Quantum Circuit from Unitary using Machine Learning

Student Name : Karan Dhingra

Roll Number : MT19025

Project Report

for

CSE622 : Introduction to Quantum Computing

Semester : Winter 2020

Under the Guidance of

Dr. Debajyoti Bera

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Problem Statement

Given an unknown Unitary function(U) of order N, design a Quantum Circuit Q_C which is closest to the Unitary, i.e. minimize $||U - Q_C||$,

Abstract

The problem of decomposing a $N \times N$ unitary into a quantum circuit was inspired from Task-4 of IBM Quantum Challenge, 2020. In that challenge, we were asked to design a quantum circuit for a predefined unitary matrix, such that it requires a minimum number of gates. One of the approaches with which this problem can be tackled is to use a machine learning algorithm, in which parameters of U3 gates are learned using gradient descent algorithm with the norm between U and circuit used as the cost function. In this project, at first analytical methods to compute the gradients were used, which worked well for classification related tasks but could not hold up to unitary estimation. After that, numerical gradient estimation techniques were used to solve the problem.

Contents

Chapter 1

Introduction

It is inspired from Task-4 of IBM May Challenge. We were asked to design a quantum circuit for a predefined unitary matrix, U

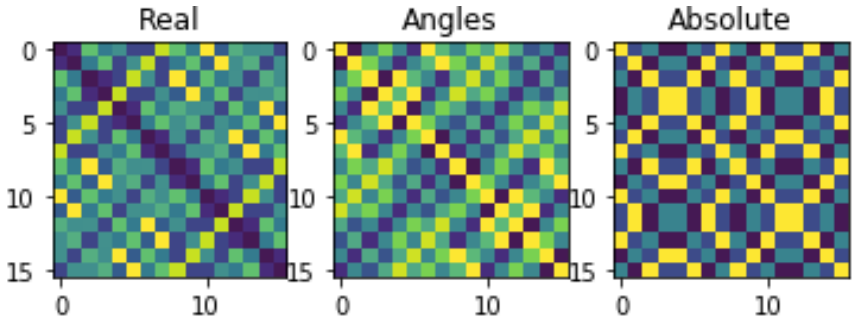

Figure 1.1: Unitary Matrix

The initial implementation, used Isometries[1] to generate circuit but it resulted in more than 150 CNOTs gates which was not permitted. In order to reduce the cost, commutative properties of the CNOTs gate which required manual analysis of each circuit.
After the submission, one submission came out to the top[2] (which was not the best) but the authors has used Quantum Machine Learning to design circuit. In my understanding, both Isometry and QML techniques aim to decompose a given unitary into circuit but QML has an additional condition of loss which allows it to design circuits with higher norm but reduced depth (in terms of gate).
The circuit centric quantum classifier constitutes one kind of machine learning algorithm, which calculates the gradients of parameters analytically[3]. There are numerical methods[3] which comes in handy when we do not have data represented by X, and y pairs, as in the case of unitary decomposition.

Chapter 2

Quantum Machine Learning Model

A quantum machine learning model is similar to the general machine learning model and involves four key blocks,

2.0.1 Encoding input features

Given a quantum register, R of n qubits, and classical input vector X = $[X_1, X_2, ...X_k]$, $k \leq 2^n$. In order to transfer X to R, we use the technique of amplitude encoding, i.e. X is encoded as amplitude of the state represented using R. In cases when k is less than 2^n, padding with non-informative data is used.

