Research Article

An Advanced Conjugate Gradient Training Algorithm Based on a Modified Secant Equation

Ioannis E. Livieris^{1,2} and Panagiotis Pintelas^{1,2}

¹ Department of Mathematics, University of Patras, 26500 Patras, Greece

² Educational Software Development Laboratory, Department of Mathematics, University of Patras, 26500 Patras, Greece

Correspondence should be addressed to Ioannis E. Livieris, livieris@upatras.gr

Received 5 August 2011; Accepted 4 September 2011

Academic Editors: T. Kurita and Z. Liu

Copyright © 2012 I. E. Livieris and P. Pintelas. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Conjugate gradient methods constitute excellent neural network training methods characterized by their simplicity, numerical efficiency, and their very low memory requirements. In this paper, 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 achieves a high-order accuracy in approximating the second-order curvature information of the error surface by utilizing the modified secant condition proposed by Li et al. (2007). Under mild conditions, we establish that the proposed method is globally convergent for general functions under the strong Wolfe conditions. Experimental results provide evidence that our proposed method is preferable and in general superior to the classical conjugate gradient methods and has a potential to significantly enhance the computational efficiency and robustness of the training process.

1. Introduction

Learning systems, such as multilayer feedforward neural networks (FNN), 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 successfully applied in many areas of artificial intelligence [1–5] and are often found to be more efficient and accurate than other classification techniques [6]. The operation of a FNN is usually based on the following equations:

$$net_j^l = \sum_{i=1}^{N_{l-1}} w_{ij}^{l-1,l} y_i^{l-1} + b_j^l, \quad y_j^l = f(net_j^l), \quad (1)$$

where net_j^l is the sum of its weighted inputs for the j th node in the l th layer ($j = 1, \dots, N_l$), $w_{ij}^{l-1,l}$ are the weights from the i th neuron at the $(l-1)$ layer to the j th neuron at the l th layer, b_j^l is the bias of the j th neuron at the l th layer, y_j^l is the output

of the j th neuron that belongs to the l th layer, and $f(net_j^l)$ is the j th neuron activation function.

The problem of *training* a neural network is to iteratively adjust its weights, in order to globally minimize a measure of difference between the actual output of the network and the desired output for all examples of the training set [7]. More mathematically, the training process can be formulated as the minimization of the error function $E(w)$, defined by the sum of square differences between the actual output of the FNN, denoted by $y_{j,p}^L$ and the desired output, denoted by $t_{j,p}$, relative to the appeared output, namely,

$$E(w) = \sum_{p=1}^P \sum_{j=1}^{N_L} (y_{j,p}^L - t_{j,p})^2, \quad (2)$$

where $w \in \mathbb{R}^n$ is the vector network weights and P represents the number of patterns used in the training set.

Conjugate gradient methods are probably the most famous iterative methods for efficiently training neural networks due to their simplicity, numerical efficiency, and their

very low memory requirements. These methods generate a sequence of weights $\{w_k\}$ using the iterative formula

$$w_{k+1} = w_k + \eta_k d_k, \quad k = 0, 1, \dots, \quad (3)$$

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

$$d_k = \begin{cases} -g_0, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{otherwise,} \end{cases} \quad (4)$$

where g_k is the gradient of E at w_k and β_k is a scalar. In the literature, there have been proposed several choices for β_k which give rise to distinct conjugate gradient methods. The most well-known conjugate gradient methods include the Fletcher-Reeves (FR) method [8], the Hestenes-Stiefel (HS) method [9], and the Polak-Ribière (PR) method [10]. The update parameters of these methods are, respectively, specified as follows:

$$\beta_k^{\text{HS}} = \frac{g_k^T y_{k-1}}{y_{k-1}^T d_{k-1}}, \quad \beta_k^{\text{FR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \beta_k^{\text{PR}} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \quad (5)$$

