Quantum Computing and Machine Learning

Contents

What Is Quantum?

Quantum is the minimum amount of any physical entity in a microphysical system, such as energy, angular momentum and electric charge.

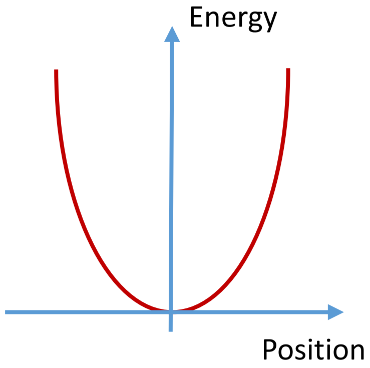
	Classical physics	Quantum physics
Energy	The physical quantity is considered continuous and the system energy can be any value. For example, a ball is released at any position at the beginning, and the system energy is a smooth and continuous parabola.	The physical quantity is quantized and the system energy can only be a discrete value. For example, the energy of a quantum harmonic oscillator can only be $\hbar\omega/2$ or $3\hbar\omega/2$, and the difference between the energy values must be an integer multiple of $\hbar\omega$.
State of motion	The motion state in the system can be completely determined and predicted. For a classical harmonic oscillator, its kinetic energy (velocity) and potential energy (position) can be 0 at the same time. For example, the position of the ball to the spring is balanced, and the system energy is 0.	The uncertainty principle of quantum mechanics shows that we cannot determine the precise position and momentum of a system at the same time. For quantum harmonic oscillators, the kinetic energy (velocity) and potential energy (position) cannot be 0 at the same time. Therefore, the system energy has the minimum value $\hbar\omega/2$.
Property	It is believed that microparticles can be composed of particles.	Quantum mechanics shows that particles have both the wave nature (wavefunction superposition) and particle nature (wavefunction collapse).

What Is Quantum?

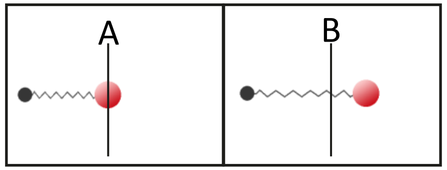
Classical physics

$$U = \frac{1}{2}kx^2$$

For example, the relationship between the system energy of a small ball (harmonic oscillator) bound to a spring and its initial position can be expressed by a parabolic curve.



Harmonic oscillator energy (classical theory)

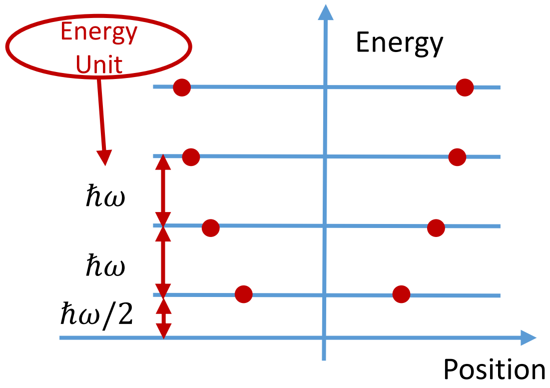


Classical oscillator

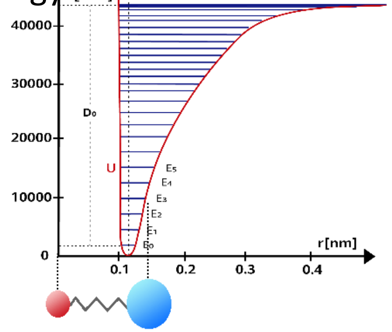
Quantum physics

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right) \quad n = 0,1,2 \ldots$$

For example, the vibration of a diatomic molecule can be approximated as a quantum harmonic oscillator, and its vibrational energy needs to be expressed in discrete energy levels.

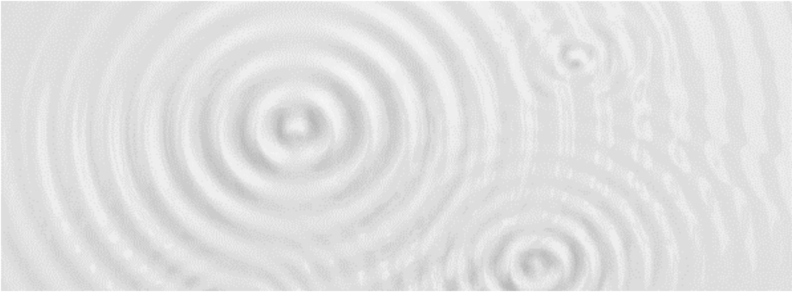


Harmonic oscillator energy (quantum theory)



Energy levels of molecular vibration of hydrogen chloride

Basic Concepts of Quantum Mechanics



Quantum Superposition

- Similar to waves in classical physics, the result of superposition of arbitrary quantum states will be an effective quantum state.
- Quantum superposition embodies the wave properties of quantum systems.
- Mathematically, quantum superposition comes from the linearity of the Schrodinger wave equation, that is, the linear combination of equation solutions is also the solution of the equation.
- Macro manifestation: double-slit interference pattern

Quantum Entanglement

- Two particles can be in an entangled state: when the first particle spins upward, the second particle spins downward, and vice versa. Neither of the particles has a definite spin orientation before the measurement.
- Now make the two particles in the entangled state leave each other in the opposite direction. Regardless of the distance, the spin orientation of the other particle can be determined immediately after the spin orientation of one particle is measured.

Quantum Uncertainty

- The position and momentum of a particle cannot be determined at the same time. The smaller the uncertainty of the position, the greater the uncertainty of the momentum, and vice versa.
- Mathematical expression: $\Delta x \Delta p \geq \frac{\hbar}{2}$

Development of Quantum Computing

Establishment of Quantum Mechanics

Development of Quantum Computing Theories

Quantum Computer Engineering

1930 · 1960 · 1980 · 1990 · 2000 · 2010 · 2020

- 1900–1930: Establishment of quantum mechanics
- 1935: Einstein, et al., put forward "EPR paradox".
- 1964: Bell inequality
- 1981: Feynman put forward "quantum simulation".
- 1985: Deutsch explained the concept of quantum Turing machine.
- 1993: Andrew Yao proved for the first time the equivalence between the quantum Turing machine model and the quantum circuit model.
- 1994: Shor proposed an algorithm for integer factorization.
- 1996: Grover proposed the quantum search algorithm.
- 2011: D-Wave announced the first commercial quantum annealer.
- 2016: IBM unveiled the 5-qubit superconducting quantum computer.
- Jan. 2017: D-Wave launched 2000Q.
- April 2018: Google released the 72-qubit superconducting computer chip Bristlecone.
- Aug. 2018: D-Wave demonstrated topology phase transition using an annealing quantum computer.
- October 2019: Google claimed its quantum computer achieved "quantum supremacy".
- Dec. 2020: Pan Jianwei team released nine chapters of optical quantum computing prototype machine to solve Gaussian wave color sampling.
- Nov. 2022: IBM released the 433-bit quantum chip.

Application Scenarios and Advantages of Quantum Computing

AI

The quantum-classical hybrid architecture is used to give full play to the advantages of quantum computing in the **NISQ phase**.

Fintech

The quantum optimization algorithm can be used to quickly search for the optimal solution in the **exponential parameter space**.

Biological computing

The functions of biological macromolecules and drug molecules can be **simulated more accurately** by general-purpose quantum computers.

Logistics and transportation

Quantum Approximate Optimization Algorithm (QAOA) can be used to find the optimal solution in **polynomial time**.

Material simulation

Quantum algorithms are used to calculate the energy potential energy surface of molecules, and the maximum **exponential acceleration** can be achieved.

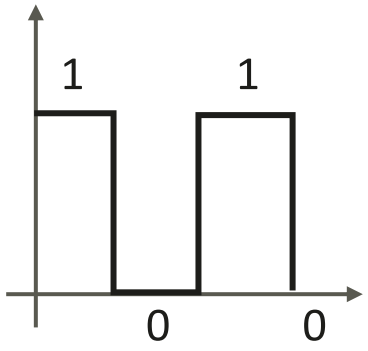
Security

The Shor algorithm can get results in **polynomial time**, whereas the classical algorithm needs **exponential time**. The Shor algorithm can easily crack the existing RSA encryption system, whereas the quantum encryption algorithm has absolute security.

