https://doi.org/10.7840/kics.2024.49.2.254

Effect of Gate Variations on MNIST Classification in QNN: Experimental Study and Analysis

Seok Bin Son◆, Hankyul Baek*, Soohyun Park°

ABSTRACT

Recently, QNNs have emerged as a promising method for big data processing because they can process data at high speeds while using fewer parameters than existing machine learning algorithms. In the structure of QNN, the performance is determined by which quantum gates are used in the PQC, which constitutes the quantum circuit. Therefore, we conduct a performance comparison experiment in this paper using various quantum gates in QNNs. Through four comparison experiments using the MNIST dataset, we confirm that performance differences occur depending on the use of gates in QNNs and discuss the reasons for the performance differences. Therefore, in this paper, we verify that the performance of QNNs varies depending on the quantum gate variation.

Key Words : QNN, Gate variation, Quantum gate, PQC, MNIST

Ⅰ. Introduction

Recently, there has been an exponential increase in the volume and diversity of data, prompting the exploration of various strategies for efficiently handling these substantial datasets. Based on this situation, various methods have been proposed to process large amounts of data efficiently. Among these approaches, one notable technique involves the introduction of quantum neural networks (QNNs)[1]. The development of modern machine learning (ML) methods[2] and the appearance of noisy intermediate scale quantum (NISQ) computing processors[3] have driven this trend. QNN has shown competitive performance comparable to classical neural networks while using fewer parameters. Notably, QNNs have an outstanding ability to process large datasets with remarkable effectiveness, making them an attractive option for addressing the problems posed by big data. The effectiveness of QNN has been verified across a spectrum of applications, including data generation[4], classification[5,6], distributed learning[7,8], and reinforcement learning[9-13], thereby highlighting their remarkable utility.

Our considering QNN uses the concept of qubits instead of the classic bits considered in classical computers. Based on quantum mechanisms such as superposition, which allows qubits to have information of both 0 and 1 at the same time, and entanglement, which is the interdependence of qubits in different states. A QNN is composed of three fundamental components: encoding, parameterized quantum circuit (PQC)[14,15], and measurement, as illustrated in Fig. 1. Traditional classical data, such as MNIST data, must be transformed into a format that can be used in quantum circuits before it can be used directly in QNNs. The *encoding* layer accomplishes this. The input data that have been

※ Enter acknowledgment here: This research was supported by the National Research Foundation of Korea (2022R1A2C2004869).
◆ First Author : Korea University Department of Electrical and Computer Engineering, lydiasb@korea.ac.kr, 학생회원
° Corresponding Author : Korea University Department of Electrical and Computer Engineering, soohyun828@korea.ac.kr, 학생회원
* Korea University Department of Electrical and Computer Engineering, 67back@korea.ac.kr, 학생회원
논문번호 : 202308-054-C-RN, Received August 21, 2023; Revised November 2, 2023; Accepted November 7, 2023

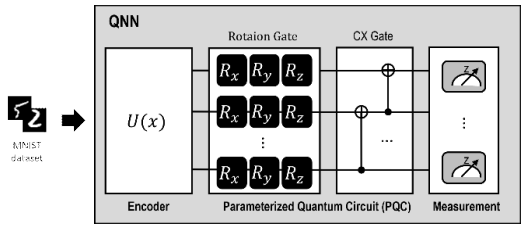


Fig. 1. Structure of QNN

transformed to qubits enter the *PQC* following the encoding layer. In terms of multiplication between hidden layers, PQC and traditional deep neural networks (DNN) are similar. PQC commonly consists of qubit rotation gates and controlled-X (CX) gates (especially, controlled-NOT (CNOT) gate) which are adjustable and fixed gates, respectively. The CX gates generate entanglement computation, which affects the performance of the QNN-based learning algorithms. Finally, the quantum states generated from the PQC are determined by *measurement* layer. Therefore, in this paper, we focus on using different types of quantum gates used in PQCs. We use the MNIST dataset to analyze the performance differences of various gates in classification.

