KLE TECHNOLOGICAL UNIVERSITY

HUBBALLI

Research Experience for Undergraduates

A Research Project on

Machine Learning on Quantum Environment

Research Project Report submitted in partial fulfilment of 7^{th} semester requirement for

Bachelor of Engineering

in

Computer Science

By

Prajwal A Nazre

Under the guidance of

Dr. Sathyadhyan Chickerur

2020-21

KLE Technological University, Hubballi, India

K.L.E SOCIETY'S

KLE TECHNOLOGICAL UNIVERSITY, HUBBALLI - 580031

School/Department Of

CERTIFICATE

This is to certify that research on *"Machine Learning on Quantum Environment"* is a bonafide work carried out by *Prajwal A Nazre*, in partial fulfillment of the award of Degree of Bachelor of Engineering in Electronics and Communication during the year 2020 – 21. The research thesis has been approved as it satisfies the academic requirement with respect to the research work prescribed for the above said program.

Guide **HoS/H.O.D** **Registrar**

External Viva:

Name of Examiners **Signature**

1.

2.

Acknowledgment

The success and eventual result of this project necessitated a great deal of support and assistance from many individuals, and I consider myself immensely fortunate to have received it during the duration of my project. Much of what I've accomplished has been possible only because of their guidance and assistance, and I'd like to express my gratitude to them. I respect and thank my project guide Dr. Sathyadhyan Chickerur, for providing me all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing his work. I am grateful to Dr. Meena S. M., School Head, Computer Science Department, for her assistance and advice in this project.

- Prajwal A Nazre

Abstract

Quantum Computing as a whole, is a very vast field in Computer Science. It is basically leveraging the concepts of Quantum Mechanics to perform the Computations. It is believed to solve a lot of complex problems with greater efficiency than traditional computing techniques. Machine Learning using Quantum Computing is an interdisciplinary research field formed from the combination of Machine Learning and Quantum Computing. It can be a perfect example to leverage the power of computation provided by Quantum computing techniques. It is believed that many machine learning algorithms could be implemented on Quantum Computers and get better efficiency. My research is to mainly focused on this aspect of the vast field.

Contents

List of Figures

Chapter 1

Introduction

1.1 Quantum Machine Learning, An Overview

In this project, we propose that the Machine Learning implementation on a quantum environment might be the best application of Quantum Computers. Quantum machine learning (QML) has revolutionized performance and speed. Quantum computation refers to a computational paradigm that relies on quantum mechanics' laws. Quantum information technologies and intelligent learning applications are both hot areas and related technologies form the evolution of informatics. The quantum information is now used within machine learning.

Quantum Information (QI) is information that is stored in the state of a quantum system. Quantum information is known by the information is used by the quantum system. Machine learning refers to a data analysis method that can be constructing the analytical model automatically. ML is an essential branch of artificial intelligence that relies on the concept of learning from data; recognize patterns and decision making with minimum human interference. The construction of quantum computers is very difficult and requires complex programs. The quantum has a big impact on the interference or entanglement happen. So the computers relying on quantum propose more efficient solutions for the chosen challenges than the computers based on the classical method.

1.1.1 Nature of Computation

On a basic level, Quantum Objects are nothing but Non-Classical objects with the de-Broglie wavelengths too significant to ignore. These objects have uncertain positions at a specific given time. To predict the end state of a quantum object, we need to take into account all the paths it could have taken to get there.

On doing a number of researches and experiments, it is concluded that directly applying the quantum objects to perform computation increases the speed when compared to the likes of traditional computing techniques. In fact, the results are much faster as the computation gets more complex. This is can be used greatly to our advantage. Quantum computers just follow the laws of physics and taking advantage of it highly depends on how we restate the computational problems in terms of equivalent quantum systems.

1.1.2 Quantum Bits

Before I jumped into Quantum Bits or Q-Bits, I made sure to go through the Logical Gates and Circuits relating to the computers including the basic gates and universal gates which I felt necessary before jumping to Quantum. Knowing the way algorithms working in traditional computers can help understand the relatively new field of Quantum when compared to the mature field of Computation. Quantum computation isn't necessarily computing in the traditional sense but it's just getting the physical laws to work on an environment that is created for a specific reason. The difficult task of constructing a quantum computer is to develop components that obey quantum mechanics laws but can be modified to solve a wide range of problems, not just simulating photon paths with photons or simulating electrons in a molecule by observing the molecule.
Traditional computers work by taking two-state structures called bits as inputs, modifying them according to an algorithm, and then producing one or more bits as measurable outputs. Quantum computation uses qubits, which are discrete quantum particles that behave quantum mechanically and have two or three states

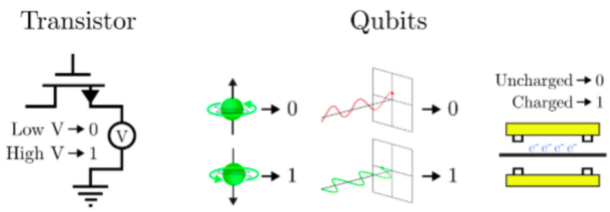

Figure 1.1: Working of Qubits

that can be separated by a calculation, similar to how classical computation uses bits.
The Bloch sphere is a geometrical representation of a two-level quantum mechanical system's pure state space (qubit) which forms the basic building block of a quantum computer.

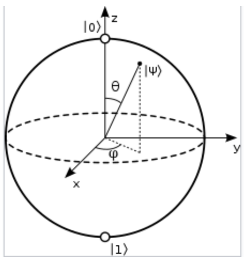

Figure 1.2: Bloch Sphere

1.1.3 Quantum Circuits

The current quantum model is defined in terms of Quantum Logic gates. This can be considered equivalent to linear-algebraic equations in mathematics. Quantum computers would be built using these logic gates. A Sequence of such gates form Quantum Circuits. A memory of n-bits of info can have around 2^n different states. Thus, a vector that represents the memory states has 2^n entries. This is a prob-

Fig. 1.1 Source : Brilliant
Fig. 1.2 Source : Wikipedia

ability vector and it can represent one particular state out of all other possible states.

In a traditional computer, one bit can have only one value and it is a fixated one. In quantum mechanics, the vectors are represented using a density matrix. Let us consider a simple qubit. There are two states in which this memory can be found, the zero state or the one state. Dirac notation is used to describe the status of this memory.

$$|0\rangle := \begin{pmatrix}1\\0\end{pmatrix}; \quad |1\rangle := \begin{pmatrix}0\\1\end{pmatrix}$$

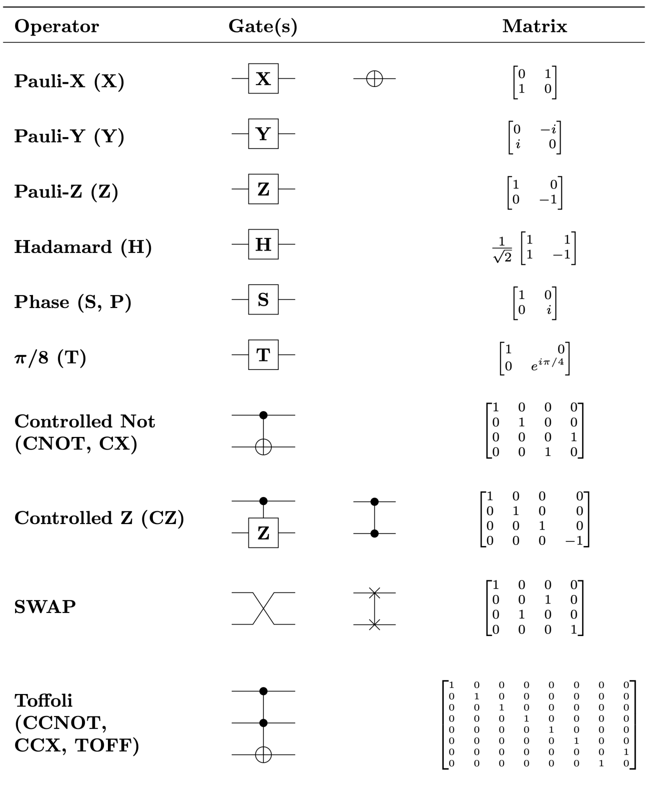

Figure 1.3: Quantum Logic Gates

Among all the gates, I am going to use Pauli-X gate. This gate acts on a single bit. It is a quantum equivalent to a not gate. It indicates rotation of the bloch sphere in the x-axis direction by π radians. Pauli-X matrix is represented as

