Neural Processing Letters **12:** 159–169, 2000.
© 2000 *Kluwer Academic Publishers. Printed in the Netherlands.*

Globally Convergent Modification of the Quickprop Method

MICHAEL N. VRAHATIS[1,3], GEORGE D. MAGOULAS[2,3]* and VASSILIS P. PLAGIANAKOS[1,3]
[1]*Department of Mathematics, University of Patras, GR-261.10 Patras, Greece.* [2]*Department of Informatics, University of Athens, GR-157.84 Athens, Greece.* [3]*University of Patras Artificial Intelligence Research Center–UPAIRC.*
e-mail: vrahatis@math.upatras.gr

Abstract. A mathematical framework for the convergence analysis of the well-known Quickprop method is described. Furthermore, we propose a modification of this method that exhibits improved convergence speed and stability, and, at the same time, alleviates the use of heuristic learning parameters. Simulations are conducted to compare and evaluate the performance of the new modified Quickprop algorithm with various popular training algorithms. The results of the experiments indicate that the increased convergence rates achieved by the proposed algorithm, affect by no means its generalization capability and stability.

Mathematics Subject Classifications (2000); 68T05, 90C53, 65H10, 65K05

Key words: Quickprop algorithm, Broyden's method, secant methods, convergence analysis, backpropagation neural networks

1. Introduction

The Quickprop (Qprop) method [6], is a very popular batch training algorithm for Feedforward Neural Networks (FNNs). It is well known that far from the neighborhood of a minimizer the morphology of the error surface, in certain cases, causes the Qprop algorithm to create inappropriate learning rates and the algorithm exhibits stability problems. Application-dependent heuristic learning parameters are used to alleviate this problem.

In this paper we analyze the Qprop method as a multivariable generalization of the secant method for nonlinear equations applied to the gradient of the batch error function. Furthermore, we present a modification of this algorithm that exhibits improved convergence speed and stability, and, at the same time, alleviates the use of heuristic learning parameters. Moreover, we prove a theorem for the convergence of the modified scheme. Using this convergence result the convergence of a class of modified Qprop algorithms is secured and the same holds for the classical method.

*G.D.M. is supported by a postdoctoral fellowship from the Greek State Scholarship Foundation (I.K.Y.).

2. Secant Methods

Let $F = (f_1, f_2, \dots, f_n) : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^n$ be a Fréchet-differentiable function and x^* a solution of the nonlinear system of equations:

$$F(x) = \Theta^n \equiv (0, 0, \dots, 0) \tag{1}$$

within the interior of $\mathcal{D}$.

The best known method for approximating x^* numerically is Newton's method. Given an initial guess x^0, this method computes a sequence of points $\{x^k\}_{k=0}^{\infty}$, obtained by solving the following *Newton's equation:*

$$F'(x^k)\bigl(x^{k+1} - x^k\bigr) = -F(x^k). \tag{2}$$

If x^0 is sufficiently close to x^*, F is continuously differentiable in a neighborhood of x^* and the Jacobian $F'(x^*)$ is nonsingular, the iterates $\{x^k\}$ of Newton's method converge quadratically to x^*. Furthermore, under the same assumptions any sequence $\{y^k\}$ which converges to x^* superlinearly is closely related to Newton's method by the fact that the relative difference between $y^{k+1} - y^k$ and the Newton correction $-F'(y^k)^{-1}F(y^k)$ will tend to zero [3].

The quadratic convergence of Newton's method is attractive. However, the method depends on a good initial approximation and it requires in general $n^2 + n$ function evaluations per iteration besides the solution of an $n \times n$ linear system.

