Basics of Quantum Information Theory and Quantum Tangent Kernel

DA6300: Quantum Computing and Machine Learning

MD Karimulla Haque

Guided by,
Dr. Dhinakaran Vinayagamurthy and Dr. Chandrashekar Lakshminarayanan

April 6, 2025

- **RSA Cryptosystem:** RSA is a widely-used public-key cryptosystem that relies on the difficulty of factoring large numbers as the foundation of its security. Classical computers would need an impractical amount of time to break RSA encryption by factoring large prime numbers.

For more details on Cryptography and RSA:
`https://mdkarimullahaque.github.io/talks/Glimpses_of_Cryptography.pdf`

Example of Areas Where Quantum Computers Excel

- **RSA Cryptosystem:** RSA is a widely-used public-key cryptosystem that relies on the difficulty of factoring large numbers as the foundation of its security. Classical computers would need an impractical amount of time to break RSA encryption by factoring large prime numbers.
- **Shor's Algorithm:** Shor's Algorithm is a quantum algorithm that can efficiently factor large numbers in polynomial time, making it much faster than any classical algorithm. This breakthrough algorithm is the primary reason why quantum computers pose a threat to RSA encryption.

For more details on Cryptography and RSA:
`https://mdkarimullahaque.github.io/talks/Glimpses_of_Cryptography.pdf`

Example of Areas Where Quantum Computers Excel

- **RSA Cryptosystem:** RSA is a widely-used public-key cryptosystem that relies on the difficulty of factoring large numbers as the foundation of its security. Classical computers would need an impractical amount of time to break RSA encryption by factoring large prime numbers.
- **Shor's Algorithm:** Shor's Algorithm is a quantum algorithm that can efficiently factor large numbers in polynomial time, making it much faster than any classical algorithm. This breakthrough algorithm is the primary reason why quantum computers pose a threat to RSA encryption.
  - **Classical vs Quantum:** For classical computers, factoring a large number, such as those used in RSA keys (often hundreds of digits), is computationally infeasible and would take billions of years to break with current technology. In contrast, a quantum computer using Shor's Algorithm could factor these numbers in a matter of seconds or minutes.

For more details on Cryptography and RSA: https://mdkarimullahaque.github.io/talks/Glimpses_of_Cryptography.pdf

Cost of Building a Quantum Computer

- **Hardware:** Quantum computers require special environments to operate, such as ultra-low temperatures (near absolute zero) for certain types of qubits like superconducting qubits.

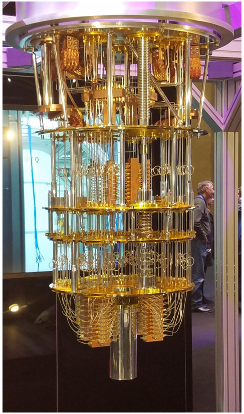


Figure: IBM Q quantum computer

Cost of Building a Quantum Computer

- **Hardware:** Quantum computers require special environments to operate, such as ultra-low temperatures (near absolute zero) for certain types of qubits like superconducting qubits.
- **Development:** Quantum computing is still in the experimental phase, meaning significant resources are required to build and maintain these systems.

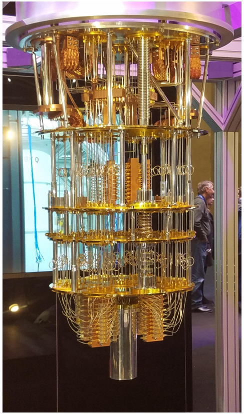


Figure: IBM Q quantum computer

- **Hardware:** Quantum computers require special environments to operate, such as ultra-low temperatures (near absolute zero) for certain types of qubits like superconducting qubits.
- **Development:** Quantum computing is still in the experimental phase, meaning significant resources are required to build and maintain these systems.
- Overall, the cost of building a quantum computer is in the range of **tens of millions of dollars**, and the technology is still in a nascent stage, meaning the cost could remain high for the next several years until it matures.

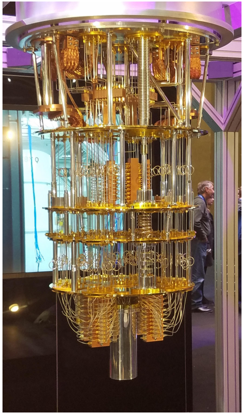
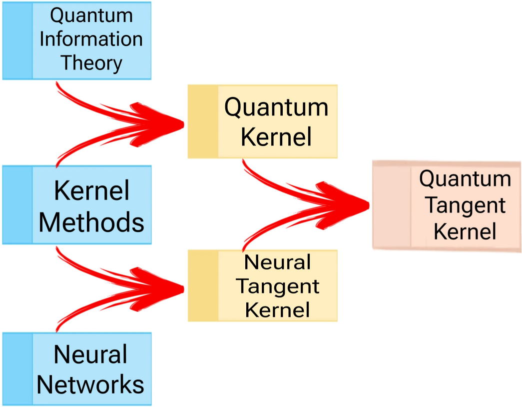