For: 1 Qubit

Given an input state (R), $|0\rangle$ and X as $[\alpha, \beta]$. We apply R_θ gate, where $\theta = 2*\cos^{-1}(\frac{\alpha}{\sqrt{\alpha^2+\beta^2}})$.

$$\begin{aligned} &\text{State}: |0\rangle \\ &R_y(\theta): \begin{pmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{pmatrix} \\ &\text{State}: \frac{\alpha|0\rangle + \beta|1\rangle}{\sqrt{\alpha^2+\beta^2}} \end{aligned} \tag{2.1}$$

For: 2 Qubit

Given an input state (R), $|00\rangle$ and X as $[\alpha, \beta, \gamma, \eta]$. We first normalise it to a unit vector:

$$
\begin{aligned}
\text{State : } & |00\rangle \\
\text{Apply } R_y(\theta) \text{ on Qubit 1 : where, } & \theta = 2 * \cos^{-1}(\sqrt{\alpha^2 + \beta^2}) \\
\text{State : } & \sqrt{\alpha^2 + \beta^2}|00\rangle + \sqrt{\gamma^2 + \eta^2}|01\rangle \\
\text{Apply } R_x \text{ on Qubit 1 : } & \\
\text{State : } & \sqrt{\alpha^2 + \beta^2}|01\rangle + \sqrt{\gamma^2 + \eta^2}|00\rangle \\
\text{Apply } CR_y(\theta) \text{ on Qubit 1} \rightarrow 0 \text{ : where, } & \theta = 2 * \cos^{-1}(\alpha) \\
\text{State : } & (\alpha|0\rangle + \beta|1\rangle)|1\rangle + \sqrt{\gamma^2 + \eta^2}|0\rangle \\
\text{Apply } R_x \text{ on Qubit 1 : } & \\
\text{State : } & (\alpha|0\rangle + \beta|1\rangle)|0\rangle + \sqrt{\gamma^2 + \eta^2}|1\rangle \\
\text{Apply } CR_y(\theta) \text{ on Qubit 1} \rightarrow 0 \text{ : where, } & \theta = 2 * \cos^{-1}(\gamma) \\
\textbf{Final State : } & \alpha|00\rangle + \beta|10\rangle + \gamma^2|10\rangle + \eta^2|11\rangle
\end{aligned}
\tag{2.2}
$$

Similarly, the algorithm for 2 qubits can be extended to n qubits.

2.0.2 Designing the Circuit

The circuit is initialised to a specific quantum geometry ansatz, which is also used in Variational Quantum Eigensolver. The Circuit C is divided into multiple blocks, $C = U_n B_n B_{n-1} ... B_1$. Each block B, consists of single qubit gates U_{t_k} and Controlled gates C_k

$$
B = \prod_{k=0}^{n-1} C_{c_k}(G_{t_k}) \prod_{j=0}^{n-1} G_j \tag{2.3}
$$

where $C_k(G_{t_k})$ applies control from k qubits to t_k qubits, such that $t_k = (kr - r)\ mod\ n$. The factor r, controls the entanglement between the qubits, and blocks with multiple r's are used to create a strong entangled circuit.

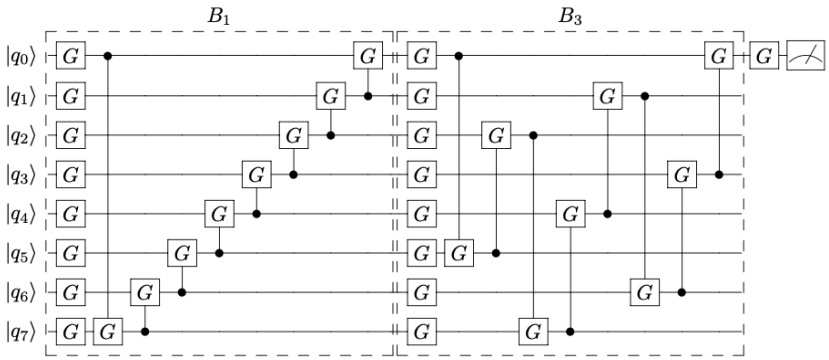

Figure 2.1: B1 block has r = 1, while B3 block has r=3

2.0.3 Learning the Parameters

The single qubit gates are designed using U3 gates, which allows the gate to be set using 3 parameters. Thus for a circuit with L blocks, there will be 3nL parameters for single qubit gates and $\sum_j 3*L*((jr-r)modn)$ 3 parameters gate, where j denotes the entanglement factor for each block. Just like VQE, strongly entangled circuits are preferred because it allows more diverse state space which we can not achieve with non entangled circuit, because in non entangled circuit there will be no relationship between different qubits, and thus the output would not be correlated.

