Globally Convergent Algorithms with Local Learning Rates

George D. Magoulas, Vassilis P. Plagianakos and Michael N. Vrahatis

Abstract— **In this paper a new generalized theoretical result is presented that underpins the development of globally convergent first–order batch training algorithms which employ local learning rates. This result allows us to equip algorithms of this class with a strategy for adapting the overall direction of search to a descent one. In this way, a decrease of the batch–error measure at each training iteration is ensured, and convergence of the sequence of weight iterates to a local minimizer of the batch error function is obtained from remote initial weights. The effectiveness of the theoretical result is illustrated in three application examples by comparing two well known training algorithms with local learning rates to their globally convergent modifications.**

Keywords— **Globally convergent algorithms, local learning rate adaptation, batch training, gradient descent, backpropagation networks, Quickprop, Silva–Almeida method, endoscopy.**

I. Introduction

The issue of changing the learning rates dynamically during training has been widely investigated and several strategies for learning rate adaptation have been proposed so far. The use of these strategies aims at finding the proper learning rate that compensates for a small magnitude of the gradient in a flat region and dampens a large weight changes in a highly deep region. To this end, the literature suggests, for example, to:

(i) start with a small learning rate, η^0, and increase it at the next iteration, $k+1$, if successive iterations reduce the error, or rapidly decrease it if a significant error increase occurs [2], [29];

(ii) start with a small η^0 and increase it at the $k+1$ iteration, if successive iterations keep gradient direction fairly constant, or rapidly decrease it if the direction of the gradient varies greatly [4];

(iii) use a *local* learning rate for each weight $w_i^k \in \mathbb{R}^n (i = 1, 2, \ldots, n)$, i.e. $\eta_1^k, \eta_2^k, \ldots, \eta_n^k$, which increases if the successive corrections of the weights are in the same direction and decreases otherwise [8], [19], [23], [27].

This paper focuses on the last approach and particularly on the special class of first–order adaptive training algorithms that employ local learning rates. These algorithms employ heuristic strategies to adapt the learning rates at each iteration and require fine tuning additional problem–dependent learning parameters that help to ensure subminimization of the error func tion along each weight direction. Nevertheless, no guarantee is provided that the network error will monotonically decrease at each iteration and that the weight sequence will converge to a minimizer of the batch error function E. To alleviate this situation, we present in this paper a new generalized theoretical result that underpins the development of globally convergent training algorithms, i.e. algorithms with the property that starting from almost any initial weight vector the sequence of the weights will converge to a local minimizer of the error function.

Note that, as stated in [5, p.5], the term globally convergent algorithm is used "to denote a method that is designed to converge to a *local* minimizer of a nonlinear function, *from almost any starting point*". Dennis and Schnabel also note that "it might be appropriate to call such methods *local* or *locally convergent*, but these descriptions are already reserved by tradition for another usage". Additionally, in [17, p.200], Nocedal defines a globally convergent algorithm as an algorithm with iterates that converge from a remote starting point. Thus, in this context, global convergence is totally different from global optimization. In a strict mathematical sense, global optimization means to find the complete set of the globally optimal solutions (global minimizers) x^* of the objective function f, and the associated global optimum value $f^* = f(x^*)$ (for analytical tractability reasons, it is assumed that x^* is at most countable). So in this paper, we do not seek global minimizers of the error function E, but we are interested in developing algorithms that will converge to a local minimizer with certainty. The interesting topic of finding global minimizers in training neural networks is described elsewhere [20], [21], [22], [28].

The paper is organized as follows. In Section II local learning rate training algorithms are presented, and their advantages and disadvantages are discussed. The proposed approach and the corresponding theoretical convergence result are presented in Section III. In order to illustrate the effectiveness of this approach, two algorithms of this class and their globally convergent modifications are comparatively evaluated. Experiments and corresponding results are reported in Section IV. Finally, Section V presents concluding remarks.

II. Local Learning Rate Adaptation Strategies

Developments in training algorithms with local learning rates are mainly motivated by the need to train neural networks in situations when a learning rate appropriate for one weight direction is not necessarily appropriate for other directions [9]. Moreover, in certain cases a learning rate may not be appropriate for all of the portions of the er-

G.D. Magoulas is with the Department of Information Systems and Computing, Brunel University, West London, UB8 3PH, United Kingdom. E-mail: George.Magoulas@brunel.ac.uk

V.P. Plagianakos and M.N. Vrahatis are with the Department of Mathematics and UP Artificial Intelligence Research Center, University of Patras, Patras, GR-261.10, Greece. E-mail: {vpp,vrahatis}@math.upatras.gr

ror surface. To this end, a common approach to avoid slow convergence in flat directions and oscillations in steep directions, as well as to exploit the parallelism inherent in the evaluation of the error, $E(w)$, and its gradient, $\nabla E(w)$, by the Back-Propagation (BP) algorithm, consists of using a different adaptive learning rate for each direction in weight space.

