Research Article

Model and Algorithm of BP Neural Network Based on Expanded Multichain Quantum Optimization

Baoyu Xu,¹ Hongjun Zhang,¹ Zhiteng Wang,¹ Huaixiao Wang,² and Youliang Zhang³

¹PLA University of Science and Technology, Nanjing 210007, China

²Bengbu Automobile NCO Academy, Bengbu 233011, China

³Logistics Academy, Beijing 100858, China

Correspondence should be addressed to Huaixiao Wang; 753048237@qq.com

Received 30 August 2015; Accepted 15 October 2015

Academic Editor: Peter Dabnichki

Copyright © 2015 Baoyu Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The model and algorithm of BP neural network optimized by expanded multichain quantum optimization algorithm with super parallel and ultra-high speed are proposed based on the analysis of the research status quo and defects of BP neural network to overcome the defects of overfitting, the random initial weights, and the oscillation of the fitting and generalization ability along with subtle changes of the network parameters. The method optimizes the structure of the neural network effectively and can overcome a series of problems existing in the BP neural network optimized by basic genetic algorithm such as slow convergence speed, premature convergence, and bad computational stability. The performance of the BP neural network controller is further improved. The simulation experimental results show that the model is with good stability, high precision of the extracted parameters, and good real-time performance and adaptability in the actual parameter extraction.

1. Introduction

Artificial neural network (ANN) is a new information processing and computer system which is based on the modern neuroscience research and is formed by abstracting, simplifying, and simulating of biological structure. The features of ANN are as follows. ANN can fully approximate to any complex nonlinear relations; all quantitative or qualitative information keeps in storage equipotentially in each neuron of network so it is with very strong robustness and fault tolerance; ANN is a kind of system which can emulate and adapt to the unknown system and is able to deal with the quantitative and qualitative knowledge at the same time [1]. The main research of ANN is how to make computer simulate and realize the self-learning and mathematical thinking ability of human to mine the inner relation from limited samples. It is mainly through studying and storing the data relations, the inference rules, and probability distribution of known sample to deduce and reveal the potential information between variables in the unknown data sample [2].

BP neural network is currently the most popular neural network model in application [3]. BP algorithm was proposed

by Rumelhart and Mclelland in 1986 which well solved the weight adjustment problems of nonlinear continuous function in the field of multilayer feedforward neural network. It is a typical error back propagation algorithm [4, 5]. Selection of activation function, design of structure parameter, and improvement of network defect have been researched a lot since the emergence of the BP neural network. In 1973, Grossberg found sigmoid function is very similar to the work situation of biological neurons so he began to explore the relationship between the features of sigmoid function and the stability of the neural network. He prompted the function to become the most commonly used activation function of BP neural network. Since then many scholars have done improvements on the limitations of the sigmoid function. In order to solve the problem that BP algorithm can obtain larger gradient values in the whole domain, many scholars try to combine activation functions with different characteristics in different intervals together to set a larger derivative value at the needful position to make up for the inadequacy of single activation function [6]. In 1991, Kung and Hu [7] proposed FARM approximate simplified method which uses Frobenius norm according to the ideas of the global optimization and

gradual optimization to delete the hidden layer units and determine the number of hidden layer nodes. Fahlman [8] found that if we adopt the sigmoid function as the activation function, the derivative of error of weight will become very small when the output value of BP network is close to 0 and 1. Salomon [9] created two BP neural networks with exactly the same initial parameters and structure and put forward a new method of network learning. The learning rate η is increased and decreased, respectively, to update the weight of two networks and network is set with fast error drop as the starting point of the next update. The effect is obvious. Dan Foresee and Hagan [10] proposed BFGS quasi-Newton method which not only can avoid calculation of the second derivative but also keeps the advantage of fast convergence of the Newton algorithm. Riedmiller and Braun [11] proposed elastic BP (RPROP) method in 1993. RPROP method introduced probability of resilient update value to modify weights directly which can reduce influence of the network structure parameters in the whole learning process and avoid unforeseen gradient error convergence of fuzzy network performance.

At present, the research based on ANN to carry out quantitative structure-activity relationship research has gradually been applied to various fields. Wang et al. [12] utilized neural network to study the quantitative structure-activity relationship of angiotensin converting enzyme inhibitors. Zhao et al. [13] established the model of antitumor activity of emodin derivatives on the basis of neural network. Cui et al. [14] applied the principal component analysis and neural network method to study the quantitative structure-activity relationship of the nitrobenzene and its homologue. González-Díaz et al. [15] established some quantitative structure-activity relationship models of synthetic compounds by neural network and verified the reliability of the model. The model can be used to design new drugs. Prado-Prado et al. [16] studied quantitative structure-activity relationship of parasite drug resistance between different kinds of parasites through neural network. Ramírez-Galicia et al. [17] established quantitative structure-activity relationship model of amoeba drug resistance through multiple linear regression, stepwise regression analysis, and neural network. The result shows that the three-dimensional structure of the model is very important.

Although BP algorithm has become the most widely used artificial neural network, the BP neural network has the following defects.

(1) It falls into local minimum value easily: the BP neural network learning algorithm based on gradient descent method is easy to fall into local extremum points and saddle point if error function is not strictly convex function and there are multiple points whose gradient is zero. Then, the network can not converge and is unable to get the optimal solution of the problem so the optimal network connection weights and threshold parameters can not be obtained [18].

(2) The speed of error convergence is slow: convergence speed of BP neural network is decided by two aspects: one is learning rate and the other one is the size of derivative of related excitation function [19]. It is usually not easy to choose the size of the learning rate. If the learning rate is too large, the oscillation or even no convergence phenomenon will happen

in the process of training. The learning rate should be a small positive number. If the learning rate is too small, the product of learning rate and negative gradient vector will become smaller. Then, the renewal speed of weights and thresholds will be affected. In addition, the size of derivative of the activation function also affects the rate of convergence. In the flat area of error curved surface, the gradient of error function is small to make update speed of weights and threshold of network slow. The network needs to pass through much iteration to be out of the flat area so the convergence speed of the network becomes slow.