Fig. 1.3 Source : Wikipedia

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

1.2 Literature Survey

1. TensorFlow Quantum: A Software Framework for Quantum Machine Learning (6 Mar 2020)

   In this paper, they show how to use TFQ to solve advanced quantum learning tasks like meta-learning, Hamiltonian learning, and sampled thermal states in this article. Data of quantum mechanical origin can be represented and generalised using a quantum model. Quantum simulations cannot generalise quantum data using only quantum processors since near-term quantum processes are still small and noisy

   Then they talked about quantum data, which is any kind of data that can be found in a real or artificial quantum device. Quantum data experiences superposition and entanglement, resulting in joint probability distributions that could take an infinite number of traditional processing capabilities to define or store. The quantum supremacy experiment demonstrated that it is possible to sample from a highly complex Hilbert space joint probability distribution. They then talked about TensorFlow Quantum (TFQ), a quantum machine learning library that allows for rapid prototyping of hybrid quantum-classical ML models. Researches conducted on quantum algorithms and frameworks will be using TensorFlow to access Google's quantum computing systems.

2. Unsupervised Machine Learning on a Hybrid Quantum Computer (December 18, 2017)

This paper explains how they practise the quantum computer using the quantum approximate optimization algorithm in combination with a gradient-free Bayesian optimization. They find evidence that classical optimization can be used to train around both coherent and incoherent imperfections using this quantum/classical hybrid algorithm, which is resilient to practical noise. They demonstrated a hybrid quantum algorithm for clustering using a 19-qubit quantum computer. This method is built on the quantum approximate optimization algorithm, which can be used to solve a wide range of combinatorial optimization problems, from image recognition to computer scheduling.

Their implementation relies on Bayesian optimization of classical parameters within the quantum circuit, and they show that the algorithm reaches the optimal solution in many fewer steps than would otherwise be expected by drawing cluster assignments uniformly at random.

3. Quantum Machine Learning: A Classical Perspective (17 January 2018)

   The shortcomings of quantum algorithms, how they relate to their best classical counterparts, and why quantum tools are supposed to have advantages for learning problems are all discussed in this article. Machine learning in the presence of noise and some computationally difficult problems have been described as potential areas for the field. Practical issues such as how to convert classical data to quantum form will be discussed. In this article, a variety of quantum methods for solving learning problems are discussed. Despite some encouraging findings, it is still impossible to infer from the theoretical data provided in current literature that the exponential benefit of quantum approaches in practical learning environments can be achieved.

4. An Introduction to Quantum Machine Learning (September 11, 2014)

Two key approaches to quantum machine learning are presented in this paper. Many scholars attempt to find quantum algorithms that replace conventional machine learning to solve a problem and demonstrate how a complete change can be achieved. This is particularly valid for the closest neighbour, kernel and clustering techniques, where costly distance measurements are accelerated by quantum computation.

Another way to explain stochastic processes is to use the probabilistic interpretation of quantum theory. With regard to the hidden Marcov quantum simulations, this was used to spread the model, though Bayesian theory is often used for truly quantum information tasks such as discriminating against quantum states.

5. Quantum Circuit Learning (April 25, 2019)

   This paper tells us a new hybrid system for machine learning with a low-depth quantum circuit, called quantum circuit learning (QCL). In QCL, the data is provided to a quantum circuit, and the circuit parameters are iteratively tuned to produce the desired performance. They have presented a machine learning framework on near-term realizable quantum computers. Its approach utilises the quantum system's infinitely vast space to combine purely inserted non-linear functions with a low-depth circuit to approximate a complex non-linear function. The results showed that they were capable of representing a function, of classifying and of fitting a relatively large quantum system. The theoretical study has also shown that QCL is able to provide unsolicited methods for dealing with high-dimensional regression or classification tasks on classic computers. We became aware of similar works recently.

6. Learning Quantum Models from Quantum or Classical Data (17 January

2020)

In this paper, we examine how a traditional data distribution can be represented in a quantum system. The suggested approach is to learn quantum Hamiltonian to approximate the classical distribution. We then illustrate how the suggested formalism of quantum learning is also possible in a classical study of results. In addition to conventional statistics, the representation of data as a density matrix rank one adds quantum statistics for classical data. We demonstrate that quantum learners produce results, both for uncontrolled learning and for classification, that can be substantially more reliable than the classic highest probability method.

7. PennyLane: Automatic Differentiation of Hybrid Quantum Classical Computations (Feb 14 2020)

In this paper we learnt about PennyLane, a Python package that extends automatic differentiation to quantum and hybrid classical-quantum information processing. They accomplished this by introducing a new quantum node abstraction which interfaces In this pa- per we learnt about PennyLane, a Python package that extends automatic differentiation to quantum and hybrid classical-quantum information processing. They accomplished this by introducing a new quantum node abstraction which interfaces.

8. Quantum Classification of the MNIST Dataset via Slow Feature Analysis (12 Jun 2018)

This paper shows that quantum computers with quantum memory can be useful when constructing an effective classifier to solve real-world problems. They have shown high accuracy classifications, comparable with the best traditional classifiers, based on the MNIST dataset. Only the dimension and number of data points are logarithmic in the runtime of the quantum process, allowing the quantum classification to be carried out at higher dimensional entries.

9. Classification with Quantum Neural Networks on Near Term Processors (30 Aug 2018)

The presentation of data as quantum overlaps between compute-based states that correspond to different label values is discussed in this article. They prove here that learning is possible by simulation. They intend to use QNN to learn a general quantum state mark. The paper is an exploration of small quantum systems and focuses on the typical simulation. The suggested QNN was planned to accommodate short-term quantum processors. This QNN can then be operated on a quantum computer in a short-term gate model, with the power exploration outside simulation. The paper also showed a general method for classifying classical data using a quantum computer. They map an entry string to a computational basis status, which is seen in the quantum computer, with classic data as inputs.

10. A Study of Complex Deep Learning Networks on High-Performance, Neuromorphic, and Quantum Computers(13 July 2017)

In this article, a common benchmark issue (MNIST) is compared with three separate computer platforms for each baseline score. The findings also remind us that any platform has its own strengths. The quantum approach will model a densely connected network, which is unworkable for conventional computers. The HPC method will analyse CNNs at great scale in order to generate an ideal CNN topology for a particular assignment. The neuromorphic method is capable of producing a low power neural network solution.

11. Q-Means: A Quantum Algorithm for Unsupervised Machine Learning(2019)

    This paper presents q-means, a modern quantum clustering algorithm. It is a quantum variant with a strong algorithm of a k-means that guarantees equal convergence and precision. We also develop a method of selecting the initial centroids corresponding to the k-means++ method.

    This algorithm now has an exponential acceleration in the amount of data point relative to the traditional K-mean algorithm. They also outline the working time of q-means for well-clustered datasets. They have a robust runtime overview for complex datasets and numerical simulations. The theorems and tools presented in this paper are also available for different quantum machine learning implementations in combination with the algorithm. They have proof of successful performance of the q-means algorithm as conventional k-means with a runtime that is slightly smaller than the conventional data-set algorithms.

12. Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits(27 Feb 2020)

    This paper shows that Quanvolutionary transformations of the classical data can improve network precision even in a wider, traditional deep neural network architecture stack. However, this study definitely showed no quantum benefit compared to other non-linear classical changes. This paper shows that the real difficulty would be to decide what characteristics and filters of quanvolutionary filters are both useful and classically challenging to emulate.

13. Quantum Algorithms for Deep Convolutional Neural Network (Nov 4 2019)

    In this paper, they design a quantum convolutional neural network (QCNN) algorithm with a mod- ular architecture that allows for any number of layers, any number and size of kernels, and that can support a large variety of non-linearity and pooling methods. The technical contributions include a new notion of a quantum convolution product, the development of a quantum sampling technique well suited for information recovery in the context of CNNs and a proposal for a quantum backpropagation algorithm for efficient training of the QCNN.

