A Framework for the Development of Globally Convergent Adaptive Learning Rate Algorithms

G.D. MAGOULAS[1,3], V.P. PLAGIANAKOS[2,3], G.S. ANDROULAKIS[2,3], M.N. VRAHATIS[2,3]

[(1)]Department of Informatics, University of Athens, GR-157.84 Athens, GREECE.

[(2)]Department of Mathematics, University of Patras, GR-261.10 Patras, GREECE.
URL:

[(3)]University of Patras Artificial Intelligence Research Center–UPAIRC.

Abstract: - In this paper we propose a framework for developing globally convergent batch training algorithms with adaptive learning rate. The proposed framework provides conditions under which global convergence is guaranteed for any training algorithm with adaptive learning rate. To this end, the learning rate is appropriately tuned along the given descent direction. Providing conditions regarding the search direction and the corresponding stepsize length this framework can also guarantee global convergence for any training algorithm with a different learning rate for each weight. In cases where the direction-related condition is not fulfilled the search direction is properly corrected and the stepsize length along the new search direction is adapted.

Keywords and phrases: Global convergence, learning rate adaptation, batch training algorithms, steepest descent, feedforward neural networks.

1 Introduction

The goal of neural network training is to iteratively update the network weights to minimize the learning error. The rapid computation of such a global minimum is a rather difficult task since, in general, the number of network weights is large and the corresponding nonconvex error function possesses multitudes of local minima and has broad flat regions adjoined with narrow steep ones.

To simplify the formulation of the equations throughout the paper we use a unified notation for the weights. Thus, for a Feedforward Neural Network (FNN) with a total of n weights, $\mathbb{R}^n$ is the n–dimensional real space of column weight vectors w with components $w_1, w_2, \dots, w_n$ and w^* is the optimal weight vector with components $w_1^*, w_2^*, \dots, w_n^*$; E is the batch error measure defined as the sum-of-squared-differences error function over the entire training set; $\partial_i E(w)$ denotes the partial derivative of $E(w)$ with respect to the ith variable w_i; $\nabla E(w)$ defines the gradient vector of the sum-of-squared-differences error function E at w while $H = [H_{ij}]$ defines the Hessian $\nabla^2 E(w)$ of E at w.

The batch training of an FNN is consistent with the theory of unconstrained optimization, since it uses information from all the training set, i.e. the true gradient, and can be viewed as the minimization of the error function E. This minimization corresponds to updating the weights by epoch and, to be successful, it requires a sequence of weight vectors $\{w^k\}_{k=0}^{\infty}$, where k indicates epochs, which converges to the point w^* that minimizes E.

The widely used batch Back–Propagation (BP) [20] is a first-order training algorithm, which minimizes the error function using the steepest descent method [7]:

$$w^{k+1} = w^k - \eta \nabla E(w^k), \tag{1}$$

where the gradient vector is usually computed by the back–propagation of the error through the layers of the FNN (see [20]) and η is a constant heuristically chosen learning rate. Appropriate learning rates help to avoid convergence to a saddle point or a maximum. In practice, a small constant learning rate is chosen ($0 < \eta < 1$) in order to secure the convergence of the BP algorithm and to avoid oscillations in the directions where the error surface is steep. However, this approach considerably slows down the training process since, in general, a small learning rate may not be appropriate for all the portions of the error surface.

Our motivation in this paper is to provide general theoretical results and strategies that are applicable to guarantee the convergence of adaptive learning rate algorithms. The algorithms differ according to the information they need to modify the learning rate. In training algorithms with a *global* learning rate, this rate is used to update all the weights in the FNN, while in algorithms with a *local* learning rate a different learning rate is used for each weight.

