Lecture Notes on Variational Quantum Algorithms (VQAs)

Le Bin Ho

Tohoku University, April 2025

Contents

1 Introduction to quantum computing

1.1 Algebra for quantum computing

Quantum computing is built on the mathematical foundations of linear algebra, complex numbers, and Hilbert spaces. To understand variational quantum algorithms (VQAs), we must first explore these fundamental mathematical structures.

1.1.1 Complex numbers

A complex number is of the form

$$z = a + bi, \tag{1}$$

where a, b are real numbers, and i is the imaginary unit satisfying $i^2 = -1$.

- The conjugate of z is $z^* = a - bi$.
- The modulus (absolute value) of z is $|z| = \sqrt{a^2 + b^2}$.
- Euler's formula:

$$e^{i\theta} = \cos\theta + i\sin\theta. \tag{2}$$

  Using this, a complex number can be expressed in polar form

$$z = |z|e^{i\theta}. \tag{3}$$

1.1.2 Vector spaces and inner product

Quantum states are represented as vectors in a Hilbert space, a vector space equipped with an inner product.

- A vector space V over a field $\mathbb{C}$ (complex numbers) consists of elements called vectors, which can be:
  - Added together: $\mathbf{v} + \mathbf{w}$.
  - Scaled by scalars: $\alpha\mathbf{v}$, where $\alpha \in \mathbb{C}$.
- The inner product of two vectors $|\psi\rangle$ and $|\phi\rangle$ is defined as

$$\langle\psi|\phi\rangle = \sum_i \psi_i^* \phi_i, \tag{4}$$

  where ψ^* denotes the complex conjugate of ψ.
- The norm of a vector is given by

$$||\psi|| = \sqrt{\langle\psi|\psi\rangle}. \tag{5}$$

1.1.3 Matrices and Operators

Quantum operations are performed using matrices. Some important types of matrices include

- Unitary matrices: satisfy $U^\dagger U = I$, ensuring that quantum operations are reversible.
- Hermitian matrices: satisfy $H^\dagger = H$, important because observables in quantum mechanics correspond to Hermitian operators.
- Pauli Matrices: the fundamental single-qubit operators are the Pauli matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{6}$$

  These matrices describe fundamental quantum operations such as bit flips and phase flips.

1.2 Basics of quantum computing

1.2.1 Qubit representation

- A qubit is the fundamental unit of quantum information. Unlike a classical bit, which can be either 0 or 1, a qubit exists in a superposition of both states $|0\rangle$ and $|1\rangle$ as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{7}$$

  where α, β are complex numbers satisfying the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1. \tag{8}$$

- The computational bases $|0\rangle$ and $|1\rangle$ can be represented by column vectors as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{9}$$

- Example: a single qubit state $|\psi\rangle$ is represented as

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{10}$$

- The Bloch sphere representation

  A single qubit state can be visualized on the Bloch sphere, where

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \tag{11}$$

  where θ represents the polar angle, and ϕ represents the azimuthal angle. This representation helps illustrate quantum gates as rotations in three-dimensional space.

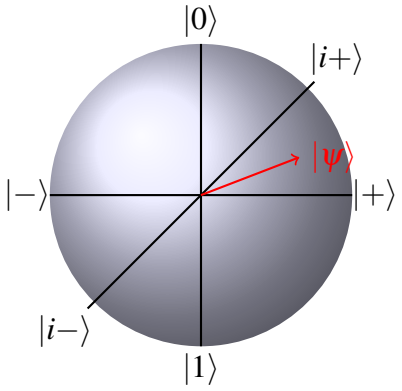


Figure 1: Representation of a single qubit state on the Bloch sphere. The state $|\psi\rangle$ is shown as a red vector.

1.2.2 Multiple qubits and tensor product

For a multi-qubit system, we use the tensor product to describe the combined quantum state. Example: If we have two qubits,

$$|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle, \quad |\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle, \tag{12}$$

the combined two-qubit state is

$$|\psi_1\rangle \otimes |\psi_2\rangle = \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\alpha_2 \\ \beta_1\beta_2 \end{pmatrix}. \tag{13}$$

For an n-qubit system, the state space grows exponentially, requiring 2^n complex numbers to describe a general quantum state.

1.2.3 Quantum gates

Quantum gates are modeled as unitary matrices, which manipulate qubits through unitary transformations. Some well-known gates are as follows.

Gate	Description and Matrix
Pauli-X Gate (Bit-flip gate)	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y Gate (Bit and phase flip)	$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z Gate (Phase-flip gate)	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard Gate H	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Phase Gate S	$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
T Gate	$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
Rotation Gates	$R_x(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$ $R_y(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$ $R_z(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$

Table 1: Single-qubit quantum gates

1.2.4 Quantum circuits

A **quantum circuit** is a model for quantum computation in which a series of quantum gates are applied to qubits to perform a computation. Quantum circuits are analogous to classical logic circuits but operate based on the principles of quantum mechanics, such as *superposition* and *entanglement*.

A quantum circuit consists of the following components:

Gate	Description and Matrix
Controlled-NOT (CNOT) Gate	$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
Swap Gate	$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
Toffoli Gate (CCNOT)	$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$
Fredkin Gate (Controlled-SWAP)	$\text{Fredkin} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Table 2: Multi-qubit quantum gates

- **Qubits:** The fundamental units of quantum information, initialized in a known state (usually $|0\rangle$).
- **Quantum gates:** Unitary transformations applied to qubits to manipulate their states.
- **Measurement:** The process of extracting classical information from qubits, collapsing the quantum state into a definite outcome.

