International Journal of Theoretical Physics (2020) 59:1978–1991
https://doi.org/10.1007/s10773-020-04470-9

Quantum Neural Network with Improved Quantum Learning Algorithm

Bu-Qing Chen[1] **· Xu-Feng Niu**[2]

Received: 19 November 2019 / Accepted: 15 April 2020 / Published online: 15 May 2020

Abstract

We present a quantum BP neural network with the universality of single-qubit rotation gate and two-qubit Controlled-NOT gate. Also, we show the process of the BP learning algorithm for the quantum model, and propose an improved BP learning algorithm based on quantum genetic algorithm. The type recognition simulation of the Matlab program shows the efficiencies of the quantum neural network and the improved learning algorithm.

Keywords Quantum neural network · BP algorithm · Quantum genetic algorithm

1 Introduction

With the great improvement of computing power, scholars are more and more enthusiastic about the research of artificial neural network (ANN). ANN is a new information processing and computational model which imitates biological neural system, and can change the structure of internal neurons according to external information. ANN [1] has a strong ability to approximate nonlinear function with any precision even without information about the relationships between data. During the last few years, ANN has grown rapidly and there are many mature kinds of ANN models, such as back propagation (BP) neural network [2–4], hopfield neural network [5–7], radial basis function neural network [8–10] and many others [11].

We are witnessing rapid progress in applications of BP neural networks for tasks like pattern recognition, function approximation, data prediction, picture processing, and so on. BP neural network is currently the most popular ANN model in application based on its

✉ Bu-Qing Chen
cbq8300@163.com

Xu-Feng Niu
361373009@qq.com

1 Teaching and Research Section of Political, PLA Information Engineering University, Luoyang, 471003, Henan, China

2 State Key Laboratory of Integrated Service Networks, Xidian University, Xi’an, 710121, Shaanxi, China

simple structure, dynamic weight vector, learning algorithm and so on. In 1986, Rumelhart et al. [12] firstly proposed BP algorithm which repeatedly adjusted the parameters in network model in order to minimize the error between the actual output vector of the BP network and the desired output vector. BP network is a kind of multi-layer feed-forward ANN which uses BP algorithm to adjust its weight vector of the connections in ANN model. However, the disadvantages of BP network also cannot be neglected, it always falls into locally optimal solution, has slow convergence rate, has uncertain structure and so on [13, 14]. In order to further improve the performance of BP neural network and overcome the shortages of it, we should find a novel way.

Both the increasing data flows of modern computations and the miniaturization of data processing units imply the future high-performance computers will be based on some quantum elements, which obey the laws of quantum mechanics. It is widely believed that computational processes can benefit by exploiting the properties of quantum mechanics [15, 16]. The concept of quantum computing was firstly proposed by famous scholar Feynman [17] and he showed that unique quantum effects such as entanglement and superposition indicated quantum computing was always exponentially faster than traditional computing. In 1994, Shor [18] designed a quantum algorithm to decompose large integer in polynomial time, while the corresponding classical algorithm still requires exponential steps. After that, Grover [19] in 1996 proposed a quantum search algorithm with a time complexity of $O(\sqrt{N})$ for unstructured data, while the corresponding classical search algorithm still has a time complexity of $O(N)$. These two proposed algorithms show the advantages of quantum computing and attract more and more attention on the research of quantum algorithm.

Quantum neural network (QNN) is consisted of a series of quantum operations and quantum or classical weight parameters between neurons, which takes advantages of both classical ANN and quantum computing. In 1995, scholar Kak [20] initially proposed the notion of QNN from a series of directions in ANN research field. Since then, QNN has aroused the research enthusiasm within scholars and there are many kinds of QNNs have been proposed, such as quantum BP neural network [21, 22] quantum hopfield neural network [23–25], universal quantum gate neural network [26, 27], and many others [28, 29]. In 1997, Karayiannis et al. [30] proposed a QNN model based on the idea of quantum superposition properties, which was unique that each hidden neuron was composed of multi-level transfer functions. In this model, each superimposed sigmoid function has a different quantum interval. By adjusting the quantum interval, different data can be mapped to different magnitude order, so that the classification has more degrees of freedom. Matsui et al. [31] in 2000 proposed a QBP model and discussed its performance on solving the 4-bit parity check problem and the function identification problem. In this model, the input is based on quantum bits. However, in many actual problems, the system input is the real vector in Hilbert space. Therefore, Li et al. [32] in 2008 improved the model proposed by Matsui et al. They showed a QBP model based on the universality of single qubit rotation gate and two-qubit controlled-NOT gate, which also has given a formula for converting real numbers into quantum states. In 2014, Schuld et al. [33] proposed a QNN concentrating on quantum hopfield-type networks and the task of associative memory, it outlined the challenge of combining the nonlinear, dissipative dynamics of neural computing and the linear, unitary dynamics of quantum computing. In 2018, Yuan et al. [34] put forward a QNN model consisting of weighting, aggregation, activation and prompting. And the multi-layer feedforward QNN was combined with BP learning to form a multi-sensor integration approach for land-vehicle navigation. In 2019, Ge and Luo [35] designed a logistic QBP neural network based on chaotic sequences incorporating quantum keys.

