IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 13, 2019, accepted January 24, 2019, date of publication January 31, 2019, date of current version April 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2896316

A Unitary Weights Based One-Iteration Quantum Perceptron Algorithm for Non-Ideal Training Sets

WENJIE LIU[1,2], PEIPEI GAO[2], YUXIANG WANG[1,2], WENBIN YU[1,2], AND MAOJUN ZHANG[3,4]

[1]Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin 541004, China
[4]College of Computer Science and Information Technology, Guangxi Normal University, Guilin 541000, China

Corresponding author: Maojun Zhang (zhang1977108@sina.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502101, Grant 61501247, Grant 61672290, and Grant 71461005, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20171458, in part by the Six Talent Peaks Project of Jiangsu Province under Grant 2015-XXRJ-013, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

ABSTRACT In order to solve the problem of non-ideal training sets (i.e., the less-complete or over-complete sets) and implement one-iteration learning, a novel efficient quantum perceptron algorithm based on unitary weights is proposed, where the singular value decomposition of the total weight matrix from the training set is calculated to make the weight matrix to be unitary. The example validation of quantum gates $\{H, S, T, \text{CNOT}, \text{Toffoli}, \text{Fredkin}\}$ shows that our algorithm can accurately implement arbitrary quantum gates within one iteration. The performance comparison between our algorithm and other quantum perceptron algorithms demonstrates the advantages of our algorithm in terms of applicability, accuracy, and availability. For further validating the applicability of our algorithm, a quantum composite gate which consists of several basic quantum gates is also illustrated.

INDEX TERMS Quantum perceptron, unitary weight, one-iteration learning, non-ideal training set, singular value decomposition, universal quantum gates.

I. INTRODUCTION

With the size of electronic devices becoming smaller and smaller, quantum effects are beginning to interfere in the function of them. The theoretical possibility of quantum computation was firstly explored by Benioff [1] and Feynman [2], and the formalization of the quantum computation model was proposed by Deutsch [3]. The main advantage of quantum computation is that the use of superposition and entanglement, which allied with the linearity of the operators, allows for a powerful form of parallelism to devise algorithms more efficient over the known classical ones. There are two well-known representative algorithms, i.e. Shor's factoring algorithm [4] and Grover's searching algorithm [5]. The first one is an algorithm for integer factorization formulated, which is NP-hard for classical computation; while Grover searching algorithm achieves a quadratic speedup on the unstructured search spaces over the classical counterparts. After that, the combination of quantum mechanics with other fields (such as information processing, encrypted transmission, machine learning, etc.) has also achieved some important results, such as quantum key agreement (QKA) [6], [7], quantum secure direct communication (QSDC) [8], [9], quantum teleportation and remote state preparation (QT&RSP) [10]–[12], quantum steganography (QS) [13]–[15], delegating quantum computation (DQC) [16], [17], and quantum machine learning [18], [19].

Artificial neural network (ANN) [20], also called neural networks (NN), is a computational model used widely in computer science and other research fields. It is based on a large collection of simple neural units (artificial neurons), which is loosely analogous to the observed behavior of a brain's axons. The perceptron [21] is the simplest type of neural network classifiers, which only consists of a single artificial neuron.

The debate on quantum approaches to NN (also called quantum neural network, QNN) emerged in the wake of a booming research field of quantum computation two decades ago. Kak [22] puts forward some preliminary ideas to find a junction between neural networks and quantum mechanics. Since then, a number of proposals have been

The associate editor coordinating the review of this manuscript and approving it for publication was Rajeeb Dey.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

proposed [23]–[26]. However, in addition to the quantum neural (also called quron), the nonlinear dissipative dynamics of neural computation is fundamentally different from the linear single dynamics of quantum computation [27]. Therefore, how to combine the fields of quantum computation and neural networks is a meaningful task for QNN.

The QNN models can be implemented into four different approaches [28]: 1) interpreting the step-function as measurement in order to combine the non-linearity of neural networks and linearity of quantum computation [22]; 2) using quantum circuits for modeling neural networks [29]; 3) describing the basic component of ANNs, the perceptron, with a quantum formalism [30]; 4) modeling ANNs with interacting quantum dots consisting of four atoms sharing two electrons [31]. In this paper, we focus on the third approach of creating a quantum equivalent of classical perceptron. Thus, it can be seen as a simple unit of QNN that harvests the advantages of quantum information processing.

Altaisky [30] firstly proposed a quantum perceptron model to overcome the limitations of the classical perceptron. Although he used the quantum form to represent the classic perceptron model, this proposal is difficult to be extended to a full neural network model. Zhou and Ding [32] proposed a quantum M-P neural network that could compute the *XOR*-function with one neuron, but it does not follow a unitary evolution and the neuron can be efficiently simulated by a classical single layer neural network. After that, an autonomous quantum perceptron algorithm was proposed by Sagheer and Zidan [33], and this algorithm has an iterative rule that can learn to solve a problem in a smaller number of iterations than Altaisky's algorithm. However, its learning weights are non-unitary. Different from the previous models, Siomau [34] introduced an autonomous quantum perceptron based on a set of positive operator-valued measures (POVMs). However, the perceptron could not implement basic quantum gates, such as *NOT* and *Hadamard* gates, which are essential for quantum computation.

In order to solve the non-unitary problem, Seow *et al.* [35] proposed a new efficient quantum perceptron algorithm in 2015. It expels the iterative learning rule by analytically calculating parameters while maintaining unitary requirements. But the perceptron cannot realize arbitrary quantum computation, and the training set is ideal (i.e, the complete training set). Different from the aforementioned quantum perceptron models [30], [32]–[35], a novel unitary weights based quantum perceptron algorithm is proposed in this study, which not only considers the case of non-ideal training sets, but also implements quantum gates $\{H, S, T, CNOT, Toffoli, Fredkin\}$ (i.e., any quantum computation) correctly within one iteration. For further validating the proposed algorithm, a quantum composite gate which consists of several basic quantum gates is also illustrated.

The remainder of this paper is divided into 5 sections. In Sect. II, some preliminaries including quantum computation, classical perceptron model and singular value decomposition, are briefly introduced. In Sect. III, a unitary weights based quantum perceptron algorithm for non-ideal training sets is proposed, and it can realize general quantum computation within one iteration. In Sect. IV, the example validations of some quantum gates $\{H, S, T, CNOT\}$ and $\{Toffoli, Fredkin\}$ are given in detail, and the performance evaluation of our algorithm is conducted in Sect. V. Finally, some remarks and future work are concluded in Sect. VI.

II. PRELIMINARIES

A. QUANTUM COMPUTATION

As we know, the classic bit is the smallest unit in classic information, and its value is either 0 or 1. In quantum computation, *qubit* (quantum bit) is quantum analog of the classic bit, but has two possible values $|0\rangle$ and $|1\rangle$ with a certain probability,

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

where $\left|\alpha^2\right| + |\beta|^2 = 1$, α, β are complex numbers. Since the vectors $|0\rangle$ and $|1\rangle$ can be represented as follows,

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}. \tag{2}$$

The qubit $|\varphi\rangle$ can be expressed in a vector form $|\varphi\rangle = \binom{\alpha}{\beta}$.

