International Journal on Advanced Science Engineering Information Technology

Vol.7 (2017) No. 4-2
ISSN: 2088-5334

Optimal Parameter Selection Using Three-term Back Propagation Algorithm for Data Classification

Nazri Mohd Nawi[#], Nurmahiran Muhammad Zaidi[#], Noorhamreeza Abdul Hamid[#], Muhammad Zubair Rehman[*], Azizul Azhar Ramli[#], Shahreen Kasim[#]

[#]*Soft Computing and Data Mining Centre, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400, Johor, Malaysia*
E-mail: nazri@uthm.edu.my

[*]*Department of Computer Science and Information Technology, University of Lahore, Islamabad Campus, Pakistan*

Abstract **— The back propagation (BP) algorithm is the most popular supervised learning method for multi-layered feed forward Neural Network. It has been successfully deployed in numerous practical problems and disciplines. Regardless of its popularity, BP is still known for some major drawbacks such as easily getting stuck in local minima and slow convergence; since it uses Gradient Descent (GD) method to learn the network. Over the years, many improved modifications of the BP learning algorithm have been made by researchers, but the local minima problem remains unresolved. Therefore, to resolve the inherent problems of BP algorithm, this paper proposed BPGD-A3T algorithm where the approach introduces three adaptive parameters which are gain, momentum and learning rate in BP. The performance of the proposed BPGD-A3T algorithm is then compared with BPGD two-term parameters (BPGD-2T), BP with adaptive gain (BPGD-AG) and conventional BP algorithm (BPGD) by means of simulations on classification datasets. The simulation results show that the proposed BPGD-A3T showed better performance and performed the highest accuracy for all dataset as compared to other.**

Keywords**— back propagation; local minima; three-term; gradient descent; local minima; search direction; classification**

I. INTRODUCTION

Artificial Neural Network (ANN) is based on the model of a human brain. ANN is composed of several neurons that act as processors which are interconnected by weighted links which are updated to obtain required outputs [1]. It uses a mathematical model for information processing which is based on the approach of computation inspired by the structure and operation of biological neurons organized into layers. Basically, there are three layers in a neural network which are Input layer, Hidden layer, and Output layer. The most common and established neural network model is the multilayer perceptron (MLP). This type of neural network is known as a supervised network since it requires the desired output in order to make sure that the network learns. The main purpose of this type of network is to create a model that correctly maps the input to the output using historical or unseen data so that the model can then be used to produce the output when the desired output is unknown. The Back propagation (BP) algorithm is very popular for supervised learning method such as multi-layered feed forward Neural Network. It is commonly used for learning algorithm for training Neural Network. In back propagation, the input data is repeatedly presented to the neural network and for every iteration of the training process, each presentation the output of the neural network is compared to the desired output in order to compute the error. The error is then fed back (back propagated) to the neural network and been used to adjust the weights such that the error decreases with each iteration. As a result, the neural model gets closer and closer to producing the desired output. This algorithm uses a gradient descent (GD) method which known to minimize the error of the network by moving down the gradient of the error curve. Furthermore, the weights of the network are adjusted by the algorithm for every iteration. Consequently, the error is reduced along a descent direction.

Recently, Artificial Neural Network (ANN) technology has gained much attention and been improved by many researchers. Some researchers had proposed some modifications to the conventional BP algorithm in order to improve the performance of Multilayer Perceptrons network training. The most simple and significant improvement of BP is by focusing on the development of ad hoc techniques [2]-[9].

In which proposed techniques some of the researchers introduced the momentum term, others used the alternative cost function or dynamic adaptation of the learning parameters. Many apply special techniques of initialization of weights.

Later, Nazri et al. [10] proved that by adaptively changing the 'gain' value for each node can reduce the training time without modifying the network topology. This is due to the effect of 'gain' parameter in reducing the steps needed to reach the minimum error. Therefore, this research takes a further step by proposing an improvement on [10] by adjusting activation function of neurons in the hidden layer in each training set. Moreover, the activation functions are adjusted by combining gain parameters together with adaptive momentum and adaptive learning rate value during the learning process. The proposed algorithm, known as, (BPGD-A3T), presents better convergence rate and can avoid the network from trapping into local minima. The performance of the proposed algorithm will be compared with the conventional BP algorithm (BPGD), back propagation gradient descent with adaptive gain (BPGD-AG), back propagation gradient descent with adaptive momentum (BPGD-AM) and back propagation gradient descent with adaptive Learning Rate (BPGD-ALR). The simulation was run were performed on five classification dataset which are glass dataset, card dataset, diabetes dataset, heart dataset and horse dataset.

