An Efficient Generalization of Battiti-Shanno's Quasi-Newton Algorithm for Learning in MLP-Networks

Carmine Di Fiore, Stefano Fanelli, and Paolo Zellini

Department of Mathematics, University of Rome "Tor Vergata", Rome, Italy
fanelli@mat.uniroma2.it

Abstract. This paper presents a novel Quasi-Newton method for the minimization of the error function of a feed-forward neural network. The method is a generalization of Battiti's well known OSS algorithm. The aim of the proposed approach is to achieve a significant improvement both in terms of computational effort and in the capability of evaluating the global minimum of the error function. The technique described in this work is founded on the innovative concept of "convex algorithm" in order to avoid possible entrapments into local minima. Convergence results as well numerical experiences are presented.

1 Introduction

It is well known that, in order to determine the global minimum of the error function of a MLP-network, it is necessary to superimpose global optimization techniques in the computational scheme of a backpropagation algorithm. An approach of this type was utilized in [7] to derive a preliminary version of a "pseudo–backpropagation" method and in [8] to refine the global algorithm introduced in [7] thereby dealing with high-dimensional problems more efficiently.

The latter approach is founded on a new definition, involving both the mathematical properties of the error function and the behaviour of the learning algorithm. The corresponding hypotheses, called "non–suspiciousness conditions" (NSC), represent a sort of generalization of the concept of convexity. Roughly speaking, a "non–suspect" minimization problem is characterized by the fact that, under general regularity assumptions on the error function, a "suitable" pseudo-backpropagation algorithm is able to compute the optimal solution, avoiding unfair entrapments into local minima.

In [7] we proved that under the NSC and with a proper choice of the step-sizes the optimal algorithm can be obtained by a classical gradient descent-type scheme.

The present paper has several aims. Firstly, we present a generalization of OSS algorithm [2]. Secondly, we show that second order methods of Quasi-Newton(QN)-type can be implemented in the frame of the NSC theory. Thirdly, we prove that the novel algorithm, named Generalized Battiti (GB), can be successfully applied to large networks.

N.R. Pal et al. (Eds.): ICONIP 2004, LNCS 3316, pp. 483–488, 2004.

A fundamental contribution towards the implementation of efficient QN-methods to MLP-networks is based on the utilization of generalized BFGS-type algorithms, named $\mathcal{L}QN$, involving a suitable family of matrix algebras $\mathcal{L}$ (see [5],[6]). The main advantage of these methods is based upon the fact that they have an $O(nlogn)$ complexity per step and that they require $O(n)$ memory allocations, where n is the number of network connections. Moreover, $\mathcal{L}QN$ methods are competitive with the most recent L-BFGS algorithms ([13],[14]), since they perform a sort of optimal second order information compression in the array of the eigenvalues of the best approximation to the Hessian matrix.

In this paper we show that a simplified variant of $\mathcal{L}QN$ methods can be easily applied in the learning process of an MLP-network. The latter variant represents a significant improvement of Battiti's algorithm and is preferable to the L-BFGS methods utilizing a comparable amount of memory.

2 Generalized BFGS-Type Algorithms

Let us consider the optimisation problem:

$$\min_{\mathbf{w}\in\Re^n} E(\mathbf{w}) \tag{1}$$

where $E(\mathbf{w})$ is the error function of an MLP-network.

Let us suppose that (1) has a solution $\mathbf{w}^*$ (global minimum) and let E_{min} denote the corresponding value of the function E.

