1st International Conference "From Scientific Computing to Computational Engineering"
1st IC–SCCE
Athens, 8–10 September, 2004

A NEW EFFICIENT VARIABLE LEARNING RATE FOR PERRY'S SPECTRAL CONJUGATE GRADIENT TRAINING METHOD

A.E. Kostopoulos, D.G. Sotiropoulos, and T.N. Grapsa

University of Patras, Department of Mathematics,
Division of Computational Mathematics & Informatics,
GR-265 00 Rio, Patras, Greece.
e-mail: {arkostop,dgs,grapsa}@math.upatras.gr.

Keywords: supervised training, backpropagation, spectral steplength, conjugate gradient methods, Perry's method, nonmonotone Wolfe conditions.

Abstract. *Since the presentation of the backpropagation algorithm, several adaptive learning algorithms for training a multilayer perceptron (MLP) have been proposed. In a recent article, we have introduced an efficient training algorithm based on a nonmonotone spectral conjugate gradient. In particular, a scaled version of the conjugate gradient method suggested by Perry, which employ the spectral steplength of Barzilai and Borwein, was presented. The learning rate was automatically adapted at each epoch according to Shanno's technique which exploits the information of conjugate directions as well as the previous learning rate. In addition, a new acceptability criterion for the learning rate was utilized based on nonmonotone Wolfe conditions. A crucial issue of these training algorithms is the learning rate adaptation. Various variable learning rate adaptations have been introduced in the literature to improve the convergence speed and avoid convergence to local minima. In this contribution, we incorporate in the previous training algorithm a new effective variable learning rate adaptation, which increases its efficiency. Experimental results in a set of standard benchmarks of MLP networks show that the proposed training algorithm improves the convergence speed and success percentage over a set of well known training algorithms.*

1 INTRODUCTION

The batch training of the MLP can be formulated as a nonlinear unconstrained minimization problem. Namely,

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

where E is the batch error measure defined as the Sum of Squared differences Error function (SSE) over the entire training set, defined by

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{N_M}\left(o_{j,p}^{M} - t_{j,p}\right)^2 \tag{2}$$

where $\left(o_{j,p}^{M} - t_{j,p}\right)^2$ is the squared difference between the actual output value at the j-th output layer neuron for pattern p and the target output value. The scalar p is an index over input-output pairs. The general purpose of the training is to search an optimal set of connection weights in the manner that the errors of the network output can be minimized.

The most popular training algorithm is the batch Backpropagation (BP) introduced by Rumelhart, Hinton and Williams [12]. Although the BP algorithm is a simple learning algorithm for training MLPs, unfortunately it is not based on a sound theoretical basis and is very inefficient and unreliable.

In order to overcome the drawbacks of the BP algorithm, many gradient based training algorithms have been proposed in the literature. An alternative to the gradient based training algorithms are the well

known nonlinear conjugate gradient algorithms. Conjugate gradient methods produce generally faster convergence than the gradient based algorithms.

In a recent article [14], we have introduced an efficient training algorithm based on the technique of nonmonotone spectral conjugate gradient, and we have achieved to improve the convergence rate and the convergence characteristics over well known training algorithms. In particular, a scaled version of the conjugate gradient method, suggested by Perry [2], [9], which employ the spectral steplength of Barzilai and Borwein [1], [10], was presented. The learning rate was automatically adapted at each epoch according to Shanno's technique which exploits the information of conjugate directions as well as the previous learning rate [13]. In addition, a new acceptability criterion for the learning rate was utilized based on nonmonotone Wolfe conditions [4]. In this contribution, we incorporate a new effective variable learning rate adaptation, instead of the previous one introduced by Shanno. The proposed algorithm is improved over the previous one in terms of both convergence rate and convergence characteristics, such as stable learning and robustness to oscillations.

