New Classes of Learning Automata Based Scheme for Adaptation of Backpropagation Algorithm Parameters

M. R. Meybodi Hamid Beigy
Computer Engineering Department
Amirkabir University of Technology
Tehran, Iran
Email : Beigy@ce.aku.ac.ir

ABSTRACT: One popular learning algorithm for feedforward neural networks is the backpropagation (BP) algorithm which includes parameters, learning rate (η), momentum factor (α) and steepness parameter (λ). The appropriate selections of these parameters have large effect on the convergence of the algorithm. Many techniques that adaptively adjust these parameters have been developed to increase speed of convergence. In this paper, we shall present several classes of learning automata (LA) based solutions to the problem of adaptation of BP algorithm parameters. By interconnection of LA to the feedforward neural networks, we use LA base scheme for adjusting the parameters η, α, and λ based on the observation of random response of the neural networks. One of the important aspects of proposed schemes is its ability to escape from local minima with high possibility during the training period. The feasibility of proposed methods are shown through the simulations on several problems.

I. INTRODUCTION

BP algorithm is a systematic method for training multilayer neural networks [Rumelhart 1986]. Despite the many successful applications of BP, it has many drawbacks. For complex problems it may require a long time to train the networks, and it may not train at all. Long training time can be the result of the non-optimum values for the parameters of the training algorithm. It is not easy to choose appropriate values for these parameters for a particular problem. The parameters are usually determined by trial and error and using the past experiences. For example, if the learning rate is too small, convergence can be very slow, if too large, paralysis and continuos instability can result. Moreover the best value at the beginning of training may not be so good later. Thus several researches have suggested algorithms for automatically adjusting the parameters of training algorithm as training proceeds.

Arabshahi et al. [Arabshahi 1992] proposed an BP algorithm in which the learning rate is adapted. They proposed that the learning rate to be adjusted using a fuzzy logic control system. Kandil et al. [Kandil 1993] used optimum, time-varying learning rate for multi-layer neural network by linearizing the neural network around weight vector at each iteration. Parlos et al. [Parlos 1994] proposed an accelerated learning algorithm for supervised training of multi-layer neural networks named adaptive BP algorithm. In their proposed algorithm the learning rate is a function of the error and the error gradient. Darken and Moody [Sarkar 1995], Solmon [Sarkar 1995], Fallside and Chan [Sarkar 1995] has proposed other schemes for adaptation of learning rate. Sperduti and Starita [Sperduti 1993] proposed an BP algorithm in which the steepness parameter is adapted using gradient descent algorithm. Several LA based procedures have been also developed [Menhaj 1994][Menhaj 1996][Menhaj 1996][Menhaj 1997][Beigy 1997][Beigy 1998]. In these methods variable structure learning automata (VSLA) or fixed structure learning automata (FSLA) have been used to find the appropriate values of parameters for the BP algorithm. In these schemes either a separate LA is associated to each layer of the network or a single LA is associated to the whole network to adapt the appropriate parameters. It is shown that the learning rate adapted in such a way not only increases the rate of convergence of the network but it bypasses the local minimum in most cases.

In this paper, we propose two new classes of LA based schemes for adaptation of appropriate learning rate or steepness parameters for BP algorithms. Unlike the existing LA based schemes, in these schemes one LA is assigned to every link or every neuron in the network for determining the parameters for that link or neuron. The simulation results show the feasibility of the proposed method and its superiority to the existing LA based schemes. The proposed schemes have two important aspects: higher speed of convergence and a high probability of escaping from the local minima. In order to evaluate the performance of proposed schemes simulations are carried out on four learning problems: digit recognition [Sperduti 1993], encoding [Rumelhart 1986], odd parity[Rumelhart 1986], and symmetry problems [Rumelhart 1986] and the results are compared with results obtained from standard BP. These problems are chosen because they posses

different error surfaces and collectively present an environment that is suitable to determine the effect of proposed method.

The rest of the paper is organized as follows: Section II briefly presents the basic BP algorithm and LA. Application of LA for adaptation of learning rate, momentum factor, and steepness parameter is given in section III. Section IV presents the proposed LA based schemes. The simulation results are given in section V. Section VI concludes the paper.