Contents

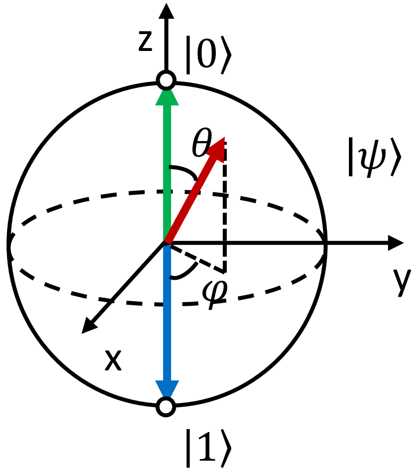
Quantum Bit and Quantum State

Bit



Bits are computing unit of classical computers for logical processing. **The classical bit that is either 0 or 1.** The N classical bits can represent only one number at a time.

Qubit



Qubits are computing unit of quantum computers for logical processing. **The qubit can be in state of 0 and 1, or a linear superposition state of 0 and 1.** The N qubits can represent **2^N** number at the same time.

Base vectors for quantum computing: state 0 is denoted as $|0\rangle$ and state 1 is denoted as $|1\rangle$ ($|\rangle$ is a right vector symbol)

Quantum superposition state: A single-bit quantum state is a vector $|\psi\rangle$ in a two-dimensional complex vector space, and may be represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Herein, α and β are complex numbers, and $|\alpha|^2 + |\beta|^2 = 1$ (the normalized single quantum bit state is a unit vector in the two-dimensional complex vector space).

The base vectors $|0\rangle$ and $|1\rangle$ are calculated and mapped to two orthogonal base vectors in a two-dimensional Hilbert space:

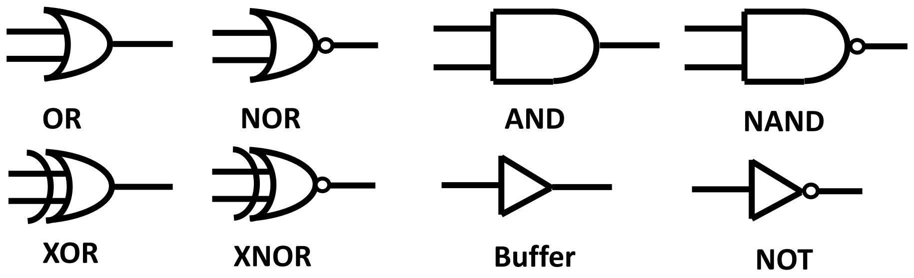
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Any single-bit quantum state can be represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Quantum Gate

Classical logic gate



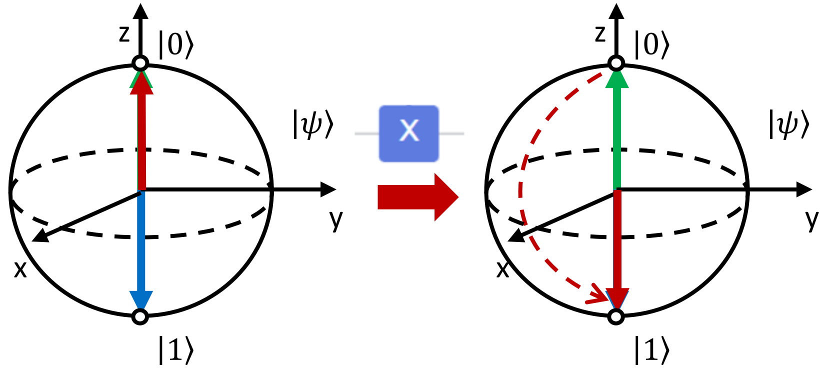
Quantum logic gate

X	Y	Z	H
Pauli-X (X)	Pauli-Y (Y)	Pauli-Z (Z)	Hadamard (H)

$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

- Different from classical logic gates, quantum logic gates are based on unitary transformation of qubits, which **makes quantum gate operations reversible**.
- Quantum gates often use the matrix representation. A single quantum gate can be represented by a unitary matrix $2^k \times 2^k$. The number of input qubits of a gate must be equal to that of the output qubits.
- **Unitary transformation**: a rotation of a quantum state, so that the inner product of the quantum state remains unchanged before and after the transformation. In the real number domain, it can be simplified as a classical rotation.

$$U^\dagger U = I \Rightarrow U^\dagger = U^{-1}$$



Basic Quantum Gates

Pauli-X (X)

X

$|0\rangle \xrightarrow{X} |1\rangle$

$|1\rangle \xrightarrow{X} |0\rangle$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pauli-Y (Y)

Y

$|0\rangle \xrightarrow{Y} i|1\rangle$

$|1\rangle \xrightarrow{Y} -i\,|0\rangle$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Pauli-Z (Z)

Z

$|0\rangle \xrightarrow{Z} |0\rangle$

$|1\rangle \xrightarrow{Z} -|1\rangle$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Hadamard (H)

H

$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Measurement gate

Measures the built quantum circuits, collects statistics on the collapsed bits at each gate, and finally obtains the statistical result.

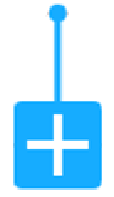
CNOT gate

The controlled-NOT (CNOT) gate is a two-qubit gate X that acts based on the input status of the control bit:

- If the control bit is 0, no operation is performed.
- **If the control bit is 1, a gate X is applied to the target bit.**

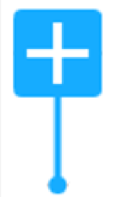
If the low level is the control bit, a matrix is represented as:

$$CONT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$



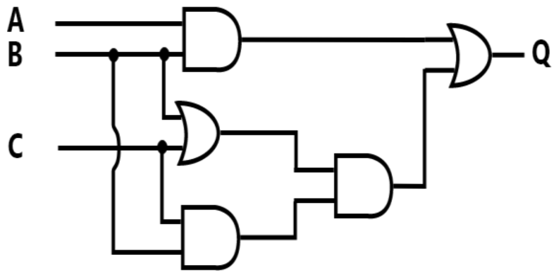
If the high level is the control bit, a matrix is represented as:

$$CONT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

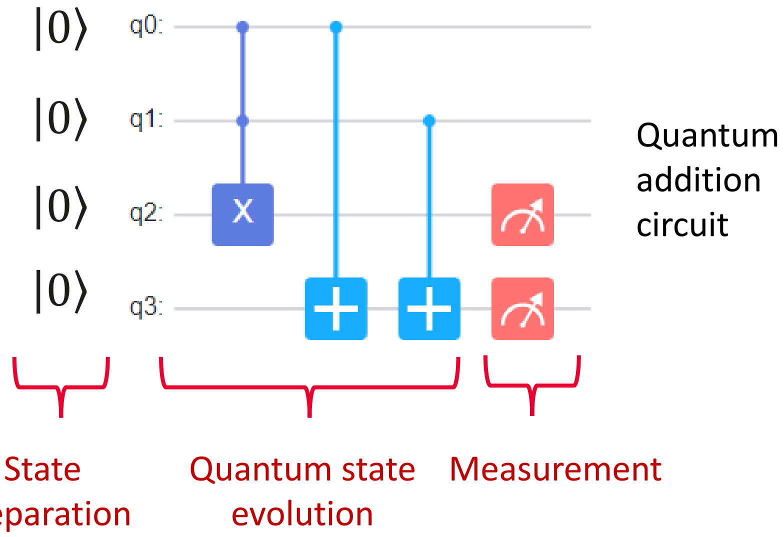


Quantum Circuit

Classical circuit



Quantum circuit



- Quantum circuits are some quantum gate sequences that act on qubits so that the quantum states of qubits evolve into new quantum states.
- Quantum circuits can be divided into three parts: state preparation, quantum state evolution, and quantum measurement.
  - State preparation ensures that qubits evolve from state $|0\rangle$.
  - A quantum circuit with a series of quantum gates works on the qubits and the qubits evolve.
  - The qubits are then measured to obtain the output result of the quantum circuit.
- Similar to a quantum gate, a quantum circuit may be considered as a larger unitary transformation or may be represented as a large unitary matrix, because the function of a quantum gate is equivalent to matrix multiplication, and the result of multiplying a series of unitary matrices is still a unitary matrix.

Preparation of Bell-state Quantum Circuits

The maximum entangled state of two qubits is called the Bell state.