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

The PR method behaves like the HS method in practical computation and it is generally believed to be one of the most efficient conjugate gradient methods. However, despite the practical advantages of this method, it has the major drawback of not being globally convergent for general functions and as a result it may be trapped and cycle infinitely without presenting any substantial progress [11]. For rectifying the convergence failure of the PR method, Gilbert and Nocedal [12], motivated by Powell's work [13], proposed to restrict the update parameter β_k of being nonnegative, namely, $\beta_k^{\text{PR}+} = \max\{\beta_k^{\text{PR}}, 0\}$. The authors conducted an elegant analysis of this conjugate gradient method (PR+) and established that it is globally convergent under strong assumptions. Moreover, although that the PR method and the PR+ method usually perform better than the other conjugate gradient methods, they cannot guarantee to generate descent directions, hence restarts are employed in order to guarantee convergence. Nevertheless, there is also a worry with restart algorithms that their restarts may be triggered too often; thus degrading the overall efficiency and robustness of the minimization process [14].

During the last decade, much effort has been devoted to develop new conjugate gradient methods which are not only globally convergent for general functions but also computationally superior to classical methods and are classified in two classes. The first class utilizes second-order information to accelerate conjugate gradient methods by utilizing new secant equations (see [15–18]). Sample works include the nonlinear conjugate gradient methods proposed by Zhang et al. [19–21] which are based on MBFGS secant equation [15]. Ford et al. [22] proposed a multistep conjugate gradient method that is based on the multistep quasi-Newton methods proposed in [16, 17]. Recently, Yabe and Takano [23] and

Li et al. [18] proposed conjugate gradient methods which are based on modified secant equation using both the gradient and function values 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 do not ensure to generate descent directions; therefore the descent condition is usually assumed in their analysis and implementations.

The second class aims at developing conjugate gradient methods which generate descent directions, in order to avoid the usually inefficient restarts. On the basis of this idea, Zhang et al. [20, 24–26] modified the search direction in order to ensure sufficient descent, that is, $d_k^T g_k = -\|g_k\|^2$, independent of the performed line search. Independently, Hager and Zhang [27] modified the parameter β_k and proposed a new descent conjugate gradient method, called the CG-DESCENT method. More analytically, they proposed a modification of the Hestenes-Stiefel formula β_k^{HS} in the following way:

$$\beta_k^{\text{HZ}} = \beta_k^{\text{HS}} - 2 \frac{\|y_{k-1}\|^2}{(d_{k-1}^T y_{k-1})^2} g_k^T d_{k-1}. \quad (6)$$

Along this line, Yuan [28] based on [12, 27, 29], proposed a modified PR method, that is,

$$\beta_k^{\text{DPR}+} = \beta_k^{\text{PR}} - \min \left\{ \beta_k^{\text{PR}}, C \frac{\|y_{k-1}\|^2}{\|g_{k-1}\|^4} g_k^T d_{k-1} \right\}, \quad (7)$$

where C is a parameter which essentially controls the relative weight between conjugacy and descent and in case $C > 1/4$ then the above formula satisfies $g_k^T d_k \leq -(1 - 1/4C)\|g_k\|^2$. An important feature of this method is that it is globally convergent for general functions. Recently, Livieris et al. [30–32] motivated by the previous works 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 training process.

In this paper, we proposed a new conjugate gradient training algorithm which has both characteristics of the previous presented classes. Our method ensures sufficient descent independent of the accuracy of the line search, avoiding thereby the usually inefficient restarts. Moreover, it achieves a high-order accuracy in approximating the second-order curvature information of the error surface by utilizing the modified secant condition proposed in [18]. 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 our proposed conjugate gradient training algorithm and in Section 3, we present its global convergence analysis. The experimental results are reported in Section 4 using the performance profiles of Dolan and Moré [33]. Finally, Section 5 presents our concluding remarks.