Figure: IBM Q quantum computer

Roadmap


Quantum Information Theory
Quantum Kernel
Kernel Methods
Quantum Tangent Kernel
Neural Tangent Kernel
Neural Networks

Basic Terminology

- Qubits

Basic Terminology

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**

Basic Terminology

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**

Basic Terminology

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**
  - If it takes *d* different states, it is called a **qudit**

Basic Terminology

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**
  - If it takes *d* different states, it is called a **qudit**
- **Ket:** $|*\rangle$, generally it's a column vector.

Basic Terminology

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**
  - If it takes *d* different states, it is called a **qudit**
- **Ket:** $|*\rangle$, generally it's a column vector.
- **Bra:** $\langle *| = |*\rangle^\dagger$, generally it's a row vector.

Basic Terminology

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**
  - If it takes *d* different states, it is called a **qudit**
- **Ket:** $|*\rangle$, generally it's a column vector.
- **Bra:** $\langle *| = |*\rangle^\dagger$, generally it's a row vector.
- **Bra-Ket:** $\langle *||*\rangle$ or, $\langle *|*\rangle$

Basic Terminology

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**
  - If it takes *d* different states, it is called a **qudit**
- **Ket:** $|*\rangle$, generally it's a column vector.
- **Bra:** $\langle *| = |*\rangle^\dagger$, generally it's a row vector.
- **Bra-Ket:** $\langle *||*\rangle$ or, $\langle *|*\rangle$
- **Ket-Bra:** $|*\rangle\langle *|$

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**
  - If it takes *d* different states, it is called a **qudit**
- **Ket:** $|*\rangle$, generally it's a column vector.
- **Bra:** $\langle *| = |*\rangle^\dagger$, generally it's a row vector.
- **Bra-Ket:** $\langle *||*\rangle$ or, $\langle *|*\rangle$
- **Ket-Bra:** $|*\rangle\langle *|$
- $|*\rangle \otimes |*\rangle = |*\rangle|*\rangle = |**\rangle$

- Qubits
  - If a quantum system admits two different states, it is called a **qubit**
  - If a quantum system admits three different states, it is called a **qutrit**
  - If it takes *d* different states, it is called a **qudit**
- **Ket:** $|*\rangle$, generally it's a column vector.
- **Bra:** $\langle *| = |*\rangle^\dagger$, generally it's a row vector.
- **Bra-Ket:** $\langle *||*\rangle$ or, $\langle *|*\rangle$
- **Ket-Bra:** $|*\rangle\langle *|$
- $|*\rangle \otimes |*\rangle = |*\rangle|*\rangle = |**\rangle$
- **Vector Notations:** $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, standard Qubits.

  $\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $\langle 1| = \begin{pmatrix} 0 & 1 \end{pmatrix}$

Dirac notation

- $\langle a||b\rangle = \langle a|b\rangle = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases}$
- $|a\rangle\langle b|$ has a 1 in the (a,b)-entry and 0 for all other entries.

Pauli matrices

- $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ and $\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

Tensor Product

- Let A be an $m \times n$ matrix and B be a $p \times q$ matrix then

Tensor Product

- Let A be an $m \times n$ matrix and B be a $p \times q$ matrix then
- $$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{3n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}$$

Tensor Product

- Let A be an $m \times n$ matrix and B be a $p \times q$ matrix then
- $A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{3n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}$
- is an $(mp) \times (nq)$ matrix.

Tensor Product

- Let A be an $m \times n$ matrix and B be a $p \times q$ matrix then
- $$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{3n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}$$
- is an $(mp) \times (nq)$ matrix.
- Example:
  - $$\sigma_x \otimes \sigma_z = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & \sigma_z \\ \sigma_z & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

- It is convenient to assume the vector $|0\rangle$ corresponds to the classical value 0, while $|1\rangle$ to 1 in quantum computation. Moreover it is possible for a qubit to be in a superposition state:

Superposition

- It is convenient to assume the vector $|0\rangle$ corresponds to the classical value 0, while $|1\rangle$ to 1 in quantum computation. Moreover it is possible for a qubit to be in a superposition state:
- $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$

- It is convenient to assume the vector $|0\rangle$ corresponds to the classical value 0, while $|1\rangle$ to 1 in quantum computation. Moreover it is possible for a qubit to be in a superposition state:
- $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$
- The fundamental requirement of quantum mechanics is that if we make measurement on $|\psi\rangle$ to see whether it is in $|0\rangle$ or $|1\rangle$, the outcome will be 0 (1) with the probability $|\alpha|^2(|\beta|^2)$,

- **Definition:** Quantum analog of classical bits; exists in superposition states $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$ $= cos(\frac{\theta}{2})|0\rangle + e^{i\varphi} sin(\frac{\theta}{2})|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$.

