International Journal of Integrated Engineering:
Special Issue 2018: Data Information Engineering, Vol. 10 No. 6 (2018) p. 11-18.
 DOI: https://doi.org/10.30880/ijie.2018.10.06.002

An Improved Back Propagation Leaning Algorithm Using Second Order Methods with Gain Parameter

Nazri Mohd Nawi[1*], Noor Haliza Mohamed Saufi[2], Avon Budiyono[3], Norhamreeza Abdul Hamid[4], Muhammad Zubair Rehman[5], Azizul Azhar Ramli[6]

[1,2,4,6]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400, Johor, Malaysia

[3]School of Industrial Engineering, Telkom University, 40257 Bandung, West Java, Indonesia
[5]Department of Computer Science and Information Technology, University of Lahore, Islamabad Campus, Pakistan

Received 28 June 2018; accepted 5 August 2018, available online 24 August 2018

Abstract: Back Propagation (BP) algorithm is one of the oldest learning techniques used by Artificial Neural Networks (ANN). It has successfully been implemented in various practical problems. However, the algorithm still faces some drawbacks such as getting easily stuck at local minima and needs longer time to converge on an acceptable solution. Recently, the introduction of Second Order Methods has shown a significant improvement on the learning in BP but it still has some drawbacks such as slow convergence and complexity. To overcome these limitations, this research proposed a modified approach for BP by introducing the Conjugate Gradient and Quasi-Newton which were Second Order methods together with 'gain' parameter. The performances of the proposed approach is evaluated in terms of lowest number of epochs, lowest CPU time and highest accuracy on five benchmark classification datasets such as Glass, Horse, 7Bit Parity, Indian Liver Patient and Lung Cancer. The results show that the proposed Second Order methods with 'gain' performed better than the BP algorithm.

Keywords: Back Propagation, second order optimization, search direction, classification, gradient descent

1. Introduction

The research on artificial neural network (ANN) was started when a group of researchers conducted by neurologist and mathematician Warren McCulloch and Walter Pitts [1]. The researchers had demonstrated that ANNs are trying to model the learning processes of human brain based in logical methods. The same principle applied in biological neurons in the brain where ANN consists of small processing units known as Artificial Neurons, which can be trained to perform very complex calculations. Moreover, it is known that as human being, we learn how to write, read, understand speech, recognize and distinguish color and patterns by learning from examples. As matter of fact, the same way with ANNs where ANNs are trained rather than programmed.

ANN is used in widespread area of applications including the field of data mining, finance, process control, flight security, medical, marketing, pattern recognition, forecasting, and regression problems [2,3,4,5]. Among many algorithms used in ANN, the most popular training algorithms is back propagation (BP) algorithm.

The BP learning is known for the establishment of the process and become the most standard method where the algorithm can adjust weight and biases for training an ANNs in many domains. The BP network learns by iteratively processing a set of training data or samples of data. The weights as parameter are modified to minimize the network's classification and actual classification for each set of training data. The network starts the process by propagating the error in forward direction which is the transfer of net output from input layer to hidden layer and next output layer. The error between the target output and the actual output is compared at output layer which makes the network back propagates into the input layer with the tuning weight parameter value.

Although the BP algorithm is one of the most widely used algorithm in neural network models [6], it also has their drawbacks such as very slow convergence or in other words slow convergence of learning algorithm as stated by Manjun [7]. Based on the steepest descent technique of the training process it can easily get stuck at local minima and the performances are decreasing when it applied on large scale applications such as pattern recognition and artificial intelligence problems [8].Thus, improvement on BP had getting attention from researchers and those improvements are crucial in order to overcome those limitations.

There have been many researchers that contribute to the improvement of BP such as the use of adaptive gain

*Corresponding author: nazri@uthm.edu.my

proposed by Nazri [9] followed by a research by Norhamreeza [10] that proposed a new procedure that change the parameter of gain adaptively as well as momentum and learning rate. Some of this research was based on the adaptive learning parameters, e.g. the Quickprop [11], the RPROP [12], delta-bar-delta rule [13],and Extended delta-bar-delta rule [14]. Combinations of different techniques can often lead to an improvement in global optimization methods [15].

