Variational Hybrid Quantum-Classical Algorithms and their Implementation in Quantum Machine Learning

Abstract

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. When compared to what is currently available, many quantum algorithms have enormous resource requirements. To overcome this disparity, the "quantum variational eigensolver," a quantum-classical hybrid optimization system , was designed with the notion that even minimal quantum resources might be rendered valuable when used in conjunction with classical processes. We extend the algorithm’s general theory and propose algorithmic improvements for practical implementations

1. Introduction

We initially learned about the fundamentals of Quantum Computing and the way in which qubits(Basic elements of a quantum computer) work.

A qubit is made from a **two-level system**, and the levels can be represented as |0> and |1> (using the Dirac Notation), analogous to the classical bit, which has two states, 0 and 1. But a qubit is more general than a classical bit, a qubit can exist in the state |0> or the state |1>, but it can also exist in a state called a **superposition state**. A superposition state is a state that is a linear combination of the states |0> and |1>. If we label this state as |ψ>, a superposition state is written as:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Where, a,b are complex numbers, and $|a|^2 + |b|^2 = 1$.

While a qubit can exist in a superposition of the states |0> and |1>, whenever a qubit is **measured**, it will only be found to be in the state |0> or the state |1>.

$|a|^2$: Tells us the probability of finding |ψ> in state |0>.
$|b|^2$: Tells us the probability of finding |ψ> in state |1>.

This fundamental difference between a qubit and a classical bit makes it possible to realise efficient information processing excitingly. The fact that the qubit is in more than one state before being measured is used extensively by researchers. Many algorithms have been designed to solve problems using a quantum computer faster than the classical computer(Quantum Supremacy). We also implemented various basic quantum circuits such as half adder, Fourier transform, quantum entanglement etc. using the ‘qiskit’ library in python.

Once we had some basic understanding of Quantum Computation, we started to learn about Machine learning and implemented various Machine Learning algorithms to understand them.

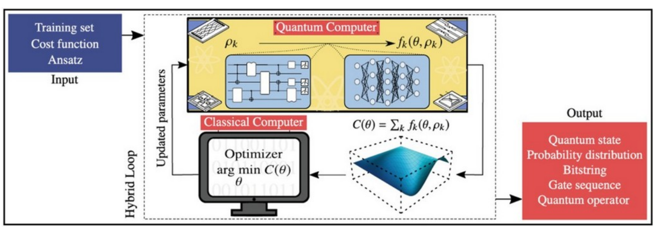
2. Problem Statement

Understanding and analysing how to integrate Quantum Computing with Machine Learning to achieve higher speeds of computation as compared to a classical computer.

3. Methodology Adopted for the Project

Study of many-body Hamiltonians are central to condensed matter physics, molecular dynamics and modeling. Variational methods are an area of extreme importance and current research. But however attractive it may seem to use, with many-body physics and molecular simulations, current and near-future quantum hardware is far from being noise-free. Our best bet is to use a shallow and small quantum-circuit, with high fidelity gates to execute algorithms without noise-correction.

Keeping the idea of Variational Quantum Circuits central to our implementation, we theorized to execute a short sequence of gates where some gates are parameterized. The control reads the result obtained and adjusts the parameters on a classical computer and the calculation is repeated on a quantum-computer using new parameters.



So out of various methodologies like HHL, Ising models and QuBO, finally we decided to use a Variational Quantum Eigensolver as they are best suited for NISQ computers and can be implemented directly on present hardwares.
Having said this, an idealized Quantum computer capable of demonstrating quantum advantage over the current classical ML algorithms is still a thing of the future. Moreover this is an area of active interest. But we know one thing for sure, that the integration of these two domains is quite possible.

4. Justification for Design Choices

Continuing from the previous section, we will run through our implementation to make a Quantum Regressor and a Quantum Binary Classifier and justify our choice of design.

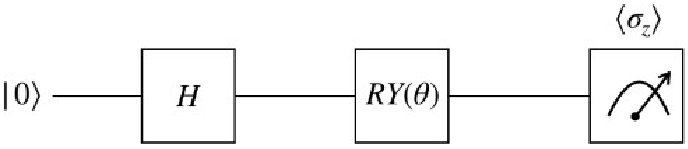
1. In both the cases, a shallow circuit was designed with not more than 4 gates to insure noise-correction is not needed.