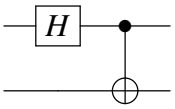
Example: A simple quantum circuit. Consider a basic quantum circuit with a single qubit where we apply a Hadamard gate followed by a measurement. This can be represented as:

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \xrightarrow{\text{Measure}} \{0,1\} \text{ (with equal probability).}$$

Example: Bell state preparation circuit. A common quantum circuit is the **Bell state** preparation, which generates an entangled state. It consists of:

1. Applying a Hadamard gate to the first qubit.
2. Applying a CNOT gate, with the first qubit as the control and the second qubit as the target.

The quantum circuit diagram is:



This transforms the initial state $|00\rangle$ into the maximally entangled Bell state:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Example: Quantum Teleportation Circuit. Quantum teleportation is a protocol that allows the transmission of an unknown quantum state from one qubit to another using entanglement and classical communication. It is a fundamental demonstration of quantum information transfer.

The quantum teleportation circuit follows these steps:

- Qubit 1 (q_1): The unknown state $|\psi\rangle$ to be teleported.
- The second qubit (q_2) is entangled with the third qubit (q_3) via a Hadamard gate and CNOT gates.
- Alice performs operations including CNOT and Hadamard gates and measurements on q_1 and q_2.
- Classical information is sent to Bob, who applies the necessary correction to q_3.

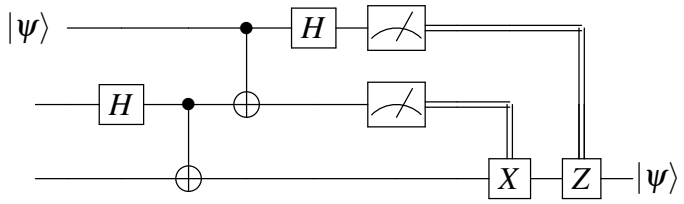


Figure 2: Quantum circuit for quantum teleportation.

Let's go through the step-by-step derivation of the teleportation process.

Step 1: Initial state of the system.

- Qubit q_1 in an unknown state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{14}$$

- Qubits q_2 and q_3 in the Bell state:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{15}$$

- Thus, the full three-qubit state is:

$$|\psi\rangle \otimes |\beta_{00}\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{16}$$

Expanding the tensor product:

$$|\psi\rangle \otimes |\beta_{00}\rangle = \frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle\Big). \tag{17}$$

Step 2: Alice applies a CNOT gate on q_1 and q_2.

- The CNOT gate (with q_1 as control and q_2 as target) maps:

$$|00\rangle \to |00\rangle, \quad |01\rangle \to |01\rangle, \quad |10\rangle \to |11\rangle, \quad |11\rangle \to |10\rangle.$$

- Applying this to our state:

$$\frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle\Big). \tag{18}$$

Step 3: Alice applies a Hadamard gate on q_1.

- The Hadamard gate transforms:

$$|0\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |1\rangle \to \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

- Applying Hadamard on q_1:

$$\frac{1}{2}\Big[|0\rangle(\alpha|00\rangle + \alpha|11\rangle + \beta|10\rangle + \beta|01\rangle) + |1\rangle(\alpha|00\rangle - \alpha|11\rangle + \beta|10\rangle - \beta|01\rangle)\Big]. \tag{19}$$

- Rewriting in terms of Bell basis states:

$$\frac{1}{2}\Big[|\beta_{00}\rangle(\alpha|0\rangle + \beta|1\rangle) + |\beta_{01}\rangle(\alpha|1\rangle + \beta|0\rangle) + |\beta_{10}\rangle(\alpha|0\rangle - \beta|1\rangle) + |\beta_{11}\rangle(\alpha|1\rangle - \beta|0\rangle)\Big]. \tag{20}$$

Step 4: Alice measures and sends classical bits to Bob.

- If Alice measures 00, Bob's qubit is already $|\psi\rangle$.
- If Alice measures 01, Bob needs to apply X (bit-flip gate).
- If Alice measures 10, Bob needs to apply Z (phase-flip gate).
- If Alice measures 11, Bob needs to apply XZ (both gates).

Importance of Quantum Teleportation:

- Enables **quantum communication** over long distances.
- Forms the basis of **quantum repeaters** in quantum networks.
- Used in **quantum cryptography** and **distributed quantum computing**.

2 Variational quantum algorithms (VQAs)

A VQA is a hybrid quantum-classical approach that utilizes the variational principle to solve complex computational problems by optimizing a parameterized quantum circuit. The variational principle, widely used in physics and optimization, states that for a given system, the best approximation of its desired solution can be found by iteratively tuning parameters to minimize (or maximize) a cost function. In VQAs, a quantum processor prepares a trial quantum state with adjustable parameters, while a classical optimizer updates these parameters based on measurements from the quantum system.

2.1 Ansatzes

In VQA, an ansatz is a **parameterized quantum circuit** used to approximate the solution to a given problem. The ansatz prepares a quantum state $|\psi(\theta)\rangle$ that depends on a set of tunable parameters θ, which are optimized to minimize (or maximize) a cost function.