II. BACKPROPAGATION ALGORITHM AND LEARNING AUTOMATA

Backpropagation Algorithm: BP training algorithm which is an iterative gradient descent algorithm is a simple way to train multilayer feedforward neural networks [Rumelhart 1986]. The BP algorithm is based on the gradient descent rule:

$$\Delta W(n) = \eta G(n) + \alpha \Delta W(n-1) \qquad (1)$$

where W is the weight vector, n is the iteration number, η is learning rate, α is momentum factor, and G is gradient of error function which is given by $G(n) = -\nabla E(n)$, where E is the sum of squared error given by:

$$E(n) = \frac{1}{2} \sum_{p=1}^{\#patterns} \sum_{j=1}^{\#outputs} [T_{p,j} - O_{p,j}]^2 \qquad (2)$$

Where T_{pj} and O_{pJ} are desired and actual outputs for pattern p at output node j. One of the major problems encountered during implementation of the BP learning rule is proper choice and update of the learning rate η to allow convergence, while keeping the number required iterations at a reasonable number. One of main reasons for investigating the possibility of the adaptive learning rate rule is the desire to reduce the sensitivity of the learning on the learning rate, without adding more tuning parameters.

The activation function of every unit is normally a sigmoid function chosen between 1/1+exp(-λnet) and tanh (λnet). The coefficient of the exponent of the exponential term determines the steep of linearity of that function and is often set to a constant value. We gain much flexibility, if we move the net inputs of the sigmoidal functions near to their active regions, where the associated gradient are not very close to zero. This makes the BP algorithm not be trapped to some points in the network parameters space where the BP algorithm would effectively stop, even though it is not close to a local minima point. This will cause the gradient of the error function to be small if the sigmoidal is shifted far outside the active region of the input to the input of function. Therefore, it is better to center each sigmoid to be inside the active region of the sigmoidal function.

Learning Automata: LA can be classified into two main families, fixed and variable structure learning automata [Narendra 1989][Meybodi 1982][Meybodi 1984][Meybodi 1987]. Examples of the FSLA type that we used in this paper are Tsetline, Krinsky, TsetlineG, and Krylov automata. An FSLA is quintuple $\langle \underline{\alpha}, \underline{\phi}, \underline{\beta}, F, G \rangle$ where:

1) $\alpha = (\alpha_1, \ldots, \alpha_R)$ is the set of actions that it must choose from.
2) $\Phi = (\Phi_1, \ldots, \Phi_s)$ is the set of states.
3) $\beta = \{0, 1\}$ is the set of inputs where "1" represents a penalty and "0" a reward.
4) $F : \Phi \times \beta \rightarrow \Phi$ is a map called the transition map. It defines the transition of the state of the LA on receiving an input.
5) $G : \Phi \rightarrow \alpha$ is the output map and determines the action taken by the automaton if it is in state Φ_j.

The selected action serves as the input to the environment which in turn emits a stochastic response $\beta(n)$ at the time n. $\beta(n)$ is an element of $\beta=\{0, 1\}$and is the feedback response of the environment to the LA. The environment penalize (i.e., $\beta(n)=1$) the LA with the penalty c_i, which is the action dependent. On the basis of the response $\beta(n)$, the state of the LA $\Phi(n)$ is updated and a new action chosen at the time (n+1). Note that the $\{c_i\}$ are unknown initially and it is desired that as a result of interaction with the environment the LA arrives at the action which presents it with the minimum penalty response in an expected sense. If the probability of the transition from one state to another state and probabilities of correspondence of action and state are fixed, the automaton is said FSLA and otherwise the automaton is said VSLA.