There are four Bell states:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$|\beta_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$

$$|\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$|\beta_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

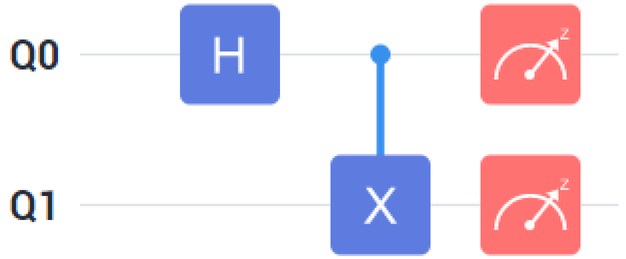
Generate Bell state

To generate Bell state $|\beta_{00}\rangle$, prepare a qubit in the $|00\rangle$ state, apply the Hadamard gate to change it to the maximum superposed state, and then use the CNOT gate.

$$U_{CNOT}(H \otimes I)|00\rangle$$
$$= U_{CNOT}(|0\rangle + |1\rangle)|0\rangle/\sqrt{2}$$

The Bell states generated vary depending on the initial states:

Input	Output
$\|00\rangle$	$(\|00\rangle + \|11\rangle)/\sqrt{2} \equiv \|\beta_{00}\rangle$
$\|01\rangle$	$(\|01\rangle + \|10\rangle)/\sqrt{2} \equiv \|\beta_{01}\rangle$
$\|10\rangle$	$(\|00\rangle - \|11\rangle)/\sqrt{2} \equiv \|\beta_{10}\rangle$
$\|11\rangle$	$(\|01\rangle - \|10\rangle)/\sqrt{2} \equiv \|\beta_{11}\rangle$



Basic Quantum Algorithms - Deutsch Algorithm

Problem

For $x \in \{0,1\}$, given an unknown Boolean function $f(x) \in \{0,1\}$, there are two possibilities of $f(x)$:

Case 1 (constant): $f(0) = f(1)$

Case 2 (balance): $f(0) \neq f(1)$

To know which case $f(x)$ belongs to, the classical algorithm needs to evolve the function at least twice for comparison—$f(0)$ and $f(1)$.

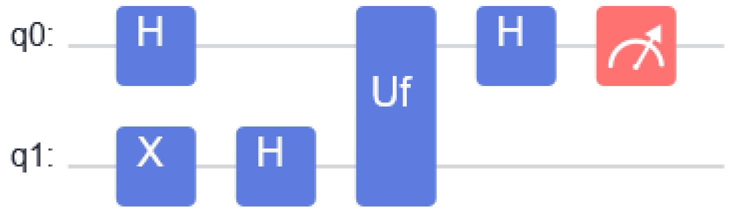
Deutsch quantum algorithm

1. Generate the superposed state: $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
2. Apply function $f(x)$:

$$\frac{1}{\sqrt{2}}\left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right)$$

For case 1, the generated quantum state is $|+\rangle$; for case 2, the generated quantum state is $|-\rangle$.

3. To discriminate between state $|+\rangle$ and state $|-\rangle$, you can use the Hadamard gate: $H|+\rangle = |0\rangle$, $H|-\rangle = |1\rangle$. Then, measure the quantum state. If state $|0\rangle$ is obtained, it is case 1. If state $|1\rangle$ is obtained, it is case 2. **With the quantum algorithm, the function only needs to be evolved once.**



General-Purpose Quantum Algorithms

Shor

$$p_1 \cdot p_2 = N \quad f(x) = a^x mod\ N$$

- **Problem: integer factorization**
- **Complexity**

  Classical: $\mathcal{O}(e^{1.9(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}})$

  Quantum: $\mathcal{O}((\log N)^3)$
- **Application scenarios and advantages:**

  The Shor algorithm can get results in **polynomial time**, whereas the classical algorithm needs **exponential time**. The Shor algorithm can easily **crack the existing RSA encryption system**, whereas the quantum encryption algorithm has absolute security.

Grover

$$f(x) = \begin{cases} \mathbf{0}, \boldsymbol{x} \neq \boldsymbol{x_{target}} \\ \mathbf{1}, \boldsymbol{x} = \boldsymbol{x_{target}} \end{cases}$$

- **Problem: unordered database search**
- **Complexity**

  Classical: $\mathcal{O}(N)$

  Quantum: $\mathcal{O}(\sqrt{N})$
- **Application scenarios and advantages:**

  The Grover algorithm is used to accelerate various algorithms and can get results **in sub-exponential time**, whereas the classical algorithm requires **exponential time**. It provides asymptotic acceleration for brute force cracking of the symmetric key algorithm (including collision attacks and original image attacks) and can crack 128-bit symmetric encryption keys in about 2^64 iterations, or 256-bit symmetric encryption keys in about 2^128 iterations.

HHL

$$A|x\rangle = |b\rangle$$

- **Problem: linear equation solving**
- **Complexity**

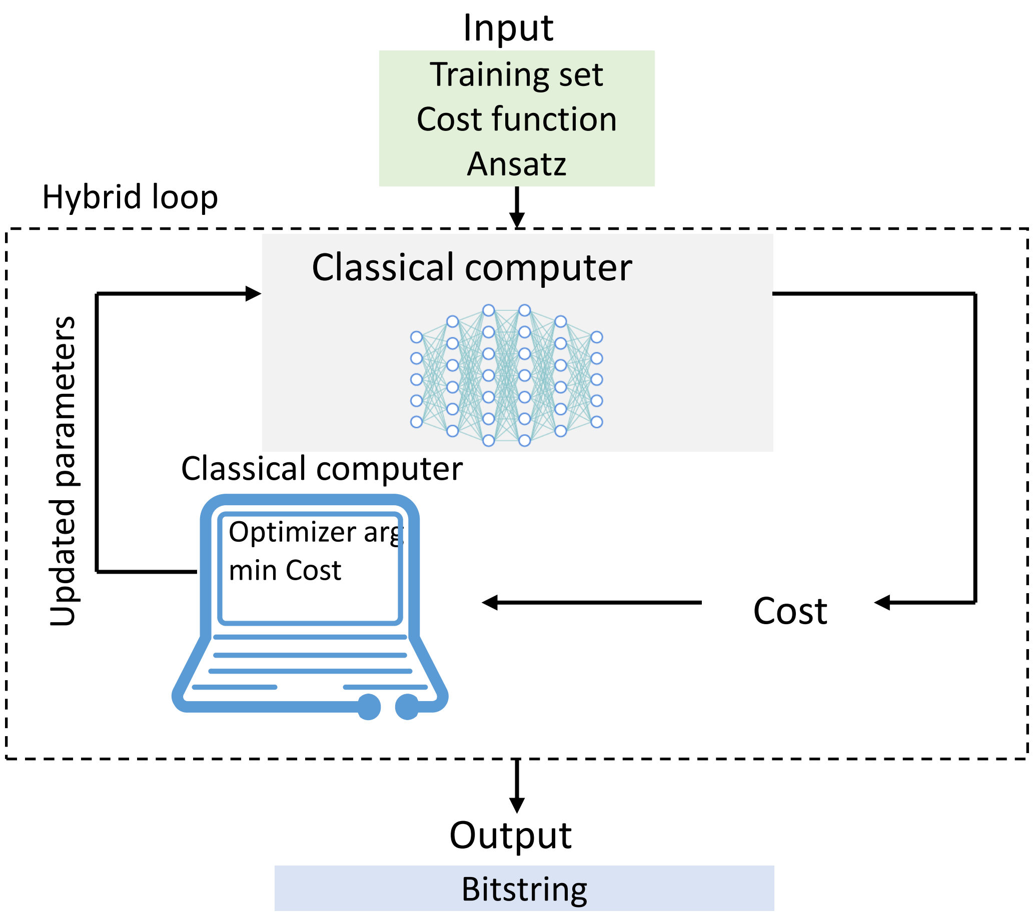
  Classical: $\mathcal{O}(N\kappa)$

  Quantum: $\mathcal{O}(\log(N)\,\kappa^2)$
- **Application scenarios and advantages:**

