

Analysis of learnability of a novel hybrid quantum-classical convolutional neural network in image classification

Tao Cheng(程涛), Run-Sheng Zhao(赵润盛), Shuang Wang(王爽), Rui Wang(王睿), and Hong-Yang Ma(马鸿洋)

Citation: Chin. Phys. B, 2024, 33 (**4**): 040303. DOI: 10.1088/1674-1056/ad1926
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Gray code based gradient-free optimization algorithm for parameterized quantum circuit

Anqi Zhang(张安琪), Chunhui Wu(武春辉), and Shengmei Zhao(赵生妹)
Chin. Phys. B, 2024, 33 (**2**): 020311. DOI: 10.1088/1674-1056/ad09cd

Variational quantum semi-supervised classifier based on label propagation

Yan-Yan Hou(侯艳艳), Jian Li(李剑), Xiu-Bo Chen(陈秀波), and Chong-Qiang Ye(叶崇强)
Chin. Phys. B, 2023, 32 (**7**): 070309. DOI: 10.1088/1674-1056/acb9fb

A new method of constructing adversarial examples for quantum variational circuits

Jinge Yan(颜金歌), Lili Yan(闫丽丽), and Shibin Zhang(张仕斌)
Chin. Phys. B, 2023, 32 (**7**): 070304. DOI: 10.1088/1674-1056/ac9b32

Quantum algorithm for neighborhood preserving embedding

Shi-Jie Pan(潘世杰), Lin-Chun Wan(万林春), Hai-Ling Liu(刘海玲), Yu-Sen Wu(吴宇森), Su-Juan Qin(秦素娟), Qiao-Yan Wen(温巧燕), and Fei Gao(高飞)
Chin. Phys. B, 2022, 31 (**6**): 060304. DOI: 10.1088/1674-1056/ac523a

Quantum partial least squares regression algorithm for multiple correlation problem

Yan-Yan Hou(侯艳艳), Jian Li(李剑), Xiu-Bo Chen(陈秀波), and Yuan Tian(田源)
Chin. Phys. B, 2022, 31 (**3**): 030304. DOI: 10.1088/1674-1056/ac1b84

Analysis of learnability of a novel hybrid quantum–classical convolutional neural network in image classification

Tao Cheng(程涛)[2], Run-Sheng Zhao(赵润盛)[1], Shuang Wang(王爽)[2], Rui Wang(王睿)[1], and Hong-Yang Ma(马鸿洋)[1,†]

[1] *School of Sciences, Qingdao University of Technology, Qingdao* 266033, *China*
[2] *School of Information and Control Engineering, Qingdao University of Technology, Qingdao* 266033, *China*

(Received 3 November 2023; revised manuscript received 19 December 2023; accepted manuscript online 28 December 2023)

We design a new hybrid quantum–classical convolutional neural network (HQCCNN) model based on parameter quantum circuits. In this model, we use parameterized quantum circuits (PQCs) to redesign the convolutional layer in classical convolutional neural networks, forming a new quantum convolutional layer to achieve unitary transformation of quantum states, enabling the model to more accurately extract hidden information from images. At the same time, we combine the classical fully connected layer with PQCs to form a new hybrid quantum–classical fully connected layer to further improve the accuracy of classification. Finally, we use the MNIST dataset to test the potential of the HQCCNN. The results indicate that the HQCCNN has good performance in solving classification problems. In binary classification tasks, the classification accuracy of numbers 5 and 7 is as high as 99.71%. In multivariate classification, the accuracy rate also reaches 98.51%. Finally, we compare the performance of the HQCCNN with other models and find that the HQCCNN has better classification performance and convergence speed.

Keywords: parameterized quantum circuits, quantum machine learning, hybrid quantum–classical convolutional neural network

PACS: 03.67.Ac, 03.67.Lx, 03.67.–a **DOI:** 10.1088/1674-1056/ad1926

1. Introduction

In recent years, machine learning has achieved significant results in various fields. The rise of machine learning technology has taken a greater step forward in human exploration of the field of artificial intelligence. This technology has numerous applications and achieved significant achievements in various fields such as computer vision, data mining, and biomedicine.[1–4] In the 20th century, with the introduction of quantum-mechanical theory, researchers proposed a more powerful quantum computation based on this theory,[5,6] which utilizes the superposition and entanglement of quantum states for computation. Its powerful parallelism provides space for the acceleration of classical computing.[7] With the development of quantum technology, we use noisy intermediate-scale quantum (NISQ) computers to perform useful computational tasks and to solve the complex and difficult problems of classical computers.[8,9] Then, researchers combined quantum computing with machine learning and proposed a new concept, i.e., quantum machine learning (QML).[10] Nowadays, QML has seen a considerable amount of theory and practice, and has become the core of quantum computing and a world application for solving problems that classical computing cannot achieve.

Quantum computers can generate more complex data distributions and naturally learn patterns in certain data distributions. QML utilizes quantum computers to achieve computational acceleration or better performance for more complex machine learning tasks.[11–15] Parameterized quantum circuits (PQCs), as a quantum computing model, have powerful parallel computing and expression capabilities, providing a broad and promising solution for quantum machine learning in the NISQ era.[16–18] PQCs perform specific tasks by adjusting the parameters of quantum circuits, which are now widely used in various fields. They have been proven in chemical simulation, combinatorial optimization, generative modeling, and image classification.[19–22] Nowadays, researchers have constructed many quantum machine learning algorithms based on PQCs. In 2014, Peruzzo *et al.*[23] proposed a variational quantum eigensolver (VQE) and implemented it using photon quantum processors and classical computers. In 2019, Parrish *et al.*[24] developed an extension of the variational quantum eigensolver (VQE) algorithm to form a multistate contracted VQE (MC-VQE), which can effectively calculate the transition energies between the molecular ground state and several low-lying states as the oscillator strength associated with these transitions. In 2021, Xin *et al.*[25] proposed a new qPCA algorithm using hybrid classical quantum control, in which parameterized quantum circuits are optimized through simple observable measurements, significantly reducing experimental complexity. In addition, researchers combined PQCs with classical machine learning to form hybrid quantum–classical machine learning.[26–29] Nowadays, with the introduction of multiple quantum machine learning algorithms based on PQCs, the su-