14. Quantum Kitchen Sinks: An algorithm for machine learning on near-term quantum computers (November 22, 2019)

    Here we learn hybrid algorithms to learn computer tasks based on the classic algorithm called random kitchen sinks. This method, known as quantum sinks, uses quantitative circuits to translate classical inputs in a non linear way into functions that can then be used in a variety of machine learning algorithms. This paper shows the potential and versatility of this proposal by using it to address both synthetic dataset binary classification problems and handwritten digits in the MNIST database. They demonstrate, by the use of Rigetti's Quantum Virtual Machinery, that a substantial performance rise over traditional classic linear algorithms provides small quantum circuits, reduced error rate of 50 percent to 0.1percent, and 4.1percent to 1.4percent in both cases.

15. Quantum algorithms for feed forward neural networks (Nov 6 2019)

    In this document we learn quantum algorithms for training on the basis of the canonical classical feedforward and back propagation algorithms and evaluation feedforward neural networks. The operating times of algorithms can be quadratically faster than their classical stan-dard counterparts since they linearly depend on the number of neurons in the network, rather than on the number of neurons as in the classical example. In addition, the networks educated by such a quantum algorithm may have inherent overfitting durability since the algorithm naturally imitates the results of the traditional techniques used for regulating networks, as for example withdrawals.

1.3 Applications

The application of Machine Learning on Quantum Computers are numerous. It is bound to change the computational time exponentially low. Artificial intelligence and machine learning are among the most important areas today, as emerging technologies have penetrated almost every aspect of human life. Some of the common applications we see every day are speech recognition, images classification, and handwriting recognition. However, as the number of applications increased, it became a difficult task for traditional computers to match the accuracy and speed. And, this is where quantum computing can help solve complex problems in much less time, something traditional computers have been around for thousands of years. Here are some of the major applications:

1. Computational Chemistry

   IBM once said that one of the most exciting quantum uses would be computer chemistry. It is thought that even in very small molecules, the number of quantum states is very large and therefore traditional computer memory is difficult to manage. Quantum computers that concentrate on the simultaneous life of 1 and 0 are able to produce huge molecular maps, thus open up a pharmaceutical science association. Some of the important problems that can be solved with quantum computing are: improving nitrogen fixation to make fertilizer based on ammonia; making superconductors at room temperature; removal of carbon dioxide for a better climate; and solid state battery generation.

2. Cybersecurity and Cryptography

   Due to the the amount of cyber-attacks that occur on a regular basis across the world, the online security room has become very fragile. Despite the fact that businesses are putting in place the requisite protection frameworks, the task is daunting and inefficient for traditional digital machines. We are becoming ever more vulnerable to these challenges as our reliance on technology grows. Quantum computing, combined with machine learning, will aid in the development of different strategies to fight these cyber-threats. Quantum computation will also aid in the development of encryption techniques, also known as quantum cryptography.

3. Financial Modeling

   To stay afloat in the economy, the finance industry must find the right balance of profitable investments based on projected returns, risk, and other factors. To do this, 'Monte Carlo' simulations are performed on traditional machines on a regular basis, using a significant amount of computing time. Companies can increase the accuracy of their solutions and also reducing the time it takes to produce them by using quantum technology to execute these large and complicated calculations. Since financial leaders deal with billions of dollars, even a small increase in the anticipated return can be very valuable to them. Another possible use is algorithmic trading, in which computers use complicated algorithms to automatically initiate share trades based on market factors, which is advantageous, particularly for high-volume transactions.

4. Logistics Optimisation

   Thanks to better data gathering and robust modelling, a wide range of organisations will be able to optimise their production line management processes and scheduling workflows. Operating models can monitor and recalculate continuously, which may have a major impact on applications, optimal routes to traffic management, fleet maintenance, air traffic control, freight and distribution. Classical computing is usually used in order to complete these tasks; however, others can become too complex to achieve an efficient computing solution whereas a quantum approach may complete these tasks. Quantum and quantum interlocking are two common quantic approaches for solving such problems. Two common quantum approaches that can be used to solve these problems are quantification and universal quantum computers. Quantum Annealing is a state-of-the-art technology for optimising the output of traditional computers. On the other hand, universal quantum computers can solve such numerical questions and yet do not have a commercial potential.

5. Weather Forecasting

   Weather forecasting takes into account a number of factors, including atmospheric pressure, temperature, and air density, making it impossible to predict correctly. Quantum machine learning can help scientists improve pattern analysis, making it possible for them to forecast extreme weather conditions and eventually saving thousands of lives per year. Meteorologists would be able to produce and analyse more complex climate simulations with quantum computers, providing more information into climate change and potential mitigation strategies. The study of weather conditions in classical computers is currently longer than the weather. However, a quantum computer's ability to easily crush vast quantities of data can help to enhance the modelling of weather systems, allowing scientists to predict developments in the weather at a time of climate change, in realtime and with high precision.

1.4 Motivation

- We became interested in quantum computing because it has the ability to do computations in seconds that would take a conventional machine years to complete.
- Machine learning can exponentially accelerate when a robust quantum computer is created, potentially cutting the time it takes to solve a problem from hundreds of thousands of years to seconds.
- Quantum circuits, a subclass of tensors, can be used to implement complex neural networks.
- It's fascinating to see how this principle can be generalised to Machine Learning, where computations take a long time to complete.

1.5 Problem definition

To implement machine learning algorithm on different quantum simulators and comparing it with classical machine learning model.

1.6 Contributions

The research is carried out using various quantum computer simulators and the quantum machine learning model implemented on those simulators. For constructing quantum models for classification MNIST dataset, we chose two simulators: tensorflow quantum and pennylane. We have used a CNN algorithm to characterise the MNIST dataset, which we compared to two other models that use quantum simulators. After choosing quantum simulators, we must create trainable quantum circuits, which are then used to build the model. The findings are then compared to a traditional standard.

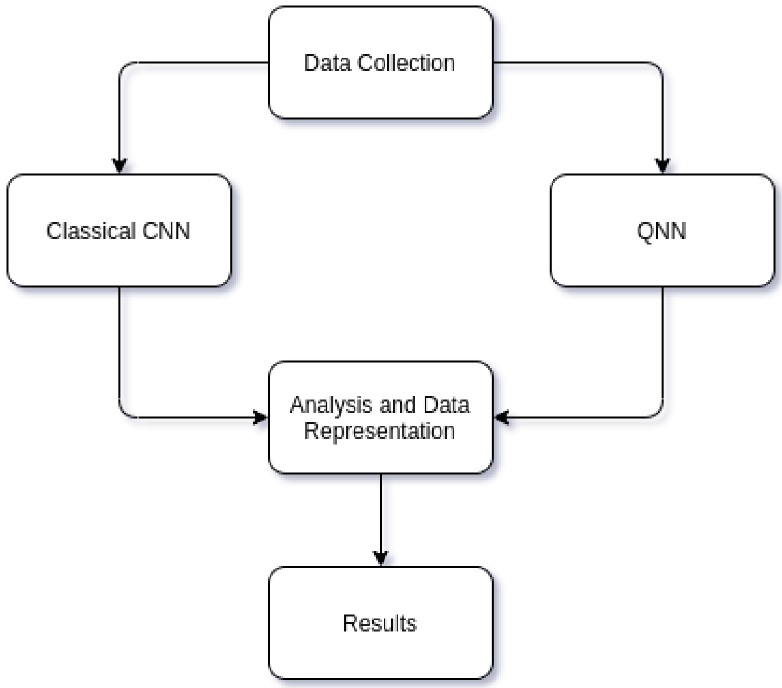

Figure 1.4: Block diagram of proposed framwork

1.7 Organization

- Chapter-1

  Covers the introduction to the research domain, the research applications, and the motivation for choosing the subject. It also discusses the research area's issue description as well as the resulting research issues.

- Chapter-2

  The selection and application of quantum circuits and quantum models on quantum simulators is covered.

- Chapter-3

  The implementation of a classical model to characterise the MNIST dataset as contrasted to a quantum model is discussed.