The paper is organized as follows. In the next section, the new training algorithm for MLPs is presented and analyzed. Experimental results are reported in Section 3 in order to evaluate the performance of the new training algorithm and compare it with other well known training algorithms. In Section 4 some remarks and conclusions are presented.

2 THE NEW TRAINING ALGORITHM

In this section, the new training algorithm is introduced and analyzed. The problem we have to deal with is the minimization of the Error function (2). Let the family of the nonlinear conjugate gradient training algorithms having the following iterative form

$$w_{k+1} = w_k + \alpha_k d_k \tag{3}$$

where w_k is the current point, d_k is the conjugate gradient search direction and α_k is the adaptive learning rate determined by some one dimensional search rules along d_k.

The conjugate gradient search directions in our method are generated by the following formula

$$d_{k+1} = -\eta_k g_{k+1} + \beta_k s_k, \quad k \geq 0 \tag{4}$$

where $g_k = \nabla E(w_k)$, $s_k = w_{k+1} - w_k$, η_k is a positive multiplier, and $d_0 = -g_0$.

Conjugate gradient methods differ in their choice for the multiplier β_k used to construct the search direction. In this method, as the previous one introduced in [2] and [14], the multiplier β_k is given by the following formula

$$\beta_k = \frac{(\eta_k y_k - s_k)^T g_{k+1}}{s_k^T y_k} \tag{5}$$

where $y_k = g_{k+1} - g_k$. For $\eta_k = 1$ the above formula is reduced to the formula introduced by Perry in [9].

In our method, as in the previous one [14], we decided to substitute the classical choice $\eta_k = 1$ with the spectral gradient choice introduced by Barzilai and Borwein in [1]. More specifically η_k is given by the formula

$$\eta_k = \frac{s_k^T s_k}{s_k^T y_k} \tag{6}$$

The step size η_k given by (6) is the inverse of the Rayleigh quotient

$$s_k^T [\int_0^1 \nabla^2 E(w_k + t s_k) dt] s_k / s_k^T s_k \tag{7}$$

which, of course, lies between the largest and the smallest eigenvalue of the Hessian average

$$\int_0^1 \nabla^2 E(w_k + t s_k) dt \tag{8}$$

This implies that the step size contains second order information without estimating the Hessian matrix.

As it was mentioned before, conjugate gradient algorithms, in order to be globally convergent, the learning rate parameter must be determined by some one dimensional line search along the conjugate direction d_k. In our method, the learning rate, achieved by the line search technique, must satisfy the nonmonotone Wolfe conditions given by

$$E(w_k + \alpha_k d_k) - \max_{0 \le j \le M} E(w_{k-j}) \le c_1 \alpha_k \nabla E(w_k)^T d_k \tag{9}$$

$$\nabla E(w_k + \alpha_k d_k)^T d_k \ge c_2 \nabla E(w_k)^T d_k \tag{10}$$

where $0 < c_1 \le c_2 < 1$ and M is a non-negative integer, named nonmonotone learning horizon. The first condition (9) allows any point to be accepted if it improves sufficiently the function value compared with the largest of the $M+1$ (or k if $k \le M$) most recent function values. The integer M controls the amount of monotonicity that is allowed. The second condition (10) ensures that the denominator of spectral gradient choice (6) is well defined and always positive, since it implies that $s_k^T y_k > 0$. Both conditions allow an increase in the error function values without affecting the global convergence properties as have been proved in [4].

A simple line search technique that we used for tuning the learning rate parameter α_k is the reduction of the learning rate by a factor $1/q$ where $q = 2$, in order to satisfy the conditions (9) and (10). This backtracking strategy ensures that the learning rate is decreased, so that (9) is satisfied. In particular, the reduction of the learning rate is given by the formula

$$\tilde{\alpha}_k = \frac{1}{2^r} \alpha_k \tag{11}$$

where $\tilde{\alpha}_k$ is the learning rate after r subdivisions needed, so that condition (9) is satisfied. Moreover, in order to avoid unnecessary reduction of the learning rate parameter, we enforce that the learning rate satisfies condition (10).