†Corresponding author. E-mail: hongyang_ma@aliyun.com

© 2024 Chinese Physical Society and IOP Publishing Ltd

periority and better performance of quantum machine learning algorithms have been strongly demonstrated.[30]

A convolutional neural network (CNN) is one of the most important networks in the field of deep learning. Due to its parameter sharing and stronger flexibility, CNNs have been widely used in image recognition, video analysis, natural language processing, and other fields.[31–34] Afterwards, with the development of quantum information, researchers combined quantum computing with CNNs to form quantum convolutional neural networks (QCNNs). In 2019, inspired by a classical CNN, Cong *et al.*[35] proposed a QCNN for phase code classification and quantum error correction code optimization, proving that the QCNN can be used to design quantum error correction schemes optimized for a given unknown error model. In 2022, Herrmann *et al.*[36] implemented a QCNN on a 7-qubit superconducting quantum processor to identify the symmetric-protected topology (SPT) phase of a spin model characterized by non-zero sequence parameters. In 2023, Chen *et al.*[37] proposed two-scale inspired local feature extraction methods based on a QCNN in the Tensorflow quantum framework for binary image classification, demonstrating that the QCNN outperforms a classical CNN in image classification accuracy.

We propose a new hybrid quantum–classical neural network model by combining PQCs with traditional neural networks. This model integrates the powerful performance of PQCs and retains the characteristics of classical neural networks. We have redesigned the classic convolutional layer using PQCs to form a new quantum convolutional layer. Moreover, we integrate the classical fully connected layer with PQCs to form a new hybrid quantum classical fully connected layer, which consists of two classical layers and one quantum layer. The classical image is transformed into quantum states as input to the quantum convolutional layer, which performs a series of unitary transformations on the quantum state. The classic pooling layer uses the maximum pooling method to perform pooling operations. The output classical data as the input of the first classical layer of the hybrid quantum–classical connected layer. Afterward, through the evolution of the quantum layer, the classification is ultimately carried out by the second classical layer. In this paper, we refer to this model as the hybrid quantum–classical convolutional neural network (HQCCNN). Simulation experiments show that the HQCCNN has stronger learning performance and higher image classification accuracy.

The contribution of this paper are as follows: In Section 2, we briefly introduce a CNN and PQC model. We provide a detailed introduction to the newly designed model architecture for the HQCCNN in Section 3. In Section 4, we describe the simulation experiment and its results. Finally, the conclusion is drawn in Section 5.

2. Related work

2.1. Convolutional neural network

A convolutional neural network (CNN) is a feedforward neural network.[38] It is mainly used to process data with grid structure, such as images (2D grid pixels) and time series data (1D grid time points).[39,40] Its structure mainly consists of convolutional layers, pooling layers, and fully connected layers. Figure 1 shows the basic structure of a convolutional neural network.

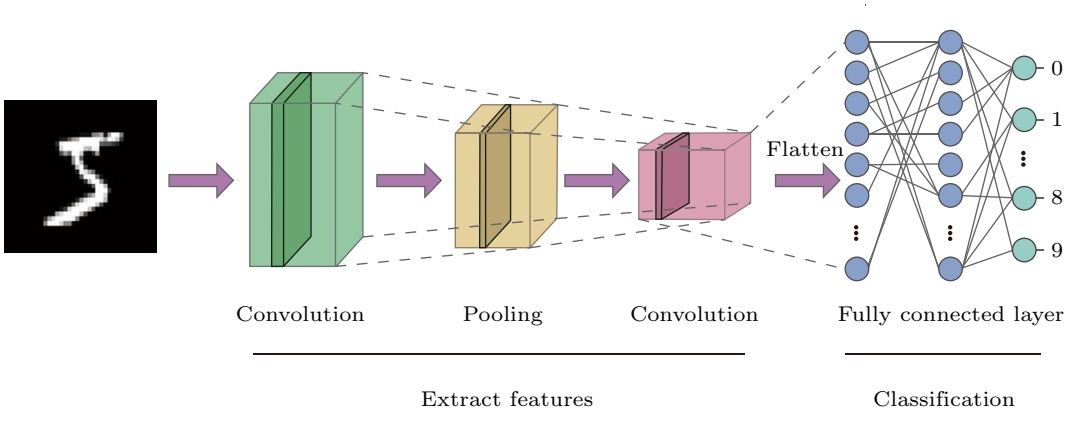


Fig. 1. The architecture of a convolutional neural network.

The filter maps a portion of the input image to a single neuron output through feature mapping and is parameterized by an array P with the same size as the input region. The specific process is as follows:

$$A_{1:m,1:n} \longrightarrow \sum_{1\le i\le m, 1\le j\le n} A_{i,j}P_{i,j}. \tag{1}$$

Suppose that a two-dimensional array A with the size of $v \times h$ is taken as the input of the filter, and the size of the first selected part area is $m \times n$, the corresponding position of the area is the upper left corner of A, and then the linear function is used to map the part of the area. The mapping process is as follows:

The next area slides to the right in steps of s until it

reaches the right edge of A. Afterwards, repeat the process until the entire image is traversed. The final output will be a two-dimensional array of size $\frac{v-m+1}{s} \times \frac{h-n+1}{s}$. If the size of the input data is a three-dimensional array of $v \times h \times d$, the final output result of the array obtained through the convolution operation of k filters with size $m \times n$ is $\frac{v-m+1}{s} \times \frac{h-n+1}{s} \times (dk)$.

After the convolution layer is the pooling layer. The main purpose is to reduce the dimension of the characteristic map output by the convolution layer, so as to reduce the calculation amount of the model and prevent overfitting. Finally, the fully connected layer is used to connect all neurons, and the extracted image features are nonlinearly combined to obtain the output.