Chapter 2

Simulating Quantum Computers

The task of simulating quantum computers is a real challenge. Thankfully, there are various platforms and frameworks to help us simulate quantum computations and we can further work on our problem statement of implementing Machine Learning. Some platforms are :

- PennyLane
- Qiskit
- Tensorflow

2.1 PennyLane by Xanadu

2.1.1 Overview

PennyLane bridged the difference between classical and quantum calculations and simplified hybrid calculations. Two famous examples are variational quantum solvers and quantum simulator models.

The basic interface for PennyLane is NumPy, but there are also interfaces for powerful machine learning libraries like PyTorch and Tensorflow. For execution, the quantum computations are sent to a computer. A system may be either actual quantum hardware or a traditional simulator. PennyLane comes with default simulator modules, but it's also well-integrated with third-party quantum circuit simulators including Xanadu's Strawberry Fields, Rigetti's Forest, IBM's Qiskit, and Google's Cirq.

PennyLane's key task is to handle the assessment of parametrized quantum circuits (also known as variational circuits) on quantum devices and making them available to machine learning libraries. PennyLane also gives the machine learning library access to quantum circuit gradients, which it can use to do backpropagation, even by quantum circuits—an important method in optimization and machine learning.

2.1.2 Features

- Follow the gradient. Built-in automatic differentiation of quantum circuits.
- Best of both worlds. Support for hybrid quantum and classical models; connect quantum hardware with PyTorch, TensorFlow, and NumPy.
- Batteries included. Provides optimization and machine learning tools.
- Device independent. The same quantum circuit model can be run on different backends. Install plugins to access even more devices, including Strawberry Fields, Amazon Braket, IBM Q, Google Cirq, Rigetti Forest, Microsoft QDK, and ProjectQ.

2.2 TensorFlow

2.2.1 Overview

TensorFlow Quantum (TFQ) is a quantum machine learning library that allows you to quickly prototype hybrid quantum-classical machine learning models. Researchers working on quantum algorithms and technologies will use TensorFlow to access Google's quantum computing systems.

TensorFlow Quantum is a TensorFlow framework that focuses on quantum data and the development of hybrid quantum-classical models. It incorporates Cirq-designed quantum computing algorithms and logic, as well as TensorFlow-compatible quantum computing primitives and high-performance quantum circuit simulators.

2.2.2 Features

- TensorFlow Quantum implements the components needed to integrate TensorFlow with quantum computing hardware.
- Quantum circuit —This represents a Cirq-defined quantum circuit within TensorFlow. Create batches of circuits of varying size, similar to batches of different real-valued datapoints.
- Pauli sum —Represent linear combinations of tensor products of Pauli operators defined in Cirq. Like circuits, create batches of operators of varying size.
- Sample from output distributions of batches of circuits.
- Calculate the expectation value of batches of Pauli sums on batches of circuits. TFQ implements backpropagation-compatible gradient calculation.
- Simulate batches of circuits and states. While inspecting all quantum state amplitudes directly throughout a quantum circuit is inefficient at scale in the real world, state simulation can help researchers understand how a quantum circuit maps states to a near exact level of precision.

2.3 Qiskit by IBM

2.3.1 Overview

Quantum computing platform Qiskit is an open-source project. It includes tools for designing and modifying quantum applications, as well as running them on IBM Quantum Experience's prototype quantum machines or local computer simulators. It is based on the circuit model for universal quantum computation and can be used for any quantum hardware that meets this model (currently superconducting qubits and trapped ions).

IBM Research developed Qiskit to facilitate software creation for IBM Quantum Experience, a cloud quantum computing service. External donors, mostly from research institutions, make contributions as well.
The Python programming language is used in the main edition of Qiskit. Swift and JavaScript versions were initially investigated, but work on these versions has come to a halt. Instead, MicroQiskit, a lightweight re-implementation of simple features that is easy to port to other platforms, is available.There are some Jupyter notebooks with illustrations of quantum computation in action. The source code for science experiments that use Qiskit, as well as a series of lessons to help people understand the fundamentals of quantum programming, are two examples.

2.3.2 Features

- Research Applications

  Qiskit makes it simple to conduct research and development for particular industry use cases with the greatest quantum advantage potential.

- Collection of Algorithms

  Qiskit contains a generic framework of cross-domain quantum algorithms upon which applications for near-term quantum computing can be built.

- Experimentalist Toolbox

Qiskit's characterization framework offers circuits and analysis methods to understand and characterize the source of noise that impacts our devices. Such parameters include T1, T* , T2, Hamiltonian parameters such as the ZZ interaction rate and control errors in the gates.

- Circuits

  Qiskit provides a set of tools for composing quantum programs at the level of circuits and pulses, optimizing them for the constraints of a particular physical quantum processor, and managing the batched execution of experiments on remote-access backends.

- Simulate Quantum Hardware

  Qiskit provides a high performance simulator framework for the Qiskit software stack. It contains optimized C++ simulator backends for executing compiled circuits, and tools for constructing highly configurable noise models for performing realistic noisy simulations of the errors that occur during execution on real devices.

Chapter 3

Setting up Quantum Environment

3.1 Initial Setup

We must ensure we take care of all dependencies and mechanisms before the implementation begins. Tensorflow 2.3.1 and the Tensorflow Quantum (TFQ) library should be mounted. TensorFlow is an acronym for Tensor Flow. Quantum is a quantum machine learning library that allows for the rapid development of hybrid quantum-classical ML models. Quantum algorithms and applications research will harness the quantum computer architectures of Google, all within TensorFlow. Seaborn, numpy, sympathetic, matplotlib and, most of all, cirq are the required libraries. Cirq is a Python library to write, manipulate, optimise and run quantum circuits against quantum computers and simulators.

3.2 Loading Data

Import the required datasets into the environment (Jupyter Notebook preferred). In my case, it is mnist fashion dataset. The raw dataset contains 60000 training set images and 10000 test set images. This dataset contains 10 labels. As I am doing a binary classification, I am filtering out datas labelled 3 (Dresses) and 6 (Shirts). Now there are 12000 training images with 2000 testing images.

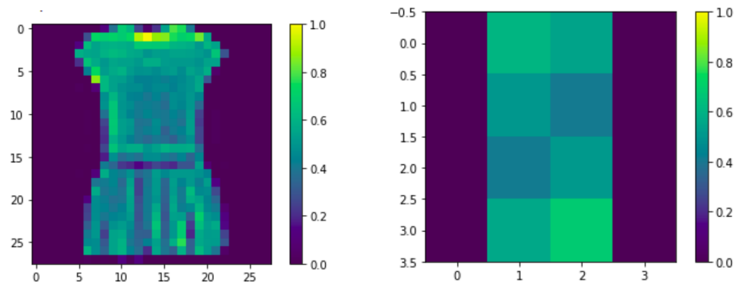

Figure 3.1: Image Quality Reduction

All the images are of the size 28 x 28. The current quantum circuits are limited. Hence, this size is too high for the circuits due to which, downscaling of the images become necessary. I downscaled the image to 4 x 4. Since the image is downscaled to a large extent, there might be images that can be contradictory i.e a same image can be present in both the labels. To make our model stronger, we would need to remove the contradictory images.

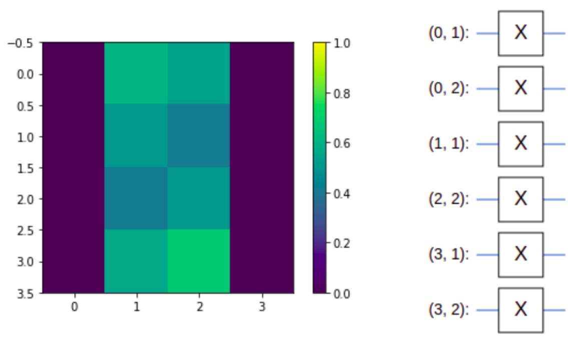

Figure 3.2: Equivalent Quantum Circuit for Image

If each pixel is to be considered as a single qubit then, an image will be converted to a circuit. If a pixel has a value greater than the threshold, it must be have the value opposite to that of the background (MNIST Dataset is grayscaled) which means that pixels that form image should have a corresponding circuit. This can be achieved using the cirq library by applying Pauli-X gate on the required pixels and couple them to form the circuit. And finally, convert the circuits to its equivalent tensors for tfq.