III. LA BASED SCHEMES FOR ADAPTATION OF BP PARAMETERS

In this section, we first, briefly describe previous LA based schemes [Menhaj 1994][Menhaj 1996][Menhaj 1996][Menhaj 1997][Beigy 1997][Beigy 1998] for adaptation of BP parameters and then introduce two new classes of LA based schemes. In all of the existing schemes, one or more LAs have been associated to the network. The LA (or

LAs) based on the observation of the random response of the neural network, adapt one or more of BP parameters. The interconnection of LA and neural network is shown in figure 1. Note that the neural network is the environment for the LA. The LA according to the amount of the error received from neural network adjusts the parameters of the BP algorithm. The actions of the LA correspond to the values of the parameters being calculated and input to the LA is some function of the error in the output of neural network. A function of error between the desired and actual outputs of network is considered as the response of environment. A window on the past values of the errors are swiped and the average value of the error in this window computed. If the difference of the average value in the two last steps is less than the predefined threshold value, the response of the environment is favorable and if this difference of average value is greater than the threshold value, the response of the environment is unfavorable. Existing LA based procedures for adaptation of BP parameters can be classified into two groups which we call them group ***A*** and group ***B***. In group ***A*** schemes, an LA is used for the whole network [Menhaj 1994][Menhaj 1996] whereas in group ***B*** schemes, separate LA one for each layer (hidden and output) are used [Menhaj 1996][Menhaj 1997][Beigy 1997][Beigy 1998]. Each group ***A*** and ***B*** depending on the type of automata used (fixed or variable structure) can be classified into two subgroups. For the sake of convenience in presentation, we use the following naming conventions to refer to different LA based schemes in class ***A*** and class ***B***.

Automata-AV(γ): A scheme in class ***A*** for adjusting parameter γ which uses VSLA **Automata**.

Automata-AF(γ): A scheme in class ***A*** for adjusting parameter γ which uses FSLA **Automata**.

Automata$_1$-Automata$_2$-BV(γ): A scheme in class ***B*** for adjusting parameter γ which uses VSLA **Automata$_1$** for hidden layer and VSLA **Automata$_2$** for output layer.

Automata$_1$-Automata$_2$-BF(γ): A scheme in class ***B*** for adjusting parameter γ which uses FSLA **Automata$_1$** for hidden layer and FSLA **Automata$_2$** for output layer.

The letters **F** and **V** in above names denotes FSLA and VSLA , respectively. For all the LA based schemes reported, it is shown through simulation that the use of LA for adaptation of BP learning algorithm parameters increases the rate of convergence by a large amount. Figure 2 borrowed from [Beigy 1998], compares the effectiveness of different LA based schemes in class ***A*** for adaptation of learning rate for the digit recognition problem. In this simulation the threshold of 0.01 and window size of 1 is chosen. For linear reward-penalty automata the reward and penalty coefficient 0.001 and 0.0001 are chosen. It is reported that FSLA based schemes have performance much higher than the VSLA based schemes [Meybodi 1998]. Also simulation studies have shown that by using LA based scheme for adaptation of learning rate or momentum factor we can compute a new point that is closer to the optimum than the point computed by BP algorithm which uses a fixed predetermined learning rate or momentum factor [Beigy 1997][Menhaj 1997] .

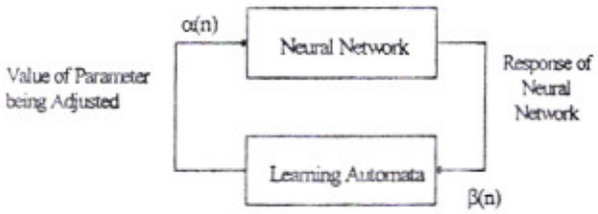

Figure 1: The interconnection of learning automata and neural network

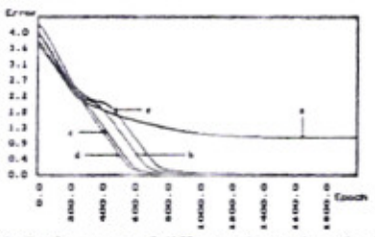

Figure 2: Performance of different class A based schemes
a: Standard BP b: Tsetline(4,4)-AF (η) c: Krinsky(2,4)-AF (η)
d: Krylov(2,4)-AF (η) e: L_{RP}-AV (η)

IV. NEW CLASSES OF SCHEMES FOR ADAPTATION OF BP PARAMETERS

In this section we propose two new classes of LA based schemes called class ***C*** and ***D***. In a class ***C*** scheme one automata is associated to each link of the network to adjust the parameter for that link and in a class ***D*** scheme one automata is associated to each neuron of the network to adjust the parameter for that neuron. Group ***C*** and ***D*** schemes may be referred to as the local parameter adaptation methods. We use **Automata-CV(γ)** and **Automata-CF(γ)** to refer to the schemes in class ***C*** and **Automata-DV(γ)** and **Automata-DF(γ)** to refer to the schemes in class ***D***.

