Conjugate Gradient Algorithm Based on Aitken's Process for Training Neural Networks

Khalil K. Abbo *Hind H. Mohammed*
kh_196538@yahoo.com *hinmath80@gmail.com*
College of Computer Science and Mathematics
University of Mosul, IRAQ

Received on: 6/11/2012 **Accepted on: 30/1/2013**

ABSTRACT

Conjugate gradient methods constitute excellent neural network training methods, because of their simplicity, numerical efficiency and their very low memory requirements. It is well-known that the procedure of training a neural network is highly consistent with unconstrained optimization theory and many attempts have been made to speed up this process. In particular, various algorithms motivated from numerical optimization theory have been applied for accelerating neural network training. In this paper, we propose a conjugate gradient neural network training algorithm by using Aitken's process which guarantees sufficient descent with Wolfe line search. Moreover, we establish that our proposed method is globally convergent for general functions under the strong Wolfe conditions. In the experimental results, we compared the behavior of our proposed method(NACG) with well- known methods in this field.

Keywords: Artificial neural front networks, Education algorithms, Aitken method, Conjugate gradient algorithms.

خوارزمية تدرج مترافق مستندة على صيغة آيتكن لتدريب الشبكات العصبية

خليل خضر عبو **هند حسام الدين محمد**

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ الاستلام : 2012/11/6 **تاريخ القبول : 2013/01/30**

الملخص

تمثل طرق التدرج المترافق طرق تدريب ممتازة للشبكات العصبية، بسبب بساطتِها، كفاءتها العددية وتطلبها لذاكرة منخفضة جداً. من المعروف أن إجراء تدريب لشبكة عصبية مرتبط إلى حدٍ كبير بنظريات الأمثلية غير المقيدة وهناك العديد من المحاولات لتسريع هذه العملية. وقد طورت خوارزميات مختلفة من نظرية الأمثلية العددية لتعجيل تدريب الشبكة العصبية. في هذا البحث، نقترح طريقة تدرج مترافق جديدة باستخدام صيغة آيتكن لتدريب شبكة عصبية والتي تضمن هبوطاً كافياً لأي بحث خطي. علاوة على ذلك، تم إثبات بأن الطريقة المقترحة متقاربة تقارباً شاملاً تحت شروطِ Wolfe القوية، أما في النَتائِج العددية فقد تم مقارنة سلوك الطريقة المقترحة(NACG) مع طرق معروفة في هذا المجال.

الكلمات المفتاحية: شبكات عصبية اصطناعية ذات التغذية الامامية، خوارزميات التعليم، طريقة آيتكن، خوارزميات التدرج المرافق.

1. Introduction

There exist many types of neural networks e. g (see [14]) , but the basic principles are similar. Each neuron in the network is able to receive input signals, to process them and to send an output signal. Each neuron is connected, at least, with one neuron, and each connection is evaluated by a real number, called weight coefficient, that reflects the degree of importance of the given connection in the neural networks.

The main advantage of neural networks is the fact, that they are able to use some a priori unknown information hidden in data (but they are not able to extract it) (see [8]). Process of capturing the unknown information is called learning of neural network or training of neural network. In mathematical formalism to learn means to adjust the with coefficients in such a way that some conditions are fulfilled.

There exist two main types of training process supervised and unsupervised training. Supervised training means that neural network known the desired output and adjusting of weight coefficients is done in such a way that the calculated and desired outputs are as close as possible [11]. This paper is concerned with supervised learning, for unsupervised learning (see [16]).

The training process of multi-layer feed forward (MLFF) neural network (AMLF neural networks consists of neurons that are ordered into layers . The first layer is called the input layer, the last layer is called output layer and the layers between are hidden layers) can be formulated as the minimization of an error function *E(w)* that depends on the connection weights *w* of the network and defined as the sum of the squared differences between the computed and required output values :

$$E(w) = \frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{M}(T_i^{(j)} - O_i^{(j)})^2 \quad \text{..}(1)$$

The variables O_{i} and T_i stand actual and desired response of i_th output neurons, respectively. The superscript denotes the particular learning pattern. The vector w is composed of all weights in the net summation of the actual errors takes place over all M output neurons and all P learning data (X,T) where the N-dimensional vector X is the input vector and the M-dimensional vector T is the target vector associated with X [14].

The most popular training algorithm for MLFF neural network is the Back-propagation(BP) algorithm introduced by Rumelhart and Williams [25] which may be proceeded in one of two basic ways: Pattern mode (online) and batch mode. In pattern mode of BP learning, weight updating is performed after the presentation of each training pattern. In the batch mode of BP learning weight updating is performed after the presentation of all the training examples (i.e. after the whole epoch) (see[8]). In this paper, we consider the batch mode training .

