Hybrid Classical-Quantum Neural Network for handwritten digit recognition

CMPE-789 Quantum Computing
Section 01

Carlos Fernandez
csf5856@rit.edu

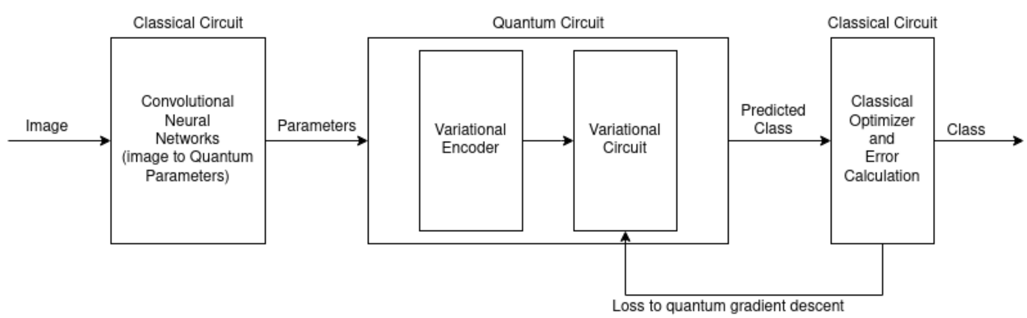
Sharjeel Sahar Ansari
sa2676@rit.edu

November 24, 2020

1 Introduction

For this project we chose to expand the implementation of a Hybrid Classical-Quantum Neural Network for handwritten digit recognition found in the "Learn Quantum Computation using Qiskit" textbook. The original network was implemented to classify only 2 classes(0 or 1) and used 1 Qubit. Our implementation can recognize 6 classes(0, 1, 2, 3, 4 or 5) and uses 3 Qubits.

A general overview of the network we are working with is as follows:



Each of the blocks in the network is discussed in subsequent sections.

The aim of this project is to show that we can use quantum computer to implement a machine learning model. It is not to achieve a industry grade accuracy. This model performs considerably slower and gives a worse accuracy compared to state-of-the art classical handwritten digit recognition models

2 Background

2.1 Parameterized Quantum Circuits

These are quantum circuits that utilize parametrized gates(for example rotation gates). These gates perform a certain operation on the qubit based on the specified parameter. Such circuits are typically composed of fixed gates. The output state of these circuits can be optimized to approximate the wanted states by changing the parameters.

Typically, PQCs constitute fixed and adjustable gates, where the initial states of all the qubits is assigned to $|0\rangle$

The output of these circuits are measurements that follow *Born Rule*, which states that the measurement outcomes corresponds to one of the eigenvalues and follows probability distribution as follows:

$$P(\lambda_i) = tr(P_i U |0\rangle \langle 0| U^\dagger)$$

and the equation for expectation of measurement is:

$$< M >= \sum_i \lambda_i P(\lambda_i) = \sum_i \lambda_i tr(P_i U |0\rangle \langle 0| U^\dagger)$$

In our work, the parameterized circuit we're working with consists of H gate followed by R_y gate for each of the three qubits. The angle of rotation for these R_y gates constitute the parameters.

2.2 Variational Encoding

This part of the quantum circuit is used to prepare a set of quantum gates with parameters generated by the input data. Let input data be $\mathbf{a} = \{a_1, a_2, ..., a_n\}$, where n is the number of qubits and a_i be the input parameter for qubit i.

Encoding $\mathbf{a}$ through variational encoding is to prepare the gate $\mathbf{G}$ as follows:

$$\mathbf{G} = \otimes_{i=0}^{n-1} g_i(f_i(a_i))$$

where,
g_i = single qubit Quantum Gates, and
f_i = Classical function to encode a_i as parameters of g_i.
In our work, the respective functions are defined as follows:
f_i = Identity
$g_i = HR_y$, where
H = Hadamard gate, and R_y = Rotation gate along $y - axis$
Thus, the gate definition for our encoder is as follows:

$$\mathbf{G} = \otimes_{i=0}^{n-1} HR_y(a_i)) |0\rangle^{\otimes n}$$

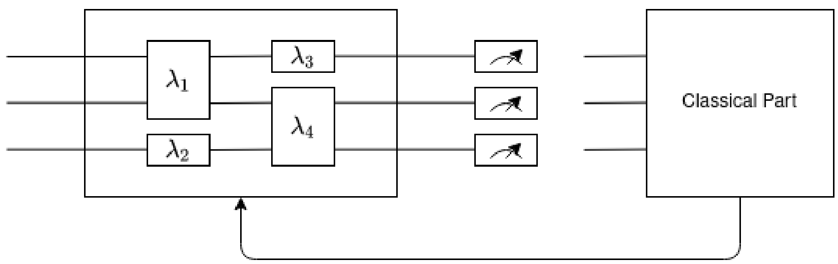
2.3 Variational Circuit

Variational circuits are dubbed as the neural network of Quantum Computing. It follows the *Universal Approximation Theorem* similar to that of classical neural networks.
Universal Approximation Theorem for Variational circuit: There always exists a quantum circuit that can represent a target function with an arbitrary

small error.
One caviat is that the circuit can get exponentially deep and therefore impractical. The solution to which was proposed in one of the paper. They stated that *"Real datasets arise from physical systems. They exhibit similarity and locality. This suggests that it is possible to use a cheap circuit and still obtain a satisfactory result."*

Hence, variational circuit aims to implement a function that can approximate the task at hand while remaining scalable in the number of parameters and depth. In practice, the circuit design follows a fixed structure of gates, which reduces model complexity.

A general structure of a variational circuit is as follows:



where $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ are the parameters of the parameters of the network. At time $t = 0$, $\Lambda^{(0)}$ acts on the three qubits as depicted, and the results are measured. These measurements are passed to classical part of the network where the error is calculated and accordingly, Λ is updated. Finally, the updated parameters are returned to Quantum part as $\Lambda^{(1)}$.

2.4 Quantum Gradients

Calculation of quantum derivatives follows *Parameter shift rule*, which states if the generator of a gate G has two unique eigenvalues (e_0 and e_1) then the derivative of this circuit expectation with respect to the gate parameter is propotional to the difference in expectations of two circuits with shifted parameters. i.e,

$$\frac{\mathrm{d}f(\theta)}{\mathrm{d}\theta} = r[f(\theta + \frac{\pi}{4r}) - f(\theta - \frac{\pi}{4r})]$$

where
$\theta =$ circuit parameter,
$r =$ shift constant, which is equal to $\frac{1}{2}(e_1 - e_0)$
Therefore, the gradient descent backpropogation as per *Parameter Shift Rule* is:

$$\nabla_\theta QC(\theta) = QC(\theta + s) - QC(\theta - s)$$

where,
$\theta =$ Circuit Parameter, and $s =$ macroscopic shift

3 Neural Network

3.1 Dataset

We will train our hybrid classical-quantum neural network on the MNIST dataset, which consists of labeled examples of handwritten digits. We used 100 random examples of each class(600 total) for training and 50 random examples of each class(300 total) for testing.

3.2 Architecture

The architecture of our network is as follows:

I. 1×10 Convolutional Input layer (ReLu).

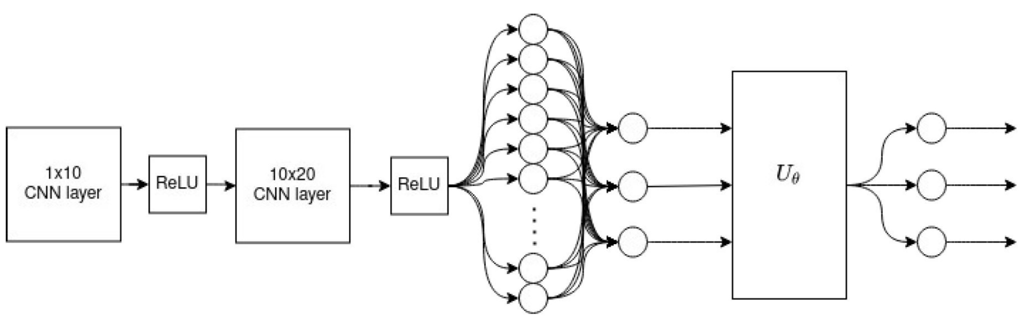
II. 10×20 Convolutional layer (ReLu).

III. 320 neuron Fully Connnected layer.

IV. 50 neuron Fully Connnected layer.

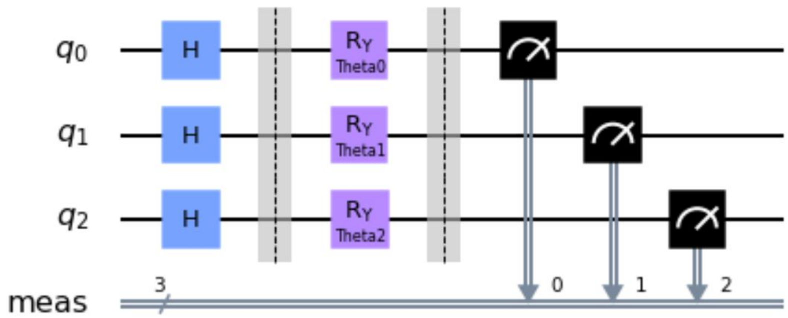
V. 3 qubit Quantum layer.