Quasi-Newton methods were developed to save computational effort of individual iterations while maintaining some convergence properties of Newton's method. They maintain approximations of x^* and the Jacobian at the solution $F'(x^*)$ as the iteration progresses. If x^k and B_k are the current approximate solution and Jacobian, then after the computation of x^{k+1}, B_k is *updated* to form B_{k+1}. The construction of B_{k+1} determines the quasi-Newton method. Given an initial guess x^0, this method computes a sequence of points $\{x^k\}_{k=0}^{\infty}$, obtained by solving the following *quasi–Newton* or *secant equation* [5]:

$$B_{k+1}\bigl(x^{k+1} - x^k\bigr) = F(x^{k+1}) - F(x^k). \tag{3}$$

The advantages of quasi-Newton methods is that they require only n function evaluations for each iteration. Hence, if a good preconditioner (initial approximation to $F'(x^*)$) can be found, these methods have an advantage in terms of function evaluation cost over Newton's method. In most quasi-Newton methods derivatives are not computed at every iteration and it is not necessary to solve completely a linear system like Equation (2). On the other hand, the local rate of convergence turns to be superlinear instead of quadratic for most of these methods.

The most used approximation to the Jacobian has been proposed by Broyden (Broyden, 1965). His method is locally superlinear convergent and therefore is a very powerful alternative to Newton's method. Broyden's algorithm for solving Equation (1) has the following general form (cf. [5]). Given an initial guess x^0

and a nonsingular matrix B_0, this method computes a sequence of steps s_k obtained as follows:

```
for $k = 0, 1, \ldots$ until convergence do:
    Solve $B_k s_k = -F(x^k)$ for $s_k$,
    Set $x^{k+1} = x^k + s_k$,
    Set $z^k = F(x^{k+1}) - F(x^k)$,
    Set $B_{k+1} = B_k + \frac{(z^k - B_k s_k) s_k^\top}{s_k^\top s_k}$.
```

Broyden's method is very popular in practice for two main reasons: first, it generally requires fewer function evaluations than a finite difference Newton's method and second, it can be implemented in ways that require only $O(n^2)$ arithmetic operations per iteration ([4], 1989, pp. 27–29).

3. The Quickprop Method

In this section, we show that the Qprop method belongs to the family of secant methods but it is related to the minimization of the error function. It is well known that in minimization problems all the local minima w^* of a continuously differentiable batch error function E satisfy the necessary conditions:

$$\nabla E(w^*) = \Theta^n, \tag{4}$$

where ∇E denotes the gradient of E. Equation (4) represents a set of n nonlinear equations, which must be solved to obtain w^*. Therefore, one approach to the minimization of E is to seek the solutions of the set of Equation (4) by including a provision to ensure that the solution found does, indeed, correspond to a local minimizer. This is equivalent to solving the following system of equations:

$$\begin{aligned} \partial_1 E(w_1, w_2, \ldots, w_n) &= 0, \\ \partial_2 E(w_1, w_2, \ldots, w_n) &= 0, \\ &\vdots \\ \partial_n E(w_1, w_2, \ldots, w_n) &= 0, \end{aligned} \tag{5}$$

where $\partial_i E$ denotes the ith coordinate of ∇E.

The classical iterative scheme of the Qprop method for the ith weight is given by:

$$w_i^{k+1} = w_i^k - \left\{ \frac{\partial_i E(w^k) - \partial_i E(w^{k-1})}{w_i^k - w_i^{k-1}} \right\}^{-1} \partial_i E(w^k).$$

Using matrix formulation the above scheme can by written as:

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

where the matrix B_k is the diagonal matrix with elements $[b_{ii}^k]$, $i = 1, 2, \dots, n$ given by:

$$b_{ii}^k = \frac{\partial_i E(w^k) - \partial_i E(w^{k-1})}{w_i^k - w_i^{k-1}}.$$

It is obvious that the matrix B_k satisfies the following secant equation:

$$B_k\bigl(w^k - w^{k-1}\bigr) = \nabla E(w^k) - \nabla E(w^{k-1}), \tag{6}$$

and thus the Qprop method belongs to the class of quasi–Newton methods.

Using the above framework a straightforward modification of the Quickprop method is the following:

$$w_i^{k+1} = w_i^k - \eta_i \left\{ \frac{\partial_i E(w^k) - \partial_i E(w^{k-1})}{w_i^k - w_i^{k-1}} \right\}^{-1} \partial_i E(w^k),$$

where η_i are arbitrary nonzero real numbers. This is so because the new $\eta_i b_{ii}^k$ satisfy the corresponding secant equation.

Based on the above analysis it is obvious that the Qprop, as well as the above modification, follows the convergence properties of the secant methods [5, 15, 16].

