2013 IEEE 11th Malaysia International Conference on Communications
26th - 28th November 2013, Kuala Lumpur, Malaysia

Dynamic Training Rate for Backpropagation Learning Algorithm

M.S Al_ Duais , A. R Yaakub, N.Yusoff
School of Computing, College of Arts and Sciences, University Utara Malaysia, 06010
UUM Sintok, Kedah, Malaysia
Neroz2025@yahoo.com, ary321@uum.edu.my, nooraini@uum.edu.my

***Abstract* - In this paper, we created a dynamic function training rate for the Back propagation learning algorithm to avoid the local minimum and to speed up training. The Back propagation with dynamic training rate (BPDR) algorithm uses the sigmoid function. The 2-dimensional XOR problem and iris data were used as benchmarks to test the effects of the dynamic training rate formulated in this paper. The results of these experiments demonstrate that the BPDR algorithm is advantageous with regards to both generalization performance and training speed. The stop training or limited error was determined by1.0** e^{-5}

***Keywords* - Artificial neural networks, Back propagation algorithm, adaptive training, dynamic training rate.**

I. Introduction

The standard back propagation (SBP) algorithm is popularly used in neural network training with multi-layer neural networks.[1]. It has been widely regarded as one of the most efficient training algorithms for multi–layer perception [2] -[3]. Gradient descent is widely used to adjust weights through the change of E. However, the gradient descent is not guaranteed to find the global minimum error because it may result in approaching the local minimum [4].

The main drawback of the SBP algorithm is slowing down training as it often takes along time to learn and get the desired results. However, the network can get stuck at a local minimum when, O_r the output training of the output layer, approach the extremes of 1 or 0 [5].

Many recent studies have attempted to solve the slow training required for the SBP algorithm through adaptation of parameters such as the training rate, which is controlled by the weight adjustment along with the descent direction [6]. Gong[7] proposed a novel algorithm of the neural network NBPNN based on a self-adaptive learning factor. Those algorithms were tested on XOR 2-bit or two dimensionally. Simulation results have shown that the proposed NBPNN helps the Back propagation algorithm avoid the local minimum and reduce the training time. Latifi and Amirii[8] presented a novel method based on the adaptation of the variable step size learning rate method to increase the convergence speed of the EBP algorithm. The proposed algorithm convergence is faster than the standard EBP algorithm. Iranmanesh and Mahdavi[9] have proposed different training rates for different locations with regards to the output layer training rate. Zhixiim and Bingqing[10] have proposed modifying the training rate by mathematical formulas based on two – step functions. Li et al[11] improved the convergence of the standard back propagation algorithm based on a mathematical formula of the training rate. The simulation results showed that the iteration time is significantly less than the SBP algorithm. Yang and Xu[12] proposed a new algorithm of back propagation that involved adapting the training rate. The new formula of the training rate helps the SBP algorithm to reduce the training time.

The remainder of this paper is organized as follows: Section II is a presentation of the neural networks (NNs) model; Section III Creating the dynamic training rate; Section IV is a presentation of the dynamic BPDR algorithm Section V is an implementation of the algorithms with XOR; Section VI is an implementation of algorithms with iris data; Section VII covers the conclusion of this study.

II. Neural Network Model

In this study, we propose an ANN model, which consists of a Multi-layer neural network composed of an input layer, hidden layer, and output layer. The input layer is the { x_1, $x_2, \dots, x_i$ } node. The hidden layers consist of two layers with four nodes. The output layer consists of one layer with one node., we will denoted by u_{0j}, v_{0j} and w_{0r} ,see figure 1. The sigmoid function is employed as an activation function, which describes the linear of the output layer [13]. The proposed neural network can be defined as { I , T ,W, A},where we denote the set of input nodes by I, T denotes the topology of NN, which covers the number of hidden layers and the number of neurons, W denotes the set of the weights and A denotes the activation function. The model appears as Figure 1.

