Variational Quantum Methods

Jiaqi Leng, Yiling Qiao, Yuxiang Peng

University of Maryland, College Park

Oct. 5, 2021

Overview

Overview

Quantum Chemistry on Google's Quantum Computer

Ecologists battle invasive coconut palms p. 1046
Digital health transformation in India p. 1050
Uncovering the origin of Earth's water pp. 1058 & 1110
Science
$15 28 AUGUST 2020 sciencemag.org
AAAS
CHEMISTRY BY QUBIT
Quantum chip simulates molecular isomerization
pp. 1054 & 1084

Google AI Quantum and collaborators used the Sycamore quantum processor (based on superconducting qubits) to simulate the energy of the diazene molecule (H_2N_2) in various conformations.

The simulations were performed on up to 12 qubits, involving up to 72 two-qubit gates. The result was published on the *Science* journal in August 2020.

Towards Universal Quantum Computer

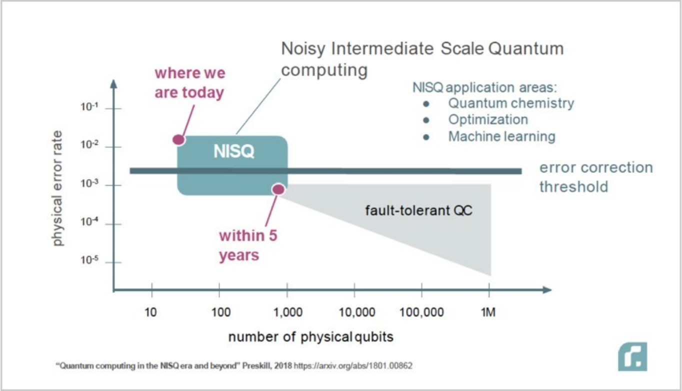


Figure: We are currently in the NISQ (Noisy Intermediate Scale Quantum computing)[Pre18] era: we only have access to quantum computers with 50-100 qubits with noise.[1]

[1] Picture credited to Nabil Laoudji.

Quantum advantage with NISQ devices

John Preskill believes NISQ technology may be able to perform tasks which “surpass the capabilities of today’s classical digital computers”, but “the 100-qubit quantum computer will not change the world right away”.

Quantum advantage with NISQ devices

John Preskill believes NISQ technology may be able to perform tasks which “surpass the capabilities of today’s classical digital computers”, but “the 100-qubit quantum computer will not change the world right away”.

A strategy that makes the best use of NISQ devices must account for:

- Limited numbers of qubits;
- Limited connectivity of qubits;
- Coherent and incoherent errors that limit quantum circuit depth.

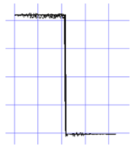
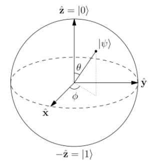
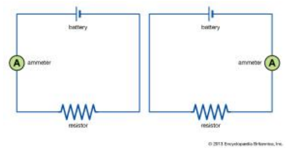
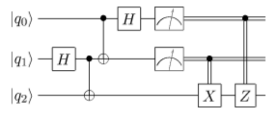
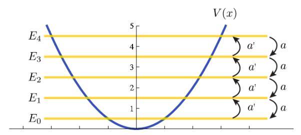
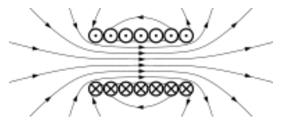
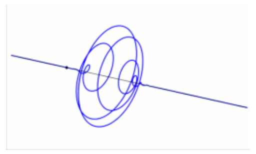
Variational Quantum Algorithms (VQAs): the leading strategy to obtain quantum advantage on NISQ devices.

