North South University

Department of Electrical and Computer Engineering

BACHELOR'S THESIS

Machine Learning In The Realm Of Quantum: The State-Of-The-Art, Challenges, Future Vision and Applications Of It.

Md. Abu Ammar - 1821944042,
Sadia Afrin Tamanna - 1812030042

Supervised by
Dr. Mahdy Rahman Chowdhury
Associate Professor
Department of Electrical and Computer Engineering

Summer 2022

Machine Learning In The Realm Of Quantum: The State-Of-The-Art, Challenges, Future Vision and Applications Of It.

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering

Submitted by

Md. Abu Ammar **1821944042**

Sadia Afrin Tamanna **1812030042**

Supervised by

Dr. Mahdy Rahman Chowdhury

Associate Professor

Department of Electrical and Computer Engineering

Department of Electrical and Computer Engineering
North South University
Dhaka-1229, Bangladesh

July, 2026

Declaration

We, do, hereby, certify that the work presented in this thesis, titled, *"Machine Learning In The Realm Of Quantum: The State-Of-The-Art, Challenges, Future Vision and Applications Of It."*, is the outcome of the investigation and research carried out by us under the supervision of *Dr. Mahdy Rahman Chowdhury*, Associate Professor, Department of ECE, NSU. We also declare that neither this thesis nor any part thereof has been submitted anywhere else for awarding any degree, diploma, or other qualifications. Any material reproduced in this project has been properly acknowledged.

Student's Names and Signatures :

1. Md. Abu Ammar

2. Sadia Afrin Tamanna

Dhaka, Bangladesh
Saturday 4th July, 2026

Approval

The thesis titled **“Machine Learning In The Realm Of Quantum: The State-Of-The-Art, Challenges, Future Vision and Applications Of It.”**, submitted by Md. Abu Ammar, Sadia Afrin Tamanna, in the Summer 2022 session, to the Department of Electrical and Computer Engineering, North South University, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering and approved as to its style and contents on Saturday 4^{th} July, 2026.

Supervisor’s Signature :

Dr. Mahdy Rahman Chowdhury
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Department Chair’s Signature :

Dr. Rajesh Palit
Professor and Chair
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Dedication

This work is dedicated to the Almighty Allah Subhanahu Ta'ala. Also our beloved parents and teachers. This possibly could not be achieved without their help and support.

Acknowledgement

First and foremost, we would like to express our deep and sincere gratitude to our supervisor Dr. Mahdy Rahman Chowdhury, Associate Professor, Department of Electrical and Computer Engineering, North South University for giving us the opportunity to work with him and giving us the required guidance to complete this project. His motivation, vision and ideas allowed us to find this research topic and work on it. I would also like to thank him for his empathy toward us.

We also acknowledge our profound sense of gratitude toward all the faculty members of North South University who have dedicated their time in enriching our knowledge in various philosophies. This enhanced our skills gradually which ultimately boosted our confidence leading to this final year project.

Finally we thank profusely all our friends and relatives that always found a way to support us emotionally, morally, physically or financially.

Dhaka, Bangladesh

Saturday 4$^{\text{th}}$ July, 2026

Contents

List of Tables

List of Figures

Abstract

For data processing, classification, and the clustering area, now a days, Machine Learning has proved to be very effective. Now, it became superior because of its process in many areas. After that, a new field has emerged named "Quantum Machine Learning" that is the combination of Quantum information Processing and classical machine learning. Machine Learning is one of the most over and over again used applications of Quantum Machine Learning. So, here we introduce a comprehensive review of state of the art advances in Quantum Machine Learning. Also, we outline some recent works n different models of the quantum machine learning, and interprets classification tasks in the realm of Quantum with several encoding methods.

1
Introduction

Our understanding of our environment has evolved as a result of quantum physics, and one of its branches, quantum mechanics, has produced accurate and reliable theoretical results. Calculations based on quantum physics are known as quantum computations. This field research is for future use, the quantum behavior of specific subatomic particles (photons, electrons, etc.) when conducting calculations and processing a lot of data. These benefits are accomplished by utilizing quantum properties like entanglement or superposition. These Quantum computers may have the edge over traditional computers in computing speed and cost compared to older computers. As a result of computational complexity (there are more bytes than atoms in the observable universe) or calculation time (thousands of years), there are now scientific problems that cannot be solved by conventional computation. One potential solution is quantum computation. To accomplish these objectives, present quantum devices do not, however, contain enough qubits or are sufficiently fault-tolerant. Moreover, with existing quantum devices, there are other industries like banking, chemistry, or machine learning where quantum computation could be valuable.

Since the 1990s, we have seen numerous highly pertinent algorithms proposed and developed that show a computational advantage over traditional computation. Some of them include Grover's search method [1] and Shor's algorithm [2], both of which greatly outperform traditional machines when it comes to integer factorization in polynomial time. The use of quantum algorithms has recently grown significantly, offering effective solutions in fields including chemistry [3] [4] [5], the solution of linear equations [6], physics simulations [7] [8], and machine learning [9] [10]. The field of machine learning has been one of the fastest-growing sectors. Classical machine learning is utilized to improve quantum procedures, and quantum algorithms have already improved classical processes with existing quantum devices and algorithms.

Recent research demonstrates the detection of quantum entanglement with unsupervised training in fully and partially entangled structures in the context of the use of classical machine learning techniques for enhancement of the quantum world [11]. Additionally, using deep learning approaches, we may analyze the structural characteristics and molecular dynamics of a quantum system [12] or minimize the noise generated by quantum systems [13] The use of hybrid graphical convolutional networks (QGCNN), which can perform better than quantum convolutional neural networks, the classical multilayer perceptron (MLP), and classical convolutional networks, has been proposed for the design and implementation of hybrid deep learning algorithms [11]. Hybrid neural networks have been used to forecast GDP growth in the field of financial applications [14]. Using Boltzmann machines in the healthcare industry to categorize people with lung cancer [15].

1.1 What is Machine Learning?

A subfield of artificial intelligence (AI) and computer science called machine learning focuses on using data and algorithms to simulate how humans learn, gradually increasing the accuracy of the system. We want to assign a label to an unlabeled item using machine learning. And there are three main approaches: segmentation, regression, and classification.

1.1.1 Classification

When classifying an instance, we attempt to anticipate its discrete label. Which one is it given the input and a list of potential labels? Here is a photo. Is that a dog or a cat? [16]

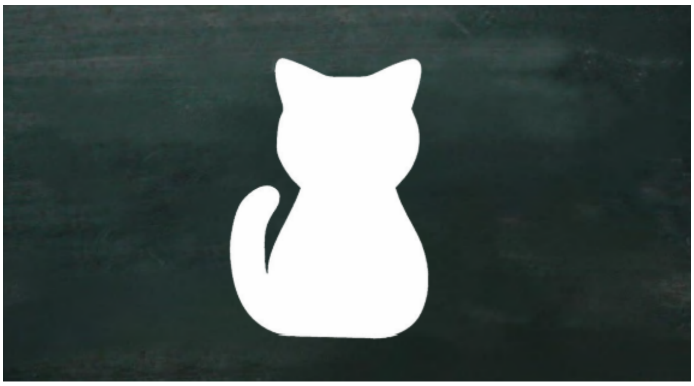

Figure 1.1: Is it dog or cat?

1.1.2 Regression

In regression, the goal is to identify a function that may be used to forecast the relationship between an input and a continuous dependent output value. For instance, even though you don't know how to calculate your tax rate, can you approximate it given your income and your friends' income and effective tax rates? [16]

1.1.3 Segmentation

Segmentation is the process of dividing the population into groups that have similar traits and are thus likely to behave similarly. Machine learning's use of customer segmentation can reduce waste, allowing us to spend less on marketing campaigns. If we know which consumers are similar to one another, we will be better equipped to target our advertising to the right people.

1.2 What is Quantum Computing?

A rapidly expanding area of research is quantum computing, which merges computer science with quantum mechanics. Feynman noted in 1982 that a quantum computer is required, or the computer must operate in the realm of quantum mechanics, in order to simulate a quantum system (QC). In 1993, the first suggestion for the actual application of a QC was made. In a QC, the quantum bit serves as the fundamental unit of quantum information (qubit). It is possible to think of a single qubit as a two-state system with two levels, such as an atom with a spin-half. The capacity of quantum systems to exist in superpositions of their fundamental states underlies the potential power of a QC. The basic states indicate a set of numbers that can all be changed at once [17].

The following fundamental prerequisites must be met in order to perform quantum computations: Qubits must meet the following requirements:

- A two-level system $|0\rangle$ and $|1\rangle$ as a qubit.
- The ability to prepare the qubit in a specific state, such as $|0\rangle$.
- The ability to measure each qubit.
- The construction of fundamental gate operations like the conditional logic gate (the control-not gate).
- And a long enough decoherence time. Since environmental interactions obliterate the superposition of states, it is crucial for a QC to be properly insulated from them. Quantum error corrections, which were developed recently, are also necessary.

The quantum system cannot be in more than one state at once. It is found in a complicated linear combination of states 0 and 1. It is an alternative combination that falls outside of either ”or” or ”and”. A unique type of computation is quantum computing. It makes use of the three essential concepts of superposition, interference, and entanglement in quantum physics. In this book, we'll go into great details about these concepts in chapter 3 but for now we are just introducing these concepts in a brief.

1.2.1 Superposition

When a quantum system may exist in numerous states simultaneously, this phenomenon is referred to as superposition.

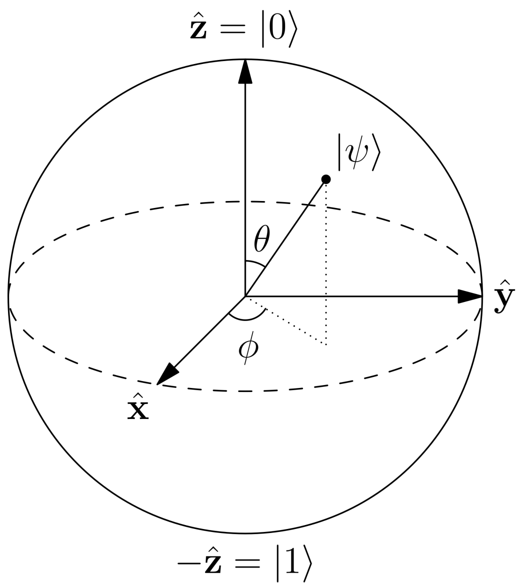

Figure 1.2: Quantum Superposition

Where $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + \exp^{i\phi}\sin\frac{\theta}{2}|1\rangle$.

1.2.2 Interference

The ability to bias quantum systems in the desired direction is due to quantum interference. The objective is to produce an interference pattern in which the paths leading to incorrect responses interfere destructively and cancel out, while the paths leading to correct answers reinforce one another. An example of interference can be like the equation

Figure 1.3: Applying double Hadamart gate on a qubit to create Interference.

1.2.3 Entanglement

A very strong correlation between quantum particles is called entanglement. Even when separated by enormous distances, entangled particles maintain their flawless correlation.

The maximally entangled Bell-States $|\phi^{\pm}\rangle$ and $|\psi^{\pm}\rangle$, are the model entangled states of a two-qubit system q_1q_2, phrases that are stated as a statement of the computational basis is as follows equation 1.1 and 1.2:

$$|\phi^{\pm}\rangle = \frac{|0\rangle_{q_1}|0\rangle_{q_2} \pm |1\rangle_{q_1}|1\rangle_{q_2}}{\sqrt{2}} \tag{1.1}$$

$$|\psi^{\pm}\rangle = \frac{|0\rangle_{q_1}|1\rangle_{q_2} \pm |1\rangle_{q_1}|0\rangle_{q_2}}{\sqrt{2}} \tag{1.2}$$

A Bell-Pair or EPR (Einstein-Podolsky-Rosen) Pair [18] is another name for the two qubits that make up a Bell-State.