Batch–type BP training algorithms of this class, [6], [8], [19], [23], [27], follow the iterative scheme

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

and try to decrease the error by searching a local minimum with small weight steps. These steps are usually constrained by problem–dependent heuristic parameters in order to avoid oscillations and ensure subminimization of the error function along each weight direction. This fact usually results in a trade–off between the convergence speed and the stability of the training algorithm. For example, the *delta-bar-delta* method [8], the Silva and Almeida's method [27] and the *Quickprop* method [6] introduce additional problem–dependent heuristic learning parameters to alleviate the stability problem. A common approach, used for example in the *Rprop* algorithm [23] and in the *BP with adaptive learning rate for each weight* [14], is to employ *learning rate lower and upper bounds* which are chosen heuristically and help to avoid the usage of an either extremely small or large learning rate component, which might misguide the resultant search direction. The learning rate lower bound helps to avoid unsatisfactory convergence rate while the learning rate upper bound limits the influence of a large learning rate component on the resultant search direction and depends on the shape of the error function.

However, the use of additional heuristics for local learning rates tuning may affect the overall adaptive search direction if the values of the heuristics are not properly chosen. In such case, the training algorithms cannot exploit the global information obtained by taking into consideration all the directions; furthermore, it is theoretically difficult to guarantee that the weight updates will converge to a local minimizer of E [6], [8], [14], [19], [23], [27].

III. Global Convergence of Algorithms with Local Learning Rates

Training of multi–layer feedforward neural networks can be considered as a highly nonlinear minimization problem, involving sigmoid functions that have infinitely broad regions with arbitrary small derivative [3], [26].

First–order training algorithms that follow the iterative scheme (1) usually evaluate the local learning rates by means of heuristic procedures that exploit information regarding the history of the partial derivative of $E(w)$ with respect to the i-th weight and/or the history of each learning rate, depending on the algorithm. For example, the *Quickprop*, [6], performs independent secant steps in the direction of each weight [31], while the *Rprop* algorithm, [23], 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 (1) 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 changes. These are 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. $-\eta_i \partial_i E(w)$. This behavior results in decreasing the error along each direction by performing small steps in the weight space so as to ensure subminimization of the error function along each weight direction and, hopefully, leads to monotone error reduction along the resultant search direction. To this end, the problem–dependent heuristic learning parameters, which are employed, act as constraints on the length of the search step, or on the length of the subminimization steps. However, enforcing monotone error reduction at each iteration using inappropriate values for the heuristic learning parameters can considerably slow the rate of training, or even lead to divergence and to premature saturation, as has been observed in certain cases [12], [14], [24]. Moreover, it seems that using heuristics it is not possible to develop globally convergent algorithms and, thus, guarantee convergence to a local minimizer from any initial condition [5].

In the context of optimization theory, the issue of making an unconstrained minimization iterative scheme globally convergent is treated as will be described below. Suppose that $f : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$ is the objective function to be minimized, using the following iterative scheme

$$x^{k+1} = x^k + \alpha^k d^k, \qquad (2)$$

where d^k is a descent direction and α^k is the step–length obtained by means of a one–dimensional line search that satisfies the Wolfe conditions [32], [33]

$$f(x^k + \alpha^k d^k) - f(x^k) \leqslant \sigma_1 \alpha^k \nabla f(x^k)^\top d^k, \qquad (3)$$

$$\nabla f(x^k + \alpha^k d^k)^\top d^k \geqslant \sigma_2 \nabla f(x^k)^\top d^k, \qquad (4)$$

where $\nabla f(x)$ is the gradient of f at x, and $0 < \sigma_1 < \sigma_2 < 1$. Then, the following theorem, due to Wolfe [32], [33] and Zoutendijk [34], can be used to obtain global convergence results.

Theorem 1 ([32], [34]) Suppose that $f : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$ is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in a neighborhood $\mathcal{N}$ of the level set $\mathcal{L} = \{x : f(x) \leqslant f(x^0)\}$, where x^0 is the starting point of the iterative scheme (2). Assume also that the gradient is Lipschitz continuous, i.e. there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|,$$

for all $x, y \in \mathcal{N}$. Then the Zoutendijk condition

$$\sum_{k \geqslant 1} \cos^2 \theta_k \left\|\nabla f(x^k)\right\|^2 < \infty, \qquad (5)$$

where

$$\cos\theta_k = \frac{-\nabla f(x^k)^\top d^k}{\|\nabla f(x^k)\|\, \|d^k\|}, \qquad (6)$$

is fulfilled.