2 Adaptive learning rate algorithms

Several adaptive learning rate algorithms have been proposed to accelerate the training procedure. The following strategies are usually suggested: (i) start with a small learning rate and increase it exponentially, if successive epochs reduce the error, or rapidly decrease it, if a significant error increase occurs [2, 22], (ii) start with a small learning rate and increase it, if successive epochs keep gradient direction fairly constant, or rapidly decrease it, if the direction of the gradient varies greatly at each epoch [4] and (iii) for each weight an individual learning rate is given, which increases if the successive changes in the weights are in the same direction and decreases otherwise [9, 16, 18, 21]. Note that all the above mentioned strategies employ heuristic parameters in an attempt to enforce the monotone decrease of the learning error and to secure the converge of the training algorithm to a minimizer of E.

A different approach is based on Goldstein's and Armijo's work on steepest–descent and gradient methods. The method of Goldstein [8] requires the assumption that E is twice continuously differentiable on $\mathcal{S}(w^0)$, where $\mathcal{S}(w^0) = \{w \colon E(w) \leq E(w^0)\}$ is bounded, for some initial vector w^0. It also requires that η is chosen to satisfy the relation $\sup \|H(w)\| \leq \eta^{-1} < \infty$ in some bounded region where the relation $E(w) \leq E(w^0)$ holds. The kth iteration of an algorithm model that follows this approach consists of the following steps:

1. **Choose** η_0 to satisfy $\sup \|H(w)\| \leq \eta_0^{-1} < \infty$ and δ to satisfy $0 < \delta \leq \eta_0$.
2. **Set** $\eta^k = \eta$, where η is such that $\delta \leq \eta \leq 2\eta_0 - \delta$ and **go to** the next step.
3. **Update** the weights $w^{k+1} = w^k - \eta^k \nabla E(w^k)$.

However, the manipulation of the full Hessian is too expensive in computation and storage for FNNs with several hundred weights [3]. Le Cun [10] proposed a technique, based on appropriate perturbations of the weights, for estimating on–line the principle eigenvalues and eigenvectors of the Hessian without calculating the full matrix H. According to experiments reported in [10] the largest eigenvalue of the Hessian is mainly determined by the FNN architecture, the initial weights and by short–term low–order statistics of the training data. This technique could be used to determine η_0, in Step 1 of the above algorithm, requiring additional presentations of the training set in the early training.

An alternative approach is based on the work of Armijo [1]. Following this approach, the value of the learning rate η is related to the value of the Lipschitz constant K, which depends on the morphology of the error surface. In this case, the BP algorithm takes the form:

$$w^{k+1} = w^k - \frac{1}{2K} \nabla E(w^k), \tag{2}$$

and converges to the point w^* which minimizes E (see [1] for conditions under which convergence occurs and a convergence proof).

However, in practice neither the morphology of the error surface nor the value of K are known a priori. In [12] a local estimation of the Lipschitz constant has been proposed, as part of a learning rate adaptation strategy that provides increased rate of convergence through the Lipschitz constant estimation and guarantees the stability of the learning procedure.

3 Monotone decrease of the error function and global convergence

A training algorithm can be made globally convergent by determining the learning rate in such a way that the error is exactly minimized along the current search direction at each epoch, i.e. $E(w^{k+1}) < E(w^k)$. To this end, an iterative search, which is often expensive in terms of error function evaluations, is required. It must be noted that the above simple condition does not guarantee global convergence for general functions, i.e. converges to a local minimizer from any initial condition (see [5] for a general discussion on globally convergent methods).

The use of adaptive learning rate algorithms which enforce monotonic error reduction using inappropriate values for the critical heuristic learning parameters can considerably slow the rate of training, or even lead to divergence and to premature saturation [11, 19]. Moreover, using heuristics it is not possible to develop globally convergent training algorithms, i.e. algorithms with the property that starting from any initial weight vector the sequence of the weights converges to a local minimizer of the error function.