The remaining of the paper is organized as follows. Section II, explains the basic operation of the back propagation algorithm and the proposed algorithm. The simulation results are discussed in Section III. This paper is concluded in the final section.

II. Material and Method

The Back Propagation (BP) algorithm is a well-known technique used in the implementation of artificial neural networks. The establishment of BP algorithm had gained attention to many researchers and been implemented in diverse disciplines and applications. The best part of BP algorithm is that it always looks for the minimum of the error function in weight space using the method of gradient descent. In which the combination of weights which minimizes the error function is considered to be a solution to the learning problem. Since this method requires computation of the gradient of the error function at each iteration step, therefore it must guarantee the continuity and differentiability of the error function. There are many activation functions that can be used, and among them, one of the most popular activation functions for back propagation networks is the sigmoid, a real function S_c :IR→ (0, 1) is defined by the expression.

$$S_c(x) = \frac{1}{1+e^{-cx}} \quad (1)$$

A differentiable activation function makes the function computed by a neural network differentiable since the network itself computes only function compositions. The error function also becomes differentiable. Furthermore, since the sigmoid always has a positive derivative, the slope of the error function provides a greater or lesser descent direction which can be followed. In most cases, local minima appear because the targets for the outputs of the computing units are values other than 0 or 1. Moreover, if a network for the computation of XOR is trained to produce 0.9 at the inputs (0,1) and (1,0), then the surface of the error function develops some protuberances, where local minima can arise. Whereas, in the case of binary target values, some local minima are also present, as shown by Lisboa and Perantonis [11] who analytically found all local minima of the XOR function. In fact, the network model represents a chain of function compositions which transform an input to an output vector. The network is a particular implementation of a composite function from input to output space, which called network function. The learning problem consists of finding the optimal combination of weights so that the network function α approximates a given function f as closely as possible [11].

The learning rate (LR) is one of the crucial factors to accelerate the convergence of BP learning and control the variable of the neuron weight adjustments at each iteration during the training process. The convergence speed is dependence on the choice of LR. The algorithm will take a longer time to converge or may never converge if the LR is too small. However, the network will accelerate the convergence rate significantly and still possibly will cause the instability if the LR value is too high. The value of LR usually set to be constant for all weights in the whole learning process. By adding some momentum coefficient (MC) to the network, it will speed up the convergence, stabilize the training procedure and avoid the local minima. Basically, the MC is set to be constant in the interval [0,1] because it was discovered from simulations that the fixed momentum coefficient value could only speed up learning 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 MC may cause the weight to be altered up the slope of the error surface as opposed to down the slope as preferred [12]. This leads to the emergence of diverse schemes for adjusting the MC value adaptively instead of being kept constant throughout the training process [13-14].

Yu and Liu [15], proposed a back propagation algorithm with adaptive learning rate and momentum. They modified the conventional back propagation algorithm by using adaptive learning rate and momentum where the learning rate and the momentum are adjusted at each iteration to speed up the training time. The modified back-propagation with adaptive learning rate and momentum outperforms the conventional back propagation with fixed momentum or without momentum in term of learning speed. Shamsuddin et al., [16] have improved the convergence rates of two-term BP model with some modification in learning strategies. The experiment results show that the modified two-term BP improved with a convergence rate much better when compared with standard BP. Iranmanesh and Mahdavi [17] proposed a differential adaptive learning rate method for BP to speed up the learning rate.

A few researchers also introduced optimization method by introducing Particle Swarm Optimization and Random walk algorithm with BP [18]-[19]. However, the calculation for

finding an optimum solution was so complex and cause extra overhead. That is why this paper only focuses on parameters such as momentum, learning rate and activation function for improving BP.

Moreover, the proposed method does not cause any extra or additional overhead since the employs of the large learning rate at the beginning of training gradually decreases the value of learning rate using the differential adaptive method. By considering the advantages of each three-term parameters to the BP performance, we believe that by combining all three parameters together the performance of BP algorithm will be further improved and faster to converge.