Basics of Quantum Computing and Quantum Optimal Control

Yuxiang Peng

CMSC838B Project Preliminary

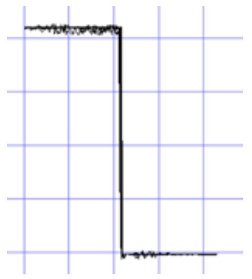
Overview

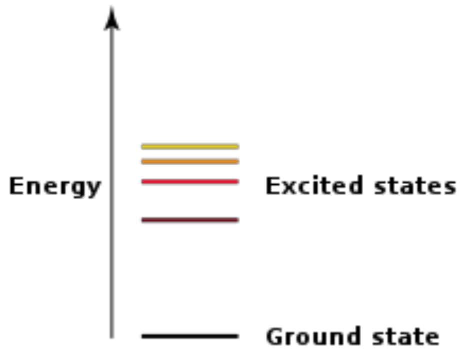
	Classical bits	Quantum bits	Fock state
	 0 or 1	 $\alpha\|0\rangle + \beta\|1\rangle$	$\sum_{j\in\mathbb{N}} c_j\|j\rangle$
Digital	Classical gates and circuits 	Quantum gates and circuits  $\|q_0\rangle$ $\|q_1\rangle$ $\|q_2\rangle$ H X Z	Ladder operators  $V(x)$ E_4 E_3 E_2 E_1 E_0 $a^\dagger$ a
Analog	Evolution of EM fields 	Evolution of waves 	

Advanced topics

From classical bits to qubits

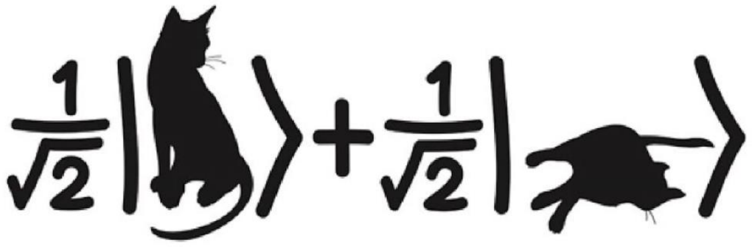
- Classical bit: Represented by voltage
  - High and low voltage represents 0 and 1.
- Quantum bit: Represented by energy levels
  - The two lowest levels are $|0\rangle$ and $|1\rangle$.
- Quantum bit can be “between” $|0\rangle$ and $|1\rangle$.
  - This is called superposition.


High voltage changes to low voltage

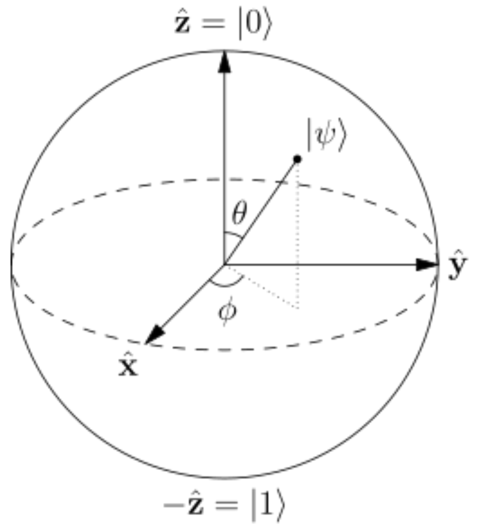


Multiple energy levels of a particle

Superposition



- The cat state: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- If we measure it, it has 50% probability to be 0 and another 50% probability to be 1.
- Bloch sphere:
  - Every point inside it represents a qubit's state.



Qubits

- A qubit can be described by 2-d vector:

$$\alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

- A system with multiple subsystems is constructed by tensor product.

$$A_{2\times 2} \otimes B = \begin{bmatrix} a_{00}B & a_{01}B \\ a_{10}B & a_{11}B \end{bmatrix}$$

- An n-qubit system can be represented by 2^n-dim vector.
- The space of such states is a 2^n-dim Hilbert space.

Entanglement

- Non-locality of quantum states: EPR pair
$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

- Local observation has non-local effect:
  - When you measure one of the two qubits, you “foresee” the measurement of the other one.

From classical gates to quantum gates

- Classical gates:
  - Mapping $\{0,1\}^n \to \{0,1\}^m$

- Example: applying NOT gate
$$\sim 0 = 1, \qquad \sim 1 = 0.$$

- Example: assigning gate
$$0b \to 00, 1b \to 11.$$

- Quantum gates:
  - Unitary transformation $U: |\varphi\rangle \to U|\varphi\rangle$.

- Example: applying X gate
$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle.$$

- Example: applying Hadamard gate
$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$
$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Reversibility and non-cloning

- Quantum gates are unitary (square matrices).
  - Information is preserved.