In this paper, we use a QNN model to develop the common BP neural network, and also propose an improved BP algorithm based on the quantum genetic algorithm (QGA). We also collect some data from the best seller in the Amazon online bookstore, and use the Matlab program to show the efficiencies of the QNN model and the improved algorithm.

The rest of this paper is organized as follows. In Section 2, we introduce the basic contents which are involved in our QBP neural network model. The principle of the proposed QBP neural network is showed in Section 3, and we design the common BP learning algorithm and the improved algorithm based on QGA of QBP neural network in the next Section 4. Then we prove the efficiencies of the QNN model and the improve QGA-QBP algorithm in Section 5. Section 6 concludes this paper.

2 Preliminary Theory

In this section, we will describe some basic contents of the QNN model.

2.1 Quantum Bits

In classical computing, binary numbers 0 and 1 are used to represent information, and they are called bits [36]. In quantum computing, $|0\rangle$ and $|1\rangle$ are used to represent basic states of microcosmic particles, and they are called the ground states of quantum bits. The state of a single quantum bit (qubit) can be represented as a linear combination of the two ground states. The difference between the classical bit and qubit is that qubit can be a linear combination of quantum states which always are called the superposition state. For example,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where, α and β are complex numbers, and they are commonly referred to as the probability amplitudes of quantum states. Namely, when we measure the quantum state $|\psi\rangle$, it will collapse to $|0\rangle$ in the probability of α^2, and collapse to $|1\rangle$ in the probability of β^2. Obviously, we get the relationship between α and β which is $|\alpha|^2 + |\beta|^2 = 1$.

In general, there are 2^n ground states $|x_1x_2...x_n\rangle$ in n-qubit system, and the quantum state in this system is determined by the 2^n probability amplitudes. Similar to single qubit, n-qubit state can also be represented as superpositions of 2^n ground states as following:

$$|\psi\rangle = \sum_{x\in(0,1)^n} a_x|x\rangle, \tag{2}$$

where each a_x is called the probability amplitudes of ground state $|x\rangle$, and they also satisfy the following relationship:

$$\sum_{x\in(0,1)^n} |a_x|^2 = 1 \tag{3}$$

For example, when $n = 3$,

$$|\psi\rangle = a_{000}|000\rangle + a_{001}|001\rangle + a_{010}|010\rangle + ... + a_{111}|111\rangle, \tag{4}$$

we can get

$$|a_{000}|^2 + |a_{001}|^2 + |a_{010}|^2 + ... + |a_{111}|^2 = 1. \tag{5}$$

2.2 Single-qubit Gate

In quantum computing, quantum rotation gate [37] is a kind of important single-qubit gate, the representation of unitary matrix is showed as following:

$$R(\theta) = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}. \tag{6}$$

When $|\phi\rangle = \begin{bmatrix} cos\alpha \\ sin\alpha \end{bmatrix}$, we can acquire the effect of quantum gate as following:

$$R(\theta)|\phi\rangle = \begin{bmatrix} cos(\alpha+\theta) \\ sin(\alpha+\theta) \end{bmatrix}. \tag{7}$$

Obviously, $R(\theta)$ can rotate the phase of a single-qubit by α angles.

2.3 Double-qubit Gate

Controlled-NOT gate [37] is a kind of typical double-qubit gate, it can reverse and select the phase of the target qubit according to the state of the controlled qubit and its expression is as follows:

$$C(m) = \begin{bmatrix} cos(\frac{\pi}{2}m - 2\alpha) & -sin(\frac{\pi}{2}m - 2\alpha) \\ sin(\frac{\pi}{2}m - 2\alpha) & cos(\frac{\pi}{2}m - 2\alpha) \end{bmatrix}. \tag{8}$$

(1) When $m = 1$, $C(m)|\phi\rangle = \begin{bmatrix} sin\alpha \\ cos\alpha \end{bmatrix}$.

(2) When $m = 0$, we have $C(k)|\phi\rangle = \begin{bmatrix} cos\alpha \\ -sin\alpha \end{bmatrix}$.

(3) When $0 < m < 1$, we have

$$C(k)|\phi\rangle = \begin{bmatrix} cos(\frac{\pi}{2}m - \alpha) \\ sin(\frac{\pi}{2}m - \alpha) \end{bmatrix}. \tag{9}$$

2.4 Classical Back Propagation Neural Network

In practice, the most common is the single-hidden-layer BP network [2, 3], which contains an input layer, a hidden layer, and an output layer, so it is also called three-layer BP neural network as shown in Fig. 1.

