CPEN 400Q / EECE 571Q Lecture 15
VQC model selection, and introducing the variational eigensolver

Tuesday 8 March 2022

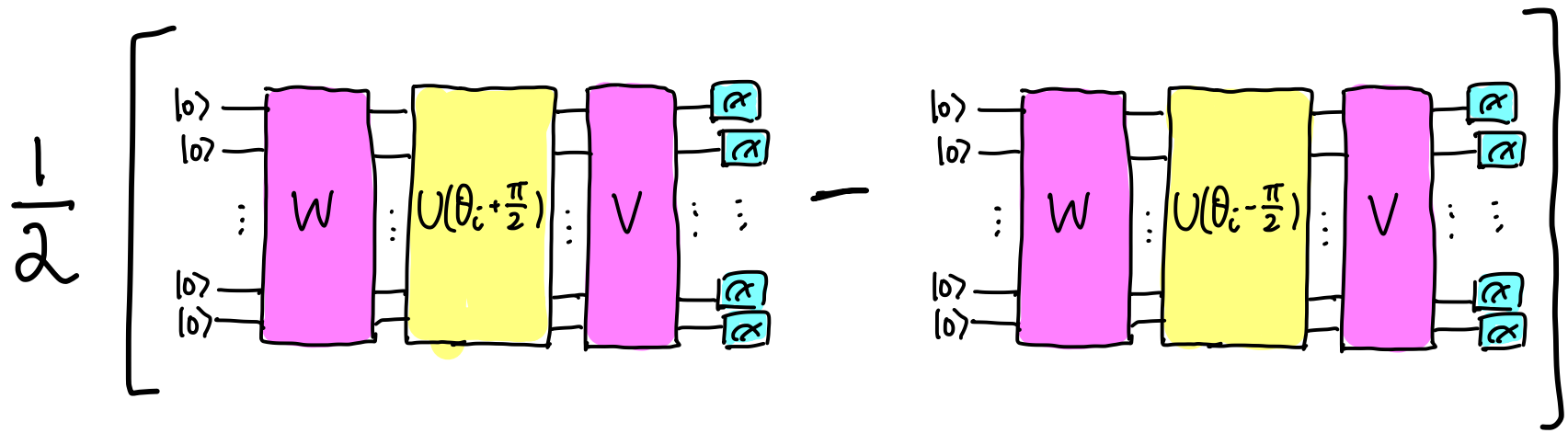
Announcements

- Assignment 3 due Friday 11 March 23:59
- Scheduling final presentations: GitHub issue with time slots will be posted *tomorrow at 10am* (note that full implementation and report are due at *end of term*)

★ Problem-solving session on Zoom w/ Gideon tomorrow @ 15:00

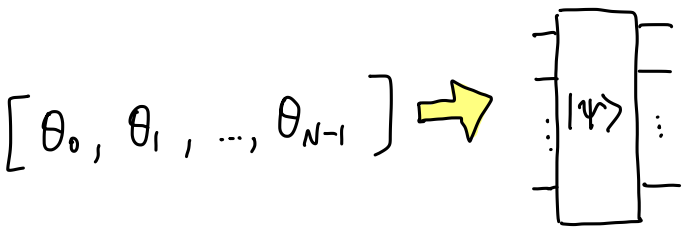
Last time

We derived the parameter-shift rule for single-qubit Pauli rotations.

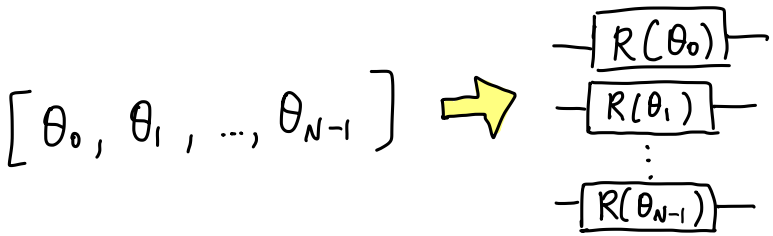


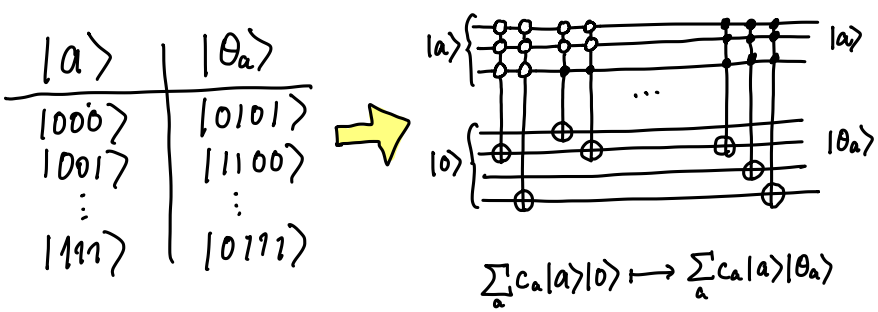
Last time

We outlined different methods for embedding classical data into quantum circuits.

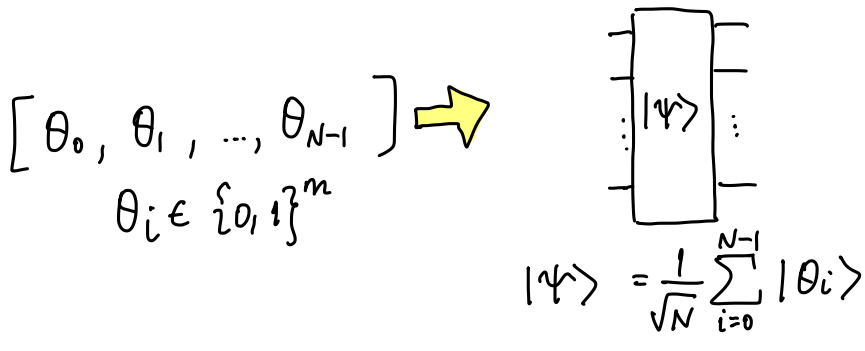


$$|\psi\rangle \approx \theta_0|0\rangle + \theta_1|1\rangle + \cdots + \theta_{N-1}|N-1\rangle$$



