Dynamic Multi-optimal Learning Rates For Neural Network

[1]HAN-LEIH LIU, [2]SENG KAH PHOOI,
[1]School of Microelectronics, Griffith University, Kessels Rd, Nathan QLD 4111,AUSTRALIA
[2]School of Engineering, Monash University, Malaysia campus, 2 Jalan Kolej, Bandar Sunway, 46150 PJ, Selangor, MALAYSIA
kp_seng@yahoo.com, sikuo@lycos.com

Abstract: - This paper presents a method called dynamic multi-optimal learning rates for neural network (NN) with backpropagation (BP) training. The stability analysis of the learning rates for a 3-layer NN to minimize the total square error is included. The optimal learning rates can be obtained by using proper numerical method. These optimal learning rates are then applied to BP training to tune the corresponding weighting factors in each layer so that the total square error is minimized rapidly. A simulation example is performed for a nonlinear first-order system identification. The results have indicated that the training or convergence speed is faster compared with the standard BP with fixed learning rates.

Key-Words: Neural network, backpropagation, learning rate

1 Introduction

During the past decade, neural networks (NN) [1],[2] have used a variety of applications in various fields and tremendous achievements have been obtained. These applications include identification of an unknown system to control nonlinear ill-defined system [5] etc. The well-known training method called back propagation (BP) is mostly used to tune the weighting factors of NN. BP can be considered as gradient descent class algorithm that attempts to minimize the error between the desired and the NN outputs. The weights of the NN are adjusted so that the error is reduced along the descent direction.

Learning rate in BP is an important factor that affects the convergence speed and stability of the NN. Several authors [6], [7] have focused on the study of learning rate in BP. Authors in [1] have developed the dynamic optimal learning rates of a certain class of fuzzy neural networks (FNNs) [3],[4]. They have performed the stability analysis of the learning rate 2-layers NN by minimizing the total squared error between the actual and desired outputs for a set of training vectors. However, they only considered the dynamic optimal learning rate in 2-layers NN in certain class of FNNs.

In this paper, we investigate the dynamic optimal learning rates for a 3-layer NN. Simultaneous equations can be derived from the Lyapunov function. A pair of optimal learning rates can be obtained by solving the simultaneous equations. The stable and optimal learning rates can lead to maximum reduction of the total square error during the back propagation process. System identification for a nonlinear first order system is simulated to verify the theoretical result. For performance comparison, two simulations for the standard BP with fixed/varied learning rate are also accomplished.

2. Dynamic Optimal Learning Rates for A 3-layer Neural Network

Figure 1 shows the 3-layer NN which is considered in this paper.

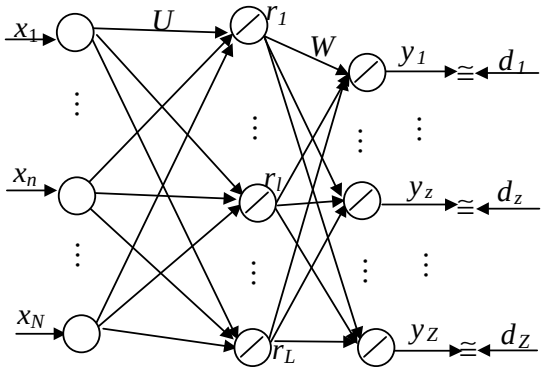

Fig. 1. Three-layer NN

$X = [\underline{x}_1 \quad \underline{x}_2 \quad \cdots \quad \underline{x}_P] \in \Re^{N \times P}$, input training matrix,

$\underline{x} = [x_1 \quad x_2 \quad \ldots \quad x_N]^T \in \Re^N$, input training vector;

$\underline{r} = [r_1 \quad r_2 \quad \ldots \quad r_L]^T \in \Re^L$, output of hidden layer,

$U_{NL} = [\underline{u}_1 \quad \underline{u}_2 \quad \cdots \quad \underline{u}_L] \in \Re^{N \times L}$, weighting matrix,

$\underline{u}_i = [u_{1i} \quad u_{2i} \quad \ldots \quad u_{Ni}]^T \in \Re^N$, i^{th} weighting vector,