Quantum operators over a qubit are all represented by 2×2 unitary matrices. An $n \times n$ matrix U is unitary if $UU^\dagger = U^\dagger U = I$, where $U^\dagger$ is the conjugate transpose of U. For instance, H (*Hadamard*), S (*phase*), and T $(\pi/8)$ operators are important quantum operators over one qubit, and they can be described as 2×2 unitary matrices as below,

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}, \quad S = \begin{pmatrix}1 & 0\\0 & i\end{pmatrix}, T = \begin{pmatrix}1 & 0\\0 & e^{i\pi/4}\end{pmatrix}. \tag{3}$$

The *CNOT* gate operates on a quantum register consisting of 2 qubits, known as the control qubit and the target qubit. If the control qubit is set to 0, then the target qubit is left alone. If the control qubit is set to 1, then the target qubit is flipped, and *CNOT* operator is described in Eq. 4,

$$CNOT = \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix} \tag{4}$$

Quantum parallelism [36] is a fundamental physical property used in many quantum algorithms. In theory, quantum parallelism, which allows quantum computers to evaluate a function $f(x)$ for many different values of x simultaneously, makes the computational power of quantum computers grow exponentially with the increase in the number of quantum bits. It works as follows, suppose $f(x) : \{0, 1\} \rightarrow \{0, 1\}$ is a function with one-bit domain and one-bit range, we define the transformation U_f: $|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$ (shown in Fig. 1), where $\oplus$ indicates addition modulo 2, the first register is called the 'data' register, and the second register is the 'target' register. In some sense, the quantum parallelism enables the evaluation of all the function $f(x)$ at the same time, even though we only evaluate $f(x)$ once.

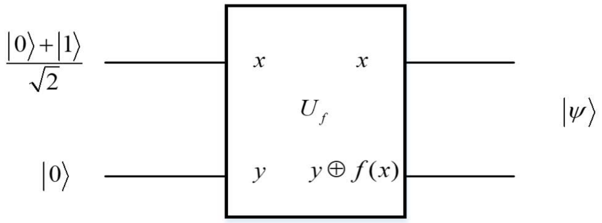


FIGURE 1. Quantum circuit for evaluating $f(0)$ and $f(1)$ simultaneously.

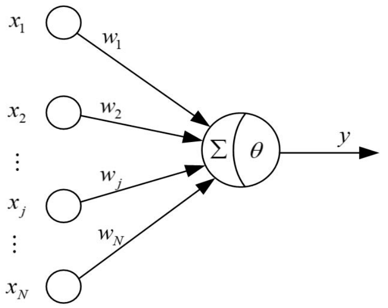


FIGURE 2. The classical perceptron model.

B. CLASSICAL PERCEPTRON

The perceptron [21] is the simplest type of neural network classifiers, which is an algorithm for learning a binary classifier. It consists of N input nodes called neurons with values $x_j = \{1, -1\}, j = 1, \ldots, N$, that feed signals into a single output neuron y (shown in Fig. 2). Each input neuron is connected to the output neuron with a certain strength denoted by a weight parameter $w_j \in [-1, 1]$. The input-output relation is governed by the activation function:

$$\hat{y} = f\left(\sum_{j=1}^{N} w_j x_j - \theta\right), \tag{5}$$

where the bias $\theta \in [-1, 1]$, and $f(\cdot)$ is the activation function. The perceptron learning algorithm works is shown in Algorithm 1.

```
Algorithm 1 The Classical Perceptron Learning Algorithm
Require: Given a training set in the form $\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$, where $x_j$ is an input and $y_j$ is the desired output.
1: The weights $w_j$ and bias $\theta$ are initialized to small random numbers.
2: The output $\hat{y}$ generated according to Eq. 5.
3: The weights are updated according to the rule $w_k(t+1) = w_k(t) + \eta(y - \hat{y})x$, where the $\eta$ is learning rate.
4: If $\hat{y} = y$, the perceptron does not change. Otherwise go to Step 3 to continue training and learning.
```

C. SINGULAR VALUE DECOMPOSITION

In linear algebra, the singular value decomposition(SVD) is a factorization of a real or complex matrix. It is the generalization of the eigendecomposition of a positive semidefinite normal matrix to any matrix via an extension of the polar decomposition. Any rectangular $m \times n$ real or complex matrix M can be decomposed into three matrices and they satisfy:

$$M = U\Sigma V^*, \tag{6}$$

where U is an $m \times m$ real or complex unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V is an $n \times n$ real or complex unitary matrix. The diagonal entries σ_i of Σ are known as the singular values of M. The columns of U and the columns of V are called the left-singular vectors and right-singular vectors of M, respectively. Applications that employ the SVD include computing the pseudoinverse, least squares fitting of data, multivariable control, matrix approximation, and determining the rank, range and null space of a matrix.

Before we introduce our quantum perceptron algorithm in next section, the key notations and descriptions is firstly listed in Table 1.

TABLE 1. Key notations and descriptions involved in the article.

Notation	Description
$\vert\varphi\rangle$	Vector, also known as a *ket*.
$\langle\varphi\vert$	Vector dual to $\vert\varphi\rangle$, also known as a *bra*.
$\vert\varphi\rangle \otimes \vert\psi\rangle$	The tensor product of $\vert\varphi\rangle$ and $\vert\psi\rangle$, which can be abbreviated as $\vert\varphi\rangle\,\vert\psi\rangle$
A^*	Complex conjugate of the A matrix.
$A^\dagger$	Conjugate-transpose of the A matrix, $A^\dagger = (A^T)^*$
$\hat{w}_j$	The j-th weight defined in Eq. 8
$\hat{w}$	The total weight defined in Eq. 9
$\vert y_{output}\rangle$	The output of our perceptron algorithm, defined in Eq. 11
H	*Hadmard* gate
S	*Phase* gate
T	$\pi/8$ gate
CNOT	Controlled-*NOT* gate
Toffoli	The three-qubit gate, also named *CCNOT* gate, defined in Eq. 12 and Fig. 3
Fredkin	The three-qubit controlled-*SWAP* gate, also named *CSWAP* gate, defined in Eq. 13 and Fig. 4

III. THE ONE-ITERATION QUANTUM PERCEPTRON ALGORITHM BASED ON UNITARY WEIGHTS

In order to solve the problem of non-ideal training sets (i.e., the less-complete or over-complete sets) and implement one-iteration learning, we propose a quantum perceptron algorithm based on unitary weights. The algorithm mainly includes: 1) find the conjugate transpose of input, 2) get the total weight, 3) use SVD to decompose weight matrix, 4) get the form of the quantum perceptron. Algorithm 2 shows the detailed procedure of our quantum perceptron algorithm.

A. FIND THE CONJUGATE TRANSPOSE OF INPUT

At the beginning of the algorithm, given a training set in the form $\{(|x_1\rangle, |y_1\rangle), \ldots, (|x_N\rangle, |y_N\rangle)\}$, where $|x_j\rangle$ is an input and $|y_j\rangle$ is the desired output. We can then find the conjugate

Algorithm 2 Quantum Perceptron Learning Algorithm Based on Unitary Weights

```
Require: Given a training set $\{(|x_1\rangle, |y_1\rangle), (|x_2\rangle, |y_2\rangle), \ldots, (|x_N\rangle, |y_N\rangle)\}$, where $|x_j\rangle$ is an input and $|y_j\rangle$ is the desired output.
1: Find the conjugate transpose of input $|x_j\rangle^\dagger = \langle x_j|$.
2: Calculate $\hat{w}_j = |y_j\rangle \otimes \langle x_j|$.
3: Get the total weight $\hat{w} = \sum_{j=1}^{N} \hat{w}_j$.
4: Decompose $\hat{w}$ into $\hat{F}\,\Sigma\hat{w}_{new}$.
5: Pull out the diagonal matrix of singular values and replace it with a unitary matrix $\Sigma_{new}$ with ones in the diagonals and zeroes elsewhere.
6: Get the result of quantum perceptron algorithm $|y_{output}\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_j\rangle$.
```