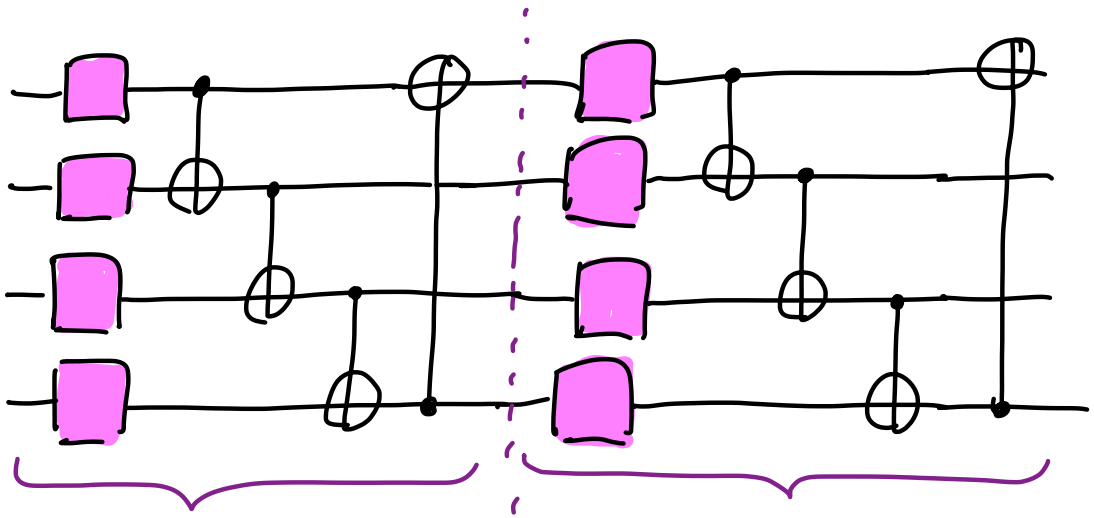


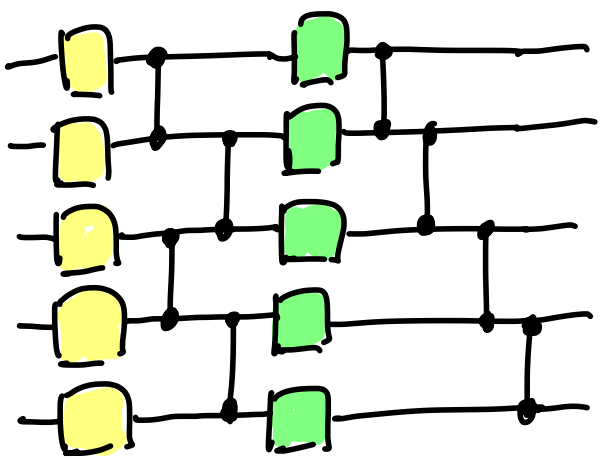
$$\sum_a c_a|a\rangle|0\rangle \longmapsto \sum_a c_a|a\rangle|\theta_a\rangle$$

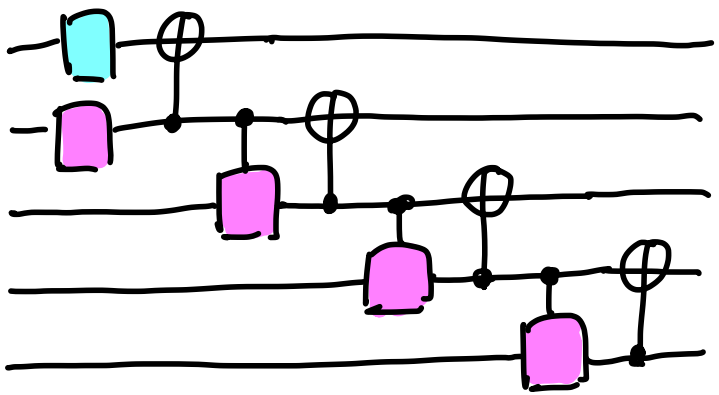


Last time

We discussed different strategies for building *parametrized quantum circuits*:

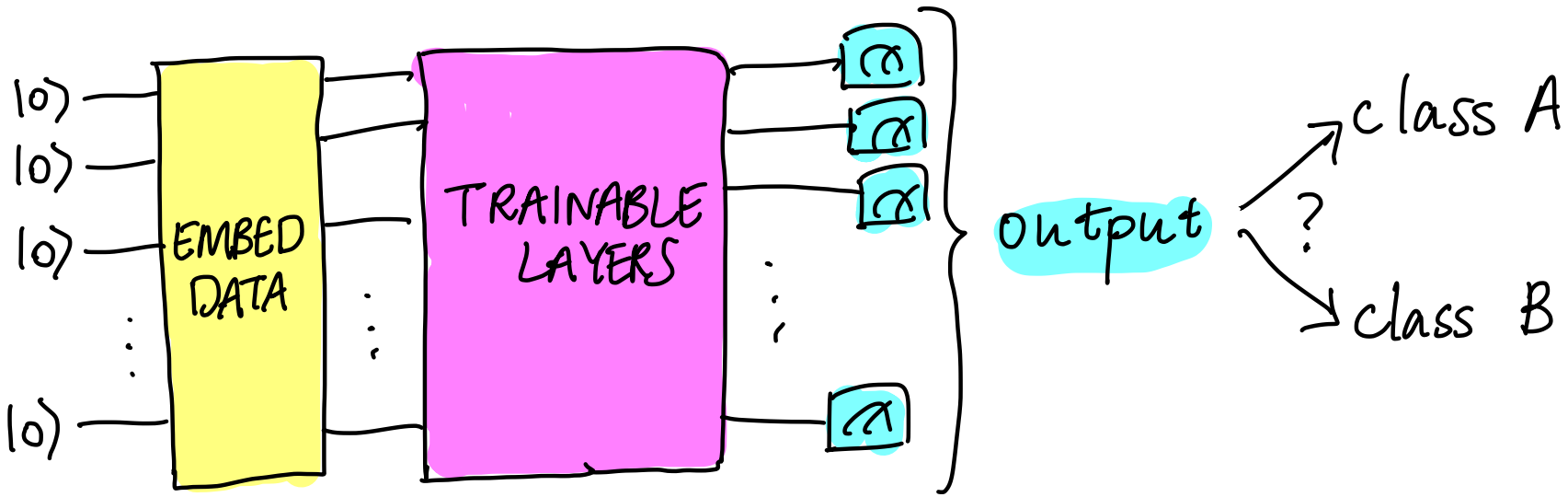






Last time

We used these ideas to start designing quantum models to classify data using the *variational quantum classifier*.