(3) Structure of network is not easy to determine: the determination of the structure of BP neural network usually refers to determination of the number of hidden layers and the number of neurons of hidden layer. Especially after Kolmogorov et al. prove the single hidden layer of BP neural network approximation theorem, how to select the number of neurons in single hidden layer neural network has always been number one of the hot and key problems. Generally, the number of neurons of input layer and output layer is easy to determine according to identification objects. The number of neurons of hidden layer is difficult to determine, and it directly affects topology structure and performance of BP neural network. It has been proved in theory that the three-layer BP neural network can approximate nonlinear functions of arbitrary precision which solves the problem of determination of the number of hidden layers in the BP neural network. If the number of the neurons of hidden layer is too small, the network can not meet the requirements of learning and the approximation performance; if the number of the neurons of hidden layer is too much, there will be adverse phenomenon in the network and it will make the hardware implementation and software calculation complicate at the same time. At present, there is no unified and complete determination theoretical framework to determine the structure of the BP neural network. The experience or grope through lots of experiments is the usual manner to estimate and adjust the structure of neural network.

In order to further improve the efficiency of the BP neural network and overcome the shortage of it, a lot of research has been conducted. Sun et al. established improved BP neural network prediction model and quantitatively researched related parameters [20]. The accuracy of the model was improved to some extent. Xiao et al. proposed BP neural network with rough set for short term load forecasting. The accuracy of prediction of BP neural network was further improved [21].

Genetic algorithm imitates evolutionary and genetic rule of biology and is a mathematical algorithm which can solve problems to find the global optimization. Due to the strong macroscopic search ability and good global optimization performance of genetic algorithm, many scholars try to use genetic algorithm to optimize the connection weights, structure, learning rules, and so forth of BP neural network. Xiao et al. [22] applied genetic algorithm to constitute GA-ANN method to optimize the optimization problems in complex engineering. The method not only takes advantage of the nonlinear mapping, network reasoning, and predicting function of neural network but also uses the characteristics

of global optimization of genetic algorithm. It can be widely applied to many complex engineering problems whose objective function is difficult to express by the form of explicit function of decision variables. Yang et al. [23] applied genetic algorithm to select parameters which overcame the restriction of symmetric weight matrix of traditional fluid neural network and broadened the application fields of this intelligent exploration method. Ge [24] applied genetic algorithm to optimize the controller parameters of structure of the neural network and applied the controller to control object with pure lag. The experiment proved that the control system optimized by genetic algorithm was with good static performance and dynamic performance. Li et al. [25] combined genetic algorithm with traditional DBD algorithm to propose a new algorithm to optimize the BP neural network, making network with large scale be able to converge fast and be out of the trap of local minimum. This algorithm is less sensitive to the selection of network parameters and obtains good effect in the application of missile comprehensive test.

In order to further overcome the shortage of the model based on genetic algorithm to optimize the BP neural network, the model and algorithm of BP neural network based on expanded multichain quantum optimization are proposed. The structure of neural network is effectively optimized. The model can overcome a series of problems of basic genetic algorithm, such as the slow convergence speed, premature convergence, and bad computational stability to further improve the performance of the BP neural network controller.

2. BP Neural Network Model and Algorithm

The main idea of BP algorithm is to divide learning process into two stages, that is, positive communication of signal and back propagation of error. In the stage of positive communication, input information is from the input layer to output layer through the hidden layer [26]. The output signal forms on the output side. The weights of network are fixed in the process of the signal transmission forward. The state of neurons of each layer only affects state of neurons of the next layer. If the desired output can not be achieved in the output layer, then the error signal will back propagate. In the back propagation stage, the error signal which failed to meet the accuracy requirement spreads forward step by step and the error is shared by all units of each layer. The connection weights are adjusted dynamically according to the error signal. The weight value between neurons keeps correcting through the cycle of forward and back adjustment. The learning stops when the error of the output signal meets the requirement of precision [27].

2.1. Topology Structure of BP Neutral Network. The simplest BP neural network is with three layers as is shown in Figure 1. It includes input layer, hidden layer, and output layer. The number of nodes in the input layer is equal to the dimension of input vector n . The number of nodes in the output layer is equal to the type of output module q . The number of nodes

in the hidden layer p is associated with specific application which is usually selected by test.

The connection weight matrix between input layer nodes and hidden layer nodes is W_1 which is defined as formula (1). Similar to it, the connection weight matrix between hidden layer nodes and the output nodes can be defined as W_2 . The threshold matrix of nodes of hidden layer and output layer is θ_1 and θ_2 , respectively:

$$W_1 = \begin{bmatrix} w_{111} & \cdots & w_{11j} & \cdots & w_{11p} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{1k1} & \cdots & w_{1kj} & \cdots & w_{1kp} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{1n1} & \cdots & w_{1nj} & \cdots & w_{1np} \end{bmatrix} \quad (1)$$

2.2. Learning Algorithm of BP Network. The learning algorithm of BP network can be divided into two stages that are forward and back propagation. It is a gradient descent algorithm which can make error of per connection weights of the neural network reduce. At the beginning of learning, assign the random numbers in the range of $[-1, +1]$ to the connection weight matrices W_1 , W_2 and threshold matrices of the hidden layer and output layer nodes θ_1 , θ_2 .

2.2.1. Forward Operation. First, enter the learning samples $\{X, T\}$, where X is the input vector of the learning samples and T is the corresponding output vector:

$$X = (x_1, x_2, \dots, x_k, \dots, x_n), \quad 1 \leq k \leq n, \quad (2)$$

$$T = (T_1, T_2, \dots, T_m, \dots, T_q), \quad 1 \leq m \leq q,$$

where n is the number of nodes of input layer; q is the number of nodes of output layer.

The data forward propagates to the output layer through the input layer and hidden layer. The produced weight value of output pattern classification is the learning result. The following steps are mainly included.

Step 1 (calculation of output value of the hidden layer nodes). The input value of nodes in hidden layer is

$$i_{1j} = \sum_{k=1}^n w_{1kj} x_k, \quad 1 \leq j \leq p, \quad (3)$$

where n is the number of nodes of input layer, p is the number of nodes of hidden layer, w_{1kj} is connection weight, and x_k is component of input vector.

The output value of node j is

$$h_j = f(i_{1j} + \theta_{1j}), \quad (4)$$

where θ_{1j} is the threshold of node j . The activation function f used is the sigmoid function given by Rumelhart:

$$f(i_{1j}) = \frac{1}{1 + e^{-i_{1j}}}. \quad (5)$$

FIGURE 1: Topology structure of BP neural network.

Step 2 (calculation of the output value of the nodes of output layer). The input values of nodes of output layer m are

$$i_{2m} = \sum_{j=1}^p w_{2jm} h_j, \quad 1 \leq m \leq q. \quad (6)$$

The output value of node m is

$$O_m = f(i_{2m} + \theta_{2m}), \quad (7)$$

where θ_{2m} is threshold of node of output layer m and f is activation function defined by type (5).

