Title: Machine Learning Lecture

Speakers: Damian Pope

Collection: Machine Learning 2023/24

Date: April 30, 2024 - 11:30 AM

URL: https://pirsa.org/24040054

Introduction to Quantum Machine Learning

Damian Pope, PhD

Some ideas from the course so far…

- Machine learning (ML)
- Supervised learning/unsupervised learning
- Ising model
- Many-body physics
- Phases of matter

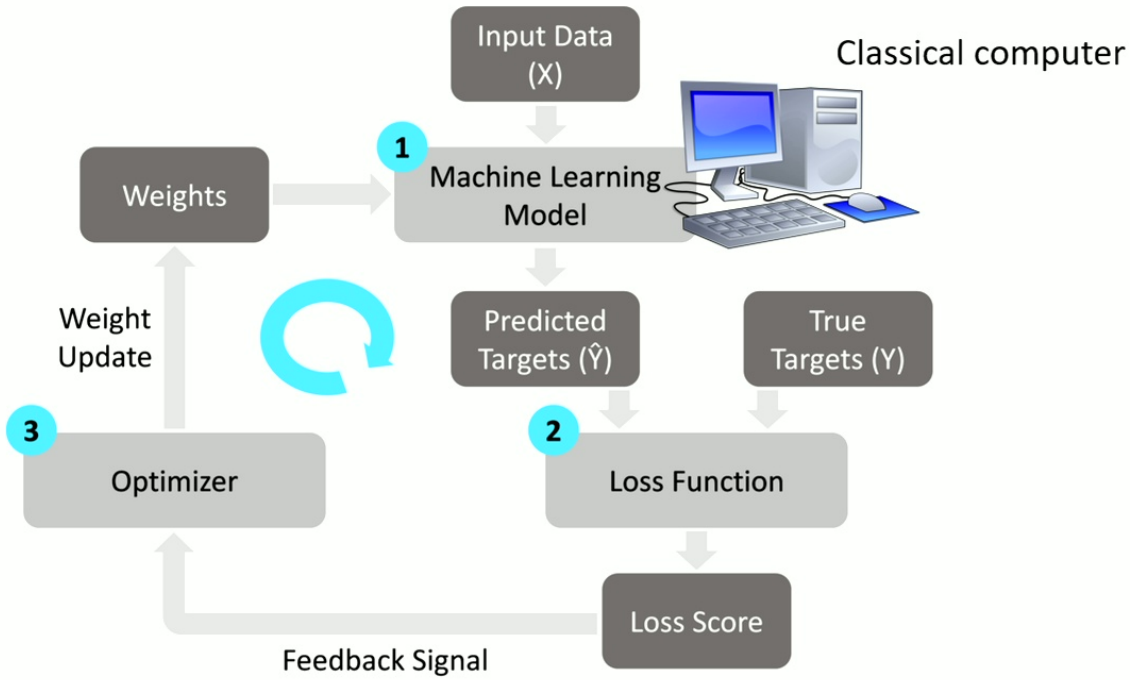
What is quantum machine learning? Definition

- 1. Classical machine learning (ML) that uses **quantum data**

E.g., using classical ML to learn when a phase transition occurs in a quantum system

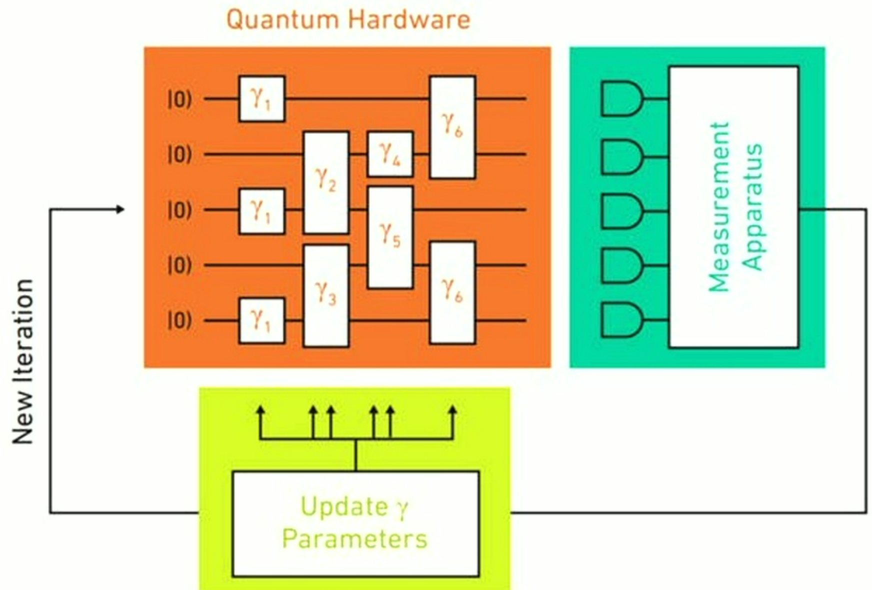
2. Machine learning done on a quantum computer instead of a regular classical computer.

High-level summary of one type of ML



Typically, QML still involves a classical computer

- Hybrid computation: quantum computer + classical computer



Why study QML?

- 1. Widely believed that, for some problems, QML is “better” than ML

(better = faster or better asymptotic scaling)

2. QML has applications in many areas of physics—**not** just in quantum information

E.g., **cosmologists** are using QML to model the quantum fields that are important to the evolution of the universe!

QML = useful tool

Why study QML?

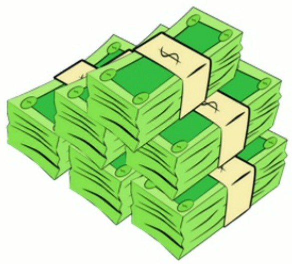
- 3. Exciting & rapidly growing field

-it's just getting started—many opportunities for young researchers to contribute & advance the field

(especially if you're physicist with good coding skills or a strong interest in coding and/or data science)

Why study QML?