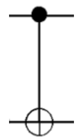
- Fact 1: Every quantum gate/circuit is reversible.
- Fact 2: Operations are linear.
- Fact 3: You cannot copy an arbitrary state.
  - There is no unitary U such that
  $$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle.$$
  - You cannot erase a state as well.

Commonly used quantum gate and circuit

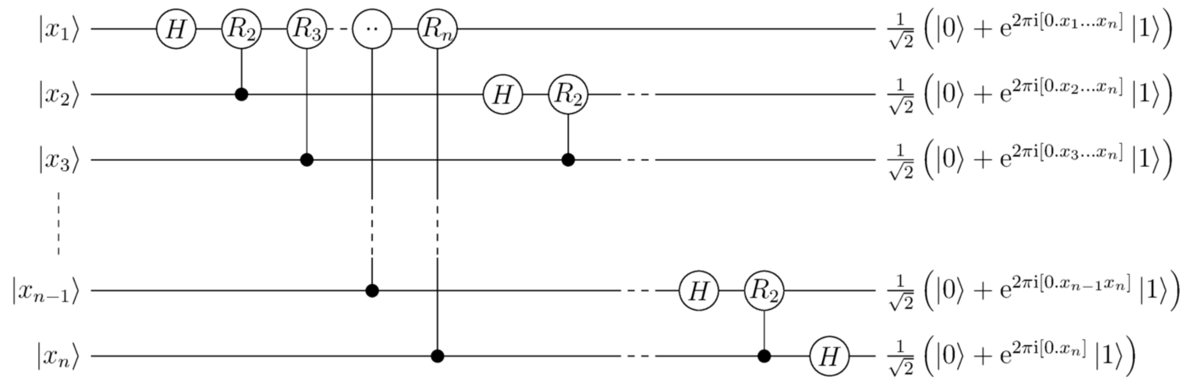
- Example one-qubit gates: Pauli matrices:
  - They are $180°$ rotations along axis in the Bloch sphere.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Example two-qubit gate: Controlled-NOT gate (CNOT).
  - Flips the second qubit if the first qubit is 1.



- Example circuit:


$|x_1\rangle$
$|x_2\rangle$
$|x_3\rangle$
$|x_{n-1}\rangle$
$|x_n\rangle$
H
R_2
R_3
R_n
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_1 \dots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_2 \dots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_3 \dots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_{n-1}x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_n]}|1\rangle\right)$

Second quantization

- Another model that physicists love.
  - Usually, it is used to describe bosonic or fermionic system.

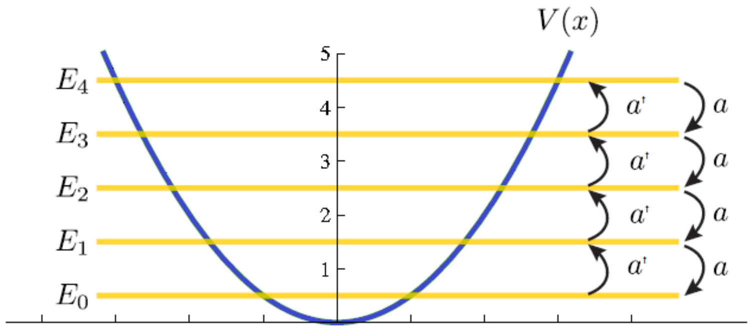
- Quantizing how many particles are in a pool.
  - $|n\rangle$ represents n particles.

- Superposition exists as well.
  - A Fock state is $\sum_{n\geq 0} c_n|n\rangle$ in an infinite-dimensional Hilbert space.

Ladder operators

- Fock states are manipulated by ladder operators.

- Ladder operators create or annihilate particles in the pool.
  - Annihilation operator a lowers the number of particles of a Fock state.
$$a|n\rangle = \sqrt{n}|n-1\rangle.$$
  - As its counter-part, creation annihilation operator $a^\dagger$ increases the number of particles:
$$a^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle$$



Ladder operators in matrix view

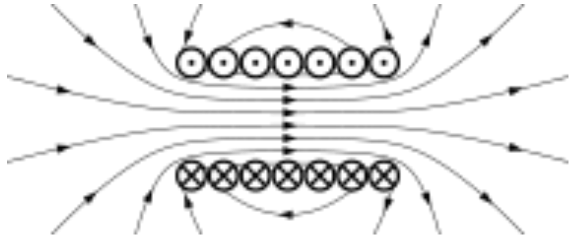
- They can be written as infinite-dimensional matrices:

$$a = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & \sqrt{2} & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & \sqrt{3} & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & \ddots & \vdots & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \sqrt{n} & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Breaking down the gates

—— from digital to analog

- Classical gates obey physical law of EM fields.
  - The Maxwell's equations.



