Zhao, J., Zhang, Y., Shao, C., Wu, Y., Guo, G., & Guo, G. (2019). Building quantum neural networks based on swap test. *Physical Review A*, *100*, 012334. https://doi.org/10.1103/PhysRevA.100.012334

Building quantum neural networks based on a swap test

Jian Zhao,[1,2] Yuan-Hang Zhang,[3] Chang-Peng Shao,[4,*] Yu-Chun Wu,[1,2,†] Guang-Can Guo,[1,2] and Guo-Ping Guo[1,2,5,‡]

[1]*Key Laboratory of Quantum Information, Chinese Academy of Sciences, School of Physics, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China*

[2]*CAS Center for Excellence in Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China*

[3]*School of the Gifted Young, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China*

[4]*Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China*

[5]*Origin Quantum Computing Hefei, Anhui 230026, People's Republic of China*

(Received 15 May 2019; published 23 July 2019)

An artificial neural network, consisting of many neurons in different layers, is an important method to simulate the human brain. Usually, one neuron has two operations: one is linear, the other is nonlinear. The linear operation is the inner product and the nonlinear operation is represented by an activation function. In this work, we introduce a kind of quantum neuron whose inputs and outputs are quantum states. The inner product and activation operator of the quantum neurons can be realized by quantum circuits. Based on the quantum neuron, we propose a model of a quantum neural network in which the weights between neurons are all quantum states. We also construct a quantum circuit to realize this quantum neural network model. A learning algorithm is proposed meanwhile. We show the validity of the learning algorithm theoretically and demonstrate the potential of the quantum neural network numerically.

DOI: 10.1103/PhysRevA.100.012334

I. INTRODUCTION

Artificial neural networks can be traced back to the McCulloch-Pitts (M-P) neurons proposed in 1943 [1]. Based on M-P neurons, Rosenblatt in 1957 proposed the perceptron model with a learning algorithm [2]. So far, artificial neural networks have had certain theoretical bases [3,4] and extensive practical applications ranging from modeling, classification, and pattern recognition to multivariate data analysis [5,6].

The quantum neural network, proposed by Kak [7] in 1995, is a class of neural network that combines quantum information theory and artificial neural networks. Different models related to quantum neural networks have been developed [8–16]. Among these models, Ref. [9] is a perceptron model with quantum input, quantum output, and weights represented by operators, in which the concrete construction is not explained; Ref. [15] uses quantum computing to achieve the potential acceleration of classical neural networks, and Ref. [16] is based on the actual physical device to construct an analog classical neural network. However, there is still no uniform standard for the rigorous definition of quantum neural networks.

Recently, Ref. [17] introduced a strategy for using quantum phase estimation to get the information for the inner product of two quantum states. Inspired by this work, we introduce a definition of a quantum neuron with quantum states as input states, weights, and a single-particle state as the output state. Accordingly we propose a quantum neural network which can be represented by quantum circuits. Besides, through theoretical analysis and the numerical experiment, we demonstrate the validity of the learning algorithm.

Our starting point is to assume that there is a large number of quantum states, each of which is labeled by a quantum state. Given these data as the training set, our goal is to predict the label of an unknown input state. It is convenient for our proposed quantum neurons to process quantum data directly. And it does not cost the classical computing resources to perform trained quantum neural networks. If using classical neural networks, one may need the method of quantum-state tomography to reconstruct the quantum data [18], which is a highly complex task itself.

This proposed neuron adapts to different kinds of data flexibly. When quantum states as the quantum data are labeled by real numbers rather than quantum states, we can slightly modify the measured strategy to realize classical outputs. Things get more complicated when both data and labels are classical. If using this proposed neuron we need to consider the state preparation problem, which requires controlling the amplitude of the desired quantum state to realize effectiveness [19,20]. A method that makes state preparation simple is to limit the structure of the data [21], in which data are limited to vectors with binary value components.

The paper is organized as follows. At the end of this section we briefly state the notations used in this paper. In Sec. II, we describe the swap test and its quantum circuit. In Sec. III, we construct a quantum neuron according to our proposed definition, and then we analyze the property of this proposed quantum neuron. The proof process is described

[*] cpshao@amss.ac.cn

[†] wuyuchun@ustc.edu.cn

[‡] gpguo@ustc.edu.cn

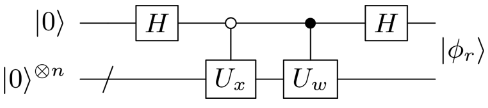

FIG. 1. Quantum circuit to prepare $|\phi_r\rangle$.

in Appendix A. In Sec. IV, based on the construction of a quantum neuron we construct a kind of feedforward neural network and a quantum circuit model representing the specific quantum neural network. We give quantitative estimations of success probability and fidelity theoretically. Some details are presented in Appendix B. We describe the training process of the quantum neural network in Sec. V. And in Sec. VI we present an experiment for numerical simulation. At last in Sec. VII, we draw the conclusions of this paper.

Notation. We use capital italic roman letters, $A, B, \ldots$, for matrices, lower case italic roman letters, $x, y, \ldots$, for column vectors, and Greek letters $\alpha, \beta, \ldots$ for scalars. For a scalar α, we denote by $\mathrm{Re}\,\alpha$ and $\mathrm{Im}\,\alpha$ the real and imaginary parts of α, respectively. Given a column vector x, x^T denotes its transpose and $x^\dagger \triangleq (\bar{x})^T$ is its conjugate transpose, and similarly for a given matrix A. Specifically, for the unitary transformation U, $U^\dagger = U^{-1}$. A quantum state $|x\rangle \in \mathbb{C}^{2^n}$ is regarded as the normalized vector. We write $R_Y(\beta) = \begin{bmatrix} \cos\frac{\beta}{2} & -\sin\frac{\beta}{2} \\ \sin\frac{\beta}{2} & \cos\frac{\beta}{2} \end{bmatrix}$ and $R_Z(\gamma) = \begin{bmatrix} e^{-i\frac{\gamma}{2}} & 0 \\ 0 & e^{i\frac{\gamma}{2}} \end{bmatrix}$.

II. SWAP TEST AND ITS QUANTUM CIRCUIT

The swap test method has been applied widely to quantum machine learning [22–24]. In this section, we describe the swap test and its quantum circuit.

Let $|x\rangle, |w\rangle \in \mathbb{C}^{2^n}$ be two quantum states that are prepared by unitary operators U_x and U_w, respectively. That is, $|x\rangle = U_x|0\rangle^{\otimes n}$, $|w\rangle = U_w|0\rangle^{\otimes n}$. The swap test is a technique that can be used to estimate $\langle x|w\rangle$. The basic procedure can be stated as follows:

Step 1. Prepare the state

$$|\phi_r\rangle = \frac{1}{\sqrt{2}}(|+\rangle|x\rangle + |-\rangle|w\rangle). \tag{1}$$

The quantum circuit to prepare $|\phi_r\rangle$ is simple (see Fig. 1). We denote the unitary to prepare $|\phi_r\rangle$ as U_{ϕ_r}.

Step 2. Construct the unitary transformation

$$\begin{aligned} G_r &= (I^{\otimes(n+1)} - 2|\phi_r\rangle\langle\phi_r|)(Z \otimes I^{\otimes n}) \\ &= U_{\phi_r}(I^{\otimes(n+1)} - 2|0\rangle^{\otimes(n+1)}\langle 0|^{\otimes(n+1)})U_{\phi_r}^\dagger(Z \otimes I^{\otimes n}), \end{aligned} \tag{2}$$

where $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ is the Pauli Z matrix. The circuit to implement G_r is represented in Fig. 2. As for the unitary operator $I^{\otimes(n+1)} - 2|0\rangle^{\otimes(n+1)}\langle 0|^{\otimes(n+1)}$, we can run it in the circuit shown in Fig. 3.