2.2.2. The Back Operation. Calculate the error of output value and expected value of output layer. The error back propagates from the output layer through the hidden layer to the input layer. The connection weight values are modified. Steps are as follows.

Step 1 (calculate the output error of nodes of output layer). The error between the learning value O_m of node of output layer m and the output value of learning samples T_m is

$$\varepsilon_m = |O_m - T_m|. \quad (8)$$

Step 2 (test learning error). ε_0 is the allowed maximum learning error, which is in the interval $[0, 1]$ set by the user.

If $\max(\varepsilon_m) \leq \varepsilon_0$, enter the next learning sample. Otherwise, adjust the weights of network and reenter the original learning sample.

The learning process is ended when all learning samples meet the aforementioned conditions.

Step 3 (calculate the learning error of nodes of output layer). The learning error of node of output layer m is

$$d_{2m} = O_m (1 - O_m) (O_m - T_m). \quad (9)$$

Step 4 (calculate learning error of nodes of hidden layer). Learning error of node of hidden layer j is

$$d_{1j} = h_j (1 - h_j) \sum_{m=1}^q w_{2jm} d_{2m}. \quad (10)$$

Step 5 (revise the value of the connection weights matrix W_2). Set the weight value at time $t + 1$ as adjusted new weight value; then

$$w_{2jm}(t + 1) = w_{2jm}(t) + \eta d_{2m} h_j + \alpha [w_{2jm}(t) - w_{2jm}(t - 1)], \quad (11)$$

where η is learning rate and α is momentum factor. Both η and α are in the scope $[0, 1]$. Using α can accelerate the learning speed and be helpful to overcome the local minimum problem of common BP algorithm.

Step 6 (revise the value of the connection weight matrix W_2). Consider

$$w_{1kj}(t + 1) = w_{1kj}(t) + \eta d_{1j} x_k + \alpha [w_{1kj}(t) - w_{1kj}(t - 1)]. \quad (12)$$

Step 7 (revise threshold θ_2). Threshold of nodes of output layer is

$$\theta_{2m}(t + 1) = \theta_{2m}(t) + \eta d_{2m} h_j + \alpha [w_{2jm}(t) - w_{2jm}(t - 1)]. \quad (13)$$

Step 8 (revise threshold θ_1). Threshold of nodes of hidden layer j is

$$\theta_{1j}(t + 1) = \theta_{1j}(t) + \eta d_{1j} + \alpha [\theta_{1j}(t) - \theta_{1j}(t - 1)]. \quad (14)$$

2.3. Run the BP Neural Network. Run the BP neural network after learning and carry out pattern classification for the input vector I . Only the forward operation of BP learning algorithm is used. The process is as follows.

Step 1. Assign the value revised by BP learning algorithm to the connection weight matrices W_1 and W_2 and threshold matrices θ_1 and θ_2 .

Step 2. Input the input vector I which needs to be recognized.

Step 4. Use formulas (5), (6), and (7) to calculate the output of the output layer, that is, the classification result of input vector I .

3. The Quantum Optimal Model Based on Multichain Expanded Coding Scheme

3.1. Principle of Quantum Coding Based on Multichain Expanded Scheme. In quantum computation, the smallest unit of information is represented by quantum bit whose state can be expressed as

$$|\varphi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (15)$$

where α and β satisfy the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1. \quad (16)$$

The complex numbers α and β satisfying formulas (15) and (16) are called probability amplitude of a quantum bit. Quantum bit can also be expressed by probability amplitude as $[\alpha, \beta]^T$. According to the property of probability amplitude, a quantum bit $|\varphi\rangle$ can be expressed by Figure 2.

Obviously, in Figure 2, $\alpha = \cos \theta$ and $\beta = \sin \theta$; therefore, quantum bit can be represented as

$$[\cos \theta, \sin \theta]^T. \quad (17)$$

The coding scheme is

$$p_i = \begin{vmatrix} \cos \theta_{i1} & \cos \theta_{i2} & \cdots & \cos \theta_{ij} \\ \sin \theta_{i1} & \sin \theta_{i2} & & \sin \theta_{ij} \end{vmatrix}, \quad (18)$$

where $\theta_{ij} = 2\pi \times \text{rand}$, $\text{rand} \in [0, 1]$; $i = 1, 2, 3, \dots, m$; $j = 1, 2, \dots, n$; m is the size of the population; n is the number of quantum bit.

FIGURE 3: Decomposition of probability amplitude of quantum bit.

Obviously, in Figure 3, $AB = \sin \theta$ can be decomposed into $[AB * \cos \theta, AB * \sin \theta]^T$ with the same way of formula (17). Combine formulas (17) and (18); quantum bit can be represented as the following form:

$$[\cos \theta \times \sin \theta, \sin \theta \times \sin \theta, \cos \theta]^T. \quad (19)$$

Formula (19) is to decompose $\sin \theta$ of formula (17) into two variables, so it also meets the condition of formula (16); however, it describes the feature from the view of two-dimensional space. Only one variable θ is not conducive to describing the dynamic behavior of the quantum objectively, comprehensively, and vividly. φ ($0 < \varphi < \pi$) is used to replace the described component of hypotenuse according to the characteristics of formulas (17) and (18). Formula (19) can be transformed to

$$[\cos \varphi \times \sin \theta, \sin \varphi \times \sin \theta, \cos \theta]^T. \quad (20)$$

Formula (9) is equivalent to translating $\sin \theta$ in formula (1) to the representation of quantum bit that is the variable in two-dimensional space which is translated to three-dimensional space. The encoding scheme based on three genes of three-dimensional space of the two angle variables is formed:

$$p_i = \begin{bmatrix} \cos \varphi_{i1} \sin \theta_{i1} \\ \sin \varphi_{i1} \sin \theta_{i1} \\ \cos \theta_{i1} \end{bmatrix} \cdots \begin{bmatrix} \cos \varphi_{in} \sin \theta_{ij} \\ \sin \varphi_{in} \sin \theta_{ij} \\ \cos \theta_{ij} \end{bmatrix}. \quad (21)$$

Similarly, β can be added to extend $\sin \varphi \sin \theta$ to form the encoding scheme based on four chains of four-dimensional space:

$$p_i = \begin{vmatrix} \cos \beta_{i1} \sin \varphi_{i1} \sin \theta_{i1} \\ \sin \beta_{i1} \sin \varphi_{i1} \sin \theta_{i1} \\ \cos \varphi_{i1} \sin \theta_{i1} \\ \cos \theta_{i1} \end{vmatrix} \dots \begin{vmatrix} \cos \beta_{ij} \sin \varphi_{ij} \sin \theta_{ij} \\ \sin \beta_{ij} \sin \varphi_{ij} \sin \theta_{ij} \\ \cos \varphi_{ij} \sin \theta_{ij} \\ \cos \theta_{ij} \end{vmatrix}. \quad (22)$$

FIGURE 4: $N + 1$ chain gene code.