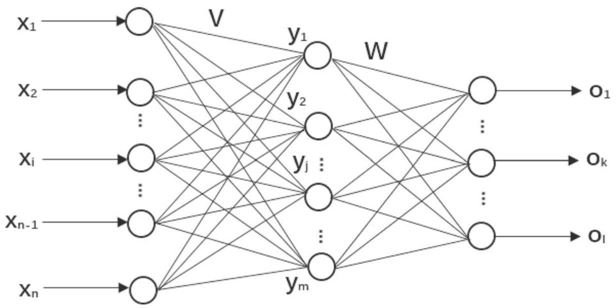

Fig. 1 The model of three-layer BP neural network

Here, input vector is $X = \begin{bmatrix} x_1 & x_2 & ...x_i & ... & x_n \end{bmatrix}^T$, output vector of the hidden layer is $Y = \begin{bmatrix} y_1 & y_2 & ...y_j & ... & y_m \end{bmatrix}^T$, final output vector of the network is $O = \begin{bmatrix} o_1 & o_2 & ...o_k & ... & o_l \end{bmatrix}^T$, the desired output vector is $D = \begin{bmatrix} d_1 & d_2 & ...d_k & ... & d_l \end{bmatrix}^T$. Besides, weight matrix between the input and hidden layers is expressed as $V = (v_{ij})_{nm}$, where v_{ij} is weight between the i-th input node and j-th hidden node. Weight matrix between the hidden and output layers is expressed as $W = (w_{jk})_{ml}$, where w_{jk} is weight between the j-th hidden node and k-th output node.

The mathematical expressions of the BP network are as follows:

$$\begin{aligned} y_j &= f(\sum_{i=1}^{n} v_{ij} x_i) \\ o_k &= f(\sum_{j=1}^{m} w_{jk} y_j), \end{aligned} \tag{10}$$

where, $j = 1, 2, \cdots, m$, $k = 1, 2, \cdots, l$, $f(x)$ and $g(x)$ are the activation functions.

When the actual output and the expected output are different, we should define the error function of the network as

$$E = \frac{1}{2}(D - O)^2 = \frac{1}{2}\sum_{K=1}^{l}(d_k - o_k)^2. \tag{11}$$

Obviously, the error is a function of weight values w_{jk} and v_{ij}. According to the gradient descent algorithm, we get

$$\begin{aligned} \Delta w_{jk} &= -\eta \frac{\partial E}{\partial w_{jk}} = \eta(d_k - o_k)o_k(1 - o_k)y_j \\ \Delta v_{ij} &= -\eta \frac{\partial E}{\partial v_{ij}} = \eta(\sum_{k=1}^{l} w_{jk}(d_k - o_k)g'(\sum_{j=1}^{n} w_{jk}))y_j(1 - y_j)x_i, \end{aligned} \tag{12}$$

where η is the learning rate. Therefore, the updating rules of weight values are as follows:

$$\begin{aligned} w'_{jk} &= w_{jk} + \Delta w_{jk} \\ v'_{ij} &= v_{ij} + \Delta v_{ij}, \end{aligned} \tag{13}$$

where, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$, $k = 1, 2, \cdots, l$.

3 Quantum Neural Network

QNN [21, 22] is a kind of ANN which uses qubits to quantize the training samples based on the topological structure of the classical ANN, at the same time, it uses quantum gates to conduct a series of operations on the parameters of the QNN.

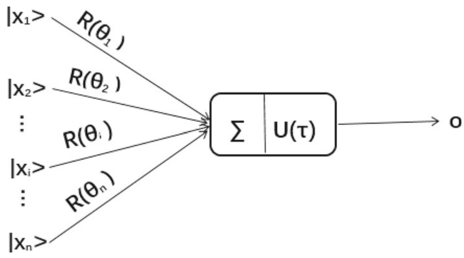

Fig. 2 The model of the quantum neuron

3.1 Quantum Neuron Model

As shown in Fig. 2, it is a quantum neuron. The input is the quantum state $|x_i\rangle$, the output is denoted as real number o. Quantum weight value $R(\theta_i)$ is the quantum rotation gate we have referred in Eq. 6, Controlled-NOT gate $U(\tau) = C(f(\tau))$, where f is the sigmoid function, and function $C(x)$ is explained in Eq. 9.

In order to explain the inner relationship between the input and output of our quantum neuron, we let

$$\sum_{i=1}^{n} R(\theta_i)|x_i\rangle = \begin{bmatrix} cos\alpha \\ sin\alpha \end{bmatrix}, \tag{14}$$

where

$$|x_i\rangle = \begin{bmatrix} cos\alpha_i \\ sin\alpha_i \end{bmatrix},$$

$$\alpha = arg(\sum_{i=1}^{n} R(\theta_i)|x_i\rangle) = arctan\frac{\sum_{i=1}^{n} sin(\alpha_i + \theta_i)}{\sum_{i=1}^{n} cos(\alpha_i + \theta_i)}.$$

After the above result is operated by Controlled-NOT gate, the following formula is obtained

$$U(\tau)\sum_{i=1}^{n} R(\theta_i)|x_i\rangle = \begin{bmatrix} cos(\frac{\pi}{2}f(\tau) - \alpha) \\ sin(\frac{\pi}{2}f(\tau) - \alpha) \end{bmatrix}. \tag{15}$$

Besides, the output of quantum neuron is the probability amplitude of $|1\rangle$, so here we can get the result is $sin(\frac{\pi}{2}f(\tau) - \alpha)$.