Therefore, this paper proposed algorithm BPGD-3T that modifies the BP algorithm with three adaptive terms which are gain, momentum coefficient, and learning rate. The advantages of using an adaptive gain value together with momentum coefficient and learning rate have been investigated. Gain update such as weight and bias update implemented for output and hidden nodes have also been explored. The iterative algorithm is proposed for the batch mode of training. For the all training set which is being presented to the network, the weights, biases, gains, momentum coefficients and learning rates are calculated and updated [17].

The pseudo code of the proposed algorithm is discussed as below:

```
Start
For a given epoch,
        For each input vector,
Step 1  Calculate the weight and bias values
        using the previously converged gain,
        momentum coefficient and learning
        rate values
Step 2  Use the weight and bias value
        calculated in Step (1) to calculate
        the new gain, momentum coefficient
        and learning rate values

        Repeat Steps (1) and (2) for each
        input vector and sum all the
        weights, biases, momentum
        coefficient, learning rate and gain
        updating terms.

        Update the weights, biases, gains,
        momentum coefficients and learning
        rates using the summed updating
        terms and repeat this procedure on
        epoch-by-epoch basis until the error
        on the entire training data set
        reduces to a predefined value.
End
```

III. Results and Discussion

The simulations were carried out using MATLAB software on five classification datasets taken from UCI machine learning repository. Those five datasets are; glass dataset, card dataset, diabetes dataset, heart dataset and horse dataset. The following algorithms are analysed and simulated on the datasets:

- Back Propagation Gradient Descent (BPGD)
- Back Propagation Gradient Descent with Adaptive Gain (BPGD-AG)
- Back Propagation Gradient Descent with Adaptive Gain and Adaptive Momentum (BPGD-AM)
- Back Propagation Gradient Descent with Adaptive Gain and Adaptive Learning Rate (BPGD-ALR)
- Back Propagation Gradient Descent with three terms parameters, Adaptive Gain, Adaptive Momentum and Adaptive Learning Rate (BPGD-A3T)

Three-layer back-propagation neural networks are used to test the models. The hidden layers are keeping constant to 5 hidden nodes while output and input layers nodes are different according to the datasets given and sigmoid activation function was used for all nodes. The maximum iteration for each problem is set to 5000 epochs, and 30 trials are run for each dataset. For each trial, the results are stored in the result file meanwhile CPU time and accuracy are recorded for each trial on every dataset.

For all training for the conventional BPGD algorithm, the initial value for momentum coefficient and learning rate is fixed generated. Furthermore, for all training for BPGD-AG, the initial value for momentum coefficient and learning rate is fixed generated. The initial value used for the gain parameter for BPGD-AG, BPGD-AM, BPGD-ALR and BPGD-3T algorithms is set to 1. For all training for BPGD-AM, BPGD-ALR and BPGD-A3T algorithms, as the gain, momentum coefficient and learning rate value were modified and the weight and biases were updated using the new value of gain, momentum coefficient, and learning rate. The initial value for momentum coefficient is fixed, and learning rate of BPGD-AM and BPGD-ALR algorithms is randomly generated. The initial value for momentum coefficient and learning rate of BPGD-A3T algorithms is randomly generated. The target error is set to 0.01.

A. Glass Dataset

This dataset was collected by B. German on fragments of glass encountered in forensic work. The glass dataset is used for separating glass splinters into six classes, namely float processed building windows, non-float processed building windows, vehicle windows, containers, tableware, or headlamps [20]. The selected architecture of the network is 9-5-6 with target error was set to 0.01, and the maximum epoch was 5000. The best momentum coefficient and learning rate value for conventional BPGD and BPGD-AG for the glass dataset are 0.2 and 0.4 respectively. For the BPGD-AM, the best momentum coefficient and learning rate value are found in the interval [0.1,0.2] and 0.4 respectively while the best momentum coefficient and learning rate value for BPGD-ALR are found in the interval 0.2 and [0.3,0.4] respectively. Meanwhile, for the proposed BPGD-A3T, the best momentum coefficient and learning rate value are found in the interval [0.1,0.2] and [0.3,0.4] respectively.