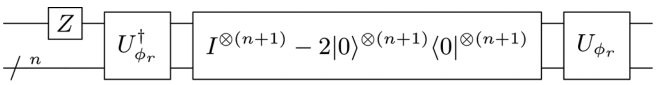
Z
n
$U_{\phi_r}^\dagger$
$I^{\otimes(n+1)} - 2|0\rangle^{\otimes(n+1)}\langle 0|^{\otimes(n+1)}$
U_{ϕ_r}

FIG. 2. Quantum circuit to implement G_r.

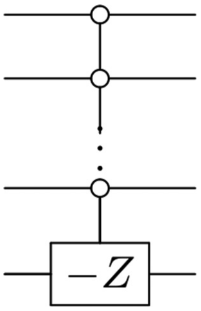

FIG. 3. Quantum circuit to run $I^{\otimes(n+1)} - 2|0\rangle^{\otimes(n+1)}\langle 0|^{\otimes(n+1)}$.

The state $|\phi_r\rangle$ can be rewritten as

$$|\phi_r\rangle = \tfrac{1}{2}(|0\rangle(|x\rangle + |w\rangle) + |1\rangle(|x\rangle - |w\rangle)). \tag{3}$$

The amplitude of $|0\rangle$ is $\sqrt{1 + \mathrm{Re}\langle x|w\rangle}/\sqrt{2}$, and the amplitude of $|1\rangle$ is $\sqrt{1 - \mathrm{Re}\langle x|w\rangle}/\sqrt{2}$. We denote $|u\rangle$ and $|v\rangle$ as the normalized states of $|x\rangle + |w\rangle$ and $|x\rangle - |w\rangle$, respectively. Then there is a real number $\theta_r \in [0, \pi/2]$ such that

$$|\phi_r\rangle = \sin\theta_r|0\rangle|u\rangle + \cos\theta_r|1\rangle|v\rangle. \tag{4}$$

Moreover, θ_r satisfies $\cos\theta_r = \sqrt{1 - \mathrm{Re}\langle x|w\rangle}/\sqrt{2}$, i.e.,

$$\mathrm{Re}\langle x|w\rangle = -\cos 2\theta_r. \tag{5}$$

We apply the Schmidt decomposition method to the quantum state $|\phi_r\rangle$, and we can decompose it into

$$|\phi_r\rangle = \frac{-\boldsymbol{i}}{\sqrt{2}}(e^{\boldsymbol{i}\theta_r}|w_+\rangle - e^{-\boldsymbol{i}\theta_r}|w_-\rangle), \tag{6}$$

where $|w_\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle|u\rangle \pm \boldsymbol{i}|1\rangle|v\rangle)$. Besides, it is easy to check that

$$G_r|w_\pm\rangle = e^{\pm\boldsymbol{i}2\theta_r}|w_\pm\rangle. \tag{7}$$

This means $|w_\pm\rangle$ are the eigenstates of G_r. The information of θ is contained in the arguments of the eigenvalues.

Step 3. Use the quantum phase estimation algorithm to estimate θ. The quantum circuit is shown in Fig. 4.

In Fig. 4, t is an integer that relates to the precision, and FT is the quantum Fourier transform. The control gate G_r^j should be regarded as a composition of a series of controlled gates $G_r^{2^i}$ by viewing the ith qubit in the first register as the control qubit, where $i = 0, \ldots, t-1$.

By Eqs. (6) and (7), the output of the quantum phase estimation is an approximate of

$$|\psi_r\rangle = \frac{-\boldsymbol{i}}{\sqrt{2}}(e^{\boldsymbol{i}\theta_r}|y_r\rangle|w_+\rangle - e^{-\boldsymbol{i}\theta_r}|2^t - y_r\rangle|w_-\rangle), \tag{8}$$

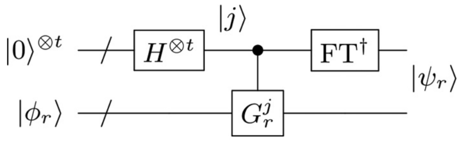

FIG. 4. Quantum phase estimation to estimate θ.

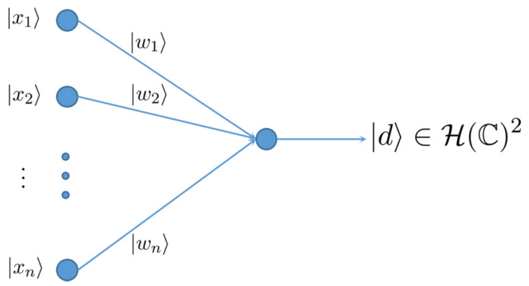

FIG. 5. Structure of a quantum neuron, where $|d\rangle = f(\langle x|w\rangle)$ is the output state.

where $y_r \in [0, 2^{t-1}]$ and $y_r\pi/2^{t-1}$ is an approximate of $2\theta_r$. By Eq. (5), we have

$$\mathrm{Re}\langle x|w\rangle \approx -\cos(\pi y_r/2^{t-1}). \tag{9}$$

Note that $\mathrm{Im}\langle x|w\rangle = -\mathrm{Re}\langle x|\boldsymbol{i}|w\rangle$; thus, the proposal to estimate the real part of the inner product is also suitable to estimate $\mathrm{Im}\langle x|w\rangle$. We only need to consider the state $|\phi_i\rangle = \frac{1}{\sqrt{2}}(|+\rangle|x\rangle - \boldsymbol{i}|-\rangle|w\rangle)$. Finally, we obtain a $y_i \in [0, 2^{t-1}]$ such that

$$\mathrm{Im}\langle x|w\rangle \approx -\cos(\pi y_i/2^{t-1}). \tag{10}$$

For convenience, the corresponding parameters, unitaries, and quantum states used to estimate $\mathrm{Im}\langle x|w\rangle$ are accordingly denoted by $\theta_i, y_i, U_{\phi_i}, G_i, U_{\psi_i}$, and $|\phi_i\rangle, |\psi_i\rangle$.

III. CONSTRUCTION OF THE QUANTUM NEURON

A. Definition of the quantum neuron

A classical neuron can be treated as a function that maps a vector $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ to a real value $z = f(x^T w)$, where $w = (w_1, \dots, w_n)^T \in \mathbb{R}^n$ and f is usually a nonlinear function. $\{x_i\}_{i=1}^n$ and $\{w_i\}_{i=1}^n$ are called the input values and synaptic weights, respectively. The function f is called the activation function. Similarly, we propose the definition of quantum neuron as follows:

Definition 1. Let $|w\rangle = |w_1, \dots, w_n\rangle \in (\mathbb{C}^2)^{\otimes n}$ be a product state. We denote $\mathcal{B}(0,1) = \{a \in \mathbb{C} : |a| \leqslant 1\}$. Assuming that f is a map from $\mathcal{B}(0,1)$ to the subspace of $\mathbb{C}^2$ with unit norm, then the map

$$F : \mathbb{C}^{2^n} \to \mathbb{C}^2,$$
$$|x\rangle \mapsto f(\langle x|w\rangle), \tag{11}$$

is called an n-variable quantum neuron.

In the n-variable quantum neuron, we call $|x\rangle$ the input state, $\{|w_i\rangle\}_i$ the (synaptic) weight states, and $f(\langle x|w\rangle)$ the output state. The map f plays the role of the activation function in defining the quantum neuron. Figure 5 shows the basic structure of the quantum neuron.

