WE BUILD QUANTUM COMPUTERS

Jan Goetz

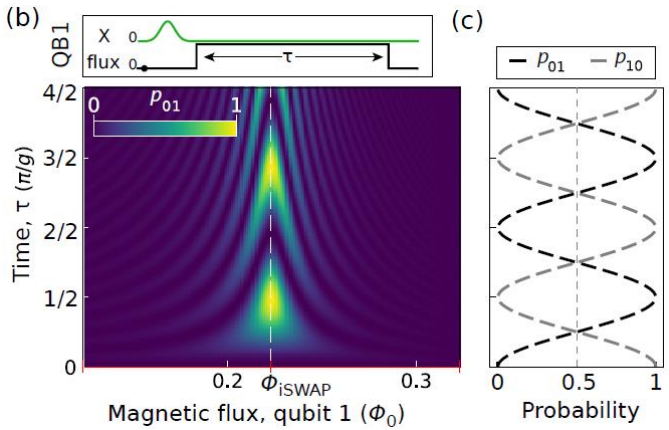
www.meetiqm.com

Lecture notes on PHYS-C0254 Quantum Circuits

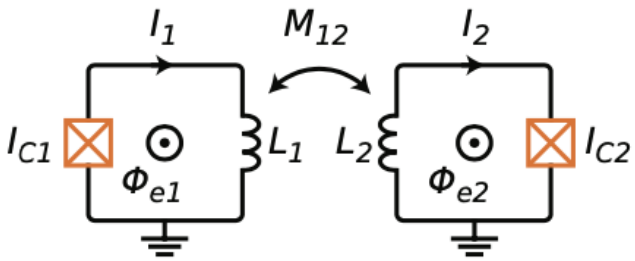
Short recap from last week

10. Two-qubit operations: Architectures for 2-qubit gates 4th DiVincenzo criteria
    a. iSWAP
    b. cPhase
    c. cNot

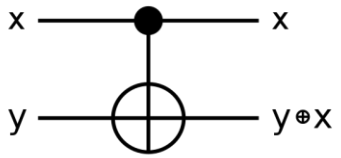
a.



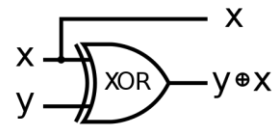
b.



c.



input		output	
x	y	x	y+x
|0⟩	|0⟩	|0⟩	|0⟩
|0⟩	|1⟩	|0⟩	|1⟩
|1⟩	|0⟩	|1⟩	|1⟩
|1⟩	|1⟩	|1⟩	|0⟩



input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

IQM

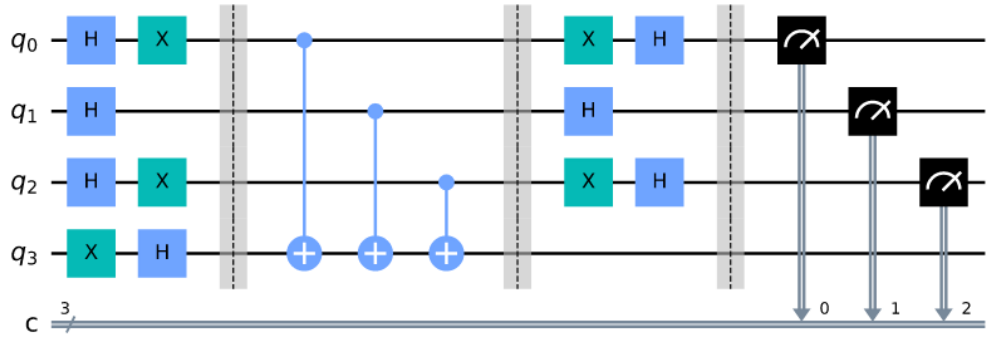
Agenda for lectures 7-12

7. Quantization of electrical networks
    a. Harmonic oscillator: Lagrangian, eigenfrequency
    b. Transfer step: LC oscillator, Legendre transform to Hamiltonian
    d. Quantization of oscillators
8. Superconducting quantum circuits
    a. Qubits: Transmon qubit, Charge qubit, Flux qubit 1st DiVincenzo criteria
    b. Circuit-QED: Rabi model
    c. Rotating Wave approximation: Jaynes-Cummings model
9. Single-qubit operations:
    a. Initialization 2nd DiVincenzo criteria
    b. Readout 5th DiVincenzo criteria
    c. Control:T1, T2 measurements, Randomized benchmarking 3rd DiVincenzo criteria
10. Two-qubit operations: Architectures for 2-qubit gates 4th DiVincenzo criteria
    a. iSWAP
    b. cPhase
    c. cNot
11. Quantum algorithms
    a. Deutsch-Josza Algorithm
    b. Parameterised circuits and VQE
12. Challenges in quantum computing
    a. Scaling
    b. SW-HW gap
    c. Error-correction