TABLE I
EPOCH, SD, CPU TIME AND ACCURACY FOR GLASS DATASET

Glass Classification Dataset					
	BPGD	**BPGD-AG**	**BPGD-AM**	**BPGD-ALR**	**BPGD-A3T**
Epoch	2312	2095	2071	2022	**1997**
SD	108.72	85.99	82.25	50.01	**46.99**
CPU Time	47.01	19.74	19.44	18.95	**17.87**
Accuracy	75.02	79.11	79.36	81.01	**81.08**

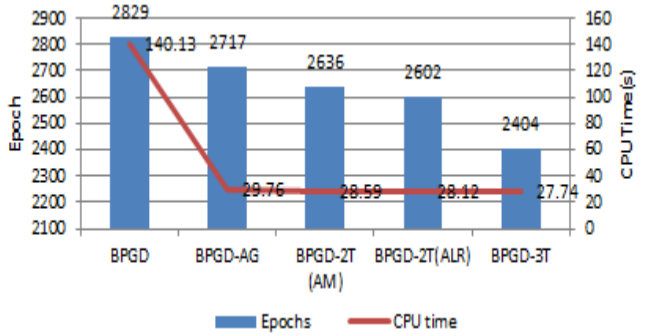

Fig. 1 Performance comparison of BPGD-A3T with BPGD-AM, BPGD-ALR, BPGD-AG and conventional BPGD on glass dataset

Table 1 and Fig. 1 shows that the proposed algorithm (BPGD-A3T) gives the best performance. Furthermore, the accuracy of the proposed algorithm is better with 81.08% as compared to BPGD-ALR, BPGD-2T AM, BPGD-AG and BPGD which are 81.01%, 79.36%, 79.11% and 75.02% respectively. Moreover, the proposed algorithm (BPGD-A3T) needs 1997 epochs to converge opposed to the conventional BPGD at about 2312 epochs, BPGD-ALR at about 2022 epochs, BPGD-AM at about 2071 epochs while BPGD-AG needs 2095 epochs to converge. The time required for the training the classification dataset is an important factor when analysing the performance. The result clearly shows that the proposed algorithm (BPGD-A3T) have the best total time of converging as compared to conventional BPGD, BPGD-AG, BPGD-AM, and BPGD-ALR.

B. Card Dataset

This dataset was predicted the approval or non-approval of a credit card to a customer [21]. Descriptions of each attribute name and values were not enclosed for confidentiality. The selected architecture of the network is 51-5-2 with target error was set to 0.01, and the maximum epoch was 5000. The best momentum coefficient and learning rate value for conventional BPGD and BPGD-AG for the glass dataset are 0.5 and 0.3 respectively. For the BPGD-AM, the best momentum coefficient and learning rate value are found in the interval [0.3,0.8] and 0.3 respectively while the best momentum coefficient and learning rate value for BPGD-ALR are found in the interval 0.5 and [0.2,0.3] respectively. The best momentum coefficient BPGD-A3T found in the range [0.3, 0.8] and [0.2,0.3] for learning rate value.

TABLE II
EPOCH, SD, CPU TIME AND ACCURACY FOR CARD DATASET

Card Classification Dataset					
	BPGD	**BPGD-AG**	**BPGD-AM**	**BPGD-ALR**	**BPGD-A3T**
Epoch	1175	1243	1176	1041	**970**
SD	871.84	969.20	636.02	343.57	**190.95**
CPU Time	44.98	13.08	12.44	11.91	**10.39**
Accuracy	90.94	92.39	93.00	90.60	**93.21**

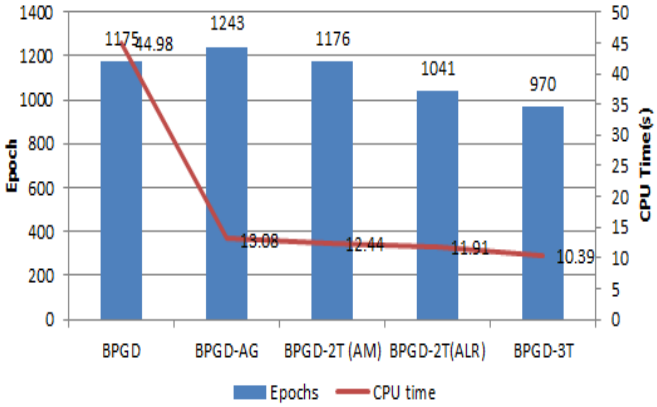

Fig. 2 Performance comparison of BPGD-A3T with BPGD-AM, BPGD-ALR, BPGD-AG and conventional BPGD on card dataset