Now, we can easily summarize the mathematical formula for this special quantum neuron as follows:

$$o = sin(\frac{\pi}{2}f(\tau) - \alpha) = sin(\frac{\pi}{2}f(\tau) - arg(\sum_{i=1}^{n} R(\theta_i)|x_i\rangle)). \tag{16}$$

3.2 Quantum Back Propagation Neural Network

Based on the designed quantum neuron model and the classical BP neural network structure, the three-layer feedforward QBP neural network is built as shown in Fig. 3.

Here, the input and hidden neurons are quantum, the output neurons are classical. Input vector is $X = \begin{bmatrix} |x_1\rangle & |x_2\rangle & ...|x_i\rangle & ... & |x_n\rangle \end{bmatrix}^T$, output vector of the hidden layer is $Y = \begin{bmatrix} y_1 & y_2 & ...y_j & ... & y_m \end{bmatrix}^T$, final output vector of the network is $O = \begin{bmatrix} o_1 & o_2 & ...o_k & ... & o_l \end{bmatrix}^T$, the desired output vector is $\begin{bmatrix} d_1 & d_2 & ...d_k & ... & d_l \end{bmatrix}^T$. Besides, the training samples are always

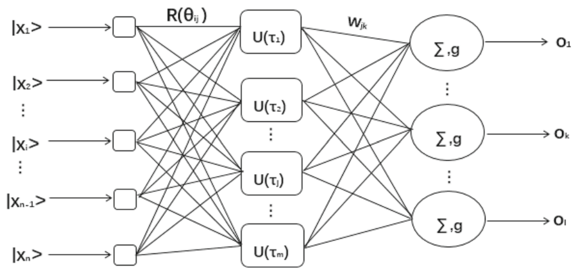

Fig. 3 The model of QBP neural network

real numbers, we should give the specific method about transferring real numbers to quantum states. When the real sample vector is $X = \begin{bmatrix} x_1 & x_2 & ...x_i & ... & x_n \end{bmatrix}^T$, we can get the corresponding quantum input vector $|X\rangle = \begin{bmatrix} |x_1\rangle & |x_2\rangle & ...|x_i\rangle & ... & |x_n\rangle \end{bmatrix}^T$, where

$$|x_i\rangle = \begin{bmatrix} cos(2\pi \cdot sigmoid(x_i)) \\ sin(2\pi \cdot sigmoid(x_i)) \end{bmatrix}. \tag{17}$$

The mathematical expressions of the QBP network are as follows

$$\begin{aligned} y_j &= sin(\frac{\pi}{2} f(\tau_j) - arg(\sum_{i=1}^{n} R(\theta_i)|x_i\rangle)) \\ o_k &= g(\sum_{j=1}^{m} w_{jk} y_j) = g(\sum_{j=1}^{m} w_{jk} sin(\frac{\pi}{2} f(\tau_j) - arg(\sum_{i=1}^{n} R(\theta_i)|x_i\rangle))), \end{aligned} \tag{18}$$

where, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$, $k = 1, 2, \cdots, l$.

4 Learning Algorithm

In this section, we will introduce learning algorithm of the propose QBP neural network based on common gradient descent, and then we make use of the quantum genetic algorithm (QGA) to improve it.

4.1 Back Propagation Algorithm

For the above designed QBP neural network, we need to update and adjust the parameters in the network. In order to update the parameters, the error function is defined as

$$E = \frac{1}{2}(D - O)^2 = \frac{1}{2}\sum_{k=1}^{l}(d_k - o_k)^2. \tag{19}$$

Then we let $|x_i\rangle = \begin{bmatrix} cos\alpha_i \\ sin\alpha_i \end{bmatrix}$, $\phi_j = arctan\frac{\sum_{i=1}^{n} sin(\alpha_i + \theta_{ij})}{\sum_{i=1}^{n} cos(\alpha_i + \theta_{ij})}$, the relationship in Eq. 18 can be written as

$$o_k = g(\sum_{j=1}^{m} w_{jk} sin(\frac{\pi}{2} f(\tau_j) - \phi_j)). \tag{20}$$

In order to obtain a simple parameter adjustment formula, we set

$$\begin{aligned} M_{ij} &= cos(\alpha_i + \theta_{ij}) \\ N_{ij} &= sin(\alpha_i + \theta_{ij}) \\ M_j &= \frac{M_{ij} \cdot \sum_{i=1}^{n} M_{ij} + N_{ij} \cdot \sum_{i=1}^{n} N_{ij}}{(\sum_{i=1}^{n} M_{ij})^2} \\ N_j &= \frac{\sum_{i=1}^{n} N_{ij}}{\sum_{i=1}^{n} M_{ij}}. \end{aligned} \tag{21}$$