2. Modified Polak-Ribière⁺ Conjugate Gradient 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 a nonlinear function E is updated so that a new matrix B_k satisfies the following secant condition:

$$B_k s_{k-1} = y_{k-1}. \quad (8)$$

Obviously, only two gradients are exploited in the secant equation (8), while the function values available are neglected. Recently, Li et al. [18] proposed a conjugate gradient method based on the modified secant condition

$$B_k s_{k-1} = \tilde{y}_{k-1}, \quad \tilde{y}_{k-1} = y_{k-1} + \frac{\max\{\theta_{k-1}, 0\}}{\|s_{k-1}\|^2} s_{k-1}, \quad (9)$$

where θ_{k-1} is defined by

$$\theta_{k-1} = 2(E_{k-1} - E_k) + (g_k + g_{k-1})^T s_{k-1}, \quad (10)$$

and E_k denotes $E(w_k)$. The authors proved that this new secant equation (9) is superior to the classical one (8) in the sense that $\tilde{y}_{k-1}$ better approximates $\nabla^2 E(w_k) s_{k-1}$ than y_{k-1} (see [18]).

Motivated by the theoretical advantages of this modified secant condition (9), we propose a modification of formula (7), in the following way:

$$\beta_k^{\text{MPR}^+} = \frac{g_k^T \tilde{y}_{k-1}}{\|g_{k-1}\|^2} - \min \left\{ \frac{g_k^T \tilde{y}_{k-1}}{\|g_{k-1}\|^2}, C \frac{\|\tilde{y}_{k-1}\|^2}{\|g_{k-1}\|^4} g_k^T d_{k-1} \right\}, \quad (11)$$

with $C > 1/4$. It is easy to see from (4) and (11) that our proposed formula $\beta_k^{\text{MPR}^+}$ satisfies the sufficient descent condition

$$g_k^T d_k \leq -\left(1 - \frac{1}{4C}\right) \|g_k\|^2, \quad (12)$$

independent of the line search used.

At this point, we present a high level description of our proposed algorithm, called modified Polak-Ribière⁺ conjugate gradient algorithm (MPR⁺-CG).

Algorithm 1 (modified Polak-Ribière⁺ conjugate gradient algorithm).

Step 1. Initiate w_0 , $0 < \sigma_1 < \sigma_2 < 1$, 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 $w_* = w_k$ and $E_* = E_k$.

Step 4. If $(g_k = 0)$, return “Error goal not met”.

Step 5. Compute the descent direction d_k using (4) and (11).

Step 6. Compute the learning rate η_k using the strong Wolfe line search conditions

$$E(w_k + \alpha_k d_k) - E(w_k) \leq \sigma_1 \alpha_k g_k^T d_k, \quad (13)$$

$$|g(w_k + \alpha_k d_k)^T d_k| \leq \sigma_2 |g_k^T d_k|. \quad (14)$$

Step 7. Update the weights

$$w_{k+1} = w_k + \eta_k d_k \quad (15)$$

and set $k = k + 1$.

Step 8. If $(k > k_{\text{MAX}})$ return “error goal not met”, else go to Step 2.

3. Global Convergence Analysis

In order to establish the global convergence result for our proposed method, we will impose the following assumptions on the error function E .

Assumption 1. The level set $\mathcal{L} = \{w \in \mathbb{R}^n \mid E(w) \leq E(w_0)\}$ is bounded.

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

$$\|g(w) - g(\tilde{w})\| \leq L \|w - \tilde{w}\|, \text{ and } \forall w, \tilde{w} \in \mathcal{N}. \quad (16)$$

Since $\{E(w_k)\}$ is a decreasing sequence, it is clear that the sequence $\{w_k\}$ is contained in $\mathcal{L}$. In addition, it follows directly from Assumptions 1 and 2 that there exist positive constraints B and M , such that

$$\|w - \tilde{w}\| \leq B, \quad \forall w, \tilde{w} \in \mathcal{L}, \quad (17)$$

$$\|g(w)\| \leq M, \quad \forall w \in \mathcal{L}. \quad (18)$$

Furthermore, notice that since the error function E is bounded below in $\mathbb{R}^n$ by zero, it is differentiable and its gradient is Lipschitz continuous [34]. Assumptions 1 and 2 always hold.

The following lemma is very useful for the global convergence analysis.

