Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

AUTO-ADAPTIVE THE WEIGHT IN BATCH BACK PROPAGATION ALGORITHM VIA DYNAMIC LEARNING RATE

MOHAMMED SARHAN AL_DUAIS
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia.
Email: sarhan2w@gmail.com

ABDUALMAJED A.G. AL- KHULAIDI
Faculty of Computer Science and Information Technology, Sana'a University, Sana'a Yemen.
Email: alkhulaidi@su.edu.ye

FATMA SUSILAWATI. MOHAMAD
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia.
Email: fatma@unisza.edu.my

BELAL AL-FUTHAIDI
Faculty of Computing and IT, University Of Science & Technology Sana'a, Yemen.
Email: belalarh@gmail.com

SADIK ALI MURSHID AL- TAWEEL
Faculty of Computing and It, University of Science & Technology Sana'a, Yemen, Altaweel.
Email: sadik@gmail.com

WALID YOUSEF
Faculty of Computing and IT, University of Science & Technology Sana'a, Yemen.
Email: walidshaher@gmail.com

MUMTAZIMAH MOHAMAD
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia.
Email: mumtaz@unisza.edu.my

Abstract

Batch back propagation (BBP) algorithm is commonly used in many applications, including robotics, automation, and global positioning systems. The man drawbacks of batch back propagation (BBP) algorithm is slow training, and there are several parameters needs to be adjusted manually, also suffers from saturation training. The objective of this study is to improve the speed uptraining of the BBP algorithm and to remove the saturation training. To overcome these problems, we have created a new dynamic learning rate to escape the local minimum, which enables a faster training time. We presented dynamic batch 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 learning 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 DBBPLR algorithm provides superior performance over the existing BBP algorithm in terms of training, the speed of training, time training, number of epochs and accuracy training and also with existing work.

Keyword: Artificial neural network, Batch Back-propagation algorithm, local minimum, Speed up Training, dynamic training rate. Auto -Adaptive the weight

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. Applications involving multilayer perceptions have seen enormous advancements thanks to the BBP algorithm [1], [2]. Because it is accurate for training, the batch BP algorithm, a new method of updating weight, is frequently used in training algorithms. Gradient descent is a popular technique for adjusting the weight using a change in error training (E); however, this approach is not ensured to find the global minimum error because the training is slow and readily converges to a local minimum [3],[4], [5]. Regarding training, the BBP algorithm performs well [6] ,[7] ,[8]. However, the main drawbacks of BBP algorithm are slow training rate, and easy convergence to a local minimum, with propensity to training saturation [9],[10],[11]. in addition to these problem ,the BBP algorithm has a number of drawbacks such as the learning rate and momentum term must be manually adjusted [12,[13],[14].Therefore, one of the requirements for speeding up the BBP algorithm is adaptive learning rate as dynamic funcation .

The adaptive learning rate is thus one of the conditions for speeding up the BBP algorithm. In order to facilitate quick convergence and minimize error training, the learning rate needs to be sufficiently high to allow for escaping the local minimum. However, the highest earning rate value results in quick training with oscillation error training. Contrarily, a low learning rate causes a weight to reflex, which causes a flat spot, which slows down the training of the BBP algorithm. The adaptive learning rate is thus one of the conditions for speeding up the BBP algorithm. In order to facilitate quick convergence and minimize error training, the learning rate needs to be sufficiently high to allow for escaping the local minimum. However, the highest training rate value results in quick training with oscillation of training error. Contrarily, a low values of learning rate causes a weight to reflex, which causes a flat spot, which slows down the training of the BBP algorithm. The BBP algorithm cannot be trained smoothly with values that are either too large or too small, so that the too big values or small values are not suitable for smooth training of BBP algorithm. Manually choosing the best or most appropriate values during training the BBP algorithm is a big challenging. [15]. Small changes to the training weight, on the other hand, cause the BBP algorithm to train more slowly, while significant changes to the weight produce training that is not smooth.