Before presenting the BPDR algorithm, let us briefydefine s ome of the notation used in the algorithm as

follows:

x_i	Input layer for neuron i
z_h	First hidden layer for neuron h
zzj	Second hidden layer for neuron j
Or	Output layer for neuron r
u_{ih}	The weight between the neuron I from input layer and neuron h from first hidden layer
U_{0h}	The weight of bias for neuron j

V_{hj} The weight between the neuron j from hidden layer Z and neuron from hidden layer ZZ

V_{0J} The weight of Bias for neuron j

Wjr The weight between the neuron j from hidden layer ZZ and neuron r from output layer

W_{0r} The weight of bias for neuron r from output layer

Δw The difference between the current and new value in the next iteration

η_{dmic} The dynamic training rate

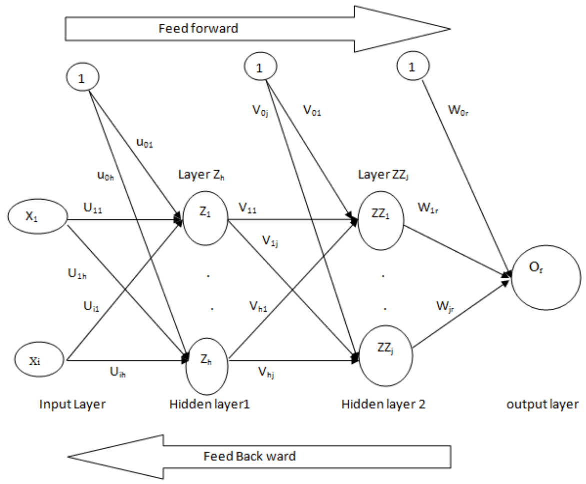

Fig. 1 Training of Back Propagation .

III. CREATING THE DYNAMIC TRAINING RATE η_{dmic}

One way of escaping the local minimum and speeding up the training of the SBP algorithm is by using a large value of training rate η in the first training. On the contrary, the small value of η leads to slow training [14]. In the SBP algorithm, the training rate is selected based on experience and a trial value of between (0, 1) is used[15]. In spite of this, there are many studies that have proposed different techniques for increasing the value of η to speed up BP through the creation of the dynamic function. But, if the increasing value of η becomes too large, it leads to oscillated output training [16]. Even with the smallest value or a large value of η, it differs from the training BP algorithm. [10] have proposed a novel BP neural network (NBPNN) based on the self- adaptive learning factor that was defined by a formula which depended on an exponential function. The weight update between the neuron k from the output layer and neuron j from the hidden layer is as follows.

$$\Delta \mathrm{w}_{\mathrm{jk}}(\mathrm{new}) = \mathrm{w}_{jk}(old) + (-\eta)\Delta \mathrm{w}_{\mathrm{jk}} \tag{1}$$

The weight is updated from equation (1) as slow training or fast training, which is dependent on some parameter which they alter to update the weight. To enhance the SBP algorithm, which is given by equation (1), in order to avoid the local minimum, or remove the saturation training that is occurring through the adaptive training rate by the dynamic function. The key to speeding up the BP algorithm is the monotonicity of the error function during training for every epoch or iteration [17]. Many studies used the adaptive training rate by adopting the monotonicity function such as [7] - [18] used the exponential to increase the speed of the BP algorithm. Based on the discussion above, the new formula of the training rate is proposed as follows.

$$\eta_{dmic}(e) = 1 + \mathrm{e}^{(1+\tan e)} \tag{2}$$

The main idea of this formula is make e error training the implicit function in the η_{dmic} .The error training starting with a big value in the beginner training will eventually result in the dynamic training rate becoming a big value in the beginner training and then the value of the e will decay with the index epoch. This step helps the BP algorithm escape the local minimum and remove the saturation training.

IV. DYNAMIC BACK PROPAGATION (BPDR)ALGORITHM

The heuristic technique is a significant method for increasing the training BP algorithm. The heuristic technique includes some parameters such as training rate and momentum term. The Training algorithm of BPDR involves three stages. They are feed forward, backward and update the weight. All steps are illustrated as follows.