Increase of the supporting angle constantly like this can conclude the encoding scheme based on $N + 1$ chain of $N + 1$ -dimensional space, where N is the number of angle variables of multiple chain code.

It can be concluded from Figure 4 that the $N + 1$ chain coding scheme is

$$p_i = \begin{bmatrix} (\sin \theta_n)_{i1} (\sin \theta_{n-1})_{i1} \cdots (\sin \theta_2)_{i1} (\sin \theta_1)_{i1} \\ (\cos \theta_n)_{i1} (\sin \theta_{n-1})_{i1} \cdots (\sin \theta_2)_{i1} (\sin \theta_1)_{i1} \\ \vdots \\ (\cos \theta_3)_{i1} (\sin \theta_2)_{i1} (\sin \theta_1)_{i1} \\ (\cos \theta_2)_{i1} (\sin \theta_1)_{i1} \\ (\cos \theta_1)_{i1} \end{bmatrix} \quad (23)$$

$$\cdots \begin{bmatrix} (\sin \theta_n)_{ij} (\sin \theta_{n-1})_{ij} \cdots (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_n)_{ij} (\sin \theta_{n-1})_{ij} \cdots (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ \vdots \\ (\cos \theta_3)_{ij} (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_1)_{ij} \end{bmatrix}.$$

3.2. Transform of Solution Space. The process of the quantum genetic optimization based on multichain code is limited

within the unit space $I^n = [-1, 1]^n$. Therefore, transformation between unit space and solution space of the optimization problem is required. Suppose that the j th solution variable of the optimization problem is $X_j \in [a_j, b_j]$ and the value of the j th quantum bit of the i th chromosome p_i is $[x_{ij}^{n+1}, x_{ij}^n, \dots, x_{ij}^2, x_{ij}^1]$, where n is the number of the angle variable of multichain code. The corresponding transformation formula of solution space is

$$\begin{aligned} X_{i1}^j &= \frac{1}{2} [b_j (1 + x_{ij}^1) + a_j (1 - x_{ij}^1)], \\ X_{i2}^j &= \frac{1}{2} [b_j (1 + x_{ij}^2) + a_j (1 - x_{ij}^2)], \\ &\vdots \\ X_{in}^j &= \frac{1}{2} [b_j (1 + x_{ij}^n) + a_j (1 - x_{ij}^n)], \\ X_{i(n+1)}^j &= \frac{1}{2} [b_j (1 + x_{ij}^{n+1}) + a_j (1 - x_{ij}^{n+1})]. \end{aligned} \quad (24)$$

3.3. Update of Quantum Chromosome. The update of the quantum rotation gate U makes each chromosome in the current population close to the optimal chromosome. During this process, better chromosome may be generated to make the population evolve. However, different coding scheme needs different quantum rotation gate to make the quantum bit rotate a corresponding angle

$$U = \begin{bmatrix} (\sin \theta_n)_{ij} (\sin \theta_{n-1})_{ij} \cdots (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_n)_{ij} (\sin \theta_{n-1})_{ij} \cdots (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ \vdots \\ (\cos \theta_3)_{ij} (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_1)_{ij} \end{bmatrix}$$

$$= \begin{bmatrix} (\sin \theta_n + \Delta \theta_n)_{ij} (\sin \theta_{n-1} + \Delta \theta_{n-1})_{ij} \cdots (\sin \theta_2 + \Delta \theta_2)_{ij} (\sin \theta_1 + \Delta \theta_1)_{ij} \\ (\cos \theta_n + \Delta \theta_n)_{ij} (\sin \theta_{n-1} + \Delta \theta_{n-1})_{ij} \cdots (\sin \theta_2 + \Delta \theta_2)_{ij} (\sin \theta_1 + \Delta \theta_1)_{ij} \\ \vdots \\ (\cos \theta_3 + \Delta \theta_3)_{ij} (\sin \theta_2 + \Delta \theta_2)_{ij} (\sin \theta_1 + \Delta \theta_1)_{ij} \\ (\cos \theta_2 + \Delta \theta_2)_{ij} (\sin \theta_1 + \Delta \theta_1)_{ij} \\ (\cos \theta_1 + \Delta \theta_1)_{ij} \end{bmatrix}. \quad (25)$$

Set $q_{0j}(x_{0j}^{n+1}, x_{0j}^n, \dots, x_{0j}^2, x_{0j}^1)$ to be the j th quantum bit of contemporary optimal chromosome, and $q_{ij}(x_{ij}^{n+1}, x_{ij}^n, \dots, x_{ij}^2, x_{ij}^1)$ is the j th quantum bit of the i th chromosome of contemporary population

$$S_n = \begin{vmatrix} x_{0j}^{n+1} & x_{ij}^{n+1} \\ x_{0j}^n & x_{ij}^n \end{vmatrix}, \quad (26)$$

$$S_{n-1} = \begin{vmatrix} x_{0j}^{n-1} & x_{ij}^{n-1} \end{vmatrix}.$$

(1) The rule of determining of rotation angle $\Delta \theta_n$ is as follows. If $S_n \neq 0$, the direction is $\text{sgn}(\Delta \theta_n) = -\text{sgn}(S_n)$; if $S_n = 0$, the direction can be either positive or negative.

(2) The rule of determining of rotation angle $\Delta \theta_{n-1}$ is as follows. If $S_{n-1} \neq 0$, the direction is $\text{sgn}(\Delta \theta_{n-1}) = -\text{sgn}(S_{n-1})$; if $S_{n-1} = 0$, the direction can be either positive or negative.

Set the maximal evolutionary generation to be G , and the size of the n th quantum rotation angle of the t th generation is $\Delta \theta_n = \Delta \theta * (1.01 - t/G)$, where $\Delta \theta$ is the maximum rotation angle. The changing type of the rotation angle can ensure the angle decreases with the increase of generation to rotate around the optimal solution.