The rest of this paper is organized as follows. Section II introduces the preliminaries of quantum. In addition, section III presents our approach. Moreover, section V discusses the result of our approach. Finally, section VI concludes this paper and presents future research directions.

Ⅱ. Preliminaries

2.1 Quantum supremacy

Quantum computers can solve challenges beyond their classical capabilities or execute noticeably more quickly when they reach the state of quantum supremacy. In October 2019, Google announced that Sycamore, a 53-qubit quantum computer, had demonstrated establishing quantum supremacy by finishing a job in 200 seconds that would have taken a supercomputer almost 10,000 years[3]. This practical application demonstrates the dependability and future development of quantum computing, subject to algorithm and hardware innovation. Qubits, as opposed to conventional bits, are the key of quantum computing. Qubits can simultaneously represent 0 and 1 thanks to probability amplitudes, increasing data density. They can represent 2^n different states with n qubits, allowing for parallelized computations and data processing with fewer input parameters than with bits.

First of all, a quantum Generative Adversarial Networks (qGAN) is designed to integrate quantum algorithms like Quantum Amplitude Estimation into Generative Adversarial Network (GAN) architectures. By utilizing qGAN, the gate complexity needed for state loading can be significantly reduced[4]. In order to demonstrate the potential of quantum computing for effective classification tasks, two quantum algorithms have been proposed that use the quantum state space as a feature space for binary classification[6]. Additionally, a novel Slimmable Quantum Federated Learning (SQFL) framework is also proposed that makes use of QNN to adjust for altering communication conditions, demonstrating improved classification accuracy and potential benefits of quantum computing in distributed learning[8]. Lastly, a quantum-based multi-agent reinforcement learning (QMARL) approach for effective multi-user cooperation and coordination in autonomous mobility systems[9].

2.2 Quantum computing

In contrast to classical computing, a qubit is an information unit representing the quantum state using a dual basis, including $|0\rangle$ and $|1\rangle$|[16]. The quantum state of a q-qubit system is represented by the probability amplitudes of the 2^q possible bases, indicated as $|\psi\rangle$, a basic notion in quantum mechanics. It may be expressed mathematically as an equation 1.

$$|\psi\rangle \triangleq \sum_{k=1}^{2^q} \alpha_k |k\rangle \quad (1)$$

where $|k\rangle$ denotes Hilbert space's basis, $\sum_{k=1}^{2^q}|\alpha_k|^2 = 1$, and $\forall_q \in N\,[1, \infty)$

It defines a superposition property of QNN where a qubit can probabilistically denote the states of 0 and 1. In other words, the states of 0 and 1 are overlapped

before we measure a single qubit, and this is shown by the symbol $\alpha|0\rangle + \beta|1\rangle$ However, following measurement, it stochastically collapses to 0 or 1 depending on the values of $|\alpha|^2$ and $|\beta|^2$. Because of the superposition, many inputs are processed when a quantum operation is performed on the qubit. Qubits can express many more states as a result than can traditional classic bits.

2.3 Structure of QNN

Traditional classical input data cannot be directly processed by the quantum circuitry of a QNN. Therefore, in order to be integrated into QNNs, traditional classical data must be transformed into a quantum state by encoding. This process of encoding not only facilitates data transformation but also effectively addresses the constraints imposed by the limitations of Noisy Intermediate Scale Quantum (NISQ) devices. The encoding method converts classical bit information into quantum states while preserving the essence of raw data within the initial quantum states, thus paving the way for subsequent quantum processing.

The PQC plays a crucial role by creating entanglement between the qubits and iteratively growing trainable parameters. PQC consists of quantum gates that execute quantum operations. Quantum gates are represented through unitary matrices, orchestrating the evolution of qubit states, and enabling their manipulation and transformation. The composition of these quantum gates encompasses both adjustable rotational and fixed gates (i.e., CX gates). Within the PQC framework, the encoded quantum state undergoes processing, whereby qubits become strongly entangled through the employment of CX gates. Thus, depending on the type of gate, qubits are transformed by performing entanglement, disentanglement, and rotation.