The remainder of this paper is organized as follows: In section 2 we introduce a brief overview of improvements back Propagation algorithm. Section 3 gives a short description of the conjugate gradient algorithms. Section 4 contains our proposed conjugate gradient training algorithm and in Section 5, we present its global convergence analysis. Finally, the experimental results are reported in Section 6.

2. Improvements on Back-Propagation Algorithm

In the standard Back-Propagation (SBP) algorithm, an initial weight (or parameter) vector w_1 of a feed-forward neural network is iteratively adapted according to the recursion:

$$\mathrm{w}_{k+1} = \mathrm{w}_{\mathrm{k}} + \gamma d_k \quad \text{...}(2)$$

To find an optimal weight vector. This adaptation is performed by presenting to the network sequentially a set pairs of input and target vectors. The positive constant of γ, which is selected by user, is called the learning rate, where $\gamma \in (0,1)$. The direction vector d_{k} is the negative of the gradient of the output error function E:

$$d_k = -g_k \quad \text{...}(3)$$

Where $g_k = \nabla E(w_k)$ several researches have investigated different modifications to speed up the convergence of the SBP algorithm. For example:

(1) Dynamical adaptation of the learning rate:

a. line search in the gradient direction i.e. at each iteration k choosing $\gamma = \gamma_k$ such that it gives the smallest nonnegative local minimum of the function $\mathrm{E}(\mathrm{w}_{\mathrm{k}} + \gamma d_k)$ [7],[15].

b. Dynamically adjusting the learning rate, either commonly for all weights [21],[26] or separately for each weight [24].

(2) Dynamical adaptation of the weight adjustments expressed by the vector d_k i.e. determining d_k for k ≥ 1 by a relation of the form :

$$\begin{aligned} d_1 &= -g_k \qquad && if\ k=1 \\ d_k &= -g_k + \beta_k\, d_{k-1} \qquad && if\ k>1 \end{aligned} \qquad \ldots\ldots(4)$$

$$w_{k+1} = w_k + \alpha_k\, d_k \qquad \ldots\ldots(5)$$

where β_k is parameter known as conjugate condition (see [18],[19],[20]).

(3) Relaxed learning rate which accelerates the back propagation algorithm in multiplicative manner of the form:

$\mathrm{w}_{k+1} = \mathrm{w}_k + \gamma\, d_k \quad if\ k=1$ and $\mathrm{w}_{k+1} = \mathrm{w}_k + \gamma\, \alpha_k\, d_k \quad if\ k>1$

where $d_k = -g_k$ and

$$\alpha_k = \frac{g_k^T\, g_k}{y_k^T\, y_k} \quad , y_k = g_k - g_{k-1} \qquad \ldots\ldots(6)$$

(see[1]) . Many other dramatically alterations of the BP algorithm. Some of which are based on combination of items listed above have been proposed, However, some of these approaches require complex and costly calculations at each iteration which offset their faster rate of convergence.

3. Conjugate Gradient Methods

Conjugate gradient (CG) methods [17] are among the most commonly and efficient used methods for large scale optimization problems due to their speed and simplicity. In general, conjugate gradient methods play an important role for efficiently training neural networks due to their simplicity and their very low memory requirements, since they don’t require the evaluation of the Hessian matrix neither the impractical storage of an approximation of it. In the literature there is a variety of conjugate gradient methods [4],[20],[18],[1] that have been intensively used for neural network training in several applications [6],[28].

The main idea for determining the search direction is the linear combination of the negative gradient vector at the current iteration with the previous search direction. The way to determine the search direction can be expressed as follows:

$$\begin{aligned} d_1 &= -g_1 \\ d_{k+1} &= -g_{k+1} + \beta_k\, d_k \end{aligned} \qquad \ldots\ldots(7)$$

Conjugate gradient methods differ in their way of defining the multiplier β_k . The most famous approaches were proposed by Fletcher–Reeves (FR), Polak–Ribere (PR) and Hestenes–Stifel (HS):

$$\beta^{FR} = \frac{g_{k+1}^T\, g_{k+1}}{g_k^T\, g_k}, \beta^{PR} = \frac{g_{k+1}^T\, y_k}{g_k^T\, g_k}, \ldots\ldots(8)\ldots\ldots \qquad \beta^{HS} = \frac{g_{k+1}^T\, y_k}{d_k^T\, y_k}$$