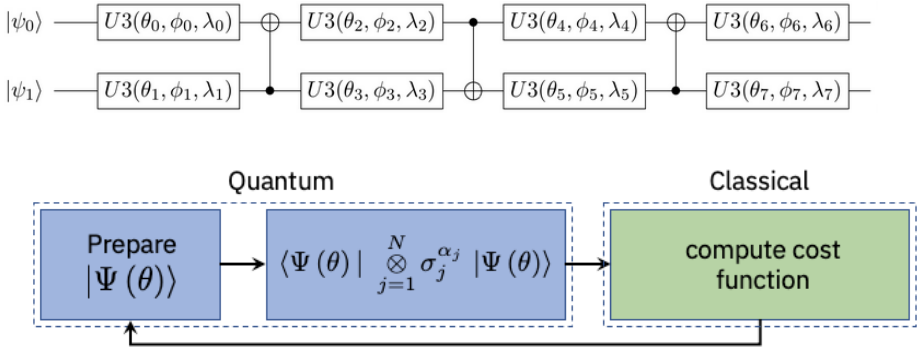
Agenda for today

11. Quantum algorithms
    a. Deutsch-Josza Algorithm
    b. Parameterised circuits and VQE

a.



b.



IQM

General approach: Quantum Algorithms

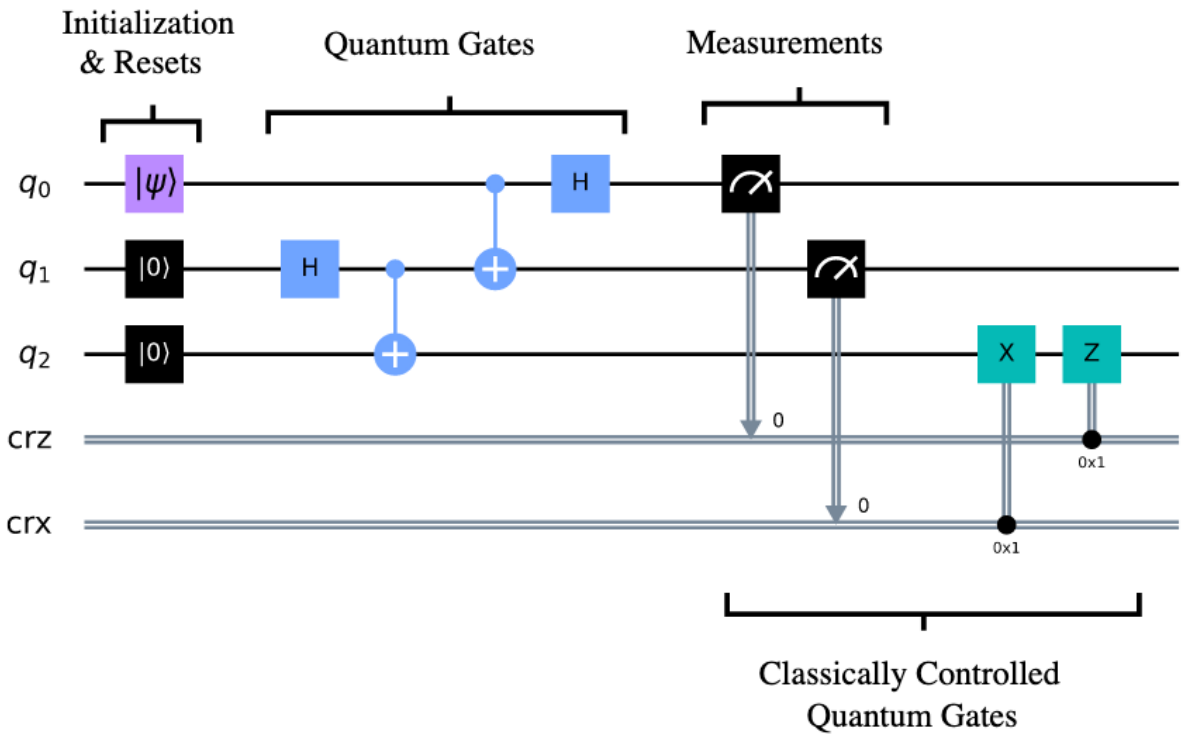
- A suitable combination of initialization, single- & two-qubit gates, as well as readout, compiles a quantum algorithm.