1.3 What is Quantum Machine Learning?

The incorporation of quantum algorithms into machine learning is known as quantum machine learning. The term is most frequently used to describe machine learning techniques used on a quantum computer to analyze classical data. Quantum machine learning makes use of qubits, quantum processes, or specific quantum systems to speed up computing and data storage performed by algorithms in a program while machine learning methods are used to calculate enormous amounts of data. This includes hybrid processing techniques that combine classical and quantum computing and outsource computationally demanding subroutines to a quantum device. On a quantum computer, these operations can be carried out more quickly and with greater complexity.

The basic building block of quantum machine learning more specifically say quantum neural network is given in fig 1.4, the concept of quantum machine learning and quantum neural network briefly discussed in chapter 3 and 4.

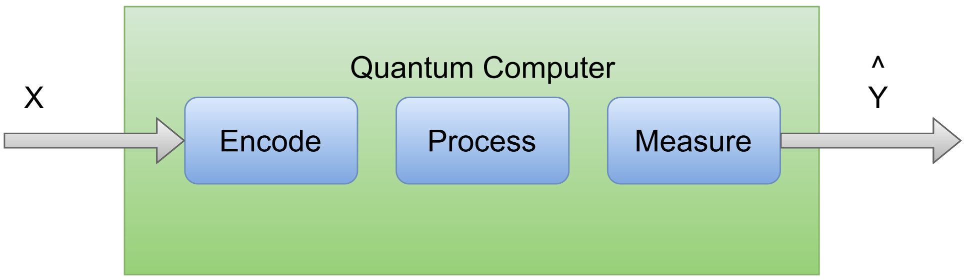

Figure 1.4: Building block of Quantum Machine Learning

1.4 Background of the research

Recent years have seen the rise of algorithms that can play chess and go better than world champions. Almost every industry that can be imagined is using machine learning, including the military, aerospace, manufacturing, agriculture, banking, and healthcare.

But because these algorithms have billions of parameters, training them gets harder and harder. Such issues are unsolvable with present computing technology, but quantum computers promise to address them. Additionally, because they can calculate several states at once, they can carry out an infinite number of superposed activities concurrently. a skill that has the potential to accelerate and enhance machine learning methods.

Quantum computing, as opposed to traditional computers based on sequential information processing, takes advantage of the superposition, entanglement, and interference phenomena in quantum physics. But it decreases the amount of processing power required to solve a problem rather than increasing the amount of computing power accessible.

However, quantum computing calls for a shift in how we view computers. A completely new set of algorithms are necessary. algorithms that use and encrypt quantum information. Algorithms for machine learning are included. Additionally, it needs a fresh group of developers. developers with an understanding of quantum computing and machine learning. Developer's ability to find solutions to real-world issues that have never been encountered. a unique kind of developer. You are already unique from everyone else since you can tackle challenges using quantum machine learning.

The disruptive potential of quantum machine learning is great. Although this integration of machine learning and quantum computing, two areas of active study, is largely of a conceptual nature, there are already some examples where it is being used to address practical issues. Many high-tech startups, including Google, Amazon, IBM, Microsoft, and others, compete to develop and market the first quantum machine learning systems.

It's a rare chance to examine a technology just as it is about to demonstrate its superiority [16].

1.5 Why Quantum in Machine Learning?

1. It has more effective dimension than classical ML models.
2. Quantum Computers as AI accelerators like gpu.
3. Theoretically it reduce the cost faster.
4. and so on

1.6 Scope of the Quantum Machine Learning

A researcher may now conduct more research with large amounts of data thanks to the past fifty years' rapid advancement in computing technology and the accessibility of high-performance computing devices. IBM recently succeeded in creating quantum processor-based computers that are much quicker than the ones in use today. Computing systems based on quantum mechanics typically include a quicker quantum bit than a binary bit. Therefore, compared to ordinary 64-bit computing equipment, a computer based on quantum computing can read and process large amounts of data much more quickly. Similar to this, existing traditional machine learning techniques based on binary bit manipulation perform poorly with large amounts of data. It is also predicted that if quantum processor-based computers become commercially viable, they would greatly assist numerous industries, and the field of quantum machine learning will be greatly opened to fresh ideas for resolving complicated issues in the future. This presentation illustrates the theories, architectures, and model construction in quantum machine learning.

1.7 Organization of the thesis

The research work performed in this study is divided into different topics and presented within six chapters including this chapter. The first Chapter of this research describes the background of the study, the scope of Quantum Machine Learning, and why Quantum Computing in Machine learning? The methodology of this study is also outlined in this chapter briefly.

The Second Chapter provides an extensive literature review based on the recent work that has been done on quantum machine learning area.

Chapter Three focuses on the key aspect of the quantum computing and quantum machine learning. This chapter will provide a basic but depth knowledge of quantum computing, quantum machine learning in contrast with the classical computing and classical machine learning. This will be an easy start for non quantum background people to learn about quantum computing and later on quantum machine learning.

The Fourth Chapter consists of the methodology of our research. Why and how we have build the models. What each quantum machine learning model.

The Fifth Chapter of this research is about the result analysis of our research where we will be folding some research question related with quantum and classical machine learning.

And finally the Sixth Chapter of this research will conclude the conclusion of our research.

2
Related Work

This chapter describes the existing work in quantum machine learning that we had found to be helpful. We tried to find out some books and papers related to this concepts and tried to find out what types of works or innovation has been done on Quantum Machine Learning.

Big data management demands a lot of time, and big data classification suffers from this limitation as well [19] [20] [21], making quantum computing-based classification a viable solution. The Quantum Least Square Support Vector Machine (Quantum LS-SVM) is one of the investigated classifiers motivated by quantum mechanics [19]. While the Library for Support Vector Machine (LIBSVM) on a classical computer has classification rates of 86.46% and 84.90%, respectively, Quantum LS-SVM has average values and standard deviations of 91.45% in low-rank datasets and 89.82% in low-rank approximate datasets.

Additionally, a quantum kernel machine can perform quantum evaluation directly in a higher-dimensional space using a quantum big data method (i.e., non-sparse matrix exponentiation for matrix inversion of the training data inner-product matrix) for implementation on a quantum computer [21]. Quantum Multiclass SVM is a different strategy that is based on the quantum matrix inversion algorithm and the one-against-all tactic. This method converts datasets into quantum states, employs QRAM for quantum parallel data access, and then exploits memory access incoherent quantum assumption to achieve quadratic speed gains over classical computer systems [20].

On the D WAVE 2000Q Quantum Annealer machine, binary classification for remote sensing (RS) of multispectral pictures is possible using quantum SVM [22]. On two ID datasets, Im16 and Im40, this method applies the RBF kernel method and formulates the classification as a Quadratic Unconstrained Binary Optimization (QUBO). For the Im16 dataset, the approach obtained an AUROC score of 0.886 and an AUPRC score of 0.930, respectively. For the other dataset Im40, AURCOC of 0.882 and AURPC of 0.870 were obtained, respectively [22]. On 50 samples from the SemCity Toulouse dataset, similar RS testing for image classification on the upgraded quantum machine D-WAVE Advantage produced an overall accuracy of 0.874 with 0.734 F1 score, which was comparable to classical SVM models and outperformed the IBM quantum machines, which lagged with 0.609 and 0.569 scores, respectively [23]. The Wisconsin breast cancer dataset was used to test QSVM with RBF kernel and SVM (the traditional equivalent) [24]. The accuracy of the QSVM, which was built on the Qiskit Aqua with a real backend quantum-chip (ibmqx4), was 80%, while that of the classical SVM was 85%. The study discovered that employing QSVM on a simulator outperformed the conventional method by achieving nearly flawless accuracy. The same study used a quantum multiclass variational SVM on the UCI ML Wine dataset and was able to obtain 100% accuracy on the StateVector simulator and 93.33% accuracy on the iqmqx4 while only achieving 90% accuracy with a classical SVM running on a local CPU context. [25]. Several datasets, including Fisher's Iris dataset, modified Iris dataset, Sonar dataset, and Wisconsin's Breast Cancer dataset, were subjected to quantum neural network (QNN) applications using the single-shot training scheme, which enables input samples to be trained in a single qubit quantum system [26]. The QNN outperformed a traditional NN with no hidden layers, delivering accuracy of 83.26%, 96.96%, 41.25%, and 90.19%, respectively [26]. However, the classical NN outperformed the QNN when two additional hidden layers were added to the architecture.

By substituting SHIFT operations for multiply operations on the ResNet, DenseNet, and AlexNet network structures, a data structure known as n-BQNN—which contains the learning framework of n-bit QNNs—can achieve an accuracy that is nearly identical to that of full-precision models while being 68.7% more energy efficient and 2.9 times faster than SVPE (shift vector processing element) [27].The ideal weights of a neural network are also discovered by a variant of Grover's quantum search technique known as BBHT optimization, which also more effectively trains a QNN for data categorization [28]. Through the activation of a Perceptron, this model is created.utilizing a step activation function, the initial qubit of the inner product of input and neuron weights is measured using quantum Fourier analysis modification [28].

To reliably display the optimal route from a certain source to destination, dynamic traffic routing can be calculated by extracting real-time data from devices on GPS landmarks, which is then preprocessed in Spark SQL and then processed by a combination of Decision tree and Random Forest [29]. In tests using the Iris Dataset, QNN's testing accuracy of 97.3%, 97.5%, and 85.5% for matching training pairings of 75, 30, and 12 was comparable to that of CVNN and RVNN's single hidden layer classical neural networks [30]. However, when compared to CVNN, which ran for 1000 epochs, and RVNN, which lasted for 5000, QNN's computational speed was 100 times faster. In addition, Variational Quantum Classifiers (VQC) based on Quantum Random Access Coding (QRAC) have been used to improve performance and efficiency by using fewer qubits on the Breast Cancer (BC) dataset and Titanic Survival (TS) dataset, respectively, with test accuracy and f1 scores of 0.682 and 0.483 for BC and 0.772 and 0.707 for TS dataset [31]. The accuracy of QNN4EO, which is made up of three convolutional 2D layers and utilized for picture classification, was higher than that of CNN (93.63%) in the Earth Observation (EO) dataset known as EuroSat [32].

3
Theory

The details of the theory of our system are discussed in this chapter. The theoretical explanation is divided into two sections

- Fundamental Of Quantum Computing.
- Fundamental Of Quantum Machine Learning.

3.1 Fundamental Of Quantum Computing

Before we move on to the concept of Quantum computers we have to know one of the most common notations that are used in quantum mechanics and quantum computing, the Dirac notation. It was invented by Paul Dirac.

Lets take a vector in 3-dimensional complex space. We can write this as a column vector like this:

$$r = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

In quantum mechanics, we write the column vector in Dirac notation as ket like this:

$$|r\rangle = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

But kets are special:

- The values (a, b, and c above) are complex numbers (they may be real numbers, imaginary numbers, or a combination of both)
- A ket is a quantum state.
- Kets can have any number of dimensions, including infinite dimensions!

The ”bra” is similar, but the values are in a row, and each element is the complex conjugate of the ket’s elements.

$$\langle r| = \begin{bmatrix} a^* & b^* & c^* \end{bmatrix}$$

3.1.1 Hilbert Space

In quantum mechanics the state of a physical system is represented by a vector in a Hilbert space :a complex vector space with an inner product. The term “Hilbert space” is often reserved for an infinite-dimensional inner product space having the property that it is complete or closed. However, the term is often used nowadays, in a way that includes finite-dimensional spaces, which automatically satisfy the condition of completeness.

3.1.2 Qubit

3.1.2.1 The Concept of Qubit

The bit-level is where processing takes place in a traditional computer. Quantum physics, in the context of quantum computers, is the specific behavior that controls the system. The interaction between various atoms is described using a number of methods from the field of quantum physics. These atoms are referred to as ”qubits” when speaking of quantum computers. A qubit functions as both a wave and a particle. Compared to a particle (or bit), a wave dispersion may store a lot of information .

A bit is a binary digit, and it is the fundamental unit of classical computing. Quantum Binary Digit is what is meant by the word ”qubit.” Qubits can exist in numerous states simultaneously, unlike bits, which can only exist in two states 0 and 1. There is a value range of 0 to 1.