In our previous work [14], we utilized Shanno's choice for the learning rate adaption given by the formula

$$\alpha_k = \begin{cases} \frac{1}{\|g_0\|_2}, & \text{if } k = 0; \\ \frac{\alpha_{k-1}\|d_{k-1}\|_2}{\|d_k\|_2}, & \text{otherwise.} \end{cases} \tag{12}$$

where d_k is the conjugate gradient direction, d_{k-1} is the previous one, and α_{k-1} is the previous learning rate. The initial learning rate α_0 is $1/\|g_0\|_2$ where g_0 is the initial steepest descent direction.

In this contribution, we introduce a new more efficient learning rate adaption scheme given by the formula

$$\alpha_k = \begin{cases} \frac{1}{||g_0||_2}, & \text{if } k = 0; \\ \frac{1}{2^{r+1}} \frac{\|d_{k-1}\|_2}{\|d_k - d_{k-1}\|_2}, & \text{otherwise.} \end{cases} \tag{13}$$

where r is the number of subdivisions needed for the previous learning rate in order to be accepted by the nonmonotone Wolfe conditions (9) and (10).

The conjugate gradient search direction d_{k+1} given by the formula (4) sometimes fails to be a descent direction. In order to avoid this pathological situation we restart the algorithm. Many restart strategies have been proposed in the literature. In our algorithm, we restart the algorithm with the spectral gradient direction given by the formula

$$d_{k+1} = -\eta_k g_{k+1} \tag{14}$$

The criterion we use to decide if the direction d_{k+1} is descent, is given by the following formula

$$d_{k+1}^T g_{k+1} \le -10^{-3} \|d_{k+1}\|_2 \|g_{k+1}\|_2 \tag{15}$$

Specifically, if (15) holds, we accept as the search direction the one given by the formula (4), otherwise, the angle between d_{k+1} and g_{k+1} is not acute enough, so we restart the algorithm with formula (14).

Nonmonotone Wolfe conditions (9) and (10) in addition with the restart condition (15) are sufficient to prove global convergence of the algorithm under reasonable assumptions. If the gradient of the Error function E, g_k, is Lipschitz continuous and E is bounded bellow it can be proved that

$$\lim_{k \leftarrow +\infty} \|g_k\|_2 = 0 \tag{16}$$

This implies that every limit point of a sequence generated by the algorithm is stationary. More details can be found in [3], [8].

At this point we will summarize the new training algorithm.

Algorithm 2.1

1. *Initialization:*
   - *1.1 Number of epochs* $k = 0$.
   - *1.2 Error goal:= eg.*
   - *1.3 Parameters* $M \geq 1$ *and* $0 < c_1 \leq c_2 < 1$.
   - *1.4 Weight vector:=*w_0.
2. *Calculate the gradient* $g_0 = \nabla E(w_0)$. *Calculate the learning rate* α_0 *using the relation (13). Set* $r = 0$.
3. *Update the weight vector* w_{k+1} *according to relation (3). Calculate the gradient* $g_{k+1} = \nabla E(w_{k+1})$ *and set* $d_{k+1} = -g_{k+1}$.
   - *3.1 If the learning rate acceptability condition (nonmonotone line search) (9) and (10) is fullfilled goto Step 4.*
   - *3.2 Set* $\alpha_k = \alpha_k/2$, $r = r + 1$ *and goto Step 3.*
4. *Check if* $E(w_{k+1}) \leq eg$, *get the final weight vector* w_* *and the corresponding value of* E. *Otherwise set* $k = k + 1$ *and goto to Step 5.*
5. *Compute the Barzilai and Borwein spectral gradient choice* η_k *using the relation (6). Check if* $1/\eta_k < e$ *or* $1/\eta_k < 1/e$. *If the relations hold then accept* η_k *else set* $\eta_k = 1$.
6. *Calculate* β_k *by Perry's formula given by the relation (5). Calculate the new search direction d according to relation (4). If the condition (15) holds then set* $d_{k+1} = d$, *otherwise set* $d_{k+1} = -\eta_k g_{k+1}$.
7. *Calculate the new learning rate* α_k *according to relation (13), set* $r = 0$ *and goto Step 3.*