2.2. Parameterized quantum circuits

Parameterized quantum circuits (PQCs) are a special type of quantum circuits, which is composed of parametric unitary gates and nonparametric unitary gates.[41] In a typical quantum circuit, quantum bits (qubits) operate through a series of quantum gates, and each quantum gate can perform a specific linear transformation, for example, Pauli gates (X, Y, Z), Hadamard gates, and CNOT gates. However, in parameterized quantum circuits, some or all of these quantum gates are replaced by parameterized single quantum bit gates, such as R_y gates or R_z gates. They can be passed through parameters θ to adjust the rotation angle. Common quantum gates are defined as follows:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{2}$$

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{3}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix},$$
$$R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \tag{4}$$

According to the quantum circuit $U(w)$, the input quantum state $|x\rangle$ obtains the output quantum state $|y\rangle$ through a series of unitary transformation in the parametric circuit, that is, $|y\rangle = U(w)|x\rangle$, where w is the parameter of the quantum circuit. PQCs can combine different quantum gates to design different quantum algorithms.

3. Hybrid quantum–classical convolutional neural network

3.1. Quantum convolutional layer

The quantum convolutional layer is composed of multiple quantum filters with PQCs, which preserves the weight sharing characteristics of classical convolutional layers.[42] This layer utilizes quantum filters to perform feature mapping and extraction on all qubits in the local space of the input image.[43] Quantum filters convert classical image data into quantum states and utilize the properties of quantum parallelism and entanglement to enable quantum convolutional layers to achieve more accurate recognition and extraction of image features on low-dimensional images. In addition, due to the nonlinearity of quantum computing, quantum filters can enhance the model's representation ability, enabling it to capture more complex data patterns. We use different types of quantum gates to design PQCs in quantum filters, which consist of single-qubit layers and double-qubit layers. The structure of the designed quantum filter is shown in Fig. 2. A single-qubit layer consists of R_y gates, and multiple CNOT gates form double-qubit layers.

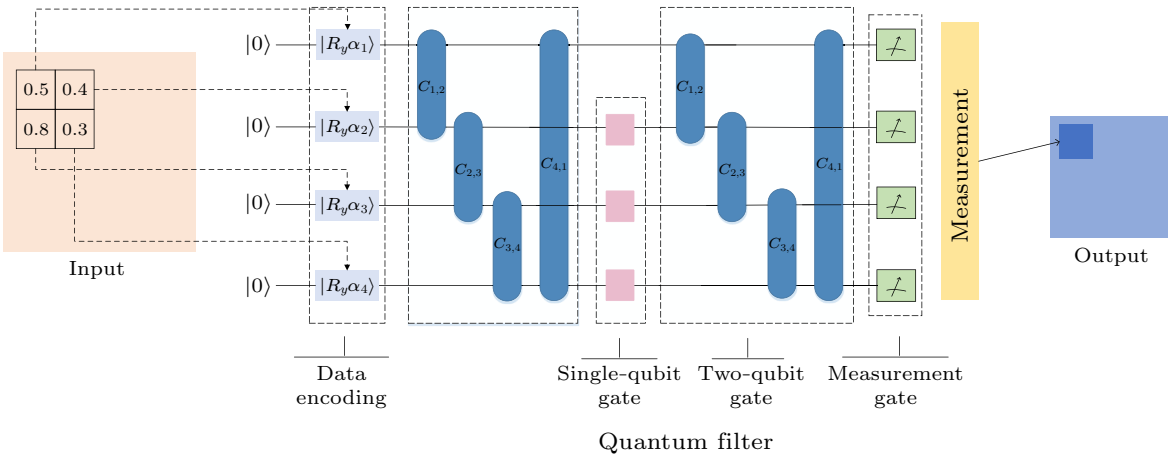


Fig. 2. The designed quantum filter architecture. It includes single-qubit layers and two-quantum layers. A single qubit layer consists of multiple R_y gates, each containing an adjustable parameter. Multiple CNOT gates form a double qubit layer, which act on two adjacent qubits. Here $\mathrm{CNOT}_{i,j}$ represents the control of the ith and jth qubits through double-qubit gates.

In quantum convolutional layers, due to the limitation of the number of qubits, we first preprocess the classical input image to reduce its dimensionality to $m \times m$ and normalize the pixel values to within the range of $[0, 1]$. Then, obtain the feature vector x, which is converted into the corresponding angle information α, and perform quantum state encoding on the preprocessed classical data. In this process, we use angle encoding to encode the angle information α corresponding to the vector as the rotation angle of single-qubit gate R_y to perform quantum state encoding on the preprocessed images, as expressed by the formula

$$|\psi_{\text{image}}\rangle = \otimes_{i=1}^{m^2} R_y(\alpha_i)|0\rangle_1|0\rangle_2 \cdots |0\rangle_{(m-1)^2}|0\rangle_{m^2}. \tag{5}$$

Afterwards, the obtained quantum state is subjected to unitary transformation and use the Pauli-Z gates to measure the expectation value. Finally, we obtain a classical output E with a dimension of $1 \times m^2$, as expressed by the formula

$$E = \langle \psi_{\text{image}}|U^{\dagger}(\theta)(Z_1, \ldots, Z_N)U(\theta)|\psi_{\text{image}}\rangle, \tag{6}$$

where $(Z_1, \ldots, Z_N)$ is the Z operator acting on different qubits.

The mapping process in the quantum convolution layer is nonlinear, so no additional nonlinear function is required to clarify. The required number of qubits is determined by the defined window size. Here, we use 4 qubits to preserve convolutional features as many as possible, and extract hidden information from quantum states. In the problem of gradient optimization, we adopt the quantum natural gradient optimization algorithm (QNGOA),[44] which can make the direction of gradient descent not target any parameter and has stronger generalization ability while accelerating the convergence speed of the network model.

