Indian Journal of Science and Technology, *Vol 9(46), DOI: 10.17485/ijst/2016/v9i46/91755, December 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Review on Enhancements to Speed up Training of the Batch Back Propagation Algorithm

Mohammed Sarhan Al_Duais* and F. S. Mohamad

Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia;
Duais2025w@gmail.com, fatma@unisza.edu.my

Abstract

Objectives: The present review is focused on determining the efficiency of some of the parameters for enhancing the time and accuracy training in the batch back propagation (BP) algorithm. **Methods:** Researchers have used many methods, including heuristic methods, flat-spots, Fletcher -Powel and Quasi-Newton methods to enhance the BP algorithm for speeding up time training. The current heuristic method covers two techniques. The first focuses on choosing the suitable value for each training rate and momentum term, either together or individually. The second technique is to create a dynamic training rate with a penalty to avoid the local minimum. **Findings:** Slow training or fast training depends on the weight adjusted in the BP algorithm. The training rate and momentum are significant parameters for controlling the updated weight, but it is difficult to choose the suitable value to adjust the weight for improving the BP algorithm. If the weights are adjusted too small, the BP algorithm gives slow training; if the weight is over-adjusted, the BP algorithm gives faster training with an oscillating value of error training. The small or large adjustment of the weights is unsuitable for learning of the BP algorithm. Existing studies do not mention the relationship between the values of training rate and momentum term with gross weight. Gross weight leads to saturation training or reduction in training accuracy. This study suggests creating the dynamic training rate with boundary and momentum terms and then establishing the relationship between them to keep the weight adjusted moderate to avoid the gross the weight being updated. **Improvements:** This study will guide researchers to create a dynamic training rate and momentum term with an inverse relationship or boundary to escape gross weight training and maintain high accuracy training.

Keywords: Batch Back Propagation Algorithm, Local Minimum, Momentum Term, Speed Up Training, Training Rate

1. Introduction

Although the batch Back Propagation (BP) algorithm is a new style for updating weight, it is widely used in training algorithms as it is accurate for training[1–6]. It uses gradient descent to adjust the weight training, but gradient descent is not guaranteed to find the global minimum error because it may result in approaching the local minimum[7–10]. The problem of the BP algorithm is slow training and there are several parameters that need to be adjusted manually, such as momentum term, learning rate and training cycle with highest saturation training[11].

Despite the training rate and momentum term being significant parameters for controlling the updated weight, it is difficult to select the best value in training[12–14]. Generally, there are two techniques for selecting the values for each training rate and momentum coefficient. The first is set to be a small constant value from interval [0,1], the second the selected series value from [0,1][15]. The learning rate should be sufficiently large to allow for escaping the local minimum to facilitate fast convergence to minimize error training[16], but the biggest value of the training rate leads to fast training with oscillation error training. To ensure a stable learning BP algorithm, the learning rate must be small[17–20]. According to[21] the best value learning rate is between 0.05 and 0.25. Another way to increase the learning rate is to modify the BP algorithm by including the momentum term[22,23]. Therefore, one of the requirements for speeding up the back propagation algorithm is adaptive training rate and momentum term together[24,25].

* Author for correspondence

The range of the error threshold influences the training time; however, there are no theories which categorically determine the value of the error threshold. Research[26] determining the error threshold as 10^{-5} finds the convergence rate to be very slow, performing 500,000 epochs; on the other hand, authors[27] determining the error threshold as 10^{-4} also find the convergence rate to be very slow, performing 10,000 epochs. However, these problems have been discussed thoroughly by many researchers. More specifically, some studies focus on improving the BP algorithm through training rate, some of them by the training rate and momentum term modifying one another.

The remainder of this paper is organized as follows: Section 2 presents the stages of training the BP algorithm. Section 3 presents the related works on improving the batch BP algorithm. Section 4. ANN Model for training (Topology). Section 5 provides validate the efficiency of the improve BP algorithm. Section 6 evaluated the efficiency of the improve BP algorithm. Section 7 provides the proposed work. Section 8 concludes this study.

2. Stages of training for the batch BP algorithm

The training of the batch BP algorithm consists of three stages, namely the forward propagation, the feed forward phase, and updating the weight. Each input unit x_i receives an input signal x_i and broadcasts this signal to the next layer up to the end layer in the system. The backward propagation in this step starts when the output of the last hidden layer reaches the end step, then it starts the feedback. The last step is updating the weight. In the batch BP algorithm, the weight adjustment stage for all layers is adjusted simultaneously. The goal of the BP algorithm is to get the The goal of the BP algorithm is to get the minimum error training between the desired output and actual data[28] error training between the desired output and actual data[28].