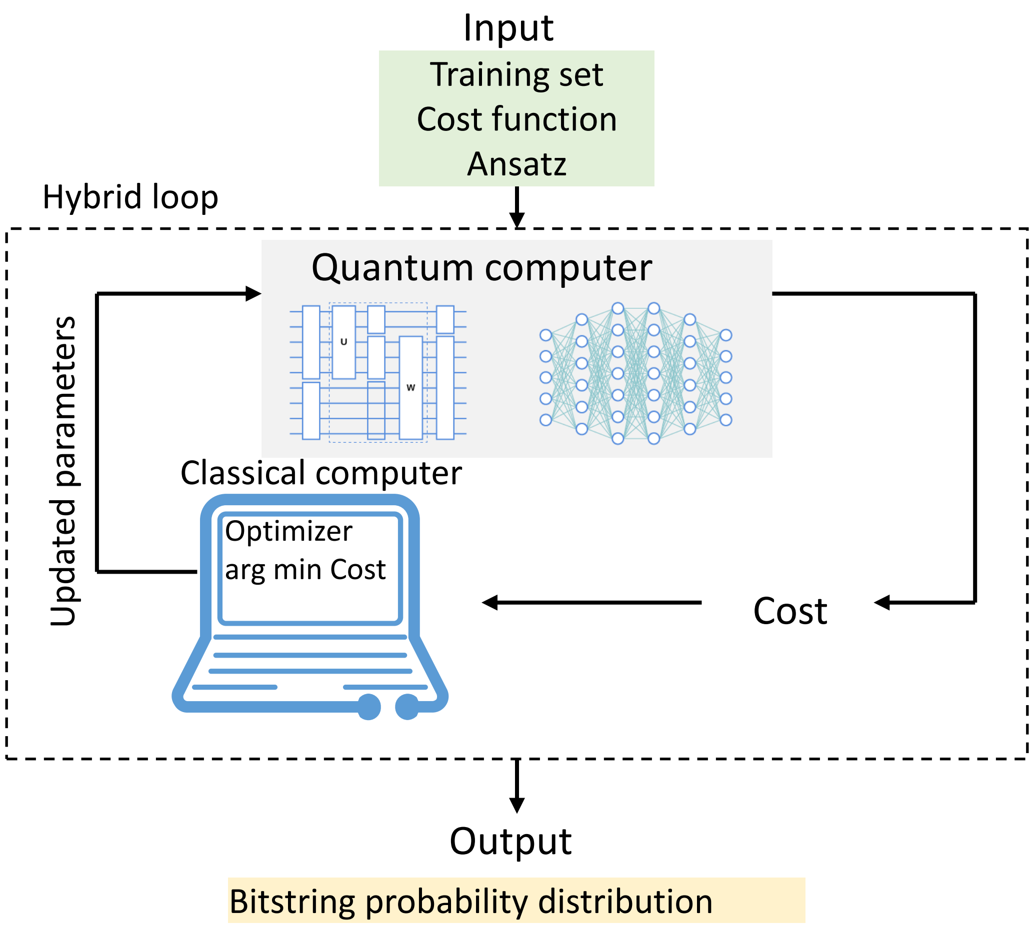
  Compared with classical algorithms, the HHL algorithm achieves **exponential accelerat**ion in solving linear equations and has advantages in **machine learning and numerical computing scenarios**. Combined with the Grover algorithm, it will be a key technology for breakthroughs in fields such as quantum machine learning and artificial intelligence in the future.

Variational Quantum Algorithm

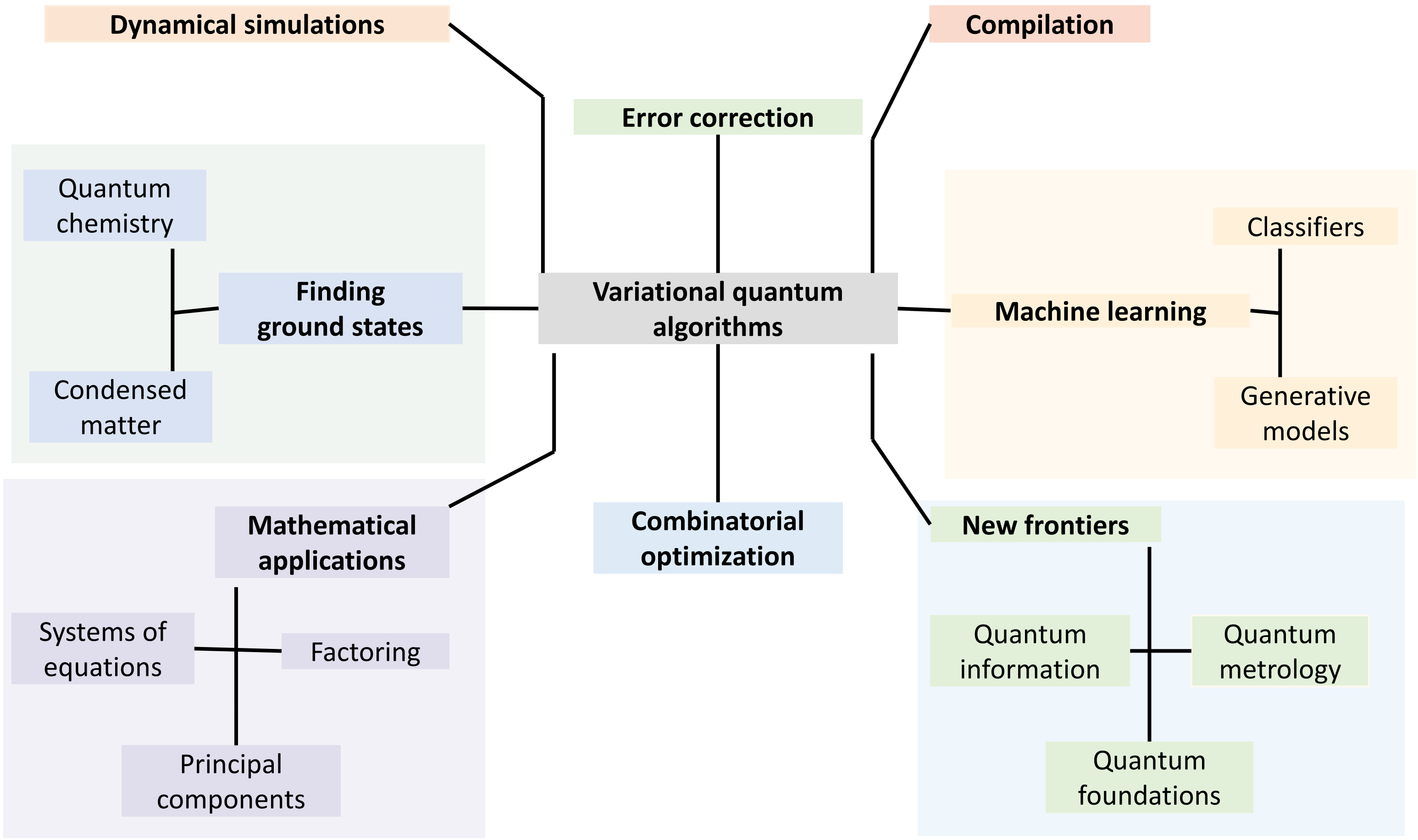
Classic Machine Learning Algorithm Flow



Variational Quantum Algorithm Flow



Application Scenarios of Variational Quantum Algorithms



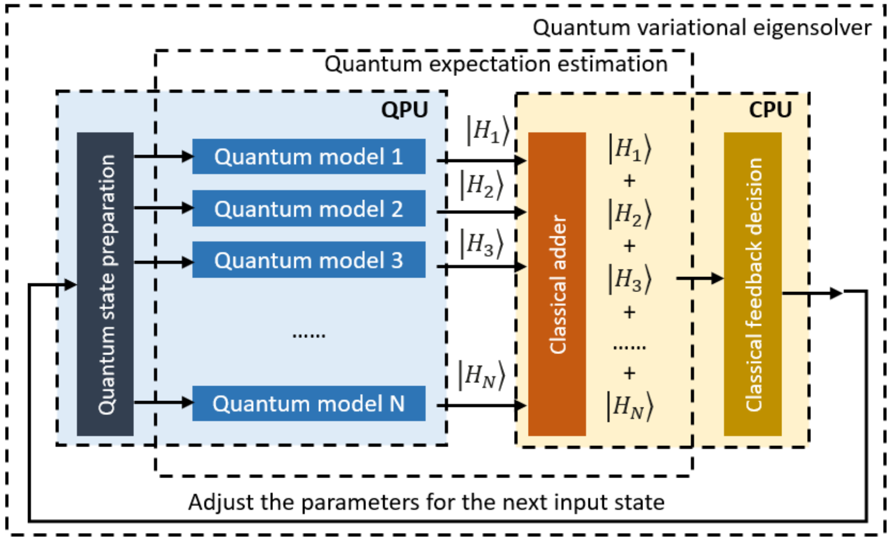
VQE for Quantum Chemistry

Challenge

Quantum chemistry aims to apply quantum mechanics to chemical systems, such as calculating the ground state energy of molecules by calculating the numerical solution of the Schrodinger equation. Quantum chemistry has become **an important means to study the physical and chemical properties** of materials.

The exact solution of the Schrodinger equation has **exponential complexity**, and **the scale of the chemical system that can be simulated is severely limited**. However, only polynomial time is required for quantum computing.

