The Future of AI Combined with the Powers of Quantum Computing

CS-664 Final project report

Aravind Badrinathan (U21804168)
Abhishek Gupta (U39625186)
Kavach Shah(U62075640)

Introduction

AI and Quantum computing are both developing domains that have shown a lot of promise in terms of changing the way we view computers. Most of us seem to look at them as independent entities moving towards their own paths to success. However, the combination of Artificial Intelligence (AI) and Quantum Computing has the potential to revolutionize many fields and create new opportunities. Quantum computing can help to improve the performance of AI algorithms, while AI can help to solve some of the key challenges in quantum computing. Together, they could unlock new capabilities and lead to breakthroughs in areas such as materials science, cryptography, and finance. The future of AI with quantum computing is an exciting and rapidly evolving field, with many possibilities yet to be explored. In this context, it is essential to understand the basics of both AI and quantum computing to appreciate the potential of this emerging technology.

A brief introduction to AI

Artificial Intelligence (AI) is a branch of computer science that aims to develop intelligent machines that can perform tasks that typically require human intelligence, such as perception, reasoning, learning, and decision-making. AI works by developing algorithms and models that can learn from data and make predictions or decisions based on that data.

The main types of AI are rule-based systems, where a set of predefined rules is used to make decisions, and machine learning, where algorithms learn from data and improve over time. Machine learning can be further divided into supervised learning, unsupervised learning, and reinforcement learning, depending on the type of feedback used to train the algorithm.

A brief introduction to quantum computing

Quantum computing is an emerging technology that has the potential to revolutionise the way we process and analyse information. Unlike classical computers, which rely on bits that can have a value of either 0 or 1, quantum computers use qubits that can be in a superposition of both states simultaneously. This allows for the possibility of parallel processing, which can solve complex problems exponentially faster than classical computers.

Quantum computing works by using the principles of quantum mechanics to manipulate and measure qubits. Quantum algorithms can be designed to take advantage of the unique properties of qubits, such as entanglement and interference, to perform calculations that are difficult or impossible for classical computers.

Why combine both?

Quantum computing has the potential to significantly improve the performance of Artificial Intelligence (AI) algorithms, leading to faster and more accurate results. Some of the areas where quantum computing provides a boost to AI are:

1. Parallel Processing: Quantum computers can perform certain types of calculations exponentially faster than classical computers by exploiting the principles of quantum mechanics. This can greatly speed up the training of machine learning models and enable the processing of much larger datasets.

2. Optimization: Quantum computing can be used to solve optimization problems that are difficult or impossible for classical computers. These problems are common in machine learning, such as finding the best weights for neural networks or the optimal route for a delivery vehicle.

3. Quantum Machine Learning: Researchers are exploring the potential of using quantum computing to develop new machine learning algorithms that can take advantage of the unique properties of qubits, such as entanglement and interference. These algorithms could lead to new insights and discoveries in fields such as materials science and drug discovery.

4. Quantum Encryption: Quantum computing could also be used to improve the security of AI systems by developing quantum-safe encryption methods that are resistant to attacks from quantum computers.

Let us investigate each of these in more detail.

Parallel Processing

Imagine you want to find the deepest place on Earth and a single point on the planet can be uniquely identified by a group of bits, like GPS coordinates. To move the point toward low ground, the processor modifies the group of bits, so they represent a neighbouring point that's further downhill. This is repeated until the point reaches a minimum. But is this the lowest minimum? To truly know if that minimum is the deepest place, the computer must search the whole planet. It must start at every point, go downhill, and then find the lowest minimum among all the discovered minimums. That’s a lot of starting points! If each point was enlarged to a square mile, the computer would need to try **196.9 million points!**

A quantum computer would solve this problem 196.9 million times faster. This is because of qubits that can represent all possible points *simultaneously*. The quantum computer processes these qubits, so all the points go downhill *simultaneously*, and thus it finds all the low points in one computation! We can then have the quantum computer compare all the low points and output the lowest one. If each point is a square mile, the quantum computer found the challenger deep (Mariana Trench) in one attempt—196.9 million times faster than the normal computer! This advantage is called quantum parallelism because the quantum computer solves for each point in parallel (meaning at the same time) as every other point.

How does it work?

Qubits are quantum bits. They are made from tiny particles, or groups of particles, like electrons, atoms, and photons. A particular property of these quantum particles (e.g., energy or magnetic moment) is mapped to a 0 or 1—just like normal bits. But qubits have a trick up their sleeve. They can also be 0 and 1 at the same time! This is called Superposition.

How did superposition allow the quantum computer to solve the lowest location problem in parallel? First, let’s say we can uniquely define a location on the planet with three bits (very unlikely but that’s beside the point). That means the normal computer had to process 8 different combinations of 0 and 1 (000, 001, 010, 011, etc.). For the quantum computer, say we start off with 000. Next, we modify the qubits so that each qubit is now in a superposition of 0 and 1. If we were to measure the qubits at this point, we’d find 000 one-eighth of the time, 001 one-eighth of the time, 010 one-eighth of the time, and so on—all the way to 111. Since all eight possible points are possible outcomes of a measurement, the qubits simultaneously represent all eight points! Now that all eight points are represented simultaneously, we modify the qubits further to travel downhill. Once they all hit a bottom, we measure the qubits and they (hopefully) output the lowest value!

