A Tutorial on Quantum Convolutional Neural Networks (QCNN)

°Seunghyeok Oh, †Jaeho Choi, and °Joongheon Kim
°School of Electrical Engineering, Korea University, Seoul, Republic of Korea
†School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea
E-mails: seunghyeokoh@korea.ac.kr, jaehochoi2019@gmail.com, joongheon@korea.ac.kr

Abstract—**Convolutional Neural Network (CNN) is a popular model in computer vision and has the advantage of making good use of the correlation information of data. However, CNN is challenging to learn efficiently if the given dimension of data or model becomes too large. Quantum Convolutional Neural Network (QCNN) provides a new solution to a problem to solve with CNN using a quantum computing environment, or a direction to improve the performance of an existing learning model. The first study to be introduced proposes a model to effectively solve the classification problem in quantum physics and chemistry by applying the structure of CNN to the quantum computing environment. The research also proposes the model that can be calculated with O(log(n)) depth using Multi-scale Entanglement Renormalization Ansatz (MERA). The second study introduces a method to improve the model's performance by adding a layer using quantum computing to the CNN learning model used in the existing computer vision. This model can also be used in small quantum computers, and a hybrid learning model can be designed by adding a quantum convolution layer to the CNN model or replacing it with a convolution layer. This paper also verifies whether the QCNN model is capable of efficient learning compared to CNN through training using the MNIST dataset through the TensorFlow Quantum platform.**

I. INTRODUCTION

Quantum computers are emerging as a new solution to problems not solved by classical computers. Quantum computers provide a computing environment that is different from classical computers. In particular, quantum computers can use superposition and entanglement, not seen in classical computing environments, and obtain powerful performance using parallelism between qubits [1]. Through these advantages, the quantum computer is considered new solutions to algorithmic problems that cannot be easily solved. Also, in the field of machine learning, various studies applying quantum computing models are in progress. There are Variational Quantum Eigensolver (VQE) [2], [3] and Quantum Approximate Optimization Algorithm (QAOA) [4], [5], which provide new ways to solve the problems of physics and chemistry in complex structures or NP-hard algorithm problems. Besides, as optimization using gradient descent method in quantum devices has been studied, learning quantum machine learning using hyperparameters can be performed efficiently [6], [7].

Convolutional Neural Network (CNN), among many classification models, has shown very high performance in computer vision [8]. Images that reflect the real world, such as photographs, have a very high correlation between surrounding pixels. The fully-connected layer, basic model in deep learning, showed strong performance in machine learning, but there is no way to keep the correlation. On the other hand, CNN can maintain correlation information directly, resulting in better performance evaluation.

CNN mainly proceeds by stacking the convolution layer and the pooling layer. The convolution layer finds new hidden data by linear combinations between surrounding pixels. The pooling layer reduces the size of the feature map, reducing the resources required for learning and avoiding overfitting. When the data size is sufficiently reduced by repeatedly applying these layers, the classification result is obtained using the fully connected layer. The loss between the acquired label and the actual label can train the model using a gradient descent method or other optimizers for better results.

However, many problems that exist in the real world are still hard to solve with classic machine learning methods. The quantum physics problem defined in the many-body Hilbert space requires converting these data into classical computer data to apply machine learning techniques. As the size of the system increases, the size of the data increases exponentially, making it difficult to solve effectively even with a machine learning method. In addition to the above case, other alternatives when data and models are no longer efficiently processed in existing computing environments.

Many studies have appeared to solve these problems with the Quantum Convolutional Neural Network (QCNN) using the quantum computing system and the CNN model together. There is an approach to apply the CNN structure itself to a quantum system to efficiently solve quantum physics problems and an approach to improve performance by adding a quantum system to problems previously solved by CNN. This paper introduces these studies.