Forward Pass Phase

In the forward pass phase, it just calculates the data layer by layer until the end out-layer in the system.

Step 0: Read and initialize the weight.

Step 1: For each training pair, do steps 2-22.

Step 2: Read the number of the neuron in the hidden layers.

Step 3: Read the pattern from the XOR problem and iris data, obtain the target, and limit the error

Backward Pass Phase

This step starts when the feed forward reaches the end step and then the start feedback, it is obvious in the figure (1). The goal of the BP is to get the minimum error training [2] as equation (3)

Step 4: Calculate the error training:

$$e_r = \sum_{r=1}^{n} (t_r - o_r) \tag{3}$$

Step 5 :Calculate the local gradient for the output o_r

$$\delta_r = e_r f'(\mathrm{o}_{\_inr}) \;,\; f'(\mathrm{o}_{\_inr}) = \mathrm{o}_{\_inr}(1 - \mathrm{o}_{\_inr}) \tag{4}$$

Step 6: Calculate weight correction term (used to updatethe newest w_{jr})

$$\Delta \mathrm{w}_{\mathrm{jr}} = -(1 + \mathrm{e}^{(1+\tan(e))})\, \delta_r zz_j \tag{5}$$

Step 7: Calculate, bias correction term (used to update the newest w_{0r} later)

$$\Delta w_{0r} = -(\ 1+\ e^{(1+\tan(e)})\delta_r \quad (6)$$

And send δ_r to hidden layer $\left(zz_j\ ,\ \ j=1,\ldots,p\right)$

Step 8 :Calculate the weighted input for layer above to get

$$\delta_{-inj} = \sum_{r=1}^{m} \delta_r w_{jr} \quad (7)$$

Step 9: Calculate the local gradient for the hidden layer (zz_j) to get

$$\delta_j = \delta_{-\text{inj}} f'\left(zz_{-\text{inj}}\right) \quad (8)$$

Step 10: Calculate the weight correction term (used to update the newest v_{hj} later).

$$\Delta v_{hj} = -(1+\ e^{(1+\tan(e)})\delta_j z_h \quad (9)$$

Step 11: Calculate the bias collection term (used to update the newest v_{0j} later).

$$\Delta v_{oj} = -(1+\ e^{(1+\tan(e)})\delta_j \quad (10)$$

And send δ_j to hidden layer $\left(z_h \quad h=1,\ldots,\ a\right)$

Step 12: Sum the weighted input from units in the layer above get:

$$\delta_{-\text{inh}} = \sum_{j=1}^{b} \delta_j v_{hl} \quad (11)$$

Step 13 :Calculate the local gradient of hidden layer z_h (expressed in terms of x_i

$$\delta_h = \delta_{-\text{inh}} f'\left(z_{-\text{inh}}\right)\ ,\quad f'\left(z_{-inh}\right) = z_{-inh}\left(1 - z_{-inh}\right) \quad (12)$$

Step 14: Calculate the weight correction (used to update the newest u_{ih} later):

$$\Delta u_{ih} = -(1+ e^{(1+\tan(e)})\delta_h x_i \quad (13)$$

Step 15 : Calculates bias weight corrective term (used to update the newest u_{0h} later)

$$\Delta u_{ih} = -(1+ e^{(1+\tan(e)})\delta_h \quad (14)$$

Update weight Phase :

The weight update for each layer according to the dynamic training rate which was created in equation 2 is as follows.

