© Universiti Tun Hussein Onn Malaysia Publisher's Office

EmAIT

Journal homepage: http://publisher.uthm.edu.my/ojs/index.php/emait
e-ISSN : 2773-5540

Emerging Advances in Integrated Technology

The Effect of Hyper-Parameters on the Performance of Third Order Neural Network Algorithms on Medical Classification Data

Nazri Mohd Nawi[1*], Prihastuti Harsani[2] ,Eneng Tita Tosida[2], Khairina Mohamad Roslan[1]

[1]Soft Computing and Data Mining Centre (SMC), Faculti Sains Komputer dan Teknologi Maklumat (FSKTM), Universiti Tun Hussein Onn Malaysia (UTHM), 86400, Parit Raja, Batu Pahat, Johor, MALAYSIA

[2]Computer Science Training Ceneter (ComSTraC), Data Science Center (DSC), Computer Science Department, Universitas Pakuan, Jl. Pakuan Po Box 452, Bogor, Jawa Barat, 16143, INDONESIA

*Corresponding Author

DOI: https://doi.org/10.30880/emait.2021.02.01.007
Received 11 April 202100; Accepted 10 Jun 2021; Available online 30 Jun 2021

Abstract: The artificial neural network (ANN) particularly back propagation (BP) algorithm has recently been applied in many areas. It is known that BP is an excellent classifier for nonlinear input and output numerical data. However, the popularity of BP comes with some drawbacks such as slow in learning and easily getting stuck in local minima. Improving training efficiency of BP algorithm is an active area of research and numerous papers have been reviewed in the literature. Furthermore, the performance of BP algorithm also highly influenced by the size of the datasets and the data preprocessing techniques that been chosen. This paper presents an improvement of BP by adjusting the two term parameters on the performance of third order neural network methods. This work also demonstrates the advantages of using preprocessing dataset in order to improve the BP convergence. The efficiency of the proposed method is verified by means of simulation on medical classification problems. The results show that the proposed implementation significantly improves the learning speed of the general back-propagation algorithm.

Keywords: Medical_diagnosis, neural network, back propagation

1. Introduction

Medical data commonly known as health information organized by technicians that help them to keep track medical records for healthcare facilities. They make sure that patient data is accurate, and all input data were kept into databases, where it can be analyzed for service quality and insurance reimbursement purposes. In addition, they ensure the confidentiality of sensitive medical information. Some medical data analysts choose to specialize in a particular area of the field, such as cancer treatment. It is important for the top management to do some decision making by using the data that are kept in database. However, there is no such systematic way that had been used by hospital to utilize the data and as a result the decision making will become very difficult.

Recently, Artificial Neural Networks (ANN) had gained popularity among researchers and specialist in medical. Neural networks are a branch of "Artificial Intelligence" where is a system loosely modeled based on the human brain. The field goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. Neural networks are a powerful technique to solve many real world problems this is because the ability to learn from experience in order to improve their performance and able to adapt with the changes in the environment [1-3].

*Corresponding author: nazri@uthm.edu.my

Moreover, ANN are able to deal with incomplete information or noisy data and can be very effective especially in situations where it is not possible to define the rules or steps that lead to the solution of a problem. Biologically, Artificial Neural Network (ANN) is an interconnected group of artificial neurons that uses mathematical or computational model for information processing based on connectionist approach to computation. In more practical terms, ANN are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data [4].

The performances of ANN particularly back propagation (BP) algorithm are very much depend on some parameter such as learning rate, momentum, target error and hidden nodes. Learning rate is defined in the context of optimization, and minimizing the loss function of a neural network. It defines a cost function for a neural network, and the goal is to minimize this cost function. For this research, we use gradient descent or other variants of it where the model parameters (here weights and biases in the network) are updated in a way to decrease the cost function. It determines how quickly or how slowly you want to update the parameters [5].

Usually, one can start with a large learning rate, and gradually decrease the learning rate as the training progresses. Momentum is a physical property that enables a particular object with mass to continue in its trajectory even when an external opposing force is applied, this means overshoots. For example, one speeds up a car and then suddenly hits the brakes, the car will skid and stop after a short distance overshooting the mark on the ground. Therefore, the networks must be designed by trial and error: this empirical approach to network design is difficult to surmount. Furthermore, there is always a danger of overtraining a neural network because that minimizing the error measure occasionally does not correspond to finding a well-generalizing neural network. Therefore, this paper analyses the performance of BP by analyzing the effect of adjusting two parameters (learning rate and momentum) on some medical datasets.

The remaining of the paper is organized as follows. The first section discusses on the basic concept of BP algorithm and its parameters are reviewed. While in the next section presents the tested on medical benchmark problems. This paper is concluded in the final section.

2. Literature Review

This section will discuss on basic concepts about BP and some parameters that contribute to the performance of BP algorithm. Towards the end of this literature review we present some adjustment on the parameters and the implementation with third order methods on medical classification data.

2.1 Classification on Medical Data

Medical classification, or medical coding, is the process of transforming descriptions of medical diagnoses and procedures into universal medical code numbers. The diagnoses and procedures are usually taken from a variety of sources within the health care record, such as the transcription of the physician's notes, laboratory results, radiologic results, and other sources [6]. Diagnosis codes track diseases and other health conditions. These diagnosis and procedure codes are used by health care providers, government health programs, private health insurance companies, workers' compensation carriers, software developers, and others for a variety of applications in medicine, public health and medical informatics, including statistical analysis of diseases and therapeutic actions, reimbursement, knowledge-based and decision support systems and direct surveillance of epidemic or pandemic outbreaks.