- The goal of using quantum algorithms is to use the two quantum resources superposition and entanglement to speed up the performance of classical algorithms

- There are “pure” quantum algorithms which run completely on the quantum computer and there are “hybrid” quantum algorithms which are executed partly on classical and partly on quantum computers.

Textbook example: Teleportation algorithm

What we will today call a quantum (logical) circuit



2QB gates in the diagram are CNOTs. This is a teleportation circuit that teleports the state of q0 to q2 by generating entanglement between the qubits:

Components:

1. Initialisation
2. The logical circuit that performs the computation
3. Measurement
   Possibly some more gates that depend on measurement outcomes
4. Classical post-processing.

IQM https://qiskit.org/textbook/ch-algorithms/teleportation.html

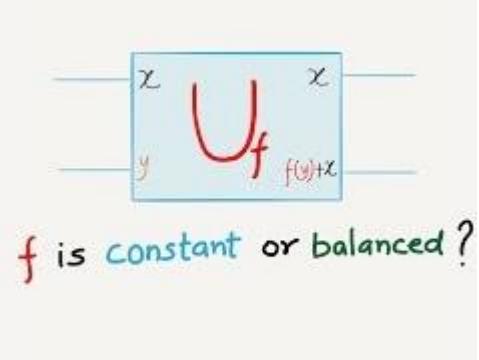
General: The Deutsch-Jozsa algorithm

- The Deutsch-Jozsa algorithm, was the first example of a quantum algorithm that performs better than the best classical algorithm. It showed that there can be advantages to using a quantum computer as a computational tool for a specific problem.

- The Deutsch-Jozsa algorithms determines if a function *f* is balanced or constant [$f(x) = x$].

- In mathematics and computer science, a balanced boolean function is a boolean function whose output yields as many 0s as 1s over its input set. This means that for a uniformly random input string of bits, the probability of getting a 1 is 1/2.

The Deutsch-Jozsa algorithm



We are given a hidden Boolean function f, which takes as input a string of bits, and returns either 0 or 1. We call this function our oracle:

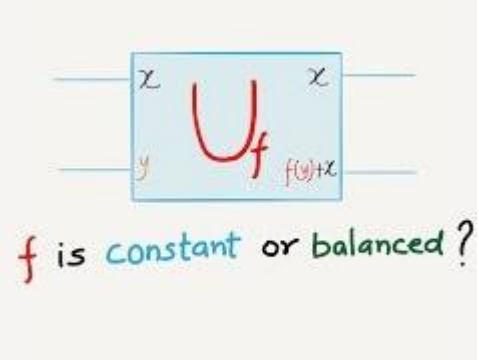
that is: f({x0,x1,x2,...}) → 0 or 1 ,

where xn is 0 or 1.

The property of the given Boolean function is that it is guaranteed to either be balanced or constant. A constant function returns all 0's or all 1's for any input, while a balanced function returns 0's for exactly half of all inputs and 1's for the other half.

IQM https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

The Deutsch-Jozsa algorithm



Classically, in the best case, two queries to the oracle can determine if the hidden Boolean function, f(x), is balanced: e.g. if we get both f(0,0,0,...)→0 and f(1,0,0,...)→1, then we know the function is balanced as we have obtained the two different outputs.

In the worst case, if we continue to see the same output for each input we try, we will have to check exactly half of all possible inputs plus one in order to be certain that f(x) is constant.