- 4.

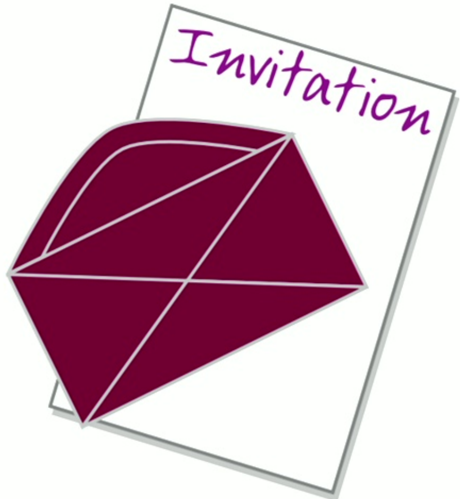


Google, GESDA and XPRIZE launch new competition in Quantum Applications

Mar 04, 2024
3 min read

XPRIZE Quantum Applications is a 3-year, $5M global competition designed to generate quantum computing (QC) algorithms that can be put into practice to help solve real-world challenges.

An invitation



- Outline: Two popular QML algorithms
    1. Variational quantum eigensolver
    2. QAOA

Caution!

- Not known how much better QML is than standard ML.
- A lot can be done using (classical) high-performance computers + sophisticated classical algorithms.
- Often, it's not obvious what the best that regular ML can achieve is.

Example of a QML algorithm
Variational quantum eigensolver (VQE)

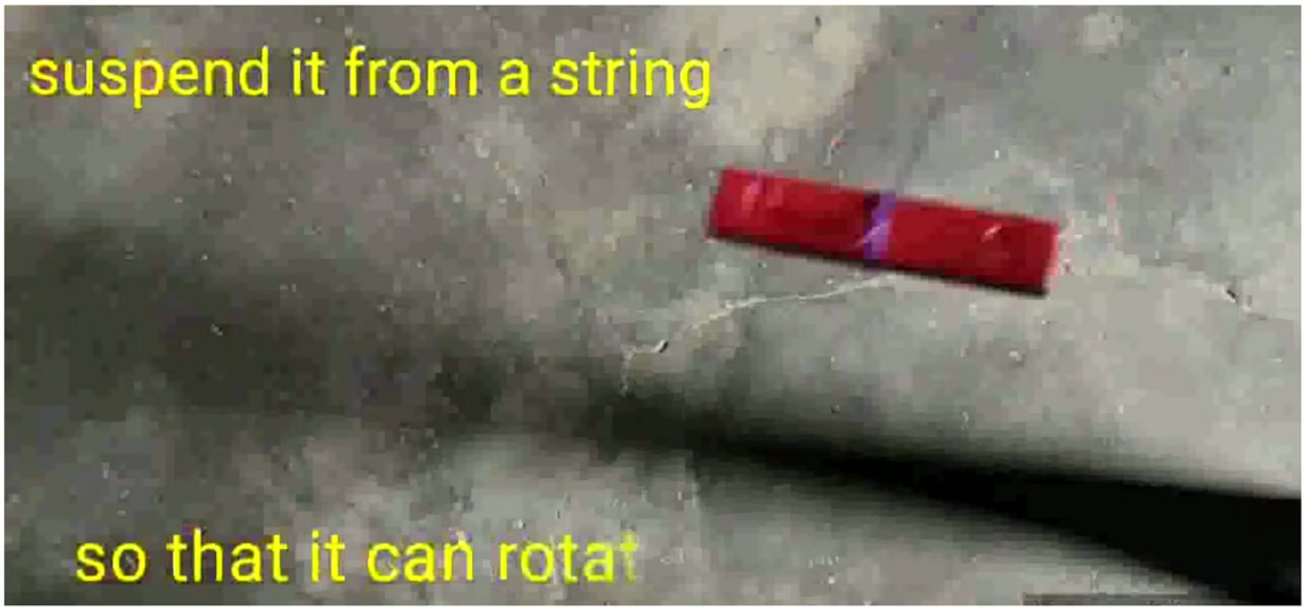
- We have a quantum system
- Would like to know its ground state.
- In many cases, it's hard to calculate.

- Calculating ground state is first step in calculating electronic properties of the system (e.g., conductivity, chemical reaction pathways)

Steps in VQE

- 1. Parameterize all the quantum states of the system
- 2. Take an educated guess at the ground state (ansatz)
- 3. Calculate the energy of your initial guess for the ground state. (QUANTUM COMPUTER)
- 4. Calculate a better guess (CLASSICAL COMPUTER/OPTIMIZER)
- 5. Calculate the energy of your new state.
- 6. Repeat steps 4. and 5. until the energy stops changing.

Classically, a bar magnet evolves to line up with the field, just like a compass needle.



Quantum case

- Consider a spin-1/2 particle in a (classical) magnetic field that's parallel to the z axis.

$$\left|\psi > = \cos\left(\frac{\theta}{2}\right)\right| \uparrow> \; + \sin\left(\frac{\theta}{2}\right) | \downarrow>$$

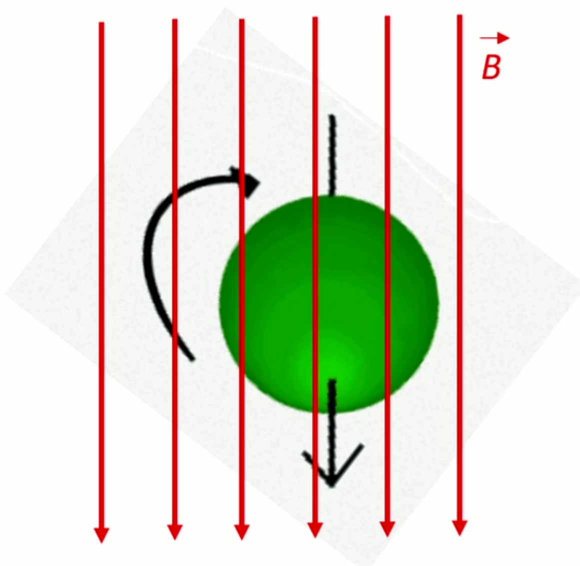
Problem: Which state minimizes $<\widehat{H}>$, the expectation value of $\widehat{H}$, the energy (or Hamiltonian) of the particle?