We use class ***C*** schemes for adaptation of learning rate and class ***D*** schemes for adaptation of steepness parameter. In class ***C*** and ***D*** schemes, the automata receives favorable response from the environment if the algebraic sign of derivative in two consecutive iterations is the same and receives unfavorable response if the algebraic sign of the derivative in two consecutive iterations alternates. The following algorithms describe class ***C*** and class ***D*** schemes.

A deterministic fixed structure learning automaton: In what follows, we introduce a new FSLA called J-automata. This automata can be used by class *C* and class *D* schemes to achieve higher degree of performance than the other known schemes in class *C* or class *D*. The proposed automata which we denote it by $J_{KN,K}$ has KN states and K actions and attempts to incorporate the past behavior of the system in its decision rule for choosing the sequence of actions. States with numbers (k-1)N+1 through kN correspond to action k. The state transition graph of this automata for favorable response and unfavorable response is shown in figure 3.

Unfavorable Response ($\beta = 1$) Favorable Response ($\beta = 0$)

Figure 3: The state transition graph for $J_{KN,K}$

The favorable response is produced by the environment if the algebraic sign of the derivative in two consecutive iterations is the same and unfavorable response produced by the environment if the algebraic sign of the derivative in two consecutive iterations alternates. When this automata is used to adjust the learning rate (steepness parameter), each action correspond to one of the values of the learning rate (steepness parameter). The automata, on reward move to the higher number state (hoping eventually increases the value of the parameter) and on penalty move to the lower number state (hoping eventually decreases the value of the parameter).
It is only when the sign of derivative remains the same for N consecutive iterations or alternates for N consecutive iteration the automata changes its action (switches from one value for the related parameter to another value). Therefore N is memory depth associated with each action and automata is said to have a total memory of KN. If parameter μ can assume K values, $\mu_1 \le \mu_2 \le \ldots \le \mu_K$ then states (k-1)N + 1 through kN correspond to the value of μ_k. Clearly such an assignment of values of parameter μ to the states of the automata causes the value used by BP increases if in N consecutive iterations the sign of the derivative do not change and decreases if in N consecutive iterations the sign of derivative alternates.

Simultaneous adaptation of learning rate and steepness parameters: The rate of convergence can be improved if both learning rate and steepness parameter are adapted simultaneously. Class *C* scheme is used for adaptation of learning rate and a class *D* scheme is used for adaptation of steepness parameter. A scheme that simultaneously adapts learning rate and steepness parameter is denoted by **Automata1-Automata2-CDF(μ,λ)**, if FSLA is used and **Automata1-Automata2-CDV(μ, λ)**, if VSLA is used.

Simultaneous adaptation of learning rate and momentum factor: A simple method of increasing the learning rate and stability of training algorithm is to modify the standard BP by including the momentum factor [Rumelhart 1986] as given in equation (1). Solving the difference equation (1) gives to the following time series equation.

$$\Delta W(n) = \eta \sum_{t=0}^{n} \alpha^{n-t} G(t) \qquad (3)$$

By inspection of equation (3), we may make the following useful observations:
- This time series converged if and only if $0 \le |\alpha| < 1$.
- When G has same algebraic sign on consecutive iterations, $|\Delta W|$ grows, and W is adjusted by a large amount. Hence the inclusion of momentum term accelerate the convergence of algorithm. To accelerate more, the momentum factor must have large value as much as possible.
- When the algebraic sign of G alternates in consecutive iterations, the $|\Delta W|$ shrinks and W is adjusted by a small amount. Hence the inclusion of momentum term stabilize the convergence of algorithm. To accelerate more, the momentum factor must have small value as much as possible.