Lemma 2 (see [18]). *Suppose that Assumptions 1 and 2 hold and the line search satisfies the strong Wolfe line search conditions (13) and (14). For θ_k and $\tilde{y}_k$ defined in (10) and (9), respectively, one has*

$$|\theta_k| \leq L \|s_k\|^2, \quad \|\tilde{y}_k\| \leq 2L \|s_k\|. \quad (19)$$

Subsequently, we will establish the global convergence of Algorithm MPR⁺-CG for general functions. Firstly, we present a lemma that Algorithm MPR⁺-CG prevents the inefficient behavior of the jamming phenomenon [35] from occurring. This property is similar to but slightly different from Property (*), which was derived by Gilbert and Nocedal [12].

Lemma 3. *Suppose that Assumptions 1 and 2 hold. Let $\{w_k\}$ and $\{d_k\}$ be generated by Algorithm MPR⁺-CG, if there exists a positive constant $\mu > 0$ such that*

$$\|g_k\| \geq \mu, \quad \forall k \geq 0, \quad (20)$$

then there exist constants $b > 1$ and $\lambda > 0$ such that

$$|\beta_k^{\text{MPR}^+}| \leq b, \quad (21)$$

$$\|s_{k-1}\| \leq \lambda \implies |\beta_k^{\text{MPR}^+}| \leq \frac{1}{b}. \quad (22)$$

Proof. Utilizing Lemma 2 together with Assumption 2 and relations (12), (14), (17), (18), (20) we have

$$\begin{aligned} |\beta_k^{\text{MPR}^+}| &\leq \frac{|g_k^T \tilde{y}_{k-1}|}{\|g_{k-1}\|^2} + C \frac{\|\tilde{y}_{k-1}\|^2}{\|g_{k-1}\|^4} |g_k^T d_{k-1}| \\ &\leq \frac{\|g_k\| \|\tilde{y}_{k-1}\|}{\|g_{k-1}\|^2} + C \frac{\|\tilde{y}_{k-1}\|^2 \sigma_2 \|g_{k-1}^T d_{k-1}\|}{\|g_{k-1}\|^4} \\ &\leq \frac{2ML\|s_{k-1}\|}{\|g_{k-1}\|^2} + C \frac{4L^2\|s_{k-1}\|^2}{\|g_{k-1}\|^4} \sigma_2 \left(1 - \frac{1}{4C}\right) \|g_{k-1}\|^2 \\ &\leq \left(\frac{2ML + 4L^2 B C \sigma_2 (1 - 1/4C)}{\mu^2}\right) \|s_{k-1}\| \triangleq D \|s_{k-1}\|. \end{aligned} \quad (23)$$

Therefore, by setting $b := \max\{2, 2DB\}$ and $\lambda := 1/Db$, we have relations (21) and (22) hold. The proof is completed. $\square$

Subsequently, we present a lemma which shows that, asymptotically, the search directions w_k change slowly. This lemma corresponds to Lemma 4.1 of Gilbert and Nocedal [12] and the proof is exactly the same as that of Lemma 4.1 in [12], thus we omit it.

Lemma 4. Suppose that Assumptions 1 and 2 hold. Let $\{w_k\}$ and $\{d_k\}$ be generated by Algorithm MPR⁺-CG, if there exists a positive constant $\mu > 0$ such that (21) holds; then $d_k \neq 0$ and

$$\sum_{k \geq 1} \|u_k - u_{k-1}\|^2 < \infty, \quad (24)$$

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

Next, by making use of Lemmas 3 and 4, we establish the global convergence theorem for Algorithm MPR⁺-CG under the strong Wolfe line search.

Theorem 5. Suppose that Assumptions 1 and 2 hold. If $\{w_k\}$ is obtained by Algorithm MPR⁺-CG where the line search satisfies the strong Wolfe line search conditions (13) and (14), then one has

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (25)$$

Proof. We proceed by contraction. Suppose that there exists a positive constant $\mu > 0$ such that for all $k \geq 0$

$$\|g_k\| \geq \mu. \quad (26)$$

The proof is divided in the following two steps.