The measurement procedure ends with creating the ultimate output generation. Before measurement, the quantum state remains in a superposed and, therefore, an inherently unstable state. It is through the act of measurement that this instability is resolved, effectuating the conversion of the quantum state into classical data.

2.4 Quantum gate

A quantum gate is a quantum computing process that manipulates and transforms quantum states. To modify the quantum state of qubits, quantum gates are used. Quantum gates function similarly to logic gates in classical computers, but quantum mechanics determines their operation. The unitary matrix is used to represent these quantum gates[17]. There are various types of gates, the specific gates that will be used in this paper are as follows.

The RX gate is a rotation gate for one qubit used in quantum computing. This gate is used to manipulate and control quantum bits by performing a rotation about the X-axis. When the RX gate is applied, the input qubit is rotated about its center around the X-axis according to a given angle. The angle of rotation is determined by the parameters applied to the gate.

$$RX(\theta) = \exp\left(-i\frac{\theta}{2}X\right) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -i\ sin(\frac{\theta}{2}) \\ -i\ sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix} \quad (2)$$

where θ denotes the constant values of the rotation angle of RX gate.

In addition, the RY gate is a rotation gate for one qubit used in quantum computing. This gate is used to manipulate and control quantum bits by performing a rotation about the Y-axis. The RY gate has one parameter, θ, which determines the angle of rotation. Depending on the rotation angle θ, the state of a quantum bit can be changed, which can be used to construct various quantum algorithms. The RY gate can be combined with other rotational gates to form complex circuits.

$$RY(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\ sin(\frac{\theta}{2}) \\ sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix} \quad (3)$$

where θ denotes the constant values of the rotation angle of RY gate.

Moreover, the RZ gate is a rotation gate for one qubit used in quantum computing. This gate has a single parameter, and depending on that parameter,

it rotates the qubit about the Z-axis by a given angle. RZ gates are used to control and manipulate quantum information. When an RZ gate is applied, the input qubit is rotated around the Z-axis by a given angle. The angle of rotation is determined by the parameters applied to the gate.

$$RZ(\theta) = \exp\left(-i\frac{\theta}{2}Z\right) = \begin{bmatrix} \exp(-i\frac{\theta}{2}) & 0 \\ 0 & \exp(i\frac{\theta}{2}) \end{bmatrix} \quad (4)$$

where θ denotes the constant values of the rotation angle of RZ gate.

Finally, the CX gate (i.e., controlled-X gate or CNOT gate) has two qubits and consists of a control and a target bit. Depending on the state of the control bit, it performs an X-gate action on the target bit. In addition, the CX gate entangles multiple qubits by executing the XOR operation on two qubits. For example, if the control bit is 1 and the target bit is in the $|0\rangle$ state, the target bit changes to the $|1\rangle$ state. In addition, if the control bit is 1 and the target bit is in the $|1\rangle$ state, the target bit changes to the $|0\rangle$ state. Finally, if the control bit is 0, nothing happens to the target bit.

$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

2.5 Entanglement

As qubits pass through the PQC and undergo quantum operations, they become entangled with each other. Entanglement signifies a strong correlation between two or more qubits, wherein the quantum states of these qubits are interdependent. In addition, describing the state of one qubit independently from the others becomes infeasible. Consequently, the degree of entanglement depends on the selection of quantum gates, which can exert a notable influence on the performance of the quantum system. Mathematically, when two states are inseparable and unfactorizable, the state is called entangled. For example, (6) could be factorized. However, (7) could not be factorized, so this matrix is entangled. Qubits may become entangled with CX gates in a system with numerous qubits.

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \quad (7)$$

Ⅲ. QNN for MNIST Classification with Differential Gates

We select RX, RY, RZ gates which are typical rotation gates, and the CX gate which creates entanglement in PQC as shown in Fig. 1. We use a set of gates consisting of the six selected gates: RXRYRZ, RYRYRY, RYRYCX, RYRYRYCX, RYRYCXCX, and RXRYRZCX gate. In addition, by using these gates, we conduct four experiments to verify the effect of gate variations.