From the above observation, we may conclude that the momentum factor could be adjusted by $J_{KN,K}$ automata in the same manner as the learning rate. This scheme is denoted by **Automata1-Automata2-CF(η, α)**.

V. SIMULATIONS

Typical simulations for these four problems for different parameter adaptation schemes are shown in figures 4 through 7. Figure 4 compares the performance of different class *A* schemes with J_CF scheme. Figure 5 indicates that J_CF scheme has higher speed of convergence than any known scheme in class *C*. Figure 6 compare the J_CF scheme with the

EUFIT '98, September 7 - 10, 1998

best scheme in class ***B***. Figures 7 through 9 indicate that if μ and λ adapted simultaneously, the performance of the BP algorithm increases by a large amount. These figures compare the performance of J_J_CDF (μ, λ) with variable learning rate (VLR) scheme [Menhaj 1995] and schemes proposed by Arabshahi (Fuzzy BP) [Arabshahi 1992] and Darken and Moody (SC) [Sarkar 1995]. For all the simulations, we have taken the momentum factor (α) to be zero. Figures 10 through 12 show the performance of different schemes when both learning rate and momentum factor adapted. The plot for each simulation is averaged over 200 runs. . For more simulations refer to [Beigy 1998].

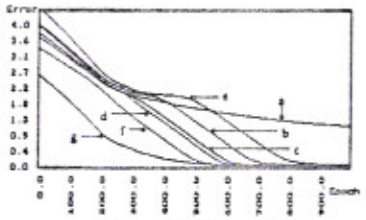

Figure 4: Digit problem
a: standard BP b: Tsetline(4, 4)-AF (η)
c: Krinsky(2, 4)-AF (η) d: Krylov(2, 4)-AF (η)
e: L_{R-P}-AV (η) f: VLR g: J(2, 1)-CF (η)

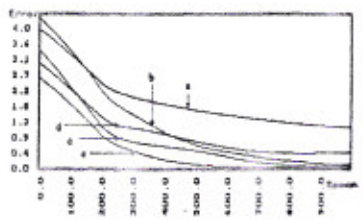

Figure 5: Digit problem
a: standard BP
b: Tsetline(4, 4)-CF (η) c: Krinsky(2, 4)-CF (η)
d: Krylov(2, 4)-CF (η) e: J(2, 1)-CF (η)

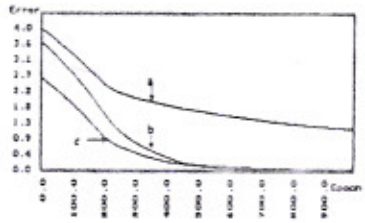

Figure 6: Digit problem
a: Standard BP
b: Tsetline(4,6)-Tsetline(2,4)-BF(η)
c: J-CF(η) (10, 1)

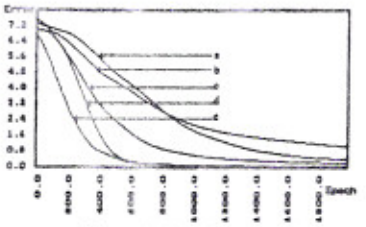

Figure 7: Encoding Problem
a: SC Scheme b: Standard BP c:VLR Scheme
d: Fuzzy BP e: J(2, 1)-J(5, 6)-CDF (η, λ)

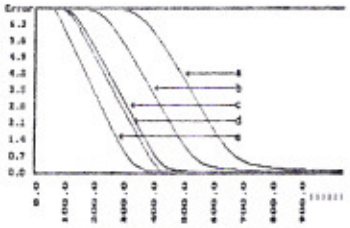

Figure 8: Symmetry Problem
a: SC Scheme b: Standard BP c:VLR Scheme
d: Fuzzy BP e: J(2, 1)-J(5, 6)-CDF (η, λ)

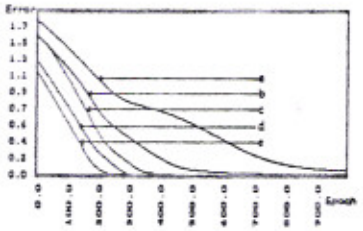

Figure 9: Parity Problem
a: SC Scheme b: Standard BP c:VLR Scheme
d: Fuzzy BP e: J(2, 1)-J(2, 1)-CDF (η, λ)

