SCALED CONJUGATE GRADIENT TYPE METHOD WITH ITS CONVERGENCE FOR BACK PROPAGATION NEURAL NETWORK

Ban A. Mitras¹, Zina Abd Al-salam ²

College of computer sciences and Mathematics, Mosul University, Mosul, Iraq

Email: Mosul University, Iraq, r_jammas@yahoo.com

Abstract
Conjugate gradient algorithms were widely used in optimization, especially for large scale optimization problems, because it was not required the storage for any matrices. The goal of the training is search an optimal set of a connection weights in the manner that the error of network output can be minimized. In this paper, we derived a proposed formula of the conjugacy coefficient based on conjugacy condition, then we proved the sufficient descent and global convergence properties for this formula. Comparative results for our proposed algorithm and standard back-propagation (BP) are presented for two test problems and the results was encourage.

Key words: conjugate gradient methods, feed forward neural network, Back propagation (BP) network, pure conjugacy condition.

1. INTRODUCTION:

The feed forward neural networks with back propagation (BP) training procedure have been used in various fields of scientific researches and engineering applications. The BP algorithm attempts to minimized the least square error of objective function, defined by the differences between the actual network outputs and desired outputs [16]. The back propagation training algorithm is a supervised learning method for multi layer feed forward neural network[16]. It is essentially a gradient descend local optimization technique which involves backward error correction of the network weights. Despite the general success of Back propagation in learning the neural networks, several neural networks, several major deficiencies are still needed to be solved. First, the back propagation algorithm will get trapped in local minima specially for non linear separable problems[12] such as the XOR problems [6].

Having trapped into local minima, back propagation may lead to failure in finding a global optimal solution. Second, the convergence rate of back propagation is still too slow even if learning can be achieved. The back propagation algorithm looks for the minimum of the error function in weight space using the method of gradient descend. the combination of weight which minimizes the error function is considered to be a solution of the learning problem. Since this method
requires combination of the gradient of the error function at each iteration step.

The patch training of FFNN is consistent with theory of unconstrained optimization and can be viewed as the minimization of the error function E defined by

\[ E = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{J} (t_j^p - o_j^p)^2 \]

…………..(1)

Where \((t_j^p - o_j^p)\) is the squared difference error between the actual output value at the j-th output layer neuron for pattern P and the target output value. A traditional way to solve this problem is by an iterative gradient-based training algorithm which generates a sequence of weights \(\{W\}\) starting from an initial point \(W_0 \in \mathbb{R}^n\) use the recurrence [11]

\[ W_{k+1} = W_k + \alpha_k d_k \]

………………(2)

Where \(K\) is the current iteration usually called epoch, \(\alpha\) is the learning rate and \(d_k = -\nabla E (K)\) is a descent search direction i.e. \(g_k^T d_k < 0\) since the appearance of back propagation[16].

2. CONJUGATE GRADIENT METHOD:

Conjugate Gradient (CG) methods are wildly used for unconstrained optimization especially when the dimension is large.

We are concerned with the following unconstrained minimization problem:

Minimize \(f(x)\)

(3)

Where \(f: \mathbb{R}^n \rightarrow \mathbb{R}\) is smooth and its gradient \(g(x) = \nabla f(x)\) is available. There are several kinds of numerical methods for solving eq.(3), which include the Steepest Descent (SD) method, the Newton method CG-method and Quasi-Newton (QN) methods. The CG methods are our choice for solving the large-scale problems, because they do not need any matrix storage. CG-methods, however, are iterative methods of the form: \(x_{k+1} = x_k + \lambda_k d_k\)

(4)

where \(\lambda_k > 0\) is a step size and \(d_k\) is a search direction. Search directions are usually defined by:

\[ d_{k+1} = \begin{cases} -g_{k+1} & \text{for } k = 1 \\ -g_{k+1} + \beta_k d_k & \text{for } k \geq 2 \end{cases} \]

(5)

where \(g_k\) denotes \(\nabla f(x_k)\) and \(\beta_k\) is a scalar.