Since the accuracy of the training and testing process is depending on the weight updation in the BP algorithm, Grzenda and Macukow [16] shows that by selecting the weight domain adaptively, it will increases the accuracy in the classification aspects. Meanwhile, Yu and Wilamowski [17] demonstrated that the used of second order algorithms such as Newton algorithm and Levernberg Marquardt (LM) algorithm also can produced better result which can converge faster than using first order algorithms. Furthermore, Aziz et al. [18, 19] also produced a very good result of using particle swarm optimization in training Elman neural network.

It is demonstrated that the combination of optimal parameters with second order method performed better and faster results as compared to the first order method. Therrefire, this research implements the advantages of Second Order Methods such as Conjugate Gradient Fletcher-Reeves (CGFR) and Conjugate Gradient Polak Ribiere (CGPR) and Quasi-Newton methods of Davidon-Feltcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) together with the optimal 'gain' parameter which we name it as CGFR-AG, CGPR-AG, DFP-AG and BFGS-AG respectively.

The remaining of this paper is organized as follows: Section two describes the most commonly used of Second Order Methods and the implementation of the proposed algorithm. In Section three, the results from the simulation testing are analyses in order to identify the algorithms that perform better on five benchmark data. In Section four, the last section concludes the research findings.

2. Numerical Model

There are two types of second order methods that are very famous and used by researchers in neural networks. The first method is conjugate gradient methods that used to minimize quadratic function of:

$$q(x) = \frac{1}{2} X^T AX - b^T X + C \quad (1)$$

Which is similar to solve $AX = b$, where A is known as a positive definite matrix. The process is computed in iterative sequence of n steps which requires performing n exact linesearches along the conjugate directions. Basically, conjugate gradient methods are used to find the nearest local minimum of a function with the computed gradient along a search direction. Summarizing all points, the methods are known as quadratic termination where a quadratic function as Equation (1) is minimized in at most sequence number of iterations.

The method replaced the previous steepest descent technique as it has drawback in terms of finding the local minimum which requires many iterations to be computed. Based on the formula stated in conjugate gradient algorithms by Fletcher and Powel and Fletcher-Reeves the positive definite quadratic function of n variables in iterations, will take n steps in order to minimize the error which generates the conjugate directions for search of the local minimum. Based on the research proposed by Nazri [9] they showed that BP algorithm had improved further with the use of adaptive gain by new weight vector and new gain vector which are calculated to minimize the number of epochs by the update of error with respect to weight and gain computed in the previous epoch. BP use gradient descent rule:

$$\Delta W_{ij}^n = -\eta^n \frac{\partial E}{\partial W_{ij}^n} \quad (2)$$

Where ΔW_{ij}^n = Weight in node i until j in layer n

η^n = learning rate at step n

∂E =change in error

∂W_{ij}^n = change in weight in node i until j in layer n

For all changes in error with respect to weight is the gradient based search direction in step n as;

$$d^n = \frac{\partial E}{\partial W_{ij}^n} = g^n \quad (3)$$

Thus, the gradient based search direction with the added gain value c_j^n can be compute by

$$d^n = g^n c_j^n \quad (4)$$

Meanwhile, the change in error with respect to gain in node j layer s (c_j^s) can be computed by

$$\frac{\partial E}{\partial c_j^s} = (\sum_k \partial_k^{s+1} W_{k,j}^{s+1}) f'(c_j^s net_j^s) net_j^s \quad (5)$$

Where net_j^s net input in node j layer s.

Thus, in this research it is noticed that the new gain value for the next epoch can also be computed by added of the old value of gain with the updated gain in current epoch as in Equation (6);

$$c_j^{new} = c_j^{old} + \Delta c_j^s \quad (6)$$

Therefore, the proposed adaptive gain is then implemented into the conjugate gradient methods where Hestenes and Stiefel stated that,

$$d_{n+1} = -g_{n+1} + \beta_n d_n \quad (7)$$

Where d_{n+1} = new search direction and g_{n+1} is new gradient based search direction as;

$$\beta_n = \frac{(g_{n+1} - g_n)^T g_{n+1}}{(g_{n+1} - g_n)^T d_n} \quad (8)$$

Where,

d_n = Search direction at step n

The types of conjugate gradient are depends on the certain conditions such as:

a) The line search along search direction d_n is exact: $d_n g_{n+1}^T = 0$

b) The function f is quadratic: $g_{n+1}^T g_{n+1} = 0$

c) The line search along d_{n-1} is exact: $d_n g_n^T = -g_n^T g_n$

CGPR [18] satisfies condition a) and c) where:

$$\beta_n = \frac{g_{n+1} - g_n)^T g_{n+1}}{g_n^T g_n} (9)$$

With the added of gain (c_n) the parameter for CGPRis

$$\beta_n = \frac{g_{n+1}^T c_{n+1} - g_n^T c_n) g_{n+1} c_{n+1}}{g_n^T c_n g_n c_n} (10)$$

Meanwhile, CGFR [19] satisfies all three conditions where the parameter β_n is

$$\beta_n = \frac{g_{n+1}^T g_{n+1}}{g_n^T g_n} (11)$$

Therefore, with the added of gain (c_n) the parameter for CGFR is

$$\beta_{n+1} = \frac{g_{n+1}^T (c_{n+1}) g_{n+1} (c_{n+1})}{g_n^T (c_n) g_n (c_n)} (12)$$

The second type of second order methods is Quasi-Newton method. Since the most crucial limitations of the back propagation algorithm is that the errors easily get stuck at local minima. Therefore it is a crucial to identify the properties in second order methods that can solve those limitations. The condition such as first derivatives and the gradient is $g = 0$ where, the second derivatives which are the computation of Hessian matrix are positive in Quasi-Newton equation is known as the inverse of matrix H_n as:

$$H_{n+1} s_n = y_n (13)$$

Where H_n satisfies Equation (14)

$$H_n = H_n + \nabla_n (14)$$

While, s_n is the change in weight and

$$s_n = W_{n+1} + W_n (15)$$

Change in gradient with the variation of gain is computed as

$$y_n = g_{n+1}(c_{n+1}) - g_n(c_n) (16)$$

Based on the Equation (13), the update represents the difference between Quasi-Newton [22,23,24] which are BFGS and DFP computations are as follow:

DFP updates is given by

$$\nabla_r = (1 + \frac{s_n^T H_n s_n}{y_n^T s_n}) \frac{y_n y_n^T}{y_n^T s_n} - \frac{y_n s_n^T H_n + H_n s_n y_n}{y_n^T s_n} (17)$$

BFGS updates is

$$\nabla_n = \frac{y_n y_n^T}{y_n^T s_n} - \frac{H_n s_n s_n H_n}{s_n^T H_n s_n} (18)$$

The error function value with respect to weight $E(W_n)$ can be computed by;

$$d = -[\nabla^2 E(w)]^{-1} \nabla E(w) (19)$$

In terms of minimization search direction, this method minimize along line between current point and the previous point by using line search technique with the added parameter of learning rate. Lastly, the convergence test is evaluated based on some indicators. For the Quasi-Newton to perform reducing in error, the error with respect to weight which is greater than the convergence tolerance will then indicate that the testing is finished as represent in Equation (19).

Considering the effectiveness of both methods together with the proposed adaptive gain, therefore this research proposed an improved method by implementing the proposed adaptive gain into both of second order methods. The pseudo-code for the proposed methods with the implementation on CGFR and CGPR are given as follow:

Start

Step 1 Randomly initialize the weight vector, the gradient vector g_0 to zero. Set the first search direction d_0 be g_0. Set β=0, epoch=1 and *n*=1. Let N_t be the total number of weight values. Select a convergence tolerance value as CT.

Step 2 At step n, the gradient vector g_n is evaluate

Step 3 Evaluate $E(w_n)$. If $E(w_n) < CT$ then stop training ELSE go to **Step 4**.

Step 4 Calculate a new gradient based search direction:

$$d_{n+1} = -g_n + \beta_{n-1} d_{n-1}$$

Step 5 If n>1 THEN, update

$$\beta_n = \frac{g_{n+1}^T c_{n+1} - g_n^T c_n) g_{n+1} c_{n+1}}{g_n^T c_n g_n c_n}$$ for CGPR-AG or

$$\beta_{n+1} = \frac{g_{n+1}^T (c_{n+1}) g_{n+1} (c_{n+1})}{g_n^T (c_n) g_n (c_n)}$$ forCGFR-AG ELSE go to **Step 6**.