Table 2 and Fig. 2 shows that BPGD needs 44.98 seconds with 1175 epochs to converge, whereas BPGD-AG needs 13.08 seconds with 1243 epochs to converge, BPGD-AM needs 12.44 seconds with 1176 epochs to converge, and BPGD-ALR needs 11.91 seconds with 1041 epochs to converge. Conversely, the proposed algorithm (BPGD-A3T) performed better, and it only needs 10.39 seconds with 970 epochs to converge. Furthermore, the accuracy of the proposed algorithm is better with 93.21% as compared to BPGD-ALR, BPGD-AM, BPGD-AG, and BPGD with 90.60%, 93.00%, 92.39% and 90.94% respectively.

C. Diabetes Dataset

This dataset that was selected from a larger data set held by the National Institutes of Diabetes and Digestive and Kidney Diseases. The constraint of this dataset are all the patients are Prima-Indian women, at least 21 years old and must be living near Pheonix, Arizona, USA [22]. The selected network topology for Diabetes classification dataset is 8-5-2, with 8 input nodes, 5 hidden nodes, and 2 output nodes. 384 instances were represented as training dataset and 192 as a testing dataset. The target error was set to 0.01, and the maximum epoch was 5000. The best momentum coefficient and learning rate value for conventional BPGD and BPGD-AG for the glass dataset are 0.3 and 0.3 respectively. For the BPGD-AM, the best momentum coefficient and learning rate value are found in the interval

[0.3,0.9] and 0.3 respectively while the best momentum coefficient and learning rate value for BPGD-ALR are found in the interval 0.3 and [0.3,0.4] respectively. Meanwhile, for the proposed algorithm (BPGD-A3T), the best momentum coefficient and learning rate value are found in the interval [0.3,0.9] and [0.3,0.4] respectively.

TABLE III
EPOCH, SD, CPU TIME AND ACCURACY FOR DIABETES DATASET.

Diabetes Classification Dataset					
	BPGD	**BPGD-AG**	**BPGD-AM**	**BPGD-ALR**	**BPGD-A3T**
Epoch	3965	3755	3272	2593	**2036**
SD	1225.3	1161.15	1685.42	1455.03	**1368.09**
CPU Time	61.69	44.08	42.34	36.72	**19.88**
Accuracy	69.01	70.76	72.14	74.45	**77.59**

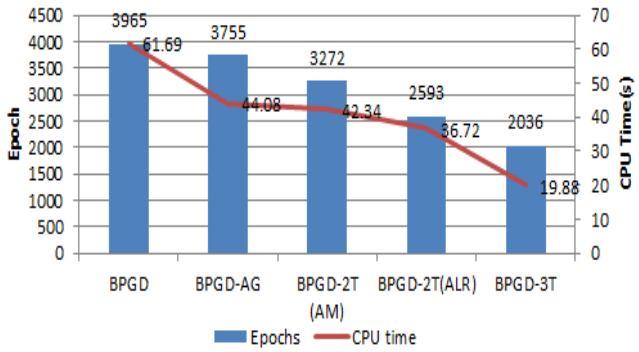

Fig. 3 Performance comparison of BPGD-A3T with BPGD-AM, BPGD-ALR, BPGD-AG and conventional BPGD on diabetes dataset

Table 3 and Fig. 3 shows that the proposed algorithm (BPGD-A3T) still outperforms other algorithms in terms CPU time and number of epochs .The proposed algorithm (BPGD-A3T) epochs need only 2036 to converge as opposed to the conventional BPGD at about 3965 epochs, BPGD-AG needs 3755 epochs to converge while BPGD-AM at about 3272 epochs and BPGD-ALR needs 2593 epochs to converge. Moreover, the time required for training the classification dataset is an important factor when analyzing the performance. The result clearly shows that the proposed algorithm (BPGD-A3T) have the better performance for a total time of converge. Furthermore, the accuracy of BPGD-A3T is much better than BPGD, BPGD-AG, BPGD-AM, and BPGD-ALR.

