Internship Report

Unveiling the Mysteries of Quantum Computing and Quantum Machine Learning

Supervisor:
Dr.Bilal Tariq

Hosted Researcher:
Sajjad Ahmad

Research Center : National Center for Physics Islamabad
Researcher's Institute : National University of Sciences and Technology (NUST), Islamabad

Duration: *July 19, 2023 to November 19, 2023*

Abstract

Quantum computation is a new way of information (quantum information) processing that harnesses the unbelievable ideas of quantum mechanics such as superposition, and entanglement. The extraordinary devices that perform quantum information processing are known as quantum computers.

At the heart of a quantum computer lies the qubit, the counterpart of bits. Unlike classical bits that can only be 0 or 1, a qubit can exist in a state of superposition, where it embodies both 0 and 1 simultaneously, each with a certain probability. This inherent duality is the key to unlocking the potential of quantum computing.

It has proven to some extent that quantum computing can tackle the computation complexity of certain problems. it can also be applied in multiple areas such as machine learning, finance, cryptography ..etc.

In this research journey, our main focus was on the basics of quantum computation and quantum information, some intricate protocols that underpin their operations. After building a strong foundation, we shifted our focus to the foundations of quantum machine learning and optimization.

We have used Python and Qiskit for the development of quantum circuits and simulations. The links for quantum circuits and simulations are embedded next to them. Also, the link to the whole GitHub repository is provided at the end of the document.

Part I

BASICS OF QUANTUM COMPUTING

1 Qubit

A quantum bit or qubit is a basic unit of quantum information. It's the counterpart of the classical bit. Unlike classical bits that can only exist in a single state 0 (reps No) or 1 (reps yes), a qubit can exist in a state of superposition, where both the states are embodied simultaneously, each with a certain probability. For example:

$$|\psi\rangle \; = \; \alpha\,|0\rangle \; + \; \beta\,|1\rangle \tag{1.1}$$

Here α and β are complex numbers although for many purposes not much is lost by thinking of them as real numbers.

2 State Vector

Quantum systems are represented by quantum state vectors, often column vectors. Quantum state vectors are the backbones of quantum information processing. Equation (1.1) is a state vector of a quantum system with two possible states, 0 and 1. Put another way, the state of a qubit is a vector in a two-dimensional complex vector space. The special states $|0\rangle$ and $|1\rangle$ are known as computational basis states and form an orthonormal basis for this vector space.

Quantum state vectors must satisfy two conditions:
1. The entries of quantum state vectors are complex numbers.
2. The sum of absolute values squared of all the entries should be 1.
For example:

$$\begin{pmatrix}1\\0\end{pmatrix} = |0\rangle \quad \textit{and} \quad \begin{pmatrix}0\\1\end{pmatrix} = |1\rangle \tag{2.1}$$

$$\begin{pmatrix}\frac{1+2i}{3}\\-\frac{2}{3}\end{pmatrix} = \frac{1+2i}{2}\,|0\rangle - \frac{2}{3}\,|1\rangle \tag{2.2}$$

Click here for Python code.
Another example: The quantum state of a fan: An electric fan has three classical states; High, Low, and Off. The quantum state vector for it will:

$$|\psi\rangle = \frac{1}{2}|High\rangle - \frac{i}{2}|low\rangle + \frac{1}{\sqrt{2}}|Off\rangle \tag{2.3}$$

3 Bloch Sphere

Bloch sphere is a physical representation of all possible qubits. In other words, it's a geometrical representation of the pure state of a pure quantum mechanical system (qubit). See Fig 1 for Bloch Sphere in Python. Check the code,here.

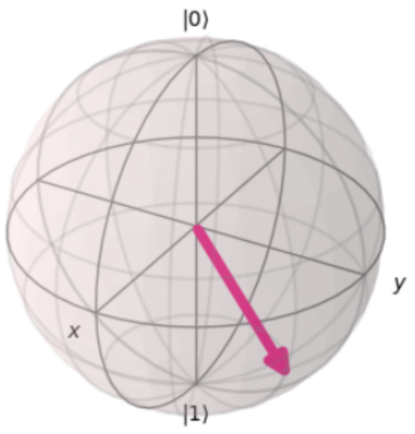


Figure 1: Visualization of Qubits on Bloch Sphere

4 Entangled States

Not all quantum state vectors of multiple systems need to be product states: for example,

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{4.1}$$

is not a product state.
If it was then there would exist quantum state vectors $|\phi\rangle$ and $|\psi\rangle$, such that:

$$|\phi\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{4.2}$$

So then, we would need: $\langle 0, \phi\rangle\ \langle 1, \psi\rangle = \langle 01, \phi \otimes \psi\rangle = 0$ it means that either $\langle 0, \phi\rangle = 0$ or $\langle 1, \psi\rangle = 0$.
But,

$$\langle 00, \phi \otimes \psi\rangle = \langle 0, \phi\rangle\langle 0, \psi\rangle = \frac{1}{\sqrt{2}} \tag{4.3}$$

and

$$\langle 11, \phi \otimes \psi\rangle = \langle 1, \phi\rangle\langle 1, \psi\rangle = \frac{1}{\sqrt{2}} \tag{4.4}$$

which contradicts.
So, the quantum state vector in (4.1) represents a correlation between systems, specifically, we say that these two systems are entangled.

4.1 Bell States

These are some important two-qubit entangled states:

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{4.5}$$

$$|\phi^-\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle \tag{4.6}$$

$$|\psi^+\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle \tag{4.7}$$

$$|\psi^-\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle \tag{4.8}$$

- The set of all the four states: $|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle$ as known as the Bell basis.
- Any quantum state vector or two qubits, or indeed any complex with entries corresponding to the four classical states of the two bits can be expressed as a linear combination of the four Bell states. e.g.

$$|00\rangle = \frac{1}{\sqrt{2}}|\phi^+]\rangle - \frac{1}{\sqrt{2}}|\psi^-\rangle \tag{4.9}$$

5 Quantum Operators and Circuits with Qiskit

In the quantum circuit model, wires represent qubits while the gates represent the operations acting on the qubits. For example:

$$THSH|0\rangle = \frac{1+i}{2}|0\rangle + \frac{1}{2}|1\rangle \tag{5.1}$$

Here H, T, and S are operators. Matrix representations of these operators are:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, T = \begin{pmatrix}1 & 0\\ 0 & \frac{1+i}{\sqrt{2}}\end{pmatrix}, S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix} \tag{5.2}$$

Operations in equation(5.1) can be represented with a circuit in qiskit. see Fig2 Operators and Circuits are extensively programmed with Qiskit. Please see it here.