Step 6 If *[(epoch +1)/N_t] = 0* THEN 'restart' the gradient vector with $d_n = -g_{n-1}$ ELSE go to **Step 7**.

Step 7 Calculate the optimal value for learning rate η_n by using line search technique.

Step 8 Update $w_n : w_{n+1} = w_n - \eta_n d_n$

Step 9 The new gradient vector g_{n+1} is evaluated.

Step 10 The new gradient based search direction: $d_n = -g_{n+1} + \beta_n d_n$ is calculated.

Step 11 Set $n = n+1$ and go to **Step 2**.

Whereas, the pseudo-code for the proposed methods with the implementation on DFP and BFGS are given as follow:

Start

Step 1 Randomly, initialize the initial weight vector *w(0)* and initialize a positive definite of the Hessian matrix H(0). Set a convergence tolerance CT.

Step 2 Compute the gradient based search direction at step by taking into account variation $d_n = -H_n g_n$

Step 3 Calculate the optimal value for η_n by using line search technique.

Step 4 Update $w_n : w_{n+1} = w_n - \eta_n d_n$

Step 5 Compute: $s_n = w_{n+1} - w_n$

$$y_n = g_{n+1} - g_n$$

$$\nabla_n = \frac{y_n^T y_n^T}{y_n^T s_n} - \frac{H_n s_n s_n^T H_n}{y_n^T s_n}$$

For BFGS methods

$$\nabla_n = (1 + \frac{s_n^T H_n s_n}{y_n^T s_n}) \frac{y_n y_n^T}{y_n^T s_n} - \frac{y_n s_n^T + H_n s_n y}{y_n^T s_n}$$

Step 6 Update the inverse matrix $H_{n+1} = H_n + \nabla_n$

Step 7 Compute the error function $E(w_n)$.

Step 8 If $E(w_n) >$ CT go to **Step 2**, else stop.

3. Results and Discussion

The simulation testing was programmed by using Matlab R2012b software.The testing was conducted by setting fixed number of 5 hidden nodes, 0.7 learning rates, 0.3 momentum, 0.01 of target error and maximum of 5000 epochs. The performances are measured in terms of lowest number of epochs, lowest CPU Time and highest accuracy.

The simulation testing was done on five benchmarks dataset from UCI Machine Learning Repository (UCIMLR) which are Glass [25], Horse [26], 7Bit Parity [27,28], Indian Liver Patient [26] and Lung Cancer [29]. The following ten algorithms were analyzed and simulated for each problem.

- ***(a)*** Back Propagation (BP)
- ***(b)*** Back Propagation with Gain (BP-AG)
- ***(c)*** Conjugate Gradient Polak Ribiere (CGPR)
- ***(d)*** Conjugate Gradient Polak Ribiere with Gain (CGPR-AG)
- ***(e)*** Conjugate Gradient Fletcher Reeves (CGFR)
- ***(f)*** Conjugate Gradient Fletcher Reeves withGain (CGFR-AG)
- ***(g)*** Davidon Feltcher Powell (DFP)
- ***(h)*** Davidon Feltcher Powell with Gain (DFP-AG)
- ***(i)*** Broyden Fletcher Goldfarb Shanno (BFGS)
- ***(j)*** Broyden Fletcher Goldfarb Shanno with Gain (BFGS-AG).

For every problem selected, 50 different trials were run. The performance in terms of epochs numbers is validated in terms of lowest number of epochs which indicates the greater performance where the convergence rate are faster as the mean square error reached during that cycle or epochs process. Meanwhile, the lowest CPU time also indicates better performance since the CPU process takes shorter time in order the algorithm to converge. Besides, the highest accuracy will give better result in terms of how much the algorithm gives the correct output based on the input processed. At the of this simulation, the simulation results such as the mean of the number of iterations (mean), the standard deviation (SD), and the number of failures are recorded.