2.2 Artificial Neural Networks

ANN have been developed as generalizations of mathematical models of biological nervous systems. Thus, ANN was applied based on the adaptations of processing unit of human’s brains and imitates the process to be modeled in the neural network. The basic elements in the ANN are the data that presented to input layers which then passed on to the hidden layer and next to output layer. Figure 1 shows the Multilayer Perceptron (MLP) structure which has 3 nodes of input, 4 nodes of hidden layer and 2 nodes on the output layers. Thus, the architecture of MLP network is 3-4-2. Each connection between the nodes has weight associated with it. Next, a learning algorithm where in this research the most popular and stable algorithm is back propagation algorithm and third order neural network algorithm is used to test the networks performance. As all the parameter was set, the data will be transforms to the network inputs and training process will begins.

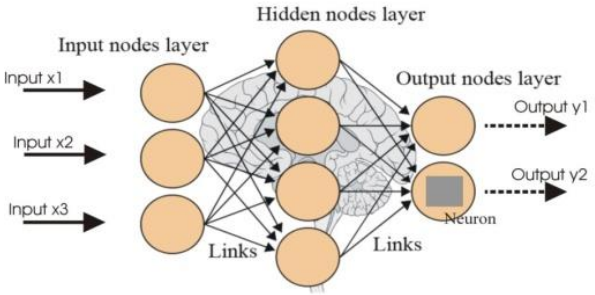

Fig. 1- Artificial neural networks

Back Propagation Neural Network was first proposed by Rumelhart and McClelland [7]. The back propagation neural network algorithm is a multi-layer feedforward network trained according to error back propagation algorithm and is one of the most widely applied neural network models. It works by approximating the non-linear relationship between input and the output by adjusting the weight values internally. It can further be generalized for the input that is not included in the training patterns or predictive abilities. Back Propagation Neural Network is also considered as one of the simplest and most general methods used for supervised training of multi-layer neural network and been used in many different types of applications.

In addition, Back Propagation Neural Network are also used for prediction and classification because they are using gradient descent (GD) rule which attempts to minimize the error of the network by moving down the gradient of the error curve, Back Propagation Neural Network. Basically back Propagation Neural Network is a multilayer network that has three or more layer which are fully connected. It means that every neuron in each layer is connected to every other neuron in the adjacent forward layer. A neuron determines its output in a manner that is similar to Rosenblatt's perceptron. The derivative of this function is easy to compute. It guarantees the neuron output is bounded between 0 and 1. Since all the hidden neuron or nodes have contributed to the errors evident in the output layer, the output error signals are transmitted backwards from the output layer to each node in the hidden layer that immediately contributed to the output layer. This process is then repeated, layer by layer, until each node in the network has received an error signal that describes its relative contribution to the overall error as shown in Figure 2.

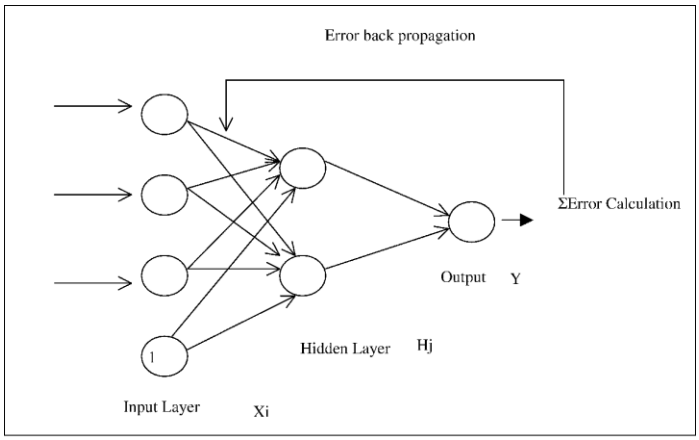

Fig 2 - Back Propagation Neural Network Typology

Once the error signal for each node has been determined, the errors are then used by the nodes to update the values for each connection weights until the network converges to a state that allows all the training patterns to be encoded. Back Propagation Neural Network algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent. The weights that minimize the error function is then considered to the learning problem.

2.3 Two Term Parameters

There are many parameters that can affect the performance of BP. However, for this research only two parameters of Back Propagation had been selected which are learning rate and momentum coefficient. Those parameters are used for controlling the weight adjustment along the descent direction.

The momentum is another possible way to improve the rate of convergence is by adding some momentum to the adjustment expression [8] and will speed up the convergence, stabilize the training procedure and avoid the local minima. Basically, the momentum is set to be constant in the interval [0, 1]. This is because, it is discovered from simulations that the fixed momentum value to speed up learning only when the recent downhill gradient of the error function and the last change in weight have a parallel direction. When the recent negative gradient is in a crossing direction to the previous update, the momentum may cause the weight to be altered up the slope of the error surface as opposed to down the slope as preferred. The modification of conventional back propagation algorithm in the proposed algorithm that uses adaptive learning rate and momentum where the learning rates are adjusted at each iteration to speed up the training time.

The learning rate is one of the most effective means to accelerate the convergence of BP learning which values lies between [0,1]. It is a crucial factor to control the variable of the neuron weight adjustments for each iteration during the training process and therefore it affects the convergence rate. Learning rate defined in the context of optimization, and minimizing the loss function of a neural network. It defines a cost function for a neural network, and the goal is to minimize this cost function. For this optimization problem, we use gradient descent or other variants of it where the model parameters (here weights and biases in the network) are updated in a way to decrease the cost function. It determines how quickly or how slowly you want to update the parameters. The convergence speed is dependence on the choice of learning rate. The algorithm will take longer time to converge or may never converge or may never converge if the learning rate is too small. Therefore, the network will accelerate the convergence rate significantly and still possibly will cause the instability if the learning rate value is too high. The value of learning rate usually set to be constant for all weights in the whole learning process.

