Applied Mathematics and Computation 221 (2013) 491–502

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A new conjugate gradient algorithm for training neural networks based on a modified secant equation

Ioannis E. Livieris *, Panagiotis Pintelas

Department of Mathematics, University of Patras, GR 265-00, Greece
Educational Software Development Laboratory, Department of Mathematics, University of Patras, GR 265-00, Greece

ARTICLE INFO

Keywords:
Artificial neural networks
Descent conjugate gradient algorithm
Modified secant equation
Global convergence

ABSTRACT

Conjugate gradient methods have been established as excellent neural network training methods, due to the simplicity of their iteration, numerical efficiency and their low memory requirements. In this work, we propose a conjugate gradient neural network training algorithm which guarantees sufficient descent using any line search, avoiding thereby the usually inefficient restarts. Moreover, it approximates the second order curvature information of the error surface with a high-order accuracy by utilizing a new modified secant condition. Under mild conditions, we establish that the global convergence of our proposed method. Experimental results provide evidence that our proposed method is in general superior to the classical conjugate gradient training methods and has a potential to significantly enhance the computational efficiency and robustness of the training process.

1. Introduction

Artificial neural networks are parallel computational models comprised of densely interconnected, adaptive processing units, characterized by an inherent propensity for learning from experience and also discovering new knowledge. Due to their excellent capability of self-learning and self-adapting, they have been extensively studied and have been widely used in many applications of artificial intelligence (see [5,21,42]) and are often found to be more efficient and more accurate than other classification techniques [24]. Although many different models have been proposed, the feedforward neural networks (FNNs) are the most common and widely used in a variety of applications.

Mathematically, the problem of *training* a FNN can be formulated as the minimization of an error function E; that is to find a minimizer $w^* = (w_1^*, w_2^*, \ldots, w_n^*) \in \mathbb{R}^n$, such that

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

where E is the batch error measure defined by the sum of square differences over all examples of the training set, namely

$$E(w) = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{N_L}(o_{j,p}^L - t_{j,p})^2, \tag{1.1}$$

where $o_{j,p}^L$ is the actual output of the jth neuron that belongs to the Lth (output) layer, N_L is the number of neurons of the output layer, $t_{j,p}$ is the desired response at the jth neuron of the output layer at the input pattern p and P represents the total number of patterns used in the training set.

* Corresponding author at: Department of Mathematics, University of Patras, GR 265-00, Greece.
E-mail addresses: livieris@upatras.gr, livieris@gmail.com (I.E. Livieris).

0096-3003/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2013.06.101

Conjugate gradient methods are probably the most famous iterative methods for efficiently training neural networks in scientific and engineering computation [1,6,27,29,28,30,34,35,40]. They are characterized by the simplicity of their iteration, numerical efficiency and their low memory requirements. There methods generate a sequence of weights $\{w_k\}$, using the iterative formula

$$w_{k+1} = w_k + \eta_k d_k, \quad k = 0, 1, \ldots, \tag{1.2}$$

where k is the current iteration usually called *epoch*, $w_0 \in \mathbb{R}^n$ is a given initial point, $\eta_k > 0$ is the learning rate and d_k is a descent search direction defined by

$$d_k = \begin{cases} -g_0, & \text{if } k = 0; \\ -g_k + \beta_k d_{k-1}, & \text{otherwise}, \end{cases} \tag{1.3}$$

where β_k is a scalar and $g_k = \nabla E(w_k)$. In the literature, there have been proposed several choices for β_k which give rise to distinct conjugate gradient methods with quite different computational efficiency and theoretical properties (see [8,17,33] and the references therein). Well known formulas for β_k include the Hestenes–Stiefel (HS) [20], the Fletcher–Reeves (FR) [10] and the Polak–Ribière (PR) [36] which are specified by

$$\beta_k^{\mathrm{HS}} = \frac{g_k^T y_{k-1}}{y_{k-1}^T d_{k-1}}, \quad \beta_k^{\mathrm{FR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \beta_k^{\mathrm{PR}} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2},$$

respectively, where $y_{k-1} = g_k - g_{k-1}$ and $\|\cdot\|$ denotes the Euclidean norm.

The HS method behaves like the PR method in practical computation and is generally regarded as one of the most efficient conjugate gradient methods thus it has undergone meticulous study in recent years. However this method has the major drawback of not being globally convergent for general functions and as a result it can cycle infinitely without presenting any substantial progress [37]. To rectify the convergence failure of the HS method, Gilbert and Nocedal [14] conducted a detailed analysis and proposed a modification of the HS formula, $\beta_k^{\mathrm{HS+}} = \max\{\beta_k^{\mathrm{HS}}, 0\}$. Although one would be satisfied with the global convergence of HS + method, this method sometimes has similar computational performance with the classical HS method when applied for training a neural network. This is due to fact that the training process is a rather difficult optimization problem since the error surface is too complex and it is characterized by a number of unhelpful features [18]. More specifically, its dimensionality is often high and the corresponding nonconvex multimodal objective function possess multitudes of local minima and has broad flat regions adjoined with narrow step ones. These specific characteristics make the training process particularly difficult and constitute a crucial factor for the neural network's performance. Hence, for improving the efficiency and robustness of the training process, we will based on ideas from the well established unconstrained optimization theory. During the last decade, much effort has been devoted to developing new conjugate gradient methods which possess strong convergence properties and are also computationally superior to classical methods. These methods can be classified in the following two classes.

The first class exploits second order information to accelerate conjugate gradient methods by utilizing modified secant equations (see [11,12,25,47,48]). Ford et al. [13] proposed a multi-step conjugate gradient method which is based on the multi-step quasi-Newton methods proposed in [11,12]. On the basis of this idea, Yabe and Takano [43], Li et al. [25] and Babaie-Kafaki et al. [3] proposed conjugate gradient methods which are based on modified secant equations with higher orders of accuracy in the approximation of the curvature. Under proper conditions these methods are globally convergent and sometimes their numerical performance is superior to classical conjugate gradient methods. However, these methods cannot guarantee to generate descent directions, therefore restarts are employed in their analysis and implementation in order to guarantee convergence. Nevertheless, there is also a worry with restart algorithms that restarts may be triggered too often; thus degrading the overall efficiency and robustness of the minimization process [32].

The second approach aims on developing conjugate gradient methods which generate descent directions. Independently, Hager and Zhang [16], Yu [45] and Yu et al. [44] proposed conjugate gradient methods which ensure sufficient descent using any line search. Moreover, an important feature of their works is that they established the global convergence of their proposed methods for general functions. Motivated by the previous works, Livieris and Pintelas [27,28] and Livieris et al. [29] presented some descent conjugate gradient training algorithms providing some promising results. Based on their numerical experiments the authors concluded that the sufficient descent property led to a significant improvement of the efficiency of the training process.