transpose of input $|x_j\rangle$,

$$|x_j\rangle^\dagger = \langle x_j| \,. \tag{7}$$

B. GET THE TOTAL WEIGHT

After finding the conjugate transpose of input, the main work of this phase is to get the total weight. We take the tensor product of $\langle x_j|$ and $|y_j\rangle$ to create its corresponding weight $\hat{w}_j$,

$$\hat{w}_j = |y_j\rangle \otimes \langle x_j| \,, \tag{8}$$

where $\otimes$ is the tensor product. After calculating the weights, we sum up all of the individual weights, and get the total weight:

$$\hat{w} = \sum_{j=1}^{N} \hat{w}_j. \tag{9}$$

C. USE SVD TO DECOMPOSE WEIGHT MATRIX

In order to preserve the quantum properties of being unitary, we decompose non-unitary matrix $\hat{w}$ into three unitary matrices using SVD. Then we can get two unitary matrices ($\hat{F}$ and $\hat{w}_{new}$) and a diagonal matrix Σ of singular values as below,

$$\hat{w} = \hat{F}\,\Sigma\hat{w}_{new}. \tag{10}$$

D. GET THE FORM OF QUANTUM PERCEPTRON

Without loss of generality, we remove the diagonal matrix and replace it with a unitary matrix Σ_{new} of ones in the diagonals and zeros everywhere else, then get the form of quantum perceptron as follows:

$$|y_{output}\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_j\rangle \,. \tag{11}$$

IV. EXAMPLE VALIDATION OF QUANTUM PERCEPTRON ALGORITHM

As we know, a quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate the quantum information [27]. Therefore, quantum computation is closely related to the implementation of quantum gates. In order to validate that our proposed algorithm can implement general quantum gates (i.e., quantum computation), the universal quantum gate set $\{H, S, T, CNOT\}$, and some complex quantum gates like *Toffoli* and *Fredkin* gates are illustrated as follows.

A. UNIVERSAL QUANTUM GATE SET

The standard set of universal gates consists of the *Hadamard*, *phase*, $\pi/8$ and *CNOT* gates [27], and our perceptron algorithm can realize the function of these universal quantum gates within one iteration, the following is the algorithm validation of the universal gates.

1) *HADAMARD* GATE

Example 1 (Over-Complete): Suppose

$$\begin{aligned}
|x_1\rangle &= |0\rangle\,, |y_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\
|x_2\rangle &= |1\rangle\,, |y_2\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\
|x_3\rangle &= \frac{a\,|0\rangle + b\,|1\rangle}{\sqrt{a^2 + b^2}}\,, \\
|y_3\rangle &= \frac{(a+b)\,|0\rangle + (a-b)\,|1\rangle}{\sqrt{2(a^2+b^2)}}(a = 1, b = 2).
\end{aligned}$$

Perceptron training: According to Algorithm 2, the conjugate transpose of inputs are firstly calculated respectively by Eq. 7, and the weights are solved by Eq. 8,

$$\begin{aligned}
\langle x_1| &= (1, 0)\,, \hat{w}_1 = |y_1\rangle \otimes \langle x_1| = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \\
\langle x_2| &= (0, 1)\,, \hat{w}_2 = |y_2\rangle \otimes \langle x_2| = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} \\
\langle x_3| &= \frac{1}{\sqrt{5}}(1, 1)\,, \hat{w}_3 = |y_3\rangle \otimes \langle x_3| = \frac{1}{\sqrt{2}}\begin{pmatrix} 0.6 & 1.2 \\ -0.2 & -0.4 \end{pmatrix}
\end{aligned}$$

In Step 3, the total weight $\hat{w}$ can be calculated as follows,

$$\begin{aligned}
\hat{w} = \sum_{j=1}^{3} \hat{w}_j &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} \\
&\quad + \frac{1}{\sqrt{2}}\begin{pmatrix} 0.6 & 1.2 \\ -0.2 & -0.4 \end{pmatrix} \\
&= \frac{1}{\sqrt{2}}\begin{pmatrix} 1.6 & 2.2 \\ 0.8 & -1.4 \end{pmatrix}.
\end{aligned}$$

In Step 4, SVD is performed on the weight matrix $\hat{w}$:

$$\hat{F} = \frac{1}{\sqrt{5}}\begin{pmatrix} -3\big/\sqrt{2} & 1\big/\sqrt{2} \\ 1\big/\sqrt{2} & 3\big/\sqrt{2} \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\hat{w}_{new} = \frac{1}{\sqrt{5}}\begin{pmatrix} -1 & -2 \\ 2 & -1 \end{pmatrix}$$

After the process of amending, we can obtain:

$$\hat{F} = \frac{1}{\sqrt{5}}\begin{pmatrix} -3\big/\sqrt{2} & 1\big/\sqrt{2} \\ 1\big/\sqrt{2} & 3\big/\sqrt{2} \end{pmatrix}, \quad \Sigma_{new} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\hat{w}_{new} = \frac{1}{\sqrt{5}}\begin{pmatrix} -1 & -2 \\ 2 & -1 \end{pmatrix}$$

Validation of correctness: The obtained quantum perceptron is used to validate the various inputs. When the perceptron predicting correctly, the output meets $\left|y_{output}\right\rangle = \left|y_j\right\rangle$.

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\hat{w}_{new}\left|x_1\right\rangle \\ &= \frac{1}{\sqrt{5}}\begin{pmatrix} -3\big/\sqrt{2} & 1\big/\sqrt{2} \\ 1\big/\sqrt{2} & 3\big/\sqrt{2} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{5}}\begin{pmatrix} -1 & -2 \\ 2 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= |y_1\rangle \end{aligned}$$

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\hat{w}_{new}\left|x_2\right\rangle \\ &= \frac{1}{\sqrt{5}}\begin{pmatrix} -3\big/\sqrt{2} & 1\big/\sqrt{2} \\ 1\big/\sqrt{2} & 3\big/\sqrt{2} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{5}}\begin{pmatrix} -1 & -2 \\ 2 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= |y_2\rangle \end{aligned}$$

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\hat{w}_{new}\left|x_3\right\rangle \\ &= \frac{1}{\sqrt{5}}\begin{pmatrix} -3\big/\sqrt{2} & 1\big/\sqrt{2} \\ 1\big/\sqrt{2} & 3\big/\sqrt{2} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{5}}\begin{pmatrix} -1 & -2 \\ 2 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ &= \frac{3\,|0\rangle - |1\rangle}{\sqrt{2\times 5}} \\ &= |y_3\rangle \end{aligned}$$

Example 2 (Less-Complete): Suppose

$$\begin{aligned} |x_1\rangle &= \frac{a\,|0\rangle + b\,|1\rangle}{\sqrt{a^2+b^2}}, \\ |y_1\rangle &= \frac{(a+b)\,|0\rangle + (a-b)\,|1\rangle}{\sqrt{2(a^2+b^2)}}(a=-11, b=7). \end{aligned}$$

Perceptron training: The same way through Algorithm 2 for training and finally get the quantum perceptron,

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\;\Sigma_{new}\;\hat{w}_{new}\left|x_j\right\rangle \\ &= \frac{1}{\sqrt{170}}\begin{pmatrix} -2\sqrt{2} & -9\sqrt{2} \\ -9\sqrt{2} & 2\sqrt{2} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{170}}\begin{pmatrix} -11 & 7 \\ -7 & -11 \end{pmatrix}\left|x_j\right\rangle. \end{aligned}$$

Validation of correctness:

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\;\Sigma_{new}\;\hat{w}_{new}\left|x_1\right\rangle \\ &= \frac{1}{\sqrt{170}}\begin{pmatrix} -2\sqrt{2} & -9\sqrt{2} \\ -9\sqrt{2} & 2\sqrt{2} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{170}}\begin{pmatrix} -11 & 7 \\ -7 & -11 \end{pmatrix}\frac{1}{\sqrt{170}}\begin{pmatrix} -11 \\ 7 \end{pmatrix} \\ &= \frac{-4\,|0\rangle - 18\,|1\rangle}{\sqrt{2\times 170}} \\ &= |y_1\rangle\,. \end{aligned}$$

