Applied Intelligence
https://doi.org/10.1007/s10489-022-03525-0

Quantum-inspired complex convolutional neural networks

Shangshang Shi[1] · Zhimin Wang[1] · Guolong Cui[1] · Shengbin Wang[1] · Ruimin Shang[1] · Wendong Li[1] · Zhiqiang Wei[1,2] · Yongjian Gu[1]

Accepted: 14 March 2022

Abstract

Quantum-inspired artificial neural network is an interesting research area, which combines quantum computing and deep learning. Several models of quantum-inspired neuron with real-valued weights have been proposed, and they were mainly used to build the three-layer feedforward neural networks. In this work, we improve the convolutional neural networks (CNNs) by utilizing the quantum-inspired way of data representation and convolutional operation. Specifically, we first improve the quantum-inspired neuron by exploiting the complex-valued weights, which have richer representational capacity and better non-linearity. Moreover, we extend the method implementing the quantum-inspired neurons to perform convolutional operations, and naturally draw the models of quantum-inspired convolutional neural networks (QICNNs) capable of processing high-dimensional data. Here five specific types of QICNNs are proposed, which are different in the way of implementing the convolutional layers and fully connected layers. We establish the detail mathematical framework to implement the QICNNs. The performances of accuracy, convergence and robustness of the five QICNNs against the classical counterpart are tested using the MNIST and CIFAR-10 datasets. The results show that (1) the QICNN can achieve higher classification accuracy (up to 99.65%) than the classical CNN when using the MNIST dataset; (2) the QICNN has faster convergence speed, which means that QICNN can be trained easily than classical CNN when they have a similar number of parameters; (3) the QICNN has better robustness in the case of employing different way of weight initialization or rotating the input data. It is expected that our QICNNs can outperform the classical counterparts in more practical learning tasks.

Keywords Quantum-inspired CNNs · Complex CNNs · Quantum-inspired neuron · Classification accuracy · Convergence · Robustness

1 Introduction

In the past few years, the field of quantum computing has witnessed many breakthroughs in both quantum processors [1, 2] and quantum algorithms [3–5]. Quantum computing performs information processing in a quantum mechanical way, that is, encoding the data into an exponentially large Hilbert space and manipulating this data space in a parallel way, which result into the exponential improvements of data representation power and computational power. On the other hand, deep learning is the art of making computers learn how to solve problems based on huge amounts of data, and now faces challenges of storage and computational resources. It is therefore only natural to ask if and how they could be combined to add something new to how machines recognize patterns in data.

At present, algorithm researches at the junction of quantum computing and deep learning focus on two active directions. One is to search for the real quantum algorithms [3]; such algorithms harness the unique properties of quantum mechanics, including the quantum superposition and entanglement, to encode and process data. The other direction is to develop the so-called quantum-inspired algorithms [6–19]; such algorithms borrow the basic ideas and formalism of quantum computing, such as the way of data representation and complex operations, to improve the existing classical algorithms or even find new algorithms. The salient difference between the two realms of algorithms is that the first kind of algorithms run on the quantum computers, while the second one run on the conventional computers.

✉ Zhimin Wang
wangzhimin@ouc.edu.cn

✉ Yongjian Gu
yjgu@ouc.edu.cn

[1] College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China

[2] High Performance Computing Center, Pilot National Laboratory for Marine Science and Technology (Qingdao), Qingdao 266100, China

Published online: 07 April 2022

Quantum-inspired techniques have been applied in various disciplines to improve the classical algorithms, such as artificial intelligence, signal processing and image processing [6–19]. Among them, quantum-inspired neural networks (QINNs) have received increasing amount of attentions [13–19]. Starting from the first model of QINN proposed by Kak in 1995 [13], a long series of QINN models and relative learning methods have been developed [14–23], including the recently proposed quantum probability-inspired graph neural network [20], quantum behaved particle swarm optimization method [21] and multiple kernel k-means algorithm [22]. More remarkably, quantum-inspired techniques have been employed in the neuromorphic computing giving rise to the quantum superposition inspired spiking neural network (QS-SNN) [23]. The QS-SNN encodes the classical input data by harnessing the basic idea of quantum superposition, which can express richer data information. In more general terms, it is intriguing to combine the quantum computing and neuromorphic computing because they are deemed two promising avenues to build the future of AI. The neuromorphic computing is biologically inspired [24, 25], and the quantum computing is quantum-mechanically inspired. The combination between them necessarily leave us much to exploit in future, for example, to combine the spiking neural networks (SNNs) [26–28] with the quantum convolutional neural networks (QCNNs) [29].

Here we focus on the topic of quantum-inspired convolutional neural networks. For now, the existing quantum-inspired neural networks are mainly simple three-layer feedforward networks, and they all use the quantum-inspired neurons with real-valued weights. However, on the one hand, quantum computing intrinsically manipulates complex weights, namely the complex probability amplitude. That is, the computing process in quantum computing can be represented by $U(\sum_i \alpha_i| \ i\rangle) = \sum_j \beta_j \ | \ j\rangle$, where U is a complex unitary matrix, and α_i and β_j are the complex column vectors. Therefore, it is a natural idea to use complex-valued weights in the quantum-inspired neurons. On the other hand, the benefits of using the complex weights in deep learning have been argued recently and the complex-valued deep neural networks were proposed [30]. Complex parameters have richer representational capacity; complex networks have the potential to enable easier optimization, better generalization characteristics, faster learning and to allow for noise-robust memory mechanisms [30]. Here we, in fact, propose an instantiation of complex neural networks by exploiting the quantum-inspired techniques.

In the present work, we first develop the quantum-inspired neurons with complex-valued weights. Then we find that the method of implementing the quantum-inspired neurons can be extended straightforwardly to implement convolutional operations with complex-valued weights. Indeed, implementing one neuron is in fact to calculate the inner product between the data vector and parameter vector, while the convolutional operation can also be seen as the inner product between the data vector (adapted from the data matrix) and kernel vector (adapted from the convolution kernel matrix). Having the quantum-inspired neurons and convolutional operations, we can naturally build the quantum-inspired convolutional neural networks (QICNNs). In general, the quantum-inspired neurons are employed in the fully connected layers and the quantum-inspired convolutional operations in the convolutional layers.