Recently, Zhang et al. [50] considered to modify the search direction (1.3) such that it ensures sufficient descent $g_k^T d_k = -\|g_k\|^2$, independent of the accuracy of the performed line search. More analytically, similar to the spectral conjugate gradient method [4], they proposed a modified FR method as follows:

$$d_k = -\left(1 + \beta_k^{\mathrm{FR}} \frac{g_k^T d_{k-1}}{\|g_k\|}\right) g_k + \beta_k^{\mathrm{FR}} d_{k-1}. \tag{1.4}$$

Notice that, if β_k^{FR} in Eq. (1.4) is replaced by another existing conjugate gradient formula, the property $g_k^T d_k = -\|g_k\|^2$ is still satisfied. Along this line, the authors proposed a modified PR method [49] and a modified HS method [51].

In this paper, we propose a new conjugate gradient method which consists of a modification of the Hestenes–Stiefel method. Our proposed method ensures sufficient descent using any line search and it is based on a new modified secant equation which approximates the second order curvature information of the error function with high-order accuracy. Under mild conditions, we establish the global convergence of our proposed method.

The remainder of this paper is organized as follows. In Section 2, we present a new modified secant equation and our proposed conjugate gradient training algorithm. Section 3 presents the global convergence analysis of our method. The experimental results are reported in Section 4 using the performance profiles of Dolan and Morè [9]. Finally, Section 5 presents our concluding remarks.

2. Modified Hestenes–Stiefel$^+$ conjugate gradient training algorithm

Firstly, we recall that for quasi-Newton methods, an approximation matrix B_{k-1} to the Hessian $\nabla^2 E(w_{k-1})$ of the nonlinear error function E is updated so that a new matrix B_k satisfies the following secant condition

$$B_k s_{k-1} = y_{k-1}, \tag{2.1}$$

where $s_{k-1} = w_k - w_{k-1}$. Wei et al. [41] expanded this condition and derived a class of modified secant condition in the form

$$B_{k-1} s_{k-1} = \tilde{y}_{k-1}, \quad \tilde{y}_{k-1} = y_{k-1} + \frac{\theta_{k-1}}{s_{k-1}^T u} u, \tag{2.2}$$

where u is any vector satisfying $s_{k-1}^T u \neq 0$ and θ_{k-1} is defined by

$$\theta_{k-1} = 2(E_{k-1} - E_k) + (g_k + g_{k-1})^T s_{k-1}. \tag{2.3}$$

where $E_k = E(w_k)$. Notice that this new quasi-Newton Eq. (2.2) contains not only gradient value information but also function value information at the present and the previous step. Moreover, Wei et al. [41] proved that if $\|s_{k-1}\|$ is sufficiently small then

$$s_{k-1}^T (\nabla^2 E(w_k) s_{k-1} - y_{k-1}) = O(\|s_{k-1}\|^3),$$
$$s_{k-1}^T (\nabla^2 E(w_k) s_{k-1} - \tilde{y}_{k-1}) = O(\|s_{k-1}\|^4).$$

Clearly, the above equations imply that the modified secant Eq. (2.2) is superior to the classical one (2.1) in the sense that $\tilde{y}_{k-1}$ better approximates $\nabla^2 E(w_k) s_{k-1}$ than y_{k-1} (see [41]).

However, for values of $\|s_{k-1}\|$ greater than one (i.e. $\|s_{k-1}\| > 1$), the standard secant Eq. (2.1) is expected to be more accurate than the modified secant Eq. (2.2). In order to overcome this difficulty we considered an extension of the modified secant Eq. (2.2) in the following way

$$B_{k-1} s_{k-1} = \tilde{y}_{k-1}^*, \quad \tilde{y}_{k-1}^* = y_{k-1} + \rho_{k-1} \frac{\max\{\theta_{k-1}, 0\}}{s_{k-1}^T u} u, \tag{2.4}$$

where parameter $\rho_{k-1} \in \{0, 1\}$ and adaptively switch between the standard secant Eq. (2.1) and the modified secant Eq. (2.4), by setting $\rho_{k-1} = 1$ if $\|s_{k-1}\| \leqslant 1$ and setting $\rho_{k-1} = 0$, otherwise.

Motivated by the theoretical advantages of the new modified secant condition (2.4), we propose a modification of formula β_k^{HS+} as follows:

$$\beta_k^{MHS+} = \max\left\{ \frac{g_k^T \tilde{y}_{k-1}^*}{d_{k-1}^T \tilde{y}_{k-1}^*}, 0 \right\}. \tag{2.5}$$

Furthermore, in order to develop a descent conjugate gradient training algorithm, avoiding thereby the usually inefficient restarts, we exploit the technique of the modified FR method [50]. Let the search direction be defined by

$$d_k = -\left(1 + \beta_k^{MHS+} \frac{g_k^T d_{k-1}}{\|g_k\|}\right) g_k + \beta_k^{MHS+} d_{k-1}. \tag{2.6}$$

It is easy to see that condition

$$g_k^T d_k = -\|g_k\|^2 \tag{2.7}$$

holds.

At this point, we present a high level description of our proposed algorithm called modified Hestenes–Stiefel$^+$ conjugate gradient algorithm (MHS$^+$-CG).

Algorithm 1. Modified Hestenes–Stiefel$^+$ Conjugate Gradient Algorithm

Step 1. Initiate w_0, E_G and k_{MAX}; set $k = 0$.
Step 2. Calculate the error function value E_k and its gradient g_k.
Step 3. If $(E_k < E_G)$ return "Error goal reached".
Step 4. If $(g_k = 0)$ return "Error goal not met".
Step 5. Compute the descent direction d_k using (2.6).
Step 6. Compute the learning rate η_k satisfying the strong Wolfe conditions

$$E(w_k + \alpha_k d_k) - E(w_k) \leqslant \sigma_1 \alpha_k g_k^T d_k, \tag{2.8}$$
$$|g(w_k + \alpha_k d_k)^T d_k| \leqslant \sigma_2 |g_k^T d_k|. \tag{2.9}$$

Step 7. Update the weights $w_{k+1} = w_k + \eta_k d_k$ and set $k = k + 1$.
Step 8. If $(k > k_{MAX})$ return "Error goal not met" else goto Step 2.

Remark. We denote the error goal and the maximum number of epochs with E_G and k_{MAX}, respectively. In Step 5 since the search direction is always a descent direction, we avoid the use of a restarting criterion. Moreover, notice that by the Wolfe condition (2.9) and Eq. (2.2), it follows that $d_{k-1}^T \tilde{y}_{k-1}^* \geqslant d_{k-1}^T y_{k-1} > 0$, which implies that formula β_k^{MHS+} is well defined.