Step 16 :Update the weight for each output layer o_r

$j = 0,1,2,\ldots p; r = 1,\ldots,m$

$$W_{jr}\left(t+1\right) = w_{jr}(t) + (1+ e^{(1+\tan(e)})\delta_r zz_j \quad (15)$$

Step 17 : Update the weight for bias w_{or}

$$W_{0r}\left(t+1\right) = w_{0r}(t) + (1+ e^{(1+\tan(e)})\delta_r \quad (16)$$

Step 18 : Update the weight for each hidden layer

$(ZZ_j \qquad h= 0,\ldots,q\ ; j=1,\ldots\ p)$

$$v_{hj}\left(t+1\right) = v_{hj}(t) + (1+ e^{(1+\tan(e)})\delta_j z_h \quad (17)$$

Step 19 : Update the weight for bias v_{0j}

$$v_{0j}(t+1) = v_{0j}(t) + (1+ e^{(1+\tan(e)})\delta_j \quad (18)$$

Step 20:Update the weight for each hidden layer z_h

$z_h(\ i=o,\ldots,n\ ;h=1,\ldots,q)$

$$u_{ih}\left(t+1\right) = u_{ih}(t) + (1+ e^{(1+\tan(e)})\delta_h x_i \quad (19)$$

Step 21 : Update the weight for bias u_{0h}

$$u_{0h}\left(t+1\right) = u_{0h}(t) + (1+ e^{(1+\tan(e)})\ \delta_h \quad (20)$$

Step 22 :Calculate the mean square error

$$MSE = 0.5\frac{1}{p}\sum_{p=1}^{n}\sum_{k}^{i}(t_{kp} - o_{kp})^2$$

Step 23: Test the conditional

V. IMPLEMENTATION OF THE ALGORITHMS ON XOR

we using XOR -2Bit parity as a benchmark and by simulation we verified our proposal.

A. Experimental test of BPDR Algorithm

10 experiments has been done the result in the table (I).

TABLE I. SPEED UP BPDR ALGORITHM AT LIMITED ERROR 0.0001

Average time-Sec	Average MSE	Average epoch
3.5864	1.00E-04	2320

From table (I) above, the formula that is proposed in equation (2) helps the back propagation algorithm to reduce the training time. Whereas t= 3.5864 seconds, the average value of MSE performances is a very small value for each epoch training. The training curve is shown in figure (2).

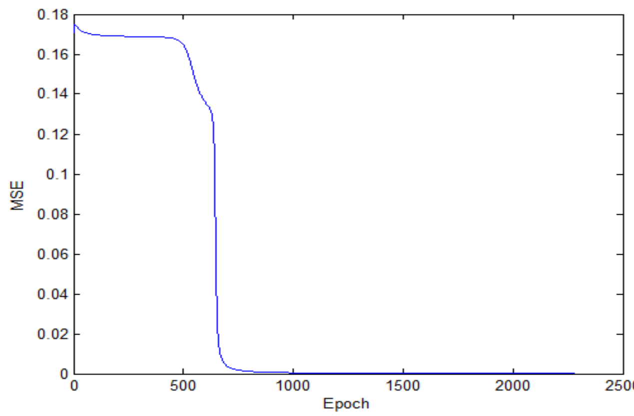

Fig. 2 Training curve of the BPDR algorithm.

From the curve above, we can evidently see that the training curve starts with a flat spot for the first 400 epochs and then decays with the inverse of the learning epoch index.

B. Experiments of the SBPAlgorithm

In this part, 10 experiment has been done The results of the experiments are tabulated in table (II).

TABLE II. SPEED UP THE TRAINING OF THE SBP ALGORITHM AT0.0001

Value of η	Time	MSE	Epoch
0.1	74.3860	9.9999e-05	66505
0.2	46.7180	1.0000e-04	43627
0.3	31.0700	9.9998e-05	28875
0.4	23.6240	9.9994e-05	21772
0.5	20.5870	9.9998e-05	17557
0.6	15.9230	9.9998e-05	14764
0.7	13.3000	9.9998e-05	12777
0.8	11.5270	9.9988e-05	11292
0.9	10.0960	9.9995e-05	10139
1	9.5800	1.0000e-04	9219

From table (II), the best performance of the SBP algorithm is achieved at η=1, whereas the time for training is 9.5800 seconds with regards to the minimum training. On the contrary, the worst performance of the training time of the SBP algorithm is achieved at η=0.1, whereas the training time is 74.3860 seconds with regards to the maximum training. The range of training time is $9.5800 \leq t \leq 74.3860$seconds. The training curve is shown below

