**Self-Adaptive the weight in Batch Back Propagation algorithm via Dynamic Training Rate**

Mohammed Sarhan Al\_Duais1 , Abdualmajed A.G. Al- khulaidi2 , Fatma Susilawati. Mohamad3, Mumtazimah Mohamad4 ,NoorainiYusoff5 ,Mohd Nizam Husen6, Azian Azamimi Abdullah has7

1,3,4 Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia ,

2, Faculty of computer science and information Technology, Sana’a university, Sana’a Yemen,

5 Department of data science , Universiti Malaysia Kelantan. , Kelantan Malaysia ,

6, Malaysian Institute of Information Technology, Universiti Kuala Lumpur (UniKL), Malaysia ,

7 Azian Azamimi Abdullah has , University Malaysia Perlis, perlis Malaysia

,

|  |  |  |
| --- | --- | --- |
| **Article Info** |  | **ABSTRACT** |
| ***Article history:***  Received -1th, 2022  Revised -, 201x  Accepted -th, 201x |  | Batch back propagation (BP) algorithm is commonly used in many applications, including robotics, automation, and global positioning systems. The learning rate is a significant parameter for increasing the efficiency of the training. The drawbacks of the BP algorithm are its slow training rate and easy convergence to the local minimum. To overcome this problem, we have created a new dynamic training rate to escape the local minimum, which enables a faster training time. We presented dynamic backpropagation algorithm(DBBPLR) which training with dynamic learning rate. This technique was implemented using a sigmoid function. The XOR problem, the Balance dataset, and the Iris dataset were used as benchmarks with different structures to test the efficiency of the dynamic training rate. The real datasets were divided into a training set and a testing set, and 75 experiments were carried out using Matlab software2016a. From the experimental results, it can be shown that the dynamic back propagation algorithm provides superior performance over the existing back propagation algorithm in terms of training, the speed of training, time training, number of epochs and accuracy training and also with existing work. |
| ***Keyword:***  Keywords: Artificial neural network; Batch Back-propagation algorithm; local minimum; Speed up Training, dynamic training rate***.***  ***Corresponding*** Author: Mohammed Sarhan Al\_Duais  \*EmailAddress: Sarhan2w@gmail.com |
| *Copyright © 201x Institute of Advanced Engineering and Science.  All rights reserved.* |

1. **Introduction**

An ANN provides a supervised learning algorithm which implements a nonlinear model within [0, 1] or [-1, 1], depending on the activation function . The BP algorithm involves parallel processing, which consists of several parameters which need to be adjusted to minimize error training. The BBP algorithm has led to tremendous breakthroughs in applications involving multilayer perceptions [1,2]. The batch BP algorithm is a new style for updating weight, it is widely used in training algorithms as it is accurate for training. Gradient descent is commonly used to adjust the weight using a change in error training(E); however, this approach is not guaranteed to find the global minimum error since the training is slow and converges easily to a local minimum [ 3- 5]. The BP algorithm is accurate in terms of training [6-8]. However, the primary problem with this algorithm is its slow training rate and easy convergence to a local minimum, and its tendency towards training saturation [9-11]. In addition to these disadvantages of the BP algorithm, several parameters need to be adjusted manually, such as learning rate and momentum term [12-14]. Therefore, one of the requirements for speeding up the  
BBP algorithm is adaptive learning rate. The learning rate should be sufficiently large to allow for escaping the local minimum to facilitate fast convergence to minimize error training. But the biggest value of the training rate leads to fast training with oscillation error training. the contrary, the small value of learning rate leads to reflex the weight that is lead to flat spot which makes B*BP* algorithm slow training. The too big values or small values are not suitable for smooth training of BBP algorithm. , It is difficult to select manually the best or suitable values in training of BBP algorithm[ 15 ] . However, a small adjustment to the training weight slows the training of the BBP algorithm, while large adjustment to the weight results in unsmooth training