We can deduce a formula for the scalar \(\beta_k\):

\[ \beta_k^{HS} = \frac{g_k^T \gamma_k}{d_k^T \gamma_k} \]

(6)

\[ \beta_k^{FR} = \frac{||g_k + l||^2}{||g_k||^2} \]

(7)
This definition of $\beta_k$; in Eq. (6) due to [8]; $\beta_k$ in Eq. (7) due to [7]; $\beta_k$ in Eq. (8) due to [14]; $\beta_k$ in Eq. (9) due to [5]; $\beta_k$ in Eq. (10) due to [4]; $\beta_k$ in Eq. (11) due to [10]; $\beta_k$ in Eqs. (12), (13), (14) due to [1]; $\beta_k$ in Eq. (15) due to [13].

To establish the convergence results of nonlinear CG-methods mentioned above, it is usually required that the step $\lambda_k$ defined in eq.(4) should satisfy the following strong Wolfe conditions:

$$f(x_k + \lambda_k d_k) - f(x_k) \leq \delta \lambda_k g_k^T d_k$$

$$|g(x_k + \lambda_k d_k)^T d_k| \leq -\sigma g_k^T d_k$$

3.PROPOSED CONJUGACY COEFFICIENT:

Dai and Liao in 2001 [3] Proposed a new formula that extended of Hestenes and Steifel method as:

$$\beta_k^{DL} = \frac{\frac{g_k^T y_k}{d_k^T y_k}}{\frac{g_k^T y_k}{d_k^T y_k}}$$

Where, $t$ is a positive scalar.

In this paper we proposed a new formula that modified Al-Bayati and Al-
Assady(1) conjugate gradient method as:

\[ \beta^{AB} = \frac{\|\gamma_k\|^2}{\|g_k\|^2} \]

\[ \beta^{AB*} = \frac{\|\gamma_k\|^2 + t \frac{(g_{k+1}^T s_k) \|\gamma_k\|^2}{\|g_k\|^2}}{\|g_k\|^2} \]

\[ = \beta^{AB} + t \frac{(g_{k+1}^T s_k) \|\gamma_k\|^2}{\|g_k\|^2} \]

......(19)

Where \( s_k = \alpha_k d_k \) and \( t \geq 0 \) are constant, for an exact line search, \( g_{k+1} \) is orthogonal to \( s_k \), hence \( \beta^{AB*} \) is reduced to \( AB \) method. But if the line search is inexact then we can compute \( t \) by multiplying equation (5) with \( y_k \), we obtain the following formula for computing \( t \):

\[ d_{k+1} = -g_{k+1} + \beta_k d_k \]

\[ y_k^T d_{k+1} = -y_k^T g_{k+1} + \beta_k y_k^T d_k \]

...(20)

Now, if the direction is inexact(ILS) then

\[ d_{k+1} = -t g_{k+1}^T s_k \]

\[ -t g_{k+1}^T s_k = -y_k^T g_{k+1} + \left( \frac{\|y_k\|^2}{\|g_k\|^2} \right) y_k^T d_k \]

\[ + t \frac{g_{k+1}^T s_k}{\|g_k\|^2} \left( \frac{\|y_k\|^2}{\|g_k\|^2} \right) y_k^T d_k \]

\[ = -y_k^T g_{k+1} + \left( \frac{\|y_k\|^2}{\|g_k\|^2} \right) y_k^T d_k \]

\[ - t g_{k+1}^T s_k = -y_k^T g_{k+1} \left( \frac{\|y_k\|^2}{\|g_k\|^2} \right) y_k^T d_k \]

\[ - t g_{k+1}^T s_k \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 \]

\[ = -y_k^T g_{k+1} \left( \frac{\|y_k\|^2}{\|g_k\|^2} \right)^2 \]

\[ + t g_{k+1}^T s_k \left( \frac{\|y_k\|^2}{\|g_k\|^2} \right)^2 \]

\[ + t g_{k+1}^T s_k \left( \frac{\|y_k\|^2}{\|g_k\|^2} \right)^2 \]

\[ t(g_{k+1} s_k \|y_k\|^2 d_k y_k + g_{k+1} s_k \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2) \]

\[ = y_k^T g_{k+1} \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 \]

\[ - \|y_k\|^2 \|g_k\|^2 d_k y_k \]

\[ t = \frac{y_k^T g_{k+1} \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 - \|y_k\|^2 \|g_k\|^2 d_k y_k}{y_k^T g_{k+1} \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 y_k + g_{k+1} s_k \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2} \]