2. In both the cases, the circuit was initially run on a simulator to make sure it works and can be implemented on an actual hardware.

3. We kept the data points in both of our examples small as Quantum circuit optimization is yet another independent domain and a larger dataset was taking a very long time to get processed. Consequently, we used 10 data points for our regression problem and 100 for training our Binary Classifier.

4. Further, we cannot work independently with a quantum class. And therefore, we needed a classical class for calculating cost function optimizer.

5. Next, much of the data handling and preprocessing was done classically and a callable Quantum class was defined which implemented the following circuit for our Binary classifier.



6. And following for Quantum Regressor

7. Both, our Quantum Regressor and Binary Classifier could not be run on actual hardware because of Server Time Runouts due to being computationally expensive.

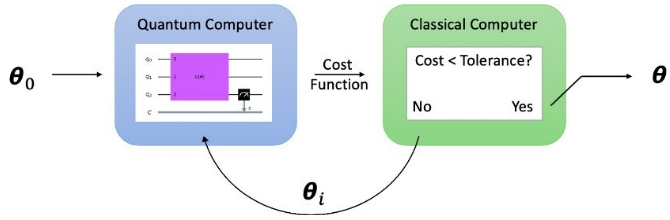
8. To make the simulator call, Qiskit Aer was used and for both the implementations a QASM Simulator was utilized because of its state of the art Noisy Quantum Circuit Simulator Backend.

For the remaining decisions, accepted conventions were used as this domain is still fairly new and the dilemma of choice is rather non-existent.

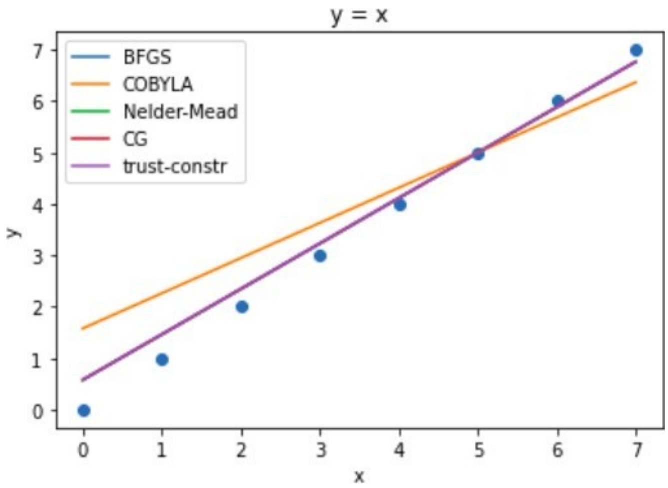
5. Result and Analysis

Variational Quantum-Regressor

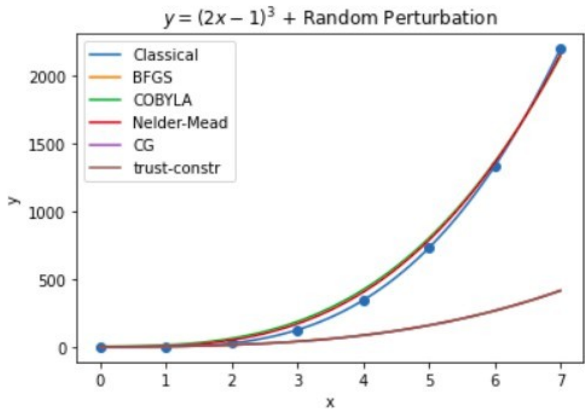
Following the model we discussed previously, we made an Ansatz parameterized by a and b (y=ax+b) . Testing the Ansatz is equivalent to minimizing the cost function. We feed the result of Ansatz to a classical computer, use different optimizers to make a better prediction about our parameters and repeat till convergence within some tolerance is achieved.