3. Global convergence analysis

In order to establish the global convergence result for Algorithm MHS$^+$-CG, we will impose the following assumptions.

Assumption 1. The level set $\mathcal{L} = \{w \in \mathbb{R}^n | E(w) \leqslant E(w_0)\}$ is bounded, namely, there exists a constant $B > 0$ such that

$$\|w\| \leqslant B, \quad \forall w \in \mathcal{L}. \tag{3.1}$$

Assumption 2. In some neighborhood $\mathcal{N} \in \mathcal{L}$, E is differentiable and its gradient g is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|g(w) - g(\tilde{w})\| \leqslant L\|w - \tilde{w}\|, \quad \forall w, \tilde{w} \in \mathcal{N}. \tag{3.2}$$

It follows directly from Assumptions 1 and 2 that there exists a positive constant $M > 0$ such that

$$\|g(w)\| \leqslant M, \quad \forall w \in \mathcal{L}. \tag{3.3}$$

Furthermore, notice that since the error function E defined in Eq. (1.1) is bounded below in $\mathbb{R}^n$ by zero, it is differentiable and its gradient is Lipschitz continuous [19], Assumptions 1 and 2 always hold.

To establish the global convergence analysis of Algorithm MHS$^+$-CG, the following lemmas are needed. The proof of the following lemma is similar to that of Case II of Theorem 5.1 in [41], thus we omit it.

Lemma 3.1. *Suppose that Assumptions 1 and 2 hold. For θ_{k-1} and $\tilde{y}_{k-1}^*$ defined by Eqs. (2.3) and (2.4), respectively, we have*

$$|\theta_{k-1}| \leqslant L\|s_{k-1}\|^2 \quad \text{and} \quad \|\tilde{y}_{k-1}^*\| \leqslant 2L\|s_{k-1}\|.$$

The following lemma is a general result of conjugate gradient methods implemented using a line search that satisfies the strong Wolfe conditions (2.8) and (2.9).

Lemma 3.2 [52]. *Suppose that Assumptions 1 and 2 hold. Consider any method of the form 1.2,1.3 where d_k is a descent direction i.e. $d_k^T g_k < 0$ and the line search satisfies the strong Wolfe line search conditions (2.8) and (2.9), then*

$$\sum_{k \geqslant 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{3.4}$$

Clearly, it is easy to get from (2.7) that (3.4) is equivalent to the following inequality

$$\sum_{k \geqslant 0} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty. \tag{3.5}$$

Next, we establish the global convergence of Algorithm MHS⁺-CG. For the purpose, we state some properties for the search direction d_k, formula β_k^{MHS+} and step s_{k-1}.

Lemma 3.3. *Suppose that Assumptions 1 and 2 hold. Consider Algorithm MHS⁺-CG, if there exists a positive constant $\mu > 0$ such that for all $k \geqslant 0$*

$$\|g_k\| \geqslant \mu, \tag{3.6}$$

then there exist positive constants C and D such that

$$|\beta_k^{MHS+}| \leqslant C\|s_{k-1}\| \tag{3.7}$$

and

$$|\beta_k^{MHS+}| \frac{|g_k^T d_{k-1}|}{\|g_k\|^2} \leqslant D\|s_{k-1}\|. \tag{3.8}$$

Proof. From (2.4), (2.7) and (2.9), we have

$$d_{k-1}^T \tilde{y}_{k-1}^* \geqslant d_{k-1}^T y_{k-1} \geqslant (\sigma_2 - 1) g_{k-1}^T d_{k-1} = (1-\sigma_2)\|g_{k-1}\|^2.$$

Utilizing this with Lemma 3.1 and relations (3.3) and (3.6), we have

$$|\beta_k^{MHS+}| \leqslant \left| \frac{g_k^T \tilde{y}_{k-1}^*}{d_{k-1}^T \tilde{y}_{k-1}^*} \right| = \frac{2ML}{(1-\sigma_2)\mu^2}\|s_{k-1}\|.$$

Letting $C = \frac{2ML}{(1-\sigma_2)\mu^2}$, then (3.7) is satisfied. Furthermore, it follows from (2.7), (2.9), (3.3), (3.6) and (3.7) that

$$|\beta_k^{MHS+}| \frac{|g_k^T d_{k-1}|}{\|g_k\|^2} \leqslant |\beta_k^{MHS+}| \frac{\sigma_2 |g_{k-1}^T d_{k-1}|}{\|g_k\|^2} \leqslant |\beta_k^{MHS+}| \frac{\sigma_2 \|g_{k-1}\|^2}{\|g_k\|^2} \leqslant C\frac{\sigma_2 M^2}{\mu^2}\|s_{k-1}\|.$$

Letting $D = \frac{\sigma_2 C M^2}{\mu^2}$, we obtain (3.8). The proof is completed. □

Subsequently, we present a lemma which shows that, asymptotically, the search directions change slowly.

Lemma 3.4. *Suppose that Assumptions 1 and 2 hold. Consider Algorithm MHS⁺-CG, if there exists a positive constant $\mu > 0$ such that (3.6) holds, then $d_k \neq 0$ and*

$$\sum_{k \geqslant 1} \|u_k - u_{k-1}\|^2 < +\infty,$$

where $u_k = d_k/\|d_k\|$.

Proof. Firstly, note that $d_k \neq 0$, for otherwise (2.7) would imply $g_k = 0$. Therefore, u_k is well defined. Now, let us define

$$r_k := \frac{v_k}{\|d_k\|} \quad \text{and} \quad \delta_k := \beta_k^{MHS+} \frac{\|d_{k-1}\|}{\|d_k\|}, \tag{3.9}$$

where

$$v_k = -\left(1 + \beta_k^{MHS+} \frac{g_k^T d_{k-1}}{\|g_k\|^2}\right) g_k.$$

Then, by Eq. (2.6), we have

$$u_k = r_k + \delta_k u_{k-1}. \tag{3.10}$$

Using this relation with the identity $\|u_k\| = \|u_{k-1}\| = 1$, we obtain

$$\|r_k\| = \|u_k - \delta_k u_{k-1}\| = \|u_{k-1} - \delta_k u_k\|.$$

Moreover, using this with the condition $\delta_k \geqslant 0$ and the triangle inequality, we get

$$\|u_k - u_{k-1}\| \leqslant \|u_k - \delta_k u_{k-1}\| + \|u_{k-1} - \delta_k u_k\| = 2\|r_k\|. \tag{3.11}$$