The conjugate gradient methods using β^{FR} update were shown to be globally convergent [3]. However the corresponding methods using β^{PR} or β^{HS} update are generally more efficient ever without satisfying the global convergence property. In the convergence analysis and implementations of CG methods, one often requires the inexact lien search such as the Wolfe line search. The standard Wolfe line search requites α_k satisfying:

$$\mathrm{E}(\mathrm{w}_k + \alpha_k d_k) \le \mathrm{E}(\mathrm{w}_k) + \rho \alpha_k g_k^T d_k \quad \ldots\ldots(9)$$

$$\mathrm{g}(\mathrm{w}_k + \alpha_K d_k)^{\mathrm{T}} d_k \ge \sigma \mathrm{g}_k^{\mathrm{T}} \mathrm{d}_k \quad \ldots\ldots(10)$$

or strong Wolfe line search:

$$\mathrm{E}(\mathrm{w}_k + \alpha_k d_k) \le \mathrm{E}(\mathrm{w}_k) + \rho \alpha_k g_k^T d_k \quad \ldots\ldots(9a)$$

$$|g_{k+1} d_k| \le -\sigma g_k d_k \quad \ldots\ldots (10a)$$

where $0 < \rho < \sigma < 1$

Moreover, an important issue of CG algorithms is that when the search direction (7) fails to be descent (by Descent, we mean $g_k^T d_k < 0 \ \forall k$) directions we restart the algorithm using the negative gradient direction to grantee convergence . A more sophisticated and popular restarting is the Powell restart.

$$|g_{k+1}^T g_k| \ge 0.2 \|g_{k+1}\|^2 \quad \ldots\ldots(11)$$

Where, $\| \ \|$ denotes to the Euclidean norm. Other important issue for the CG methods is that the search directions generated from equation (7) are conjugate if the objective function is convex and line search is exact i.e:

$$d_i^T G d_j = 0 \ , \forall \ i \ne j \quad \ldots\ldots(12)$$

Where, G is the Hessian matrix for the objective function . Dai and Lioa in [9] showed that the equation (12) can be written as follows:

$$d_{k+1}^T y_k = 0 \quad \ldots\ldots(13)$$

which is called pure conjugacy condition and generalize to the

$$d_{k+1}^T y_k = -t g_{k+1}^T s_k \ , t > 0 \quad , \quad \mathrm{s}_k = \mathrm{w}_{k+1} - \mathrm{w}_k \quad \ldots\ldots(14)$$

for general objective function with inexact line search.

4. Suggested Conjugate Gradient Algorithms

When a sequence or an iterative process is slowly converging a convergence acceleration process has to be used , Aitken's process is the most well-known convergence acceleration for linearly converging sequence, based on this observation Abbo and Mohammed [2] suggested a modification to the standard BP algorithm as follows:

Let $\{w_k\}_{k+1}^{\infty}$ be linearly convergent sequence generated by SBP algorithm, then accelerated sequence can be written as

$$\overline{w} = w_k - \frac{(w_{k+1} - w_k)^T (w_{k+1} - w_k)}{w_{k+2} - 2w_{k+1} + w_k} \quad \ldots\ldots(15)$$

From the above equation, we have :

$$\mathrm{w}_{k+1} - \mathrm{w}_k = -\gamma \ g_k \quad \ldots\ldots(16)$$

$$\mathrm{w}_{k+2} - 2\mathrm{w}_{k+1} + \mathrm{w}_k = \mathrm{w}_{k+2} - \mathrm{w}_{k+1} - (\mathrm{w}_{k+1} - \mathrm{w}_k) = -\gamma \ y_k \quad \ldots\ldots(17)$$

Let $d = \overline{w} - w_k$ ……(18)

Use equations (16), (17) and (18) in (15) and multiply the numerator and denominator d_k to get :

$d = \frac{\gamma g_k^T g_k}{y_k^T d_k} d_k$ …………..………………….………………………………(19)

4.1 New Conjugate Gradient Based on Aitken's Process (NACG Say)

Now consider the direction given equation (19) and assume that there exists a CG search direction which is parallel to the direction d_k i.e.:

$- g_{k+1} + \beta^{NA} d_k = \frac{\gamma g_k^T g_k}{y_k^T d_k} d_k$ ……...……………………………………………(20)

multiply both sides by $g_{k+1}^T y_k < 0$ to obtain:

21)…………...……………………………………………($\beta^{NA} = \frac{\gamma g_k^T g_k + g_{k+1}^T y_k}{y_k^T d_k}$

Therefore, the new search direction is :

$d_{k+1} = -g_{k+1} + \beta_k^{NA} d_k$ …………...……………………………………………(22)

Where, β_k^{NA} is defined in equation (21) .We summarize our proposed conjugate gradient neural network training algorithm (NACG) as follows:

4.1.1 The Algorithm NACG

Step 1: Initialize w_1 and choose σ, ρ such that $0 < \rho < \sigma < 1$, $\gamma \in (0,1), E_G, \varepsilon > 0$ and $\mathrm{K}_{\max}$, set $\mathrm{k} = 1$.

Step 2: Calculate the error function value E_k and its gradient g_k.

Step 3: IF $(E_k < E_G) or \|g_k\| < \varepsilon$,set $w^* = w_k$ and $E^* = E_k$, return goal is meet and stop .

Step 4: Compute the descent direction :

If $k = 1$ then, $d_k = -g_k$ go to step 6

Else $d_k = -g_k + \beta_{k-1}^{NA} d_{k-1}$.