Remark 2.1 *Parameter e is the one that Raydan in [11] introduces in order to avoid having the spectral gradient choice very large or too small.*

3 EXPERIMENTAL RESULTS

The proposed training algorithm, as described in the previous section, has been applied to several test problems in order to be evaluated fairly. The problems have been tested, are the eXclusive OR Problem, the 3-bit Parity Problem, the Font Learning Problem and the Continuous Function Approximation problem. On each test problem, five batch training algorithms have been simulated: backpropagation with constant learning rate (BP) [12]; backpropagation with constant learning rate and constant momentum (BPM) [5]; adaptive backpropagation with adaptive momentum (ABP) [15]; nonmonotone spectral conjugate gradient (NSCG); modified nonmonotone spectral conjugate gradient (MNSCG). For the simulations, Matlab version 6.5 has been used. We have utilized Matlab Neural Network Toolbox version 4.0 for the BP, BPM and ABP training algorithms. The NSCG and MNSCG training algorithms have been implemented in Matlab environment.

The selection of initial weights is very important in feedforward neural network training. Thus, in order to evaluate the performance of the training algorithms better, the simulations conducted using the same initial weight vectors that have been chosen by the Nguyen - Widrow method [6]. This technique of weight initialization results in distributing the initial weights at the hidden layer in such a way that it is more likely that each input pattern will cause a hidden neuron to learn efficiently, accelerating convergence and preventing premature saturation.

Concerning the heuristic parameters of the BP, BPM and ABP training algorithms, Neural Network Toolbox default values have been used, unless stated otherwise. For the NSCG and MNSCG training

algorithms, the values of the parameters M, c_1, c_2 and e have been fixed to 10, 10^{-4}, 0.5 and 10^{-3}, respectively, for all the experiments. Also, we have to mention that for the BP, BPM, ABP training algorithms one gradient evaluation and one error function evaluation are necessary in each epoch. In NSCG and MNSCG training algorithms, there are an additional number of gradient and error function evaluations when the nonmonotone Wolfe conditions are not satisfied (same number of gradient and error function evaluations required due to the nonmonotone Wolfe conditions). Thus we compare the training algorithms in terms of both gradient and error function evaluations.

For each training problem, we present a table which summarizes the performance of the algorithms for simulations that have reached solution. The resulted parameters are: *min* the minimum number, *max* the maximum number, *mean* the mean value, *s.d.* the standard deviation of gradient and error function evaluations and *succ.* the simulations succeeded out of 1000 simulations. If an algorithm fails to converge, it is considered that it fails to train the feedforward neural network, but its function and gradient evaluations are not included in the statistical analysis of the algorithms. This fact clearly favors BP, BPM and ABP that require too many epochs to complete the task and/or often fail to converge.

3.1 THE EXCLUSIVE - OR PROBLEM

The first problem we have been encountered is the eXclusive - OR (XOR) Boolean function problem, which is considered as a classical problem for the feedforward neural network training. The XOR function maps two binary inputs to a single binary output. As it is well known this function is not linearly separable.

Algorithm	min	max	mean	s.d.	succ.
BP	28	992	94.5329	114.441	74.5%
BPM	23	905	111.821	136.753	76.1%
ABP	18	835	46.5517	73.1005	74.5%
NSCG	28	991	133.136	170.856	87.3%
MNSCG	16	993	115.201	173.146	90.9%

Table 1: Comparative Results for the XOR Problem

The selected architecture of the feedforward neural network is 2-2-1 (six weights and three biases) with hyperbolic tangent activation functions in the hidden neurons and with a linear output neuron. The termination condition has been set to 0.01 and the maximum gradient and error function evaluations to 1000. For the BP and BPM algorithms the learning rate is chosen to be 0.1 instead of the default value 0.01 to accelerate their convergence, since they converge slowly with the default value in this problem. The results of the simulations are presented in Table 1.

