Quantum Machine Learning

Design Project

About

Title: QML

Author:

Abu Shahid (B20CS003)

Chakshu Anup Dhannawat (B20AI006)

Mentor: V. Narayanan
Course Code: CSN1020

Motivation

- The field of Quantum Computing is developing at a rapid phase and has several applications in various areas.

- In this project, we explored Quantum Computing and Machine Learning, and implemented algorithms that use Quantum Computing for Machine learning tasks.

- The aim of the project was to study and understand Quantum Computation and analyse areas in Machine learning, where Quantum Computing can be used to complete tasks faster than a classical computer.

Objectives

- Understanding Quantum Computation and learn to implement them on a quantum computer.
- Learn various Machine Learning techniques and understand how quantum computing can be of use in areas related to it.
- Integration of Quantum with Machine Learning with appropriate tasks to be carried out.

Work Done

Machine Learning

The project was anything but fundamental and needed an in-depth knowledge of machine learning.

- Elementary Course on Machine Learning on Coursera (Andrew Ng)
- Tutorials from Code with Tim
    - Of course, while learning we made dummy projects as part of the learning
    - Especially various Linear Regression Models and Classifiers**

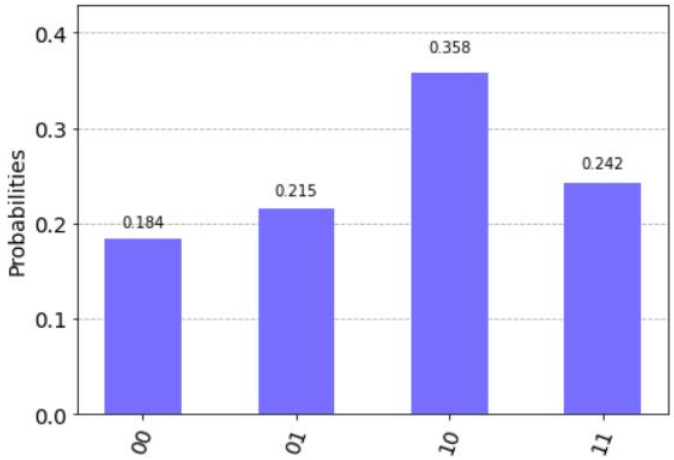
Quantum Computing

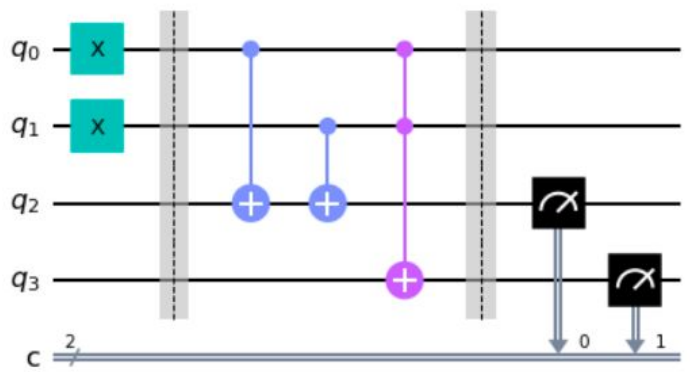
- Quantum Algorithms and Qiskit
- Qiskit Notebook
- Quantum Computation and Quantum Information (Nielsen Chuang)
    - To get started with coding using Quantum Computing, various algorithms and protocols were implemented
    - ...

Work Done

Various protocols that we implemented included: Deutz Jozsa Algorithm, Grovers Algorithm, Vaidman's game, Quantum Fourier Transform and Quantum Random Walk to name a few.

Notable: We were able to implement most, if not all, of these algorithms on real Quantum Hardware.