3.2. Max-pooling layer

This layer adopts the method of maximum pooling,[45] with the aim of further reducing the dimensionality of the image information extracted by quantum convolutional layers to reduce computational costs, enhance the feature invariance and robustness of the image, and improve the performance of the training model. This layer divides the input image into multiple small regions and selects the maximum value within the divided small regions to extract the strongest significant features of the image. Furthermore, the extracted maximum values are merged to form a new feature map and converted into a one-dimensional array as input for subsequent sections. The detailed process of maximum pooling is shown in Fig. 3.

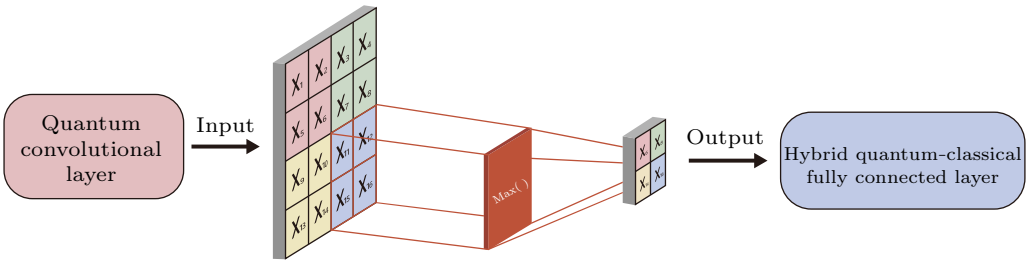


Fig. 3. Detailed process of maximum pooling.

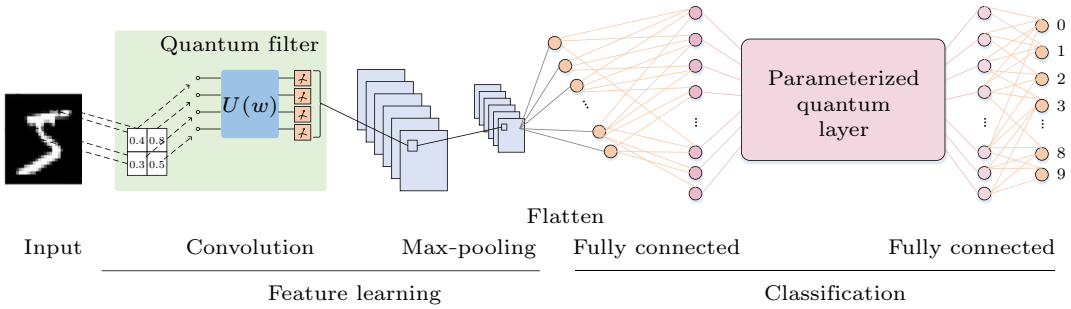


Fig. 4. Architecture of the hybrid quantum–classical convolutional neural network (HQCCNN). This layer is composed of two classical fully connected layers and one variational quantum layer. The input image is feature extracted and dimensionally reduced through quantum convolutional layers and maximum pooling layers, and compressed into multiple planes as input for the hybrid quantum–classical fully connected layer. The quantum layer is used for classification, and the second classical layer outputs the classification results.

3.3. Hybrid quantum–classical fully connected layer

After the max-pooling layer, there is a hybrid quantum–classical fully connected layer, which consists of two classical fully connected layers and one parameterized quantum layer. The purpose of this layer design is to ensure the quantity of input and output features. Its structure is shown in Fig. 4. This layer converts the merged feature images of the maximum pooling layer into X features as input, where the value of X is the total number of encoding parameters determined by the quantum part. Then, the features are classified through the quantum layer, and the output is used as the input of the second fully connected layer and mapped to 10 output

features to achieve the purpose of classification. After each classical layer, apply batch normalization and ReLu activation functions.[46,47]

3.4. Structure of the quantum layer

The quantum layer of the HQCCNN includes a data encoding layer, a variable layer, and a measurement layer. The designed parameterized quantum layer is shown in Fig. 5. This layer uses the classical image feature data of the previously classical layer as the input of PQCs, and encodes it into quantum states through the encoding layer. This section is jointly completed by H gates and R_y gates.

The encoding process is as follows:

The H-gate is used to convert the initialized quantum state $|0\rangle \otimes \cdots \otimes |0\rangle$ into an entangled state, as expressed by the formula

$$\begin{aligned}(H|0\rangle)^{\otimes q} &= \frac{1}{\sqrt{2^q}}(|0\rangle + |1\rangle)^{\otimes N} \\ &= \frac{1}{\sqrt{2^q}}(|0\rangle \otimes \cdots \otimes |0\rangle + \cdots + |1\rangle \otimes \cdots \otimes |1\rangle) \\ &= \frac{1}{\sqrt{2^q}} \sum_{i=0}^{2^q-1} |i\rangle. \end{aligned} \tag{7}$$

We use angle encoding to encode classical data features into quantum states.[48] Apply classical data X_q $(q = 1,2,3,\ldots)$ to the angle parameter $\theta_{q,1} = \arctan(x_q)$ to generate $2q$ rotation angles; $\theta_{i,2}$ controls the rotation of the R_y gate around the y axis, and resulting in a quantum state of $(\otimes_{i=0}^{q-1})R_y(\arctan(x_q))H|0\rangle^{\otimes q}$.

Multiple CNOT gates and single-qubit rotation gates form variable layers. The image feature information encoded by quantum states undergoes a series of unitary transformations in the variable layer. The number of qubits used in this article is 4, so the design process of variable layering is as follows:

$$\begin{aligned}&(\otimes_3^{i=0} R_y(w_i)\mathrm{CNOT}_{2,3}\mathrm{CNOT}_{2,1}\mathrm{CNOT}_{0,1}) \\ &\mathrm{CNOT}_{0,3}(\mathrm{CNOT}_{1,3}\mathrm{CNOT}_{0,2})(\mathrm{CNOT}_{3,1}\mathrm{CNOT}_{2,0}),\end{aligned} \tag{8}$$

where $\mathrm{CNOT}_{i,j}$ refers to the double-qubit gate controlled at the ith and jth qubits.