To solve this problem, several techniques have been developed to improve learning, to speed up the BBP algorithm or to escape a local minimum, such as the heuristic approach. Heuristic methods are currently the most widely used methods for improving the training rate of the BP algorithm. Training rate is the most significant parameter affecting the weight update of a neural network. Many studies of this have been carried out, such as that in [ 16 ] proposed a new algorithm involving adjusting the dynamic training rate with a penalty for escaping from local minima. The weight is updated together with the penalty and the relationship between the training rate and the penalty. In this approach, the training rate η is a fixed at 0.013 and the penalty parameter is set as 0.001. The results are compared with the (SBP) algorithm. The study in [ 17 ] improves the batch BPAP algorithm using a proposed dynamic training rate with a penalty. The penalty coefficient is set at *λ* > 0, and the training rate set at *λ* =0.15. The algorithm has a 2:2:1 structure and uses a sigmoid activation function. The weight update in the batch BPAP algorithm is bounded during training. The experimental results show that the BPAP has faster training than the BP, with a fixed learning rate. In [ 18 ] investigated the effects of the input parameters using three different structures: the BP algorithm, the BP algorithm with a momentum factor, and the BP algorithm using a conjugate gradient descent. A sigmoid function was used as the activation function. The goals of this research were to study the ways in which differences in learning rates affected the recognition rate; the experiments, therefore, began with a sample structure and varied the value of the training rate. The second method used various values of training rates and hidden nodes, while the third method used a BP algorithm with a conjugate gradient descent. The experimental results indicated that the BP algorithm with the momentum factor provided the highest recognition rate, whereas the recognition rate for the other methods was 0.99. The work in [ 19 ] presented a dynamic BP algorithm for training with a boundary. In this case, the weight was updated under the constraints of this boundary; a sigmoid function was used as the activation function. The boundary helps to increase the rate of training of the BP algorithm and enhances the classification rate; in this study, the value of the classification correction was 91.1%.

The remainder of this paper is organized as follows: Section 2 describes the materials and method used; Section 3 presents the implementation; Section 4 presents the experimental results; Section 5 a discussion; Section 6 evaluation of the Performance of improving algorithm; Section 7 conclusions of the study; and Section 8 discusses future work.

**2. Materials and Method**

This kind of this research belongs to the heuristic method. This method includes the learning rate and momentum factor. To Investigate the aims of this study there are many steps as follows:

**2.1 Dataset**

The data set is very important for verification to improve the BBP algorithm. In this study, all data are taken from UCI Machine Learning Repository through the link <https://archive.ics.uci.edu/ml/index.html>. All real dataset changes to become normalization dataset between [0,1]. All data set divided into two set training set and testing set.

***2.1 Neural network Model***

The training BP algorithm is consist of three stages , namely ,forward propagation , in the feed forward phase ,each input unit xi receives an input signal xi and broadcasts this signal to the next layer until the end layer  
in the system. The backward propagation this step is starting when the output of the last layer reach to end step  
then the start feedback. The end steep is update the weight, in the batch BP algorithm the weight adjustment stage for all layer adjusted simultaneously.

We propose an ANN model, which consist of the three-layer neural network that has an input, hidden, and output layer. The input layer is considered to be {, ,...,  }, which represents the nodes; the nodes depend on the types or attributes of the data. The hidden layer is made of two layers with four nodes. Whereas the  and  are the first and second layer respectively. The output layer  is made of one layer with one neuron. Three basis, two of them are used in the hidden and one in the output layer, which is denoted by , and . is the weight between neuron h from hidden layer  and neuron j from the hidden layer .  is the weight between neuron  in the input layer and neuron h in the hidden layer. Finally, the sigmoid function is employed as an activation function.

**2.2 Create Dynamic Training rate (DBBPL)**

The weight update between neuron *k* from the output layer and neuron *j* from the hidden layer is as follows:

 ()

where  is a weight change . The weight is updated for each epoch in Equation , and the speed of the training depends on a parameter that affects the updating of the weight. There are two primary techniques for selecting the value for the learning rate. The first is to set it to a small constant value in the interval [0,1]; the second is to use a series value in the range [0,1] [ 20 ]. However, the large values or small values of the learning rate does not suitable for training BP algorithm .To enhance the BP algorithm, as given in Equation (), and to avoid local minima and training saturation, a dynamic function can be used to obtain an adaptive training rate as dynamic learning rate with boundary as follow:

 (2)

where K is the average of the activation function. The activation function used in this study is a sigmoid function. This formula uses (1-) as an implicit function in *η* to ensure that the expression cos( 1-Or ) (the boundary function) is a positive value for every value of .The dynamic training rate has both an upper and a lower bound; updating of the weight in the BP algorithm is bounded. We substitute  from Equation (2) into Equation (1) to obtain:

 (3)

The weight update is automatic for every layer under effect the dynamic training rate ( ).

