Classical-Quantum Hybrid Convolutional Neural Network on IBM Qiskit

Quantum Ant

Hansung University

1 Problem 1

We train the MNIST dataset through a classical-quantum hybrid neural network to solve problem 1. Figure 1 shows the configuration diagram of a hybrid neural network. First, the classic data is loaded and then encoded into the quantum data. The encoded data is used as input to the quantum circuit. When performing a quantum circuit, measurements are taken at the end of the circuit after going through each gate. With this process, the state of the qubit is determined, and these values are used as input to the classical neural network. The training process of the classic neural network is the same as the existing learning process. Firstly, data is entered in batch units. And then, the loss value is calculated. Finally, the parameter is updated by reflecting loss values. In the hybrid neural network, both parameters of the classical neural network and parameterized quantum circuit are updated.

1.1 Data Encoding

Quantum circuits use qubits that have both 0 and 1 as probabilities, rather than bit units used in conventional computers. Therefore, to use a quantum circuit, the input must be stored in the form of qubits. For this reason, the quantum data encoding must be performed. The quantum data encoding determines *theta* values based on classical data. Then, filter values are determined based on rotation with *theta* values. The filter is used in quanvolution layer like filter of classical convolution neural network. For this reason, the threshold must be set. Since the input data is a black and white image, it has pixel values of 0 to 255. Therefore, 127, which is the median value, is set as the threshold. If the input data is less than the threshold, *theta* becomes 0. If it is greater than the threshold, *theta* becomes *pi* (π; 3.141592...). The rotation is performed as much as the determined *theta* angle to calculate all possible filter values. Rotation operators, such as rx and ry, are applied to each qubit, and the rotation angle is set to 0 or *pi* based on the set threshold. The state of the qubit is changed by rotating with the *theta* value. Circuits can be parameterized through this process. Since the unitary matrix can change the state of qubits, quantum circuits are constructed using rotating gates (e.g. rx, rry, crx, cry, and crz) For rx, ry, and rz gates, it is a rotation operation about the x, y and z axes, respectively. crx, cry, and crz are rotational operations, where two qubits are entangled.

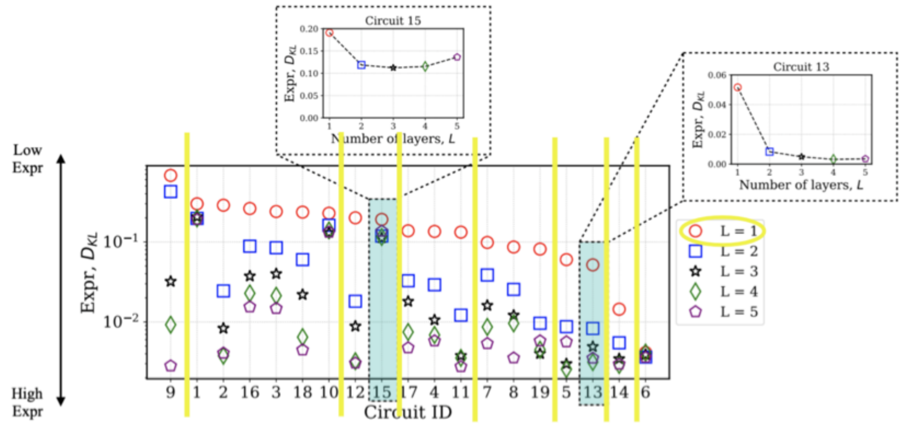

Fig. 1: Architecture of classical-quantum hybrid convolutional neural network.