2) *PHASE* GATE

Example 3 (Over-Complete): Suppose

$$\begin{aligned} &|x_1\rangle = |0\rangle\,, \quad |y_1\rangle = |0\rangle \\ &|x_2\rangle = |1\rangle\,, \quad |y_2\rangle = i\,|1\rangle \\ &|x_3\rangle = \frac{a\,|0\rangle + b\,|1\rangle}{\sqrt{a^2+b^2}}, \quad |y_3\rangle = \frac{a\,|0\rangle + bi\,|1\rangle}{\sqrt{a^2+b^2}}(a=2, b=2) \end{aligned}$$

Perceptron training: The quantum perceptron is

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\;\Sigma_{new}\;\hat{w}_{new}\left|x_j\right\rangle \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ -i & -i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}\left|x_j\right\rangle. \end{aligned}$$

Validation of correctness:

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\;\Sigma_{new}\;\hat{w}_{new}\left|x_1\right\rangle \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ -i & -i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= |0\rangle \\ &= |y_1\rangle \end{aligned}$$

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\;\Sigma_{new}\;\hat{w}_{new}\left|x_2\right\rangle \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ -i & -i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= i\,|1\rangle \\ &= |y_2\rangle \end{aligned}$$

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\;\Sigma_{new}\;\hat{w}_{new}\left|x_3\right\rangle \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ -i & -i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} \\ &= |y_3\rangle \end{aligned}$$

Example 4 (Less-Complete): Suppose

$$\begin{aligned} |x_1\rangle &= \frac{a\,|0\rangle + b\,|1\rangle}{\sqrt{a^2+b^2}}, \\ |y_1\rangle &= \frac{a\,|0\rangle + bi\,|1\rangle}{\sqrt{a^2+b^2}} (a=1, b=3). \end{aligned}$$

Perceptron training: The quantum perceptron is

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\ &= \frac{1}{\sqrt{10}}\begin{pmatrix} -1 & 3i \\ -3i & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{10}}\begin{pmatrix} -1 & -3 \\ -3 & 1 \end{pmatrix}\left|x_j\right\rangle \end{aligned}$$

Validation of correctness:

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_1\rangle \\ &= \frac{1}{\sqrt{10}}\begin{pmatrix} -1 & 3i \\ -3i & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{10}}\begin{pmatrix} -1 & -3 \\ -3 & 1 \end{pmatrix}\frac{1}{\sqrt{10}}\begin{pmatrix} 1 \\ 3 \end{pmatrix} \\ &= \frac{|0\rangle + 3i\,|1\rangle}{\sqrt{10}} \\ &= |y_1\rangle \end{aligned}$$

3) $\pi/8$ GATE

Example 5 (Over-Complete): Suppose

$$\begin{aligned} |x_1\rangle &= |0\rangle\,, |y_1\rangle = |0\rangle \\ |x_2\rangle &= |1\rangle\,, |y_2\rangle = e^{i\pi/4}\,|1\rangle \\ |x_3\rangle &= \frac{a\,|0\rangle + b\,|1\rangle}{\sqrt{a^2+b^2}}, \\ |y_3\rangle &= \frac{a\,|0\rangle + be^{i\pi/4}\,|1\rangle}{\sqrt{a^2+b^2}} (a=-8, b=-9). \end{aligned}$$

Perceptron training: The quantum perceptron is

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\ &= \frac{1}{\sqrt{145}}\begin{pmatrix} -8 & 9 \\ -9e^{i\pi/4} & -8e^{i\pi/4} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{145}}\begin{pmatrix} -8 & -9 \\ 9 & -8 \end{pmatrix}\left|x_j\right\rangle \end{aligned}$$

Validation of correctness:

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_1\rangle = |0\rangle = |y_1\rangle \\ \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_2\rangle = e^{i\pi/4}\,|1\rangle = |y_2\rangle \\ \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_3\rangle = \frac{-8\,|0\rangle - 9e^{i\pi/4}\,|1\rangle}{\sqrt{145}} = |y_3\rangle \end{aligned}$$

Example 6 (Less-Complete): Suppose

$$\begin{aligned} |x_1\rangle &= \frac{a\,|0\rangle + b\,|1\rangle}{\sqrt{a^2+b^2}}, \\ |y_1\rangle &= \frac{a\,|0\rangle + be^{i\pi/4}\,|1\rangle}{\sqrt{a^2+b^2}} (a=13, b=-10). \end{aligned}$$

Perceptron training: The quantum perceptron is

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\ &= \frac{1}{\sqrt{269}}\begin{pmatrix} -13 & 10e^{i\pi/4} \\ 10e^{i\pi/4} & 13i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &\quad \times \frac{1}{\sqrt{269}}\begin{pmatrix} -13 & 10 \\ -10 & -13 \end{pmatrix}\left|x_j\right\rangle \end{aligned}$$

Validation of correctness:

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_1\rangle \\ &= \frac{13\,|0\rangle - 10e^{i\pi/4}\,|1\rangle}{\sqrt{269}} = |y_1\rangle \end{aligned}$$

4) *CNOT* GATE

Example 7 (Over-Complete): Suppose that the example is:

$$\begin{aligned} |x_1\rangle &= |00\rangle\,, \quad |y_1\rangle = |00\rangle \\ |x_2\rangle &= |01\rangle\,, \quad |y_2\rangle = |01\rangle \\ |x_3\rangle &= |10\rangle\,, \quad |y_3\rangle = |11\rangle \\ |x_4\rangle &= |11\rangle\,, \quad |y_4\rangle = |10\rangle \\ |x_5\rangle &= \frac{a\,|00\rangle + b\,|11\rangle}{\sqrt{a^2+b^2}}, \quad |y_5\rangle = \frac{a\,|00\rangle + b\,|10\rangle}{\sqrt{a^2+b^2}} (a=3, b=4) \end{aligned}$$

Perceptron training: The same way through Algorithm 2 for training and learning and finally get the quantum perceptron:

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\ &= \begin{pmatrix} -0.6 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ -0.8 & -0.6 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &\quad \times \begin{pmatrix} -0.6 & 0 & 0 & -0.8 \\ 0.8 & 0 & 0 & -0.6 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}\left|x_j\right\rangle \end{aligned}$$

Validation of correctness:

$$\begin{aligned} \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_1\rangle = |00\rangle = |y_1\rangle \\ \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_2\rangle = |01\rangle = |y_2\rangle \\ \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_3\rangle = |11\rangle = |y_3\rangle \\ \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_4\rangle = |10\rangle = |y_4\rangle \\ \left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_5\rangle = \frac{3\,|00\rangle + 4\,|10\rangle}{5} = |y_5\rangle \end{aligned}$$

Example 8 (Less-Complete): Suppose

$$\begin{aligned} |x_1\rangle &= |00\rangle\,, \quad |y_1\rangle = |00\rangle \\ |x_2\rangle &= \frac{a\,|01\rangle + b\,|11\rangle}{\sqrt{a^2+b^2}}, \\ |y_2\rangle &= \frac{a\,|01\rangle + b\,|10\rangle}{\sqrt{a^2+b^2}} (a=5, b=-6) \end{aligned}$$

Perceptron training: The quantum perceptron is

$$
\begin{aligned}
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle\sum_{i=1}^{n}X_i \\
&= \frac{1}{\sqrt{61}}\begin{pmatrix} 0 & \sqrt{61} & 0 & 0 \\ -5 & 0 & 0 & 6 \\ 6 & 0 & 0 & 5 \\ 0 & 0 & \sqrt{61} & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&\quad \times \frac{1}{\sqrt{61}}\begin{pmatrix} 0 & -5 & 6 & 0 \\ \sqrt{61} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{61} \\ 0 & 6 & 5 & 0 \end{pmatrix}\left|x_j\right\rangle
\end{aligned}
$$

Validation of correctness:

$$
\begin{aligned}
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_1\right\rangle = \left|00\right\rangle = \left|y_1\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_2\right\rangle = \frac{5\left|01\right\rangle - 6\left|10\right\rangle}{\sqrt{61}} = \left|y_2\right\rangle
\end{aligned}
$$

B. COMPLEX QUANTUM GATES

In addition to the basic quantum gates illustrated before, there are also some complex gates which are frequently used, such as the *Toffoli* and *Fredkin* gates. Here, we take these two gates as an example to validate the applicability of our proposed algorithm.