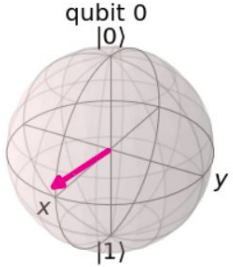



```
print(f" the experimental accuracy rate is : {exp_count['10']/sum(exp_count.values()) :.3f}%")
```

```
the experimental accuracy rate is : 0.358%
```

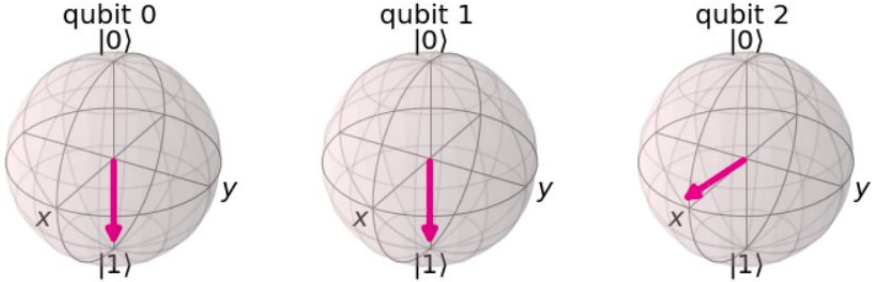
Half adder results

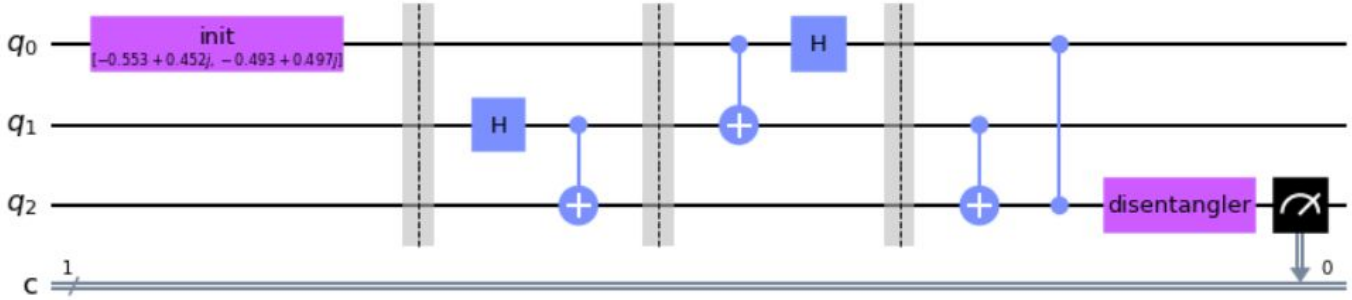
Teleportation Results




```
the experimental error rate is : 0.165%
```

```
##lastly
# backend: ibmq_belem
# Validating: 795ms
# In queue: 3m 42s
# Running: 32.1s
# time in system 4.6s
```