3.4. Mutation of Quantum Chromosome. The mutation of quantum chromosome is realized by quantum Not gate V . The quantum bit with the amplitude of θ_n is changed to $\pi/2 - \theta_n$ through the role of quantum Not gate. Consider

$$V \begin{bmatrix} (\sin \theta_n)_{ij} (\sin \theta_{n-1})_{ij} \cdots (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_n)_{ij} (\sin \theta_{n-1})_{ij} \cdots (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ \vdots \\ (\cos \theta_3)_{ij} (\sin \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_2)_{ij} (\sin \theta_1)_{ij} \\ (\cos \theta_1)_{ij} \end{bmatrix} = \begin{bmatrix} \sin \left(\frac{\pi}{2} - \theta_n \right)_{ij} \sin \left(\frac{\pi}{2} - \theta_{n-1} \right)_{ij} \cdots \sin \left(\frac{\pi}{2} - \theta_2 \right)_{ij} \sin \left(\frac{\pi}{2} - \theta_1 \right)_{ij} \\ \cos \left(\frac{\pi}{2} - \theta_n \right)_{ij} \sin \left(\frac{\pi}{2} - \theta_{n-1} \right)_{ij} \cdots \sin \left(\frac{\pi}{2} - \theta_2 \right)_{ij} \sin \left(\frac{\pi}{2} - \theta_1 \right)_{ij} \\ \vdots \\ \cos \left(\frac{\pi}{2} - \theta_3 \right)_{ij} \sin \left(\frac{\pi}{2} - \theta_2 \right)_{ij} \sin \left(\frac{\pi}{2} - \theta_1 \right)_{ij} \\ \cos \left(\frac{\pi}{2} - \theta_2 \right)_{ij} \sin \left(\frac{\pi}{2} - \theta_1 \right)_{ij} \\ \cos \left(\frac{\pi}{2} - \theta_1 \right)_{ij} \end{bmatrix}. \quad (27)$$

The specific form of V can be determined by the method of undetermined coefficients. Diversity of population can be increased though the quantum Not gate which can prevent local convergence and avoid premature phenomenon in the process of evolution.

3.5. Algorithm Description. The main realization steps of multichain quantum genetic algorithm are as follows.

Step 1 (population initialization). Generate population with m initial individuals $Q(t) = \{q_1^t, q_2^t, \dots, q_m^t\}$ according to formula (12) randomly.

Step 2 (transformation of solution space). Map the multiple approximate solutions of each chromosome from unit space $I^n = [-1, 1]^n$ to the solution space of optimization problem Ω to get the solution set $X(t)$.

Step 3. Calculate the fitness of the approximate solution to obtain the contemporary optimal solution $\text{Best}X$ and contemporary optimal chromosome $\text{Best}C$.

Step 4. Make $\text{Best}X$ be the optimal solution GX ; make $\text{Best}C$ be the global optimal chromosome GC .

Step 5. Do the iteration cycle, $t = t + 1$, obtaining new species by updating and mutation.

Step 6. Transform solution space of $Q(t)$ to get the solution of optimal problem $X(t)$.

Step 7. Evaluate $X(t)$ to access the contemporary optimal solution $\text{Best}X$ and optimal chromosome $\text{Best}C$.

Step 8. If $\text{fit}(\text{Best}X) < \text{fit}(GX)$, update the contemporary optimal solution $\text{Best}X = GX$ and update the contemporary optimal chromosome $BC = GC$ at the same time to prevent the degradation phenomenon. Otherwise, $GX = \text{Best}X$, $GC = BC$. Such control can ensure the convergence to the optimal value.

Step 9. Go back to Step 5.

4. The Model of BP Neural Network Based on Multichain Code Quantum Optimization Algorithm

4.1. Code. Connect each weight and threshold of BP network in order to form a long string of an array of real number. The string is as a chromosome. The decoding value of individual is the corresponding weights and thresholds.

4.2. Fitness Function. The purpose of using multichain quantum algorithm to optimize BP network is to simplify the structure of the network to minimize the error of network. Therefore, the fitness function is

$$f = \frac{1}{E}, \quad (28)$$

$$E = \frac{1}{2} \sum_{p=1}^m \sum_{j=1}^n (d_{ij} - y_{pj})^2,$$

where E is the global error of network; m is the number of training samples; n is the number of units of output layer; d_{ij} is the desired output value; y_{pj} is the actual output value.

The combination of multichain quantum optimization algorithm and neural network is realized by using multichain quantum optimization algorithm to optimize the parameters of the neural network. The specific ways are as follows.

- (1) Use the algorithm to optimize the initial weights and thresholds of neural network.
- (2) Use the algorithm to optimize structure parameters of BP network, the main of which is to optimize the unit number of hidden layers of network.
- (3) Use the algorithm to optimize the selection the learning rate and momentum factor of BP network.

The first optimization way is adopted to realize the optimization of neural network in this paper. The specific implementation process is as below.

The basic idea of using multichain quantum optimization algorithm to optimize the initial weight and threshold value of BP network is as follows. The multichain quantum optimization algorithm is applied to ensure the appropriate initial weight and threshold value to find the optimal solution of BP algorithm.

The specific implementation steps of MCQA-BP (BP neural network based on multichain quantum optimization) algorithm are as follows.

- (1) Initial population with N individuals is produced by multichain quantum optimization algorithm. Each individual in the population composed the chromosome strings with the weight, threshold, and the number of units of hidden layer of BP network.
- (2) Enter the neural network module. Apply BP algorithm to do the learning process and apply multichain quantum optimization algorithm to do the fitness inspection. If the fitness is unqualified, update the chromosome until the fitness meets the requirements. The individuals which meet requirements are updated by multichain quantum optimization algorithm to calculate fitness. The unqualified individuals are updated until they meet the accuracy requirement. Then end the algorithm.

5. Simulation Test

5.1. Test Case 1. Apply MCQA-BP algorithm to diagnose fault state of machine. The data sample after normalization of machine status is shown in Table 1. The algorithm adopts network with three layers. The function of hidden layer is sigmoid tangent function; the function of output layer is sigmoid logarithmic function. The formula $n_2 = 2 \times n_1 + 1$ is used to calculate the number of neurons of hidden layer, where n_1 is the number of neurons of input layer; n_2 is the number of neurons of hidden layer. Since there are 15 input parameters and 3 output parameters, the structure of BP neural network is 15-31-3. That is, the number of nodes of input layer is 15, the number of nodes of hidden layer is 31, and the number of nodes of output layer is 3. The number of weight values is $15 \times 31 + 31 \times 3 = 558$ and the number of thresholds is $31 + 3 = 34$. The total number of all the parameters that need to be optimized is 592. The training frequency of network is 1000; the goal of training is 0.01. The minimum value of error of network is as the fitness function. The state of machine failure is divided into 3 types which are, respectively, normal: (1, 0, 0); crack: (0, 1, 0); defect: (0, 0, 1). In order to test the trained network, three sets of new data are presented in Table 2.