Denote by $\nabla E(\mathbf{w})$ and by $\nabla^2 E(\mathbf{w})$ the gradient vector and the Hessian matrix of E in $\mathbf{w}$, respectively. The matrix B_{k+1}, replacing $\nabla^2 E(\mathbf{w}_{k+1})$ in the $BFGS$ method, is a rank-2 perturbation of the previous positive definite Hessian approximation B_k, defined in terms of the two current difference vectors $\mathbf{s}_k = \mathbf{w}_{k+1} - \mathbf{w}_k$ and $\mathbf{y}_k = \nabla E(\mathbf{w}_{k+1}) - \nabla E(\mathbf{w}_k)$ by the following formula:

$$B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k) = B_k + \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}\mathbf{y}_k\mathbf{y}_k^T - \frac{1}{\mathbf{s}_k^T B_k \mathbf{s}_k} B_k \mathbf{s}_k \mathbf{s}_k^T B_k. \tag{2}$$

The method introduced by Battiti [2], named OSS, has $O(n)$ complexity per step, being a simple memory-less modification of equation (2), in which the identity matrix I is used, instead of B_k, to compute the new approximation B_{k+1}. In [2] it is shown that OSS can be extremely competitive with the original $BFGS$ method to perform optimal learning in a large MLP-network. Unfortunately, by the very nature of the memory-less approach, the amount of second order information contained in OSS is considerably reduced in comparison with the standard $BFGS$ method.

The main problem connected to the calculation of the Hessian approximation B_{k+1} is, in fact, to minimize the computational complexity per iteration, by maintaining a QN rate of convergence.

In [9] we introduced a generalized iterative scheme, named $\mathcal{L}QN$, where the matrix B_{k+1} is defined by utilizing a suitable matrix $\tilde{B}_k$ instead of B_k ($BFGS$) or I (OSS):

$$B_{k+1} = \varphi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k) \tag{3}$$

We point out that in the $\mathcal{L}QN$ methods $\tilde{B}_k$ is in general a *structured dense* matrix. The method considered in [13] is instead obtained from (3) by setting :

$$\tilde{B}_k = \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} I \tag{4}$$

Notice that (4) is the simplest L-BFGS algorithm (i.e. L-BFGS for $m = 1$ [1]). We study here other possible choices in the family of matrices:

$$\tilde{B}_k = \alpha_k I \tag{5}$$

It is interesting to observe (see [10] for the rigorous proof) that *the choice of* α_k *given in* (4) *is associated to the minimum value of the condition number of* $B_{k+1} = \varphi(\alpha_k I, \mathbf{s}_k, \mathbf{y}_k)$. In [13],[14] the latter choice was only justified by weaker arguments based on experimental observations.

In this paper, we have implemented in particular the following values of α_k:

$$\alpha_k^* = \left(1 - \frac{1}{n}\right) \alpha_{k-1}^* + \frac{1}{n} \frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{y}_{k-1}^T \mathbf{s}_{k-1}} \tag{6}$$

$$\begin{cases} \alpha_k^{**} = \alpha_{k-1}^{**} + \frac{1}{n-1} \left(\frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}} - \frac{\mathbf{v}_k^T \mathbf{v}_k}{\mathbf{s}_k^T \mathbf{v}_k} \right); \\ \mathbf{v}_k = \alpha_{k-1}^{**} \left(\mathbf{s}_k - \frac{\mathbf{s}_{k-1}^T \mathbf{s}_k}{\|\mathbf{s}_{k-1}\|^2} \mathbf{s}_{k-1} \right) + \frac{\mathbf{y}_{k-1}^T \mathbf{s}_k}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}} \mathbf{y}_{k-1} \end{cases} \tag{7}$$

The formula (6) can be derived by minimizing $\|\alpha_k I - \varphi(\alpha_{k-1}^* I, \mathbf{s}_{k-1}, \mathbf{y}_{k-1})\|_F$, being $\|\cdot\|_F$ the Frobenius norm. *We underline that the resulting matrix* $\alpha_k^* I$ *is the best least squares fit* $\mathcal{L}_{B_k}$ *of* B_k *in the space* $\mathcal{L} = \{zI : z \in C\}$ (see [9]), i.e. (6) gives rise to a $\mathcal{L}QN$ method. The formula (7) is obtained by solving the minimum problem:

$$\min_{\alpha_k} \|\varphi(\alpha_k I, \mathbf{s}_k, \mathbf{y}_k) - \varphi(\varphi(\alpha_{k-1}^{**} I, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}), \mathbf{s}_k, \mathbf{y}_k)\|_F,$$

or, in other words, by minimizing the distance between B_{k+1} and the matrix defining the search direction in the L-BFGS algorithm, for m=2. In Section 4,we shall compare the performances of the algorithms based on the use of (5)-(6) and (5)-(7) with the original OSS and (4).

