Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Recent Advances in Quantum Machine Learning

Yao Zhang | Qiang Ni

[1]School of Computing and Communications, Lancaster University, United Kingdom

Summary

Machine learning is a branch of artificial intelligence, and it has been widely used in many science and engineering areas, such as data mining, natural language processing, computer vision, biological analysis and so on. Quantum computer is considered as one of the most promising technologies of human beings in the near future. With the development of machine learning and quantum computing, researchers consider to combine these two aspects to gain more benefits. As a result, a novel interdisciplinary subject has emerged—quantum machine learning. This paper reviews the state-of-the-art research of algorithms of quantum machine learning and shows a path of the research from the basic quantum information to quantum machine learning algorithms from the perspective of people in the field of computer science.

KEYWORDS:
Quantum machine learning, quantum machine learning algorithms, quantum computing, machine learning

1 | INTRODUCTION

Machine learning has been developed for more than half a century, and with the improvement of computational ability, it has become a very important part of computer science. Ignoring the definition of machine learning, the learning is usually divided into three types: supervised learning, unsupervised learning and reinforcement learning. Since these three types of learning have already been clearly defined in the learning theory, here the concepts will not be emphasised in this review, but we only focus on the corresponding machine learning algorithms. Although the computing power has increased quite fast over a couple of decades and new algorithms have come up continuously, the increment of data is much greater than the growth of the computers' performance. Therefore, the lack of computing power becomes deficiency gradually in the field of machine learning, which relies on big data in many cases.

Quantum computing is based on phenomena of quantum mechanics, such as superposition and entanglement. Because of the paramount feature for high speed computing, the parallelism can be designed into specific algorithms to solve specific problems. These classical problems usually cannot be solved as efficient as they are in the quantum system. Shor's algorithm[1] shows that the quantum computing is able to provide an exponential speedup to solve the problem of big integer factorisation, which is impossible by using any classical method. After that, a plenty of quantum algorithms are proposed to solve the specific problems. For instance, Grover's algorithm is proved that it can give a quadratic speedup in searching an unstructured database[2].

Since machine learning is under pressure from lack of computing power and quantum computing has this strong computational ability, people consider the possibilities of the combination of quantum computing and machine learning. The development of quantum computer has made some progress recently. For example, D-wave special-purpose quantum computer, which can perform quantum annealing, allows some classical machine learning algorithms to run efficiently[3]. On the other hand, some companies and research institutions have produced actual prototype machines of universal quantum computers based on the quantum circuit model, which make experiments are able to be carried out with quantum computational operations on a small

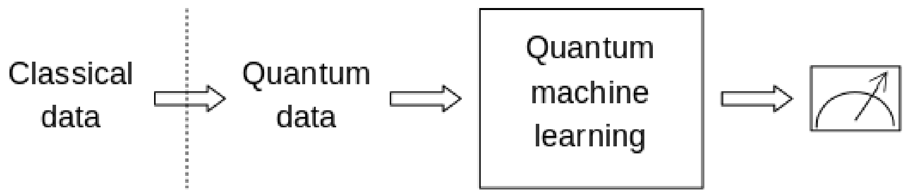

FIGURE 1 The basic structure of quantum machine learning. The learning process deal with the quantum data.

number of qubits via cloud platforms. Nowadays, the general large-scale quantum computer is still being developed. However, the research of potential quantum machine learning algorithms has got some progress. Several famous machine learning algorithms have got their quantum counterparts, such as quantum support vector machine (QSVM), quantum k-means clustering etc. In addition, some algorithms have been implemented and tested on the forementioned real quantum computer[4,5,6,7].

Obviously, there are two essential parts in machine learning—the data and the learning process. Likewise, in quantum domain, it also includes these two parts. Looking back to history, the quantum computer was proposed and supposed to simulate the quantum system in physics in the earliest days to solve the problem that some "data and actions" in quantum system are very hard to be simulated by classical computers. Therefore, in early days, only quantum data was considered. However, people consider to make it more useful in many other areas since it has unprecedented computing power. Because the quantum computing is supposed to deal with the quantum data, the classical data should be pre-processed into quantum data, so that quantum computing can work as it is imagined. In daily life, it is believed that most information people deal with is classical, so it is necessary to do this pre-processing. However, there is a special case that the quantum data may be processed directly. For example, quantum communication has become a hot topic in recent years, and in quantum channels, there may be some noise itself is quantum[8]. Fig. 1 illustrates the structure of quantum machine learning.

There have been some survey papers which mainly overview general ideas of different machine learning algorithms in quantum version[9,10,11,12,13]. In our review work, we would like to provide a different perspective on this interdisciplinary area, which introduces the quantum computing in machine learning from fundamentals to applications. The rest of this paper is organised as follows. In section 2, we review some basic knowledge of quantum information, which is the underpinning of quantum computing and even quantum machine learning. In section 3, we introduce some basic algorithms for quantum machine learning, namely the subroutines of whole machine learning algorithms in quantum version, which are the core parts of quantum machine learning algorithms and mainly provide the speedup over classical machine learning algorithms. In section 4, we introduce some popular machine learning algorithms and their quantum conterparts as the application of quantum machine learning, including both supervised and unsupervised cases. Finally, the conclusion is given in section 5 which introduces the practical problems and challenges in the area of quantum machine learning.

2 | BACKGROUND

2.1 | Qubit and quantum state

Like the bit in the classical information, qubit is the fundamental unit of quantum information, which is usually denoted by Dirac notation:

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

and

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

Unlike classical bits, qubits can exist in a superposition state:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \tag{3}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. It means that the state $|\Psi\rangle$ is in both $|0\rangle$ and $|1\rangle$ simultaneously, but when it is measured, it will collapse to the state $|0\rangle$ with probability $|\alpha|^2$, or to the state $|1\rangle$ with probability $|\beta|^2$. Similarly, a two-qubit system can be expressed as:

$$|\Psi\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}, \tag{4}$$

where $a, b, c, d \in \mathbb{C}$ and $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$.

Suppose that we have a two-qubit system which is expressed as:

$$\begin{aligned} |\psi_{12}\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \right] \\ &= \frac{1}{\sqrt{2}} (|0\rangle_1 \, |1\rangle_2 + |1\rangle_1 \, |0\rangle_2) \\ &= \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle), \end{aligned} \tag{5}$$

where the subscripts $1, 2$ are the order number of these two qubits. It turns out that the composite system $|\psi_{12}\rangle$ cannot be expressed as a tensor product of two independent qubits. In this case, it is called Entanglement. A much more detailed instruction of quantum information can be found in Nielsen and Chuang's book[14].

Another representation of quantum states is called density matrix. The density matrix can be used to describe part of a composite systems as well. It is given by the outer product of the state with itself:

$$\rho = |\psi\rangle \langle\psi| = \begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} a^* & b^* \end{pmatrix} = \begin{pmatrix} |a|^2 & ab^* \\ a^*b & |b|^2 \end{pmatrix}. \tag{6}$$

The quantum state $|\psi\rangle$ is a pure state. In addition, the density matrix can also describe a set of pure states $|\psi_i\rangle$, which is a mixed states, with probabilities p_i [14,15]:

$$\rho = \sum_i p_i \, |\psi_i\rangle \langle\psi_i| , \tag{7}$$

where $\sum_i p_i = 1$.

2.2 | Quantum gates

In the quantum circuit model of computation, the quantum gate is the basic quantum circuit operating on qubits, which is the building blocks of quantum circuits like the classical logic gate is for conventional digital circuits. Unlike classical logic gates, quantum gates are all reversible[14,16]. Therefore, quantum gates are represented by unitary matrices. That means the quantum gates in the circuits always have the same number of inputs and outputs. Quantum gates are actually operators—unitary matrices that act on a quantum state and transform the quantum state into another quantum state, which has the form

$$\begin{aligned} U\,|\psi\rangle &= \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} a \\ b \end{pmatrix} \\ &= |\phi\rangle . \end{aligned} \tag{8}$$

Table 1 lists some basic quantum gates.