Conventional CNNs has been widely used in deep learning [31–34], while as far as we know the present QICNNs are the first complex-valued CNN models utilizing the quantum-inspired way of data representation and convolutional operation. Specifically, in QICNNs, the real input space is first mapped to the complex feature space, and then the parameters are searched in the feature space using the operations adapted from the quantum computing. The present QICNNs are expected to be powerful in learning the complex pattern existed in the high-dimensional raw data.

This paper is organized as follows. In Section 2, we describe the improved quantum-inspired neuron with complex-valued weights, and the approach to extend it to the convolutional operations. Section 3 presents five types of QICNNs with different structures in the convolutional and fully connected layers. In Section 4, the performances of our QICNNs, including the classification accuracy, convergence and robustness, are tested using the MNIST and CIFAR-10 dataset. Finally, conclusions are given in Section 5.

2 Quantum-inspired neuron and convolutional operation

2.1 An improved quantum-inspired neuron

In quantum computing, a qubit is implemented using a two-level quantum system. Its state can be described mathematically by a two-dimensional complex Hilbert space. From the quantum principle of state superposition, an arbitrary one-qubit state can be written as $|\varphi > =\alpha|0 > +\beta|1 >$ with α and β being the probability amplitude of the computational basis $|0 >$ and $|1>$, respectively. The amplitudes α and β are complex numbers and satisfy the normalization condition, i.e. $|\alpha|^2 + |\beta|^2 = 1$. Considering the normalization constraint, the state can be rewritten as

$$|\varphi\rangle = e^{i\gamma}\left(\cos\theta|0\rangle + e^{i\varphi}\sin\theta|1\rangle\right), \tag{1}$$

where γ, θ and φ are real numbers, and the factor $e^{i\gamma}$ and $e^{i\varphi}$ is the global and relative phase, respectively. Now we ignore the phase information in Eq. (1) and combine the magnitude information as a complex number, and then we obtain a qubit-inspired representation of input data,

$$f(\theta) = e^{i\theta} = \cos\theta + i\sin\theta. \tag{2}$$

Next, we discuss how to operate the data represented as Eq. (2). As it is well known, a single-qubit state corresponds to a point on the surface of Bloch sphere with the coordinate (θ, φ); and the evolution of single-qubit state can be taken as a rotation of the vector on the Bloch sphere [35]. Here, Eq. (2) corresponds to a ring on the surface of Bloch sphere, and operating the data of Eq. (2) corresponds to the rotation along latitude direction. Thus, the data of Eq. (2) can be operated using a similar way as quantum state evolution,

$$\begin{aligned} U \cdot f(\theta_0) &= U \cdot (\cos(\theta_0) + i\sin(\theta_0)) \\ &= \cos(\theta + \theta_0) + i\sin(\theta + \theta_0). \end{aligned} \tag{3}$$

Note that, here the operation U can be itself expressed as a complex number, that is $U = f(\theta) = e^{i\theta}$, so $U \cdot f(\theta_0) = e^{i\theta} \cdot e^{i\theta_0} = e^{i(\theta+\theta_0)}$. Particularly, using $\theta' = \frac{\pi}{2}\delta - \theta$, we can define an operation similar to CNOT gate as follows,

$$f\left(\frac{\pi}{2}\delta - \theta\right) = \begin{cases} \cos\theta - i\sin\theta & (\delta = 0) \\ \sin\theta + i\cos\theta & (\delta = 1) \end{cases} \tag{4}$$

When $\delta = 0$, the data remains unchanged if we ignore the change of the phase information, while when $\delta = 1$, the real and imaginary part interchange (corresponding to the bit flip in Eq. (1)).

So far we have defined the quantum inspired data representations and operations. Based on these we can obtain a typical model of quantum-inspired neuron as shown in Fig. 1a. In the previous works, the weights u_i are taken as real numbers because it is convenient to be implemented in the commonly used framework of machine learning. Now we improve the quantum-inspired neuron by directly using complex weights as shown Fig. 1b. The output of the improved quantum-inspired neuron is

$$S_2 = f(\theta), \tag{5}$$

with

$$\theta = \frac{\pi}{2} - \arg(\Sigma), \tag{6}$$

$$\Sigma = \sum_{n=1}^{N} f(u_n) f\left(\frac{\pi}{2} x_n\right) - f(b). \tag{7}$$

In the equations, $f(\cdot)$ represents the function as defined in Eq.(2); arg(·) is to calculate the argument of a complex number; $\{x_1, \ldots x_N\}$are the inputs which are real numbers. We have performed several numerical experiments and find that the fully connected network with complex-valued weights indeed performs better than that with real weights. So below we will employ the improved quantum-inspired neuron to construct the convolutional neural networks.

2.2 Quantum-inspired convolutional operation

In general, the convolutional neural network (CNN) is comprised of two main parts, namely the convolutional layers and the fully connected layers. The quantum-inspired neurons can be employed to build the fully connected layers. Moreover, a similar way can be used to perform the complex convolutional operation as depicted in Fig. 2. The input pixel and the convolutional kernel are first transformed into complex space (using Eq. (2)), and then the multiplication between them is just like the process of weighted summation as shown in Fig. 1b. After the convolutional operation (namely a rotation operation as shown in Eq. (3), a NOT operation is added as it is done in the quantum-inspired neurons.