2.1.1 Key characteristics of an ansatz

- Parameterized structure
  - The ansatz is built using quantum gates that depend on parameters θ, such as rotation gates $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$.
  - The parameters θ are optimized iteratively using classical optimization methods.
- Expressibility vs. trainability trade-off
  - A good ansatz must be expressive enough to represent the optimal quantum state.
  - However, it should also be trainable, meaning it avoids issues like barren plateaus (where gradients vanish, making optimization difficult).
- Application-specific design
  - Different problems require different ansatze. For example, in the **Variational Quantum Eigensolver (VQE)**, the ansatz must be able to approximate the ground state of a given Hamiltonian. In **Quantum Approximate Optimization Algorithm (QAOA)**, the ansatz structure is problem-inspired.

2.1.2 Types of ansatz in VQAs

- Hardware-efficient ansatz (HEA)
  - Designed to be compatible with the native gate set of a quantum device.
  - Uses layers of **parameterized single-qubit gates** (e.g., $R_y(\theta)$) and **entangling gates** (e.g., CNOT gates).
  - Example:

$$U(\theta) = \prod_{l=1}^{d} \left[\prod_{i=1}^{N} R_y^i(\theta_{l,i}) \cdot \prod_{\langle i,j \rangle} \mathrm{CNOT}_{i,j} \right]. \tag{21}$$

  - **Advantage**: Easy to implement on quantum hardware.

- **Disadvantage**: May not efficiently represent the target state, leading to slow convergence.

- Problem-inspired ansatz
  - Derived from the structure of the problem being solved.
  - Often used in **VQE**, where the ansatz reflects the physical system's Hamiltonian.
  - Example: Unitary Coupled Cluster (UCC) ansatz for quantum chemistry simulations.

$$|\psi(\theta)\rangle = e^{\sum_i \theta_i (T_i - T_i^\dagger)} |\psi_0\rangle \tag{22}$$

  - **Advantage**: Physically motivated, leading to more efficient optimization.
  - **Disadvantage**: Often requires complex gate sequences that may be hard to implement on near-term devices.

- Layered ansatz (Alternating ansatz)
  - Used in **QAOA**.
  - Alternates between problem Hamiltonian evolution and mixing Hamiltonian.

$$|\psi(\theta)\rangle = e^{-i\beta H_M} e^{-i\gamma H_C} \cdots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C} |\psi_0\rangle, \tag{23}$$

    where H_C encodes the cost function and H_M is a mixing Hamiltonian.
  - **Advantage**: Well-structured, with a clear connection to classical optimization problems.
  - **Disadvantage**: Limited expressibility for certain problems.

- Hardware-aware ansatz
  - Designed specifically to reduce circuit depth and minimize errors for a given quantum hardware topology.
  - **Example**: Ansatz optimized for superconducting qubits where connectivity is limited.

2.1.3 Choosing the Right Ansatz

The choice of ansatz depends on:

- **Expressibility**: Can the ansatz represent the solution?
- **Trainability**: Does it avoid barren plateaus?
- **Hardware Constraints**: Can it be implemented efficiently?

2.2 Cost functions

In the context of a VQA, the cost function (often referred to as the loss function or objective function) is a classical function that measures how well a quantum circuit or quantum state aligns with the desired outcome of the quantum algorithm. The goal of the optimization in a VQA is to minimize (or maximize) this cost function by adjusting the parameters (typically angles or coefficients) of the quantum gates.

2.2.1 General form of a cost function in VQA

In many cases, the cost function $C(\theta)$ is defined as an expectation value of

$$C(\theta) = \langle \psi(\theta) | \hat{O} | \psi(\theta) \rangle, \tag{24}$$

where:

- θ is the vector of parameters (such as gate rotation angles) that define the quantum state or quantum circuit.
- $|\psi(\theta)\rangle$ is the quantum state prepared by the quantum circuit with parameters θ.
- $\hat{O}$ is an observable, which is a Hermitian operator that represents the quantity we want to measure or optimize.

2.2.2 Types of cost functions

- Expectation value of an observable.
  - The most common cost function is the expectation value of an observable, typically represented as:

    $$C(\theta) = \langle \psi(\theta) | \hat{H} | \psi(\theta) \rangle, \tag{25}$$

    where $\hat{H}$ is a Hamiltonian (or some other observable), and $|\psi(\theta)\rangle$ is the quantum state parameterized by θ.
  - In variational quantum algorithms like VQE, $\hat{H}$ is usually the Hamiltonian of a quantum system (e.g., the Hamiltonian of a molecule).
  - In QAOA, the observable $\hat{H}$ could be related to the cost function in an optimization problem.
- Measurement-based cost functions
  - In some cases, the cost function is based on measurements of certain quantum observables, and it might involve multiple measurements of the quantum state to estimate the expectation value.
- Distance or fidelity-based cost functions
  - If the goal is to find a quantum state that is as close as possible to some target state, the cost function could be based on the **fidelity** or **distance** between the current quantum state and the target state:

    $$C(\theta) = 1 - F(\psi(\theta), \psi_{\text{target}}), \tag{26}$$

    where $F(\psi(\theta), \psi_{\text{target}})$ is the fidelity between the state $\psi(\theta)$ and the target state ψ_{target}.
- Custom cost functions: In specific applications, the cost function can be tailored to the problem at hand. For example:
  - In quantum machine learning, the cost function might be related to the accuracy of a classifier trained with quantum states.
  - In quantum metrology, the cost function might involve the precision of parameter estimation.

2.2.3 Optimization of the cost function

Once the cost function is defined, it is minimized (or maximized) using classical optimization algorithms, such as gradient descent, Nelder-Mead, or more advanced techniques like Adam or COBYLA. The parameters θ of the quantum circuit are adjusted iteratively to reduce the cost function.