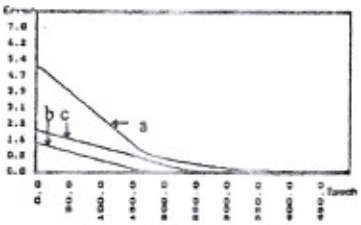

Figure 10 : Digit problem
a: J(5, 10)-CF(η)
b: J(5, 10)-J(5, 10)-CF(η,α)
c:J(5, 10)-CF(η) with constant momentum factor

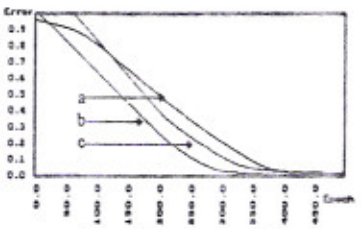

Figure 11 : Parity problem
a: J(5,10)-J(5,10)-CDF(η,λ)
b: J(5, 10)-J(5, 10)-CF(η,α)
c: J(5, 10)-CF(η) with fixed momentum factor

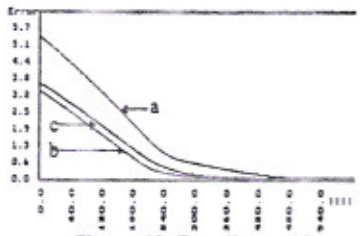

Figure 12: Encoding problem
a: J(5, 10)-J(5, 6)-CDF(η, λ)
b: J(5, 10)-J(5, 10)-CF(η,α)
c: J(5, 10)-CF(η) with fixed momentum factor

V. CONCLUSIONS

Dynamic learning rate adjustment could be indirectly or directly. Momentum and conjugate gradient methods are example of indirect methods. Since indirect methods haven't been satisfactory enough in many cases direct adjustment of the learning rate have been proposed. Direct methods can be classified into two groups: local methods, such as Bold Driver method [Sarkar 1995] in which one learning rate is used for all the weights and global methods such as SAB [Sarkar 1995] and Super SAB [Sarkar 1995]methods in which one learning rate is used for each weight.

In this paper two new classes of direct methods called ***B*** and ***C*** classes and a new class of indirect method called ***D*** class are presented. Since class ***B*** schemes adapt two different learning rates one for hidden layer and one for output layer it may be called quasi-global method. All the schemes in these classes uses learning automata of fixed or variable structure type to adapt BP parameter. The adaptation is based on the error surface behavior. With the introduction of these new classes of schemes we propose a new classification tree for learning rate adjustment methods. Figure 28 shows this new classification. To evaluate the performance of these methods, simulation studies were carried out on several learning problems with different error surfaces. Simulations indicate that dynamic adaptation of BP parameters

using the proposed methods increase the speed of convergence of the standard BP and superior to most previous schemes reported for adaptation of BP parameters.

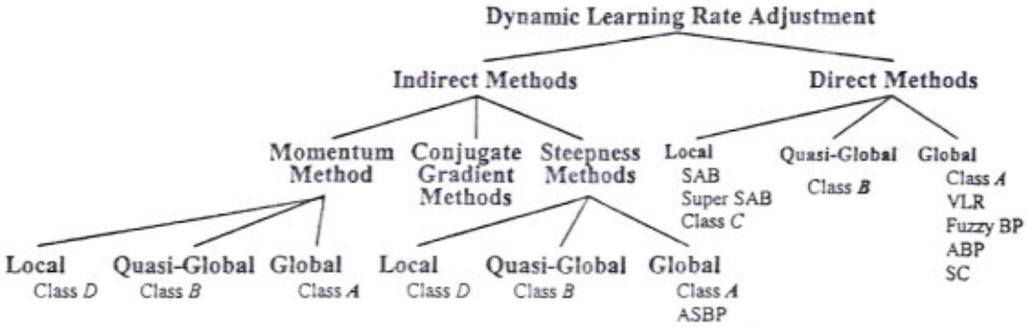

Figure 28 : Classification tree for dynamic learning rate adjustment schemes

VI. REFERENCES

Arabshahi, P., Choi J. J., Marks, R. J., and Caudell, T. P. 1992, Fuzzy Control of Back propagation", Proc. of IEEE Int. Conf. on Fuzzy Systems, pp. 967-972.