Let, $\alpha = r_1 e^{i\theta_1}$ and $\beta = r_2 e^{i\theta_2}$

Then, $r_1^2 + r_2^2 = 1$ and $\theta_1 = \theta_2 + \varphi$

Take, $r_1 = cos(\frac{\theta}{2}), r_2 = sin(\frac{\theta}{2})$

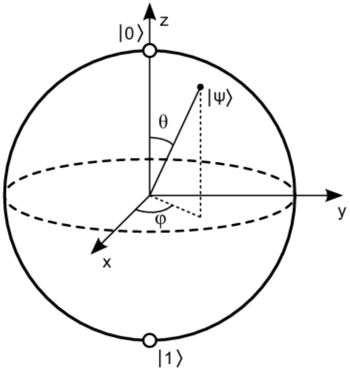


Figure: The Bloch sphere is a geometrical representation of a qubit. Qubits can take as value each point on the surface described by the two angles φ and θ. The pole points are $|0\rangle$ or $|1\rangle$.

Entangled States

- A state $|\psi\rangle \in \mathcal{H}$ written as a tensor product of two vectors as $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ (where $|\psi_a\rangle \in \mathcal{H}_a$) is called a **separable state** or a **tensor product state.**

Entangled States

- A state $|\psi\rangle \in \mathcal{H}$ written as a tensor product of two vectors as $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ (where $|\psi_a\rangle \in \mathcal{H}_a$) is called a **separable state** or a **tensor product state.**
- Let us consider a state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$ of two separated electrons. Suppose $|\psi\rangle$ may be decomposed as $|\psi\rangle = (c_1|0\rangle + c_2|1\rangle) \otimes (d_1|0\rangle + d_2|1\rangle) =$ $c_1 d_1|0\rangle \otimes |0\rangle + c_1 d_2|0\rangle \otimes |1\rangle + c_2 d_1|1\rangle \otimes |0\rangle + c_2 d_2|1\rangle \otimes |1\rangle$

Entangled States

- A state $|\psi\rangle \in \mathcal{H}$ written as a tensor product of two vectors as $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ (where $|\psi_a\rangle \in \mathcal{H}_a$) is called a **separable state** or a **tensor product state.**
- Let us consider a state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$ of two separated electrons. Suppose $|\psi\rangle$ may be decomposed as $|\psi\rangle = (c_1|0\rangle + c_2|1\rangle) \otimes (d_1|0\rangle + d_2|1\rangle) =$ $c_1 d_1|0\rangle \otimes |0\rangle + c_1 d_2|0\rangle \otimes |1\rangle + c_2 d_1|1\rangle \otimes |0\rangle + c_2 d_2|1\rangle \otimes |1\rangle$
- However this decomposition is not possible since we must have $c_1 d_2 = c_2 d_1 = 0, c_1 d_1 = c_2 d_2 = \frac{1}{\sqrt{2}}$ simultaneously, and it is clear that the above equations have no common solution. Therefore the state $|\psi\rangle$ is not separable.

Entangled States

- A state $|\psi\rangle \in \mathcal{H}$ written as a tensor product of two vectors as $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ (where $|\psi_a\rangle \in \mathcal{H}_a$) is called a **separable state** or a **tensor product state.**
- Let us consider a state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$ of two separated electrons. Suppose $|\psi\rangle$ may be decomposed as $|\psi\rangle = (c_1|0\rangle + c_2|1\rangle) \otimes (d_1|0\rangle + d_2|1\rangle) =$ $c_1 d_1|0\rangle \otimes |0\rangle + c_1 d_2|0\rangle \otimes |1\rangle + c_2 d_1|1\rangle \otimes |0\rangle + c_2 d_2|1\rangle \otimes |1\rangle$
- However this decomposition is not possible since we must have $c_1 d_2 = c_2 d_1 = 0, c_1 d_1 = c_2 d_2 = \frac{1}{\sqrt{2}}$ simultaneously, and it is clear that the above equations have no common solution. Therefore the state $|\psi\rangle$ is not separable.
- Such non-separable states are called **entangled** in quantum theory.

- we have introduced qubits to store information.