It follows from the definition of v_k in Eq. (3.10) and relations (3.3) and (3.8) that

$$\|v_k\| \leqslant \left(1 + |\beta_k^{MHS+}| \frac{|g_k^T d_{k-1}|}{\|g_k\|^2}\right)\|g_k\| \leqslant (1+D)M. \tag{3.12}$$

Therefore, using this relation with (3.5) and (3.9), we have

$$\sum_{k\geqslant 1}\|r_k\|^2 \leqslant \sum_{k\geqslant 1}\frac{\|v_k\|^2}{\|d_k\|^2} \leqslant \sum_{k\geqslant 1}\frac{\|v_k\|^2}{\|g_k\|^4}\frac{\|g_k\|^4}{\|d_k\|^2} \leqslant \left(\frac{(1+D)M}{\mu^2}\right)^2\sum_{k\geqslant 1}\frac{\|g_k\|^4}{\|d_k\|^2} < +\infty.$$

which together with (3.11) completes the proof. □

Next, making use of Lemmas 3.3 and 3.4 we establish the global convergence theorem for Algorithm MHS$^+$-CG whose proof is similar to that Theorem 3.2 in [16], however we present it here for completeness.

Theorem 3.1. *Suppose that Assumptions 1 and 2 hold. If $\{w_k\}$ is obtained by Algorithm MHS$^+$-CG, then we have*

$$\liminf_{k\to\infty}\|g_k\| = 0. \tag{3.13}$$

Proof. We proceed by contraction, we suppose that the conclusion (3.13) is not true. That is, there exists a positive constant $\mu > 0$ such that for all k, $\|g_k\| \geqslant \mu$. The proof is divided in the following steps:

Step I. A bound on the step s_k. Observe that for any $l \geqslant k$, we have

$$w_l - w_k = \sum_{j=k}^{l-1}(w_{j+1} - w_j) = \sum_{j=k}^{l-1}\|s_j\|u_j = \sum_{j=k}^{l-1}\|s_j\|u_k + \sum_{j=k}^{l-1}\|s_j\|(u_j - u_k).$$

By the triangle inequality together with Assumption 1, we get

$$\sum_{j=k}^{l-1}\|s_j\| \leqslant \|w_l - w_k\| + \sum_{j=k}^{l-1}\|u_j - u_k\| \leqslant B + \sum_{j=k}^{l-1}\|s_j\|(u_j - u_k). \tag{3.14}$$

Let Δ be a positive integer, chosen large enough that

$$\Delta \geqslant 4BC, \tag{3.15}$$

where B and C are defined in (3.1) and (3.7), respectively. By Lemma 3.4, we can choose k_0 large enough such that

$$\sum_{i\geqslant k_0}\|u_{i+1} - u_i\|^2 \leqslant \frac{1}{4\Delta}. \tag{3.16}$$

For any $j > k \geqslant k_0$ with $j - k \leqslant \Delta$, using (3.16) with the Cauchy–Schwartz inequality, we obtain

$$\|u_j - u_k\| \leqslant \sum_{i=k}^{j-1}\|u_{i+1} - u_i\| \leqslant \sqrt{j-k}\left(\sum_{i=k}^{j-1}\|u_{i+1} - u_i\|\right)^{1/2} \leqslant \sqrt{\Delta}\left(\frac{1}{4\Delta}\right)^{1/2} = \frac{1}{2}.$$

Utilizing this with (3.14), yields

$$\sum_{j=k}^{l-1}\|s_j\| < 2B, \tag{3.17}$$

where $l > k > k_0$ and $l - k \geqslant \Delta$.

Step II. A bound on the search directions d_l. We rewrite (2.6) as follows

$$d_l = -g_l + \beta_k^{\mathrm{MHS}+}\left(I - \frac{g_l g_l^T}{\|g_l\|^2}\right)d_{l-1}. \tag{3.18}$$

Since g_l is orthogonal to $\left(I - \frac{g_l g_l^T}{\|g_l\|^2}\right)d_{l-1}$ and $I - \frac{g_l g_l^T}{\|g_l\|^2}$ is a project matrix, we have from(3.3), (3.7) and (3.18) that

$$\|d_l\|^2 \leqslant 2\|g_l\|^2 + 2|\beta_k^{\mathrm{MHS}+}|^2\|d_{l-1}\|^2 \leqslant 2M^2 + 2C^2\|s_{l-1}\|^2\|d_{l-1}\|^2.$$

Defining $S_i = 2C^2\|s_i\|^2$, we have that for $l > k_0$,

$$\|d_l\|^2 \leqslant 2M^2\left(\sum_{i=k_0+1}^{l}\prod_{j=i}^{l-1}S_j\right) + \|d_{k_0}\|^2\prod_{j=k_0}^{l-1}S_j. \tag{3.19}$$

Let us consider as follows a product of Δ consecutive S_j, where $k \geqslant k_0$. Utilizing (3.15) and (3.17) together with the Cauchy–Schwartz inequality, we get

$$\prod_{j=k}^{k+\Delta-1}S_j = \prod_{j=k}^{k+\Delta-1}2C^2\|s_j\|^2 \leqslant \left(\frac{\sum_{j=k}^{k+\Delta-1}\sqrt{2}C\|s_j\|}{\Delta}\right)^{2\Delta} \leqslant \left(\frac{2\sqrt{2}BC}{\Delta}\right)^{2\Delta} \leqslant \frac{1}{2^{\Delta}}.$$

Since the product of Δ consecutive S_j is bounded by $1/2^{\Delta}$, it immediate follows from (3.19) that

$$\|d_l\|^2 \leqslant c_1 l + c_2$$

for a certain positive constant c_1 independent of l. Therefore, we have from the previous relation that

$$\sum_{k\geqslant 0} \frac{\|g_k\|^4}{\|d_k\|^2} \geqslant \sum_{k\geqslant 0} \frac{\mu^2}{c_1 k + c_2} = +\infty,$$

which contradicts with (3.5). This completes the proof. □

4. Experimental results

In this section, we will present experimental results in order to evaluate the training and generalization performance of our proposed conjugate gradient algorithm MHS$^+$-CG in five famous classification problems acquired by the UCI Repository of Machine Learning Databases [31]: the Fisher's iris problem, the diabetes problem, the yeast problem, the sonar problem and the escherichia coli problem.

All networks had logistic activation functions and received the same sequence of input patterns. For evaluating classification accuracy we have used the standard procedure called *k-fold cross-validation* [23]. The implementation code was written in Matlab 6.5 based on the SCG code of Birgin and Martínez [4]. All methods are implemented with the line search proposed in CONMIN [39] which employs various polynomial interpolation schemes and safeguards in satisfying the strong Wolfe line search conditions. Moreover, for all training algorithms the parameters were set as $\sigma_1 = 10^{-4}$ and $\sigma_2 = 0.5$ for all experiments as in [27,44]. The simulations have been carried out on a Pentium computer (2.4 MHz, 512Mbyte RAM) running Windows XP operating system while the results have been averaged over 500 simulations.