To solve this issue, a number of techniques, including the heuristic approach, have been developed to enhance learning, speed up the BBP algorithm, or escape local minima. The BP algorithm's training rate can be increased using a variety of techniques, but heuristic methods are currently the most popular. The most important factor influencing how often a neural network updates its weights is its training rate.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

2. LITERATURE REVIEW

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. Numerous studies have been done on this, such as the one in [16], which proposed a new algorithm involving adjusting the dynamic training rate with a penalty for escaping from local minim. In this study, the training rate is fixed at 0.013 and the penalty parameter is set as 0.001. The outcomes are compared with the (SBP) algorithm. The study in [17] suggests a dynamic learning rate with a penalty λ to enhance the batch BBP algorithm. The learning rate is set to = 0.15, and the penalty coefficient is set at $\lambda > 0$ the algorithm used a sigmoid activation function. The structure of algorithm has t three layer, one input layer with WO node, two hidden layer and one layer with one node during training, the batch BBP algorithm's weight update is constrained. The experimental findings demonstrate that the BPAP, with a fixed learning rate, trains more quickly than the BBP algorithm. In [18], three different BBP algorithm structures each with a momentum factor were used to examine the effects of the input parameters. Utilizing conjugate gradient descent is the BBP algorithm. The activation function was a sigmoid function. The experiments started with a sample structure and varied the value of the learning rate in order to achieve the research's objectives of examining how variations in learning rates affected the recognition rate. The third method used a BBP algorithm with a conjugate gradient descent, while the second method used a range values of learning rates and hidden node values. The experimental results showed that the momentum factored BBP algorithm had the highest recognition rate. Whereas the recognition rate for the other methods was 0.99. The work in [19] presented a dynamic BBP 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.

3. RESEARCH MATERIALS AND METHODS

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:

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

3.1 Dataset

The data set is crucial for verification in order to strengthen the BBP algorithm. All data used in this study were obtained from the UCI Machine Learning Repository using the following link: https://archive.ics.uci.edu/ml/index.html. Every real dataset is transformed into a normalization dataset between [0, 1]. The entire data set was divided into two sets: a training set and a testing set.

3.2 Neural Network Model

The training BBP algorithm consists of three stages, namely, forward propagation, in which each input unit x_i receives an input signal x_i and broadcasts this signal to the next layer until the end layer in the system. When the output from the previous layer reaches the end step and the start feedback, the backward propagation of this step begins. In the batch BP algorithm, the weight adjustment stage for every layer is adjusted simultaneously at the end steep [20],[21]. This study proposed a three-layer neural network with an input, hidden layer, and output layer that makes up the ANN model. The nodes that make up the input layer—which are thought of as {x_1, x_2,..., x_i } depend on the different types or attributes of the data. Two hidden layer layers with four nodes make up whereas the Z_h .and ZZ_r are the first and second layer respectively. The output layer O_r is made of one layer with one node. Three basis, two of them are used in the hidden and one in the output layer, which is denoted by u_{0j}, v_{0k} and w_{0r}. v_{hj} Is the weight between neuron h from hidden layer Z_h and neuron j from the hidden layer ZZ_r . u_{ih} Is the weight between neuron i in the input layer and neuron h in the hidden layer. Finally, the sigmoid function is employed as an activation function.

In this study proposed two structures, in order to verify the improved of dynamic algorithm with several dataset. However, the first structure (I: 2H: O) is consist of a multi-neural network (ML) with three layers – an input layer (I), two hidden layers (2H) with four nodes. And output layer (O) which consists of one layer with one neuron. Three biases are used in the two in hidden layers, and one is used in the output layer. In the contrary the second structure is (I: H: O), which consists of an input layer, one hidden layer with two nodes, and one output layer with one node. Two biases are used in the one in hidden layers, and one is used in the output layer. However, the bias takes one value. The training of BBP algorithm shown in the figure 1 below

ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

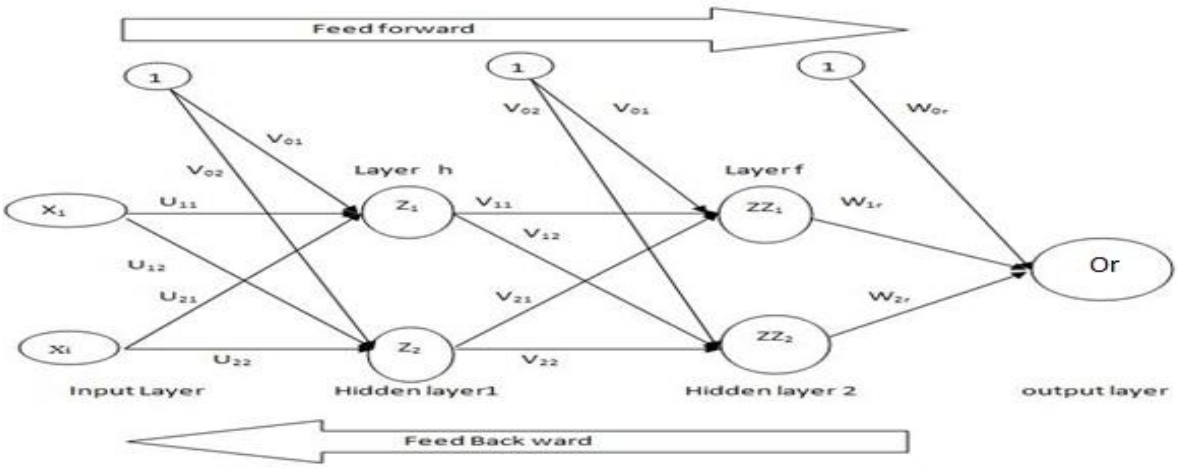

Figure 1: Batch BP Algorithm Structure

Before presenting the DBBPLR algorithm, let us briefly define some of the notations used in the algorithms follows

Z_h First hidden layer for neuron h, h = 1, …, q

ZZ_r Second hidden layer for neuron j, j = 1,..., p

Or Output layer for neuron r

u_{ih} The weight between neuron i in the input layer and neuron h in the hidden layer

u_{0h} The weight of the bias for neuron j

v_{hj} The weight between neuron h from hidden layer z and neuron j from the hidden layer ZZ

v_{0j} The weight of the bias for neuron j

w_{jr} The weight between neuron k from the hidden layer ZZ and neuron r from the output layer O

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

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

δ_r The error back propagation at neuron r

δ_j The error back propagation at neuron j

3.3. Create Dynamic Training Rate (DLR)

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

$$\Delta \mathrm{w}_{\mathrm{jk}}\left(\mathrm{t}+1\right) = \mathrm{w}_{jk}(t) - \gamma \frac{\partial E}{\partial W_{jk}(t)} \tag{1}$$

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

Where $\Delta w_{jk}(t)$ is a weight change. The weight is updated for each epoch in Equation1, and the speed of the training depends on a parameter that affects the updating of the weight.

To choose of the learning rate value can be made using one of two main techniques. The first option is to set it too a small constant value between [0,1], and the second techniques is to use a series value between [0,1] [22]. However, the training of the BBP algorithm is not appropriate for the large or small values of the learning rate. To improve the BP algorithm given in Equation (1)

To avoid local minima and training saturation, a dynamic function can be used to obtain an adaptive learning rate as dynamic learning rate with boundary. We dented the dynamic training rate by γ_{dmic} as follow:

$$\gamma_{dmic} = k + \frac{1}{\cos(1-o_r)} \tag{2}$$

Where K is the average of the activation function. The activation function used in this study is a sigmoid function. This formula uses (1-o_r) as an implicit function in γ to ensure that the expression cos(1-O_r) (the boundary function) is a positive value for every value of o_r, where o_r is the output layer at neuron 'r'. The dynamic learning rate has both an upper and a lower bound; updating of the weight in the BBP algorithm is bounded. We substitute γ_{dmic} from Equation (2) into Equation (1) to obtain to getting equation (3):

$$\Delta \mathrm{w}_{\mathrm{jk}}\left(\mathrm{t+1}\right) = \mathrm{w}_{jk}(t) - [k + \frac{1}{\cos(1-o_r)}]\frac{\partial E}{\partial W_{jk}(t)} \tag{3}$$

The weight update is automatic for every layer under effect the dynamic learning rate (γ_{dmic}).

3.4. Dynamic Batch Backpropagation (DBBPLR) Algorithm

The dynamic batch backpropagation **(**DBBPLR**)** has three training phases: the forward phase, the feedback phase, and the weight update phase. The weights are computed for each layer during the forward phase until the end layer or output layer is reached, at which point the feedback phase starts.

3.4.1 Update the Weight Phase in (DBBPLR) Algorithm

In the batch BP algorithm, the weight in all layers is adjusted simultaneously during the weight update stage. The weight update is determined using the formula below:

For hidden layer ZZ_j the weight updating as equation (4)

$$ZZ_k \quad \mathrm{W}_{\mathrm{jr}}\left(\mathrm{t+1}\right) = w_{jr}(t) + [\ k + \frac{1}{\cos(1-o_r)}]\delta_r ZZ_j \tag{4}$$

For bias the weight updating as equation (5)

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

$$\mathrm{W}_{0\mathrm{r}}\left(\mathrm{t+1}\right)=w_{jr}(t)+[\ k+\frac{1}{\cos(1-o_r)}]\delta_r \tag{5}$$

For hidden layer Z_h the weight updating as equation (6)

$$\mathrm{v}_{\mathrm{hj}}\left(\mathrm{t+1}\right)=v_{hj}(t)+(\ k+\frac{1}{\cos(1-l_r)})\delta_j Z_h \tag{6}$$

For biases the weight updating as equation 7

$$\mathrm{v}_{0\mathrm{j}}\left(\mathrm{t+1}\right)=v_{0j}(t)+(\ k+\frac{1}{\cos(1-l_r)})\delta_j \tag{7}$$

For layer x_i the weight updating as equation (8)

$$\mathrm{u}_{\mathrm{ih}}\left(\mathrm{t+1}\right)=u_{ih}(t)+(\ k+\frac{1}{\sin(1-o_r)})\delta_h x_i \tag{8}$$

For the biases the weight updating as equation (9)

$$\mathrm{u}_{0\mathrm{h}}\left(\mathrm{t+1}\right)=u_{0h}(t)+(\ k+\frac{1}{\cos(1-o_r)})\delta_h \tag{9}$$

4. EXPERIMENTAL RESULTS

In this part we going to implement both algorithms, namely DBBPLR algorithm and BBP algorithm. In this study, the DBBPLR algorithm is trained using a dynamic function for the learning rate which created in equation (2), whereas the BBP algorithm was implemented using a manually values for the learning rate from the range [0,1]. Several datasets have been used such as, XOR problem, balance, Iris dataset. The tests conducted to verify the proposed algorithm are described in this part. We compute training accuracy as follows [23]

$$\text{Accuracy } (\%) = \frac{1-\text{absolut}(\mathrm{T_i}-\mathrm{O_i}\quad)}{\mathrm{UP}-\mathrm{LW}} * 100$$

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. In the same way this study used some criteria to test the performance such as average training time, average S.D, number epoch.

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

Ten experiments (EX) were carried out using MATLAB, Several criteria were used in this study to measure the effectiveness of the dynamic learning rate which created in equation 2, and the average (AV) of those criteria was taken. The experimental results are tabulated in Table 1 below.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

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

According to Table 1, the average training time for the first structure is t = 2.3037s and the epoch is 3700, while the average training time for the second structure is t = 5.1185s and the epoch is 3323. For both structures, the S.D. is nearly zero for training time. The below-mentioned Figure 2 depicts the training curve.

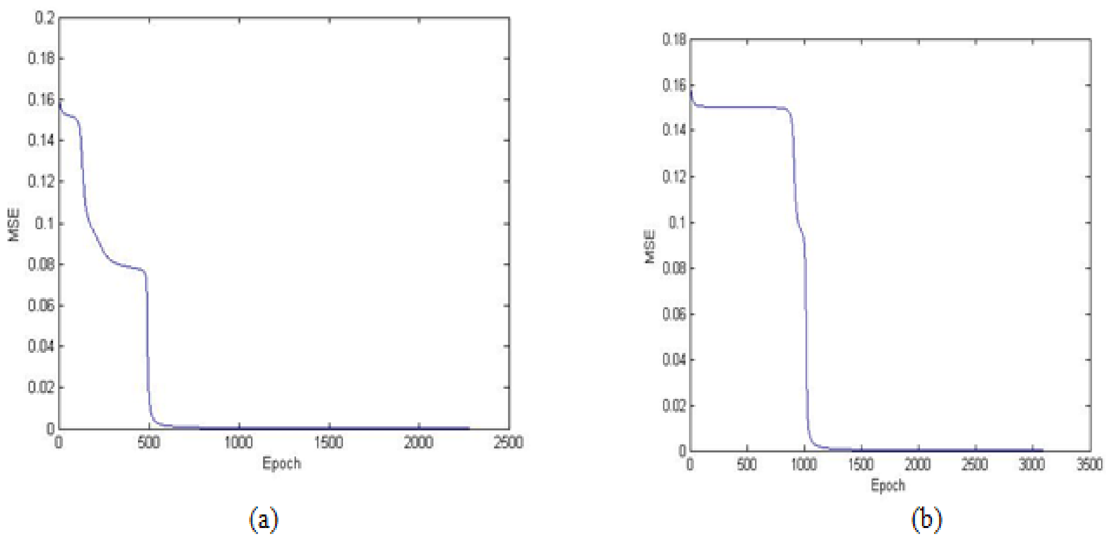

Figure 2: Training Curve of DBBPLR Algorithm with XOR

As seen in Figure. 2(a), the training curve of DBBPLR algorithm for the first structure is decreases with an index epoch to reach the global minimum before 500 epochs, the weight training in Figure 2(b) remains the mean. This means that the DBBPLR algorithm each curve converges quickly to reach the global minimum.

4.2. Experimental Results for the BP Algorithm Using the XOR Problem

In this part implement the BBP algorithm, which present the results, along with trial or manual values for each learning rate. The average results of each experiment (EX) are tabulated in Table 2.

Table 2: Average the Performance of the Training of BBP Algorithm with XOR Problem

	First structure		Second structure	
Values of γ	**Time - sec**	**Epoch**	**Time - sec**	**Epoch**
AV	225.7362	1707567	288.8325	2091897
S. D	127.1295	2541604.076	260.2652	2258757

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

According to Table2, the first structure's average training time with 1707567 epoch is 225.7362 seconds. The second structure's average training time was 288.8325 seconds with 2091897 epochs. Both structures have S.D that are higher than 1.

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	0.9999
S.D	0.0352	0	0	1.2615048	7.4860	0.0001

According to Table 3, in the first structure average training time with 1 epoch is 0.1808 seconds. in the second structure average training time for 50 epochs is 9.1166 seconds. The training accuracy for the first structure is one, but it is nearly one for the second structure. The high accuracy results suggest that the dynamic learning rate aids in preventing training saturation, achieving a higher training rate, and reaching the global minimum for the DBBPLR algorithm.

4.4 Experimental Results for the BBP 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 BBP Algorithm with Balance- Training set

	First structure		Second structure	
Values of γ	**Time - sec**	**Epoch**	**Time-sec**	**Epoch**
AV	477.2925	2804	240.7302	767
S.D	693.2013	4029.163	265.0521	889.7967

According to Table 4, in the first structure average training time is 477.2925 s with average epoch is 2804, whereas the second structure average training time is 240.7302 s and its average epoch is 767.

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	0.9859
S.D	0.0192	0	1.11E-16	0.4808	3.324	5E-05

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

Form table 5 shows that the dynamic learning rate technique, it helps the DBBPLR algorithm for reduce training period and improves the convergence of the training period. For the first structure average training took an is 0.196 seconds with one epoch. With an average training time was 97 epochs. For the second structure average training time was 10.33 seconds. The average S.D for time for both structures was less than one, and both provided excellent training accuracy.

4.6 Experimental Results for the BBP 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 BBP Algorithm with Balance-Testing Set

	First structure		Second structure	
Values of γ	Time - sec	Epoch	Time - sec	Epoch
AV	1229.0513	6991	1110.9285	14636

From Table 6 above, it can be seen that for first structure the average time of training is 1229.0513 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	2.6906	69	0.9962
S.D	0.0312	0	2.22E-16	0.1561	0	0

According to Table 7 above, the average training time for both structures is extremely brief: for the first structure, the training time is 0.1143s with one epoch, while for the second structure, the average training time is 2.6906s with 69 epochs. Furthermore, the DBBPLR algorithm has robustness of average accuracy for the two structures is 0.9998 and 0.9962, respectively. The training curve is displayed in Figure 3 below.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

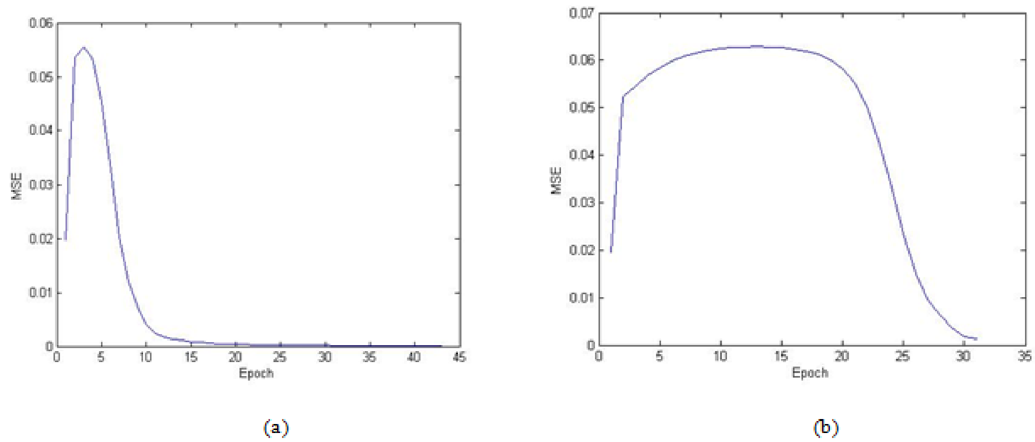

Figure 3: Training Curve of DBBPLR algorithm with Iris –Training dataset

From Figure. 3(a), the DBBPLR algorithm to avoid the flat spot after 10 epochs, while in Figure. 3(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 BBP Algorithm with the Iris Training Dataset

Table 8: Average Training BP algorithm with Iris – Training

	First structure		Second structure	
Values of γ	Time - sec	Epoch	Time - sec	Epoch
A. V	5108.759	254670	735.555	16062
S.D	10185.	10185.22	742.6004	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 epoch.

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

Ten experiments were carried out in Matlab 2016 a. 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	1.0638	46.3121	0.0011	0.0719	0	0

According to Table 9, the average training time for both structures is quite brief. The first structure , the average time is 4.7898 seconds, whereas the second structure , the average time takes 0.9837 seconds. Both structures have the highest accuracy rate. Additionally, the average accuracy of DBBPLR algorithm for each structure is 0.96415 and 0.9571, respectively.

ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

4.10. Experiments of the BBP algorithm with Iris - Testing Set

Table 10: Average Training BBP algorithm with Iris – Testing

	First structure		Second structure	
Values of γ	**Time-sec**	**Epoch**	**Time - sec**	**Epoch**
AV	2087.4398	63339.4	1379.2998	10751
S.D	2038.5963	65565.41043	1639.3694	5747.54

From Table 10, the first structure, the average training time is 2087.4398 seconds with 63339 epochs. For the second structure, the average training time 1379.2998-second with 10751 epochs. 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 [24], [25].

$$\text{Processing Time Improved} = \frac{\text{Execution time of B}BP\ algorithm}{\text{Execution time of DBBPLR}\ algorithm}$$

5.1. Performance Training of the DBBPR Algorithm versus the BBP Algorithm for both Structure

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

Table 11: Processing time Improved DBBPLR Algorithm Versus BBP with First Structure

	First structure			Second structure		
Data set	**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	1229.513	6273.02602	10.33	1110.9285	107.543901
Iris Training	0.1143	5109.759	44704.8031	2.6906	735.555	273.3795
Iris Testing	4.7898	2087.438	435.8093	0.9837	1379.298	1402.1549
Average Processing Time improved			10830.7352			373.1826

It is clear from Table 11 that, performance of the DBBPLR algorithm, which training with dynamic learning rate gives superior performance over the BBP algorithm for all datasets. For the first structure, the processing time improved is 10830.7352s faster than the BBP algorithm, and for the second structure, it is 373.1826times faster than the BBP algorithm.

6. EVALUATION THE PERFORMANCE OF DBBPLR ALGORITHM

To evaluation the performances of the DBBPLR algorithm for speed up training via compared with the performance of previous studies. In this study, the error or stop training is set at 0.0001; elsewhere, this value is fixed at 0.0001 in the earlier literature, such as Liu *et al*. (2015). In this study, the error or stop training is set at 0.0001; elsewhere, this value is fixed at 0.0001 in the earlier literature, such as Liu *et al*. (2015). Abbas et al. (2016) put the stop training at 500 iterations. For the purpose of the comparison between the results of this study and the previous works, in this study rerun again of DBBPLR algorithm with different training values of the stop training. 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

A common technique for enhancing the batch BP algorithm is learning rate, which is also a crucial parameter for controlling weight training. The DBBPLR algorithm, which trains by dynamic learning ate, is introduced in this study. The weight of each hidden layer and output layer is influenced by the dynamic learning rate, which also prevents training saturation. The dynamic learning rate has several benefits, including a reduction in training time, training error, and number of epochs. The experimental results demonstrate that the DBBPLR algorithm outperforms both the BBP algorithm currently in use and also exsist studies.

References

1) P.Moallem ," Improving Back-Propagation VIA an efficient Combination of A Saturation Suppression Method," Neural Network World," vol.20, no.2 ,pp. 207-222, 2010,doi:
2) D.Xu, H. Shao, and H. Zhang, "A new adaptive momentum algorithm for split-complex recurrent neural networks," Neurocomputing, vol. 93, pp. 133-136 , 15 Sep. 2012 , doi :org/10.1016/j.neucom.2012.03.013.
3) M.S.Al_Duais and 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, Dec2018 , doi: 10.11591/ijai.v7.i4.pp170-178.
4) S.Nandy, P.P Sarkar and A.Das ,"An Improved Gauss-Newtons Method based Back-propagation algorithm for fast convergence," International Journal of Computer Applications, vol.39, no.8, pp. 0975 – 8887. Feb.2012 ,doi: 0.17485/ijst/2016/v9i5/87267
5) JM.Rizwan, PN.Krishnan, R.Karthikeyan and SR.Kumar," Multi-layer perception type artificial neural network based traffic control," Indian Journal of Science and Technology, vol. 9, no.5 ,pp.1-6, Feb .2016, d0i: 10.17485/ijst/2016/v9i5/87267
6) Y. Shao, C. Zhao, Y. Bao and Y.He, "Quantification of nitrogen status in rice by least squares support vector machines and reflectance spectroscopy," Food and Bioprocess Technology.2012 ; vol.5, no.1, pp. 100-107 , Jan 2012, doi: 10.1007/s11947-009-0267
7) L.Wang, Y.Zeng and T.Chen , " Back propagation neural network with adaptive differential evolution algorithm for time series forecasting," Expert Systems with Applications , vol.42, no.2, PP. 855-863, Feb.2015,doi: doi.org/10.1016/j.eswa.2014.08.018

8) Q.Dai and N.Liu," A two-phased and Ensemble scheme integrated Backpropagation algorithm," Neurocomputing , vol.94 ,pp. 1124-1135 , NOV.2014 , doi: 10.1016/j.asoc.2014.08.012