3.1 Glass Classification problem

First classification problem dataset is Glass dataset. It contains 214 instances and 10 attributes such as Magnesium, Aluminium, Silicon, Potassium and others. There are 9 inputs and 6 outputs. The data was used to predict type of glass such as tableware, containers and headlamps. Moreover, the datasets was applied in the criminological investigation where at the scene of the crime, the glass left can be used as evidence.

Table 1 demonstrates that the proposed methods outperform other methods in terms of number of epochs and CPU time.As we can see that the proposed DFP-AG and CGPR-AG reached the lowest epoch numbers which indicates better performance as compared to BP-AG.Numerically, both methods are 55 times faster and only need 42 epochs to reach target error as compared to BP-AG with 2335 epochs. Meanwhile, CGFR-AG takes

shortest time to converge with 2.86 seconds which is nearly 0.96 improvement ratio when compared to CGFR.

In terms of accuracy, the highest accuracy was achieved at 80.75% by CGFR and 80.53% by CGFR-AG methods. Meanwhile, the proposed QN methods of BFGS-AG performed better with 3.14 percent when compared to BFGS. Although second order methods without gain parameter perform better, yet the proposed CGFR-AG performs 10.42% better than BP-AG. However, it is noticed that of all methods in this problem, the proposed CGFR-AG method was the fastest second order method with shortest CPU time and highest accuracy.

Table 1: Summary of algorithms performance for Glass datasets

Performance	Mean number of epochs	Mean CPU Time(s) to converge	Accuracy (%)
BP	3104	80.87	74.95
BP-AG	**2335**	**33.12**	**70.11**
CGPR	63	5.47	79.38
CGPR-AG	**42**	**3.70**	**78.36**
CGFR	58	3.82	80.75
CGFR-AG	**43**	**2.86**	**80.53**
DFP	57	9.82	74.85
DFP-AG	**42**	**7.33**	**70.26**
BFGS	132	23.52	68.10
BFGS-AG	**43**	**7.40**	**71.24**

3.2 Horse Classification problem

The benchmark dataset contains 368 instances with 300 used for training and the rest for testing. Besides, it has 28 numbers of attributes which related to horse physical information such as age, surgery, pulse and others. The source of the dataset was created by Mary McLeish and Matt Cecile from Department of Computer Science of University of Guelph in Ontario, Canada.

The disparity between the convergence rate of first and second order methods for Horse classification problem is illustrated in Table 2. The result clearly demonstrates that algorithms which implement the proposed method exhibit very good average performance in order to reach the target error particularly when it is compared with BP method. The proposed method DFP-AG took only 54 iterations to converge as compared to BP-AG method with 2728 iterations which is a significant improvement with 50 times less epochs. Meanwhile, the lowest CPU time is achieved when the proposed CGFR-AG methods are implemented with 4.05 seconds in results with difference of 1.08 when compared to CGFR. Whereas, the proposed CGPR-AG required only 4.23 seconds faster when compared to CGPR.

At the same time, the proposed CGFR-AG performs better with nearly 46 times shortest in CPU time than BP method. In terms of accuracy, CGFR-AG is the only methods that achieved the highest percentage of accuracy of 75.53 when compared to the other Second Order Methods with adaptive gain and it increase nearly 3 % as compared to CGFR. At the same time, the highest accuracy among all methods tested was achieved by CGPR with 77.33 % which is 2.18 % higher than CGPR-AG. Meanwhile, CGFR-AG and CGPR-AG have the accuracy of 75.53% and 75.15% respectively which shows the better performance of the proposed methodas compared to the BP-AG methods with the accuracy of 71.90. The proposed CGFR-AG is the best second order method in the classification problems since it has highest accuracy and lowest CPU Time.

Table 2: Summary of algorithms performance for Horse classification

Performance	Mean number of epochs	Mean CPU Time(s) to converge	Accuracy (%)
BP	4272	186.05	72.63
BP-AG	**2728**	**40.61**	**71.90**
CGPR	76	14.90	77.33
CGPR-AG	**55**	**10.67**	**75.15**
CGFR	73	5.13	72.56
CGFR-AG	**55**	**4.05**	**75.53**
DFP	74	15.74	68.25
DFP-AG	**54**	**11.75**	**66.90**
BFGS	39	8.58	65.72
BFGS-AG	**53**	**11.61**	**66.42**

3.3 7Bit Parity Classification problem

The dataset consist of a mixture of even and odd parity tuples. It contains 128 instances where the network structural design has 7 inputs 5 hidden neurons in the hidden layer, and 1 output in the output layer. It has forty connection weights and six biases.