According to the principle of BP algorithm, we can get

$$\begin{aligned} \Delta\theta_{ij} &= -\eta \frac{\partial E}{\partial \theta_{ij}} = -\eta \cdot \sum_{k=1}^{l} (d_k - o_k) g' w_{jk} cos(\frac{\pi}{2} f(\tau_j) - \phi_j) \cdot \frac{M_j}{1 + N_j^2} \\ \Delta\tau_j &= -\eta \frac{\partial E}{\partial \tau_j} = \frac{\pi}{2} \eta \cdot \sum_{k=1}^{l} (d_k - o_k) g' w_{jk} cos(\frac{\pi}{2} f(\tau_j) - \phi_j) f' \\ \Delta w_{jk} &= -\eta \frac{\partial E}{\partial w_{jk}} = \eta (d_k - o_k) g' sin(\frac{\pi}{2} f(\tau_j) - \phi_j), \end{aligned} \tag{22}$$

where, η is the learning rate. Therefore, we acquire the updating rule of parameters in the QBP model, and it is shown as follows:

$$\begin{aligned} \theta'_{ij} &= \theta_{ij} + \Delta\theta_{ij} \\ \tau'_j &= \tau_j + \Delta\tau_j \\ w'_{jk} &= w_{jk} + \Delta w_{jk}. \end{aligned} \tag{23}$$

4.2 Improved Back Propagation Algorithm

Since the gradient descent method is used by BP algorithm, it is easy to fall into a situation of local optimization, and the speed of convergence is too slow, sometimes even it is not convergent. In order to improve the convergence speed and generalization ability of the proposed QBP networks, we should pay more and more attention to various global optimization algorithms.

GA is good at solving the global optimization problem, it can efficiently jump out of the local optimal point to find the global optimal point. However, when the selection, crossover and mutation operators are not appropriate, GA will cost many iterations, have slow convergence, premature convergence and many other defects. In order to overcome some of these defects, we should find an efficient method to improve classical GA for QBP neural network.

Luckily, QGA is a kind of fusion algorithm that introduces efficient quantum computing into GA, uses quantum state to encode, and selects quantum rotating gate to perform genetic operation. Compared with ordinary GA, QGA can keep the population diversity because of the quantum superposition state, and it simplifies the calculation and reduces the operation steps through using the quantum rotating gate. Combining the global search QGA and the precise search BP algorithm, we can minimize the training error of the QBP neural network and effectively avoid the training falling into a local optimization situation.

(1) Quantum coding

According to the probability amplitude , single-qubit can be expressed as $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, and m-qubit can be written as $\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix}$, where $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, 2, \cdots, m$. m-qubit contain the information of 2^m quantum states at the same time. For example, when the probability amplitude of 3-qubit is as follows

$$\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{5}} \\ \frac{\sqrt{2}}{\sqrt{5}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{5}} \end{bmatrix}. \tag{24}$$

we can easily obtain the quantum system is

$$\begin{aligned} |\phi\rangle &= \frac{2}{\sqrt{30}}|100\rangle + \frac{1}{\sqrt{30}}|001\rangle + \frac{2}{\sqrt{30}}|010\rangle + \frac{1}{\sqrt{30}}|011\rangle \\ &\quad + \frac{2\sqrt{2}}{\sqrt{30}}|100\rangle + \frac{2\sqrt{2}}{\sqrt{30}}|110\rangle + \frac{\sqrt{2}}{\sqrt{30}}|101\rangle + \frac{\sqrt{2}}{\sqrt{30}}|111\rangle, \end{aligned} \tag{25}$$

and when we measure the system $|\phi\rangle$ based on the basis $\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |110\rangle, |101\rangle, |111\rangle\}$, the probabilities that we get $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|110\rangle$, $|101\rangle$, $|111\rangle$ are $\frac{4}{30}, \frac{1}{30}, \frac{4}{30}, \frac{1}{30}, \frac{8}{30}, \frac{8}{30}, \frac{2}{30}, \frac{2}{30}$, respectively. Obviously, the 3-qubit system shown in Eq. 24 can contain information of 8 quantum states at the same time. Therefore, we choose the above way to code in our improved algorithm.

When we use QGA to optimize the initial m parameters of QBP neural network, these parameters are called chromosomes and each element of a chromosome is called a gene. In our model, each chromosome is coded by qubits as follows $q_j^t = \begin{bmatrix} \alpha_{11}^t & \cdots & \alpha_{1k}^t & \alpha_{21}^t & \cdots & \alpha_{2k}^t & \cdots & \alpha_{m1}^t & \cdots & \alpha_{mk}^t \\ \beta_{11}^t & \cdots & \beta_{1k}^t & \beta_{21}^t & \cdots & \beta_{2k}^t & \cdots & \beta_{m1}^t & \cdots & \beta_{mk}^t \end{bmatrix}$, where q_j^t is the j-th chromosome of the t-th generation, k represents is the number of qubits coding each gene, m is the number of genes on the chromosome.