3. The Proposed Third Order Neural Network

According to Fletcher and Powel [9] and Fletcher and Reeves [10] most widely used Conjugate Gradient (CG) algorithms is ability to generate in very economical fashion, a set of vectors with a property known as conjugacy. Both these procedures generate conjugate search directions and therefore aim to minimize a positive definite quadratic function of variables in steps. The proposed algorithm referred to Rivaie, Mustafa, Ismail and Leong (RMIL/AG) begins the minimization process with an initial estimate and an initial search direction as:

$$d_0 = -\nabla E(w_0) = -g_0 \tag{1}$$

The search direction at $(n+1)^{th}$ iteration is calculated as:

$$d_{(n-1)} = -\frac{\partial E}{\partial w_{(n+1)}}(c_{i,n+1}) + \beta_{(n+1)} d_n(c_{i,n}) \tag{2}$$

where the scalar $\beta_{(n+1)}$ is to be determined by the requirement that d_n and d_{n+1} must fulfil the conjugacy property [11]. There are many formulae for the parameter $\beta_{(n+1)}$ and the choice of the formulae for selection of $\beta_{(n+1)}$ is problem dependent [11]. In this paper, common formula as referred by Rivaie, Mustafa, Ismail and Leong [11] (RMIL) is used which has been stated as:

$$\beta_n^{RMIL} = \frac{g_n^T (g_n - g_{n-1})}{\| d_{n-1} \|^2} \tag{3}$$

Like $\beta_{(n)}$, the computation of learning rate η also requires knowledge as that of $\beta_{(n)}$. The learning rate η can be optimally chosen as to minimize the error $E(\eta)$ along the chosen search direction d_n.

$$E(\eta) = E(w_{(n)}(\eta)) = E(w_{n-1} + \eta_{n-1} d_{n-1}) \tag{4}$$

The given us an automatic procedure for the setting the learning rate, once the search direction is chosen. This

procedure is also referred to as 'line search' method.

In this paper we used golden section search technique to obtain optimized learning rate. The golden search technique starts by restricting η in $[\eta_1, \eta_h]$. In this paper wet set $\eta_1 > 0$ and $\eta_h < 1$, then the following steps are performed.

Compute $E(\eta_1),\ E(\eta_h)$

If $E(\eta_1) < E(\eta_h)$, then set $\eta_h = \eta_1 - 0.618(\eta_1 - \eta_h)$

If $E(\eta_1) > E(\eta_h)$, then set $\eta_h = \eta_1 + 0.618(\eta_1 - \eta_h)$

The process is repeated until $(\eta_1 - \eta_h) < \varepsilon$ and then set $\eta = \frac{\eta_1 + \eta_h}{2}$.

The complete RMIL/AG [11] [10] algorithm works as follows:

Step 1 Initialize the weight vectors randomly, the gradient vector g_0 to zero and gain vector to unit values. Let the first search direction d_0 be g_0. Set $\beta_0 = 0$, $epoch = 1$ and $n = 1$. Let *Nt* be the total number of weight values. Select a convergence tolerance *CT*.

Step 2 At step *n*, evaluate gradient vector $g_n(c_n)$.

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 which is a function of gain parameter: $d_n = -g_n(c_n) + \beta_n d_{n-1}$.

Step 5 IF $n > 1$ THEN,

update $\beta_n = \frac{g_n^T(c_n)(g_n(c_n) - g_{n-1}(c_n))}{||d_{n-1}(c_n)||^2}$

ELSE go to ***step 6***.

Step 6 IF $[(epoch + 1)/Nt] = 0$ THEN 'restart' the gradient vector with $d_n = -g_{n-1}(c_{n-1})$ 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 Evaluate new gradient vector $g_{n+1}(c_{n+1})$ with respect to gain value c_{n+1}.

Step 10 Calculate new search direction:

$d_{n+1} = -g_{n+1}(c_{n+1}) + \beta_{n+1}(c_n) d_n$

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

4. Results and Discussions

The performance criterion used in this research focuses on the speed of convergence, measured in number of iterations, CPU time and accuracy. Two algorithms have been utilized in these researches which are Halley with BFGS and Halley with DFP methods which is representing Third Order method. 5 hidden nodes were selected throughout this research because it is the most stable architecture for selected datasets. Since the third order neural networks perform very fast for their convergence therefore, this research selected the maximum of 1500 epoch for running the simulation. This research was carried out by testing the algorithms performance using three medical data classification such as Breast Cancer, Diabetes, and Heart. The simulation testing was done by using Matlab R2010b software and performed on a CPU of Intel(R) 1017U, with 1.60 GHz processor. Meanwhile, some values are set as shown in Table 1.

Table 1 - Fixed variables

Variables	*Value*
Hidden Nodes	5
Target Error	0.001
Maximum Epoch	1500
Trials Total	50
Momentum	0.3, 0.4, 0.5
Learning Rate	0.3, 0.5, 0.7

The simulation required data such as epoch, CPU time and accuracy. The data is then calculated into average. The results are recorded into a table of summary of epoch, CPU time, accuracy, Halley with BFGS and Halley with DFP.

4.1 Breast Cancer Data Set

The first benchmark problem data set is Breast Cancer data set. This is one of three domains provided by the Oncology Institute that has repeatedly appeared in the machine learning literature. Obtained from UCI Machine Learning Website, this data set includes 350 instances altogether. While for the testing example are 174 instances. The instances are described by 9 attributes of input and 2 attributes of output. The results of the testing were recorded as in Table 2 and 3. As summarized in the Table 3, Halley with DFP performed better than Halley with BFGS with the highest average accuracy of 93.57 percent at parameter with the learning rate 0.5 and momentum 0.3. Table 2 shows that, Halley with BFGS performed last at parameter learning rate of 0.3 and momentum of 0.3 and achieves accuracy of 93.06 percent. Other than that, Halley with BFGS performs better in the term of CPU time average where Halley with BFGS is faster than Halley with DFP. As Halley with DFP reach 63 epochs to converge, Halley with BFGS performs with less epochs average that is 51 epochs. This means Halley with BFGS converges to global minima within fewer epochs, faster than Halley with DFP.