It is worth noticing the performance of the MNSCG training algorithm since it exhibits the best success performance (90.9%) while the gradient and error function evaluations are competitive with the other training algorithms.

3.2 THE 3-BIT PARITY PROBLEM

In this simulation we have been considered the 3-bit parity problem, which can be considered as a generalized XOR problem, but it is more difficult. This problem maps three binary inputs to a single binary output. The target of the output is 1, if the number of 1 bits in the input is odd, and 0 otherwise. The selected architecture of the feedforward neural network is 3-2-1 (eight weights and three biases) with hyperbolic tangent activation functions in the hidden neurons and with a linear output neuron. The termination condition has been set to 0.01 and the maximum gradient and error function evaluations to 1000. For the BP and BPM algorithms the learning rate is chosen to be 0.1 instead of the default value 0.01 to accelerate their convergence, since they converge slowly with the default value in this problem. The results of the simulations are presented in Table 2.

Despite the selection of the learning rate for the BP training algorithm, it has failed to converge within the gradient and error function evaluations limit in all the simulations. BPM and ABP performed better, with BPM having better success performance, gradient and error function evaluations over ABP. The NSCG and MNSCG training algorithms, as shown in table 2, had the best success performance as well as the least average gradient and error function evaluations. Also, it is obvious, that the MNSCG training algorithm is significant better in performance than the NSCG training algorithm.

Algorithm	min	max	mean	s.d.	succ.
BP	-	-	-	-	0.0%
BPM	185	996	492.716	197.034	53.5%
ABP	431	982	596.344	128.165	47.4%
NSCG	171	986	429.5	175.026	73.8%
MNSCG	112	992	323.792	174.252	79.4%

Table 2: Comparative Results for the 3-bit Parity Problem

3.3 THE ALPHABETIC FONT LEARNING PROBLEM

The third test problem we have been encountered is the alphabetic font learning problem. We present to the network 26 matrices with the capital letters of the English alphabet. Each letter has been defined in terms of binary values on a grid of size 5×7. The selected architecture of the feedforward neural network is 35-30-26 (1830 weights and 56 biases) with logistic activation functions in the hidden layer and with a linear output neuron.

Algorithm	min	max	mean	s.d.	succ.
BP	1100	1999	1548.22	197.517	75.6%
BPM	1246	1995	1579.5	175.82	4.8%
ABP	1234	1999	1785.87	156.965	36.1%
NSCG	406	1210	707.732	131.425	100.0%
MNSCG	288	801	486.909	84.5928	100.0%

Table 3: Comparative Results for the Alphabetic Font Learning Problem

In order to improve the performance of the methods with fixed learning rate (i.e. BP and BPM), the weights have been initialized following the Nguyen - Widrow method, as we have stated in the beginning of this section, but afterwards the weights and biases of the output layer have been multiplied by 0.01. The termination criterion has been set to 0.1 and the maximum gradient and error function evaluations to 2000. The results of the simulations are presented in Table 3.

It is worth noticing the performance of the MNSCG training algorithm since it exhibits the best success performance (100%) while the mean of gradient and error function evaluations are the lower. Also, as it is shown by the table 3, the MNSCG training algorithm would have the same success performance even if we had not increased the gradient and error function evaluations limits from 1000 to 2000 (the maximum gradient and error function evaluations are 801).