Quantum Chemistry VQE Algorithm Process



Quantum Chemistry Computing Method

The core problem of quantum chemistry is to solve the Schrodinger equation. In quantum chemistry simulation, the variational method is usually used, that is, a trial wavefunction containing parameters is constructed, and the wavefunction with the **lowest expected energy is found by continuously optimizing the parameters**.

Variational Quantum Eigensolver (VQE): Because a wavefunction needs to be constructed, quantum computers based on quantum state evolution have natural advantages and can efficiently evolve to obtain the trial wavefunction. After the output (the expected energy value of the wavefunction) is obtained by using the quantum computer, the variational parameters in the quantum circuit may be updated by using the classical optimizer, and iterations are performed repeatedly until the ground state energy and the ground state wavefunction are found.

Quantum Approximate Optimization Algorithm(1)

Quantum Approximate Optimization Algorithm

Quantum Approximation Optimization Algorithm (QAOA) is a quantum algorithm used to solve combinatorial optimization problems. For a given NP-Hard problem, QAOA can find a good approximation solution in **polynomial time**. In addition, QAOA has a better approximation rate than any known classical polynomial-time algorithm and can be used in various scenarios such as **transportation, logistics, and finance**.

Max-Cut Problem Quantization

1. Each vertex in the graph is assigned a qubit. When the vertex is distributed to the left, the vertex is set to the $|0\rangle$ state. When the vertex is distributed to the right, the vertex is set to the $|1\rangle$ state.

2. A proper Hamiltonian quantity is selected, so that when connected vertices are in the same quantum state (on the same side), the expected value of the Hamiltonian quantity is 0, and when connected vertices are in different quantum states (on different sides), the expected value of the Hamiltonian quantity is –1.

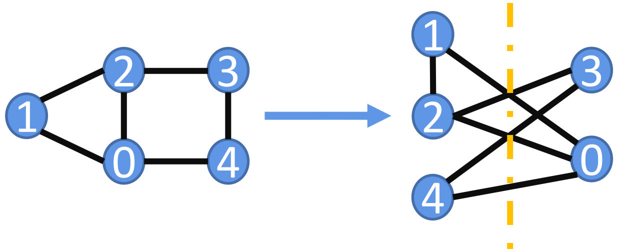
3. An approximate optimal solution can be obtained by minimizing the expected value of the Hamiltonian quantity.

Overall, we turn the max-cut problem into **finding the ground state of the Hamiltonian quantity**. So, all we need to do is setting up a proper ansatz circuit based on the Hamiltonian quantity and continuously optimize the circuit parameters until the optimal solution is found. The figure on the right shows the detailed process.

Quantum Approximate Optimization Algorithm(2)

Max-Cut Problem

The max-cut problem is an NP-complete problem in the graph theory. It needs to divide vertices of a graph into two parts and make the most edges be cut, as shown in the figure on the right.



Max-Cut Problem Solving Process

Set up a parameterized circuit.

Initialize parameters.

Execute the circuit to obtain the quantum state $|\psi\rangle$.

Calculate $\langle\psi|H_C|\psi\rangle$.

Does the loss function converge?

No

Optimize parameters.

Yes

Obtain the optimal approximation solution through calculation.

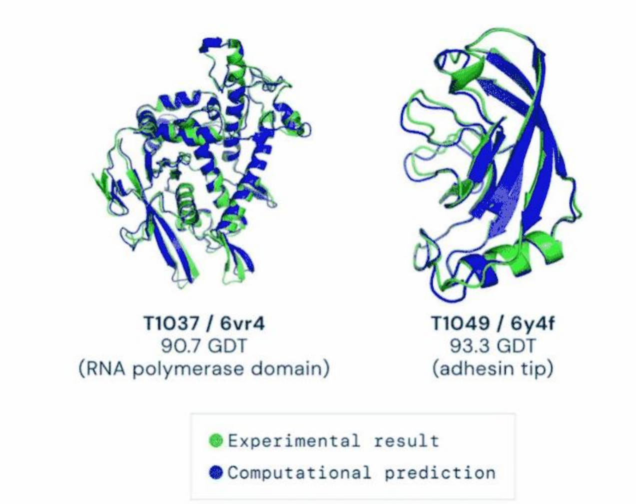
HUAWEI

Contents

Classical Machine Learning

Classical machine learning includes supervised learning, unsupervised learning, and reinforcement learning, and is widely used in various fields such as classification, image recognition, and behavior learning.

In 2017, AlphaGo trained in deep reinforcement learning defeated Ke Jie, who is the world's number one Go player.

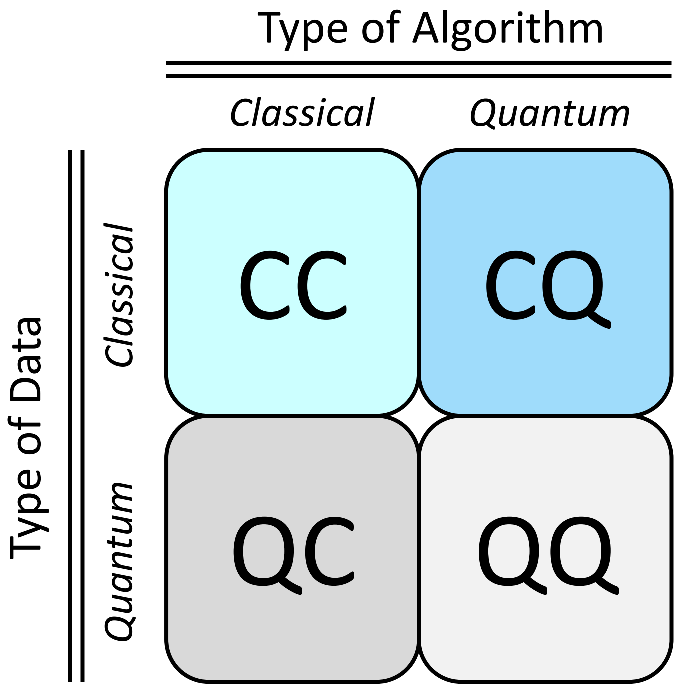


In 2020, the accuracy of protein structure prediction using AlphaFlod 2 reaches 1.6 Å.

- Bottlenecks faced by classical machine learning: As data grows, the volumes overload classical computers.
- The unique **superposed state** and **state entanglement** properties of qubits can implement **exponential parallelism** to reduce computing complexity and improve data processing capability.
- For a 28 x 28 MNIST dataset, only **10 qubits** are required for encoding, outcompeting **classical 784 bits**.

Quantum Machine Learning

Machine learning can be classified into the following four types by **data and algorithm**:



- **CC: classical data + classical algorithm (traditional machine learning)**
- **CQ (AI4Quantum): classical data + quantum algorithm, that is, classical machine learning is applied to the quantum field.** Quantum data, such as quantum states and Hamiltonian quantities, can be represented by classical neural networks and tasks can be completed by training the parameters of the classical neural networks.
- **QC (Quantum4AI): quantum data + classical algorithm** , which is classified into two types:
  - **A quantum version of traditional machine learning algorithms**, such as quantum principal component analysis (QPCA), quantum support vector machine (QSVM), and quantum reinforcement learning. These algorithms can obtain quantum acceleration from corresponding quantum algorithms, but are not suitable for near-term quantum devices.
  - **QNN:** Parameterized quantum circuits replace neural networks. For example, quantum convolutional neural networks (QCNNs) are used for classification, and quantum generative adversarial networks (QGANs) are used for generation.

QQ: quantum data + quantum algorithm, that is, fully quantum machine learning.

Application scenario: Quantum data is unknown, and the data can be regenerated in other quantum systems through quantum machine learning.

Advantages of Quantum Machine Learning

- By using properties such as quantum state superposition and entanglement in quantum computing, a plurality of qubits can represent more complex states or implement more complex operations, thereby implementing quantum acceleration.
- Quantum systems are more suitable for linear algebra operations (distance estimation of high-dimensional vectors).
- Quantum algorithms with polynomial or exponential acceleration: HHL, quantum PCA (qPCA), quantum-enhanced SVM (QSVM), Grover search, etc.
- The NISQ phase requires a QC hybrid architecture that uses classical machine learning to control parameters in quantum circuits.

Algorithm	Complexity
Bayesian inference	$O(\sqrt{N})$
Online perceptron	$O(\sqrt{N})$
Least-squares fitting	$O(\log N)$
Classical Boltzmann machine	$O(\sqrt{N})$
Quantum Boltzmann machine	$O(\log N)$
Quantum PCA	$O(\log N)$
Quantum SVM	$O(\log N)$
Quantum reinforcement learning	$O(\sqrt{N})$

Iris Classification Algorithm

Challenge

The iris dataset is widely used in classical machine learning. By using the quantum neural network, a classifier can be trained to solve the classification problem of the dataset.