From this simple example, we can clearly see how parallel computation can be sped up using the powers of Cubits.

Optimization

Optimization is the process of finding the best solution to a problem from a set of possible options, given its desired outcome and constraints.

The best solution can be defined in many ways: it could be the option with the lowest cost, the quickest runtime, or perhaps the lowest environmental impact. To keep things simple, best is usually defined as a cost function to be minimised. If you wanted to maximise the cost function instead (for example, if you wanted to maximise energy output from a solar cell), all you would need to do is multiply the cost by a negative one and then minimise it.

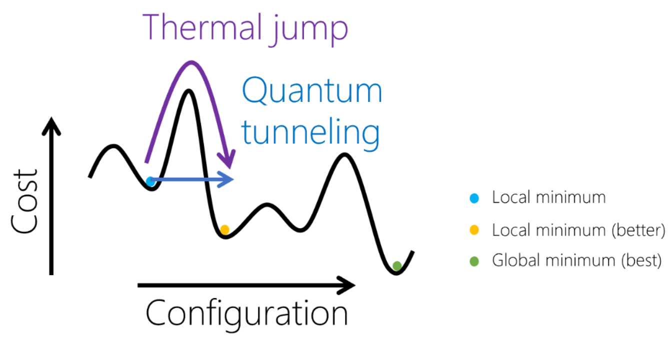
How does it work?

There exist several methods for finding the global minimum of a cost function, one of the most successful and commonly used heuristics is simulated annealing. A heuristic is a technique for finding an approximate solution, especially in situations where finding an exact solution can take too long. You can think of the technique as a random walk through the search space, where each walker creates a path through the optimization landscape.

In simulated annealing, the algorithm simulates a walker that, ideally, always moves downhill but can also take uphill moves with some non-zero probability. This creates the possibility for the walker to escape from local minima and then descend into deeper neighbouring minima. The uphill moves are called thermal jumps. That is because simulated annealing is an algorithm from physics that mimics the behaviour of materials as they are slowly cooled.

Quantum-inspired optimization makes use of the techniques for solving combinatorial problems of simulated annealing but applying quantum mechanical effects.

Quantum annealing is a quantum algorithm that is similar in spirit to simulated annealing, but it differs in a few ways. In simulated annealing, the search space is explored by making thermal jumps from one solution to the next, while quantum annealing makes use of a quantum effect called quantum tunnelling, which allows the walker to travel through these energy barriers.



In this graph, you can see the difference between the classical and the quantum approach. In simulated annealing, thermal fluctuations help a walker overcome an energy barrier, and in quantum tunnelling, quantum effects allow a walker to pass through the energy barrier.

Quantum Machine Learning

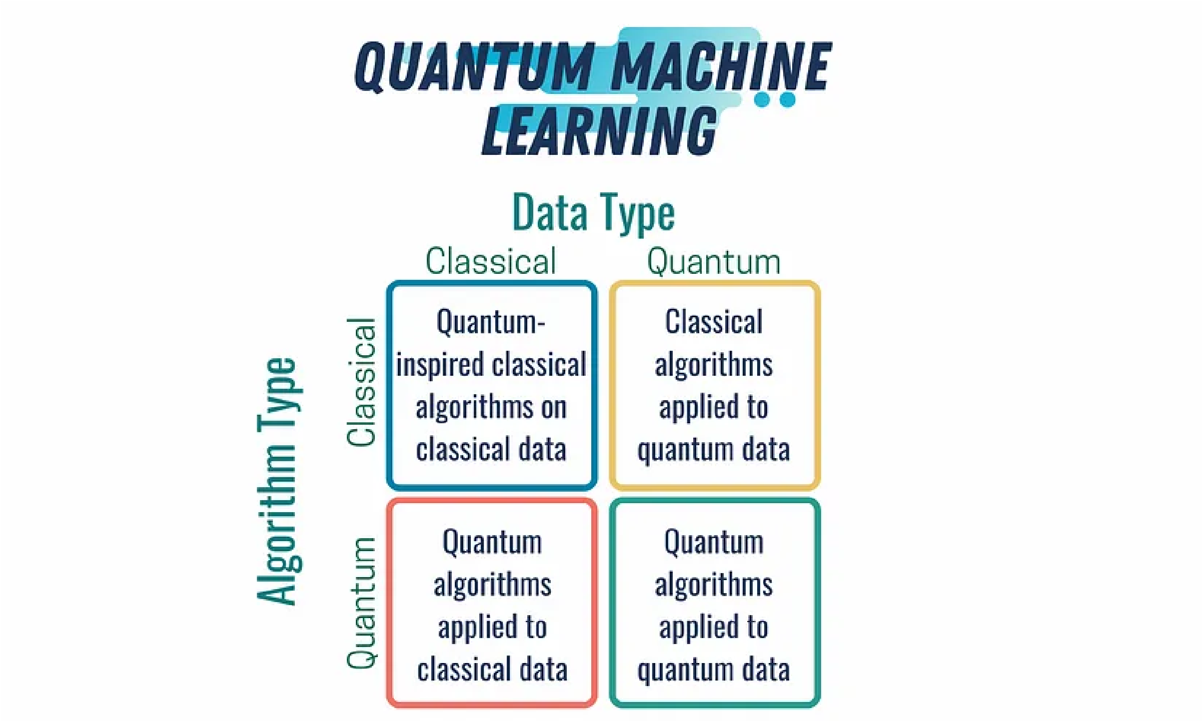
Quantum machine learning is a term used to cover 4 types of scenarios:

1. Quantum-inspired classical algorithms on classical data: such as tensor network and de-quantized recommendation systems algorithms.