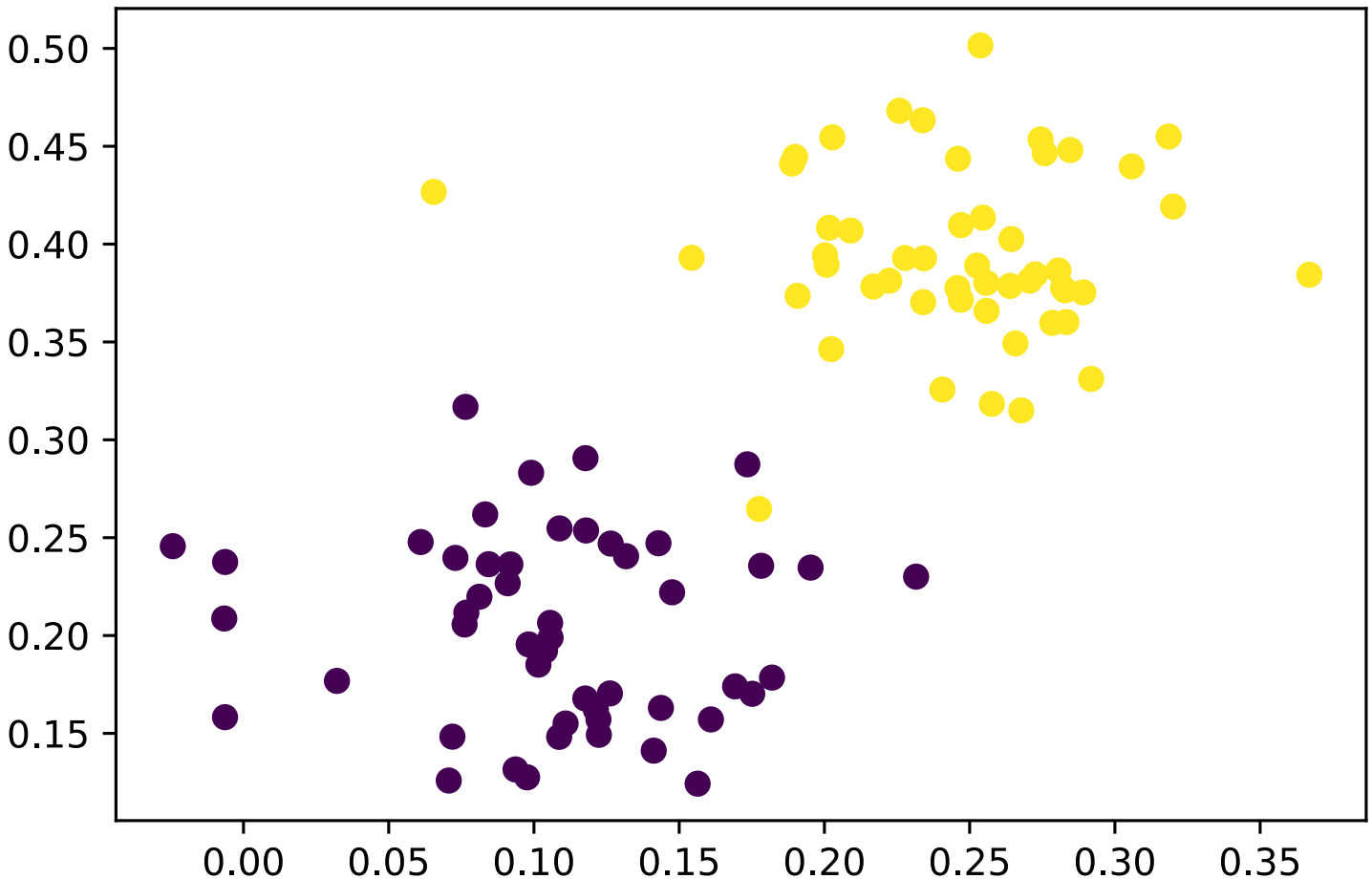


Learning outcomes

- Compare and contrast embedding strategies and quantum models for variational classification
- Define a *Hamiltonian*, and construct one in PennyLane
- Use the variational quantum eigensolver to find the ground state energy of a Hamiltonian

Variational quantum classifier

We are working with 2-dimensional data:



Variational quantum classifier

Our cost function is a simple least-squares fit.

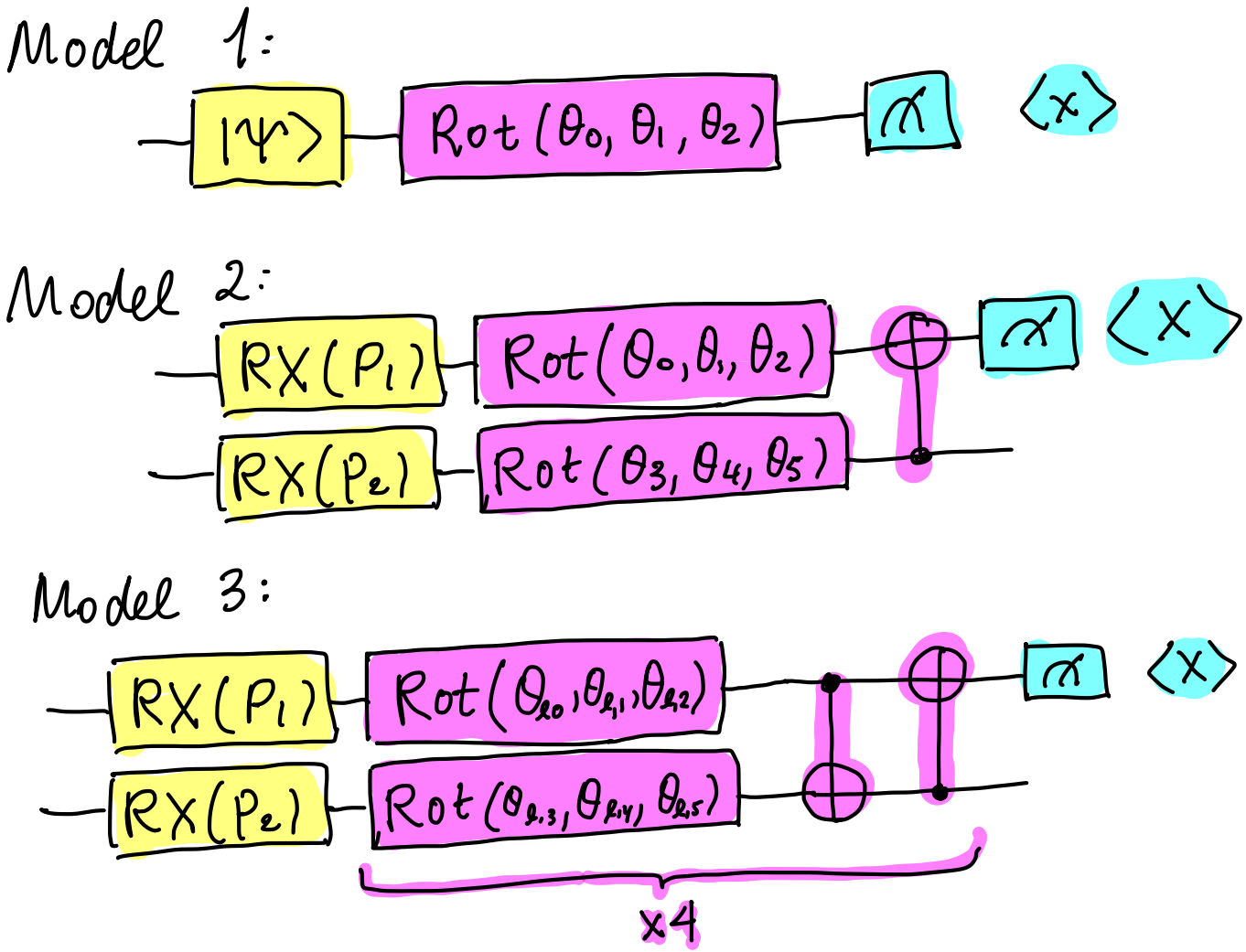
```
def cost(data, true_labels):
    total = 0.0

    for data_point, label in zip(data, true_labels):
        computed_exp_val = circuit(data_point)
        total += (computed_exp_val - label) ** 2

    return total / len(data)
```