To alleviate this situation it is preferable to tune the learning rate, which is evaluated by an adaptive learning rate algorithm, so that the error function is sufficiently decreased at each epoch, accompanied by a significant change in the value of w. A strategy of this kind consists in accepting a positive learning rate η^k along the search direction φ^k if it satisfies the *Wolfe conditions*:

$$E(w^k + \eta^k \varphi^k) - E(w^k) \leq \sigma_1 \eta^k \langle \nabla E(w^k), \varphi^k \rangle, \quad (3)$$

$$\langle \nabla E(w^k + \eta^k \varphi^k), \varphi^k \rangle \geq \sigma_2 \langle \nabla E(w^k), \varphi^k \rangle, \quad (4)$$

where $0 < \sigma_1 < \sigma_2 < 1$ and $\langle \cdot, \cdot \rangle$ stands for the usual inner product in $\mathbb{R}^n$. The first inequality ensures that the error is reduced sufficiently and the second prevents the learning rate from being too small. It can be shown that if φ^k is a descent direction, if E is continuously differentiable and if E is bounded below along the ray $\{w^k + \eta \varphi^k \mid \eta > 0\}$, then there always exist learning rate satisfying (3)–(4) [14, 5]. Relation (4) can be replaced by

$$E(w^k + \eta^k \varphi^k) - E(w^k) \geq \sigma_2 \eta^k \langle \nabla E(w^k), \varphi^k \rangle, \quad (5)$$

where $\sigma_2 \in (\sigma_1, 1)$ (see [5]).

An alternative strategy has been proposed in [17]. It is applicable to any descent direction φ^k and uses two parameters $\alpha, \beta \in (0,1)$. Following this approach the learning rate is $\eta^k = \beta^{m_k}$, where $m_k \in \mathbb{Z}$ is any integer such that

$$E(w^k + \beta^{m_k} \varphi^k) - E(w^k) \leq \beta^{m_k} \alpha \langle \nabla E(w^k), \varphi^k \rangle \quad (6)$$

$$E(w^k + \beta^{m_k - 1} + \varphi^k) - E(w^k) >$$
$$> \beta^{m_k - 1} \alpha \langle \nabla E(w^k), \varphi^k \rangle. \quad (7)$$

An algorithm model that incorporates the above strategy is given below. It can be implemented in two versions depending on the input value of the parameter s.

Algorithm 1

1. **Input** $\{E; w^0; \alpha, \beta \in (0,1); s \in \{0,1\}; m^* \in \mathbb{Z}; MIT; \varepsilon\}$.
2. **Set** $k = 0$.
3. **If** $\|\nabla E(w^k)\| \leq \varepsilon$ **go to** Step 6. **Else, compute** a descent direction φ^k.
4. **If** $s = 0$, **set** $M^* = \{m \in \mathbb{Z} \mid m \geq m^*\}$ and **compute** the learning rate $\eta^k = \beta^{m_k}$ by $\beta^{m_k} = \arg \max_{m \in M^*} \Big\{ \beta^m \mid E(w^k + \beta^m \varphi^k) - E(w^k) \leq \beta^m \alpha \langle \nabla E(w^k), \varphi^k \rangle \Big\}$.
   Else ($s = 1$) **compute** the $\eta^k = \beta^{m_k}$, where $m_k \in \mathbb{Z}$ is any integer such that $E(w^k + \beta^{m_k} \varphi^k) - E(w^k) \leq \beta^{m_k} \alpha \langle \nabla E(w^k), \varphi^k \rangle$
   and
   $E(w^k + \beta^{m_k - 1} \varphi^k) - E(w^k) > \beta^{m_k - 1} \alpha \langle \nabla E(w^k), \varphi^k \rangle$.
5. **Set** $w^{k+1} = w^k + \eta^k \varphi^k$. **If** $k < MIT$, **replace** k by $k+1$, and **go to** Step 3; **otherwise go to** Step 6.
6. **Output** $\{w^k; E(w^k); \nabla E(w^k)\}$.

The selection $s = 0$ is normally used with second order algorithms, with $m^* = 0$ to ensure superlinear convergence. The selection $s = 0$ is not very good for first-order algorithms because, on the average, it requires considerably more function evaluations than the selection $s = 1$. So, $s = 1$ is used in first-order algorithms.

All the above strategies must be combined with tuning subprocedures generating learning rates that satisfy conditions (3)–(4) or (6)–(7) in order to guarantee global convergence. This issue is the subject of the next section.