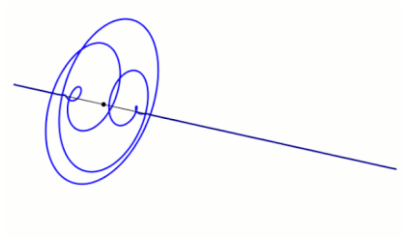
$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \left(\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right)$$

- Quantum gates obey quantum mechanics.
  - The Schrodinger equation.



$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \hat{H} |\psi(t)\rangle$$

Hamiltonian

- Physically, Hamiltonian is the sum of kinetic and potential energy in the system.

- Mathematically, time-dependent Hamiltonian is a mapping from time to a Hermitian matrix.

- Examples: $H_1(t) = X \otimes X + Z \otimes Z,\ \ H_2(t) = \cos(t)\, aa^\dagger$.

- Hamiltonian determines how a quantum system evolves.

Schrödinger equation

- How a state time-evolve under a Hamiltonian.

$$i\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = H(t)|\psi(t)\rangle.$$

  - The change of physical system subject to the “force”.

- If $H(t) \equiv H_0$ which is time independent, the solution is “trivial”:

$$|\psi(t)\rangle = e^{-iH_0 t}|\psi(0)\rangle.$$

- Otherwise, the solution will be “non-trivial”:

$$|\psi(t)\rangle = e^{\int -iH(\tau)\,\mathrm{d}\tau}|\psi(0)\rangle.$$

Parameterizing the evolution

- To have something to differentiate, we need to parameterize the time-dependent Hamiltonian.

- A typical model is:

$$H(t) = H_0 + \sum_{j=1}^{n} u_j(t) H_j \,.$$

  - u_j is a time-dependent complex function.
  - H_j is a time-independent Hermitian matrix.
  - The system evolves for time interval $[0, 1]$.

- We then send the “pulses” u_j to the machine for execution.

Quantum optimal control

- Trying to find the best $u_j(t)$'s such that the evolution fulfills:
  - The evolution results is close to a target state $|\psi_{\text{target}}\rangle$.
  - The evolution transformation is close to a unitary U_{target}.

- Let the evolution be $V = e^{\int_0^1 H(\tau)\,\mathrm{d}\tau}$.
- Loss function:
  - For the first one: $1 - \langle\psi_{\text{target}}|V|\psi_0\rangle$.
  - For the second one: $1 - \mathrm{tr}\left[U_{\text{target}}V^{\dagger}\right]$.

A solution in the literature

- Discretization: divide the time interval to mini-intervals $[t_k, t_{k+1}]$.
- For $[t_k, t_{k+1}]$, the Hamiltonian keeps the same:

$$H^{(k)}(u_{1k}, u_{2k}, \dots, u_{nk}) = H_0 + \sum_{j=1}^{n} u_{jk} H_j$$

- Then the evolution approximately is

$$V(u_{11}, \dots, u_{nm}) \approx \prod_{k=m}^{1} e^{-iH^{(k)}(u_{1k}, u_{2k}, \dots, u_{nk})\Delta t_k}$$

- Differentiation goes naturally.
- This is typically done on a classical computer. (GRAPE algorithm)

SWITCH TO OTHER SLIDES

Overview

Parameter Shift

Problem: Solving optimization problems on quantum computers using gradient-based methods.

Objective function: We will start with a general form and give a concrete example in Chapter 3.

Current Methods: Pulse-based parametrization.

- Advantage: Easy to compute gradients.
- Disadvantage: Not efficient enough. Not analog.
- We will talk about how to differentiate the pulse-based representation in the following slides.

Our goal: How to find a more efficient representation (with differentiation) for analog simulation.