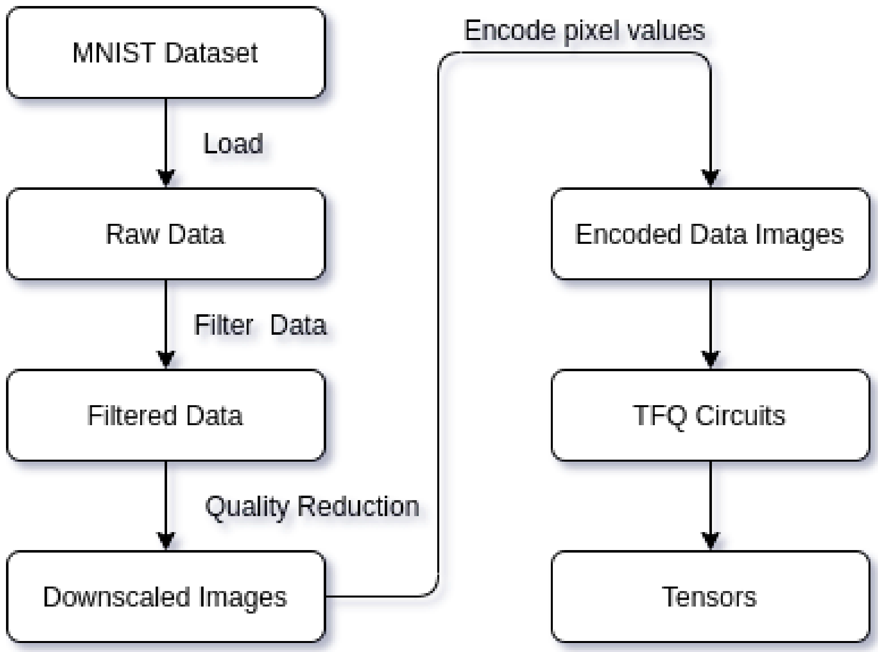

Figure 3.3: Loading process of the Images

Chapter 4

Implementing QNN and CNN

4.1 Quantum Neural Network

We need to build a quantum circuit structure to classify the image. So, lets go with an approach of running a simple RNN across the pixels. We need to initialize 4 x 4 image grid as the data for our QNN. A special qubit known as readout qubit must be created which is used to read the values from the grid. This bit is going to be connected all other qubits. To prepare this bit, I am passing it through one Hadamard Gate(H) and one Pauli-X Gate(X). This makes it eligible for reading out the values of it's connected bit. For this dataset, a double layered QNN would be more than enough. Hence, we must add 2 layers of a double Pauli-X circuit and Pauli-Z circuit. The 'XX' denotes that it is accepting 2 different values. Since, our numpy array of the images are in this form, it would match the input.To end the ciruit, append a final Hadamard Gate for reading. The final circuit would look something like this. Use SVGCircuit to print your circuit.

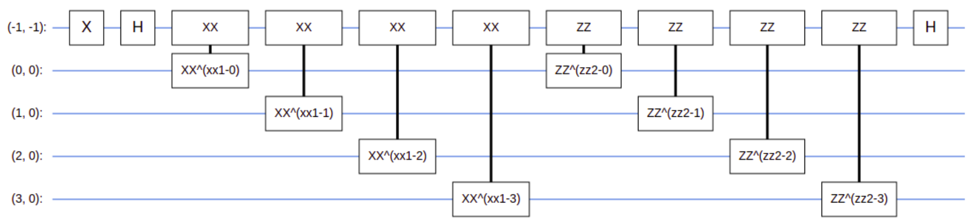

Figure 4.1: Quantum Circuit Model

All the preparations are done now. We have to link the data and the Quantum Circuit Model we built just now. Keras has a special layer type for accepting quantum data and circuits. This special layer is known as Parametrized Quantum Circuit layer to train the model circuit, on the quantum data. This can be accessed using 'tfq.layers.PQC'. Our model and data, both needs to be fed into Keras model. To classify these images, we must take the expectation of a readout qubit in a parameterized circuit. The expectation returns a value between 1 and -1. Then, portray the preparation strategy to the model, utilizing the Compile Method. Since the normal readout is in the range [-1,1], optimizing the hinge loss is required. To utilize the hinge loss, we must make two little changes. First convert the data from boolean to [-1,1], true to form by the hinge loss.

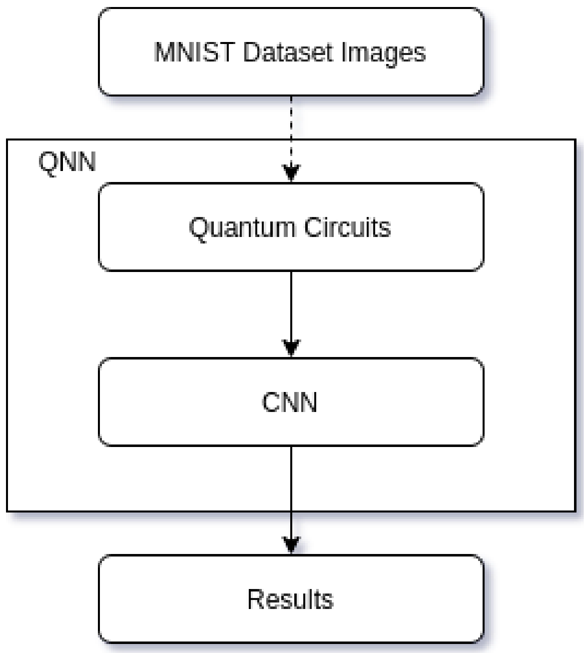

Figure 4.2: QNN Structure

Train the model (takes around 30 mins). On the off chance that you would prefer not to stand by that long, utilize a little subset of the dataset. This doesn't actually influence the model's advancement during training (it just has 32 parameters). Utilizing lesser data simply closes training early, yet runs adequately long to show that it is making progress in the validation logs.

4.2 Convolutional Neural Networks

Multilayer perceptrons are regularised versions of CNNs. Multilayer perceptrons are normally fully networks connected, such that each neuron is connected to all the neurons on the next level in one layer. The ”absolute access” of these networks leaves them vulnerable to data transmission. Regularisation or minimising overfitting in many ways can be achieved, including penalization of fitness parameters (such as weight loss) or trimming connectivity (skipped connections, dropout, etc.) The CNNs have a specific regulatory approach: they use the hierarchical data structure to compile patterns of increasing complexity by using smaller and simpler patterns embossed in their filters.

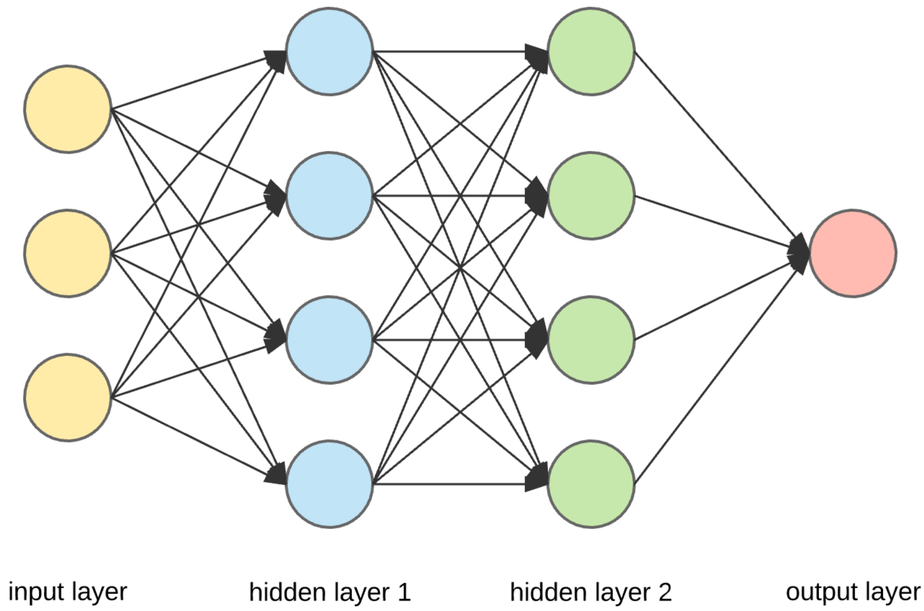

Figure 4.3: CNN Structure

For the sake of comparison, lets build a Classical Machine Learning model. Create a keras model for CNN, add required number of layers and compile the model. Fit the model with the original raw dataset and check for the results and parameters. The model would easily converge at 99.9 percent accuracy since it is a MNIST dataset.