Take a look at the coin toss as an example to help you better understand this idea. One side of a coin is Heads (1) and the other is Tails (0). We don’t know which side the coin is on when it’s being tossed until we stop it or it hits the ground. Can you identify its side? Depending on your angle, it displays either 0 or 1; only when you pause it to take a closer look does it display just one side. Similar circumstances apply to qubits. The superposition of two states is what is meant by this. Accordingly, the likelihood of measuring a 0 or 1 is typically neither 0.0 nor 1.0. In other words, there’s a chance the qubit could be in several states at once. When we learn the outcome of the coin flip (heads or tails), the superposition collapses

3.1.2.2 Qubit with Mathematical Notations

It is a two-dimensional quantum system in mathematics with the orthonormal basis states $|0\rangle$ and $|1\rangle$, also referred to as the computational basis [25]. Any superposition of the base states, which can be described as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{3.1}$$

can be the state $|\psi\rangle$ of a system.

where, α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. The final equality states that since a qubit's states are simply vectors in a complex Hilbert space, linear algebra's whole framework may be applied to manipulate them. The ground state $|g\rangle$ and an excited state $|e\rangle$ of a matter system, or the horizontally polarized state $|H\rangle$ and vertically polarized state $|V\rangle$ of a single-photon system, could both be examples of orthonormal basis states.

3.1.3 Quantum Register

In classical computer the register is form by flip-flop and in quantum computer the register is merely a collection of qubits.

If any input state is entangled with any other state then 2^n bit of input state and overall $2^{(n+1)}$ data can be store initially. This need more memory space, but this space is required in instantinuous situation.

So an n size classical register refers to an array of n flip flops where as an n size of quantum register is merely a collection of n qubits.

In classical computer we worked with 0 or 1 at a time but in quantum computer for qubits we get more advantages as a quantum register of n qubits (register of size n or length n) is a state of the 2^n dimensional product space or Hilbert Space.

Suppose we want to store a number between 0 to 7 in a register for this we need 3 bits. A classical register would thus store one of the following configurations

$$0 = (000)$$

$$1 = (001)$$

$$2 = (010)$$

$$3 = (011)$$

$$4 = (100)$$

$$5 = (101)$$

$$6 = (110)$$

$$7 = (110)$$

A system of 3 bits can be occupied by the any following product states:

$$0 : |000\rangle$$
$$1 : |001\rangle$$
$$2 : |010\rangle$$
$$3 : |011\rangle$$
$$4 : |100\rangle$$
$$5 : |101\rangle$$
$$6 : |110\rangle$$
$$7 : |111\rangle$$

As we can generate superposition like:

$$|q\rangle = \sum_{x,y,z \in [0,1]} C_{xyz} |xyz\rangle \tag{3.2}$$

Therfore, we can conclude that the sate vector allows us to store the $2^3 = 8$ numbers in C_{xyz} at once and generally 2^N numbers with N qubits.This can be all done before measuremnts as after measurement the superposition states of qubits collapses.

3.1.4 Quantum Logic Gates

Qubits, as opposed to bits, are used by quantum computersas. A qubit can exist in a "superposition" of 0 and 1, in contrast to conventional bits, which can only be 0 or 1. Quantum computers are incredibly powerful due to their capacity to live in several states simultaneously.

But until you can perform a quantum calculation on a qubit, it is meaningless. Additionally, a set of fundamental operations known as quantum logic gates are used to accomplish these quantum computations. There are lots of types of quantum gates from single qubit to multiple qubit gates.

3.1.4.1 Single Qubit Gate

Single-qubit gates exist that enable both the creation of superposition states (Hedamart gate) and the flipping of a qubit from 0 to $1(PaulyXgate)$ and also rotate the bit along different axis$(PaulyY, Z)$. Details mentioned in the table 3.1.

3.1.4.2 Two Qubit Gates

These enable interaction between the qubits and can be used to produce quantum entanglement, which is a condition in which two or more qubits are correlated in a way that is

Operator	**Gate(s)**	**Matrix**	**Transformation**
Hadamard gate	$-\boxed{H}-$	$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$	Maps $Z \rightarrow X$, and $X \rightarrow Z$. This gate is required to make superpositions.
Pauli-X	$-\boxed{X}-$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	π rotation around the X axis, $Z \rightarrow -Z$. It basically like not gate of classical gate means it flip the bit from $0 \rightarrow 1$ or $1 \rightarrow 0$.
Pauly-Y	$-\boxed{Y}-$	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	π rotation around the Y axis, $Y \rightarrow -Y$.
Pauly-Z	$-\boxed{Z}-$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	π rotation around the Z axis, $X \rightarrow -X$. Also referred to as a phase-flip.
Phase (S,P)	$-\boxed{S}-$	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	maps $X \rightarrow Y$. This gate extends H to make complex superpositions. ($\pi/2$ rotation around Z axis).

Table 3.1: Single Qubit Gates with their matrices and transformation on a qubit.

not consistent with classical physics.

The controlled-NOT (CNOT) gate is an illustration of a 2qubit gate. It is a two-qubit operation, with the control qubit as the first qubit and the target qubit as the second. It will flip the target's qubit state from $|0\rangle$ to $|1\rangle$ or vice versa if the control qubit is $|1\rangle$.

Quantum algorithms with numerous qubits can be decomposed into a series of one-qubit and two-qubit quantum gates, constituting a universal quantum gate set, much like conventional computers.

Therefore, we can utilize these quantum gates to implement every algorithm conceivable on our quantum computer if we can make them operate reliably and scale to many qubits.

3.1.5 Quantum Entanglement

One of the most strange phenomena that can be observed when things are extremely small or inside the quantum realm is quantum entanglement. No matter how far apart two or more particles are in space, their states remain linked when they hook up in a specific way. That indicates that they are in a single, shared quantum state. Therefore, regardless of how far apart the particles are, observations of one of the particles can automatically reveal information about the other entangled particles. Furthermore, any change to one of these particles will inevitably have an effect on the others in the entangled system.

As they figured out the mechanics of the quantum realm in the early 20th century, physicists discovered the basic concepts of entanglement. They discovered that they needed to employ something known as a quantum state in order to adequately characterize subatomic processes.

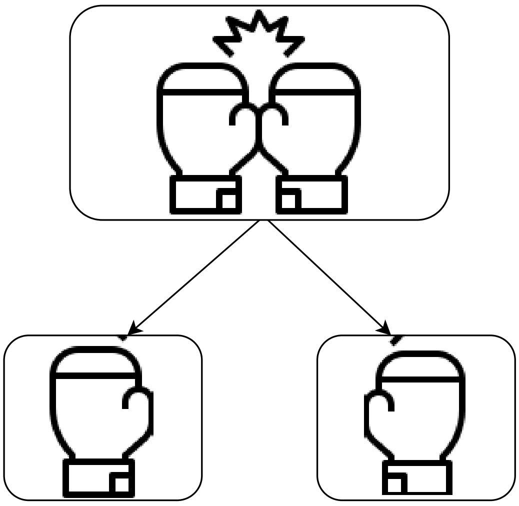

Figure 3.1: Engtangled Gloves

The concept of quantum entanglement asserts that two qubits are constantly superposed in two different states. Here is one instance. Imagine there is a package with a pair of gloves inside of it. One glove is pulled at random from the box. The package is then brought to another room. We can infer that the glove that is still in the box is left-handed because the glove that was taken out was discovered to be right-handed.

With qubits, the situation is the same. If one is spinning up, the other is put into a spin-down position by default. A situation in which both qubits are in the identical state is not possible. In other words, they are continuously intertwined. Quantum entanglement is what is happening here.

3.1.6 Quantum Superposition

Superposition is the capacity of a quantum system to exist in numerous states concurrently up until it is measured. The property of a quantum system whereby it can exist in multiple simultaneous quantum states. As an illustration, electrons have a quantum property called spin, which is a form of inherent angular momentum. The electron can exist in either of two distinct spin states—spin up or spin down—in the presence of a magnetic field. Each electron will have a finite possibility of existing in either state up until it is measured. It only seems to be in a particular spin state when it is being measured. In common experience a coin with its face up has a clear value: either it is a head or a tail. You are confident that the coin must be a head or a tail even if you don't look at it.

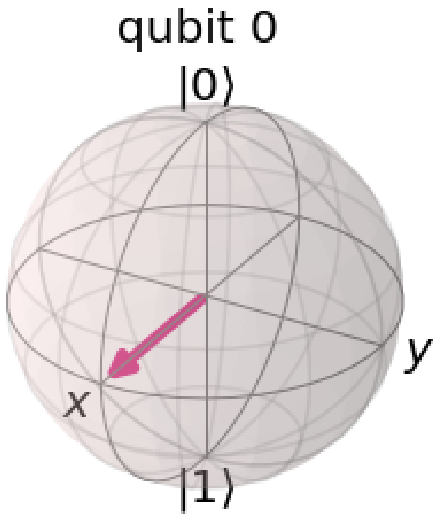

Figure 3.2: Example of a Superposition State

3.1.7 Measurement

A qubit may be measured out in the computational basis once a quantum computation task has been completed. As a result, one of the basic states $|0\rangle$ or $|1\rangle$ is reached for the qubit state. The value obtained could be kept in a conventional register and used for additional processing.

All qubits must initially be in the state $|0\rangle$ and can only be measured in the computational basis, according to the rules of quantum computing. Prior to the measuring operation, other states must be established by employing the requisite unitary operations, and measurement in other bases may be accomplished by using the appropriate gates. [33]

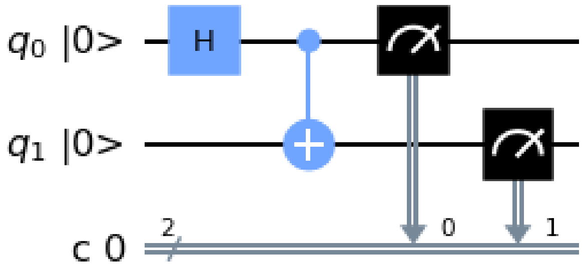

Figure 3.3: Example of Qubit Measurement

3.2 Fundamental of Quantum Machine Learning

3.2.1 Quantum Embeddings

Through the use of a quantum feature map, a quantum embedding encodes classical data as quantum states in a Hilbert space. A quantum state $|\psi_x\rangle$ is produced by translating a classical data point x into a collection of gate parameters in a quantum circuit. This procedure is essential for creating quantum algorithms and has an impact on their computing capacity.

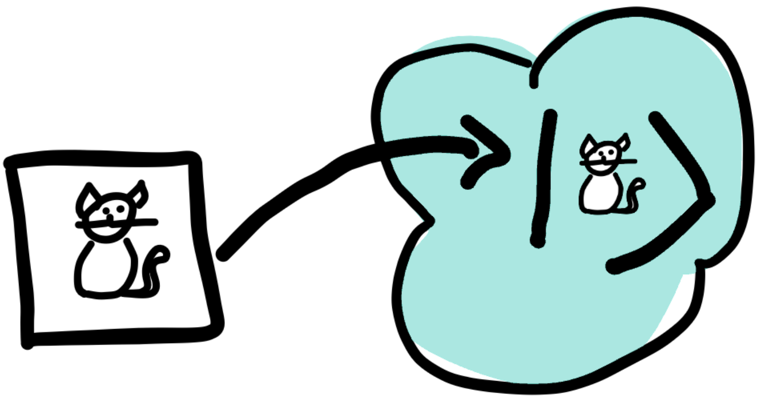

Figure 3.4: Example of encoding the Classical Data to Quantum Data

Let's look at a standard set of input data that consists of M examples and N features per example,

$$D = \{x^{(1)}, \ldots, x^{(m)}, \ldots, x^{(M)}\}, \tag{3.3}$$

where, for m=1,...,M, $x^{(m)}$ is an N-dimensional vector. We can utilize different embedding approaches, some of which are briefly discussed below, to embed this data into n quantum subsystems (n qubits or n qumodes for discrete- and continuous-variable quantum computing, respectively).