...(21)

now substitute the value of \( t \) in (21) in equation (19) we get:

\[ \beta^* = \frac{\|y_k\|^2}{\|g_k\|^2} \]

\[ = \frac{y_k^T g_{k+1} \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 - \|y_k\|^2 \|g_k\|^2 d_k y_k}{y_k^T g_{k+1} \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 y_k + g_{k+1} s_k \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2} \]

\[ = \frac{y_k^T g_{k+1} \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 - \|y_k\|^2 \|g_k\|^2 d_k y_k}{y_k^T g_{k+1} \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2 y_k + g_{k+1} s_k \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2} \]

\[ \beta^{new} = \frac{\|y_k\|^2}{\|g_k\|^2} + \frac{y_k^T g_{k+1} \|g_k\|^2}{\|y_k\|^2 \|g_k\|^2 d_k y_k + \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2} \]

\[ = \frac{(\|y_k\|^2) d_k y_k + \|g_k\|^2}{\|y_k\|^2 \|g_k\|^2 d_k y_k + \left( \frac{\|g_k\|^2}{\|g_k\|^2} \right)^2} \]

...(22)
and we use the last $\beta_k^{new}$ in equation (22) to prove the convergence analysis of our algorithms.

4. CONVERGENCE ANALYSIS:

In order to establish the global convergence analysis, we make the following assumptions for the objective function $f$.

**ASSUMPTION (1)**

i. The level set $\xi = \{x \mid f(x) \leq f(x_0)\}$ is bounded, namely, there exists a constant $B > 0$ such that $\|x\| \leq B$ for all $x \in \xi$

ii. In some neighborhood $N$ of $\xi$, $f$ is continuously differentiable, and its gradient is globally Lipschitz continuous, namely, there exists a constant $L > 0$ such that $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$ for all $x, y \in N$ [18].

**THEOREM (1)**

Suppose that $d_{k+1}$ is given by (5) and $\beta_k^*$ which is defined in (22) then, the following result is satisfies:

$$\mathcal{a}_k^T \mathcal{a}_k + 1 \leq c \|\mathcal{a}_k + 1\|^2$$

Proof:

By induction for $k=1$ we have $d_1 = -g_1$ then $d_1^T g_1 < 0$, then we assume that $d_k^T g_k < 0 \quad \forall k \geq 2$

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \frac{\|y_k\|^2}{\|g_k\|^2} - \frac{\|\nabla f(x_k)^T g_k\|^2}{\|g_k\|^4 a_k^T g_k} g_{k+1}^T d_k$$

It follows from strong Wolfe condition (16) and (17) that:

$$g_{k+1}^T d_{k+1} \leq -\|g_{k+1}\|^2 + \frac{\|y_k\|^2}{\|g_k\|^2} - \frac{\|\nabla f(x_k)^T g_k\|^2}{\|g_k\|^4 a_k^T g_k} (\sigma g_k^T d_k)$$

$$g_{k+1}^T d_{k+1} + \|g_{k+1}\|^2 \leq \left(\frac{\|y_k\|^2}{\|g_k\|^2} - \frac{\|\nabla f(x_k)^T g_k\|^2}{\|g_k\|^4 a_k^T g_k} \right) (\sigma g_k^T d_k)$$

dividing both side by $\|g_{k+1}\|^2$ and invert the inequality:

$$\frac{\|g_{k+1}\|^2}{g_{k+1}^T d_{k+1} + \|g_{k+1}\|^2} \geq \left(\frac{\|y_k\|^2}{\|g_k\|^2} - \frac{\|\nabla f(x_k)^T g_k\|^2}{\|g_k\|^4 a_k^T g_k} \right) (\sigma g_k^T d_k)$$
Now also it follows from strong Wolfe condition (16) and (17) that 

\[ g_k^T d_k \leq -y_k^T d_k \Rightarrow -g_k^T d_k \geq \frac{y_k^T d_k}{(\sigma + 1)} \]

\[ \frac{\|g_{k+1}\|^2}{g_{k+1}^T d_{k+1} + \|g_{k+1}\|^2} \geq (\frac{\|y_k\|^2}{\|y_k\|^2 y_k^T g_{k+1}} + \frac{\|g_k\|^4}{\|y_k\|^2 y_k^T g_{k+1}}) \|g_{k+1}\|^2(\sigma + 1) \]