Remark 1. Suppose that an iterative scheme of the form (2) follows a descent direction d^k, which does not tend to be orthogonal to the gradient $\nabla f(x^k)$, for which

$$\cos\theta_k \geqslant \zeta > 0,$$

for all k. Then, from the Zoutendijk condition (5), holds that

$$\lim_{k\to\infty} \|\nabla f(x^k)\| = 0, \tag{7}$$

which means that the sequence of gradients converges to zero.

For an iterative scheme of the form (2), the limit (7) is the best type of global convergence result that can be obtained (see [17] for a detailed discussion). From the above, it is evident that no guarantee is provided that the iterative scheme (2) will converge to a global minimizer, x^*, but only that it possesses the global convergence property [5], [17] to a local minimizer.

In neural network training, the sum–of–squared–differences error function E is bounded from below, since $E(w) \geqslant 0$. For a finite set of training patterns and a fixed network architecture, if w^* exists such that $E(w^*) = 0$, then w^* is a global minimizer. Otherwise, the weight vector w with the smallest error function value is the global minimizer.

For neural networks with *smooth enough* activation functions (the derivatives of at least order p are available and continuous), such as the well known hyperbolic tangent, the logistic activation function etc., the error function E is also smooth enough.

Although it is possible to verify the assumptions of Theorem 1, in neural network training this task is considered to be computationally expensive for large networks, and in practice is omitted.

In general, any batch–type BP training algorithm of the form (2) can be made globally convergent if

(i) the adopted search direction d^k is a descent direction and it does not tend to be orthogonal to the gradient direction ($\cos\theta_k$ in Relation (6) should be positive)

(ii) the learning rate α^k satisfies the two Wolfe conditions (3)–(4). Notice that, since d^k is a descent direction and E is continuously differentiable and bounded below along the radius $\{w^k + \alpha d^k \mid \alpha > 0\}$, then there always exist α^k satisfying the Wolfe's conditions (3) and (4) [5], [17].

For example, the well known batch BP algorithm that employs the steepest descent method with a common learning rate for all weights that satisfies the Wolfe conditions (3)–(4) is globally convergent because in this case we have $\cos\theta_k = 1 > 0$.

With regards to batch–type BP algorithms with a different learning rate for each weight (local learning rate), no strategy is available, to the best of our knowledge, to make these methods globally convergent. Below we present a strategy that ensures the search direction followed is, indeed, a descent one, and a theoretical result for globally convergent local learning rate algorithms. It is important to emphasize that this result is independent of the local learning rates adaptation procedure, and can be used to prove convergence for any batch–type training algorithm that adopts the strategy:

(i) define $(n-1)$, say $\{\eta_1, \eta_2, \ldots, \eta_{i-1}, \eta_{i+1}, \ldots, \eta_n\}$, out of the n learning rates, $\{\eta_1, \eta_2, \ldots, \eta_n\}$, as computed directly by an adaptive learning rate evaluation procedure, and

(ii) calculate the remaining one, say η_i, analytically using the values of the others, $\{\eta_1, \eta_2, \ldots, \eta_{i-1}, \eta_{i+1}, \ldots, \eta_n\}$, as it will be shown below.

Next we present a theoretical result that applies to adaptive training algorithms with local learning rates.

Corollary 1. Suppose that the conditions of Theorem 1 on $f(x)$ and $\nabla f(x)$ are also true for the error function $E(w)$ and its gradient $\nabla E(w)$. Then, for a given point $w^0 \in \mathbb{R}^n$, the sequence $\{w^k\}_{k=0}^{\infty}$ which is generated by the iterative scheme:

$$w^{k+1} = w^k + \alpha^k\, d^k, \tag{8}$$

where $\alpha^k > 0$ satisfies the Wolfe's conditions (3)–(4), $d^k = -\mathrm{diag}\{\eta_1^k, \ldots, \eta_i^k, \ldots, \eta_n^k\}\, \nabla E(w^k)$ denotes the search direction, η_m^k for $m = 1, 2, \ldots, i-1, i+1, \ldots, n$ are arbitrarily chosen small positive learning rates, and

$$\eta_i^k = -\frac{\delta}{\partial_i E(w^k)} - \frac{1}{\partial_i E(w^k)} \sum_{\substack{j=1 \\ j\neq i}}^{n} \eta_j^k\, \partial_j E(w^k), \tag{9}$$

$$0 < \delta \ll \infty \qquad \text{and} \qquad \partial_i E(w^k) \neq 0,$$

is globally convergent to a local minimizer.