- we have introduced qubits to store information.
- it is time to consider operations acting on them

Quantum Computation

- we have introduced qubits to store information.
- it is time to consider operations acting on them
- If they are simple, these operations are called gates, or more precisely **quantum gates**

- we have introduced qubits to store information.
- it is time to consider operations acting on them
- If they are simple, these operations are called gates, or more precisely **quantum gates**
- More complicated **quantum circuits** are composed of these simple gates.

- we have introduced qubits to store information.
- it is time to consider operations acting on them
- If they are simple, these operations are called gates, or more precisely **quantum gates**
- More complicated **quantum circuits** are composed of these simple gates.
- A collection of quantum circuits for executing a complicated **quantum algorithm**

Quantum Computation

- we have introduced qubits to store information.
- it is time to consider operations acting on them
- If they are simple, these operations are called gates, or more precisely **quantum gates**
- More complicated **quantum circuits** are composed of these simple gates.
- A collection of quantum circuits for executing a complicated **quantum algorithm**
- a quantum algorithm, is a part of a **quantum computation.**

Continue...

- A quantum computation is a collection of the following three elements:

Continue...

- A quantum computation is a collection of the following three elements:
- A register or a set of registers

Continue...

- A quantum computation is a collection of the following three elements:
- A register or a set of registers
- A unitary matrix U, which is taylored to execute a given quantum algorithm

Continue...

- A quantum computation is a collection of the following three elements:
- A register or a set of registers
- A unitary matrix U, which is taylored to execute a given quantum algorithm
- Measurements to extract information we need.

Quantum Gates

- Let us consider the gate I whose action on the basis vectors are defined by $I : |0\rangle \to |0\rangle, |1\rangle \to |1\rangle$

Quantum Gates

- Let us consider the gate I whose action on the basis vectors are defined by $I : |0\rangle \to |0\rangle, |1\rangle \to |1\rangle$
- The matrix expression of this gate as

$$I = |0\rangle\langle 0| + |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

NOT, shift gate

- Similarly we introduce
- $X : |0\rangle \rightarrow |1\rangle, |1\rangle \rightarrow |0\rangle$
- $Y : |0\rangle \rightarrow -|1\rangle, |1\rangle \rightarrow |0\rangle$
- $Z : |0\rangle \rightarrow |0\rangle, |1\rangle \rightarrow -|1\rangle$

NOT, shift gate

- Similarly we introduce
- $X : |0\rangle \to |1\rangle, |1\rangle \to |0\rangle$
- $Y : |0\rangle \to -|1\rangle, |1\rangle \to |0\rangle$
- $Z : |0\rangle \to |0\rangle, |1\rangle \to -|1\rangle$
- whose matrix representations are

$$X = |1\rangle\langle 0| + |0\rangle\langle 1| = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \sigma_x$$

$$Y = |0\rangle\langle 1| - |1\rangle\langle 0| = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = -i\sigma_y$$

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \sigma_z$$

$$\text{input} \; — \boxed{U} — \; \text{output}$$

CNOT (controlled-NOT) gate

- It is a two-qubit gate

CNOT (controlled-NOT) gate

- It is a two-qubit gate
- The gate flips the second qubit (the target qubit) when the first qubit (the control qubit) is $|1\rangle$, while leaving the second bit unchanged when the first qubit state is $|0\rangle$.

CNOT (controlled-NOT) gate

- It is a two-qubit gate
- The gate flips the second qubit (the target qubit) when the first qubit (the control qubit) is $|1\rangle$, while leaving the second bit unchanged when the first qubit state is $|0\rangle$.
- Let $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ be a basis for the two-qubit system.

CNOT (controlled-NOT) gate

- It is a two-qubit gate
- The gate flips the second qubit (the target qubit) when the first qubit (the control qubit) is $|1\rangle$, while leaving the second bit unchanged when the first qubit state is $|0\rangle$.
- Let $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ be a basis for the two-qubit system.
- $|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$ and $|11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$

CNOT (controlled-NOT) gate

- It is a two-qubit gate
- The gate flips the second qubit (the target qubit) when the first qubit (the control qubit) is $|1\rangle$, while leaving the second bit unchanged when the first qubit state is $|0\rangle$.
- Let $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ be a basis for the two-qubit system.
- $|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ and $|11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$
- $U_{CNOT} : |00\rangle \rightarrow |00\rangle, |01\rangle \rightarrow |01\rangle, |10\rangle \rightarrow |11\rangle$ and $|11\rangle \rightarrow |10\rangle$ In the following, we use the standard basis vectors with components

Continue...