Assuming that $a \in \mathbb{C}$, a commonly used activation function in this paper is

$$f(a) = R_Z\left(-\frac{\pi}{2}\right)R_Z(\arccos -\mathrm{Im}a)R_Y(\arccos -\mathrm{Re}a)|0\rangle$$
$$= \begin{bmatrix} \cos\left(\frac{\arccos -\mathrm{Re}a}{2}\right)e^{\boldsymbol{i}(\frac{\pi}{4} - \frac{\arccos -\mathrm{Im}a}{2})} \\ \sin\left(\frac{\arccos -\mathrm{Re}a}{2}\right)e^{-\boldsymbol{i}(\frac{\pi}{4} - \frac{\arccos -\mathrm{Im}a}{2})} \end{bmatrix}. \tag{12}$$

The operator $R_Z(-\pi/2)$ is added to make sure that if $a \in \mathbb{R}$ then

$$f(a) = \begin{bmatrix} \cos\left(\frac{\arccos -a}{2}\right) \\ \sin\left(\frac{\arccos -a}{2}\right) \end{bmatrix} \in \mathbb{R}^2.$$

B. Realization of the output state in the quantum circuit

Now we assume that the activation function f is defined by Eq. (12). Let $|x\rangle \in \mathbb{C}^{2^n}$ be an input state and $|w\rangle \in (\mathbb{C}^2)^{\otimes n}$ be a weight state. In this section, we show how to realize $f(\langle x|w\rangle)$ in the quantum circuit.

We first show how to realize $f(\langle x|w\rangle)$ in the quantum circuit in the ideal case, then extend it into the general case. By ideal, we mean both $\frac{\arccos -\mathrm{Re}\langle x|w\rangle}{2\pi}$ and $\frac{\arccos -\mathrm{Im}\langle x|w\rangle}{2\pi}$ can be represented in binary form with t bits precisely. As a result, a swap test can approximate these two values with no error; i.e., Eqs. (9) and (10) are exact.

By Eqs. (9), (10), and (12),

$$f(\langle x|w\rangle) = R_Z(-\pi/2)R_Z(\pi y_i/2^{t-1})R_Y(\pi y_r/2^{t-1})|0\rangle. \tag{13}$$

To prepare the state (13), first we consider $|\psi_r\rangle|0\rangle$, where $|\psi_r\rangle$ is given in Eq. (8). We want to generate the state $R_Y(\pi y_r/2^{t-1})|0\rangle$ in the third register of $|\psi_r\rangle|0\rangle$ by viewing $|y_r\rangle$ and $|2^t - y_r\rangle$ as control registers. That is to obtain the following transformation:

$$|\psi_r\rangle|0\rangle = \frac{-\boldsymbol{i}}{\sqrt{2}}(e^{i\theta_r}|y_r\rangle|w_+\rangle - e^{-i\theta_r}|2^t - y_r\rangle|w_-\rangle)|0\rangle$$
$$\mapsto |\psi_r\rangle R_Y(\pi y_r/2^{t-1})|0\rangle.$$

The control rotation generated by $|y_r\rangle$ gives $R_Y(\pi y_r/2^{t-1})$ directly. However, the control rotation generated by $|2^t - y_r\rangle$ gives $R_Y(\pi(2^t - y_r)/2^{t-1}) = -XR_Y(\pi y_r/2^{t-1})X$. To modify this, it suffices to add a control X and control $-X$ gate. More precisely, assuming that $y_r' \in \{y_r, 2^t - y_r\}$ and $y_r' = \sum_{j=0}^{t-1} 2^j y_{r,t-j-1}'$ in binary form, then the control qubit is $|y_{r,0}'\rangle$. If $y_{r,0}' = 0$, then we know $y_r' = y_r$ and we just apply the control rotation $R_Y(\pi y_r'/2^{t-1})$ to $|0\rangle$. If $y_{r,0}' = 1$, then we know $y_r' \in \{2^{t-1}, 2^t - y_r\}$. In this case, we apply the X gate to $|0\rangle$ first, then apply the control rotation $R_Y(\pi y_r'/2^{t-1})$, and finally apply $-X$ to the result. The quantum circuit is shown in Fig. 6(a).

If we consider $|\psi_i\rangle R_Y(\pi y_r/2^{t-1})|0\rangle$, then based on the fact that $R_Z(\pi(2^t - y_i)/2^{t-1})|0\rangle = -XR_Z(\pi y_i/2^{t-1})X$ and the above analysis, we can generate $|\psi_i\rangle R_Z(\pi y_i/2^{t-1})R_Y(\pi y_r/2^{t-1})|0\rangle$ by the quantum circuit of Fig. 6(b).

Finally, we conclude the above two procedures in Fig. 7 by adding $R_Z(-\pi/2)$ to generate $f(\langle x|w\rangle)$, where $R_{Y_{f_r}}(t)$ and $R_{Z_{f_i}}(t)$ are short for the control operators used in Figs. 6(a) and 6(b), respectively.

Generally, $\frac{\arccos -\mathrm{Re}\langle x|w\rangle}{2\pi}$ and $\frac{\arccos -\mathrm{Im}\langle x|w\rangle}{2\pi}$ cannot be written in binary form precisely. And y_r and y_i only give approximates of them. By introducing measurements to the original circuit, the quantum circuit given in Fig. 8 returns an approximate of $f(\langle x|w\rangle)$ with high probability. For a detailed proof see Appendix A.

Theorem 1. Let $|x\rangle \in \mathbb{C}^{2^n}$ be a quantum state and $|w\rangle \in (\mathbb{C}^2)^{\otimes n}$ be a product state. Let $t = m + \lceil \log_2(2 + \frac{1}{\sigma}) \rceil$ be the number of ancilla qubits used in the quantum phase

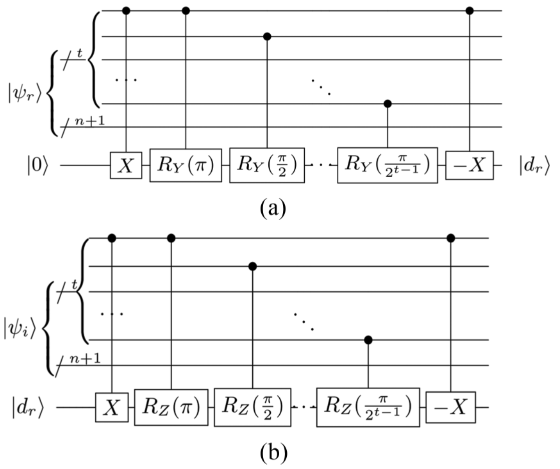

FIG. 6. The quantum neuron to generate $R_Z(\pi y_i/2^{t-1})|d_r\rangle$, where $|d_r\rangle = R_Y(\pi y_r/2^{t-1})|0\rangle$.

estimation, where $\sigma \in (0, 1)$ and $m \in \mathbb{Z}^+$. Assume that $|\tilde{d}\rangle$ is the state obtained by the quantum circuit given in Fig. 8. Then with success probability at least $1 - \sigma$, we have $\||\tilde{d}\rangle - |d\rangle\| \leqslant \pi/2^{m-1}$, where $|d\rangle = f(\langle x|w\rangle)$.

In Fig. 8, the purpose of performing measurements is simply to convert the mixed state (A1) in the ancillary registers into a pure state $|\tilde{d}\rangle$ that is close to $f(\langle x|w\rangle)$. However, it is unnecessary to record or store the measured results, which makes it possible to perform quantum neurons without the classical resources.

One thing worth noting is that the quantum neuron model defined by Fig. 8 can be used to analyze quantum data with real number labels through analyzing the measured results $|\tilde{d}\rangle$. More precisely, we assume that $|\tilde{d}\rangle = p_0|0\rangle + p_1|1\rangle$ is the output of Fig. 8. By Eq. (12), if we perform measurements on $|\tilde{d}\rangle$, then we can estimate

$$|p_1|^2 \approx \sin^2\left(\frac{\arccos -\mathrm{Re}\langle x|w\rangle}{2}\right) = \frac{1 + \mathrm{Re}\langle x|w\rangle}{2}.$$

The probability $|p_1|^2$ characterizes the closeness between $|\tilde{d}\rangle$ and $|1\rangle$. It can be viewed as the label of the input state $|x\rangle$. Note that to solve the classification problems with classical neural networks, we need to calculate a function of the inner product between the input and the weight. However, this inner product is already included in $|p_1|^2$. Thus classical classification problems can also be solved by a quantum neuron. Especially for binary classification problems, we can simply define the label of $|x\rangle$ as a quantum state, e.g., $|0\rangle$ or $|1\rangle$.