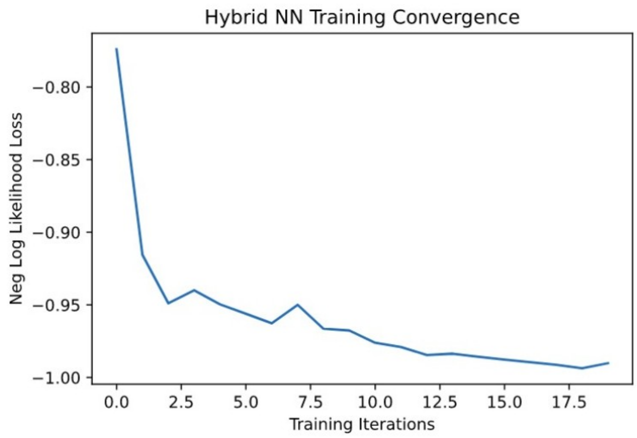
Calculation of loss was done using Quantum Logic while optimization was done classically.



Original data distribution v/s our predictions using different optimizaters

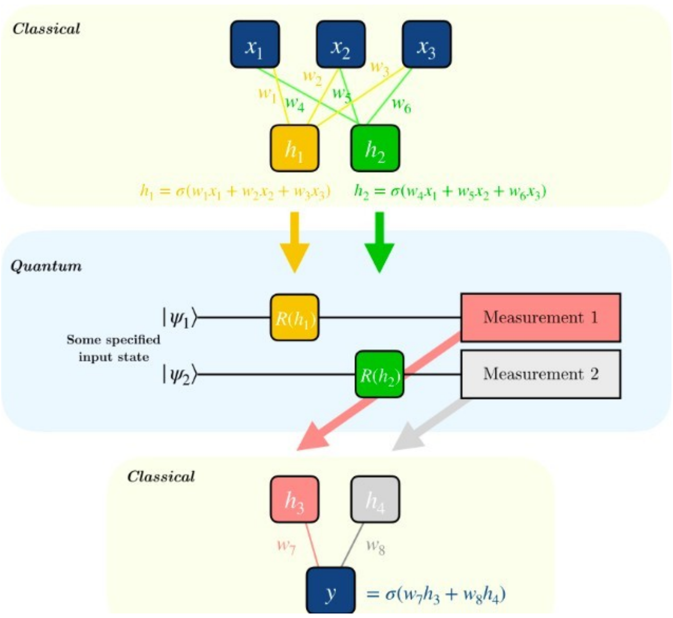


***fresh data was generated and a polynomial regressor was implemented similarly**

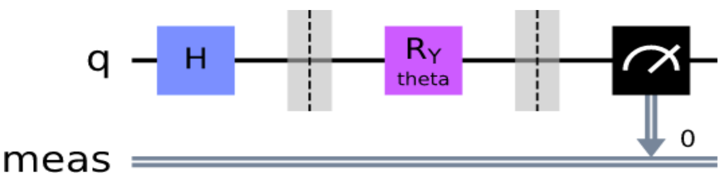


Binary Classifier

For our classifer, we implemented the following pipeline.



We initialized a very standard and straight-forward neural network to classify, except we also added one more hidden layer consisting of a quantum circuit.



Training data was drawn, pre-processed and fed into the Hybrid neural network and following is the resulting loss curve.

Remark: The classical layer of the neural network trains perfectly fine(in fact better) without the quantum layers. Moreover, the quantum layer does not generate entanglement and continues to be classically replicable as we scale this architecture.

6. Conclusion

Machine learning has established itself as a core field which has changed our life for good. But we can see its limiting use case reaching its potential as the size of the data we deal with is growing faster than ever. This is where Quantum computing comes into picture.
Having said that, currently there exists no algorithm or semi-quantum architecture that performs better than the best Machine learning algorithms known. But this is an active area of research and we are sure that sooner or later, we will be having Near- Noise resilient quantum computers which is going to make mathematical burden on our classical processors cake walk and when that happens, Machine Learning shall have its fair share too. Certain elementary tasks can be performed using the present generation of hardware but they are not going to establish quantum supremacy in this domain(unlike other probabilistic domains.
Eg: Simon's Algorithm

But one thing is for sure, that the integration of classical computing and quantum computing will be a harmonious one, ***all we need is a Quantum circuit shallow enough, gates with enough fidelity and qubits coherent enough.***

7. References

[1] Quantum Implementation wrt the project
[2] Quantum Regressor
[3] Quantum Binary Classifier
[4] AER Simulator
[5] Quantum Machine Learning: An Applied Approach
[6]Qiskit ML Tutorials
[7] Quantum Algorithms and Qiskit