Table 2 - The performance of Halley with BFGS on Breast cancer data set

HALLEY WITH BFGS			
PARAMETER	**ACCURACY (%)**	**CPU TIME**	**EPOCH**
LR= 0.3 , CM=0.3	**93.06**	**2.92**	**51**
LR= 0.3 , CM=0.4	93.18	3.00	52
LR= 0.3 , CM=0.5	93.23	3.02	53
LR= 0.5 , CM=0.3	93.11	3.00	51
LR= 0.5 , CM=0.4	93.09	2.98	52
LR= 0.5 , CM=0.5	93.07	3.04	53
LR= 0.7 , CM=0.3	93.11	3.08	55
LR= 0.7 , CM=0.4	93.19	3.04	54
LR= 0.7 , CM=0.5	93.09	2.96	52

Table 3 - The performance of Halley with DFP on Breast cancer data set

HALLEY WITH DFP			
PARAMETER	**ACCURACY (%)**	**CPU TIME**	**EPOCH**
LR= 0.3 , CM=0.3	93.44	3.10	57
LR= 0.3 , CM=0.4	93.34	3.18	56
LR= 0.3 , CM=0.5	93.45	3.17	57
LR= 0.5 , CM=0.3	**93.57**	**3.53**	**63**
LR= 0.5 , CM=0.4	93.53	3.30	60
LR= 0.5 , CM=0.5	93.36	3.17	58
LR= 0.7 , CM=0.3	93.31	3.07	55
LR= 0.7 , CM=0.4	93.44	3.13	55
LR= 0.7 , CM=0.5	93.44	3.14	56

4.2 Heart Data Set

The second benchmark problem data set is Heart data set. This data set contains 37 attributes, where 35 were the input and 2 were the output. The data sets taken have 152 instances. Meanwhile, 76 instances were used for testing. Taken from UCI Machine Learning Website, the data set was already processed. The simulation result for this benchmark data set was summarized in Table 4 and 5. As shown in the table, Halley with DFP was more performed than Halley with BFGS with the highest average accuracy of 74.15 percent at parameter for the learning rate is 0.5 and momentum is 0.3. In last place, Halley with BFGS at parameter learning rate is 0.5 and momentum is 0.5 with accuracy is 70.98 percent. Other than that, Halley with BFGS performs better in the term of CPU time average where Halley with BFGS is faster than Halley with DFP. As Halley with DFP reach 695 epochs at learning rate is 0.5 and momentum is 0.4 as the average from parameter, Halley with BFGS performs with less epochs average that is 118 epochs. This means Halley with BFGS converges to global minima within fewer epochs, faster than Halley with DFP.

Table 4 - The performance of Halley with BFGS on Heart data set

	HALLEY WITH BFGS		
PARAMETER	**ACCURACY (%)**	**CPU TIME**	**EPOCH**
LR= 0.3 , CM=0.3	71.24	8.16	134
LR= 0.3 , CM=0.4	71.16	7.72	118
LR= 0.3 , CM=0.5	71.29	8.40	136
LR= 0.5 , CM=0.3	71.26	8.98	147
LR= 0.5 , CM=0.4	71.29	6.95	120
LR= 0.5 , CM=0.5	**70.98**	**7.00**	**120**
LR= 0.7 , CM=0.3	71.26	7.90	138
LR= 0.7 , CM=0.4	71.22	7.40	129
LR= 0.7 , CM=0.5	71.34	7.64	132

Table 5 - The performance of Halley with DFP on Heart data set

	HALLEY WITH DFP		
PARAMETER	**ACCURACY (%)**	**CPU TIME**	**EPOCH**
LR= 0.3 , CM=0.3	73.38	33.36	538
LR= 0.3 , CM=0.4	73.83	39.30	575
LR= 0.3 , CM=0.5	73.81	36.39	587
LR= 0.5 , CM=0.3	**74.15**	**39.15**	**635**
LR= 0.5 , CM=0.4	74.05	42.03	695
LR= 0.5 , CM=0.5	74.07	34.16	579
LR= 0.7 , CM=0.3	73.40	28.60	465
LR= 0.7 , CM=0.4	73.56	35.94	614
LR= 0.7 , CM=0.5	73.64	41.11	666

4.3 Diabetes Data Set

The last benchmark problem data set is Diabetes data set. Taken from UCI Machine Learning Website, this data set describes that diabetes patient records were obtained from two sources: an automatic electronic recording device and paper records. Therefore, this data set consists of 384 instances where 192 instances were used for testing. As for the attribute, there are 10 attributes where 8 attributes are for input and 2 attributes are for output. Table 7 and 8 summarizes the simulation testing result on Diabetes data set. In this data set from Table 6 and 7, Halley with DFP was more performed than Halley with BFGS with the highest average accuracy of 64.64 percent at parameter for the learning rate is 0.3 and momentum is 0.5. In last place, Halley with BFGS at parameter learning rate is 0.3 and momentum is 0.3 with accuracy is 63.07 percent. Other than that, Halley with DFP performs better in the term of CPU time average where Halley with DFP is faster than Halley with BFGS. As Halley with DFP reach 139 epochs as the average, Halley with BFGS performs with less epochs average that is 112 epochs. This means Halley with BFGS converges to global minima within fewer epochs, faster than Halley with DFP.

