_k)$$

Step 2: compute the error for each layer (backward)

$$\delta_k = (t_k - y_k)\,\acute{f}(y\_in_k)$$

Compute the learning rate for output layers:

$$\alpha_{out} = \acute{f}(|t_k - y_k|)$$

Compute the weight changes with new learning rate:

$$\Delta w_{jk} = \alpha_{out}\delta_k z_j$$

$$\Delta w_{ok} = \alpha_{out}\delta_k$$

$$\delta\_in_j = \sum_{k=1}^{m}\delta_k w_{jk}$$

$$\delta_j = \delta\_in_j \dot{f}(z\_in_j)$$

Compute the learning rate for hidden layers:

$$\alpha_{hid} = \dot{f}(1/h_{number}\sum_{j=1}^{p}\delta\_in_j)$$

Compute the weight changes with new learning rate:

$$\Delta v_{ij} = \alpha_{hid}\delta_j x_i$$

$$\Delta v_{oj} = \alpha_{hid}\delta_j$$

Update the weights:

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

Step 3: repeat step 1-3 whilst stopping condition is false

V. EXPERIMENTS

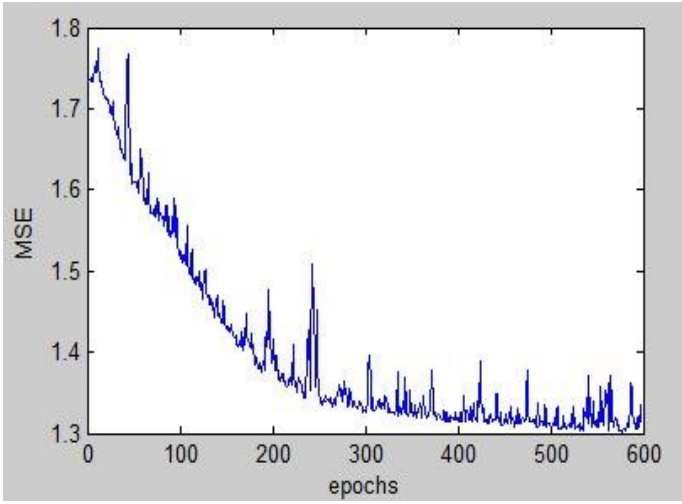

Fig. 1 OBP algorithm

The DALRM was tested using the following benchmark example that considers a neural network with 2 units for input layer, 3 units for hidden layer, and 1units for output layer. The activation function used for hidden layer is tan-hyperbolic and for output unit is linear function. The results shown in experiment used XOR function for training .The input vector represented in binary form. This example will train the network using DALRM, and then it trains it using standard BP, OBP and one of the improved BP that's written by Phil Brierley[3]. It then compares the final results obtained from DALRM, OBP, standard BP, and the improved BP.

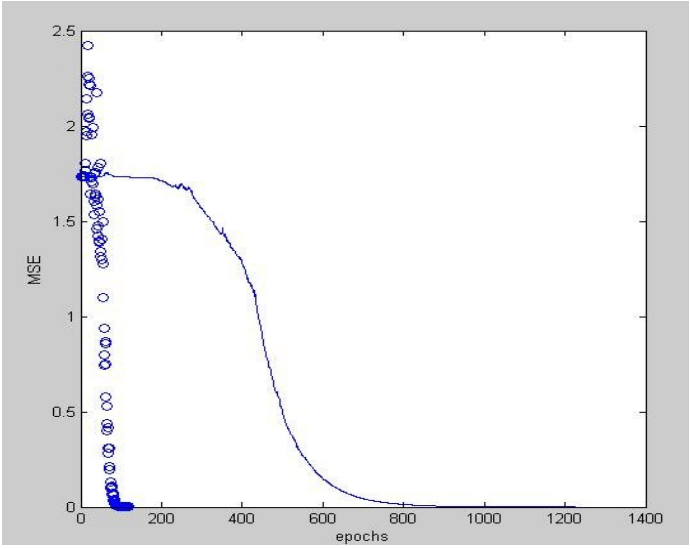

Fig .2 comparing DALRM and improved BP (BP --, DALRM O)

VI. CONCLUSION

The result shown in figure-1 demonstrated that OBP didn't converged to absolute minimum and at early epochs, minimization is very rapid but after many epochs the rate of minimization is reduced while the frequency of change in weights is variable. The result presented in figure-2 shows that the problem associated with the above method have been completely solved, where improved BP method and DALRM method have been compared. It is clearly obvious that the DALRM has a much faster convergence rate where it converge to zero in 115 epochs whilst the improved BP converged to zero in 1230 epochs.

Table-1 below shows the comparison between five methods where it considers the number of epochs of each method to converge to MSE<=0.05 with this parameters (2-4-1 BP net, binary input).

TABLE I COMPARISON OF METHODS FOR DECREASE OF MSE <= 0.05

Methods\parameters	Initial Learning rate	Epochs (mean)
Standard BP	0.02	2891
Nguyen-Widrow	0.02	1935
Improved BP	0.02	1193
OBP	0.02	412
DALRM	0.02	58

The minimum epochs for DALRM in XOR training for the above table was 39 epochs.From the above data and the previously presented figures it can be calculated that DALRM can escape from local minima and converge to absolute minimum. This method has also been tested on any other benchmark examples and has shown excellent results.

REFERENCES

[1] Mohammed A. Otair & Walid A. Salameh (2005), "Speeding Up Back-Propagation Neural Networks," Proceedings of the 2005 Informing Science and IT Education Joint Conference 167-173

[2] Fahlman, S. E. (1988), "Faster-learning variations on backpropagation: An empirical study," Proceedings of the 1988 connectionist Models Summer School, 38-51

[3] Phil brierley www.philbrierley.com

[4] Minai, A. A., & Williams, R. D (1990), "Acceleration of back-propagation through learning rate momentum adaptation," Proceedings of the International Joint Conference on Neural Networks, 1676-1679.

[5] Otair, M. A. & Salameh, W. A. (2004), "An improved back-propagation neural networks using a modified non-linear function," Proceedings of the IASTED International Conference, 2004, 442-447.

[6] M. Moreira & E. Fiesler (1995), "Neural networks with adaptive learning rate and momentum terms," IDIAP technical report

[7] Liang Jin and Madan M. Gupta, "Stable Dynamic Back-propagation Learning in Recurrent Neural Networks," IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 10, NO. 6, NOVEMBER 1999 1321-1334

[8] M. Giudici, F. Queirolo and M. Valle, "Evaluation of gradient descent learning algorithms with adaptive and local learning rate for recognising hand-written numerals," ESANN'2002 proceedings – European Symposium on Artificial Neural Networks Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 289-294