Chapter 5

Results

5.1 Comparison

The QNN produced a result of 69 percent while the CNN produced a whopping 99.9 percent. The QNN was built on the grounds of quantum circuits and considering each image as an indvidual circuit. The computations it undergoes are complex. The loss factors needs to be optimized even further to get better results.

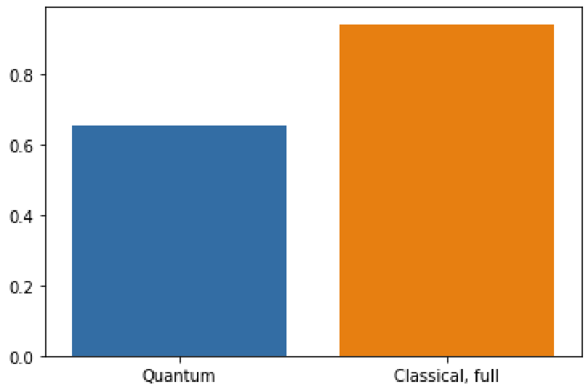

Figure 5.1: QNN vs CNN with Full Data Feed

A question arises as to what would happen when the data fed to both the models would be reduced. To find out, the data was halved and fed to both QNN and CNN. Interestingly, the accuracy of the QNN increased to 73 percent while that of CNN decreased significantly to 86 percent. CNN behaves bad when data fed is low while QNN took a bit of an advantage in this case. So, it is quite probable that QNN might have an application when the initial data is relatively less.

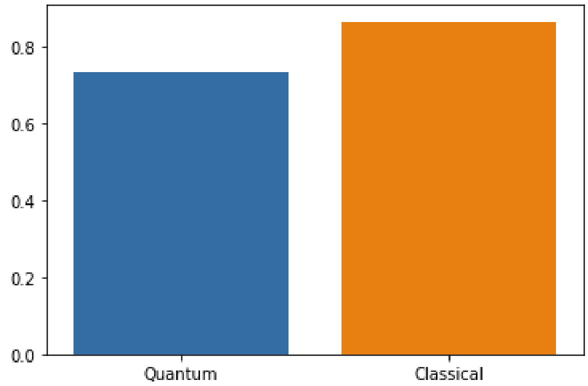

Figure 5.2: QNN vs CNN with Half Data Feed

5.2 Future Scope

Quantum Computers today are still in their infancy. Many big names like IBM, Google, Amazon are trying to reach this domain and come up with methods to leverage the full potential of the Quantum Computers. This research of mine is just a demonstration that Machine Learning on a quantum environment is possible. But, it would take another decade or two to finally implement them on real systems and utilize the phenomenal speed and accuracy. Further research on this is possible as to how we can improve the accuracy of the QNN model. However, all the actual results would have to wait until we perform these on actual Quantum Computers.

Chapter 6

Paper Published/Submitted

Abstract

Quantum Computing as a whole, is a very vast field in Computer Science. It is basically leveraging the concepts of Quantum Mechanics to perform the Computations. It is believed to solve a lot of complex problems with greater efficiency than traditional computing techniques.
Machine Learning using Quantum Computing is an interdisciplinary research field formed from the combination of Machine Learning and Quantum Computing. It can be a perfect example to leverage the power of computation provided by Quantum computing techniques. It is believed that many machine learning algorithms can be implemented on Quantum Computers and get better efficiency. My research is to mainly focused on this aspect of the vast field.

6.1 Introduction

Quantum machine learning (QML) has revolutionized performance and speed. Quan- tum computation refers to a computational paradigm that relies on quantum mechanics' laws. Quantum information technologies and intelligent learning applications are both hot areas and released technologies form the evolution of informatics. The quantum information is now used within machine learning (ML). Quantum Information (QI) is information that is stored in the state of a quantum system. Quantum information is known by the information is used by the quantum system. Machine learning refers to a data analysis method that can be constructing the analytical model automatically. ML is an essential branch of artificial intelligence that relies on the concept of learning from data; recognize patterns and decision making with minimum human interference. The construction of quantum computers is very difficult and requires complex programs. The quantum has a big impact on the interference or entanglement happen. So the computers relying on quantum propose more efficient solutions for the chosen challenges than the computers based on the classical method.

6.2 Working of Quantum Computers

6.2.1 Nature of Computation

On a basic level, Quantum Objects are nothing but Non-Classical objects with the de-Broglie wavelengths too significant to ignore. These objects have uncertain positions at a specific given time. To predict the end state of a quantum object, we need to take into account all the paths it could have taken to get there.
On doing a number of researches and experiments, it is concluded that directly applying the quantum objects to perform computation increases the speed when compared to the likes of traditional computing techniques. In fact, the results are much faster as the computation gets more complex. This is can be used greatly to our advantage. Quantum computers just follow the laws of physics and taking advantage of it highly depends on how we restate the computational problems in terms of equivalent quantum systems.

6.2.2 Quantum Bits

Before I jumped into Quantum Bits or Q-Bits, I made sure to go through the Logical Gates and Circuits relating to the computers including the basic gates and universal gates which I felt necessary before jumping to Quantum. Knowing the way algorithms working in traditional computers can help understand the relatively new field of Quantum when compared to the mature field of Computation. Quantum computer isn't really computing at all. It's just allowing the laws of physics to act on a system designed for a very particular purpose. The challenge of building a quantum computer is designing components that behave according to the laws of quantum mechanics but can be adapted to solve many problems — not just simulating the path of photons using photons or simulating the electrons in a molecule by studying that molecule.

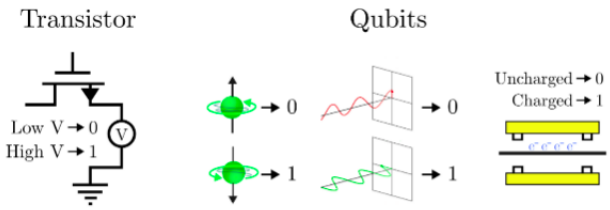

Figure 6.1: Working of Qubits

Classical computers perform tasks by using two-state objects called bits as inputs, manipulating them according to some algorithm, then yielding one or more bits as an output that can be measured. Much as classical computation uses bits, quantum computation uses qubits, isolated quantum objects that behave quantum mechanically, with two or more states that can be distinguished with a measurement. The Bloch sphere is a geometrical representation of the pure state space of a two-level quantum mechanical system (qubit). This forms a fundamental building block of quantum computer.

6.2.3 Quantum Circuits

The current quantum model is defined in terms of Quantum Logic gates. This can be considered equivalent to linear-algebraic equations in mathematics. Quantum computers would be built using these logic gates. A Sequence of such gates form Quantum Circuits. A memory of n-bits of info can have around 2^n different states.