D. Heart Dataset

The selected architecture of the network is 36-5-2 with target error was set to 0.01, and the maximum epoch was 5000. The best momentum coefficient and learning rate value for conventional BPGD and BPGD-AG for the glass dataset are 0.7 and 0.3 respectively. For the BPGD-AM ,the best momentum coefficient and learning rate value are found in the interval [0.5,0.7] and 0.3 respectively while the best momentum coefficient and learning rate value for BPGD-ALR are found in the interval 0.7 and [0.2,0.3] respectively. Meanwhile, for the proposed BPGD-A3T, the best momentum coefficient and learning rate value are found in the interval [0.5,0.7] and [0.2, 0.3] respectively.

TABLE IV
EPOCH, SD, CPU TIME AND ACCURACY FOR HEART DATASET

Heart Classification Dataset					
	BPGD	**BPGD-AG**	**BPGD-AM**	**BPGD-ALR**	**BPGD-A3T**
Epoch	1702	1691	1502	1438	**1717**
SD	203.85	317.62	470.54	518.25	**148.63**
CPU Time	76.08	18.70	18.05	18.04	**15.63**
Accuracy	88.58	88.78	90.38	90.66	**90.90**

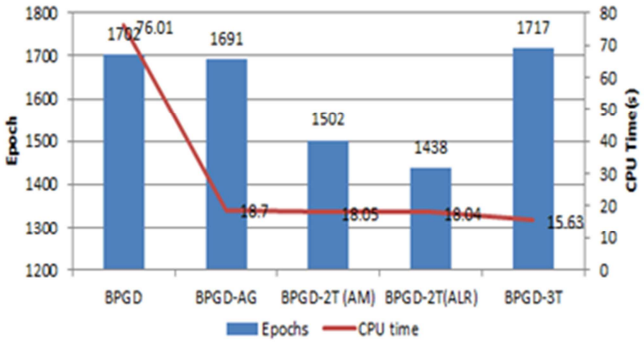

Fig. 4 Performance comparison of BPGD-A3T with BPGD-AM, BPGD-ALR, BPGD-AG and conventional BPGD on Heart dataset

Table 4 and Fig. 4 shows that the proposed algorithm (BPGD-A3T) deliver the best performance. Furthermore, the accuracy of the proposed algorithm is better with 90.90 % as compared to BPGD-ALR, BPGD-AM, BPGD-AG, and BPGD which are 90.66%, 90.38%, 88.76% and 88.58% respectively. Moreover, the proposed algorithm (BPGD-A3T) needs 1717 epochs to converge opposed to the conventional BPGD at about 1702 epochs, BPGD-AG needs 1691 epochs to converge while BPGD-ALR at about 1502 epochs, BPGD-AM at about 1438 epochs. Apart from the speed of convergence, the time required for the training the classification dataset is an important factor when analysing the performance. The results clearly show that the proposed algorithm (BPGD-A3T) have the best total time of converging compared to outperforms conventional BPGD, BPGD-AG, BPGD-AM , and BPGD-ALR.

E. Horse Dataset

The selected architecture of the network is 58-5-2 with target error was set to 0.01, and the maximum epoch was 5000. The best momentum coefficient and learning rate value for conventional BPGD and BPGD-AG for the glass dataset are 0.5 and 0.4 respectively. For the BPGD-AM, the best momentum coefficient and learning rate value are found in the interval [0.5,0.6] and 0.4 respectively while the best

momentum coefficient and learning rate value for BPGD-ALR are found in the interval 0.5 and [0.3,0.4] respectively. The momentum coefficient value for BPGD-A3T is found in the range [0.5,0.6] for momentum coefficient and [0.3,0.4] for learning rate value.