2.1 Mathematical Framework

BP Algorithm with Training rate and Momentum term

The weight update between the neuron k from output layer and neuron j from hidden layer as follows.

$$\Delta w_{jk}(t+1) = w_{jk}(t) - \eta \frac{\partial E}{w_{jk}(t)} + \alpha \Delta w(t-1)$$

All steps of the BP algorithm with training rate and momentum shown as below

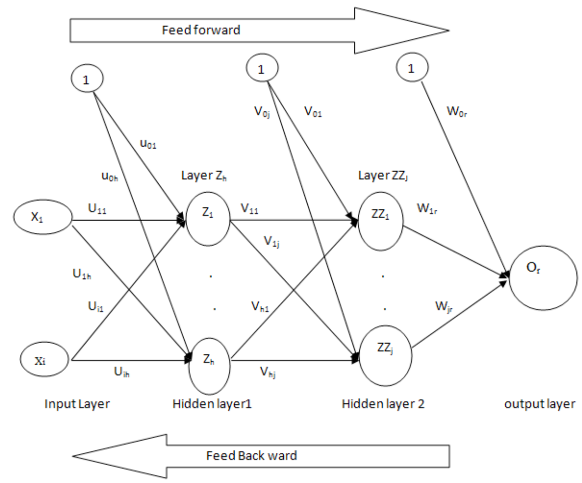

Figure 1. Training of back propagation.

Before presenting the BP algorithm, let us briefly define some of the notations used in the algorithm as follows.

Z_h	Where h, h=1,…,q is neuron for first layer
ZZ_k	where j, j=1,. . .,p is neuron for second hidden layer
O_r	Output layer for neuron r
u_{ih}	The weight for each neuron i and h from input layer and hidden layer respectively.
u_{0h}	The weight of the bias for neuron j
v_{hj}	The weight for each neuron h and j from hidden layer z and hidden layer zz respectively.
v_{0j}	Initial weight of the bias for neuron j
w_{jr}	The weight for each neuron j and r from hidden layer zz and output layer O respectively.
w_{0r}	Initial weight of the bias for neuron r from the output layer
Δw	The rate of change the weight between the current and the next iteration
η	Manually value of the training, rate
α	Manually value of the momentum factor
α_{dmc}	The dynamic of momentum term
η_{dmc}	The dynamic of training, rate
MSE	Mean Square error

2.1.1 Forward Pass Phase

In feed forward phase, each input unit x_i receives an input signal x_i and broadcasts this signal to each of the hidden unit is $z_1,...,z_i$. The input signal z_{-inh} for each hidden layer. All steps as follows :

Step 0: Read and initialize the weight.
Step 1: For each training pair, do steps 2-22.
Step 2: Read the number of the neuron in the hidden layer.
Step 3: Calculate input signal.
Step 4: Read the pattern from the data set, obtain the target, and limit the error.
Step 5: Calculates the input for the first layer of hidden

$$z_{-inh} = u_{oh} + \sum_{i=1}^{h} x_i u_{ih}$$

Step 6: Computes the output of the first layer Z_h

$$z_h = f(z_{-inp})$$

Step 7: send its output signal to all the units in the second hidden layer (z_{jj} j = 1, 2, ..., p)

Step 8: Calculates the input for second layer

$$zz_{-inj} = v_{oh} + \sum_{i=1}^{i} z_h v_{hj}$$

Step 9: Compute the output layer for the last layer

$$O_{-inr} = w_{0r} + \sum_{j=1}^{p} zz_j w_{jr}$$

Step 10: Compute the output layer signal

$$O_r = f(O_{-inr})$$

2.2.2 Backward Pass Phase

This step is starting when the output of the last hidden layer reach to end step then the start. The goal of the *BP* algorithm is to get the minimum error train between the desired output and actual data, as follows
Step 11: Calculates the error training

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

Step 12: Calculate the local gradient for the o_r

$$\delta_r = e_r f'(o_{-inr}), f'(o_{-inr}) = o_{-inr}(1-o_{-inr})$$

Step 13: Calculate weight correction term (update newest one later)

$$\Delta w_{jr} = -\eta\, \delta_r\, zz_j + \alpha \Delta w_{jr}\,(t-1)$$

Step 14: Calculate, bias correction term (update newest one w_{0r} later)

$$\Delta w_{0r} = -\eta\, \delta_r + \alpha \Delta w_{0r}\,(t-1)$$

Step 15: send δ_r to the hidden layer, each hidden unite (zz_j, j =1,...,p)