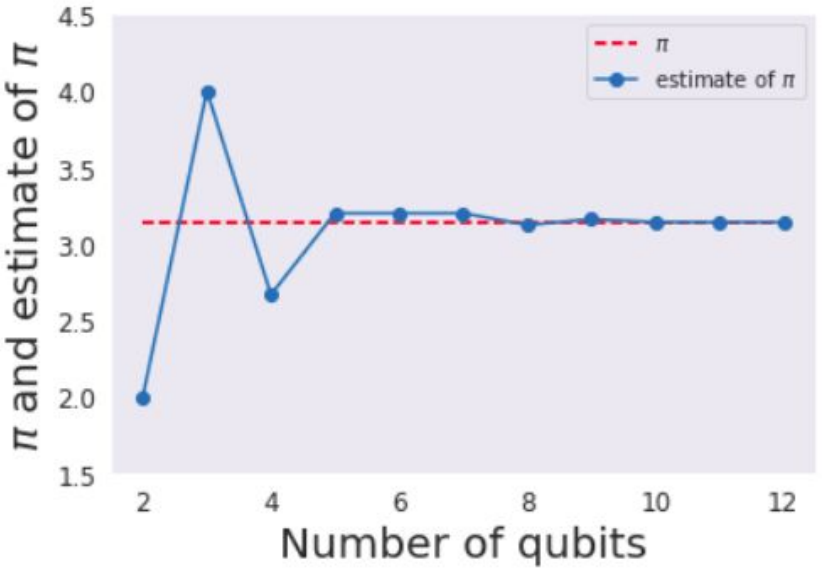




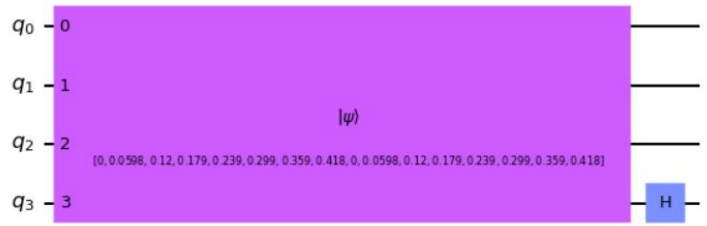
Work Done

- Similarly, in Quantum Coin Game, we were able to implement the circuit and our Oracle was the winner 99.1% times.
- Grover's Search Algorithm ran with 92.5% results in favour of the hardware.
- We also used quantum phase estimation protocol to estimate the value of Pi.

```
Job Status: job has successfully run
2 qubits, pi ≈ 2.0
Job Status: job has successfully run
3 qubits, pi ≈ 4.0
Job Status: job has successfully run
4 qubits, pi ≈ 2.6666666666666665
Job Status: job has successfully run
5 qubits, pi ≈ 3.2
Job Status: job has successfully run
6 qubits, pi ≈ 3.2
Job Status: job has successfully run
7 qubits, pi ≈ 3.2
Job Status: job has successfully run
8 qubits, pi ≈ 3.1219512195121952
Job Status: job has successfully run
9 qubits, pi ≈ 3.1604938271604937
Job Status: job has successfully run
10 qubits, pi ≈ 3.1411042944785277
Job Status: job has successfully run
11 qubits, pi ≈ 3.1411042944785277
Job Status: job has successfully run
12 qubits, pi ≈ 3.1411042944785277
```



Work Done



Quantum Regressor:

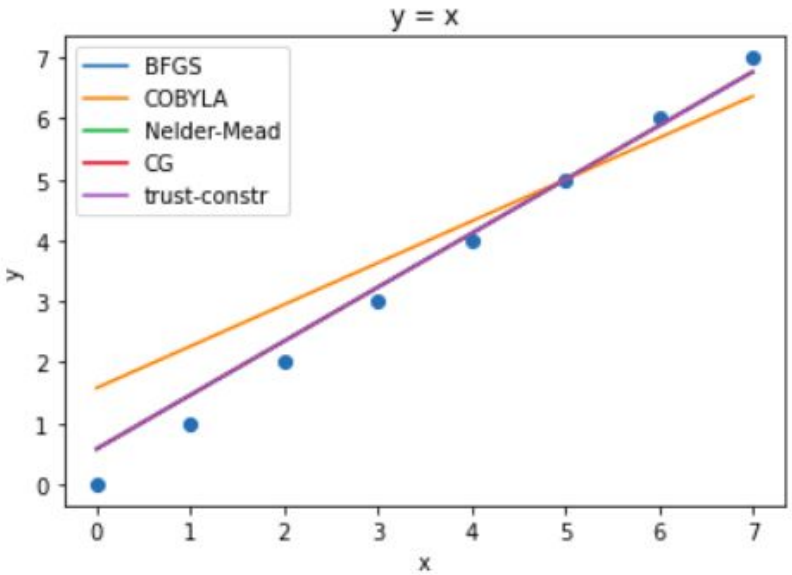
- Structured our pipeline.
- Data drawn
- Implemented a quantum class to calculate inner product of two vectors.
- Implemented a classical parameter optimizer.
- Constructed a hybrid quantum-classical class to take in the data and spit out the regression graph.