3.1 Quantum gate variation

Firstly, the RXRYRZ gate is constructed through a sequential arrangement of three constituent gates, such as the RX gate, the RY gate, and the RZ gate. The RXRYRZ gate rotates the quantum state in different directions (i.e., X-axis, Y-axis, and Z-axis). In addition, the RYRYRY gate is constructed through the sequential arrangement of RY gates, consisting of a series of three RY gates applied in succession. Each RY gate performs a rotation about the Y-axis of the Bloch sphere, and when combined in sequence, these rotations result in a collective transformation of the quantum state of the involved qubits. Moreover, the RYRYCX gate and RYRYRYCX are constructed by sequentially arranging RY gates and a CX gate, consisting of two or three RY gates followed by a CX gate applied in sequence, respectively. This configuration enables the controlled manipulation of quantum states through rotations and entanglement generation. Furthermore, the RYRYCXCX gate is constructed through the sequential arrangement of RY

gates and CX gates, comprising a series of two RY gates followed by two CX gates applied in sequence. This arrangement facilitates controlled rotations and the establishment of entanglement patterns, contributing to the controlled evolution of quantum states. Finally, RXRYRZCX gate is constructed through the RX gate, the RY gate, the RZ gate, and CX gate. This allows for more complex quantum state transformations by rotating and entangling in different directions.

3.2 Four comparison

Firstly, comparison 1 consists of the comparison between RXRYRZ and RYRYRY gate. We compare the performance of these gates to verify the performance difference between rotating gates. Moreover, comparison 2 consists of a comparison of the RYRYCX and RYRYRY gates. By comparing the performance of these two gates, we can see the difference in performance between using only rotating gates and using a combination of rotating gates and CX when the number of gates is the same. Furthermore, comparison 3 consists of the comparison between RXRYRZCX and RYRYRYCX gate. We compare the performance of these gates to identify performance differences due to the use of CX gates and the diverse rotation directions. Finally, comparison 4 consists of the comparison between RYRYCXCX and RYRYCX gate. We compare the performance of these gates to identify performance differences due to the number of CX gates.

Table 1. The result of average accuracy of gate variation.

Gate	RYRYRY	RYRYCX	RXRYRZ
Accuracy (%)	39.31	40.47	41.36
Gate	RYRYRYCX	RYRYCXCX	RXRYRZCX
Accuracy (%)	44.79	48.00	54.56

Ⅳ. Results with four comparisons of experiment

This paper considers four comparisons between quantum gates for evaluating the performance effect of gate variations in QML. The experiments were performed out using 3.8.10 Python and 2.0.0 PyTorch, the Adam optimizer, 5×10^{-3} learning rate, 256 batch size, and 100 epochs. Our proposed comparisons consist of four parts, i.e., (i) Comparison 1 (refer to Section 4.1), (ii) Comparison 2 (refer to Section 4.2), (iii) Comparison 3 (refer to Section 4.3), and (iv) Comparison 4 (refer to Section 4.4), respectively.

4.1 Comparison 1

In this section, we compare the performance of the RXRYRZ gate and the RYRYRY gate. As shown in Table I, the average accuracy of RXRYRZ and RYRYRY are 41.36% and 39.31%. Therefore, the RXRYRZ gate outperforms the RYRYRY gate. Fig. 2 (a) shows the experiment result of comparison 1. Specifically, Fig. 2 (a) shows that the RYRYRY gate exhibits a shifting pattern, and initially displays better performance than the RXRYRZ gate until the 20 epochs. Beyond this epoch, however, there is a noticeable change, with the RXRYRZ gate continually exhibiting higher accuracy levels than the RYRYRY gate. In addition, Fig. 2 (b) also shows that the RYRYRY gate exhibits a shifting pattern until 25 epochs. Among this epoch, the loss convergence of RXRYRZ gate is comparable to RYRYRY gate.