3.2.1.1 Several Types of Embeddings

- **Basis Embedding:** Each input is connected to a qubit system's computational basis state using basis embedding. Therefore, binary strings are the only way to represent classical data. The bit-wise translation of a binary string to the appropriate states of the quantum subsystems creates the embedded quantum state. For instance, the 4-qubit quantum state $|1001\rangle$ represents the value x=1001 . Consequently, one quantum subsystem can represent one bit of classical information.

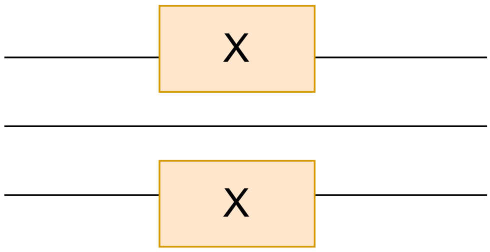

Figure 3.5: Basis Embedding Circuit

Think about the classical dataset in equation (3.3) that was mentioned earlier. Each example must be an N-bit binary string for basis embedding, where $x^{(m)} = (b1, ..., bN)$ and $b_i = \{0, 1\}$ for $i = 1, ..., N$. Each input example $x^{(m)}$ can be directly mapped to the quantum state $|x^{(m)}\rangle$ if all features are recorded with unit binary precision (one bit). In light of this, n, the number of quantum subsystems, must at least be equal to N. In superpositions of computational basis states, a complete dataset can be represented as

$$|D\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} |x^{(m)}\rangle$$

Let's say, for example, that we have a classical dataset with two samples, $x^{(1)} = 01$ and $x^{(2)} = 11$. Two qubits are used in the corresponding basis encoding to represent the values $|x^{(1)}\rangle = |01\rangle$ and $|x^{(2)}\rangle = |11\rangle$, resulting in

$$|D\rangle = \frac{1}{\sqrt{2}} |01\rangle + \frac{1}{\sqrt{2}} |11\rangle$$

- **Amplitude Embedding:** Data is encoded into a quantum state's amplitudes using the amplitude-embedding technique. The amplitudes of an n-qubit quantum state $|\psi_x\rangle$ are used to represent a normalized classical N-dimensional datapoint x as,

$$|\psi_x\rangle = \sum_{i=1}^{N} x_i |i\rangle$$

where $|i\rangle$ is the i-th computational basis state, x_i is the i-th element of x, and $N = 2^n$. However, in this situation, x_i may contain several numeric data types, such as integer or floating point.

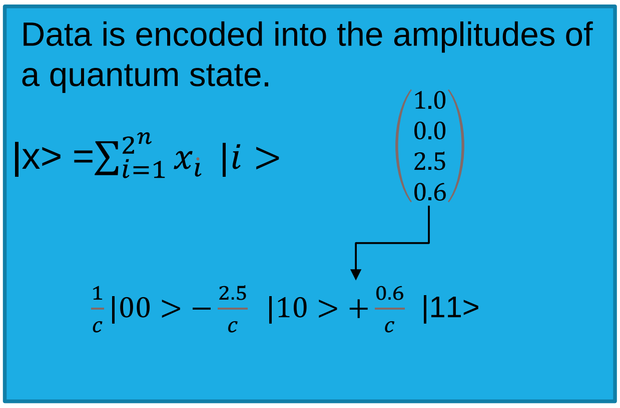

Figure 3.6: Amplitude Embedding

- **Angle Embedding:** Encodes N features into the rotation angles of n qubits, where $N \leq n$.

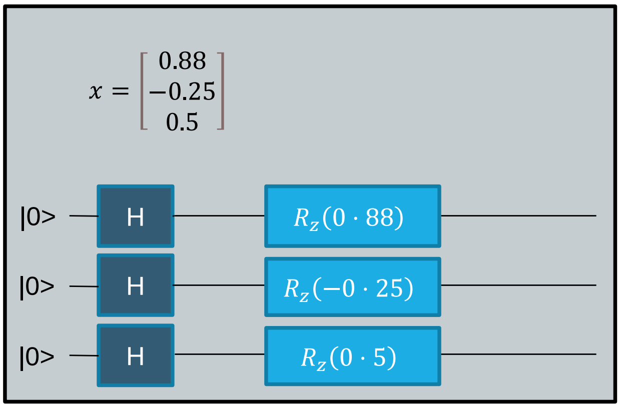

Figure 3.7: Angle Embedding

According to the rotation parameter, the rotations can be either RX, RY, or RZ gates:

1. rotation='X' uses the features as angles of RX rotations.
2. rotation='Y' uses the features as angles of RY rotations.
3. rotation='Z' uses the features as angles of RZ rotations.

Features must have lengths that are less than or equal to the amount of qubits. The circuit does not apply the remaining rotation gates if there are fewer entries in features than rotations.

- **Displacement Embedding:** Encodes N features into the displacement amplitudes r or phases ϕ of M modes, where $N \leq M$

  The mathematical definition of the displacement gate is given by the operator.

$$D(\alpha) = \exp\left(r\left(e^{i\phi}\hat{a}^{\dagger} - e^{-i\phi}\hat{a}\right)\right)$$

where $\hat{a}$ and $\hat{a}^{\dagger}$ are the bosonic creation and annihilation operators

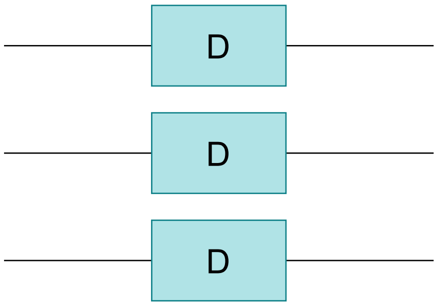

Figure 3.8: Displacement Embedding Circuit

3.2.2 Quantum Feature Map

To make it easier to work with or because the new space might have some useful qualities, many traditional machine learning techniques re-express their input data in a separate space. Support vector machines, which categorize data using a linear hyperplane, are a typical example. When the data are already linearly separable in the original space, a linear hyperplane is effective; however, this is unlikely to be the case for many data sets. By using a feature map to translate the data into a new space where it is linear, it might be able to get past this problem.

Let χ represent a set of input data more formally. A function that has the form $\phi : \chi \rightarrow F$, where F is the feature space, is called a feature map (ϕ). $\phi(x)$ for all $x \in X$,

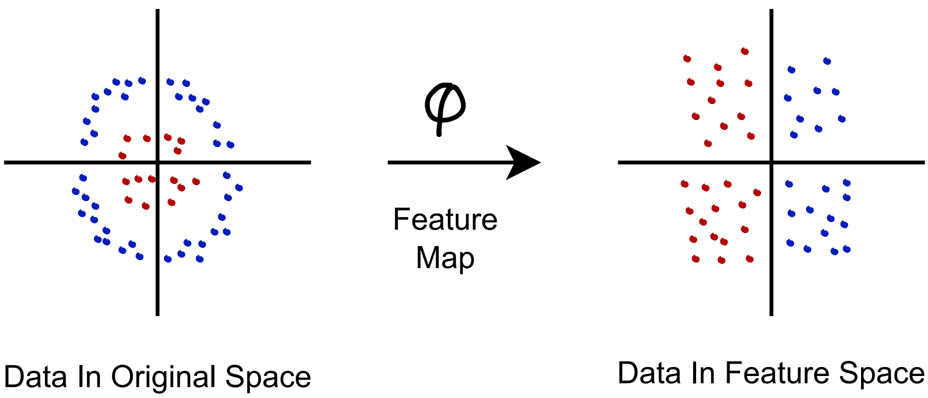

Figure 3.9: Data can be transformed into an easier-to-process format using a feature map.

the outputs of the map on the individual data points, are referred to as feature vectors.

A quantum feature map, F is typically merely a vector space. $\phi : \chi \rightarrow F$ is a feature map where the feature vectors are quantum states and the vector space F is a Hilbert space. The unitary transformation $U_\phi(x)$, which is often a variational circuit whose parameters depend on the input data, is used by the map to convert $x \rightarrow |\phi(x)\rangle$ by means of its parameters.

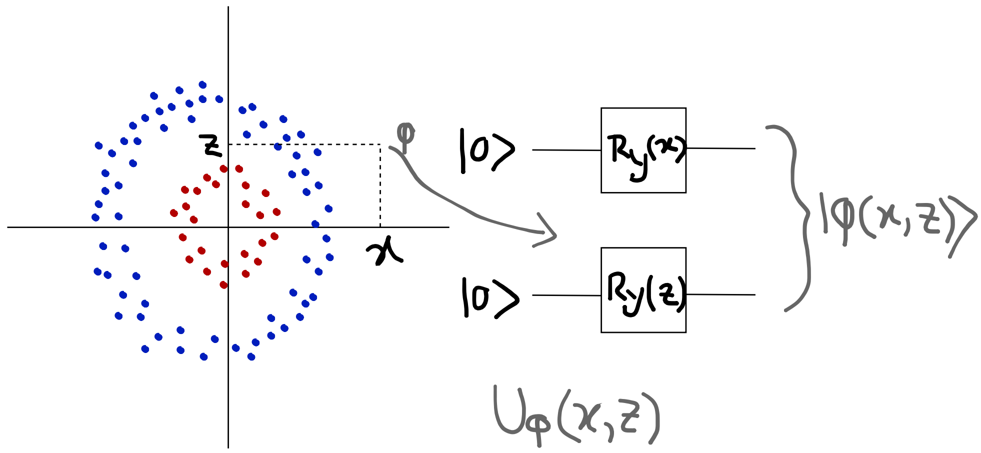

Figure 3.10: Data Transformation Using Unitary Operation.

3.2.3 Quantum Gradient

The expected value of a measurement observable, which may be formally expressed as a parameterized "quantum function" $f(\theta)$ with tunable parameters $\theta = \theta_1, \theta_2, ..., \theta_n$, is the output of a variational circuit. Partially derivatives of f with respect to its parameters can be defined, just like with any other similar function. [34]

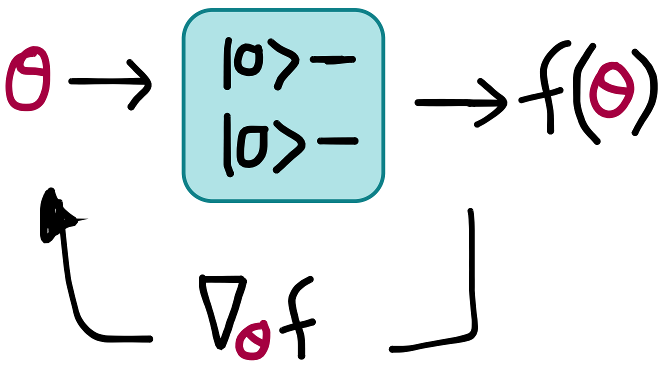

Figure 3.11: The Flow Of Quantum Gradients

The vector of partial derivatives of a quantum function $f(\theta)$ is defined as a quantum gradient.

$$\nabla_\theta f(\theta) = \begin{pmatrix} \partial_{\theta_1} f \\ \partial_{\theta_2} f \\ \vdots \\ \partial_{\theta_n} f \end{pmatrix}$$

Usually, for instance if many qubits are measured, quantum nodes are characterized by a number of expectation values. The output in this situation is characterized by the vector-valued function $\vec{f}(\theta) = (f_1(\theta), f_1(\theta), ...)^T$, and the quantum gradient turns into a "quantum Jacobian":

$$J_\theta f(\theta) = \begin{pmatrix} \partial_{\theta_1} f_1 & \partial_{\theta_1} f_2 & \cdots \\ \partial_{\theta_2} f_1 & \partial_{\theta_2} f_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

It turns out that using parameter-shift principles, the gradient of a quantum function $f(\theta)$ may frequently be represented as a linear combination of other quantum functions. This implies that quantum computers are capable of computing quantum gradients, enabling gradient-based optimization techniques like gradient descent, which are commonly used in machine learning, to be used to quantum computing.