Table 3 shows the summary of algorithms' performance for 7Bit Parity dataset. In terms of epochs, BFGS gives better performance among all methods with lowest number of 53 epochs. Besides, DFP-AG performs better followed by CGPR-AG methods with the values of 73 and 76 numbers of epochs respectively out of 5000 epochs tested as compared to other methods. Meanwhile, the proposed CGFR-AG performs 4 times better as compared with BP methods where CGFR-AG needs 1065 epochs compare to BP with 4709 epochs.

In other aspects, the performance in terms of lowest CPU time is recorded by using the proposed CGPR-AG with 4.97 seconds followed by BP-AG with 5.31 seconds. Next, DFP-AG and CGFR-AG performs 8.03 seconds and 13.34 seconds better when compared to DFP and CGFR respectively.

In terms of accuracy, the highest accuracy is achieved by using the methods of the proposed BP-AG with

66.10% accuracy followed by CGFR of accuracy 57.96 with the difference of 8.14% and the proposed CGFR-AG performs better with more than 50% accuracy of 57.75%.

Table 3: Summary of algorithms performance for 7Bit Parity classification

Performance	Mean number of epochs	Mean CPU Time(s) to converge	Accuracy (%)
BP	4708	70.28	63.13
BP-AG	**1658**	**5.31**	**66.10**
CGPR	109	7.02	40.68
CGPR-AG	**76**	**4.97**	**40.57**
CGFR	1035	82.77	57.96
CGFR-AG	**1065**	**69.43**	**57.75**
DFP	103	27.46	43.03
DFP-AG	**73**	**19.43**	**43.06**
BFGS	53	14.00	45.66
BFGS-AG	**69**	**18.39**	**44.85**

3.4 Indian Liver Patient Disease(ILPD)

The benchmark dataset contain of 583 instances where 416 are the liver patient records and 167 non liver patient records. It also contains 10 attributes such as age and gender of patients. The dataset was taken from north east of Andhra Pradesh, India. The dataset was used to identify whether the patient has the liver disease.

Table 4 shows the summary of performances of all algorithms for ILPD classification. In terms of average number of epochs, all second order methods with adaptive gain perform better as compared to all methods without gain and BP methods. As we can see that the proposed CGPR-AG, DFP-AG and BFGS-AG are lowest in epoch numbers which indicates their better performance when compared to BP-AG. Numerically, all three proposed methods are 82 times faster and only need 60 epochs to reach target error as compared to BP with 4957 epochs. Meanwhile, the proposed CGFR-AG performs better with difference of 22 epochs when compared to CGFR and 78 times better than BP methods.

Another performance aspect which demonstrates significant improvement is CPU Time, all proposed methods also performs better. Based on Table 3, the proposed CGPR-AG performs better 60 time shortest than BP-AG with 5.38 seconds followed by CGFR-AG, DFP-AG and BFGS-AG with 5.49, 11.03 and 11.12 seconds respectively. Thus, the proposed CGPR-AG performs 12 times shortest than BP-AG and 17 times shortest than BP.

In terms of accuracy, CGPR performs better with accuracy of 66.63% which is 1.83% higher than CGPR-AG and 3.31% higher than BP-AG. Next, CGFR-AG and BFGS-AG both are 4.14% and 0.54% better in accuracy when compared with CGFR and BFGS. The proposed CGPR-AG is the best second order methods since it converge with shortest CPU time and require less iteration to converge.