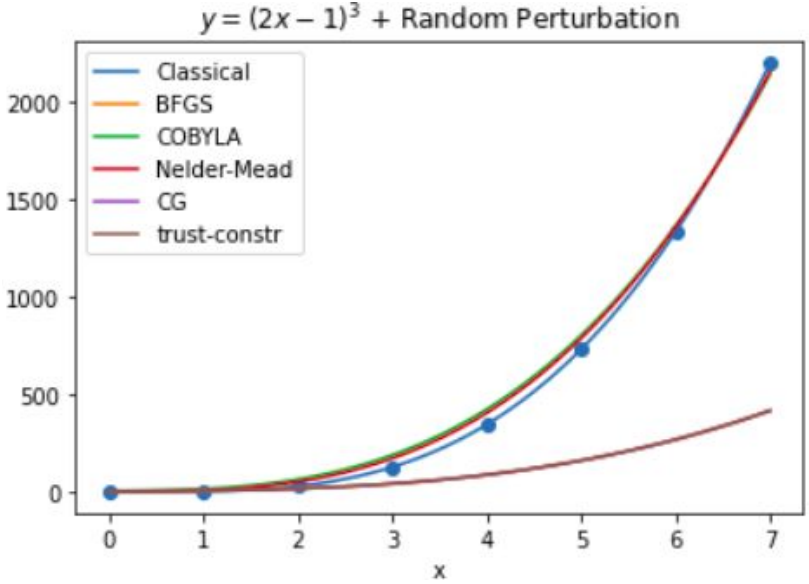
$$|\phi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|x\rangle + |1\rangle|y\rangle\right)$$

$$\left|\tilde{\phi}\right\rangle = \frac{1}{2}\Big(|0\rangle\left(|x\rangle + |y\rangle\right) + |1\rangle\left(|x\rangle - |y\rangle\right)\Big)$$

$$P(0) = \frac{1}{2}\Big(1 + Re\big[\langle x|y\rangle\big]\Big)$$

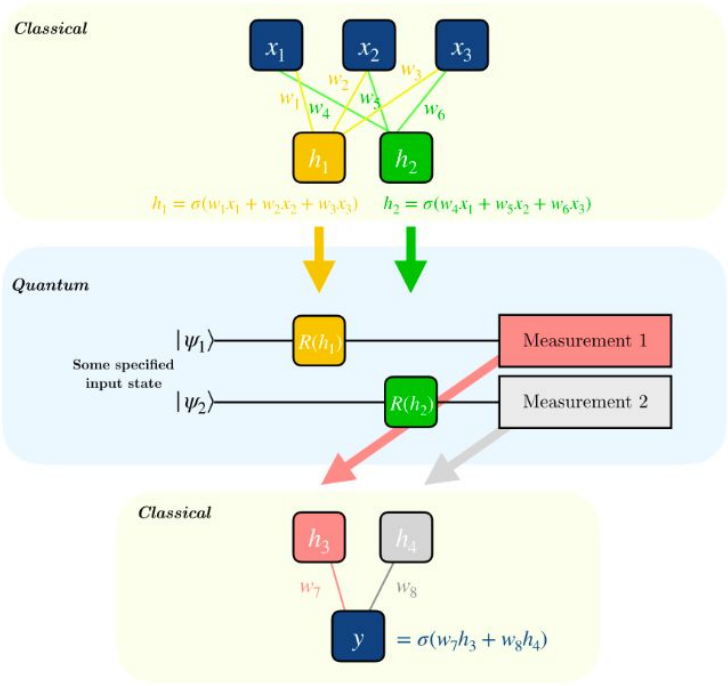
$$Re\big[\langle x|y\rangle\big] = 2P(0) - 1$$