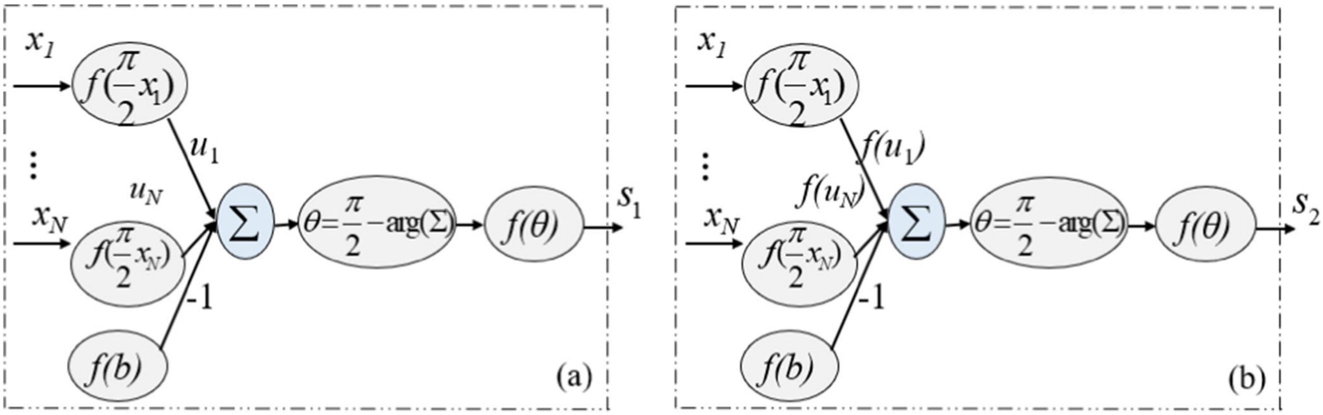

Fig. 1 The structure of the common quantum-inspired neuron with real valued weights (**a**), and the improved quantum-inspired neuron with complex weights (**b**). The weights$\{u_1, u_2 \ldots u_N\}$are real numbers. The module Σ represents weighted summation operation

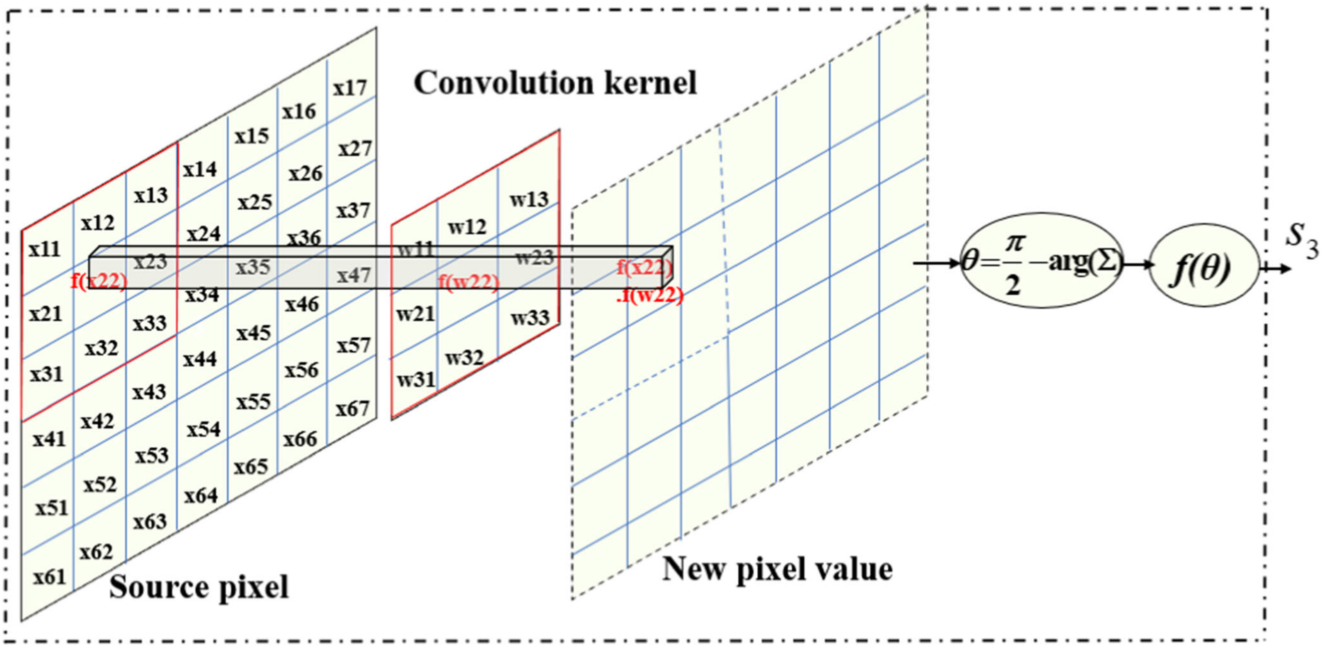

Fig. 2 The quantum-inspired method of implementing the convolutional operations used to construct convolutional neural networks

Now that we have developed the models of quantum-inspired neutron and quantum-inspired convolutional operation, we can naturally construct the quantum-inspired convolutional neural networks (QICNNs) by combining them. However, since most present machine learning frameworks are based on real-valued representations and operations, we first need to adapt the complex-valued neutron and convolutional operations to be implemented conveniently on the common frameworks. The approach is as follows.

As shown in Eq. (2) and (3), both the data and operation can be expressed as a complex number. A complex number $z = a + ib$ has a real component a and an imaginary component b. The basic idea is that the real part and imaginary part are represented as logically distinct two real-valued entities and perform complex arithmetic using real-valued arithmetic internally [30]. For example, given a typical real-valued input data with m channels, we transform them to complex-valued channels as shown in Fig. 1. After separating the real and imaginary part of the complex features, the original m channels are expanded to 2 m, of which the first m ones are allocated to represent the real components and the remaining m the imaginary ones.

Next, the convolutional operations on the complex values are performed based on the multiplication principle of complex numbers. Specifically, the data can be expressed as $x = x_{real} + ix_{imag}$ with x_{real} and x_{imag} being the real matrices of two channels, and the complex filter matrix is $W = w_{real} + iw_{imag}$ with w_{real} and w_{imag} being the real matrices to be learned. Then the convolutional operation is implemented as follows,

$$W{*}x = \left(w_{real}{*}x_{real} - \mathrm{w}_{imag}{*}x_{imag}\right) \\ + i\left(w_{imag}{*}x_{real} + \mathrm{w}_{real}{*}x_{imag}\right). \tag{8}$$

According to the computational procedure in Fig. 2, the quantum-inspired convolutional operation is realized in the way as shown in Fig. 3.

In summary, the quantum-inspired convolutional operation is implemented through the following five steps:

(I) Transform the input data and filter parameters to complex ones using Eq. (2), i.e. $x_0 \mapsto f(x_0)\, x_0 \mapsto f(w_0)$. Then we have $x = x_{real} + ix_{imag}$ and $w = w_{real} + iw_{imag}$.