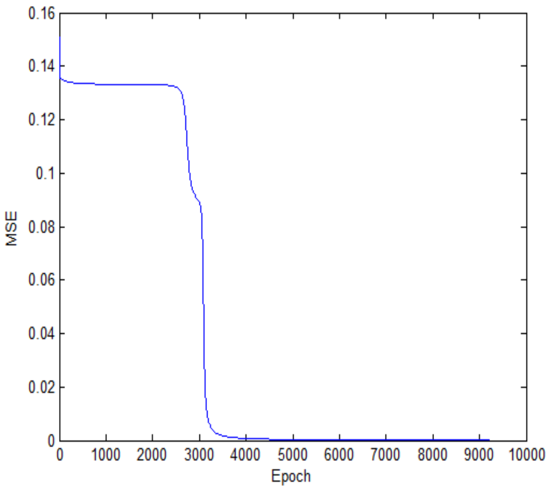

Fig. 3Training curve of the SBP algorithm .

It is obvious that the curve for training starts flat for the first 3000 epochs, and then decays with the inverse of the learning epoch index.

C. Speed up Training BPDR versusSBPAlgorithm

In this section, we compare the performance of the BPDR algorithm and the SBP algorithm to discover which provides the better training. The result is tabulated in table (III).

TABLE III. SPEED UP OF BPDR VERSUS SBP ALGORITHM AT 0.0001

Algorithm	Time –Sec	MSE	Epoch	Value of η
BP DR	3.5864	1.00E-04	2320	
SBP	74.3860	1.9999e-5	66505	0.1
	46.7180	1. e04	43627	0.2
	31.0700	1.9998e-5	28875	0.3
	23.6240	9.9994e-5	21772	0.4
	20.5870	9.9998e-5	17557	0.5
	15.9230	9.9998e-5	14764	0.6
	13.3000	9.9998e-5	12777	0.7
	11.5270	9.9988e-5	11292	0.8
	10.0960	9.9995e-5	10139	0.9
	9.5800	1e-4	9219	1

The BPDR algorithm is 20.741 $\approx$21 times faster than the SBP algorithm at the maximum training time. In addition , the BPDR algorithm is 3 times faster than the SBP algorithm at the minimum training time.

VI. IMPLEMENTATION OF BPDR ALGORITHM ON IRIS DATA

The dataset has 150 patterns. We divided the iris data into two sets, a training set that consists of 60% of the data, we consider the other 40% of the data as the testing set.

A. Implementation of BPDR algorithm on Training set

1) An Experimental BPDR algorithm for Training

We tested the performance of the dynamic training that was created by equation (2). The results of these experiments are tabulated in table (IV).

TABLE IV. SPEEDUP OF BPDR ALGORITHM IRIS –TRAININGSET

Averag		Average Epoch	Average MSE
Time –sec	S.D		
2.7986	0.453249	329	9.98E-6

The value of time is very small, and this indicates that the dynamic of the training rate that was proposed helps the BP algorithm to speed up training. The curve for training is shown as follows:

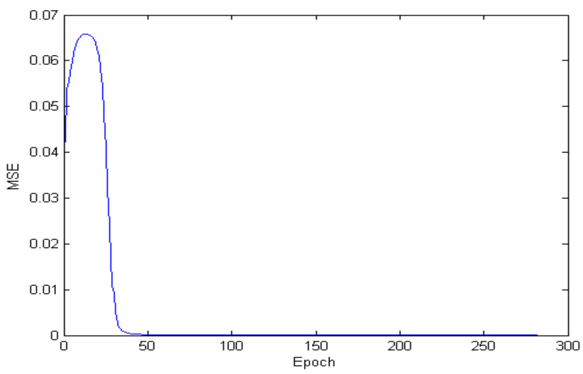

Fig. 4 Training Curve of the BPDR algorithm .