3.2.4 Parameter-Shift Rules

A variational circuit's output (i.e., the expectation of an observable) can be expressed as a "quantum function" denoted by the parameters $f(\theta)$ and parametrized by $\theta = \theta_1, \theta_2, ...$ In many instances, the partial derivative of $f(\theta)$ can be written as a linear combination of various quantum functions. It's significant that these other quantum functions frequently employ the same circuit, with the sole difference being a shift in the argument. This indicates that the same variational circuit architecture can be used to compute partial derivatives of a variational circuit.

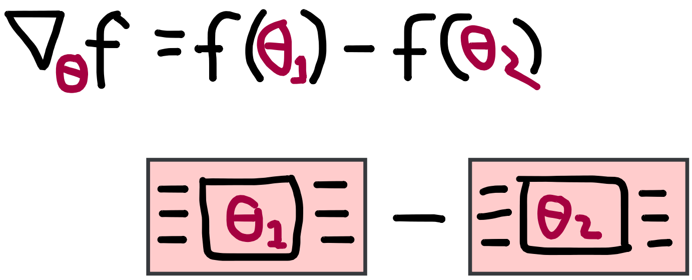

Figure 3.12: Example of Parameter-Shift Rules

Parameter-shift rules are instructions on how to obtain partial derivatives by evaluating parameter-shifted examples of a variational circuit. They were first introduced to quantum machine learning in [35] and expanded in [36].

3.2.5 Circuit Ansatz

An ansatz in the context of variational circuits typically refers to a subroutine made up of a series of gates applied to particular wires. Similar to the design of a neural network, this just specifies the basic structure; the variational technique can be used to optimize the types of gates and/or their free parameters.

The quantum computing community has put up a number of variational circuit theories. The strength of an ansatz depends on the targeted use-case, and what constitutes a good ansatz is not always obvious.

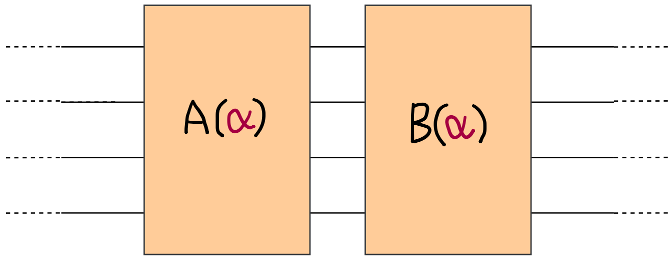

Figure 3.13: A Simple Circuit Ansatz

3.2.6 Hybrid Computation

The method of combining classical and quantum computations is referred to as a hybrid in the domain of quantum computing. This is the fundamental idea of optimizing variational circuits, which involves using a classical co-processor to support a quantum algorithm.

To establish how ”excellent” the quantum circuits are, averages of measurement outcomes are typically estimated by quantum devices. This is done by combining these averages into a single classical cost function.

3.2.7 Quantum Differentiable Programming

In quantum computing, it is possible to automatically calculate variational circuits’ derivatives with regard to their input parameters. This is used by a paradigm called quantum differentiable programming to create quantum algorithms differentiable and thus trainable.

The derivatives of functions with respect to program inputs are calculated via automatic differentiation in classical differentiable programming. Although differentiable programming and deep learning are conceptually similar, they differ in some ways. The structure of models like neural networks, including the number of nodes and hidden layers, is often statically set in early deep learning frameworks. This is referred to as a ”define-and-run” strategy. Although the network is trainable via automatic differentiation and backpropagation, the program’s basic structure remains the same.

In contrast, more modern methods employ a ”define-by-run” strategy where the model’s structure isn’t predicated in any way. The models’ structure, which is dynamic and made up of parameterized function blocks, can alter depending on the input data while still

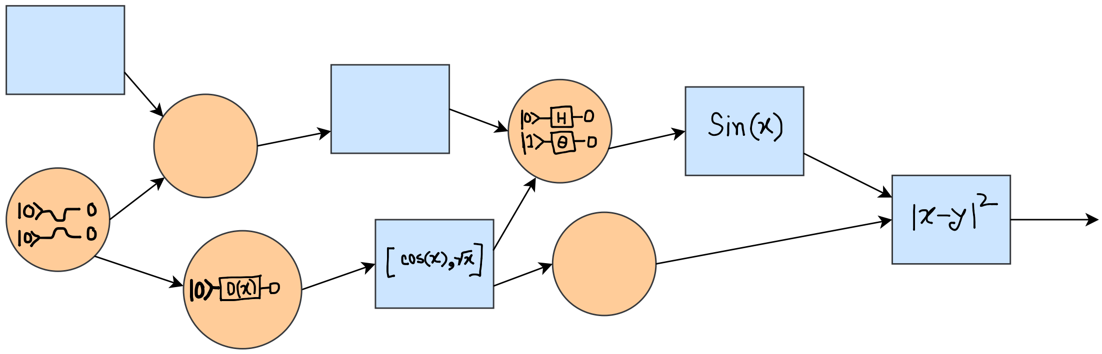

Figure 3.14: In quantum differentiable programming, hybrid computations made up of both classical (blue) and quantum (pink) components are inherently differentiable.

being trainable and differentiable. This is true even in the presence of traditional control flow structures like for loops and if statements, which is an important part of it. With the help of these concepts, we may use diffentiable programming to create end-to-end differentiable programs for hybrid classical-quantum computing.

4 Methodology

In this chapter, we will discuss about our research methodology.

4.1 Methond with System Diagram

The quantum circuit is differentiable, as was already mentioned it in section 3.2.7, and it can function as a deep learning model. Nearly all datasets in the world are classical datasets, so in order to train a quantum machine learning model, we must transform the dataset into a format that quantum computers can understand. Quantum Embedding is the process of converting traditional data into quantum data. There are several ways to convert classical data into quantum data, as explained in the section on quantum embeddings. According to the methodology of quantum machine learning as shown in figure basic architecture of QML, we must first and foremost encode the classical data in quantum readable data before using some unitary operations that will function as deep learning neurons. We then perform measurements to obtain the prediction, calculate the loss using the classical loss function, and then backpropagate through the circuit. The quantum machine learning algorithm functions in this manner overall.

So in the quantum neural networks, the goal is to translate the classical mathematical formula $L(x) = \phi(Wx + b)$ into a quantum state $L(|x\rangle) = |\phi(Wx + b)\rangle$. The key elements for transforming classical neural networks into quantum circuits are:

- data encoding: $x \to |\psi(x)\rangle$
- affine transformation $W|\psi(x)\rangle + |b\rangle$
- non-linear activation function $\phi(|A\rangle)$

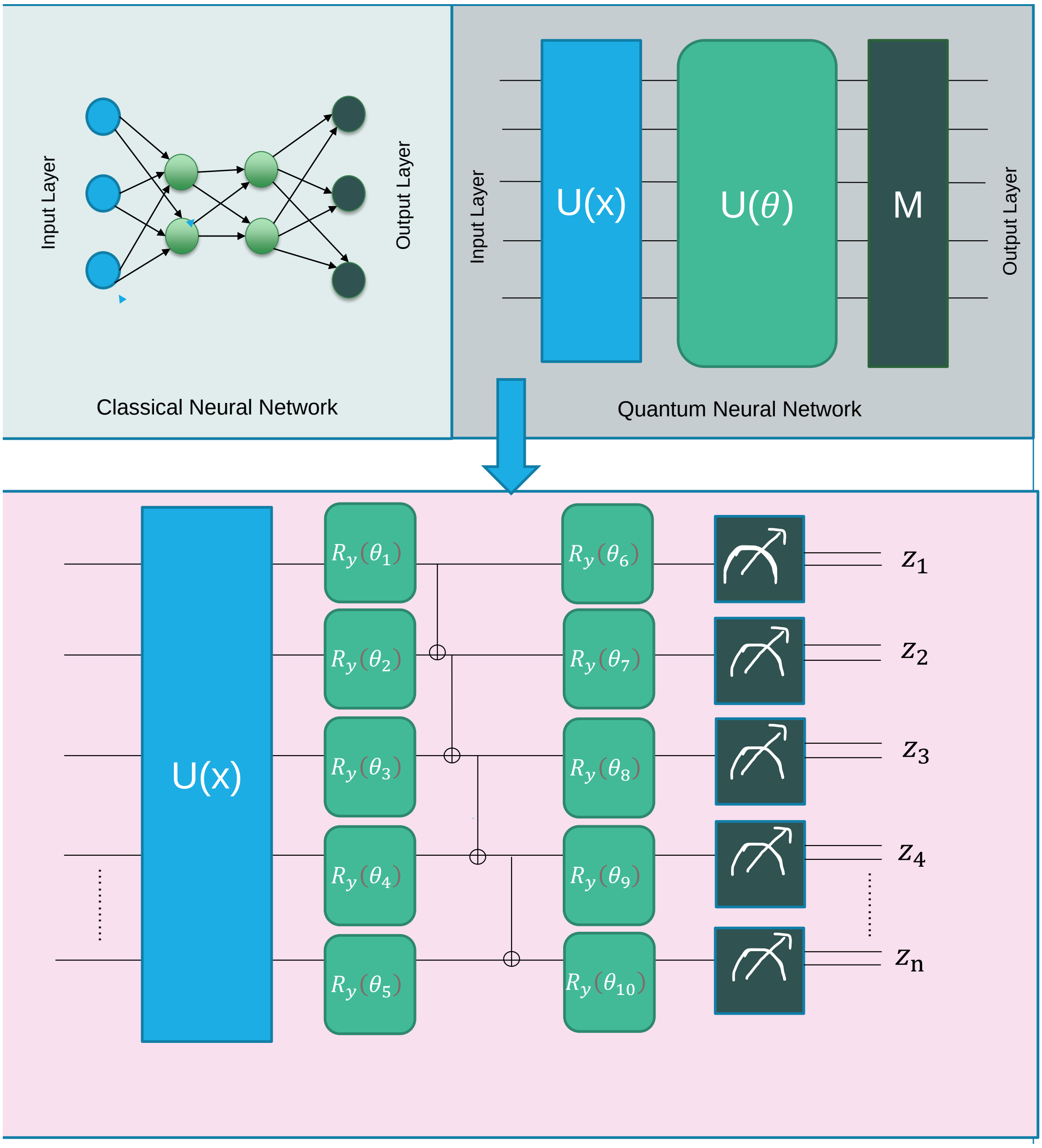

Figure 4.1: Extended Building block of Quantum Machine Learning in Contrast with Classical Machine Learning.

All unitary gates that are possible in the qubit model are linear. As a result, the model does not directly translate the bias addition component and the non-linear activation function component of classical neural networks into quantum. The displacement gate and the Kerr gate in the CV model, however, permit a direct conversion from classical to quantum.

That's why we have chosen two architectures of QNN one from the qubit model (Quanvolutional Neural Network) and another from qumode model (Continuous variable Quantum Neural Network). Where our primary objective is to develop a quantum neural network model and compare its result to a classical machine learning model. For two different quantum machine learning models (1. QNN; 2. CVQNN) of two different quantum computing architectures, we primarily compare three things.

1. Either classical or quantum model reach the optimum accuracy state faster.
2. Either the classical or quantum model reaches the optimum loss faster.
3. And lastly their test accuracry.

Our research models will be conducted as visual on the figure 4.1 to build an effective model for the classification task of multiple classes of MNIST handwritten digit classification.

4.2 Model 01: Quanvolutional Neural Networks(QNN)

We use the Quanvolutional Neural Network, a quantum machine learning model first presented by Henderson[37]. A common model in classical machine learning that is very effective at processing images is the convolutional neural network (CNN). The model is built around the concept of a convolution layer, where a local convolution is used to process a portion of the input data rather than processing the entire set with a global function.

Small local regions are processed consecutively with the same kernel when the input is an image. The findings for each region are typically connected to various output channels of a single pixel. A new object that resembles an image is created by adding together all of the output pixels and can then be treated by other layers.

The same concept can be expanded to apply to quantum variational circuits. The subsequent process, which is quite similar to the one utilized in [37], provides a potential strategy. The figure additionally depicts the scheme.