4 Global convergence by tuning the learning rate

In this section we propose learning rate tuning subprocedures and establish useful convergence theorems due to Wolfe [23, 24] and Polak [17].

The strategy based on Wolfe's conditions provides an efficient and effective way to ensure that the error function is globally reduced sufficiently. In practice, the condition (4) or (5) generally is not needed because the use of a backtracking strategy avoids very small learning rates. A simple backtracking strategy to tune the length of the minimization step, so that it satisfies conditions (3)–(4) at each epoch, is to decrease the learning rate by a reduction factor $1/q$, where $q > 1$ [15]. This has the effect that the learning rate is decreased by the largest number in the sequence $\{q^{-m}\}_{m=1}^{\infty}$, so that the condition (3) is satisfied. We remark here that the selection of q is not critical for successful learning, however it has an influence on the number of error function evaluations required to satisfy the condition (3). Thus, when seeking to satisfy (3) it is important to ensure that the learning rate is not reduced unnecessarily so that the condition (4) is not satisfied. Since, in training the gradient vector is known only at the beginning of the iterative search for a new weight vector, the condition (4) cannot be checked directly (this task requires additional gradient evaluations at each epoch), but is enforced simply by placing a lower bound on the acceptable values of the learning rate. This bound on the learning rate has the same theoretical effect as the condition (4) and ensures global convergence [5]. The value $q = 2$ is usually suggested in the literature [1] and indeed it was found to work without problems in the experiments (see [13]).

In this framework, an important theorem due to Wolfe [5] states that if E is bounded below, then the sequence $\{w^k\}_{k=0}^{\infty}$ generated by any algorithm that follows a descent direction φ^k whose angle θ_k with $-\nabla E(w^k)$ is such that

$$\cos\theta_k = \frac{\langle -\nabla E(w^k), \varphi^k \rangle}{\|\nabla E(w^k)\| \|\varphi^k\|} > 0, \qquad (8)$$

and satisfy the Wolfe's conditions, will also obey $\lim_{k\to\infty} \nabla f(w^k) = 0$ [5, 14].

Theorem 1[5, 14, 23, 24]. *Suppose that the error function* $E : \mathbb{R}^n \to \mathbb{R}$ *is continuously differentiable on* $\mathbb{R}^n$ *and assume that* ∇E *is Lipschitz continuous on* $\mathbb{R}^n$. *Then, given any* $w^0 \in \mathbb{R}^n$, *either* E *is unbounded below, or there exists a sequence* $\{w^k\}_{k=0}^{\infty}$ *obeying the Wolfe's conditions (3)–(4) and either*

(1) $\langle \nabla E(w^k), (w^{k+1} - w^k) \rangle < 0$, *or*

(2) $\nabla E(w^k) = 0, \quad \text{and} \quad w^{k+1} - w^k = 0,$

for each $k > 0$. *Furthermore, for any such sequence, either*

(a) $\nabla E(w) \neq 0$ *for some* $k \geq 0$, *or*

(b) $\lim_{k\to\infty} E(w^k) = -\infty$, *or*

(c) $\lim_{k\to\infty} \langle \nabla E(w^k), (w^{k+1} - w^k) \rangle / \|w^{k+1} - w^k\| = 0.$

This is also true when Relation (4) is replaced by Relation (5) [5](cf. Relation (c) of Step 4 of Algorithm 1). For a relative convergence result where the sequence $\{w^k\}_{k=0}^{\infty}$ converges q-superlinearly to a minimizer w^* see [5].

Regarding Polak's approach, if the error function E is bounded from below the following subprocedure can be used to find an m_k satisfying Relations (b) and (c) of Step 4 of the Algorithm 1. This subprocedure uses the last used learning rate $\eta^{k-1} = \beta^{m_{k-1}}$ as the starting point for the computation of the next one [17]:

1. **If** $k = 0$, **set** $m' = m^*$. **Else set** $m' = m_{k-1}$.
2. **If** $m_k = m'$ satisfies Relations (b) and (c) of Step 4 of Algorithm 1, **stop**.
3. **If** $m_k = m'$ satisfies (b) but not (c), **replace** m' by $m' - 1$, and **go to** Step 2.