The cumulative total for a performance metric over all simulations does not seem to be too informative, since a small number of simulations can tend to dominate these results. For this reason, we utilize the performance profiles of Dolan and Morè [9] to present perhaps the most complete information in terms of robustness, efficiency and solution quality. The use of performance profiles eliminate the influence of a small number of simulations on the benchmarking process and the sensitivity of results associated with the ranking of solvers [9]. The performance profile plots the fraction P of simulations for which any given method is within a factor τ of the best training method. The horizontal axis of each plot shows the percentage of the simulations for which a method is the fastest (efficiency), while the vertical axis gives the percentage of the simulations that the neural networks were successfully trained by each method (robustness). The reported performance profiles measure the efficiency and the robustness of our method in terms of computational time (CPU time) and function/gradient evaluations (FE/GE).

The curves in the following figures have the following meaning:

- "HS" stands for the Hestenes–Stiefel conjugate gradient method.
- "HS$^+$" stands for the Hestenes–Stiefel$^+$ conjugate gradient method.
- "AMSP" stands for the adaptive spectral Perry conjugate gradient method [28].
- "MHS$_1^+$" stands for Algorithm MHS$^+$-CG with $u = s_{k-1}$.
- "MHS$_2^+$" stands for Algorithm MHS$^+$-CG with $u = y_{k-1}$.

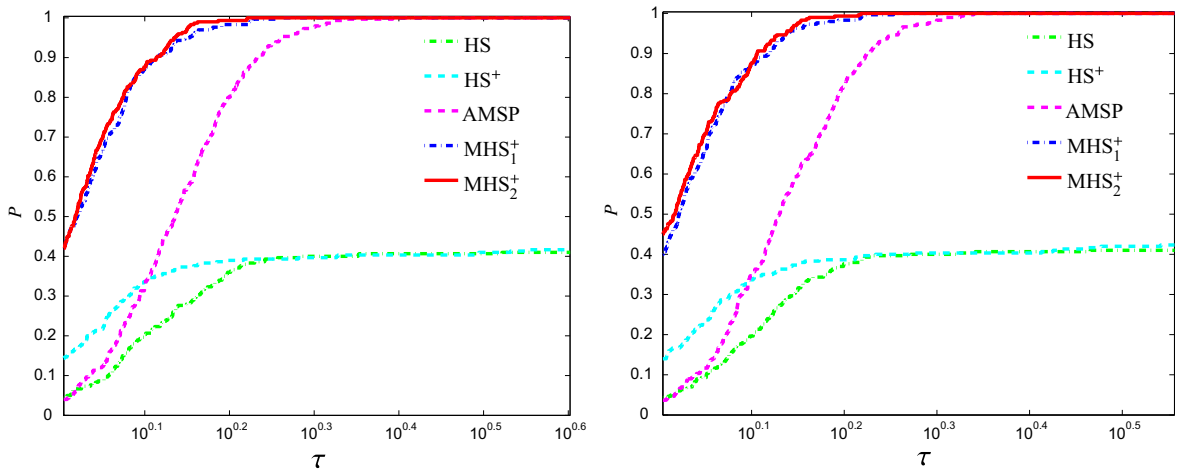
4.1. Fisher's Iris classification problem

This benchmark is perhaps the most best known to be found in the pattern recognition literature [31]. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The network architectures constitute of 1 hidden layer with 7 neurons and an output layer of 3 neurons. The termination criterion is set to $E_G \leqslant 0.01$ within the limit of 1000 epochs and all networks were tested using 10-fold cross-validation [27].

Fig. 1 illustrates the performance profiles for the iris classification problem, investigating the efficiency and robustness of each training method. Clearly, our proposed methods MHS$_1^+$ and MHS$_2^+$ significantly outperform the classical training methods since the curves of the former lie above the curves of the latter, regarding both performance metrics. More analytically, the performance profiles show that the probability of MHS$_1^+$ and MHS$_2^+$ to successfully train a neural network within a factor 1.65 of the best solver is 100%, in contrast with HS, HS$^+$ and AMSP which exhibit probability 38.6%, 39.7% and 87.6%, respectively.

4.2. Diabetes classification problem

The aim of this real-world classification task is to decide when a Pima Indian female is diabetes positive or not. The data of this benchmark consists of 768 different patterns each of them having 8 features of real continuous values and a class label (diabetic positive or not). We have used neural networks with 2 hidden layers of 4 neurons each and an output layer of 2



(a) Performance based on CPU time. (b) Performance based on FE/GE.

Fig. 1. Log_{10} scaled performance profiles for the iris classification problem.

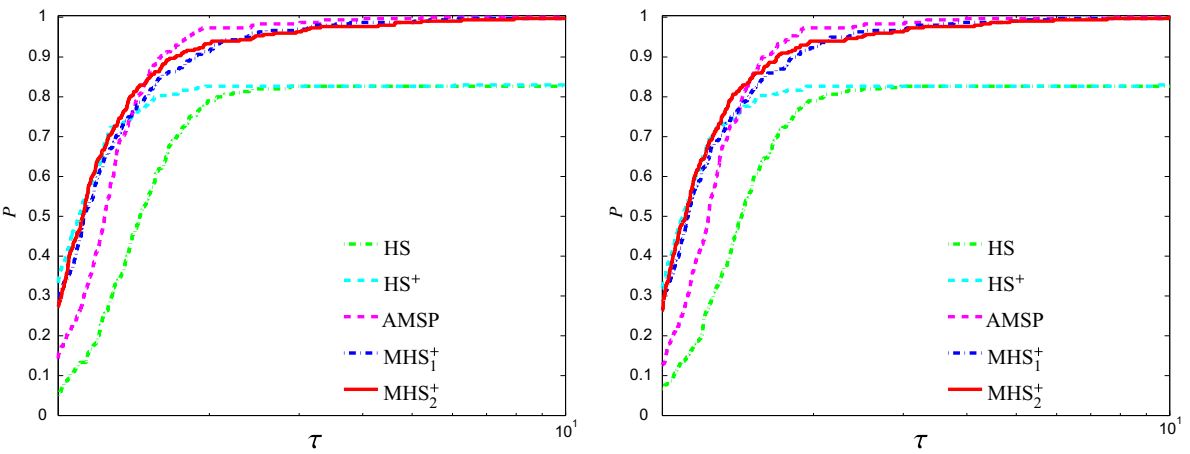
neurons [38]. The termination criterion is set to $E_G < 0.14$ within the limit of 2000 epochs and all networks were tested using 10-fold cross validation [46].