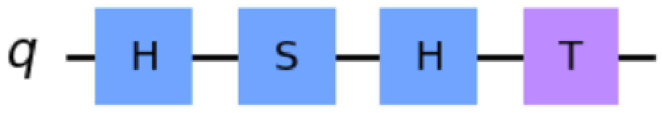


Figure 2: Multiple gates applied on a qubits in qiskit

5.1 Properties of quantum circuits

- Quantum circuits often have all the qubits initialized to $|0\rangle$.
- Default names for qubits in Qiskit are q0, q1, q2, ... etc.
- If we have only a single qubit like $|0\rangle$, so the default name is q.

6 Quantum Measurements

To extract information from a quantum system we have to measure it. Our measurements are always probabilistic, what it means is that a certain state will provide us information with some probability.

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{6.1}$$

Let's say we want to get information about a quantum system with what probability to be in state $|00\rangle$, and state $|11\rangle$. In other words, we want to measure the quantum system represented by equation (6.1). This measurement is represented with a quantum circuit in Fig 4.

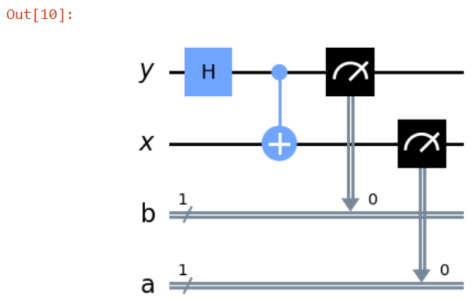


Figure 3: Bell's states measurement circuit

We may get the following results if we simulate this circuit with qiskit AerSimulator for 1000 shots. it says that the system is in state $|00\rangle$ with 51 percent probability, and 49 percent of the system is in state $|11\rangle$. see Fig 4.
This and some other measurements are programmed here.

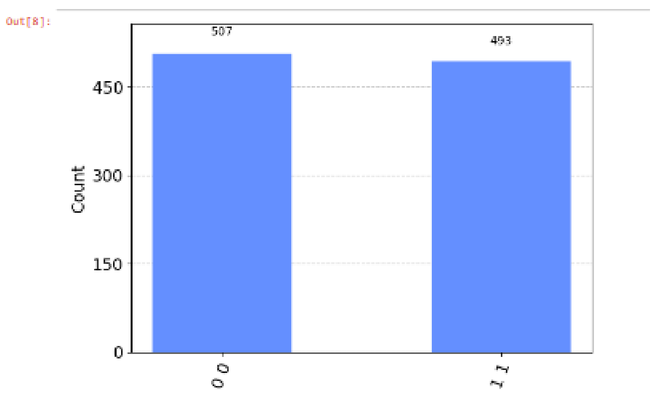


Figure 4: Simulating Above circuit for 1000 shots

7 Teleportation

It is a protocol[1] where a sender (Alice) transits a qubit to a receiver (Bob) by making use of a shared entangled quantum state (one ebit, to be specific) along with two bits of classical communication. In reality, matter is not transported in quantum teleportation, rather quantum information is teleported.

Set-up for Teleportation:

- Alice and Bob share a Bell pair,$|\phi^+\rangle$.
- Alice wants to transmit[2] an arbitrary qubit, ψ, to Bob.
- Alice applies two gates to her qubits.
- Alice measures her qubits and sends the results (classically) to Bob.
- Bob applies the appropriate corrections and recovers ψ.
  See Figure 5 for Quantum teleportation.

Someone might ask whether Alice and Bob can accomplish their task without the help of an entangled qubit. Or, Is there any way to transit qubit using classical communication alone?
The answer is no. Somehow the answer comes from the cloning theorem.

[1]A quantum protocol is a set of rules for using quantum properties to perform tasks efficiently, especially in quantum computing and cryptography.

[2]By transmission Alice means that Bob is like to be holding a qubit in the same state that ψ was in at start

A quantum circuit that describes the teleportation protocol.

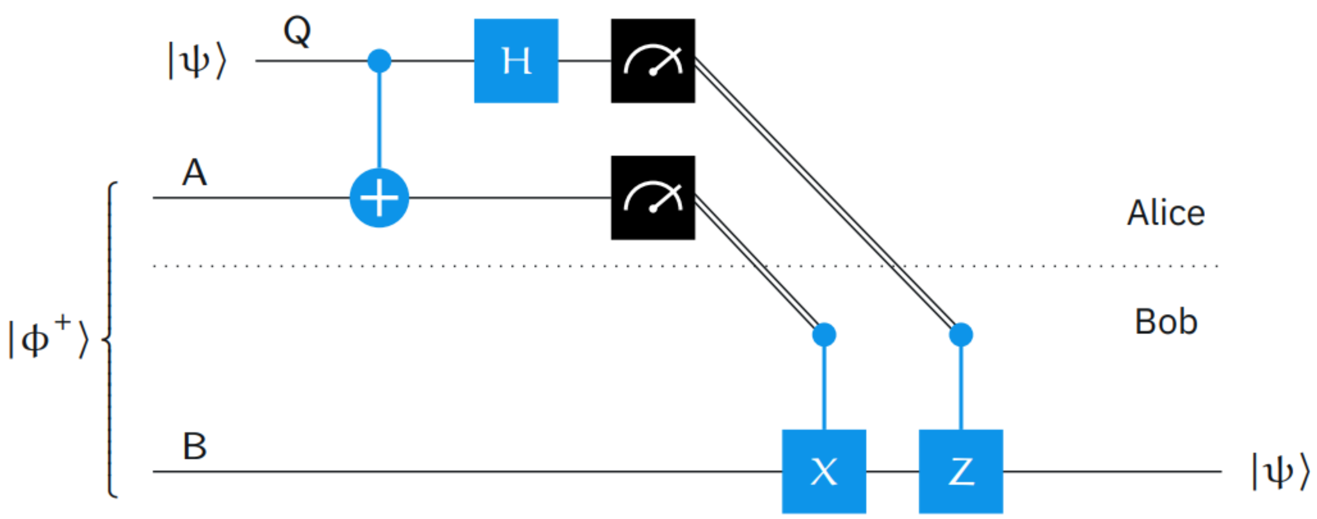


Figure 5: Quantum Teleportation Circuit[ref]

.