II. QCNN USING THE STRUCTURE OF CNN

QCNN proposed by *Cong* extends the main features and structures of the existing CNN to quantum systems [9]. In moving the quantum physics problem defined in the many-body Hilbert space to the classical computing environment, the data size exponentially increases according to the system size, so it is not suitable to solve efficiently. In a quantum environment, the data can express using qubits, so the problem can be avoided by applying a CNN structure to a quantum
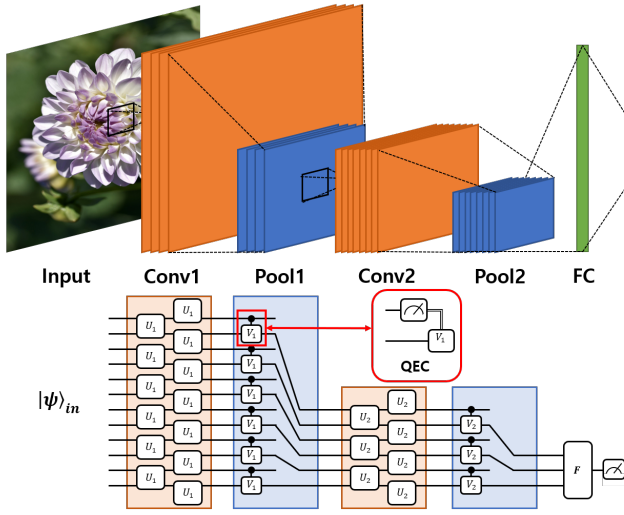
Fig. 1. Simple example of CNN and QCNN. QCNN, like CNN, consists of a convolution layer that finds a new state and a pooling layer that reduces the size of the system. QEC can be additionally applied by using measurement instead of controlled-gate in the pooling layer [9].
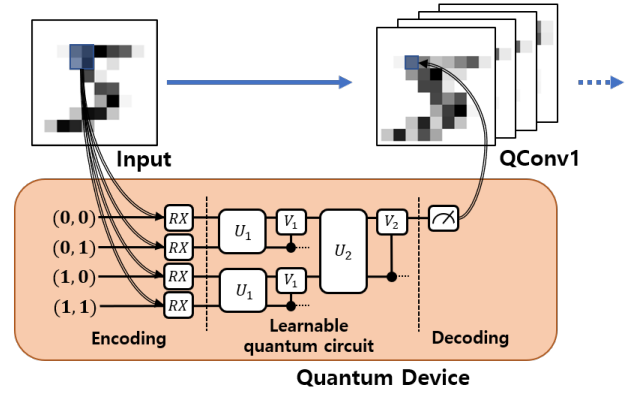

Fig. 2. Example of quantum convolution layer for image classification. The learnable quantum circuit can choose in various methods that can enhance performance.

computer. This section briefly introduces the structural design of this QCNN.

The model of QCNN applies the convolution layer and the pooling layer, which are the main features of CNN, to quantum systems, as shown in Fig. 1. The concept proceeds as follows:

1) The convolution circuit finds the hidden state by applying multiple qubit gates between adjacent qubits.
2) The pooling circuit reduces the size of the quantum system by observing the fraction of qubits or applying 2-qubit gates such as CNOT gates.
3) Repeat the convolution circuit and pooling circuit defined in 1)-2).
4) When the size of the system is sufficiently small, the fully connected circuit predicts the classification result.

The model used to satisfy this structure is typically Multiscale Entanglement Renormalization Ansatz (MERA) [10]. MERA is a model designed to simulate many-body state quantum systems efficiently. At this time, MERA exponentially increases the size of the quantum system for each depth by adding qubits of $|0\rangle$. QCNN uses this MERA in the reverse direction. The reversed MERA reduces the size of the quantum system exponentially from the given data, which is suitable as a model of QCNN.

The QCNN model proposed by *Cong* suggests an additional performance improvement through the Quantum Error Correction (QEC) to this MERA model [11]. There is a representative state $|\psi\rangle$ for each label in MERA. Since QCNN uses the reverse direction of MERA, if its $|\psi\rangle$ is given as input data, the corresponding label can be obtained as a definitive solution. On the other hand, if $|\psi'\rangle$ that cannot be generated in MERA is given as input data, QCNN cannot obtain a definitive solution. This problem can be corrected and solved by applying QEC to give additional degrees of freedom.