(II) Perform the complex-valued convolutional operation using the formula $x * w = (x_{real} * w_{real} - x_{imag} * w_{imag}) + i(x_{real} * w_{imag} + x_{imag} * w_{real})$.

(III) Calculate the phase using $\arg(\upsilon) = \arctan\left(\frac{x_{real}{*}w_{imag} + x_{imag}{*}\mathrm{w}_{real}}{x_{real}{*}w_{real} - \mathrm{x}_{imag}{*}w_{imag}}\right)$.

(IV) Perform an NOT operation, namely $\theta = \frac{\pi}{2} - \arg(\upsilon)$.

(V) Compute the output using $y = f(\theta)\ and\ output = |\mathrm{Im}(y)|^2$.

Note that the convolutional operation in (II) represents a rotation operation on each input data, while step (IV) is a global NOT operation on the sum of all the input data.

3 The architecture of quantum-inspired convolutional neural networks

In order to have a reference to benchmark the QICNNs, we take a classical CNN [31–34] as the template to build the QICNNs. The classical template CNN is adapted from the

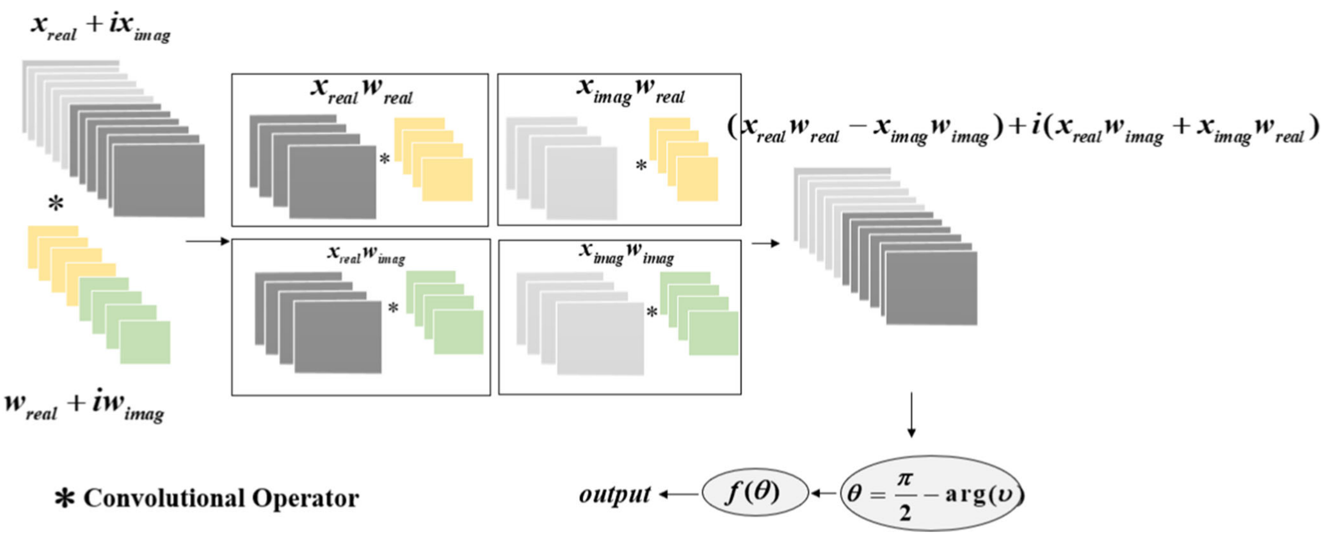

Fig. 3 The mathematical details of implementing the quantum-inspired convolution operations with complex-valued weights

LeNet-5 model, which is widely used for image recognition of handwritten digit and English letter. In general, the template CNN consists of three convolution blocks and two fully connected layers, where each convolution block includes a convolutional layer, a ReLu layer and a pooling layer. Figure 4a shows the implementing details of the template CNN.

The straightforward way to build the QICNNs is that employ the quantum-inspired neuron in the fully connected layers and employ the quantum-inspired convolutional operation in the convolutional layer. Moreover, we find that employing the two quantum-inspired techniques as shown in Fig. 4b in different stages of the template CNN will have a great influence on the performance of the resulted QICNNs.