**2.3 Dynamic Backpropagation (DBBPLR) algorithm**

The three stages of training in the backpropagation algorithm are the forward phase, the feedback phase, and the weight update phase. At the forward phase, the weights are calculated for each layer until the end layer or output layer is reached, at which point the feedback phase begins.

Update the Weight Phase

At the weight update stage, all of the layers are adjusted simultaneously. The weight update is calculated as follows:

For hidden layer LL  (4)

For bias  (5)

For hidden layer L  **(**6)

For biases  (7)

For each hidden layer  (8)

For the biases  (9)

**3. Implementation Dynamic algorithms (DBBPLR)**

The dataset is very important for verification to improve the algorithm. In this study, all data are taken from the UCI Machine Learning Repository available online at <https://archive.ics.uci.edu/ml/index.html>. These real data sets are divided into two parts, a training set, and a testing set. The BP algorithm was implemented using a fixed value for the training rate from the range [0,1], while the DBBPLR algorithm is trained using a dynamic function for the training rate.

**4. Experimental Results**

This section describes the experiments carried out to validate our proposed algorithm. We calculate the accuracy of training as follows [ 21] .

where *UP* and *LW* are the upper bound and lower bound of the activation function. A sigmoid function was used, and thus *UP* = 1 and *LW* = 0.

**4.1 Experimental Results for the DBBPLR Algorithm with the XOR Problem**

Ten experiments (EX) were carried out using Matlab, and the average (AV) was taken of several criteria used in this study for measurement of the training performance.The experimental results are tabulated in Table 1 below.

Table 1**.** Average the performance of DBBPLR algorithm with XOR with different structure

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | First structure | | | Second structure | | | | |
| Item | Time - sec | Epoch | Accuracy | | Time - sec | Epoch | Accuracy Training |
| Average | 2.3037 | 3700 | 0.9868 | | 5.1185 | 3323 | 0.9847 |
| S.D | 0.362968056 | 0 | 0 | | 0.201240776 | 0 | 1.11022E-16 |

From Table, for the first structure, the average time for training is *t* = 2.3037s and the epoch is 3700, while for the second structure the average training time is *t* = 5.1185s and the epoch is 3323. The S.D is close with zero for e time for both structure. The training curve is shown in Figure 1 below.

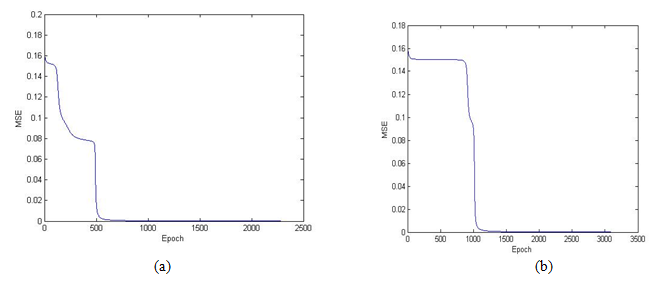


Figure 1. Training Curve of Dynamic algorithm with XOR

From Fig. 1(a), for the first structure, it can be seen that the curve training is a daisy with index epoch to meet the global minimum. in Fig. 1(b), the weight training does not change before 500 epochs. This means that the DBBPLR algorithm both curve converges quickly to give the minimum error.

**4.2 Experimental Results for the BP Algorithm Using the XOR Problem**This section presents the results of implementing the BP algorithm, as given in Equation 2, with a trial or manual values for each training rate. The average (AV) of the results of each experiment (EX) results are tabulated in Table 2.