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

$$d^k = -\mathrm{diag}\{\eta_1^k, \ldots, \eta_i^k, \ldots, \eta_n^k\}\, \nabla E(w^k),$$

which is a descent direction if $\eta_m^k,\ m = 1, 2, \ldots, i-1, i+1, \ldots, n$ are arbitrarily chosen learning rates (positive real numbers) and η_i^k is given by Relation (9), since

$$\nabla E(w^k)^\top d^k < 0.$$

Moreover, in our case Relation (6) becomes

$$\cos\theta_k = \frac{-\nabla E(w^k)^\top d^k}{\|\nabla E(w^k)\|\, \|d^k\|} > 0. \tag{10}$$

Thus, from the previous discussion it is evident that the sequence $\{w^k\}_{k=0}^{\infty}$ is globally convergent to a local minimizer. ■

In Relation (9), we choose a coordinate direction with no zero partial derivative. Of course always exists such a direction; otherwise we would have found a minimizer already. In addition, the parameter δ, $0 < \delta \ll \infty$, is introduced to alleviate problems with limited precision that may occur in simulations. It should take a small value proportional to

the square root of the relative machine precision. In the experiments reported in the next section the value $\delta = 10^{-6}$ has been used in an attempt to test the convergence accuracy of the proposed strategy.

With regards to the parameter α^k, the use of $\alpha^k = 1$, for all k, is suggested in practical applications. This has the effect that the minimization step along the resultant search direction is explicitly defined by the values of the local learning rates. The length of the minimization step can be regulated through α^k tuning so that the Wolfe's conditions are satisfied and the weights are updated in a descent direction. To this end, a simple backtracking strategy could be used to decrease α^k by a reduction factor $1/q$, where $q > 1$. This has the effect that α^k is decreased by the largest number in the sequence $\{q^{-m}\}_{m=1}^{\infty}$ [18]. 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 Wolfe's conditions. A value of $q = 2$ is generally suggested in the literature [1], [18] and, indeed, it has been found to work without problems in our experiments.

In reference to the Wolfe conditions (3)–(4), Inequality (3) ensures that the error is reduced sufficiently, while Inequality (4) prevents the minimization step from becoming too small. Consequently, when seeking to satisfy Condition (3) it is important to ensure that α^k is not reduced unnecessarily so that Condition (4) is not satisfied. However, at the k-th training epoch the gradient vector is only known at the beginning of the iterative search for a new set of weights, w^{k+1}. Thus, Condition (4) cannot be checked directly, as this task would require additional gradient evaluations at each iteration of the training algorithm. This problem can be easily tackled (see [5]) by replacing Inequality (4) with relation

$$E(w^k + \alpha^k d^k) - E(w^k) \geqslant \sigma_2 \alpha^k \nabla E(w^k)^\top d^k, \qquad (11)$$

and, thus, avoid the computationally expensive backward passes.

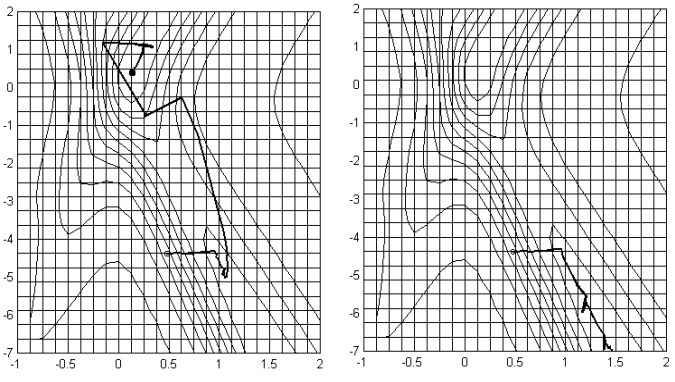

Fig. 1. Illustration of the Quickprop method for training a node with two weights. The modification converges to the desired minimum (left), while the classic method converges to an undesired local minimum (right).

At this point, it is useful to illustrate the behavior of the proposed strategy by means of a simple example, which concerns the case of a single node with two weights and logistic activation function [13]. This minimal architecture is trained using the classic Quickprop method and its globally convergent modification, which uses a positive learning rate value η_1^k computed by the Quickprop formula and η_2^k given by Relation (9). Starting from the same initial conditions, the globally convergent modification successfully locates the feasible minimum (see Figure 1, left), while the classic Quickprop generates a discretized path in the weight space (see Figure 1, right) that leads to an undesired local minimum (undesired local minimizers are those having error function values higher than the desired error goal).