The optimization process generally involves:

- Preparing the quantum state using the current parameter values θ.
- Measuring the observable $\hat{O}$ or computing the expectation value.
- Evaluating the cost function $C(\theta)$.
- Using classical optimization to update the parameters θ.

2.2.4 Example: Cost function in VQE

In VQE, the goal is to find the ground state energy of a quantum system. The cost function is the expectation value of the Hamiltonian $\hat{H}$, which represents the energy of the system:

$$C(\theta) = \langle \psi(\theta) | \hat{H} | \psi(\theta) \rangle. \tag{27}$$

The optimization process aims to minimize $C(\theta)$, which corresponds to finding the lowest energy state of the system.

2.3 Gradient

2.3.1 Finite difference method

A simple method is the **finite difference method**, where the gradient is approximated numerically as:

$$\frac{\partial C}{\partial \theta_i} \approx \frac{C(\theta_i + \varepsilon) - C(\theta_i - \varepsilon)}{2\varepsilon}, \tag{28}$$

where ε is a small step size. However, this approach introduces numerical errors and requires choosing an appropriate ε.

2.3.2 Parameter-shift rule

Due to the discrete nature of quantum measurements, computing gradients in quantum circuits differs from classical automatic differentiation. One of the most common methods for computing gradients in VQAs is the **parameter shift rule**, which allows exact gradient computation. For a general function $f(x)$, the parameter-shift rule is given by

$$f'(x) = r\Big(f(x+s) - f(x-s)\Big), \tag{29}$$

where s is a shift parameter and r is a scaling factor.

- For a linear function

$$f(x) = ax + b, \tag{30}$$

we have

$$\begin{aligned} f'(x) &= \frac{f(x+s)-f(x-s)}{2s} \\ &= \frac{[a(x+s)+b]-[a(x-s)+b]}{2s} \\ &= a. \end{aligned} \tag{31}$$

- For a quadratic function

$$f(x) = ax^2 + bx + c, \tag{32}$$

we have

$$\begin{aligned} f'(x) &= \frac{f(x+s)-f(x-s)}{2s} \\ &= \frac{[a(x+s)^2+b(x+s)+c]-[a(x-s)^2+b(x-s)+c]}{2s} \\ &= \frac{\left[a(x^2+2sx+s^2)+bx+bs+c\right]-\left[a(x^2-2sx+s^2)+bx-bs+c\right]}{2s} \\ &= \frac{ax^2+2asx+as^2+bx+bs+c-ax^2+2asx-as^2-bx+bs-c}{2s} \\ &= \frac{4asx+2bs}{2s} \\ &= 2ax+b. \end{aligned} \tag{33}$$

- For a higher-order function such as a cubic function:

$$f(x) = ax^3 + bx^2 + cx + d, \tag{34}$$

we can apply the parameter-shift rule by decomposing it into quadratic and linear components.

- Now, let us give a detailed explanation and proof of the parameter-shift rule in VQAs, specifically for gates of the form

$$U(\theta) = e^{-i\frac{\theta}{2}G}, \tag{35}$$

where G is a generator with eigenvalues ± 1, such as in rotation gates $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$. In VQAs, the cost function $C(\theta)$ is typically given as the expectation value of an observable O with respect to a variational quantum state

$$C(\theta) = \langle \psi(\theta)|O|\psi(\theta)\rangle. \tag{36}$$

The variational state is prepared using a parametrized quantum circuit

$$|\psi(\theta)\rangle = U(\theta)|\psi\rangle, \tag{37}$$

where $|\psi\rangle$ is the intial quantum state, usually choose as $|00\cdots0\rangle$.

To optimize the cost function, we need to compute gradients with respect to the parameters θ_i. To compute the gradient $\partial C/\partial\theta_i$, we express

$$\frac{\partial C}{\partial \theta_i} = \frac{\partial}{\partial \theta_i}\langle\psi(\theta)|O|\psi(\theta)\rangle. \tag{38}$$

Since the parameterized unitary has the form $U(\theta_i) = e^{-i\theta_i G}$, we consider an infinitesimal shift in θ_i

$$\frac{\partial}{\partial \theta_i}U(\theta_i) = -iGU(\theta_i). \tag{39}$$

Using this, the derivative of the cost function can be rewritten as

$$\frac{\partial C}{\partial \theta_i} = -\frac{i}{2}\langle\psi|U^\dagger[G,O]U|\psi\rangle. \tag{40}$$

Since G is Hermitian, i.e, ($G^\dagger = G$) and satisfies $G^2 = I$. The commutator $[G,O]$ gives

$$[G,O] = -i\Big[U^\dagger\Big(\frac{\pi}{2}\Big)OU\Big(\frac{\pi}{2}\Big) - U^\dagger\Big(\frac{\pi}{2}\Big)OU\Big(-\frac{\pi}{2}\Big)\Big]. \tag{41}$$

Substituting it to Eq. (40), we get

$$\frac{\partial C}{\partial \theta_i} = \frac{C(\theta_i+\pi/2) - C(\theta_i-\pi/2)}{2}. \tag{42}$$

This is the simple form of the **parameter-shit rule**. In general, we can express the derivative using shifted parameter values as

$$\frac{\partial C}{\partial \theta_i} = \frac{C(\theta_i+s) - C(\theta_i-s)}{2\sin(s)}. \tag{43}$$

This method only requires evaluating the cost function at two shifted parameter values, making it **efficient and hardware-friendly**.