When the data given to QCNN is $|\psi\rangle$, the result measured in the pooling layer should be $|0\rangle$ the same as the newly given state in MERA. On the other hand, if $|\psi'\rangle$ which MERA cannot generate is given as input data, there is a possibility that it will be $|1\rangle$ in the measured result. Using them, if $|1\rangle$ is measured, an additional gate is applied to the surrounding qubits to correct the result. The method can give better performance through additional deterministic measurement outcomes.

III. QCNN FOR IMAGE CLASSIFICATION

Image classification is one of the most applied fields in neural networks, such as CNN. Quantum computers have potent advantages in terms of superposition and parallel computation. Quantum Convolutional Neural Network proposed by *Henderson* applies quantum environments in CNN to improve the performance of CNN [12]. This section briefly introduces the research that suggested how to apply a quantum computing system to CNN.

The quantum convolution layer defines a layer that behaves like a convolution layer in a quantum system. The quantum convolution layer applies a filter to the input feature map to obtain feature maps composed of new data. However, the quantum convolution layer uses a quantum computing environment for filter operation, unlike the convolution layer.

Quantum computers have the advantages of superposition and parallel computation that do not exist in classical computing, which can reduce the learning time and evaluation time. However, existing quantum computers are still limited to small quantum systems. The quantum convolution layer does not apply the entire image map to a quantum system at once, but processes it as much as the filter size at a time, so small quantum computers can construct the quantum convolution layer.

The quantum convolution layer can construct, as shown in Fig. 2. The concept proceeds as follows:

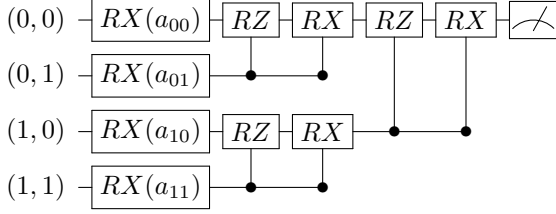1) The encoding process stores the pixel data corresponding to the filter size in qubits.

Fig. 3. Circuit example of a simple quantum convolution layer used for learning of the MNIST dataset. The learnable hyperparameters in variable gates optimize the model.

2) The learnable quantum circuits apply the filters that can find the hidden state from the input state.
3) The decoding process gets new classical data by measurement.
4) Repeat steps 1) to 3) to complete the new feature map.

The encoding process of 1) is a process necessary to convert classical information into quantum information. The simplest method is to apply a rotation gate corresponding to a pixel data to qubits. Of course, various encoding methods are possible, and the selected encoding method can change the number of qubits required and the learning efficiency. The decoding process of 3) is determined according to measuring one or more quantum states. By measuring quantum states, classical data are determined.

The random quantum circuit in 2) can be made from a combination of multiple gates. Also, the circuit can perform optimization using the gradient descent method by adding variable gates. This circuit can be designed in various ways that can affect learning performance depending on the design method. When using MERA, the classical environment generally requires $O(n^2)$ operations in an n^2-sized filter, but in a quantum system, the parallelism of qubits can design the filters with $O(log(n))$ depths.

IV. LEARNING MNIST USING QCNN SIMULATION

In this section, simulations are performed to verify that the actual quantum convolutional neural network works properly in image classification using the MNIST dataset [13]. QCNN's quantum computing simulation used the TensorFlow Quantum platform [14]. However, because the quantum computing simulation environment uses many resources, it has the following limitations.

- The 28×28 size MNIST dataset was downscaled to 10×10 size.
- The filter size of the quantum convolution layer was limited to 2×2.
- In each epoch, 2500 random images out of 60,000 are selected for learning.

To evaluate the performance of QCNN, fully-connected, CNN, and QCNN models are defined as follows:

- QCNN model: The quantum convolution layer consists of a quantum circuit defined by Fig.3. The quantum convolution layer returns a feature map with 8 channels. The returned feature map predicts classification results by
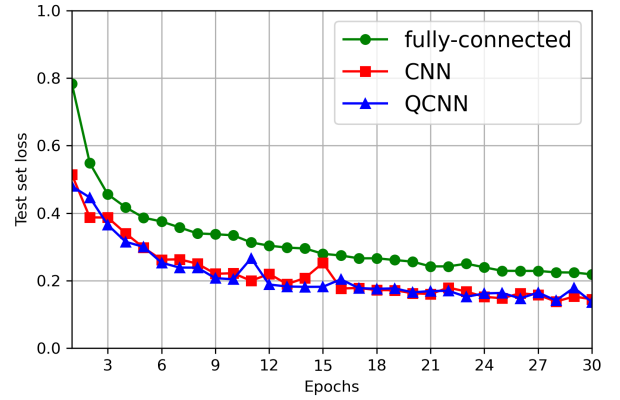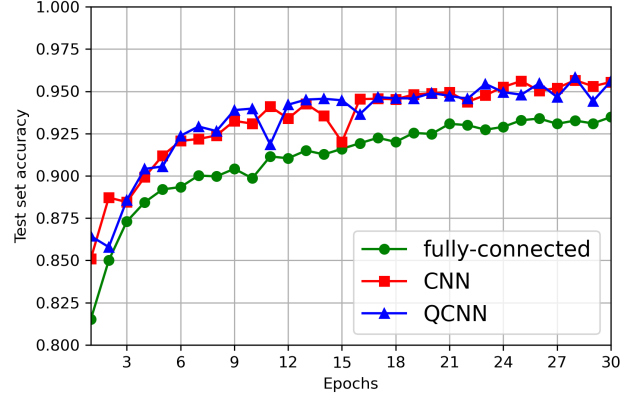

Fig. 4. The performance of the QCNN model compared to fully-connected layers only model, and CNN model.

connecting the feature map with a fully-connected layer comprising 64 and 10 hidden units.
- fully-connected model: Construct the model using only the fully-connected layer to check whether the quantum convolution layer affects learning.
- CNN model: The convolution layer that returns the feature map of the same channel length replaces the quantum convolution layer to compare the performance difference.

Fig.4 shows the results of this simulation. First, the QCNN model always shows good learning results compared to the fully-connected layer. It confirmed that the quantum convolution layer could improve learning performance. Second, in the comparison between the QCNN model and the CNN model, it can be seen that similar learning results appear. In other words, the QCNN model can have the same learning performance as the CNN model.

V. CONCLUDING REMARKS AND FUTURE WORK

QCNN uses a CNN model and a quantum computing environment to enable various approaches. The QCNN model can be a solution in the field of physical and chemical classification not solved simply, and in addition to the existing CNN model, it can be a more effective and efficient learning model method. Furthermore, in the quantum computer of the NISQ era,

the QCNN model can expect more efficient and high-level results in more complex and large-scale learning [15]. In this experiment's results, the simulations were performed at the microscopic scale, but we plan to apply the QCNN model to more complex data.

QCNN is available in more detailed implementation and approaches. Depending on how the internal quantum circuit is designed, the performance evaluation of the learning model can be improved. Besides, when applying the QCNN model to the field of imaging processing, by including more information in one qubit according to an encoding method, much more efficient learning may be possible. In the future, we will study the QCNN model that is more efficient and has better learning performance through simulation using various approaches.

ACKNOWLEDGMENT

REFERENCES

[1] S. Bravyi, D. Gosset, and R. König, "Quantum advantage with shallow circuits," *Science*, vol. 362, no. 6412, pp. 308–311, 2018.

[2] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, p. 4213, 2014.

[3] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023023, Feb 2016. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/18/2/023023

[4] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[5] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, 2019.

[6] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019.

[7] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, "Quantum natural gradient," *Quantum*, vol. 4, p. 269, 2020.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[9] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[10] G. Vidal, "Class of quantum many-body states that can be efficiently simulated," *Physical review letters*, vol. 101, no. 11, p. 110501, 2008.

[11] J. Preskill, "Lecture notes for physics 229: Quantum information and computation," *California Institute of Technology*, vol. 16, 1998.

[12] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quanvolutional neural networks: powering image recognition with quantum circuits," *Quantum Machine Intelligence*, vol. 2, no. 1, pp. 1–9, 2020.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters *et al.*, "Tensorflow quantum: A software framework for quantum machine learning," *arXiv preprint arXiv:2003.02989*, 2020.

[15] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.