Step 16: Calculate input of the layer above to get

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

Step 17: Calculate the local gradient for hidden layer (zz_j) to get

$$\delta_j = \delta_{-inj} f'(zz\text{-}_{-inj})$$

Step 18: Calculate weight correction term (update newest one v_{hj} later)

$$\Delta v_{hj}(t+1) = -\eta\, \delta_j\, z_h + \alpha \Delta v_{hj}\,(t-1)$$

Step 19: Calculate, bias correction term (update newest one v_{oj} later)

$$\Delta v_{0j} = -\eta\, \delta_j\, z_h + \alpha \Delta v_{0j}\,(t-1)$$

Step20: Calculate the weighted input in layer above

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

Step 21: Calculate the local gradient of hidden layer z_h (expressed in terms of x_i

$$\delta_h = \delta\text{-}_{inh} f'(z_{-inh}), f'(z_{-inr}) = z_{-inh}(1-z_{-inh})$$

Step 22: Calculate weight correction (used to update u_{ih} later)

$$\Delta u_{hj}(t+1) = -\eta\, \delta_h\, x_i + \alpha \Delta u_{ih}\,(t-1)$$

Step 23: Calculate weight correction of bias (used to update u_{0h} later)

$$\Delta u_{0h} = -\eta\, \delta_j + \alpha u_{0h}\,(t-1)$$

Update Weight Phase

The weight adjustment stage, for all layers is adjusted simultaneously.

Step 24: Update the weight for each output layer or
(j = 0,1,2,...p; r =1..., m)
$$W_{jr}(t+1) = w_{jr}(t) - \eta \Delta w_{jr}(t) + \alpha \Delta w_{jr}(t-1)$$

Step 25: Updated the weight for bias w_{0r}
$$W_{0r}(t+1) = w_{0r}(t) - \eta \Delta w_{0r}(t) + \alpha \Delta w_{0r}(t-1)$$

Step 26: Updated the weight for layer (zz_{j} $_{h=0,\ldots,q;\, j=1,\ldots,p)}$
$$v_{hj}(t+1) = v_{hj}(t) - \eta \Delta v_{hj}(t) + \alpha \Delta v_{hj}(t-1)$$

Step 27: Updated the weight for bias
$$v_{0j}(t+1) = v_{0j}(t) - \eta \Delta w_{0j}(t) + \alpha \Delta w_{0j}(t-1)$$

Step 28: Updated the weight for hidden layer Z_h
Z_h(i= 0,...,n; h = 1,...,q)
$$u_{ih}(t+1) = u_{ih}(t) - \eta \Delta u_{ih} + \alpha \Delta v_{ih}(t+1)$$

Step 29: Updated the weight for bias
$$u_{0h}(t+1) = u_{0h}(t) - \eta \Delta u_{0h}(t) + \alpha \Delta u_{0h}(t-1)$$

3. Related Works on Improving the Batch BP Algorithm

In the batch BP algorithm, the values of the training rate and momentum coefficient are set to be constant values between [0, 1]. To enhance the BP algorithm, there are many methods such as flat-spot, conjugate gradient descent, and the heuristic method. All these methods have the same aim, but with a different technique to avoid the local minimum and to remove the saturation training. The gain is a direct effect on the slope of the activation function[29,30]. Along the same lines, some studies improved the BP algorithm by creating dynamic a training rate and momentum. These studies focused on modifying the training rate and momentum.

3.1 Improving the Heuristic Method by Adapting some Parameters

The heuristic method is widely used to improve the convergence rate of training the BP algorithm, and includes two parameters, namely the training rate and the momentum coefficient. The heuristic method is very important to increase the training algorithm. Both the training rate and momentum coefficient are significant to control the weight training[31].

This method is divided into two parts: the first part focuses on the adaptive the training rate, which selects the value for each training rate and momentum coefficient as a random and a manual value in the interval [0, 1]. The second part is focused on modifying the training rate η and momentum coefficient α through creating a mathematics formula or dynamic formulae. The relative work of the heuristic method is divided as follows.