- whose matrix representation is

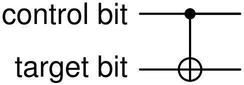
$$U_{CNOT} = |00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 10| + |10\rangle\langle 11|$$

$$= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- whose matrix representation is

$$U_{CNOT} = |00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 10| + |10\rangle\langle 11|$$
$$= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

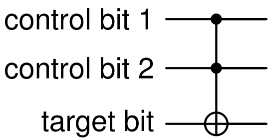
- The action of CNOT on the input state $|i\rangle|j\rangle$ is written as $|i\rangle|i \oplus j\rangle$,where $i \oplus j$ is an addition mod 2,
that is, $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1$ and $1 \oplus 1 = 0$.



Toffoli gate or, CCNOT(Controlled-Controlled-NOT) gate

- The explicit form of the CCNOT gate is

$$U_{CCNOT} = (|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I + |11\rangle\langle 11| \otimes X$$



Hadamard gate

- $U_H : |0\rangle \to |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |1\rangle \to |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Hadamard gate

- $U_H : |0\rangle \to |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |1\rangle \to |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
- whose matrix representation is

$$\mathrm{U}_H = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\langle 0| + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\langle 1| = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\text{input} \;—\boxed{H}—\; \text{output}$$

Arithmetic Circuit

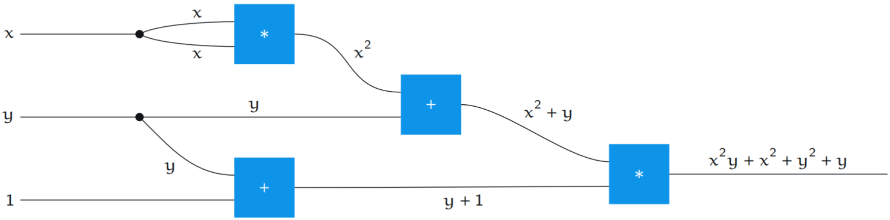


Figure: Wires store numbers and gates represent arithmetic operations, such as addition (+) and multiplication (*)

Example of Quantum Circuit

$$|0\rangle \;-\!\boxed{H}\!-\!\boxed{S}\!-\!\boxed{H}\!-\!\boxed{T}\!-\; \frac{1+i}{2}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

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

We get $THSH = \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \end{pmatrix}$

Why Kernel?

- Assume the feature map $\phi : \mathcal{X} \to \mathcal{F}$
- For example, $\phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix} \in \mathcal{F}$, where $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathcal{X}$

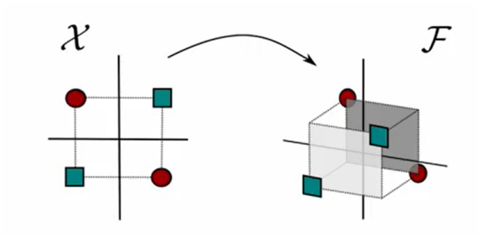


Figure: Separation may be easier in higher dimensions

Kernel Methods

- Training dataset $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$, where x_i is input data and y_i is corresponding teacher data.

Kernel Methods

- Training dataset $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$, where x_i is input data and y_i is corresponding teacher data.
- Kernel method employs a non-linear map ϕ from the original space to the higher dimensional feature space: $\phi : \mathcal{X} \to \mathcal{F}$,
  i.e., $x_i \to \phi(x_i)$,
  where $\mathcal{X}$ is an original space of the data and $\mathcal{F}$ is a higher dimensional feature space.

Kernel Methods

- Training dataset $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$, where x_i is input data and y_i is corresponding teacher data.
- Kernel method employs a non-linear map ϕ from the original space to the higher dimensional feature space: $\phi : \mathcal{X} \to \mathcal{F}$,
  i.e., $x_i \to \phi(x_i)$,
  where $\mathcal{X}$ is an original space of the data and $\mathcal{F}$ is a higher dimensional feature space.
- For two data x_i and x_j , we define $K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$, where $\langle\cdot, \cdot\rangle$ denotes the inner product on $\mathcal{F}$.

QML as a Kernel Method

- First, data are encoded into quantum states by $U_\phi(x)$ i.e.,

$$\begin{aligned} y(x,\theta) &= \langle 0^n | U^\dagger(x,\theta) O U(x,\theta) | 0^n \rangle \\ &= \langle 0^n | U_\phi^\dagger(x) V^\dagger(\theta) O V(\theta) U_\phi(x) | 0^n \rangle \qquad \{\because \mathrm{U(x,}\theta) = V(\theta) U_\phi(x)\} \\ &= Tr(O(\theta)\rho(x)), \end{aligned}$$

where weight vector $O(\theta) = V^\dagger(\theta) O V(\theta)$ and feature vector $\rho(x) = U_\phi(x)|0^n\rangle\langle 0^n|U_\phi^\dagger(x)$