Elementary quantum gates	Pauli-X	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \equiv \sigma_x$	It maps $\|0\rangle$ to $\|1\rangle$ and $\|1\rangle$ to $\|0\rangle$. It is equialent to NOT gate.
	Pauli-Y	$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \equiv \sigma_y$	It maps $\|0\rangle$ to $i\,\|1\rangle$ and $\|1\rangle$ to $-i\,\|0\rangle$.
	Pauli-Z	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \equiv \sigma_z$	It leaves the basis state $\|0\rangle$ unchanged and maps $\|1\rangle$ to $-\,\|1\rangle$. It is also called phase-flip.
	Hadamard	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	It creates a superposition by mapping $\|0\rangle$ to $\frac{\|0\rangle+\|1\rangle}{\sqrt{2}}$ and $\|1\rangle$ to $\frac{\|0\rangle-\|1\rangle}{\sqrt{2}}$.
	Phase shift	$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$	It leaves the basis state $\|0\rangle$ unchanged and maps $\|1\rangle$ to $e^{i\phi}\,\|1\rangle$.
Two-qubit gates	Controlled NOT	$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	It acts on 2 qubits, and performs the NOT operation on the target qubit only when the control qubit is $\|1\rangle$.
	SWAP	$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	It acts on 2 qubits, and swaps these two qubits.

TABLE 1 Basic quantum gates

2.3 | Measurement

In quantum mechanics, the definition of measurement is the core problem of the interpretation of quantum mechanics. However, the interpretation of quantum mechanics is currently no consensus. Here we do not need to worry about philosophical differences, but just consider the practical physics measurement.

According to quantum postulate 3[14], quantum measurements are described by a collection M_m of measurement operators. These operators are acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$, then the probability that the result m occurs is given by:

$$p(m) = \langle\psi|\, M_m^\dagger M_m \,|\psi\rangle\,, \tag{9}$$

and the state of the system after the measurement is:

$$\frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}. \tag{10}$$

The measurement operators satisfy the completeness equation

$$\sum_m M_m^\dagger M_m = I. \tag{11}$$

The completeness equation expresses the fact that probabilities sum to 1:

$$1 = \sum_m p_m = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle . \tag{12}$$

For instance, the measurement of a qubit in the computational basis is defined as operators $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. Thus the operator M is Hermitian and obeys the completeness: $I = M_0^\dagger M_0 + M_1^\dagger M_1$. Suppose that the state being measured is $|\psi\rangle = a|0\rangle + b|1\rangle$. Then the probabilities of obtaining measurement outcome 0 and 1 are seperately

$$p(0) = \langle\psi| M_0^\dagger M_0 |\psi\rangle = |a|^2 \tag{13}$$

and

$$p(1) = \langle\psi| M_1^\dagger M_1 |\psi\rangle = |b|^2. \tag{14}$$

In addition, the states after measurement that corresponding to those two cases are

$$\frac{M_0 |\psi\rangle}{|a|} = \frac{a}{|a|} |0\rangle \tag{15}$$

and

$$\frac{M_1 |\psi\rangle}{|b|} = \frac{b}{|b|} |1\rangle , \tag{16}$$

where the multipliers $\frac{a}{|a|}$ and $\frac{b}{|b|}$ that related to phase factor can be securely ignored.

3 | BASIC ALGORITHMS FOR QUANTUM MACHINE LEARNING

3.1 | Grover's algorithm

Grover's algorithm is a famous quantum algorithm, since it uses the quantum parallelism to give the speedup which has no corresponding counterparts in the classical computation. Grover's algorithm solves the problem of unstructured or unsorted database search. Specifically, the problem can be described as, given a set of N elements $X = \{x_1 ... x_n\}$, and also given a boolean function $f : X \to \{0, 1\}$, the goal is to find an element x^* in X to satisfy that $f(x^*) = 1$. Classically, solving linear search problem has the complexity of $O(N)$ in time. Grover's algorithm is proved that it takes up $O(\sqrt{N})$ in time and is the fastest possible quantum algorithm[17]. It provides quadratic speedup, while some other quantum algorithms can give exponential speedup over their classical counterparts. However, even quadratic speedup is considerable when N is large.

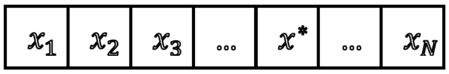

FIGURE 2 The sketch of an unstructured database. x^* is the item that needs to be searched.

As it is mentioned above, the boolean function $f : X \to \{0, 1\}$ can tell whether the result it searched is right if $f(x^*) = 1$, otherwise wrong if $f(x) = 0$. The oracle is actually a unitary operator that is used for the function which checks if the searched result is right. The oracle operator is usually represented as O, and meets the relation

$$O|x\rangle = (-1)^{f(x)} |x\rangle . \tag{17}$$

From this equation, it is clear to see that the phase of the quantum state $|x^*\rangle$ would be flipped since only x^* makes the function $f(x^*)$ outputs 1, where x^* is which item it is searching for. In another word, the problem of database searching is given the input

x, finding the entry that corresponds to x. Thus, in reality, the quantum oracle is already given to the algorithm so that it can be used to determine or test if the processing returns true.

For a list of items in the database, there is no idea where the target one is before looking at the list. Thus, any guess of its location is the same, which can be expressed as a superposition of a quantum state. That is

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle . \tag{18}$$

Where $|x\rangle$ is the standard basis. If it is measured at this point, this superposition will collapse to any one of the basis states with the same probability of $\frac{1}{N}$, where N is 2^n. So if we want to pick the target state, enhancing the probability amplitude of the target state is a spontaneous thought. This is called amplitude amplification. The amplitude amplification can cause the collapse to skew to the state that has the amplified probability amplitude. Therefore, what makes Grover's algorithm work is to find ways to increase the probability amplitude of the particular state. Fig. 3 shows the flow of the algorithm.

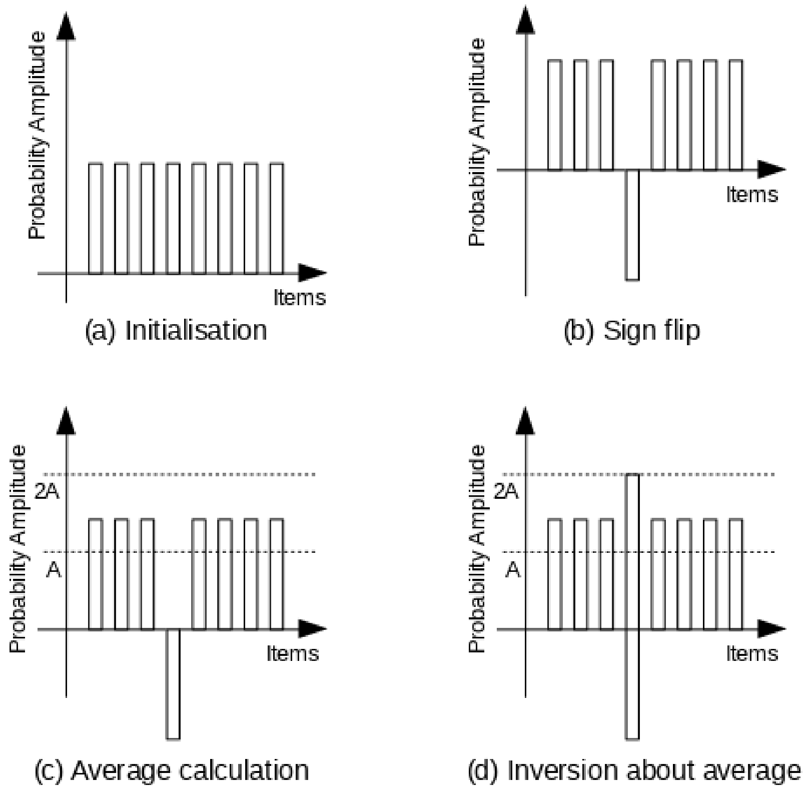

FIGURE 3 The flow of Grover's algorithm.