- In our case, a 2×2 square from the input image is implanted into a quantum circuit. This is demonstrated here by applying parametrized rotations to qubits that were started in the ground state.
- The system performs a quantum computation linked to a unitary U. A variational quantum circuit or, more simply, a random circuit, as suggested in [37], could produce the unitary.
- After measuring the quantum system, a list of classical expectation values is obtained. The measurement results might also be post-processed traditionally, as suggested in [37].
- Each expectation value is assigned to a distinct channel of a single output pixel, much like a classical convolution layer.
- By repeating the same process over several regions of the image, we may scan the entire input image and create an output object that is organized as a multi-channel image.

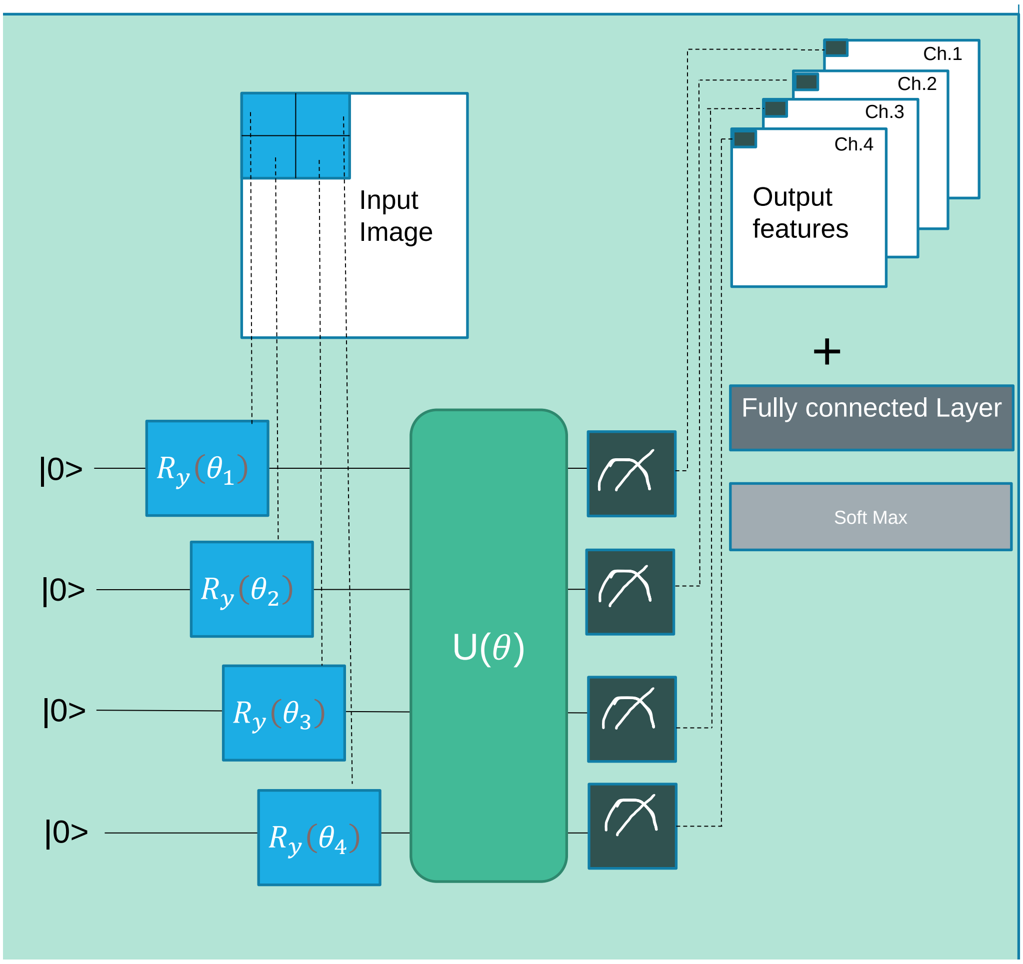

Figure 4.2: Model 01: Quanvolutional Neural Network

- Further quantum layers or classical layers may come after the quantum convolution.

A quantum circuit can produce extremely complex kernels whose computation could, at least in theory, be classically intractable. This is the major difference between a quantum circuit and a classical convolution.

4.2.1 Data Encoding

The 28×28 matrix used in the MNIST dataset is in its classical states. We must use a quantum data encoding approach to convert this classical data from classical to quantum states. To convert the classical data into quantum data in this case, we employed the angle embedding technique (discussed in section 3.2.1). An embedding layer of R_y rotations with angles scaled by a factor off π makes up the quantum circuit of our model 1 (QNN) in fig 4.2.

4.2.2 Quantum Circuit

By following the architecture of Quanvolutional Neural Network we have implemented a random circuit from the PennyLane framework on 4qubits. Hence it is a 4 qubit system simulated on PennyLane $default.qubit$ simulation device.

4.2.3 Measurement

For qubit based classification, PennyLane provides several measurement techniques including expected value method which has been used for our first model.

We have got 4 expected value from for qubits that are then mapped into 4 diferent channels of a single ouptput pixel. By repeating the process a multiple of times we have mapped a full 4 output channel object just like CNN extract the important features from image into several channels. This process halves the resolution of the input image. In the standard language of CNN, this would correspond to a convolution with a 2×2 kernel and a stride equal to 2.

Inspecting how a batch of samples respond to the quantum convolution layer in fig 4.3. we can finalize that the resolution has been downscaled, and the quantum kernel has also caused some localized distortion. On the other hand, as is conventional for a convolution layer, the image's overall shape remains kept.

4.2.4 Classical Layer

The characteristics that come from applying the quantum convolution layer are then fed into a classical neural network, which will be trained to classify the 10 distinct digits in the MNIST dataset.

We only utilize a single fully connected layer with ten output nodes and a softmax activation function as our model.

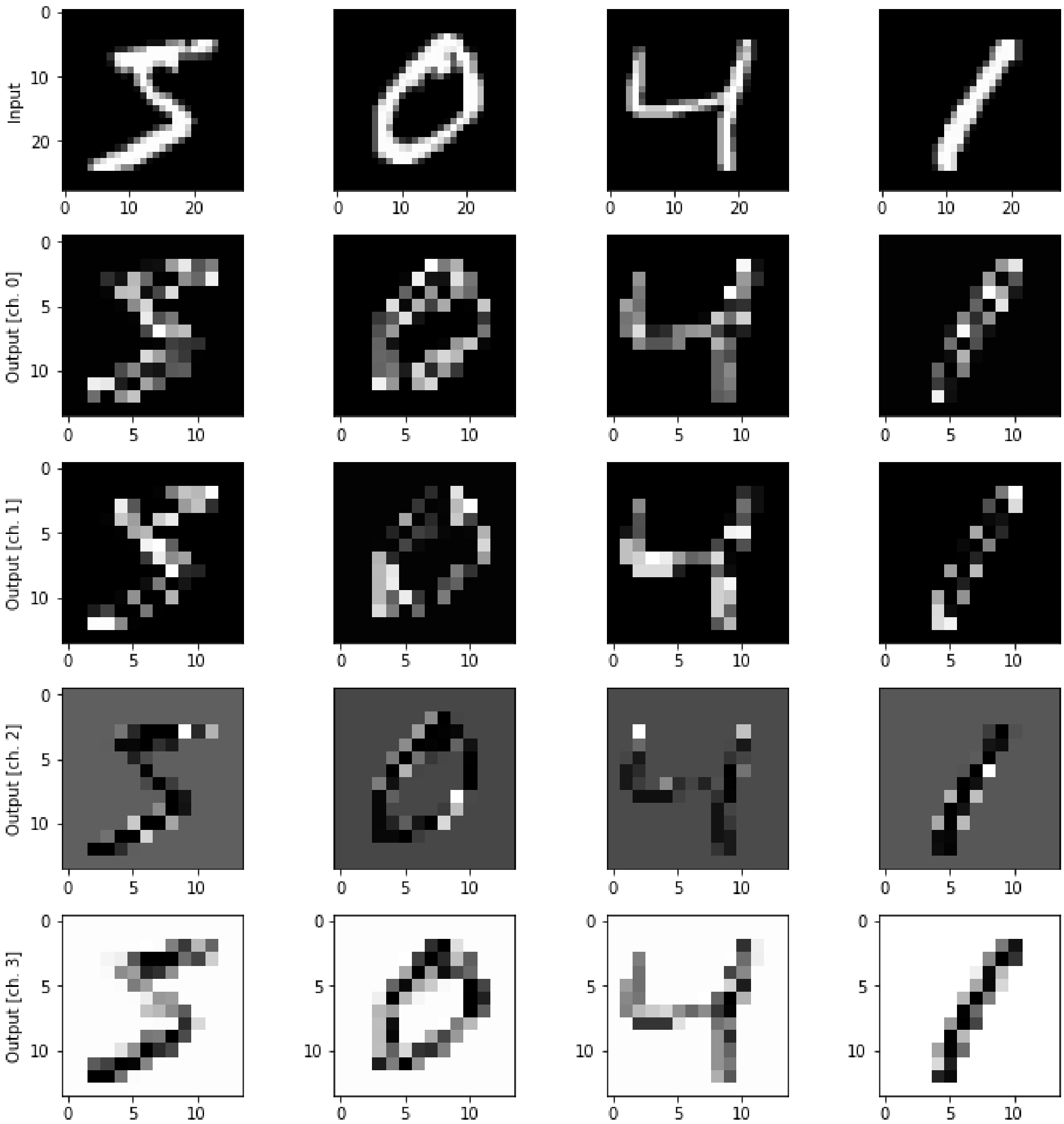

Figure 4.3: Grayscale images of the four quantum convolution output channels.

A cross-entropy loss function and a stochastic gradient descent optimizer are used in the model's construction.

4.3 Model 02: Continuous variable Quantum Neural Network(CVQNN)

The CV binary classifier design(14) serves as an inspiration for the multi-classifier model provided in this section. The CVQNN moel runs on Xanadu's $X8$ photonic quantum simulator which is made up of $8 - qumodes$. The model's measurement outputs are interpreted as one-hot encoded predictions of the labels for the images.

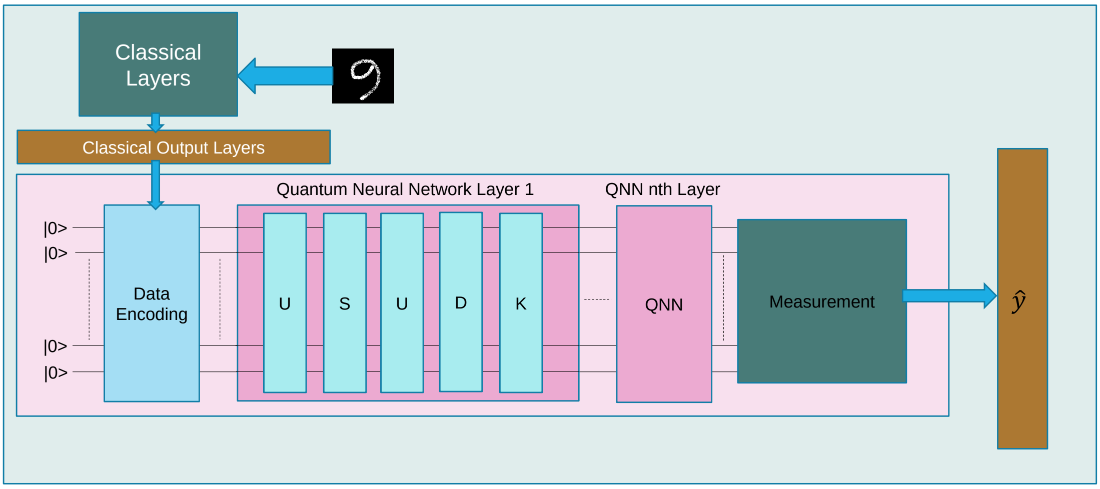

Figure 4.4: Model 02: Hybrid Continuous Variable Quantum Neural Network

In order to convert the original image matrices of size $28 \times 28 = 784$ into vectors of shorter length that the data encoding circuit can handle, a classical neural network is utilized as a pre-processing step. Squeezers, an interferometer, displacement gates, and Kerr gates are employed in the data encoding circuit. The quantum circuit that implements $L(|x\rangle) = |\phi(Wx + b)\rangle$ as in the binary classifier(14) is employed for the quantum neural network.