Beigy, H. and Meybodi, M. R. 1997, Adaptation of Momentum Factor and Steepness parameter in Backpropagation Algorithm Using Fixed Structure Learning Automata, Technical Report, Computer Eng. Dept., Amirkabir university of Technology, Tehran, Iran .

Beigy, H. , Meybodi, M. R. 1998, New Class of Learning Automata Based Scheme for Adaptation of Backpropagation Algorithm Parameters, Technical Report, Computer Eng. Dept., Amirkabir university of Technology, Tehran, Iran.

Beigy, H. , Menhaj, M. B., and Meybodi, M. R. 1998, Adaptation of Learning Rate in Backpropagation Algorithm Using Fixed Structure Learning Automata, Proc. of ICEE-98, Vol. III, pp. 117-123.

Kandil, N., Khorasani K., Patel R. V., and Sood V. K. 1993, Optimum Learning Rate for Back propagation Neural Networks, In Neural Networks Theory, Technology, and Applications, Ed. P. K. Simpson, pp. 249-251.

Narendra, K. S and Thathachar, M. A. L. 1989, Learning Automata: An Introduction, Prentice-Hall, Englewood cliffs.

Menhaj, M. B. and Meybodi, M. R. 1994, A Novel Learning Scheme for Feedforward Neural Nets", Proc. of ICEE-95, University of Science and Technology, Tehran, Iran.

Menhaj, M. B. and Hagen, M. H. 1995, Rapid Learning Using Modified Backpropagation Algorithms for Multi-Layer Feedforward Neural Nets, Proc. of ICEE-95, University of Science and Technology, Tehran, Iran.

Menhaj, M. B. and Meybodi, M. R. 1996, Flexible Sigmodal Type Functions for Neural Nets Using Game of Automata, Proc. of Second Annual CSI Computer Conference CSIC'96, Tehran, Iran , pp. 221-232.

Menhaj, M. B. and Meybodi, M. R. 1996, Application of Learning Automata to Neural Networks, Proc. of Second Annual CSI Computer Conference CSIC'96, Tehran, Iran , pp. 209-220.

Menhaj, M. B. and Meybodi, M. R. 1997, Using Learning Automata in Backpropagation Algorithm with Momentum", Technical Report, Computer Eng. Dept., Amirkabir University of Technology, Tehran, Iran.

Meybodi, M. R. and Lakshmivarhan, S. 1982, Optimality of a General Class of Learning Algorithm, Information Science, Vol. 28, pp. 1-20.

Meybodi, M. R. and Lakshmivarhan, S. 1984, On a Class of Learning Algorithms which have a Symmetric Behavior Under Success and Failure, Springer Verlag Lecture Notes in Statistics, pp. 145-155.

Meybodi, M. R. 1987, Results on Strongly Absolutely Expedient Learning Automata, Proc. of OU Inference Conf. 86, ed. D. R. Mootes and R. Butrick, Athens, Ohio: Ohio University Press, pp. 197-204.

Parlos, A. G., Fernadez, B., Atya, A. F., Muthusami, J., and Tsai W. K. 1994, An Accelerated Learning Algorithm for Multi-Layer Preceptron Networks, IEEE Trans. on Neural Networks, Vol. 5, No. 3, pp. 493-497.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. 1986, Learning Internal Representations by Error Propagation , In Parallel distributed processing, Cambridge, MA: MIT Press.

Sarkar, D. 1995, Methods to Speedup Error Backpropagation Learning Algorithm, ACM Computing Surveys, No. 4, Vol. 27.

Sperduti, A. and Starita, A. 1993, Speed Up Learning and Network Optimization with Extended Backpropagation", Neural Networks, Vol. 6, pp. 365-383.