Cost function

Given an input x, the output of the circuit is going to be:

$$P(s = k, x, \theta) = \langle C|P_k|C\rangle \tag{2.4}$$

where P_k is the projector corresponding to the post measurement state of $|k\rangle$. In the paper [2], aim of the network was to generate the probability of state $|1\rangle$ and use that probability for classification task. But for circuit decomposition, we need to estimate U', which can be formulated as:

We generate random state $|\phi_i\rangle$ as the input to the network, and observe output state $|\phi_o\rangle$ then the objective function for hybrid gradient scheme can be defined as:

$$\pi(\theta, |\phi_i\rangle, b) = p(|\phi_o\rangle, |\phi_i\rangle, \theta) + b, \tag{2.5}$$

where b is the bias variable and p is the probability of observing state $|\phi_o\rangle$ when input to the network is $|\phi_i\rangle$. Thus the cost function is,

$$C(\theta, |\phi_i\rangle) = ||\pi(\theta, |\phi_i\rangle, b) - U_{|\phi_i\rangle,|\phi_o\rangle}||, \text{mean square error} \tag{2.6}$$

where $U_{|\phi_i\rangle,|\phi_o\rangle}$ is the probability of target unitary.

Hybrid Gradient Scheme

Using gradient descent scheme, each parameter $\theta_i \in \theta$ is updated using

$$\theta_i \leftarrow \theta_i - \eta \frac{\partial C(\theta, |\phi_i\rangle)}{\partial \theta_i}, \tag{2.7}$$

where η is the learning rate.

$$\frac{\partial C}{\partial \theta_i} = (\pi(\theta, x) - U_{i,o}) \frac{\partial \pi(\theta, x)}{\partial \theta_i} \tag{2.8}$$

$$\frac{\partial C}{\partial b} = (\pi(\theta, x) - U_{i,o}) \frac{\partial \pi(\theta, x)}{\partial b} \tag{2.9}$$

where $\frac{\partial \pi(\theta,x)}{\partial b} = 1$,

$$\frac{\partial \pi(\theta_i, |\phi_i\rangle)}{\partial \theta_i} = \frac{\partial p(|\phi_o\rangle, |\phi_i\rangle, \theta_i)}{\partial \theta_i} \tag{2.10}$$

In this representation, we are required to calculate the gradient of the unitary estimated from the circuit which is not possible because the gradient might turns out to be non unitary, which can not be implemented on Quantum Circuit

Linear Combination of Unitary

In[2], the network is converted into linear combination of the unitaries, after that gradient of each parameter will depend on its either single qubit gate or controlled gate only. The gradients of single qubits gates can be calculated as:

$$\begin{aligned}\frac{\partial U(\theta,\phi,\lambda)}{\partial\theta} &= \frac{\partial \begin{pmatrix} cos(\theta/2) & -e^{i\lambda}sin(\theta/2) \\ e^{i\phi}sin(\theta/2) & e^{i(\phi+\lambda)}cos(\theta/2)\end{pmatrix}}{\partial\theta} \\ &= \begin{pmatrix} -sin(\theta/2) & -e^{i\lambda}cos(\theta/2) \\ e^{i\phi}cos(\theta/2) & -e^{i(\phi+\lambda)}sin(\theta/2)\end{pmatrix} \\ &= U(\theta+\frac{\pi}{2},\phi,\lambda)\end{aligned} \tag{2.11}$$

Similarly, it can be calculated for other parameters too. Thus, to calculate the gradient, circuit can be implemented as Linear combination of individual derivative gates. After measurement, these gradients are used to update the individual parameters.