Parameter Shift

In Yuxiang's slides, the state $|\psi(T)\rangle$ at time T is,

$$|\psi(T)\rangle = e^{\int_0^T -iH(t)dt} |\psi(0)\rangle \tag{1}$$

where

$$H(t) = \sum_j u_j(t) H_j \tag{2}$$

$u_j(t)$s are the parameters to optimize.
A scalar loss function l can be defined as,

$$l = \langle\psi(T)| B |\psi(T)\rangle \tag{3}$$

Bra-ket notation: $\langle v_1| M |v_2\rangle = \bar{v_1}^T M v_2$

Parameter Shift

$u_j(t)$ are continuous functions and hard to optimize. People nowadays discretize $u_j(t)$ in time to perform optimization. Time interval $[0, T]$ is divided to mini-intervals $[t_k, t_{k+1}]$. During each intervals, the $H(t)$ keeps constant.

$$H(t_k) = \sum_j u_{jk} H_j \tag{4}$$

In this setting, the state $|\psi(T)\rangle$ can be written into,

$$\begin{aligned} |\psi(T)\rangle &= \prod_k e^{-iH(t_k)\Delta t} |\psi(0)\rangle \qquad (5) \\ &= \prod_k \prod_j e^{-iu_{j,k}H_j \Delta t} |\psi(0)\rangle \qquad (6) \\ &= \prod_k \prod_j U_j(u_{j,k}) |\psi(0)\rangle \qquad (7) \end{aligned}$$

Parameter Shift

Let's do the differentiation

$$
\begin{aligned}
l &= \langle\psi(T)| B |\psi(T)\rangle & (8)\\
\frac{\partial l}{\partial u_{jk}} &= \left\langle \frac{\partial\psi(T)}{\partial u_{jk}} \right| B |\psi(T)\rangle + \langle\psi(T)| B \left| \frac{\partial\psi(T)}{\partial u_{jk}} \right\rangle & (9)\\
&= \left\langle \frac{\partial\psi(T)}{\partial u_{jk}} \right| B |\psi(T)\rangle & (10)\\
&+ \langle\psi(T)| B \left| ..U_{j+1}(u_{j+1,k})(-i\Delta t H_j)U_j(u_{j,k})U_{j-1}(u_{j-1,k})...|\psi(0)\right\rangle & (11)\\
&= \langle v| B_j |v\rangle & (12)
\end{aligned}
$$

where B_j is also a Hermitian matrix that can be computed by quantum computers,

$$
\begin{aligned}
|v\rangle &= \left| U_j(u_{j,k})U_{j-1}(u_{j-1,k})...|\psi(0)\right\rangle & (13)\\
B_j &= \Delta t[H_jU_{j+1}(u_{j+1,k})...B - iB..U_{j+1}(u_{j+1,k})H_j] & (14)
\end{aligned}
$$

Parameter Shift

More details about the parameter shift technique can be found in [MNKF18], [SBG$^+$19]).

The next lecture will also elaborate on this method.

Problems: how to compute the gradient for analog simulation? How to design a programming language for the differentiation?

Overview

Background

The **Quantum Approximate Optimization Algorithm (QAOA)** was proposed by Farhi et. al. [FGG14] in 2014. This algorithm produces approximate solutions for discrete optimization problems.

A **discrete optimization problem** is to find the maxima (or minima) of a function defined over a discrete domain.

Discrete optimization problems include Boolean Satisfaction problem (SAT), Traveling salesman problem (TSP), Max-Cut problem, etc. These problems are usually $\mathtt{NP}$-hard.

Max-Cut problem

Suppose $G = (V, E)$ is a graph. A **cut** of a graph is a partition of the vertices in the graph into two disjoint subsets. We can write a cut as $C = (S, T)$.

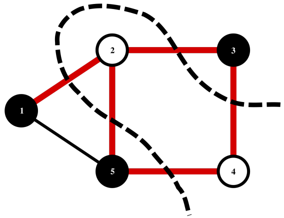


Figure: A cut on a 5-node graph.

In the above graph, we have a cut $C = (\{1, 3, 5\}, \{2, 4\})$.

Max-Cut problem

The **size** of a cut $C = (S, T)$ is the number of edges between the set S and the set T. A cut $C = (S, T)$ is called **maximal** if it has the largest size among all cuts.

The **Max-Cut** problem is to find a maximal cut of the graph.