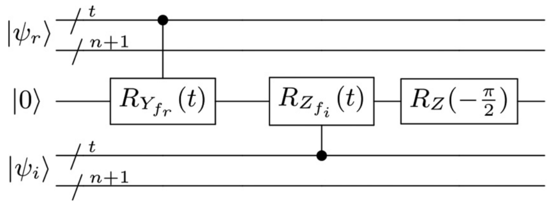

FIG. 7. The quantum neuron in the ideal case, where $|d\rangle = f(\langle x|w\rangle)$.

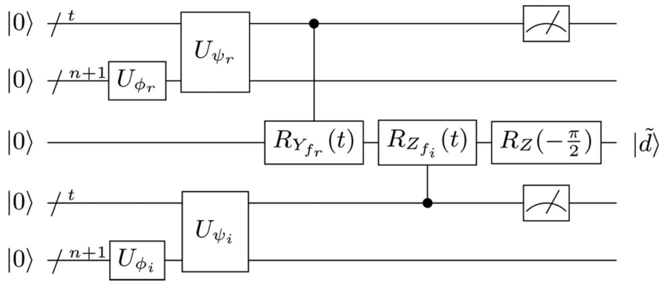

FIG. 8. The quantum neuron in the general case.

IV. CONSTRUCTION OF THE QUANTUM NEURAL NETWORK

The classical feedforward neural network has been used to process data to simulate unknown nonlinear functions [25–27]. In this section we introduce a quantum feedforward neural network to accomplish a similar task.

Let $\mathcal{M} \triangleq \{|x^i\rangle : i = 1, \dots, q\} \subset \mathbb{C}^{2^n}$ be a quantum data set. We want to apply some kind of quantum feedforward neural network to capture the property and structure of $\mathcal{M}$ theoretically. More precisely, suppose that the information of $\mathcal{M}$ is included in an unknown function F_0 mapping $\mathcal{M}$ to a product state space with dimensions 2^s, that is,

$$\begin{aligned} F_0 : \quad & M \to (\mathbb{C}^2)^{\otimes s} \\ & |x^i\rangle \mapsto |d^i\rangle = \big|d_1^i, \dots, d_s^i\big\rangle. \end{aligned} \tag{14}$$

Our purpose is to construct a neural network based on the quantum neuron to simulate F_0 efficiently.

Let $|x\rangle \in \mathcal{M}$ be the input state and it is allowed to be entangled. For convenience, we assume $|x\rangle$ is a product state; that is, $|x\rangle = |x_1, x_2, \dots, x_n\rangle$. The state $|x\rangle$ constitutes the input layer, i.e., the zeroth layer, of the quantum neural network. We denote it as $|z^{(0)}\rangle$. Suppose we have $K - 1$ hidden layers and one output layer. The output layer is also known as the Kth layer. We denote the number of neurons in the kth layer as p_k, where $k = 1, \dots, K$ and $p_K = s$.

For $k = 1, \dots, K$, the jth neuron in the kth and $(k-1)$th layers are connected by an edge with weight $|w_{ij}^{(k)}\rangle$, where $i = 1, \dots, p_{k-1}$, $j = 1, \dots, p_k$. The state of each neuron in the kth layer is determined by the weights and the states of the $(k-1)$th layer. Thus, if we denote $|z_j^{(k)}\rangle$ as the state of the jth

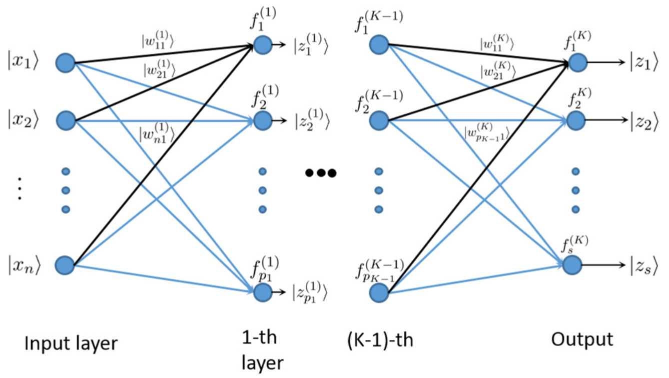

FIG. 9. The quantum feedforward neural network.

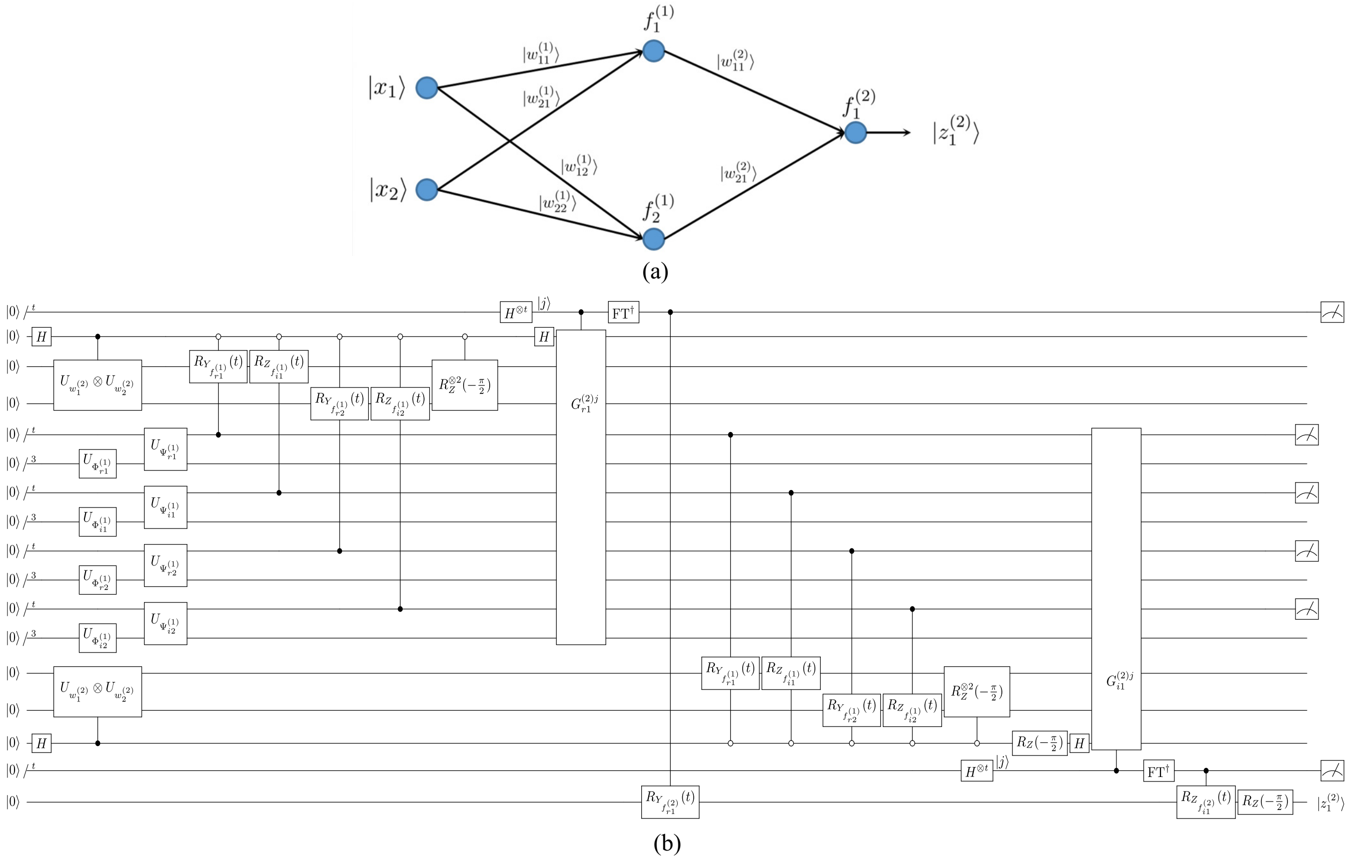

FIG. 10. Construction of the quantum feedforward neural network. Here the input states are $|x_1\rangle$ and $|x_2\rangle$, and the output state is $|z_1^{(2)}\rangle$. (a) Quantum neural network model with three neurons. (b) The quantum neural network represented by a circuit. The transformations $G_{r1^{(2)}}$ and $G_{i1}^{(2)}$ are controlled by $4t$ qubits compared to before (see Fig. 11).