QML as a Kernel Method

- First, data are encoded into quantum states by $U_\phi(x)$
i.e., $y(x,\theta) = \langle 0^n | U^\dagger(x,\theta) O U(x,\theta) | 0^n \rangle$
$$= \langle 0^n | U_\phi^\dagger(x) V^\dagger(\theta) O V(\theta) U_\phi(x) | 0^n \rangle \qquad \{\because \mathrm{U(x,}\theta) = V(\theta)U_\phi(x)\}$$
$$= Tr(O(\theta)\rho(x)),$$
where weight vector $O(\theta) = V^\dagger(\theta) O V(\theta)$ and feature vector $\rho(x) = U_\phi(x)|0^n\rangle\langle 0^n| U_\phi^\dagger(x)$

- Now the conventional quantum kernel is defined as
$K_q(x_i, x_j) = Tr(\rho(x_i)\rho(x_j)) = |\langle \phi(x_i) | \phi(x_j) \rangle|^2$, where $|\phi(x)\rangle = U_\phi(x)|0^n\rangle$

Accuracy Compare

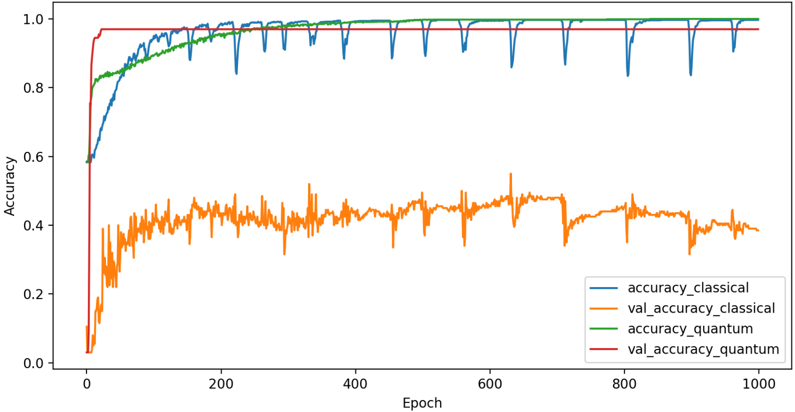


Figure: Comparison between Classical and Quantum Models

Neural Tangent Kernel (NTK)

- Suppose $f(x_i, \theta)$ is the output of a neural network where θ is parameters in the network, and x_i is the input data.

Neural Tangent Kernel (NTK)

- Suppose $f(x_i, \theta)$ is the output of a neural network where θ is parameters in the network, and x_i is the input data.
- Now we approximated by the first-order expansion with respect to the parameters around the initial values:
  $f(x_i, \theta) \simeq f(x_i, \theta_0) + \nabla_\theta f(x_i, \theta_0)^T(\theta - \theta_0)$, where θ_0 is the initial values of parameters of a neural network.

Neural Tangent Kernel (NTK)

- Suppose $f(x_i, \theta)$ is the output of a neural network where θ is parameters in the network, and x_i is the input data.
- Now we approximated by the first-order expansion with respect to the parameters around the initial values: $f(x_i, \theta) \simeq f(x_i, \theta_0) + \nabla_\theta f(x_i, \theta_0)^T(\theta - \theta_0)$, where θ_0 is the initial values of parameters of a neural network.
- This approximation allows us to interpret the neural network as a linear model with a feature map $\phi(x) = \nabla_\theta f(x, \theta_0)$

Neural Tangent Kernel (NTK)

- Suppose $f(x_i, \theta)$ is the output of a neural network where θ is parameters in the network, and x_i is the input data.
- Now we approximated by the first-order expansion with respect to the parameters around the initial values: $f(x_i, \theta) \simeq f(x_i, \theta_0) + \nabla_\theta f(x_i, \theta_0)^T(\theta - \theta_0)$, where θ_0 is the initial values of parameters of a neural network.
- This approximation allows us to interpret the neural network as a linear model with a feature map $\phi(x) = \nabla_\theta f(x, \theta_0)$
- Using this feature map, we define the following neural tangent kernel (NTK): $K_{ntk}(x_i, x_j) = \nabla_\theta f(x_i, \theta_0)^T \nabla_\theta f(x_j, \theta_0)$

- Now from K_q and K_{ntk}, we define the following quantum tangent kernel (QTK):
  $K_{qtk}(x_i, x_j) = \nabla_\theta y(x_i, \theta_0)^T \nabla_\theta y(x_j, \theta_0)$,
  where $y(x, \theta) = \langle 0^n | U^\dagger(x, \theta) O(\theta) U(x, \theta) | 0^n \rangle$

Numerical Experiment

- First one is a circuit where x is encoded only at the first layer as in the bellow Figure. $U_{shallow}(x,\theta) = V(\theta)U_{\phi}(x)$, where $V(\theta)$ is a parameterized unitary and $U_{\phi}(x)$ is a quantum feature map to encode data.