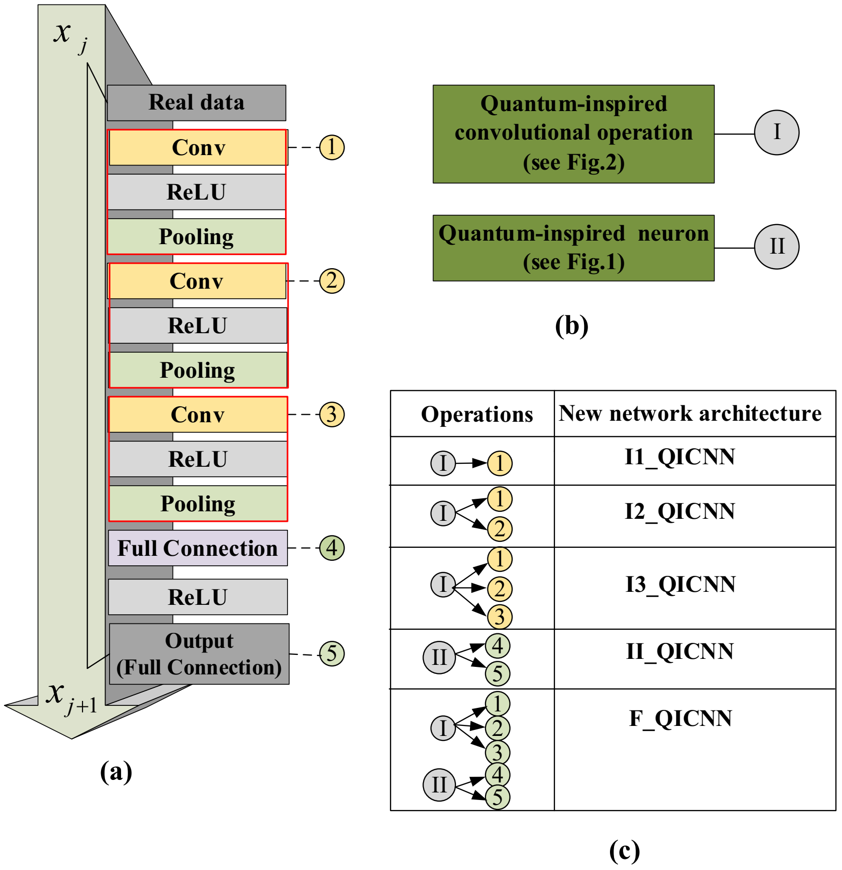

Fig. 4 The strategy to build QICNNs based on the classical template CNN. **a** The implementing details of the template CNN; **b** the two building blocks of the QICNN, namely the quantum-inspired convolutional operation (I) and the improved quantum-inspired neuron (II); **c** five ways to build QICNNs by embedding the building blocks in the template CNN

Here, we exploit five types of QICNNs obtained by embedding the quantum-inspired techniques in the template CNN in the way as shown in Fig. 4c.

More specifically, the first QICNN is called I1_QICNN for its first convolutional layer uses the quantum-inspired convolutional operation and the others are the classical ones. Accordingly, it is easy to understand the structures of I2_QICNN and I3_QICNN. The first three models are also called I_QICNN because they use only the quantum-inspired convolutional operation. The fourth one is called II_QICNN because it uses the quantum-inspired neuron in the fully connected layers. The last one called F_QICNN is the full quantum case where both the quantum-inspired convolutional operation and neuron are used. Below we present the mathematical details of implementing the QICNNs.

3.1 Mathematical details of implementing I_QICNN

We take the model of I3_QICNN as an example to describe the mathematical details of the I_QICNN algorithms. It goes through the following computational steps.

(1) Input layer. In this layer, the input matrix is rescaled into the range [0, 1], and then multiplied by π/2 to get the phase values into the range [0, π/2]. The output of the input layer is as follows,

$$x_{input1}=\frac{\pi}{2}x_{data} \tag{9}$$

$$x_{real}=\cos\left(x_{input1}\right),x_{imag}=\sin\left(x_{input1}\right) \tag{10}$$

(2) Convolutional layer 1. This is the first convolutional layer in which a quantum-inspired convolutional operation is used. The real and imaginary part of the convolutional kernel matrix are obtained by $w_{real1} = \cos(w_0)$, $w_{imag1} = \sin(w_0)$. It is the same for the bias term, i.e. $b_{real1} = \cos(b_0)$, $b_{imag1} = \sin(b_0)$. Then the convolutional operation is performed as follows,

$$\begin{aligned} y_{c1\_real} &= conv\left(x_{real},w_{real1}\right)-conv\left(x_{imag},w_{imag1}\right)-b_{real1} \\ y_{c1\_imag} &= conv\left(x_{imag},w_{real1}\right)+conv\left(x_{real},w_{imag1}\right)-b_{imag1} \end{aligned} \tag{11}$$

In order to perform the controlled NOT operation as shown in Fig. 3, we calculate the angle using $\arg_1(\upsilon)=\arctan\left(\frac{y_{c1\_imag}}{y_{c1\_real}}\right)$. The output is $y_{c\_1}=f\left(\frac{\pi}{2}-\arg_1(\upsilon)\right)$.

(3) ReLu layer 1. The ReLu function is used as the activation function, i.e. $y_{relu1\_real} = \mathrm{Re}\,LU(y_{c\_1\_real})$, $y_{relu1\_imag} = \mathrm{Re}\,LU(y_{c\_1\_imag})$.

(4) Pooling layer 1. The pooling is the down-sampling (sub-sampling) process. It can reduce the amount of data to be processed without loss of useful information. Maximum pooling is used here, namely $y_{pool1\_real} = \max\ pool(y_{relu1\_real})$ and $y_{pool1\_imag} = \max\ pool(y_{relu1\_imag})$.

(5 ~ 7) The second convolutional layer. The above steps of (2), (3) and (4) are repeated for Convolutional layer 2, ReLu layer 2 and Pooling layer 2.

(8 ~ 10) The third convolutional layer. Since the following fully connected layer after this third convolutional layer is based on real values, the imaginary part of the output of Convolutional layer 3 is remained as the input of ReLu layer 3. That is, after the complex-valued convolutional operation (namely Eqs. (5) (7)), the angle is $\arg_3(\upsilon)=\arctan\left(\frac{y_{c3\_imag}}{y_{c3\_real}}\right)$, then the output is $y_{c\_3}=\left|\mathrm{Im}\left(f\left(\frac{\pi}{2}-\arg_3(\upsilon)\right)\right)\right|^2$. The operations for ReLu layer 3 and Pooling layer 3 are the same as above.

(11) Fully connected layer 1. The role of the fully connected layer is mainly to achieve classification. This layer uses the real-valued classical neuron. First, the weighted summation is done, that is $y_{f1} = y_{pool3} * w_{f1} + b_{f1}$, where $\mathrm{y}_{\mathrm{pool3}}$ is the output of the Pooling layer 3; w_{f1} the connection weights between convolutional layer 3 and fully connected layer 1; b_{f1} the threshold of the fully connected layer 1. Then the ReLu function is performed, i.e. $y_{f1\_relu} = relu(y_{f1})$.

(12) Output layer. The output of the network is $y_{output} = y_{f1\_relu} * w_0 + b_0$ with w_0 being the weight of the output layer and b_0 being the threshold of the output layer.

3.2 Mathematical details of implementing II_QICNN

The type of II_QICNN is the network that uses classical convolutional operation in the convolutional layers and uses the improved quantum-inspired neurons in the fully connected layers. For the II_QICNN, the first three convolutional-ReLu-pooling operations are the same as the classical ones. The computational procedures of the fully connected layers is implemented as follows.

(1) Input layer. The input data is transformed into complex space with phase values in the range [0, π/2]. That is $x_{input}=\frac{\pi}{2}y_{pool3}$ and $y_{input} = f(x_{input})$ with y_{pool3} being the output of the third Pooling operation.