Circuit's explanation:

1. Q, A, and B are qubits. a and b are cbits.
2. Alice performs CNOT operation on (A, Q) and then operates Hadammard on Q.
3. Alice measures both A and Q, w.r.t standard basis, and transmits the classical outcomes to Bob.
4. Bob receives a and b from Alice.
   - If a = 1, Bob performs X-operation on qubit B, otherwise no operation.
   - If b = 1 Bob performs Z-operation on the qubit B, otherwise no operation.

That is, conditioned on ab being 00, 01, 10, 11; Bob performs the operations **I**, Z, X, ZX on qubit B.

When this circuit runs, the qubit B will be in whatever state Q was in a prior state to the protocol being executed; including whatever correlations it had with any other system.

The qiskit code for quantum teleportation can be found here.
If we want to transmit the following state;

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{7.1}$$

to Bob's qubit before measuring we will get the following state.

$$\begin{gathered}\frac{1}{2}(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|00\rangle + \\ \frac{1}{2}(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\,|01\rangle + \\ \frac{1}{2}(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|10\rangle + \\ \frac{1}{2}(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\,|11\rangle\end{gathered} \tag{7.2}$$

After simulating the circuit we may get the following probabilistic result in Fig 6. As we can see from the equation (7.2) and the bar graph in Fig. 6 the plus

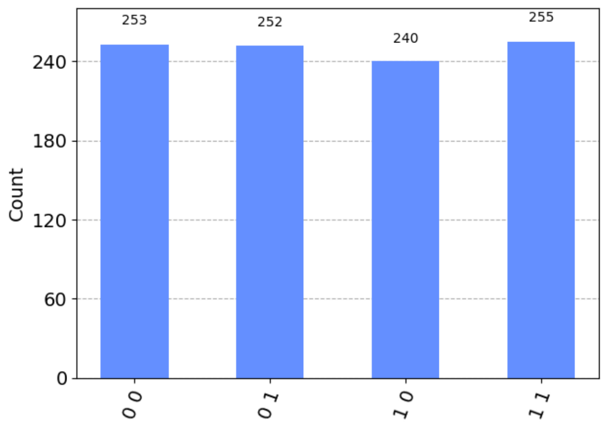


Figure 6: Transmitting plus state 7.1 to $|0\rangle$ state.

state is safely teleported to Bob with 25 percent probability.

The qiskit code for quantum teleportation is available: here.

8 Super-dense Coding

In some sense, it is a protocol that achieves a complementary aim to teleportation.
Instead of using two classical bits to transmit one qubit at the cost of one entangled bit, super-dense coding allows for transmitting two classical bits using one qubit of quantum communication (again at the cost of one ebit of entanglement).

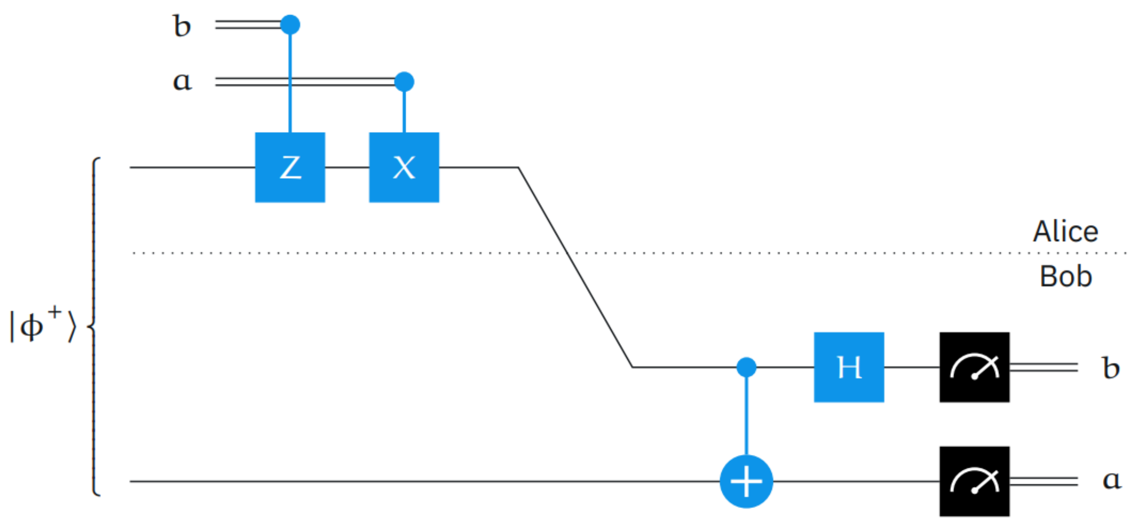


Figure 7: Alice wishes to transmit two cbits (a and b) to Bob. [ref]

- Without the entangled bit, Alice and Bob's task will be impossible.
- **Holevo's theorem** implies that two classical bits of communication cannot be reliably transmitted by a single qubit alone.

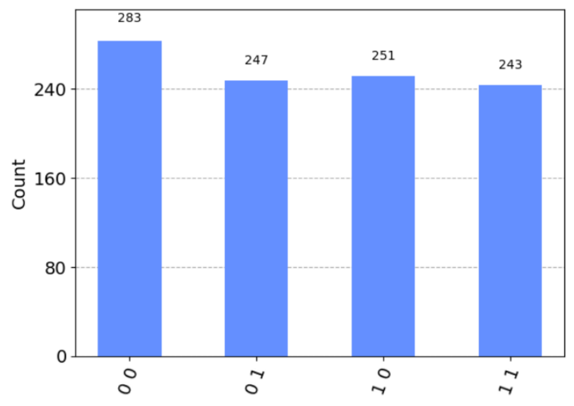


Figure 8: Simulation of super-dense coding

The qiskit code for super-dense coding is available: here.

Part II

QUANTUM MACHINE LEARNING

Machine learning learns from provided data and seeks to find patterns in the data.
On the other hand, quantum machine learning can narrowly be defined as quantum-assisted machine learning.
OR,
Machine Learning with quantum computers[3].
Quantum machine learning becomes an application of quantum computing rather than truly an interdisciplinary field of research. As we move forward this fact will be proved.