Variational quantum classifier

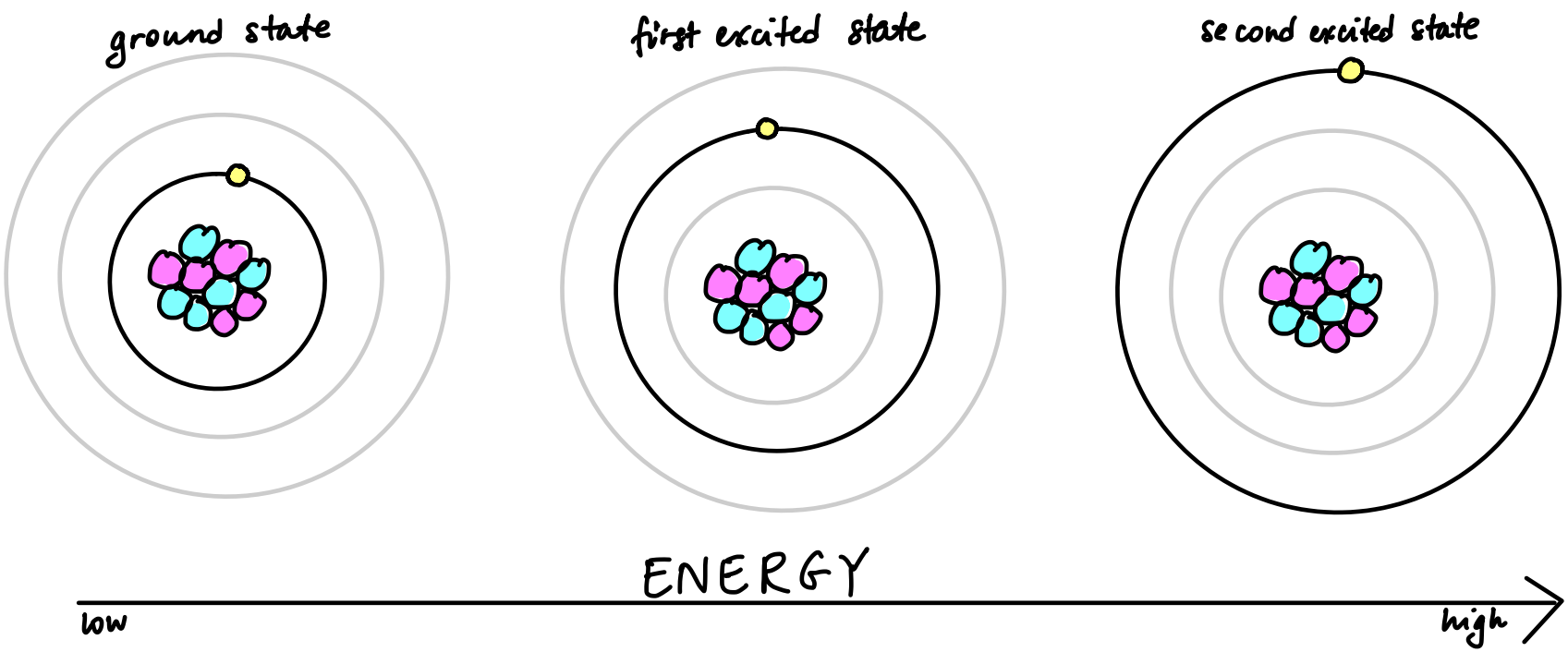
We want to compare the performance of a number of different quantum models.



Variational quantum eigensolver

Energy of a physical system

Determining the energy levels of physical systems is relevant for applications of quantum computing like quantum chemistry. Energy levels allow us to compute useful properties of molecules (e.g., activation energies and reaction rates).



Hamiltonians

In physics, a special operator called a Hamiltonian describes the energy of a system. Hamiltonians are *Hermitian* operators. We will denote them by $\hat{H}$.

$\hat{H}^\dagger = \hat{H}$ H: Hadamard

The Hamiltonian can be used to determine how a quantum system evolves over time. Let $|\psi(0)\rangle$ be the state of the system at time 0. Then at time t,

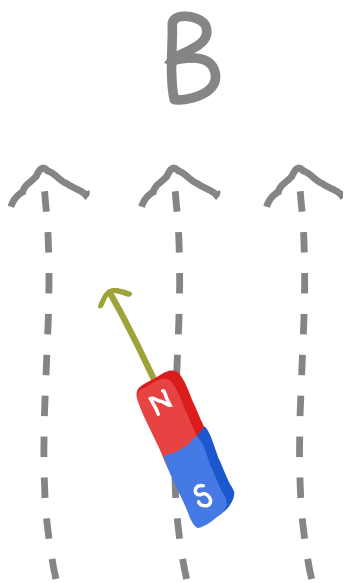
$|\psi(t)\rangle = e^{-i\hat{H}t}|\psi(0)\rangle$

This comes from the Schrödinger equation in physics.

(If $\hat{H}$ is Hermitian, then $e^{-i\hat{H}t}$ is unitary, so we can perform this time evolution on a quantum computer! This is called *Hamiltonian simulation*; we will cover this in the course if time permits)

Hamiltonians

Example: consider a bar magnet in a magnetic field oriented in the Z direction.