Let 

\[ \frac{\|g_k\|^2}{\|y_k\|^2 y_k^T g_{k+1}} + \frac{\|g_k\|^4}{\|y_k\|^2 y_k^T g_{k+1}} \frac{\|g_{k+1}\|^2(\sigma + 1)}{(y_k^T d_k)} \]

\[ = \delta > 0 \]

\[ \Rightarrow \]

\[ = \delta > 0 \]

5. GLOBAL CONVERGENCE THEOREM:

Under Assumption ii, we give a useful lemma which was essentially proved [27]

**LEMMA (1)**: Suppose that \( x_1 \) is a starting point for which Assumption (1) is satisfied. Consider any method of the form (2), where \( d_k \) is a descent direction and \( \alpha_k \) satisfies Wolfe conditions (7) and (8) then we have:

\[ \sum_{k=1}^{\infty} \frac{1}{\|d_k\|^2} = \infty \]

**THEOREM (3)**: Suppose that \( x_1 \) is a starting point for which Assumption (1) holds. Let \( \{ x_k, k = 1, 2, \ldots \} \) be generated by our method. Then the algorithm either terminates at a stationary point or converges in the sense that 

\[ \lim \inf_{k \to \infty} \| g_k \| = 0 \]

\[ \frac{\|g_{k+1}\|^2}{g_{k+1}^T d_{k+1} + \|g_{k+1}\|^2} \geq \delta \]

\[ \Rightarrow \]

\[ g_{k+1}^T d_{k+1} + \|g_{k+1}\|^2 \leq \frac{1}{\delta} \]

\[ \Rightarrow g_{k+1}^T d_{k+1} \leq \frac{1}{\delta} \|g_{k+1}\|^2 - \|g_{k+1}\|^2 \]

and if we assume \( 1 - \frac{1}{\delta} = c \) and \( \frac{1}{\delta} \in (0,1) \)

then we complete the proof

\[ g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2 \]

**PROOF**: 

Suppose that the conclusion does not hold, that is to say their exist appositive constant \( \varepsilon \) such that \( \|g_k\| \geq \varepsilon \) for all \( k \). Since

\[ d_k + 1 = -g_k + 1 + \beta_k d_k \]

which is can be written as

\[ \|d_k + 1\| \leq \|g_k + 1\| + \|\beta_k \| \|d_k\| \]

and since:

\[ |\beta|^n \| = \|y_k \| \| \leq \|g_{k+1} \| \|y_k \| \| \]

\[ \|g_{k+1} \|^2 - \|g_{k+1} \|^2 \]

\[ \Rightarrow \]

27
\[
|\beta^{\text{new}}| \leq \frac{\|y_k\|^2}{\|g_k\|^2} \left( \frac{y_k g_{k+1} \|y_k\|}{\|y_k\|^2} + \frac{y_k g_{k+1} \|y_k\|}{\|g_k\|^2} - \frac{(\|y_k\|)^2 - \|y_k\| g_k d_k (\sigma - 1) + \|g_k\|^2}{\|g_k\|^2\|y_k\|^2} \right) - \frac{(\|y_k\|)^2 - \|y_k\| g_k d_k (\sigma - 1) + \|g_k\|^2}{\|g_k\|^2\|y_k\|^2}
\]

then

\[
|\beta^{\text{new}}| \leq \frac{y^2}{\xi} + \frac{y \psi y^2}{\xi \psi (\sigma - 1) + \psi^4} - \frac{y^4 \psi (\sigma - 1)}{\xi^6 - \xi^2 \psi^2 \psi (\sigma - 1)}
\]

such that \( b \) is a constant

\[
|\beta_k^{\text{new}}| \leq \left| \frac{y^2}{\xi} + \frac{y \psi y^2}{\xi \psi (\sigma - 1) + \psi^4} - \frac{y^4 \psi (\sigma - 1)}{\xi^6 - \xi^2 \psi^2 \psi (\sigma - 1)} \right| = b
\]

and with this contradiction we complete the prove that is

\[
\sum_{i=1}^{\infty} \frac{1}{\|d_i\|^2} \geq \frac{1}{(\gamma + b \eta)^2} \sum_{k=1}^{\infty} 1 = \infty
\]

6. NUMERICAL EXPERIMENTS:

Now we present a numerical experiments whose objective function is compared with AB1 algorithms on the same set of unconstrained optimization test problem. For each test function (Andrei, 2008). All algorithms implemented with the same line search and with the same parameters. The comparison is based on number of iteration (NOI), and number of function evaluation (NOF). Our algorithms has converged as soon as \( \|g_k\|^2 \leq 10^{-6} \).

