Development and Convergence Analysis of Training Algorithms with Local Learning Rate Adaptation

G.D. Magoulas[1,3], V.P. Plagianakos[2,3], M.N. Vrahatis[2,3]
(1)Department of Informatics, University of Athens, GR-157.84 Athens, Greece
e-mail: magoulas@di.uoa.gr
(2)Department of Mathematics, University of Patras, GR-261.10 Patras, Greece
e-mail: {vpp,vrahatis}@math.upatras.gr
(3)University of Patras Artificial Intelligence Research Center–UPAIRC

Abstract— **A new theorem for the development and convergence analysis of supervised training algorithms with an adaptive learning rate for each weight is presented. Based on this theoretical result, a strategy is proposed to automatically adapt the search direction, as well as the stepsize length along the resultant search direction. This strategy is applied to some well known local learning algorithms to investigate its effectiveness.**

Keywords and phrases: Globally convergent algorithms, local learning rate adaptation, batch training algorithms, gradient descent, feedforward neural networks.

1 Introduction

Supervised learning is a classical method to bring the weights of a neural network towards optimality. A finite set of arbitrarily ordered examples is presented at the input of the network and associated to appropriate references through an error correction process. Batch training, which is a special case of supervised learning, is consistent with the theory of unconstrained optimization. This can be viewed as the minimization of a batch error measure, which is usually defined as the sum-of-squared-differences error function E over the entire training set:

$$w^* = \min_{w \in \mathbb{R}^n} E(w), \tag{1}$$

where $w^* = (w_1^*, w_2^*, \ldots, w_n^*) \in \mathbb{R}^n$ is a minimizer of E. The rapid computation of such a minimizer is a rather difficult task since, in general, the number of network weights is high and the corresponding nonconvex error function possesses multitudes of local minima and has broad flat regions adjoined with narrow steep ones.

Let us consider the family of gradient–based supervised learning algorithms having the iterative form:

$$w^{k+1} = w^k + \eta^k d^k, \qquad k = 0, 1, 2, \ldots \tag{2}$$

where w^k is the current weight vector, d^k is a search direction, and η^k is a *global* learning rate, i.e. the same learning rate is used to update all the weights of the network. Various choices of the direction d^k give rise to distinct algorithms. A broad class of methods uses the search direction $d^k = -\nabla E(w^k)$, where the gradient $\nabla E(w)$ can be obtained by means of back–propagation of the error through the layers of the network [20]. The most popular training algorithm of this class, named batch Back–Propagation (BP), minimizes the error function using the steepest descent method [6] with constant, heuristically chosen, learning rate η. In practice, a small value for the learning rate is chosen ($0 < \eta < 1$) in order to secure the convergence of the BP training algorithm and to avoid oscillations in a direction where the error function is steep. It is well known that this approach tends to be inefficient. This happens, for example, when the search space contains long ravines that are characterized by sharp curvature across them and a gently slopping floor [8, 20]. Obtaining efficient convergence of BP training algorithms utilizing a constant learning rate is considered particularly difficult [9, 10]. On the other hand, there are theoretical results that guarantee the convergence when the learning rate is constant. In this case the learning rate is proportional to the inverse of the Lipschitz constant which, in practice, is not easily available [1, 12].

Alternatively, several heuristic methods have been suggested to dynamically adapt the global learning rate during training to accelerate the convergence [2, 3, 22]. A different approach is to exploit the local shape of the error surface as described by the direction cosines [7] or the local estimation of the Lipschitz constant [12].

This paper focuses on a special class of adaptive training algorithms that employ *local* learning rates, i.e. a different learning rate for each weight. A general theoretical result is presented that underlies the development of globally convergent training algorithms of this class, i.e. algorithms with the property that starting from any initial weight vector the sequence of the weights will converge to a local minimizer of the error function. The paper is organized as follows. In Section 2 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 3. Experiments are presented in Section 4 in order to evaluate and compare the performance of two algorithms of this class with their globally convergent modifications. Section 5 presents the conclusions.

2 Local learning rate adaptation strategies

Studying the sensitivity of a minimizer to small changes by approximating the error function quadratically, it is known that, in a sufficiently small neighborhood of w^*, the directions of the principal axes of the corresponding elliptical contours (n-dimensional ellipsoids) will be given by the eigenvectors of $\nabla^2 E(w^*)$, while the lengths of the axes will be inversely proportional to the square roots of the corresponding eigenvalues. Hence, a variation along the eigenvector corresponding to the maximum eigenvalue will cause the largest change in E, while the eigenvector corresponding to the minimum eigenvalue gives the least sensitive direction. Thus, in general, a learning rate appropriate in one weight direction is not necessarily appropriate for other directions. Moreover, it may not be appropriate for all the portions of a general error surface.