VI. 3 neuron Fully Connnected Output layer(Output is represented in binary).



3.2.1 Quantum Layer

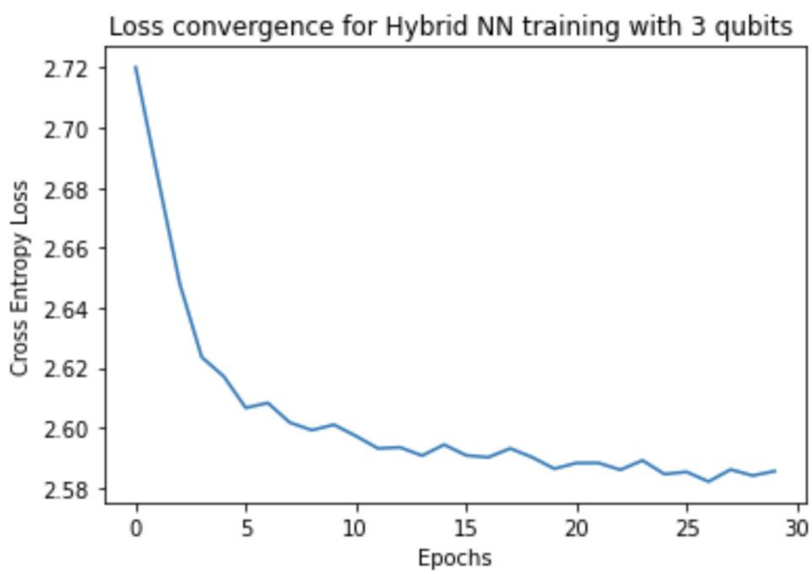
Our quantum layer consist of 3 qubits with a Hadamard gate and a Rotate Y gate. The parameters for the Ry gates are the input for the quantum layer. Here is the circuit:



4 Results

4.1 Digits from 0 to 5

We trained the network with 30 Epochs, a learning rate of 0.0009 using the Adam optimizer and cross entropy as our loss function, It took about 30 minutes to train on a i7-7500U CPU. Here is a graph showing the loss convergence:



After training, our model achieved an accuracy of **91%**.

4.2 Other attempts

We tried making a 4 bit quantum layer to predict all 9 digits in the dataset, however, this took much longer to train and yielded lower accuracy(75%) when compared to our network with the 3 qubit layer.

We also tried training the network to recognize digits from 0 to 7(000_b to 111_b) but this took a bit longer to train and the maximum accuracy that we got with this was 83%.

This is why we decided to only classify digits from 0 to 5.

5 Conclusion and Future work

Through this project, we saw that neural networks can be simulated using a hybrid quantum-classical network by combining PQC. We showed that the quantum-classical neural network can be trained to obtain a fairly decent accuracy at recognizing handwritten digits from MNIST dataset, and demonstrates the power of the proposed hybrid network.

Furthermore, we can use other Quantum circuit models to replace classical aspects in the above model, such as replacing Adam optimizer (classical) with QAOA - *Quantum Approximate Optimization Algorithm*, a purely quantum optimizer. And if there exists a purely quantum encoder to encode the physical parameters into the circuit, one can achieve a 100% quantum neural networks.

As for improving the accuracy of our model is concerned, we can play around with combination of parameters such as *learning rate, number of epochs, number of shots* and so on

References

[1] Gavin E. Crooks. *Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition.*

[2] Marcello Bendetti, Erika Lloyd, Stefan Sack and Mattia Fiorentini. *Parameterized Quantum Circuits as Machine Learning Models.*

[3] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu and Dacheng Tao. *The Expressive Power of Parameterized Quantum Circuits.*

[4] Rongxin Xia and Sabre Kais. *Hybrid Quantum-Classical Neural Network for Calculating Ground State Energies of Molecules.*

[5] Junhua Liu, Kwan Hui Lim, Kristin L. Wood, Wei Huang, Chu Guo and He-Liang Huang. *Hybrid Quantum-Classical Convolutional Neural Networks.*

[6] IBM Quantum Experience. *Hybrid quantum-classical Neural Networks with PyTorch and Qiskit.*