Table 4: Summary of algorithms performance for ILPD classification

Performance	Mean number of epochs	Mean CPU Time(s) to converge	Accuracy (%)
BP	4957	96.10	60.61
BP-AG	**3841**	**66.00**	**63.32**
CGPR	89	7.81	66.63
CGPR-AG	**60**	**5.38**	**64.80**
CGFR	85	7.47	58.68
CGFR-AG	**63**	**5.49**	**62.82**
DFP	86	15.98	61.94
DFP-AG	**60**	**11.03**	**61.49**
BFGS	61	11.61	61.26
BFGS-AG	**60**	**11.12**	**61.80**

3.5 Lung Cancer

The benchmark dataset were created by Stefan Aeberhard on 1st May 1992 which consists of 32 numbers of instances and 56 numbers of attributes. The data was published in journal "Optimal Discriminant Plane for a Small Number of Samples and Design Method of Classifier on the Plane", by Hong, Z.Q. and Yang, J.Y in pattern recognition application on 1991.

Table 5 shows summary of algorithm's performances for Lung cancer classification. The result clearly shows that algorithms which implement proposed method exhibit good average performance in order to reach target error. As we can see that DFP-AG is 60 times faster as compared to BP-AG. The proposed method of DFP-AG only needs 48 iterations to converge as compared to BP-AG method with 2909 iterations.

Meanwhile, the lowest CPU time is recorded when the proposed CGPR-AG methods are implemented with 5.73 seconds in results with 0.43 seconds better when compared to CGPR and 16 times faster than BP. While, CGFR performs 76.15 seconds which is better with nearly 1.2 times when compared with BP.

The highest accuracy achievedwere 73.50% by using the methods of BP-AG followed by CGPR-AG with 69.03% with difference of 4.47%. The proposed CGPR-AG is the best second order methods since it require less number of epochs and lowest CPU time to converge.

Table 4: Summary of algorithms performance for Lung Cancer classification

Performance	Mean number of epochs	Mean CPU Time(s) to converge	Accuracy (%)
BP	5000	92.08	0.00
BP-AG	**2909**	**9.15**	**73.50**
CGPR	74	6.16	68.02
CGPR-AG	**69**	**5.73**	**69.03**
CGFR	910	76.15	68.91
CGFR-AG	**778**	**67.32**	**67.49**
DFP	69	12.38	62.50
DFP-AG	**48**	**8.64**	**63.60**
BFGS	38	7.11	61.11
BFGS-AG	**49**	**9.14**	**62.20**

4. Summary

The performance of the second order methods with adaptive gain (CGPR-AG, CGFR-AG, CFP-AG and BFGS-AG) against the standard second order methods without gain (CGPR, CGFR, DFP, BFGS) and Back Propagation methods (BP and BP-AG) have been presented in this research. All algorithms were evaluated and compared in-terms of speed of convergence, number of epochs, CPU time and accuracy on five benchmark problems. The simulation results showed that the proposed second order methods with adaptive gain performed significantly better than the other methods and are good training alternatives than first order ANNs.

Acknowledgement

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education (MOHE) Malaysia for financially supporting this Research under IGSP grants note U420 and under Trans-displinary Research Grant Scheme (TRGS) vote no. T003.

References

[1] Rumelhart, D. E., Hinton, G. E., Williams, R. J.: Learning Internal Representations by error Propagation, J. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1. (1986)

[2] Samsudin, N.A., Bradley, A.P. “Group-based meta- classification”. Proceedings - International Conference on Pattern Recognition. Institute of Electrical and Electronics Engineers Inc., 2008.

[3] Samsudin, N.A., Bradley, A.P. “Extended naïve bayes for group based classification”. Advances in Intelligent Systems and Computing. Vol. 287, Springer Berlin Heidelberg, 2014. Page. 497-506.

[4] Yassin, I. M., Jailani, R., Ali, M. S. A. M., Baharom, R., Hassan, A. H. A., & Rizman, Z. I. (2017). Comparison between cascade forward and multi-layer perceptron neural networks for NARX functional electrical stimulation (FES)-based muscle model. International Journal on Advanced Science, Engineering and Information Technology, 7(1), 215-221.

[5] Jabril Ramdan,Khairuldin Omar and Mohammad Faidzul,"A Novel Method to Detect Segmentation points of Arabic Words using Peaks and Neural Network," International Journal on Advanced Science, Engineering and Information Technology, vol. 7, no. 2, pp. 625-631, 2017.

[6] Li, Jing, et al. "Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its improvement." Advances in Computer Science and Information Engineering. Springer Berlin Heidelberg, 2012. 553-558.