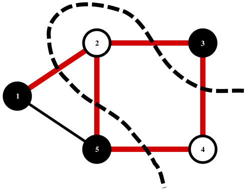


Figure: An example of a maximal cut.

The Max-Cut problem has applications in various fields. e.g., theoretical physics, circuit design, etc.

Encoding Max-Cut by bit strings

We can represent a cut $C = (S, T)$ of an n-node graph G by an n-bit string:

$$C = b_1 b_2 ... b_n,$$

with $b_j = 1$ if the j-th node is in S, and $b_j = 0$ of it is in T.

For example, the cut in Figure 2 can be expressed as $C = 10101$.

This problem is known to be $\texttt{NP}$-complete.

Encoding Max-Cut by qubits

Suppose s is a bit string specifying a cut $C = (S, T)$, we can use the computational basis $|s\rangle$ in a n-qubit register to represent the same cut C. For example, $|10101\rangle$.

Recall that σ_z^j is the Pauli-Z operator at the j-th site: $\sigma_z^j |0\rangle_j = |0\rangle_j$ and $\sigma_z^j |1\rangle_j = -|1\rangle_j$. Define a new operator

$$C_{j,k} = \frac{1}{2}\left(1 - \sigma_z^j \otimes \sigma_z^k\right).$$

Exercise: check that

$$\langle 00|C_{1,2}|00\rangle = \langle 11|C_{1,2}|11\rangle = 0,$$

$$\langle 10|C_{1,2}|10\rangle = \langle 01|C_{1,2}|01\rangle = 1.$$

Encoding Max-Cut by qubits

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \sigma_z^1 \otimes \sigma_z^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$\begin{aligned} |00\rangle &= \left|[1,0,0,0]^T\right\rangle \\ |01\rangle &= \left|[0,1,0,0]^T\right\rangle \\ |10\rangle &= \left|[0,0,1,0]^T\right\rangle \\ |11\rangle &= \left|[0,0,0,1]^T\right\rangle \end{aligned}$$

Encoding Max-Cut by qubits

Let $|s\rangle$ be a cut of a graph G, and there is an edge $(j,k) \in E$. We find that $\left\langle s\middle|C_{j,k}\middle|s\right\rangle = 0$ if the vertices j,k are in the same subset; otherwise, $\left\langle s\middle|C_{j,k}\middle|s\right\rangle = 1$.

Encoding Max-Cut by qubits

Let $|s\rangle$ be a cut of a graph G, and there is an edge $(j,k) \in E$. We find that $\left\langle s \middle| C_{j,k} \middle| s \right\rangle = 0$ if the vertices j,k are in the same subset; otherwise, $\left\langle s \middle| C_{j,k} \middle| s \right\rangle = 1$.

It is now clear that if we define a “size” operator

$$C = \sum_{(j,k)\in E} C_{j,k},$$

then $\langle s|C|s\rangle$ is the size of the cut $|s\rangle$.

C is a Hermitian matrix, for a quantum state $|\psi\rangle$ in the register, we can perform a measurement by C and the result is

$$\langle \psi|C|\psi\rangle = \mathrm{Tr}\left[C\,|\psi\rangle\,\langle\psi|\right]. \tag{16}$$

Encoding Max-Cut by qubits

Note that the operator $C_{j,k}$ is diagonal in computational basis, we can directly measure a state $|\psi\rangle$ and compute $\left\langle\psi\middle|C_{j,k}\middle|\psi\right\rangle$. An n-node (undirected) graph can have at most $\frac{n(n-1)}{2} = O(n^2)$ edges, so we can always compute $\langle\psi|C|\psi\rangle$ efficiently by

$$\langle\psi|C|\psi\rangle = \sum_{(j,k)\in E} \left\langle\psi\middle|C_{j,k}\middle|\psi\right\rangle .$$

Encoding Max-Cut by qubits

Note that the operator $C_{j,k}$ is diagonal in computational basis, we can directly measure a state $|\psi\rangle$ and compute $\langle\psi|C_{j,k}|\psi\rangle$. An n-node (undirected) graph can have at most $\frac{n(n-1)}{2} = O(n^2)$ edges, so we can always compute $\langle\psi|C|\psi\rangle$ efficiently by

$$\langle\psi|C|\psi\rangle = \sum_{(j,k)\in E} \langle\psi|C_{j,k}|\psi\rangle .$$

In fact, we find that the operator C is diagonal, and its largest eigenvalue is the largest size of a cut. In other words,

$$|\mathrm{MaxCut}| = \max_{|\psi\rangle} \langle\psi|C|\psi\rangle .$$

Idea: perhaps we can prepare some state $|\psi\rangle$ that approximates a max-cut state $|s_{\max}\rangle$?