Thus, the fundamental algorithmic issue is to find the proper learning rate that compensates for the small magnitude of the gradient in the flat regions and dampens the large weight changes in highly deep regions. A common approach to avoid slow convergence in the flat directions and oscillations in the steep directions, as well as to exploit the parallelism inherent in the evaluation of $E(w)$ and $\nabla E(w)$ by the BP algorithm, consists of using a different learning rate for each direction in weight space. Various batch–type BP training algorithms with an adaptive learning rate for each weight have been suggested in the literature [5, 8, 16, 18, 21]. Following this approach equation (2) is reformulated to the following scheme:

$$w^{k+1} = w^k - \operatorname{diag}\{\eta_1^k, \ldots, \eta_n^k\} \nabla E(w^k). \tag{3}$$

The algorithms that follow the above scheme try to decrease the error by searching a local minimum with small weight steps. These steps are usually constraint by problem–dependent heuristic parameters in order to avoid oscillations and to ensure subminimization of the error function in 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] or the *Quickprop* method [5] introduce additional problem–dependent heuristic learning parameters to alleviate the stability problem. A common approach is to use heuristically chosen *learning rate lower and upper bounds* that would help to avoid the usage of an extremely small or large learning rate component, which misguides 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.

A well known difficulty of this approach is that the use of inappropriate heuristic values for a weight direction misguides the resultant search direction [13]. In these cases, the training algorithm cannot exploit the global information obtained by taking into consideration all the directions. This is the case of many well known training algorithms that employ additional heuristics for properly tuning the local learning rates [5, 8, 16, 18, 21] and no guarantee is provided that the weight updates will converge to a minimizer of E.

3 Global convergence by adapting the search direction

A batch BP algorithm with a different learning rate for each weight, as defined in relation (3), evaluates 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 ith weight and/or the history of each learning rate, depending on the algorithm. For example, the *Quickprop* [5] performs independent secant steps in the direction of each weight [23], while 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 equation (3) 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 in each direction by performing small steps in the weight space. These steps are usually constraint by problem–dependent heuristic parameters to ensure subminimization of the error function in each weight direction and hopefully obtain monotone error reduction. However, enforcing monotone error reduction using inappropriate values for the heuristic learning parameters can considerably slow the rate of training, or even lead to divergence and to premature saturation [11, 19]. Moreover, it seems that 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 will converge to a local minimizer of the error function.

To alleviate this situation, we propose that the search direction is obtained using any $n-1$ out of the n learning rate values that are directly computed by means of an adaptive learning rate strategy and analytically calculate the remaining one, using the values of the other $n-1$ learning rates. This approach has the effect that the search direction followed is, indeed, a descent one. The following theorem provides a global convergence result for local learning rate algorithms.

Theorem 1. *Suppose that: (a) the error function $E : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and (b) that ∇E is Lipschitz continuous on $\mathbb{R}^n$, i.e. for any two points w and $v \in \mathbb{R}^n$, ∇E satisfies the Lipschitz condition*

$$\|\nabla E(w) - \nabla E(v)\| \leq L\|w - v\|, \tag{4}$$

where $L > 0$ denotes the Lipschitz constant. Then, for any given point $w^0 \in \mathbb{R}^n$ and any sequence $\{w^k\}_{k=0}^{\infty}$, generated by the iterative scheme

$$w^{k+1} = w^k - \tau^k \operatorname{diag}\{\eta_1^k, \eta_2^k, \ldots, \eta_n^k\} \nabla E(w^k), \tag{5}$$

where $\eta_m^k,\ m = 1, 2, \ldots, i-1, i+1, \ldots, n$ are arbitrarily chosen positive real numbers,

$$\eta_i^k = -\frac{\delta}{\partial_i E(w^k)} - \frac{1}{\partial_i E(w^k)} \sum_{\substack{j=1 \\ j \neq i}}^{n} \eta_j^k\, \partial_j E(w^k), \qquad \delta > 0 \tag{6}$$

and $\tau^k > 0$ satisfies the Wolfe's conditions

$$E(w^{k+1}) - E(w^k) \leq \sigma_1 \tau^k \nabla E(w^k)^\top d^k, \tag{7}$$

$$\nabla E(w^{k+1})^\top \nabla E(w^k) \geq \sigma_2 \nabla E(w^k)^\top d^k, \qquad 0 < \sigma_1 < \sigma_2 < 1 \tag{8}$$

where d^k denotes the search direction, holds 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

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

which is a descent direction if $\eta_m^k,\ m = 1, 2, \ldots, i-1, i+1, \ldots, n$ are arbitrarily chosen positive real numbers and η_i^k is given by relation (6), since

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

Now, since d^k is a descent direction and E is continuously differentiable and bounded below along the radius $\{w^k + \tau d^k \mid \tau > 0\}$, then there always exist τ^k satisfying the Wolfe's conditions (7) and (8) [4, 14].