This part starts with a basic review of some terminologies, followed by an introduction to parameterized quantum circuits. It ends by analyzing some gradient descent methods for training parameterized quantum circuits.

1 Fields of Quantum Machine Learning

Just like classical machine learning quantum machine learning can be roughly split into three subfields.[1]

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

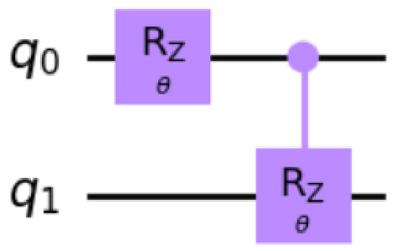
1.1 Supervised Learning

In supervised learning, we have labeled data and want to train a model with given data for predicting unseen labels in unseen data.
For example, the classification of different animal photos.

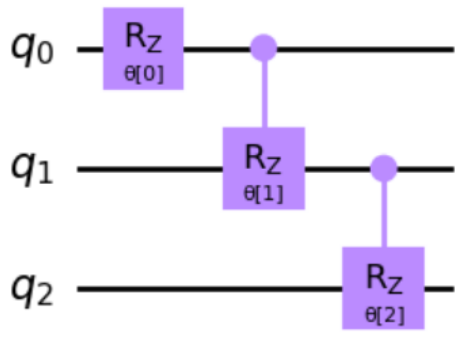
1.2 Unsupervised Learning

In the case of unsupervised learning have to find the patterns in unlabeled data.
e.g. Grouping a set of viewers based on their movie viewing history.

[3]A computer whose computation can only be described with the laws of quantum theory.



(a)



(b)

Figure 1: Parameterized circuits with same and different parameters.

1.3 Reinforcement Learning

Learning from feedback, rather than learning from given labeled data is what reinforcement learning is about. e.g. Teaching a robot to navigate a maze by rewarding it for reaching the goal and penalizing wrong moves.

2 Parameterized Quantum Circuits

Parameterized quantum circuits (also called Variational Quantum Circuits or VQCs) where gates are defined by a tunable parameter. They are the building blocks of QML algorithms.
In Fig. 1 some examples of parameterized quantum circuits are shown.

All quantum gates used in a parameterized quantum circuit are unitary, a parameterized quantum circuit itself can be described as a unitary operation on n qubits, U_θ, acting on some state $|\phi_0\rangle$, often set to $|0\rangle^{\otimes n}$.[2]
The resulting parameterized quantum state:

$$|\psi_\theta\rangle = U_\theta |\phi_o\rangle \tag{2.1}$$

- $\vec{\theta}$: is a vector of polynomial numbers of circuit parameters.
- It corresponds to angles of rotation gates (e.g. θ in $R_x(\theta)$, but more generally, they can represent any tunable parameters in quantum operations.

Qiskit code for parameterized quantum circuits is here.

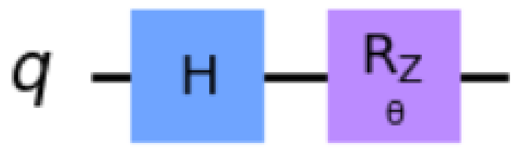
3 Properties of Parameterized Quantum Circuits

How do we choose one VQC over another? Well, there are two important descriptors mark capture the properties of VQCs.

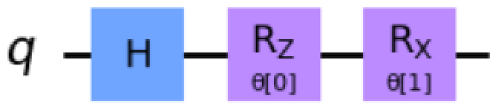
- Expressibility
- Entangling Capability
- Hardware Efficiency

3.1 Expressibility

The extent to which a circuit can generate (pure) states that are well representative of Hilbert space. [2]
It is essentially the coverage of the Hilbert space by the circuit's hypothesis space.
Highly expressive circuit parameterized quantum circuits can represent many different unitaries.



(a)



(b)

Figure 2: Parameterized circuits with same and different parameters.

If we look at the circuit (a) in Fig.2 the output states are distributed above the equator of the Bloch-sphere.
On the other hand, in circuit (b) in Fig.2 the output states cover all the Bloch-sphere, but the convergence is not uniform. There is concentration on the +X and -X poles of the sphere. So the circuit (b) is more expressive than (a). [2]

3.2 Entangling Capability

The ability of a quantum circuit to generate entangled states. Or, The average Meyer-Wallach [4] entanglement of states generated by the circuit. Estimation of the entangling capability to be:

$$Ent = \frac{1}{|S|} \sum_{\theta_i \in S} Q(|\psi_{\theta i}\rangle) \tag{3.1}$$

[4]A measurement of the average entanglement between qubits in the range between 0 to 1.

Where S = θi is the set of sampled circuit parameter vectors.
This measurement for unentangled states is 0, but for highly entangled states it is 1, such as for Bell states.[2]

3.3 Hardware Efficiency

It describes how well the circuit will perform on near-term quantum hardware.

3.4 PQCs for quantum machine learning

In quantum machine learning PQCs tend to be used for two things:

- To encode data: Where the parameters are determined using the data being encoded.
- As a quantum model, where the parameters are found by an optimization process.

An example of a data encoding circuit: The following parameterized circuit can be used to encode data:[3]

$$U_\phi(\vec{x}) = \prod_d \mathbf{U}_\phi(\vec{x}) H^{\otimes n} \tag{3.2}$$

where H denotes the conventional Hadamard and

$$\mathbf{U}_\phi(\vec{x}) = exp\left(i \sum_{S\subseteq[n]} \phi(\vec{x}) \prod_{k\in S} P_i\right) \tag{3.3}$$

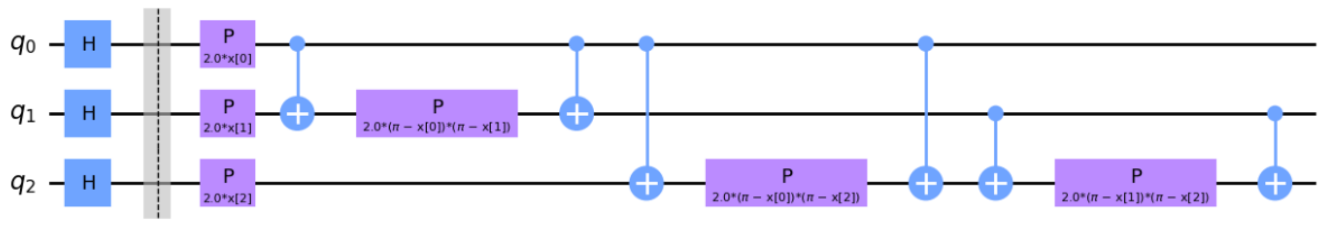
is a diagonal gate on the Puali-Z basis. It is an entangling block in a quantum circuit. The unitary in eq(3.3) is classically difficult to compute but tractable on near-term hardware.