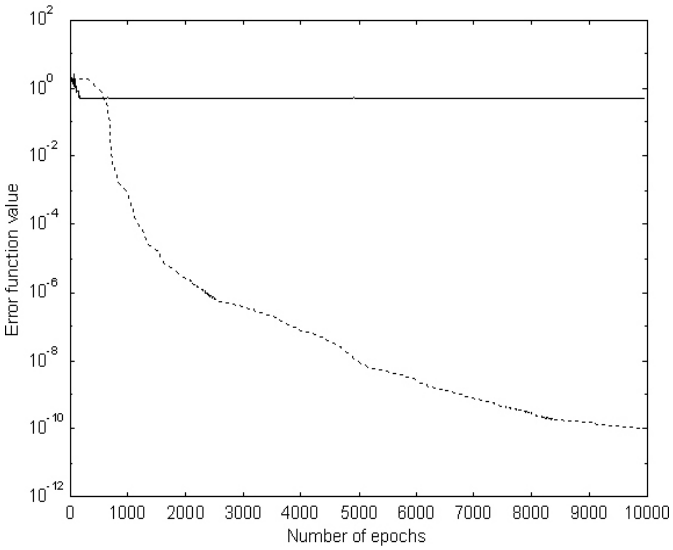

Fig. 2. Learning curves for the Quickprop method (solid line) and the globally convergent Quickprop (dotted line) for the 3–bit parity problem. Both methods start from the same initial conditions.

In a more difficult problem, learning the 3–bit parity [7], [25], a typical run for the Quickprop method and its globally convergent modification is shown in Figure 2. Starting with the same initial weights and learning parameters, the modified Quickprop (dotted line) successfully converges to a feasible solution ($E(w) \leqslant 10^{-10}$), while the original Quickprop (solid line) got stuck in a local minimum with higher error function value.

IV. Empirical Study

The proposed strategy has been incorporated in two first–order training algorithms, the Silva–Almeida method [27] and the Quickprop algorithm [6], to develop new globally convergent modifications. These modified schemes have been implemented and tested on different training problems and have been compared in terms of gradient and error function evaluations and rate of success with the original methods. Note that in training practice, a gradient evaluation is considered three times more costly than an error function evaluation for certain classes of problems [14], [15], [16].

Our experience from the simulations is that the proposed strategy behaves predictably and reliably. Below, we ex-

hibit results from 100 runs for the Silva–Almeida method (SA), the Quickprop algorithm (Qprop) and their globally convergent modifications in two applications, using initial weights from the interval $[-1, 1]$. The globally convergent modifications use the same initial values for the learning parameters and are tested under the same initial weight conditions as the original methods.

The heuristic learning parameter *maximum growth factor* of the Quickprop method has been set to the classical value $m = 1.75$. The *learning rate increment and decrement factors* of the Silva–Almeida method have been tuned appropriately and received the values $u = 1.02$ and $d = 0.5$, respectively. It is worth mentioning that Relation (9) has been employed cyclically over the local learning rates, i.e. at the k-th iteration $i = k \bmod n$, in all of the experiments reported in this paper.

In the first experiment, a network with 64 input, 6 hidden and 10 output nodes (444 weights, 16 biases) is trained to recognize 8×8 pixel machine printed numerals from 0 to 9 in helvetica italic [13]. The network is based on logistic activation neurons. The training performance of the algorithms is shown in Tables I and II, where the following notation is adopted: μ is the mean number of gradient or error function evaluations, denoted as GRD and EFE respectively, σ is the corresponding standard deviation, Min/Max denotes the minimum/maximum number of gradient or error function evaluations, D indicates that the algorithm diverged, and % is the percentage of simulations that converge to a desired minimum.

The termination conditions for all algorithms tested were: $E \leqslant 10^{-1}$, in the first case (Table I), and $E \leqslant 10^{-2}$, in the second case (Table II). For both cases, a limit of 5000 error function evaluations was set and the algorithms were tested under the same initial conditions (learning parameters and weights).

As shown in Table I, the Globally convergent Quickprop (G–Qprop) is faster and more reliable than the original method, which fails to converge in all cases. In the same problem, the Silva–Almeida method, although it is faster than the globally convergent modification (G–SA), fails to converge in 43 out of the 100 runs, due to convergence to undesired local extrema (points that possess error function values higher than the desired error goal). The algorithm exhibits stability problems because the learning rates increase exponentially when many iterations are performed successively. This behavior results in minimization steps that increase some weights by large amounts, pushing the outputs of some neurons into saturation and consequently into convergence to a local minimum or maximum. On the other hand, the globally convergent modification overcomes these problematic situations by exploiting the analytic evaluation of the i-th learning rate. For the globally convergent modifications, the 1 percent (1%) of failure represents runs that the algorithm failed to converge to a desired local minimum within the maximum allowed number of error function evaluations.