   If $m_k = m'$ satisfies (c) but not (b), **replace** m' by $m' + 1$, and **go to** Step 2.

In practice, only a very small number of iterations of the above subprocedure are required to compute the learning rate. When a very small learning rate occurs for several iterations, causing slow convergence, the user can revert to setting $s = 0$ for one or two iterations.

The search strategy of Algorithm 1 allows us to establish the following useful convergence theorem due to Polak [17]. This theorem requires the search direction φ^k to be bounded from above, it imposes a restriction on the angle between $\nabla E(w^k)$ and φ^k (see Relation (8)) and states that Algorithm 1 is well defined in the sense that whenever $\nabla E(w^k) \neq 0$, the search for a learning rate η^k is a finite process, whether $s = 0$ or $s = 1$.

Theorem 2 [17]. *Assume that (i) the error function* $E : \mathbb{R}^n \to \mathbb{R}$ *is Lipschitz continuously differentiable on bounded sets; (ii) the sequences* $\{w^k\}_{k=0}^{\infty}$ *and* $\{\varphi^k\}_{k=0}^{\infty}$ *are constructed by Algorithm 1; (iii) there exist two continuous functions* $N_1 : \mathbb{R}^n \to \mathbb{R}$ *and* $N_2 : \mathbb{R}^n \to \mathbb{R}$ *such that*

(1) for all w *satisfying* $\nabla E(w) \neq 0$, $N_1(w) > 0$, $N_2(w) > 0$ *and* $N_1(w) = 0$ *if and only if* $\nabla E(w) = 0$ *and*

(2) for all $k \in \mathbb{N}$, *the* w^k *and* φ^k *satisfy the inequalities* $\langle \nabla E(w^k), \varphi(w^k) \rangle \leq -N_1(w^k)$, *and* $\|\varphi^k\| \leq N_2(w^k)$.

Under these assumptions,

(a) if w^k *is such that* $\nabla E(w^k) \neq 0$, *then* η^k *is computed by Algorithm 1 using a finite number of function evaluations and*

(b) any accumulation point w^* *of the sequence* $\{w^k\}_{k=0}^{\infty}$ *satisfies* $\nabla E(w^*) = 0$.

5 Global convergence by adapting the search direction

A batch BP algorithm with a different learning rate for each weight is defined by the iterative scheme:

$$w^{k+1} = w^k - \mathrm{diag}\{\eta_1^k, \eta_2^k, \ldots, \eta_n^k\}\, \nabla E(w^k). \qquad (9)$$

The learning rates are evaluated employing heuristic procedures that exploit information regarding the history of the partial derivative of $E(w)$ with respect to the ith weight and/or the history of the corresponding learning rate, depending on the algorithm. Appropriate values of the heuristics ensure that the error function is decreased in each weight direction, every epoch. The well known *delta-bar-delta* method [9] and Silva and Almeida's method [21] follow this approach. Another method, named *quickprop* [6] is based on independent secant steps in the direction of each weight. The *Rprop* algorithm [18] updates the weights using the learning rate and the sign of the partial derivative of the error function with respect to each weight.

Clearly, the weight vector in Eq. (9) is not updated in the direction of the negative of the gradient; instead, an alternative adaptive search direction is obtained by taking into consideration the weight change, evaluated by multiplying the length of the search step, i.e. the value of the learning rate, along each weight direction by the partial derivative of $E(w)$ with respect to the corresponding weight, i.e. $-\eta_i \partial_i E(w)$. In other words, the algorithms of this class try to decrease the error in each direction, by searching the local minimum with small weight steps. These steps are usually constraint by problem-dependent heuristic parameters in order to ensure subminimization of the error function in each weight direction.