neuron in the kth layer, then

$$\left|z_j^{(k)}\right\rangle = f_j^{(k)}\left(\left\langle z^{(k-1)}\middle|w_j^{(k)}\right\rangle\right), \tag{15}$$

where $|z^{(k-1)}\rangle = |z_1^{(k-1)},\dots,z_{p_{k-1}}^{(k-1)}\rangle$, $|w_j^{(k)}\rangle = |w_{1j}^{(k)},\dots,w_{p_{k-1}j}^{(k)}\rangle$, and $f_j^{(k)}$ is defined by Eq. (12). Figure 9 shows the basic structure of the quantum neural network.

Example. We set $n = 2$, $p_1 = 2$, $K = 2$, and $p_2 = s = 1$. In this case the quantum neural network and the corresponding quantum circuit are shown in Figs. 10(a) and 10(b), respectively.

In the construction of circuits we use the strategy of postponing measurement. To be specific, we postpone the measured process of each neuron in all hidden layers until the last layer. In Fig. 10(b) we postpone $4t$ measured results in the first layer.

The strategy of postponing measurement is necessary. Suppose we want to get the output state of the neuron in hidden layers; we need to measure the corresponding qubits to convert the mixed state to a random pure state. Without postponing measurement we cannot use the method of the swap test to get the subsequent output states, which means the neural network is interrupted. This implies that the intermediate state is unreadable in the quantum neural network and we do not care about the state of hidden layer neurons naturally.

In this quantum neural network, we give quantitative estimations of success probability and fidelity for the output state. Its proof is presented in Appendix B.

Theorem 2. Given a quantum neural network as defined in Fig. 9, suppose the number of the neurons in the kth layer is p_k. Let $p = \max\{p_1,\dots,p_K\}$, $\epsilon \in (0,1)$ and $\sigma \in (0,1)$. Set $m = \lceil \log_2[(\frac{2\pi^2 p^2}{\epsilon})^{2^{K-1}}\pi]\rceil + 1$ and $t = m + \lceil \log_2(2 + \frac{Kp}{\sigma})\rceil$. Then with success probability at least $1-\sigma$ we have the fidelity

$$\left\| \left|z^{(K)}\right\rangle - \left|\tilde{z}^{(K)}\right\rangle \right\| \leqslant \epsilon. \tag{16}$$

V. TRAINING PROCESS

In this section, we introduce the training process of the proposed quantum neural network. We transform the quantum neural network into a quantum circuit containing parameters to be optimized. The training process of parametrized quantum circuits has been used in many quantum algorithms [28–30].

Suppose the quantum neural network has n neurons in the input layer and has s neurons in the output layer. In the training

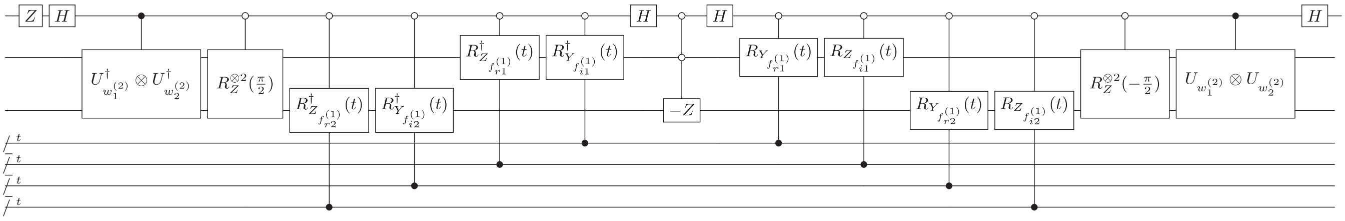
Z
H
$U^\dagger_{w_1^{(2)}} \otimes U^\dagger_{w_2^{(2)}}$
$R_Z^{\otimes 2}(\frac{\pi}{2})$
$R^\dagger_{Z_{f_{r2}^{(1)}}}(t)$
$R^\dagger_{Y_{f_{i2}^{(1)}}}(t)$
$R^\dagger_{Z_{f_{r1}^{(1)}}}(t)$
$R^\dagger_{Y_{f_{i1}^{(1)}}}(t)$
H
$-Z$
H
$R_{Y_{f_{r1}^{(1)}}}(t)$
$R_{Z_{f_{i1}^{(1)}}}(t)$
$R_{Y_{f_{r2}^{(1)}}}(t)$
$R_{Z_{f_{i2}^{(1)}}}(t)$
$R_Z^{\otimes 2}(-\frac{\pi}{2})$
$U_{w_1^{(2)}} \otimes U_{w_2^{(2)}}$
H
t
t
t
t

FIG. 11. The transformation $G_{r1}^{(2)}$.

process, we choose the mean square loss

$$\mathcal{L}(\mathcal{M},\mathcal{W}) = \frac{1}{q}\sum_{i=1}^{q} ||z^i\rangle - |d^i\rangle|^2$$
$$= \frac{1}{q}\sum_{i=1}^{q}(2 - 2\text{Re}\langle z^i|d^i\rangle). \tag{17}$$

Each input state $|x^i\rangle \in \mathcal{M}$ has a fixed label $|d^i\rangle = |d_1^i,\dots,d_s^i\rangle$. Each output state $|z^i\rangle = |z_1^i,\dots,z_s^i\rangle$ produced by quantum circuits can be closed to $|\tilde{z}^i\rangle$ with high success probability according to Theorem 2, where $|\tilde{z}^i\rangle$ is the ideal output state decided by all the weights $|w_j^{(k)}\rangle$ and the activation function f defined in expression (12).

Our goal is to find a set $\mathcal{W} \triangleq \{|w_j^{(k)}\rangle : k = 1,\dots,K, j = 1,\dots,p_k\}$ of weight states such that they minimize the mean-square loss.

Since $|w_j^{(k)}\rangle = |w_{1j}^{(k)},\dots,w_{p_{k-1}j}^{(k)}\rangle$ is a product state, we assume that

$$\big|w_{ij}^{(k)}\big\rangle = e^{\boldsymbol{i}\delta_{ijk}}R_Z(\gamma_{ijk})R_Y(\beta_{ijk})|0\rangle \tag{18}$$

for some parameters $\beta_{ijk},\gamma_{ijk},\delta_{ijk} \in [0,2\pi)$ to be tuned. We denote the parameter vector by $\theta = (\theta_1,\dots,\theta_L)^{\text{T}}$, where $\theta_i \in \{\beta_{ijk},\gamma_{ijk},\delta_{ijk}\}$ and $L = |\{\beta_{ijk},\gamma_{ijk},\delta_{ijk}\}|$. As in Figs. 8 and 10(b), the output state $|z^i\rangle$ always can be obtained by performing a unitary transformation, denoted by $U^i(\theta)$, to the initial state $|0\rangle$ and adding some measurements. Let $|Z^i\rangle = U^i(\theta)|0\rangle$; then the output state $|z^i\rangle$ is decided by the parameter vector θ and measurement results. We denote the map from $|0\rangle$ to $|z^i\rangle$ by $F^i(\theta)$. Thus, $\mathcal{L}$ can be viewed as a function of θ.

We explain the training process as follows.

Step 1. Initial value selection. Randomly try the initial parameter vector θ and choose the optimal parameter denoted by $\theta^{(0)}$ such that the value of $\mathcal{L}$ is minimum.