1.2 Quantum Circuit

In QNN, quantum circuits are used as filters. After applying the *theta* value stored in the input data to the quantum circuit filter, the measurement is performed on the result value. Afterward, measured values are transmitted to the classical neural network. In the quantum circuit filter, qubits as many as the square of kernel size are allocated and use for calculation. Since we use square filters, we generate as many qubits as the size of the filter is squared and assign them to each element of the filter. Therefore, the filter size must be set within the range of qubits supported by quantum hardware or simulators. Then, we set the backend, shot, and threshold. For the backend, we can use real quantum devices provided by IBM, or we can use a simulator. However, we have confirmed that when learning with real quantum devices, it takes too much time compared with the simulator. Shot is a factor that indicates how many times a quantum circuit will be run, and the default is set to 1,024. It is recommended to run more than once and select a value observed more than 900 times out of 1,024 times. Since the MNIST dataset is a black-and-white image, it has pixel values between 0 and 255. Therefore, the threshold value is set to 127, which is an intermediate value. When the pixel value of MNIST dataset exceeds that the threshold, a *pi* value is added to *theta*, and 0 is added in the opposite case. In the quantum circuit filter, the state of the qubit is calculated by the rotation according to the *theta* value stored in the data value. Finally, the qubit that has passed through the quantum circuit filter circuit is measured and the measured value is transmitted to the existing neural network.

In [1], 19 quantum circuit filters in Figure 2 is presented. The accuracy according to the number of layers was measured when 19 quantum circuit filters were applied. The accuracy of the quantum circuit filter is shown in Figure 3. There are various quantum circuit filters. The accuracy is different for each circuit. The more complex the circuit, the higher the accuracy, but it is important to consider the cost aspect according to the depth and parameter and to use an appropriate quantum circuit filter.

Circuit 9 has the lowest accuracy. However, it has a low depth and it can be calculated with a small gate cost. Due to the small depth, the gate cost is low, but the accuracy is very low. For this reason, it is expected that it will not act as a good filter. Circuits 1, 2, 3, and 4 show relatively low accuracy. Also, the gate cost is not large because the circuit is not deep. Circuit 2 is the addition of $CNOT - gate$ to Circuit 1, and Circuit 3 is the change of $CNOT - gate$ of Circuit 2 to $R_z - gate$. Also, Circuit 4 is the $R_z - gate$ of Circuit 3 changed to $R_x - gate$. Circuits 17, 4, and 11 have appropriate accuracy and depth is also considered appropriate. It has moderately high accuracy and the depth of the circuit is also suitable. It does not cost a lot of gate cost and it is considered suitable for use as a filter. Circuits 7 and 8 show relatively high accuracy. In addition, the gate cost is not large because the depth of the circuit is not deep. It has moderately high accuracy and the circuit depth is not deep. The gate cost is not too high and it is good to use as a filter. Circuit 5 and Circuit 13 both showed high accuracy. Circuit 5 showed lower accuracy than Circuit 13 despite using a lot of gate cost according to the deeper depth than Circuit 13. Circuit 5 and 13 both have high accuracy, but Circuit 13 has a lower depth than Circuit 5 and shows higher accuracy. Using Circuit 13 will be much more effective. Circuit 6 shows the highest accuracy among quantum filters. Due to the deep circuit depth, a large gate cost is consumed. Although it has high accuracy, it consumes a lot of gate cost due to its deep depth. It is judged that it is more appropriate to use a filter with a slightly lower accuracy and a lower gate cost.

1.3 Selected Circuit

We checked the performance by selecting Circuit 6, Circuit 8, Circuit 9, Circuit 11, and Circuit 13 among the 19 proposed quantum filters.

Criteria for which we chose the circuit

- Circuit 6 in Figure. 4
  It is judged that the feature with the highest accuracy is suitable for the comparison with other circuits.
- Circuit 8 in Figure. 5
  Circuit 8 has improved accuracy by modifying Circuit 11 and shows relatively high accuracy. Circuit 8 has improved accuracy by modifying Circuit 11 and shows relatively high accuracy.
- Circuit 11 in Figure. 6
  This is for comparison with Circuit 8.
- Circuit 9 in Figure. 7
  It is judged that the feature with the lowest accuracy is suitable for comparison with other circuits.