2.3.3 Analytical Gradients via Quantum Differentiable Programming

For certain ansatze, analytical gradient methods can be used, such as computing gradients through hybrid quantum-classical backpropagation (e.g., in PennyLane or TensorFlow Quantum).

2.4 Optimizes

Once gradients are computed, classical optimization algorithms update the parameters θ. Common gradient-based optimizers include:

2.4.1 Standard Gradient Descent (SGD)

- Updates parameters using the rule:

$$\theta_{t+1} = \theta_t - \eta\nabla C(\theta_t), \tag{44}$$

  where η is the learning rate.

- Simple but can struggle with noisy gradients.

2.4.2 Adam Optimizer

- An adaptive gradient method that uses momentum to accelerate convergence:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t} + \varepsilon} m_t, \tag{45}$$

  where m_t and v_t are first and second moment estimates of the gradient.

- Commonly used due to stability and fast convergence.

2.4.3 Natural Gradient Descent (NGD)

- Incorporates the quantum Fisher Information Matrix (QFIM) to improve optimization:

$$\theta_{t+1} = \theta_t - \eta F^{-1} \nabla C(\theta_t), \tag{46}$$

  where F is the QFIM.

- Helps navigate the curvature of the parameter space efficiently.

3 Specific VQAs

3.1 Quantum Approximate Optimization Algorithm (QAOA)

QAOA is a hybrid quantum-classical algorithm designed to solve combinatorial optimization problems by using a sequence of quantum operations that alternate between two Hamiltonians. This section provides a detailed breakdown of QAOA's key components and its application.

3.1.1 Algorithm Overview

QAOA operates by iteratively applying two Hamiltonians:

- Problem Hamiltonian (H_P): Encodes the optimization problem as an energy function, where lower energy corresponds to better solutions.

- Mixing Hamiltonian (H_M): Facilitates quantum transitions between different solutions, preventing the algorithm from getting stuck in local minima.

The quantum system evolves through a series of alternating unitary transformations parameterized by angles γ and β. These parameters are optimized using classical algorithms to minimize the cost function.

3.1.2 Mathematical Formulation

- Initialization: The quantum register is initialized in an equal superposition state:

$$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

  This is achieved by applying Hadamard gates to all qubits.

- Alternating Unitary Evolution: The quantum state is evolved by applying alternating unitary transformations:

$$|\psi_p(\gamma,\beta)\rangle = \prod_{j=1}^{p} U(H_M,\beta_j)U(H_P,\gamma_j)|\psi_0\rangle$$

The unitary operators are defined as:

$$U(H_P,\gamma) = e^{-i\gamma H_P}$$

$$U(H_M,\beta) = e^{-i\beta H_M}$$

- Measurement and Optimization: The final quantum state is measured in the computational basis to obtain a candidate solution. A classical optimizer (e.g., gradient-based or Bayesian optimization) updates γ and β to improve performance. The process is repeated until convergence.

3.1.3 Application to the MAX-CUT Problem

One of the most common applications of QAOA is the MAX-CUT problem, which involves partitioning a graph into two sets while maximizing the number of edges between them.

- Problem Hamiltonian:

$$H_P = \sum_{(i,j)\in E} \frac{1}{2}(1 - Z_i Z_j)$$

where Z_i and Z_j are Pauli-Z operators acting on qubits corresponding to vertices i and j.

- Mixing Hamiltonian::

$$H_M = \sum_i X_i$$

where X_i are Pauli-X operators.

After running QAOA, the measured bitstring provides an approximate solution for MAX-CUT.

3.1.4 Key Properties and Advantages

Feature	**Description**
Hybrid approach	Uses both quantum circuits and classical optimization.
Tunability	Performance improves with increasing p.
Quantum advantage	For some problems, QAOA outperforms classical heuristics.
Noise resilience	Works well even on noisy quantum hardware.

Table 3: Key properties of QAOA

3.2 Variational Quantum Eigensolver (VQE)

The Variational Quantum Eigensolver (VQE) is a hybrid quantum-classical algorithm designed to estimate the ground-state energy of a given Hamiltonian, particularly in quantum chemistry and materials science applications. The algorithm leverages the variational principle, which states that for a trial wavefunction $|\psi(\theta)\rangle$ parameterized by a set of classical parameters θ, the expectation value of the Hamiltonian provides an upper bound on the true ground-state energy:

$$E(\theta) = \langle \psi(\theta)|H|\psi(\theta)\rangle \geq E_0, \tag{47}$$

where E_0 is the true ground-state energy of the system.

3.2.1 Algorithm Overview

VQE follows an iterative optimization approach involving both quantum and classical computation:

1. **Ansatz Preparation:** A parameterized quantum circuit $U(\theta)$ is prepared on a quantum processor to generate a trial wavefunction $|\psi(\theta)\rangle = U(\theta)|0\rangle$.
2. **Hamiltonian Expectation Evaluation:** The expectation value $E(\theta)$ is computed by measuring the prepared quantum state across multiple bases.
3. **Classical Optimization:** A classical optimizer (e.g., gradient-based methods, COBYLA, or Nelder-Mead) adjusts θ to minimize $E(\theta)$.
4. **Convergence Check:** The process repeats until the optimization converges to a value close to E_0.