The value of $\text{Re}\langle z^i|d^i\rangle$ for each vector θ can be obtained by reusing the quantum swap test. Then we classically calculate and compare the different values of $\mathcal{L}(\theta)$ to obtain the optimal initial value.

Step 2. Iteration process. We use the gradient descendent method. In the $(i+1)$th step,

$$\theta_l^{(i+1)} = \theta_l^{(i)} - \eta\frac{\partial\mathcal{L}}{\partial\theta_l}, \tag{19}$$

where η is an adjustable positive step length and $l = 1,\dots,L$. Combining expressions (17) and (19), we can use the quantum-classical hybrid method to acquire the gradient:

$$\frac{\partial\mathcal{L}}{\partial\theta_l} = -\frac{2}{q}\sum_{i=1}^{q}\frac{\text{Re}\partial\langle d^i|z^i\rangle}{\partial\theta_l}, \tag{20}$$

$$\frac{\partial|z^i\rangle}{\partial\theta_l} = \frac{\partial F^i}{\partial\theta_l}|0\rangle. \tag{21}$$

The partial derivative of F^i can be obtained by first deriving the partial derivative of U^k and then adding the corresponding measurements.

To be specific, theoretically for arbitrary unitary transformation, it always can be represented by the basic unitary gates: the single-particle rotation gates and the controlled X gates. For example, if $U^i = (R_X(2g_1(\theta_l))\otimes R_Z(2g_2(\theta_l)))(\text{CNOT})(I \otimes R_Z(2g_3(\theta_{l'})))$, where $g_j(\theta_l) \in [0,2\pi)$ denotes the rotation angle for the single-particle gate in the form of the basic unitary gates, $j = 1,2,3$,

$$\frac{\partial U^i|0\rangle}{\partial\theta_l} = -g_1'(\theta_l)|\boldsymbol{i}(X\otimes I)U^i|0\rangle - g_2'(\theta_l)|\boldsymbol{i}(I\otimes Z)U^i|0\rangle. \tag{22}$$

As in expression (22), we can construct the quantum circuit for the unitary transformations $\boldsymbol{i}(X\otimes I)U^i$ and $\boldsymbol{i}(I\otimes Z)U^i$, respectively. Then we measure and record the corresponding registers, collapsing $\boldsymbol{i}(X\otimes I)U^i|0\rangle$ and $\boldsymbol{i}(I\otimes Z)U^i|0\rangle$ to the states denoted by $|z_{p1}^i\rangle$ and $|z_{p2}^i\rangle$, respectively. At last, we use the swap test to get the value of $\text{Re}\langle d^i|z_{p_1}^i\rangle$ and $\text{Re}\langle d^i|z_{p_2}^i\rangle$ and calculate the gradient of $\mathcal{L}$ by

$$\frac{\text{Re}\partial\langle d^i|z^i\rangle}{\partial\theta_l} = -g_1'(\theta_l)\text{Re}\big\langle d^i\big|z_{p_1}^i\big\rangle - g_2'(\theta_l)\text{Re}\big\langle d^i\big|z_{p_2}^i\big\rangle.$$

VI. NUMERICAL EXPERIMENT: CLASSIFICATION ON A CHECKERBOARD

In this section, we numerically validate our model with the following checkerboard classification task.

Consider a product state $R_Y(\theta_1)|0\rangle \otimes R_Y(\theta_2)|0\rangle$. This state has two parameters $\theta_1,\theta_2 \in [0,2\pi)$, which form a square area $C \triangleq \{[0,2\pi)\times[0,2\pi)\}$. Now, suppose we divide the square C into a 2×2 checkerboard with two disjoint parts:

$$C_0 = \{[0,\pi)\times[0,\pi)\} \cup \{[\pi,2\pi)\times[\pi,2\pi)\},$$
$$C_1 = \{[0,\pi)\times[\pi,2\pi)\} \cup \{[\pi,2\pi)\times[0,\pi)\}. \tag{23}$$

The task is to classify whether the input quantum state is in the region C_0 (labeled by $|0\rangle$) or C_1 (labeled by $|1\rangle$). To this end, we constructed a four-layered quantum neural network: the input layer has two neurons corresponding to the input state, followed by two hidden layers with eight neurons in each of them, and the output layer has one neuron.

During the training process, we randomly generated 10^5 data points and applied the stochastic gradient descent algorithm to minimize the loss function defined in Eq. (17). In this numerical experiment, since the vector forms of samples are

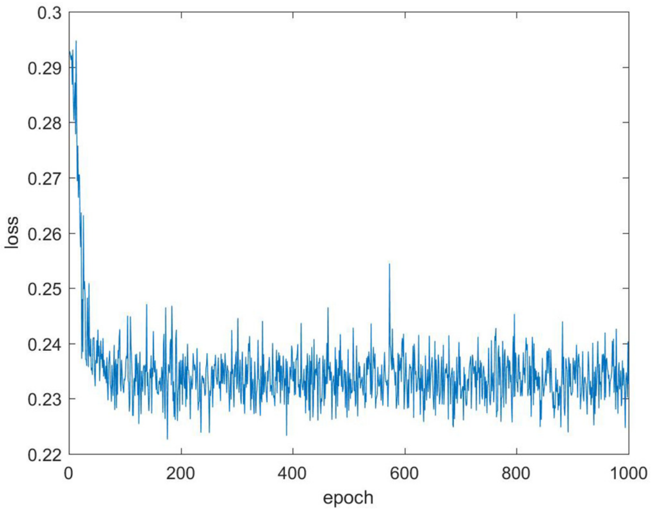

FIG. 12. The learning curve.

known, we calculated the loss function and the gradient in the classical way. The learning curve is shown in Fig. 12, from which we can see that the loss converged to about 0.23.

After training, we further generated 10 000 samples to test our quantum neural network. The result is plotted in Fig. 13, in which the classification accuracy achieved is 99.25%.

VII. CONCLUSIONS

The quantum neural network is introduced and its explicit expression is obtained. The validity of the training process of the neural network is proved theoretically. The numerical example illustrates the potential of this model. Although there exists the process of measurement, we do not need to record or store any measured result, which means performing the quantum neural networks does not cost the resources of classical calculations.

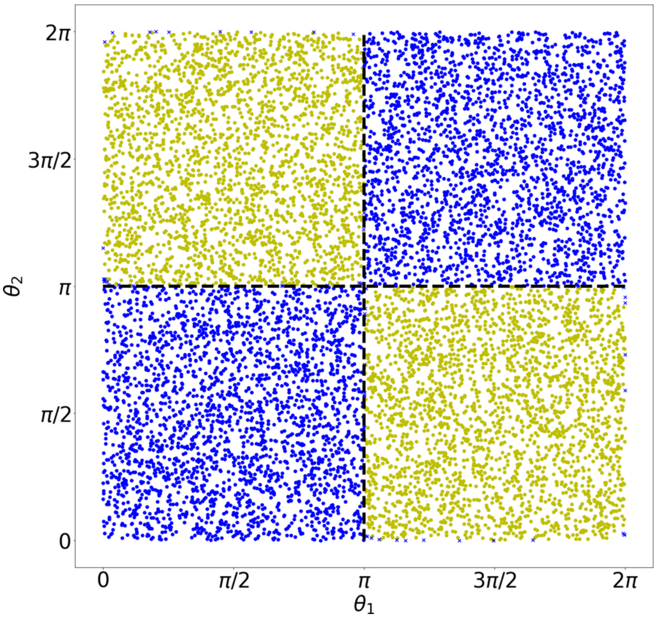

FIG. 13. The testing result: The correct predictions are represented with dots, and the incorrect predictions are labeled with crosses.

This proposed quantum neural network includes some situations of classical neural networks, where the weights constitute a vector belonging to the product state space. And it can be used to process both quantum data with classical labels directly and classical data with classical labels by using state preparation.