$$\widehat{H} = -B\hat{Z} = -B\hat{\sigma}_z$$

where $\hat{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \hat{\sigma}_z$ (*Pauli Z operator*)

Answer

$|\psi> = |\downarrow>$



We can also find this answer using VQE

1. Parameterize the quantum states of the system

$$\left|\psi >= \cos\left(\frac{\theta}{2}\right)\right| \uparrow> \ + \sin\left(\frac{\theta}{2}\right) | \downarrow>$$

(We've simplified by setting the azimuthal angle ϕ = 0, where $\left|\psi >= \cos\left(\frac{\theta}{2}\right)\right| \uparrow> \ + \sin\left(\frac{\theta}{2}\right) e^{i\phi} | \downarrow>$)

2. Take an educated guess at the ground state (ansatz)

- Let's try θ = π/2
- $\cos\left(\frac{\pi}{4}\right) = \sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}}$
- This gives $|\psi >= \frac{1}{\sqrt{2}}(\uparrow> \ + | \downarrow>)$ $=|\psi_0 >$

- **3. Calculate the energy of your initial guess for the ground state.**

(QUANTUM COMPUTER)

Prepare the spin-1/2 particle (qubit) in $= |\psi_0>$

Map spin-1/2 particle to a qubit

$$|0> \equiv |\uparrow>$$
$$|1> \equiv |\downarrow>$$

How do we prepare the initial state?

Typically, qubits are prepared to initially be in the state $|\psi_0>=|0>\equiv$
$|\uparrow>$

Rotate the particle about the y-axis by the angle θ:

$$\hat{R}_Y(\theta)=\begin{bmatrix} cos\frac{\theta}{2} & -sin\frac{\theta}{2} \\ sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix}$$

$|\psi>=\hat{R}_Y(\theta)|\psi_0>$ QUANTUM GATE

RECALL $\hat{H} = -B\hat{Z} = -B\hat{\sigma}_z$

- As $\hat{H} \propto \hat{Z}$, we can calculate $\langle\hat{H}\rangle$ by measuring $\hat{Z}$ many times and taking the average.

- **4. Calculate a better guess** (CLASSICAL COMPUTER/OPTIMIZER)

- One way of doing this is by calculating $\frac{d<\hat{H}>}{d\theta}$
- Then, calculating a new value for the parameter: $\theta' = \theta - \Delta \frac{d<\hat{H}>}{d\theta}$

Δ = step size (or *learning rate*)

This is just **gradient descent**.

- **5. Calculate the energy of the new state.**
- prepare the new state: $|\psi' >= \hat{R}_Y(\theta')|\psi_0 >$
- Measure *Z* many times and take the average.
- $<\hat{H}>$ = - B $< \hat{Z} >$

- **6. Repeat steps 4. and 5. until the energy stops changing.**

VQE is performed on computers.