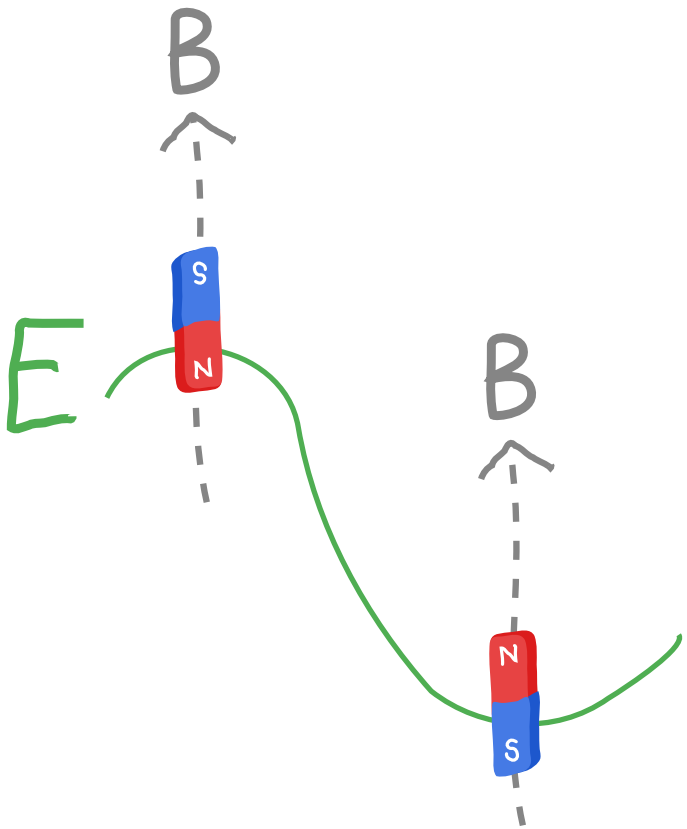


Every orientation of the magnet has an energy associated with it.

Image credit: Xanadu Quantum Codebook node H.3

Mathematical description of energy

Bar magnets like to *align* with the external field.



Aligned configuration is the *ground state* (lowest energy state).

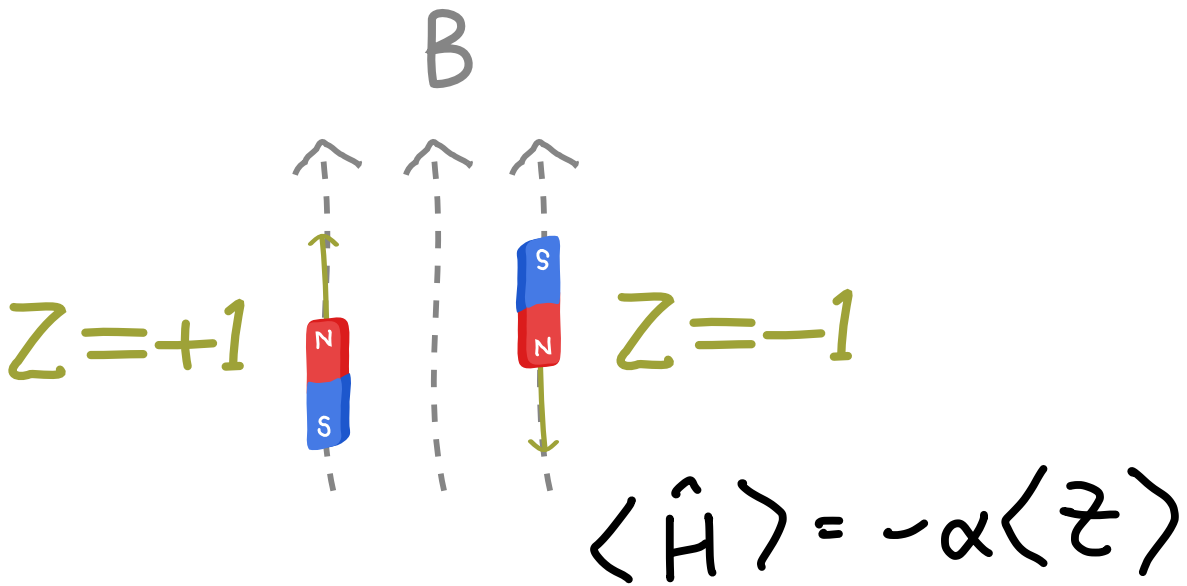
Image credit: Xanadu Quantum Codebook node H.3

Hamiltonians

We can write down a *Hamiltonian* that describes the energy of the system:

$$\hat{H} = -\alpha Z$$

where α is a coefficient that depends on some physical values, and Z is the Pauli Z operator.



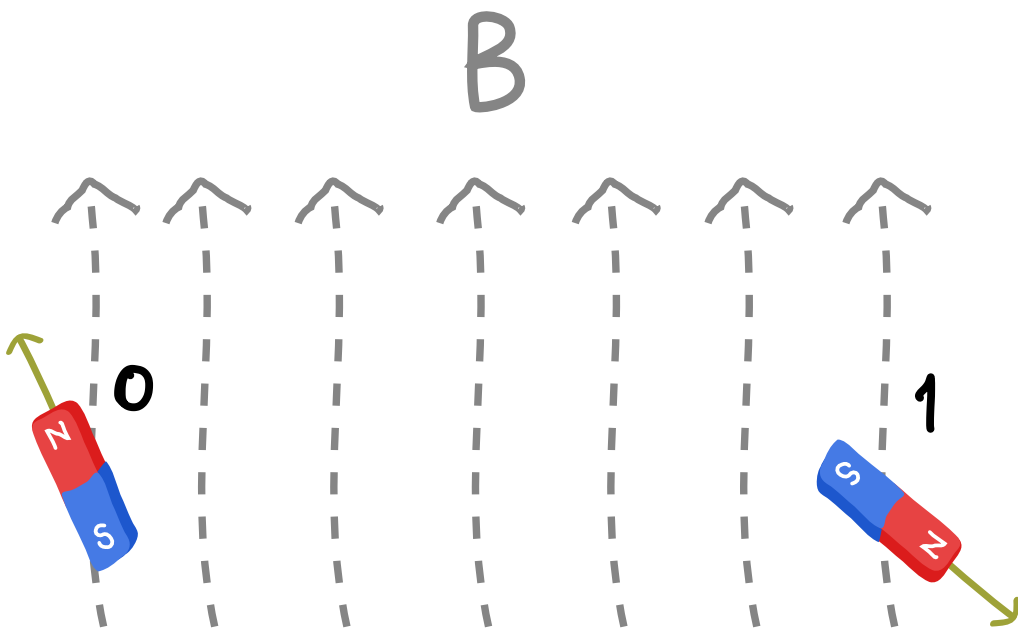
Computing the energy of the system corresponds to *measuring the expectation value of* $\hat{H}$ (which in this case, is just $-\alpha\langle Z\rangle$).