In general, the matrix B_k may contain non positive entries. This fact results in a non positive definite matrix, which in practice means that the method may take uphill or zero steps in the corresponding directions. To alleviate this problem, a heuristic parameter called ‘the maximum growth factor’ has been introduced [16].

4. Globally Convergent Adaptive Learning Rate Algorithms

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 (for a general discussion on globally convergent methods see [5]).

The use of adaptive learning rate algorithms that 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 [10, 18]. Moreover, 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, by the use of heuristics. To deal with this problem it is preferable to tune the adaptive learning rate so that the error function is sufficiently decreased at each epoch, accompanied by a significant change in the value of w.

To this end, for the iterative scheme

$$w^{k+1} = w^k + \eta^k \varphi^k, \tag{7}$$

where φ^k is the search direction, the following conditions due to Wolfe can be used:

$$E(w^{k+1}) - E(w^k) \leqslant \sigma_1 \eta^k \langle \nabla E(w^k), \varphi^k \rangle, \tag{8}$$

$$\langle \nabla E(w^{k+1}), \varphi^k \rangle \geqslant \sigma_2 \langle \nabla E(w^k), \varphi^k \rangle, \tag{9}$$

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 function is reduced sufficiently, and the second prevents the steps 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 radius $\{w^k + \eta\varphi^k \mid \eta > 0\}$, then there always exist stepsize satisfying Equations (8)–(9) [14, 22, 23].

The following theorem, due to [5, 14, 22, 23], 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:

$$\cos\theta_k = \frac{\langle -\nabla E(w^k), \varphi^k \rangle}{\|\nabla E(w^k)\| \, \|\varphi^k\|} \geqslant \delta > 0, \tag{10}$$

and satisfies the Wolfe conditions, then it holds that either $\nabla E(w^k) = 0$ for some k, or $\lim_{k\to\infty} \nabla E(w^k) = 0$ [4].

THEOREM 1. [5, 14, 22, 23]. *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 conditions (8)–(9) and either:*

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

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

(a) $\nabla E(w) = 0$ *for some $k \geqslant 0$, or*
(b) $\lim_{k\to\infty} E(w^k) = -\infty$, *or*
(c) $\lim_{k\to\infty} \langle \nabla E(w^k), (w^{k+1} - w^k) \rangle / \|w^{k+1} - w^k\| = 0.$

For a relative convergence result where the sequence $\{w^k\}_{k=0}^{\infty}$ converges q-superlinearly to a minimizer w^* see [5, p. 123].

In practice, the condition (9) is not needed because the use of a backtracking strategy avoids very small steps [13]. A simple backtracking strategy to tune the length of the minimization step, so that it satisfies conditions (8)–(9) at each epoch, is to decrease the learning rate by a reduction factor $1/q$, where $q > 1$ [13, 15]. Also it can be proved (see [5]) that if (9) is replaced by:

$$E(w^k + \eta^k \varphi^k) - E(w^k) \geqslant \sigma_2 \eta^k \langle \nabla E(w^k), \varphi^k \rangle, \tag{11}$$

where $\sigma_2 \in (\sigma_1, 1)$, then Theorem 1 still holds.

5. The Modified Qprop Algorithm

To avoid tuning the problem dependent heuristics of the Qprop method and to guarantee the desirable property of positive definiteness of B_k we propose the following modification:

$$w_i^{k+1} = w_i^k - \eta \left\{ \frac{|\partial_i E(w^k) - \partial_i E(w^{k-1})|}{|w_i^k - w_i^{k-1}|} \right\}^{-1} \partial_i E(w^k),$$

where the coefficient η can be properly tuned. In this way, the length of the minimization step is regulated to satisfy the Wolfe conditions, while the weights are updated in a descent direction.