2. Classical algorithms are applied to quantum data: such as neural network-based quantum States and optimising pulse sequences.

3. Quantum algorithms are applied to classical data: such as quantum optimization algorithms and quantum classification of classical data.

4. Quantum algorithms are applied to quantum data: such as quantum signal processing and quantum hardware modelling.



How does Quantum Computing help?

In 2019, A team at IBM found what they considered a prime example of a problem-solving method compatible with quantum physics — a sort of trick used in statistics, involving something called kernels.

A kernel is a measure of how related two data points are with respect to a particular feature. Think of a simple data set containing three items: BLUE, RED, and ORANGE. If you examine them as colour, RED, and ORANGE are neighbours. But if you look at the number of characters, BLUE sits between RED and ORANGE. Kernels are like lenses that allow an algorithm to classify data in different ways to find patterns that help distinguish future inputs. Implementing them is a trick to recast information in a new light, allowing you to zero in on strong relationships otherwise hidden in data.

Kernels have no inherent connection to quantum physics. But quantum computers manipulate data in a similar way, so IBM suspected that their team could design a quantum algorithm for kernels. And for supervised learning problems in particular — where the system learns from a set of labelled data — the combo could excel at learning and applying patterns. And they did it! IBM showed how, by using quantum kernels, one can learn to find the pattern hiding in the seemingly random output produced by the discrete log problem. The technique uses kernels and superpositions to both reinterpret the data points and quickly estimate how they compare to one another. Initially, the data appears random, but the quantum approach finds the right “lens” to reveal its pattern. Data points that share some key trait no longer appear randomly distributed but come together as neighbors. By making these connections, the quantum kernels help the system learn how to classify the data.

Quantum Encryption

Data collection and AI algorithms are becoming the cornerstone of the cybersecurity industry. Automated decision-making and evaluation processes provide a wider range of protection from malicious activity than legacy solutions. For instance, AI can be proactive and monitor devices for suspicious activity, instead of relying on slowly updated malware databases.

Gartner, one of the world’s leading research and IT companies, predicted that 60% of digital businesses could suffer major losses due to the inability of their security teams to manage digital risks.

After the high-profile security breaches of the past two years, businesses are becoming more aware of risk-management techniques. In fact, 79% of global executives of companies rank cybersecurity risk management as one of their top priorities.
Most CISOs have actively started to make transformations in their security and IT culture by dispersing security responsibility throughout the organisation. AI models were developed to resolve several tasks in cybersecurity such as risk assessment , encryption/decryption, breach control etc.

Why were all of these ‘obvious’ paragraphs mentioned above? It’s because still many people think that AI and Security are unrelated and two separate entities.

Encryption is the practice of converting information into a code, or cipher, that cannot be read by anyone other than the sender and (hopefully) the recipient. The information is converted into meaningless data using an encryption key, which is a number that the encryption software uses to convert the data. For the recipient to read the information, a decryption key must be used.

This key may be the same as the key used for encryption, in which case it’s symmetrical encryption, or it may be different, which is asymmetrical encryption.

Public Key Encryption uses one key that’s known by anyone to encrypt the data and another private key to decrypt it. Public key encryption is considered less secure than symmetric encryption and isn’t quantum-proof.

Modern computers are good at solving maths problems. Encryption and decryption are maths problems, but cracking strong encryption is computationally challenging, so much so that the time required to solve the problem can exceed the age of the universe. However, for quantum computers, it will take a mere few hours to crack such encryption.

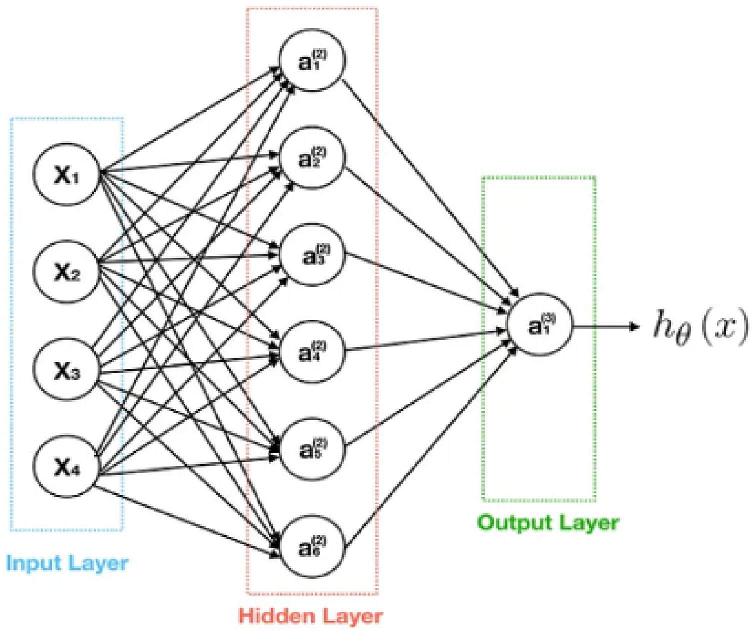
While increasing the size of encryption keys is an effective security technique, its success depends on being able to increase key sizes faster than the adversary can build a bigger quantum computer. By contrast, quantum key distribution is inherently quantum-resistant because the keys are generated by intrinsically quantum processes.