Fig. 2 presents the performance profiles for the diabetes classification problem, regarding both performance metrics. MHS_1^+ and MHS_2^+ exhibit the best overall performance, significantly outperforming the classical training methods HS and HS^+ and are more robust than AMSP. Notice that, MHS_1^+, MHS_2^+ and AMSP are the only methods reporting excellent (100%) probability of successful training. Moreover, as regards our proposed methods, they exhibit similar performance, relative to both performance metrics.

4.3. Yeast classification problem

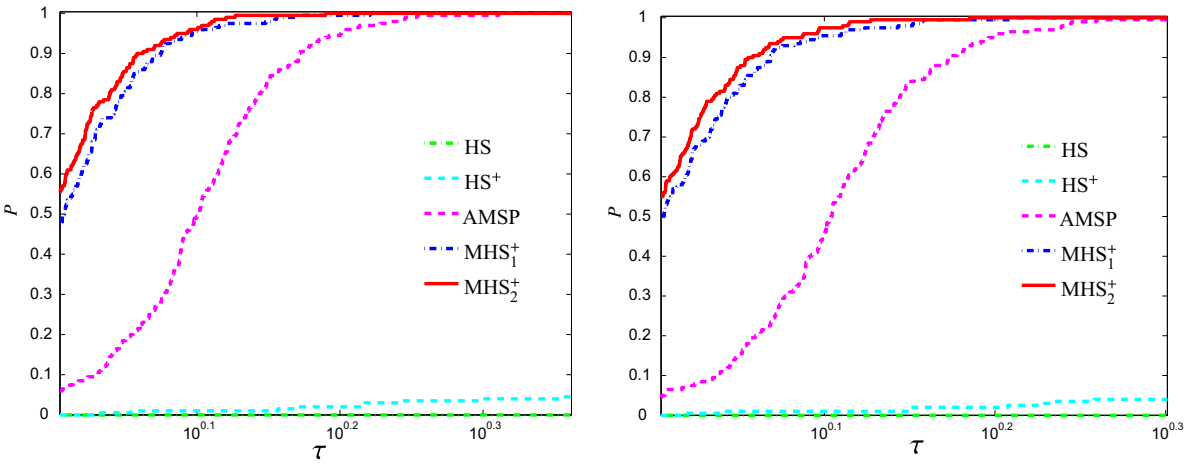
This problem is based on a drastically imbalanced dataset and concerns the determination of the cellular localization of the yeast proteins into ten localization sites. *Saccharomyces cerevisiae* (yeast) is the simplest Eukaryotic organism. The network architecture for this classification problem consists of 1 hidden layer of 16 neurons and an output layer of 10 neurons [2]. The termination criterion is set to $E_G < 0.05$ within the limit of 2000 epochs and all networks were tested using 10-fold cross validation [22].

Fig. 3 presents the performance profiles for the yeast classification problem. Obviously, our proposed methods MHS_1^+ and MHS_2^+ exhibit the best overall performance, significantly outperforming the classical training methods since the curves of the former lie above the curves of the latter, regarding both performance metrics. Moreover, it is worth noticing that MHS_1^+ and



(a) Performance based on CPU time. (b) Performance based on FE/GE.

Fig. 2. Log_{10} scaled performance profiles for the diabetes classification problem.



(a) Performance based on CPU time. (b) Performance based on FE/GE.

Fig. 3. Log_{10} scaled performance profiles for the yeast classification problem.

MHS_2^+ exhibit 50% and 55% of simulations with the least number of CPU time and computational cost while $AMSP^+$ reports only 9% of simulations, in the same situations.

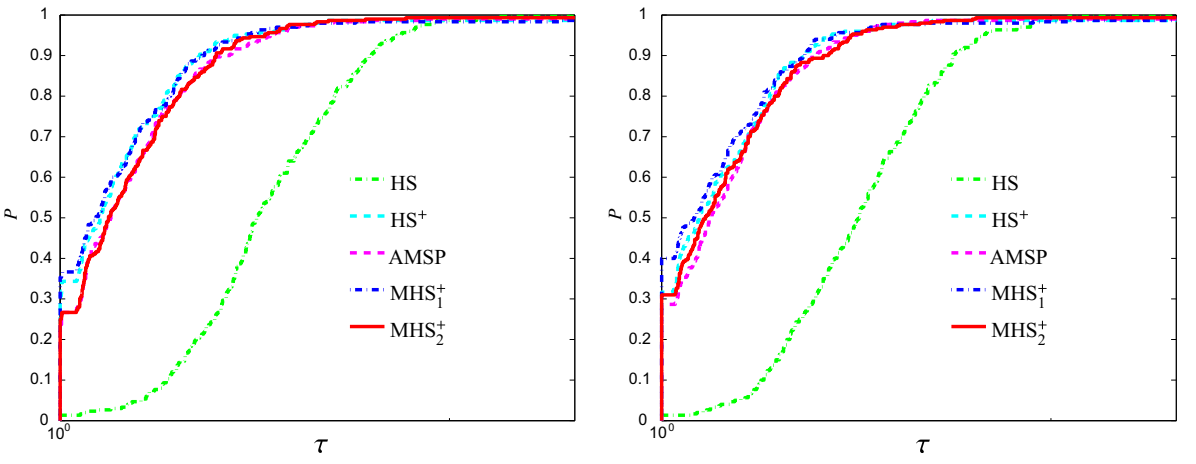
4.4. Sonar classification problem

This is the dataset used by Gorman and Sejnowski in their study of the classification of sonar signals using a neural network [15]. The transmitted sonar signal is a frequency-modulated chirp, rising in frequency. The dataset contains signals obtained from a variety of different aspect angles, spanning 90° for the cylinder and 180° for the rock. The network architecture for this problem constitute of 1 hidden layer with 24 neurons and an output layer of 2 neurons [15]. The termination criterion is set to $E_G \leqslant 0.1$ within the limit of 1000 epochs and all networks were tested using 3-fold cross validation [28].

Fig. 4 shows the performance profiles for the sonar classification problem investigating the efficiency and robustness of each training method. MHS_1^+ exhibits the best overall performance, relative to both performance metrics. More specifically, MHS_1^+ reports 5–35% more simulations with the least number of CPU time and computational cost than the other conjugate gradient training methods.