2.0.4 Numerical Method for Gradient Estimation

Using analytical methods, we are required to calculate gradient for a specific circuit, which prohibits it from calculating it for the complete unitary and that's why we have to sample multiple states and calculate gradients for those states. Another technique could be to use numerical techniques to estimate the gradients.

For parameter α, the gradient

$$\frac{\partial\pi(\alpha,|\phi\rangle\langle\phi|)}{\partial\alpha} = \frac{|\phi_{\alpha+s}\rangle\langle\phi_{\alpha+s}| - |\phi_{\alpha-s}\rangle\langle\phi_{\alpha-s}|}{s} \tag{2.12}$$

where s is a very small quantity and $|\phi_{\alpha+s}\rangle$ represents quantum circuit with parameter α increased by s value. The benefit of this method is that, not only it is faster than analytical because we have to run the circuit single time per parameter when compared with 2 or 4 times for analytical. But, it provides an estimate of the gradient only.

Chapter 3

Experiments

The implementation of Analytical algorithm required us to design gradient scheme too, while Adam algorithm from Qiskit was used for numerical gradients. We explored two datasets, IRIS and Moon dataset for classification tasks, and Unitaries upto 8×8 for circuit decomposition.

3.1 Classification

The IRIS dataset, worked as baseline to compare the performance, as it is linearly separable. The moon dataset allowed us to evaluate the algorithm's performance on non-convex decision boundaries.

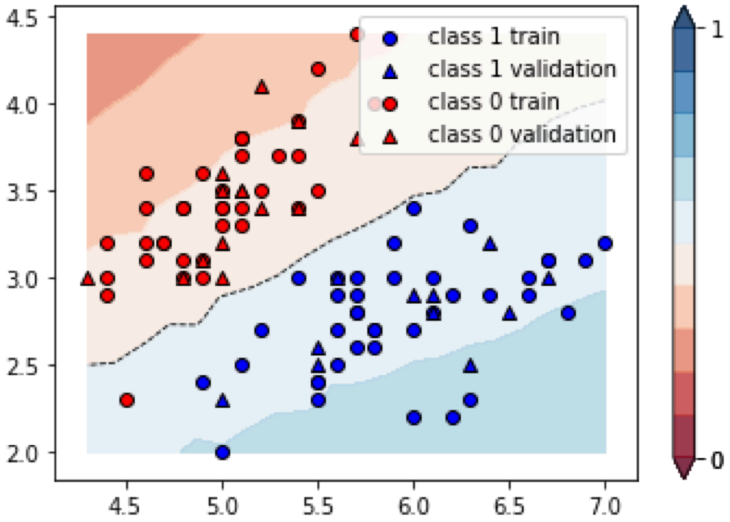

Figure 3.1: Quantum Classifier : IRIS Dataset

The Quantum Classifier, was able to separate both of the classes using only one Qubit and U3 gate. Also, the boundary is not straight line arising the possibilities of learning more complex boundaries in complex datasets. In Moon dataset, we did not observe improvement in the performance when we jumped from 1 Qubit implementation to 2 Qubit implementation. Also, the boundary learned by the Classifier should be able to learn non-convex

boundary which did not happen even when the input was transformed using square kernel,

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \rightarrow \begin{pmatrix} X_1^2 \\ X_1 * X_2 \\ X_2 * X_1 \\ X_2^2 \end{pmatrix} \tag{3.1}$$