Use BP, GA-BP, DCQA-BP (BP neural network based on double-chain quantum optimization algorithm), and MCQA-BP (BP neural network based on multichain quantum optimization algorithm, where the number of chain is 4). In GA-BP and DCQA-BP, the population number is set to be 50; the maximal evolutionary generation is set to be 30.

TABLE 1: The training sample data.

Sample order	Characteristic value of sample	State
1	0.2286, 0.1292, 0.072, 0.1592, 0.1335, 0.0733, 0.1159, 0.094, 0.0522, 0.1345, 0.009, 0.126, 0.3619, 0.069, 0.1828	Normal
2	0.209, 0.0947, 0.1393, 0.1387, 0.2558, 0.09, 0.0771, 0.0882, 0.0393, 0.143, 0.0126, 0.167, 0.245, 0.0508, 0.1328	Normal
3	0.0442, 0.088, 0.1147, 0.0563, 0.3347, 0.115, 0.1453, 0.0429, 0.1818, 0.0378, 0.0092, 0.2251, 0.1516, 0.0858, 0.067	Normal
4	0.2603, 0.1715, 0.0702, 0.2711, 0.1491, 0.133, 0.0968, 0.1911, 0.2545, 0.0871, 0.006, 0.1793, 0.1002, 0.0789, 0.0909	Crack
5	0.369, 0.2222, 0.0562, 0.5157, 0.1872, 0.1614, 0.1425, 0.1506, 0.131, 0.05, 0.0078, 0.0348, 0.0451, 0.0707, 0.088	Crack
6	0.0359, 0.1149, 0.123, 0.546, 0.1977, 0.1248, 0.0624, 0.0832, 0.164, 0.1002, 0.0059, 0.1503, 0.1837, 0.1295, 0.07	Crack
7	0.1759, 0.2347, 0.1829, 0.1811, 0.2922, 0.0655, 0.0774, 0.02273, 0.2056, 0.0925, 0.0078, 0.1852, 0.3501, 0.168, 0.2668	Defect
8	0.0724, 0.1909, 0.134, 0.2409, 0.2842, 0.045, 0.0824, 0.1064, 0.1909, 0.1586, 0.0116, 0.1698, 0.3644, 0.2718, 0.2494	Defect
9	0.2634, 0.2258, 0.1165, 0.1154, 0.1074, 0.0657, 0.061, 0.2623, 0.2588, 0.1155, 0.005, 0.0978, 0.1511, 0.2273, 0.322	Defect

TABLE 2: The test sample data.

Sample order	Characteristic value of sample	State
1	0.2101, 0.095, 0.1298, 0.1359, 0.2601, 0.1001, 0.0753, 0.089, 0.0389, 0.1451, 0.0128, 0.159, 0.2452, 0.0512, 0.1319	Normal
2	0.2593, 0.18, 0.0711, 0.2801, 0.1501, 0.1298, 0.1001, 0.1891, 0.2531, 0.0875, 0.0058, 0.1803, 0.0992, 0.0802, 0.1002	Crack
3	0.2599, 0.2235, 0.1201, 0.0071, 0.1102, 0.0683, 0.0621, 0.2597, 0.2602, 0.1167, 0.0048, 0.1002, 0.1521, 0.2281, 0.3205	Defect

TABLE 3: The experiment results of the four algorithms.

Algorithm	Error	1	2	3	4	5	6	7	8	9	10	Average value
BP	Error 1	0.9455	1.0252	1.1493	1.0368	0.8473	0.9781	1.0185	1.0967	0.9795	1.0955	1.0172
	Error 2	1.9559	1.9604	2.0556	1.9821	1.8676	1.9592	1.9683	2.0947	1.9839	1.9868	1.9815
GA-BP	Error 1	0.8828	0.3893	0.5888	0.6893	0.8178	0.7885	0.3947	0.5838	0.5635	0.8384	0.6537
	Error 2	1.6535	1.421	1.3859	1.5857	1.8434	1.6883	1.4641	1.6759	1.5937	1.8758	1.6187
DCQA-BP	Error 1	0.6463	0.4027	0.5394	0.4859	0.6673	0.6375	0.3375	0.5473	0.5532	0.6391	0.5456
	Error 2	1.5874	1.3867	1.3589	1.4845	1.5838	1.5423	1.5984	1.5873	1.4839	1.5868	1.52
MCQA-BP	Error 1	0.5263	0.3017	0.4194	0.3559	0.5273	0.4275	0.5175	0.6273	0.3332	0.3291	0.4365
	Error 2	1.5174	1.3167	1.4389	1.3545	1.4438	1.3123	1.5284	1.4373	1.3239	1.3268	1.4

The 10 times average values of the performance of algorithms are shown in Table 3.

Apply the test method of mean difference of two normal populations (t -test) to analyze the error of BP, GA-BP, DCQA-BP, and MCQA-BP.

The data of error of the four methods can be considered to be samples from normal population $N(\mu, \sigma^2)$. The means of the sample of the four methods are $\bar{X}_1$, $\bar{X}_2$, $\bar{X}_3$, and $\bar{X}_4$, respectively, and the variances are S_1^2 , S_2^2 , S_3^2 , and S_4^2 . The t -statistic is introduced to be the test statistic:

$$t = \frac{(\bar{X}_i - \bar{X}_j) - \delta}{S_w \sqrt{1/n_i + 1/n_j}}, \quad (29)$$

where $\delta = \mu_i - \mu_j$, $S_w^2 = ((n_i - 1)S_i^2 + (n_j - 1)S_j^2)/(n_i + n_j - 2)$, $S_w = \sqrt{S_w^2}$.

The form of rejection region is $|(\bar{X}_i - \bar{X}_j) - \delta|/S_w \sqrt{1/n_i + 1/n_j} \geq k$.

Consider $\{\text{Reject } H_0 \text{ when } H_0 \text{ is true}\} = P_{\mu_i - \mu_j = \delta} \{|(\bar{X}_i - \bar{X}_j) - \delta|/S_w \sqrt{1/n_i + 1/n_j} \geq k\} = \alpha$; then $k = t_{\alpha/2}(n_i + n_j - 2)$.

So the rejection region is $|t| = |(\bar{X}_i - \bar{X}_j) - \delta|/S_w \sqrt{1/n_i + 1/n_j} \geq t_{\alpha/2}(n_i + n_j - 2)$.