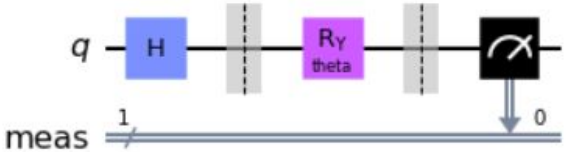

y = x
BFGS
COBYLA
Nelder-Mead
CG
trust-constr
y
x
0
1
2
3
4
5
6
7

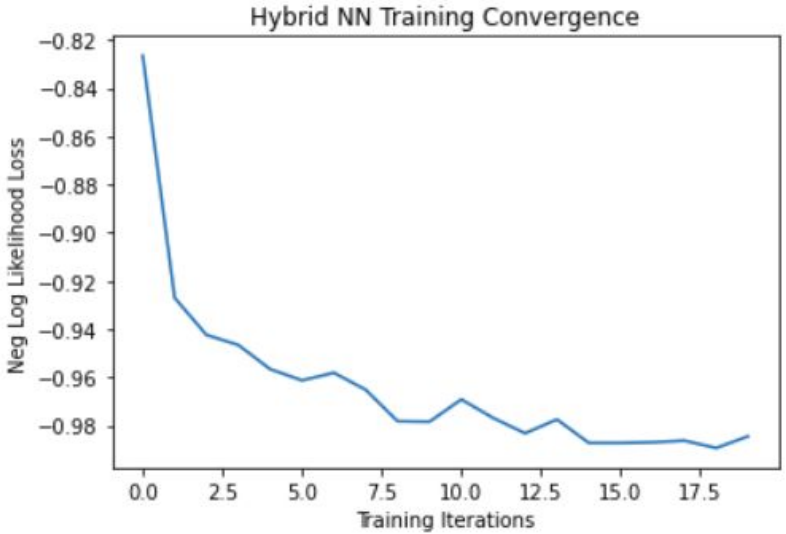

$y = (2x - 1)^3$ + Random Perturbation
Classical
BFGS
COBYLA
Nelder-Mead
CG
trust-constr
x
0
500
1000
1500
2000
0
1
2
3
4
5
6
7

Work Done

Quantum Binary Classifier

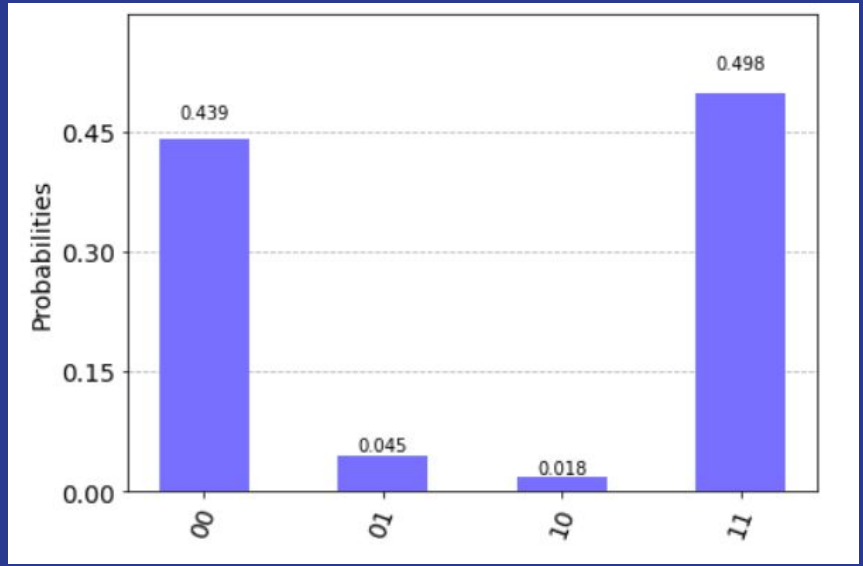






Result and Conclusions

- Learnt to draw quantum circuits.
- Learnt to weight outputs with computation cost and other tradeoffs.
- Did we have any benefit of integrating Quantum computing to our standard machine Learning Algorithms??




```
Performance on test data:
        Loss: -0.9727
        Accuracy: 100.0%
```

Thank You.
:)