1) *TOFFOLI* GATE

The *Toffoli* gate has three input bits and three output bits. Two of the bits are control bits that are unaffected by the action of the *Toffoli* gate. The third bit is a target bit that is flipped if both control bits are set to 1, and otherwise is left alone. The Fig. 3 illustrates the circuit of *Toffoli* gate, and Eq. 12 gives the matrix form of *Toffoli* operator.

$$
Toffoli = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{12}
$$

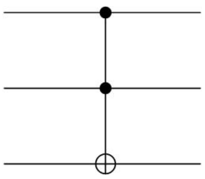


FIGURE 3. Quantum circuit of the *Toffoli* Gate.

Example 9 (Over-Complete): Suppose

$$
\begin{aligned}
&\left|x_1\right\rangle = \left|000\right\rangle, \quad \left|y_1\right\rangle = \left|000\right\rangle. \ \left|x_2\right\rangle = \left|001\right\rangle, \quad \left|y_2\right\rangle = \left|001\right\rangle \\
&\left|x_3\right\rangle = \left|010\right\rangle, \quad \left|y_3\right\rangle = \left|010\right\rangle. \ \left|x_4\right\rangle = \left|011\right\rangle, \quad \left|y_4\right\rangle = \left|011\right\rangle \\
&\left|x_5\right\rangle = \left|100\right\rangle, \quad \left|y_5\right\rangle = \left|100\right\rangle. \ \left|x_6\right\rangle = \left|101\right\rangle, \quad \left|y_6\right\rangle = \left|101\right\rangle \\
&\left|x_7\right\rangle = \left|110\right\rangle, \quad \left|y_7\right\rangle = \left|111\right\rangle. \ \left|x_8\right\rangle = \left|111\right\rangle, \quad \left|y_8\right\rangle = \left|110\right\rangle \\
&\left|x_9\right\rangle = \frac{a\left|001\right\rangle + b\left|110\right\rangle}{\sqrt{a^2+b^2}}, \\
&\left|y_9\right\rangle = \frac{a\left|001\right\rangle + b\left|111\right\rangle}{\sqrt{a^2+b^2}}(a = 3, b = 4)
\end{aligned}
$$

Perceptron training: The quantum perceptron is

$$
\begin{aligned}
&\left|y_{output}\right\rangle \\
&= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\
&= \frac{1}{5}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ -3 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -5 & 0 & 0 \\ -4 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
&\times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
&\times \frac{1}{5}\begin{pmatrix} 0 & -3 & 0 & 0 & 0 & 0 & -4 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & -5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5 \\ 0 & 0 & -5 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\left|x_j\right\rangle
\end{aligned}
$$

Validation of correctness:

$$
\begin{aligned}
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_1\right\rangle = \left|000\right\rangle = \left|y_1\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_2\right\rangle = \left|001\right\rangle = \left|y_2\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_3\right\rangle = \left|010\right\rangle = \left|y_3\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_4\right\rangle = \left|011\right\rangle = \left|y_4\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_5\right\rangle = \left|100\right\rangle = \left|y_5\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_6\right\rangle = \left|101\right\rangle = \left|y_6\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_7\right\rangle = \left|111\right\rangle = \left|y_7\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_8\right\rangle = \left|110\right\rangle = \left|y_8\right\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_9\right\rangle = \frac{3\left|001\right\rangle + 3\left|111\right\rangle}{\sqrt{3^2+4^2}} = \left|y_9\right\rangle
\end{aligned}
$$

Example 10 (Less-Complete): Suppose

$$
\begin{aligned}
|x_1\rangle &= |000\rangle, \quad |y_1\rangle = |000\rangle \\
|x_2\rangle &= |110\rangle, \quad |y_2\rangle = |111\rangle \\
|x_3\rangle &= \frac{a\,|101\rangle + b\,|111\rangle}{\sqrt{a^2+b^2}}, \\
|y_3\rangle &= \frac{a\,|101\rangle + b\,|110\rangle}{\sqrt{a^2+b^2}} (a = 1, b = 2)
\end{aligned}
$$

Perceptron training: The quantum perceptron is

$$
\begin{aligned}
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\
&= \frac{1}{\sqrt{5}}\begin{pmatrix}
0 & \sqrt{5} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{5} \\
0 & 0 & 0 & 0 & 0 & \sqrt{5} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \sqrt{5} & 0 \\
0 & 0 & 0 & 0 & \sqrt{5} & 0 & 0 & 0 \\
-1 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\
-2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \sqrt{5} & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \\
&\times \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \\
&\times \frac{1}{\sqrt{5}}\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & -1 & 0 & -2 \\
\sqrt{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \sqrt{5} & 0 \\
0 & 0 & 0 & 0 & 0 & -2 & 0 & 1 \\
0 & 0 & 0 & 0 & \sqrt{5} & 0 & 0 & 0 \\
0 & 0 & \sqrt{5} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \sqrt{5} & 0 & 0 & 0 & 0 \\
0 & \sqrt{5} & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}\left|x_j\right\rangle
\end{aligned}
$$

Validation of correctness:

$$
\begin{aligned}
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_1\rangle = |000\rangle = |y_1\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_2\rangle = |111\rangle = |y_2\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\,|x_3\rangle = \frac{|101\rangle + 2\,|110\rangle}{\sqrt{5}} = |y_3\rangle
\end{aligned}
$$

2) *FREDKIN* GATE

The *Fredkin* gate (also called *CSWAP* gate) is a computational circuit suitable for reversible computation. The *Fredkin* gate is a circuit or device with three inputs and three outputs, the first bit is control bit and the remaining two bits are target bits. The Fig. 4 shows that the *Fredkin* gate can transmit the first bit unchanged and swap the last two bits if the first bit is 1, and the Eq. 13 gives the matrix form of *Fredkin* operator.

