Introduction to Quantum Neural Networks

Joongheon Kim
School of Electrical Engineering, Korea University, Seoul, Republic of Korea
E-mail: joongheon@korea.ac.kr

***Abstract*—This paper introduces quantum neural network (QNN) architectures and organization, which consists of state encoding, parameterized quantum circuit, and measurement.**

I. Introduction

This paper basically introduces the neural network design based on the concepts of quantum computing [1].

II. Quantum Neural Network

To design and compute QNN using qubits, the qubits should be controllable for training the neural network. The control is achieved via the utilization of basic quantum gates in order to control the positions of qubits over Bloch sphere [2]. Representative examples of the basic quantum gates are rotation gates, which are expressed as R_x, R_y, and R_z, which are for the rotation over x-, y-, and z-axes. For more details, the gate functions are performed as unitary operations on a single qubit, causing it to rotate by a specific value in the given directions of x-, y-, and z-axes. These gates not only control qubits but also encode classical bit-scale data. While basic quantum rotation gates dd are single qubit gates that can only be applied to a single qubit simultaneously, there are also multiple qubit gates acting on two or more qubits simultaneously. For example, a *CNOT* gate causes entanglement among several qubits by performing an *XOR* operation on two qubits [3]. Based on the above theories and concepts, QNN models are built by assembling various types of gates. Conventional QNN models consist of following three components, *i)* state encoding circuit (A in Fig. 1), *ii)* parameterized quantum circuit (PQC) (B in Fig. 1), and *iii)* quantum measurement (C in Fig. 1) layers.

State Encoding. First, the encoding layer's function is to encode classical data into quantum states because quantum circuits cannot take classical bits as input. Therefore, the state encoder converts bits into qubits by passing q number of $|0\rangle$ into an array of rotation gates using classical data used as parameters denoted as θ_{enc}. Additionally, the input data X is split into $[x_1 \cdots x_N]$ such that they can be individually used as parameters, where N is the number of split data for the input X. Then, the output quantum state of the encoding layer will contain the information of classical data.

Parameterized Quantum Circuit (PQC). Secondly, there is PQC which carries out the desired computation, and it is equivalent to a classical neural network (NN), especially accumulated hidden layer multiplication. In the layer of PQC, the input quantum state is rotated by a specific angle using quantum gates such that the output will give the required value like the action and state values. In our paper, the qubits are computed using the *Controlled-Universal* (CU) gate which has flexible control over the direction of rotation, entanglement, and disentanglement. The structure of the QNN model is presented in Fig. 1, and it can be seen that the encoding layer followed by the CU3 layer is repeated several times. The particular structure is due to the data re-uploading technique [3], simultaneously encoding and rotating the qubits. As a result, the computation efficiency of each qubit is maximized, *i.e.*, the number of qubits is decreased which is required to produce the values needed for multi-agent reinforcement learning.

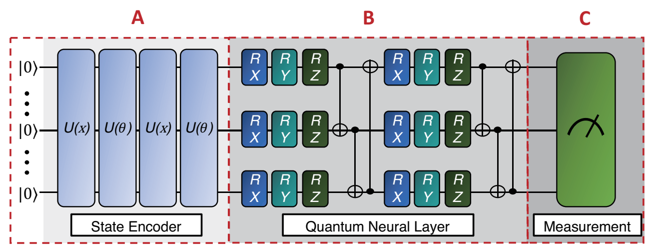

Fig. 1: Architecture of quantum neural networks.

Measurement. Lastly, the quantum state produced from PQC becomes the input of the measurement layer. In this stage, the input is measured such that the quantum data are decoded back into classical data for optimization. The measurement operation is equivalent to the multiplication of a projection matrix with respect to z-axis. While the z-axis is most commonly used for measurement, it can be any other properly defined directions. After conducting the measurement of the quantum state, the quantum state collapses, and it becomes an *observable*. After the decoding procedure, the *observable* is used to minimize the loss function. Then, it should be differentiated for backpropagation. However, quantum data cannot be differentiated because applying the chain rule will completely collapse the state of qubits. Thus, the technique to obtain the loss gradient via the symmetric difference quotient of loss function of observable is used for QNN training.

References

[1] H. Baek, W. J. Yun, S. Park, and J. Kim, "Stereoscopic scalable quantum convolutional neural networks," *Neural Networks*, vol. 165, pp. 860–867, August 2023.

[2] J. P. Kim, W. J. Yun, H. Baek, and J. Kim, "Modern trends in quantum AI: Distributed and high-definition computation," in *Proc. IEEE International Conference on Information Networking (ICOIN)*, Bangkok, Thailand, January 2023, pp. 750–754.

[3] 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 Journal*, pp. 1–1, 2023 (Early Access).