Within entangling blocks:

- $\mathbf{U}_\phi(\vec{x})$: $P_i \in I, X, Y, Z$, Pauli matrices.
- S describes connectivity between different qubits. $S \in \binom{n}{k} combination, k = 1, ..., n$
- $\phi_s(\vec{x})$ is a data mapping function:

$$\phi_s : \vec{x} = \begin{cases} x_i, & if \;\; S = \{i\} \\ (\pi - x_i)(\pi - x_j), & if \; S = \{i, j\} \end{cases}$$

Especially, k = 2, $P_o = Z$, and $P_1 = ZZ$ is used in [3], Which in qiskit is ZZFeaturMap circuit: fig. 3



q_0
q_1
q_2
H
H
H
P
2.0*x[0]
P
2.0*x[1]
P
2.0*x[2]
P
2.0*(π − x[0])*(π − x[1])
P
2.0*(π − x[0])*(π − x[2])
P
2.0*(π − x[1])*(π − x[2])

Figure 3: ZZFeatureMap circuit.

4 Training Parameterized Quantum Circuits or

In this section, we will shift our attention to how to train circuit-based models using gradient descent-based methods. In this section, we will only use gradient-based methods.

4.1 Introduction

The learning from arbitrary data is mathematically expressed as a cost function or $f(\vec{\theta})$, known as the objective function function.
When trying to train a parameterized quantum circuit model, the function we are trying to minimize is the expectation value[5]:

$$\langle\psi(\vec{\theta})|\,\hat{H}\,|\psi(\vec{\theta})\rangle$$

where $\hat{H}$ is Hamiltonian.

Update parameters $\theta_n \longrightarrow \theta_{n+1}$	$\xrightarrow{\theta_n}$ $\xleftarrow[FunctionValue]{}$	Compute $\langle\psi(\vec{\theta})|\,\hat{H}\,|\psi(\vec{\theta})\rangle$

4.2 Computing Quantum Gradients

To compute a quantum gradient, first, we need to create a variational circuit; that is to apply a sequence of $U(\vec{\theta})$ and find the expectation value of measurement $\hat{H}$.

$$f(\vec{\theta}) := \langle\hat{H}\rangle = \langle\psi(\vec{\theta})|\,\hat{H}\,|\psi(\vec{\theta})\rangle$$

Where the trial state is prepared by our parameterized circuit.

$$|\psi(\vec{\theta})\rangle = U(\vec{\theta})\,|0000.....0\rangle$$

[5]The expectation value is what we'd expect to measure if we averaged out a large number of results.

and $\hat{H}$ is problem Hamiltonian or operator.
It is an abstract definition of a parameterized quantum circuit. Its physical implementation on a quantum device runs the quantum algorithm several times and averages the samples to get an estimate of $f(\vec{\theta})$.

Expectation values can be very easily simulated with qiskit. Code is available here.

4.3 Gradients

Let's have a function $f(\vec{\theta})$ and its gradient $\vec{\nabla} f(\vec{\theta})$, starting from an initial point. The simplest way to minimize this function it update its parameters in the deepest descent:

$$\vec{\theta_n} + 1 = \vec{\theta_n} - \eta \vec{\nabla} f(\vec{\theta_n})$$

Here, the term η: Learning rate - a small, positive hyperparameter [6] controlling the size of the update. $\vec{\nabla} f(\vec{\theta})$ represent partial derivatives w.r.t all parameters. We continue the updating parameters until we achieve local minima of the function.

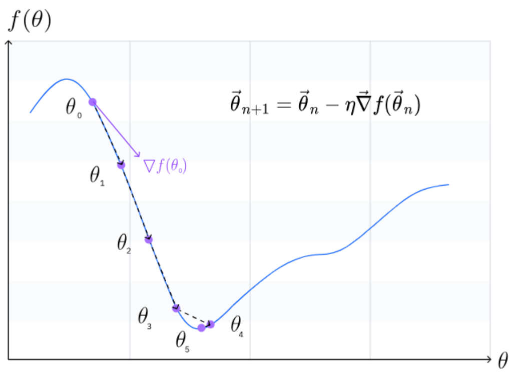


Figure 4: Gradient descent or vanilla gradient descent. Ref: Here

We will focus on the following gradient methods specifically:

- Finite Difference Gradient
- Analytic Gradients
- Quantum Natural Gradients
- Simultaneous Perturbation Stochastic Approximation

Note: All the following simulations are carried out for:

- Hamiltonian:

$$\hat{H} = \hat{Z} \otimes \hat{Z}$$

[6]A parameter different than 'theta' that our algorithm is trying to find.

- Real Amplitude quantum circuit: These circuits consist of alternating layers of Y-rotations and cx entanglements.

4.4 Finite Difference Gradient

The simplest way to find the gradient of $f(\vec{\theta})$ is that we can chose a small distance ϵ and calculate $f(\vec{\theta}+\epsilon)$ and $f(\vec{\theta}-\epsilon)$ and divide their difference by the distance:

$$Vec\nabla f(\vec{\theta}) \approx \frac{1}{2\epsilon}\Big(f(\vec{\theta}+\epsilon) - f(\vec{\theta}-\epsilon\Big)$$

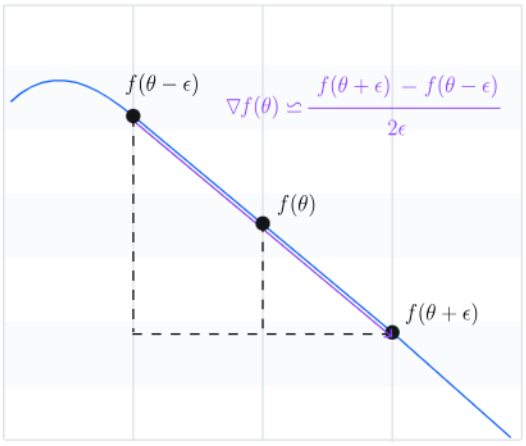

$f(\theta-\epsilon)$
$\nabla f(\theta) \backsimeq \frac{f(\theta+\epsilon) - f(\theta-\epsilon)}{2\epsilon}$
$f(\theta)$
$f(\theta+\epsilon)$