The hypotheses $H_0: \mu_i - \mu_j < 0$ and $H_1: \mu_i - \mu_j \geq 0$ need to be tested.

The mean and variance of Error 1 of BP, GA-BP, DCQA-BP, and MCQA-BP are as follows: $n_1 = 10$, $\bar{X}_1 = 1.0172$, and $S_1^2 = 0.0075$; $n_2 = 10$, $\bar{X}_2 = 0.6537$, and $S_2^2 = 0.0319$; $n_3 = 10$, $\bar{X}_3 = 0.5456$, and $S_3^2 = 0.0122$; $n_4 = 10$, $\bar{X}_4 = 0.4365$, and $S_4^2 = 0.0118$. Then, $S_{w1}^2 = 0.0197$, $S_{w2}^2 = 0.0221$, and $S_{w3}^2 = 0.024$.

Consider $t_1 = 5.7910 > t_{0.005}(18) = 2.8784$. So reject H_0 ; that is, Error 1 of BP is larger than the GA-BP with probability of more than 99.5%.

Consider $t_2 = 1.6260 > t_{0.10}(18) = 1.3304$. So reject H_0 ; that is, Error 1 of GA-BP is larger than the DCQA-BP with probability of more than 90%.

Consider $t_3 = 1.5747 > t_{0.10}(18) = 1.3304$. So reject H_0 ; that is, Error 1 of DCQA-BP is larger than the MCQA-BP with probability of more than 90%.

The mean and variance of Error 2 of BP, GA-BP, DCQA-BP, and MCQA-BP are as follows: $n_1 = 10$, $\bar{X}_1 = 1.9815$, and $S_1^2 = 0.0037$; $n_2 = 10$, $\bar{X}_2 = 1.6187$, and $S_2^2 = 0.0272$; $\bar{X}_3 = 1.52$, $S_3^2 = 0.0078$; $n_4 = 10$, $\bar{X}_4 = 1.4$, and $S_4^2 = 0.007$. Then, $S_{w1}^2 = 0.0155$, $S_{w2}^2 = 0.0175$, and $S_{w3}^2 = 0.0074$.

TABLE 4: Contrast between 15 kinds of simulation of damage index value and predicted value.

Order number	Value of laboratory	Prediction value of BP	Prediction value of GA-BP	Prediction value of MCQA-BP	Error value of BP	Error value of GA-BP	Error value of MCQA-BP
1	0.5753	0.6743	0.6642	0.5912	17.21%	15.45%	2.76%
2	0.3735	0.3574	0.3671	0.3719	4.31%	1.71%	0.43%
3	1.0842	1.1218	1.1189	1.0901	3.47%	3.20%	0.54%
4	1.2436	1.2120	1.1203	1.2392	2.54%	9.91%	0.35%
5	1.8643	1.9212	1.9200	1.8765	3.05%	2.99%	0.65%
6	0.607	0.5716	0.5825	0.5923	5.83%	3.23%	2.42%
7	0.5394	0.6067	0.5923	0.5532	12.48%	9.81%	2.56%
8	1.1865	1.2077	1.2065	1.1965	1.79%	1.69%	0.84%
9	1.3987	1.3844	1.3865	1.3899	1.02%	0.87%	0.63%
10	1.4826	1.4863	1.4861	1.4830	0.25%	0.24%	0.03%
11	0.5436	0.5627	0.5512	0.5311	3.51%	1.40%	2.30%
12	0.0817	0.0780	0.079	0.0813	4.53%	3.30%	0.49%
13	0.9045	0.8847	0.8859	0.8899	2.19%	2.06%	1.61%
14	1.4768	1.4950	1.4854	1.4801	1.23%	0.58%	0.22%
15	1.3164	1.3490	1.3387	1.3214	2.48%	1.69%	0.38%

FIGURE 5: The evolutionary process of BP, GA-BP, DCQA-BP, and MCQA-BP.

Consider $t_1 = 6.5161 > t_{0.005}(18) = 2.8784$. So reject H_0 ; that is, Error 2 of BP is larger than the GA-BP with probability of more than 99.5%.

Consider $t_2 = 1.6683 > t_{0.10}(18) = 1.3304$. So reject H_0 ; that is, Error 2 of GA-BP is larger than the DCQA-BP with probability of more than 90%.

Consider $t_3 = 3.1193 > t_{0.005}(18) = 2.8784$. So reject H_0 ; that is, Error 2 of DCQA-BP is larger than the MCQA-BP with probability of more than 99.5%.

The simulation results show that MCQA-BP is significantly superior to BP, GA-BP, and DCQA-BP. The evolutionary process of BP, GA-BP, DCQA-BP, and MCQA-BP is shown in Figure 5.

It can be seen that the fitting results of different models are roughly the same. The trend of these algorithms is the same. It shows that the method of this paper is scientific and effective. The fitting error of the model of this paper is less

than results of other models. So the model in this paper is better than other models.

For the reason that MCQA-BP with super parallel and ultra-high speed which can overcome a series of problems existing in the BP and GA-BP such as slow convergence speed, premature convergence, and bad computational stability optimizes the structure of the neural network effectively, the computational cost of the MCQA-BP is lower than the method of BP, GA-BP, and DCQA-BP in test case 1 and test case 2.

5.2. Test Case 2. The training data is from the results of numerical simulation of 1365 kinds of target board damage under the kinetic energy rod collision of penetration cases. 15 kinds of destruction values predicted by neural network are randomly chosen and shown in Table 4. In the listed penetration value forecast under collision situation, the fitting results of different models are different. The errors between predicted values and the actual values of different models are distinguished from each other. The error of fitting result of MCQA-BP model is less than results of other models. The efficiency of MCQA-BP is further proved.

It can be seen from the table that the prediction effect of MCQA-BP is better than BP network and GA-BP on the same sample data. The error range of MCQA-BP network is less than that of BP neural network and GA-BP neural network under the same learning times. The high efficiency of MCQA-BP network model is fully illustrated.