3.1.1 Adapting either the Training Rate or Momentum with a Penalty

The gross weight kept the error training value large, which has effect on the training accuracy. However, the previous work in some studies introduced a way to solve this problem through proposing a penalty or a boundary to control the weight update, for example. In[32] proposed a new algorithm by adapting the square error function with a penalty for escaping the local minimum. The weight updated under the effect of the penalty. This study created a relationship between the training rate and the penalty. The training rate η is a fixed learning rate of 0.013 and the penalty parameter is set to 0.001. The results were compared to the Standard Back Propagation (SBP) algorithm. A relationship between the learning rate parameter and the penalty parameter is given to guarantee the convergence. But, each training rate and penalty set as manual value. In[33] improved the batch BPAP algorithm through proposing a dynamic training rate with a penalty. The penalty coefficient was $\lambda > 0$, and in this study $\lambda = 0.15$. The structure of the algorithm is 2:2:1 and it used the Sigmoid as an activation function. The weight updated in the batch BPAP algorithm, which was bounded during training. From the experiment, the result was that the BPAP reached the global minimum after spending 1000 iterations. In[34] presented the penalty to get the term proportion of the norm of the weight or to prove the boundedness of the weights in the network training process. The formula of penalty was proposed. The learning rate is set to be a small constant or an adaptive series using a mathematics formula. The weight chosen in [-0.5, 0.5], the initial training rate, is fixed to be a small constant at $\eta = 0.05$, and the penalty factor $\lambda = 0.001$. The results showed that the BPAP algorithm gave a better convergence compared to existing work. The conclusion, this kind of these techniques set the each training rate and penalty as manual value.

3.1.2 Adapting the Training Rate

The BP algorithm trains by a manual training rate, and in this case, the weight is updated manually based on the effect the value of the training rate. In this case, the BP algorithm suffers from slow training. One way to escape the local minimum and save training time in the BP algorithm is by using a large value of training rate η in the first training. On the contrary, a small value of training rate η leads to slow training, but a smaller value of the training rate η training rate leads to the BP algorithm having a slow convergence[35,36].

Many studies have tried to improve the BP algorithm through creating a dynamic training such as[37] presented the effect of input parameters with three different structures, e.g., the BP algorithm, BP algorithm with a momentum factor, and BP algorithm using conjugate gradient descent. A Sigmoid function was used as the activation function. The goals are to know how the different learning rates affect the recognition rate: therefore, the experiments start with a sample structure and different values of training rate. The second method uses various values of training rates and hidden nodes; and the third method uses a BP algorithm with conjugate gradient descent. From the experimental results, the BP algorithm with the momentum factor provides highest recognition rate, whereas the recognition rate is 0.99 in any other method. In[38] proposed a novel algorithm (NBPNN) that has a self-adaptive training rate; the experimental results showed that NBPNN gave a more accurate result than the BP algorithm. In[39] improved the BP algorithm: their structure consists of two layers and two output layers through the created dynamic training rate. The dynamic training rate depends on the current and previous values of the error training. In the same way, the BP algorithm can be improved by implicit the momentum on the dynamic training rate. The created training rate depends on the value of the training rate that is fixed at values η = 1.0, 0.4 and 1.4. The results show that the improved BP algorithm gave superior performance in training than the BP algorithm. In[40] provided the dynamic BP algorithm for training with a boundary. In this case, the weight is updated under the effect of this boundary. The Sigmoid function is used as the activation function. The boundary helps the BP algorithm to improve the speed up training and enhances the classification rate, whereas the value of classification correction is 91.1%. Many studies have tried to improve the BP algorithm through creating a dynamic training rate using a dynamic fractional function, or an exponential function such as[41,42].

3.1.3 Adapting the Momentum Coefficient

In the BP algorithm, the value of the momentum is set to be a constant in the interval [0, 1]. The BP algorithm with momentum has been created. By[43], improved the BP algorithm by creating dynamic momentum. In[44] author compared the BP algorithms, the first trained without the momentum coefficient and the second BPM algorithm, which trained with momentum. The BPM algorithm had faster convergence training than the BP algorithm, whereas the BPM algorithm reached the global minimum after 500 epochs, and the BP algorithm reached the global minimum after 5000 epochs. In conclusion, the momentum helped the BP algorithm to speed up training and get smooth curve training. By[45] investigated the applicability of the third term in the BP algorithm. The BP algorithms, which were enhanced by adding the new term, gave a better result for others that trained without the third term. Some studies have tried to overcome the local minimum and increase the speeding up of training for the BP algorithm through adding the third term with a long training rate and a momentum term. In[46] improved the back-propagation algorithm based on the adaptive momentum term. A new algorithm was tested using the two-dimensional XOR. The simulation results show that the new algorithm is better than the standard BP algorithm. On the other hand, some studies have focused on an adaptive training rate.