[7] Manjun , L. S. Z (2002). Computer and Information Department Hefei University of Technology Hefei, Anhui ,China, (1), pp.293–296.

[8] Bi, W., Wang, X., Tang, Z., & Tamura, H.(2005). Avoiding the Local Minima Problem in Backpropagation Algorithm with Modified Error Function. IEICE Trans. Fundam. Electron. Commun. Comput. Sci.,E88-A (12), pp. 3645-3653.

[9] Nawi, N., Rozaida Ghazali, and M. Salleh."An approach to improve back-propagation algorithm by using adaptive gain." Biomedical Soft Computing and Human Sciences 162 (2010): 125-134.

[10] Abdul Hamid, Norhamreeza. The effect of adaptive parameters on the performance of back propagation. Diss. Universiti Tun Hussein Onn Malaysia, 2012.

[11] Fahlman, S. E. (1988). Faster-learning variations on backpropagation: An empirical study. Proceedings of the 1988 Connectionist Models Summer School, 38-51.

[12] Riedmiller, M & Braun, H. (1993). A direct adaptive method for faster backpropagation learning the PROP algorithm. Proceedings of the IEEE international conference on Neural Networks (ICNN), Vol. I, San Francisco, CA, 586-591.

[13] Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. Neural Networks, 1, 169-180

[14] Minai, A. A., & Williams, R. D (1990). Acceleration of back-propagation through learning rate momentum adaptation. Proceedings of the International Joint Conference on Neural Networks, 1676-1679.

[15] Yu, H., and B. M. Wilamowski. "Neural network training with second order algorithms." Human

Computer Systems Interaction: Backgrounds and Applications 2. Springer Berlin Heidelberg, 2012. 463-476.

[16] Grzenda, M.; Macukow, B., "The role of weight domain in evolutionary design of multilayer perceptrons," Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on, vol.6, pp.596-599, vol.6, 2000.

[17] Yu, H., and B. M. Wilamowski. "Neural network training with second order algorithms." Human Computer Systems Interaction: Backgrounds and Applications 2. Springer Berlin Heidelberg, 2012. 463-476.

[18] Ab Aziz, M.F., Hj Shamsuddin, S.M."Quantum particle swarm optimization for Elman recurrent network". AMS2010: Asia Modelling Symposium 2010 - 4th International Conference on Mathematical Modelling and Computer Simulation, (2010), page. 133-137.

[19] Ab Aziz, M.F., Hj Shamsuddin, S.M. "Enhancement of quantum particle Swarm Optimization in Elman recurrent network with bounded VMAX function". Jurnal Teknologi, (2016) Vol. 78, Issue 12, Page 43-48.

[20] R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization. British Computer J., 1963: p. 163-168.

[21] Fletcher R. and Reeves R. M., Function minimization by conjugate gradients. Comput. J., 1964.7(2): p. 149-160.

[22] Adrian J. Sheperd, Second Order Methods for Neural Networks-Fast and Reliable Training Methods for Multi-layer Perceptrons, ed. J.G.Taylor. 1997: Springer. 143.

[23] Polak E., *Computational Methods in Optimization.* (New York: Academic Press), 1971.

[24] Dennis J. E. and Schnabel R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* 1983, Englewood Cliffs NJ:Prentice-Hall.

[25] Ian W. Evett and Ernest J. Spiehler. Rule Induction in Forensic Science. CentralResearch Establishment. Home Office Forensic Science Service. Aldermaston, Reading, Berkshire RG7 4PN

[26] Julie Greensmith. New Frontiers For An Artificial Immune System. Digital Media Systems Laboratory HP Laboratories Bristol. 2003.

[27] 7-Bit Parity Training data, *http://homepages.cae.wisc.edu/~ece539/data/parity7r*.

[28] 7-Bit Parity Testing data, *http://homepages.cae.wisc.edu/~ece539/data/parity7t.*

[29] Bendi Venkata Ramana, Prof. M. S. Prasad Babu and Prof. N. B. Venkateswarlu, A Critical Comparative Study of Liver Patients from USA and INDIA: An Exploratory Analysis, International Journal of Computer Science Issues, ISSN:1694-0784, May 2012.