Figure 5: Finite gradient. Ref: [4]

4.5 Analytic or Vanilla Gradient

It evaluates the analytic formula for the gradients. In general, this is fairly difficult as we have to do manual calculations, but for circuit-based gradients, the *Parameter Shift Rules* gives an easy formula for calculating gradients.[5]

For a single circuit consisting of only Pauli rotations, without any coefficients, then this rule says the analytic gradient is:

$$\frac{\partial f}{\partial \theta_i} = \frac{f(\vec{\theta}+\frac{\pi}{2}\vec{e}_i) - f(\vec{\theta}-\frac{\pi}{2}\vec{e}_i)}{2} \tag{4.1}$$

In the following plot, we can see that the expectation is minimized as the number of interactions increases. Around 140 iterations it reaches the target. The results on a real device will be very noisy.

Qiskit code for finite difference and analytic gradient can be found: here

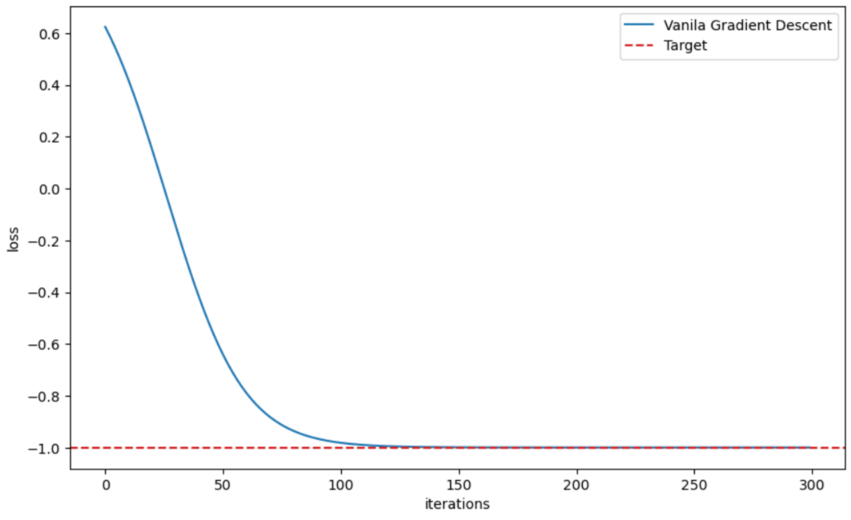


Figure 6: Minimizing the expectation value of our Hamiltonian, $\hat{H}$.

4.6 Quantum Natural Gradient

The gradient descent is not always the best strategy to find the minimum of a function. For example, if we look at the left diagram in fig.1, given the initial point $\theta_o = (x_o, y_o)$ on the edge of the loss landscape and learning rate η, we can approach the minimum in the center.
However, for the same conditions, we can't approach the minimum in the center of the right diagram. This is because we're incorrectly assuming the loss landscape varies at the same rate concerning each parameter.

Both models show the same Euclidean distance between $(x_1, y_1), (x_o, y_o)$, but this is insufficient for Model B because this metric fails to capture the relative sensitivities.
Euclidean distance between two parameters:

$$d = ||\vec{\theta}_{n+1} - \vec{\theta}_n||^2$$

With Quantum Natural gradients, we use the distance that depends on our model.

$$d = ||\langle\psi(\vec{\theta}_n)|\,|\psi(\vec{\theta}_{n+1}\rangle)||^2$$

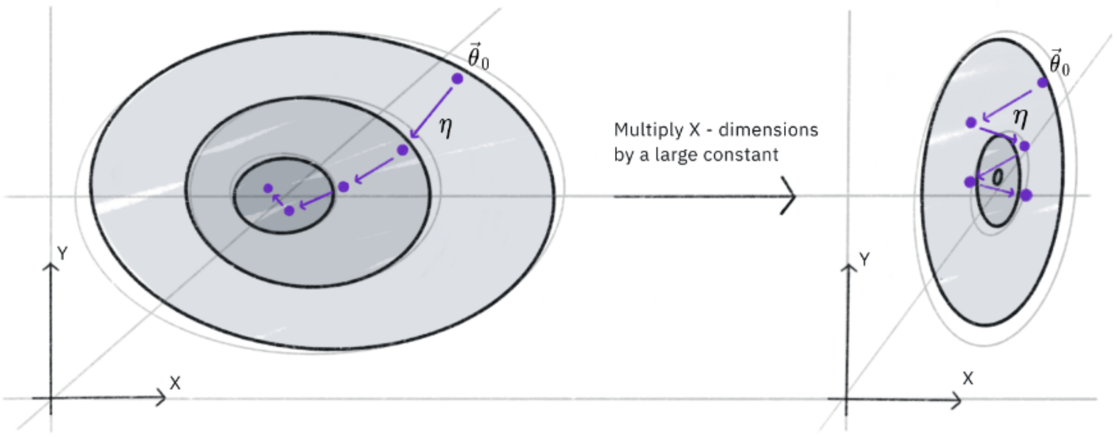


Figure 7: Loss landscape: The relationships between all possible parameters and the loss of those associated points. [4]

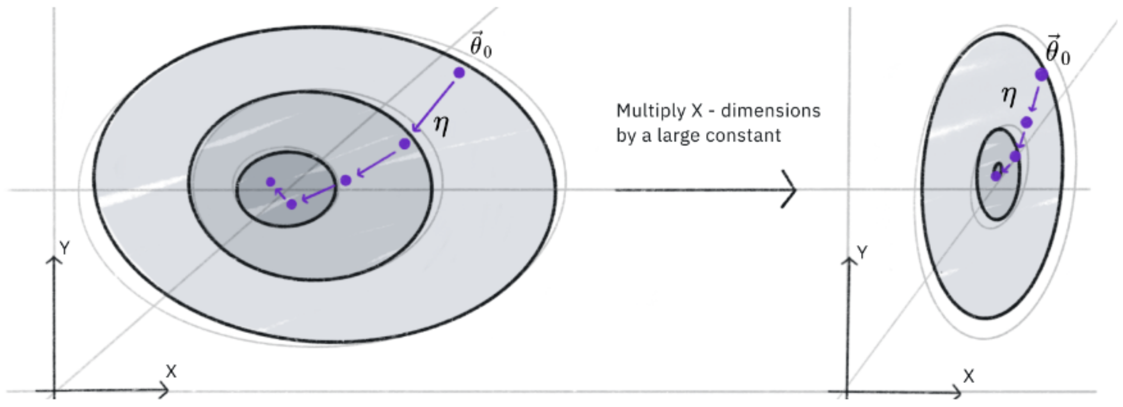


Figure 8: Models with Quantum natural gradient. [4]