3.1.4 Adapting the Training Rate and Momentum

The training rate and momentum are significant parameters to control the weight update, this kind of algorithm has the ability to speed up the BP algorithm. This kind of technique improved the training of the BP algorithm through making for each training rate and momentum coefficient dynamic, such as[47] 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, 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[48] compare 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, at 10, 50, 100, 200, 500 and 1000. The back-propagation algorithm with adaptive learning rate and momentum gave superior accuracy training at 1000 epochs. In[49] enhanced the BP algorithm to obtain a good web selection algorithm for providing the most suitable service to meet the requirements of telecommunication technology for smart distribution grids and to improve the quality of the telecommunication service. To this end, improved BP algorithm by modifying the training rate and momentum. The value of the training rate selected depends on the ratio between the new error and the previous error training. The Sigmoid function used as activation function. The simulation results show an optimization of the training speed and oscillation reduction duration training. Some studies have improved the training by using the multi-step function, such as the training rate. In[50] created dynamic training in order to escape the local minimum and to speed up the training of the BP algorithm. This study created the dynamic training rate that consists of multi-steps. The value of the training rate η_1 is set at 1.14 and training rate η_2 set at 0.91; the momentum factor is set as 0.91. The Sigmoid function is used as an activation function with a single hidden layer. The Mean Square Error (MSE) was used for optimization and convergence of the network. From the experimental results, the improved algorithm was overall efficient, both in visual effect and quality. In[51] improved the BP algorithm by creating dynamic a training rate and momentum coefficient. The experiment resulted in the improvement of the algorithm, which had superior training than that of the SBP algorithm for the time it took to train, whereas the improved algorithm was 50 times faster than the SBP algorithm. The performance improved back propagation and dynamic back propagation compared to the BP algorithm[52,53].

3.2.1 Adapting the Gain

More specifically, some studies modified the BP algorithm to become faster training through adapting some parameters as a modified activation function in the BP algorithm[54]. The Δw_{jk} is affected on the slope value. The value of the gain and momentum have an influence on the efficiency of the training in the BP algorithm, so the[55] adapted each parameter's gain and momentum to remove the saturation. These problems of the BP algorithm easily convergence to the local minimum. In[56] improved the BP algorithm to avoid the local minimum to improve the performance of the training, through adapting the gain with the momentum and training rate. But each training rate and momentum were selected as a manual value from [0, 1], and depended on the kind of data set. The result demonstrates that the proposed algorithm gave superior training than that of the SBP algorithm.

3.3 Discussion of Previous Studies

The heuristic method is a current method for improving the BP algorithm, which covers two parameter training rate and momentum term. Literature review, focused on improving the BP algorithm through selecting the suitable value of each training rate and momentum term from, [0, 1] like study[47–49]. The weaknesses, the values of training, rate and momentum are manual values. On the other hand, each[32,34] improves BP algorithm through creating dynamic training rate. But the weaknesses of these dynamic training depend on the initial value of training rate and penalty. The[57] determines the problem of BP algorithm as, there are several parameters which need to be adjusted manually, such as momentum, learning rate, and training cycle and slow training with highest saturation training.

To fill the gap in the literature review, we need to avoid the gross weight training, through creating dynamic parameter with boundary to control the weight updated.

4. Artificial Neural Network ANN Model for Training (Topology)

Almost of previous studies were designed the model (Topology) of the training with a Sigmoid function as activation function.Sigmoide function is most commonly used in a training batch BP algorithm, training machine[58]. No any theory to determine the best structure. But, the number of hidden layers has affected the training *BP* algorithm[59]. One hidden layer suffices for many application[60]. Many studies used single hidden layer such[61]. Two hidden layers give suitable of the training and accurate of generalization, but the large number of hidden nodes an excessive number of nodes in the hidden layer may endanger the process[62]. From the previous study, the common structure is N:2h:1, where, h denoted the hidden layer and N is the number of patterns of the input training which effect by the kind of the data. By[63],test three structures, from experiment result the best structure which consist only one output layer.The model of neural network will define as { I, T,W,A},where I denoted the set of input node, T denoted the topology of NN, which cover the number hidden layer and number neurons, W denoted the set of the weight. A denoted of activation function.

5. Validate the Efficiency of the Improve the Batch BP Algorithm

This study belongs to the heuristic method. In this kind of method, the adaptive training rate and momentum coefficient are there to control the weight update. In order to verify or validate the efficiency of the proposed algorithm, simulations or experiments results on a data set will be performed, and then the performance of the proposed algorithm will be compared with the standard SBP algorithm against one MSE, time training, the number of iterations, and errors in training. From previous studies, the performance of the dynamic algorithm was compared to the BP algorithm, such as the studies[64–67]. From relative work, the performance of dynamic a lgorithm measurement by using some criteria such as average time- second, number of epoch or iteration, sum square error, accuracy training and speed convergence rate.

6. Evaluate the Efficiency of the Improve of BP Algorithm

The performances of the improved BP algorithm are compared to previous research works. The limited error or stop training must be same. From previous studied almost studies used one criteria such time training, epoch, and error training or accuracy of training. But those criteria are not robustness when the comparison between two algorithms with different data which, it has different size of the data set. This study puts the recommended use Standard Deviation (S.D) and coefficient correlation (C.v).