In Fig. 3 (a), it initialises n qubits at the state $|\psi_0\rangle = |0 \cdots 0\rangle$ and then create a superposition with all qubits that are with the same probability amplitude:

$$|\psi_1\rangle = H^{\otimes n} |0 \cdots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{N-1} |k\rangle . \tag{19}$$

Then the quantum oracle O is applied to flip the sign of searched input:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} O \sum_{k=0}^{N-1} |k\rangle . \tag{20}$$

This is an iterative operation that the result gets closer to the target in every process, which means the probability amplitude of the searched item increases in each iteration.

Then, for implementing the inversion about the average, a G gate (representing Grover iteration, more details in[14]), which is $H^{\otimes n}(2\left|0^{\otimes n}\right\rangle\left\langle 0^{\otimes n}\right| - I^{\otimes n})H^{\otimes n}$, is applied on the state $\left|\psi_2\right\rangle$. It gets

$$\left|\psi_3\right\rangle = H^{\otimes n}\left(2\left|0^{\otimes n}\right\rangle\left\langle 0^{\otimes n}\right| - I^{\otimes n}\right)H^{\otimes n}\left|\psi_2\right\rangle. \tag{21}$$

Finally, after repeating steps corresponding to the Eq. (20) and Eq. (21), the measurement of the state $\left|\psi_3\right\rangle$ will give the target searched item with a high probability. The times of Grover iteration is proved as $iteration_{max} = \frac{\pi}{4}\sqrt{2^n}$ times. A thorough derivation of the iteration times is given in[18]. Thus, the full circuit of Grover's algorithm can be expressed as the Fig. 4.

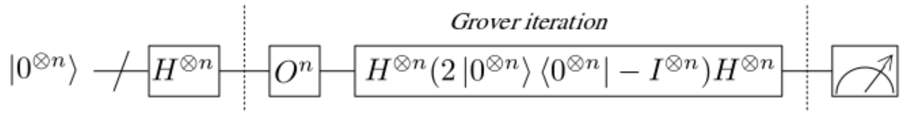
Grover iteration
$|0^{\otimes n}\rangle$
$H^{\otimes n}$
O^n
$H^{\otimes n}(2|0^{\otimes n}\rangle\langle 0^{\otimes n}| - I^{\otimes n})H^{\otimes n}$

FIGURE 4 The structure of Grover's algorithm.

Grover's algorithm was applied on the optimisation problem, which is of finding the minimum[19]. It applies Grover's algorithm with quantum oracles that tells which items are smaller than the threshold and performs it several times to find out the solution. In addition, some researchers made significant improvements in the quantum search algorithm based on Grover's algorithm[20,21].

3.2 | Swap-test

Swap-test is a simple and basic algorithm that is able to evaluate the overlap of two states. The overlap is a measure of similarity between two quantum states, and it is noted as $\langle\psi|\phi\rangle$. Fig. 5 shows the quantum circuit of this algorithm.

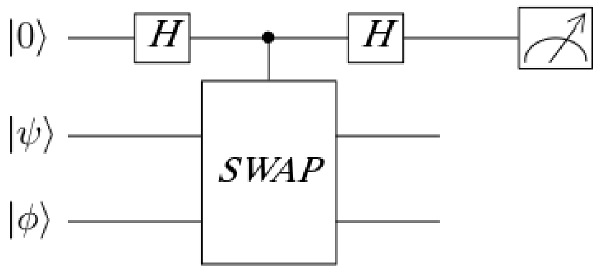

FIGURE 5 The circuit of swap-test algorithm.

For describing the procedure of swap-test algorithm, suppose there are two states $|\psi\rangle$ and $|\phi\rangle$, as well as a control qubit $|0\rangle$, which constitute a quantum state

$$\left|s_0\right\rangle = |0\rangle_1 \otimes |\psi\rangle_2 \otimes |\phi\rangle_3 = |0,\psi,\phi\rangle. \tag{22}$$

The subscripts represent what number of the state it is.

First of all, a Hadamard gate is applied on the control qubit that initialised as $|0\rangle$. It results in a superposition:

$$\begin{aligned}\left|s_1\right\rangle &= (H_1 \otimes I_2 \otimes I_3)\left|\psi_0\right\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle_1|\psi\rangle_2|\phi\rangle_3 + |1\rangle_1|\psi\rangle_2|\phi\rangle_3) \\ &= \frac{1}{\sqrt{2}}(|0,\psi,\phi\rangle + |1,\psi,\phi\rangle).\end{aligned} \tag{23}$$

Then, the *SWAP* gate is applied on both $|\psi\rangle$ and $|\phi\rangle$ to swap $|\psi\rangle$ and $|\phi\rangle$ with the control qubit in state $|1\rangle$. For how to construct the *SWAP* gate is another question, we would not talk about it here. This *controlled-SWAP* gate is also called *Fredkin* gate. As the result, following state will be achieved:

$$\begin{aligned}|s_2\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_1|\psi\rangle_2|\phi\rangle_3+|1\rangle_1|\phi\rangle_2|\psi\rangle_3)\\ &= \frac{1}{\sqrt{2}}(|0,\psi,\phi\rangle+|1,\phi,\psi\rangle).\end{aligned} \tag{24}$$

At the third stage, another Hadamard gate is applied on the control qubit and result in the state:

$$\begin{aligned}|s_3\rangle &= \frac{1}{\sqrt{2}}H_1(|0,\psi,\phi\rangle+|1,\phi,\psi\rangle)\\ &= \frac{1}{\sqrt{2}}(H_1|0\rangle_1|\psi\rangle_2|\phi\rangle_3+H_1|1\rangle_1|\phi\rangle_2|\psi\rangle_3)\\ &= \frac{1}{\sqrt{2}}(\frac{|0\rangle_1+|1\rangle_1}{\sqrt{2}}|\psi\rangle_2|\phi\rangle_3+\frac{|0\rangle_1-|1\rangle_1}{\sqrt{2}}|\phi\rangle_2|\psi\rangle_3)\\ &= \frac{1}{2}(|0\rangle_1|\psi\rangle_2|\phi\rangle_3+|1\rangle_1|\psi\rangle_2|\phi\rangle_3)+\frac{1}{2}(|0\rangle_1|\phi\rangle_2|\psi\rangle_3-|1\rangle_1|\phi\rangle_2|\psi\rangle_3)\\ &= \frac{1}{2}|0\rangle(|\psi,\phi\rangle+|\phi,\psi\rangle)+\frac{1}{2}|1\rangle(|\psi,\phi\rangle-|\phi,\psi\rangle).\end{aligned} \tag{25}$$

Finally, the control qubit is measured. The probability of getting state $|0\rangle$ that measures the control qubit with the basis state $|0\rangle$ is given by:

$$\begin{aligned}P(|0\rangle) &= |\frac{1}{2}\langle 0|0\rangle(|\psi,\phi\rangle+|\phi,\psi\rangle)+\frac{1}{2}\langle 0|1\rangle(|\psi,\phi\rangle-|\phi,\psi\rangle)|^2\\ &= \frac{1}{4}|(|\psi,\phi\rangle+|\phi,\psi\rangle)|^2\\ &= \frac{1}{4}(|\psi,\phi\rangle+|\phi,\psi\rangle)^\dagger(|\psi,\phi\rangle+|\phi,\psi\rangle)\\ &= \frac{1}{4}((\langle\phi,\psi|+\langle\psi,\phi|)(|\psi,\phi\rangle+|\phi,\psi\rangle)\\ &= \frac{1}{4}((\langle\phi,\psi|\psi,\phi\rangle+\langle\phi,\psi|\phi,\psi\rangle+\langle\psi,\phi|\psi,\phi\rangle+\langle\psi,\phi|\phi,\psi\rangle)\\ &= \frac{1}{4}((\langle\phi|\psi\rangle\langle\psi|\phi\rangle+\langle\phi|\phi\rangle\langle\psi|\psi\rangle+\langle\psi|\psi\rangle\langle\phi|\phi\rangle+\langle\psi|\phi\rangle\langle\phi|\psi\rangle)\\ &= \frac{1}{4}(|\langle\psi|\phi\rangle|^2+1+|\langle\psi|\phi\rangle|^2+1)\\ &= \frac{1}{2}+\frac{1}{2}|\langle\psi|\phi\rangle|^2.\end{aligned} \tag{26}$$

From this equation, the probability $P(|0\rangle)=0.5$ means that the state $|\psi\rangle$ and the state $|\phi\rangle$ are orthogonal, and the probability $P(|0\rangle)=1$ means that these two states are identical. Likewise, the probability of getting state $|1\rangle$ that using the basis state $|1\rangle$ to measure the control qubit is:

$$P(|1\rangle)=\frac{1}{2}-\frac{1}{2}|\langle\psi|\phi\rangle|^2. \tag{27}$$

Thus, the overlap $\langle\psi|\phi\rangle$ makes a connection with the measurement probability of the control qubit successfully in the final quantum state.