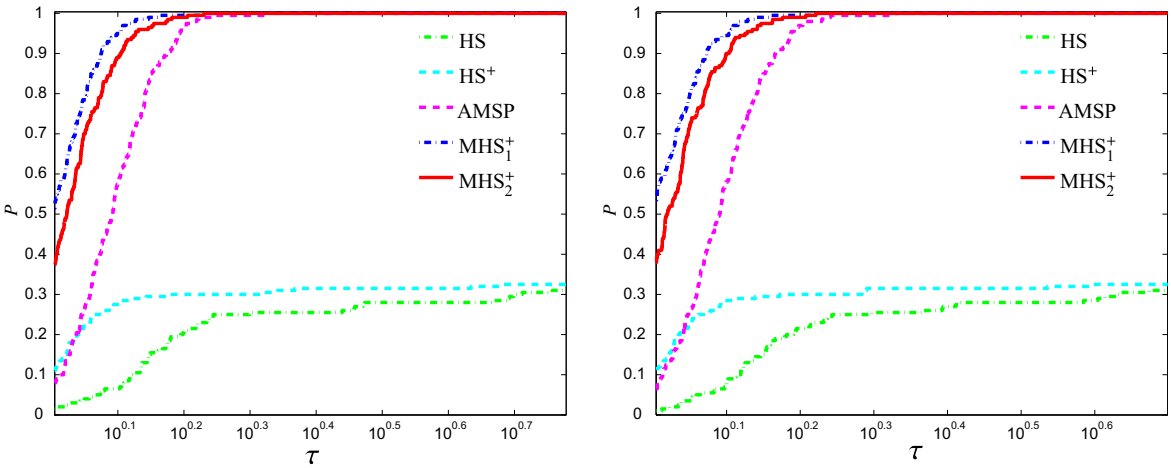
4.5. Escherichia coli classification problem

This problem is based on a drastically imbalanced data set of 336 patterns and concerns the classification of the *E. coli* protein localization patterns into eight localization sites. *E. coli*, being a prokaryotic gram-negative bacterium, is an



(a) Performance based on CPU time. (b) Performance based on FE/GE.

Fig. 4. Log_{10} scaled performance profiles for the sonar classification problem.



(a) Performance based on CPU time. (b) Performance based on FE/GE.

Fig. 5. Log_{10} scaled performance profiles for the escherichia coli classification problem.

Table 1
Generalization results of conjugate gradient methods HS, HS$^+$, AMSP, MHS_1^+ and MHS_2^+.

Method	Classification problems				
	Iris (%)	Diabetes (%)	Yeast (%)	Sonar (%)	E. coli (%)
HS	95.5	75.5	–	77.4	95.4
HS$^+$	95.7	75.8	92.2	77.6	95.4
AMSP	98	76.0	**92.7**	77.6	95.9
MHS_1^+	**98.1**	76.1	**92.7**	**77.7**	**96**
MHS_2^+	98	**76.3**	**92.7**	77.6	95.9

important component of the biosphere. Three major and distinctive types of proteins are characterized in *E. coli*: enzymes, transporters and regulators. The largest number of genes encodes enzymes (34%) (this should include all the cytoplasm proteins) followed by the genes for transport functions and the genes for regulatory process (11.5%) [26]. The network architectures constitute of 1 hidden layer with 16 neurons and an output layer of 8 neurons [2]. The termination criterion is set to $E_G \leqslant 0.02$ within the limit of 2000 epochs and all neural networks were tested using 4-fold cross-validation [22].

In Fig. 5 are presented the performance profiles for the escherichia coli classification problem. Similar observations can be made with the previous benchmarks. More specifically, the interpretation in Fig. 5 highlights that MHS_1^+ and MHS_2^+ are the best training methods since their curves lie on the top, with MHS_1^+ being more robust, relative to both performance metrics. More analytically, MHS_1^+ and MHS_2^+ exhibit 52% and 40% of simulations with the least number of CPU time and computational cost while HS, HS$^+$ and AMSP$^+$ report 1%, 10% and 9% of simulations, respectively in the same situation.

4.6. Generalization performance

In Table 1 are summarized the generalization results of the conjugate gradient methods HS, HS$^+$, AMSP$^+$, MHS_1^+ and MHS_2^+, measured by the percentage of testing patterns that were classified correctly in the presented classification problems. For each test problem, we present a table summarizing the performance of the algorithms for simulations that reached solution within a predetermined limit of epochs. Each row reports the average performance in percentage for each problem and the best conjugate gradient method for a problem is illustrated in boldface. Moreover, "–" means that the method reported 0% training success.

The interpretation on Table 1 illustrates that MHS_1^+ and MHS_2^+ significantly outperform the classical training algorithms in all classification problems. Furthermore, MHS_1^+ is an excellent generalizer since it manages to have the highest generalization performance in four out of five classification problems. Summarizing, it worths noticing that MHS_1^+ and MHS_2^+ are not only the best training methods but also they exhibit better generalization performance than the classical methods.

5. Conclusions

In this paper, we proposed a new conjugate gradient method which consists of a modification of the Hestenes–Stiefel method. An important property of our proposed method is that it ensures sufficient descent independent of the accuracy

of the line search. Moreover, it achieves a high-order accuracy in approximating the error surface by utilizing a new modified secant condition. Under mild conditions, we established the global convergence of our proposed method under the strong Wolfe line search conditions. Based on our experimental results, we concluded that our proposed method is efficient and has a potential to significantly enhance the computational efficacy of the training process.