2) AnExperimental SBP algorithm for trainingset

we test the performance of the SBP algorithm, which is given in equation (1). 100 experiments have been done. The results of the experiments are tabulated in table (V).

TABLE V. SPEED UP OF SBP ALGORITHM FORTRAINING-SET

Value of η	Average		Average Epoch	Average MSE
	Time-	S.D		
0.1	202.33	48.58	10029	1.0E-05
0.2	100.05	37.56	5204	1.0E-05
0.3	71.34	19.59	3620	1.0E-05
.04	39.32	7.95	2167	1.0E-05
0.5	35.19	12.33	6628	3.7E-05
0.6	36.76	12.72	1598	3.7E-05
0.7	35.92	10.31	1289	9. 9E-06
0.8	35.40	10.45	1272	1.0E-05
0.9	23.85	7.32	952	9.9E-06
1	24.98	8.32	991	9.9E-06

From the table above the best performance of the SBP algorithm is achieved at training rate = 0.9 whereas the average time is 23.85 seconds as the minimum training time. On the contrary, the worst performance of the SBP algorithm is achieved at training rate=0.1, whereas the average time training is 202.33 seconds at the maximum time.

3) Speeding up the BPDR versus the SBP algorithm

We compare the BPDR algorithm and SBP algorithm to discover which gives the superior training time as follows.

TABLE VI. SPEED UP BPDR VERSUS SBP ALGORITHM FOR TRAINING SET

Algorithms	Average		Average Epoch	Average MSE	Value of η
	Time	S.D			
BP DR	2.7986	0.453249	329	9.98E-6	
SBP	202.33	48.58	10029	1.0E-05	0.1
	100.05	37.56	5204	1.0E-05	0.2
	71.34	19.59	3620	1.0E-05	0.3
	39.32	7.95	2167	1.0E-05	0.4
	35.19	12.33	6628	3.7E-05	0.5
	36.76	12.72	1598	3.7E-05	o.6
	35.92	10.31	1289	9. 9E-06	0.7
	35.40	10.45	1272	1.0E-05	0.8
	23.85	7.32	952	9.9E-06	0.9
	24.98	8.32	991	9.9E-06	1

We can easily see that the BPDR algorithm gives superior training compared to the SBP algorithm, whereas the BPDR algorithm is 72.29686 $\approx$ 72times faster than the SBP algorithm at the maximum training time. In addition, the BPDR algorithm is 8.5221 $\approx$ 9 times faster than the SBP algorithm at the minimum training time. On other hand the S.D for BPDR algorithm is very smaller than S.D of SBP algorithm this is indicated the BPDR algorithm more robust than SBP algorithm

B. IMPLEMENTAtion of BPDRalgorithm on Testing set

1) An experimental BPDR algorithm

There were 60 patterns used as a testing set as a benchmark to test the performance of the equation (1). 10 experiments have been done The experiment result is written down as below.

TABLE VII. SPEED UP OF BPDR ALGORITHM FORIRIS - TEST SET

Average		Averg Epoch	Average MSE
Time -Src	S.D		
4.51	0.457138	432	9.99E-06

The BPDR algorithm reaches 9.99E-06 after spending 4.51 seconds with 432 epoch ,which is a very short time. The curves for training are shown below:

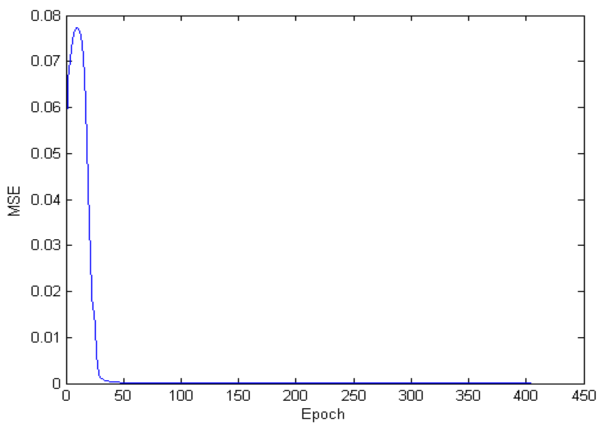

Fig. 5 Training curve of the BPDR .