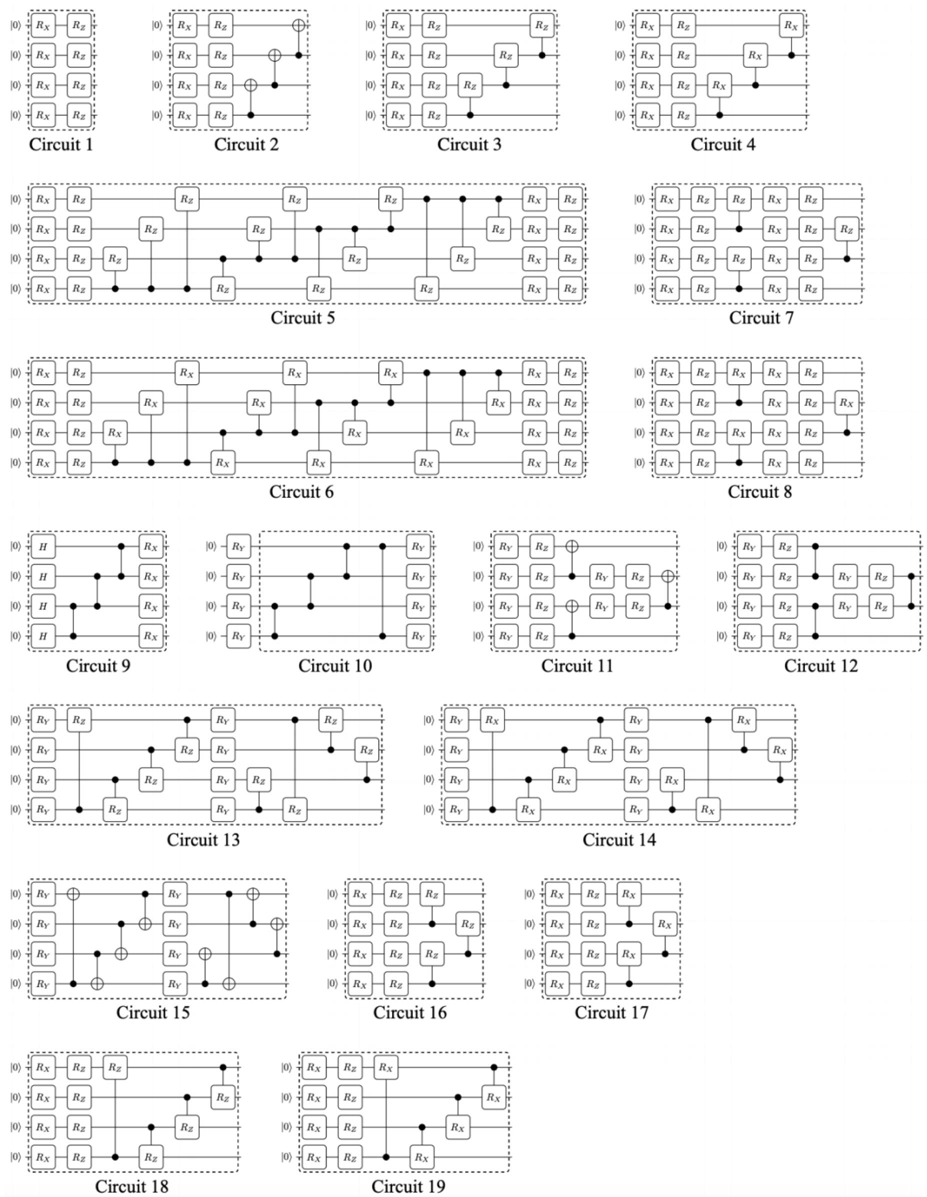

Fig. 2: Quantum filter of [1].