Step I. A bound on the step s_k . Let Δ be a positive integer, chosen large enough that

$$\Delta \geq 4BD, \quad (27)$$

where B and D are defined in (17) and (23), respectively. For any $l > k \geq k_0$ with $l - k \leq \Delta$, following the same proof as the case II of Theorem 3.2 in [27], we get

$$\sum_{j=k}^{l-1} \|s_j\| < 2B. \quad (28)$$

Step II. A bound on the search directions of d_l . It follows from the definition of d_k in (4) together with (18) and (23), we obtain

$$\|d_l\|^2 \leq \|g_l\|^2 + |\beta_k^{\text{MPR}^+}|^2 \|d_{l-1}\|^2 \leq M^2 + D^2 \|s_{l-1}\|^2 \|d_{l-1}\|^2. \quad (29)$$

Now, the remaining argument is standard in the same way as case III in Theorem 3.2 in [27], thus we omit it. This completes the proof. $\square$

4. Experimental Results

In this section, we will present experimental results in order to evaluate the performance of our proposed conjugate gradient algorithm MPR⁺-CG in five famous classification problems acquired by the UCI Repository of Machine Learning Databases [36]: the iris problem, the diabetes problem, the sonar problem, the yeast problem, and the *Escherichia coli* problem.

The implementation code was written in Matlab 6.5 on a Pentium IV computer (2.4 MHz, 512 Mbyte RAM) running Windows XP operating system based on the SCG code of Birgin and Martínez [37]. All methods are implemented with the line search proposed in CONMIN [38] which employs various polynomial interpolation schemes and safeguards in satisfying the strong Wolfe line search conditions. The heuristic parameters were set as $\sigma_1 = 10^{-4}$ and $\sigma_2 = 0.5$ as in [30, 39]. All networks have received the same sequence of input patterns and the initial weights were generated using the Nguyen-Widrow method [40]. For evaluating classification accuracy we have used the standard procedure called *k-fold cross-validation* [41]. The results have been averaged over 500 simulations.

4.1. Training Performance. 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 use the performance profiles proposed by Dolan and Moré [33] to present perhaps the most complete information in terms of robustness, efficiency, and solution quality. 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),

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

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 have been created using the Libopt environment [42] for measuring 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.

- (i) “PR” stands for the Polak-Ribière conjugate gradient method.
- (ii) “PR+” stands for the Polak-Ribière+ conjugate gradient method.
- (iii) “MPR+” stands for Algorithm MPR+ -CG.

4.1.1. Iris Classification Problem. This benchmark is perhaps the most best known to be found in the pattern-recognition literature [36]. 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 training goal was set to $E_G \leq 0.01$ within the limit of 1000 epochs and all networks were tested using 10-fold cross-validation [30].

Figure 1 presents the performance profiles for the iris classification problem, regarding both performance metrics. MPR+ illustrates the best performance in terms of efficiency and robustness, significantly outperforming the classical training methods PR and PR+. Furthermore, the performance profiles show that MPR+ is the only method reporting an excellent (100%) probability of being the optimal training method.

4.1.2. Diabetes Classification Problem. The aim of this real-world classification task is to decide whether 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 (diabetes positive or not). We have used neural networks with 2 hidden layers of 4 neurons each and an output layer of 2 neurons [43]. The training goal was set to $E_G < 0.14$ within the limit of 2000 epochs and all networks were tested using 10-fold cross-validation [44].

Figure 2 illustrates the performance profiles for the diabetes classification problem, investigating the efficiency and robustness of each training method. Clearly, our proposed method MPR+ significantly outperforms the conjugate gradient methods PR and PR+ 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 MPR+ to successfully train a neural network within a factor 3.41 of the best solver is 100%, in contrast with PR and PR+ which have probability 84.3% and 85%, respectively.

4.1.3. Sonar Classification Problem. This is the dataset used by Gorman and Sejnowski [45] in their study of the classification of sonar signals using a neural network. The dataset contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock. The network architecture for this problem constitutes of 1 hidden layer of 24 neurons and an output layer of 2 neurons [45]. The training goal was set to $E_G = 0.1$ within the limit of 1000 epochs and all networks were tested using 3-fold cross-validation [45].

FIGURE 2: Log₁₀ scaled performance profiles for the diabetes classification problem.