Image credit: Xanadu Quantum Codebook node H.3

Hamiltonians

Hamiltonians can be used to describe systems with multiple parts. For example, two magnets that are far away have the Hamiltonian

$$\hat{H} = \hat{H}_0 + \hat{H}_1 = -\alpha Z_0 - \alpha Z_1$$



The energy of this system is $\langle \hat{H} \rangle = -\alpha \langle Z_0 \rangle - \alpha \langle Z_1 \rangle$

Image credit: Xanadu Quantum Codebook node H.5

Hamiltonians

Two magnets that are close together may have different terms in the Hamiltonian that describe *how they interact*:

$$\hat{H} = \hat{H}_0 + \hat{H}_1 + \hat{H}_{01} = -\alpha Z_0 - \alpha Z_1 + \beta(X_0X_1 + Y_0Y_1 + Z_0Z_1)$$

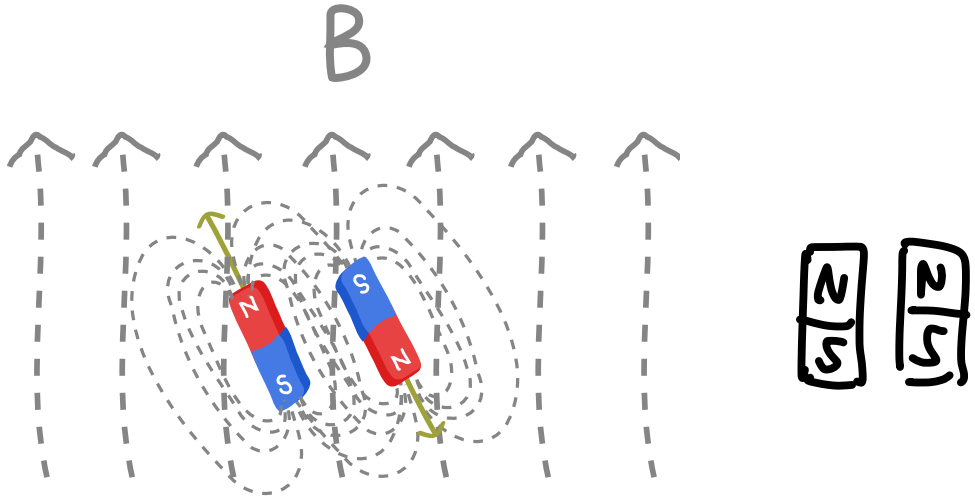


Image credit: Xanadu Quantum Codebook node H.5

Computing the energy of a Hamiltonian

Given an arbitrary $\hat{H}$, how can we compute its energy?

Option: *exact diagonalization* to find its eigenvalues.

Many other purely classical methods too. But we're more interested in the quantum ones.

Computing the energy of a Hamiltonian

Let $\hat{H}$ be a Hamiltonian and $|E\rangle$ an eigenstate with energy E. If $U = e^{-i\hat{H}t}$,

$$\hat{H}|E\rangle = E|E\rangle$$

$$U|E\rangle = e^{-i\hat{H}t}|E\rangle$$

$$= \left(I + (-i\hat{H}t) + \frac{1}{2!}(-i\hat{H}t)^2 + \dots\right)|E\rangle$$

$$= \left(|E\rangle + (-it)\hat{H}|E\rangle + \frac{1}{2!}(-it)^2\hat{H}^2|E\rangle + \dots\right)$$

$$= \left(|E\rangle + (-it)E|E\rangle + \frac{1}{2!}(-it)^2E^2|E\rangle + \dots\right)$$

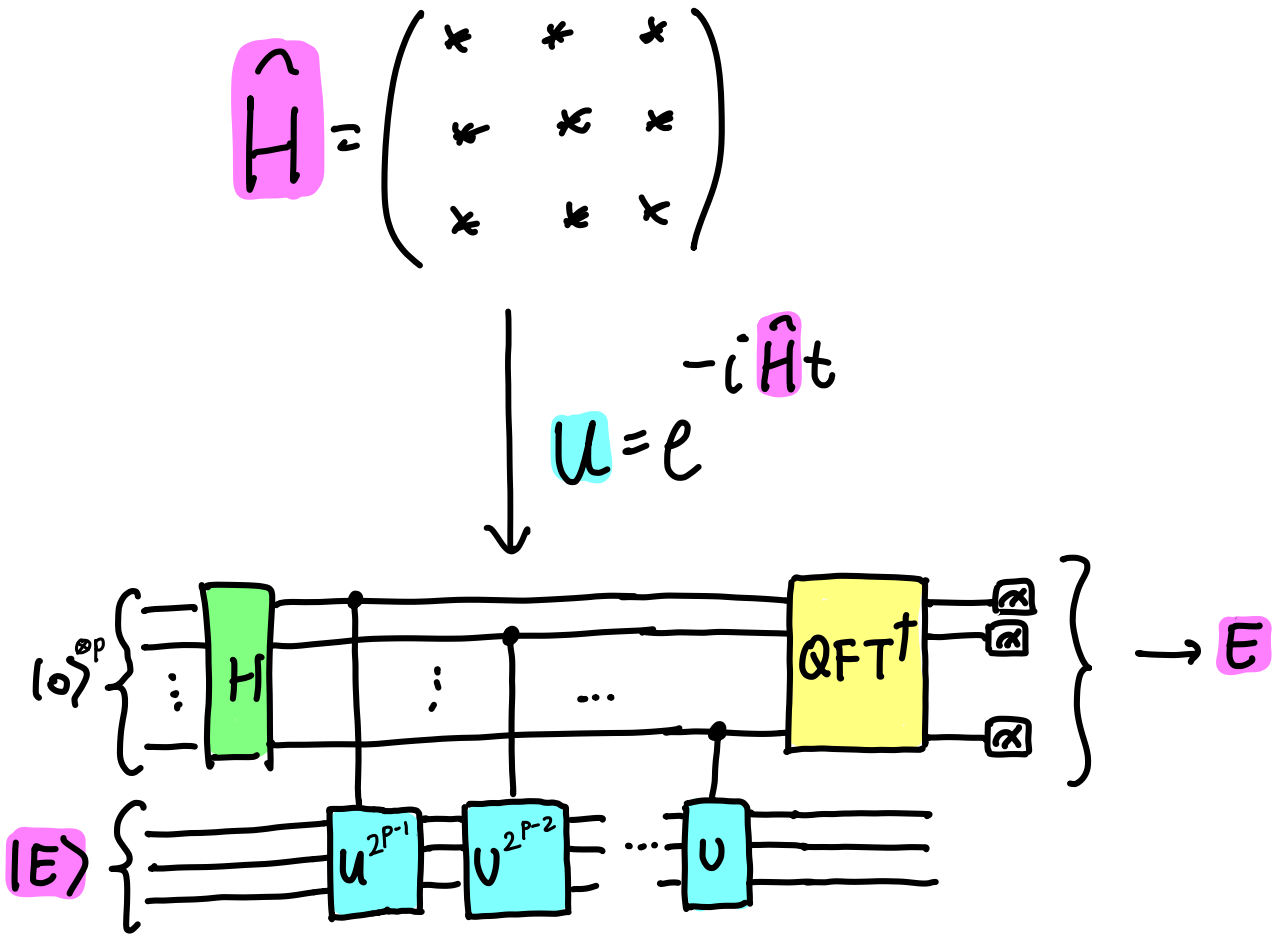
$$= \left(1 + (-iEt) + \frac{1}{2!}(-iEt)^2 + \dots\right)|E\rangle$$

$$= e^{-iEt}|E\rangle$$

So e^{-iEt} is an eigenvalue of U... look familiar?