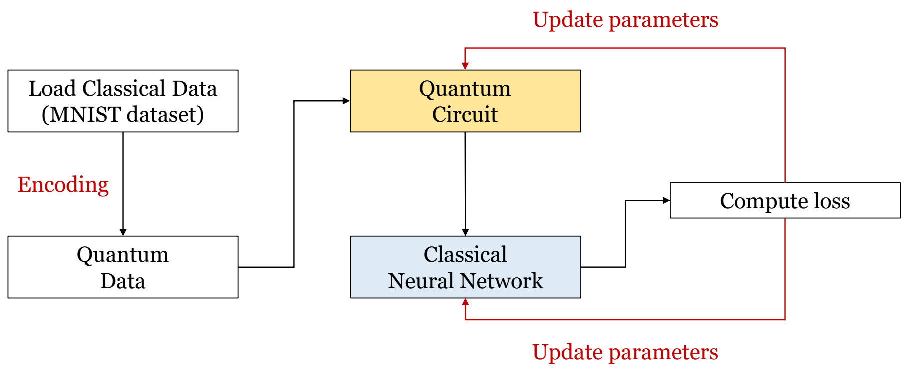

Fig. 3: Accuracy according to the number of layers of [1].

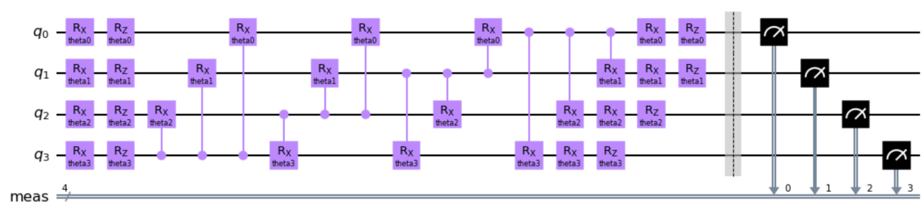

Fig. 4: Quantum circuit 6 of [1].

- Circuit 13 in Figure. 8
  It is judged to be a circuit that properly maintains the circuit depth and accuracy. It is considered the most suitable for use as a filter.

1.4 Classical-Quantum Hybrid Convolutional Neural Network

Hybrid neural networks consist of quantum circuits and classical neural networks. Encoding into quantum data, quantum circuit execution, and measurement in the quantum circuit, and classical neural network are performed in the following order. For this reason, the quantum circuit must be configured as a layer of hybrid neural networks. The quantum circuit is the same as previously described, and consists of functions, such as forward propagation and back propagation. The classical neural network consists of a convolutional neural network that is well-known for its excellent image classification performance for training the MNIST dataset. A classical neural network is performed after a quantum circuit is performed and takes the output of the quantum circuit as input values.

A classical convolutional neural network consists of a convolutional layer, a max pooling layer, and 2 fully connected layers. In the convolution layer, a matrix multiplication operation is performed between the filter (kernel) and the input data. When the filter and stride values are set to 2, 1, the 2×2 filter is

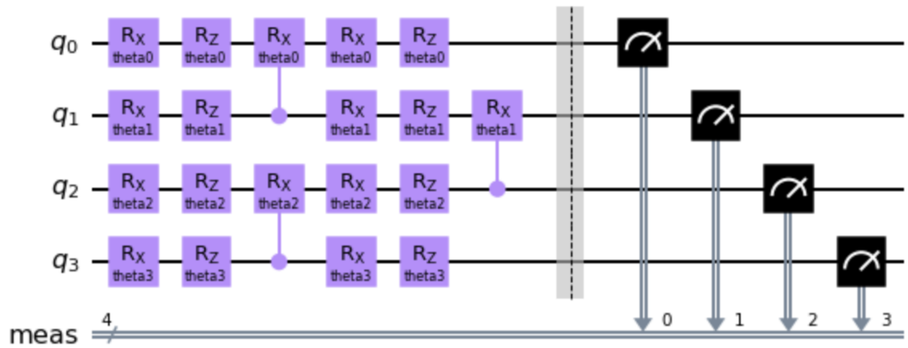

Fig. 5: Quantum circuit 8 of [1].