$W_{LZ} = [\underline{w}_1 \quad \underline{w}_2 \quad \cdots \quad \underline{w}_Z] \in \Re^{L \times Z}$, weighting matrix,

$\underline{w}_i = [w_{1i} \quad w_{2i} \quad \ldots \quad w_{Li}]^T \in \Re^L$, i^{th} weighting vector,

$Y = [\underline{y}_1 \quad \underline{y}_2 \quad \cdots \quad \underline{y}_P]^T \in \Re^{P \times Z}$, actual output matrix,

$\underline{y} = [y_1 \quad y_2 \quad \dots \quad y_Z]^T \in \Re^Z$, actual output vector,

$D = [\underline{d}_1 \quad \underline{d}_2 \quad \cdots \quad \underline{d}_P]^T \in \Re^{P \times Z}$, desired output matrix.

$\underline{d} = [d_1 \quad d_2 \quad \dots \quad d_Z]^T \in \Re^Z$, desired output vector

"T" denotes matrix transpose. The symbol "/" in Fig. 1 denotes that the transfer function of each neuron is pure linear.

Given P training vectors, there should be P desired output vectors. The actual output matrix Y can be generated as:

$$Y = X^T U W \tag{1}$$

The weighting matrixes U and W are trained so that the actual output y_z will converge to its corresponding desired value d_z. The error function is defined as

$$E = Y - D = X^T U W - D \tag{2}$$

And the total squared error J can be computed

$$J = \frac{1}{2PZ} Tr(EE^T). \tag{3}$$

To update U and W, we employ the famous back propagation method as follows:

$$\begin{aligned} W^{t+1} &= W^t - \beta_t \frac{\partial J}{\partial W^t} = W^t - \beta_t \frac{1}{PZ} U^T X E \\ U^{t+1} &= U^t - \alpha_t \frac{\partial J}{\partial U^t} = U^t - \alpha_t \frac{1}{PZ} X E W^T \end{aligned} \tag{4}$$

where t denotes the t^{th} iteration. If the zero error is obtained after learning, we can have $D = X^T U W$. From Eq.(4), we notice that the two learning rates α_t β_t are changed for each iteration. The following shows how the optimal learning rates α_t, β_t are generated.

A candidate Lyapunov function is defined as

$$V = J^2 \tag{5}$$

The difference of Lyapunov function is $\Delta V = J_{t+1}^2 - J_t^2$. If $\Delta V<0$, the response of the system is to be stable and we have

$$J_{t+1} - J_t < 0. \tag{6}$$

After some manipulations, $J_{t+1} - J_t$ is a function of variables α, β can be expressed as

$$\begin{aligned} F(\alpha, \beta) = J_{t+1} - J_t = A\alpha^2\beta^2 + B\alpha\beta^2 + C\beta^2 \\ + I\alpha^2 + E\alpha\beta + F\alpha^2\beta + G\beta + H\alpha \end{aligned} \tag{7}$$

The coefficient A~C, I, E~H, are shown as below:

$$\begin{aligned} A &= \frac{1}{2}(PZ)^{-5} Tr(X_{PM}^T X_{MP} E_{PZ} W_{ZL}^T U_{LM}^T X_{MP} \\ &\quad E_{PZ} E_{ZP}^T X_{PM}^T U_{ML} W_{LZ} E_{ZP}^T X_{PM}^T X_{MP}) > 0 \\ B &= -(PZ)^{-4} Tr(X_{PM}^T U_{ML} U_{LM}^T X_{MP} \\ &\quad E_{PZ} E_{ZP}^T X_{PM}^T U_{ML} W_{LZ} E_{ZP}^T X_{PM}^T X_{MP}) \\ C &= \frac{1}{2}(PZ)^{-3} Tr(X_{PM}^T U_{ML} U_{LM}^T X_{MP} \\ &\quad E_{PZ} E_{ZP}^T X_{PM}^T U_{ML} U_{LM}^T X_{MP}) > 0 \\ I &= \frac{1}{2}(PZ)^{-3} Tr(X_{PM}^T X_{MP} E_{PZ} W_{ZL}^T W_{LZ} W_{ZL}^T W_{LZ} \\ &\quad E_{ZP}^T X_{PM}^T X_{MP}) > 0 \\ E &= 2(PZ)^{-3} Tr(X_{PM}^T U_{ML} W_{LZ} E_{ZP}^T \\ &\quad X_{PM}^T U_{ML} W_{LZ} E_{ZP}^T X_{PM}^T X_{MP}) \\ &\quad -(PZ)^{-3} Tr(X_{PM}^T X_{MP} E_{PZ} W_{ZL}^T U_{LM}^T X_{MP} E_{PZ} D_{ZP}^T) \\ F &= -(PZ)^{-4} Tr(X_{PM}^T X_{MP} E_{PZ} W_{ZL}^T W_{LZ} \\ &\quad E_{ZP}^T X_{PM}^T U_{ML} W_{LZ} E_{ZP}^T X_{PM}^T X_{MP}) \\ G &= (PZ)^{-2} Tr(-X_{PM}^T U_{ML} W_{LZ} E_{ZP}^T X_{PM}^T U_{ML} U_{LM}^T X_{MP} \\ &\quad + X_{PM}^T U_{ML} U_{LM}^T X_{MP} E_{PZ} D_{ZP}^T) \\ H &= (PZ)^{-2} Tr(-X_{PM}^T U_{ML} W_{LZ} W_{ZL}^T W_{LZ} E_{ZP}^T X_{PM}^T X_{MP} \\ &\quad + X_{PM}^T X_{MP} E_{PZ} W_{ZL}^T W_{LZ} D_{ZP}^T) \end{aligned}$$

To get a critical point $(\alpha,\beta) \in R \times R$ of F, the following simultaneous equations have to be solved

$$\begin{aligned} \frac{\partial F}{\partial \alpha} &= 2A\alpha\beta^2 + B\beta^2 + 2I\alpha + E\beta + 2F\alpha\beta + H = 0 \\ \frac{\partial F}{\partial \beta} &= 2A\alpha^2\beta + 2B\alpha\beta + 2C\beta + E\alpha + F\alpha^2 + G = 0 \end{aligned} \tag{8}$$

Furthermore, we can find minimum value of F by computing

$$\begin{aligned} \Theta &= \frac{\partial^2 F}{\partial\alpha\partial\beta} = \frac{\partial^2 F}{\partial\beta\partial\alpha} = 4A\alpha\beta + 2B\beta + E + 2F\alpha \\ \Phi &= \frac{\partial^2 F}{\partial\alpha\partial\alpha} = 2A\beta^2 + 2F\beta + 2I; \\ \Gamma &= \frac{\partial^2 F}{\partial\beta\partial\beta} = 2A\alpha^2 + 2B\alpha + 2C \end{aligned} \tag{9}$$

Note that if $\Theta^2 < \Gamma\Phi$ and $\Phi>0$, then F has a minimum value at (α,β). Otherwise the fixed values (<1) of α, β are assumed.

3 Simulation Examples

In this example [2], [7], the system to be identified is the following nonlinear form:

$$y(k+1) = g[y(k), u(k)]$$

where the unknown function g has the following form

$$g(x_1, x_2) = \frac{x_1}{1 + x_1^2} + x_2^3$$

and controller $u(k)=\sin(2\pi k/25)+\sin(2\pi k/10)$. The series-parallel identification model is

$$\hat{y}(k+1)=\hat{f}[y(k),u(k)] \qquad (10)$$

where $\hat{f}$ is in the form of (10) with two fuzzy variables x_1, and x_2.

90 training data items are generated for training purposes. The initial values of weighting matrixes $U_{2\times 8}$ and $W_{8\times 1}$ are randomly chosen from ranges [-0.05 0.05], [-0.005 0.005], respectively. Fig. 2 shows the performance comparison of the proposed method and the standard BP. The result of the proposed method indicates that the total squared error convergences after 7 iterations. The final total squared error J, ≈ 0.5359 is achieved. The total squared errors of conventional BP methods converge after 45 iterations. Fig. 3 shows the outputs of plant and the NN model after the training program is completed. Better results can be obtained if more hidden and output nodes are used.