Computing the energy of a Hamiltonian

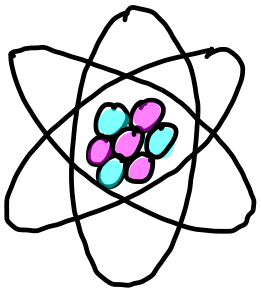
Option: *quantum phase estimation*.



But this is not practical on near-term quantum devices.

Computing the energy of a Hamiltonian

Option: *variational quantum eigensolver* (VQE).



$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$$

$$H = \sum_i c_i P_i$$

CPU

QPU

θ

measurement outcomes

choose:
ansatz $U(\theta)$
initial θ

iterate:
run on qpu w/params θ
compute $\langle H \rangle$ from results
choose new value of θ

$U(\theta)$

estimate of relevant physical quantity

Variational quantum eigensolver (VQE)

The VQE is a variational algorithm that works by:

- training a parametrized circuit to prepare the ground state of a system
- measuring the expectation value of the problem Hamiltonian to compute the energy

The optimization works because of the *variational principle*. If E_g is the ground state energy, then

$$E = \langle \hat{H} \rangle = \langle \Psi | \hat{H} | \Psi \rangle \geq E_g$$

for *any other state* $|\psi\rangle$.

Anatomy of a Hamiltonian

So far, we've been computing expectation values of multi-qubit *Pauli observables* involving X, Y, and Z.

Pauli operators are Hermitian. Even better: they form a basis for the space of Hermitian operators.

It is thus the case that any Hamiltonian can be written as a linear combination of Paulis:

$$\hat{H} = \sum_i c_i P_i$$

$$= c_1 XXX + c_2 XYX + c_3 ZZZ + \ldots$$

Anatomy of a Hamiltonian

Example: $\hat{H} = 0.5Z \otimes Z + 1.5X \otimes X$.

$$\hat{H} = 0.5\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + 1.5\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0.5 & 0 & 0 & 1.5 \\ 0 & -0.5 & 1.5 & 0 \\ 0 & 1.5 & -0.5 & 0 \\ 1.5 & 0 & 0 & 0.5 \end{pmatrix}$$

Anatomy of a Hamiltonian

$\hat{H} = 0.5Z \otimes Z + 1.5X \otimes X$ has 4 eigenvalues/vectors:

$$E_0 = -2 \quad E_1 = -1 \quad E_2 = 1 \quad E_3 = 2$$

$$|E_0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix} \quad |E_1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad |E_2\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ -1 \\ -1 \\ 0 \end{pmatrix} \quad |E_3\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

ground state

Anatomy of a Hamiltonian

How do we compute the expectation value of arbitrary states, e.g., the uniform superposition

$$|\psi\rangle = \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Option: compute analytically from H directly.

$$\langle\hat{H}\rangle = \langle\psi|\hat{H}|\psi\rangle = 1.5$$

But consider that, when we run a computation on a quantum computer, we can measure only in the computational basis. How can we compute the expectation value of an arbitrary Hamiltonian?

Anatomy of a Hamiltonian

Since

$$\hat{H} = \sum_i c_i P_i$$

and inner products / computation of expectation values is linear:

$$\begin{aligned}
\langle \hat{H} \rangle &= \langle \psi | \hat{H} | \psi \rangle \\
&= \langle \psi | \left(\sum_i c_i P_i \right) | \psi \rangle \\
&= \sum_i c_i \langle \psi | P_i | \psi \rangle \\
&= \sum_i c_i \langle P_i \rangle
\end{aligned}$$

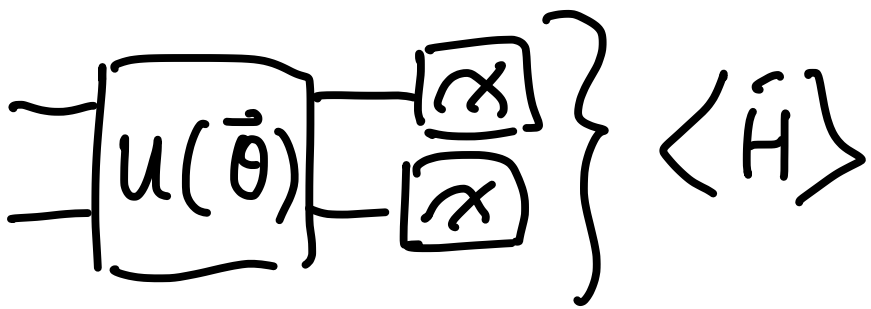
So we can compute expectation values of Hamiltonians simply by computing expectation values of individual Paulis.

VQE example

Suppose we want to *learn* the ground state energy of

$$\hat{H} = 0.5\, Z \otimes Z + 1.5\, X \otimes X$$

We need a parametrized circuit capable of preparing the desired eigenstate *at some values of the parameters*.



for what $\vec{\theta}$ does $\langle \hat{H} \rangle = E_g$?

VQE example

We are lucky here: we know that the ground state is

$$|E_0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}$$

So we need a parametrized circuit $U(\theta)$ that can produce arbitrary combinations of $|01\rangle$ and $|10\rangle$:

$$U(\theta)|00\rangle = \alpha|01\rangle + \beta|10\rangle$$

VQE example

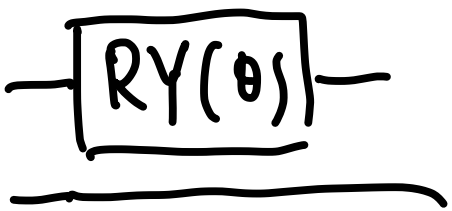
We can make a very tailored ansatz with just a single parameter here. Let

$$U(\theta)|00\rangle = \alpha|01\rangle + \beta|10\rangle$$

Since $|\alpha|^2 + |\beta|^2 = 1$, we can write

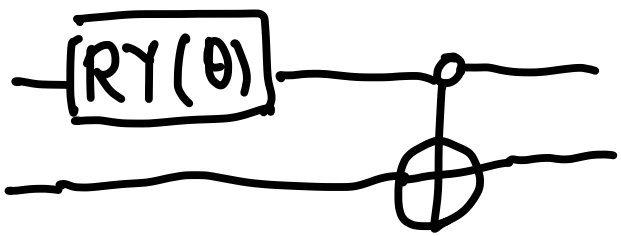
$$U(\theta)|00\rangle = \cos\frac{\theta}{2}|01\rangle + \sin\frac{\theta}{2}|10\rangle$$

What circuit makes states like this?



$$RY(\theta)|00\rangle = \cos\frac{\theta}{2}|00\rangle + \sin\frac{\theta}{2}|10\rangle$$