Table 2 Average the performance of the training of BP algorithm with XOR Problem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | First structure Second structure | | | |
| Values of | Time - sec | Epoch | Time - sec | Epoch |
| AV | 225.7362 | 1707567 | 288.8325 | 2091897 |
| S.D | 127.1295117 | 2541604.076 | 260.2652117 | 2258757 |

From Table2, for first structure, the average training time is 225.7362 seconds with 1707567 epoch. For the second structure, the average training time 288.8325 second with 2091897 epoch. The S.D for both structures is greater than one.

**4.3 Experimental Results for the DBBPLR algorithm with the Balance Training Dataset**. Ten experimental has been done and the result recorded in Table 3 below.

Table 3. Average the performance of DBBPLR algorithm with balance -Training Set

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | First structure | | | Second structure | | |
| Item | Time - sec | Epoch | Accuracy | Time - sec | Epoch | Accuracy Training |
| AV | 0.1808 | 1 | 1 | 9.1166 | 50 | 0.9998333 |
| S.D | 0.03526131 | 0 | 0 | 1.2615048 | 7.4859869 | 0.0001333 |

From Table 3, for the first structure, the average training time is 0.1808s, with 1 epoch. For the second structure, the average training time is 9.1166, with 50 epochs. For first structure the training accuracy is one while the second structure close to one. The high values for accuracy indicate that the dynamic training rate helps the back-propagation algorithm to avoid training saturation, to achieve a higher training rate and to reach the global minimum.

**4.4 Experimental Results for the BP Algorithm with the Balance Training Dataset**

The performance was tested using 250 patterns as a form of training. The results of the simulation are given in Table 4.

Table 4.Average the performance of BP algorithm with balance-Training set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | First structure | | Second structure | |
| Values of | Time - sec | Epoch | Time-sec | Epoch |
| AV | 477.2925 | 2804 | 240.7302 | 767 |
| S.D | 693.2013113 | 4029.163 | 265.0520825 | 889.7967 |

From Table 4 for first structure, the average of time is 477.2925 477 s with average epoch is 2804, while the second structure the average of time training is 240.7302241 s with 767 epoch.

**4.5 Experimental Results for the DBBPLR Algorithm with the Balance Testing Dataset**

The experimental results are recorded in Table 5 below.

Table 5. Average the performance of DBBPLR algorithm with balance- Testing set

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | First structure | | | Second structure | | |
| Item | Time - sec | Epoch | Accuracy | Time - sec | Epoch | Accuracy Training |
| AV | **0.196** | **1** | **0.9666** | **10.33** | **97** | **0.98595** |
| S.D | **0.01920417** | **0** | **1.11E-16** | **0.4807896** | **3.324154** | **5E-05** |

From Table 5, it can be seen that the dynamic approach for training rate reduces the time required for training and enhances the convergence of the time training. For the first structure, the average training time was 0.196s with an average one epoch. For the second structure, the average training time was 10.33s with an average of 97 epochs. Both structures gave high training accuracy, and the average SD of time for both structures was less than one.

**4.6 Experimental Results for the BP algorithm with the Balance Testing Dataset**

In this section, we implement the backpropagation algorithm using 250 patterns, representing the testing dataset. The experimental results are given in Table 6 below.

Table 6**.** The performance of the training of BP algorithm with Balance- Testing set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | First structure Second structure | | | |
| Values of | Time - sec | Epoch | Time - sec | Epoch |
| AV | 1229.051333 | 6991 | 1110.9285 | 14636 |
| S.D | 1676.514917 | 11334 | 1439.474401 | 23117 |

From Table 6 above, it can be seen that for first structure the average time training is 1229.051333 second with 6991 epoch. For second structure the average time is 1110.9285 seconds with 14636 epoch.

**4.7 Experimental Results for the DBBPLR Algorithm with the Iris Training Dataset**

The experimental results are given in Table 7 below

Table 7. Average Training DBBPLR algorithm with Iris – Training

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | First structure | | | Second structure | | |
| Item | Time - sec | Epoch | Accuracy | Time - sec | Epoch | Accuracy Training |
| AV | 0.1143 | 1 | 0.9998 | 2.6905556 | 69 | 0.9962 |
| S.D | 0.03118028 | 0 | 2.22E-16 | 0.1560599 | 0 | 0 |

From Table 7 above, for both structures, the average training time is very short: for the first structure, the training time is 0.1143s with one epoch, for the first structure and 2.6905556s with 69 epochs. In addition, the average accuracy of the dynamic algorithm for both structures is 0.9998 and 0.9962 respectively. The training curve is shown in Figure 2 below.