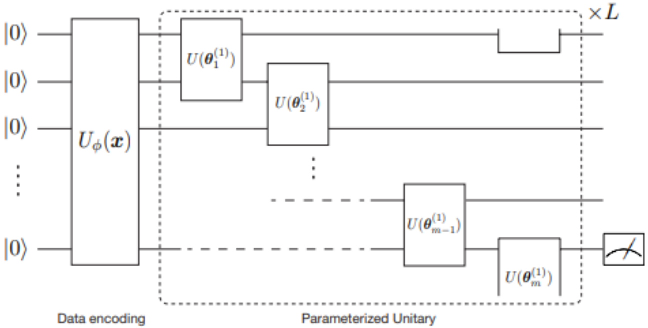


Figure: The m-qubit ansatz used for numerical simulations. $U_{\phi}(x)$ is the quantum feature map for encoding classical data. $U(\theta_j^{(i)}) \in SU(4)$ is the parameterized unitary of the i-th layer. At the output, we measure the Pauli Z expectation value of the final qubit.

- Next, in order to increase the non-linearity, we consider a multi-layered circuit that alternates between data encoding and a parameterized unitary as shown in the bellow Figure. $U_{deep}(x, \theta) = \Pi_{i=1}^{L} V(\theta_i) U_{\phi}(x)$,

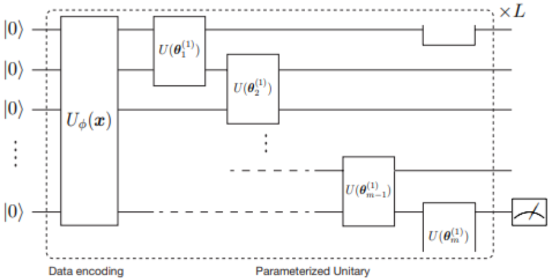


Figure: Quantum circuit that defines deep quantum tangent kernel. $U_{\phi}(x)$ is the feature map and $U(\theta_j^{(i)}) \in SU(4)$ is the parameterized unitary of the i-th layer. To increase the non-linearity of the kernel, the feature map $U_{\phi}(x)$ and the parameterized unitary are iteratively applied. At the output, we measure the Pauli Z expectation value of the final qubit.

Numerical Experiment

- The "ansatz-generated" dataset is generated in the following manner. Four dimensional random value data x_i are inputted into $U_{deep}(x, \theta)$ consisting of n = 4 qubits and L = 10 layers.

- The “ansatz-generated” dataset is generated in the following manner. Four dimensional random value data x_i are inputted into $U_{deep}(x, \theta)$ consisting of n = 4 qubits and L = 10 layers.

- Then, evaluate the expectation value $I(x_i, \theta) = \langle 0^n | U^\dagger_{deep}(x_i, \theta) Z_4 U_{deep}(x_i, \theta) | 0^n \rangle$ with a randomly chosen θ. We label each x_i as 1 and -1 if $I(x, \theta) \geq 0$ and $I(x, \theta) < 0$, respectively.

Numerical Experiment

- The “ansatz-generated” dataset is generated in the following manner. Four dimensional random value data x_i are inputted into $U_{deep}(x, \theta)$ consisting of n = 4 qubits and L = 10 layers.
- Then, evaluate the expectation value $I(x_i, \theta) = \langle 0^n | U^{\dagger}_{deep}(x_i, \theta) Z_4 U_{deep}(x_i, \theta) | 0^n \rangle$ with a randomly chosen θ. We label each x_i as 1 and -1 if $I(x, \theta) \geq 0$ and $I(x, \theta) < 0$, respectively.
- Here, generate the 15,000 samples of the four dimensional input data x_i and its label y_i. They are splitted into 10,000 training and 5000 test data.

- Behavior of training losses and training accuracies for different number of layers during training.

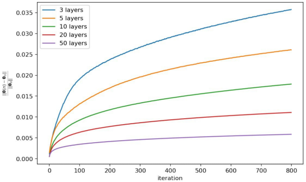


Figure: Relative norm change in the parameters of the quantum circuit from initial values during training by gradient descent. $\theta(n)$ is the parameter at the n-th iteration. θ_0 is initial value of the parameter.

