

Research and Improvement on Error Back Propagation Neural Network and Learning Algorithm

Huang Yudong, Yang Gaoqiang, Hao Rui, Guan Jianhe

School of Information Engineering
China University of Geosciences (Beijing)
Beijing 100083, China

andyroddick6@foxmail.com, auyanggao@gmail.com, haorui2009@163.com, guanjh@cugb.edu.cn

Abstract—As the important subfield and the quintessence of Artificial Neural Network, BP Neural Network accelerated the development in this field. BP algorithm uses the steepest descent algorithm, thus there are two main shortcomings of slow convergence rate and easy to fall into local minimum. According to this, this paper researches on the basis of the standard BP algorithm and presented many improved back propagation algorithms from the factor of the learning rate, error function, activation function, optimization algorithm, the network structure and other aspects.

Keywords—Artificial Neural Network; local minimum; BP algorithm; learning rate

I. INTRODUCTION

Rumelhart and some other scholars advanced the Error Back Propagation theory in 1985, which was improved as BP Neural Network theory today. BP Neural Network is short for Error Back Propagation Neural Network, which includes one input layer, one or more hide layers and one output layer, and there are several neurons in each layer. BP Neural Network has explicit algorithmic process, complete theory system, powerful data identification and simulation functions.

Error back propagation algorithm is an algorithm which is to estimate the error of the output layer directly leading layer (such as hidden layer) by the error of the output layer, and then again to estimate the error of previous layer (such as input layer) by this error(the estimated error). Repeatedly calculated to go through such layers, it is easy to get from the input layer to the output layer error estimates for all other layers. By the use of such a method, the calculated error of the output layer can propagate to the direction of the input layer of the artificial neural network step by step and layer by layer. With such propagation, the weights of each layer can be corrected. Multilayer forward network which uses BP algorithm learning and training is called the BP network. BP algorithm provides a multilayer artificial neural network training guidance.

BP algorithm owns the excellent ability to solve non-linear problem, therefore, the value of practical application is outstanding. Along with researching deeply, the defects of BP Neural Network have been found, such as low convergence speed, long training time, falling into local minimum easily, bad generalization ability, few principles to build network structure. These defects can depress the

accuracy of BP Neural Network and damage the practical effect. Therefore, improving BP Neural Network step by step is significant not only for the theory of BP Neural Network but also for the practical application.

II. MODEL OF BP NETWORK

A typical four-layer BP neural network is clearly shown in Fig. 1, and its number of the input layer note is n , input variable is $x_i(p)$ ($i = 1, 2, \dots, n$); the number of the neuron in the hidden layer of the first level is l , the number of the neuron in the hidden layer of the second level is s , the number of the output layer note is m ; the weights between the input layer note and the hidden layer note of the first level is w_{ij} ($i = 1, 2, \dots, n; j = 1, 2, \dots, l$), the thresholds of the hidden layer note of the first level is θ_j ($j = 1, 2, \dots, l$), the output of the hidden layer note of the first level is $o_j(p)$; the weights between the hidden layer note of the first level and the hidden layer note of the second level is v_{jt} ($j = 1, 2, \dots, l; t = 1, 2, \dots, s$), the thresholds of the hidden layer note of the second level is δ_t ($t = 1, 2, \dots, s$), the output of the hidden layer note of the second level is $o_t(p)$; the weights between the hidden layer note of the second level and the output layer note is u_{tk} ($t = 1, 2, \dots, s; k = 1, 2, \dots, m$), the thresholds of the output layer note is φ_k ($k = 1, 2, \dots, m$), the output of the output layer note is $y_k(p)$.

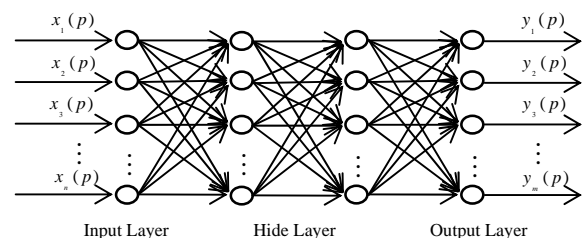


Figure 1. Structure diagram of a four-layer BP neural network

The hidden layer can be two layers as shown in Fig. 1, and also can be monolayer or multilayer; the front layer

connects to the rear layer by weights. The hidden layer nodes in general use the sigmoidal type function, and sigmoidal type function or a linear function can be used in the input and output nodes.