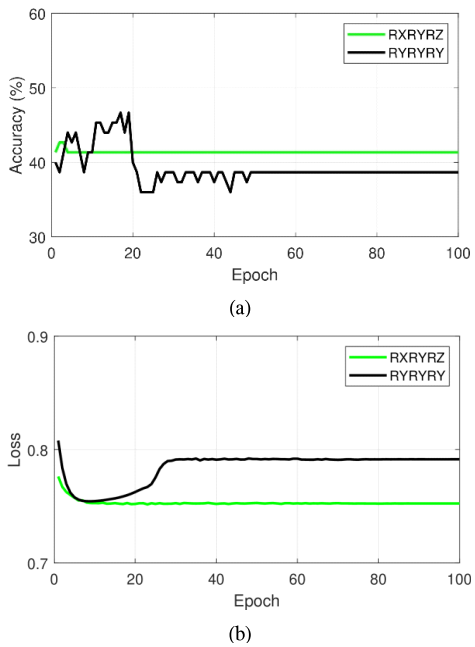
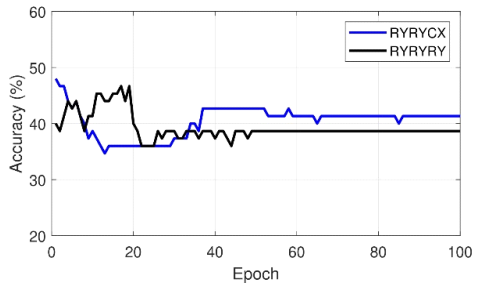


Fig. 2. Experiment result of comparison 1: (a) Accuracy, (b) Loss.

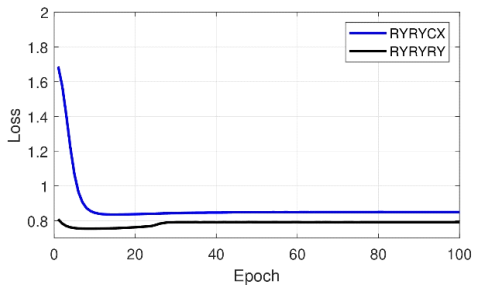
However, after 25 epochs, RXRYRZ gate has lower loss convergence than RYRYRY gate.

4.2 Comparison 2

In addition, in this section, we also compare the performance of the RYRYCX gate and the RYRYRY gate. As shown in Table I, the average accuracy of RYRYCX and RYRYRY are 40.47% and 39.31%, respectively. Therefore, the RYRYCX gate outperforms the RYRYRY gate. Fig. 3 (a) shows the experiment result of comparison 2. Especially, as depicted in Fig. 3 (a), until the 35 epochs, the performance trajectory of the RYRYCX and RYRYRY gates exhibited fluctuations and variability. Beyond this epoch, however, there is a noticeable change, with the RYRYCX gate continually showing higher accuracy levels than the RYRYRY gate. In addition, as shown in Fig. 3 (b), the RYRYCX gate can see a rapid convergence of loss. However, it seems to be less convergent than the RYRYRY gate.



(a)

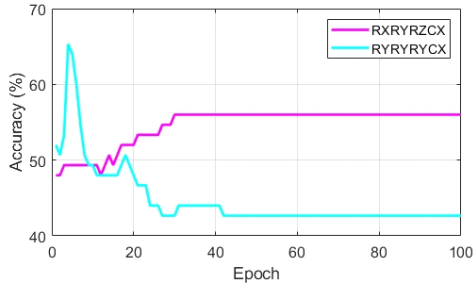


(b)

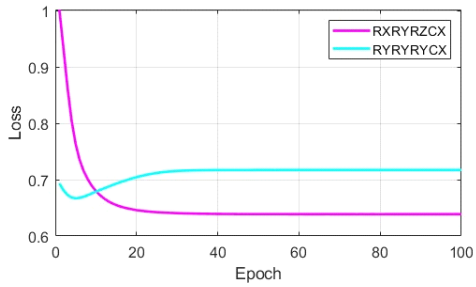
Fig. 3. Experiment result of comparison 2: (a) Accuracy, (b) Loss.