What does all of this result in?? Meet Quantum AI

Quantum AI (Artificial Intelligence) is an emerging field that combines the principles of quantum computing and machine learning to develop new algorithms and models for data analysis, prediction, and decision-making. Quantum AI aims to improve the performance of machine learning models by using the unique properties of qubits, such as superposition and entanglement, to enable faster and more accurate processing of large datasets. A core part of quantum AI is QNNs (Quantum neural networks) which are used to build them.

Classical Neural Networks:

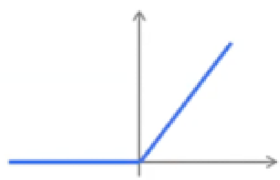
The design and operation of biological neural networks serve as a model for classical neural networks. They are made up of linked nodes arranged in layers, where each neuron performs mathematical computations on inputs to generate an output that is then sent to the following layer. To reduce the error between the expected and actual outputs, an algorithm is used to optimise the weights and biases during training. ANNs are effective in many applications, although they have drawbacks including overfitting, trouble with sequential data, and lack of interpretability. Newer architectures, like RNNs, CNNs, and transformers, have been created to get around these restrictions.



$$\hat{y} = f(x, \theta)$$

Linear model $$\hat{y} = \theta x + b$$

Perceptron model $$\hat{y} = \sigma(\theta x + b)$$



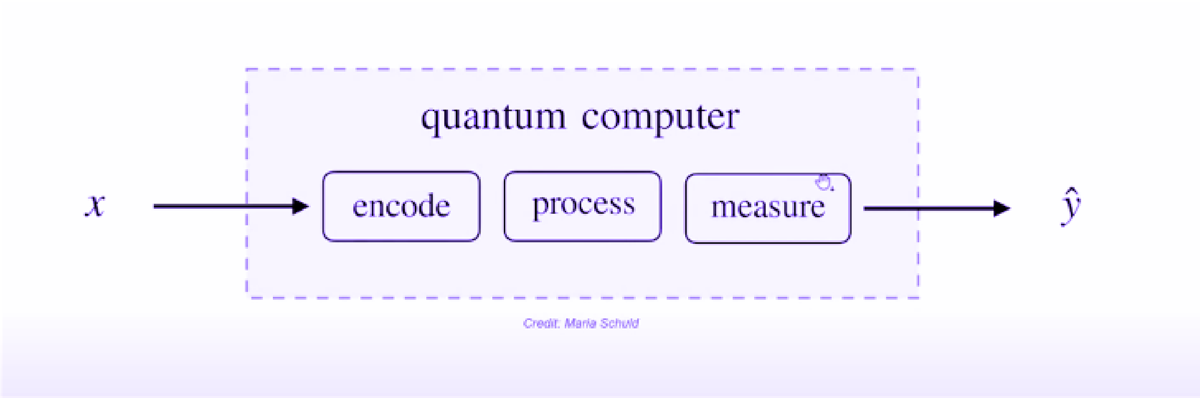
Feedforward neural network with one hidden layer $$\hat{y} = \sigma_2(\theta_2 \sigma_1(\theta_1 x + b_1) + b_2)$$

The above figure shows the equations of different Neural Networks.

The layers of linked neurons that make up traditional neural networks apply an activation function to the weighted sum of their inputs. To reduce the difference between expected and real output, ANNs are trained utilising supervised learning and optimization techniques. They have been used in a variety of applications, including voice and picture recognition. However, they have drawbacks like overfitting, difficulty handling sequential data, and poor interpretability.

QNNs

Quantum neural networks (QNNs) are a type of neural network that utilises the principles of quantum mechanics to improve the performance of classical neural networks. In a classical neural network, information is processed using classical bits, while in a QNN, information is processed using qubits, the basic unit of quantum information.