Table (1) shows the Comparison of algorithms with respect to NOI and NOF for \( n=1000, n=10000, n=100000 \) respectively.

<table>
<thead>
<tr>
<th>Test Functions</th>
<th>standard AB method</th>
<th>modified AB method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOI</td>
<td>NOF</td>
</tr>
<tr>
<td>Cubic</td>
<td>30</td>
<td>77</td>
</tr>
<tr>
<td>Sallow Function</td>
<td>32</td>
<td>75</td>
</tr>
</tbody>
</table>
Table (2) Comparison between standard AB method and modified AB method with respect to (NOI and NOF) for n=10000

<table>
<thead>
<tr>
<th>Test Functions</th>
<th>Standard AB method</th>
<th>Modified AB method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOI</td>
<td>NOF</td>
</tr>
<tr>
<td>Cubic</td>
<td>33</td>
<td>84</td>
</tr>
<tr>
<td>Shallow Function</td>
<td>38</td>
<td>87</td>
</tr>
<tr>
<td>Rosen</td>
<td>94</td>
<td>208</td>
</tr>
<tr>
<td>Beale</td>
<td>199</td>
<td>401</td>
</tr>
<tr>
<td>Nondiagonal</td>
<td>333</td>
<td>1158</td>
</tr>
<tr>
<td>Sum</td>
<td>119</td>
<td>521</td>
</tr>
<tr>
<td>Strait</td>
<td>41</td>
<td>88</td>
</tr>
<tr>
<td>Reciep</td>
<td>65</td>
<td>151</td>
</tr>
<tr>
<td>Wood Function</td>
<td>656</td>
<td>1318</td>
</tr>
</tbody>
</table>
Table (3) Comparison between standard AB method and modified AB method with respect to (NOI and NOF) for n=100000

<table>
<thead>
<tr>
<th>Test Functions</th>
<th>Standard AB method</th>
<th>Modified AB method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOI</td>
<td>NOF</td>
</tr>
<tr>
<td>Cubic</td>
<td>33</td>
<td>84</td>
</tr>
<tr>
<td>Sallow Function</td>
<td>44</td>
<td>99</td>
</tr>
<tr>
<td>Rosen</td>
<td>124</td>
<td>268</td>
</tr>
<tr>
<td>Beale</td>
<td>217</td>
<td>437</td>
</tr>
<tr>
<td>Nondiagonal</td>
<td>147</td>
<td>307</td>
</tr>
<tr>
<td>Sum</td>
<td>204</td>
<td>818</td>
</tr>
<tr>
<td>Strait</td>
<td>43</td>
<td>92</td>
</tr>
<tr>
<td>Reciep</td>
<td>71</td>
<td>165</td>
</tr>
<tr>
<td>Wood Function</td>
<td>704</td>
<td>1414</td>
</tr>
<tr>
<td>Panalty 2Function</td>
<td>65</td>
<td>147</td>
</tr>
<tr>
<td>Total</td>
<td>1652</td>
<td>3831</td>
</tr>
</tbody>
</table>

7. CONCLUSION:

The architecture of the FFNN is 1-15-1 with sigmoid function in hidden layer and linear function in output layer used to approximate y=sin(x)*cos(3x) in the interval [-pi,pi].

For the test problems, a table summarizing the performance of the
algorithms for simulations that reached solution is presented. Where the standard parameters are the goal of error (GE), the minimum number of epochs (MIN/EP), the maximum number of epochs (MAX/EP), the average value of epochs (AV/EP), the average of total time (AV/TM) back-propagation (SBP) and the Proposed Algorithm (P-CG-BPNN). The reported and successful performance (SUCC/PERF). The succeeded simulations out of (100) trials within the error function evaluations limit.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SBP</td>
<td>0.001</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0%</td>
</tr>
<tr>
<td>Standard AB method</td>
<td>0.01</td>
<td>0.0464</td>
<td>614</td>
<td>75</td>
<td>344.5</td>
<td>100%</td>
</tr>
<tr>
<td>Modified AB method</td>
<td>0.001</td>
<td>0.0249</td>
<td>355</td>
<td>60</td>
<td>207.5</td>
<td>100%</td>
</tr>
</tbody>
</table>

8. REFERENCES:


