I.J. Intelligent Systems and Applications, **2020, 1, 43-54**
Published Online February 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2020.01.05

Enhancement Processing Time and Accuracy Training via Significant Parameters in the Batch BP Algorithm

Mohammed Sarhan Al _ Duais
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
E-mail: Sarhan2w@gmail.com

Fatma Susilawati Mohamad
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
E-mail: Fatma@ unisza.edu.my

Mumtazimah Mohamad
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
E-mail: Mumtazimah@ unisza.edu.my

Mohd Nizam Husen
Malaysian Institute of Information Technology , Universiti Kuala Lumpur (UniKL), Malaysia
E-mail: mnizam@unikl.edu.my

Received: 29 January 2019; Accepted: 22 November 2019; Published: 08 February 2020

***Abstract*—**The batch back prorogation algorithm is anew style for weight updating. The drawback of the BBP algorithm is its slow learning rate and easy convergence to the local minimum. The learning rate and momentum factor are the are the most significant parameter for increasing the efficiency of the BBP algorithm. We created the dynamic learning rate and dynamic momentum factor for increasing the efficiency of the algorithm. We used several data set for testing the effects of the dynamic learning rate and dynamic momentum factor that we created in this paper. All the experiments for both algorithms were performed on Matlab 2016 a. The stop training was determined ten power -5. The average accuracy training is 0.9909 and average processing time improved of dynamic algorithm is 430 times faster than the BBP algorithm. From the experimental results, the dynamic algorithm provides superior performance in terms of faster training with highest accuracy training compared to the manual algorithm. The dynamic parameters which created in this paper helped the algorithm to escape the local minimum and eliminate training saturation, thereby reducing training time and the number of epochs. The dynamic algorithm was achieving a superior level of performance compared with existing works (latest studies).

***Index Terms*—**Enhancement processing time, accuracy Training, Dynamic momentum factor, Dynamic learning rate, Batch Back-propagation algorithm.

I. Introduction

An artificial neural network (ANN) is a mathematical model inspired by biological node systems [1,2]. This is a description that offers great flexibility in modeling quantitative methods. The ANN has a strong ability to complete tasks many and is considered an active tool for training and classification[3-6]. An ANN provides a supervised learning algorithm which implements a nonlinear model within [0, 1] or [-1, 1], depending on the activation function [7, 8]. An ANN involves parallel processing, which consists of several parameters which need to be adjusted to minimize error training

Heuristic techniques are the most current methods for improving the training of the BBP algorithm. These methods include learning rate and momentum factor; these are significant parameters for adjusting the weights in training [9-10] . In general, the values of learning rate and momentum factor are fixed and are chosen from the interval [0, 1]. Manually set values for each training rate and momentum factor do not help the BBP algorithm escape from local minima or to meet the requirements of speeding up the training of the BBP algorithm.

Generally, there are two techniques for selecting the values for each learning rate and momentum factor. The first is set to be a small constant value from interval [0,1], the second the selected series value from[0,1], [11,12]. Therefore, one of the requirements for speeding up the BBP algorithm is adaptive learning rate and momentum

factor together [13,14]. 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 [15,16] .On 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 [17,18]. The big values or small values are not suitable for smooth training. It is difficult to select the best or suitable values in training [19, 20].

The heuristic method is a current method for improving the BP algorithm, which covers two parameter training rate and momentum factor. The weaknesses, the values of training, rate and momentum are selected manual values. On the other hand, improves BP algorithm through creating dynamic training rate. But the weaknesses of these dynamic training depend on the initial value of training rate. Manually set values for learning rate and momentum factor do not help the BBP algorithm avoid local minima or meet the requirements of increasing training speed and difficult to choose a suitable value for learning rate and momentum factor. One way of avoiding localminima and speeding up the training of the BBP algorithm is to use a large value for training rate.

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. Most previous studies have tried to escape the local minimum through adaptive learning rates or momentum factors to improve the BBP algorithm. The persisting major problem, though, is the accumulation of weight in the BBP, which decreases accuracy and increases the time of training.

To fill the gap, we need to avoid the gross weight training, through creating dynamic parameter each learning rate and momentum factor with boundary to control the weight updated.

The remainder of this paper is organized as follows: Section II, literature review ; Section III, theoretically of training batch of back-Propagation algorithm; Section IV materials and method; Section V Created the dynamic learning rate and momentum factor, Section VI Experimental results; Section VII Discussion to validate the performance of improve algorithm; Section VIII Evaluation of the Performance of Improved Batch BP algorithm Section XI Conlusion.

II. Related Works

Currently, works have been carried out, such as author, In [21] Improved back propagation Algorithm by adaptive the momentum factor by dynamic function. The dynamic function which proposed is depend on the two manual factors parameters. The learning rate selected by manual value to control the weight updated. Each factor and learning rate selected different value for each data set. To validate, this study through compares the performance of improved algorithm with BP algorithm.The proposed algorithm gave superior training algorithm than other with whole data set. In [22] improved the BP algorithm through two techniques, the training rate and momentum factor, values of training rate were fixed at different values. The idea of this study is to set the value of training, the rate to be large initially, and then to look at the value of error training after iteration. If the error (e) training is increased, the fit produced changes the value of training, rate multiplied by less than one and then recalculated in the original direction. If the iteration error can be reduced, this integration is effective. Therefore, by changing the training rate multiplied by a constant greater than one, the next iteration is calculated continuously. In[23] compared several techniques such as BP with momentum, BP with the adaptive learning rate, BP with adaptive learning rate and momentum, Polak-Ribikre CGA, Powell-Beale CGA, scaled CGA, resilient BP (RBP) conjugate gradient algorithm (CGA), and Fletcher-Reeves. The epochs of training are fixed with different values. The BP algorithm with adaptive learning rate and momentum gave superior accuracy training at 1000 epochs.

III. Theoretically of Training Batch of Back-Propagation Algorithm

In this section we will foucas on the theoretivall of BBP algorithm which cover the mathematical framework with significant parameters such as learning rate and momentum factor, and the gradient descent method. The BBP algorithm was designed to train models to achieve a balance between the actual data and the target correctly in the input patterns; this is done using the gradient descent method to minimize the error training of the BBP algorithm.

A. Mathematical Framework BBP Algorithm with γ and μ

The training BP algorithm is consist of three stages , namely ,forward propagation , in the feed forward phase , 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. 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. The training of *BP* algorithm shown in the figure below

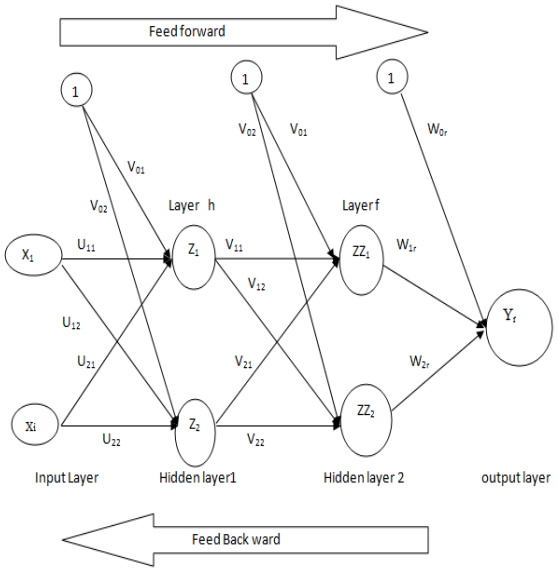

Fig.1. Batch BP algorithm stractur

The BP Algorithm with Training rate and Momentum term. The weight updates between the neuron k from output layer and neuron j from hidden layer as follows All steps training of the BP algorithm with training rate and momentum. Any neuron k from output layer and neuron j from hidden layer, the weight is adjusted between them as follows:

$$\Delta w_{jr}(t+1) = w(t) - \gamma \frac{\partial E}{w_{jk}(t)} + \mu \Delta w_{jr}(t-1) \qquad (1)$$

All steps training of the BP algorithm with training rate and momentum.

Forward Pass Phase: In this stage, the weight calculates layer by layer until reaching the end layer or output layer of the end layer under affected the activation function. Every variable $f'(x)$ receives an input with initial weight and broadcasts to next the layer $L_1,...,L_i$. The input signal L_{-inh} for each hidden unit is the sum of each input signal x_j. Each input layer as follows:

$$L_{-inh} = u_{0h} + \sum_{i=1}^{k} u_{ik} x_i$$

Compute the output of first hidden layer of L_h

$$L_h = f(L_{-inh}) \qquad (2)$$

Send the result to LL_j (LL_j $j = 1,2,...,p$) and Calculate the input signal

$$LL_j = v_{0h} + \sum_{i=1}^{i} v_{hj} L_h \qquad (3)$$

Calculate the output layer of second hidden LL

$$LL_j = f(LL_{-inh}) \qquad (4)$$

Send the output LL to input Y_{-inr} and then get the input layer of Y_{-inr}

$$Y_{-inh} = w_{0h} + \sum_{i=1}^{k} LL_J w_{jr} \qquad (5)$$

Compute the output layer of Y_r

$$Y_r = f(Y_{-inr})$$

Backward Pass Phase: when the weight reaches to the output of the last hidden layer to end step then the start all steps as follow

$$e_r = \sum_{r=1}^{n} (t_r - Y_r) \qquad (6)$$

The local gradient of output layer Y_r at neuron r is defined by below equation

$$\delta_r = e_r f(Y_{-inr}), \; f'(Y_{-inr}) = Y_{-inr} f(1 - Y_{-inr}) \qquad (7)$$

Compute weight correction term (update later)

$$\Delta w_{jr} = -\gamma \delta_r LL_j + \mu \Delta w_{jr}(t-1) \qquad (8)$$

Calculate, bias correction term (update w_{0r} later

$$\Delta w_{0r} = -\gamma \delta_{0r} + \mu \Delta w_{0r}(t-1) \qquad (9)$$

Send δ_r to hidden layer , each hidden unite $LL_j, j = 1,...,p)$

The single error or error back propagation from units in layer above to given as below

$$\delta_{-inj} = \sum_{r=1}^{m} \delta_r W_{jr} \qquad (10)$$

The local gradient of hidden layer LL_j at neuron j is defined by below equation

$$\delta_j = -\delta_{-inj} f'(LL_{-inJ}) \quad (11)$$

Compute weight correction term (update v_{hj} later)

$$\Delta v_{hj}(t+1) = -\gamma\delta_j L_h + \mu\Delta w_{hj}(t-1) \quad (12)$$

Calculate, bias correction (update v_{0j} later)

$$\Delta v_{jr}(t+1) = -\gamma\delta_j + \mu\Delta v_{0r}(t-1) \quad (13)$$

Send δ_j to first layer h=1,…,q).

The single error or error backpropagation is given as below

$$\delta_{-inh} = \sum_{j=1}^{b} \delta_j v_{hl} \quad (14)$$

The local gradient of hidden layer L_h compute as follow

$$\delta_h = \delta_{-inh} f(L_{-inh}),\ f'(Y_{-inr}) = Y_{-inr} f(1-Y_{-inr}) \quad (15)$$

Compute weight correction (update u_{ih} later)

$$\Delta u_{ih} = -\gamma\delta_h x_i + \mu\Delta u_{ih}(t-1) \quad (16)$$

Calculate bias weight corrective (used to update u_{0h} later)

$$\Delta u_{0h} = -\gamma\delta_h + \mu\Delta u_{0h}(t-1) \quad (17)$$

Update Weight Phase: the weight is update of the output layer $j = (0,1,2,...,p; r = 1...,m)$ as follows

$$w_{jr}(t+1) = w(t) - \gamma\Delta w_{jr}(t) + \mu\Delta w_{jr}(t-1) \quad (18)$$

For Bias

$$w_{0r}(t+1) = w(t) - \gamma\delta_j + \mu\Delta w_{jr}(t-1) \quad (19)$$

For each hidden layer $(LL_J h = 0,...,q;\ j = 1,...p)$

$$v_{hj}(t+1) = v_{hj}(t) - \gamma\Delta w_{hj}(t) + \mu\Delta w_{hj}(t-1) \quad (20)$$

For Bias

$$v_{0j}(t+1) = v_{0j}(t) - \gamma\Delta w_{0j}(t) + \mu\Delta w_{0j}(t-1) \quad (21)$$

For each hidden layer $L_h (i = 0,...,n; h = 1,...,q)$

$$u_{ih}(t+1) = u_{ih}(t) - \gamma\Delta u_{ih}(t) + \mu\Delta u_{ih}(t-1) \quad (22)$$

For Bias

$$u_{0h}(t+1) = u_{0h}(t) - \gamma\Delta u_{0h}(t) + \mu\Delta u_{0h}(t-1) \quad (23)$$

The flowchart of the BP algorithm with trail value for each training rate and

B. Gradient Descent Backpropagation Algorithm

The gradient descent method is one of the most popular methods used to adjust the weight for minimizing the training error in the BP algorithm [24, 25] Unfortunately, the gradient descent method is not guaranteed to find the global error [26, 27] .It may, therefore, result in leading to a local minimum, as the gradient descent method is not powerful enough to adjust for the weight, apart from limited changes [28] .The batch gradient descent method computes gradients using pattern training and updates the weight in the final stages of training [29]. This method focuses on modifying one parameter or more than one as adaptive of activation function or modifying the gain. The value of the gain is seen in the effect of the behavior of the convergence algorithm [30]. The author [31] proposed a new algorithm through the modified gain in the activation function. The proposed algorithm gives superior training compared with the BBP an algorithm.

Gradient descent ΔE taken partial of the differentiate of error with respect the weight w_{jk} in each layer. *ANN* which is consist one input, one hidden layer and one output also focus for one neuron as k from output layer and one neuron j from hidden, as follows

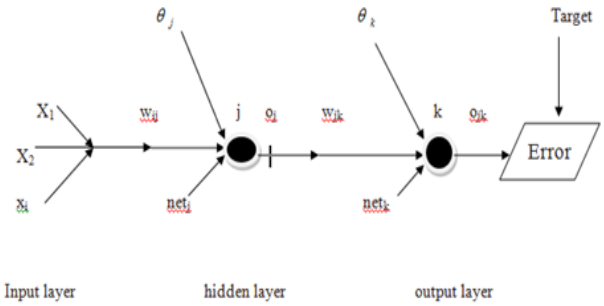

Fig.2. Gradient Descent for one Hidden Layer

Let the error training is given by $E = \frac{1}{2}\sum_{k=1}^{k}(t_k - o_k)^2$, activation function of training is sigmoid function $f(x) = \frac{1}{1+Exp(-x)}$. When the gain equals is one. The weight update between the neuron K, from output and neuron j from hidden layer as follow

$$\Delta w_{jk}(new) = \Delta w_{jk}(old) - \gamma \Delta w_{jk} + \mu \Delta v_{hj}(t-1) \quad (24)$$

The gradient between neuron K from output layer and neuron j from hidden layer

$$\Delta w_{jk} = \frac{\partial E}{\partial W_{jk}} = \frac{\partial E}{\partial o_k} \cdot \frac{\partial o_{jk}}{\partial net_k} \cdot \frac{\partial net_k}{w_{jk}} \quad (25)$$

$$\frac{\partial E}{\partial o_k} = \frac{1}{2} \frac{\partial}{\partial o_k} (t_k - o_k)^2$$

$$\frac{\partial E}{\partial o_k} = -(t_k - o_k) \quad (26)$$

$$\frac{\partial o_{jk}}{\partial net_k} = \frac{\partial}{\partial net_k} \left(\frac{1}{1+e^{-x}} \right)$$

$$\frac{\partial o_{jk}}{\partial net_k} = o_k(1-o_k) \quad (27)$$

$$\frac{\partial net_k}{w_{jk}} = x_j \quad (28)$$

Insert equation 25, 26, 27 into equation (25) it becomes as follow

$$\Delta w_{jk} = -(t_k - o_k) o_k (1-o_k) x_j \quad (29)$$

Insert equation (29) into equation (24) to get a new equation for update the weight as below

$$\Delta w_{jk}(new) = \Delta w_{jk}(old) - \gamma (t_t - o_k) o_k (1-o_k) x_j \quad (30)$$

Put the $\delta_k = (t_k - o_k) o_k (1-o_k)$ then get the equation

$$\Delta w_{jk}(new) = w_{jk(old)} + \eta \delta_k x_j \quad (31)$$

For calculate the local gradient between neuron j from hidden layer and neuron i from input layer. In this case the weight update as follow

$$\Delta w_{ij}(t+1) = w_{ij}(t) - \gamma \Delta w_{ij}(t) \quad (32)$$

The gradient between neuron j from hidden layer and neuron i from input layer given as follow

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial net_{ij}} \cdot \frac{\partial net_{ij}}{\partial w_{ij}} \quad (33)$$

$$\frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j} \quad (34)$$

$$\frac{\partial E}{\partial o_j} = \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial o_j} \quad (35)$$

Insert equation (35) into equation (34) it becomes as below

$$\frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j} \quad (36)$$

$$\Delta w_{ij} = \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ij}} \quad (37)$$

$$\frac{\partial E}{\partial net_k} = \frac{\partial E}{\partial o_k} \cdot \frac{\partial o_k}{\partial net_k}$$

$$\frac{\partial E}{\partial net_k} = -(t_t - o_k) o_k (1-o_k) \quad (38)$$

Put $\delta_k = (t_t - o_k) o_k (1-o_k)$ then get

$$\frac{\partial E}{\partial net_k} = \delta_k \quad (39)$$

$$\frac{\partial net_k}{\partial o_j} = \frac{\partial}{\partial o_j} \left(\sum_{j=1}^{j} o_j w_{ij} + \theta_j \right)$$

$$\frac{\partial net_k}{\partial o_j} = w_{ij} \quad (40)$$

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial}{\partial net_j} \left(\frac{1}{1+e^{net_j}} \right)$$

$$\frac{\partial o_j}{\partial net_j} = f'(net_j) = \bar{o}_j (1-\bar{o}_j) \quad (41)$$

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left(\sum_{k=1}^{k} x_j w_{ij} + \theta_j \right)$$

$$\frac{\partial net_j}{\partial w_{ij}} = x_i \quad (42)$$

Insert equation (26, 27, 28) into equation (25) it becomes as below

$$\Delta w_{ij} = \frac{\partial E}{\partial W_{ij}} = -\sum_{k=1}^{k} (t_t - o_k) o_k (1-o_k) w_{ij} \bar{o}_i (1-\bar{o}_j) x_i$$

From equation (29) replace $(t_t - o_k) o_k (1-o_k)$ by δ_k

$$\Delta w_{ij} = \frac{\partial E}{\partial W_{ij}} = -\sum_{k=1}^{k=K} \delta_k w_{ij} \bar{o}_i (1-\bar{o}_j) x_i \quad (43)$$

Put $\bar{\delta}_j = \sum_{j=1}^{j=i} \delta_k w_{ij} \bar{o}_i (1-\bar{o}_j)$ in (42) to get

$$\Delta w_{ji} = \frac{\partial E}{\partial W_{ij}} = \bar{\delta}_j x_i \quad (44)$$

Insert (43) into equation (32), then the weight update between neuron j from hidden layer and neuron i from input layer as below

$$\Delta W_{ij}(new) = w_{ij}(old) + \gamma \bar{\delta}_j x_i \quad (45)$$

IV. 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:

A. Data set

The data set is very important for verification of improved DBBPLM 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 data set to change to become nomination data set between [0,1] and divided into two set training set and testing set.

B. Neural Network Model

We propose an ANN model, which is a three-layer neural network that has an input layer, hidden layer, and output layer. The input layer is considered to be $\{x_1, x_2, \ldots, x_i\}$, 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. The output layer Y_r is made of one layer with one neuron. Three basis, two of them are used in the hidden layers and one in the output layer. Finally, the sigmoid function is employed as an activation function

V. Created the Dynamic the Prameters Learning Rate and Momentum Factor

The weight update between neuron k from the output layer and neuron j from the hidden layer (w_{jr}) in equqtion (1) which recorded as above.

To enhance the BBP algorithm, which is given by the Equation above (1) to avoid the local minima and to avoid saturation training. The learning rate should be sufficiently large to allow escape from the local minimum and to facilitate fast convergence to minimize error training . However, the value of the learning rate which is too large leads to fast training with an oscillation in error training. To ensure a stable learning BP algorithm, the learning rate must be small. Depend on above, we will create the dynamic training rate with boundary to keep the weight as moderate as follows:

$$\gamma_{dmic} = \frac{\sec h(sh(1-sh)}{2} + \exp(sh.e) \quad (46)$$

Where the $sech\theta$ is Hyperbolic functions, θ is angle, and the $sech\theta = 2/(e^{\theta} + e^{-\theta})$. The sh(1-sh) is slop the first derivative of the activation function of the out put of second hidden layer and (e) is the error training. We can see the value of (sh) is located between 0 and 1, the value of (sh) is a boundary. In the same way, the training error (e) is boundary value $0 \leq e \leq 1$. Each sh and e boundary function, that is lead to sh×e is boundary function, $0 \leq sh \times e \leq 1$. Depend on the above we can find the $1 \leq \leq 2.78$. the dynamic γ_{dmic} is boundary funcation.

The second significant parameter is momentum. To get smooth training and avoid inflation in the gross weight of the added values for momentum factor, the fitting producer through creating dynamic momentum factor and implicate the Depend above we can create the dynamic momentum factor μ_{dmic} as follow

$$\mu_{dmic} = \frac{1}{\frac{\sec h(sh(1-sh)}{2} + \exp(sh.e)} \quad (47)$$

We Insert the equation (46) and (47) into equation (1), then the weight is updated between any layer as below

$$w_{jk}(t+1) = w_{jk}(t) - [\frac{\sec h(sh(1-sh)}{2} + \exp(sh.e)] \frac{\partial E}{w_{jk}(t)} + \frac{1}{\frac{\sec h(sh(1-sh)}{2} + \exp(sh.e)} \Delta w_{jk}(t-1) \quad (48)$$

For the equation (48) the dynamic algorithm will be updated the weight under affected the dynamic learning rate and dynamic momentum factor.

VI. Experimental Results

The accuracy of training is calculated as follows [32,33]

Accuracy (%) $= \frac{1-\text{absolut}(T_i - O_i)}{UP-LW} * 100$ where UP and LW are the upper bound and lower bound of the activation function.

A. Experimental Results for the DBBPLM Algorithm Using the XOR Problem

Ten experiments were carried out using Matlab, and the average (AV) was taken for several criteria used for

measurement of the training performance. Ten experimental has been done for DBBPLM with XOR and then taken the average and S.D.The average and S.D of experimental results are tabulated in Table 1.

Table 1. Average the Performance of DBBPLM algorithm with XOR

	Time-sec	Epoch	MSE	Accuracy
Average	2.2463	1681	9.998E-05	0.9859
S.D	0.36127166	0	0	2.22E-16

From Table 1., the formulas proposed in equations (46 and 47) help the back-propagation algorithm to enhance the performance of the training. Whereas the average time training is t = 2.2463 seconds and the epoch is 1681 epoch. The smallest value of standard deviation S.D indicates the scatter of time or error training is very close for every experiment, whereas the S.D = 0.36127166 second. The accuracy rate is 0.9859.

B. Experiment result of the BBP algorithm with XORProblem

For the dynamic DBBPLM algorithm we run the dynamic algorithm 10 time with XOR problem and then take the averge for time , epoch and accuracy rate as follow

Table 2. Performance of BBP algorithm with XOR

Value of		times	MSE	Epoch
γ	μ			
0.1	0.004	52.4760	9.999e-05	67490
0.0001	0.0001	6530	0.5011	194345
0.004	0.95	5670	1.635e-04	153329
0.1	0.1	55.9070	9.999e-05	76190
0.2	0.2	31.0090	9.997e-05	37839
Average		2467.8784	0.1003127	105838.6
S.D		2978.0706	0.2003937	58416.301

From Table 2., the range of the training time is 31.0090.0090≤ t≤ 6530 seconds. We consider the value 31.0090second is as minimum training time and the value6530 second is as maximum training time. The best performance of the BP algorithm is achieved at γ = 0.2 and at μ =0.2 whereas the training time is 21.0090 seconds, the worst performance of the BP algorithm is achieved at γ = 0.0001 and μ =0.0001 for first, whereas the training time is 65300 second.

C. Experiments result of the DBBPLM algorithm with Balance- Train set

We implement the DBBPLM algorithm ten time with the balance data training set. and then the average was taken for several criteria. The experiment's result is tabulated in the Table 3.

Table 3. Average the Performance of DBBPLM algorithm with Iris- Train set

	Time-sec	Epoch	MSE	Accuracy
Average	3.3663	61	9.887E-05	0.992
S.D	1.55506733	13.169662	9.139E-07	0.0016697

From Table 3., the dynamic approach for training rate and momentum term reduce the time required for training and enhance the convergence of the time training. The average training time is 3.3663 seconds at an average epoch is 61 epochs. The average accuracy training is 0.992

D. Experiments of the BBP Algorithm with Balance-Train set

We tested the BBP algorithm with several optimum values. The experiments result recorded as table V with iris –training set

Table 4. Performance of BBP bbp algorithm with Iris- Train set

Value of		times	MSE	Epoch
γ	μ			
0.1	0.004	67.5020	9.7199e-05	323
0.01	0.25	61.5210	9.8827e-05	1129
0.004	0.95	220.0530	9.9736e-05	4361
0.005	0.09	153.8200	9.9950e-05	2993
0.005	0.001	179.8890	9.9960e-05	3535
Average		129.157	9.913E-05	2468.2
S.D		71.7656181	1.053E-06	1509.4044

From Table 4., the best performance of the *BP* algorithm was at γ = 0.01 and μ =0.25 whereas the time training is 61.5210 ≈ 52 second while the best performance of the *BP* algorithm is achieved at γ = 0.004 and μ =0.95 whereas the time training is $220.0530 \approx 220$ seconds. In addition, the range of the average training time is $61.5210 \leq t \leq 220.0530$seconds.

E. Experiments result of the DBBPLM Algorithm with Balance- Testing set

Ten experiments has been done with balance testing and then taken the average of several criteria. The experiment result is tabulated in the Table 5.

Table 5. Performance of DBBPLM with Iris- Testing set

	Time-sec	Epoch	MSE	Accuracy
Average	3.6801	88	9.821E-05	0.9906
S.D	0.98730577	20.813697	1.143E-06	0.0019355

From Table 5., the average training time is 3.6801 seconds at an average epoch is 88 epoch. The average S.D of time is 0.98730577. The accuracy rate is very highest about 0.9887.

F. Experiments result of the BBP Algorithm with Balance- Testing set

We tested the BBP algorithm with several optimum values. The experiments result recorded in the Table 6., with iris –testing set

From Table 6., the best performance of the BP algorithm was at γ = 0.1 and μ = 0.1, whereas the time training is 79.9910 ≈ 80 seconds, while the worst performance of the BBP algorithm is achieved at γ =0.0001 and μ =0.0001 whereas the time training is 5600 second. The range of the average training time is 79.9910≤ t ≤ 5600 seconds. The average S.D of time is greater than one.

Table 6. Performance of BBP Algorithms with Iris- Testing set

Value of		times	MSE	Epoch
γ	μ			
0.1	0.004	79.9910	5.5408e-05	2867
0.0001	0.0001	5600	0.2049	96800
0.005	0.09	4868	0.8193	7352
0.005	0.001	1.422e+3	5.0238e-05	39854
0.01	0.25	591.4530	5.0937e-05	14140
0.1	0.1	70.0050	5.6064e-05	1840
Average		2105.2415	0.1707354	27142.167
S.D		2267.46456	0.2995366	33674.904

G. Experiments result of the DBBPLM Algorithm with Iris.- Train set

Ten experimental has been done for DBBLM with balance data set and then taken the average and S.D.The average and S.D of experimental results are tabulated in Table 7.

Table 7. Average the Performance of DBBPLM Algorithm with Balance - Train set

	Time-sec	Epoch	MSE	Accuracy
Average	9.9023	77	2.9658e-04	100
S.D	0.52140945	0	0	0

From Table 7., the experiments result indicates the dynamic learning rate and momentum factor helps the BBP algorithm for reducing the time training. The average time is 9.9023 ≈10 seconds with 77 epoch. The average S.D of time is 0.52140945. .

H. Experiments result of the BP Algorithm with Balance - Train set

We test the manual algorithm at several values for each learning rate and momentum factor. The result are recoded in the Table 8.

Table 8. Performance of DBBPLM Algorithm with Balance - Train set

Value of		times	MSE	Epoch
γ	μ			
0.1	0.004	105.5750	9.996e-05	6882
0.004	0.95	96.5370	1.0000e-04	6176
0.005	0.09	1.85e+3	9.99e-05	124715
0.005	0.001	4963	1.185e-04	117460
0.1	0.1	93.0560	9.98e-05	6289
0.2	0.2	76.7300	9.91e-05	9554
Average		1197.483	0.0001029	45179.333
S.D		1197.483	0.0001029	45179.333

From Table 8., the range of the training time is 76.7300≤ t ≤ 105.5750 seconds, the range of training time is very widely.

I. Experiments result of the DBBPLM Algorithm with iris.- Testing set

Ten experimental has been done for DBBPLM with Balance –Testing set and then taken the average and S.D.The average and S.D of experimental results are tabulated in Table 9.

Table 9. Average the Performance of DBBPLM Algorithm with Balance - Testing set

	Time-sec	Epoch	Accuracy
Average	6.482	76	0.9860
S.D	0.3571277	0	0

From Table 9., the average time is 6.482 6 seconds with 76 epochs. The accuracy rate is 0.9860, the accuracy rate is the nearest one

J. Experiments result of the manually BBP Algorithm with Iris.- Testing set

We test the manual algorithm at several optimum values for each learning rate and momentum factor . The result are recoded in the Table 10.

Table 10. Performance of BBP Algorithm with Balance - Testing set

Value of		Time-S	MSE	Epoch
γ	μ			
0.1	0.004	81.9550	2.772e-4	11490
.0001	0.0001	4976	0.0077	369071
0.004	0.95	3693	0.0077	152912
0.005	0.09	3546	5.35e-04	129706
0.005	0.001	1.97e+03	2.68e-04	273293
0.1	0.1	94.9780	2.76e-04	9833
0.2	0.2	75.6650	2.979e-04	5204
Average		2062.514	0.0024363	135929
S.D		1893.50603	0.0033302	132031

From Table 10., the range of the average training time is in the interval $75.6650 \leq t \leq 4976$ seconds this means the range of time training is wide. The best performance of the BP algorithm is achieved at γ =0.2 and μ =0.2 the worst performance of the BP algorithm is achieved at γ = 0.0001 and μ =0.0001. The BBP algorithm suffers the highest saturation at γ = 0.0001 and μ =0.0001.

VII. Discussion to Validate the Performance of Improved Algorithm

To validate the efficiency of the improved algorithm, through comparing the performance of the DBBPML algorithm with the performance of the batch BP algorithm based on certain criteria[34, 35].We calculate the speed up training using the following formula [36].

$$\text{Proccessing Time}=\frac{\text{Execution time of } BBP\ algorithm}{\text{Execution time of } DBBPLM\ algorithm}$$

In this part we will compare the performance for each DBBPLM dynamic algorithm which listed in the from Table.1, 3,5,7,9 with the performance of manually algorithm wich listed in the Tables 2,4,6,8,10. The comparission result are tabulated in Table 11.

Table 11. Processing Time Improved the DBBPLM Algorithm versus BP with first structure

	Dynamic (DBBPLM)algorithm		A manually BBP algorithm		
	AV Time	AV Epoch	AV Time	AV Epoch	Processing Time Improved
XOR	2.2463	1681	2467.8784	105839	1098.641499
Balance Training Balance Testing	9.9023	77	1197.483	45179	120.929784
	6.482	76	2062.514	135930	318.1909904
Iris Training Iris Testing	3.3663	61	129.157	2468	38.367644
	3.6801	88	2105.2415	27142	572.0609494

From Table 11., it is evident that the dynamic algorithm provides superior performance over the Manually algorithm for all datasets. The range of the training time of the Dynamic algorithm is $2.2463 \leq t \leq 9.9023$ s; this is a narrow interval, meaning that the dynamic parameters which created help the algorithm to reaches the global minimum in a short time and with few epochs. The range of training times of the BBP algorithm is $129.157 \leq t \leq 2105$ s; this is a wide interval, meaning that the BBP algorithm has a long training time and a high level of training saturation. The dynamic algorithm is $2360.166 \approx 2360$ times faster than the BBP algorithm at its maximum.

Easily we can see in the end Column in the table above, dynamic algorithm which created in this paper gave faster training than manual algorithm. For example the dynamic algorithm is 1098.641499se 1099 time faster than manual algorithm at its maximum time training. Also dynamic algorithm which created in this paper gave faster training than manually algorithm. For example the dynamic algorithm is 38.367644se 38 time faster than manual algorithm at its minimum time training

From Figure 3 Obviously, we can see the dynamic algorithm taken a short time to reach the glob minimum, for all data set while the Manual algorithm takes a long time to reach a global minimum. The dynamic algorithm created in this study is able to escape the local minimum and remove the saturation training.

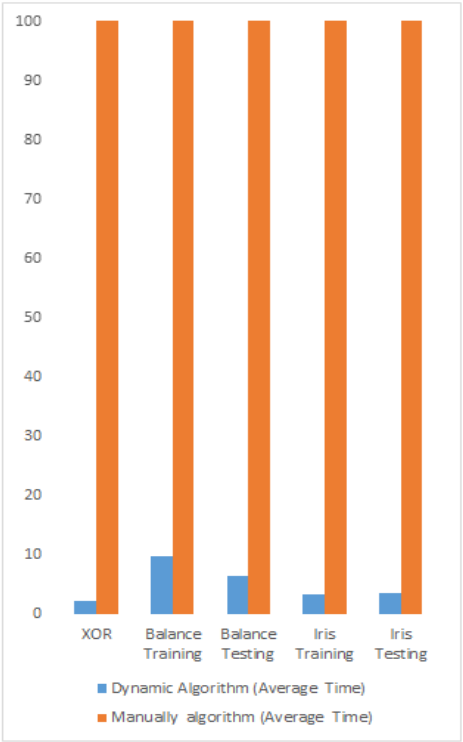

Fig.3. The performance of Dynamic algorithm versus Manually algorithm for time

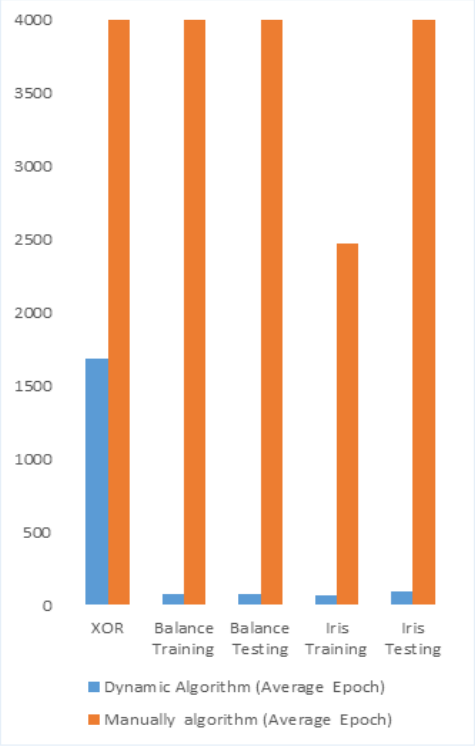

Fig.4. The performance of Dynamic algorithm versus Manually algorithm for epoch

From Figure 4 it is evident that the dynamic algorithm which created in this study provides superior performance over the Manually algorithm for all datasets. The dynamic algorithm taken few epochs to reach to the global minimum while the Manual algorithm takes a big value of epochs to reach the global minimum.

VIII. Evaluation of the Performance of Improved Batch BP Algorithm

To evaluated the performances of the improved batch BP algorithm for speeding up training the performances of the improved batch BP algorithm are compared to previous research works such, Hameed *et al* (2016) and Azami, (2015), the stop training for both study determined by number of iteration, the number iteration set by 2000 iteration and 1000 iteration respectively. For the purpose of the comparison between the results of this study and the previous works, the fit is re-run again with different stop training values according to previous work. The performance of the improved algorithm which proposed in this study give superior performance than exists works.

IX. Conclusion

We have introduced the dynamic DBBPLM algorithm, which trains by creating a dynamic function for each the learning rate and momentum factor. Sevral dataset were used, all data are taken from the UCI Machine Learning Repository through the following link: http://mir.cs.umass.edu/ml/dataset.html. All datasets should be changed to become normalized in the interval [0, 1]. The dataset was divided into two sets, a training set and a testing set for validating the performance of the contribution. This function influences for the weight. Of the DBBPLM algorithm were carried out on Matlab 2012a with same goal error or limited error training. One of the main advantages of the dynamic training and dynamic momentum factor are reduces the training time, the training error, number of epochs and enhancement the accuracy of the training. The performance of dynamic DBBPLM algorithm which presented in this study gave superior performance compared with exists work.

References

[1] M. S. Al_Duais , F. S. 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.

[2] F.Ortega-Zamorano, J.M. Jerez, I. Gómez, & L.Franco,"Layer multiplexing FPGA implementation for deep back-propagation learning". Integrated Computer Aided Engineering , 24(2), PP.171-185,2017

[3] Y.Hou, & H.Zhao, "Handwritten Digit Recognition Based on Improved BP Neural Network," Proceedings of 2017 Chinese Intelligent Systems Conference.. CISC 2017 ,pp. 63-70, 2018.

[4] E. Ibrahim, Hussien, & Z.E.Mohamed, "Improving Error Back Propagation Algorithm by using Cross Entropy Error Function and Adaptive Learning rate," International Journal of Computer Applications, 161(8), PP.5-9, 2017.

[5] C.B.Khadse, M.A Chaudhari, & V.B.Borghate, "Conjugate gradient backpropagation based artificial neural network for real time power quality assessment, " Electrical Power and Energy Systems, 82, PP.197–206 ,2016.

[6] M. S. Al_Duais , F. S 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.

[7] Z. X Yang, G. S. Zhao, H. J. Rong & J.Yang, "Adaptive backstepping control for magnetic bearing system via feed forward networks with random hidden nodes," Neurocomputing, 174, PP.109–120, 2016.

[8] Li, Jie, R.Wang , T.Zhang , & X.Zhang, "Predication Photovoltaic Power Generation Using an Improved Hybrid Heuristic Method. Sixth International conference on information Science and Technology ,pp. 383-387, 2016.

[9] M.Sheikhan, M.A Arabi, &D. Gharavian, "Structure and weights optimization of a modified Elman network emotion classifier using hybrid computational intelligence algorithms: a comparative study," Connection Science, 27(4), PP.340-357, 2015, doi:10.1080/09540091.2015.1080224

[10] R. Karimi, F.Yousefi, M.Ghaedi & K.Dashtian," Back propagation artificial neural network and central composite design," Chemometrics and Intelligent Laboratory Systems, 159, 127–137,PP.2016

[11] R.Kalaivani, K.Sudhagar K, Lakshmi P, "Neural Network based Vibration Control for Vehicle Active Suspension System", Indian Journal of Science and Technology.9(1), 2016.

[12] S. X. Wu, D. L.Luo, Z. W. Zhou, J. H.Cai, & Y. X. Shi, "A kind of BP neural network algorithm based on grey interval," ,International Journal of Systems Science, 42(3) ,PP.389-96,2011.
[13] H.. Mo, J.Wang ,H. Niu, "Exponent back propagation neural network forecasting for financial cross-correlation relationship," Expert Systems with Applications , 53 , PP.106-1016 , 2016.
[14] l.v Kamble, ;D.R Pangavhane, & T.P Singh, " Improving the Performance of Back-Propagation Training Algorithm by Using ANN,"International Journal of Computer, Electrical, Automation, Control and Information Engineering, 9(1), PP.187- 192, 2015.
[15] J.M.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.
[16] X.Xue, Y.Pan, R. Jiang, & Y. Liu, "Optimizing Neural Network Classification by Using the Cuckoo Algorithm," 11 the International Conference on Natural Computation (ICNC) , PP.24-30,2015.
[17] L. He , Y.Bo, & G.Zhao," Speech-oriented negate-motion recognition," Proceedings of the 34th Chinese Control Conference ,PP.3553 -3558 , 2015.
[18] Y.Zhang, S.Zhao, & L.Tang, "Energy Consumption Prediction for Steelmaking Production Using PSO-based BP Neural Network,"*Congress on Evolutionary Computation (CEC) PP*.3207- 3214, 2016.
[19] Q.Abbas, F. Ahmad, & M. Imran, "Variable Learning rate Based Modification in Backpropagation algorithm (MBPA) of Artificial neural Network for Data Classification," *Science International*, *28*(3), 2016.
[20] V. P. S. Kirar, "Improving the Performance of Back-Propagation Training Algorithm by Using ANN," World Academy of Science, Engineering, and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 9(1) ,PP.187-192,2015
[21] A. A. Hameed , B. Karlik, & M. S. Salman, "Back-propagation algorithm with variable adaptive momentum", Knowle dge-Base d Systems, 114,PP.79–87, 2016.
[22] W. Zhang, Z. Li, W .Xu, H.Zhou , "A classifier of satellite signals based on the back-propagation neural network," In 8th International Congress on Image and Signal Processing (CISP), PP.1353-1357,2015.
[23] H. Azami , & J.Escudero, "A comparative study of breast cancer diagnosis based on neural network ensemble via improved training algorithms," Proceedings of 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) ,PP.2836-2839,2015.
[24] Leema, N., Nehemiah, H. K., & Kannan, A. (2016). Neural Network Classifier Optimization using Differential Evolution with Global Information and Back Propagation Algorithm for Clinical Datasets. Applied Soft Computing. doi:http://dx.doi.org/doi:10.1016/j.asoc.2016.08.001
[25] Kamble, L V; Pangavhane, D R; Singh, T P. (2015). Improving the Performance of Back-Propagation Training Algorithm by Using ANN. International Journal of Computer, Electrical, Automation, Control and Information Engineering, 9(1), 187- 192.
[26] Liew, S. S., Hani, M. K., & Bakhteri, R. (2016). An Optimized Second Order Stochastic Learning Algorithm for Neural Network Training.Neurocomputing. Neurocomputing, 186, 74 – 89.
[27] Andrade, A., Costa, M., Paolucci,, L., Braga, A., Pires, F., Ugrinowitsch, H., & Menzel, H.-J. (2015). A new training algorithm using artificial neural networks to classify gender-specific dynamic gait patterns. Computer Methods in Biomechanics and Biomedical Engineering, 4, 382-390. doi:10.1080/10255842.2013.803081
[28] Kirar, V. P. (2015). Improving the Performance of Back-Propagation Training Algorithm by Using. International Journal of, 9(1), 187-192.
[29] Yui, M., & Kojima, I. (2013). A Database-Hadoop Hybrid Approach to Scalable Machine Learning. IEEE International Congress on Big Data (pp. 1-8). IEEE.
[30] Hamid, N. A., Nawi, N. M., & Ghazali, R. (2012). The effect of adaptive gain and adaptive momentum in improving training time of gradient descent back propagation algorithm on classifation problems. International Scientific Conferenc. 15. ISC.
[31] Nawi, N. M., Khan, A., & Rehman, M. Z. (2013). A new levenberg marquardt based back propagation algorithm training with cuckoo search. The 4th International Conference on Electrical Engineering and Informatics (pp. 18-23). ICEEI.
[32] N.M. Nawi, N. A Hamid, R. S , R.Ransing, Ghazali & M. N. Salleh, "Enhancing Back Propagation NeuralNetwork Algorithm with Adaptive Gain on Classification Problems," networks, 4(2) ,2011.
[33] T.N .Bui & H. Hasegawa, "Training Artificial Neural Network using modification of Differential Evolution Algorithm," International journal of Machine Learning and computing, 5(1) , PP.1-6, 2015.
[34] 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, 9(46), 2016.
[35] M. S. Al Duais,& F.S.Mohamad, "Dynamically-adaptive Weight in Batch Back Propagation Algorithm via Dynamic Training Rate for Speedup and Accuracy Training. Training,"Journal of Telecommunications and Information Technology. 4, 2017, doi.org/10.26636/jtit.2017: 113017
[36] S. Scanzio , S.Cumani, R.Gemello, R., F.Mana &F.PLaface, "Parallel implementation of Neural Network training for speech recognition," Pattern Recognition Letters. Vol..1, no.11, PP.1302–1309,2010.

Authors' Proflies

Fatma Susilawati Mohamad is an Associate Professor in the Department of Information Technology, Faculty of Informatics and Computing, UniSZA. She services 18 years in academic field and has been actively involved in teaching, research, publications, professional services, consulting, and administration. She was also been appointed for holding various administrative positions in faculty and university level. Now she serves as a Deputy Dean at the Graduate School, UniSZA. She has graduated a Bachelor of Science (Information System) from Oklahoma City University, U.S.A in 1997. She has continued his studies at Universiti Kebangsaan Malaysia and obtained her Master of Information Technology (Computer Science) in 2004. She received his Ph.D. in Computer Science from Universiti Teknologi Malaysia in 2012. During her tenure as an academic staff, she has taught various courses for undergraduate studies

(Diploma and Bachelor Degree). Her experience gained not only from the aspect of academic, but also from the aspect of research, leadership, management and administration. She has published in more than 30 articles in various conferences and refereed journals where most of her publications came from Scopus and ISI indexed journals. She has led several research grants and her work is recognized in both national and international level. She was also invited to give talk in several conferences and technical workshops national and internationally. Currently, she has more than 10 postgraduate students under supervision where 5 of them were already graduated. Her specialization is in Computer Science focusing on Pattern Recognition and Image Processing.

Associate Professor Dr Fatma Susilawati Mohamad, Department of Information Technology, Faculty of Informatics and Computing Universiti Sultan Zainal Abidin (UniSZA), 21300 Gong Badak, Kuala Nerus, Terengganu, Telephone: 09-6688796, E-mail: fatma@unisza.edu.my, Website: http:/fik.unisza.edu.my/fatma

Mumtazimah Mohamad completed her B.Sc in Information Technology (2000) from University Kebangsaan Malaysia, Malaysia, and her MSc. in Software Engineering (2005) from University Putra Malaysia. She received her Ph.D (2014) from Universiti Malaysia Terengganu in Computer Science. She joined Kolej Ugama Sultan Zainal Abidin in 2000 as a lecturer. She then joins Universiti Sultan Zainal Abidin, Terengganu, Malaysia in 2006, where she is currently employed as the Deputy Dean for Academic & Postgraduate since

Mohammed Sarhan completed his B.S and Master from Sana'a university.He received her Ph.D (2018) from Universiti Sultan Zainal Abidin, Terengganu, Malaysia in Computer Science . The field is artificial intelligence more specially, improve the training of back propagation algorithm . He is an author in several scientific papers and publications in the field of Arterial intelligence. He is a reviewer for several journals which index under ISI journal or Scopus such, Engineering Applications of Artificial Intelligence, Neural Processing Letters (NEPL), Journal of Electrical Systems and Information Technology International Journal of Electrical and Computer Engineering (IJECE) and 2018 IEEE International Conference on Automatic Control and Intelligent Systems and many others. E-mail:Sarhan2w@gmail.com.

Mohd Nizam Husen received the B.S. degree in computing from University of Portsmouth, United Kingdom, in 1997 and the M.S. degree in information technology from Universiti Utara Malaysia, Malaysia, in 2007. He received the Ph.D. degree in computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2017.

From 1997 to 2008, he was a Lecturer with the Universiti Kuala Lumpur (UniKL), Malaysia. In 2008, he was promoted as a Senior Lecturer at the same institution. He is currently the Deputy Dean (Academic & Technology) at UniKL. He is the author of more than 30 research articles. His research interests include intelligent systems, visual attention, pattern recognition, and indoor localization. He is also a Reviewer of the IEEE Communications Letters and Springer Mobile Networks and Applications journal and many others.

How to cite this paper: 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