3 Theoretical Results

The following assumptions, named "non–suspiciousness conditions" were originally introduced in [3], [12] and redefined in a more suitable form in [7].

Definition 1. *The non-suspiciousness conditions hold if* $\exists \lambda_k$*:*

1. $\forall \epsilon_a \in \Re^+$, $\exists \epsilon_s \in \Re^+$*:* $\|\nabla E(\mathbf{w}_k)\| > \epsilon_s$ *during the optimization algorithm, apart from k:* $E(\mathbf{w}_k) - E_{min} < \epsilon_a$*;*

2. $\lambda_k \|\nabla E(\mathbf{w}_k)\|^2 \leq \epsilon_a$*;*

3. $E \in C^2$ *and* $\exists H > 0 : \|\nabla^2 E(\mathbf{w})\| \leq H$.

The following convergence result was proved in [12](see also [4]):

Theorem 1. *Let the non-suspiciousness conditions hold for problem* (1)*. Then,* $\forall \epsilon_a \in \Re^+$, $\exists k^{**} : \forall k > k^{**}$*:*

$$E(\mathbf{w}_k) - E_{min} < \epsilon_a, \tag{8}$$

by using the gradient descent-type scheme: $\mathbf{w}_{k+1} = \mathbf{w}_k - \lambda_k \nabla E(\mathbf{w}_k)$.

Let E.S.W. be the following λ-subset of the classical Strong Wolfe conditions

$$\begin{cases} E(\mathbf{w}_k + \lambda \mathbf{d}_k) \leq E(\mathbf{w}_k) + c_1 \lambda \nabla E(\mathbf{w}_k)' \mathbf{d}_k, \\ c_3 |\nabla E(\mathbf{w}_k)' \mathbf{d}_k| \leq |\nabla E(\mathbf{w}_k + \lambda \mathbf{d}_k)' \mathbf{d}_k| \leq c_2 |\nabla E(\mathbf{w}_k)' \mathbf{d}_k| \end{cases} \tag{9}$$

being $\lambda > 0$ and $0 < c_1 \leq c_3 < c_2 < 1$ proper constants. E.S.W. *is not empty whenever* $\mathbf{d}_k$ *is a descent direction* (see [10],[14]). It can be proved (see again [10]) the fundamental:

Theorem 2. *Let the condition* **1.** *of* **Definition 1.** *hold for problem (1). Moreover, assume that* $E \in C^2$ *and* $\exists m, M > 0$ *:*

$$\frac{\|\mathbf{y}_k\|}{\|\mathbf{s}_k\|} \geq m \quad ; \quad \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M \tag{10}$$

Then, $\forall \epsilon_a \in \Re^+$, $\exists k^{**} : \forall k > k^{**}$*:*

$$E(\mathbf{w}_k) - E_{min} < \epsilon_a, \tag{11}$$

by using the QN-type scheme:

$$\begin{cases} \mathbf{w}_{k+1} = \mathbf{w}_k - \lambda_k B_k^{-1} \nabla E(\mathbf{w}_k) \\ B_k = \varphi(\tilde{B}_{k-1}, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}) \end{cases} \tag{12}$$

Theorem 2 shows that QN-type methods can be implemented in the frame of the NSC theory. It is important to emphasize that the scalars λ_k, evaluated by (9) in connection with (10), perform a sort of *experimental translation* of the condition **1.** of **Definition 1.** (see once again [10]). Observe that, if E were a convex function, the second inequality in (10) would be satisfied ([9]). Furthermore, the first inequality implies that $\|\nabla^2 E(\xi_k)\| \geq m$, being $\xi_k = \mathbf{w}_k + t(\mathbf{w}_{k+1} - \mathbf{w}_k), 0 < t < 1$. This fact justifies the following:

Definition 2. *A QN-method satisfying the conditions* (10) *is called a convex algorithm*(see [10] for more details).