4.3 Comparison 3

Moreover, in this section, we also compare the performance of the RXRYRZCX gate and the RYRYRYCX gate. As shown in Table I, the average accuracy of RYRYCX and RYRYRY are 54.56% and 44.79%, respectively. Therefore, the RXRYRZCX gate outperforms the RYRYRYCX gate. Fig. 4 (a) shows the experiment result of comparison 3. Especially, as depicted in Fig. 4 (a), RYRYRYCX gate initially displays better performance than the RXRYRZCX gate until the 10 epochs. Beyond this epoch, however, there is a noticeable change, with the RXRYRZCX gate continually showing higher accuracy levels than the RYRYRYCX gate. In addition, as shown in Fig. 4 (b), the RXRYRZCX gate can see a rapid convergence of loss, but still shows less convergent than the RYRYRYCX gate until the 10 epochs. However, beyond this epoch, there is noticeable change, which shows that RYRYRYCX gate seems to be less convergent than the RXRYRZCX gate.



(a)



(b)

Fig. 4. Experiment result of comparison 3: (a) Accuracy, (b) Loss.

4.4 Comparison 4

Furthermore, in this section, we also compare the performance of the RYRYCXCX gate and the RYRYCX gate. As shown in Table I, the average accuracy values of RYRYCXCX and RYRYCX are 48.00%, and 40.47%, respectively. Therefore, the

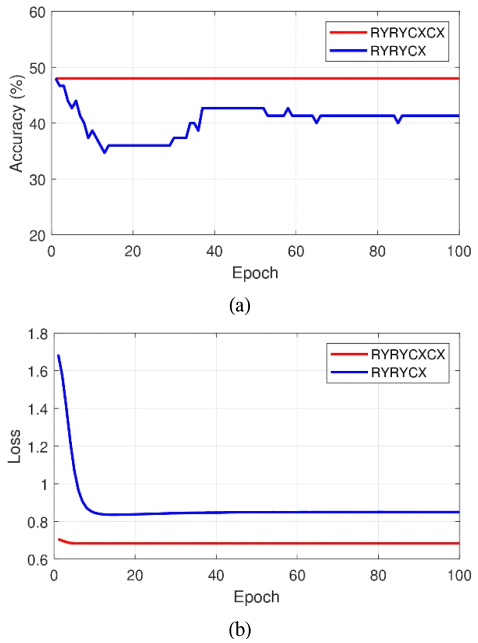


Fig. 5. Experiment result of comparison 2: (a) Accuracy, (b) Loss.

RYRYCXCX gate outperforms the RYRYCX gate. Fig. 5 (a) shows the experiment result of comparison 4. In particular, the RYRYCXCX gate outperforms the RYRYRY gate at every epoch, as shown in Fig. 5 (a). In the case of RYRYCXCX of Fig. 5 (a), the results reach saturation from the very early epoch. However, as the epoch progresses, you can see that the degraded performance from the initial accuracy is achieved. However, in the RYRYCXCX gate, where CX is added to RYRYCX, the significant results are maintained in the initially achieved performance until the end. In addition, as shown in Fig. 5 (b), the RYRYCX gate can see a rapid convergence of loss. However, it seems to be less convergent than the RYRYCXCX gate.

V. Discussion

In this section, we discuss the result of the experiment of comparison 1, 2, 3, and 4. The more detail of discussion is as follows.

5.1 Discussion of comparison 1

As shown in Table I and Fig. 2, the RXRYRZ gate outperforms the RYRYRY gate. Since each rotating gate of RXRYRZ gate rotates the qubit state around a different axis, RXRYRZ gate can perform various state transformations. This is expected to allow for more flexible quantum computation. The RYRYRY gate, on the other hand, consists of successive applications of the RY gate. In this case, it is limited to successively rotating the state of the qubit around the Y-axis. Since the number of axes that can be rotated is limited, the versatility of quantum operations can be relatively limited. Therefore, it is thought that the RXRYRZ gate may perform better than the RYRYRY gate because it allows for a wider variety of state transformations, giving quantum operations more flexibility.