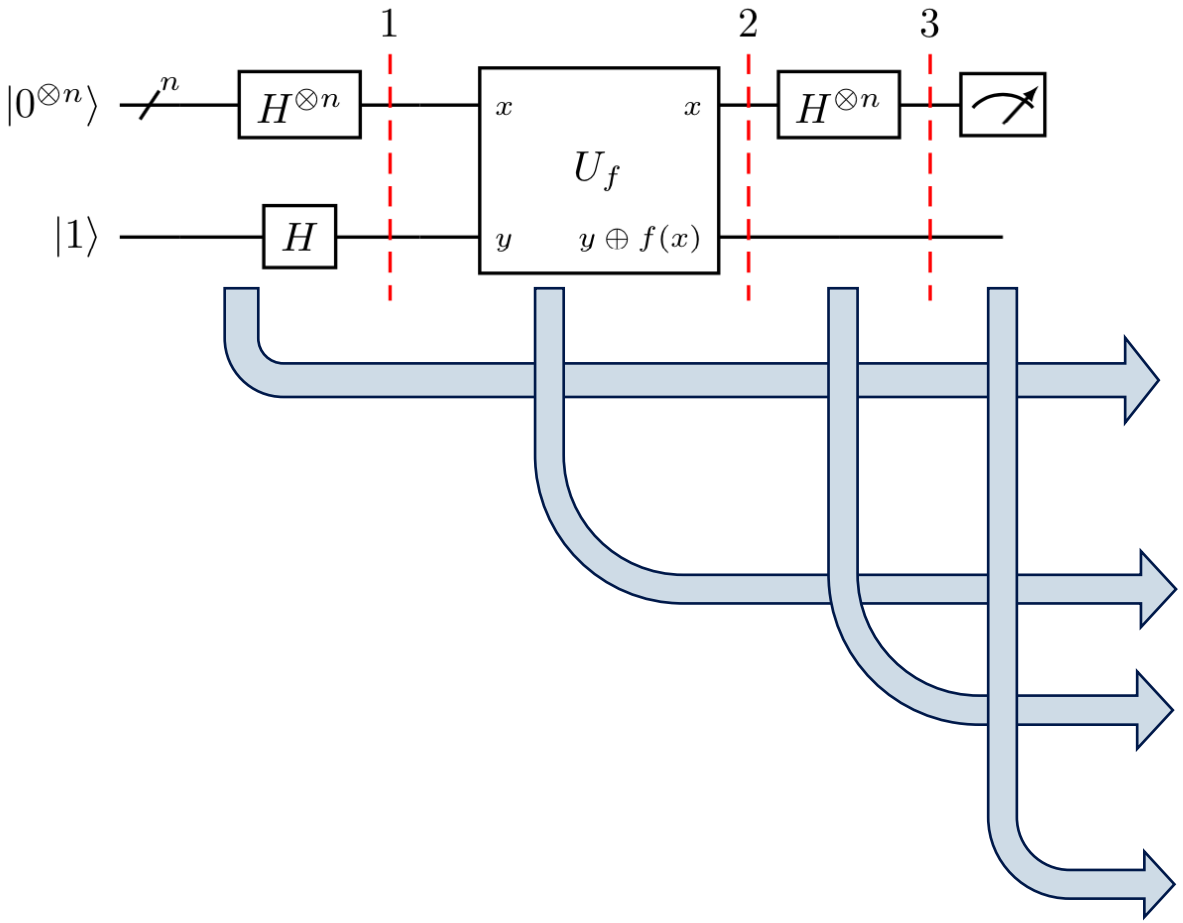
Since the total number of possible inputs is 2n, this implies that we need 2n−1+1 trial inputs to be certain that f(x) is constant in the worst case. For example, for a 4-bit string, if we checked 8 out of the 16 possible combinations, getting all 0's, it is still possible that the 9th input returns a 1 and f(x) is balanced.

Probabilistically, this is a very unlikely event. In fact, if we get the same result continually in succession, we can express the probability that the function is constant as a function of k inputs as:

$$P_{\text{constant}}(k) = 1 - \frac{1}{2^{k-1}} \qquad \text{for } 1 < k \leq 2^{n-1}$$

IQM https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

The Deutsch-Jozsa algorithm



Using a quantum computer, we can solve this problem with 100% confidence after only one call to the function f(x), provided we have the function f implemented as a quantum oracle, which maps the state $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$, where $\oplus$ is addition modulo 2.

Prepare two quantum registers. The first is an n-qubit register initialised to $|0\rangle$, and the second is a one-qubit register initialised to $|1\rangle$. Apply a Hadamard gate to each qubit.

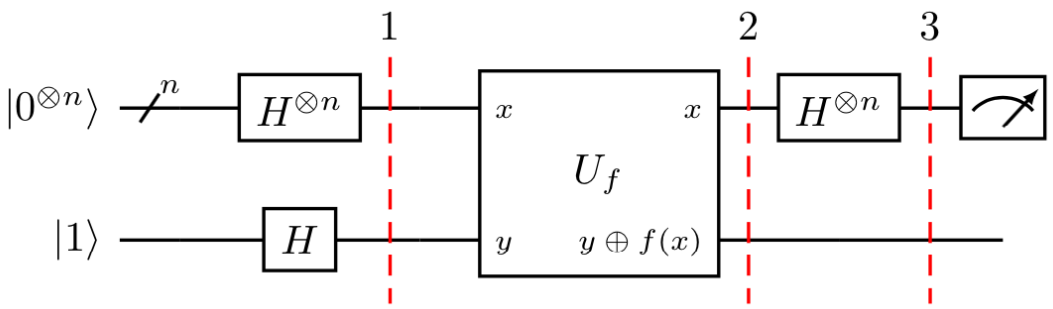
Apply the quantum oracle $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$.