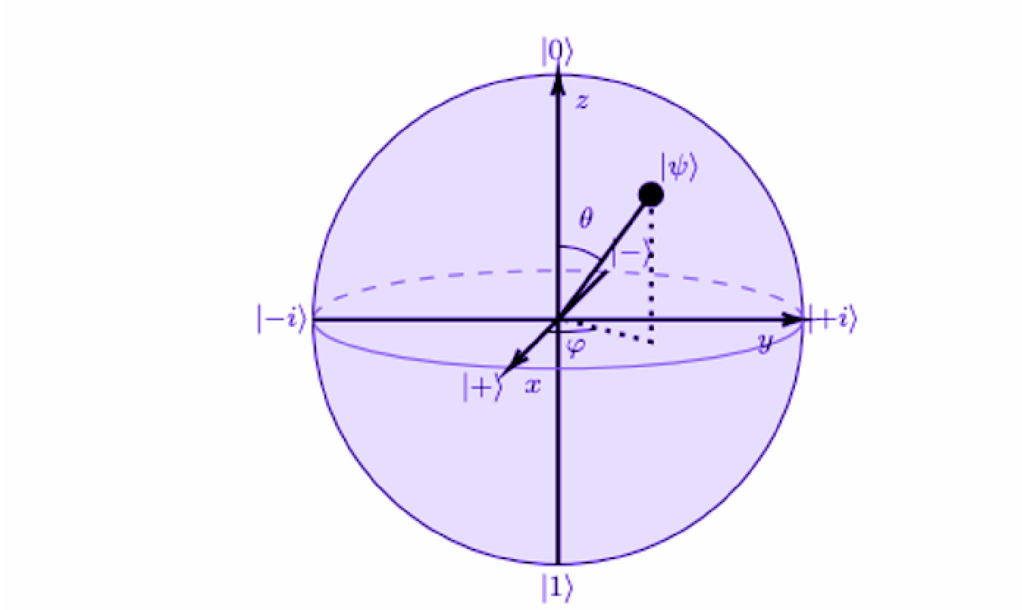
QNNs use a layered architecture like classical neural networks, with input, hidden, and output layers. However, instead of classical neurons, each node in a QNN is represented by a qubit, and the weights between the nodes are represented by quantum gates. The quantum gates allow the qubits to become entangled, which enables QNNs to perform certain tasks more efficiently than classical neural networks.

One of the primary advantages of QNNs is their ability to perform quantum parallelism (as discussed previously), where a single QNN can perform many calculations in parallel on different inputs. This is made possible by the superposition of qubits, which allows a QNN to evaluate all possible inputs at once. This can lead to significant speedups in certain types of calculations, such as image and speech recognition.

Another advantage of QNNs is their ability to perform quantum interference (also used in quantum machine learning), which enables them to detect subtle patterns in data that may be difficult to identify using classical neural networks. Quantum interference occurs when the output of a QNN is influenced by the phase relationship between the inputs and the quantum gates, which can amplify certain patterns while suppressing others.

Angle Encoding:

In quantum computing and quantum machine learning, angle encoding, often referred to as angle-based encoding, is a method for representing conventional data in a quantum state. The information is encoded as angles between quantum states, which are subsequently put into quantum gates as parameters.



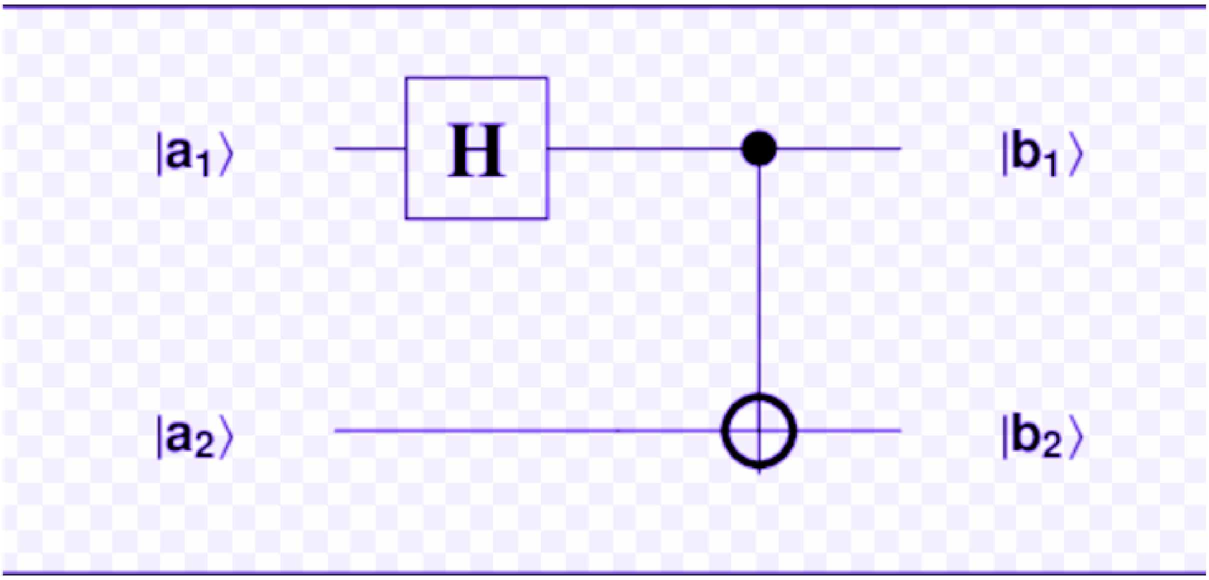
The ability to encode classical data for use in quantum neural networks (QNNs) and other quantum machine learning methods makes angle encoding particularly relevant in the field of quantum machine learning. We can compute the quantum state using quantum gates by encoding the data as angles, which might possibly yield exponential speedups over classical computation.