3.2.2 Choice of Ansatz

The effectiveness of VQE depends significantly on the choice of the ansatz. Common ansätze include:

- **Hardware-Efficient Ansatz:** Uses native gate operations to minimize circuit depth.
- **Unitary Coupled Cluster (UCC) Ansatz:** Inspired by classical quantum chemistry methods, particularly useful for molecular simulations.
- **Problem-Specific Ansatz:** Tailored circuits designed based on problem symmetry and structure.

3.2.3 Advantages and Challenges

Advantages:

- Suitable for near-term noisy quantum devices (NISQ era).
- Requires fewer quantum resources compared to full quantum eigensolvers.
- Adaptable to various Hamiltonians and quantum systems.

Challenges:

- Requires many quantum measurements for accurate expectation values.
- Classical optimization may suffer from barren plateaus, slowing convergence.

- The choice of ansatz significantly affects performance and accuracy.

VQE has been successfully applied to small molecular systems such as hydrogen (H_2) and lithium hydride (LiH), demonstrating its potential for quantum chemistry applications. Further improvements in ansatz design and classical optimization techniques are critical for scaling VQE to larger and more complex molecules.

3.3 Quantum Compilation Algorithms

4 Applications of VQAs

Variational Quantum Algorithms (VQAs) are designed to harness quantum computing's power in solving complex problems by combining quantum circuits with classical optimization methods. VQAs are especially valuable for near-term quantum devices with noise and limited qubits. This section explores several important application areas of VQAs, including Quantum Chemistry, Optimization Problems, Quantum Machine Learning, and more.

4.1 Quantum Chemistry

Quantum chemistry is one of the most promising fields for quantum computing, as it involves complex calculations related to the electronic structure of molecules. The VQE has emerged as a key algorithm for solving molecular simulations and determining the ground state energy of molecules. The algorithm works by variationally optimizing the parameters of a quantum circuit, which encodes the molecular wavefunction. The goal is to minimize the expectation value of the *Hamiltonian* H, representing the total energy of the molecular system.

The quantum state $|\psi(\theta)\rangle$ is parameterized by a set of variational parameters θ, and the energy expectation value is calculated as:

$$E(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$$

where H represents the Hamiltonian of the molecule, and $|\psi(\theta)\rangle$ is the trial wavefunction prepared by the quantum circuit.

4.1.1 VQE Procedure in Quantum Chemistry

The VQE algorithm proceeds through the following steps to approximate the ground state energy of a molecule:

1. **State Preparation:** A quantum circuit prepares a trial state $|\psi(\theta)\rangle$ based on the current values of the variational parameters θ. This trial state is a quantum approximation of the true molecular wavefunction.

2. **Measurement:** The quantum circuit is measured to obtain an estimate of the energy expectation value $E(\theta)$. The energy is computed by measuring the Hamiltonian H in the trial quantum state.

3. **Optimization:** Classical optimization techniques are then used to adjust the variational parameters θ in the quantum circuit to minimize the energy expectation value $E(\theta)$. This iterative process refines the approximation of the ground state energy.

This hybrid quantum-classical approach allows VQE to be effective on noisy, intermediate-scale quantum computers (NISQ devices) that are available today.

4.1.2 Example: Hydrogen Molecule Simulation

Consider a simple example of simulating the ground state energy of a hydrogen molecule using VQE. The Hamiltonian for a hydrogen molecule is typically represented as:

$$H = -\frac{1}{2}\nabla^2 + V_{\text{int}}$$

where ∇^2 is the Laplacian operator (representing kinetic energy), and V_{int} is the interaction potential between electrons and nuclei.

In VQE, the quantum circuit would prepare a trial quantum state $|\psi(\theta)\rangle$ that approximates the electronic structure of the hydrogen molecule. By optimizing the parameters θ, the algorithm minimizes the energy expectation value $E(\theta)$, which gives the ground state energy of the molecule.

As the optimization proceeds, the quantum circuit gets closer to the true wavefunction, and the energy converges to a value that approximates the real energy of the hydrogen molecule.

4.2 Optimization Problems

Optimization problems, especially combinatorial ones, are often NP-hard and difficult for classical algorithms to solve efficiently. Variational Quantum Algorithms (VQAs), in particular the *Quantum Approximate Optimization Algorithm (QAOA)*, have been proposed as a way to approximate solutions to such problems.

4.2.1 Quantum Approximate Optimization Algorithm (QAOA)

The QAOA algorithm works by preparing a quantum state that encodes the solution space of an optimization problem. The quantum circuit alternates between applying a problem-specific Hamiltonian H_{P} and a mixer Hamiltonian H_{M}. The quantum state is parameterized, and the parameters are optimized to minimize the expected value of the problem Hamiltonian.

The quantum state for QAOA is given by:

$$|\psi(\gamma,\beta)\rangle = \left(e^{-i\gamma H_{\text{P}}} e^{-i\beta H_{\text{M}}}\right)^p |+\rangle$$

where p is the number of layers, and γ and β are the parameters that control the evolution in the Hamiltonian.

4.2.2 Example: Max-Cut Problem

In the Max-Cut problem, the objective is to partition the nodes of a graph into two sets in such a way as to maximize the number of edges between them. The problem Hamiltonian for Max-Cut is:

$$H_{\text{P}} = \sum_{\langle i,j\rangle} \left(\frac{1 - Z_i Z_j}{2}\right)$$

where Z_i and Z_j are the Pauli-Z operators acting on qubits i and j, respectively, and $\langle i,j\rangle$ represents all pairs of connected nodes. By optimizing the parameters γ and β, QAOA approximates the Max-Cut solution, providing a potential quantum advantage over classical heuristics.