Moreover, the Wolfe's Theorem [4, 14] suggests that if the cosine of the angle θ_k between the descent direction d^k and the $-\nabla E(w^k)$ is positive then $\lim_{k\to\infty} \nabla E(w^k) = 0$. In our case

$$\cos\theta_k = \frac{-\nabla E(w^k)^\top d^k}{\|\nabla E(w^k)\| \|d^k\|} > 0. \tag{9}$$

Thus, the theorem is proved.

Remark 1: The effect of assumptions (a) and (b) is to place an upper bound on the degree of the nonlinearity of the error function and to ensure that the first derivatives are continuous.

Remark 2: Note that for neural networks with sigmoid activation functions the assumption of continuous differentiability of the error function is redundant.

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 values of the local learning rates. The length of the minimization step can be regulated through τ^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 τ^k by a reduction factor $1/q$, where $q > 1$. This has the effect that τ^k is decreased by the largest number in the sequence $\{q^{-m}\}_{m=1}^{\infty}$ [15]. 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, 15] and, indeed, it has been found to work without problems in our experiments.

It is worth noticing that the inequality (7) ensures that the error is reduced sufficiently, while the inequality (8) prevents the minimization step from becoming too small. Consequently, when seeking to satisfy the condition (7) it is important to ensure that τ^k is not reduced unnecessarily so that the condition (8) is not satisfied. In a training epoch the gradient vector is only known at the beginning of the iterative search for a new weight vector. Thus, the condition (8) cannot be checked directly, since this task would require additional gradient evaluations at each epoch of the training algorithm. This problem can be easily tackled (see [4]) by replacing relation (8) with

$$E(w^k + \tau^k d^k) - E(w^k) \geq \sigma_2 \tau^k \nabla E(w^k)^\top d^k \tag{10}$$

and thus avoid the computationally expensive backward passes.

Next, we visualize the behavior of the proposed strategy by means of a simple example, which concerns the case of a single neuron with two weights and logistic activation function [12]. This minimal architecture is trained using the classical Quickprop method and its globally convergent modification, which uses the absolute value of the learning rate η_1^k computed by the Quickprop method (in order to satisfy Theorem 1) and η_2^k is given by relation (6). The classical Quickprop formula (see Figure 1, left) generates a discretized path in the weight space that leads to an undesired local minimum. On the other hand, the globally convergent modification (see Figure 1, right) successfully locates the desired minimum.

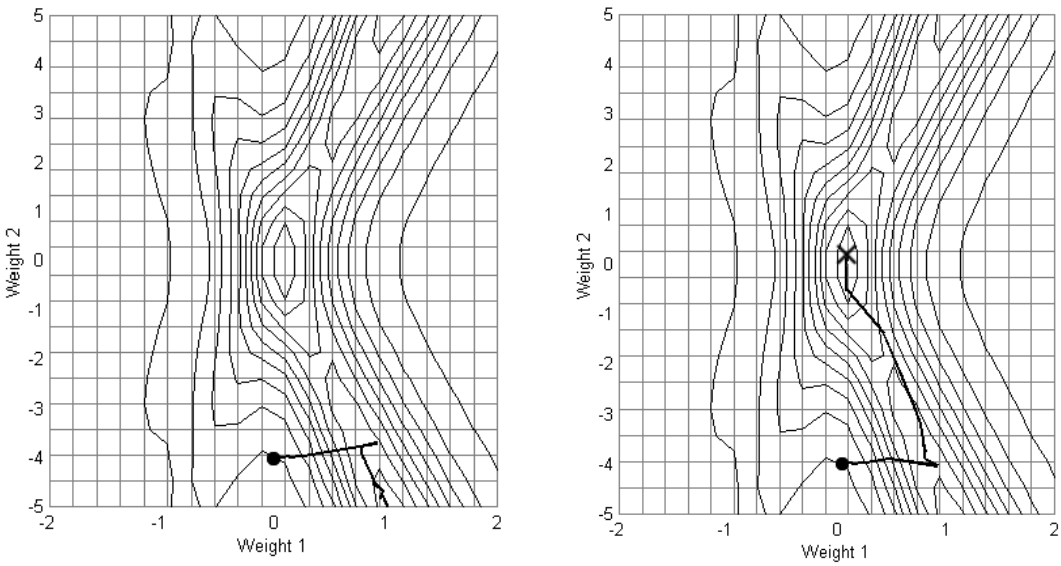

Figure 1: Illustration of the Quickprop method for training a network with two weights. The classical method converges to an undesired local minimum (left), while the modification to the desired minimum (right).