QAOA for Max-Cut

The idea of QAOA is to introduce a parametrized quantum circuit $U(\boldsymbol{\beta}, \boldsymbol{\gamma})$ with p layers:

$$U(\boldsymbol{\beta}, \boldsymbol{\gamma}) = U(B, \beta_p)U(C, \gamma_p)...U(B, \beta_1)U(C, \gamma_1), \tag{17}$$

where $U(M, t) = e^{-itM}$ is a parametrized evolution operator.

Here, B is called a *driving operator*, and we must have $[B, C] \neq 0$. A canonical choice of B is the sum of all single site Pauli-X operators,

$$B = \sum_{l=1}^{n} \sigma_x^l.$$

QAOA for Max-Cut

We initialize the register with the uniform superposition over all computational basis:

$$|s\rangle = |+\rangle \otimes ... \otimes |+\rangle ,$$

and the quantum state after implementing the quantum circuits $U(\beta, \gamma)$ is now

$$|\beta, \gamma\rangle := U(\beta, \gamma) |s\rangle . \tag{18}$$

QAOA for Max-Cut

We initialize the register with the uniform superposition over all computational basis:

$$|s\rangle = |+\rangle \otimes \dots \otimes |+\rangle \, ,$$

and the quantum state after implementing the quantum circuits $U(\boldsymbol{\beta}, \boldsymbol{\gamma})$ is now

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle := U(\boldsymbol{\beta}, \boldsymbol{\gamma}) \, |s\rangle \, . \tag{18}$$

Define

$$F_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) := \left\langle \boldsymbol{\beta}, \boldsymbol{\gamma} \middle| C \middle| \boldsymbol{\beta}, \boldsymbol{\gamma} \right\rangle . \tag{19}$$

QAOA$_p$ formulation of Max-Cut:

$$M_p := \max_{\boldsymbol{\beta}, \boldsymbol{\gamma}} F_p(\boldsymbol{\beta}, \boldsymbol{\gamma}).$$

How to train the QAOA algorithm?

Here we borrow the word "train" from machine learning, which basically means "to optimize".

1. **Gradient-free method**: the function value of $F_p(\beta, \gamma)$ can be efficiently evaluated. For fixed p, we can make regular mesh grid in the parameter space and compute function values on this grid. Other heuristic methods like Nelder–Mead can be used as well.

2. **Gradient-based method**: for QAOA, the gradient can also be efficiently computed (through a technique called *parameter shift* [MNKF18], [SBG$^+$19]).

Reference I

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, *A quantum approximate optimization algorithm*, arXiv preprint arXiv:1411.4028 (2014).

Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii, *Quantum circuit learning*, Physical Review A **98** (2018), no. 3, 032309.

John Preskill, *Quantum computing in the nisq era and beyond*, Quantum **2** (2018), 79.

Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran, *Evaluating analytic gradients on quantum hardware*, Physical Review A **99** (2019), no. 3, 032331.

SWITCH BACK

Unknown areas

A quantum solution?

- Is it possible to use a quantum evolution to compute the differentiations?

- Motivation: it is possible for circuit model.
    - Variational quantum eigen-solver.
    - Differentiable quantum while-language.
- A parametrized circuit’s gradients can be obtained by another circuit.

Another parametrization?

- Our target is to find optimal pulse wave.

- Discretization returns non-smooth results.
- High frequency signals are hard to simulate on machines.

- Parametrize by Fourier transformation? Or wavelet transformation?

Other applications?

- Current applications of optimal control are state preparation and gate synthesis.
- Variational quantum eigensolver finds ground states.
- Are there other ways to formulate the loss function?

A differentiable PL?

- We don’t have a PL to describe Hamiltonian evolution, for now.
- How to develop a differentiable variant of it?
- How to write down the rules for differentiation?

Q&A