Table 6 - The performance of Halley with BFGS on diabetes data set

	HALLEY WITH BFGS		
PARAMETER	**ACCURACY (%)**	**CPU TIME**	**EPOCH**
LR= 0.3 , CM=0.3	**63.07**	**6.78**	**116**
LR= 0.3 , CM=0.4	63.19	5.87	115
LR= 0.3 , CM=0.5	63.24	5.57	117
LR= 0.5 , CM=0.3	63.21	6.26	115
LR= 0.5 , CM=0.4	63.28	6.19	120
LR= 0.5 , CM=0.5	63.13	5.29	119
LR= 0.7 , CM=0.3	62.88	4.49	112
LR= 0.7 , CM=0.4	63.17	4.57	117
LR= 0.7 , CM=0.5	63.10	5.28	114

Table 7 - The performance of Halley with BFGS on heart data set

	HALLEY WITH DFP		
PARAMETER	**ACCURACY (%)**	**CPU TIME**	**EPOCH**
LR= 0.3 , CM=0.3	63.92	7.03	129
LR= 0.3 , CM=0.4	64.15	7.35	132
LR= 0.3 , CM=0.5	**64.64**	**8.32**	**147**
LR= 0.5 , CM=0.3	64.20	7.60	134
LR= 0.5 , CM=0.4	64.00	7.53	132
LR= 0.5 , CM=0.5	64.33	7.76	139
LR= 0.7 , CM=0.3	64.18	7.29	136
LR= 0.7 , CM=0.4	64.33	6.24	139
LR= 0.7 , CM=0.5	64.07	4.61	136

4.4 Analysis on Algorithms Efficiency

Figure 3 and Table 8 summarize the average accuracy for breast cancer data set per each method. In Figure 3, Halley with DFP shows highest accuracy for Breast Cancer is at parameter learning rate is 0.5 and momentum 0.3. While Halley with BFGS at parameter for learning is 0.5 and momentum is 0.5 shows the lowest accuracy in Breast Cancer data set compared to other algorithms. However, Halley with BFGS shows the lowest accuracy for Breast Cancer data set than Halley with DFP.

Table 8 - The summary of accuracy result performance for breast cancer data set

	HALLEY WITH BFGS	**HALLEY WITH DFP**
PARAMETER	**ACCURACY (%)**	**ACCURACY (%)**
LR= 0.3 , CM=0.3	93.06	93.44
LR= 0.3 , CM=0.4	93.18	93.34
LR= 0.3 , CM=0.5	**93.23**	**93.45**
LR= 0.5 , CM=0.3	93.11	93.57
LR= 0.5 , CM=0.4	93.09	93.53
LR= 0.5 , CM=0.5	93.07	93.36
LR= 0.7 , CM=0.3	93.11	93.31
LR= 0.7 , CM=0.4	93.19	93.44
LR= 0.7 , CM=0.5	93.09	93.44

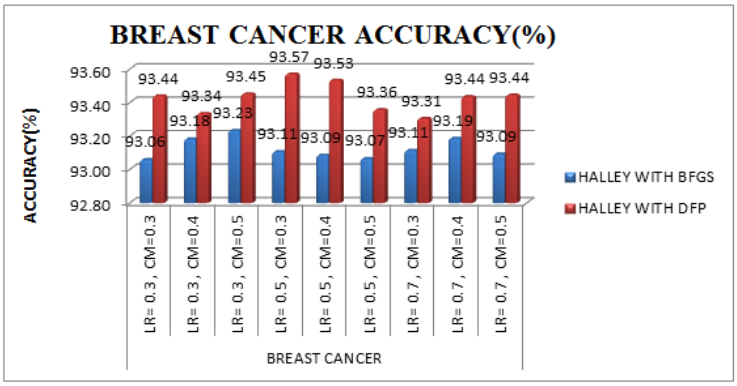

Fig. 3 - The summary of average accuracy for breast cancer data set

Figure 4 and Table 9 summarized the average accuracy for heart data set per each method. In Figure 4, Halley with DFP shows highest accuracy for Heart is at parameter learning rate is 0.5 and momentum 0.3. While Halley with BFGS at parameter for learning is 0.5 and momentum is 0.5 shows the lowest accuracy in Heart data set compared to other algorithms. However, Halley with BFGS shows the lowest accuracy for heart data set than Halley with DFP.

Table 9 - The summary of accuracy for breast cancer data set

PARAMETER	HALLEY WITH BFGS ACCURACY (%)	HALLEY WITH DFP ACCURACY (%)
LR= 0.3 , CM=0.3	71.24	73.38
LR= 0.3 , CM=0.4	71.16	73.83
LR= 0.3 , CM=0.5	71.29	73.81
LR= 0.5 , CM=0.3	71.26	**74.15**
LR= 0.5 , CM=0.4	71.29	74.05
LR= 0.5 , CM=0.5	70.98	74.07
LR= 0.7 , CM=0.3	71.26	73.40
LR= 0.7 , CM=0.4	71.22	73.56
LR= 0.7 , CM=0.5	71.34	73.64

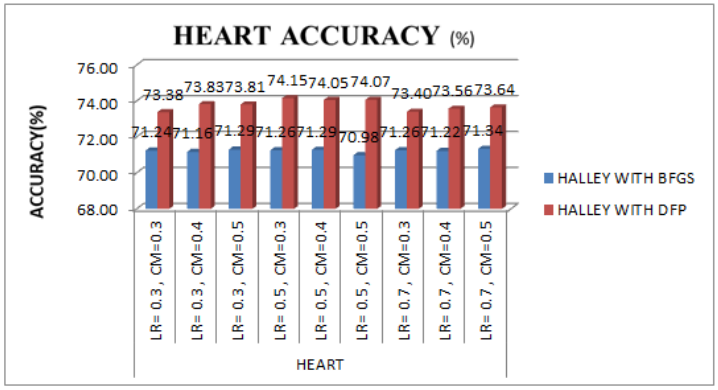

Fig. 4 - The summary of average accuracy heart data set

Figure 5 and Table 10 summarized the average accuracy for diabetes data set per each method. In Figure 5, Halley with DFP shows highest accuracy for diabetes is at parameter learning rate is 0.3 and momentum 0.5. While Halley with BFGS at parameter for learning is 0.7 and momentum is 0.3 shows the lowest accuracy in diabetes data set compared to other algorithms. However, Halley with BFGS shows the lowest accuracy for diabetes data set than Halley with DFP.