4 Experimental Results

In this section we study the local convergence properties of the algorithms described in section 2. In particular, it is shown that the novel values of α_k given in (6),(7) are the most competitive. Some well known non-analytical tasks taken from UCI Repository of machine learning databases (IRIS and Ionosphere [11]) are selected as benchmarks. We consider the training of *i-h-o* networks where i, h and o are the number of input, hidden and output nodes, respectively.

Since thresholds are associated with hidden and outer nodes, the total number of connections (weights) is $n = ih+ho+h+o$. In the learning process of IRIS and Ionosphere n= 315, 1408, respectively (for more details see [5]). CPU-time is referred to a Pentium 4-M, 2 GHz with a machine precision of $.1 \times 10^{-18}$.

LN, GB1, GB2 indicate the algorithms utilizing the values of α_k given in (4), (6) and (7), respectively. In all the algorithms we implement the same line-search technique, i.e. the classical efficient Armijo-Goldstein (A.G.) conditions ([7],[8]).

Define $p_k = \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k}$

$$
\begin{array}{l}
\mathbf{w}_0 \in \Re^n,\ \mathbf{d}_0 = -\nabla E(\mathbf{w}_0) \\
For\ k = 0, 1, \ldots : \\
\left\{ \begin{array}{l}
\mathbf{w}_{k+1} = \mathbf{w}_k + \lambda_k \mathbf{d}_k, \quad \lambda_k \in A.G. \\
\mathbf{s}_k = \mathbf{w}_{k+1} - \mathbf{w}_k,\ \mathbf{y}_k = \nabla E(\mathbf{w}_{k+1}) - \nabla E(\mathbf{w}_k) \\
\textbf{define}\ \alpha_k : \\
OSS:\ \alpha_k = 1 \\
LN:\ \alpha_k = p_k \\
GB1:\ if(k=0)\{\alpha_k = 1 \quad or \quad \alpha_k = p_k\} else\{ \\
\qquad \alpha_k = \left(1 - \frac{1}{n}\right) \alpha_{k-1} + \frac{1}{n} \frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{y}_{k-1}^T \mathbf{s}_{k-1}}; \\
\qquad\qquad \} \\
GB2:\ if(k=0)\{\alpha_k = 1 \quad or \quad \alpha_k = p_k\} else\{ \\
\qquad \mathbf{v}_k = \alpha_{k-1} \left(\mathbf{s}_k - \frac{\mathbf{s}_{k-1}^T \mathbf{s}_k}{\|\mathbf{s}_{k-1}\|^2} \mathbf{s}_{k-1} \right) + \frac{\mathbf{y}_{k-1}^T \mathbf{s}_k}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}} \mathbf{y}_{k-1}; \\
\qquad \alpha_k = \alpha_{k-1} + \frac{1}{n-1} \left(\frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}} - \frac{\mathbf{v}_k^T \mathbf{v}_k}{\mathbf{s}_k^T \mathbf{v}_k} \right); \\
\qquad\qquad \} \\
B_{k+1} = \varphi(\alpha_k I, \mathbf{s}_k, \mathbf{y}_k) \\
\mathbf{d}_{k+1} = -B_{k+1}^{-1} \nabla E(\mathbf{w}_{k+1})
\end{array} \right.
\end{array}
$$

The following table reports the number of iterations (the seconds) required by the algorithms to obtain $E(\mathbf{w}_k) < 10^{-1}$, where E is the error function of the corresponding MLP. Experiments are related to different initial weights of IRIS and Ionosphere networks. Notice that GB1 and GB2 are often more efficient than LN. *OSS in some cases performs less iterations, but it is always dominated in terms of CPU time* . The latter experimental result depends upon the fact that, in order to evaluate $\lambda_k \in$ A.G., OSS requires more computational effort. The general $\mathcal{L}QN$, using *dense and structured matrices* $\mathcal{L}$ ($\mathcal{L}$=Hartley algebra, see [9]), outperforms all the other algorithms.