5.2 Discussion of comparison 2

As shown in Table I and Fig. 3, the RYRYCX gate outperforms the RYRYRY gate. From these results, we can see that for the same number of gates, the combination of rotating gates and CX outperforms rotating gates alone. The RYRYCX gate consists of the application of the CX gate in addition to the rotation gate that continues to rotate in one direction only. The CX gate represents an interaction between two quantum bits, where the state of one quantum bit applies a rotation to the other. This interaction can be made complex, resulting in the formation of quantum entanglement between the quantum bits. Quantum entanglement represents the interdependence between quantum states and is a very powerful tool in quantum computation. Entangled qubits are strongly connected to each other, and a measurement of one qubit affects the state of the other qubits. It makes it possible to perform complex quantum computations. Therefore, the RYRYCX gate performs better than the RYRYRY gate because the CX gate allows quantum entanglement to form, allowing for more complex and powerful quantum interactions.

5.3 Discussion of comparison 3

As shown in Table I and Fig. 4, the RXRYRZCX gate outperforms the RYRYRYCX gate. These results show that for the same number of gates, combinations of gates rotated in multiple directions and

combinations of gates with CX outperform combinations of gates with a single direction of rotation and CX. The RXRYRZCX gate consists of a rotation gate that continues to rotate in various directions, plus a CX gate. This interaction is more complex and can lead to the formation of quantum entanglement between quantum bits, which is why the RXRYRZCX gate outperforms the RYRYRCX gate.

5.4 Discussion of comparison 4

As shown in Table I and Fig. 5, the RYRYCXCX gate outperforms the RYRYCX gate. While the RYRYCX gate already uses one CX gate, the RYRYCXCX gate uses two consecutive CX gates. It further complicates the interactions between the quantum bits and strengthens the quantum entanglement. The two CX gate operations make the quantum bits physically more strongly connected and interacting, which allows for more complex quantum states to form. Therefore, the RYRYCXCX gate performs better than the RYRYCX gate because it contains more CX gates, strengthening the quantum entanglement between the quantum bits and making their interactions more complex.

Ⅵ. Conclusions and future work

This paper presents an introduced and assessed experimental investigation involving gate diversification to ascertain the pivotal quantum gate within the array of options offered by PQCs, thereby enhancing the efficacy of QNNs. Through four comprehensive comparative analyses encompassing six quantum gate variations, it has been discerned that heightened gate complexity correlates with increased employment of using various rotation gates and CX gates, subsequently leading to strengthened entanglement and consequent performance enhancement.

In future work, we will conduct experiments with more diverse gates. In addition, we will proceed with improved performance through more sophisticated gate combinations. Moreover, we will analyze the strengths and weaknesses of related work and conduct studies to improve them. Furthermore, we will expand the validated understanding of quantum gate performance to encompass a wider range of quantum applications, including QFL and other diverse fields that rely on QNN.

References

[1] R. Huang, X. Tan, and Q. Xu, “Learning to learn variational quantum algorithm,” *IEEE Trans. Neural Netw. and Learn. Syst.*, pp. 1-11, 2022. (https://doi.org/10.1109/TNNLS.2022.3151127)

[2] N. Burkart and M. F. Huber, “A survey on the explainability of supervised machine learning,” *J. Artificial Intell. Res.*, vol. 70, pp. 245-317, 2021. (https://doi.org/10.1613/jair.1.12228)

[3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al., “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505-510, 2019. (https://doi.org/10.1038/s41586-019-1666-5)

[4] C. Zoufal, A. Lucchi, and S. Woerner, “Quantum generative adversarial networks for learning and loading random distributions,” *npj Quantum Inf.*, vol. 5, no. 1, p. 103, 2019. (https://doi.org/10.1038/s41534-019-0223-2)

[5] M. Schuld, “Supervised quantum machine learning models are kernel methods,” *arXiv preprint arXiv:2101.11020*, 2021. (https://doi.org/10.48550/arXiv.2101.11020)

[6] V. Havlicek, A. D. Corcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209-212, 2019. (https://doi.org/10.1038/s41586-019-0980-2)

[7] Y. Kwak, W. J. Yun, J. P. Kim, H. Cho, J. Park, M. Choi, S. Jung, and J. Kim, “Quantum distributed deep learning architectures: Models, discussions, and applications,” *ICT Express*, vol. 9, no. 3, pp. 486-491, 2023.