2) An Experimental SBP algorithm on the testing set

we test the performance of the SBP algorithm , which is given in equation (1), by using 60 patterns.100 experiments have been done. The experiments are shown in the table VIII

TABLE VIII. SPEED UP OF SBP ALGORITHM FOR TESTING SET

Value of η	Averag		Average Epoch	Average MSE
	Time	S.D		
0.1	187.32	45.7	14324	1.0E-05
0.2	92.17	35.26	7722	1.0E-05
0.3	55.46	13.32	4675	1.0E-05
0.4	41.03	7.78	3414	1.0E-05
0.5	44.95	10.98	2941	1.0E-05
0.6	32.16	12.17	2092	1.0E-05
0.7	27.65	7.05	1930	1.0E-05
0.8	30.75	8.89	1904	1.0E-05
0.9	30.19	6.53	1660	1.0E-05
1	25.81	7.32	1412	1.0E-05

The best performance of the SBP algorithm is achieved at η =1, whereas the time training is 25.81 seconds at the minimum training time. On the contrary, the worst performance of the SBP algorithm is achieved at η=0.1, whereas the training time is 187.32 seconds at the maximum training time. The range of the training times is 25.81 $\leq$ t $\leq$ 187.32 seconds.

3) Speed up BPDR Versus SBP algorithm

We compare the BPDR algorithm and SBP algorithm for iris training -set to discover which gives the superior training time as follows .

TABLE IX. PEED UP OF BPDR VERSUS SBP ALGORITHM FOR TESTING

Algorithms	Time –Sec		Average Epoch	Average MSE	Value of η
	Average	S.D			
BP DR	4.51	0.457138	432	9.99E-06	
SBP	187.32	45.7	14324	1. E-05	0.1
	92.17	35.26	7722	1. E-05	0.2
	55.46	13.32	4675	1. E-05	0.3
	41.03	7.78	3414	10E-05	0.4
	44.95	10.98	2941	1.E-05	0.5
	32.16	12.17	2092	1. E-05	o.6
	27.65	7.05	1930	1 E-05	0.7
	30.75	8.89	1904	1 E-05	0.8
	30.19	6.53	1660	1.E-05	0.9
	25.81	7.32	1412	1.E-05	1

The BPDR algorithm was given superior training compared with the SBP algorithm, whereas the BPDR algorithm is 41.5343≈42 times faster than the SBP algorithm at the maximum training time. In addition, the BPDR algorithm is 5.7228 ≈6 times faster than the SBP algorithm at the minimum training time.

VII. CONCLUSION

The back propagation BP algorithm suffers from slow training.To overcome this problem, this study creating dynamic training rate. This study introduced the BPDR algorithm which is training by dynamic training rate. The dynamic training rate affected the weight for each hidden layer and output layer and eliminated the saturation training in the BPDR algorithm. One of the main advantages of the dynamic training is that it reduces the training time and reduces the error training and number of epochs.The experiments results have shown that BPDR algorithm gave a superior performance with regard straining time compared with the SBP algorithm. However, the BPDR algorithm is 21 times faster than the SBP algorithm at the maximum training time. In addition, the BPDR algorithm is 3 times faster than the SBP algorithm at the minimum training time . For the iris data training set, the BPDR algorithm is 72 time faster than the SBP algorithm at the maximum training time. In addition, the BPDR algorithm is 9 times faster than the SBP algorithm at the minimum training time. For the iris data testing set, the BPDR algorithm is 42 faster than the SBP algorithm at the maximum training time. In addition, the BPDR algorithm is 6 times faster than the SBP algorithm at the minimum training time at a limited error of $1e^{-5}$

ACKNOWLEDGMENT

I would like to thank to binti Abdullah for her comments

REFERENCES

[1] M. Asaduzzaman, M. Shahjahan, and K. Murase, “Faster training using fusion of activation functions for feed forward neural networks.,” International journal of neural systems, vol. 19, no. 6, pp. 437–48, Dec. 2009

[2] Y. Fukuoka, H. Matsuki, H. Minamitani, and Akimasa Ishida, “A modified back-propagation method to avoid false local minima.,” Neural networks□: the official journal of the International Neural Network *Society*, vol. 11, no. 6, pp. 1059–1072, Aug. 1998.