9) M. S. Al_Duais, & F. S .Mohamad , "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, Dec.2016 , 10.17485/ijst/2016/v9i46/9175

10) E.Noersasongko , F.T.Julfia, R.A.Syukur.Pramunendar and C. A Supriyanto ,"tourism arrival forecasting using genetic algorithm based neural network," Indian Journal of Science and Technology , vol.;9 .no.4 , pp. 1-55 , Jan.2016, **d0i:**10.17485/ijst/2016/v9i4/78722

11) Y. Liu., Z., Li, D. Yang, K.S.Mohamed, L.Wang, and W.Wu, " Convergence of batch gradient learning algorithm with smoothing L 1/2 regularization for Sigma–Pi–Sigma neural networks", Neurocomputing, vol. , no. , pp.333-341 , Mar. 2015 , doi. 10.1016/j.neucom.2014.09.031

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, vol,13, pp. 174-79,Apr. 2015, doi: 10.1016/j.neucom.2014.11.039

13) C. Kaensar, "Analysis on the Parameter of Back Propagation Algorithm with Three Weight Adjustment Structure for Hand Written Digit Recognition," *2013 10th International Conference on Service Systems and Service Management*, 2013, pp. 18-22, doi: 10.1109/ICSSSM.2013.6602610.

14) D.Sha and V.B.Bajic, "An optimized recursive learning algorithm for three-layer feedforward neural networks for mimo nonlinear system identifications," Intelligent Automation & Soft Computing, vol. 17, no.2, PP.133-147, Jan.2015, .

15) M.S Al_Duais, A. Yaakub, N. Yusoff and F. 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 , vol.9, no.3, pp.189-200,Jan. 2015, doi: 10.19026/rjaset.9.1394.

16) S.Sureerattanan , N. Phien ,N.Sureerattanan and N.Mastorakis.,"The optimal multi-layer structure of back propagation networks," Proceedings of the 7th WSEAS International Conference on Neural Networks, Croatia, pp.108-13 , jun 2006

17) ZX.Yang, GS.Zhao, HJ.Rong, J.Yang ,"Adaptive backstepping control for magnetic bearing system via feedforward networks with random hidden nodes," Neurocomputing , vol. 174, pp.109-120, Jan.2016 , doi: 10.1016/j.neucom.2014.12.116

18) Y. Chao and R. Xu, "Adaptation Learning Rate Algorithm of Feed-Forward Neural Networks," *2009 International Conference on Information Engineering and Computer Science*, 2009, pp. 1-3, doi: 10.1109/ICIECS.2009.5366919.

19) M. S. Al Duais , F. S Mohamad," Dynamically-adaptive Weight in Batch Back Propagation Algorithm via Dynamic Training Rate for Speedup and Accuracy Training ," Journal Telecommunications and Information Technology , vol. 4, no.4 , PP.82-89, Jun. 2017, *doi.org/10.26636/jtit.2017.113017*,

20) V.M. Joy and S. Krishnakumar," Optimal design of adaptive power scheduling using modified ant colony optimization algorithm," International Journal of Electrical and Computer Engineering , Vol. 10, no.1 , pp738-745, Feb 2020, doi: 10.11591/ijece.v10i1

21) R,Malemnganbi, Benjamin A. Shimray, "An integrated multiple layer perceptron-genetic algorithm decision support system for photovoltaic power plant site selection ," International Journal of Electrical and Computer Engineering , Vol. 12, no. 2, pp. 1965~1972 April.2022, doi: 10.11591/ijece.v12i2.pp1965-1972.

22) 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," International Journal of Database Theory and Application, vol.4, no.2, pp.65-76 Jun.2011,.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8348528

23) 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, vol.7 , no.4, pp. 124-130 , Des.2018 , doi: 10.11591/ijai.v7.i4.pp170-178.

24) 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. Vol.43, no.1 pp. 32-41 , Jan.2013, doi. 0.1016/j.compbiomed.2012.10.006

25) M.S 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, Feb.2020. DOI: 10.5815/ijisa.2020.01.05.