<u>Example:</u>

Suppose we have a real number x that we want to encode into a quantum state.
We can first use the formula:
$\theta = \cos^{-1}(\mathrm{sqrt}(x))$
to calculate the angle θ.
This angle corresponds to the rotation around the x-axis of the Bloch sphere,
Next, we use the formula
$\phi = 2\pi x$
to calculate the angle ϕ.
This angle corresponds to the rotation around the z-axis of the Bloch sphere.
Using these two angles,
we can construct the quantum state
$|\Psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\phi} \sin(\theta/2)\,|1\rangle$,
.The state |0⟩ represents the 0 bit value and the state |1⟩ represents the 1 bit value.

Quantum Circuit:

A quantum algorithm or quantum computer's operation can be mathematically represented as a quantum circuit.



It is composed of quantum gates, the fundamental constituents of quantum circuits. The basic building blocks of quantum information are quantum bits (qubits), and each gate represents a unitary operation that operates on one or more of these qubits.

Qubits can exist in a superposition of both states concurrently, in contrast to conventional circuits where each bit can only be in one of two potential states

(0 or 1). Since quantum algorithms can take advantage of this property to complete some calculations much faster than classical computers, it opens the door for the possibility of much more powerful computations.

CNOT:

A crucial gate in quantum computing is the CNOT (controlled-NOT) gate, which flips the target qubit conditionally based on the state of the control qubit.

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

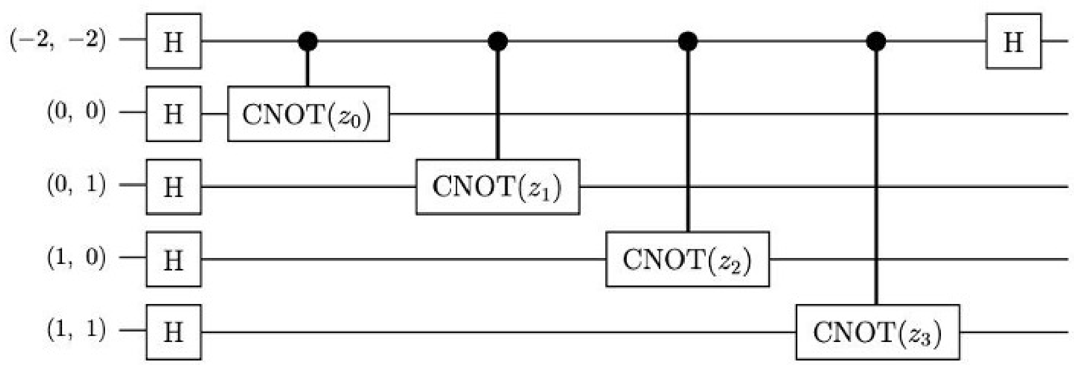
The Figure shows matrix Representation of CNOT

The control qubit is represented by the matrix's first two rows and columns being in the 0 and 1 states, respectively, and the matrix's final two rows and columns being in the 1 and 0 states, respectively. The CNOT gate leaves the target qubit unaltered while the control qubit is in the 0 state. The CNOT gate applies a NOT gate (bit flip) to the target qubit when the control qubit is in the state of 1.

Working:

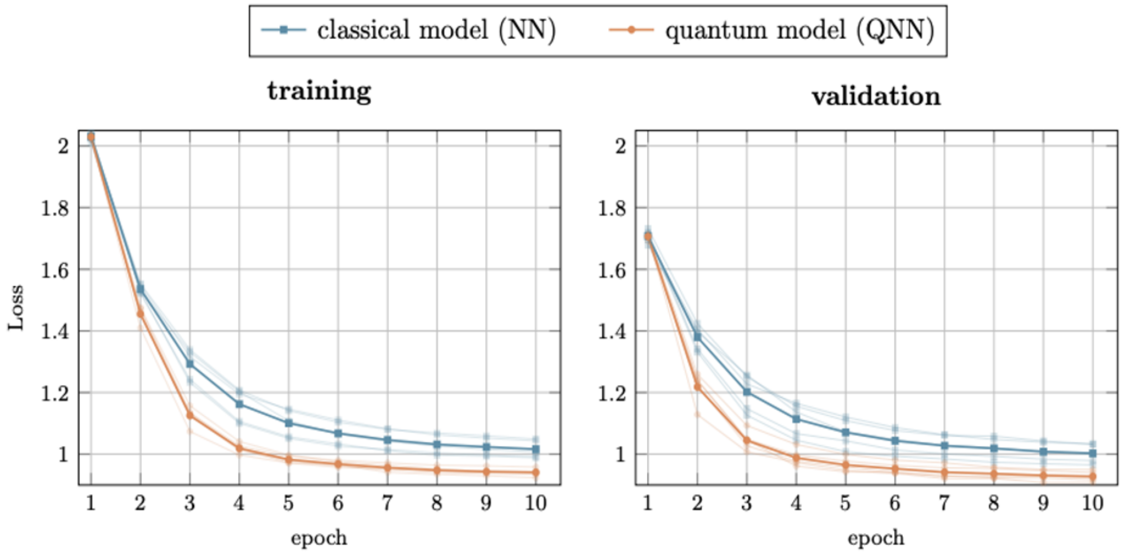
A QNN operates on a similar fundamental level as a traditional neural network.