III. CONCEPTS AND DEFINITION OF BP NETWORK LEARNING ALGORITHM

A. Concepts of BP Network Learning Algorithm

- Process of Learning: BP neural network adjusts and changes its network connection weights of each node of each layer, so that the network output error is getting smaller and smaller, that is gradually close to the expected output.
- Essence of Learning: Dynamical adjustment of the connection weights of the neural network for its nodes of each layer.
- Rule of Learning: Weight adjustment rules of the nodes of various layers, that is, the BP algorithm, which is propagating the mean square error between the actual output and the expected output back to each node of the hidden layer and the input layer in the learning process.
- Type of Learning: Unsupervised learning, which is learning with a teacher.
- Idea of Learning: Back-propagation, which is propagating the output error in some form layer by layer through the hidden layer to the input layer.

B. Definition of BP Network Learning Algorithm

- Input vector: $x = (x_1, x_2, \dots, x_n)$.
- Hide layer input vector: $hi = (hi_1, hi_2, \dots, hi_p)$.
- Hide layer output vector: $ho = (ho_1, ho_2, \dots, ho_p)$.
- Output layer input vector: $yi = (yi_1, yi_2, \dots, yi_q)$.
- Output layer output vector: $yo = (yo_1, yo_2, \dots, yo_q)$.
- Expected output vector: $d_o = (d_1, d_2, \dots, d_q)$.
- Weight between input layer and hide layer: w_{ih} .
- Weight between hide layer and output layer: w_{ho} .
- Threshold of each neuron of hide layer: b_h .
- Threshold of each neuron of output layer: b_o .
- Number of sample data: $k = 1, 2, \dots, m$.
- Activation function: $f(\bullet)$.
- Error function: $e = \frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2$.

IV. PROCESS OF BP NETWORK LEARNING ALGORITHM

According to the concepts and definition of BP network learning algorithm, BP network learning algorithm can be designed as following steps.

Algorithm 1: BP network learning algorithm

Step 1: Network initialization, which is to assign a random value that is in $(-1, 1)$ to each weight, set error function e , and set calculation accuracy value ε and the maximum number of learning times M .

Step 2: Randomly select k -th input sample and the corresponding expected output:

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k))$$

$$d_o(k) = (d_1(k), d_2(k), \dots, d_q(k))$$

Step 3: Calculate the input and output of each neuron node of the hidden layer:

$$hi_h(k) = \sum_{i=1}^n w_{ih} x_i(k) - b_h \quad h = 1, 2, \dots, p$$

$$ho_h(k) = f(hi_h(k)) \quad h = 1, 2, \dots, p$$

$$yi_o(k) = \sum_{h=1}^p w_{ho} ho_h(k) - b_o \quad o = 1, 2, \dots, q$$

$$yo_o(k) = f(yi_o(k)) \quad o = 1, 2, \dots, q$$

Step 4: Calculate partial derivative $\delta_o(k)$ of the error function for each neuron of the output layer by comparing the expected output with the actual output of neural network:

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o(k)} \frac{\partial yi_o(k)}{\partial w_{ho}}$$

$$\frac{\partial yi_o(k)}{\partial w_{ho}} = \frac{\partial (\sum_{h=1}^p w_{ho} ho_h(k) - b_o)}{\partial w_{ho}} = ho_h(k)$$

$$\frac{\partial e}{\partial yi_o(k)} = \frac{\partial \left(\frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2 \right)}{\partial yi_o(k)}$$

$$= -(d_o(k) - yo_o(k)) yo_o'(k)$$

$$= -(d_o(k) - yo_o(k)) f'(yi_o(k)) \triangleq -\delta_o(k)$$

Step 5: Calculate partial derivative $\delta_h(k)$ of the error function for each neuron of the hide layer by using the weights between hide layer and output layer and $\delta_o(k)$ of output layer:

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o(k)} \frac{\partial yi_o(k)}{\partial w_{ho}} = \delta_o(k) ho_h(k)$$

$$\frac{\partial e}{\partial w_{ih}} = \frac{\partial e}{\partial hi_h(k)} \frac{\partial hi_h(k)}{\partial w_{ih}}$$

$$\frac{\partial hi_h(k)}{\partial w_{ih}} = \frac{\partial \left(\sum_{i=1}^n w_{ih} x_i(k) - b_h \right)}{\partial w_{ih}} = x_i(k)$$

$$\frac{\partial e}{\partial hi_h(k)} = \frac{\partial e}{\partial ho_h(k)} \frac{\partial ho_h(k)}{\partial hi_h(k)}$$