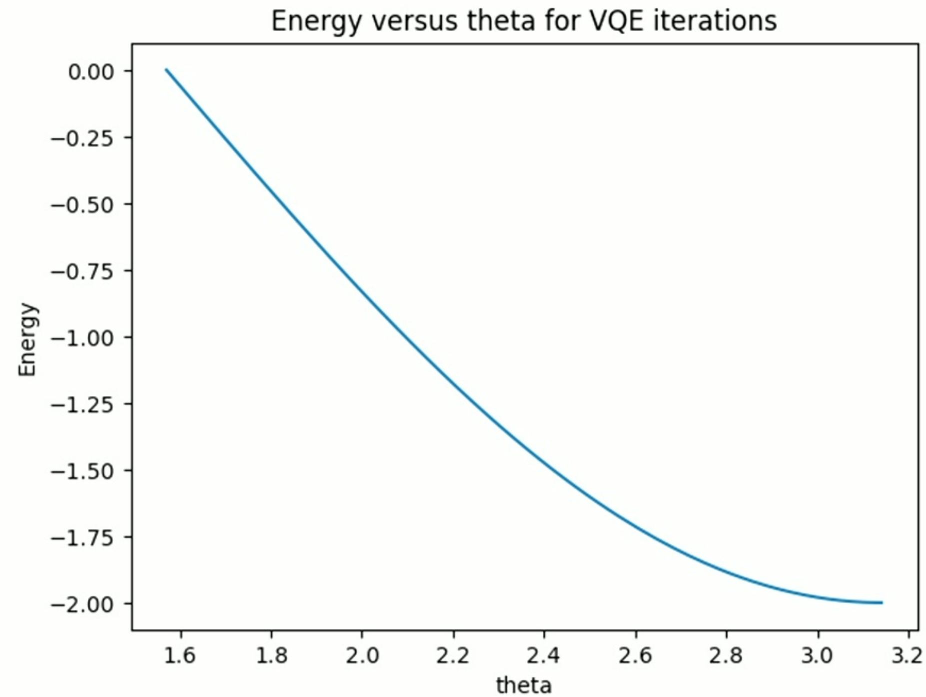
Python Libraries for QML

PENNYLANE

Qiskit

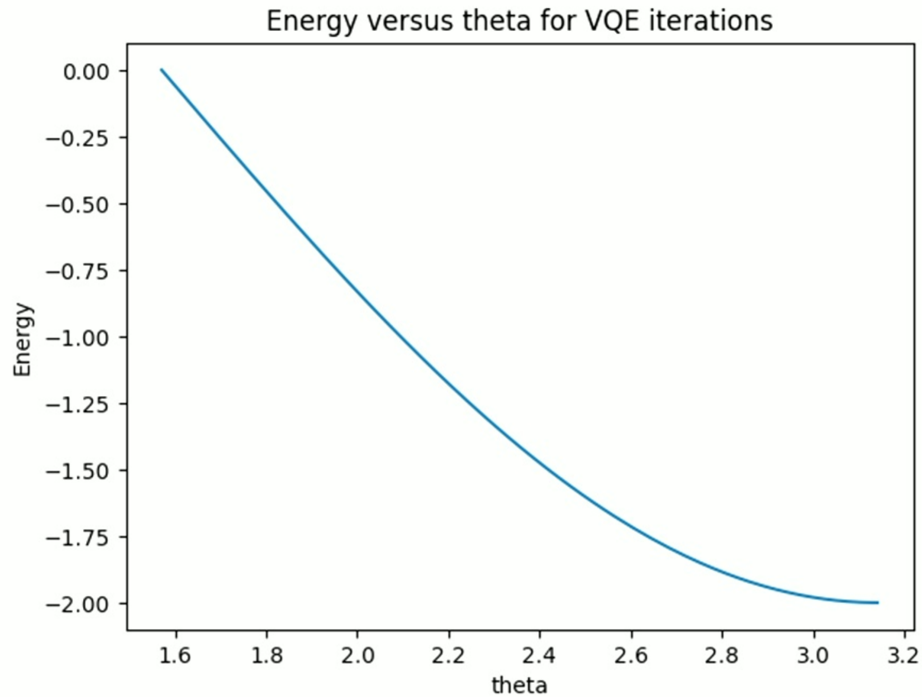
+ many others (Cirq, TKet, Amazon braket etc.)

Results



- Obviously, using VQE is overkill for this simple problem.

Results

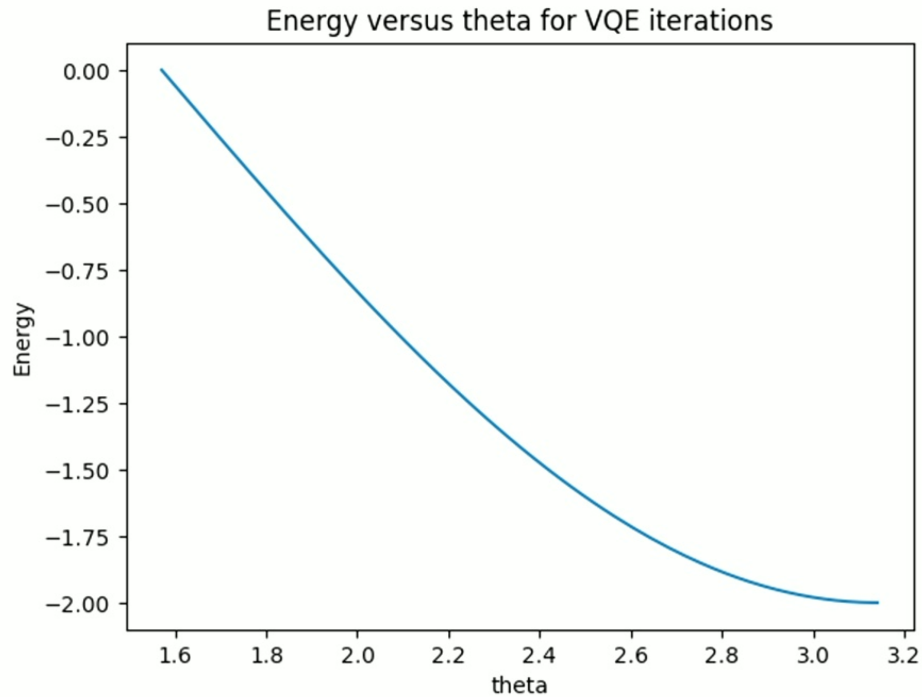


- Obviously, using VQE is overkill for this simple problem.

Applications: What's the ground state of the <u>quantum</u> Ising model?

- You saw the *classica*l Ising model in Lecture 6
- Let's look at the *quantum* Ising model
- Classical spins (up or down) → qubits
- See tutorial code for details

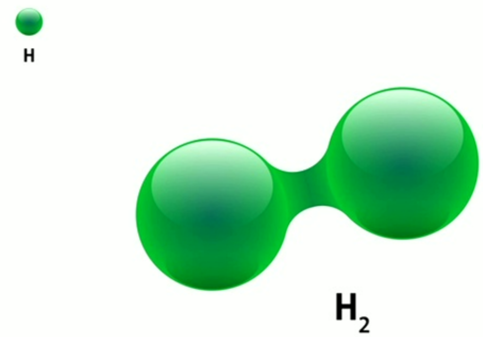
Results



- Obviously, using VQE is overkill for this simple problem.

Ground states of molecules

- E.g., H_2 molecule.
- Map orbitals to qubits.
- See tutorial code for details



Ground states of molecules

- Useful in modeling complex chemical reactions in carbon capture!

Many different ways to optimize!

- **Gradient-based**
- Momentum-based gradient descent
- Stochastic gradient descent
- Quantum Natural Gradient

- **Gradient-free**
- COBYLA

Intuition for why VQE might be useful/better

- For *N* qubits, the classical computing overhead seems to increase exponentially.

- Resource scaling of VQE is **polynomial**

Note that this is a variational algorithm

- Many QML algorithms are variational.