4 Application examples

The proposed strategy has been incorporated in various first–order training algorithms 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. Our experience is that the proposed strategy behaves predictably and reliably. In this section we exhibit results from 100 independent runs for the Silva–Almeida's method [21], the Quickprop algorithm [5] and their globally convergent modifications in two applications.

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 Silva–Almeida's method have been appropriately tuned and received the values $u = 1.02$ and $d = 0.5$, respectively.

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 [12]. The network is based on neurons of the logistic activation model. The termination condition for all algorithms tested is an error value $E \leq 10^{-1}$ within 2000 error function evaluations. Detailed results regarding the training performance of the algorithms are presented in Table 1, where μ denotes the mean number of gradient or error function evaluations, σ the corresponding standard deviation, Min/Max the minimum and maximum number of gradient or error function evaluations, and % denotes the percentage of simulations that converge to a desired minimum.

Table 1: Comparative Results for the Numeric Font Learning Problem

Algorithm	Gradient Evaluation			Function Evaluation			Success
	μ	σ	Min/Max	μ	σ	Min/Max	%
Silva–Almeida	127.16	15.847	103/200	127.16	15.847	103/200	56
Global Silva–Almeida	410.14	129.432	148/862	736.25	317.644	148/1996	99
Quickprop	-	-	-	-	-	-	0
Global Quickprop	88.70	87.504	27/550	176.21	249.913	27/1550	99

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). 20 input/output pairs are taken, scattered in the interval $[0, 2\pi]$ and the termination condition is $E \leq 0.1$ within 10000 error function evaluations. The network is based on hidden neurons with hyperbolic tangent activations and on a linear output neuron. Comparative results are exhibited in Table 2.

Table 2: Comparative Results for the Function Approximation Problem

Algorithm	Gradient Evaluation			Function Evaluation			Success
	μ	σ	Min/Max	μ	σ	Min/Max	%
Silva–Almeida	23.11	116.18	84/150	23.11	116.18	84/150	11
Global Silva–Almeida	382.67	167.11	47/1378	724.89	455.15	47/4005	99
Quickprop	362.81	268.55	58/953	362.81	268.55	58/953	27
Global Quickprop	514.89	686.28	39/2764	1477.10	2308.99	49/8263	61

5 Discussion

A framework for the development of globally convergent batch training algorithms with local learning rates has been proposed. The proposed framework provides conditions under which global convergence is guaranteed and a strategy for adapting the search direction and tuning the length of the minimization step. The applicability of the proposed theorem has been illustrated in two test cases. From Tables 1 and 2 it is shown that the globally convergent modifications of the tested algorithms provide stable learning and therefore a greater possibility of good performance. They exhibit significantly better percentage of success than the original methods, but they generally require additional error function and gradient evaluations.

As shown in Table 1 the Global Quickprop is faster and more reliable than the classical method, which fails to converge within the function evaluations limit. In the same problem, the Silva–Almeida's method fails to converge in 44 out of the 100 runs, due to convergence to undesired local extrema. For the same reason, this method converges only 11 times (see Table 2) in the function approximation problem. In this problem, the Global Quickprop outperforms the classical method in the number of successful runs. However, it fails to converge within the error function evaluations limit in 39 runs. On the other hand, the classical Quickprop method succeeded only in 27 runs due to local minima.

Finally, it is worth mentioning that all the experiments have been performed employing relation (6) cyclically over the local learning rates, i.e. at the kth iteration $i = k \bmod n$. This issue needs further investigation in order to develop techniques that will properly choose η_i^k depending on the learning process.

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

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

[5] 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.

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

[7] H.-C. Hsin, C.-C. Li, M. Sun and R.J. Sclabassi, An adaptive training algorithm for back–propagation neural networks. IEEE Transactions on System, Man and Cybernetics, 25, 512–514, 1995.

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

[9] C.M. Kuan and K. Hornik, Convergence of learning algorithms with constant learning rates, IEEE Transactions on Neural Networks, 2, 484–488, 1991.

[10] R. Liu, G. Dong and X. Ling, A convergence analysis for neural networks with constant learning rates and non–stationary inputs, in *Proc. of the 34th Conference on Decision and Control*, New Orleans, 1278–1283, 1995.

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

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

[13] 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.

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

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

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

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

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

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

[20] D.E. Rumelhart, G.E. Hinton and R.J. Williams, 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] F. Silva and L. Almeida, Acceleration techniques for the back–propagation algorithm, *Lecture Notes in Computer Science*, 412, 110–119, Springer–Verlag, Berlin, 1990.

[22] 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.

[23] M.N. Vrahatis, G.D. Magoulas and V.P. Plagianakos, Convergence analysis of the quickprop method, in Proc. of the *International Joint Conference on Neural Networks (IJCNN'99)*, Washington DC, #848, Session: 5.3, 1999.