3.3 | Phase Estimation

Phase estimation is one of the core components of HHL-like algorithms. HHL algorithm is proposed by Harrow, Hassidim and Lloyd[22], thus the algorithm is abbreviated by these three researchers. Particularly, HHL algorithm solves the problem of solving linear equations, and it is able to achieve the exponential acceleration over classical algorithms under some specific conditions additionally. Therefore, it has a widely use in the area of quantum machine learning. Here we would like to talk about the key part of HHL algorithm—Phase Estimation.

Consider this equation,

$$I|0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1 \cdot |0\rangle, \tag{28}$$

This is a simple example to explain that 1 is the eigenvalue of the given eigenvector $|0\rangle$. In general, the problem is that given an equation

$$U|\psi\rangle = \lambda|\psi\rangle, \tag{29}$$

where U is an unitary quantum gate, we would like to find an eigenvalue λ of the eigenvector $|\psi\rangle$ satisfying it. Because U is unitary, the eigenvalue can be expressed as:

$$\lambda = e^{2\pi i\theta}, \tag{30}$$

where the phase θ is in $[0, 1)$. Go back to Eq. (28) and (30), from the eigenvalue λ equals 1, it is clear that the phase θ is estimated as 0. Thus, the target of phase estimation is to estimate the unknown phase θ that the unitary U has an eigenvector which has an eigenvalue $e^{2i\pi\theta}$.

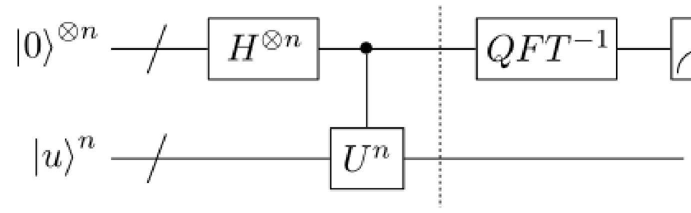

FIGURE 6 The structure of phase estimation algorithm.

Fig. 6 shows the fundamental structure of the whole algorithm. It includes two parts. In the left parts, the states are initialised as

$$|\psi_0\rangle = |0\rangle^{\otimes n}|u\rangle, \tag{31}$$

where $|u\rangle$ is the eigenvector state. Here the state $|0\rangle$ includes n qubits and the state $|u\rangle$ is also n-dimentional. Then the Hadamard gate is applied on the initialised basis state $|0\rangle$ and it gets the state

$$|\psi_1\rangle = (H \otimes I)|0\rangle^{\otimes n}|u\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n}|u\rangle. \tag{32}$$

After this, a controlled U gate is applied on the eigenvector state $|u\rangle$ and results in the state

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2^n}}\left(|0\rangle + U^{2^{n-1}}|1\rangle\right) \otimes \left(|0\rangle + U^{2^{n-2}}|1\rangle\right) \otimes \ldots \otimes \left(|0\rangle + U^{2^0}|1\rangle\right) \otimes |u\rangle \\ &= \frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i 2^{n-1}\theta}|1\rangle\right) \ldots \otimes \left(|0\rangle + e^{2\pi i 2^0\theta}|1\rangle\right) \otimes |u\rangle. \end{aligned} \tag{33}$$

At this point, the left part has been completed and it stores the eigenvalues of the operator U into the probability amplitude of the first n-qubit state.

In the right-hand part, for fitting some particular patterns when it implements the inverse quantum Fourier transform, the last equation can be re-written as:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i\frac{k2^n\theta}{2^n}}|k\rangle|u\rangle. \tag{34}$$

Compare the control qubit state $\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i\frac{k2^n\theta}{2^n}}|k\rangle$ with the form of Fourier-transformed state

$$|j\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi i\frac{kj}{N}}|k\rangle, \tag{35}$$

when $N = 2^n$ and $j = 2^n\theta$, the state $|j\rangle = |2^n\theta\rangle$ can be recovered by applying an inverse Fourier transform. Namely, it will get the state:

$$\begin{aligned}
|\psi_3\rangle &= QFT^{-1}\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i\frac{k2^n\theta}{2^n}}|k\rangle|u\rangle \\
&= \frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi ik\theta}e^{-2\pi i\frac{kj}{2^n}}|j\rangle|u\rangle \\
&= \frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi i\frac{k(2^n\theta-j)}{2^n}}|j\rangle|u\rangle\,. \qquad (36)
\end{aligned}$$

Because the state $|j\rangle$ is the basis state, it can only be encoded discretely. While the phase θ is a continuous variable which should be $0 \le \theta < 1$.

3.4 | HHL Algorithm

HHL algorithm is designed by Aram Harrow, Avinatan Hassidim and Seth Lloyd, and named after them[22]. This algorithm, published in 2009, led to a real take-off in the field of quantum machine learning, and HHL-based machine learning algorithms have emerged in a number of papers over the past few years. HHL is a famous quantum algorithm for solving linear systems. Linear system is the core of many scientific and engineering fields. Because HHL algorithm achieves the exponential speedup over classical algorithm under specific conditions, it can be widely used in data processing, machine learning, numerical calculation and other scenarios in the future.

The core steps of HHL algorithm are shown in Fig. 7. As it is shown, HHL algorithm mainly includes three subroutines—phase estimation, controlled rotation and inverse phase estimation. The description of this procedure is summarised as below.

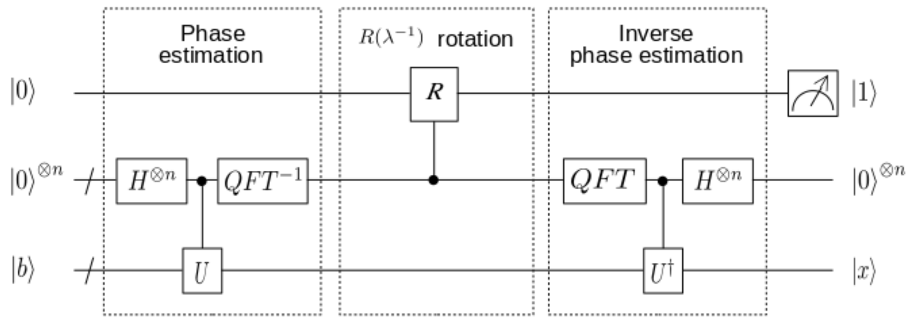

FIGURE 7 The procedure of HHL algorithm.

1. Encode the vector $\vec{b}$ into the quantum state:

$$|b\rangle = \sum_{i=1}^{N} b_i\,|i\rangle\,. \qquad (37)$$

2. Apply the unitary operator e^{iAt} to $|b\rangle$, where A is a Hermitian matrix and $A = \sum_j \lambda_j u_j u_j^{\dagger}$. After the processing of the phase estimation, the state becomes into

$$e^{iAt}\,|b\rangle = \sum_{j=1}^{N} \beta_j \left|u_j\right\rangle\left|\lambda_j\right\rangle, \qquad (38)$$

where $\left|u_j\right\rangle$ is the eigenvector of A, λ_j is the eigenvalue of A, and

$$|b\rangle = \sum_{j=1}^{N} \beta_j \left|u_j\right\rangle. \qquad (39)$$

It is worth noting that if A is not Hermitian, in the paper[22], a method to deal with this problem is presented. Define a Hermitian matrix C as the input, and make C

$$C = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}. \tag{40}$$

Solve the equation

$$C\vec{y} = \begin{pmatrix} \vec{b} \\ 0 \end{pmatrix} \tag{41}$$

to obtain

$$\vec{y} = \begin{pmatrix} 0 \\ \vec{x} \end{pmatrix}, \tag{42}$$

so it can solve for $\vec{x}$ for the original problem $A\vec{x} = \vec{b}$.