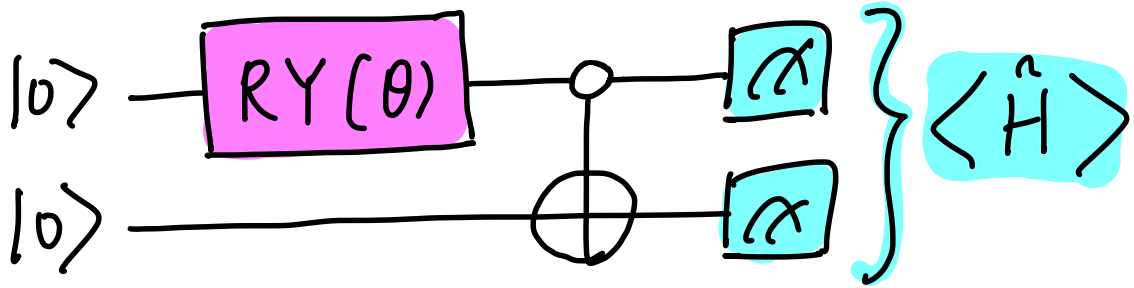
$|01\rangle$?



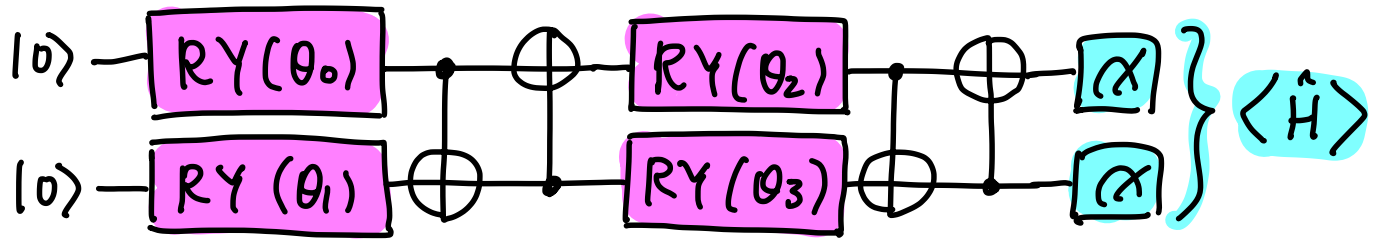
$$\cos\frac{\theta}{2}|01\rangle + \sin\frac{\theta}{2}|10\rangle$$

VQE example

Problem-specific ansatz:



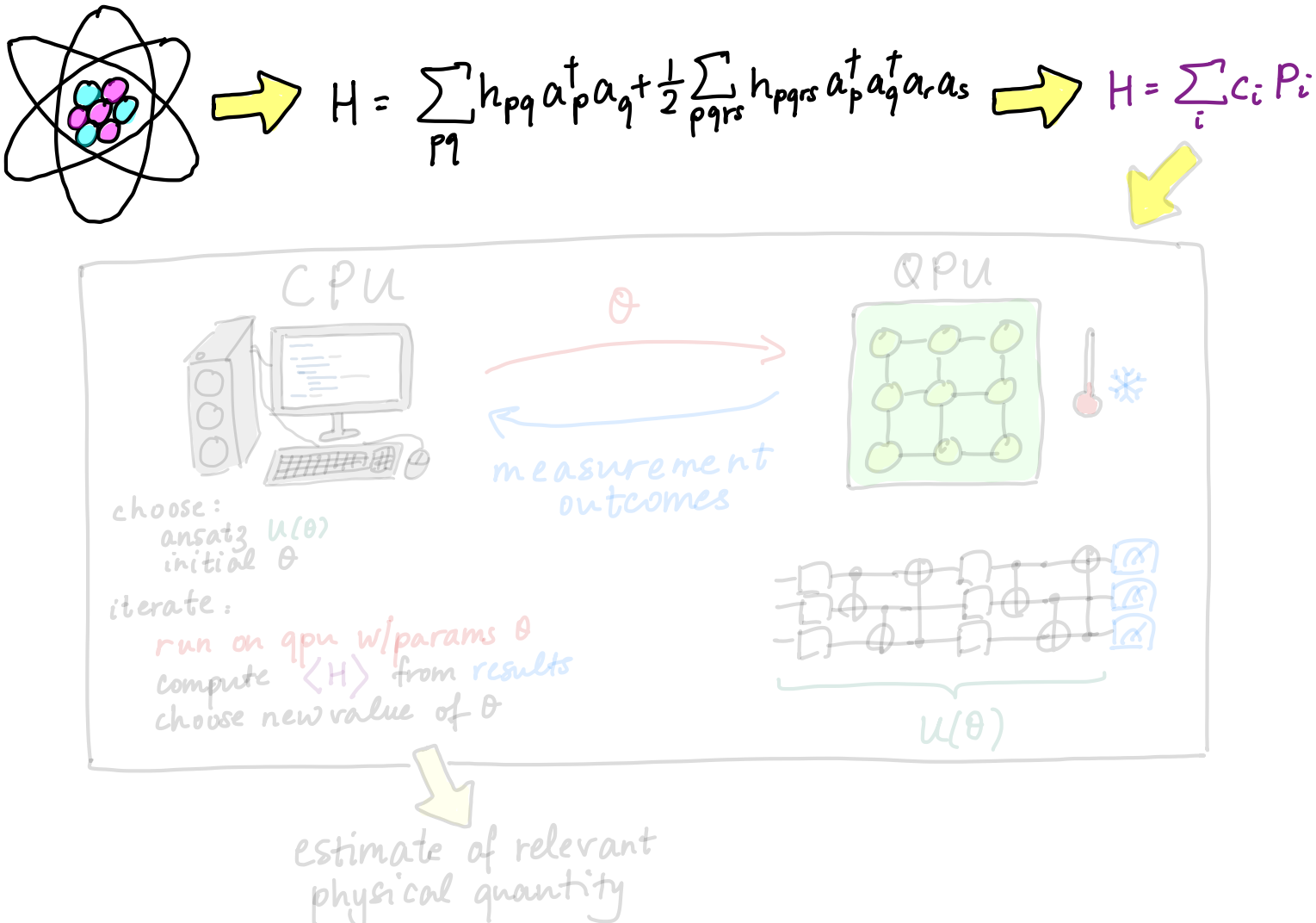
Generic ansatz:



Let's try these out!

VQE example

You might be wondering about this part:



We will discuss this in a future class / assignment 4.

Next time

Content:

- Density matrices, mixed states, and noise channels

Action items:

1. Prototype implementation for project
2. Schedule final project presentation 10am tomorrow
3. Assignment 3

Recommended reading:

- Codebook node H.3 (Hamiltonians)
- VQE demo: `https://pennylane.ai/qml/demos/tutorial_vqe.html`