By comparing the results of Tables I and II one can see that the globally convergent modifications of the tested algorithms exhibit consistent behavior. On the other hand, the performance of the original Silva–Almeida method is getting worse as the accuracy of the required solution increases. Note that the results of Tables I and II have been produced by using the same initial weights for all the algorithms tested; only the desired error goal varied from an $E \leqslant 10^{-1}$ to an $E \leqslant 10^{-2}$ in the two sets of experiments.

TABLE I
RESULTS FOR THE NUMERIC FONT LEARNING PROBLEM ($E \leqslant 10^{-1}$)

Algorithm		μ	σ	Min/Max	%
SA	GRD	124.21	10.56	109/151	57
	EFE	124.21	10.56	109/151	
G–SA	GRD	403.21	114.35	145/798	99
	EFE	711.92	298.40	148/1428	
Qprop	GRD	D	D	D	0
	EFE	D	D	D	
G–Qprop	GRD	82.42	77.30	26/485	99
	EFE	172.31	202.85	26/1023	

Thus, the adaptive learning rate schedule helps SA to converge very fast in 26 of the runs, but fails to allow the method to reach the local minimizer with accuracy. Thus, as shown in Table II, the success percentage of the Silva–Almeida method reduces as the method gets stuck to undesired local minima. A typical run for the Silva–Almeida method and its globally convergent modification is shown in Figure 3. In this case, the networks were trained exhaustively to reach an error value $E \leqslant 10^{-2}$ starting from the same initial weights and learning parameters. The modified Silva–Almeida (dotted line) successfully converges to a feasible solution ($E(w) \cong 10^{-5}$), while the original Silva–Almeida (solid line) gets stuck to a local minimum with higher error function value.

TABLE II
RESULTS FOR THE NUMERIC FONT LEARNING PROBLEM ($E \leqslant 10^{-2}$)

Algorithm		μ	σ	Min/Max	%
SA	GRD	218.23	9.77	204/237	26
	EFE	218.23	9.77	204/237	
G–SA	GRD	712.92	235.48	335/1526	99
	EFE	1423.41	796.50	335/4556	
Qprop	GRD	D	D	D	0
	EFE	D	D	D	
G–Qprop	GRD	160.06	147.20	35/641	100
	EFE	372.36	443.81	35/1778	

In the second experiment, the continuous function $f(x) = \sin(x)\cos(2x)$ is approximated by a 1-15-1 neural network (thirty weights, sixteen biases) using 20 input/output pairs, scattered in the interval $[0, 2\pi]$. The termination condition is $E \leqslant 0.1$ within 10000 error function evaluations, and the network is based on hidden neurons with hyperbolic tangent activations and on a linear output neuron. Comparative results are exhibited in Ta-

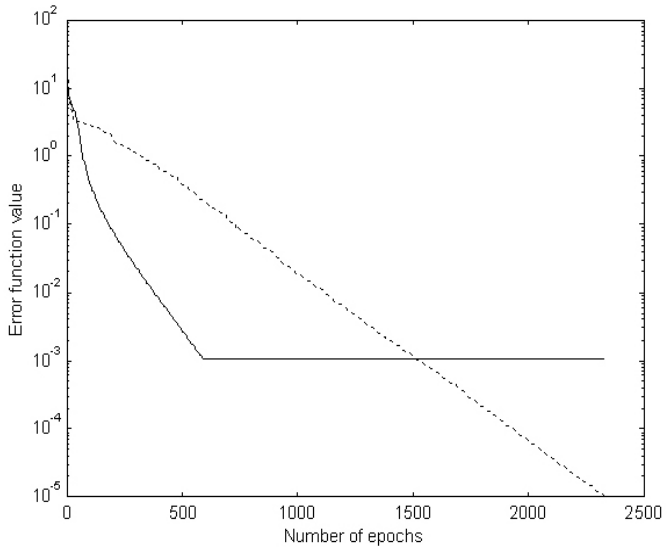

Fig. 3. Learning curves for the Silva–Almeida method (solid line) and the globally convergent Silva–Almeida (dotted line) for the numeric font learning problem. Both methods start from the same initial conditions.

ble III. The SA method exhibits the highest percentage of failure due to convergence to undesired local extrema; the method converges only 11 times (see Table III), although the best available values for the heuristics have been used. In the same problem, the G–Qprop outperforms the original method in the number of successful runs (99% success). On the other hand, the original Quickprop method succeeded only in 27 runs due to entrapment in neighborhoods of undesired local minima. Although, training was allowed for 10000 iterations (the worst case run took 953 epochs to converge), this didn't help Qprop to escape from the undesired minima.