Step 5: Compute the learning rate α_k using standard Wolfe conditions (9) and (10).

Step 6: Update the weights

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

and set $k = k + 1$.

Step 7: If $k > k_{\max}$ return Error goal not meet and stop else go to step (2).

4.2 The Descent Property of the Suggested Algorithm

In this section, we shall show our new conjugate gradient (NACG) algorithm satisfies the descent property with standard Wolfe conditions as stated in the following theorem:

4.2.1 Theorem

Consider any method of the form (4) and(5) where the learning rate α_k satisfies the standard Wolfe conditions equation (9) and (10) and the search direction computed by the equation (21) and (22), then for $k \geq 1$:

$d_k^T g_k < 0$ if $\gamma g_k^T g_k \geq g_{k+1}^T y_k$ ……...……………………………………………(23)

Proof :

The conclusion can be proved by induction for $k+1$ we have $d_1 = -g_1$, then $d_k^T\, g_k = -\|g_k\| < 0$. Now, we need to prove that (23) holds for $k+1$. From (21)and (22) we have :

$$d_{k+1} = -g_{k+1} + \frac{\gamma\, g_k^T\, g_k + g_{k+1}^T\, y_k}{d_k^T\, y_k} d_k \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(24)$$

Notice that, by Wolfe condition (10), we get that $0 < \rho < \sigma < 1$ and $y_k^T d_k > 0$ therefore:

$$y_k^T\, d_k = g_{k+1}^T\, d_k - g_k^T\, d_k \geq \sigma\, g_k^T\, d_k - g_k^T\, d_k = (\sigma - 1)\, g_k^T\, d_k$$

Hence,

$$\frac{1}{y_k^T\, d_k} \leq \frac{1}{(\sigma-1) g_k^T\, d_k} \qquad ..\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(25)$$

Then,

$$d_{k+1}^T\, g_{k+1} \leq -\, g_{k+1}^T\, g_{k+1} + \frac{\gamma\, g_{k+1}^T\, g_k + g_{k+1}^T\, y_k}{(\sigma-1) g_k^T\, d_k} d_k^T\, g_{k+1} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(26)$$

again by the second Wolfe condition with $(\sigma - 1) = -(1-\sigma)$ and $-\, g_{k+1}^T\, d_k \leq -\sigma\, g_k^T\, d_k$ then,

$$d_{k+1}^T\, g_{k+1} \leq -g_{k+1}^T\, g_{k+1} + \frac{\gamma\, g_k^T\, g_k + g_{k+1}^T\, y_k}{(1-\sigma) g_k^T\, d_k} (-\sigma) g_k^T\, d_k$$

$$\therefore d_{k+1}^T\, g_{k+1} \leq -g_{k+1}^T\, g_{k+1} - \frac{(\gamma\, g_k^T\, g_k + g_{k+1}^T\, y_k)\sigma}{(1-\sigma)} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(27)$$

Since $\sigma < 1, \gamma > 0$ then $1-\sigma > 0$ and $\gamma\, g_k^T\, g_k > 0$ we can consider the following cases:

Case (1): if $g_{k+1}^T y_k < 0$ then, by the condition given in equation (23), we have $\gamma\, g_k^T\, g_k + g_{k+1}^T\, y_k = \delta > 0$ then,

$$d_{k+1}^T\, g_{k+1} = -g_{k+1}^T\, g_{k+1} - \frac{\delta\,\sigma}{(1-\sigma)}$$

The above equation can be written as:

$d_{k+1}^T\, g_{k+1} \leq -c\|\, g_{k+1}\|^2$, where $\mathrm{c} > 0$

Therefore, sufficient descent is held .

Case (2): if $g_{k+1}^T\, y_k = 0$ then,

$$d_{k+1}^T\, g_{k+1} = -g_{k+1}^T\, g_{k+1} - \frac{(\gamma\, g_k^T\, g_k)\ \sigma \|g_{k+1}\|^2}{(1-\sigma)\ \|g_{k+1}\|^2}$$

$$= -(1 + \frac{(\gamma\, g_k^T\, g_k)\ \sigma}{(1-\sigma)\ \|g_{k+1}\|^2})\ \|g_{k+1}\|^2 < 0$$

Case (3): if $g_{k+1}^T\, y_k > 0$ then,

$$g_{k+1}^T\, y_k = g_{k+1}^T\, g_{k+1} - g_{k+1}^T\, g_k > 0$$

We get: $g_{k+1}^T\, g_{k+1} > g_{k+1}^T\, g_k$ therefore,

$$d_{k+1}^T g_{k+1} \le -g_{k+1}^T g_{k+1} - \frac{(\gamma g_k^T g_k + g_{k+1}^T g_{k+1})\sigma}{(1-\sigma)}$$

$$= -c\|g_{k+1}\|^2 < 0$$

This is complete for the proof.