5 Conclusions

All GB-algorithms examined in Table 1 require $6n$ memory allocations (see [14] for L-BFGS $m = 1$). Moreover, the computational complexity per step is cn, being c a very small constant. Further experimental results have shown that the use of L-BFGS methods, for $m = 2$ or $m = 3$, does not reduce the number of iterations. Since minimizing the error function of a neural network is a nonlinear least squares problem, one could use the well known Levenberg-Marquardt(LM)

Table 1. k (seconds): $f(\mathbf{x}_k) < 10^{-1}$.

Algorithms	iris1	iris2	iono1	iono2	iono3
OSS	49528 (1037)	42248 (863)	3375 (277)	2894 (236)	2977 (245)
LN	41291 (361)	40608 (358)	4054 (165)	5156 (212)	5171 (215)
GB1, $\alpha_0 = 1$	24891 (207)	34433(286)	4415 (174)	3976 (157)	4159 (164)
GB2, $\alpha_0 = 1$	20086 (166)	17057 (142)	3821 (152)	4519 (178)	3794 (149)
General $\mathcal{LQN}$	12390 (112)	15437 (140)	993 (49)	873 (43)	1007 (48)

method. Unfortunately, LM needs at least $O(n^2)$ memory allocations and its implementation requires the utilization of more expensive procedures than GB algorithms (i.e. Givens and Householder for QR factorizations). As a matter of fact, the original algorithm OSS turns out to be much more efficient than LM for large scale problems (see [15]).

References

1. M.Al Baali, Improved Hessian approximations for the limited memory BFGS method, *Numer. Algorithms*, Vol. 22, pp.99–112, 1999.
2. R. Battiti, First- and second-order methods for learning: between steepest descent and Newton's method, *Neural Computation*, Vol. 4, pp. 141–166, 1992.
3. M. Bianchini, S. Fanelli, M.Gori, M.Protasi, Non-suspiciousness: a generalisation of convexity in the frame of foundations of Numerical Analysis and Learning, *IJCNN'98*, Vol.II, Anchorage, pp. 1619–1623, 1998.
4. M.Bianchini, S.Fanelli, M.Gori, Optimal algorithms for well-conditioned nonlinear systems of equations, *IEEE Transactions on Computers*, Vol. 50, pp. 689-698, 2001.
5. A.Bortoletti, C.Di Fiore, S.Fanelli, P.Zellini, A new class of quasi-newtonian methods for optimal learning in MLP-networks, *IEEE Transactions on Neural Networks*, Vol. 14, pp. 263–273, 2003.
6. C.Di Fiore, S.Fanelli, P.Zellini, Matrix algebras in quasi-newtonian algorithms for optimal learning in multi-layer perceptrons, *ICONIP Workshop and Expo*, Dunedin, pp. 27–32, 1999.
7. C.Di Fiore, S.Fanelli, P.Zellini, Optimisation strategies for nonconvex functions and applications to neural networks, *ICONIP 2001*, Vol. 1, Shanghai, pp. 453–458, 2001.
8. C. Di Fiore, S.Fanelli, P.Zellini, Computational experiences of a novel algorithm for optimal learning in MLP-networks, *ICONIP 2002*, Vol. 1, Singapore, pp. 317–321, 2002.
9. C. Di Fiore, S. Fanelli, F. Lepore, P. Zellini, Matrix algebras in Quasi-Newton methods for unconstrained optimization, *Numerische Mathematik*, Vol. 94, pp. 479–500, 2003.
10. C. Di Fiore, S.Fanelli, P.Zellini, Convex algorithms for optimal learning in MLP-networks, *in preparation*
11. R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.
12. P. Frasconi, S. Fanelli, M. Gori, M. Protasi, Suspiciousness of loading problems, *IEEE Int. Conf. on Neural Networks*, Vol. 2, Houston, pp. 1240–1245, 1997.
13. D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Programming*, Vol. 45, pp.503–528, 1989.
14. J. Nocedal, S.J. Wright, *Numerical Optimization.* New York: Springer-Verlag, 1999
15. http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/backpr14.html