Table 10 - The summary of accuracy for diabetes data set

PARAMETER	HALLEY WITH BFGS ACCURACY (%)	HALLEY WITH DFP ACCURACY (%)
LR= 0.3 , CM=0.3	63.07	63.92
LR= 0.3 , CM=0.4	63.19	64.15
LR= 0.3 , CM=0.5	63.24	**64.64**
LR= 0.5 , CM=0.3	63.21	64.20
LR= 0.5 , CM=0.4	63.28	64.00
LR= 0.5 , CM=0.5	63.13	64.33
LR= 0.7 , CM=0.3	62.88	64.18
LR= 0.7 , CM=0.4	63.17	64.33
LR= 0.7 , CM=0.5	63.10	64.07

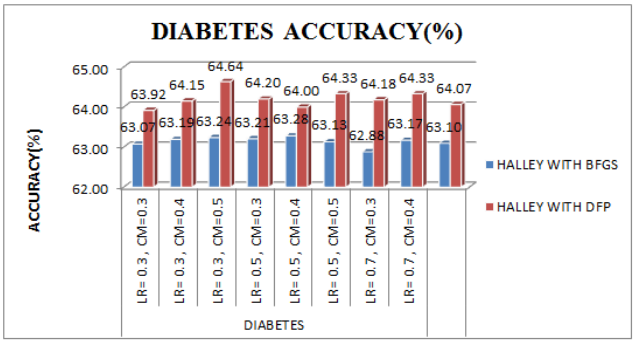

Fig. 5 - The summary of average accuracy diabetes data set

According to these figure shows that Halley with DFP shows the higher accuracy for three data sets which were Diabetes, Heart and Breast Cancer data sets. For heart and breast cancer are higher at parameter for learning is 0.5 and momentum 0.3. Meanwhile, for diabetes at parameter for learning is 0.3 and momentum is 0.5 shows the lowest accuracy. Besides that, Halley with BFGS shows the lowest accuracy for three data sets which were Diabetes, Heart and Breast Cancer data sets. For heart and breast cancer are lower at parameter for learning is 0.5 and momentum is 0.5. Meanwhile, for diabetes at parameter for learning is 0.7 and momentum is 0.3 shows the lowest accuracy.

Figure 6 and Table 11 shows the summary of CPU time average for heart data sets. It shows that CPU time for Halley with BFGS is a lot faster than other algorithms in breast cancer data set at parameter for learning is 0.3 and momentum 0.3. Meanwhile, the slowest CPU time in each data set was Halley with DFP at parameter for learning is 0.5 and momentum 0.3.

Table 11 - The summary of CPU for breast cancer data set

PARAMETER	HALLEY WITH BFGS CPU TIME	HALLEY WITH DFP CPU TIME
LR= 0.3 , CM=0.3	**2.92**	3.10
LR= 0.3 , CM=0.4	3.00	3.18
LR= 0.3 , CM=0.5	3.02	3.17
LR= 0.5 , CM=0.3	3.00	3.53
LR= 0.5 , CM=0.4	2.98	3.30
LR= 0.5 , CM=0.5	3.04	3.17
LR= 0.7 , CM=0.3	3.08	3.07
LR= 0.7 , CM=0.4	3.04	3.13
LR= 0.7 , CM=0.5	2.96	3.14

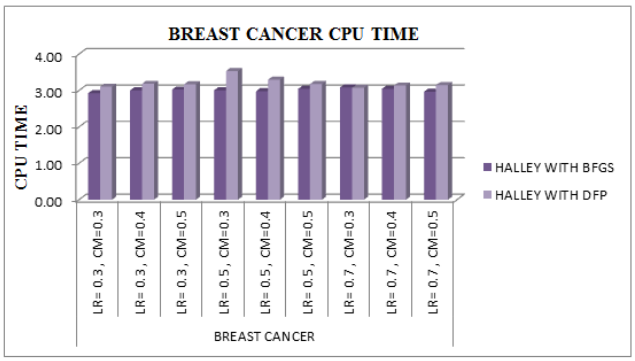

Fig. 6 - The summary of CPU time for breast cancer data set

Figure 7 and Table 12 shows the summary of CPU time average for heart data sets. It shows that CPU time for Halley with BFGS is a lot faster than other algorithms in breast cancer data set at parameter for learning is 0.5 and momentum 0.4. Meanwhile, the slowest CPU time in each data set was Halley with DFP at parameter for learning is 0.5 and momentum 0.4.