5. Convergence Analysis

5.1 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, which have often been used in the literature [10],[13] to analyze the global convergence of CG methods by using inexact line searches.

We make the following basic assumptions on the objective function E.

5.1.1 Assumption

1. The objective function E is bounded below in the level set $L = \{w \in R^n : E(w) \le E(w_1)\}$ in some neighborhood N of the level set L.
2. The objective function E is continuously differentiable and its gradient g is Lipschitz's condition i.e $\exists\ \ell > 0$ such that :

   $\|g(x) - g(z)\| < \ell\|x - z\| \qquad \forall x, z \in R^n$ ………………………...………(28)
3. The level L is compact.

5.1.2 Property [12]

Consider general Conjugate gradient method, and suppose that $0 < \delta \le \|g_k\| \le \bar{\delta}$ holds. We call a method has property 5.1.2 if there exists two constants $b > 1$ and $\lambda > 0$ such that for all k, $|\beta_k| \le b$ and if $\|S_k\| \le \lambda \Rightarrow |\beta_k| \le \frac{1}{2b}$.

5.1.3 Lemma

Suppose that assumption 5.1.1 hold. if there exists a constant $\delta > 0$ such that $\|g_k\| \ge \delta$ for all $k \ge 0$ then, the following holds. if d_k satisfies the sufficient descent condition : $-g_k^T d_k \ge c\|g_k\|^2$, $c > 0$ and α_k is obtained by the Wolfe conditions (9) and (10) then, β^{NA} has property 5.1.2 .

Proof:

By the compactness of the level set L, there exists constants $M > 0$, $M_1 > 0, \bar{\delta} > 0$ such that :

$\|w_k\| \le M, \|E_k\| \le M_1, \|g_k\| \le \bar{\delta}, \forall w_k \in L$

From the sufficient descent condition and second Wolfe condition we have:

$$d_k^T y_k \ge (\sigma - 1) g_k^T d_k \ge (1-\sigma) c\, \|g_k\|^2$$

Then, by Lipschitz condition and Cauchy Schwarz inequality, we have:

$$\beta^{NA} = \frac{\gamma g_k^T g_k + g_{k+1}^T y_k}{y_k^T d_k} \le \frac{\gamma\|g_k\|^2 + \ell\, \|g_{k+1}\|\, \|S_k\|}{(1-\sigma)c\, \|g_k\|^2} \le \frac{\gamma \bar{\delta}^2 + \ell\, \bar{\delta}\, \|S_k\|}{(1-\sigma) c\, \delta^2} = b$$

Now define λ as: $\lambda = \dfrac{(1-\sigma)c\ \delta^2}{2b(\gamma\bar{\delta}^2 + \ell\ \bar{\delta}\ \|S_k\|)}$

if $\|S_k\| \le \lambda$ then :

$$\beta^{NA} \le \frac{\gamma\bar{\delta}^2 + \ell\ \bar{\delta}\ \|S_k\|\lambda}{(1-\sigma)c\ \delta^2}$$

$$\therefore \beta^{NA} \le \frac{1}{2b}\ \blacksquare.$$

5.2 Convergence analysis of the NACG method with exact line search:

In this subsection, we focus on the convergence behavior of β^{NA} method with exact the line search before proceeding we state the following lemma.

5.2.1 Lemma [28]

Suppose that the assumption 5.1.1 holds. Consider any method in the form of $\mathrm{w}_{k+1} = \mathrm{w}_k + \gamma_k d_k$ where, d_k is descent direction and γ_k satisfied the standard Wolfe conditions (9) and (10) then we have

$$\sum_{k\ge 0} \frac{(g_k^T\ d_k)^2}{\|d_k\|^2} < \infty \qquad \text{………………………………………………..………(29)}$$

5.2.2 Theorem

Suppose that the assumption 5.1.1 holds. Consider the NACG method, where γ_k is obtained by the exact line search. If $\|S_k\| \to 0, k \to \infty$ then $\lim_{k\to\infty}(\inf\|g_k\|) = 0$.

Proof :

We now prove the theorem by contradiction which is similar to the proof given in [5]. Assume that there exists some constant $\delta > 0$ such that $\|g_k\| \ge \delta$ for all $k \ge 0$. The compactness of the level set L implies that there exists a constant $\bar{\delta} > 0$ such that $\|g_k\| \le \bar{\delta}$. Since, $\|S_k\| \to 0$ then, there exists a $\bar{k}$ for all $k > \bar{k}$, $\|S_k\| < \lambda$, where λ is the same as in lemma 5.1.3 then, it follows from property that for all $k > \bar{k}$:

$$\|d_k\| \le \|g_k\| + \beta\|d_{k-1}\| \le \bar{\delta} + \frac{1}{2b}\|d_{k-1}\| \le \bar{\delta}\left(\sum_{i=0}^{k-\bar{k}-1} \frac{1}{(2b)^i}\right) + \frac{1}{(2b)^{k-\bar{k}}}\|d_{\bar{k}}\|$$