4.3 Quantum Machine Learning

Quantum Machine Learning (QML) is an emerging field that explores the potential of quantum computing to enhance classical machine learning models. One of the key components of QML is the *Quantum Neural Networks (QNNs)*, which utilize quantum circuits to improve performance in various machine learning tasks, such as classification, regression, and clustering.

4.3.1 Quantum Neural Networks (QNNs)

Quantum Neural Networks (QNNs) employ quantum circuits to encode data and perform machine learning tasks. Similar to classical neural networks, the parameters of the quantum circuit are optimized to minimize a loss function. However, the data is represented as quantum states, and the optimization process can harness quantum parallelism to potentially outperform classical models.

The quantum state for a quantum classifier is given by:

$$|\psi(\theta)\rangle = U(\theta)|x\rangle$$

where $U(\theta)$ is the unitary operator that applies a series of quantum gates to the input data, represented as the quantum state $|x\rangle$. The goal of the QNN is to optimize the parameters θ such that the classification error is minimized.

4.3.2 Quantum Support Vector Machines (QSVM)

Quantum Support Vector Machines (QSVM) are a quantum version of classical Support Vector Machines (SVMs). QSVMs use quantum circuits to map classical data into a high-dimensional quantum space, allowing the quantum computer to exploit quantum properties like superposition and entanglement. In QSVM, the classical kernel function is replaced by a quantum kernel, which is defined as:

$$K(x,y) = |\langle \psi_x | \psi_y \rangle|^2$$

where $|\psi_x\rangle$ and $|\psi_y\rangle$ are quantum states corresponding to the classical data points x and y, and the inner product $\langle \psi_x | \psi_y \rangle$ measures the similarity between the two data points in the quantum space. This quantum kernel can potentially offer an advantage in terms of computational complexity for high-dimensional data.

4.4 Quantum Cryptography

Quantum cryptography leverages the principles of quantum mechanics to achieve secure communication. Variational Quantum Algorithms (VQAs) can be employed to enhance cryptographic protocols, particularly in areas such as the generation of cryptographic keys and secure authentication systems.

4.4.1 Quantum Key Distribution (QKD)

Quantum Key Distribution (QKD) protocols, such as the *BB84* protocol, enable two parties to securely share a secret key over a public channel. The security of QKD protocols arises from the fact that any attempt to eavesdrop on the quantum channel introduces detectable errors due to the no-cloning theorem and quantum measurement effects. This makes it impossible for an eavesdropper to obtain any information without disturbing the quantum state in a detectable manner.

VQAs can be used to optimize certain aspects of QKD, such as the selection of quantum states for encoding information, in order to maximize both security and efficiency. This optimization can improve the overall performance of QKD systems, especially in noisy or resource-constrained environments.

4.4.2 Quantum Secret Sharing (QSS)

Quantum Secret Sharing (QSS) is a cryptographic technique used to divide a secret into multiple parts, which are then distributed among several parties. By employing VQAs, the division of the secret can be optimized to ensure that no single party has access to the full secret. This enhances the security of the protocol, ensuring that even if some parties are compromised, the secret cannot be reconstructed without the cooperation of multiple participants.

4.5 Quantum Simulation

Quantum simulation refers to the use of quantum computers to simulate quantum systems that are too complex for classical computers to handle efficiently. Variational Quantum Algorithms (VQAs) can be applied to optimize the simulation of quantum systems, enhancing the efficiency of algorithms designed to simulate many-body systems, quantum field theories, and other complex physical systems.

4.5.1 Quantum Many-Body Simulation

Simulating quantum systems with many interacting particles is a challenge for classical computers due to the exponential scaling of the problem size. VQAs, such as the Variational Quantum Simulation (VQS) algorithm, offer a promising approach for simulating these systems by optimizing the parameters of a quantum state to approximate the true ground state of the system.

4.5.2 Example: Ising Model

The Ising model is a widely studied mathematical model in statistical mechanics, describing the interaction between spins in a lattice, which is used to simulate magnetism. The Hamiltonian for the Ising model can be expressed as:

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z$$

where J represents the interaction strength between spins, and σ_i^z is the Pauli-Z operator acting on the i-th spin. By applying VQS, the quantum state can be optimized to approximate the ground state energy of the Ising model, providing insights into the system's behavior.

4.6 Quantum Finance

Quantum algorithms have significant applications in finance, particularly in areas such as portfolio optimization, risk analysis, and option pricing. Variational Quantum Algorithms (VQAs) are especially effective in solving optimization problems in finance, where the solution space is large and complex.

4.6.1 Portfolio Optimization

In portfolio optimization, the objective is to choose a combination of investments that minimizes risk while maximizing returns. VQAs, particularly the Quantum Approximate Optimization Algorithm (QAOA), can be applied to optimize portfolios by modeling the problem as a combinatorial optimization task.

4.6.2 Example: Mean-Variance Optimization

The mean-variance optimization problem aims to minimize the variance of returns for a given expected return. The optimization problem can be formulated as:

$$\min_{\mathbf{x}} \mathbf{x}^T \Sigma \mathbf{x}$$

where $\mathbf{x}$ is the vector of portfolio weights, and Σ is the covariance matrix of returns. QAOA can be employed to solve this problem and find the optimal portfolio configuration by minimizing the objective function.