The quantum natural gradient can be written as:

$$\vec{\theta}_{n+1} = \vec{\theta}_n - \eta g^{-1}(\vec{\theta})\vec{\nabla} f(\vec{\theta}_n) \tag{4.2}$$

where, $g_{ij}(\vec{\theta}) = Re[G_{ij}(\vec{\theta})]$ is quantum fisher information or fubini-study metric tensor and $G_{ij}(\vec{\theta})$ is quantum geometric tensor.[6]

$$G_{ij}(\vec{\theta}) = \langle\frac{\partial\psi(\vec{\theta})}{\partial\vec{\theta}^i}, \frac{\partial\psi(\vec{\theta})}{\partial\vec{\theta}^j}\rangle - \langle\frac{\partial\psi(\vec{\theta})}{\partial\vec{\theta}^i}, \psi(\vec{\theta})\rangle\langle\psi(\vec{\theta}), \frac{\partial\psi(\vec{\theta})}{\partial\vec{\theta}^j}\rangle \tag{4.3}$$

Having considered this information, the above two models will look.Fig.8

In fig.9, we have simulated a quantum natural gradient versus an analytic gradient. We can see that the quantum natural gradient reaches the target faster than the analytic gradient descent. However, this comes at the cost of needing to evaluate many more quantum circuits.
The qiskit code for the simulation is available:Here

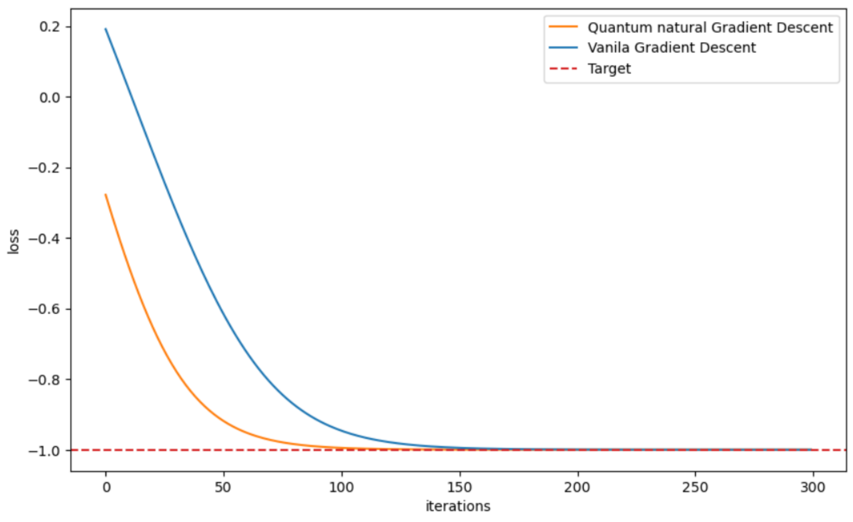


Figure 9: Vanilla Gradient versus quantum natural gradient

4.7 Simultaneous Perturbation Stochastic Approximation (SPSA)

It is an optimization technique where we randomly sample from the gradient, to reduce the number of evaluations. since we don't care about the exact values but only about the convergence, an unbiased sampling should on average work equally well.
Note well: The exact gradient will follow a smooth path to the minimum, SPSA will jump around due to the random sampling, but it will converge, given the same boundary conditions as for the exact gradient. Fig.11

If $f(\vec{\theta})$ is a vector and $\vec{\nabla} f(\vec{\theta})$ its partial derivatives w.r.t each parameter, we would need 2N function evaluations for N-parameters to calculate the gradient.

$$\vec{\nabla} f(\vec{\theta}) = \begin{pmatrix} \frac{\partial f}{\partial \theta_1} \\ . \\ . \\ . \\ . \\ \frac{\partial f}{\partial \theta_n} \end{pmatrix} \approx \frac{1}{2\epsilon} \begin{pmatrix} f(\vec{\theta} + \epsilon \vec{e}_1) - f(\vec{\theta} - \epsilon \vec{e}_1) \\ . \\ . \\ . \\ . \\ f(\vec{\theta} + \epsilon \vec{e}_n) - f(\vec{\theta} - \epsilon \vec{e}_n) \end{pmatrix} \tag{4.4}$$

If we chose a random perturbation:

$$\vec{\triangle} \in \{1, -1\}$$

and simultaneously perturb all the dimensions at once, so the gradient will become.

$$\vec{\nabla} f(\vec{\theta}) \approx \frac{f(\vec{\theta} + \epsilon \vec{\triangle}) - f(\vec{\theta} - \epsilon \vec{\triangle})}{2\epsilon} \vec{\triangle}^{-1} \tag{4.5}$$