FIGURE 3: Log₁₀ scaled performance profiles for the sonar classification problem.

In Figure 3 are presented the performance profiles for the sonar classification problem, relative to both performance metrics. Our proposed conjugate gradient method MPR⁺ presents the highest probability of being the optimal training method. Furthermore, MPR⁺ significantly outperforms PR and is slightly more robust than PR⁺, regarding both performance metrics.

4.1.4. 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 [46]. The training

FIGURE 4: Log_{10} scaled performance profiles for the yeast classification problem.

goal was set to $E_G < 0.05$ within the limit of 2000 epochs and all networks were tested using 10-fold cross validation [47].

Figure 4 presents the performance profiles for the yeast classification problem, regarding both performance metrics. The interpretation in Figure 4 highlights that our proposed conjugate gradient method MPR^+ is the only method exhibiting an excellent (100%) probability of successful training. Moreover, it is worth noticing that PR and PR^+ report very poor performance exhibiting 0% and 5% probability of successfully training, respectively, in contrast with our proposed method MPR^+ which has successfully trained all neural networks.

4.1.5. *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 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 encoding enzymes (34%) (this should include all the cytoplasm proteins) is followed by the genes for transport functions and the genes for regulatory process (11.5%) [48]. The network architectures constitute of 1 hidden layer with 16 neurons and an output layer of 8 neurons [46]. The training goal was set to $E_G \leq 0.02$ within the limit of 2000 epochs and all neural networks were tested using 4-fold cross-validation [47].

In Figure 5 are presented the performance profiles for the *Escherichia coli* classification problem. Similar observations can be made with the previous benchmarks. More specifically, MPR^+ significantly outperforms the classical training

TABLE 1: Generalization results of conjugate gradient methods PR, PR^+ , and MPR^+ .

Method	Classification problems				
	Iris	Diabetes	Sonar	Yeast	<i>E.coli</i>
PR	95.0%	76.1%	75.9%	—	95.8%
PR^+	95.3%	76.1%	75.6%	92.0%	96.0%
MPR^+	98.1%	76.4%	75.9%	92.5%	96.0%

methods PR and PR^+ , since the curves of the former lie above the curves of the latter, regarding both performance metrics. Moreover the performance profiles show that the probability of MPR^+ is the only method reporting excellent (100%) probability of being the optimal training method.

4.2. Generalization Performance. In Table 1 are summarized the generalization results of PR, PR^+ , and MPR^+ conjugate gradient methods, measured by the percentage of testing patterns that were classified correctly in the presented classification problems. 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 MPR^+ is an excellent generalizer since it manages to have the highest generalization performance, outperforming the classical training methods PR and PR^+ in all classification problems.

5. Conclusions

In this paper, we proposed a conjugate gradient method for efficiently training neural networks. An attractive property

FIGURE 5: Log_{10} scaled performance profiles for the *Escherichia coli* classification problem.

of our proposed method is that it ensures sufficient descent, avoiding thereby the usually inefficient restarts. Furthermore, it achieves a high-order accuracy in approximating the second-order curvature information of the error surface by utilizing the modified secant equation proposed in [18]. Under mild conditions, we established that our proposed method is globally convergent. Based on our numerical experiments, we concluded that our proposed method outperforms classical conjugate gradient training methods and has a potential to significantly enhance the computational efficiency and robustness of the training process.