A high level description of this modified Qprop (MQprop) algorithm is given below, where *MIT* indicates the maximum number of iterations and ϵ the desired accuracy:

ALGORITHM 1. **Modified Quickprop Algorithm – MQprop**

```
(1) Input $\{E; w^0; \eta^0; (\lambda_1^0, \lambda_2^0, \dots, \lambda_n^0); MIT; \epsilon\}$.
(2) Set $k = -1$.
(3) If $k < MIT$, replace $k$ by $k + 1$, set $\eta = \eta^0$, and go to the next step; otherwise, go to
    Step 8.
(4) If $k \geqslant 1$ and $\Lambda_i^k = |\partial_i E(w^k) - \partial_i E(w^{k-1})|/|w_i^k - w_i^{k-1}| \neq 0$, for all $i = 1, \dots, n$, set
    $\lambda_i^k = 1/\Lambda_i^k$; otherwise set $\lambda_i^k = \lambda_i^0$.
(5) Tune $\eta$ by means of a tuning subprocedure.
(6) Set $w^{k+1} = w^k - \eta \, \mathrm{diag}\{\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k\} \nabla E(w^k)$.
(7) If $\|\nabla E(w^k)\| \leqslant \epsilon$ go to Step 8; otherwise go to Step 3.
(8) Output $\{w^k; E(w^k); \nabla E(w^k)\}$.
```

Assume now that the tuning subprocedure of Step 5 of Algorithm 1 consists of the pair of relations (8)–(9). The following theorem states that if E is bounded below, then the sequence $\{w^k\}_{k=0}^{\infty}$ generated by Algorithm 1 converges to a point w^* for which $\nabla E(w^*) = 0$.

THEOREM 2. *Suppose that the error function $E : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and bounded below on $\mathbb{R}^n$ and 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 Algorithm 1, satisfying the Wolfe conditions (8)–(9) implies that* $\lim\limits_{k\to\infty} \nabla E(w^k) = 0$.

Proof. The sequence $\{w^k\}_{k=0}^{\infty}$ follows the direction

$$\varphi^k(w^k) = -\operatorname{diag}\{1/\Lambda_1^k, \ldots, 1/\Lambda_n^k\}\nabla E(w^k),$$

which is a descent direction since

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

Moreover, the restriction on the angle θ_k is fulfilled since, as it can be easily justified utilizing Relation (10), $\cos\theta_k > 0$. Thus, by Theorem 1 holds that $\lim\limits_{k\to\infty} \nabla E(w^k) = 0$. 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. Moreover, the batch error function is always bounded below on $\mathbb{R}^n$.

6. Experimental Study

Next, we give quantitative results on three neural network applications applying the following methods:

(i) the batch BackPropagation with constant learning rate (BP) [19];
(ii) the Steepest Descent with Line Search for the learning rate proposed by Polak [16] (SDLS);
(iii) the batch BackPropagation with constant stepsize and Momentum (BPM) [19];
(iv) the Adaptive BackPropagation with heuristics (ABP) [21];
(v) the Fletcher–Reeves (FR) method [7];
(vi) the Polak–Ribiere (PR) method [7];
(vii) the Polak–Ribiere (PR) method constrained by the FR method (PR-FR) [7];
(viii) the MQprop method.

In the implementation of FR, PR, and PR-FR, the Armijo line search proposed by Polak [16] has been used. In Step 5 of the MQprop method (see Algorithm 1), a simple backtracking strategy has been used in the tuning subprocedure, i.e. η is reduced by a factor $1/q$, where $q = 2$ [13]. For all cases, the results are exhibited in terms of the average number of iterations (μ_{IT}) required to obtain a local minimum, the average number of gradient and function evaluations (μ_{FE}) and the number of successful runs out of 1000 (Success).

The selection of initial points is very important in FNN training. Very small initial values lead to very small corrections of the variables so that eventually some variables remain practically unchanged and more iterations are necessary to train

Table I. Results for the XOR problem, ($n = 9$).

Algorithm	μ_{IT}	μ_{FE}	Success
BP	549	1098	810/1000
SDLS	64	435	810/1000
BPM	803	1606	810/1000
ABP	157	314	810/1000
FR	84	282	130/1000
PR	21	169	380/1000
PR-FR	22	171	410/1000
MQprop	52	234	810/1000

the network [19]. In the worst case, the learning may stop in an undesired local minimum. On the other hand, very large initial values can speed up the learning process but in many cases they can lead neurons to saturation and generate very small gradient values. In such cases, learning is considerably slow [11]. A well known initialization heuristic for FNNs is to select the points with uniform probability from an interval $(w_{\min}, w_{\max})$, where usually $w_{\min} = -w_{\max}$. A common choice is the interval $(-1, +1)$. Thus, 1000 initial starting points have been randomly selected from this interval to test the different methods.