Fig. 1 Source : Brilliant

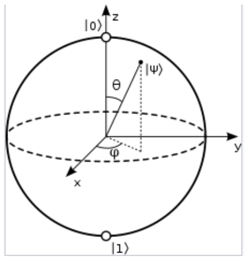

Figure 6.2: Bloch Sphere

Thus, a vector that represents the memory states has 2^n entries. This is a probability vector and it can represent one particular state out of all other possible states.
In a traditional computer, one bit can have only one value and it is a fixated one. In quantum mechanics, the vectors are represented using a density matrix. Let us consider a simple qubit. This memory may be found in one of two states: the zero state or the one state. We represent the state of this memory using Dirac notation

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Among all the gates, I am going to use Pauli-X gate. This gate acts on a single bit. It is a quantum equivalent to a not gate. It indicates rotation of the bloch sphere in the x-axis direction by π radians. Pauli-X matrix is represented as

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

6.3 Simulating Quantum Computers

The task of simulating quantum computers is a real challenge. Thankfully, there are various platforms and frameworks to help us simulate quantum computations and we can further work on our problem statement of implementing Machine Learning. Some platforms are :

- PennyLane

Fig. 2 Source : Wikipedia

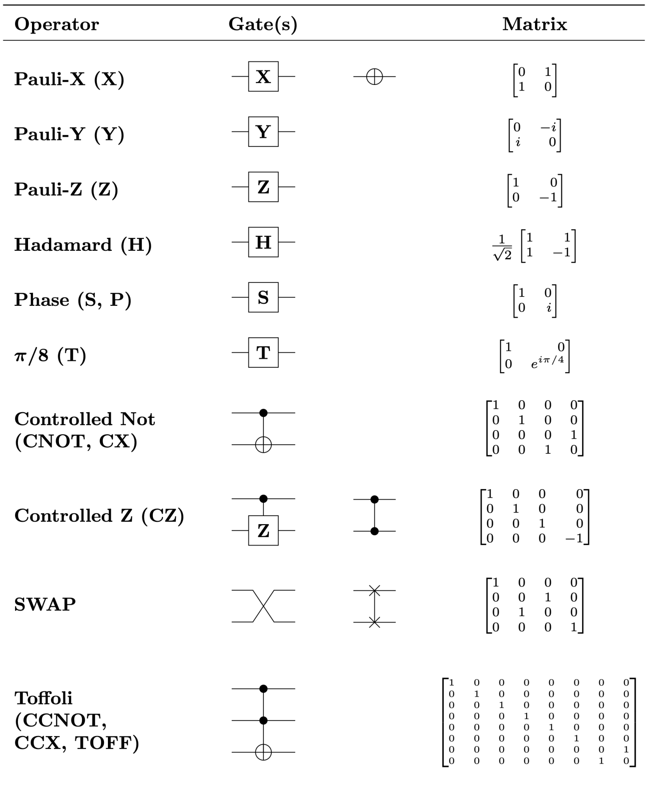

Figure 6.3: Quantum Logic Gates

- Qiskit
- Tensorflow

6.3.1 PennyLane

A cross-platform Python library for differentiable programming of quantum computers. We can train a quantum computer just like a neural network. Some great features are as follows:

- Automatic differentiation of Quantum Circuits
- Hybrid models, connecting Quantum hardware to PyTorch, Tensorflow and Numpy
- Open-Source Tools for Quantum Machine Learning
- Resources and documents to learn and get familiar with the environment

6.3.2 Qiskit by IBM

Qiskit is an open-source SDK for working with quantum computers at the level of pulses, circuits and algorithms. Qiskit accelerates the development of quantum applications by providing the complete set of tools needed for interacting with quantum systems and simulators.

6.3.3 Tensorflow

Tensorflow needs no introduction. This major machine learning open-source framework has been one of the best tools for programming models alongside keras. Recently, Tensorflow Quantum library has been added to its framework which has been phenomenal to all the Quantum enthusiasts. This library is used to prototype hybrid QML(Quantum Machine Learning) models. I have decided to go with this as Tensorflow has provided with a ton of instructions, references and resources to learn from. The fact that it is an open-source makes it free to use where anybody could try simulating Quantum Circuits.

6.4 Implementing ML on Quantum Environment

6.4.1 Initial Setup

Before starting the implemetation, we must make sure that we have taken care of all the dependancies and frameworks. Install Tensorflow 2.3.1 along with Tensorflow Quantum (TFQ) library. TensorFlow Quantum is a quantum machine learning library for prototyping of hybrid quantum-classical ML models. Research in quantum algorithms and applications can leverage Google's quantum computing frameworks, all from within TensorFlow. Some necessary libraries like seaborn, numpy, sympy, matplotlib and most importantly cirq. Cirq is a Python library for

Fig. 3 Source : Wikipedia

writing, manipulating, and optimizing quantum circuits and running them against quantum computers and simulators.

6.4.2 Load Data

Import the required datasets into the environment (Jupyter Notebook preferred). In my case, it is mnist fashion dataset. The raw dataset contains 60000 training set images and 10000 test set images. This dataset contains 10 labels. As I am doing a binary classification, I am filtering out datas labelled 3 (Dresses) and 6 (Shirts). Now there are 12000 training images with 2000 testing images.
All the images are of the size 28 x 28. The current quantum circuits are limited. Hence, this size is too high for the circuits due to which, downscaling of the images become necessary. I downscaled the image to 4 x 4. Since the image is downscaled to a large extent, there might be images that can be contradictory i.e a same image can be present in both the labels. To make our model stronger, we would need to remove the contradictory images.
If each pixel is to be considered as a single qubit then, an image will be converted to a circuit. If a pixel has a value greater than the threshold, it must be have the value opposite to that of the background (MNIST Dataset is grayscaled) which means that pixels that form image should have a corresponding circuit. This can be achieved using the cirq library by applying Pauli-X gate on the required pixels and couple them to form the circuit. And finally, convert the circuits to its equivalent tensors for tfq.

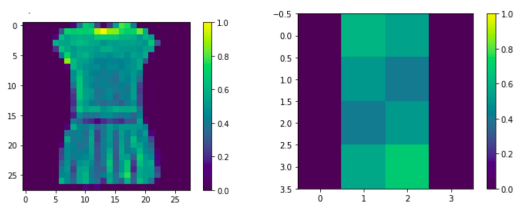

Figure 6.4: Equivalent Quantum Circuits for Images

6.4.3 Quantum Neural Network

We need to build a quantum circuit structure to classify the image. So, lets go with an approach of running a simple RNN across the pixels. We need to initialize 4 x 4 image grid as the data for our QNN. A special qubit known as readout qubit must be created which is used to read the values from the grid. This bit is going to be connected all other qubits. To prepare this bit, I am passing it through one Hadamard Gate(H) and one Pauli-X Gate(X). This makes it eligible for reading out the values of it's connected bit. For this dataset, a double layered QNN would be more than enough. Hence, we must add 2 layers of a double Pauli-X circuit and Pauli-Z circuit. The 'XX' denotes that it is accepting 2 different values. Since, our numpy array of the images are in this form, it would match the input.To end the ciruit, append a final Hadamard Gate for reading. The final circuit would look something like this. Use SVGCircuit to print your circuit.

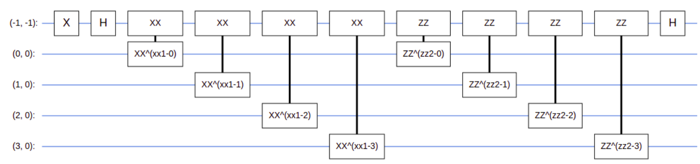

Figure 6.5: Quantum Circuit Model

All the preparations are done now. We have to link the data and the Quantum Circuit Model we built just now. Keras has a special layer type for accepting quantum data and circuits. This special layer is known as Parametrized Quantum Circuit layer to train the model circuit, on the quantum data. This can be accessed using 'tfq.layers.PQC'. Our model and data, both needs to be fed into Keras model. To classify these images, we must take the expectation of a readout qubit in a parameterized circuit. The expectation returns a value between 1 and -1. Then, portray the preparation strategy to the model, utilizing the Compile Method. Since the normal readout is in the range [-1,1], optimizing the hinge loss is required. To utilize the hinge loss, we must make two little changes. First convert the data from boolean to [-1,1], true to form by the hinge loss.
Train the model (takes around 30 mins). On the off chance that you would pre-

fer not to stand by that long, utilize a little subset of the dataset. This doesn't actually influence the model's advancement during training (it just has 32 parameters). Utilizing lesser data simply closes training early, yet runs adequately long to show that it is making progress in the validation logs.

6.4.4 Classical Neural Network

For the sake of comparison, lets build a Classical Machine Learning model. Create a keras model for CNN, add required number of layers and compile the model. Fit the model with the original raw dataset and check for the results and parameters. The model would easily converge at 99.9 percent accuracy since it is a MNIST dataset.

6.5 Results

6.5.1 Comparison

The QNN produced a result of 69 percent while the CNN produced a whopping 99.9 percent. The QNN was built on the grounds of quantum circuits and considering each image as an indvidual circuit. The computations it undergoes are complex. The loss factors needs to be optimized even further to get better results.

