Variational Quantum Algorithms

March 4, 2025 11:35 AM

1. Introduction to Variational Quantum Algorithms (VQAs)

Why Quantum Computing?

Quantum computing has exponential speedup over classical computing, this comparision being coined, "quantum advantage", researchers have tried to build the necessary hardware to run quantum software for over a decade

The First Quantum Computer & its Limitations

-> In 2016, access to the first cloud-based quantum computer became available, but noise (error) & qubit limitations prevented serious implementations of quantum algorithms (unreliable for deep quantum circuits). These computers are called Noisy Intermediate-Scale Quantum (NISQ) computers.

-> Current scale devices range in size from 50 to 100 qubits, which allows us to achieve, "quantum supremacy": outperforming even the best classical supercomputer for certain contrived mathematical tasks.

Q: So how do we make use of today's NISQ devices to achieve quantum advantage?

Accounting for qubit limitations & errors that limit quantum circuit depth.

↳ An integer # that counts the maximum length in the circuit b/wn the input & output. The length is usually defined in terms of layers of gates acting in parallel.

The Solution: VQAs

Variational Quantum Algorithms (VQAs) are the leading strategy used to obtain quantum advantage using current NISQ devices. Which account for the constraints requiring an optimization or learning based approach. Where VQAs are arguably the quantum analogue of machine learning methods (e.g. neural networks)

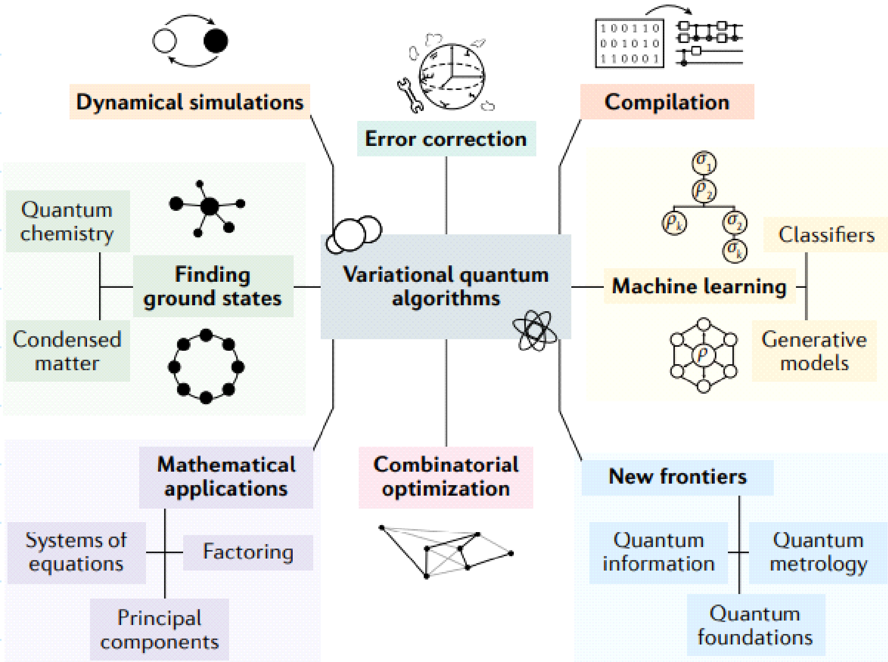
quantum equivalent

VQAs combine the use of quantum computing w/ classical optimization using parameterized quantum circuits to be run on a quantum computer, & then outsourcing the parameter optimization to a classical optimizer.

-> This approach gives advantage by making quantum circuit depth shallow, which in turn mitigates noise, which is useful for Noisy Intermediate-Scale Quantum (NISQ) computers.

noise, which is useful for Noisy Intermediate-Scale Quantum (NISQ) computers.

Variational Quantum Algorithms are already used for a plethora of applications. As seen here:



One of the main advantages of VQAs is that they provide a general framework to solve variety of problems.

Key points

- Variational quantum algorithms (VQAs) are the leading proposal for achieving quantum advantage using near-term quantum computers.
- VQAs have been developed for a wide range of applications, including finding ground states of molecules, simulating dynamics of quantum systems and solving linear systems of equations.
- VQAs share a common structure, where a task is encoded into a parameterized cost function that is evaluated using a quantum computer, and a classical optimizer trains the parameters in the VQA.
- The adaptive nature of VQAs is well suited to handle the constraints of near-term quantum computers.
- Trainability, accuracy and efficiency are three challenges that arise when applying VQAs to large-scale applications, and strategies are currently being developed to address these challenges.

2. How Variational Quantum Algorithms Work

The Basic Building Blocks of VQAs,

→ Step 1: define a cost (or loss) fct^n C

→ Step 2: Include an ansatz → a quantum operation depending on a set of continuous or discrete parameters (θ) that can be optimized.

→ Step 3: Train the ansatz using a quantum-classical optimization continuous or discrete parameters (θ) that can be optimized.

→ Step 3: Train the ansatz using a quantum-classical optimization loop to solve: $\boxed{\theta^* = \arg\min C(\theta)}$ The optimizer

→ value of θ that minimizes the cost fctⁿ

VQAs use a quantum computer to estimate the cost fctⁿ, $C(\theta)$, (or its gradient) while leveraging classical optimizers to train the parameters, θ.

These parameters, θ, are then mapped to real #s by the cost fctⁿ, $C(\theta)$.

The cost defines a hypersurface, called the cost landscape, such that the optimizer navigates the landscape to find the global minima. θ^*

The cost is generally expressed as:

$$\boxed{C(\theta) = f(\{\rho_k\}, \{O_k\}, U(\theta))}$$

where,
- f - some fctⁿ
- $U(\theta)$ - a parametrized Unitary
- ρ_k - the input states from a training set
- O_k - the set of observables

Where, given indices k, it's often useful to define cost as:

$$\boxed{C(\theta) = \sum_k f_k\left(\mathrm{Tr}\left[O_k U(\theta) \rho_k U^\dagger(\theta)\right]\right)}$$ for some set of fctⁿs $\{f_k\}$

→ Trace of the matrix

→ Parameterized Quantum Circuit (Variational Ansatz)

Criteria the Cost Fctⁿ Should Meet

1. Must be 'faithful' s.t. the minimum of $C(\theta)$ corresponds to the solⁿ of the problem.
2. Must be able to efficiently estimate $C(\theta)$ by performing quantum computing measurements, possibly performing classical post-processing.
3. Cost should not be efficiently computable w/ a classical computer; otherwise no quantum advantage in using VQA
4. Useful to be 'operationally meaningful' s.t. smaller cost values indicate better solⁿ quality.
5. Finally, cost must be 'trainable' → should be possible to efficiently optimize parameters, θ.

So for a given VQA to be implementable in NISQ hardware, The quantum

So, for a given VQA to be implementable in NISQ hardware, The quantum circuits used to estimate $C(\theta)$ <u>must</u> keep the requirements for circuit depth & ancilla qubits (Auxillary qubits used during a quantum computation) small.

→ This is to account for NISQ device gate errors & limited qubit counts & their short decoherence times.

3. Types of Ansatz (Quantum Circuits) Used in VQAs

An ansatz is a structured quantum circuit that, given its form, determines what the parameters $\{\theta\}$ are, hence how they can be trained to minimize the cost.

The structure of an ansatz depends on the task, & in many cases the type of problem is used to determine the type of ansatz used.