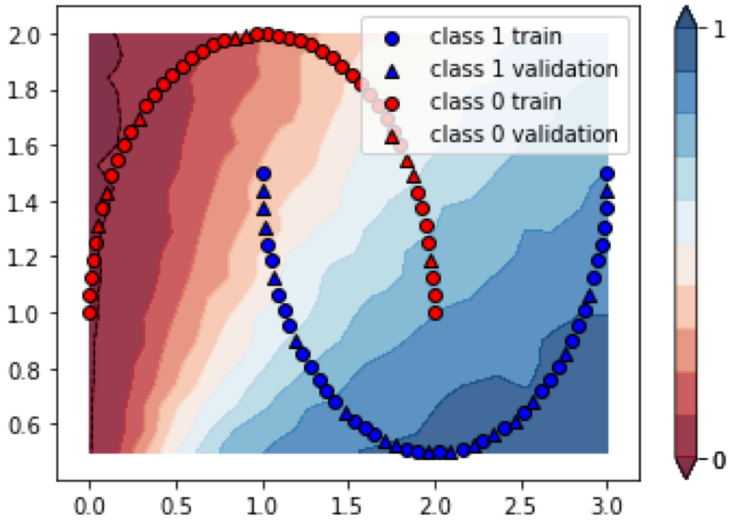

Figure 3.2: Quantum Classifier : Moon's Dataset

Even though the boundary learned is non-convex, it is still not able to draw boundary around non linear data properly. Though, we are still verifying the results of [2] regarding non-convex boundary with Square kernel.

3.1.1 Unitary Decomposition

With the analytical gradient, our solution stucked at local minima, as it can be observed in figure 3.3. This spiral also signifies that by inputing random value of X to estimate the unitary, we also need to ensure high diversity in the samples of X to ensure better generalization. In order to prove these points, we tested the same unitaries using Numerical gradient techniques.

With numerical gradients, we were able to estimate the circuits for all three different unitaries and the performance improved when more blocks are added to learnable parameters.

Unitary Size	B = 1	B = 1, 3
2x2	1E-8	-
4x4	0.52	1E-8
8x8	.75	0.11

As, it can be observed from the results that the same network was able to estimate the correct parameters, which analytical gradients failed.

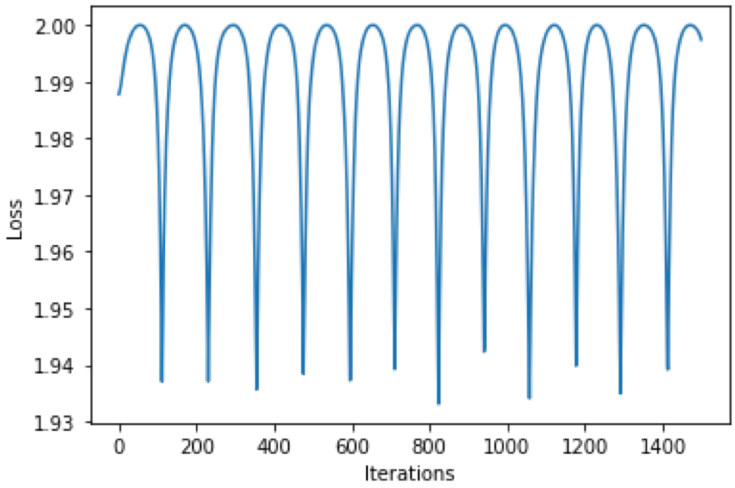

Figure 3.3: Loss function : 1 Qubit Unitary

3.2 Analysis

The results shows that, analytical gradients technique are going to be more efficient in tasks where we have to fit the certain form of data, but it gets stuck in the local minima when the task is fit the complete unitary. In case of numerical gradients, because it has better estimates of unitary gradients than analytical it performs better than analytical gradients. But, in real quantum computer all the advantages offered by numerical gradients with the help of unitary simulator will not work and analytical gradient would work better in comparison.

Chapter 4

Conclusion

In this project, we explored different machine learning techniques which can be applied in Quantum algorithms. Also, we observed how principles of Quantum theory can improve the decision capabilities of machine learning model. We observed a roadblock with analytical gradient techniques for unitary decomposition, and later on the future prospects would be to test the analytical gradients techniques to understand where the local minima occurs and why the decision boundary generated does not match with the one demonstrated in [2].

Bibliography

[1] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, "Quantum circuits for isometries," *Physical Review A*, vol. 93, no. 3, p. 032318, 2016.

[2] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, 2020.

[3] 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.