EXAMPLE 1. *The XOR problem* [9, 19]. The classification of the four XOR patterns in two classes is an interesting problem because it is sensitive to initial points as well as to learning rate variations, and presents a multitude of local minima. The patterns are classified using an 2-2-1 FNN with 9 variables.

The termination condition for all algorithms tested is to find a local minimizer with batch error function value $E \leqslant 0.04$. The results are summarized in Table I.

In this case the number of successful runs is related to the local minima problem. Thus FR, PR and PR-FR usually converge to an undesired local minimum, i.e. a minimizer with function value $E > 0.04$ which means that some of the patterns are not correctly classified. MQprop exhibits better performance than FR, PR and PR-FR with regards to the number of successful runs. MQprop also outperforms BP, SDLS, BPM and FR in training speed, measured by the mean number of function and gradient evaluations needed to successfully classify the patterns. Note that PR and PR-FR require less function evaluations than MQprop but they reveal a smaller number of successful runs. It is worth noticing that the classical Qprop without heuristics fails to converge in this experiment.

EXAMPLE 2. *Texture classification problem* [12]. A total of 12 Brodatz texture images [1]: 3, 5, 9, 12, 15, 20, 51, 68, 77, 78, 79, 93 (see Figure 1 in [12] of size

Table II. Results for the texture classification problem, ($n = 244$)

Algorithm	μ_{IT}	μ_{FE}	Success
BP	15839	31678	960/1000
SDLS	13256	26517	965/1000
BPM	12422	24844	940/1000
ABP	560	1120	1000/1000
FR	1624	12674	250/1000
PR	140	810	990/1000
PR-FR	145	1005	996/1000
MQprop	406	1228	1000/1000

512×512 is acquired by a scanner at 150 dpi. From each texture image 10 subimages of size 128×128 are randomly selected, and the co-occurrence method, introduced by Haralick [8] is applied. In the co-occurrence method, the relative frequencies of gray–level pairs of pixels at certain relative displacements are computed and stored in a matrix. The combination of the nearest neighbor pairs at orientations 0°, 45°, 90° and 135° are used in the experiment. A set of 10 sixteenth-dimensional training patterns are created from each image. An 16-8-12 FNN with 244 variables is trained to classify the patterns to the 12 texture types.

Detailed results regarding the training performance of the algorithms are presented in Table II. The termination condition is a classification error $CE < 3\%$; that is the network classifies correctly 117 out of the 120 patterns. Again, in this experiment the heuristic free Qprop exhibits extremely poor convergence characteristics and it is not included in Table II.

The successfully trained FNNs are tested for their generalization capability, using test patterns from 20 subimages of the same size randomly selected from each image. To evaluate the generalization performance of the FNN the *max* rule is used, i.e. a test pattern is considered to be correctly classified if the corresponding output neuron has the greatest value among the output neurons. The average percentage of success for each algorithm is: BP = 90.0%; SDLS = 90.0%; BPM = 90.0%; ABP = 93.5%; FR = 92.0%; PR = 92.6%; PR − FR = 93.5%; MQprop = 94.0%.

PR exhibits the best performance in terms of the average number of gradient and error function evaluations required during the training phase. On the other hand ABP, and MQprop are more robust in the sense that they exhibit larger number of successes providing also good generalization capability.

EXAMPLE 3. *Numeric font learning problem* [12, 20]. This experiment refers to the training of a multilayer FNN with 460 variables in order to recognize an 8×8 pixel

Table III. Results for the numeric font learning problem, ($n = 460$).