References

- [1] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford, UK, 1995.
- [2] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College, New York, NY, USA, 1994.
- [3] A. Hmich, A. Badri, and A. Sahel, "Automatic speaker identification by using the neural network," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS '11)*, pp. 1–5, April 2011.
- [4] H. Takeuchi, Y. Terabayashi, K. Yamauchi, and N. Ishii, "Improvement of robot sensor by integrating information using neural network," *International Journal on Artificial Intelligence Tools*, vol. 12, no. 2, pp. 139–152, 2003.
- [5] C. H. Wu, H. L. Chen, and S. C. Chen, "Gene classification artificial neural system," *International Journal on Artificial Intelligence Tools*, vol. 4, no. 4, pp. 501–510, 1995.
- [6] B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein, "A comparative study of neural network based feature extraction paradigms," *Pattern Recognition Letters*, vol. 20, no. 1, pp. 7–14, 1999.
- [7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. Rumelhart and J. McClelland, Eds., pp. 318–362, Cambridge, Mass, USA, 1986.
- [8] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *Computer Journal*, vol. 7, pp. 149–154, 1964.
- [9] M. R. Hestenes and E. Stiefel, "Methods for conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.
- [10] E. Polak and G. Ribière, "Note sur la convergence de methods de directions conjuguées," *Revue Francais d'Informatique et de Recherche Operationnelle*, vol. 16, pp. 35–43, 1969.
- [11] M. J. D. Powell, "Nonconvexminimization calculations and the conjugate gradientmethod," in *Numerical Analysis*, vol. 1066 of *Lecture notes in mathematics*, pp. 122–141, Springer, Berlin, Germany, 1984.
- [12] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," *SIAM Journal of Optimization*, vol. 2, no. 1, pp. 21–42, 1992.
- [13] M. J. D. Powell, "Convergence properties of algorithms for nonlinear optimization," *Siam Review*, vol. 28, no. 4, pp. 487–500, 1986.
- [14] J. Nocedal, "Theory of algorithms for unconstrained optimization," *Acta Numerica*, vol. 1, pp. 199–242, 1992.
- [15] D. H. Li and M. Fukushima, "A modified BFGS method and its global convergence in nonconvex minimization," *Journal of Computational and Applied Mathematics*, vol. 129, no. 1-2, pp. 15–35, 2001.
- [16] J. A. Ford and I. A. Moghrabi, "Multi-step quasi-Newton methods for optimization," *Journal of Computational and Applied Mathematics*, vol. 50, no. 1-3, pp. 305–323, 1994.
- [17] J. A. Ford and I. A. Moghrabi, "Using function-values in multi-step quasi-Newton methods," *Journal of Computational and Applied Mathematics*, vol. 66, no. 1-2, pp. 201–211, 1996.
- [18] G. Li, C. Tang, and Z. Wei, "New conjugacy condition and related new conjugate gradient methods for unconstrained