Apply a Hadamard gate to each qubit in the first register.

Measure the first register. Notice that it evaluates to 1 if f(x) is constant and 0 if f(x) is balanced.

IQM https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

The Deutsch-Jozsa algorithm



Why Does This Work?

Case 1: Constant Oracle

When the oracle is *constant*, it has no effect (up to a global phase) on the input qubits, and the quantum states before and after querying the oracle are the same. Since the H-gate is its own inverse, in Step 2 we reverse Step 1 to obtain the initial quantum state of |00…0⟩ in the first register.

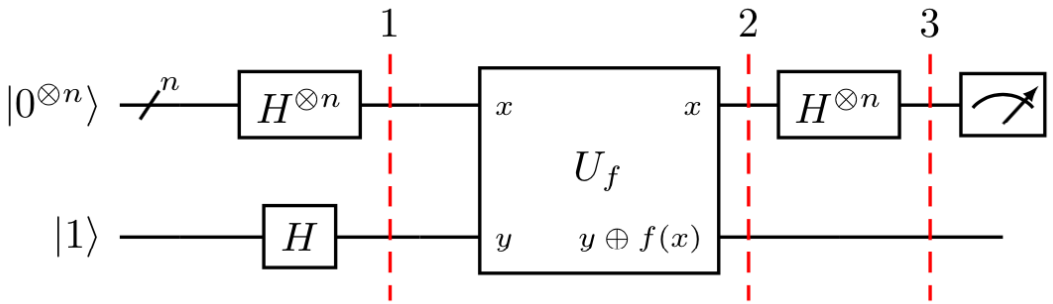
Case 2: Balanced Oracle

After step 1, our input register is an equal superposition of all the states in the computational basis. When the oracle is *balanced*, phase kickback adds a negative phase to exactly half these states.

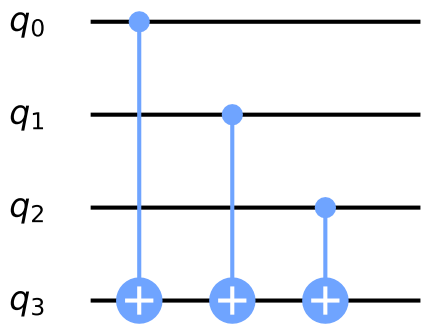
The quantum state after querying the oracle is orthogonal to the quantum state before querying the oracle. Thus, in Step 4, when applying the H-gates, we must end up with a quantum state that is orthogonal to |00…0⟩. This means we should never measure the all-zero state.

IQM https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

The Deutsch-Jozsa algorithm



How does an oracle look in reality?



Let's see some different ways we can create a quantum oracle.

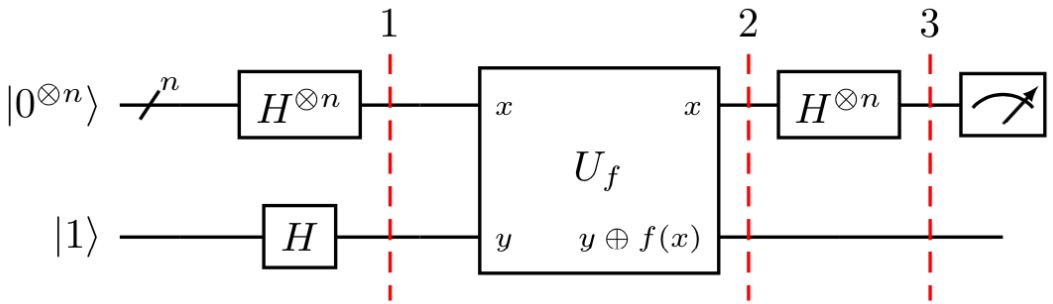
For a constant function, it is simple:
1. if f(x) = 0, then apply the I gate to the qubit in register 2.
2. if f(x) = 1, then apply the X gate to the qubit in register 2.

For a balanced function, there are many different circuits we can create. One of the ways we can guarantee our circuit is balanced is by performing a CNOT for each qubit in register 1, with the qubit in register 2 as the target.