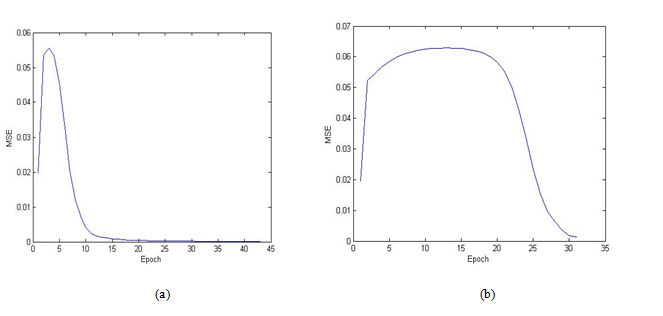


Figure 2. Training Curve of Dynamic algorithm with Iris –Training dataset

From Fig. 2(a), the dynamic algorithm to avoid the flat spot after 10 epochs, while in Fig. 2(b), training does not change before 30 epochs for the second structure. For both structures, the training curve converges quickly to give the minimum error.

**4.8 Experimental Results for the BP Algorithm with the Iris Training Dataset**

Table 8. Average Training BP algorithm with Iris – Training

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | First structure Second structure | | | |
| Values of | Time - sec | Epoch | Time - sec | Epoch |
| A.V | 5108.759 | 254670 | 735.555 | 16062 |
| S.D | 10185. | 10185.22 | 742.6004044 | 17109.799 |

From Table 8, for first structure the average time training is 5108.759 second with 254670 epochs while second structure the average time is 735.555 seconds with 16062 epochs.

**4.9 Experimental Results for the DBBPLR Algorithm with the Iris Testing Dataset**

Ten experiments were carried out in Matlab. The average of several criteria was used in this study for the measurement of the training performance. The experimental results are given in Table 9.

Table 9**.** Average Training Improve DBBPLR algorithm with Iris – Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | First structure | | | Second structure | | |
| Item | Time - sec | Epoch | Accuracy | Time - sec | Epoch | Accuracy Training |
| AV | 4.7898 | 161.7 | 0.96415 | 0.9837 | 35 | 0.9571 |
| S.D | 1.06378953 | 46.312093 | 0.0010929 | 0.0719 | 0 | 0 |

From Table 9, Both the structures, the average of the training time is very short. The average time is 4.7898seconds for the first structure while the average time for the second structure is 0.9837for second structure. Both structures have the highest accuracy rate. In addition, the average accuracy of a dynamic algorithm for both structures is 0.96415 and 0.9571respectively.

**4.10Experiments of the BP algorithm with Iris - Testing Set**

Table 10. Average Training BP algorithm with Iris – Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | First structure Second structure | | | |
| Values of | Time-sec | Epoch | Time - sec | Epoch |
| AV | 2087.4398 | 63339.4 | 1379.29975 | 10751 |
| S.D | 2038.59631 | 65565.41043 | 1639.369388 | 5747.54 |

For first structure, the average training time is 2087.4398 seconds with 63339 epochs. For the second structure, the average training time 1379.29975-second with 10751 epoch. The S.D for both structures is greater than one.

**5. Discussion**

This section presents a discussion of the performance of the training time for each DBBPLR and BP algorithm to determine which is superior. We calculate the processing time improved following formula [ 22 , 23 ]

Processing Time Improved =

**5.1 Performance Training of the DBBPR Algorithm Versus the BBP Algorithm for the First Structure**

To validate the improved BP algorithm, we compare the performance of the DBBPLR algorithm and the BP algorithm. The speed-up obtained in training is shown in Table 11 below.