(2) Fully connected layer 1. According to Eqs. (5)–(7), the output of this layer is calculated as $\upsilon_1 = f(w_1)y_{input} - f(b_1)$, $\theta_1=\frac{\pi}{2}-\arg(\upsilon_1)$ and $y_{output1} = f(\theta_1)$. The parameter w_1 is the connection weight matrix between the Pooling layer and the fully connected layer 1, and b_1 is the threshold vector of the fully connected layer 1.

(4) Fully connected layer 2. This layer is the output layer. The output is slightly different from that of fully connected layer 1. The output is calculated by $\upsilon_2 = f(w_2)y_{output1} - f(b_2),\theta_2=\frac{\pi}{2}-\arg(\upsilon_2)$, $y_{output2} = f(\theta_2)$ and $y_{output} = |\mathrm{Im}(y_{output2})|^2$. The parameter w_2 is the connection weight matrix between fully connected layer 1 and fully connected layer 2; b_1 is the threshold vector of the fully connected layer 2; y_{output} is the final output of II_QICNN.

4 Experimental results

In order to test the performance of the present QICNNs against the template CNN, we evaluate all the networks using the MNIST and CIFAR-10 classification dataset. MNIST is a dataset used for recognition of handwritten digits with figures from 0 to 9 [36]. CIFAR-10 is a dataset used for object detection, which is labeled a subset of 800 million tiny images [37]. Both MNIST and CIFAR-10 datasets contain 50,000 training images and 10,000 testing images. Below we first show the performances of convergence rate and classification accuracy of the QICNNs, and then discuss the robustness of the networks.

4.1 Convergence and accuracy

The training process of the five QICNNs as well as the template CNN are shown in Fig. 5. In general, the convergence performance of the QICNNs depends on both the structure of QICNNs and the dataset processed. For the MINIST dataset, the type of II_QICNN has the fastest rate of convergence, which is remarkably better than the template CNN. However, for the CIFAR-10 dataset, the performance of template CNN is slightly better than II_QICNN. That is, the QICNNs can achieve better performance for specific problems. In addition, it can be seen that employing the quantum-inspired neurons in the fully connected layers can produce better performance than employing the quantum-inspired convolutional operations in the convolutional layers.

The performance of classification accuracy of the five QICNNs are shown in Fig. 6. Similarly, the accuracy performance of the QICNNs depends on both the specific structure of QICNNs and the dataset processed. For the MINIST dataset, the model of II_QICNN has the maximum classification accuracy (achieving 0.9965), which is remarkably better than the template CNN (namely 0.9950). For the CIFAR-10 dataset, the accuracy of the template CNN is slightly better than II_QICNN. As before, employing the quantum-inspired neurons in the fully connected layers results into better performance than employing the quantum-inspired convolutional operations in the convolutional layers. Moreover, by comparing the accuracy of the three I_QICNN networks, it seems that the more quantum-inspired convolutional operations are used, the lower the accuracy is. This may due to the fact that more quantum-inspired operations result into a larger feature space which is harder to train. We leave the detail explorations of this phenomena for the future work.

4.2 Robustness

We further explore the robustness of the QICNNs. Here we focus on the II_QICNN networks because it has the best performance both in the convergent rate and classification accuracy. The robustness of the networks is verified using the MNIST dataset from two aspects, that is, (1) employing different weight initializations and (2) rotating the input data to different degrees. Three ways of weight initialization are used, which are (1) MSRA as $w_{MSRA} \sim G\left[0, \sqrt{2/F_{in}}\right]$ (namely the He initialization), (2) truncated normal distribution as $w_{TND} \sim N(0,0.1)$, (3) random uniform distribution as $w_{RUD} \sim \left[-\sqrt{6/F_{in}}, \sqrt{6/F_{in}}\right]$ with F_{in} being the number of input neurons. As can be seen from Tables 1 and 2, the II_QICNN network is generally more robust than the template CNN. It is expected that the QICNNs can maintain better performance characteristics even when it is disturbed.

According to the above numerical experiment results, we can see that the QICNN can indeed perform better than the classical CNN with respect to classification accuracy, convergent rate and robustness. Here we remark that the presented improvements of QICNN depend on the specific structure of the networks and the dataset it processes. Particularly, the more usages of quantum-inspired convolutional operations in the network do not always result into better performance, or even make it worse. Therefore, in principle, the complex-

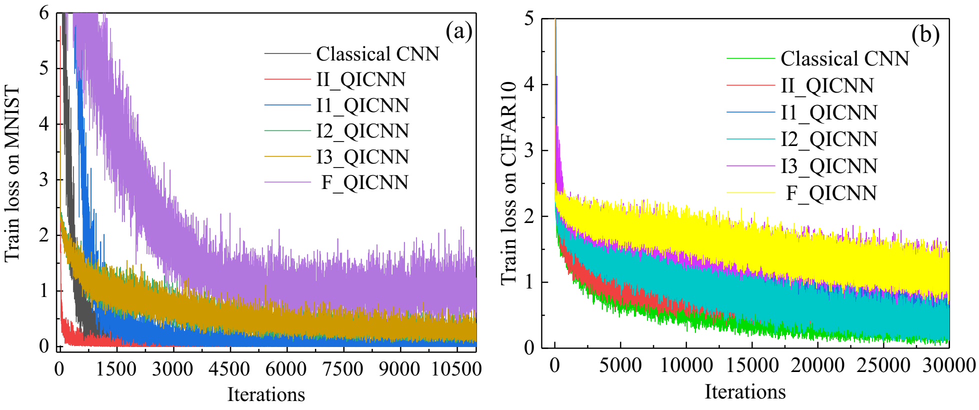

Fig. 5 The loss curves of training the five QICNNs as well as the classical template CNN using the dataset of MNIST (**a**) and CIFAR-10 (**b**)