The dataset consists of 150 samples from three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Each sample contains four characteristics: calyx length, calyx width, petal length, and petal width.

Solution Process of Classification Algorithm Based on the Quantum Neural Network

Preprocessing	Divide the dataset into a **training set and a test set** and shuffle the dataset to increase randomness.
Encoding	**Encode data to quantum states** using a parameterized quantum circuit (encoding circuit).
Ansatz setup	According to the problem type and characteristics, select a proper parameterized quantum circuit as ansatz.
Hamiltonian quantity	Obtain the Hamiltonian quantity from the quantum state evolved from the quantum circuits (encoding and ansatz circuits) that are set up in the previous steps and calculate the **loss function and the parameter gradient**.
Training	• Evolve qubits through the quantum circuits and obtain the expected value of Hamiltonian quantity as the output. • Calculate the loss function and parameter gradient, and use the classical optimizer to update the parameters of the ansatz circuit to complete an iteration. **Repeat iterations until the loss function converges.**
Testing	Apply the trained model to the test set to **test the accuracy of the model**.

Contents

HiQ Quantum Computing Full-Stack Solution

Quantum applications	AI; Biological computing; Material simulation; Logistics and transportation
Quantum computing cloud platform	**HiQ Composer**; **Jupyter Notebook (ModelArts)**; **CloudIDE**
Quantum computing framework	**MindSpore Quantum**: **Variational algorithm library**, **General algorithm library**, Quantum Simulator; **Quantum Computing Simulation Acceleration Engine**; **MindSpore deep learning framework**
Quantum operating system	**Euler Quantum**: **Task scheduling**, **Virtual QPU**, **Pulse optimization**, **Autocalibration**
Quantum hardware	Quantum Electronics Design Automation (QEDA); Classical computers: **CPU**, **GPU**, **Ascend**; Quantum computers: **Superconducting**, **Ion-trapped**

Open

Rich APIs, facilitating development and integration and enabling industry applications

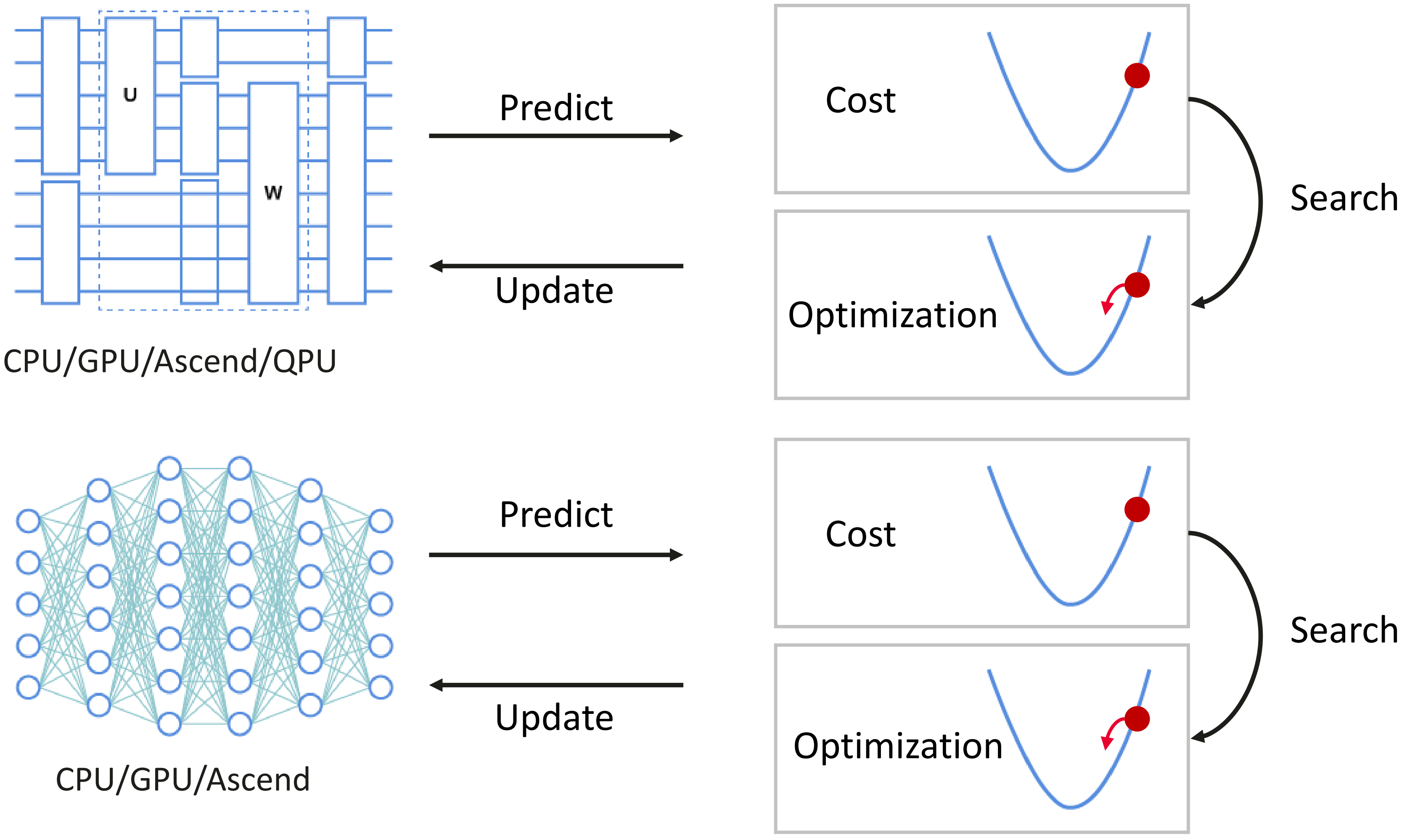
HPC framework

Integrated with AI development and DL platforms to achieve industry-leading performance

Full-stack

Integrated software and hardware solution

MindSpore Quantum



MindSpore Quantum

Quantum computing framework

MindSpore

Deep learning framework

Evolution of MindSpore Quantum: Pursuing Excellence and Continuous Innovation

HDC 2021 | **HC 2021** | **APTQS 2022** | **Quantum Physics and Logic 2022** | **CPS International Young Scientists Forum on Quantum Computing**

2021.03 — V0.1
Official open source
- Provides quantum circuit simulation capabilities.
- Provides variational quantum algorithm simulation capabilities, including VQE, QAOA, and QNN.

2021.10 — V0.3
- Provides various ansatz circuits.
- Provides multiple quantum neural network operators.
- Provides unique quantum circuit operators.

2022.01 — V0.5
- Completed framework refactoring.
- Simplified the quantum machine learning process.

2022.04 — V0.6
- Provides the quantum noise simulation capability, supporting multiple channels.
- Provides SVG drawing of quantum circuits and sampling results.

2022.07 — V0.7
- **Optimizes the construction process of quantum neural networks.**
- **Releases more simulator interfaces.**
- **Implements efficient quantum Fisher information computing.**
- **Provides the dynamic display capability of the Bloch sphere.**

2022.12 — V0.8
Coming soon
- New CPU/GPU quantum circuit simulator

MindSpore Quantum: Making Quantum Computing Reachable

Superb experience

- Rich **variational circuit algorithm libraries**, enabling frontiers
- Efficient and convenient **template-based** quantum neural networks
- In-depth integration with **MindSpore**
- **Various hardware platforms**
- **Visualized circuit rendering**
- Non-Hermitian operations such as **gradient calculation of expected values**

Ultimate performance

- Industry-leading **variational quantum computing performance**
- Industry-leading **28-qubit** quantum chemical simulation and **32-qubit** acceleration (QuPack)
- **High-performance simulator**, supporting analog computing of 30+ qubits.
- Up to 10x and 2x increase in GPU and CPU performance (compared with version 0.7)

Simplified development

- **Pre-integrated with HiQ cloud services**, installation-free, and out-of-the-box
- **One-click PIP installation**, supporting various operating systems
- Built-in quantum machine learning, chemical simulation, and combinatorial optimization modules, providing **rich tutorials**
- Rich **programming interfaces** and development instances, simple and easy to use