References

[1] H. Adeli, S.L. Hung, An adaptive conjugate gradient learning algorithm for efficient training of neural networks, Applied Mathematics and Computation 62 (1) (1994) 81–102.
[2] A.D. Anastasiadis, G.D. Magoulas, M.N. Vrahatis, New globally convergent training scheme based on the resilient propagation algorithm, Neurocomputing 64 (2005) 253–270.
[3] S. Babaie-Kafaki, R. Ghanbari, N. Mahdavi-Amiri, Two new conjugate gradient methods based on modified secant equations, Journal of Computational and Applied Mathematics 234 (2010) 1374–1386.
[4] E.G. Birgin, J.M. Martínez, A spectral conjugate gradient method for unconstrained optimization, Applied Mathematics and Optimization 43 (1999) 117–128.
[5] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford, 1995.
[6] C. Charalambous, Conjugate gradient algorithm for efficient training of artificial neural networks, IEEE Proceedings 139 (3) (1992) 301–310.
[8] Y.H. Dai, Y.X. Yuan, Nonlinear Conjugate Gradient Methods, Shanghai Scientific and Technical Publishers, Shanghai, 2000.
[9] E. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2002) 201–213.
[10] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Computer Journal 7 (1964) 149–154.
[11] J.A. Ford, I.A. Moghrabi, Multi-step Quasi-Newton methods for optimization, Journal of Computational and Applied Mathematics 50 (1994) 305–323.
[12] J.A. Ford, I.A. Moghrabi, Using function-values multi-step Quasi-Newton methods, Journal of Computational and Applied Mathematics 66 (1996) 201–211.
[13] J.A. Ford, Y. Narushima, H. Yabe, Multi-step nonlinear conjugate gradient methods for unconstrained minimization, Computational Optimization and Applications 40 (2008) 191–216.
[14] J.C. Gilbert, J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM Journal of Optimization 2 (1) (1992) 21–42.
[15] R.P. Gorman, T.J. Sejnowski, Analysis of hidden units in a layered network trained to classify sonar targets, Neural Networks (1988) 75–89.
[16] W.W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM of Journal Optimization 16 (2005) 170–192.
[17] W.W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, Pacific of Journal Optimization 2 (2006) 35–58.
[18] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan, College Publishing Company, New York, 1994.
[19] J. Hertz, A. Krogh, R. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Reading, MA, 1991.
[20] M.R. Hestenes, E. Stiefel, Methods for conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards 49 (1952) 409–436.
[21] A. Hmich, A. Badri, A. Sahel, Automatic speaker identification by using the neural network, in: IEEE 2011 International Conference on Multimedia Computing and Systems (ICMCS), 2011, pp. 1–5.
[22] P. Horton, K. Nakai, Better prediction of protein cellular localization sites with the k-Nearest Neighbors classifier, in: Intelligent Systems in Molecular Biology, 1997, pp. 368–383.
[23] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: IEEE International Joint Conference on Artificial Intelligence, AAAI Press and MIT Press, 1995, pp. 223–228.
[24] B. Lerner, H. Guterman, M. Aladjem, I. Dinstein, A comparative study of neural network based feature extraction paradigms, Pattern Recognition Letters 20 (1) (1999) 7–14.
[25] G. Li, C. Tang, Z. Wei, New conjugacy condition and related new conjugate gradient methods for unconstrained optimization, Journal of Computational and Applied Mathematics 202 (2007) 523–539.
[26] P. Liang, B. Labedan, M. Riley, Physiological genomics of Escherichia coli protein families, Physiological Genomics 9 (2002) 15–26.
[27] I.E. Livieris, P. Pintelas, Performance evaluation of descent CG methods for neural network training, in: E.A. Lipitakis (Ed.), Ninth Hellenic European Research on Computer Mathematics & its Applications Conference (HERCMA'09), 2009, pp. 40–46.
[28] I.E. Livieris, P. Pintelas, An improved spectral conjugate gradient neural network training algorithm, International Journal on Artificial Intelligence Tools 21 (1) (2012).
[29] I.E. Livieris, D.G. Sotiropoulos, P. Pintelas, On descent spectral CG algorithms for training recurrent neural networks, in: 13th Panellenic Conference of Informatics, 2009, pp. 65–69.
[30] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.
[31] P.M. Murphy, D.W. Aha, UCI repository of machine learning databases, University of California, Department of Information and Computer Science, Irvine, CA, 1994.
[32] J. Nocedal, Theory of algorithms for unconstrained optimization, Acta Numerica 1 (1992) 199–242.
[33] J. Nocedal, S.J. Wright, Numerical Optimization, Springer-Verlag, New York, 1999.
[34] C.-C. Peng, G.D. Magoulas, Adaptive nonmonotone conjugate gradient training algorithm for recurrent neural networks, in: 19th IEEE International Conference on Tools with Artificial Intelligence, 2008, pp. 374–381.
[35] C.-C. Peng, G.D. Magoulas, Advanced adaptive nonmonotone conjugate gradient training algorithm for recurrent neural networks, International Journal on Artificial Intelligence Tools 17 (5) (2008) 963–984.
[36] E. Polak, G. Ribière, Note sur la convergence de methods de directions conjuguees, Revue Francais d'Informatique et de Recherche Operationnelle 16 (1969) 35–43.
[37] M.J.D. Powell, Nonconvex minimization calculations and the conjugate gradient method, in: Numerical Analysis, Lecture Notes in Mathematics, vol. 1066, Springer-Verlag, Berlin, 1984, pp. 122–141.
[38] L. Prechelt, PROBEN1-A set of benchmarks and benchmarking rules for neural network training algorithms, Technical Report 21/94, Fakultt fr Informatik, University of Karlsruhe, 1994.
[39] D.F. Shanno, K.H. Phua, Minimization of unconstrained multivariate functions, ACM Transactions on Mathematical Software 2 (1976) 87–94.
[40] J. Wang, W. Wu, M. Zurada, Deterministic convergence of conjugate gradient method for feedforward neural networks, Neurocomputing 74 (2011) 2368–2376.
[41] Z. Wei, G. Li, L. Qi, New Quasi-Newton methods for unconstrained optimization problems, Applied Mathematics and Computation 175 (2006) 1156–1188.
[42] C.H. Wu, H.L. Chen, S.C. Chen, Gene classification artificial neural system, International Journal on Artificial Intelligence Tools 4 (4) (1995) 501–510.
[43] H. Yabe, M. Takano, Global convergence properties of nonlinear conjugate gradient methods with modified secant condition, Computational Optimization and Applications 28 (2004) 203–225.
[44] G. Yu, L. Guan, W. Chen, Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization, Optimization Methods and Software 23 (2) (2008) 275–293.

[45] G.H. Yu, Nonlinear self-scaling conjugate gradient methods for large-scale optimization problems, Ph.D. Thesis, Sun Yat-Sen University, 2007.
[46] J. Yu, S. Wang, L. Xi, Evolving artificial neural networks using an improved PSO and DPSO, Neurocomputing 71 (2008) 1054–1060.
[47] J.Z. Zhang, N.Y. Deng, L.H. Chen, New Quasi-Newton equation and related methods for unconstrained optimization, Journal of Optimization Theory and Applications 102 (1999) 147–167.
[48] J.Z. Zhang, C.X. Xu, Properties and numerical performance of Quasi-Newton methods with modified Quasi-Newton equations, Journal of Computational and Applied Mathematics 137 (2001) 269–278.
[49] L. Zhang, W. Zhou, D. Li, A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence, IMA Journal of Numerical Analysis 26 (4) (2006) 629–640.
[50] L. Zhang, W. Zhou, D. Li, Global convergence of a modified Fletcher–Reeves conjugate gradient method with Armijo-type line search, Numerische Mathematik 104 (2006) 561–572.
[51] L. Zhang, W. Zhou, D. Li, Some descent three-term conjugate gradient methods and their global convergence, Optimization Methods and Software 22 (2007) 697–711.
[52] G. Zoutendijk, Nonlinear programming, in: J. Abadie (Ed.), Integer and Nonlinear Programming, North Holland, Amsterdam, 1970, pp. 37–86.