- optimization,” *Journal of Computational and Applied Mathematics*, vol. 202, no. 2, pp. 523–539, 2007.
- [19] L. Zhang, “Two modified Dai-Yuan nonlinear conjugate gradient methods,” *Numerical Algorithms*, vol. 50, no. 1, pp. 1–16, 2009.
 - [20] L. Zhang, W. Zhou, and D. Li, “Some descent three-term conjugate gradient methods and their global convergence,” *Optimization Methods and Software*, vol. 22, no. 4, pp. 697–711, 2007.
 - [21] W. Zhou and L. Zhang, “A nonlinear conjugate gradient method based on the MBFGS secant condition,” *Optimization Methods and Software*, vol. 21, no. 5, pp. 707–714, 2006.
 - [22] J. A. Ford, Y. Narushima, and H. Yabe, “Multi-step nonlinear conjugate gradient methods for unconstrained minimization,” *Computational Optimization and Applications*, vol. 40, no. 2, pp. 191–216, 2008.
 - [23] H. Yabe and M. Takano, “Global convergence properties of nonlinear conjugate gradient methods with modified secant condition,” *Computational Optimization and Applications*, vol. 28, no. 2, pp. 203–225, 2004.
 - [24] L. Zhang, W. Zhou, and D. Li, “Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search,” *Numerische Mathematik*, vol. 104, no. 4, pp. 561–572, 2006.
 - [25] L. Zhang, W. Zhou, and D. Li, “A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence,” *IMA Journal of Numerical Analysis*, vol. 26, no. 4, pp. 629–640, 2006.
 - [26] L. Zhang and W. Zhou, “Two descent hybrid conjugate gradient methods for optimization,” *Journal of Computational and Applied Mathematics*, vol. 216, no. 1, pp. 251–264, 2008.
 - [27] W. W. Hager and H. Zhang, “A new conjugate gradient method with guaranteed descent and an efficient line search,” *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 170–192, 2005.
 - [28] G. Yuan, “Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems,” *Optimization Letters*, vol. 3, no. 1, pp. 11–21, 2009.
 - [29] G. H. Yu., *Nonlinear self-scaling conjugate gradient methods for large-scale optimization problems*, Ph.D. thesis, Sun Yat-Sen University, 2007.
 - [30] I. E. Livieris and P. Pintelas, “Performance evaluation of descent CG methods for neural network training,” in *Proceedings of the 9th Hellenic European Research on Computer Mathematics & its Applications Conference (HERCMA '09)*, E. A. Lipitakis, Ed., pp. 40–46, 2009.
 - [31] I. E. Livieris and P. Pintelas, “An improved spectral conjugate gradient neural network training algorithm,” *International Journal on Artificial Intelligence Tools*. In press.
 - [32] I. E. Livieris, D. G. Sotiropoulos, and P. Pintelas, “On descent spectral CG algorithms for training recurrent neural networks,” in *Proceedings of the 13th Panellenic Conference of Informatics*, pp. 65–69, 2009.
 - [33] E. Dolan and J. J. Moré, “Benchmarking optimization software with performance profiles,” *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.
 - [34] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Reading, Mass, USA, 1991.
 - [35] M. J. D. Powell, “Restart procedures for the conjugate gradient method,” *Mathematical Programming*, vol. 12, no. 1, pp. 241–254, 1977.
 - [36] P. M. Murphy and D. W. Aha, *UCI Repository of Machine Learning Databases*, University of California, Department of Information and Computer Science, Irvine, Calif, USA, 1994.
 - [37] E. G. Birgin and J. M. Martínez, “A spectral conjugate gradient method for unconstrained optimization,” *Applied Mathematics and Optimization*, vol. 43, no. 2, pp. 117–128, 2001.
 - [38] D. F. Shanno and K. H. Phua, “Minimization of unconstrained multivariate functions,” *ACM Transactions on Mathematical Software*, vol. 2, pp. 87–94, 1976.
 - [39] G. Yu, L. Guan, and W. Chen, “Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization,” *Optimization Methods and Software*, vol. 23, no. 2, pp. 275–293, 2008.
 - [40] D. Nguyen and B. Widrow, “Improving the learning speed of 2-layer neural network by choosing initial values of adaptive weights,” *Biological Cybernetics*, vol. 59, pp. 71–113, 1990.
 - [41] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the IEEE International Joint Conference on Artificial Intelligence*, pp. 223–228, AAAI Press and MIT Press, 1995.
 - [42] J. C. Gilbert and X. Jonsson, LIBOPT—an environment for testing solvers on heterogeneous collections of problems—version 1. CoRR, abs/cs/0703025, 2007.
 - [43] L. Prechelt, “PROBEN1—a set of benchmarks and benchmarking rules for neural network training algorithms,” Tech. Rep. 21/94, Fakultät für Informatik, University of Karlsruhe, 1994.
 - [44] J. Yu, S. Wang, and L. Xi, “Evolving artificial neural networks using an improved PSO and DPSO,” *Neurocomputing*, vol. 71, no. 4–6, pp. 1054–1060, 2008.
 - [45] R. P. Gorman and T. J. Sejnowski, “Analysis of hidden units in a layered network trained to classify sonar targets,” *Neural Networks*, vol. 1, no. 1, pp. 75–89, 1988.
 - [46] A. D. Anastasiadis, G. D. Magoulas, and M. N. Vrahatis, “New globally convergent training scheme based on the resilient propagation algorithm,” *Neurocomputing*, vol. 64, no. 1–4, pp. 253–270, 2005.
 - [47] P. Horton and K. Nakai, “Better prediction of protein cellular localization sites with the k nearest neighbors classifier,” *Intelligent Systems for Molecular Biology*, pp. 368–383, 1997.
 - [48] P. Liang, B. Labedan, and M. Riley, “Physiological genomics of escherichia coli protein families,” *Physiological Genomics*, vol. 2002, no. 9, pp. 15–26, 2002.