Algorithm	μ_{IT}	μ_{FE}	Success
BP	14489	28978	660/1000
SDLS	12225	24454	990/1000
BPM	10142	20284	540/1000
ABP	1975	3950	910/1000
FR	620	3121	420/1000
PR	649	2124	960/1000
PR-FR	750	3473	1000/1000
MQprop	159	739	1000/1000

machine printed numerals from 0 to 9. The network has 64 input neurons and 10 output neurons representing 0 through 9. Numerals are given in a finite sequence $C = (c_1, c_2, \dots, c_p)$ of input–output pairs $c_p = (u_p, t_p)$ where u_p are the binary input vectors in $\mathbb{R}^{64}$ determining the 8×8 binary pixel and t_p are binary output vectors in $\mathbb{R}^{10}$, for $p = 1, \dots, 10$, determining the corresponding numerals. The termination condition is to locate a minimizer with function value less than or equal to 0.001. The results are summarized in Table III. It is clear that MQprop achieves faster training than all other methods tested. Note that the classical Qprop scheme without heuristic parameters did not converge.

7. Conclusions

In this Letter the convergence of the Qprop method has been considered. A modification of the classical Qprop algorithm has been presented and a strategy for alleviating the use of highly problem-dependent heuristic learning parameters that are necessary in order to secure the stability of the classical algorithm have been proposed. A new theorem that guarantees the convergence of the proposed modified Qprop has been proved. This modified Qprop scheme exhibits rapid convergence and provides stable learning and therefore, a greater possibility of good performance.

References

1. Brodatz, P.: *Textures – a Photographic Album for Artists and Designers.* Dover, New York, 1966.
2. Broyden, C. G.: A class of methods for solving nonlinear simultaneous equations. *Math. Comp.* **19** (1965), 577–593.

3. Dennis, J. E. Jr and Moré, J.: A characterization of superlinear convergence and its applications to quasi Newton methods. *Math. Comp.* **28** (1974), 577–593.
4. Dennis, J. E. Jr and Schnabel, R. B.: A view of unconstrained optimization. In: G. L. Nemhauser et al., (eds), *Handbooks in OR & MS, Vol. 1.* Elsevier Science Publishers, 1989.
5. Dennis, J. E. Jr and Schnabel, R. B.: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* SIAM, Philadelphia, 1996. (Originally published: Prentice Hall, Inc., New Jersey, 1983.).
6. Fahlman, S. E.: Faster-learning variations on back–propagation: an empirical study. In: D. S. Touretzky, G. E. Hinton and T. J. Sejnowski, (eds), *Proc. 1988 Connectionist Models Summer School.* Morgan Kaufmann, San Mateo, CA, 1988, pp. 38–51.
7. Gilbert J. C. and Nocedal. J.: Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optimization* **2** (1992), 21–42.
8. Haralick, R., Shanmugan, K. and Dinstein, I.: Textural features for image classification. *IEEE Trans. System, Man and Cybernetics* **3** (1973), 610–621.
9. Jacobs, R. A.: Increased rates of convergence through learning rate adaptation. *Neural Networks* **1** (1988), 295–307.
10. Lee, Y., Oh, S. H. and Kim, M. W.: An analysis of premature saturation in backpropagation learning. *Neural Networks* **6** (1993), 719–728.
11. Magoulas, G. D., Vrahatis, M. N. and Androulakis, G. S.: A new method in neural network supervised training with imprecision. *Proc. IEEE 3rd Int. Conf.Electronics, Circuits and Systems,* 1996, 287–290.
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* **11** (1999), 1769–1796.
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, New York, 1970.
16. Polak, E.: *Optimization: Algorithms and Consistent Approximations.* Springer-Verlag, New York, 1997.
17. Rigler, A. K., Irvine, J. M. and Vogl, T. P.: Eescaling of variables in backpropagation learning. *Neural Networks* **4** (1991), 225–229.
18. 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.* MIT Press, Cambridge, Massachusetts, 1, 1986, pp. 318–362.
19. Sperduti, A. and Starita, A.: Speed up learning and network optimization with extended back-propagation. *Neural Networks* **6** (1993), 365–383.
20. 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.
21. Wolfe, P.: Convergence conditions for ascent methods. *SIAM Review* **11** (1969), 226–235.
22. Wolfe, P.: Convergence conditions for ascent methods. II: Some corrections. *SIAM Review* **13** (1971), 185–188.