Quantum Approximation Optimization Algorithm = QAOA

- **What problem does it solve?**

- **Let's look at an example!**

- Classical Ising model from Lecture 6

- *N* spins
- Each spin is either up or down (2 classical states)
- h_i = local magnetic field applied to i^{th} spin
- Nearest-neighbour interactions

This is a *combinatorial optimization problem*

- z_i = spin of i^{th} particle

- Minimize the function H(**z**) = $-J \sum_{i=1}^{N-1} z_i z_{i+1} + \sum_{i=1}^{N} h_i z_i$

where **z** = $\{z_1, z_2, z_3 \ldots z_n\}$ and

J = nearest-neighbour interaction strength

In principle, we could solve the problem via adiabatic quantum computation

Map the *classical* energy $H_{\text{Ising}}(\mathbf{z})$ to the energy of the qubits

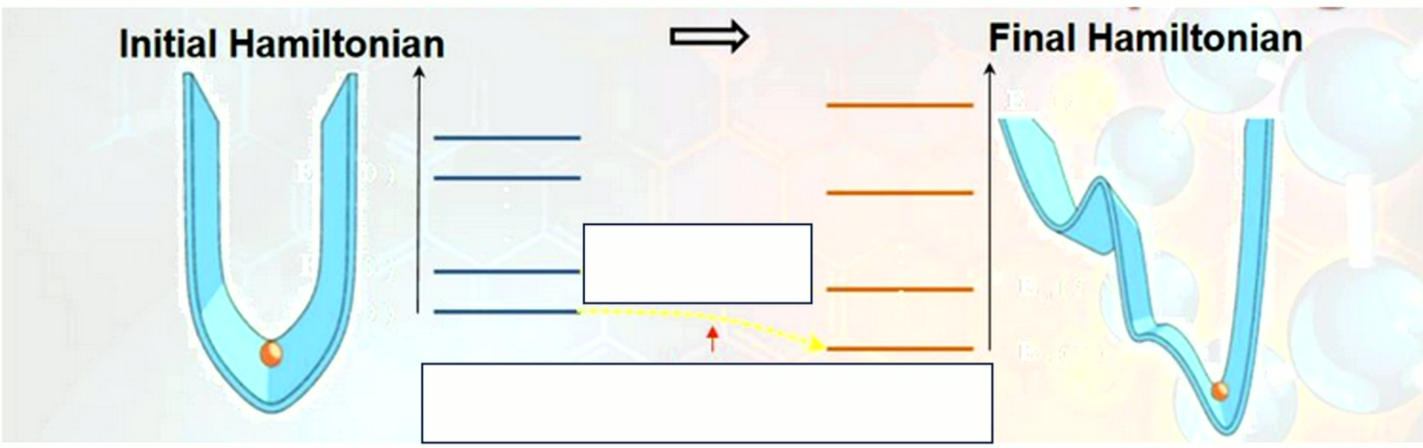
H($\mathbf{z}$) = $-\mathrm{J}\sum_{i=1}^{N-1} z_i z_{i+1} + \sum_{i=1}^{N} h_i z_i$ H($\mathbf{z}$) → $\hat{H}_{Ising} = -\mathrm{J}\sum_{i=1}^{N-1} \hat{z}_i \hat{z}_{i+1} + \sum_{i=1}^{N} h_i \hat{z}_i$

CLASSICAL QUANTUM

Adiabatic quantum evolution

- adiabatic = very slow
- Start the *N* qubits in the ground state of a "simple" Hamiltonian $\widehat{H}_S$
- **Slowly** (i.e., adiabatically) change the Hamiltonian from $\widehat{H}_S$ to $\widehat{H}_{Ising}$

- Quantum system stays in ground state
- We end up with the ground state of $\widehat{H}_{Ising}$!



$\hat{H}(t) = (1 - t/T)\hat{H}_S + t/T\ \hat{H}_{Ising}$, for $0 \le t \le T$

T = total evolution time

This is theoretically possible, but hard to do in practice

QAOA approximates this ideal approach

Let the simple Hamiltonian be $\widehat{H}_S = \sum_{i=1}^{N} \widehat{X}_i$

where $X_i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \hat{\sigma}_{X,i}$ (Pauli X operator for the i^{th} qubit)

We want to simulate the following time evolution:

$\widehat{U}$(t) = f ($\widehat{H}(t)$)

$$\widehat{H}(t) = (1 - t/T)\widehat{H}_S + t/T\ \widehat{H}_{Ising}$$

But, how?

Let the simple Hamiltonian be $\widehat{H}_S = \sum_{i=1}^{N} \hat{X}_i$

where $X_i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \hat{\sigma}_{X,i}$ (Pauli X operator for the i^{th} qubit)

Start system in ground state of $\widehat{H}_S$:

$$\Big|\psi > = \widehat{H}_S{}^{\otimes N}\Big| 1 > = \frac{1}{\sqrt{2}} \sum_{i=1}^{2^N} (-1)^i |i >$$

Use a **constant** Hamiltonian within each segment

Within the m^{th} time segment,

$\boldsymbol{\hat{U}}$ **(t$_m$) = exp[(-i/$\hbar$) $\boldsymbol{((1 - t_m/T)\hat{H}_S + t_m/T\hat{H}_{Ising})}$ Δt]**

This is of the form $\hat{U} = \exp((\hat{A} + \hat{B})\Delta t)$

- $\exp((\hat{A} + \hat{B})\Delta t) \neq \exp((\hat{A})\Delta t)\, \exp((\hat{B})\Delta t)$

It’s known that **exp**$\left(\left(\boldsymbol{\hat{A}} + \boldsymbol{\hat{B}}\right)\boldsymbol{\Delta t}\right) \approx \prod_{n}$ **[exp(**$\hat{A}$ **Δ*t/n*) exp(**$\hat{B}$ **Δ*t/n*)]*^n***

for Δ*t* << 1 and *n* >> 1

(*Lie-Trotter formula*)

So, we can approximate $\hat{U}$ (t_m) as follows:

1. Apply $\hat{H}_{Ising}$ for a time (t_m/T) Δt/n

2. Apply $\hat{H}_S$ for a time $[(1 - t_m/T)$ Δt/n]

3. Repeat 1 and 2 *n* times.

- Repeat the entire procedure for each of the M intervals.