A well known difficulty of this approach is that the use of inappropriate heuristic values for a weight direction misguides the resultant search direction. In such cases, the training algorithm cannot exploit the global information obtained by taking into consideration all the directions. To alleviate this situation, we propose the search direction to be obtained by taking into consideration $n-1$ learning rates, as directly evaluated by any adaptive learning rate algorithm and analytically evaluate the remain one. This approach has the effect that the search direction is properly corrected and ensures that the direction followed is indeed a descent one. The following theorem provides a global convergence result for training algorithms with a different learning rate for each weight.

Theorem 3. *Suppose that the error function* $E : \mathbb{R}^n \to \mathbb{R}$ *is continuously differentiable. Assume that* ∇E *is Lipschitz continuous on* $\mathbb{R}^n$. *Then, given any point* $w^0 \in \mathbb{R}^n$, *for any sequence* $\{w^k\}_{k=0}^{\infty}$, *generated by the iterative scheme:*

$$w^{k+1} = w^k - \tau^k \operatorname{diag}\{\eta_1^k, \eta_2^k, \dots, \eta_n^k\} \nabla E(w^k), \quad (10)$$

where η_m^k, $m = 1, 2, \dots, i-1, i+1, \dots, n$ *are arbitrarily chosen positive real numbers and*

$$\eta_i^k = -\frac{\delta}{\partial_i E(w^k)} - \frac{1}{\partial_i E(w^k)} \sum_{\substack{j=1 \\ j\neq i}}^{n} \eta_j^k \, \partial_j E(w^k), \quad (11)$$

where δ *is a positive real number and* $\tau^k > 0$ *satisfies the Wolfe's conditions (3)–(4) implies that*

$$\lim_{k\to\infty} \nabla E(w^k) = 0.$$

Proof: Evidently, the error function E is bounded below on $\mathbb{R}^n$. The sequence $\{w^k\}_{k=0}^{\infty}$ follows the direction

$$\varphi^k(w^k) = -\operatorname{diag}\{\eta_1^k, \eta_2^k, \dots, \eta_n^k\} \nabla E(w^k),$$

which is a descent direction since

$$\left\langle \nabla E(w^k), \varphi^k(w^k) \right\rangle < 0.$$

Now, since φ^k is a descent direction and since E is continuously differentiable and bounded below then there always exist τ^k satisfying the Wolfe's conditions:

$$E(w^k + \tau^k \varphi^k) - E(w^k) \leq \sigma_1 \tau^k \langle \nabla E(w^k), \varphi^k \rangle, \quad (12)$$

$$\langle \nabla E(w^k + \tau^k \varphi^k), \varphi^k \rangle \geq \sigma_2 \langle \nabla E(w^k), \varphi^k \rangle, \quad (13)$$

for $0 < \sigma_1 < \sigma_2 < 1$. Moreover, the restriction on the angle θ_k is fulfilled since, as it can be easily justified utilizing Relation (8), $\cos\theta_k > 0$. Thus, by the Wolfe's Theorem [5], it holds that $\lim_{k\to\infty} \nabla E(w^k) = 0$. Thus the Theorem is proved.

Remark 1: Note that for neural networks with sigmoid activation functions the assumption on continuous differentiability of the error function is redundant.

Remark 2: A relative convergence result can be proved for any sequence $\{w^k\}_{k=0}^{\infty}$ satisfying the relations (3) and (5).

Remark 3: The use of $\tau^k = 1$ is suggested. This has the effect that the minimization step along the resultant search direction is defined by the value of the learning rates. By tuning τ^k, the length of the minimization step is regulated to satisfy the Wolfe's conditions, while the weights are updated in a descent direction.

6 Concluding remarks

A framework for the development of globally convergent batch training algorithms with adaptive learning rates has been proposed. The proposed framework provides conditions under which global convergence is guaranteed and strategies for tuning the adaptive learning rate and search direction. A new general result for the global convergence has been established which is applicable to any training algorithm with a different learning rate for each weight.

References

[1] Armijo L., Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific Journal of Mathematics*, 16, 1966, 1–3.