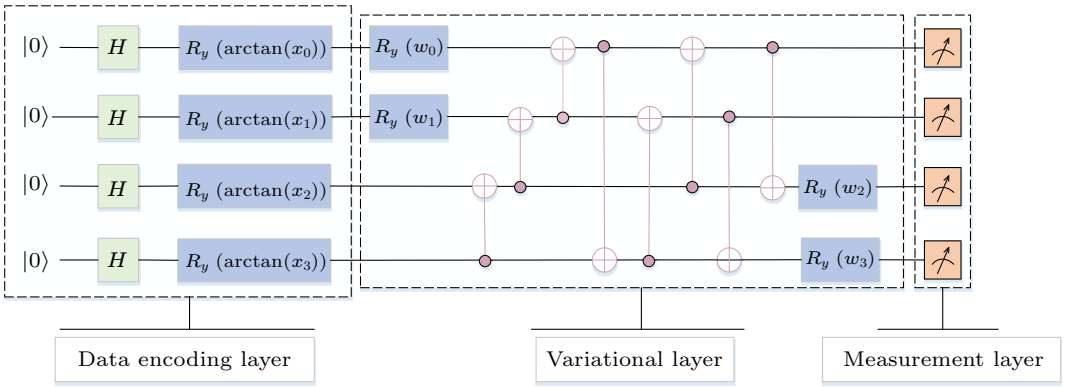


Fig. 5. The structure of parameterized quantum circuits in the quantum layer. The architecture consists of three parts: encoding layers, variable layers, and quantum measurement layers. The encoding layer is composed of H gates and R_y gates, the variable layer is composed of R_z gates, and the Pauli-Z gates form the quantum measurement layer.

The quantum measurement layer is used to output quantum states, measures each quantum state, and extracts useful information as much as possible from the quantum circuit, as expressed the formula

$$\begin{aligned}f(x;\theta) &= \langle 0|U_0^\dagger(x)U_i^\dagger(\theta_i)ZU_i(\theta_i)U_0(\theta_0)|0\rangle \\ &= \langle x|U_i^\dagger(\theta_i)ZU_i(\theta_i)|x\rangle,\end{aligned} \tag{9}$$

where $U_0(x)$ represents the conversion of x into a quantum state in the encoding layer, and $U_i(\theta_i)$ represents the rotation of a single qubit.

3.5. Quantum natural gradient optimization algorithm

In order to improve the convergence speed of the model and more effectively achieve the optimization task of image classification, and avoid the impact of gradient disappearance or explosion on the experimental results, we use QNGOA to update the parameters of the quantum system. This algorithm is a quantum version based on natural gradient optimization, which considers the special geometric shapes of the quantum state space. In quantum systems, small changes in parameters can lead to significant changes in quantum states due to the properties of Hilbert space. By introducing the Fubini–Study metric tensors[49] to measure the importance of different directions in the parameter space to adjust the direction and step size of optimization, we select the specific optimization process of QNGOA as follows:

Let the loss function $L(x;\theta)$ be

$$\begin{aligned}L(x;\theta) &= \langle 0|U_0^\dagger(x)U_i^\dagger(\theta_i)HU_i(\theta_i)U_0(\theta_0)|0\rangle \\ &= \langle x|U_i^\dagger(\theta_i)HU_i(\theta_i)|x\rangle.\end{aligned} \tag{10}$$

Here, $U_0(x)$ is the quantum R_y gate used for angle encoding, and the Pauli-Z matrix is the Hamiltonian operator of the loss

function. The gradient of the middle loss function is

$$\nabla L(x;\theta) = \frac{\partial L(x;\theta_t)}{\partial \theta_t} = \frac{1}{2}\left[L\left(x;\theta_t+\frac{\pi}{2}\right) - L\left(x;\theta_t-\frac{\pi}{2}\right)\right]. \quad (11)$$

Finally, the iterative-discrete time dynamic system is used to find the local optimal solution of the loss function $L(x;\theta)$,

$$\theta_{t+1} = \text{argmin}\left[\langle\theta-\theta_t, \nabla L(\theta_t)\rangle + \frac{1}{2\eta}\|\theta-\theta_t\|^2_{g(\theta_t)}\right], \quad (12)$$

$$\|\theta-\theta_t\|^2_{g(\theta_t)} = \langle\theta-\theta_t, g(\theta_t)(\theta-\theta_t)\rangle, \quad (13)$$

where η is the learning rate and $g(\theta_t)$ is the Fubini–Study metric tensor.

4. Experimental results and analysis

In the experiment, we use the tensorflow quantum framework[50] on the classical computer to complete numerical simulations of the experiment. For the HQCCNN, we use the backpropagation algorithm to optimize the parameters of the classical parts to calculate the loss function and gradient, and allow it to be optimized by gradient descent.[51] However, in the quantum parts, we employ the QNGOA to guide the optimization process by using the measurement of quantum state space.

We use the MNIST dataset to simulate and analyze the HQCCNN. For each class in the MNIST dataset, we randomly select 200 images with a size of 28×28 as the training set and 40 images as the test set. In order to comply with the current quantum experimental simulation environment, we downsample on the images and ultimately obtained 2000 training images and 400 test images of size 4×4.

The setup of this experiment is as follows: We conduct binary and multivariate classification on the selected MNIST dataset to test the classification performance and accuracy of the HQCCNN. Then, we conduct performance analysis and comparison with a CNN with the same structure. To further demonstrate the performance of the HQCCNN, we conduct multivariate classification comparison experiments with other models. In binary and multivariate classification tasks, we use one and two classical hidden layers containing ReLu nonlinear functions, respectively, and use the classic optimizer Adam[52] to optimize parameters. Set the batch size to 32, the learning rate to 0.01, and the number of epochs to 100.

4.1. Evaluation indicators

In order to evaluate the performance of the HQCCNN more intuitively, the ultimate goal is to minimize the loss function during optimization, and we take the loss value of the training set and the accurate value of the test set as metrics. We use the cross-entropy function as the loss function. The function is defined as

$$H = -\sum_{c=1}^{k} y_c \log p_c, \quad (14)$$

where p_c is the probability vector predicted by the model, y_c is 0 or 1, which is used to determine whether the image belongs to the predicted category, and k is the number of categories. Accuracy is defined as

$$\text{Acc} = \frac{1}{N}\sum_{N}^{i=1}(\text{argmax}\, p_c = y_c), \quad (15)$$

where N represents the size of the testing set and the position of the maximum output value of the argmax function in a vector.