The foundation of the $10 - class$ classifier is two qumodes. When an image matrix's label $k\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ is transformed into a one-hot encoded vector, the result is a vector of length 10 with all zeros but the k^{th} entry as 1. The cutoff dimensions are chosen so that the length of the output vectors is more than 10. For this model, we added some padding zeros to the one-hot encoded labels to make them fit the circuit's output size. The table 4.1 below provides a description of the cutoff dimension utilized for the model CVQNN.

The architecture in Figure 6 depicts the data flow of $image\ matrix \rightarrow classical\ layers \rightarrow reduced\ output\ vectors \rightarrow data\ encoding \rightarrow QNN \rightarrow measurement \rightarrow output\ vectors\ as\ one-hot\ encoded\ labels$.

number of qumodes	cutoff dimension	output size	number o padding 0's
2	4	$4^2 = 16$	6

Table 4.1: Calculation of cutoff dimension

Interferometers, a set of m squeezers, a set of m displacement gates, and a set of m Kerr gates, respectively, are represented by the boxes $U, S, D, andK$, where m is the number of qumodes.

4.3.1 Classical layers

The size of the data images is reduced from $28 \times 28 = 784$ to smaller size vectors that fit the amount of parameters available for data encoding using traditional feed forward neural networks. The image matrices are first reduced to vectors of smaller sizes using Keras dense layer operations and the activation function "ELU," flattening them to vectors of size $28 \times 28 = 784$. The data encoding quantum circuit subsequently transforms the output vectors into quantum states.

4.3.2 Data encoding

The classical neural network's output vectors are in classical states. Classical states are transformed into quantum states by the quantum data encoding circuit. Squeezers, interferometers, displacement gates, and Kerr gates are examples of quantum gates that are used for data encoding. These parameterized quantum gates use the entries of a classical vector as their parameters. Beamsplitters on pairs of nearby qumodes, along with rotation matrices, make up an interferometer. When viewing the parameters $z, \alpha \in \mathbb{C}$ in the Euler formula $z = a + bi = r(cos\phi + isin\phi) = re^{i\phi}$, squeezers $S(z)$ and displacement gates $D(\alpha)$ can be viewed as either one parameter gates or two parameter gates. We employ them as two-parameter gates for data encoding.

These gates can handle $(8m - 2)$ parameters, where m is the number of qumodes, in $m - qumode$ circuits. These values calculated from table 4.2 served as the basis for determining the size of the classical network output vectors.

Squeezers	Base Shift	Rotation	Displacement	Kerr
$2m$	$2(m-1)$	m	$2m$	m

Table 4.2: Calculation of Squeezers, Base Shift, Rotation, Displacement, and Kerr gate

4.3.3 Quantum circuit

The QNN circuit implements a quantum equivalent of the classical neural network $L(x) = (Wx + b)$ as

$$L(|x\rangle) = |\phi(Wx + b)\rangle = \phi \circ D \circ U_2 \circ S \circ U_i |x\rangle \tag{4.1}$$

where $|x\rangle$ is a quantum data encoded state of the original data vector x, U_k interferometers, S squeezers, D displacement gates, and $\phi(\cdot)$ Kerr gates, respectively. Calculations for the number of parameters for a set of gates on m-qumodes are shown in the table 4.3 .

U_k	S	Dt	$\phi(\cdot)$	Total
$2(m-1)+m$	m	m	m	$2\{(2m-1)m\}+m+m+m=9m-4$

Table 4.3: Equation to calculate total Parameter

On our 2-qumode CVQNN multiclass classifier, 4 layers of QNN were implemented. The table 4.4 lists how many parameters there are for 2-qumodes.

Number of Qumodes	Parameter per Layer	Total Parameter
2	$9 \times 2 - 4 = 14$	$4 \times 14 = 56$

Table 4.4: Calculation of Parameters

4.3.4 Measurement

For CV-based calculations, Pennylane provides three different measurement techniques: expected value, variance, and probability. For each qumode, the expectation value and variance procedures produce a single-valued result. Vectors of size n^m are produced by the probability approach, where n is the cutoff dimension and m is the number of qumodes.

For the measurement we have use the probability method for our quantum network

4.3.5 Parameter update

The quantum circuit is transformed into a Keras layer using the PennyLane Tensorflow plug-in capabilities, and it is then added to the classical layers. Then, parameter updating can be done using the built-in loss and optimizer methods of Keras. Categorical Cross entropy and Stochastic Gradient Descent are employed as the optimizer and loss function, respectively, for our $2-qumode$ CVQNN multiclass classifier model. The quantum gates' parameters are subsequently set using the revised parameters in the following training iteration.

5
Result Analysis

We primarily compare three things of classical and quantum machine learning model for both of our model.

1. Either classical or quantum model reach the optimum accuracy state faster.

2. Either the classical or quantum model reaches the optimum loss faster.

3. And lastly their test accuracry.

5.1 Result Analysis of Model 01

As far as our first analysis of model 1(QNN) from the evaluation graph of figure [5.1] we can visualize that the classical machine learning model reach out to the optimum state of accuracy matrix before than the Quantum Machine Learnin model.

In comparison to the classical model, the hybrid quanvolutional neural network model for the second analysis achieves the ideal state of loss reduction more quickly can be enriched from figure [5.1].

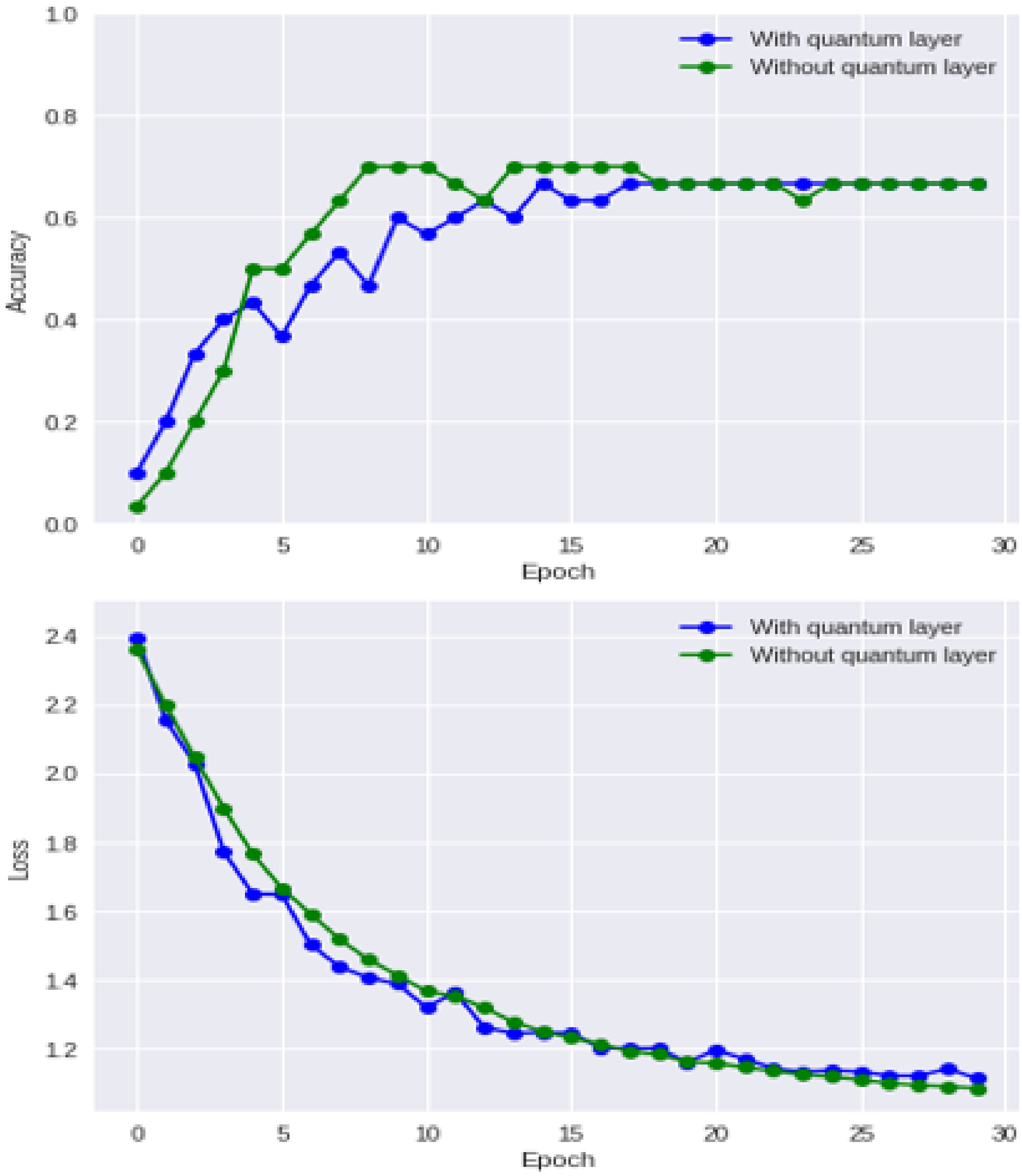

Figure 5.1: Evaluation Loss and Accuracy of QNN Model in contrast with a Classical Neural Network Model..

model	Test Accuracy	Train Accuracy
Classical Neural network	0.96	0.99
Quanvolutional Neural network(QNN)	0.92	0.98

Table 5.1: Test Accuracy of Classial And Quantum Model 1.

5.2 Result Analysis of Model 02

According to our first study of Model 2, we can see from figure [**??**]that the classical machine learning model reached the optimal accuracy matrix state before the quantum machine learning model from the evaluation graph.

For the second study,as per figure [5.3] the hybrid continuous variable quantum neural network takes longer to get to the ideal loss-reduce state than the conventional neural network model.

As the we are comparing the first generation quantum machine learning models to the optimized classical machine learning models it is still quit close to the classical machine learning and also quantum models takes longer time than classical model as it runs on the simulator rather than the actual quantum computer, so when we will be able to do these thing on real quantum computer it will be faster than classical one.

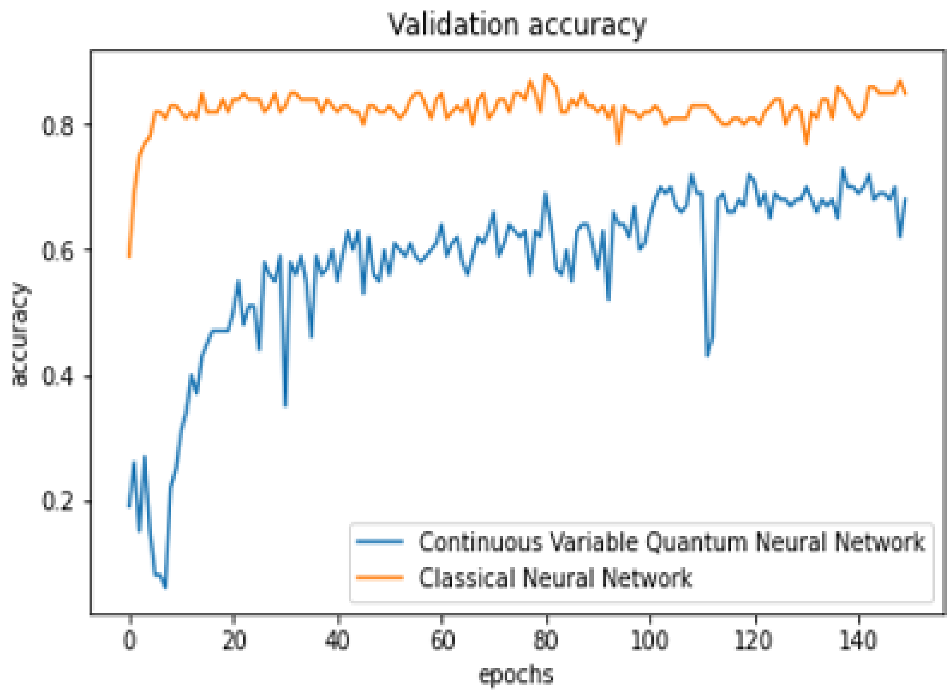

Figure 5.2: Validation Accuracy of CVQNN Model along with a Classical Neural Network.