[2] Battiti R., Accelerated backpropagation learning: two optimization methods, *Complex Systems*, 3, 1989, 331–342.

[3] Becker S. and Le Cun Y., Improving the convergence of the back–propagation learning with second order methods, in *Proc. of the 1988 Connectionist Models Summer School*, D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski (eds.), 29-37, Morgan Koufmann, San Mateo, CA, 1988.

[4] Chan L.W. and Fallside F., An adaptive training algorithm for back–propagation networks, *Computers Speech and Language*, 2, 1987, 205–218.

[5] Dennis J.E. and Schnabel R.B., *Numerical Methods for Unconstrained Optimization and nonlinear equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] Fahlman S.E., Faster-learning variations on back-propagation: an empirical study, in *Proc. of the 1988 Connectionist Models Summer School*, D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski (eds.), 38-51, Morgan Kaufmann, 1989.

[7] Gill P.E., Murray W. and Wright M.H., *Practical Optimization*, Academic Press, NY, 1981.

[8] Goldstein,A.A., Cauchy's method of minimization, *Numer. Math.*, 4, 1962, 146–150.

[9] Jacobs, R.A., Increased rates of convergence through learning rate adaptation, Neural Networks, 1, 1988, 295-307.

[10] Le Cun Y., Simard P.Y. and Pearlmutter B.A., Automatic learning rate maximization by on–line estimation of the Hessian's eigenvectors, in *Advances in Neural Information Processing Systems 5*, S.J. Hanson, J.D. Cowan, and C.L. Giles (eds.), 156–163, Morgan Kaufmann, San Mateo, CA, 1993.

[11] Lee Y., Oh S.-H. and Kim M.W., An analysis of premature saturation in backpropagation learning, *Neural Networks*, 6, 1993, 719–728.

[12] Magoulas G.D., Vrahatis M.N. and Androulakis G.S., Effective back-propagation with variable stepsize, *Neural Networks*, 10, 1997, 69-82.

[13] Magoulas G.D., Vrahatis M.N. and Androulakis G.S., Improving the convergence of the back-propagation algorithm using learning rate adaptation methods, *Neural Computation*, accepted for publication.

[14] Nocedal J., Theory of algorithms for unconstrained optimization, *Acta Numerica*, 1992, 199–242.

[15] Ortega J.M. and Rheinboldt W.C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, NY, 1970.

[16] Pfister M. and Rojas R., Speeding-up backpropagation -A comparison of orthogonal techniques, in *Proc. of the Joint Conference on Neural Networks*, Nagoya, Japan, 517–523, 1993.

[17] Polak E., *Optimization: Algorithms and Consistent Approximations*, Springer-Verlag, NY, 1997.

[18] Riedmiller M. and Braun H., A direct adaptive method for faster back-propagation learning: the Rprop algorithm, in *Proc. of the IEEE Int. Conf. on Neural Networks, San Francisco*, CA, 586-591, 1993.

[19] Rigler A.K., Irvine J.M. and Vogl T.P., Rescaling of variables in backpropagation learning, *Neural Networks*, 4, 1991, 225–229.

[20] Rumelhart D.E., Hinton G.E. and Williams R.J., Learning Internal Representations by Error Propagation, in D. E. Rumelhart, and J. L. McClelland (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, 318–362. MIT Press, Cambridge, MA, 1986.

[21] Silva F. and Almeida L., Acceleration techniques for the back-propagation algorithm, *Lecture Notes in Computer Science*, 412, 1990, 110-119, Springer-Verlag, Berlin.

[22] Vogl T.P, Mangis J.K., Rigler J.K., Zink W.T. and Alkon D.L., Accelerating the convergence of the back-propagation method, *Biological Cybernetics*, 59, 1988, 257–263.

[23] Wolfe P., Convergence conditions for ascent methods, *SIAM Review*, 11, 1969, 226–235.

[24] Wolfe P., Convergence conditions for ascent methods. II: Some corrections, *SIAM Review*, 13, 1971, 185–188.