The qubit code of each individual in the population is initialized to $\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$, which means that all possible states expressed by each chromosome are equal.

(2) Quantum rotation gate

The dynamic adjustment strategy of the quantum rotating angle is showed in Table 1, where x_i and $best_i$ are the i-th bit of the current chromosome and the best chromosome, respectively, $f(x)$ is the fitness function, $\Delta\theta_i$ is magnitude of the rotating angle and $s(\alpha_i, \beta_i) = \alpha_i\beta_i$ is direction of the rotating angle.

(3) Simulation

The specific flowchart of our improved BP algorithm on the basis of the QGA is shown in Fig. 4.

5 The Experiment Results

In this section, we will prove the efficiency of the QBP neural network and the improved QGA-QBP learning algorithm through simulating a typical example based on the Matlab program.

Table 1 The dynamic adjustment strategy of the quantum rotating angle

x_i	$best_i$	$f(x) > f(best)$	$\Delta\theta_i$	$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	false	0	0	0	0	0
0	0	true	0	0	0	0	0
0	1	false	0.01π	+1	-1	0	±1
0	1	true	0.01π	-1	+1	±1	0
1	0	false	0.01π	-1	+1	±1	0
1	0	true	0.01π	+1	-1	0	±1
1	1	false	0	0	0	0	0
1	1	true	0	0	0	0	0

5.1 Example

We selecte the Amazon's best-seller list released in October 2019 as the analysis object, and capture relevant samples of the top 50 best-selling books. The popularity of books is related to many factors. In this case, we choose eight factors, namely the publisher country, format, price, content capacity, publisher, relate score, number of comments, and the date of publication. In order to verify the performances of the proposed QNN and its improved

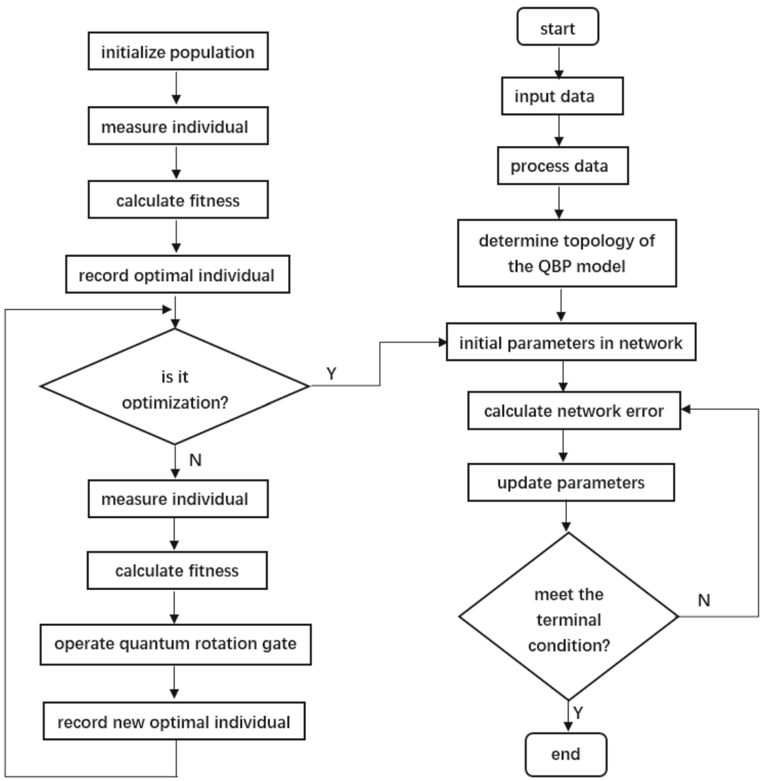

Fig. 4 The training of QBP neural network base on improved QGA-QBP learning algorithm

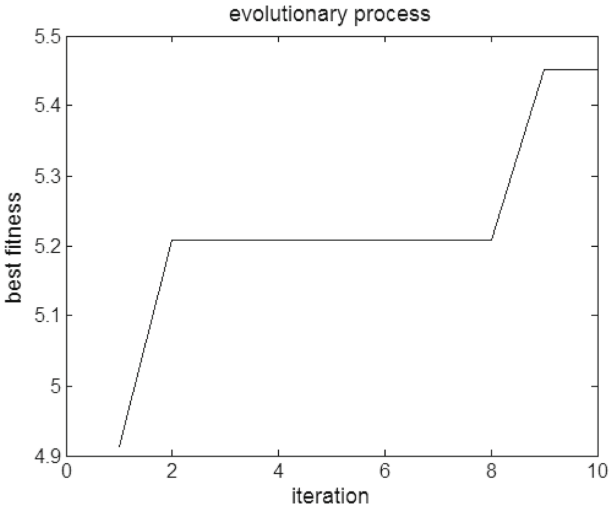

Fig. 5 The evolution of QGA-QBP learning algorithm

algorithm QGA-QBP, we eliminate 5 incomplete samples and divide the remaining samples into 3 categories. The specific classification standards are as follows: the top 10 books belong to type 1 with high best-selling degree, the books ranking between 11 and 35 belong to type 2 with medium best-selling degree, and the books ranking between 36 and 45 belong to type 3 with common best-selling degree.