The network receives input data, processes it through a few mathematical operations, and outputs the results.



The main distinction between a QNN and a conventional neural network is the use of quantum gates, which are created to take advantage of the special qualities of quantum systems like superposition and entanglement.
It may be applied to many different tasks, including grouping, regression, and classification. The usage of a quantum circuit as the network's computing core is a typical method for creating QNNs.

<u>Comparison of NN and QNN:</u>



The graphs of Loss VS Epoch are shown above, one can clearly see that the difference between loss of NN and QNN is significant as the number of Epoch increases.

Demo

Quantum Machine Learning (QML): Quantum computing application in the field of AI has been in both theory and practice in recent years. We intend to demonstrate how quantum computing can help advance classical machine learning.
With QML, there have been promising results when it comes to training datasets with lower cost and loss and higher accuracy. We will see how a classical machine learning engineer can get started with QML. For implementation, we will use a classification variant of QML called the QML Variational classifier.

QML Variational Classifier: Variational quantum classifiers (VQCs) are hybrid machine learning classification models that use quantum computing to their advantage.

Dataset and model Information:
In the demo, we are going to use the popular Irish data set for demonstration. For quantum simulation, we are going to use the *pennylane.ai* python package pennylane.

<u>Circuit preparation</u>: create a device using the qml API by pennylane with 2 wires/qubits. To prepare the circuit we first need to prepare states, angles, and layers required in Quantum Neural Networks (QNN).
After my above functions are prepared, I attach my device to my circuit with all relevant functions declared before being called in my circuit function.

<u>Model evaluation functions</u>: We have created cost, square_loss, and accuracy functions to evaluate our qml variational classifier model.

<u>Adding Amplitude and dimension to the original dataset</u>: The original dataset has been transformed for quantum computing with the rotation of qubits with angles around the y-axis. Amplitude is then added via the activation function to transform the data into a higher dimension. Normalisation is then performed and finally, we have transformed the whole feature set into quantum data.

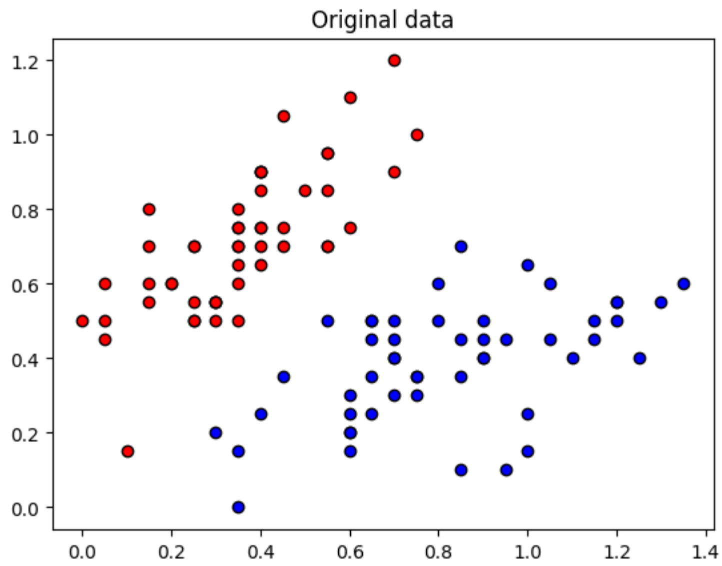


Fig: Original Dataset

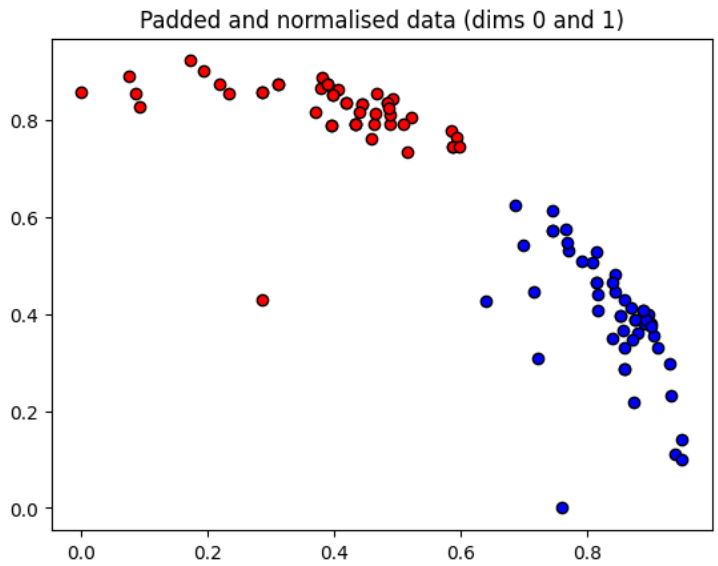


Fig: Snapshot (dimensions 0 and 1) of transformed data into a higher dimension.