3.4 THE CONTINUOUS FUNCTION APPROXIMATION PROBLEM

The forth test problem we have been considered is the approximation of the continuous trigonometric function $F(x) = \sin(x)\cos(x)$. This problem maps one real input to a single real output. The input values are 20 equally spaced points $x_i \in [0, 2\pi]$ and the target values are the mapping of these points from function $F(x)$. As it is cleared, we have 20 patterns and each pattern is consisted of one input $x \in [0, 2\pi]$ and one target value $F(x)$.

The selected architecture of the feedforward neural network is 1-10-1 (twenty weights and eleven biases) with logistic activation functions in the hidden layer and with a linear output neurons. The termination criterion has been set to 0.1 and the maximum gradient and error function evaluations to 1000. The results of the simulations are presented in Table 4.

Algorithm	min	max	mean	s.d.	succ.
BP	398	994	767.359	154.092	7.8%
BPM	389	995	760.506	157.334	7.7%
ABP	116	999	610.51	217.033	29.0%
NSCG	112	998	486.212	179.492	70.4%
MNSCG	84	983	343.841	168.9	76.9%

Table 4: Comparative Results for the Continuous Function Approximation Problem

The performance of the BP and BPM training algorithms can be characterized inefficient, as it is

shown by the table 4. The ABP has performed better, but not sufficient. The NSCG and MNSCG training algorithms exhibit very good average performance, having the best success rate. Also, the MNSCG outperforms the NSCG training algorithm in both success rate and gradient and error function evaluations.

4 CONCLUSIONS

In this paper, a new efficient variable learning rate has been proposed for use in conjunction with Perry's nonmonotone spectral conjugate gradient training algorithm. It is shown that the NSCG training method was considerably improved by using the new learning rate in terms of convergence speed and success percentage. The gradient and function evaluations are reduced over all the training algorithms that we used in order to evaluate the new training algorithm.

Further work must be done in order to evaluate the new learning rate, by incorporating it to other well known conjugate gradient training algorithms.

References

[1] J. Barzilai and J.M. Borwein, Two point step size gradient methods, *IMA J. Numer. Anal.*, **8**, 141–148, (1988).

[2] E.G. Birgin and J.M. Martinez, A spectral conjugate gradient method for unconstrained optimization, *Applied Mathematics and Optimization*, **43**, 117–128, (1999).

[3] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM J. Numer. Anal.*, **23**, 707–716, (1986).

[4] J. Han, J. Sun and W. Sun, Global Convergence of Non-Monotone Descent Methods for Unconstrained Optimization Problems, *Journal of Computational and Applied Mathematics*, **146**, 89–98, (2002).

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

[6] D. Nguyen and B. Widrow, Improving the learning speed of 2-layer neural network by choosing initial values of the adaptive weights, *IEEE First International Joint Conference on Neural Networks*, **3**, 21–26, (1990).

[7] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research, (1999).

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

[9] A. Perry, A modified conjugate gradient algorithm, *Operations Research*, **26**, 26–33, (1978).

[10] V.P. Plagianakos, D.G. Sotiropoulos and M.N. Vrahatis, Automatic adaptation of learning rate for backpropagation neural networks In: *Recent Advances in Circuits and Systems*, N.E. Mastorakis ed., pp.337-341, World Scientific Publishing Co. Pte. Ltd., (1998).

[11] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM J. Optim.*, **7**, 26–33, (1997).

[12] D.E. Rumelhart and J.L. McClelland(eds), Parallel distributed processing: explorations in the microstructure of cognition, Vol. 1: Foundations, MIT Press, 1986.

[13] D.F. Shanno and K.H. Phua, Minimization of unconstrained multivariate functions, *ACM Transactions on Mathematical Software*, **2**, 87–94, (1976).

[14] D.G. Sotiropoulos, A.E. Kostopoulos and T.N. Grapsa, A spectral version of Perry's conjugate gradient method for neural network training, In: *Proceedings of 4th GRACM Congress on Computational Mechanics*, 27-29 June, University of Patras, Greece, (2002).

[15] 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).