$$\begin{aligned}
 &= \frac{\partial \left(\frac{1}{2} \sum_{o=1}^q (d_o(k) - y_{o_o}(k))^2 \right)}{\partial h_{o_h}(k)} \frac{\partial h_{o_h}(k)}{\partial h_i(k)} \\
 &= \frac{\partial \left(\frac{1}{2} \sum_{o=1}^q (d_o(k) - f(y_{i_o}(k)))^2 \right)}{\partial h_{o_h}(k)} \frac{\partial h_{o_h}(k)}{\partial h_i(k)} \\
 &= \frac{\partial \left(\frac{1}{2} \sum_{o=1}^q \left(d_o(k) - f \left(\sum_{h=1}^p w_{ho} h_{o_h}(k) - b_o \right) \right)^2 \right)}{\partial h_{o_h}(k)} \frac{\partial h_{o_h}(k)}{\partial h_i(k)} \\
 &= - \sum_{o=1}^q (d_o(k) - y_{o_o}(k)) f'(y_{i_o}(k)) w_{ho} \frac{\partial h_{o_h}(k)}{\partial h_i(k)} \\
 &= - \sum_{o=1}^q \delta_o(k) w_{ho} f'(h_i(k)) \triangleq -\delta_h(k)
 \end{aligned}$$

Step 6: Fix the connection weights $w_{ho}(k)$ by using $\delta_o(k)$ of output layer neuron and output of hide layer neuron:

$$\begin{aligned}
 \Delta w_{ho}(k) &= -\mu \frac{\partial e}{\partial w_{ho}} = \mu \delta_o(k) h_{o_h}(k) \\
 w_{ho}^{N+1} &= w_{ho}^N + \eta \delta_o(k) h_{o_h}(k)
 \end{aligned}$$

Step 7: Fix the connection weights $w_{ih}(k)$ by using $\delta_h(k)$ of hide layer neuron and input of input layer neuron:

$$\begin{aligned}
 \Delta w_{ih}(k) &= -\mu \frac{\partial e}{\partial w_{ih}} = \mu \delta_h(k) x_i(k) \\
 w_{ih}^{N+1} &= w_{ih}^N + \eta \delta_h(k) x_i(k)
 \end{aligned}$$

Step 8: Calculate the global error:

$$E = \frac{1}{2m} \sum_{k=1}^m \sum_{o=1}^q (d_o(k) - y_{o_o}(k))^2$$

Step 9: Judge whether the net error meets the requirement: when the error reaches the preset accuracy or learning times are greater than the preset maximum number of learning times, the algorithm should be ended immediately. Otherwise, select the next learning sample and its expected output, and return to the third step for the learning of next round.

V. IMPROVEMENT OF BP NETWORK LEARNING ALGORITHM

BP network learning algorithm has many advantages, such as easy, simple, small amount of calculation and parallelism. However, BP network learning algorithm also has many defects, for example, easy to fall into local minimum state, slow convergence rate and low learning efficiency. Therefore, improvement of BP network learning algorithm is very necessary.

A. Pretreatment of the input data

Preprocessing of the input data can accelerate the learning speed. The general selection of initial value of the network weights and offset values is to choose a different

initial value to train the network, and then select the most accurate network for practical use. But this method is with arbitrary, and sometimes it is unable to find a better initial value. The following algorithm can get a better selection of initial value of the network weights and offset values.

Definition:

n_0 : The number of input of BP network.

n_1 : The number of hide layer neuron.

γ : Scaling factor.

Algorithm 2: BP network weights initialization algorithm

Step 1: Calculate the scaling factor: $\gamma = 0.7^{n_0/n_1}$.

Step 2: Select random number which is from interval $[-0.5, 0.5]$ for the initial value of weight w_{ij} .

Step 3: Re-initialize the weights: $w_{ij} = \gamma \frac{w_{ij}}{\sum_{i=1}^{n_1} w_{ij}^2}$.

Step 4: For the i-th neuron of hide layer, its offset value is selected as the random number which is taken from the interval $[-w_{ij}, w_{ij}]$.