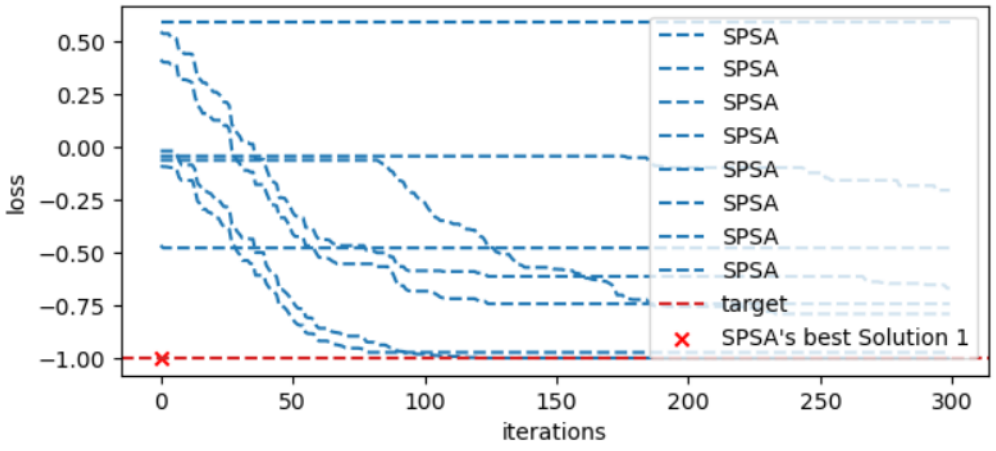


Figure 10: SPSA's results

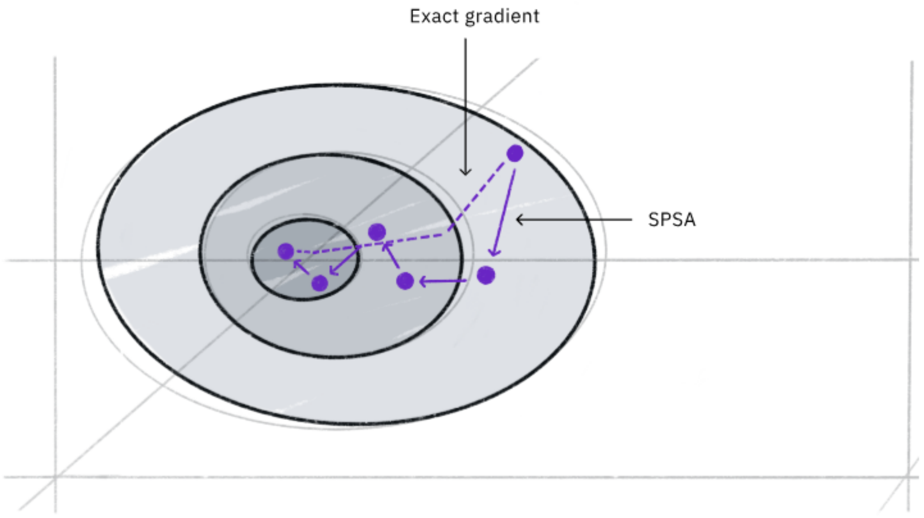


Figure 11: Finding the minimum with SPSA. Ref: [4]

If we look at figure 10, there are multiple possible solutions because of simultaneous perturbations.
In fig.[12], We have compared all of the above gradient methods. As we can see some of the results of the SPSA method converge very fast. But the quantum natural gradient gives us a specifiable solution because we can find only one best solution in contrast to SPSA.

Qiskit code for SPSA and the comparison simulation is available: here.

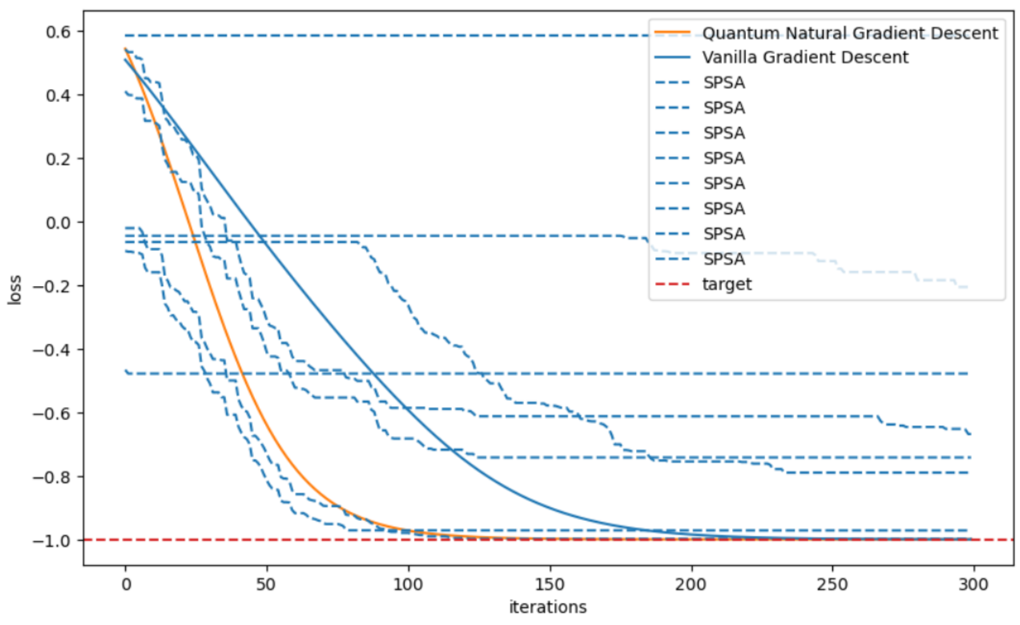


Figure 12: Comparing Analytic, Quantum Natural, and SPSA methods.

4.8 Limitations

Training with gradients works well on the small example models. Can we expect the same for a large number of qubits? To investigate it we measure the variance [7] of the gradient for different model sizes.

While finding the gradient, we always start from some random initial point and go in the direction of a decreasing slope, doing so we might end up stuck in a local minima. It is called exponentially vanishing gradients or the barren plateaus problem, which is an open research problem.

Conclusion

In the first phase of this research journey, we studied the basics of quantum computing and learned how to design gate-based quantum circuits using qiskit. Having familiarized ourselves with the basics of quantum computation, we then simulated some well-known protocols of quantum information with qiskit and analyzed their results.
In the second phase, we turned our attention to quantum machine learning. After learning the basics, we studied the backbones of optimization, the gradients.
We simulated different gradient methods for quantum machine learning and analyzed their results.

[7] A measure how much a variable deviates from the the mean.

Our codes for the quantum circuits and simulations are publicly available at:
https://github.com/Sajjad-Ahmad-phy/Quantum-Computing-and-Quantum-Machine-Learning-Internship-Qiskit-.git

References

[1] Vedran Dunjko, Jacob M. Taylor, and Hans J. Briegel. Quantum-enhanced machine learning. *Phys. Rev. Lett.*, 117:130501, Sep 2016.

[2] Sukin Sim, Peter D. Johnson, and Alá n Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12), oct 2019.

[3] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, mar 2019.

[4] https://learn.qiskit.org/course/machine-learning/training-quantum-circuitstraining-19-0.

[5] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3), mar 2019.

[6] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, may 2020.