A possible future research topic is to generalize the form of the weights in each layer, such as that $|w_j^{(k)}\rangle$ is not limited to the product state. One can also generalize the activation operator f, which still retains validity, or to generalize the output state of the neural network into an entangled state.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (Grant No. 2016YFA0301700) and the Anhui Initiative in Quantum Information Technologies (Grant No. AHY080000).

APPENDIX A: THE PROOF OF THEOREM 1

Proof. Denote $y_r' = \lfloor 2^t\theta_r/\pi \rfloor$ and $y_r'' = 2^t - y_r'$ [see Eq. (4) for the meaning of θ_r]. By quantum phase estimation (see Ref. [31]), $\forall\sigma' \in (0,1)$ we can choose $t = m + \lceil \log_2(2 + \frac{1}{2\sigma'}) \rceil$ and approximate θ_r/π to precision 2^{-m} with probability at least $1-\sigma'$; thus, the exact form of the state $|\psi_r\rangle$ in Eq. (8) should be

$$\frac{-i}{\sqrt{2}}\left[e^{i\theta}\left(\sum_{\substack{\tilde{y}_r':|\tilde{y}_r'-y_r'| \\ \leqslant 2^{t-m}-1}} \beta_{\tilde{y}_r'}|\tilde{y}_r'\rangle + \sum_{\substack{\hat{y}_r':|\hat{y}_r'-y_r'| \\ > 2^{t-m}-1}} \beta_{\hat{y}_r'}|\hat{y}_r'\rangle \right)|w_+\rangle \right.$$
$$\left. - e^{-i\theta}\left(\sum_{\substack{\tilde{y}_r'':|\tilde{y}_r''-y_r''| \\ \leqslant 2^{t-m}-1}} \beta_{\tilde{y}_r''}|\tilde{y}_r''\rangle + \sum_{\substack{\hat{y}_r'':|\hat{y}_r''-y_r''| \\ > 2^{t-m}-1}} \beta_{\hat{y}_r''}|\hat{y}_r''\rangle \right)|w_-\rangle \right]. \quad \text{(A1)}$$

Moreover,

$$\sum_{\substack{\tilde{y}_r':|\tilde{y}_r'-y_r'| \\ \leqslant 2^{t-m}-1}} \frac{|\beta_{\tilde{y}_r'}|^2}{2} \geqslant \frac{1-\sigma'}{2}, \qquad \sum_{\substack{\tilde{y}_r'':|\tilde{y}_r''-y_r''| \\ \leqslant 2^{t-m}-1}} \frac{|\beta_{\tilde{y}_r''}|^2}{2} \geqslant \frac{1-\sigma'}{2}.$$

In $|\psi_r\rangle$, all $\tilde{y}_r'$ provide 2^{-m} approximates of θ_r/π, i.e., $|\tilde{y}_r'/2^t - \theta_r/\pi| \leqslant 2^{-m}$. We also have $\tilde{y}_r'' = 2^t - \tilde{y}_r'$. Applying the control rotation shown in Fig. 6(a) to $|\psi_r\rangle|0\rangle$, then with probability at least $1-\sigma'$ we get

$$|\tilde{d}_r'\rangle = R_Y(\tilde{y}_r'\pi/2^{t-1})|0\rangle \quad \text{(A2)}$$

in the third register. We denote the angle between $|\tilde{d}_r'\rangle$ and $|d_r\rangle := R_Y(2\theta_r)|0\rangle$ in the Bloch sphere as η_r; then

$$\eta_r = \left| \tilde{y}_r' - \frac{2^t\theta_r}{\pi} \right| \frac{\pi}{2^{t-1}} \leqslant \frac{\pi}{2^{m-1}}. \quad \text{(A3)}$$

Thus,

$$\| |\tilde{d}_r'\rangle - |d_r\rangle \| \leqslant \sqrt{2 - 2\cos(\eta_r/2)} = 2\sin(\eta_r/4) \leqslant \pi/2^m. \quad \text{(A4)}$$

Similarly, with probability at least $1-\sigma'$, we can obtain a $\tilde{y}_i'$ such that $|\tilde{y}_i'/2^t-\theta_i/\pi|\leqslant 2^{-m}$. By definition,

$$|\tilde{d}\rangle=R_Z(-\pi/2)R_Z(\tilde{y}_i'\pi/2^{t-1})R_Y(\tilde{y}_r'\pi/2^{t-1})|0\rangle,$$
$$|d\rangle=R_Z(-\pi/2)R_Z(2\theta_i)R_Y(2\theta_r)|0\rangle.$$

Therefore,

$$\begin{aligned}\||\tilde{d}\rangle-|d\rangle\|&\\ &\leqslant \|R_Z(\tilde{y}_i'\pi/2^{t-1})(R_Y(\tilde{y}_r'\pi/2^{t-1})|0\rangle\\ &\quad-R_Z(\tilde{y}_i'\pi/2^{t-1})R_Y(2\theta_r)|0\rangle)\|\\ &\quad+\|R_Z(\tilde{y}_i'\pi/2^{t-1})R_Y(2\theta_r)|0\rangle)\\ &\quad-R_Z(2\theta_i)R_Y(2\theta_r)|0\rangle\|\\ &\leqslant \||\tilde{d}_r'\rangle-|d_r\rangle\|+\|R_Z(\tilde{y}_i'\pi/2^{t-1})-R_Z(2\theta_i)\|\\ &\leqslant \pi/2^m+\pi/2^m=\pi/2^{m-1}.\end{aligned}$$

The success probability is $(1-\sigma')^2>1-2\sigma'$. We choose $\sigma=2\sigma'\in(0,1)$ and $t=m+\lceil\log_2(2+\frac{1}{\sigma})\rceil$. ■

APPENDIX B: THE DETAILS OF THEOREM 2

Lemma 1. Assume that $|x\rangle=|x_1,\ldots,x_n\rangle$, $|\tilde{x}\rangle=|\tilde{x}_1,\ldots,\tilde{x}_n\rangle$, where $\||x_i\rangle-|\tilde{x}_i\rangle\|\leqslant\epsilon$ for all i. Assume that $|w\rangle=|w_1,\ldots,w_n\rangle$. Then

(1) $\||x\rangle-|\tilde{x}\rangle\|\leqslant n\epsilon$.

(2) Let $g(y)=\arccos(-y)$, $y\in[-1,1]$. $\forall\delta\in(0,2)$, if $|y_1-y_2|<\delta$, then

$$|g(y_1)-g(y_2)|\leqslant\pi\sqrt{\delta}/\sqrt{2}.$$

(3) Suppose that $y_r\pi/2^{t-1}$, $y_i\pi/2^{t-1}$, $\tilde{y}_r\pi/2^{t-1}$, and $\tilde{y}_i\pi/2^{t-1}$ are $\pi/2^m$ approximates of $2\theta_r=\arccos-\mathrm{Re}\langle x|w\rangle$, $2\theta_i=\arccos-\mathrm{Im}\langle x|w\rangle$, $\arccos-\mathrm{Re}\langle\tilde{x}|w\rangle$, and $\arccos-\mathrm{Im}\langle\tilde{x}|w\rangle$, respectively, then

$$|\tilde{d}\rangle=R_Z(-\pi/2)R_Z(\tilde{y}_i\pi/2^{t-1})R_Y(\tilde{y}_r\pi/2^{t-1})|0\rangle,$$
$$|d\rangle=R_Z(-\pi/2)R_Z(2\theta_i)R_Y(2\theta_r)|0\rangle,$$

satisfies $\||\tilde{d}\rangle-|d\rangle\|\leqslant\pi/2^{m-1}+\pi\sqrt{n\epsilon}/\sqrt{2}$.

Proof.