7. Proposed Works

From the previous work, we can propose alternative works, to avoid inflation in the gross (overshoot) weight and enhance the speeding up of training the BP algorithm as follows:

- Create the dynamic training rate with a boundary to control the weight update.
- Create a dynamic training rate and momentum coefficient that have a relationship with each other to keep the weight adjusted as moderate .Thus, we place an implicit momentum function in the training rate $\alpha_{dmc} = f(\eta_{dmc})$.

8. Conclusion

This paper presented techniques for improving the BP algorithm and determining the best structure (topology) of the BP algorithm. Many studies have sought to improve the performance of the batch BP algorithm. However, further research is required to improve the batch BP algorithm. In this paper, we recommended a new strategy to improve the BP algorithm, consisting of multiple steps to avoid inflation in gross weight training. The fitting is done by making a relationship between the dynamic training rate and the dynamic momentum. As a result, we placed an implicit dynamic momentum term in the dynamic training rate. This procedure kept the weights as moderate as possible (neither too small nor too large).

9. References

1. Rizwan JM, Krishnan PN, Karthikeyan R, Kumar SR. Multi layer perception type artificial neural network based traffic control. Indian Journal of Science and Technology. 2016 Feb 9; 9(5).
2. 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): 100–7.
3. Kostopoulos AE, Grapsa TN. Self-scaled conjugate gradient

training algorithms. Neurocomputing. 2009; 72(13):3000–19.
4. Zweiri YH, Whidborne JF, Seneviratne LD. A three term back propagation algorithm. Neurocomputing. 2003; 50:305–18.
5. Shukla A, Karmakar S. Identification of global minima of back-propagation neural network in the prediction of chaotic motion. International Journal of Computer Applications. 2015; 112(4).
6. Kalaivani R, Sudhagar K, Lakshmi P. Neural network based vibration control for vehicle active suspension system. Indian Journal of Science and Technology. 2016 Feb 7; 9(1).
7. Mo H, Wang J, Niu H. Exponent back propagation neural network forecasting for financial cross-correlation relationship. Expert Systems with Applications. 2016; 53:106–16.
8. Wang L, Zeng Y, Chen T. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Systems with Applications. 2015; 42(2):855–63.
9. Dai Q. Back-propagation with diversive curiosity: An automatic conversion from search stagnation to exploration. Applied Soft Computing. 2013; 13(1):483–95.
10. Li LK, Shao S, Yiu KF. A new optimization algorithm for single hidden layer feedforward neural network. Applied Soft Compting. 2013; 13(5):2857–62.
11. Malleswaran M, Vaidehi V, Sivasankari N. A novel approach to the integration of GPS and INS using recurrent neural networks with evolutionary optimization techniques. Aerospace Science and Technology. 2014; 32(1):169–79.
12. Abbas Q, Ahmad F, Imran M. VARIABLE learning rate based Modification In Backpropagation Algorithm (MBPA) of artificial neural network for data classification. Science International. 2016; 28(3):2369–78.
13. Kirar VPS. 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. 2015; 9(1):187–92.
14. Ramadhena C, Ibrahim AO, Sulaiman S. Weights Adjustment of two -term back- propagation network using adaptive and fixed learning methods. International Journal Advance Soft Computer Applied. 2013; 5(2) .
15. Wu SX, Luo DL, Zhou ZW, Cai JH, Shi YX. A kind of BP neural network algorithm based on grey interval. International Journal of Systems Science. 2011 ; 42(3):389–96.
16. Azami H, Sanei S, Mohammadi K. Improving the neural network training for face recognition using adaptive learning rate, resilient back propagation and conjugate gradient algorithm. Journal of Computer Applications. 2011 Nov; 34(2):22–6.
17. He L, Bo Y, Zhao G. Speech-oriented negative emotion recognition. Proceedings 34th Chinese in Control Conference (CCC); 2015 Jul 28. p. 3553–8.
18. Ge J, Sha J, Fang Y. An new back propagation algorithm with chaotic learning rate. International Conference on Software Engineering and Service Sciences; 2010 Jul 16. p. 404–7.
19. Iranmanesh S, Mahdavi MA. A differential adaptive learning rate method for back-propagation neural networks. World Academy of Science, Engineering and Technology. 2009 Mar 23; 50(1):285–8.
20. Zainuddin Z, Mahat N, Hassan YA. Improving the convergence of the backpropagation algorithm using local adaptive techniques. International Conference on Computational Intelligence; 2004 Dec. p. 173–6.
21. Rimer ME. Improving neural network classification training [Unpublished PhD thesis]. Brigham Young University; 2007.
22. Wen H, Xie W, Pei J. A pre-radical basis function with deep back propagation neural network research. Proceeding in 12th International Conference on Signal Processing (ICSP); 2014 Oct 19. p. 1489–94.
23. Zhang N. An online gradient method with momentum for two-layer feedforward neural networks. Applied Mathematics and Computation. 2009 Jun 15; 212(2):488–98.
24. Charif O, Omrani H, Trigano P. Optic modified backpropagation training algorithm for fast convergence of feedforward neural network. International Conference on Telecoounication Technology and Applications (CSIT); 2011. p. 132–7.
25. Liyi Z, Ting L, Jingyu Z. Analysis of momentum factor in neural network blind equalization algorithm. International Conference on Communications and Mobile Computing; 2009 Jan 6. p. 345–8.
26. Li Y, Fu Y, Li H, Zhang S-W. The improved training algorithm o back propagation neural network with self adaptive learing rate. International Conference on Computational Intelligence and Natural Computing CNC'09; Wuhan. 2009. p. 73–6.
27. Cheung C-C, Ng S-C, Lui AK, Xu SS. Enhanced two-phase method in fast learning algorithms. International Joint Conference on Neural Networks (IJCNN); 2010 Jul 18-23. p. 1–7.
28. Örkcü HH, Bal H. Comparing performances of back propagation and genetic algorithms. Expert Systems with Applications. 2011; 38(4):3705–9.
29. Moallem P. Improving back-propagation VIA an efficient combination of a saturation suppression method. Neural Network World. 2010; 20(2):207–23.
30. Xie J, Lin P, Liang H, Lu M. The improved rapid convergence algorithm of the connecting rights in the BP network. International Symposium on Knowledge Acquisition and Modeling; Wuhan. 2008. p. 819–22.
31. Shrestha SB, Song Q. Adaptive learning rate of spike prop based on weight convergence analysis. Neural Networks. 2015; 63:185–98.
32. Liu Y, Li Z, Yang D, Mohamed KS, Wang J, Wu W. Convergence of batch gradient leaning algorithm with smoothing L 1/2 regularization for sigma–pi–sigma neural networks. Neurocomputing. 2015; 151:333–41.