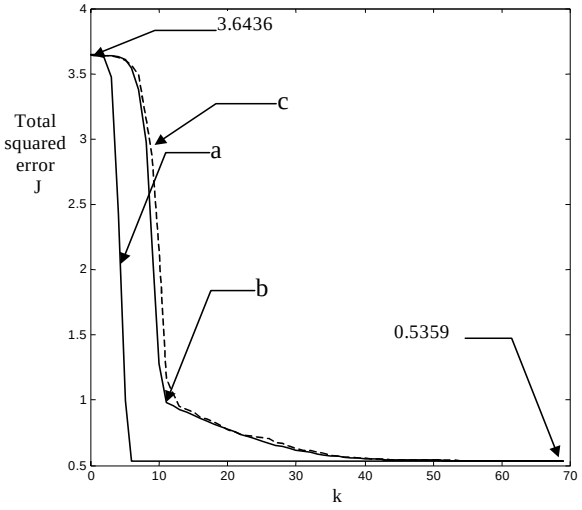

Fig. 2. Performance comparison. Case *a*: Optimal α, β; Case *b*: α=0.1, β=0.1; Case *c*: random α, β ∈(0, 0.2).

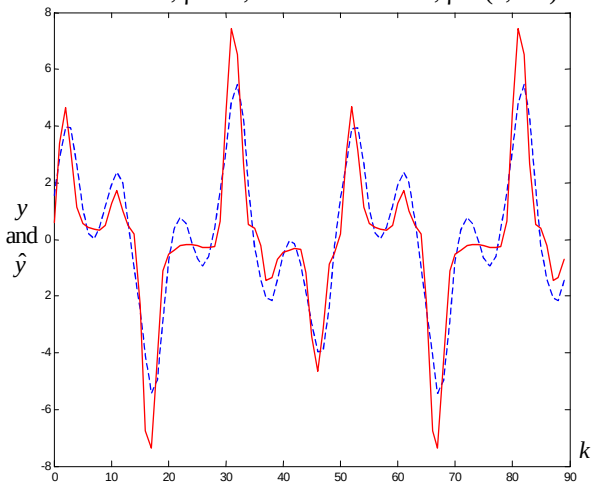

Fig. 3. Outputs of the plant *y* (solid) and $\hat{y}$ (dashed).

4 Conclusion

A new approach of training NN with dynamic optimal learning rates BP is presented systemically in this paper. The stability analysis of these optimal learning rates is presented. By solving a set of simultaneous equations that are derived based on Lyapunov theory, the optimal learning rates can be computed. They are then fed into BP algorithm such that the minimum total square error can be achieved. The proposed method can speed up the total error convergence rate. Simulation examples for the nonlinear first order system identification are performed to verify the theoretical analyses. The results have revealed that our approach provides faster total squared error convergence.

References:

[1] B. Kosko, *Neural Network and Fuzzy Systems*, Prentice Hall, Eaglewood Cliffs, NJ. 1992.

[2] K. S. Narendra, and K. Parthasarathy, "Identification and Control of Dynamical Systems using Neural Networks", *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4-26, Mar. 1990.

[3] S. Horikawa, T. Furuhasi and Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neural Networks with Back-Propagation Algorithm", *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 801-806, Sept. 1992.

[4] C. T. Lin and Y. C. Lu, "A Neural Fuzzy System with Fuzzy Supervised Learning", *IEEE Trans. Syst. Man Cyber.*, Vol. 26, No. 5, pp. 744-763 ,Oct. 1996.

[5] Suleiman Barada and Harpreet Singh, "Generating Optimal Adaptive Fuzzy-Neural Models of Dynamical Systems with Applications to Control", *IEEE Trans. on Syst. Man Cyber.*, Vol. 28, No. 3, pp. 371-390 , Aug. 1998.

[6] X. H. Yu, G. A. Chen, S. X. Cheng, "Dynamic Learning Rate Optimization of the Backpropagation Algorithm", *IEEE Trans. On Neural Networks*, Vol. 6, No. 3, pp. 669- 677, May 1995.

[7] C. H. Wang, H. L. Liu, and C. T. Lin, "Dynamic Optimal Learning Rates of a Certain Class of Fuzzy Neural Networks and its Applications with Genetic Algorithm", *IEEE Trans. on Syst. Man Cyber*. Part B, Vol. 31, No. 3, pp. 467-475, June. 2001.