Table 12 - The summary of CPU time for heart data set

PARAMETER	HALLEY WITH BFGS CPU TIME	HALLEY WITH DFP CPU TIME
LR= 0.3 , CM=0.3	8.16	33.36
LR= 0.3 , CM=0.4	7.72	39.30
LR= 0.3 , CM=0.5	8.40	36.39
LR= 0.5 , CM=0.3	8.98	39.15
LR= 0.5 , CM=0.4	**6.95**	42.03
LR= 0.5 , CM=0.5	7.00	34.16
LR= 0.7 , CM=0.3	7.90	28.60
LR= 0.7 , CM=0.4	7.40	35.94
LR= 0.7 , CM=0.5	7.64	41.11

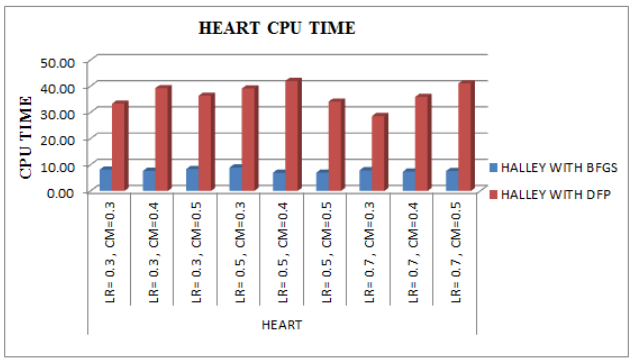

Figure 7 - The summary of CPU time heart for data set

Figure 8 and Table 13 shows the summary of CPU time average for heart data sets. It shows that CPU time for Halley with BFGS is a lot faster than other algorithms in breast cancer data set at parameter for learning is 0.7 and momentum 0.5. Meanwhile, the slowest CPU time in each data set was Halley with DFP at parameter for learning is 0.3 and momentum 0.5.

Table 13 - The summary of CPU time for diabetes data set

	HALLEY WITH BFGS	HALLEY WITH DFP
PARAMETER	**CPU TIME**	**CPU TIME**
LR= 0.3 , CM=0.3	6.78	7.03
LR= 0.3 , CM=0.4	5.87	7.35
LR= 0.3 , CM=0.5	5.57	**8.32**
LR= 0.5 , CM=0.3	6.26	7.60
LR= 0.5 , CM=0.4	6.19	7.53
LR= 0.5 , CM=0.5	5.29	7.76
LR= 0.7 , CM=0.3	**4.49**	7.29
LR= 0.7 , CM=0.4	4.57	6.24
LR= 0.7 , CM=0.5	5.28	4.61

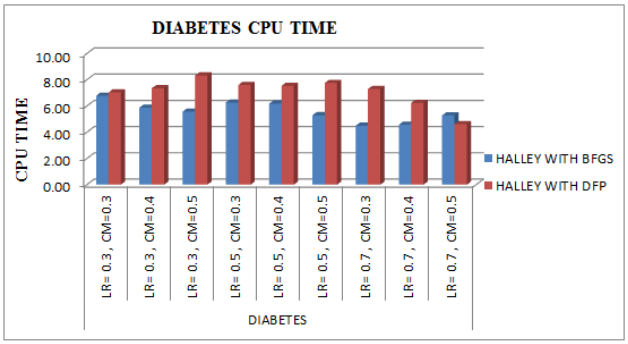

Fig. 8 - The summary of CPU time for diabetes data set

It shows that CPU time for Halley with BFGS is a lot faster than other algorithms in every data set than Halley with DFP. Average size data sets Breast Cancer data set each algorithm needs less CPU time to complete the simulation than Heart and Diabetes. Figure 9 and Table 14 illustrates the summary of epochs where in breast cancer data sets, Halley with BFGS and Halley with DFP converge to global minima just within lesser epoch in this data set.

Table 14 - The summary of epoch for breast cancer data set

	HALLEY WITH BFGS	HALLEY WITH DFP
PARAMETER	**EPOCH**	**EPOCH**
LR= 0.3 , CM=0.3	**51**	57
LR= 0.3 , CM=0.4	**52**	56
LR= 0.3 , CM=0.5	**53**	57
LR= 0.5 , CM=0.3	**51**	63
LR= 0.5 , CM=0.4	**52**	60
LR= 0.5 , CM=0.5	**53**	58
LR= 0.7 , CM=0.3	**55**	55
LR= 0.7 , CM=0.4	**54**	55
LR= 0.7 , CM=0.5	**52**	56

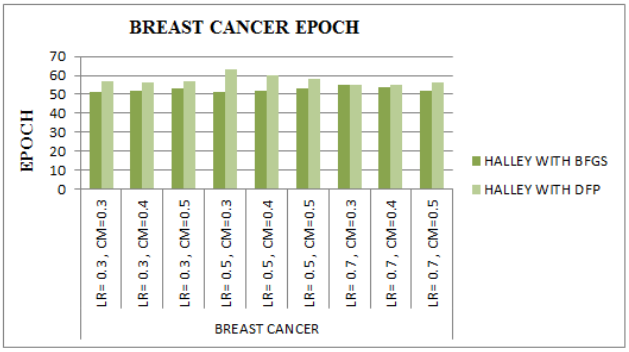

Fig. 9 - The summary of epoch breast for cancer data set

Figure 10 and Table 15 illustrates the summary of epochs where in heart data sets, Halley with BFGS converge to global minima just within lesser epoch in this data set. Halley with DFP almost reaches for maximum value of epochs.