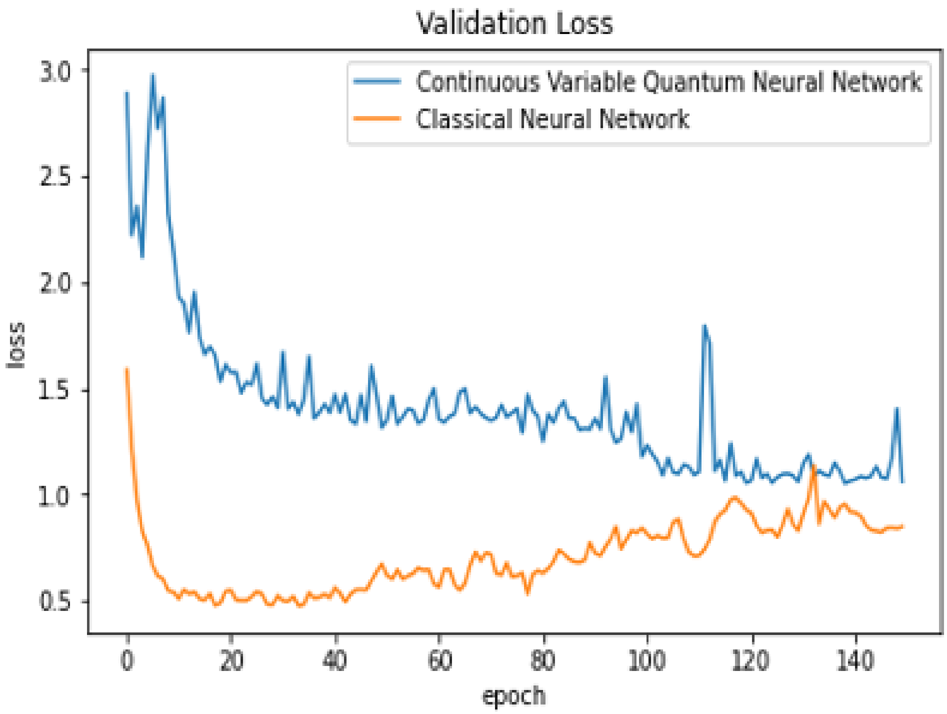

Figure 5.3: Validation Loss of CVQNN Model along with a Classical Neural Network

model	Test Accuracy	Train Accuracy
Classical Neural network	0.88	0.99
Continuous Variable Quantum Neural network(CVQNN)	0.72	0.98

Table 5.2: Test Accuracy of Classical And Quantum Model 2

6

Conclusion

The world is moving towards the making Quantum Computer to ease the complex computation that take years in classical computer. We need to introduce the classical ML algorithm in quantum and that is what we wanted and have done in this project. For data processing, classification, and the clustering area, now a days, Machine Learning has proved to be very effective. Now, it became superior because of its process in many areas. After that, a new field has emerged named "Quantum Machine Learning" that is the combination of Quantum information Processing and classical machine learning. As most of our data is classical and converting all classical data to quantum data will be very costly, there need a bridge between quantum and classical computer and this hybrid model is exactly what we need.

So, here we introduce a comprehensive review of state-of-the-art advances in Quantum Machine Learning. Also, we outline some recent works on different models of the quantum machine learning, and interprets classification tasks in the realm of Quantum with several encoding methods.

Appendix

A block of sample **python** codes are given below.

```python
# Initialize a Quantum circuit in PannyLane
dev = qml.device("default.qubit", wires=4)
# Random circuit parameters
rand_params = np.random.uniform(high=2 * np.pi, size=(n_layers, 4))

@qml.qnode(dev)
def circuit(phi):
    # Encoding of 4 classical input values
    for j in range(4):
        qml.RY(np.pi * phi[j], wires=j)

    # Random quantum circuit
    RandomLayers(rand_params, wires=list(range(4)))

    # Measurement producing 4 classical output values
    return [qml.expval(qml.PauliZ(j)) for j in range(4)]
```

```python
# Data encoding program for CVQNN (Model 02) model
def data_encoding(x):
    qml.Squeezing(x[3], x[4], wires=0)
    qml.Squeezing(x[9], x[10], wires=1)

    qml.Beamsplitter(x[5], x[6], wires=[0,1])

    qml.Rotation(x[7], wires=0)
    qml.Rotation(x[8], wires=1)

    qml.Displacement(x[1], x[2], wires=0)
    qml.Displacement(x[11], x[12], wires=1)

    qml.Kerr(x[0], wires=0)
    qml.Kerr(x[13], wires=1)
```

```
#  CVQNN (Model 02) model circuit
def qnn_layer(v):
    # Interferometer 1
    qml.Beamsplitter(v[0], v[1], wires=[0,1])
    qml.Rotation(v[2], wires=0)
    qml.Rotation(v[3], wires=1)

    # Squeezers
    qml.Squeezing(v[4], 0.0, wires=0)
    qml.Squeezing(v[5], 0.0, wires=1)

    # Interferometer 2
    qml.Beamsplitter(v[6], v[7], wires=[0,1])
    qml.Rotation(v[8], wires=0)
    qml.Rotation(v[9], wires=1)

    # Bias addition
    qml.Displacement(v[10], 0.0, wires=0)
    qml.Displacement(v[11], 0.0, wires=1)

    # Non-linear activation function
    qml.Kerr(v[12], wires=0)
    qml.Kerr(v[13], wires=1)
```

Bibliography

[1] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997. [Online]. Available: https://doi.org/10.1137/S0097539795293172

[3] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya *et al.*, "Quantum chemistry in the age of quantum computing," *Chemical reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.

[4] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Química computacional cuántica," *Rev. Mod. Phys*, vol. 92, no. 015003, pp. 10–1103, 2020.

[5] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, "Quantum algorithms for quantum chemistry and quantum materials science," *Chemical Reviews*, vol. 120, no. 22, pp. 12 685–12 717, 2020.

[6] C. W. Bauer, B. Nachman, and M. Freytsis, "Simulating collider physics on quantum computers using effective field theories," *Physical review letters*, vol. 127, no. 21, p. 212001, 2021.

[7] B. Nachman, D. Provasoli, W. A. De Jong, and C. W. Bauer, "Quantum algorithm for high energy physics simulations," *Physical review letters*, vol. 126, no. 6, p. 062001, 2021.

[8] C. W. Bauer, B. Nachman, and M. Freytsis, "Simulating collider physics on quantum computers using effective field theories," *Physical review letters*, vol. 127, no. 21, p. 212001, 2021.

[9] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Aprendizaje automático cuántico," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[10] M. Schuld and F. Petruccione, "Quantum machine learning." 2017.

[11] Y. Chen, Y. Pan, G. Zhang, and S. Cheng, “Detecting quantum entanglement with unsupervised learning,” *Quantum Science and Technology*, vol. 7, no. 1, p. 015005, 2021.

[12] L. Domingo and F. Borondo, “Deep learning methods for the computation of vibrational wavefunctions,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 103, p. 105989, 2021.

[13] M.-Z. Ai, Y. Ding, Y. Ban, J. D. Martín-Guerrero, J. Casanova, J.-M. Cui, Y.-F. Huang, X. Chen, C.-F. Li, and G.-C. Guo, “Experimentally realizing efficient quantum control with reinforcement learning,” *arXiv preprint arXiv:2101.09020*, 2021.

[14] D. Alaminos, M. B. Salas, and M. A. Fernández-Gámez, “Quantum computing and deep learning methods for gdp growth forecasting,” *Computational Economics*, vol. 59, no. 2, pp. 803–829, 2022.

[15] S. Jain, J. Ziauddin, P. Leonchyk, S. Yenkanchi, and J. Geraci, “Quantum and classical machine learning for the classification of non-small-cell lung cancer patients,” *SN Applied Sciences*, vol. 2, no. 6, pp. 1–10, 2020.

[16] F. Zickert, “Hands-on quantum machine learning with python,” 2021.

[17] S.-S. Li, G.-L. Long, F.-S. Bai, S.-L. Feng, and H.-Z. Zheng, “Quantum computing,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 21, pp. 11 847–11 848, 2001.

[18] J. S. Bell, “On the einstein podolsky rosen paradox,” *Physics Physique Fizika*, vol. 1, no. 3, p. 195, 1964.

[19] C. Ding, T.-Y. Bao, and H.-L. Huang, “Quantum-inspired support vector machine,” *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[20] A. K. Bishwas, A. Mani, and V. Palade, “Big data classification with quantum multiclass svm and quantum one-against-all approach,” in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2016, pp. 875–880.

[21] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.

[22] G. Cavallaro, D. Willsch, M. Willsch, K. Michielsen, and M. Riedel, “Approaching remote sensing image classification with ensembles of support vector machines on the d-wave quantum annealer,” in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 1973–1976.

[23] M. Masum, M. Nazim, M. J. H. Faruk, H. Shahriar, M. Valero, M. A. H. Khan, G. Uddin, S. Barzanjeh, E. Saglamyurek, A. Rahman *et al.*, “Quantum machine

learning for software supply chain attacks: How far can we go?" *arXiv preprint arXiv:2204.02784*, 2022.

[24] C. Havenstein, D. Thomas, and S. Chandrasekaran, "Comparisons of performance between quantum and classical machine learning," *SMU Data Science Review*, vol. 1, no. 4, p. 11, 2018.

[25] ——, "Comparisons of performance between quantum and classical machine learning," *SMU Data Science Review*, vol. 1, no. 4, p. 11, 2018.

[26] S. Adhikary, S. Dangwal, and D. Bhowmik, "Supervised learning with a quantum classifier using multi-level systems," *Quantum Information Processing*, vol. 19, no. 3, pp. 1–12, 2020.

[27] J. Chen, L. Liu, Y. Liu, and X. Zeng, "A learning framework for n-bit quantized neural networks toward fpgas," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1067–1081, 2020.

[28] F. M. de Paula Neto, T. B. Ludermir, and W. R. de Oliveira, "Quantum neural networks learning algorithm based on a global search," in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2019, pp. 842–847.

[29] S. Boyapati, S. R. Swarna, and A. Kumar, "Quantum neural networks for dynamic route identification to avoid traffic," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2020, pp. 1018–1022.

[30] N. H. Nguyen, E. C. Behrman, M. A. Moustafa, and J. E. Steck, "Benchmarking neural networks for quantum computations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2522–2531, 2019.

[31] N. Thumwanit, C. Lortaraprasert, and R. Raymond, "Trainable discrete feature embeddings for quantum machine learning," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1352–1355.

[32] D. A. Zaidenberg, A. Sebastianelli, D. Spiller, B. Le Saux, and S. L. Ullo, "Advantages and bottlenecks of quantum machine learning for remote sensing," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 5680–5683.

[33] M. B. Plenio and S. S. Virmani, "An introduction to entanglement theory," in *Quantum information and coherence*. Springer, 2014, pp. 173–209.

[34] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.

[35] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, “Quantum circuit learning,” *Physical Review A*, vol. 98, no. 3, sep 2018. [Online]. Available: https://doi.org/10.1103%2Fphysreva.98.032309

[36] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, “Evaluating analytic gradients on quantum hardware,” *Physical Review A*, vol. 99, no. 3, mar 2019. [Online]. Available: https://doi.org/10.1103%2Fphysreva.99.032331

[37] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, “Quanvolutional neural networks: Powering image recognition with quantum circuits,” 2019. [Online]. Available: https://arxiv.org/abs/1904.04767

Index