In additional simulations, keeping initial conditions the same and changing only the desired error goal to an $E \leqslant 10^{-2}$, the algorithms exhibited behavior similar to the one of the numeric font learning problem, whilst the globally convergent algorithms exhibited consistent convergence behavior and 100% success.

TABLE III
COMPARATIVE RESULTS FOR THE FUNCTION APPROXIMATION PROBLEM

Algorithm		μ	σ	Min/Max	%
SA	GRD	23.11	116.18	84/150	11
	EFE	23.11	116.18	84/150	
G–SA	GRD	352.44	105.21	48/764	99
	EFE	688.26	197.02	48/2354	
Qprop	GRD	362.81	268.55	58/953	27
	EFE	362.81	268.55	58/953	
G–Qprop	GRD	176.51	119.98	40/694	99
	EFE	252.10	179.31	50/1033	

The results of the two experiments reported above provide empirical evidence verifying that the proposed strategy performs in practice reasonably well in different types of problems. The globally convergent modifications of the tested algorithms provide stable learning and therefore a greater possibility of good performance in terms of successfully finding a local minimizer; however, in the case of the Silva–Almeida method its globally convergent modification requires additional iterations to converge.

To investigate how the distributions of initial weights affect the success of the learning process in large networks, we have conducted a third experiment. A 16–40–2 (720 weights and 42 biases) network with logistic activations has been trained to detect two different types of abnormalities in colonoscopy images taken from two different colons. Image 1 (Figure 4-top left) is considered histologically as a low grade cancer (a Type–IIIs lesion macroscopically [11]). Image 2 (Figure 4-top right) is considered histologically as a moderately differentiated carcinoma (a Type-V lesion macroscopically). Textures from 10 normal and 10 abnormal tissue samples have been randomly chosen from each image and used for training the network to discriminate between malignant and normal regions with 3% classification error (see [10] for further technical details). We have used 100 initial weight sets generated randomly from uniform distributions within six different intervals, and we have trained the networks with the original and the Globally convergent Quickprop.

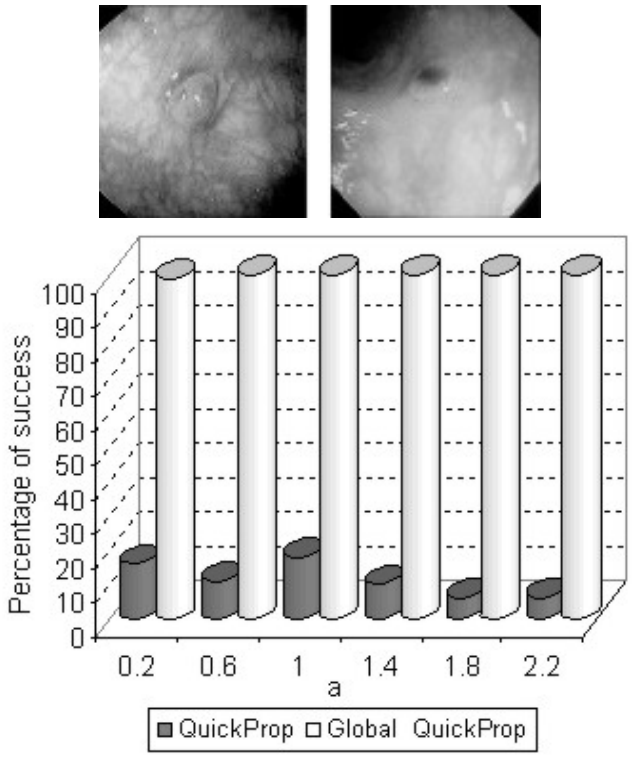

Fig. 4. Colonoscopy images used in the experiments (top) and percentage of success with respect to different initial weights ranges (bottom).

Figure 4 (bottom) shows a plot of the average percentage of success with respect to the six initial weight ranges

$(-a, a)$, where $a \in \{0.2, 0.6, 1, 1.4, 1.8, 2.2\}$, for simulations that reached a desired minimum out of 100 runs. It is evident that the G–Qprop outperforms the original method. According to our experience, the improvement obtained justifies, especially in certain real-life applications, the additional programming effort for the implementation of the proposed strategy.