(https://doi.org/10.1016/j.icte.2022.08.004)

[8] W. J. Yun, J. P. Kim, S. Jung, J. Park, M. Bennis, and J. Kim, "Slimmable quantum federated learning," *arXiv preprint arXiv: 2207.10221*, 2022. (https://doi.org/10.48550/arXiv.2207.10221)

[9] S. Park, J. P. Kim, C. Park, S. Jung, and J. Kim, "Quantum multi-agent reinforcement learning for autonomous mobility cooperation," *IEEE Commun. Mag.*, pp. 1-7, Aug. 2023. (https://doi.org/10.1109/MCOM.020.2300199)

[10] C. Park, W. J. Yun, J. P. Kim, T. K. Rodrigues, S. Park, S. Jung, and J. Kim, "Quantum multi-agent actor-critic networks for cooperative mobile access in multi-uav systems," *IEEE Internet of Things J.,* vol. 10, no. 22, Nov. 2023. (https://doi.org/10.1109/JIOT.2023.3282908)

[11] W. J. Yun, J. P. Kim, S. Jung, J. Kim, and J. Kim, "Quantum multiagent actor-critic neural networks for internet-connected multirobot coordination in smart factory management," *IEEE Internet of Things J.*, vol. 10, no. 11, pp. 9942-9952, 2023. (https://doi.org/10.1109/JIOT.2023.3282908)

[12] W. J. Yun, J. Park, and J. Kim, "Quantum multi-agent meta reinforcement learning," in *Proc. AAAI Conf. Artificial Intell.*, pp. 11 087-11 095, Washington, DC, USA, Feb. 2023. (https://doi.org/10.1609/aaai.v37i9.26313)

[13] W. J. Yun, Y. Kwak, J. P. Kim, H. Cho, S. Jung, J. Park, and J. Kim, "Quantum multi-agent reinforcement learning via variational quantum circuit design," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.,* pp. 1332-1335, Bologna, Italy, Jul. 2022. (https://doi.org/10.1109/ICDCS54860.2022.00151)

[14] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Phys. Rev. Res.*, vol. 1, no. 3, p. 033063, 2019. (https://doi.org/10.1103/PhysRevResearch.1.033063)

[15] H. Baek, W. J. Yun, S. Park, and J. Kim, "Stereoscopic scalable quantum convolutional neural networks," *Neural Netw.*, vol. 165, pp. 860-867, 2023. (https://doi.org/10.1016/j.neunet.2023.06.027)

[16] D. Bouwmeester and A. Zeilinger, "The physics of quantum information: basic concepts," in *The physics of quantum information: Quantum cryptography, quantum teleportation, quantum computation*, pp. 1-14, 2000. (https://doi.org/10.1007/978-3-662-04209-0_1)

[17] I. Hamamura and T. Imamichi, "Efficient evaluation of quantum observables using entangled measurements," *npj Quantum Inf.*, vol. 6, no. 1, p. 56, 2020. (https://doi.org/10.1038/s41534-020-0284-2)

Seok Bin Son

Feb. 2022 : B.S. degree, Seoul Women's University

Mar. 2022~Current : Ph.D. student, Korea University

<Research Interests> Deep learning algorithms and their applications to information security

[ORCID:0000-0002-3692-0752]

Hankyul Beak

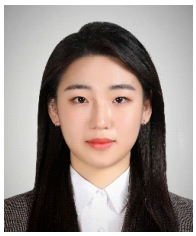
Feb. 2020 : B.S. degree, Korea University

Mar. 2021~Current : Ph.D. student, Korea University

<Research Interests> Quantum machine learning and its applications

[ORCID:0009-0007-4670-6817]

Soohyun Park



Feb. 2019 : B.S. degree, Chung-Ang University

Mar. 2020~Aug. 2023 : Ph.D. degree, Korea University

Aug. 2023~Current : Postdoctoral, Korea University

<Research Interests> Deep learning algorithms and their applications to computer networking, autonomous mobility platforms, and quantum multi-agent distributed autonomous systems.

[ORCID:0000-0002-6556-9746]