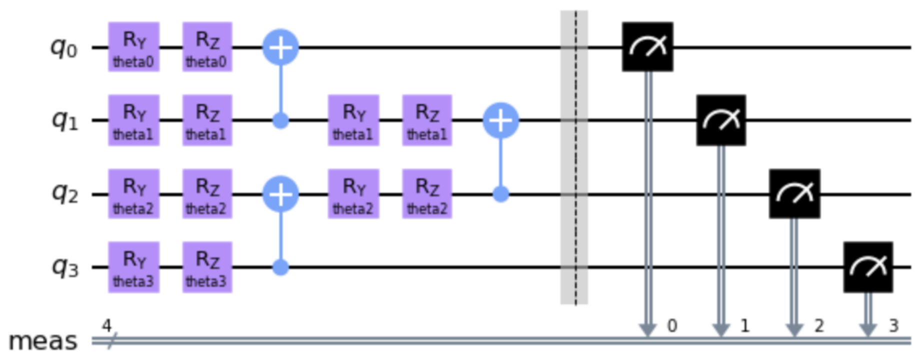

Fig. 6: Quantum circuit 11 of [1].

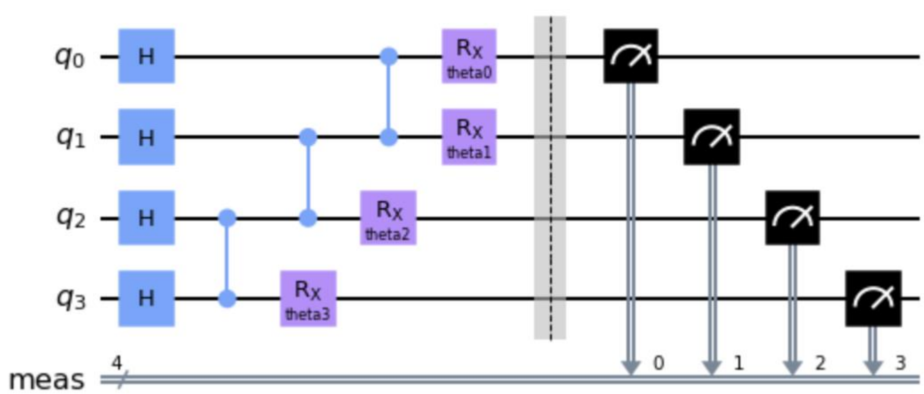

Fig. 7: Quantum circuit 9 of [1].

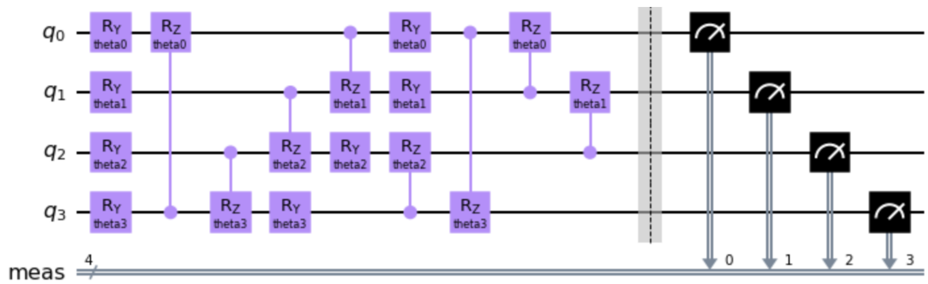

Fig. 8: Quantum circuit 13 of [1].

computed on the input image, then this filter is moved by 1 and then the same operation is performed again. The maximum values of feature maps obtained after the convolution layer are extracted through max-pooling. By these processes, characteristics of the input data can be extracted and enhanced. Since it is a multiple classification problem (the number of label is 10), the activation function of the output layer is set to softmax and the unit is set to 10, and a categorical cross-entropy loss function is used. In the hidden layer, ReLU, which is faster than other activation functions, was used, and Adam was used as the optimization function for this network. Detailed description is given in Table 1. Finally, a hybrid neural network that combines a quantum layer (quantum circuit) and a convolutional neural network calculates a loss by performing forward and back propagation like a classical neural network. The loss is fed back to classical neural networks and quantum circuits to update the parameters of both models.