33. Shao H, Wang J, Liu L, Xu D, Bao W. Relaxed conditions forconvergence of batch BPAP for feed forward neural networks. Neurocomputing. 2015; 153:174–9.
34. Zhang H, Wu W, Yao M. Boundedness and convergence of bach back-propagation algorithm with penalty for feedward neural networks. Neurocomputing. 2012; 89:141–6.
35. Yang G, Qian FJ. A fast and efficient two-phase sequential learning algorithm. Applied Soft Computing. 2013; 25:129–38.
36. Nawi NM, Ranaing R, Ransing S. An improved learning algorithm besed on the Brroyden-Fletcher-Goldfarb-Shanno (BFGS) method for back propagation neural networks. Proceeding of the 6th Internatioal conference on Intelligent Systems Design and Applications; 2006 Oct 16-18. p. 152–7.
37. Kaensar C. Analysis on the parameter of back propagation algorithm with three weight adjustment structure for hand written digit recognition. 10th International Conference on Service Systems and Service Management. 2013. p. 18–22.
38. Shao H, Zheng G. Convergence analysis of a back-propagation algorithm with adaptive momentum. Neurocomputing. 2011 Feb 28; 74(5):749–52.
39. Sha D, Bajic VB. An optimized recursive learning algorithm for three-layer feedforward neural networks for mimo nonlinear system identifications. Intelligent Automation and Soft Computing. 2011; 17(2):133–47.
40. Feng Q, Daqi G. Dynamic learning algorithm of multi-layer perceptrons for letter recognition. International Joint Conference on in Neural Networks (IJCNN); 2013. p. 1-6.
41. Latifi N, Amiri A. A novel VSS-EBP algorithm based on adaptive variable learning rate. 2011 3rd International Conference on Computational Intelligence a Modelling and Simulation. Langkawi. 2011. p. 14–7.
42. Gong B. A novel learning algorithm of back-propagation neural network. Control, Automation and Systems Engineering. CASE 2009. IITA International Conference; 2009 Jul 11. p. 411–4.
43. Xinbo Z, Lili W. Handwritten digit recognition based on improved learning rate BP algorithm. International Conference in Information Engineering and Computer; 2010. p. 1–4.
44. Masmoudi MS, Klabi I, Masmoudi M. Performances improvement of back propagation algorithm applied to a lane following system. 2013 World Congress on in Computer and Information Technology (WCCIT); Sousse. 2013. p. 1–5.
45. Abdulkadir SJ, Shamsuddin SM, Sallehuddin R. Three term back propagation network for moisture prediction. International Conference on Clean and Green Energy; 2012. p. 103–7.
46. Zhixin S, Bingqing L. Research of improved back propagation neural network algorithm. Proceedings 12th IEEE International Conference on Communication Technology; Nanjing. 2010. p. 763–6.
47. Zhang W, Li Z, Xu W, Zhou H. A classifier of satellite signals based on the back-propagation neural network. 8th International Congress on Image and Signal Processing (CISP); 2015 Oct 14. p. 1353–7.
48. Azami H, Escudero J. A comparative study of breast cancer diagnosis based on neural network ensemble via improved training algorithms. Proceedings in 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2015. p. 2836–9.
49. Rui L, Xiong Y, Xiao K, Qiu X. BP neural network-based web service selection algorithm in the smart distribution grid. Proceedings 16th Asia-Pacific in Network Operations and Management Symposium (APNOMS); 2014. p. 1–4.