Since, the first term in the above equation is geometric series, therefore,

$$\|d_k\| \le \bar{\delta}\left(\frac{2b}{2b-1}\right) + \|d_{\bar{k}}\| = \bar{\lambda} \qquad \text{…………………………………………………(30)}$$

Furthermore,

$$\sum_{u=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \ge \sum \frac{\|g_k\|^4}{\|d_k\|^2} \ge \sum \frac{\delta^4}{\|d_k\|^2} \ge \infty \qquad \text{…………………………………..……(31)}$$

Zoutendijk condition together with (31) yield $\sum \dfrac{\delta^4}{\|d_k\|^2} < \infty$ which contradictions (30) hence, $\lim_{k\to\infty} (\inf\|g_k\|) = 0\ \blacksquare$.

5.3 Convergence Analysis of the NACG Method with Inexact Line Search

Finally, we discuss the global convergence of the NACG method with inexact line search, to establish the convergence of the method, we need the following lemma

5.3.1 Lemma [5]

Suppose that assumption 5.1.1 holds. Consider any conjugate gradient method, where d_k is a descent direction and α_k is obtained by the strong Wolfe line search conditions (9a) and (10a). if $\sum_{k\geq 0} \frac{1}{\|d_k\|} < \infty$ then, $\lim(\inf\|g_k\|) = 0$.

Now, we ready to show the convergence of NACG method with strong Wolfe conditions according to the following theorem.

5.3.2 Theorem

Suppose that assumption 5.1.1 holds and d_k is a descent direction or sufficient descent i.e.

$$g_k^T d_k \leq -c_1 \|g_k\| \quad ,c_1 > 0 \qquad \text{……………………………………………...………(31)}$$

Consider the NACG method, where α_k is obtained by the strong Wolfe line search (9a) and (10a). if there exists a constant $\bar{\delta} > 0$ such that $\delta \leq \|g_k\| \leq \bar{\delta}$ for all $k \geq 0$, then $\|d_k\| \neq 0$ and $\sum \|u_{k+1} + u_k\|^2 < \infty$, where $u_k = \frac{d_k}{\|d_k\|}$.

Proof:

This proof is similar to lemma (9) in [5].

Since, $\|g_k\|$ is bounded a way form zero then $d_k \neq 0, \forall k$, by definition of β^{NA},

we have: $\beta^{NA} = \frac{\gamma g_{k-1}^T g_{k-1}}{d_{k-1}^T y_{k-1}} + \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} = M_k^1 + M_k^2$ Where,

$$M_k^1 = \frac{\gamma g_{k-1}^T g_{k-1}}{d_{k-1}^T y_{k-1}} > 0 \quad , \quad M_k^2 = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}$$

Define: $r_k = \frac{-g_k + M_k^2 d_{k-1}}{\|d_k\|} \quad , \quad t_k = \frac{M_k^1 \|d_{k-1}\|}{\|d_k\|}$

Then, by $d_k = -g_k + \beta_k^{NA} d_{k-1}$ and noting $\beta^{NA} = M_k^1 + M_k^2$, we have:

$$u_k = r_k + t_k u_{k-1} \quad ,k \geq 1 \qquad \text{……………………………………………...……(32)}$$

Then,

$$\|u_k - u_{k-1}\| \leq \|r_k - r_{k-1}\| + \|t_k u_{k-1} - t_{k-1} u_{k-2}\| \quad ,k \geq 1$$

By definition u_{k-1} , u_{k-2} in (32) :

$$\therefore \|u_k - u_{k-1}\| \leq 2\|r_k - r_{k-1}\| \qquad \text{……………………………………..……(33)}$$

Also, from second Wolfe condition :

$$d_{k-1}^T y_{k-1} \geq (\sigma - 1) g_{k-1}^T d_{k-1}$$

$$\therefore \|d_{k-1}^T y_{k-1}\| \geq (\sigma - 1) \ \|g_{k-1}\| \, \|d_{k-1}\|$$

Then,

$$\|r_k\| \le \frac{1}{\|d_k\|}(\|g_k\| + M_k^2 \|d_{k-1}\|) \le \frac{1}{\|d_k\|}(\bar{\delta} + \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}\|d_{k-1}\| + \frac{\gamma\, g_{k-1}^T g_{k-1}}{d_{k-1}^T y_{k-1}}\|d_{k-1}\|)$$

Then, by bounded of $\|g_k\|$,$\forall k$,we have

$\|r_k\| \le \frac{1}{\|d_k\|}(\bar{\delta} + \frac{\bar{\delta}\|y_{k-1}\|}{(\sigma-1)\ \|g_{k-1}\|} + \frac{\gamma\bar{\delta}}{(\sigma-1)}) = \frac{1}{\|d_k\|}(\bar{\delta} + \frac{\bar{\delta}^2}{(\sigma-1)\ \delta} + \frac{\gamma\bar{\delta}}{(\sigma-1)}) = \frac{1}{\|d_k\|}\bar{M} < \infty$ it follows from (33) and $\sum \frac{1}{\|d_k\|} < \infty$ that

$$\sum_{k\ge 0}\|u_k - u_{k-1}\|^2 \le 4\sum_{k\ge 0}\|r_{k+1}\|^2 < 4\sum_{k}\frac{\bar{M}}{\|d_{k+1}\|^2} < \infty$$

Then by lemma (5.3.1) $\lim_{k\to\infty}(\inf\|g_k\|) = 0$ ■.