3. Then the key step is the rotation gate in the middle of the circuit. Rotate the ancilla qubit with $\left|\lambda_j\right\rangle$ as the control qubit, and after the rotation, it will get the state:

$$\sum_{j=1}^{N} \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} \left|0\right\rangle + \frac{C}{\lambda_j} \left|1\right\rangle \right) \beta_j \left|\lambda_j\right\rangle \left|u_j\right\rangle, \tag{43}$$

which means it performs the linear map taking the state $\left|\lambda_j\right\rangle$ to $\frac{C}{\lambda_j}\left|\lambda_j\right\rangle$, where C is a normalising constant.

4. The final step is the inverse phase estimation. In this step, it performs the linear map taking the state $\left|\lambda_j\right\rangle$ to the state $\left|0\right\rangle$. Observe Eq. (43), there is a proportion left of the state:

$$\sum_{j=1}^{N} \beta_j \left|\lambda_j\right\rangle \left|u_j\right\rangle. \tag{44}$$

If the result of the measurement of the ancilla qubit is 1, the final result of $\left|x\right\rangle$ will be

$$\left|x\right\rangle = \sqrt{\frac{1}{\sum_{j=1}^{N} C^2 |\beta_j|^2 / |\lambda_j|^2}} \sum_{j=1}^{N} \beta_j \frac{C}{\lambda_j} \left|u_j\right\rangle. \tag{45}$$

Here, the $\left|x\right\rangle$ it obtains is a quantum way to represent the desired solution of vector $\vec{x}$. At this stage, it cannot read out all components of $\vec{x}$. However, most of cases are one is not interested in $\vec{x}$ itself, but rather the expectation value of a linear operator M acting on $\vec{x}$. By making a measurement M, it allows to obtain an estimation of the expectation value $\left\langle x\right| M \left|x\right\rangle$. This indicates that a variety of useful features of the vector $\vec{x}$, such as normalisation, weights in different parts of the state space, and so on, can be extracted without actually computing all values of the solution vector $\vec{x}$. To sum up, HHL is not exactly an algorithm for solving a system of linear equations in logarithmic time, but rather an algorithm for appoximately preparing a quantum superposition of the form $\left|x\right\rangle$, where x is the solution of a linear system $A\vec{x} = \vec{b}$[23].

4 | APPLICATION ALGORITHMS FOR QUANTUM MACHINE LEARNING

For using the basic quantum algorithms as subroutines, machine learning algorithms will be able to get speedup in different degrees, since the subroutines such as HHL and amplitude amplification can provide exponential speedup and quadratic speedup separately. Table 2 lists the speedup of some quantum machine learning algorithms[24,22,25,26,27].

Although the speedup occurs in different ways in different machine learning settings[25], it still can be perceived intuitively. As it is known, classical data is in the form of N-dimensional vectors. To map these N-dimensional vectors onto a quantum state, it needs only $\log_2 N$ qubits, which means if the data is stored in a quantum random access memory (QRAM), this mapping will take $O(\log_2 N)$ steps[28]. Some researchers have already addressed the time complexity of specific steps of quantum operations. For instance, quantum Fourier transform and matrix inversion take time $O(poly(\log N))$, and distance estimation and inner product take time $O(\log N)$[22,14]. In addition, the problem of clustering, be more specific, assigning N-dimensional vectors into one of clusters of M states, takes time $O(\log(MN))$ on a quantum computer. It is an exponential speedup due to that the best known classical algorithm takes time $O(poly(MN))$[25].

Algorithms	Subroutines	Speedup
Quantum support vector machine	HHL	Exponential
Quantum k-means clustering	Phase estimation, swap-test, Grover's algorithm	Quadratic
Quantum principle component analysis	HHL	Exponential
Quantum linear discriminant analysis	HHL	Exponential

TABLE 2 Speedup for some given machine learning algorithms

4.1 | Quantum Support Vector Machine

The support vector machine (SVM) algorithm is a popular supervised machine learning algorithm, which is used to solve problems of binary classification. The main idea of this algorithm is to find a hyperplane that can distinguish two classes of data including different features and be used as a decision boundary for more data classification in the future[29,30,31]. Fig. 8 illustrates this kind of classification problems. In mathematics, this classification problem is formulated as finding the maximum margin

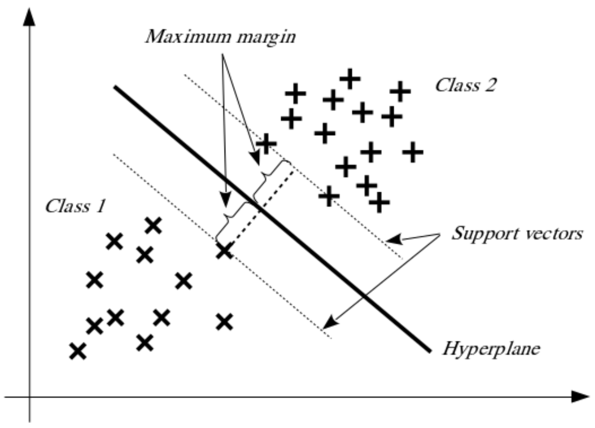

FIGURE 8 The illustration of SVM for classification problems

between the hyperplane and the closest data points to it, so that the classifier hyperplane makes these two classified classes have the maximum margin. Thus, SVM is also called maximum margin classifiers. In Fig. 8, there are two classes of data—45-degree cross as class 1 and vertical-horizontal cross as class 2. Imagine that this is a sectional view and one is looking at it from the side, then this data set is linearly separable in some dimensions. The data points in the parallel dashed lines are called support vectors. In addition, the line in the middle of these two classes is actually a hyperplane, which is corresponding to the equation:

$$\vec{w} \cdot \vec{x} + b = 0, \tag{46}$$

where $\vec{w}$ is the normal vector of the hyperplane and b is a constant that represents the offset. Therefore, it has the relationship:

$$\begin{aligned} &\vec{w} \cdot \vec{x}_i + b \geq 1, && \text{for } \vec{x}_i \text{ in the positive class,} \\ &\vec{w} \cdot \vec{x}_i + b \leq -1, && \text{for } \vec{x}_i \text{ in the negative class.} \end{aligned} \tag{47}$$

Here, the problem turns into an optimisation problem and the optimisation objective is to estimate parameters $\vec{w}$ and b that makes:

$$\begin{aligned} &\max_{\vec{w},b} \frac{2}{|\vec{w}|} \\ &s.t. \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \end{aligned} \tag{48}$$

where $\frac{2}{|\vec{w}|}$ is the margin between these two classes.Once the parameters $\vec{w}$ and b have estimated, the classification result of a new vector $\vec{x}_0$ can be determined by

$$y(\vec{x}_0) = \mathrm{sgn}(\vec{w} \cdot \vec{x}_0 + b). \tag{49}$$

In the classical SVM, it will traverse all the samples and the features of every sample which will result in a polynomial complexity of $O(poly(NM))$, where N is the number of training data points and M is the number of features. However, QSVM alogorithm is able to provide an exponential speedup, which is $O(log(NM))$[24]. In addition, QSVM is one of the machine learning algorithms which has already been implemented on quantum computing hardware[4]. Actually, there are two versions of QSVM that gives quadratical and exponential speedup separately. One of them focuses on solving non-convex optimisation problems by involving Grover's algorithm as a subroutine[32]. The other one focuses on the analasis of the least-squares approximation of SVM[24]. In this paper, we will summarise and explain the latter method.

The normal vector $\vec{w}$ can be represented as

$$\vec{w} = \sum_{i=1}^{N} \alpha_i \vec{x}_i, \tag{50}$$

where α_i is the weight of the ith training vector $\vec{x}_i$. Then the parameters need to be optimised turn to α_i and b. There is a detailed description of QSVM algorithm in[24]. The main idea of that work is to employ the least-squares reformulation of SVM that circumvents the quadratic programming and obtains the parameters from solving a linear equation:

$$F\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} \equiv \begin{pmatrix} 0 & 1 \\ I & K+\gamma^{-1}I_N \end{pmatrix}\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y} \end{pmatrix}. \tag{51}$$

Here, K is the linear kernel matrix that defined as $K_{i,j} = \vec{x}_i \cdot \vec{x}_j$.