Superb Experience of MindSpore Quantum

Rich variational circuit algorithm libraries

Hardware efficient

max-cut

max-2-sat

uccsd/uccsd0

IQPencoding

In-depth integration with MindSpore

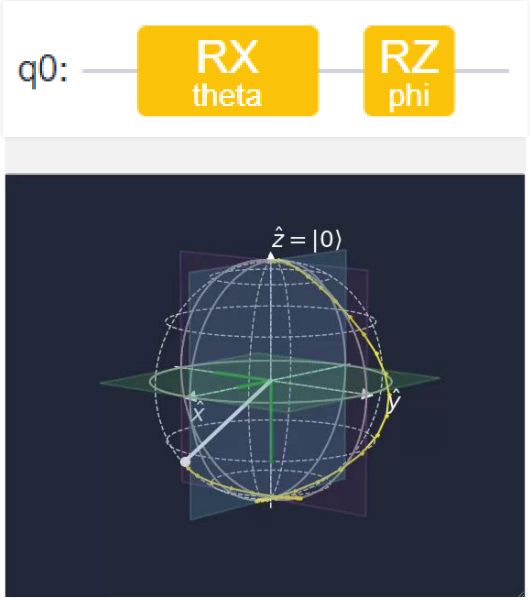
Auto differentiation

Quantum neural network training & inference

Various hardware platforms

CPU/GPU/Ascend/QPU

Multi-format visualized circuit rendering

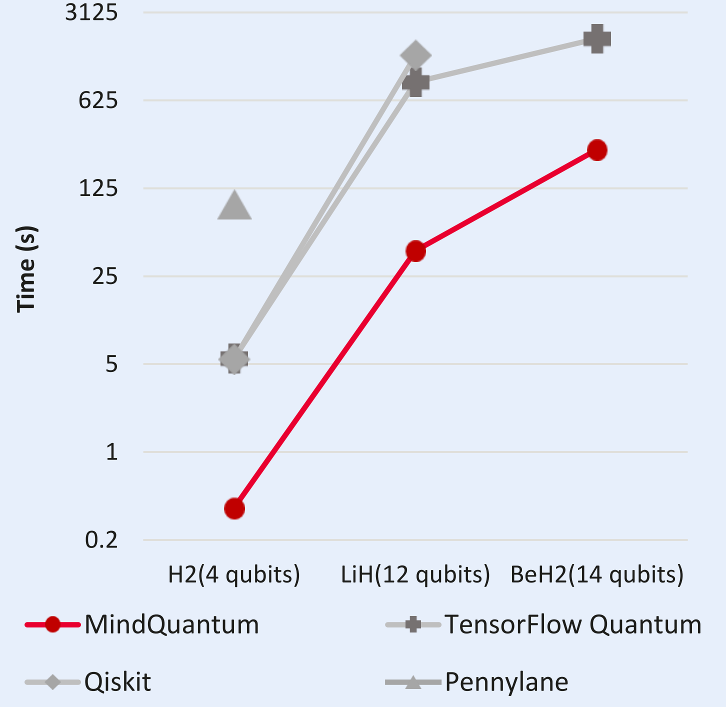


Based on the high-performance in-depth AI development platform, MindSpore Quantum provides a rich algorithm library, enabling developers to easily develop quantum software and algorithms based on the framework.

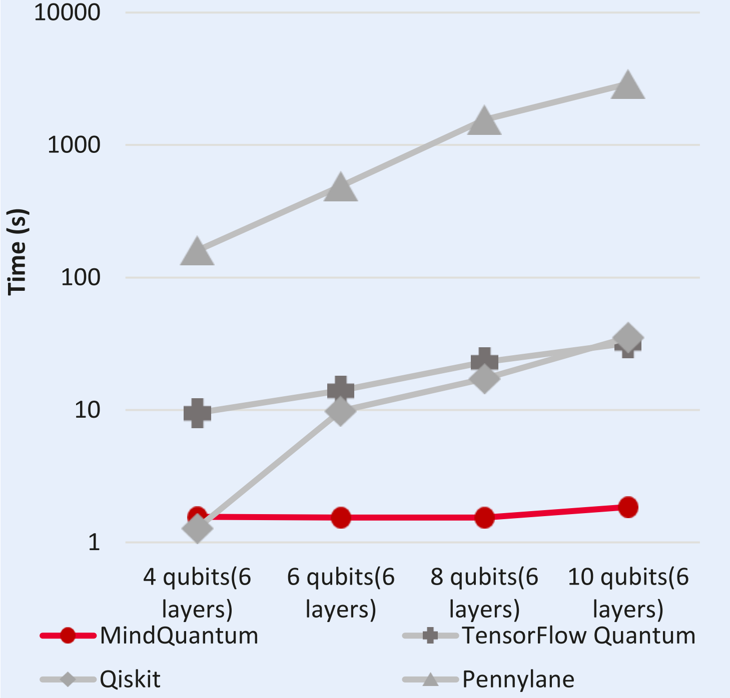
HUAWEI

Ultimate Performance of MindSpore Quantum: Industry-leading Variational Quantum Algorithms

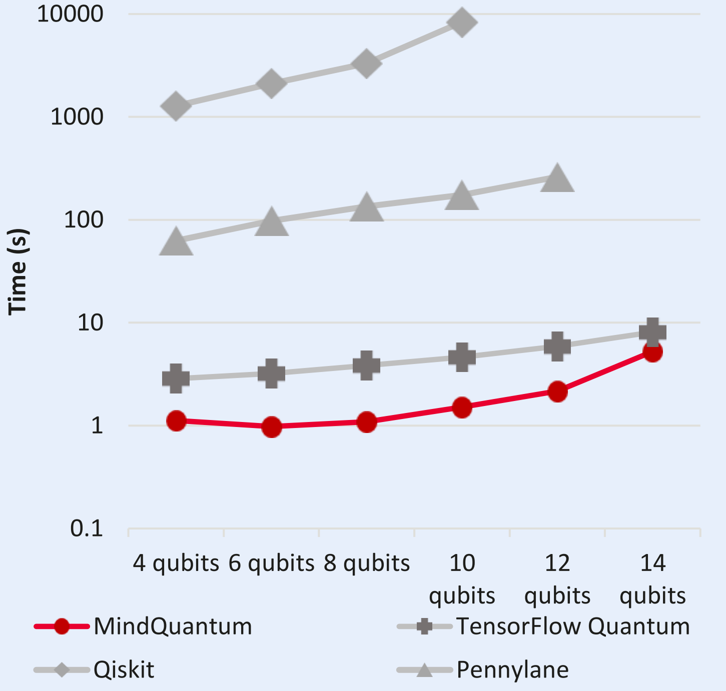
VQE performance improved 3+ times



QAOA performance improved 5+ times



QNN performance improved 4+ times



Molecular simulation | **Quantum material simulation** | **Research on new materials**

Max-Cut | **TSP** | **Logistics and transportation problems**

NLP | **Classification problems** | **GAN**

Beta version

MindQuantum 0.5.0 (2022.02.28)
qiskit-terra 0.19.2 (2022.02.09)

TensorFlow Quantum 0.6.1 (2022.02.04)
Pennylane 0.21.0 (2022.02.08)

Simplified Development on MindSpore Quantum

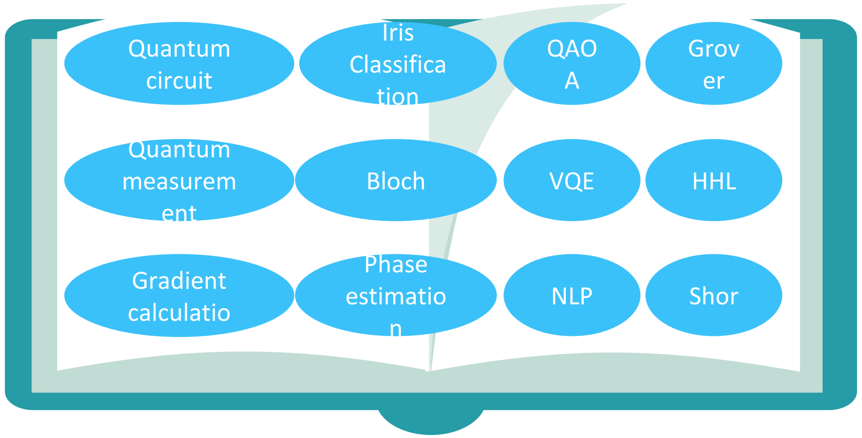
Pre-integration, installation-free

One-click installation, efficient and fast

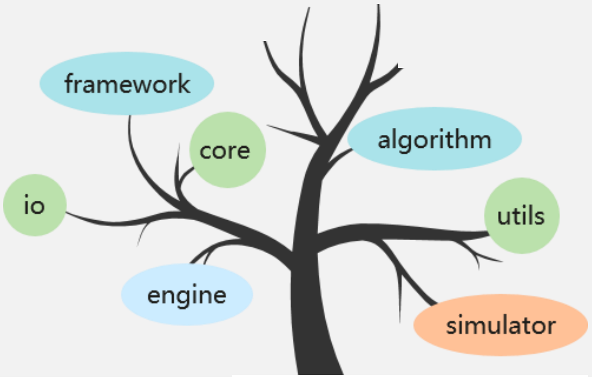
PIP installation command:
pip install mindquantum