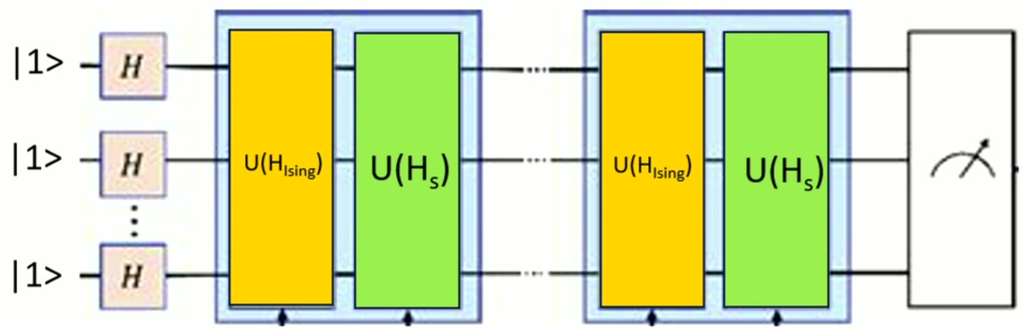
For m = 1 to M:

1. Apply $\widehat{H}_{Ising}$ for a time $(t_m/T)\ \Delta t/n$
2. Apply $\widehat{H}_S$ for a time $[(1 - t_m/T)\ \Delta t/n]$
3. Repeat 1 and 2 *n* times.

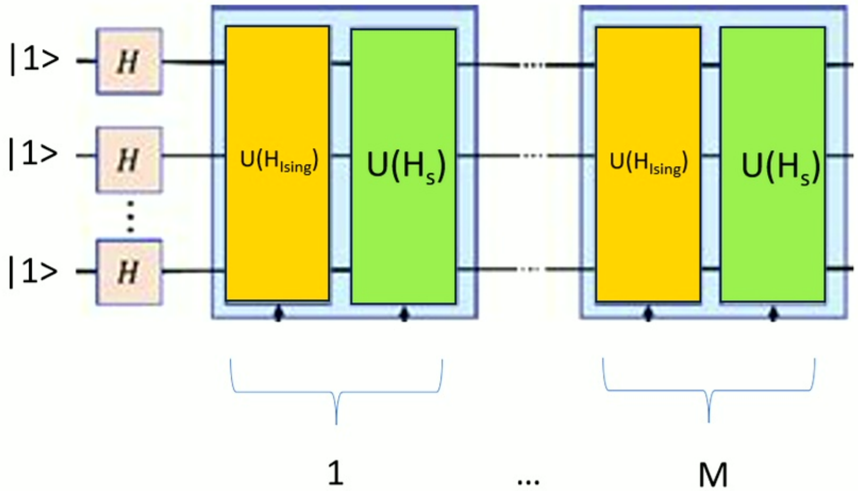
At the end, measure each qubit in the computational (i.e., *Z*) basis.

The bitstring of results represents a possible solution.

E.g., for N = 3, we *might* measure '0 1 0'. This represents ↑↓↑ in the Ising model.



Visualization



Steps in QAOA

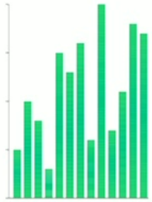
1. Start in ground state of $\hat{H}_S$: $\left|\psi > = \hat{H}_S{}^{\otimes N}\right| 1 > = \frac{1}{\sqrt{2}} \sum_{i=1}^{2^N} (-1)^i |i >$

2. For p = 1 to P:
   a. Apply $\hat{H}_{Ising}$ for a *randomly chosen length of time*
   b. Apply $\hat{H}_S$ for a randomly chosen length of time
   (Note: P = M n)

3. Measure each qubit in the computational (i.e., Z) basis.

4. Repeat Steps 1. to 3. a number of times. This produces a sample the distribution of outcomes.

5. Calculate the value of *H* for each possible outcome & output the lowest H (energy value)



6. Classically optimize all $2P$ time durations for $\hat{H}_{Ising}$ and $\hat{H}_S$.

(This is where the machine learning comes in.)

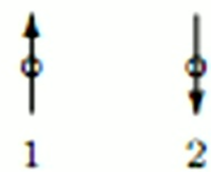
7. Repeat steps 1. to 6. until either:

a) the result stops changing or

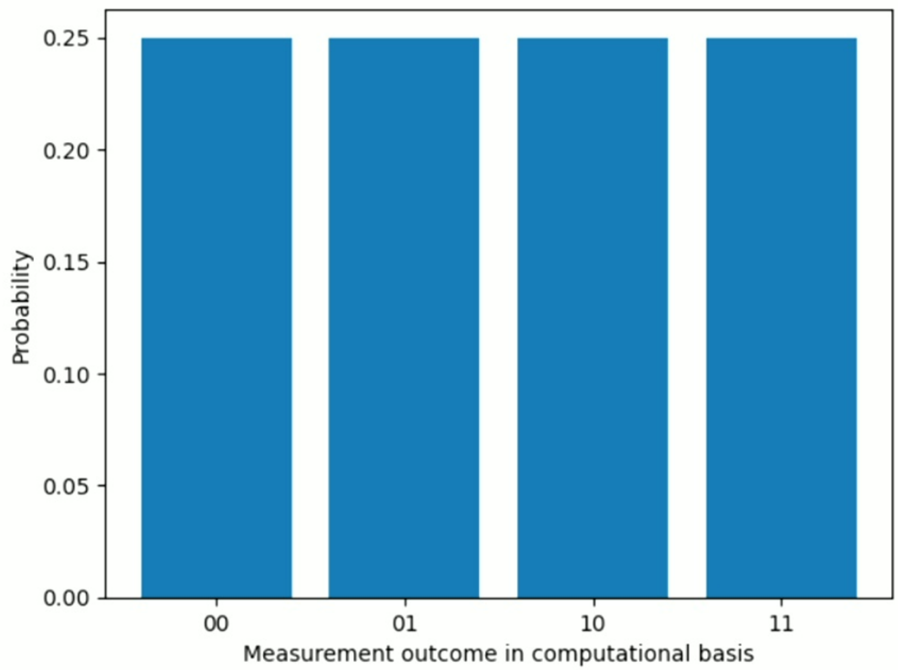
b) we're confident enough that we have a good enough approximation to the ground state.