5 Challenges in VQAs

Variational Quantum Algorithms (VQAs) are a promising approach for leveraging quantum computing in various optimization and simulation tasks. However, there are several challenges that hinder their practical implementation and efficient execution on real quantum hardware. These challenges include barren plateaus, noise and decoherence, and classical bottlenecks. Below, we provide a detailed overview of each challenge.

5.1 Barren Plateaus

Barren plateaus refer to regions in the parameter space of the quantum circuit where the gradient of the cost function with respect to the variational parameters becomes exceedingly small. This phenomenon makes the optimization process difficult, especially for deep quantum circuits. In these regions, the optimization algorithm struggles to make progress because the gradient is insufficient to guide the search for an optimal solution.

Mathematically, barren plateaus are characterized by the gradient of the cost function $C(\theta)$ becoming very small for large p, where p denotes the depth of the circuit. Specifically, for certain types of circuits, the variance of the gradient may decrease exponentially with the number of qubits N, which results in an exponentially vanishing gradient:

$$\text{Var}\left(\frac{\partial C(\theta)}{\partial \theta_k}\right) \sim O\left(\frac{1}{2^N}\right)$$

This leads to a situation where the optimization becomes highly inefficient, as the optimization algorithm cannot effectively navigate the parameter space.

5.2 Noise and Decoherence

Quantum computers are highly sensitive to noise and decoherence, which can severely affect the performance of VQAs on real hardware. Errors in quantum gates, readout, and other quantum operations can lead to incorrect results. These errors are often amplified as the number of qubits and the circuit depth increase, making it difficult to maintain accuracy.

Noise can be categorized into two main types: bit-flip errors and phase-flip errors. These errors cause qubits to deviate from their intended states, leading to incorrect measurements. Decoherence, on the other hand, arises when quantum systems lose their quantum properties due to interaction with the environment, causing entanglement to degrade and the system to behave classically.

For example, the effect of noise can be represented by a quantum channel $\mathscr{E}$ acting on the quantum state $|\psi\rangle$, which transforms the state as:

$$|\psi\rangle \rightarrow \mathscr{E}(|\psi\rangle)$$

This transformation introduces imperfections in the quantum state and leads to errors in the optimization process. Strategies such as error correction codes, noise mitigation techniques, and hardware improvements are being explored to address these issues.

5.3 Classical Bottlenecks

Although the quantum part of VQAs shows potential for exponential speedups, the optimization process itself is still heavily reliant on classical optimization methods. Classical optimizers, such as gradient descent or more advanced algorithms like the Nelder-Mead method, are used to tune the parameters of the quantum circuit. However, these classical optimizers can face significant challenges when dealing with high-dimensional, noisy, or complex landscapes.

In particular, the classical optimization becomes more difficult as the number of variational parameters increases. The optimization landscape in high dimensions may have many local minima and saddle points, which can trap classical algorithms in suboptimal solutions. This is especially problematic for high-dimensional VQA applications, where classical optimization algorithms are not guaranteed to find the global optimum.

Mathematically, the difficulty arises when the number of parameters d increases, and the objective function becomes highly non-convex. The classical optimization algorithm may then become stuck in local minima:

$$\min_{\theta} C(\theta) \quad \text{where} \quad C(\theta) \quad \text{is highly non-convex.}$$

Efforts to improve classical optimization techniques, such as adaptive algorithms and hybrid quantum-classical methods, are essential to address this bottleneck.

6 Further Reading

References

[1] Farhi, E., Goldstone, J., & Gutmann, S. (2014). A Quantum Approximate Optimization Algorithm. [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).

[2] Peruzzo, A., et al. (2014). A variational eigenvalue solver on a quantum processor. Nature Communications, 5, 4213. [DOI: 10.1038/ncomms5213](https://doi.org/10.1038/ncomms5213).

[3] McClean, J. R., et al. (2018). Barren plateaus in quantum neural network training. [arXiv:1803.11173](https://arxiv.org/abs/1803.11173).

[4] Farhi, E., et al. (2016). Quantum approximate optimization algorithm and variational quantum eigensolver. [arXiv:1602.07674](https://arxiv.org/abs/1602.07674).

[5] Benedetti, M., et al. (2019). Quantum-assisted learning of hardware-efficient variational quantum circuits. Quantum Science and Technology, 4(4), 045003. [DOI: 10.1088/2058-9565/ab4d32](https://doi.org/10.1088/2058-9565/ab4d32).

[6] Wang, Z., et al. (2020). Noise resilient variational quantum algorithms for optimization. Quantum, 4, 271. [DOI: 10.22331/q-2020-12-01-271](https://doi.org/10.22331/q-2020-12-01-271).

[7] Kiani, M. D., et al. (2020). Variational quantum algorithms for optimization. Journal of Physics A: Mathematical and Theoretical, 53(42), 423001. [DOI: 10.1088/1751-8121/abafad](https://doi.org/10.1088/1751-8121/abafad).

[8] Schuld, M., & Petruccione, F. (2018). Supervised Learning with Quantum Computers. Springer Nature. [DOI: 10.1007/978-3-030-14912-7](https://doi.org/10.1007/978-3-030-14912-7).

[9] Helsen, J., et al. (2020). Optimizing variational quantum algorithms: A comparison of classical optimizers. Quantum, 4, 318. [DOI: 10.22331/q-2020-12-21-318](https://doi.org/10.22331/q-2020-12-21-318).

[10] Luo, X., et al. (2020). Variational Quantum Algorithms: A Survey of Recent Developments and Future Directions. [arXiv:2003.06503](https://arxiv.org/abs/2003.06503).