5.2 Simulation Comparison Experiment

In order to ensure the validity of the following simulation comparison results, we select the same network topology structure with the same parameters. Based on the above example, the input number of our network should be 8, and the output results should represent the different 3 categories, here we choose binary 100, 010, 001 to respectively represent type 1, 2, 3. Therefore, the topologies of QBP and common BP neural network all are 8-17-3.

(1) The evolution of quantum genetic algorithm
During the process of simulation, the size of population is $N = 10$, each individual of the population contained $S = 204$ genes (parameters of QBP neural network), fitness function is the inverse of mean squared error (MSE), termination generation is 10. The evolution of QGA based on the QBP neural network is shown in Fig. 5, the fitness continues to increase during each iteration, and we also can gain the optimized parameters of our network.

(2) Efficiency of the quantum back propagation neural network
We will use the BP, QBP and QGA-QBP neural networks to simulate the above example of type identification, so as to verify that the proposed QBP neural network has better performance than the common BP neural network and the improved QGA is more efficient than common BP algorithm used in QBP neural network. In the process of simulation, the limiting error is set to 0.01, the maximum iteration step is set to 5000, and the learning rate is taken from the elements of set {0.01, 0.02, ..., 0.1}. The above example is simulated 50 times for each learning rate by the proposed QBP and common BP neural network, respectively. As shown in Fig. 6, the simulation results

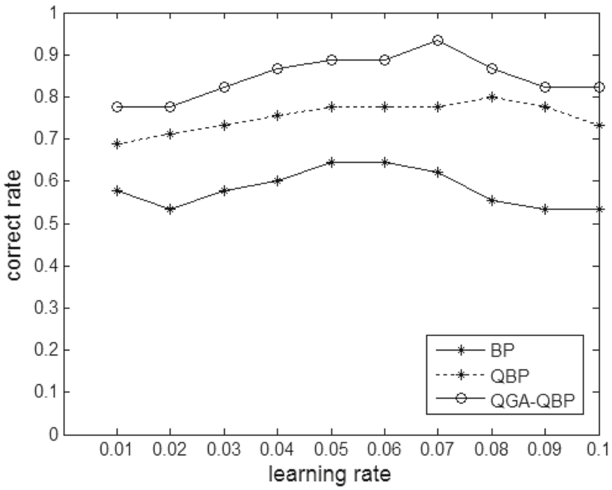

Fig. 6 The training of QBP neural network base on improved QGA-BP learning algorithm

clearly show that the type recognition accuracy of QGA-QBP is higher than that of QBP neural network on the whole, and the recognition rate of QBP is generally higher than that of common BP neural network on the whole. Besides, the correct rate of QGA-QBP neural network can reach 93.3 percent, the highest accuracy rate of QBP neural network is 80 percent, while the highest accuracy rate of common BP neural network is just 64 percent. More importantly, the QGA-QBP neural network has a minimum recognition rate of 77.8 percent, which is better than common BP network's maximum recognition capability.

6 Conclusion

In this paper, we propose a QBP neural network model, it is composed of quantum neurons, classical neurons, and their connections. We also use the quantum rotation gate and Controlled-NOT gate instead of classical weight and threshold to process data, and the input vector is quantum state instead of real number in our quantum network. Besides, we show the specific mathematic process of the common BP algorithm and also propose an improved algorithm based on the principle of QGA. We also capture some data from the internet to prove the abilities of our quantum model and the improved algorithm.

References

1. Hsu, K.L., Gupta, H.V., Sorooshian, S.: Artificial neural network modeling of the rainfall-runoff process. Water Resour. Res. **31**(10), 2517–2530 (1995)
2. Feng, C.X.J., Gowrisankar, A.C., Smith, A.E., et al.: Practical guidelines for developing BP neural network models of measurement uncertainty data. J. Manuf. Syst. **25**(4), 239–250 (2006)