(1) We prove the result by induction. The result is true for $n=1$. We denote $|x'\rangle=|x_2,\ldots,x_n\rangle$ and $|\tilde{x}'\rangle=|\tilde{x}_2,\ldots,\tilde{x}_n\rangle$; then by induction $\||x'\rangle-|\tilde{x}'\rangle\|\leqslant(n-1)\epsilon$. Thus,

$$\||x\rangle-|\tilde{x}\rangle\|\leqslant\||x_1,x'\rangle-|\tilde{x}_1,x'\rangle\|+\||\tilde{x}_1,x'\rangle-|\tilde{x}_1,\tilde{x}'\rangle\|\leqslant n\epsilon.$$

(2) Since $|y_1-y_2|\leqslant\delta$, we have $|g(y_1)-g(y_2)|\leqslant\arccos(1-\delta)$. Note that $\cos(\pi\sqrt{\delta}/\sqrt{2})<1-\delta$; then $|g(y_1)-g(y_2)|\leqslant\pi\sqrt{\delta}/\sqrt{2}$.

(3) By step 1, we have $\||x\rangle-|\tilde{x}\rangle\|\leqslant n\epsilon$; thus, $|\langle w|x\rangle-\langle w|\tilde{x}\rangle|\leqslant n\epsilon$. We denote $2\tilde{\theta}_r=\arccos-\mathrm{Re}\langle\tilde{x}|w\rangle$, $2\tilde{\theta}_i=\arccos-\mathrm{Im}\langle\tilde{x}|w\rangle$; then by step 2, $|\tilde{\theta}_r-\theta_r|\leqslant\pi\sqrt{n\epsilon}/2\sqrt{2}$, $|\tilde{\theta}_i-\theta_i|\leqslant\pi\sqrt{n\epsilon}/2\sqrt{2}$. Setting $|d'\rangle=R_Z(-\pi/2)R_Z(2\tilde{\theta}_i)R_Y(2\tilde{\theta}_r)|0\rangle$, then

$$\begin{aligned}\||\tilde{d}\rangle-|d\rangle\|&\leqslant\||\tilde{d}\rangle-|d'\rangle\|+\||d\rangle-|d'\rangle\|\\ &\leqslant\pi/2^{m-1}+\pi\sqrt{n\epsilon}/\sqrt{2}.\end{aligned}$$

This completes the proof. ■

Then combining Lemma 1 and Theorem 1, we give the proof of Theorem 2.

Proof. We denote the error to generate $|z^{(k)}\rangle$ as ϵ_k; then $\epsilon_0=0$. We assume that $m=\lceil\log_2(\pi/\delta)\rceil+1$ for some δ such that $\delta\leqslant\frac{\pi}{\sqrt{2}}\sqrt{\epsilon_k}$ for all $k\geqslant 1$.

By Lemma 1, $\epsilon_1\leqslant p_1\frac{\pi}{2^{m-1}}\leqslant p\delta$. When $k\geqslant 2$ and $\epsilon_{k-1}\leqslant 2$,

$$\begin{aligned}\epsilon_k&\leqslant p_k\left(\frac{\pi}{2^{m-1}}+\frac{\pi}{\sqrt{2}}\sqrt{\epsilon_{k-1}}\right)\\ &\leqslant p\left(\delta+\frac{\pi}{\sqrt{2}}\sqrt{\epsilon_{k-1}}\right)\\ &\leqslant\sqrt{2}\pi p\sqrt{\epsilon_{k-1}}.\end{aligned}$$

Thus,

$$\begin{aligned}\epsilon_k&\leqslant(\sqrt{2}\pi)^{1+\frac{1}{2}+\cdots+\frac{1}{2^{k-2}}}p^{1+\frac{1}{2}+\cdots+\frac{1}{2^{k-1}}}\delta^{\frac{1}{2^{k-1}}}\\ &\leqslant 2\pi^2p^2\delta^{\frac{1}{2^{k-1}}}.\end{aligned}$$

Setting $\epsilon_K=\epsilon$ shows that $\delta=(\epsilon/2\pi^2p^2)^{2^{K-1}}$. And we can check that $\epsilon_k<2\pi^2p^2\delta^{\frac{1}{2^{k-1}}}<\epsilon<2$.

By Theorem 1, if $t=m+\log_2(2+\frac{1}{\sigma'})$, the success probability is $(1-\sigma')^{Kp}$. Let $\sigma=Kp\sigma'\in(0,1)$; then

$$(1-\sigma')^{Kp}=\left(1-\frac{\sigma}{Kp}\right)^{Kp}\geqslant 1-\sigma.$$

■

[1] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. **5**, 115 (1943).

[2] F. Rosenblatt, The perceptron, a perceiving and recognizing automaton Project Para, Cornell Aeronautical Laboratory, 1957 (unpublished).

[3] M. Minsky and S. A. Papert, *Perceptrons: An introduction to computational geometry* (MIT Press, Cambridge, MA, 2017).

[4] J. Hospfield, Neural networks and physical systems with emergent collective computer abilities, Proc. Natl. Acad. Sci. USA **79**, 2554 (1982).

[5] I. A. Basheer and M. Hajmeer, Artificial neural networks: Fundamentals, computing, design, and application, J. Microbiol. Methods **43**, 3 (2000).

[6] H. S. Hippert, C. E. Pedreira, and R. C. Souza, Neural networks for short-term load forecasting: A review and evaluation, IEEE Trans. Power Syst. **16**, 44 (2001).

[7] S. Kak, On quantum neural computing, Inf. Sci. **83**, 143 (1995).

[8] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, Quantum Inf. Process. **13**, 2567 (2014).

[9] M. V. Altaisky, Quantum neural network, arXiv:quant-ph/0107012.

[10] M. Andrecut and M. K. Ali, A quantum neural network model, Int. J. Mod. Phys. C **13**, 75 (2002).

[11] A. da Silva, T. B. Ludermir, and W. R. de Oliveira, Quantum perceptron over a field and neural network architecture selection in a quantum computer, Neural Networks **76**, 55 (2016).

[12] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim, Quantum generalisation of feedforward neural networks, npj Quantum Inf. **3**, 36 (2017).

[13] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik, Quantum neuron: An elementary building block for machine learning on quantum computers, arXiv:1711.11240.

[14] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002.

[15] A. Daskin, A simple quantum neural net with a periodic activation function, in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE, Miyazaki, Japan, 2018), pp. 2887–2891.

[16] G. Tóth, C. S. Lent, P. D. Tougaw, Y. Brazhnik, W. Weng, W. Porod, Ruey-Wen Liu, and Yih-Fang Huang, Quantum cellular neural networks, Superlattices Microstruct. **20**, 473 (1996).

[17] C. Shao, A quantum model for multilayer perceptron, arXiv:1808.10561.

[18] A. I. Lvovsky and M. G. Raymer, Continuous-variable optical quantum-state tomography, Rev. Mod. Phys. **81**, 299 (2009).

[19] A. N. Soklakov and R. Schack, State preparation based on Grover's algorithm in the presence of global information about the state, Opt. Spectrosc. **99**, 211 (2005).

[20] A. N. Soklakov and R. Schack, Efficient state preparation for a register of quantum bits, Phys. Rev. A **73**, 012307 (2006).

[21] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, An artificial neuron implemented on an actual quantum processor, npj Quantum Inf. **5**, 26 (2019).

[22] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411.

[23] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[24] Xin-Ding Zhang, Xiao-Ming Zhang, and Zheng-Yuan Xue, Quantum hyperparallel algorithm for matrix multiplication, Sci. Rep. **6**, 24910 (2016).

[25] T. D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, Neural Networks **2**, 459 (1989).

[26] G. Bebis and M. Georgiopoulos, Feed-forward neural networks, IEEE Potentials **13**, 27 (1994).

[27] G. B. Huang, Q. Y. Zhu, and C. K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in *2004 IEEE International Joint Conference on Neural Networks* (IEEE, Budapest, Hungary, 2004), pp. 985–990.

[28] A. Peruzzo, J. McClean, P. Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[29] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[30] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[31] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2002).