6. Conclusions

A kind of super parallel ultra-fast BP neural network model and algorithm based on multichain quantum optimization algorithm are proposed. The algorithm makes full use of local time domain feature of the BP network and global optimization search capability of multichain quantum optimization

algorithm to enhance the intelligent search ability of network. It overcomes the disadvantages of BP network to improve the effectiveness of the optimization and accelerate search efficiency and convergence speed. The model controller based on MCQA-BP network can evaluate damage effect of penetration target board of kinetic energy well. The antijamming ability of the model is fine. The MCQA-BP network controller is effective as can be seen from the actual simulation results.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] S. Yu, K. Zhu, and F. Diao, "A dynamic all parameters adaptive BP neural networks model and its application on oil reservoir prediction," *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 66–75, 2008.
- [2] S. S. Haykin, *Neural Networks and Learning Machines*, vol. 3, Pearson Education, Upper Saddle River, NJ, USA, 2009.
- [3] Y. Wang, D. Niu, and L. Ji, "Short-term power load forecasting based on IVL-BP neural network technology," *Systems Engineering Procedia*, vol. 4, pp. 168–174, 2012.
- [4] X. You and X. Cao, "Study of liquid lithium coolant interaction based on BP neural network optimized by genetic algorithm," *Journal of Fusion Energy*, vol. 34, no. 4, pp. 918–924, 2015.
- [5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
- [6] W. Jia, D. Zhao, T. Shen, S. Ding, Y. Zhao, and C. Hu, "An optimized classification algorithm by BP neural network based on PLS and HCA," *Applied Intelligence*, vol. 43, no. 1, pp. 176–191, 2015.
- [7] S. Y. Kung and Y. H. Hu, "A Frobenius approximation reduction method (FARM) for determining optimal number of hidden units," in *Proceedings of the Seattle International Joint Conference on Neural Networks (IJCNN '91)*, vol. 2, pp. 163–168, IEEE, Seattle, Wash, USA, July 1991.
- [8] S. E. Fahlman, "An empirical study of learning speed in back-propagation networks," Tech. Rep. CMU-CS-88-162, Carnegie-Mellon University, 1988.
- [9] R. Salomon, "Adaptive Regelung der Lernrate bei back-propagation," Technische Universität Berlin FB 20, Institut für Softwaretechnik und Theoretische Informatik, 1989.
- [10] F. Dan Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *Proceedings of the International Conference on Neural Networks (IJCNN '97)*, vol. 3, pp. 1930–1935, IEEE, Houston, Tex, USA, June 1997.
- [11] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," in *Proceedings of the IEEE International Conference on Neural Networks (IJCNN '93)*, vol. 1, pp. 586–591, IEEE, San Francisco, Calif, USA, March-April 1993.
- [12] H. Wang, B. Chen, and S. Yao, "Quantitative structure-activity relationship modeling of angiotensin converting enzyme inhibitors by back propagation artificial neural network," *Chinese Journal of Analytical Chemistry*, vol. 34, no. 12, pp. 1674–1678, 2006 (Chinese).
- [13] C. B. Zhao, Z. L. Wang, X. H. Guo, and L. H. Li, "QSAR model of emodin derivatives against the cell line of MCF-7 on artificial neural network," *Journal of Shaanxi University of Technology (Natural Science Edition)*, vol. 23, no. 4, pp. 64–67, 2007 (Chinese).
- [14] X. Cui, Z. Zhang, X. Yuan, Z. Su, and S. Liu, "QSAR studies on nitrobenzene derivatives with PCA-ANN," *Computers and Applied Chemistry*, vol. 22, no. 11, pp. 1038–1040, 2005.
- [15] H. González-Díaz, I. Bonet, C. Terán et al., "ANN-QSAR model for selection of anticancer leads from structurally heterogeneous series of compounds," *European Journal of Medicinal Chemistry*, vol. 42, no. 5, pp. 580–585, 2007.
- [16] F. J. Prado-Prado, X. García-Mera, and H. González-Díaz, "Multi-target spectral moment QSAR versus ANN for antiparasitic drugs against different parasite species," *Bioorganic & Medicinal Chemistry*, vol. 18, no. 6, pp. 2225–2231, 2010.
- [17] G. Ramírez-Galicia, H. Martínez-Pacheco, R. Garduño-Juárez, and O. Deeb, "Exploring QSAR of antiamebic agents of isolated natural products by MLR, ANN, and RTO," *Medicinal Chemistry Research*, vol. 21, no. 9, pp. 2501–2516, 2012.
- [18] Y. Zhang and G. Ruan, "Bernoulli neural network with weights directly determined and with the number of hidden-layer neurons automatically determined," in *Advances in Neural Networks—ISNN 2009*, vol. 5551 of *Lecture Notes in Computer Science*, pp. 36–45, Springer, Berlin, Germany, 2009.
- [19] F. M. Ham and I. Kostanic, *Principles of Neuro Computing for Science & Engineering*, McGraw-Hill, New York, NY, USA, 2001.
- [20] Z. Sun, X. Wang, J. Zhang, and H. Yang, "Prediction and control of equiaxed α in near- β forging of TA15 Ti-alloy based on BP neural network: for purpose of tri-modal microstructure," *Materials Science and Engineering A*, vol. 591, pp. 18–25, 2014.
- [21] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "BP neural network with rough set for short term load forecasting," *Expert Systems with Applications*, vol. 36, no. 1, pp. 273–279, 2009.
- [22] Z. Xiao, R. Xu, and X. Gong, "A practical method for solving the optimal problems in complicated engineering," *Journal of Hydraulic Engineering*, vol. 2, pp. 23–27, 1999 (Chinese).
- [23] Z. Yang, G. Jiang, and H. Wen, "A study on the application of fluid neuron network under nonstandard condition," *Systems Engineering—Theory & Practice*, vol. 19, no. 8, pp. 54–58, 1999 (Chinese).
- [24] H. Ge, "Training the controller based on neural network with genetic algorithm," *Journal of South China Normal University (Natural Science Edition)*, vol. 1, pp. 33–38, 2000 (Chinese).
- [25] J. Li, G. Hu, and X. Sun, "Application of a modified BP algorithm of neural networks in missile-testing expert system," *Computer Applications and Software*, vol. 22, no. 1, pp. 100–102, 2005 (Chinese).
- [26] Z. Xie, Q. Du, F. Ren, X. Zhang, and S. Jamiesone, "Improving the forecast precision of river stage spatial and temporal distribution using drain pipeline knowledge coupled with BP artificial neural networks: a case study of Panlong River, Kunming, China," *Natural Hazards*, vol. 77, no. 2, pp. 1081–1102, 2015.
- [27] S. Wang, D. Chen, P. Hou, X. Wang, Z. Wang, and M. Wei, "Determination of the sodium methylparaben content based on spectrum fluorescence spectral technology and GA-BP neural network," *Spectroscopy and Spectral Analysis*, vol. 35, no. 6, pp. 1606–1610, 2015.