V. Concluding Remarks

A theoretical result that underpins the development of globally convergent batch training algorithms with local learning rates has been proposed in this paper. This result allows us to provide conditions under which global convergence is guaranteed and introduce a strategy for adapting the search direction and tuning the length of the minimization step. Two well known training algorithms with local learning rates have been equipped with the proposed strategy to illustrate its applicability. Their modified versions exhibit significantly better percentage of success in reaching local minimizers than the original methods, but they may require additional error function and gradient evaluations depending on the algorithm, as has been observed with the Globally convergent Silva–Almeida method.

Nevertheless, the issue of developing techniques that will choose the appropriate i-th local learning rate, η_i^k, to be calculated by Relation (9), as well as the optimal value of δ should be investigated further to fully exploit the potential of the suggested strategy, and improve the convergence speed of the globally convergent algorithms.

Acknowledgments

The authors would like to thank the TNNL editor, and the referees for constructive comments and useful suggestions that helped to improve the paper.

References

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

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

[3] R. Battiti, "First- and second-order methods for learning: between steepest descent and Newton's method", *Neural Computation*, 4, 141–166, 1992.

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

[5] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and nonlinear equations*, SIAM, Philadelphia, 1996.

[6] S.E. Fahlman, "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] M.E. Hohil, D. Liu, and S.H. Smith, "Solving the N–bit parity problem using neural networks", *Neural Networks*, 12, 1321–1323, 1999.

[8] R.A. Jacobs, "Increased rates of convergence through learning rate adaptation", *Neural Networks*, 1, 295–307, 1988.

[9] K. Joe, Y. Mori and S. Miyake, "Construction of a large scale neural network: simulation of handwritten Japanese character recognition on ncube", *Concurrency*, 2, 79–107, 1990.

[10] S. Karkanis, G.D. Magoulas and N. Theofanous, "Image Recognition and Neuronal Networks: Intelligent Systems for the Improvement of Imaging Information", *Minimally Invasive Therapy and Allied Technologies*, 9, 225–230, 2000.

[11] S. Kudo, *Early Colorectal Cancer*, Igaku–Shoin Publischers, Tokyo, 1996.

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

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

[14] G.D. Magoulas, M.N. Vrahatis and G.S. Androulakis, "Improving the convergence of the back–propagation algorithm using learning rate adaptation methods", *Neural Computation*, 11, 1769–1796, 1999.

[15] M. F. Møller, "A scaled conjugate gradient algorithm, for fast supervised learning", *Neural Networks*, 6, 525–533, 1993.

[16] E. Mizutani and S. E. Dreyfus, "On complexity analysis of supervised MLP-learning for algorithmic comparisons" in *Proc. of the International Joint Conf. on Neural Networks*, Washington, DC, USA, 347–352, 2001.

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

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

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

[20] V.P. Plagianakos, G.D. Magoulas and M.N. Vrahatis, "Learning in multilayer perceptrons using global optimization strategies", *Nonlinear Analysis: Theory, Methods and Applications*, 47, 3431–3436, 2001.

[21] V.P. Plagianakos, G.D. Magoulas and M.N. Vrahatis, "Supervised training using global search methods", in N. Hadjisavvas and P. Pardalos (eds.), *Advances in Convex Analysis and Global Optimization, vol. 54, Noncovex Optimization and its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, Chapter 26, pp.421-432.

[22] V.P. Plagianakos, G.D. Magoulas and M.N. Vrahatis, "Improved learning of neural nets through global search", in Janos D. Pintér (ed.), *Global Optimization – Selected Case Studies*, edited volume of case studies in global (nonconvex) optimization, NOIA series, Kluwer Academic Publishers, to appear.

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

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

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

[26] S. Saarinen, R. Bramley, and G. Cybenko, "Ill–conditioning in neural network training problems", *SIAM Journal of Scientific Computing*, 14, 693–714, 1993.

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

[28] N.K. Treadgold and T.D. Gedeon, "Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm", *IEEE Transactions on Neural Networks*, 9, 662–668.

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

[30] M.N. Vrahatis, G.S. Androulakis, J.N. Lambrinos and G.D. Magoulas, "A class of gradient unconstrained minimisation algorithms with adaptive stepsize", *Journal of Computational and Applied Mathematics*, 114, 367-386, 2000.

[31] M.N. Vrahatis, G.D. Magoulas and V.P. Plagianakos, "Globally convergent modification of the quickprop method", *Neural Processing Letters*, 12, 159–169, 2000.

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

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

[34] G. Zoutendijk, "Nonlinear Programming, Computational Methods", in *Integer and Nonlinear Programming*, J. Abadie (ed.), 37–86, North–Holland, Amsterdam, 1970.