[3] H. R. Maier and G. C. Dandy, “Understanding the behavior and optimizing the performance of back-propagation neural networks□: an empirical study,” vol. 13, pp. 179–191, 1998.

[4]F. Han, T. Gu, and S. Ju, “An Improved Hybrid Algorithm Based on PSO and BP for Feed forward Neural Networks,” International Journal of Digital Content Technology and itsApplicationsvol. 5, no. 2, pp. 106–115, February2011.

[5]C. Cheung, S. Ng, S. Member, A. K. Lui, and S. S. Xu, “Enhanced Two-Phase Method in Fast Learning Algorithms.” International Joint Conference on Neural Networks , Barcelona,2010, pp 18-23

[6] S. J. Abdulkadir, S. M. Shamsuddin, and R. Sallehuddin, “Three Term Back Propagation Network for Moisture Prediction,” International Conference on Clean and Green Energy, 2012.pp103-107

[7] B. Gong, “A Novel Learning Algorithm of Back-Propagation Neural Network,” IITA International Conference on Control, Automation and Systems Engineering , 2009, pp. 411–414 .

[8] N. Latifi and A. Amiri, “A Novel VSS-EBP Algorithm Based on Adaptive Variable Learning Rate,” Third International Conference on Computational Intelligence, Modelling & Simulation, 2011 , pp. 14–17.

[9] S. Iranmanesh and M. A. Mahdavi, “A Differential Adaptive Learning Rate Method for Back-Propagation Neural Networks,”. World Academy of Science, Engineering and Technology.,v0l.50,no 1 , pp285–288 , 2009.

[10] S. ZhIxIm and L. Bingqing “Research of Improved Back-propagation Neural□! Network Algorithm” IEEE International Conference on Communication Technology, 2020, pp763-766.

[11]J. Li, L. Lian, and Y. Jiang, “An Accelerating Method of Training Neural Networks Based on Vector Epsilon Algorithm,” Third International Conference on Information and Computing, , 2010, pp. 292–295.

[12] C. Yang and R. Xu, “Adaptation Learning Rate Algorithm of Feed-Forward Neural Networks,” International Conference on Information Engineering and Computer Science, 2009, pp. 1–3.

[13]Y. Bassil, “Neural Network Model for Path-Planning of Robotic Rover Systems,” Journal of Science and Technology vol. 2, no. 2, pp. 94–100,,. February 2012.

[14]J. Ge, J. Sha, and Y. Fang, “An new back propagation algorithm with chaotic learning rate,” IEEE International Conference on Software Engineering and Service Sciences, 2010 ,pp. 404–407.

[15]W. Jin-wei, “The improvements of BP neural network learning algorithm,”. 5th International Conference on Signal Processing Proceedings. 2000, pp. 1647–1649.

[16]P. Moallem, “Improving Back-Propagation VIA an efficient Combination of A Saturation Suppression Method,” . Neural Network World Vol ,2no,10 pp. 207–223. , January , 2010.

[17] W. Wu, N. Zhang, Z. Li, L. Li, and Y. Liu, “convergence of Gradient Method with Momentum for back- propagation Neural Networks . Journal of Computational Mathematics, Vol.26, No.4, pp613–623, 2008

[18]H. Shao and G. Zheng, “A New BP Algorithm with Adaptive Momentum for FNNs Training,” 2009 WRI Global Congress on Intelligent Systems, 2009 ,pp. 16–20, 2009.