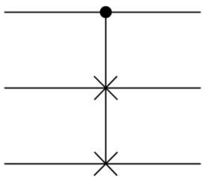


FIGURE 4. Quantum circuit of the *Fredkin* gate.

$$
Fredkin = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \tag{13}
$$

Example 11 (Over-Complete): Suppose

$$
\begin{aligned}
|x_1\rangle &= |000\rangle, \quad |y_1\rangle = |000\rangle.\ |x_2\rangle = |001\rangle, \quad |y_2\rangle = |001\rangle \\
|x_3\rangle &= |010\rangle, \quad |y_3\rangle = |010\rangle.\ |x_4\rangle = |011\rangle, \quad |y_4\rangle = |011\rangle \\
|x_5\rangle &= |100\rangle, \quad |y_5\rangle = |100\rangle.\ |x_6\rangle = |101\rangle, \quad |y_6\rangle = |110\rangle \\
|x_7\rangle &= |110\rangle, \quad |y_7\rangle = |101\rangle.\ |x_8\rangle = |111\rangle, \quad |y_8\rangle = |111\rangle \\
|x_9\rangle &= \frac{a\,|010\rangle + b\,|101\rangle}{\sqrt{a^2+b^2}}, \\
|y_9\rangle &= \frac{a\,|010\rangle + b\,|110\rangle}{\sqrt{a^2+b^2}} (a = 2, b = -7)
\end{aligned}
$$

Perceptron training: The quantum perceptron is

$$
\begin{aligned}
&\left|y_{output}\right\rangle \\
&\quad = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\
&\quad = \frac{1}{\sqrt{53}}\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{53} \\
0 & 0 & 0 & 0 & 0 & 0 & \sqrt{53} & 0 \\
-2 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \sqrt{53} & 0 & 0 & 0 & 0 \\
0 & 0 & \sqrt{53} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sqrt{53} & 0 & 0 & 0 \\
7 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \sqrt{53} & 0 & 0
\end{pmatrix} \\
&\quad \times \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{aligned}
$$

$$\times \frac{1}{\sqrt{53}}\begin{pmatrix} 0 & 0 & -2 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{53} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{53} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{53} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{53} \\ 0 & \sqrt{53} & 0 & 0 & 0 & 0 & 0 & 0 \\ \sqrt{53} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\left|x_j\right\rangle$$

Validation of correctness:

$$\begin{aligned}
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_1\right\rangle = |000\rangle = |y_1\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_2\right\rangle = |001\rangle = |y_2\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_3\right\rangle = |010\rangle = |y_3\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_4\right\rangle = |011\rangle = |y_4\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_5\right\rangle = |100\rangle = |y_5\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_6\right\rangle = |110\rangle = |y_6\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_7\right\rangle = |101\rangle = |y_7\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_8\right\rangle = |111\rangle = |y_8\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_9\right\rangle = \frac{2\,|010\rangle - 7\,|110\rangle}{\sqrt{2^2+7^2}} = |y_9\rangle
\end{aligned}$$

Example 12 (Less-Complete): Suppose

$$\begin{aligned}
|x_1\rangle &= |010\rangle\,, \quad |y_1\rangle = |010\rangle \\
|x_2\rangle &= |101\rangle\,, \quad |y_2\rangle = |110\rangle \\
|x_3\rangle &= \frac{a\,|000\rangle + b\,|110\rangle}{\sqrt{a^2+b^2}}, \\
|y_3\rangle &= \frac{a\,|000\rangle + b\,|101\rangle}{\sqrt{a^2+b^2}}(a=3, b=2)
\end{aligned}$$

Perceptron training: The quantum perceptron is

$$\begin{aligned}
&\left|y_{output}\right\rangle \\
&\quad = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\
&\quad = \frac{1}{\sqrt{13}}\begin{pmatrix} 0 & -3 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{13} & 0 & 0 \\ \sqrt{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{13} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{13} \\ 0 & -2 & 0 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{13} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{13} & 0 & 0 & 0 \end{pmatrix} \\
&\quad\times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

$$\times \frac{1}{\sqrt{13}}\begin{pmatrix} 0 & 0 & \sqrt{13} & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\sqrt{13} & 0 & 0 \\ 0 & \sqrt{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{13} \\ -2 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & \sqrt{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{13} & 0 & 0 & 0 \end{pmatrix}\left|x_j\right\rangle$$

Validation of correctness:

$$\begin{aligned}
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_1\right\rangle = |010\rangle = |y_1\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_2\right\rangle = |110\rangle = |y_2\rangle \\
\left|y_{output}\right\rangle &= \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_3\right\rangle = \frac{3\,|000\rangle + 2\,|101\rangle}{\sqrt{13}} = |y_3\rangle
\end{aligned}$$

V. PERFORMANCE EVALUATION

Now it is worth discussing the performance of the proposed algorithm. We select Altaisky's algorithm [30], Seow *et al.*'s algorithm [35] as references, and evaluate our algorithm in the aspects of the training set, output result, and iteration times.

It is well known that training set used for a perceptron algorithm are not necessarily ideal, so the cases of the non-ideal training set, i.e., the less-complete and over-complete training sets, should be taken into account in practical scenarios. In [30] and [35], they only consider the ideal case (i.e., complete training set), and it cannot be applied to those cases of less-complete and over-complete training sets. Table 2 detailedly compares the availabilities of Altaisky's, Seow *et al.*'s and our algorithms. As shown in Table 2, our algorithm has higher availability than other two algorithms, i.e., it is not only suitable for a complete training set, but also for the less-complete and over-complete training sets.

TABLE 2. Availability comparison among Altaisky's, Seow *et al.*'s and our algorithms under the complete, less-complete and over-complete training sets.

Quantum gates	Training set	Altaisky's [30]	Seow et al's [35]	Ours
H	complete set	√	√	√
	less-complete set	×	×	√
	over-complete set	×	×	√
S	complete set	√	√	√
	less-complete set	×	×	√
	over-complete set	×	×	√
T	complete set	√	√	√
	less-complete set	×	×	√
	over-complete set	×	×	√
CNOT	complete set	√	√	√
	less-complete set	×	×	√
	over-complete set	×	×	√
Toffoli	complete set	√	×	√
	less-complete set	×	×	√
	over-complete set	×	×	√
Fredkin	complete set	√	√	√
	less-complete set	×	×	√
	over-complete set	×	×	√

Assume that the training set is complete, we select six quantum gates $\{H, S, T, CNOT, Toffoli, Fredkin\}$ to compare

TABLE 3. Comparison of iteration times and output results among Altaisky's, Seow *et al.*'s and our algorithms.

Quantum gates	Comparative content	Altaisky's [30]	Seow et al's [35]	Ours
H	Iteration times	30	1	1
	Output	$\begin{pmatrix} 0.6771 \\ -0.7195 \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$
S	Iteration times	30	1	1
	Output	$\begin{pmatrix} 0 \\ 0.9576i \end{pmatrix}$	$\begin{pmatrix} 0 \\ i \end{pmatrix}$	$\begin{pmatrix} 0 \\ i \end{pmatrix}$
T	Iteration times	30	1	1
	Output	$\begin{pmatrix} 0 \\ 0.9576e^{\frac{i\pi}{4}} \end{pmatrix}$	$\begin{pmatrix} 0 \\ e^{\frac{i\pi}{4}} \end{pmatrix}$	$\begin{pmatrix} 0 \\ e^{\frac{i\pi}{4}} \end{pmatrix}$
CNOT	Iteration times	30	1	1
	Output	$\begin{pmatrix} 0 \\ 0 \\ 0.9576 \\ -0.0424 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
Toffoli	Iteration times	30	–	1
	Output	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.9576 \\ -0.0424 \end{pmatrix}$	–	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
Fredkin	Iteration times	30	1	1
	Output	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.9576 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

the accuracy (i.e., the difference between desirable results and output results) and applicability (i.e., applicable quantum gates) among three algorithms. Since Altaisky's algorithm converges to an approximate value after 30 iterations, so we take 30 as the iteration times, and the approximate value as the final output result. Different from Altaisky's algorithm, Seow *et al.*'s algorithm can get the accurate output results of five gates $\{H, S, T, CNOT, Fredkin\}$ after only one iteration. In our algorithm, we also implement the *Toffoli* gate within one iteration. As shown in Table 3, Seow *et al.*'s and our proposed algorithm have advantages over Altaisky's algorithm in terms of accuracy and iteration times, and our algorithm is more widely applicable than Seow's algorithm.

In summary, considering the aspects of availability, accuracy, and applicability, our algorithm has advantages over Altaisky's and Seow *et al.*'s algorithms.