TABLE V
EPOCH, CPU TIME AND ACCURACY FOR HORSE DATASET

Horse Classification Dataset					
	BPGD	**BPGD-AG**	**BPGD-AM**	**BPGD-ALR**	**BPGD-A3T**
Epoch	2829	2717	2636	2602	**2404**
SD	573.13	481.17	737.42	461.83	**582.34**
CPU Time	140.13	29.76	28.59	28.16	**27.74**
Accuracy	79.37	79.64	79.91	80.86	**80.99**

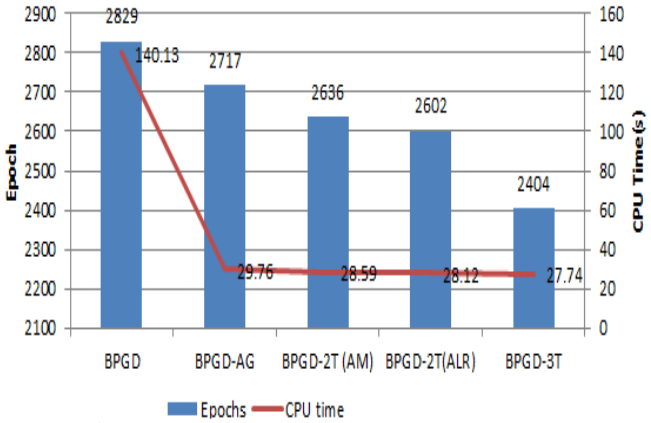

Fig. 5 Performance comparison of BPGD-A3T with BPGD-AM, BPGD-ALR, BPGD-AG and conventional BPGD on Horse dataset

Table 5 shows that the proposed algorithm required 2404 epochs with 27.74 seconds CPU times to achieve the target error by 80.99%. Whereas BPGD-ALR required 2602 epochs with 28.12 seconds CPU times with 80.86% accuracy while BPGD-AM required 2636 epochs with 28.60 seconds CPU times with 79.91% accuracy. At the same time, BPGD-AG required 2717 epochs with 29.76 seconds CPU times with 79.64% accuracy, and BPGD required 2829 epochs with 140.13 seconds CPU times with 79.37 % accuracy. Fig. 5 shows that the proposed algorithm (BPGD-A3T) still outperformed other algorithms in terms of the number of epochs, CPU time and accuracy.

IV. CONCLUSION

As a popular and most widely used algorithm, Back Propagation (BP) Neural Network is known to be able to train Artificial Neural Networks (ANN) successfully. However, BP algorithms have some drawbacks which are getting stuck in local minima, and slow convergence rate and this algorithm still need some improvement. In this paper, the BPGD-A3T algorithm is proposed to train BPNN in order to achieve fast convergence, avoid local minima and enhance accuracy. The proposed algorithm adaptively changes the gain parameter of the activation function together with momentum and learning rate to overcome the inherent problems of BP. The performance of the BPGD-A3T algorithm is then compared with the BPGD-AM, BPGD-ALR, BPGD-AG and conventional BP algorithm. The performance of the proposed BPGD-A3T is verified by means of simulations on Glass classification dataset, Card classification dataset, Diabetes classification dataset, Heart classification dataset and Horse classification dataset are used respectively. The simulation results show that the proposed BPGD-A3T showed better performance and performed the highest accuracy for all dataset compared to other algorithms.

ACKNOWLEDGMENT

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) Ministry of Higher Education (MOHE) Malaysia for financially supporting this Research under Trans-disciplinary Research Grant Scheme (TRGS) vote no. T003. This research also supported by GATES IT Solution Sdn. Bhd under its publication scheme.

REFERENCES

[1] Krasnopolsky, V.M. Neural Network Applications to Developing Hybrid Atmospheric and Oceanic Numeric Model. In: Haupt, S.E., Pasini, A. and Marzban, C. (Ed.). Artificial Intelligence Methods in the Environmental Science. New York City: Springer. pp. 217 – 234; 2009.

[2] Rumelhart D. E., Hinton G. E., and Williams R. J., Learning internal representations by back-propagation errors. Parallel Distributed Processing, 1986. 1 (Rumelhart D.E. et al. Eds.): p. 318-362.

[3] Sotiropoulos D.G., Kostopoulos A.E., and G. T.N., A spectral version of Perry's Conjugate gradient method for neural network training. 4th GRACM Congress on Computational Mechanics (GRACM 2002), 2002. 1: p. 291-298.

[4] Fahlman S.E., Faster learning variations of back propagation: An empirical study. D. Touretzky, G.E. Hinton and T.J. Sejnowski (editors) Proceedings of the1988 Connectionist Models Summer School, 1988: p. 38-51.