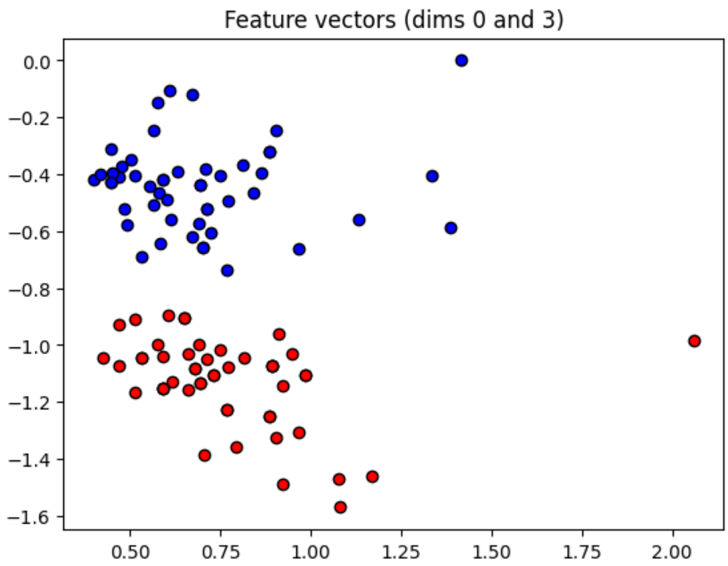


Fig: Snapshot of dimensions 0 and 3

Training Time:

It is time to train our classification model. We have introduced weights and biases for each layer. We need to initialise the optimizer function to train our model. After the training is performed, the cost has significantly reduced. The accuracy has topped to 100%.

```
Iter:    1 | Cost: 1.4490948 | Acc train: 0.4933333 | Acc validation: 0.5600000
Iter:    2 | Cost: 1.3312057 | Acc train: 0.4933333 | Acc validation: 0.5600000
Iter:    3 | Cost: 1.1589332 | Acc train: 0.4533333 | Acc validation: 0.5600000
Iter:    4 | Cost: 0.9806934 | Acc train: 0.4800000 | Acc validation: 0.5600000
.
.
.
Iter:   58 | Cost: 0.3698736 | Acc train: 0.9600000 | Acc validation: 0.9200000
Iter:   59 | Cost: 0.3420686 | Acc train: 1.0000000 | Acc validation: 1.0000000
Iter:   60 | Cost: 0.3253480 | Acc train: 1.0000000 | Acc validation: 1.0000000
```

Plotting our results: The plot represents how the final classification model looks like. With the heatmap, we can see how our model transformed after each iteration of training.

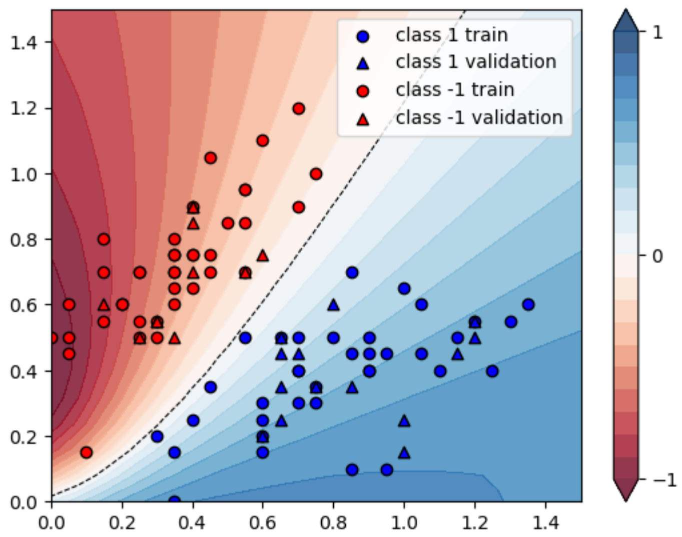


Fig: Final plot of QML variational classifier model on Irish dataset

References:-

- Quantum convolutional neural networks , Authors:Iris Cong1 , Soonwon Choi 1,2* and Mikhail D. Lukin1

- Training an Artificial Neural Network Using Qubits as Artificial Neurons: A Quantum Computing Approach, Authors: Avinash Chalumuria , Raghavendra Kuneb, B. S. Manoja,

- https://www.sciencedirect.com/topics/engineering/quantum-parallelism

- https://www.ericsson.com/en/blog/2021/9/machine-learning-on-encrypted-data

- https://www.hpe.com/us/en/insights/articles/quantum-of-encryption-1707.html

- https://towardsdatascience.com/how-may-quantum-computing-benefit-machine-learning-c96de0bef0d4

- https://blogs.iu.edu/sciu/2019/07/13/quantum-computing-parallelism/

- https://arxiv.org/abs/1802.06002

- https://learn.qiskit.org/course/ch-states/single-qubit-gates#cnot

- https://arxiv.org/abs/1810.03787

- https://arxiv.org/abs/2001.04449

- https://www.ibm.com/blogs/research/2020/06/quantum-neural-networks/

- https://www.tensorflow.org/tutorials/quickstart/beginner

- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9016555&tag=1