Table 1: Details of classical-quantum hybrid convolutional neural network.

Category	Description
Layers	Quanvolution, Convolution, Maxpooling, Fully-connected
Qubits	4 (for quantum filter)
Gates	H, rx, ry, rz, crx, cry, crz
Threshold	127
Activation	ReLu (hidden layer), Softmax (output layer)
Loss	Categorical crossentropy
Optimizer	Adam
Batch size	10
Epoch	20

Table 2: Comparison of performance results by circuit

Circuit	Accuracy	Loss	Depth
Circuit 6	10%	2.19	16 × (epoch × (data size / batch size))
Circuit 8	10%	2.19	7 × (epoch × (data size / batch size))
Circuit 11	10%	2.18	7 × (epoch × (data size / batch size))
Circuit 9	10%	2.16	6 × (epoch × (data size / batch size))
Circuit 13	10%	2.18	10 × (epoch × (data size / batch size))

2 Experiment Result

We measured the performance of Classical-Quantum Hybrid Convolutional Neural Network with the selected quantum circuit filter (circuits 6, 8, 11, 9, and 13) applied. When we ran our hybrid model on IBM's real quantum hardware, it was too slow to check the result. When running on the simulator, the training speed was still very slow. For this reason, we couldn't experiment circuits with variable hyper-parameters. Table 2 shows the comparison of performance result by circuits. The overall depth of the circuit depends on the epoch and batch size and is shown as an equation in Table. 2. We checked the performance of the circuit with small datasets and small epochs due to the limitations of IBM's simulation speed. Therefore, relatively accurate results were not obtained. We think this is probably a problem with small datasets and small epochs.

2.1 Circuit Performance Results

We did not get accurate results due to the limitations of the development environment. So, to evaluate the filter performance, we refer to the results of [1].

Circuit 6 The execution result showed the highest accuracy. Since the circuit depth is large, it is an unsuitable filter in terms of gate cost.

Circuit 8 and 11 As a result of execution, both Circuits 5 and 13 had high accuracy, but Circuit 5 showed lower accuracy than Circuit 13 despite having a deeper depth. When comparing the two circuits, using Circuit 13 is effective both in terms of gate cost and accuracy.

Circuit 9 Since the depth is small, the gate cost is small. However, the accuracy is low. It is confirmed that it is not suitable for use as a quantum circuit filter.

Circuit 13 As a result of the experiment, Circuit 13 showed quite high accuracy. Not only that, but it has a depth that is not very deep. Considering both accuracy and gate cost, it is considered to be the best filter.

3 Conclusion

We implemented quantum circuit filter and classic-quantum hybrid neural network. In addition, we trained a hybrid neural network applying various quantum circuits.

3.1 Quantum backend

When the hybrid model was run on IBM's real quantum hardware, the result could not be confirmed due to the too slow training speed. When running on the simulator, the training speed was also too slow to experiment with circuits with various hyperparameters. We considered the reason as follows. In quantum circuits, in order to obtain accurate results, the circuit must be repeated several times when it is performed once, and repeated execution is repeated due to the batch unit training of neural networks.

The number of qubits provided by real quantum hardware on IBM is different. Since hybrid neural networks require qubits to represent quantum data (filters), the kernel size must be set within the number of qubits supported by quantum hardware. In general, a classical neural network is better with more kernels (channels). Therefore, the performance is expected to improve when more qubits are available.

3.2 Quantum circuit

There are various quantum circuits that can be used as quantum filters in hybrid models, and each circuit yields different accuracy. However, it is thought that it is necessary to design an appropriate quantum circuit in consideration of the cost and accuracy according to the depth and parameters of the circuit. Also, the number of qubits supported by real quantum hardware is different. Therefore, when constructing a quantum neural network, it is also necessary to design a circuit considering the number and accuracy of supported qubits. That is, we have to design a suitable quantum circuit for our environment and purpose.

References

1. S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms," *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1900070, 2019.