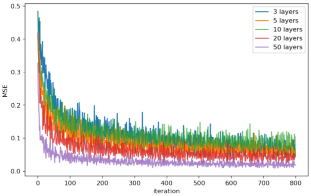


Figure: Mean Square Error

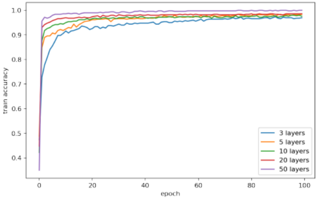


Figure: Train Accuracy

Classification accuracy for three types of kernels

- Classification accuracy for SVM with three types of kernels. Three SVMs classify the ansatz-generated dataset generated by a quantum circuit for deep QTK

Kernel	Accuracy
Quantum kernel	0.7842
Shallow quantum tangent kernel	0.7484
Deep quantum tangent kernel	0.812

The distribution of outputs

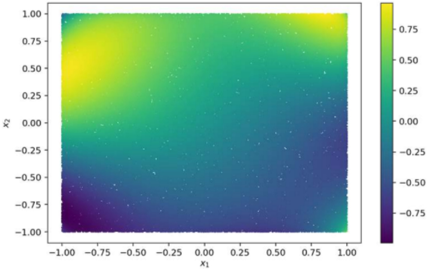


Figure: The distribution of outputs of the quantum circuit with L = 10 layers which can be interpreted as the conventional quantum kernel method.

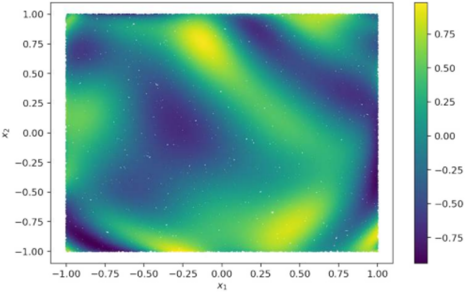


Figure: The distribution of outputs of the quantum circuit with L = 10 layers beyond the conventional quantum kernel

Conclusion

- Here quantum tangent kernel (QTK) and deep quantum tangent kernel which cannot be interpreted as conventional quantum kernel methods

Conclusion

- Here quantum tangent kernel (QTK) and deep quantum tangent kernel which cannot be interpreted as conventional quantum kernel methods
- QTK is defined by applying the formulation of NTK to parametrized quantum circuits.

Conclusion

- Here quantum tangent kernel (QTK) and deep quantum tangent kernel which cannot be interpreted as conventional quantum kernel methods
- QTK is defined by applying the formulation of NTK to parametrized quantum circuits.
- It imply that deep parameterized quantum circuits with repetitive data encoding unitary have a higher representation power and better performance for quantum machine learning than the conventional quantum kernel method.

Conclusion

- Here quantum tangent kernel (QTK) and deep quantum tangent kernel which cannot be interpreted as conventional quantum kernel methods
- QTK is defined by applying the formulation of NTK to parametrized quantum circuits.
- It imply that deep parameterized quantum circuits with repetitive data encoding unitary have a higher representation power and better performance for quantum machine learning than the conventional quantum kernel method.
- Hence, better ansatz constructions and parameter optimization methods are crucially important.

References

- Huang, H. Y., et al. **Power of data in quantum machine learning**. Nature Communications, 12.1 (2021): 2631. Available at: https://www.nature.com/articles/s41467-021-22539-9
- Sebastian E., et al. **A hyperparameter study for quantum kernel methods**. Springer Nature, 6.44 (2024). Available at: https://link.springer.com/article/10.1007/s42484-024-00172-1
- Shirai, N., Kubo, K., Mitarai, K., Fujii, K. (2024). **Quantum tangent kernel**. Physical Review Research, 6(3), 033179. Available at: https://doi.org/10.1103/PhysRevResearch.6.033179
- Michael A. Nielsen and Isaac L. Chuang - **Quantum Computation and Quantum Information**. Cambridge University Press. Available at: https://doi.org/10.1017/CBO9780511976667
- Peter Wittick - **Quantum Machine Learning: What Quantum Computing Means to Data Mining**. Elsevier Inc. Available at: https://doi.org/10.1016/C2013-0-19170-2
- Mikio Nakahara and Tetsuo Ohmi - **Quantum Computing: From Linear Algebra to Physical Realizations**. CRC Press. Available at: https://doi.org/10.1201/9781420012293
- John Watrous - **Basics of quantum information and Fundamentals of quantum algorithms**, IBM Quantum Learning. Available at: https://learning.quantum.ibm.com
- MD Karimulla Haque - **Glimpses of Cryptography**. MAC Math Fest (2021). Available at: https://mdkarimullahaque.github.io/talks/Glimpses_of_Cryptography.pdf
- Pictures were taken from Google Images.

Thank You