First of all, by applying the training-data oracle, the classical training data is encoded as

$$|\vec{x}_i\rangle = \frac{1}{|\vec{x}_i|}\sum_{i=1}^{M}(\vec{x}_i)_j\,|j\rangle\,. \tag{52}$$

From the initial state $(1/\sqrt{M})\sum_{i=1}^{M}|i\rangle$ and the training-data oracle, we can prepare the state

$$|s\rangle = \frac{1}{\sqrt{N_s}}\sum_{i=1}^{M}|\vec{x}_i|\,|i\rangle\,|\vec{x}_i\rangle \tag{53}$$

Then to optimise the hyperplane parameters b and a_i, use HHL algorithm to solve the linear equations. The hyperplane parameters are determined by the matrix inversion:

$$(b, \vec{a}^T)^T = F^{-1}(0, \vec{y}^T)^T. \tag{54}$$

Due to the quantum register is initialised as

$$|0, y\rangle = \frac{1}{\sqrt{N_{0,y}}}\left(|0\rangle + \sum_{i=1}^{M} y_i\,|i\rangle\right), \tag{55}$$

by performing the matrix inversion of F, the quantum state is transfered to

$$|b, a\rangle = \frac{1}{\sqrt{N_{b,a}}}\left(b\,|0\rangle + \sum_{i=1}^{M} a_i\,|i\rangle\right). \tag{56}$$

With the optimised parameters b and a_i, the classification result then can be represented as

$$y(\vec{x}_0) = \mathrm{sgn}\left(\sum_{i=1}^{M} a_i(\vec{x}_i \cdot \vec{x}_0) + b\right). \tag{57}$$

4.2 | Quantum K-means Clustering

K-means clustering is one of the typical unsupervised machine learning algorithms. The clustering is a method to automatically divide a pile of unlabeled data into several classes. This method should ensure that the data of the same class has similar features. K-means algorithm has the property that the number of classes should be given in advance. In addition, there is an important assumption of k-means that the similarity between data can be measured by Euclidean distance, which means that the smaller the Euclidean distance, the higher the similarity between two data. The most important step in k-means algorithm is the calculation of Euclidean distance between given centroids of each cluster and data points[33,34,35]. For calculating Euclidean distance in quantum k-means, the basic algorithm swap-test is employed[36,25]. The whole quantum k-means algorithm is decribed as follows.

As it is in classical case, number of k clusters must be determined in advance. The initialisation of these k cluster centroids can be done by any methods that adopted in classical k-means algorithm. To access the data in quantum way, the information in vector $\vec{a}$ is encoded as:

$$\frac{\vec{a}}{|\vec{a}|} \rightarrow |a\rangle = \sum_{j=1}^{N} \frac{a_j}{|a|} |j\rangle . \tag{58}$$

To calculate the Euclidean distance $|\vec{a} - \vec{b}|^2$, two states are prepared and initialised as:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0, a\rangle + |1, b\rangle), \tag{59}$$

$$|\phi\rangle = \frac{1}{\sqrt{Z}}(|a|\,|0\rangle - |b|\,|1\rangle), \tag{60}$$

where $Z = |a|^2 + |b|^2$. From the prepared state $|\psi\rangle$ and $|\phi\rangle$, the overlap of these two states is

$$\langle\phi|\psi\rangle = \frac{1}{\sqrt{2Z}}(|a|\,|a\rangle - |b|\,|b\rangle). \tag{61}$$

Refer to the quantum information encoding at Eq. (58), the overlap can be further expressed as

$$\langle\phi|\psi\rangle = \frac{1}{\sqrt{2Z}}(a - b). \tag{62}$$

Swap positions, it is

$$a - b = \sqrt{2Z}\,\langle\phi|\psi\rangle . \tag{63}$$

Therefore, the Euclidean distance $|\vec{a} - \vec{b}|^2$ is

$$|a - b|^2 = 2Z|\,\langle\phi|\psi\rangle\,|^2. \tag{64}$$

By using the basic algorithm swap-test, the overlap $\langle\phi|\psi\rangle$ can be evaluated and then the Euclidean distance is obtained.

After all the distance between the data point and each cluster centroid obtained, the data point can be subsumed under the cluster that has the minimum distance with it and new centroids should be recalculated. To complete the algorithm, the whole procedure should be repeated several times until the result reaches convergence.

4.3 | Quantum Dimensionality Reduction

In the field of machine learning, dimensionality reduction means that some mapping methods are used to map the data points in the original high-dimensional space to the low-dimensional space. Dimensionality reduction is actually to learn a mapping function $f : x \rightarrow y$, where x is the original data points, which is usually represented by the vector, and y is the low-dimensional vector representation after data points mapping. Additionally, the dimension of y is usually less than the dimension of x. The reason why the dimensionality reduction is needed is that in the original high-dimensional space, a mass of redundant information and noise information is included, so that errors and the low accuracy are caused in practical applications. By dimensionality reduction, it hopes to reduce the errors caused by redundant information and improve the accuracy of identification. On the other hand, it hopes to find the essential structural characteristics of data through the dimensionality reduction algorithms as well[37,38,39]. In this paper, we mainly focus on two popular dimensionality reduction algorithms in quantum machine learning—Quantum Principal Component Analysis (QPCA) and Quantum Linear Discriminant Analysis (QLDA).

4.3.1 | QPCA

Principal component analysis (PCA) is a widely used dimensionality reduction method. It is an unsupervised dimensionality reduction method, which compresses high dimensional eigenvectors into low dimensional eigenattributes. The goal of this algorithm is to map high-dimensional data to low-dimensional space representation through some linear projection, and expect that the variance of data on the projected dimension is the largest, so as to use smaller data dimension and retain more characteristics of original data points[40]. QPCA involves the quantum phase estimation subroutine to deal with eigenvectors and eigenvalues. This algorithm is proved that it is able to provide exponential speedup over any known classical algorithm[26]. the procedure of QPCA is described as below.

Firstly, the classical data should be normalised and encoded as quantum states. For a set of N dimensional vectors x^i with $i = 1, 2, ...M$, we subtract the mean $\bar{x}$:

$$\begin{aligned} x^i &\rightarrow x^i - \bar{x} \\ \bar{x} &= \frac{1}{M}\sum_{i=1}^{M} x^i, \end{aligned} \tag{65}$$

and normalise as

$$\begin{aligned} x^i &\rightarrow \frac{x^i}{|x^i|} \\ |x| &= \sqrt{\sum_{k=1}^{N} x_k^2}. \end{aligned} \tag{66}$$

Then encode the classical data as quantum states:

$$x \rightarrow |x\rangle = \sum_{k=1}^{N} x_k |k\rangle . \tag{67}$$

As the core of QPCA algorithm, phase estimation requires to construct two input parts in advance—quantum state ρ and controlled U operator. Here, the quantum state ρ can be defined by the following density matrix:

$$\begin{aligned} \rho &= \frac{1}{M}\sum_{i=1}^{M} |x^i\rangle \langle x^i| \\ &= \frac{1}{M}\begin{bmatrix} \sum_{i=1}^{M} x_1^i x_1^i & \sum_{i=1}^{M} x_1^i x_2^i & \cdots & \sum_{i=1}^{M} x_1^i x_N^i \\ \sum_{i=1}^{M} x_2^i x_1^i & \sum_{i=1}^{M} x_2^i x_2^i & \cdots & \sum_{i=1}^{M} x_2^i x_N^i \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{M} x_N^i x_1^i & \sum_{i=1}^{M} x_N^i x_2^i & \cdots & \sum_{i=1}^{M} x_N^i x_N^i \end{bmatrix}, \end{aligned} \tag{68}$$

while the controlled U operator can be generated as $e^{-i\rho t}$ according to the reference[26].