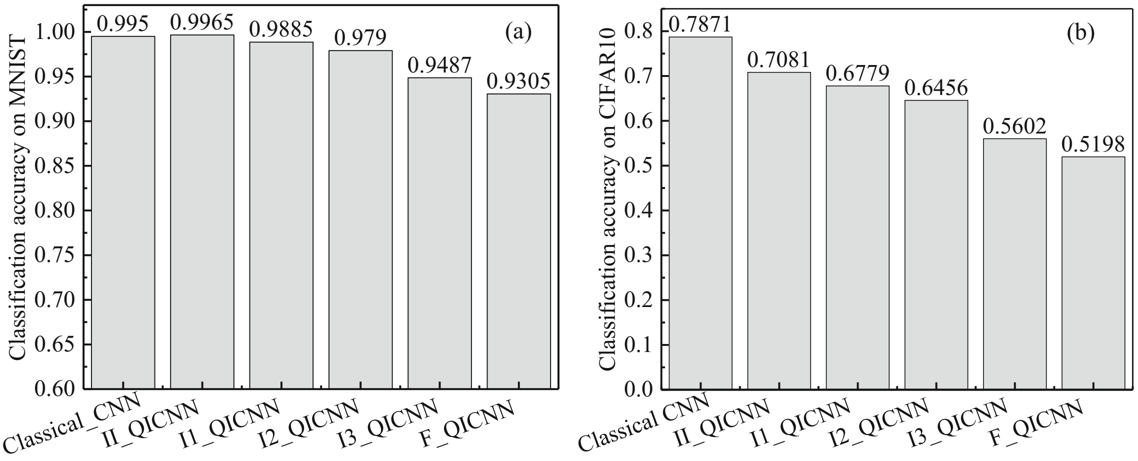

Fig. 6 The classification accuracy of the five QICNNs as well as the classical template CNN using the dataset of MNIST (**a**) and CIFAR-10 (**b**)

valued data representation and convolutional operations borrowed from quantum computing can bring about larger data encoding space and faster search of data space; but, in practice, in order to pick out an efficient QICNN model much numerical experiments should be performed based on the specific dataset of the learning task. In addition, it is worth mentioning that the quantum-inspired method of data representation and convolutional operation proposed in this work (i.e. Eq. (2) and Eqs. (5 ~ 7) respectively) can also be applied as subroutines in other image processing algorithms, such as image segmentation [38] and target extraction [39]. This is possible because data encoding and convolutional operations are the basic components of these algorithms. The broader combination would give rise to more efficient algorithms.

5 Conclusions

In the present work, we improve the quantum-inspired neurons by exploiting the complex-valued weights to have richer representational capacity and better non-linearity. Then the basic idea of quantum-inspired neuron is extended naturally to perform the complex-valued convolutional operations. By employing the quantum-inspired neurons in the fully connected layers and/or the convolutional operations in the convolutional layers, we develop five types of quantum-inspired convolutional neuron networks (QICNNs). The detail mathematical framework to implement the QICNNs are developed, which can be executed based on the common real-valued framework of machine learning. The performances of the five QICNNs are studied using the MNIST and CIFAR-10 dataset. The results show that the QICNN using the quantum-inspired neuron in the fully connected layers has the best performance. The performance of classification accuracy, convergent rate and robustness is remarkably higher than the classical template CNN.

In the future, we will first optimize the QICNNs by improving the mathematical methods of implementing the complex-valued deep neural networks. We deem that more advantages of QICNNs can be revealed if we use a specialized learning framework that supports complex-valued parameters and operations. Secondly, we will find more learning tasks that QICNNs can outperform the classical CNNs, which are essential for the further applications of QICNNs. Finally, it is apparent, in principle, the quantum-inspired technologies proposed in this work can be used to implement other kinds of neural network, such as the SNN. We will study the way of applying the quantum-inspired technologies in the digital neuromorphic computing [24–28] to produce more inspiring algorithms.

Table 1 Classification accuracy on MNIST with different weight initializations

	II_QICNN	template CNN
MSRA	99.61%	99.46%
Truncated normal distribution	99.65%	99.50%
Random uniform distribution	99.33%	99.38%

Table 2 Classification accuracy on MNIST when rotating the input data with different angles

	II_QICNN	template CNN
0°	99.38%	99.30%
-15°~15°	99.65%	99.50%
-30°~30°	99.24%	99.30%

Acknowledgements The present work is supported by the National Natural Science Foundation of China (Grant No. 12005212, 6157518), the Natural Science Foundation of Shandong.

Province of China (ZR2021ZD19) and the Pilot National Laboratory for Marine Science and Technology (Qingdao).