Linux
Windows
macOS

Rich tutorials, easy to get started



Easy-to-use APIs



MindSpore Quantum Provides Rich Tutorials for Learning and Scientific Research

Step 1

Quantum computing beginner and advanced tutorials & training videos

Step 2

Tutorial for solving typical application problems of quantum computing

Step 3

Academic papers and solution verification, providing cases for quantum software research and innovation

2022 QWorld Quantum Computing and Programming Course

https://hiq.huaweicloud.com/tutorial

MindSpore Quantum Tutorial and Developer Guide

https://hiq.huaweicloud.com/tutorial

Download Notebook

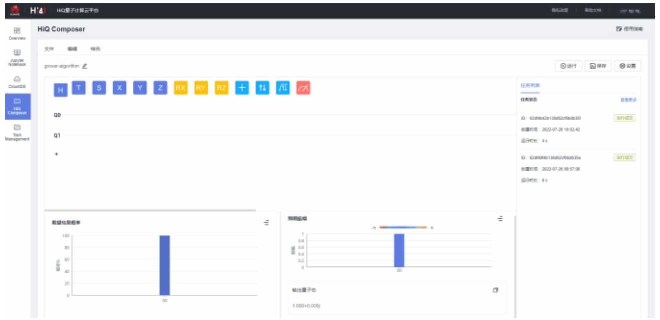
Download Sample

View source on Gitee

MindSpore Quantum papers, open source code

https://hiq.huaweicloud.com/consult/paper

HiQ Quantum Computing Cloud Platform Provides Various Frontend and Backend Services

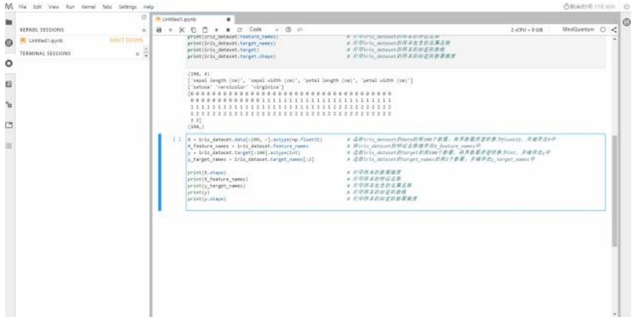


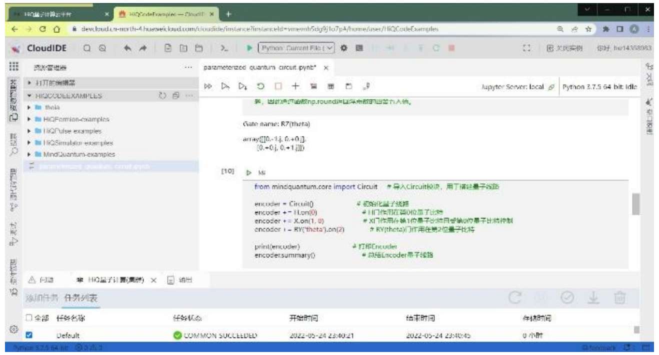
HiQ Composer
GUI tool
(For beginners)

Drag-and-drop quantum circuit setup, easy to use.

Jupyter Notebook
Interactive IDE
(Teaching)

Interactive programming IDE to facilitate quantum algorithm development.



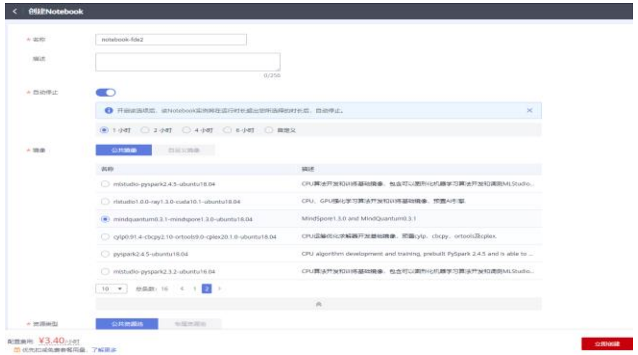


CloudIDE
Cloud Native IDE
(Lightweight development)

A professional programming interface similar to VS Code and highly integrated with tools such as Git

ModelArts
AI development platform
(High-performance computing)

Paid services with more powerful computing resources to meet the requirements of large-scale algorithm research



Contents

Creating a Quantum Programming Development Environment

Step 1

Log in to the HiQ platform and create an instance.

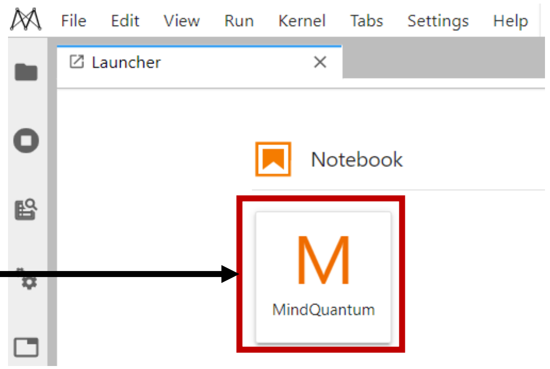
https://hiq.huaweicloud.com/portal/home

Step 2

Start the instance.

Step 3

Create a Notebook file.



Name	Status	Image	Flavor	Description	Created At		Operation
notebook-4fda	Creating	mindquantum0.6.0-mindspore1.7.0-...	CPU: 2vCPUs 8GB	--	Dec 15, 2022 09:46:02 GMT+08:00		Open | Start | Stop | More

Practice 1: Setting Up a Quantum Circuit

1. Import the dependency package.

```python
import numpy as np
# import quantum gate
from mindquantum.core.gates import X, H, RY
# import quantum circuit
from mindquantum.core.circuit import Circuit
```

2. Create a quantum circuit and add a quantum gate.

```python
circ = Circuit()          # Create a quantum circuit.
circ += RY('a').on(0) # Add a RY gate to bit 0 of the circuit.
circ += X.on(1, 0)     # Add an H gate to bit 1 of the circuit.
```

3. Print the quantum circuit.

```python
print(circ)
```

```
q0: ──RY(a)────●──
               │
q1: ───────────X──
```


HUAWEI

Practice 2: Variational Quantum Computing

4. Create the Hamiltonian quantity.

```
ops = QubitOperator('X0 X1') # Create the Hamiltonian quantity wrapper.
ham = Hamiltonian(ops)              #Create the Hamiltonian quantity.
print(ham)            # Print the Hamiltonian quantity.
```

```
1 [X0 X1]
```

5. Calculate the expected value and gradient.

```
sim = Simulator('projectq', 2)     # Create a 2-qubit simulator.
# Obtain a function that returns the forward value and the gradient of the circuit parameter.
grad_ops = sim.get_expectation_with_grad(ham, circ)
grad_ops(np.array([1.0]))       # Calculate the gradient and expected value.
```

(array([[0.84147098+0.j]]),	Expected value
array([[[0.54030231+0.j]]]))	Gradient

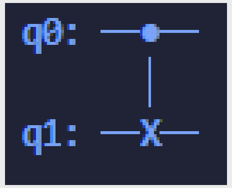
Quiz

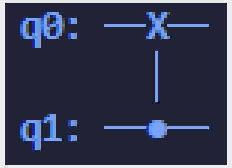
1. (True or false) The Shor algorithm can efficiently perform integer factorization, which is much faster than the classical algorithm.

   A. True

   B. False

2. (Single-choice) Which of the following quantum circuits corresponds to MindSpore Quantum X.on (0,1)?

   A. 

   B. 

Recommendations

- Huawei official websites:
  - HiQ quantum computing: https://hiq.huaweicloud.com/home
  - MindSpore: https://mindspore.cn/
- Quantum software
  - MindQuantum: https://gitee.com/mindspore/mindquantum
  - HiQ quantum computing cloud platform: https://hiq.huaweicloud.com/portal/home

HiQ official website

MindSpore official website

MindQuantum community

HiQ Platform

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

HUAWEI