Table 11**.** Processing time Improved DBBPLR algorithm versus BP with first structure

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| First structure | | | | Second structure | | |
|  | DBBPLR algorithm | BBP algorithm | Processing Time Improved | DBBPLR algorithm | BBP algorithm | Processing Time Improved |
|  | AV time | AV time | (BBP/DBBPLR) | AV time | AV time | (BBP/DBBPLR) |
| XOR | 2.3037 | 225.7362 | 97.9885402 | 5.1185 | 288.8325 | 56.4291296 |
| Balance Training | 0.1808 | 477.2925 | 2639.89215 | 9.1166 | 240.7302 | 26.4056995 |
| Balance Testing | 0.196 | 1229.5131 | 6273.02602 | 10.33 | 1110.9285 | 107.543901 |
| Iris Training | 0.1143 | 5109.759 | 44704.8031 | 2.6905556 | 526.3825 | 195.640819 |
| Iris Testing | 4.7898 | 1379.2997 | 287.966032 | 0.9837 | 526.3825 | 535.104707 |
| Average Processing Time improved |  |  | 10800.7352 |  |  | 184.224851 |
|

From Table 11, it is evident that the dynamic DBBPLR algorithm provides superior performance over the BBP algorithm for all datasets. The processing time improved is = 10800.7352s time faster than BBP algorithm for first structure and The processing time improved is = 184.224851times faster than BBP algorithm for second structure.

**6. Evaluation of the Performance of Improved Batch BP algorithm**

We have evaluated the performances of the DBBPLR algorithm for speed up training. The performances of the improved BBPLR algorithm are compared with the performance of previous works. The limited error or stop training is set at 0.0001 in this study; elsewhere, this value is set at 0.0001 in the previous literature such as Liu et al (2015). While the Abbas *et al* (2016), set the stop training at 500 epochs. For the purpose of the comparison between the results of this study and the previous works, the fit is rerun again with different stop training values. In conclusion, the results of the current study prove that the DBBPLR algorithm outperforms the previous studies in the form of the training time, epoch, and the accuracy training.

**7. Conclusions.**

The training rate is a widely used technique for improving the batch BP algorithm and an important parameter for controlling the weight training. This study introduces the DBBPLR algorithm which training by training rate. The dynamic training rate affects the weight of each hidden layer and output layer and eliminates training saturation. One of the main advantages of the dynamic learning rate is that it reduces training time, error training and the number of epochs. The experimental results show that the DBBPLR algorithm gives superior performance compared with the existing of BP algorithm and exsist studies.

**Future Work**

This study was carried out using a dynamic training rate algorithm. However, it is possible to create a dynamic function for each training rate and momentum, in order to reduce the training time in the BP algorithm further.

**REFERENCES**

[1].P.Moallem . Improving Back‐Propagation VIA an efficient Combination of A Saturation Suppression Method. Neural Network World. 20,2(2010).

[2].D.Xu, H. Shao, and H.Zhang, H. A new adaptive momentum algorithm for split-complex recurrent neural networks. Neurocomputing, 93(2012). 133-136.

[3] M.S.Al\_Duais , F.Mohamad. “ Improved Time Training with Accuracy of Batch Back Propagation Algorithm Via Dynamic Learning Rate and Dynamic Momentum Factor.” IAES International Journal of Artificial Intelligence , Vol. 7, No. 4, , pp. 170~178, 2018

[ 4].S.Nandy, P.P Sarkar and A.Das (2012). An Improved Gauss-Newtons Method based Back-propagation algorithm for fast convergence. International Journal of Computer Applications.2012 Feb 8, 39(8) 1206.4329.

[ 5]. JM.Rizwan, PN.Krishnan, R.Karthikeyan , SR.Kumar. Multi-layer perception type artificial neural network based traffic control. Indian Journal of Science and Technology. 9,5(2016).

[ 6]. Shao, Y., Zhao, C., Bao, Y., & He, Y. Quantification of nitrogen status in rice by least squares support vector machines and reflectance spectroscopy. Food and Bioprocess Technology.2012 ; 5(1), 100-107.

[7]. L.Wang, Y.Zeng, T.Chen.Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Systems with Applications. 42,2(2015) ,PP 855-63.

[8].Q.Dai and N.Liu. A two-phased and Ensemble scheme integrated Backpropagation algorithm. Neurocomputing.94(2014). 1124-1135.