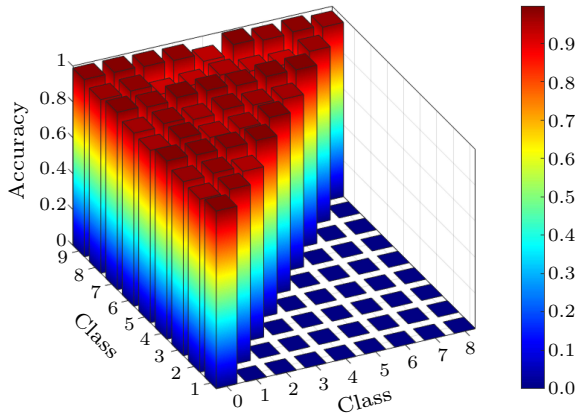


Fig. 6. The result of binary classification accuracy.

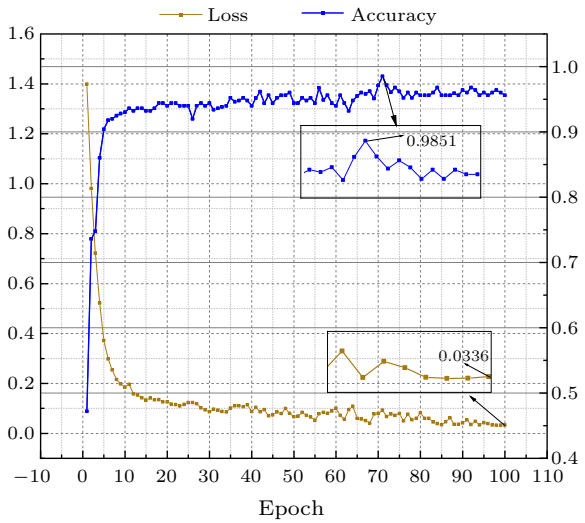


Fig. 7. The results of multivariate classification. The blue curve represents the accuracy of classification, and the brown curve represents the loss value.

4.2. Binary classification and multivariate classification

In this section, we perform binary and multivariate classification on the selected MNIST dataset, where the HQCCNN sets up a quantum convolutional layer with a quantum convolution kernel and no pooling layer. The results of binary

classification are shown in Fig. 6. As can be seen from Fig. 6, the HQCCNN has the highest classification accuracy. Among them, the classification accuracy of numbers 5 and 7 is the highest, i.e., 99.71%. The lowest classification is the numbers 2 and 8, which is 92.19%. We conduct 100 epochs in the execution of the multivariate classification task, with the highest accuracy reaching 98.51%. The final loss value is 0.0336. The results are shown in Fig. 7.

The experiment shows that the HQCCNN is fully suitable for binary and multivariate classification tasks. However, from the overall classification results, in binary classification tasks, when other numbers are classified with 8, the classification accuracy is significantly lower than other numbers. In addition, there is also a state of unstable classification accuracy when performing multivariate classification. The reason for these results may be that when downsampling the dataset, the similarity of individual images increases due to the loss of features.

4.3. Comparative experiment

We explore the differences in classification performance between the HQCCNN and CNN. We set the HQCCNN and CNN to two similar structures, that is, they have the same number of parameters, and divide the experiment into two groups, one with pooling layers and the other without pooling layers. Then, classify the 10 classes of the MNIST dataset and analyze the changes in the result curves. The HQCCNN is set to 1 convolutional layer and 2 convolutional kernels. The CNN has a convolutional layer filled with 1, containing 2 convolutional kernels of size 2×2, with a step size of 1 and an activation function of ReLu. Finally, there is a classic fully connected layer with the ReLu activation function. For a group with pooling layers, both of them are set to have 1 pooling layer, and both use max pooling with a step size of 1 and a kernel size of 2×2. The loss values of the training set and the accuracy of the test set are shown in Fig. 8.

Next, in maintaining a similar structure between the HQCCNN and the CNN, we conduct a complex binary classification experiment. The experimental setup is also divided into two groups, one with the pooling layer, and the other without the pooling layer. The classification results are listed in Tables 1 and 2. In addition, we conduct comparative experiments on our model with another HQCNN[53] and differential convolutional neural network,[54] the results are shown in Fig. 9.

From Fig. 8, it can be seen that in the group without pooling layers, the loss values of the HQCCNN and the CNN are almost the same. However, the classification accuracy of the former is far superior to the latter. In the group with pooling layers, it is more evident that the HQCCNN has higher classification accuracy and faster convergence speed. From Tables 1 and 2, it can also be seen that the HQCCNN has better classification performance and stronger learning ability compared to the CNN. Experiments show that under similar structures, the HQCCNN performs better than the CNN. The experimental results in Fig. 9 demonstrate that our model performs better than the HQCNN and differential neural networks in multivariate classification tasks.