VI. DISCUSSION AND CONCLUSION

In this paper, we present an efficient one-iteration perceptron algorithm based on unitary weights. Different from the previous quantum perceptron algorithms, our proposed algorithm has good availability, high accuracy, and wide applicability. To be specific, our perception algorithm is not only suitable for the less-complete and over-complete training set, but also can accurately implement universal quantum gates within one iteration. In order to further validate its applicability, Appendix A gives an example to show that our algorithm can be applied to an arbitrary quantum composite gate (which can be viewed as arbitrary quantum computation).

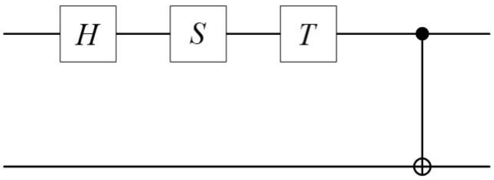


FIGURE 5. Quantum circuit of the composite gate.

The proposed algorithm discussed in our paper is a learning algorithm for the single-layer perceptron. For multi-layer perceptrons, where a hidden layer exists, more sophisticated algorithms such as backpropagation must be used. So, one of our next research efforts is how to use quantum technology to solve some problem of multi-layer perceptron. On the other hand, we know that when multiple perceptrons are combined in an artificial neural network, each output neuron runs independently of all other neurons; therefore, each output can be considered in isolation. Therefore, another work we would like to carry out is to select some neural network models and validate whether our quantum perceptron can be effectively used to construct a large-scale quantum neural network.

ACKNOWLEDGMENT

The authors would like to express heartfelt gratitude to the anonymous reviewers and editor for their comments that improved the quality of this paper. And the support of all the members of the quantum research group of NUIST is especially acknowledged, their professional discussions and advice have helped them a lot.

APPENDIX A
TRAINING ARBITRARY QUANTUM COMPOSITE GATE WITH OUR ALGORITHM

In order to validate that the proposed quantum perceptron algorithm is suitable for the multiple gates composite calculation, the *Hadamard* gate (H), *phase* gate (S), $\pi/8$ gate (T), *CNOT* gate are combined to construct the composite gate (shown in Fig. 5). Through the training of composite gate to further validate the applicability of our algorithm.

Example 13 (Over-Complete): Suppose

$$|x_1\rangle = |00\rangle, \quad |y_1\rangle = \frac{|00\rangle + ie^{i\pi/4}|11\rangle}{\sqrt{2}}$$

$$|x_2\rangle = |01\rangle, \quad |y_2\rangle = \frac{|01\rangle + ie^{i\pi/4}|10\rangle}{\sqrt{2}}$$

$$|x_3\rangle = |10\rangle, \quad |y_3\rangle = \frac{|00\rangle - ie^{i\pi/4}|11\rangle}{\sqrt{2}}$$

$$|x_4\rangle = |11\rangle, \quad |y_4\rangle = \frac{|01\rangle - ie^{i\pi/4}|10\rangle}{\sqrt{2}}$$

$$|x_5\rangle = \frac{a|00\rangle + b|10\rangle}{\sqrt{a^2+b^2}},$$

$$|y_5\rangle = \frac{(a+b)|00\rangle + (a-b)ie^{i\pi/4}|11\rangle}{\sqrt{2(a^2+b^2)}}(a = 6, b = 8).$$

Perceptron training: The quantum perceptron is

$$\begin{aligned}
&\left|y_{output}\right\rangle \\
&\quad = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\
&\quad = \frac{1}{10\sqrt{2}}\begin{pmatrix} -14 & 2 & 0 & 0 \\ 0 & 0 & 10 & 10 \\ 0 & 0 & -10ie^{i\pi/4} & 10ie^{i\pi/4} \\ 2ie^{i\pi/4} & 14ie^{i\pi/4} & 0 & 0 \end{pmatrix} \\
&\qquad \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} -0.6 & 0 & -0.8 & 0 \\ 0.8 & 0 & -0.6 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}\left|x_j\right\rangle
\end{aligned}$$

Validation of correctness:

$$\left|y_{output}\right\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_1\right\rangle = \frac{|00\rangle + ie^{i\pi/4}|11\rangle}{\sqrt{2}} = |y_1\rangle$$

$$\left|y_{output}\right\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_2\right\rangle = \frac{|01\rangle + ie^{i\pi/4}|10\rangle}{\sqrt{2}} = |y_2\rangle$$

$$\left|y_{output}\right\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_3\right\rangle = \frac{|00\rangle - ie^{i\pi/4}|11\rangle}{\sqrt{2}} = |y_3\rangle$$

$$\left|y_{output}\right\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_4\right\rangle = \frac{|01\rangle - ie^{i\pi/4}|10\rangle}{\sqrt{2}} = |y_4\rangle$$

$$\left|y_{output}\right\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_5\right\rangle = \frac{14\,|00\rangle - 2ie^{i\pi/4}|11\rangle}{\sqrt{2\times 100}} = |y_5\rangle$$

Example 14 (Less-Complete): Suppose

$$\begin{aligned}
&|x_1\rangle = |00\rangle, \quad |y_1\rangle = \frac{|00\rangle + ie^{i\pi/4}|11\rangle}{\sqrt{2}} \\
&|x_2\rangle = \frac{a\,|01\rangle + b\,|11\rangle}{\sqrt{a^2+b^2}}, \\
&|y_2\rangle = \frac{(a+b)\,|01\rangle + (a-b)ie^{i\pi/4}\,|10\rangle}{\sqrt{2(a^2+b^2)}}(a=2, b=1)
\end{aligned}$$

Perceptron training: The quantum perceptron is

$$\begin{aligned}
&\left|y_{output}\right\rangle \\
&\quad = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_j\right\rangle \\
&\quad = \frac{1}{\sqrt{10}}\begin{pmatrix} -\sqrt{5} & 0 & 0 & \sqrt{5}e^{i\pi/4} \\ 0 & -3 & e^{i\pi/4} & 0 \\ 0 & -ie^{i\pi/4} & 3 & 0 \\ -\sqrt{5}ie^{i\pi/4} & 0 & 0 & \sqrt{5} \end{pmatrix} \\
&\qquad \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\sqrt{5}\begin{pmatrix} -\sqrt{5} & 0 & 0 & 0 \\ 0 & -2 & 0 & -1 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & \sqrt{5} & 0 \end{pmatrix}\left|x_j\right\rangle
\end{aligned}$$

Validation of correctness:

$$\left|y_{output}\right\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_1\right\rangle = \frac{|00\rangle + ie^{i\pi/4}|11\rangle}{\sqrt{2}} = |y_1\rangle$$

$$\left|y_{output}\right\rangle = \hat{F}\,\Sigma_{new}\,\hat{w}_{new}\left|x_2\right\rangle = \frac{3\,|01\rangle + ie^{i\pi/4}|10\rangle}{\sqrt{2\times 5}} = |y_2\rangle$$

REFERENCES

[1] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," *J. Stat. Phys.*, vol. 22, no. 5, pp. 563–591, May 1980.

[2] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, nos. 6–7, pp. 467–488, 1982.

[3] D. Deutsch, "Quantum theory, the church–turing principle and the universal quantum computer," *Proc. Royal Soc. A*, vol. 400, no. 1818, pp. 97–117, 1985.

[4] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

[5] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, 1997.

[6] W. Huang, Q. Su, B. Liu, Y.-H. He, F. Fan, and B.-J. Xu, "Efficient multiparty quantum key agreement with collective detection," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 15264.