6. Experimental Results

In this following section, we compare the performance of conjugate gradient methods with Fletcher_Reeves update (CGFR), Polack_Ribiere update (CGPR) and our proposed conjugate gradient algorithm (CGNA) in two famous classification problem: the SPECT heart problem and Monk's problem.

The simulations have been carried out using MATIAB(7.6) the performance of the AIBP has been evaluated and compared with batch versions of the conjugate gradient method with Fletcher_ Reeves update (CGFR) is known as (traincgf) and the conjugate gradient method with Polack_Ribiere update which is known as (traincgp) see appendix. Toolbox default values for the heuristic parameters of the above algorithms are used unless stated otherwise. The algorithms were tested by using the initial weights, initialized by the Nguyen –windrow method [22] and received the same sequence of input patterns . The weights of the network are updated only after the entire set of patterns to be learned has been presented .

For each of the test problems, the network architectures consists of one hidden layer with 3 neurons and an output layer of one neuron. The termination criterion is set to E ≤ 0.1 within the limit of 1000 epochs Table(1) summarizes the results of all algorithms i.e. for 50 simulations, a table summarizing the performance of the algorithms for simulations that reached solution is presented. The reported parameters are min the minimum number of epochs, mean the mean value of epochs, Max the maximum number of epochs, Tav the average of total time and Suc, the succeeded simulations out of (50) trails within error function evaluations limit.

6.1 SPECT Heart Classification Problem

SPECT is a good data set for testing ML algorithms; it has 267 instances that are described by 22 binary attributes. This dataset contains data instances derived from cardiac Single Proton Emission Computed Tomography (SPECT) images from the University of Colorado. This is also a binary classification task, where patients heart images are classified as normal or abnormal. The class distribution has 55 instances of the abnormal class (20.6%) and 212 instances of the normal class (79.4%). From them, there have been selected 80 instances for the training process and the remainder 187 for testing the neural networks generalization capability [19].The network architectures for this medical classification problem consists of one hidden layer with 3 neurons and an output layer of one neuron. Table(1) summarizes the results of all algorithms.

Table(1): Results of Simulations for the Heart Problem

Algorithms	Min	Max	Mean	Tav	Succ
FRCG	26	90	48.5	0.44154	50
PRCG	23	65	42.72	0.4588	50
NACG	17	89	41.14	0.38904	50

From table (1), we noted that the algorithm NACG is the beast algorithm with respect to the mean number of the epochs and the time.

6.2 Monk's Problems

The MONK's problems were the basis of a first international comparison of learning algorithms. The result of this comparison is summarized in "The MONK's Problems - A Performance Comparison of Different Learning algorithms". This data is a collection of three binary classification problems relying on the artificial robot domain, in which robots are described by six binary different attributes. These benchmarks are made of a numeric base of examples and of a set of symbolic rules. There are three MONK's problems. The domains for all MONK's problems are the same(432 patterns). For each problem the domain has been partitioned into a train and test set[17]:

1. MONK-1 consists of 124 patterns which were selected randomly from the data set for training, while the remaining 308 were used for the generalization testing (table(2)).

Table(2): Results of Simulations for the MONK-1 Problem

Algorithms	Min	Max	Mean	Tav	Succ
FRCG	22	87	45.6	0.65816	50
PRCG	17	79	36.64	0.59778	50
NACG	15	62	33.14	0.50146	50

From table (2), we conclude that the algorithm NACG is the beast algorithm with respect to the mean number of the epochs and the time.

2. MONK-2 comprises by 169 randomly selected examples from the data set for training, while the rest 263 were used for testing (table(3)).

Table(3): Results of Simulations for the MONK-2 Problem

Algorithms	Min	Max	Mean	Tav	Succ
FRCG	29	314	124.18	1.4569	50
PRCG	28	159	64.04	0.90308	50
NACG	19	163	61.08	0.81032	50

From table (3), we noted that the algorithm NACG is the best algorithm with respect to the mean number of the epochs and the time.

3. MONK-3 problem consists of 122 patterns for training and the remaining 310 patterns were used for testing (table(4)) .