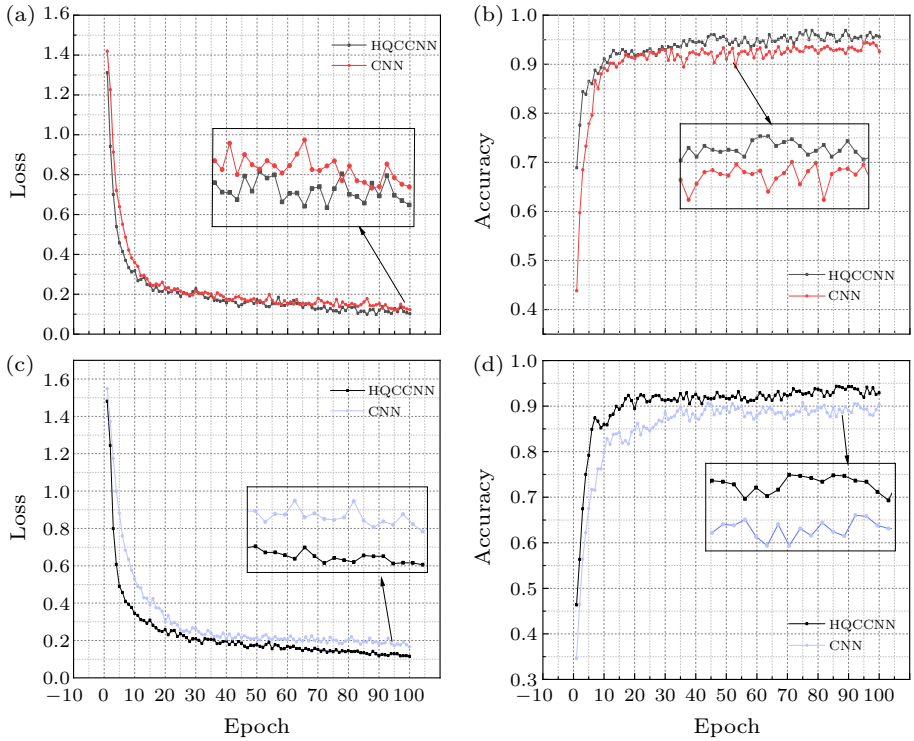


Fig. 8. The numerical comparison results between the HQCCNN and the CNN. [(a), (b)] The numerical results without adding pooling layers, where the solid grey line represents the results of the HQCCNN and the red line represents the CNN. [(c), (d)] The results of adding pooling layers, where black represents the result of the HQCCNN and cyan represents the CNN.

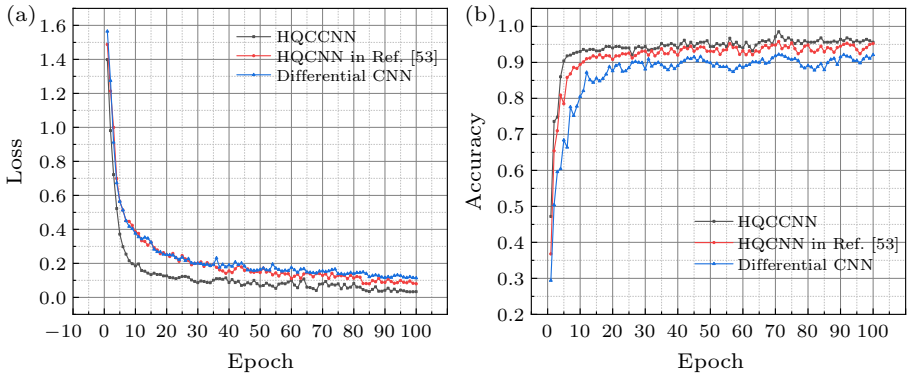


Fig. 9. Comparison of the numerical results between the HQCCNN and other models. Here the black curve represents the HQCCNN, the red curve represents the HQCNN, and the blue curve represents the differential convolutional neural network.

Table 1. Binary classification results with pooling layer.

Tasks	Model	Experimental setup	Accuracy
Odd number			0.9157
Even number	HQCCNN	with pooling layer	0.9241
Both			0.9184
Odd number			0.9012
Even number	CNN	with pooling layer	0.8912
Both			0.8981

Table 2. Binary classification results without pooling layer.

Tasks	Model	Experimental setup	Accuracy
Odd number			0.9318
Even number	HQCCNN	without pooling layer	0.9397
Both			0.9412
Odd number			0.9147
Even number	CNN	without pooling layer	0.9078
Both			0.9083

5. Conclusion

In this paper, we propose a new model of hybrid quantum–classical convolutional neural networks, which replaces the classical part by designing PQCs to form a new quantum convolutional layer and a new hybrid quantum–classical fully connected layer. This method greatly improves the feature extraction and classification accuracy of images, and accelerates the convergence speed. We use the designed HQCCNN to perform binary and multivariate classification on the MNIST dataset to test model performance, and then conduct comparative experiments with a CNN. The results show that under similar structures, the HQCCNN outperforms the CNN in classification performance and convergence speed. Finally, we compare the model with other models to further demonstrate the performance of the HQCCNN.

In the future, the development of quantum computing provides the possibility to solve complex classical problems. Due to the limitations of qubits, there are still significant issues in the processing of ultra-high dimensional data. Therefore, for ultra-high dimensional images, we must perform dimensionality reduction processing on the images. However, various factors such as noise can greatly reduce the accuracy of image feature extraction, which greatly affects the accuracy of image classification. With the continuous efforts of researchers, quantum computers that can facilitate the processing of high-dimensional data problems are constantly emerging, which further enhances the possibility of solving more complex classical problems.

Acknowledgments

Project supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2021MF049) and the Joint Fund of Natural Science Foundation of Shandong Province (Grant Nos. ZR2022LLZ012 and ZR2021LLZ001).

References

[1] Benbarrad T, Salhaoui M, Kenitar S B and Arioua M 2021 *J. Sensor Actuator Networks* **10** 7
[2] Teng X Y and Gong Y X 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **392** 062202
[3] Ren S Q, He K M, Girshick R and Sun J 2015 *Adv. Neural Inf. Process. Syst.* **28** 1137
[4] Litjens G, Kooi T, Bejnordi B E, Setio A A A, Ciompi F, Ghafoorian M, van der Laak J A, Van Ginneken B and Sánchez C I 2017 *Med. Image Anal.* **42** 60
[5] Li S S, Long G L, Bai F S, Feng S L and Zheng H Z 2001 *Proc. Natl. Acad. Sci. USA* **98** 11847