[5] Jacobs R.A., Increased rates of convergence through learning rate adaptation. Neural Networks, 1988. 1: p. 295–307.

[6] Kamarthi S. V. and Pitner S., Accelerating Neural Network Training using Weight Extrapolations. Neural Networks, 1999. 12: p. 1285-1299.

[7] Leonard J. and Kramer M. A., Improvement to the backpropagation algorithm for training neural networks. Computer and Chemical Engineering, 1990. 14(3): p. 337-341.

[8] Looney C. G., Stabilization and Speedup of Convergence in Training Feed Forward Neural Networks. Neurocomputing, 1996. 10(1): p. 7-31.

[9] Perantonis S. J. and Karras D. A., An Efficient Constrained Learning Algorithm with Momentum Acceleration. Neural Networks, 1995. 8(2): p. 237-249.

[10] Nazri, M. N., Ransing, R. S. and Ransing, M. S., An Improved Conjugate Gradient Based Learning Algorithm for Back Propagation Neural Networks. International Journal of Information and Mathematical Sciences. 4(1): p. 46-55. (2008).

[11] Back Propagation Algorithm Theory [Online]. Available:http://page.mi.fuberlin.de/rojas/neural/chapter/K7.pdf. [Accessed: May. 3, 2015].

[12] Nazri Mohd Nawi, R.S. Ransing, Mohd Najib Mohd Salleh, Rozaida Ghazali, Norhamreeza Abdul Hamid :"The Effect of Gain Variation in Improving Learning Speed of Back propagation Neural Network Algorithmon Classification Problems". Proceeding in Symposium on Progress in Information and Communication Technologies (SPICT'09), Malaysia, 7-8 December 2009.

[13] N. A. Hamid, N. M. Nawi and R. Ghazali, (2011). "The Effect of Adaptive Gain and Adaptive Momentum in Improving Training Time of Gradient Descent Back Propagation Algorithm on Classification Problems," Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011. Putrajaya, Malaysia.

[14] Nazri Mohd Nawi, Mohd Najib Mohd Salleh, Rozaida Ghazali, Norhamreeza Abdul Hamid. "Learning Efficiency Improvement of Back Propagation Algorithm by Adaptively Changing Gain Parameter together with Momentum and Learning Rate", Proceeding in International Conference on Mathematical and Computational Biology 2011 (ICMCB 2011), Renaissance Melaka Hotel, Malacca 12-14 April 2011.

[15] Yu, C.-C., and Liu, B.-D. (2002). A backpropagation algorithm with adaptive learning rate and momentum coefficient .Proceedings of the International Joint Conference on Neural Networks, May 12-17, IEEE Xplore Press, Honolulu, HI, pp: 1218-1223.

[16] Shamsuddin, S. M., Sulaiman, M. N., and Darus, M. (2001). An improved error signal for the backpropagation model for classification problems. International Journal of Computer Mathematics, 76(3), 297-305.

[17] Iranmanesh, S., and Mahdavi, M. A. (2009). A differential adaptive learning rate method for back-propagation neural networks. World Academy of Science, Engineering and Technology, 38, 289-292.

[18] Choon Sen Seah, Shahreen Kasim, Mohd Saberi Mohamad: Specific Tuning Parameter for Directed Random Walk Algorithm Cancer Classification. International Journal on Advanced Science, Engineering and Information Technology, Vol. 7 (2017) No. 1, pages: 176-182

[19] Moaath Shatnawi, Mohammad Faidzul Nasrudin, Shahnorbanun Sahran: A new initialization technique in polar coordinates for Particle Swarm Optimization and Polar PSO. International Journal on Advanced Science, Engineering and Information Technology, Vol. 7 (2017) No. 1, pages: 242-249

[20] Evett, I. W. and Spiehler, E. J., Rule induction in forensic science, in Knowledge Based Systems. Halsted Press. p. 152-160. (1988).

[21] Quinlan, J. R.: Simplifying Decision Trees". J.Man Machine Studies. vol. 27, pp. 221--234. (1987).

[22] Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, In: The Symposium on Computer Applications and Medical Care, IEEE Computer Society Press, pp. 261--265. (1988).