[9] M. S. Al\_Duais, & F. S .Mohamad (2016). “A Review on Enhancements to Speed up Training of the Batch Back Propagation Algorithm”, Indian Journal of Science and Technology, vol.9.no.46. pp. 1-10, 2016.

[ 10].E.Noersasongko , FT.Julfia, A.Syukur.Pramunendar RA, Supriyanto C. A tourism arrival forecasting using genetic algorithm based neural network. Indian Journal of Science and Technology. 2016 Jan 16;9(4).

[ 11] Y. Liu., Z., Li, D. Yang, K.S.Mohamed, L.Wang, & W.Wu, “ Convergence of batch gradient learning algorithm with smoothing L 1/2 regularization for Sigma–Pi–Sigma neural networks”, Neurocomputing, 151,333-341(2015).

[ 12].H.Shao, J.Wang, L.Liu, D.Xu, and W.Bao. Relaxed conditions for convergence of batch BPAP for feed forward neural networks. Neurocomputing, pp. 174-79,153(2015).

[ 13]. C.Kansas. Analysis on the parameter of back propagation algorithm with three weight adjustment structure for hand written digit recognition. Proceedings of the10th International Conference on Service Systems and Service Management, 2013,pp. 18-22.

[14]. D.Sha, & V.B.Bajic. An optimized recursive learning algorithm for three-layer feedforward neural networks for mimo nonlinear system identifications. Intelligent Automation & Soft Computing.17,2, PP.133-147,2015.

[ 15 ] Mohameed Sarhan Al\_Duais, AbdRazak Yaakub, Nooraini Yusoff and Faudziah Ahmed “ A Novel Strategy for Speed up Training for Back Propagation Algorithm via Dynamic Adaptive the Weight Training in Artificial Neural Network “Research Journal of Applied Sciences, Engineering and Technology 9(3): 189-200, 2015

[ 16 ] S.Sureerattanan ,H.P. PhienH. Sureerattanan NMastorakis E. The optimal multi-layer structure of back propagation networks. Proceedings of the 7th WSEAS International Conference on Neural Networks, Croatia, 2006, 108-13.

[ 17 ] .ZX.Yang, GS.Zhao, HJ.Rong, J.Yang . Adaptive backstepping control for magnetic bearing system via feedforward networks with random hidden nodes. Neurocomputing. 2016 Jan 22;174:109-20.

[ 18] .C.Yang & R.Xu. Adaptation Learning Rate Algorithm of Feed-Forward Neural Networks. International Conference in Information Engineering and Computer Science (2009) Dec 19. (pp. 1-3).

[ 19 ] M. S . Al Duais , Mohamad, F. S.(2017) Dynamically-adaptive Weight in Batch Back Propagation Algorithm via Dynamic Training Rate for Speedup and Accuracy Training. Journal Telecommunications and Information Technology. 4 *doi*.*org*/*10.26636*/*jtit*.*2017.113017*, 4*,* PP.82-89

[ 20 ] N.MNawi, N.A.Hamid, R.S Ransing ,R. Ghazali and M.N.M Salleh. Enhancing Back Propagation Neural Network Algorithm with Adaptive Gain on Classification Problems. Networks . 4,2(2011).

[ 21 ] M.S. Al\_Duais, F.Mohamad, M. Mohamad. Improved the Speed Up Time and Accuracy Training in the Batch Back Propagation Algorithm via Significant Parameter. International Journal of Engineering & Technology, 7 (3.28) , pp. 124-130 , 2018

[ 22 ] F.Saki , A.Tahmasbi, H.Soltanian-Zadeh and S.B. Shokouhi. Fast opposite weight learning rules with application in breast cancer diagnosis. Computers in biology and medicine. 43,1 (2013).32-41.

[ 23 ] [1] Mohammed Sarhan Al \_ Duais, Fatma Susilawati Mohamad, Mumtazimah Mohamad, Mohd Nizam Husen, "Enhancement Processing Time and Accuracy Training via Significant Parameters in the Batch BP Algorithm", International Journal of Intelligent Systems and Applications(IJISA), Vol.12, No.1, pp.43-54, 2020. DOI: 10.5815/ijisa.2020.01.05.