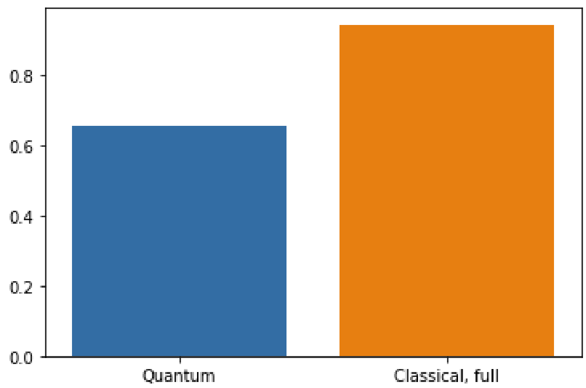

Figure 6.6: QNN vs CNN with Full Data Feed

A question arises as to what would happen when the data fed to both the models would be reduced. To find out, the data was halved and fed to both QNN and

CNN. Interestingly, the accuracy of the QNN increased to 73 percent while that of CNN decreased significantly to 86 percent. CNN behaves bad when data fed is low while QNN took a bit of an advantage in this case. So, it is quite probable that QNN might have an application when the initial data is relatively less.

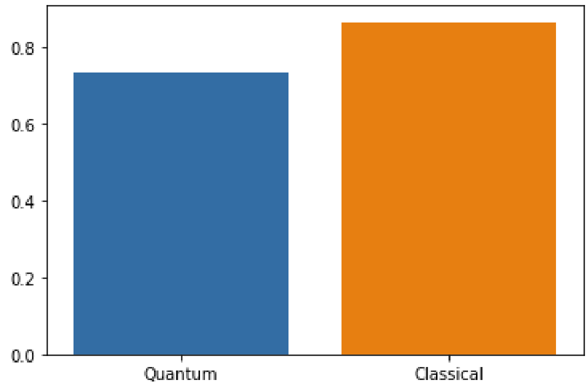

Figure 6.7: QNN vs CNN with Half Data Feed

6.5.2 Future Scope

Quantum Computers today are still in their infancy. Many big names like IBM, Google, Amazon are trying to reach this domain and come up with methods to leverage the full potential of the Quantum Computers. This research of mine is just a demonstration that Machine Learning on a quantum environment is possible. But, it would take another decade or two to finally implement them on real systems and utilize the phenomenal speed and accuracy. Further research on this is possible as to how we can improve the accuracy of the QNN model. However, all the actual results would have to wait until we perform these on actual Quantum Computers.

Acknowledgment

This research was made possible from the guidence of my professor Dr. Satyadhyan Chickerur. I thank him as well as our University for providing me an opportunity to do research on Quantum Computing and Machine Learning aspect of it.

Bibliography

[1] Google Quantum AI. Cirq framework. https://quantumai.google/cirq/. Usage of the cirq library.

[2] Brilliant. https://brilliant.org/practice/quantum-bits/?p=3. Figure 1.

[3] Brilliant. Quantum computing. https://brilliant.org/courses/quantum-computing/. Basics of Quantum Computers.

[4] Alessandro Davide Ialongo Massimiliano Pontil Andrea Rocchetto Simone Severini Carlo Ciliberto, Mark Herbster and Leonard Wossnig. Quantum machine learning: a classical perspective. 2018.

[5] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. 2018.

[6] IBM. Qiskit by ibm. https://qiskit.org/. Knowing the platform.

[7] Iordanis Kerenidis Jonathan Allcock, Chang-Yu Hsieh and Shengyu Zhang. Quantum algorithms for feedforward neural networks. 2019.

[8] Aaron Kalvani. Machine learning on quantum computing in 2020. https://aijourn.com/a-summary-of-machine-learning-with-quantum-computing-in-2020/. Machine Learning on Quantum Computing.

[9] Iordanis Kerenidis and Alessandro Luongo. Quantum classification of the mnist dataset via slow feature analysis. 2018.

[10] Ilya Sinayskiya Maria Schulda and Francesco Petruccionea. An introduction to quantum machine learning. 2014.

[11] Trevor McCourt Antonio J. Martinez Michael Broughton, Guillaume Verdon. Tensorflow quantum: A software framework for quantum machine learning. 2020.

[12] Pennylane. Pennylane by xanadu. https://pennylane.ai/. Knowing the platform.

[13] Rigetti. Forest by rigetti. https://www.rigetti.com/forest. Knowing the platform.

[14] Kevin Bonsor Jonathan Strickland. How quantum computers work. https://computer.howstuffworks.com/quantum-computer.htm. Working of Quantum Computers.

[15] Tensorflow. Machine learning on quantum computing using tensorflow. https://tensorflow.org/.

[16] Tensorflow. Quantum computing on tensorflow. https://www.tensorflow.org/quantum. Usage of Tensorflow Quantum Framework.

[17] Wikipedia. Quantum computing. https://en.wikipedia.org/wiki/Quantum_computing. Basics of Quantum Computing.

[18] Wikipedia. Quantum logic gates. https://en.wikipedia.org/wiki/Quantum_logic_gate. Basics of Quantum Logic Gates.

[19] Wikipedia. Quantum machine learning. https://en.wikipedia.org/wiki/Quantum_machine_learning. Basics of Quantum Machine Learning.

REU Poster

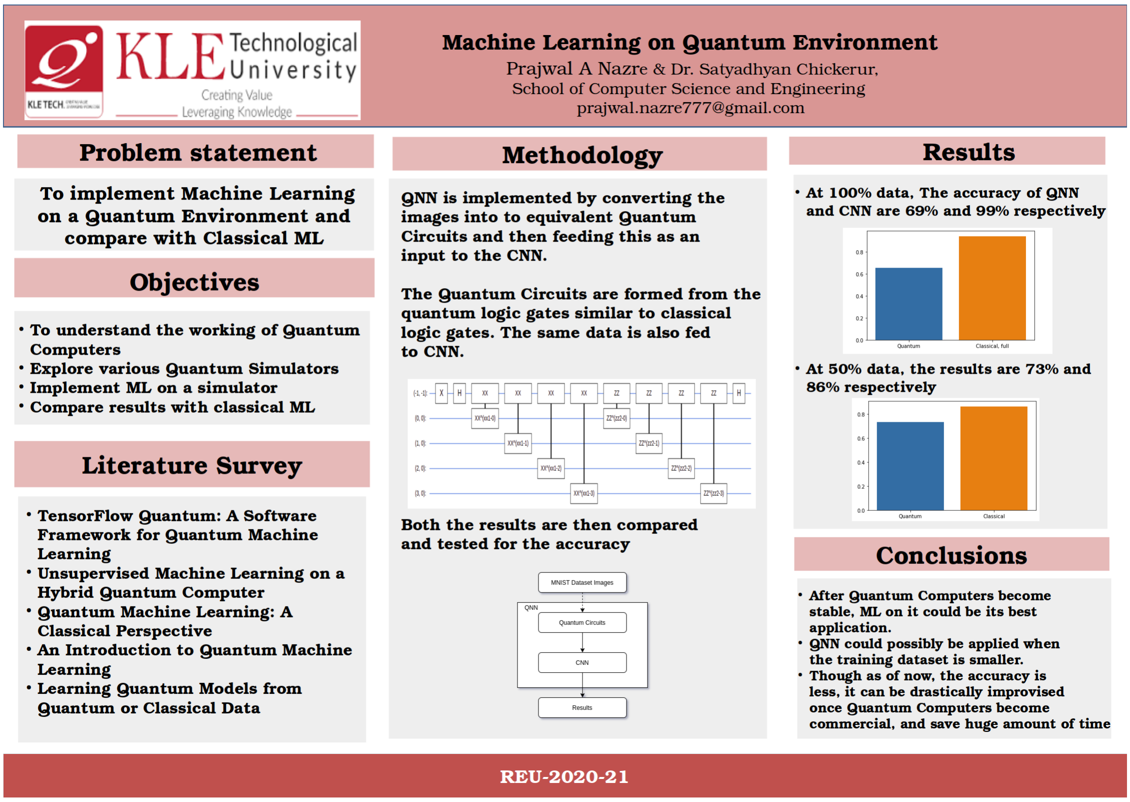

Figure 6.8: REU Poster