B. Addition of momentum factor

The adjustment of connection weights in the traditional BP algorithm does not consider the gradient direction before time t, but only adjusts the weights by the gradient descent direction of the error at time t. Therefore, BP algorithm often makes the learning process oscillated and convergence slow. In order to avoid oscillation and slow convergence speed in learning process of BP network, it needs to consider the effects of the last weight values change to the right weight values change and can be added a momentum factor α :

$$\begin{aligned}
 w_{ho}^{N+1} &= \alpha w_{ho}^N + (1 - \alpha) \eta \delta_o(k) h_{o_h}(k) \\
 w_{ih}^{N+1} &= \alpha w_{ih}^N + (1 - \alpha) \eta \delta_h(k) x_i(k)
 \end{aligned}$$

In the upper formulas, w_{ho}^{N+1} is the N+1-th hide layer-output layer weights matrix, w_{ih}^{N+1} is the N+1-th input layer-hide layer weights matrix, η is learning rate, α is momentum factor, $\alpha \in [0, 1]$ define momentum item αw_{ho}^N , it reflects the previously accumulated experience of adjustment. When $\alpha = 0$, the weights correction only relates to the current negative gradient; When $\alpha = 1$, the weights correction depends entirely on previous cycle of negative gradient. The added momentum term in this method can be substantially regarded as a low pass filter which smoothes the oscillation of the learning process, thereby it accelerates the speed of convergence.

At the same time, when the local gradient appears the local minimum value, although $\delta_o(k) h_{o_h}(k) \rightarrow 0$, due to $\alpha w_{ho}^N \neq 0$, it can make the network out of the local minimum area to avoid it converging to the local minima.

VI. ANALYSIS OF THE IMPROVED BP NETWORK LEARNING ALGORITHM

According to the steps of BP network learning algorithm, we can bring about BP Network easily. Now, we use some real data to test the difference between the traditional BP network learning algorithm and the improved BP network learning algorithm.

Suppose we use pretreatment of the input data and addition of momentum factor to improve the algorithm, and take the momentum factor $\alpha = 0.1$, set calculation accuracy value $\varepsilon = 0.0001$, and the number of hide layer node is 16. We can get the test result which is shown in Fig. 2, which is clear that the improved BP network learning algorithm is more efficient than the traditional one.

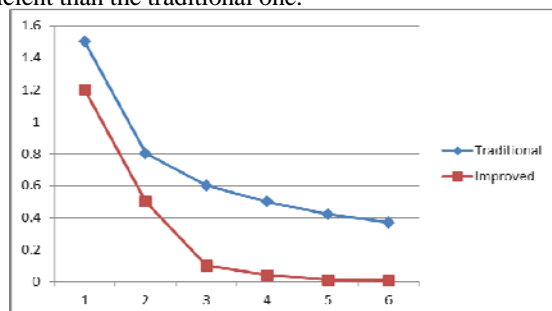


Figure 2. Compare of the improved algorithm and the traditional one

VII. CONCLUSION

With the development of science and technology, BP network is being widely used and improving quickly. In this paper, we researched the method and process of traditional

BP network learning algorithm. However, there is no a feasible satisfactory method to avoid the defects of BP network learning algorithm. Pretreatment of the input data and addition of momentum factor can improve the BP network learning algorithm, but the two methods are not all-purpose. They maybe lost their improvement in some conditions. However, they are still giving us some inspiration. At the end of the paper, a comparison between the traditional BP network learning algorithm and the improved BP network learning algorithm is given. It is clearly that the improved BP network learning algorithm is more effective and feasible.

REFERENCES

- [1] Paulito P. Palmes, Yaichi Hayasaka, Shiro Usui, "Mutation-Based Genetic Neural Network," IEEE Transactions on Neural Networks, 2005, 16(3), pp. 587-600.
- [2] Wei Gao, "Study on evolutionary neural network based on ant colony optimization," 2007 International Conference on Computational Intelligence and Security Workshops, 2007, pp. 3-6.
- [3] Arun D. Kulkarni, "Neural Network Models for Classification," Applied Intelligence, 2000, 12(3), pp. 207-215.
- [4] Y.H. Zweiri, J.F. Whidborne, L.D. Seneviratne, "A three-term back-propagation algorithm," Neurocomputing, 2003, pp. 305-318.
- [5] Yi-Jen Wang, Chin-Teng Lin, "A second-order learning algorithm for multilayer network based on block Hessian matrix," Neural Networks, 1998, pp. 1607-1622.
- [6] Wei Gao, "Study on New Evolutionary Neural Network," Proceeding of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003, pp. 1287-1292.
- [7] Ma Da, Hu Hai-guang, Guan Jian-he, "The Naive Bayesian Approach in Classifying the Learner of Distance Education System," 2nd International Conference on Information Engineering and Computer Science - Proceedings, 2010, 12(2), pp. 978-981.