Table 15 - The summary of epoch for heart data set

	HALLEY WITH BFGS	HALLEY WITH DFP
PARAMETER	**EPOCH**	**EPOCH**
LR= 0.3 , CM=0.3	**134**	538
LR= 0.3 , CM=0.4	**118**	575
LR= 0.3 , CM=0.5	**136**	587
LR= 0.5 , CM=0.3	**147**	635
LR= 0.5 , CM=0.4	**120**	695
LR= 0.5 , CM=0.5	**120**	579
LR= 0.7 , CM=0.3	**138**	465
LR= 0.7 , CM=0.4	**129**	614
LR= 0.7 , CM=0.5	**132**	666

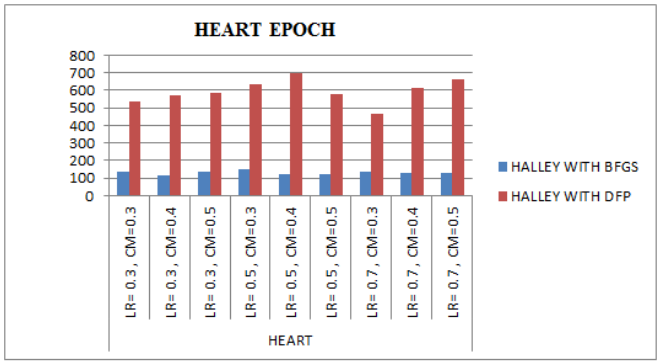

Fig. 10 - The summary of epoch for heart data set

Figure 11 and Table 16 illustrates the summary of epochs where in diabetes data sets, Halley with BFGS and Halley with DFP converge to global minima just within lesser epoch in this data set.

Table 16- The summary of epoch for diabetes data set

	HALLEY WITH BFGS	HALLEY WITH DFP
PARAMETER	**EPOCH**	**EPOCH**
LR= 0.3 , CM=0.3	**116**	129
LR= 0.3 , CM=0.4	**115**	132
LR= 0.3 , CM=0.5	**117**	147
LR= 0.5 , CM=0.3	**115**	134
LR= 0.5 , CM=0.4	**120**	132
LR= 0.5 , CM=0.5	**119**	139
LR= 0.7 , CM=0.3	**112**	136
LR= 0.7 , CM=0.4	**117**	139
LR= 0.7 , CM=0.5	**114**	136

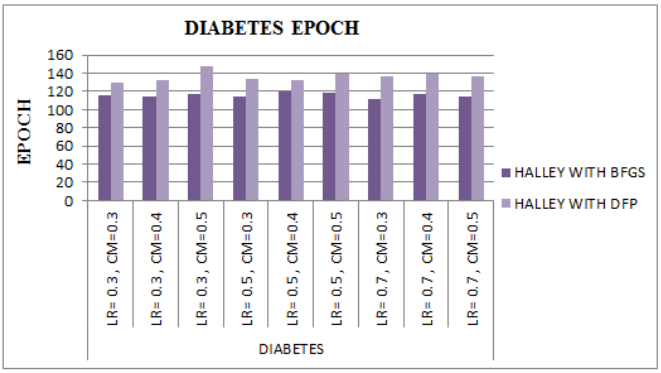

Fig. 11 - The summary of epoch for diabetes data set

5. Conclusion

The limitation of BP algorithm has been improved in this research by implementing third order method. Furthermore, this research also shows that the performance of the network depends on the choice of proper parameters such as learning rate and momentum value. This research proposed the effect of hyper-parameters on the performance of third order neural networks. The performance of third order neural network had been analyzed by changing the learning rate and momentum value for all nodes in the learning process. The performance of Halley with BFGS and Halley with Davidon-Fletcher Powell (DFP) had been evaluated by testing on three benchmark medical data sets. The result of simulation shows that by changing the hyper-parameters value in Halley with DFP, it performs better as compared to others it is considered as the best approach for medical classification problems like Heart, Breast Cancer, and Diabetes.

Acknowledgement

The Authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) and Faculty of Computer Science and Information Technology (FSKTM) for supporting this Research.

References

[1] A. Shakeri Abdolmaleki, A. Gholamalizadeh Ahangar, and J. Soltani, “Artificial Neural Network (ANN) Approach for Predicting Cu Concentration in Drinking Water of Chahnimeh1 Reservoir in Sistan-Balochistan, Iran,” *Heal. Scope*, vol. 2, no. 1, pp. 31–38, 2013

[2] A. Shakeri Abdolmaleki, A. Gholamalizadeh Ahangar, and J. Soltani, “Artificial Neural Network (ANN) approach for predicting Cu concentration in drinking water of Chahnimeh1 Reservoir in Sistan-Balochistan,” Iran. *Health scope*, vol. 2, no. (1), pp. 31-38, 2013

[3] M.W. Gardnera and S.R Dorlinga, "Artificial Neural Networks (The Multilayer Perceptron) - A Review of Application in the Atmospheric Science," *Atmospheric Sciences*, vol. 32, no. 14-15, pp. 2627-2636, 1998

[4] *K. Rama, and K. Taranjit, "Backpropagation Algorithm: An Artificial Neural Network Approach for Pattern Recognition," International Journal of Scientific. Engineering Research, vol. 3, no. 6, pp. 2229-5518. 2012*

[5] I. Ernest, and M. Tony, "Improved Backpropagation Learning in Neural Networks with Windowed Momentum," In International Journal of Neural System, vol. 12, no. 3&4, pp. 303-318, 2002

[6] Medical classification, 2017. https://en.wikipedia.org/wiki/Medical_classification

[7] Rumelhart D. E., Hinton G. E., and Williams R. J., (1986). Learning Internal Representations by Error Propagation: in D.E. Rumelhart and J.L. McClelland (eds), Parallel Distributed Processing, vol. 1, p. 318-362.

[8] Yu, C. C., & Liu, B. D. (2002). A backpropagation algorithm with adaptive learning rate and momentum coefficient. In Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on (Vol. 2, pp. 1218-1223). IEEE

[9] Fletcher, R. and M.J.D. Powell, A Rapidly Convergent Descent Method for Minimization. The Computer Journal, 1963. 6(2): p. 163-168

[10] Fletcher, R. and C.M. Reeves, Function minimization by conjugate gradients. The Computer Journal, 1964. 7(2): p. 149-154

[11] Rivaie, M., et al., A new class of nonlinear conjugate gradient coefficients with global convergence properties. Applied Mathematics and Computation, 2012. 218(22): p. 11323-11332