Let's look at a toy example

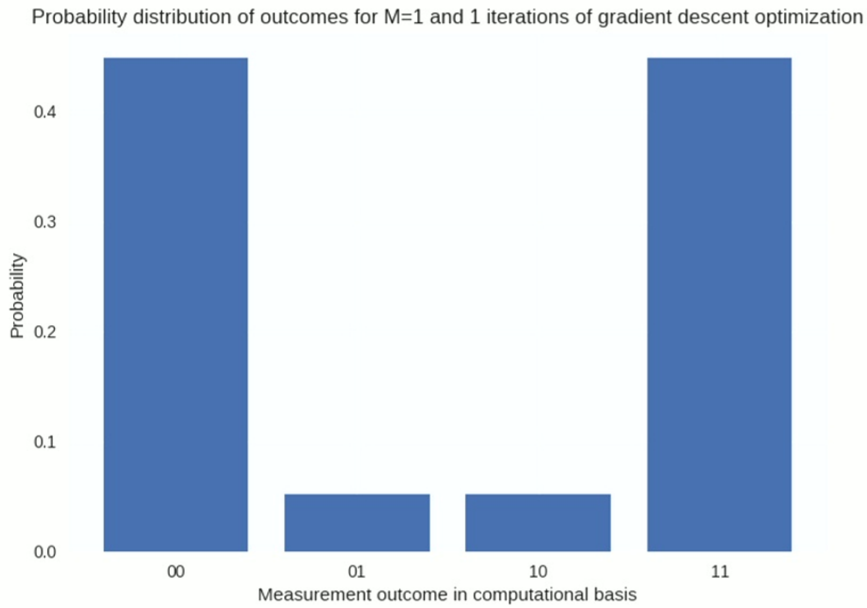
- Classical Ising model with two spins, N = 2 and $h_1 = h_2 = 0$



Probability distribution for ground state of $\hat{H}_s$



Some results for M = 1 (and n = 1)



1 optimization iteration

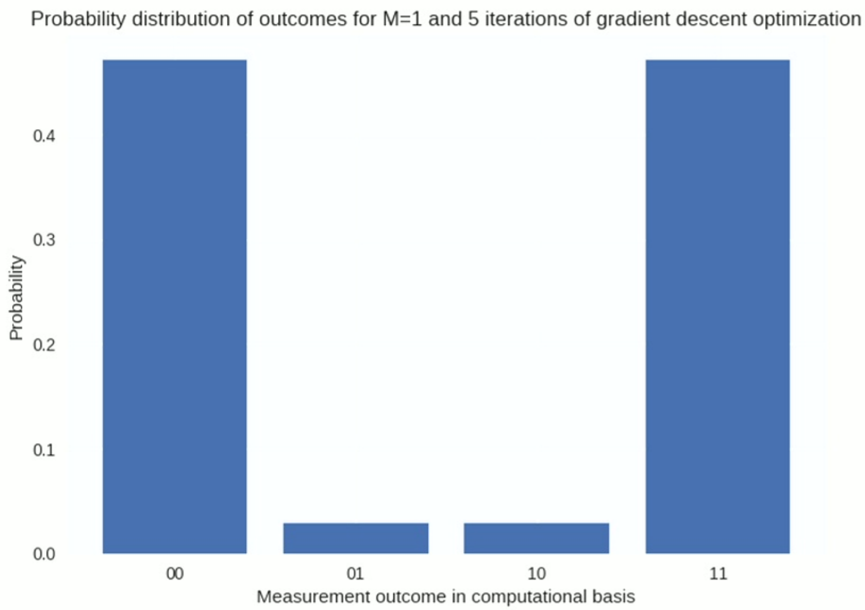
0 0 → ↑↑

1 1 → ↓↓

Calculate H:
H(↑↑) = -(-1)(-1) = -1
H(↓↓) = -(+1)(+1) = -1

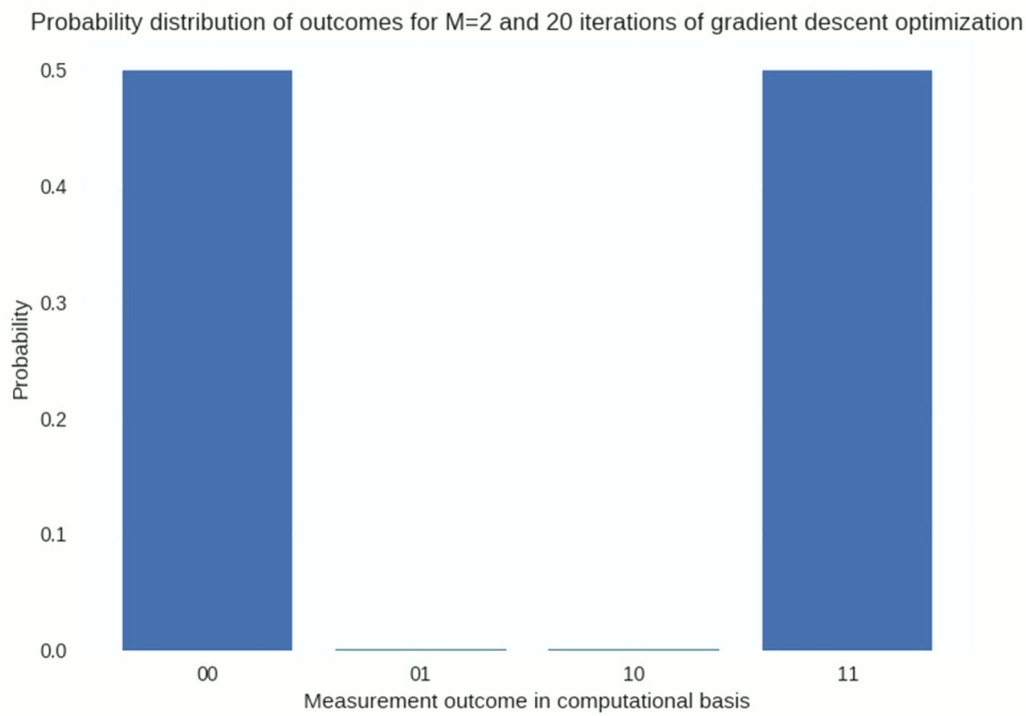
Ground state!

