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 nonlinear 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 \( x_p \left( i = 1, 2, \ldots, n \right) \); 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 \( w_{i,j} \left( i = 1, 2, \ldots, n; j = 1, 2, \ldots, l \right) \), the thresholds of the hidden layer note of the first level is \( \theta_{j} \left( j = 1, 2, \ldots, l \right) \), the output of the hidden layer note of the first level is \( o_{j} \left( j = 1, 2, \ldots, l \right) \); the weights between the hidden layer note of the first level and the hidden layer note of the second level is \( v_{j,k} \left( j = 1, 2, \ldots, l; k = 1, 2, \ldots, s \right) \), the thresholds of the hidden layer note of the second level is \( \delta_{i,k} \left( i = 1, 2, \ldots, s \right) \), the output of the hidden layer note of the second level is \( o_{k} \left( j = 1, 2, \ldots, s \right) \); the weights between the hidden layer note of the second level and the output layer note is \( u_{k,p} \left( k = 1, 2, \ldots, m \right) \), the thresholds of the output layer note is \( \phi_{k} \left( k = 1, 2, \ldots, m \right) \), the output of the output layer note is \( y_p \left( k = 1, 2, \ldots, m \right) \).

![Structure diagram of a four-layer BP neural network](image_url)

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 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 note 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:  \( x = (x_1, x_2, \ldots, x_n) \).
- Hide layer input vector:  \( h_i = (h_{i1}, h_{i2}, \ldots, h_{ip}) \).
- Hide layer output vector:  \( h_o = (h_{o1}, h_{o2}, \ldots, h_{op}) \).
- Output layer input vector:  \( y_i = (y_{i1}, y_{i2}, \ldots, y_{iq}) \).
- Output layer output vector:  \( y_o = (y_{o1}, y_{o2}, \ldots, y_{oq}) \).
- Expected output vector:  \( d_o = (d_{o1}, d_{o2}, \ldots, d_{oq}) \).
- Weight between input layer and hide layer:  \( w_{hi} \).
- 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:  \( k = 1, 2, \ldots, m \).
- Activation function:  \( f(\bullet) \).
- Error function:  \( e = \frac{1}{2} \sum_{i=1}^{m} (d_i(k) - y_o(i,k))^2 \).

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  \( \varepsilon \) and the maximum number of learning times  \( M \).

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

- \( x(k) = (x_1(k), x_2(k), \ldots, x_n(k)) \)
- \( d_o(k) = (d_{o1}(k), d_{o2}(k), \ldots, d_{oq}(k)) \)

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

- \( h_i(k) = \sum_{i=1}^{p} w_{hi}(k) x_i(k) - b_h \quad h = 1, 2, \ldots, p \)
- \( h_o(k) = f(h_i(k)) \quad h = 1, 2, \ldots, p \)
- \( y_i(k) = \sum_{i=1}^{q} w_{ho}(k) h_i(k) - b_o \quad o = 1, 2, \ldots, q \)
- \( y_o(k) = f(y_i(k)) \quad o = 1, 2, \ldots, q \)

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

\[
\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial y_i} \frac{\partial y_i}{\partial h_o} = \delta(k) h_o(k)
\]

\[
\frac{\partial y_i}{\partial h_o} = \frac{1}{2} \sum_{i=1}^{q} (d_o(k) - y_o(k))^2
\]

\[
\frac{\partial e}{\partial y_i} = -\left( d_o(k) - y_o(k) \right) y_o'(k) = -\left( d_o(k) - y_o(k) \right) f'(y_i(k)) \nabla \delta(k)
\]

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

\[
\frac{\partial e}{\partial w_{hi}} = \frac{\partial e}{\partial y_i} \frac{\partial y_i}{\partial h_i} = \delta(k) h_o(k)
\]

\[
\frac{\partial e}{\partial h_i} = \frac{\partial e}{\partial y_i} \frac{\partial y_i}{\partial h_i}
\]

\[
\frac{\partial h_i}{\partial w_{hi}} = \left( \sum_{i=1}^{p} w_{hi}(k) x_i(k) - b_h \right)
\]

\[
\frac{\partial e}{\partial b_h} = \frac{\partial e}{\partial y_i} \frac{\partial y_i}{\partial b_h}
\]

Published by Atlantis Press, Paris, France. © the authors 0419
Algorithm 2: BP network weights initialization algorithm

Step 1: Calculate the scaling factor: \( \gamma = 0.7^{n_i} \).

Step 2: Select random number which is from interval \([-0.5, 0.5]\) for the initial value of weight \( w_i \).

Step 3: Re-initialize the weights: \( w_i = \gamma \frac{w_i}{\sum_{i=1}^{n_i} w_i} \).

Step 4: For the i-th neuron of hidden layer, its offset value is selected as the random number which is taken from the interval \([-w_i, w_i]\).

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 \( \alpha \) :

\[
\begin{align*}
\Delta w_{hi}^{(t)} &= \alpha w_{hi}^{(t-1)} + (1 - \alpha) \eta \delta_i(k)x_i(k) \\
\end{align*}
\]

In the upper formulas, \( w_{hi}^{(t)} \) is the \( N+1 \)-th hidden layer weights matrix, \( w_{hi}^{(t-1)} \) is the \( N+1 \)-th hidden layer weights matrix, \( \eta \) is learning rate, \( \alpha \) is momentum factor, \( \alpha \in [0, 1] \) define momentum item \( \alpha w_{hi}^{(t)} \), 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, due to \( \alpha w_{hi}^{(t)} \neq 0 \), 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 0.1 $\alpha = 0.1$, 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](image)

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