Table(4): Results of Simulations for the MONK-3 Problem

Algorithms	Min	Max	Mean	Tav	Succ
FRCG	5	23	12.54	0.12684	50
PRCG	5	26	11.4	0.12728	50
NACG	4	18	10.08	0.09844	50

From table (4), we noted that the algorithm NACG is the best algorithm with respect to the mean number of the epochs and the time.

Appendix

1. traincgf: is matlab function (in the matlab toolbox), which is a network training function that updates weight and bias values according to the conjugate gradient back propagation with Fletcher-Reeves updates.

2. traincgp: is matlab function (in the matlab toolbox) utilize the conjugate gradient back propagation with Polak-Ribiere updates to minimize error function E (training the network).

REFERENCES

[1] Abbo. K (2010)." Developing of Gradient Algorithms for Solving Unconstrained Non-linear Problems with Artificial Neural Networks". Ph.D. thesis, University of Mosul.

[2] Abbo K. and Mohammed .H (2012). " Improving the learning rate of the back propagation by Aitkin process". Appear soon.

[3] AL-Baali M .(1999). "Descent property and global convergence of Fletcher and Reeves method with inexact line search". IMA J. of Numerical Analysis.

[4] Birgin E. and Martinez J.(1999). "A spectral conjugate gradient method for unconstrained optimization" Applied Mathematics and Optimization, 43:117-128.

[5] Caiying W. and Guoqing C. (2010). "New type of conjugate gradient algorithms for unconstrained optimization problems". J. of Systems Engineering and Electronics. Vol. 21, No. 6.

[6] Charalambous C. (1992). " Conjugate gradient algorithm for efficient training of artificial neural networks". IEEE Proceedings, 139(3):301-310.

[7] Dahl D. (1987). "Accelerated learning using the generalized delta rule', proceedings of the IEEE International conference on neural networks, 2.

[8] Daniel S., Vladimir K. and pospichal J. (1997). "Introduction to multi-layer feed-forward neural networks", Chemo metrics and Intelligent laboratory system 39.

[9] Dai Y and Liao Z (2001). "new conjugacy conditions and related nonlinear conjugate gradient methods", Applied Mathematics and Optimization 43.

[10] Dia Y. and Yuan Y. (2000). "non-linear conjugate gradient methods", Shanghai Scientific and Technical publishers.

[11] Enrique C., Bertha G., Oscar F. and Amparo (2006). "A very fast learning method for neural network Based on sensitivity analysis", J. of Machine learning Research7.

[12] Gilbert G. and Nocedal J. (1992). "Global convergence property of conjugate gradient methods for optimization. SIAN Journal on Optimization

[13] Hager W. and Zhang H. (2006). "A survey of non-linear conjugate gradient methods", Pacific of J. Optimization 2.

[14] Haykin S. (1994). "Neural Networks –A comprehensive foundation", Macmillan .

[15] Hush D. and Salas M. (1988). "Improving the learning rate of back propagation with the gradient reuse algorithm". proceedings of the IEEE International conference on networks, 1.

[16] Kohonen T.(1988). "Self-organization and associative Memory",

Springer Verlag, Berlin.

[17] Livieris I. and Pintelas P. (2008). "'A survey of algorithms for training artificial neural networks", Technical Report NO. TR08-01.

[18] Livieris I. and Pintelas R. (2011). "An advanced conjugate gradient training algorithm based on a modified secant equation", Technical Report NO. TR11-03.

[19] Livieris I . and Pintelas P. (2012). "An Improved spectral conjugate gradient neural network training algorithm", Technical Report NO. TR10-02.

[20] Moller M. (1993). "A scaled conjugate gradient algorithm for fast supervised learning", Neural Networks,6.

[21] Nachtsheim P. (1994). "A first order adaptive learning rate algorithm for back propagation networks", Proceedings of the IEEE International conference on neural networks, 1.

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

[23] Peng C. and Magoulas G.(2008)" Adaptive nonmonotone conjugate gradient training algorithm for recurrent neural networks". In 19th IEEE International Conference on Tools with Artificial Intelligence, pages 374-381.

[24] Pirez M. and Sarkar D. (1993). "Back-propagation algorithm with controlled oscillation of weights", Proceeding of IEEE International conference on neural networks,1.

[25] Rumelhart D. and Mc.Clelland J. (1986) "puralled distributed processing: explorations in the microstructure of cognition", Vol.1: Foundations, MIT press.

[26] Salomon R. and Van Hemmen J. (1996). "Accelerating back propagation through dynamic self-adaptation ". Neural network 9(4) .

[27] Sotiropoulos D. , Kostopoulos A., and Grapsa T. (2002)." A spectral version of Perry's conjugate gradient method for neural network training". In Proceedings of the 4th GRACM Congress on Computational Mechanics, University of Patras.

[28] Zoutendijk G. (1970). "Non_linear programming, computional methods". Abadie J. ed. Integer and Non_linear Programming, Amsterdam.