50. Yonghao D, Peng Z, Yuming S, Sanyuan Z. Improvements of coefficient learning in BPNN for image restoration restoration ICSAI. International Conference on Systems; Yantai. 2012. p. 2692–4.
51. Wang K, Zhuo L, Lu H, Guo H, Xu L, Zhang Y. An improved BP algorithm overout-of-order streams for big data. Proceedings 8th International ICST Conference in Communications and Networking; Guilin. 2013. p. 840–5.
52. Saki H, Tahmasbi A, Saltanian-Zadah H, Shokouhi SB. Fast opposite weight learning rules with application in breast cancer. Computers in biology and medicine. 2013; 43(1):32–41.
53. Scanzio S, Cumani S, Gemello R, Mana F, Laface P. Parallel implementation of artificial neural network training for speech recognition. Pattern Recognition Letters. 2010; 31(11):1302–9.
54. El-Melegy MT, Essai MH, Ali AA. Robust training of artificial feedforward neural network. Foundations of Computational Intelligence, Springer-Verlag: Berlin Heidelberg; 2009. p. 217–42.
55. Hamid NA, Nawi NM, Ghazali R. The effect of adaptive gain and adaptive momentum in improving training time of gradient descent back propagation algorithm on classifation problems. International Journal on Advanced Science, Engineering and Information Technology. 2011; 1(2):178–84.
56. Hamid NA, Nawi NM. Ghazali R, Salleh MN Accelerating learning performance of back propagation algorithm by using adaptive gain together with adaptive momentum and adaptive learning rate on classification problems. International Conference on Ubiquitous Computing and Multimedia Applications; 2011. p. 559–70.
57. Noersasongko E, Julfia FT, Syukur A, 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).
58. Wang J, Wu W, Zurada JM. Deterministic convergence of conjugate gradient method for feedforward neural networks. Neurocomputing. 2011 Jul 31; 74(14):2368–76.
59. Shao HM, Wu W, Liu LJ. Convergence of online gradient method with penalty for BP neural networks. Journal of Mathematical Research. 2010; 26(1):67–75.
60. Sureerattanan S, Phien HN, Sureerattanan N, Mastorakis E. The optimal multi-layer structure of back propagation networks. Proceedings of the 7th WSEAS International Conference on Neural Networks; Croatia. 2006. p. 108–13.

61. Yang ZX, Zhao GS, Rong HJ, Yang J. Adaptive backstepping control for magnetic bearing system via feedforward networks with random hidden nodes. Neurocomputing. 2016 Jan 22; 174:109–20.
62. Huang Y. Advances in artificial neural networks–methodological development and application. Algorithms. 2009 Aug 3; 2(3):973–1007.
63. Albarakati N, Kecman V. Fast neural network algorithm for solving classification tasks: Batch error back-propagation algorithm. Proceedings of IEEE in Southeastcon; Jacksonvill. 2013. p. 1–8.
64. Bui NT, Hasegawa H. Training artificial neural network using modification of differential evolution algorithm. International Journal of Machine Learning and computing. 2015; 5(1):1–6.
65. Chen C, Wang P, Xing J. Indoor location algorithm of back propagation neural network based on residual analysis. 33rd Control Conference (CCC); 2014. p. 467–71.
66. Dai Q, Ma Z, Xie Q. A two-phased and Ensemble scheme integrated Backpropagation algorithm. Applied Soft Computing. 2014; 24:1124–35.
67. Dai Q, Liu N. Alleviating the problem of local minima in backpropagation through competitive learning. Neurocomputing. 2012; 94:152–8.