[7] W.-J. Liu, Y. Xu, C.-N. Yang, P.-P. Gao, and W.-B. Yu, "An efficient and secure arbitrary N-party quantum key agreement protocol using bell states," *Int. J. Theor. Phys.*, vol. 57, no. 1, pp. 195–207, 2018.

[8] W.-J. Liu, H.-W. Chen, T.-H. Ma, Z.-Q. Li, Z.-H. Liu, and W.-B. Hu, "An efficient deterministic secure quantum communication scheme based on cluster states and identity authentication," *Chin. Phys. B*, vol. 18, no. 10, p. 4105, 2009.

[9] J. Zhong, Z. Liu, and J. Xu, "Analysis and improvement of an efficient controlled quantum secure direct communication and authentication protocol," *Comput., Mater. Continua*, vol. 57, no. 3, pp. 621–633, 2018.

[10] X. Tan, X. Li, and P. Yang, "Perfect quantum teleportation via bell states," *Comput., Mater. Continua*, vol. 57, no. 3, pp. 495–503, 2018.

[11] W.-J. Liu, Z.-F. Chen, C. Liu, and Y. Zheng, "Improved deterministic N-to-one joint remote preparation of an arbitrary Qubit via EPR Pairs," *Int. J. Theor. Phys.*, vol. 54, no. 2, pp. 472–483, 2015.

[12] Z. Qu, S. Wu, M. Wang, L. Sun, and X. Wang, "Effect of quantum noise on deterministic remote state preparation of an arbitrary two-particle state via various quantum entangled channels," *Quantum Inf. Process.*, vol. 16, no. 12, p. 306, 2017.

[13] F. Wu, X. Zhang, W. Yao, Z. Zheng, L. Xiang, and W. Li, "An advanced quantum-resistant signature scheme for cloud based on eisenstein ring," *Comput., Mater. Continua*, vol. 56, no. 1, pp. 19–34, 2018.

[14] Z. Qu, T. Zhu, J. Wang, and X. Wang, "A novel quantum stegonagraphy based on brown states," *Comput., Mater. Continua*, vol. 56, no. 1, pp. 47–59, 2018.

[15] Z. Qu, Z. Cheng, W. Liu, and X. Wang, 'A novel quantum image steganography algorithm based on exploiting modification direction," in *Multimedia Tools and Applications*, 2018. [Online]. Available: https://www.deepdyve.com/search?query=10.1007%2Fs11042-018-6476-5. doi: 10.1007/s11042-018-6476-5.

[16] W.-J. Liu, Z.-Y. Chen, S. Ji, H.-B. Wang, and J. Zhang, "Multi-party semi-quantum key agreement with delegating quantum computation," *Int. J. Theor. Phys.*, vol. 56, no. 10, pp. 3164–3174, 2017.

[17] W. Liu, Z. Chen, J. Liu, Z. Su, and L. Chi, "Full-blind delegating private quantum computation," *Comput., Mater. Continua*, vol. 56, no. 2, pp. 211–223, 2018.

[18] S. Lloyd, M. Mohseni, and P. Rebentrost. (2013). "Quantum algorithms for supervised and unsupervised machine learning." [Online]. Available: https://arxiv.org/abs/1307.0411

[19] W.-J. Liu, P.-P. Gao, W.-B. Yu, Z.-G. Qu, and C.-N. Yang, "Quantum Relief algorithm," *Quantum Inf. Process.*, vol. 17, no. 10, p. 280, 2018.

[20] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.

[21] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[22] S. Kak, "On quantum neural computing," *Inf. Sci.*, vol. 83, nos. 3–4, pp. 143–160, Mar. 1995.

[23] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural network methods in quantum mechanics," *Comput. Phys. Commun.*, vol. 104, nos. 1–3, pp. 1–14, 1997.

[24] G. Purushothaman and N. B. Karayiannis, "Quantum neural networks (QNNs): Inherently fuzzy feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 679–693, May 1997.

[25] M. Andrecut and M. K. Ali, "A quantum neural network model," *Int. J. Mod. Phys. C*, vol. 13, no. 1, pp. 75–88, 2002.

[26] S. Gupta and R. K. P. Zia, "Quantum neural networks," *J. Comput. Syst. Sci.*, vol. 63, no. 3, pp. 355–383, 2001.

[27] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, Cambridge, 2000.

[28] M. Schuld, I. Sinayskiy, and F. Petruccione, "The quest for a quantum neural network," *Quantum Inf. Process.*, vol. 13, no. 11, pp. 2567–2586, Nov. 2014.

[29] M. Panella and G. Martinelli, "Neural networks with quantum architecture and quantum learning," *Int. J. Circuit Theory Appl.*, vol. 39, no. 1, pp. 61–77, 2011.

[30] M. V. Altaisky. (2001). "Quantum neural network." [Online]. Available: https://arxiv.org/abs/quant-ph/0107012

[31] E. C. Behrman, J. E. Steck, and S. R. Skinner, "A spatial quantum neural computer," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 1999, pp. 874–877.

[32] R. Zhou and Q. Ding, "Quantum M-P neural network," *Int. J. Theor. Phys.*, vol. 46, no. 12, pp. 3209–3215, 2007.

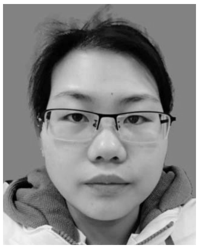
[33] A. Sagheer and M. Zidan. (2013). "Autonomous quantum perceptron neural network." [Online]. Available: https://arxiv.org/abs/1312.4149

[34] M. Siomau, "A quantum model for autonomous learning automata," *Quantum Inf. Process.*, vol. 13, no. 5, pp. 1211–1221, 2014.

[35] K.-L. Seow, E. Behrman, and J. Steck. (2015). "Efficient learning algorithm for quantum perceptron unitary weights." [Online]. Available: https://arxiv.org/abs/1512.00522

[36] S. 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 Netw.*, vol. 76, pp. 55–64, Apr. 2016.

WENJIE LIU was born in 1979. He received the bachelor's degree from HZAU, China, in 1997, the master's degree from WHU, China, in 2004, and the Ph.D. degree from SEU, in 2011. He is currently an Associate Professor of computer science with the Nanjing University of Information Science and Technology, China. He is a Visiting Professor with the Centre for Quantum Computing and Intelligent Systems, University of Technology Sydney. In recent five years, he has published over 20 SCI-indexed articles. His research interests include quantum machine learning, quantum secure multi-party computation, and quantum cryptography communication. He was a recipient of the China NSF Grant, the Jiangsu NSF Grant, and Six Talent Peaks Project in Jiangsu Province.



PEIPEI GAO was born in 1994. She received the bachelor's degree from the Nanjing University of Information Science and Technology, China, in 2016, where she is currently pursuing the master's degree with the School of Computer and Software. Her research interest includes quantum machine learning.

YUXIANG WANG received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, China, in 2010. In 2010, he joined the Nanjing University of Information and Technology, China, where he is currently a Lecturer with the School of Computer and Software. His research interests include intelligent computing, wireless sensor networks, data mining, and pervasive computing.

WENBIN YU received the Ph.D. degree in the field of quantum information processing. He is currently a Lecturer with the School of Computer Science and Software, Nanjing University of Information Science and Technology, China. His research interest includes quantum information processing. He is a Leading Researcher of the National Science Foundation of China, Youth Program.

MAOJUN ZHANG was born in 1979. He received the bachelor's degree from TYNU, China, in 2001, the master's degree from GXU, China, in 2004, and the Ph.D. degree from DLUT, China, in 2007. He is currently a Professor of mathematics and computing science with the Guilin University of Electronic Technology, China. He has published over ten SCI-indexed articles. His research interests include optimization, statistics, and machine learning. He was a recipient of two China NSF grants and a Guangxi NSF Grant.