Five optimization iterations



5 optimization iterations

M = 2 and 20 optimization iterations



It's known that $\mathbf{exp}\big((\hat{\boldsymbol{A}} + \hat{\boldsymbol{B}})\boldsymbol{\Delta t}\big) \approx \prod_{n} [\mathbf{exp}(\hat{A}\ \Delta t/n)\ \mathbf{exp}(\hat{B}\ \Delta t/n)]^{n}$

for $\Delta t \ll 1$ and $n \gg 1$

(*Lie-Trotter formula*)

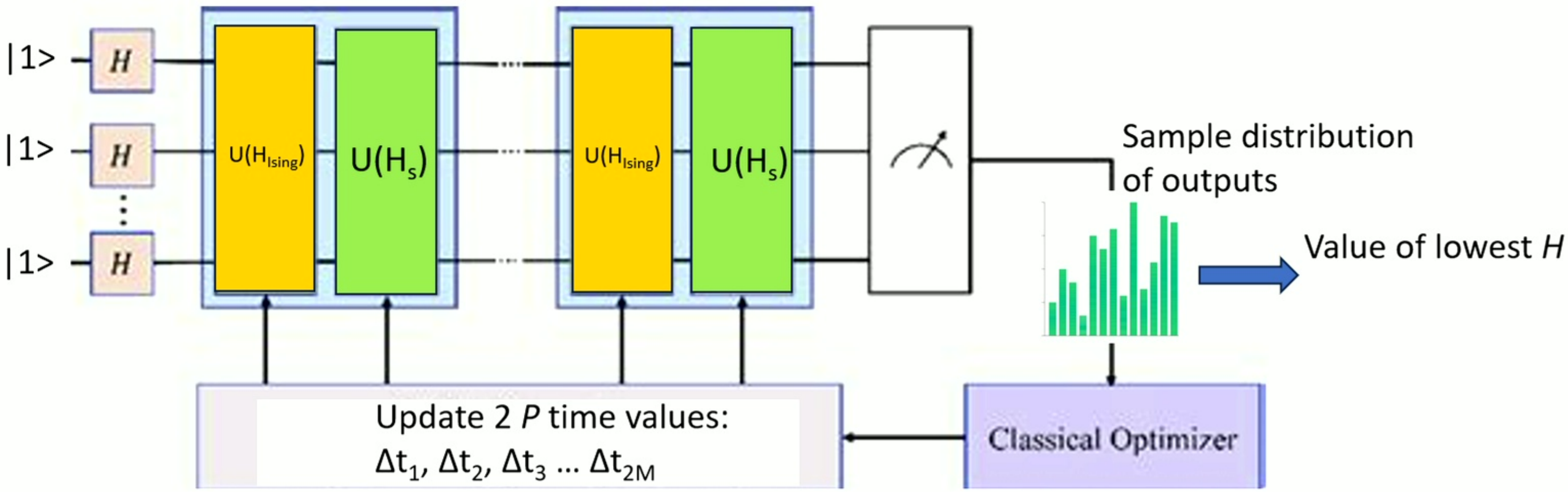
So, we can approximate $\hat{U}$ (t_m) as follows:

1. Apply $\hat{H}_{Ising}$ for a time $(t_m/T)\ \Delta t/n$

2. Apply $\hat{H}_S$ for a time $[(1 - t_m/T)\ \Delta t/n]$

3. Repeat 1 and 2 *n* times.

Steps in QAOA



Many real-world problems can be represented as COP problems

- Finance
- Logistics
- Chemistry
- Materials science
- Many-body physics

- QAOA has many applications within & outside of physics

- Performance of QAOA versus the best classical algorithms?

Lots of interest in VQE and QAOA

Part of the reason is that we can run them on quantum computers with a relatively small number of quantum gates.

Shallow quantum circuits.

Doable on today's (and near future) quantum computers. NISQ era

NISQ = noisy intermediate scale quantum computers

Further references to continue your QML journey

CODE USED IN THIS LECTURE TO GET THE RESULTS FOR VQE & QAOA: https://drive.google.com/drive/folders/1vVcwu_JxasUOuXOH_V5bSMI-1qdvAAxm?usp=drive_link

QML MOOC, Peter Wittek

https://www.youtube.com/playlist?list=PLmRxgFnCIhaMgvot-Xuym_hn69lmzIokg

(41 Lectures!)

QAOA: A different perspective | PennyLane Tutorial

https://www.youtube.com/watch?v=cMZcA2SQnYQ

An Introduction to Quantum Optimization Approximation Algorithm (University of Maryland)

https://www.cs.umd.edu/class/fall2018/cmsc657/projects/group_16.pdf

Maria Schuld, I. Sinayskiy, F. Petruccione

An Introduction to Quantum Machine Learning

https://arxiv.org/abs/1409.3097

Big questions in QML

- How much better are VQE and QAOA than the best possible classical algorithms?

- What other QML algorithms exist?
- (thinking “quantumly”)

- What problems (that we’re interested in) can QML algorithms solve?
- Physics problems & real-world problems