3. Wang, X.Q.: Study of construction project bidding based on the BP neural network improved by GA. China Civil Eng. J. **40**(7), 93–98 (2007)
4. Gong, K., Guan, J.H., Kim, K., et al.: Iterative PET image reconstruction using convolutional neural network representation. IEEE Transactions on Medical Imaging 38(3), 675–685 (2019)
5. Giulia, M.: Hopfield neural network. Int. J. Addict. **33**(2), 481–488 (1998)
6. Su, Y.F.: Integrating a scale-invariant feature of fractal geometry into the Hopfield neural network for super-resolution mapping. Int. J. Remote Sens. **40**(23), 1–22 (2019)
7. Frigura, M., Frigura, M.F., Balcu, I., et al.: Algorithm for solving economical and environmental dispatch problems of thermal power plants. IOP Conf. Series Earth Environ. Sci. **219**, 012010 (2019)
8. Yang, F., Paindavoine, M.: Implementation of an RBF neural network on embedded eystems: real-time face tracking and identity verification. IEEE Trans. Neural Netw. **14**(5), 1162–1175 (2003)
9. Mojtaba. M., Ali, G.: Hybrid routing scheme using imperialist competitive algorithm and RBF neural networks for VANETs. Wirel. Netw. **25**(5), 2831–2849 (2019)
10. Li, J.B., Liu, X.G.: Melt index prediction by RBF neural network optimized with an adaptive new ant colony optimization algorithm. J. Appl. Polymer Sci. **119**(5), 3093–3100 (2012)
11. Ding, D.W., Yao, X.L., Wang, N.: Adaptive synchronization of fractional-order complex-valued uncertainty dynamical network with coupling delay. Int. J. Theor. Phys. **58**(7), 2357–2371 (2019)
12. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
13. AlDuais, M.S., Yaakub, A.R., Yusoff, N.: Dynamic training rate for backpropagation learning algorithm. IEEE Malaysia International Conference on Communications. 277–282 (2013)
14. Tian, X.S., Zhang, T.J.: Methods to improve BP neural network. J. Huaihua Univ. **25**(2), 126–130 (2006)
15. Wang, T.Y., Ma, J.F., Cai, X.Q.: The postprocessing of quantum digital signatures. Quant. Inf. Process. **16**(1), 19 (2017)
16. Niu, X.F., Zhang, J.Z., Xie, S.C., Chen, B.Q.: A third-party e-payment protocol based on quantum multi-proxy blind signature. Int. J. Theor. Phys. **57**(8), 2563–2573 (2018)
17. Feynman, R.: Simulating physics with computers. Int. J. Theor. Phys. **21**(6-7), 467–488 (1982)
18. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994)
19. Grover, L.K.: A fast quantum mechanical algorithm for database search. Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, pp. 212–219. ACM Press, New York (1996)
20. Kak, S.C.: On quantum neural computing. Inform. Sci. **83**(3-4), 143–160 (1995)
21. Zuo, X.G., Zhang, Z.X.: Image compression method based on quantum BP network. Comput. Eng. **38**(13), 205–211 (2012)
22. Sun, W., He, Y.J., Meng, M.: A novel quantum neural network model with variable selection for short term load forecasting. Appl. Mech. Mater. **20**, 612–617 (2010)
23. Akazawa, M., Tokuda, E., Asahi, N., et al.: Quantum hopfield network using single-electron circuits–a novel hopfield network free from the local-minimum difficulty. Analog Integr. Circ. Sig. Process **24**(1), 51–57 (2000)
24. Behrman, E.C., Gaddam, K., Steck, J.E., et al.: Microtubules as a quantum hopfield network. Frontiers Collection 351–370 (2006)
25. Qin, X., Ma, Y.Q.: Tricritical points and reentry in the quantum hopfield neural-network model. Commun. Theor. Phys. **34**(2), 217–222 (2000)
26. Gao, Z.C., Gong, S.R.: License plate character recognition based on neural network with quantum gate. Comput. Eng. **34**(23), 227–229 (2008)
27. Li, S., Zhang, P., Li, B., et al.: Application of universal quantum gate neural network in gear fault diagnosis. China Mech. Eng. **26**(6), 773–777 (2015)
28. Altman, C., Pykacz, J., Roman, R.Z.: Superpositional quantum network topologies. Int. J. Theor. Phys. **43**(10), 2029–2040 (2004)
29. Zhou, R.G.: Quantum competitive neural network. Int. J. Theor. Phys. **49**(1), 110–119 (2010)
30. Karayiannis, N.B.: Purnshothaman. G.: Fuzzy pattern classification using feed forward neural networks with multilevel hidden neurons. IEEE Int. Neural Netw. **5**(2), 127–132 (1994)
31. Matsui, N., Kouda, N., Nishimura, H.: Neural network based on QBP and its performance. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks **3**, 247–252 (2000)
32. Li, P.C., Li, S.Y.: Learning algorithm and application of quantum BP neural networks based on universal quantum gates. J. Syst. Eng. Electron. **19**(1), 167–174 (2008)
33. Schuld, M., Sinayskiy, I., Petruccione, F.: The quest for a quantum neural network. Quantum Inf. Process **13**(11), 2567–2586 (2014)
34. Yuan, D., Cai, L., Li, M., et al.: Multi-sensor integration based on a new quantum neural network model for land-vehicle navigation. Neuroquantology **16**(6) (2018)

35. Ge, B., Luo, H.B.: Image encryption application of chaotic sequences incorporating quantum keys. Int. J. Autom Comput https://doi.org/10.1007/s11633-019-1173-z (2019)
36. Niu, X.F., Zhang, J.Z., Xie, S.C.: A quantum multi-proxy blind signature scheme based on entangled four-qubit Cluster state. Commun. Theor. Phys. **70**(1), 43–48 (2018)
37. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)