After we have these inputs, the quantum phase estimation subroutine can be applied. For QPCA algorithm, the unitary operator $U = e^{-i\rho t}$ is not applied on the eigenvector, but density matrix ρ instead. The output of this application of quantum phase estimation is:

$$\sum_{j=1}^{M} \lambda^j \left|\tilde{\lambda^j}\right\rangle \left\langle \tilde{\lambda^j}\right| \otimes |x^j\rangle \langle x^j| . \tag{69}$$

Lastly, sampling from this state is able to help us obtain features of eigenvectors.

4.3.2 | QLDA

LDA is an another dimensionality reduction algorithm designed for classification. For the problem of classification, the feature space of input data is mapped from high dimension to low dimension, so as to achieve the dimensionality reduction. LDA retains the information of classes discrimination as much as possible, that is, data belongs to a class in the original high-dimensional space also belongs to a class in the low-dimensional space, and vice versa. The goal of LDA dimensionality reduction algorithm is to find a mapping, which can maximise the distance between classes and minimise the distance within classes[41,42]. Consider that there are M real value input data vectors $x_i \in \mathbb{R}^N, 1 \leq i \leq M$ that each belongs to one of k classes, we have the between-class scatter matrix of the dataset

$$S_B = \sum_{c=1}^{k} (\mu_c - \bar{x})(\mu_c - \bar{x})^T, \tag{70}$$

where μ_c is the within-class mean of class c and $\bar{x}$ denotes the mean of all data vectors x. And also, we have the within-class scatter matrix of the dataset

$$S_W = \sum_{c=1}^{k} \sum_{x\in c} (x - \mu_c)(x - \mu_c)^T. \tag{71}$$

Then the goal is to find a direction of projection $\vec{w} \in \mathbb{R}^N$ that maximises the between-class variance $\vec{w}^T S_B \vec{w}$ relative to the within-class variance $\vec{w}^T S_W \vec{w}$. In mathematics, it is assumed that the classes have approximate multi-variable Gaussian distribution and the covariance is similar additionally, it is the problem of maximising the objective function

$$J(\vec{w}) = \frac{\vec{w}^T S_B \vec{w}}{\vec{w}^T S_W \vec{w}}. \tag{72}$$

And the equivalent optimisation problem is

$$\begin{aligned} \min_{\vec{w}} \quad & -\vec{w}^T S_B \vec{w}, \\ s.t. \quad & \vec{w}^T S_W \vec{w} = 1. \end{aligned} \tag{73}$$

Then minimise the Lagrangian

$$\mathcal{L}_P = -\vec{w}^T S_B \vec{w} + \lambda(\vec{w}^T S_W \vec{w} - 1), \tag{74}$$

where λ is the desired Lagrange multiplier. Following the Karush-Kuhn-Tucker (KKT) conditions, we have

$$\frac{S_B}{S_W}\vec{w} = \lambda \vec{w}. \tag{75}$$

It means that $\vec{w}$ is an eigenvector of $\frac{S_B}{S_W}$. Therefore, the objective function can be obtained as $J(\vec{w}) = \lambda$. In general case, keeping p eigenvecotrs for each data is necessary, thus a projection subspace with $N \times p$ dimensions is needed. Then the objective function is generalised as:

$$J(W) = \frac{W^T S_B W}{W^T S_W W}, \tag{76}$$

where W is an $N \times p$ matrix.

The goal of QLDA is to speed up the classical LDA algorithm. Compared with the classical LDA algorithm, the exponential speedup is achieved on the number of training vectors M and dimension N of feature space[27]. In the paper[27], the author proposes the algorithm as 4 steps as follows:

1. Construct S_B:

$$S_B = \frac{1}{A}\sum_{c=1}^{k} |\mu_c - \bar{x}|^2 \left|\mu_c - \bar{x}\right\rangle \left\langle \mu_c - \bar{x}\right|, \tag{77}$$

   and S_W:

$$S_W = \frac{1}{B}\sum_{c=1}^{k}\sum_{i\in c} |x_i - \mu_c|^2 \left|x_i - \mu_c\right\rangle \left\langle x_i - \mu_c\right|. \tag{78}$$

2. Use the method proposed in the paper[26], to construct density matrix $S_B^{1/2} S_W^{-1} S_B^{1/2}$.

3. Apply quantum phase estimation algorithm to solve the problem

$$S_B^{1/2} S_W^{-1} S_B^{1/2} v = \lambda v. \tag{79}$$

   It can obtain an approximation to the state

$$\rho = \sum_i \lambda_i \left|v_i\right\rangle \left\langle v_i\right| \otimes \left|\lambda_i\right\rangle \left\langle \lambda_i\right|, \tag{80}$$

   where λ_i is the eigenvalue of $S_B^{1/2} S_W^{-1} S_B^{1/2}$ and v_i is the eigenvector of $S_B^{1/2} S_W^{-1} S_B^{1/2}$. Furthermore, it obtains the corresponding p principal eigenvectors v_r of $S_B^{1/2} S_W^{-1} S_B^{1/2}$.

4. Again, use the technique in[26] to obtain eigenvectors of $S_W^{-1} S_B$

$$w_r = S_B^{-1/2} v_r. \tag{81}$$

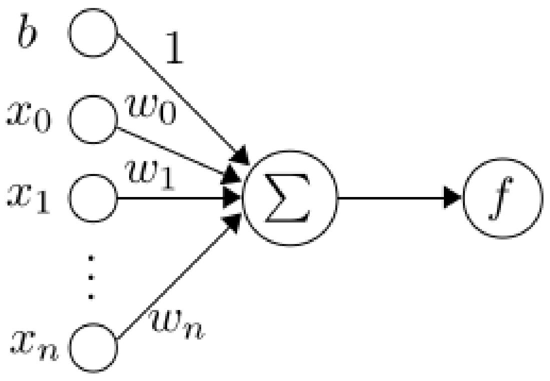

FIGURE 9 Single neuron model.

4.4 | Quantum Neural Networks

Researchers have studied a number of quantum machine learning algorithms that corresponding to their classical counterparts, some of them are able to achieve exponential speedup, while some of them are able to obtain quatratical speedup. However, the benefits of quantum neural networks are not yet specified in comparing with the classical one. The first step in creating a usable quantum neural network is to model a single quantum neuron. In 2018, researchers from the university of Pavia in Italy implemented the first single-layer neural network on a quantum computer[43]. In a classical neural network with a single neuron, the output is a weighted sum that maps the input vector to the binary output through the activation function. In quantum neural networks, the first layer encodes input vectors as quantum states. Then the second layer performs the unitary transform on the input, similar to the way the weight vector works in classical neural networks. Finally, the output is written on the ancilla qubit, producing the final output. It shows the quantum circuit in Fig. 10. They proposed the quantum neuron model and designed the

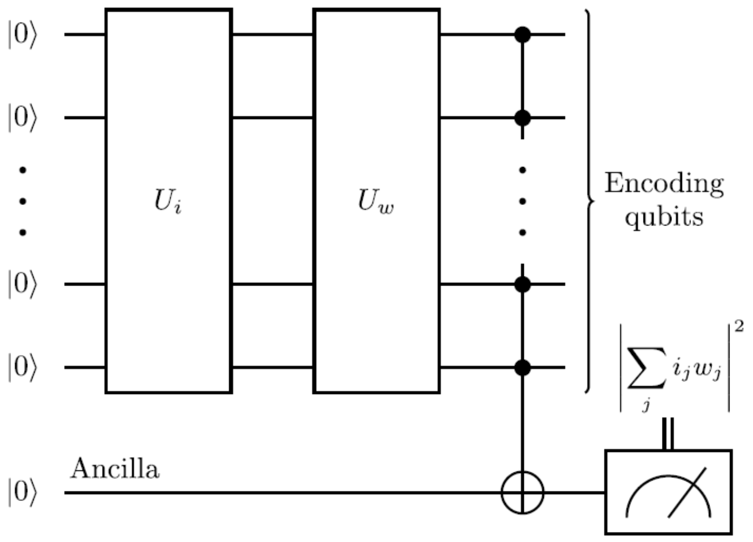

FIGURE 10 The quantum scheme of artificial neuron on quantum processor.

unitary transform.