[6] Knill E 2010 *Nature* **463** 441
[7] Ladd T D, Jelezko F, Laflamme R, Nakamura Y, Monroe C and O'Brien J L 2010 *Nature* **464** 45
[8] Zhong H S, Wang H, Deng Y H, Chen M C, Peng L C, Luo Y H,Qin J, Wu D, Ding X, Hu Y, Hu P, Yang X Y, Zhang W J, Li H, Li Y X, Jiang X, Gan L, Yang G W, You L X, Wang Z, Li L, Liu N L, Lu C Y and Pan J W 2020 *Science* **370** 1460
[9] Preskill J 2018 *Quantum* **2** 79
[10] Biamonte J, Wittek P Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 7671
[11] Zhao J B, Zhang T, Jiang J W, Fang T and Ma H Y 2022 *Sci. Rep.* **12** 14253
[12] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys.* **10** 631
[13] Mu X Y, Wang H W, Bao R Y, Wang S M and Ma H Y 2023 *Quantum Inf. Process.* **22** 223
[14] Rebentrost P, Mohseni M and Lloyd S 2014 *Phys. Rev. A* **113** 130503
[15] Liu G Z, Li W, Fan X K, Li Z, Wang Y X and Ma H Y 2022 *Entropy* **24** 608
[16] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 *New J. Phys.* **18** 023023
[17] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 *Quantum Sci. Technol.* **4** 043001
[18] Liu Y, Wang D Y, Xue S C, Huang A Q, Fu X, Qiang X G, Xu P, Huang H L,, Deng M T, Guo C, Yang X J and Wu J J 2020 *Phys. Rev. A* **101** 052316
[19] Lee J, Magann A B, Rabitz H A and Arenz C 2021 *Phys. Rev. A* **104** 032401
[20] Farhi E, Goldstone J, Gutmann S and Neven H 2017 1703.06199 [quant-ph]
[21] Farhi E and Neven H 2018 arXiv:1802.06002 [quant-ph]
[22] Benedetti M, Garcia-Pintos D, Perdomo O, Leyton-Ortega V, Nam Y and Perdomo-Ortiz A 2020 *npj Quantum Inf.* **5** 45
[23] Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O'brien J L 2014 *Nat. Commun.* **5** 4213
[24] Parrish R M, Hohenstein E G, McMahon P L and Martínez T J 2019 *Phys. Rev. A* **122** 230401
[25] Xin T, Che L, Xi C, Singh A, Nie X, Li J, Dong W and Lu D W 2021 *Phys. Rev. A* **126** 110502
[26] Gircha A I, Boev A S, Avchaciov K, Fedichev P O and Fedorov A K 2023 *Sci. Rep.* **13** 8250
[27] Bokhan D, Mastiukova A S, Boev A S, Trubnikov D N and Fedorov A K 2023 *Front. Phys.* **10** 1069985
[28] Lin X, Chen Z and Wei Z 2021 *Phys. Rev. A* **107** 062409
[29] Zhang S X, Wan Z Q, Lee C K, Hsieh C Y, Zhang S and Yao H 2022 *Phys. Rev. A* **128** 120502
[30] Liu J, Lim K H, Wood K L, Huang W, Guo C and Huang H L 2021 *Sci. China Phys. Mech. Astron.* **64** 290311
[31] Liu Y H 2018 *J. Phys.: Conf. Ser.* **1087** 062032
[32] Chen L Y, Li S B, Bai Q, Yang J, Jiang S L and Miao Y M 2021 *Remote Sensing* **13** 4712
[33] Nishani E and Çiço B 2017 2017 *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, June 11–15, 2017, Bar, Montenegro, p. 1
[34] Scardino G, Scicchitano G, Chirivì M, Costa P J, Luparelli A and Mastronuzzi G 2022 *Remote Sensing* **14** 2994
[35] Cong I, Choi S and Lukin M D 2019 *Nat. Phys.* **15** 1273
[36] Herrmann J, Llima S M, Remm A, Zapletal P, McMahon N A, Scarato C, Swiadek F, Andersen C K, Hellings C, Krinner S, Lacroix N, Lazar S, Kerschbaum M, Zanuz D C, Norris G J, Hartmann M J, Wallraff A and Eichler C 2022 *Nat. Commun.* **13** 4144
[37] Chen G M, Chen Q, Long S, Zhu W H, Yuan Z D and Wu Y L 2023 *Pattern Anal. Appl.* **26** 655
[38] Véstias M P 2021 *Encyclopedia of Information Science and Technology*, 5th edn. (IGI Global) p. 12
[39] Pelletier C, Webb G I and Petitjean F 2019 *Remote Sensing* **11** 523
[40] Sharma N, Jain V and Mishra A 2018 *Procedia Comput. Sci.* **132** 377
[41] Li W, Chu P C, Liu G Z, Tian Y B, Qiu T H and Wang S M 2015 *J. Infr. Millimeter Waves* **34** 497
[42] Habibi Aghdam H, Jahani Heravi E, Habibi Aghdam H and Jahani Heravi E 2017 *Guide to Convolutional Neural Networks* (Spain: Spinger Link) p. 85
[43] Li Y C, Zhou R G, Xu R Q, Luo J and Hu W W 2020 *Quantum Sci. Techno.* **5** 044003
[44] Stokes J, Izaac J, Killoran N and Carleo G 2020 *Quantum* **4** 269
[45] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 *Proc. IEEE* **86** 2278
[46] Ioffe S and Szegedy C 2015 *ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning* vol. 37 p. 448
[47] Agarap A F 2018 arXiv:1803.08375 [cs.NE]
[48] Ovalle-Magallanes E, Alvarado-Carrillo D E, Avina-Cervantes J G, Cruz-Aceves I and Ruiz-Pinales J 2023 *Appl. Soft Comput.* **14** 110307
[49] Cheng R 2013 arXiv:1012.1337 [quant-ph]
[50] Broughton M, Verdon G, McCourt T, *et al.* 2021 arXiv:2003.02989 [quant-ph]
[51] Rumelhart D E, Hinton G E and Williams R J 1986 *Nature* **323** 533
[52] Kingma D P and Ba J Adam 2014 arXiv:1412.6980 [cs.LG]
[53] Christol Y 2023 *Github: christorange QC-CNN*
[54] Sarıgül M, Ozyildirim B M and Avci M 2019 *Neural Netw.* **116** 279