Code availability The codes are available upon request from the authors.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Google AI Quantum and Collaborators (2019) Quantum supremacy using a programmable superconducting processor. Nature 574: 505–510
2. 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, Jiang X, Gan L, Yang G, You L et al (2020) Quantum computational advantage using photons. Science 370:1460–1463
3. Montanaro A (2016) Quantum algorithms: an overview. Npj quantum. Information 2:15023
4. Google AI Quantum and Collaborators (2020) Hartree-Fock on a superconducting qubit quantum computer. Science 369:1084–1089
5. Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, McClean JR, Mitarai K, Yuan X, Cincio L, Coles PJ (2021) Variational quantum algorithms. Nature reviews 3:625–644
6. Manju A, Nigam MJ (2014) Applications of quantum inspired computational intelligence: a survey. Artif Intell Rev 42:79–156
7. Boev AS, Rakitko AS, Usmanov SR, Kobzeva AN, Popov IV, Ilinsky VV, Kiktenko EO, Fedorov AK (2021) Genome assembly using quantum and quantum-inspired annealing. Sci Rep 11:13183. https://doi.org/10.1038/s41598-021-88321-5
8. Ding C, Bao T, Huang H (2021) Quantum-inspired support vector machine. In: IEEE Transactions on Neural Networks and Learning Systems https://doi.org/10.1109/TNNLS.2021.3084467, 1, 13
9. Mozaffaria A, Azada NL, Emamib M, Fathi A (2016) Mixed continuous/binary quantum-inspired learning system with nonnegative Least Square optimisation for automated design of regularised ensemble extreme learning machines. Journal of Experimental & Theoretical Artificial Intelligence 28(3):581–606
10. Narayanan A, Moore M (1996) Quantum-inspired genetic algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp 61–66. https://doi.org/10.1109/ICEC.1996.542334
11. Xiao J, Yan Y, Zhang J, Tang Y (2010) A quantum-inspired genetic algorithm for K-means clustering. Expert Syst Appl 37:4966–4973
12. Li PC, Li SY (2008) Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits. Neurocomputing 72(1):581–591
13. Kak SC (1995) Quantum neural computation. Advances in Imaging and Electron Physics, pp 259–313. https://doi.org/10.1016/0020-0255(94)00095-S
14. Matsui N, Takai M, Nishimura H (2000) A network model based on Qubitlike neuron corresponding to quantum circuit. Electronics and Communications in Japan 83(10):67–73
15. Mori K, Isokawa T, Kouda N, Matsui N, Nishimura H (2006) Qubit inspired neural network towards its practical applications. In: 2006 international joint conference. https://doi.org/10.1109/IJCNN.2006.1716095
16. Li PC, Li SY (2008) Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits. J Syst Eng Electron 19(1):167–174
17. Li PC, Xiao H, Shang FH, Tong X, Li X, Cao M (2013) A hybrid quantum-inspired neural networks with sequence inputs. Neurocomputing 117(6):81–90
18. Patel OP, Tiwari A (2014) Quantum inspired binary neural network algorithm. In:2014 international conference on information technology, pp 270–274. https://doi.org/10.1109/ICIT.2014.29
19. Ganjefar S, Tofighi M (2018) Optimization of quantum-inspired neural network using memetic algorithm for function approximation and chaotic time series prediction. Neurocomputing 291(24): 175–186
20. Yan P, Li L, Jin M, Zeng D (2021) Quantum probability-inspired graph neural network for document representation and classification. Neurocomputing 445:276–286
21. Li Y, Xiao J, Chen Y, Jiao L (2019) Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classification. Neurocomputing 362:156–165
22. Liu X, Zhu X, Li M, Wang L, Zhu E, Liu T, Kloft M, Shen D, Yin J, Gao W (2020) Multiple kernel k-means with incomplete kernels. IEEE Trans Pattern Anal Mach Intell 42(5):1191–1204
23. Sun Y, Zeng Y, Zhang T (2021) Quantum superposition inspired spiking neural network. 24:102880. https://doi.org/10.1016/j.isci.2021.102880
24. Yang S, Gao T, Wang J, Deng B, Lansdell B, Linares-Barranco B (2021) Efficient spike-driven learning with dendritic event-based processing. Front Neurosci 15. https://doi.org/10.3389/fnins.2021.601109
25. Yang S, Wang J, Deng B, Azghadi MR, Barranco BL (2021) Neuromorphic context-dependent learning framework with fault-tolerant spike routing. In: IEEE Trans Neural Netw Learn Syst https://doi.org/10.1109/TNNLS.2021.3084250, PP
26. Yang S, Wang J, Hao X, Li H, Wei X, Deng B, Loparo KA (2021) BiCoSS: toward large-scale cognition brain with multigranular neuromorphic architecture. In: IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2020.3045492, PP
27. Yang S, Deng B, Wang J, Li H, Lu M, Che Y, Wei X, Loparo KA (2020) Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. IEEE Trans Neural Netw Learn Syst 31(1):148–162. https://doi.org/10.1109/TNNLS.2019.2899936
28. Yang S, Wang J, Zhang N, Deng B, Pang Y, Azghadi MR (2021) CerebelluMorphic: large-scale neuromorphic model and architecture for supervised motor learning. In: IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2021.3057070, PP
29. Cong I, Choi S, Lukin MD (2019) Quantum convolutional neural networks. Nat Phys 15:1273–1278
30. Trabelsi C, Bilaniuk O, Zhang Y, Serdyuk D, Subramanian S, Santos JF, Mehri S, Rostamzadeh N, Bengio Y, Pal CJ (2018) Deep complex networks. In: ICLR 2018 conference. https://doi.org/10.48550/arXiv.1705.09792
31. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In:2017 international conference on engineering and technology (ICET), pp 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186
32. Kayed M, Anter A, Mohamed H (2020) Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture. In: 2020 international conference on innovative trends in communication and computer engineering (ITCE), pp 238–243. https://doi.org/10.1109/ITCE48509.2020.9047776
33. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: proceedings of the 25th international conference on neural information processing systems, pp 1097–1105. https://doi.org/10.1145/3065386

34. Chauhan R, Ghanshala KK, Joshi RC (2018) Convolutional neural network (CNN) for image detection and recognition. In: 2018 first international conference on secure cyber computing and communication (ICSCCC), pp 278–282. https://doi.org/10.1109/ICSCCC.2018.8703316
35. Lou Y, Cong S (2011) State transfer control of quantum systems on the bloch sphere. J Syst Sci Complex 24:506–518
36. LeCun Y, Cortes C, CBurges CJ (2008a) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/ - MNIST database. Accessed 16 Jan 2021
37. Krizhevsky A, Sutskever I (2013a) The CIFAR-10 dataset. https://www.cs.toronto.edu/~kriz/cifar.html-CIFAR-10 database. Accessed 14 Jul 2021
38. Yu X, Zhou Z, Gao Q, Li D, Riha K (2018) Infrared image segmentation using growing immune field and clone threshold. Infrared Phys Technol 88:184–193
39. Yu X, Ye X, Zhang S (2022) Floating pollutant image target extraction algorithm based on immune extremum region. Digital Signal Processing 123:103442

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

ShangshangShi is a PhD candidate of the Faculty of Information Science and Engineeringat Ocean University of China. Her research focuses on the quantum machine learning, especially the quantum deep learning.

ZhiminWang is an associateprofessor in the Facultyof Information Science and Engineering at Ocean University of China.He received his PhD from Peking University 2007 with a background inNuclear Science and Technology. Now he engages in the research ofquantum computing and quantum algorithms as well as neutron physics.

YongjianGu receiveda PhD from University of Science and Technology of China in 2004. Heis a professor in the Faculty of Information Science and Engineeringat Ocean University of China. His research interests are in the areasof quantum optics, marine optics and quantum information, especiallyquantum communication and quantum algorithms. Recent projects haveincluded underwater quantum key distribution and quantum DeepLearning. He has reviewed for the journals including Physical ReviewA, Physics Letters A and Chinese Physics Letters. He is a council member of Chinese Optical Society, and is co-author of over 100articles.