Some other work such as[44] and[45] are also inspiring. In[44], a small quantum circuit defines a building block as the quantum neuron which can natually simulate neurons with threshold activation. In[45], it proposes a model of a quantum neural network based on the quantum neuron by the subroutine of a swap test. In addition, Google's Quantum AI team has built a theoretical model of a deep neural network that could be trained on a quantum computer[46]. Although there is no current hardware to actually implement this model, their results are encouraging. Once the hardware is available in the future, the framework they created will allow people to adopt quantum machine learning immediately.

In general, the reason why people develop quantum neural networks is to compare with classical neural networks, quantum neural networks have many advantages, including exponential memory capacity, fewer hidden neurons but higher performance, faster learning and processing speed, smaller scale and higher stability. These advantages are able to solve most of the limitations of classical neural networks. At present, people are trying to realise fully functional quantum neural network.

5 | CONCLUSION

This paper reviews some of the current research on quantum machine learning. It is worth noticing that this is not a complete review, but focused on the quantum version of some given supervised and unsupervised machine learning algorithms, which is based on the quantum circuit model. Even though some quantum versions of machine learning algorithms have been shown by many researchers to provide speedup over their classical conterparts, there are still some problems in this area. For instance, the problem of data input and output. It means that in some cases, reading the classical data may dominate the cost of quantum algorithms, so that it cannot speedup the whole algorithm at the macro level, and exactly reading out the data may be infeasible, which cannot meet the computing needs in some tasks of learning. Due to these problems, it is believed that applying quantum computing to quantum data directly is much more effective and efficient than to classical data. Therefore, quantum machine learning is better suited to solving problems in quantum systems themselves. However, with the development of research and theory, in the near future, we can verify whether these quantum machine learning algorithms can effectively help people solve problems about the data and decision-making that encountered today.

ACKNOWLEDGMENTS

This work was supported by HIRP under grant HO2018105132.

References

1. Shor PW. Proceedings of the 35th Annual Symposium on Foundations of Computer Science. *IEE Computer society press, Santa Fe, NM* 1994.

2. Grover LK. Quantum mechanics helps in searching for a needle in a haystack. *Physical review letters* 1997; 79(2): 325.

3. Hu F, Wang BN, Wang N, Wang C. Quantum machine learning with D-wave quantum computer. *Quantum Engineering* 2019; 1(2): e12. e12 QUE-2019-0007.R1doi: 10.1002/que2.12

4. Li Z, Liu X, Xu N, Du J. Experimental realization of a quantum support vector machine. *Physical review letters* 2015; 114(14): 140504.

5. Hu W. Empirical Analysis of a Quantum Classifier Implemented on IBM's 5Q Quantum Computer. *Journal of Quantum Information Science* 2018; 8(01): 1.

6. Xin T, Wei S, Cui J, et al. A Quantum Algorithm for Solving Linear Differential Equations: Theory and Experiment. 2018.

7. Xin T, Wang BX, Li KR, et al. Nuclear magnetic resonance for quantum computing: Techniques and recent achievements. *Chinese Physics B* 2018; 27(2): 020308. doi: 10.1088/1674-1056/27/2/020308

8. Zhang Y, Ni Q. Design and analysis of random multiple access quantum key distribution. *Quantum Engineering* 2020: e31. doi: 10.1002/que2.31

9. Oshurko I. Quantum machine learning. 2016.

10. Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. *Nature* 2017; 549(7671): 195.

11. Schuld M, Killoran N. Quantum machine learning in feature Hilbert spaces. *arXiv preprint arXiv:1803.07128* 2018.

12. Schuld M, Sinayskiy I, Petruccione F. An introduction to quantum machine learning. *Contemporary Physics* 2015; 56(2): 172–185.

13. Adcock J, Allen E, Day M, et al. Advances in quantum machine learning. *arXiv preprint arXiv:1512.02900* 2015.

14. Nielsen MA, Chuang IL. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press . 2010

15. Bennett CH, DiVincenzo DP. Quantum information and computation. *Nature* 2000; 404(6775): 247.

16. DiVincenzo DP. Two-bit gates are universal for quantum computation. *Physical Review A* 1995; 51(2): 1015.

17. Zalka C. Grover's quantum searching algorithm is optimal. *Physical Review A* 1999; 60(4): 2746.

18. Nannicini G. An Introduction to Quantum Computing, Without the Physics. *arXiv preprint arXiv:1708.03684* 2017.

19. Durr C, Hoyer P. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014* 1996.

20. Long GL, Li YS, Zhang WL, Niu L. Phase matching in quantum searching. *Physics Letters A* 1999; 262(1): 27–34. doi: 10.1016/s0375-9601(99)00631-3

21. Long GL. Grover algorithm with zero theoretical failure rate. *Physical Review A* 2001; 64(2). doi: 10.1103/physreva.64.022307

22. Harrow AW, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations. *Physical review letters* 2009; 103(15): 150502.

23. Aaronson S. Read the fine print. *Nature Physics* 2015; 11(4): 291.

24. Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. *Physical review letters* 2014; 113(13): 130503.

25. Lloyd S, Mohseni M, Rebentrost P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411* 2013.

26. Lloyd S, Mohseni M, Rebentrost P. Quantum principal component analysis. *Nature Physics* 2014; 10(9): 631.

27. Cong I, Duan L. Quantum discriminant analysis for dimensionality reduction and classification. *New Journal of Physics* 2016; 18(7): 073011.

28. Giovannetti V, Lloyd S, Maccone L. Quantum random access memory. *Physical review letters* 2008; 100(16): 160501.

29. Smola AJ, Schölkopf B. A tutorial on support vector regression. *Statistics and computing* 2004; 14(3): 199–222.

30. Burges CJ. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 1998; 2(2): 121–167.

31. Suykens JA, Vandewalle J. Least squares support vector machine classifiers. *Neural processing letters* 1999; 9(3): 293–300.

32. Anguita D, Ridella S, Rivieccio F, Zunino R. Quantum optimization for training support vector machines. *Neural Networks* 2003; 16(5-6): 763–770.

33. Likas A, Vlassis N, Verbeek JJ. The global k-means clustering algorithm. *Pattern recognition* 2003; 36(2): 451–461.

34. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 2002(7): 881–892.

35. Wagstaff K, Cardie C, Rogers S, Schrödl S, others . Constrained k-means clustering with background knowledge. In: . 1. ; 2001: 577–584.

36. Wittek P. 10 - Clustering Structure and Quantum Computing. In: Wittek P. , ed. *Quantum Machine Learning*Boston: Academic Press. 2014 (pp. 99 - 107)

37. Van Der Maaten L, Postma E, Herik V. dJ. Dimensionality reduction: a comparative. *J Mach Learn Res* 2009; 10: 66–71.

38. Sarwar B, Karypis G, Konstan J, Riedl J. Application of dimensionality reduction in recommender system-a case study. tech. rep., Minnesota Univ Minneapolis Dept of Computer Science; 2000.

39. Lee JA, Verleysen M. *Nonlinear dimensionality reduction.* Springer Science & Business Media . 2007.

40. Jolliffe I. Principal component analysis. In: Springer. 2011 (pp. 1094–1096).

41. Ye J, Janardan R, Li Q. Two-dimensional linear discriminant analysis. In: ; 2005: 1569–1576.

42. Mika S, Ratsch G, Weston J, Scholkopf B, Mullers KR. Fisher discriminant analysis with kernels. In: Ieee. ; 1999: 41–48.

43. Tacchino F, Macchiavello C, Gerace D, Bajoni D. An artificial neuron implemented on an actual quantum processor. *npj Quantum Information* 2019; 5(1): 26.

44. Cao Y, Guerreschi GG, Aspuru-Guzik A. Quantum neuron: an elementary building block for machine learning on quantum computers. *arXiv preprint arXiv:1711.11240* 2017.

45. Zhao J, Zhang YH, Shao CP, Wu YC, Guo GC, Guo GP. Building quantum neural networks based on swap test. *arXiv preprint arXiv:1904.12697* 2019.

46. Farhi E, Neven H. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002* 2018.

□