Input states that output 0	Input States that output 1
000	001
011	100
101	010
110	111

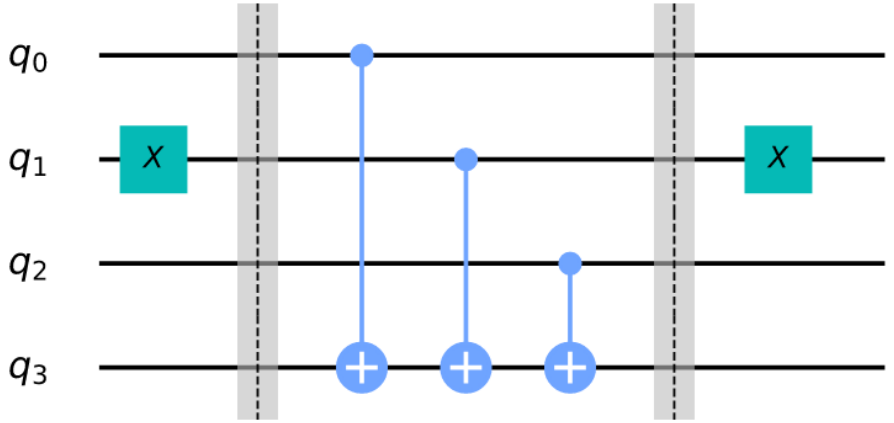
IQM https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

The Deutsch-Jozsa algorithm



How does an oracle look in reality?

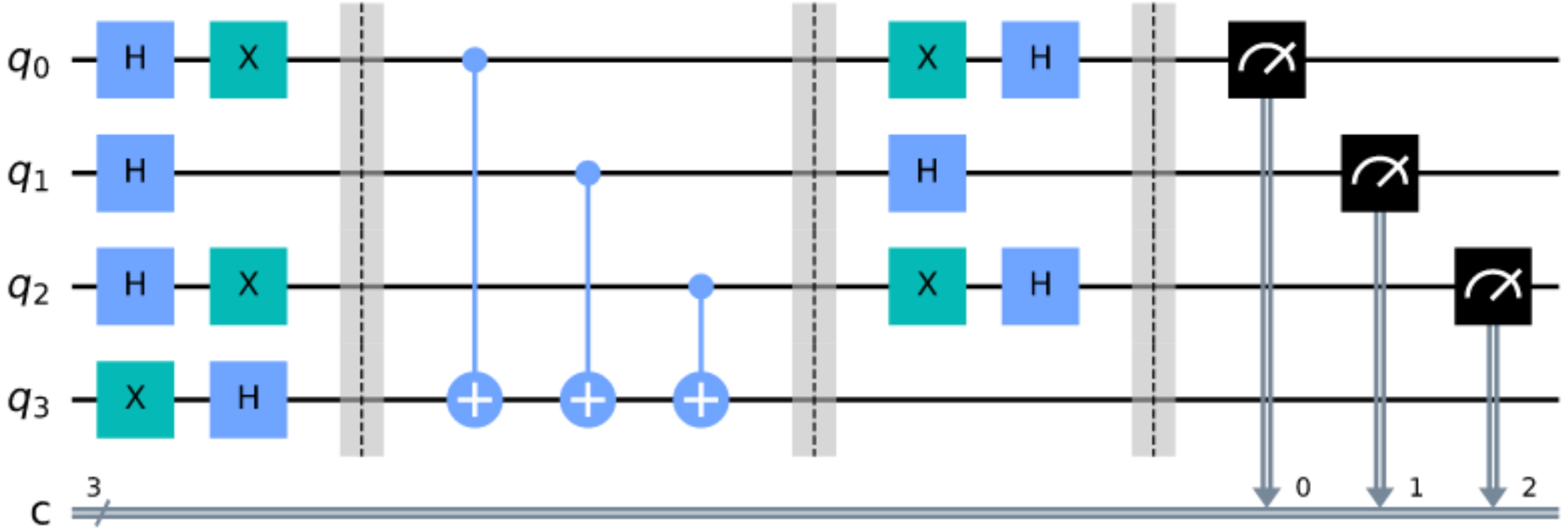
We can change the results while keeping them balanced by wrapping selected controls in X-gates. For example, see the circuit and its results table below:



Input states that output 0	Input states that output 1
001	000
010	011
100	101
111	110

The Deutsch-Jozsa algorithm

This is the Deutsch-Jozsa algorithm that determines if a Boolean function is balanced or constant in one step (it is guaranteed to be either one)

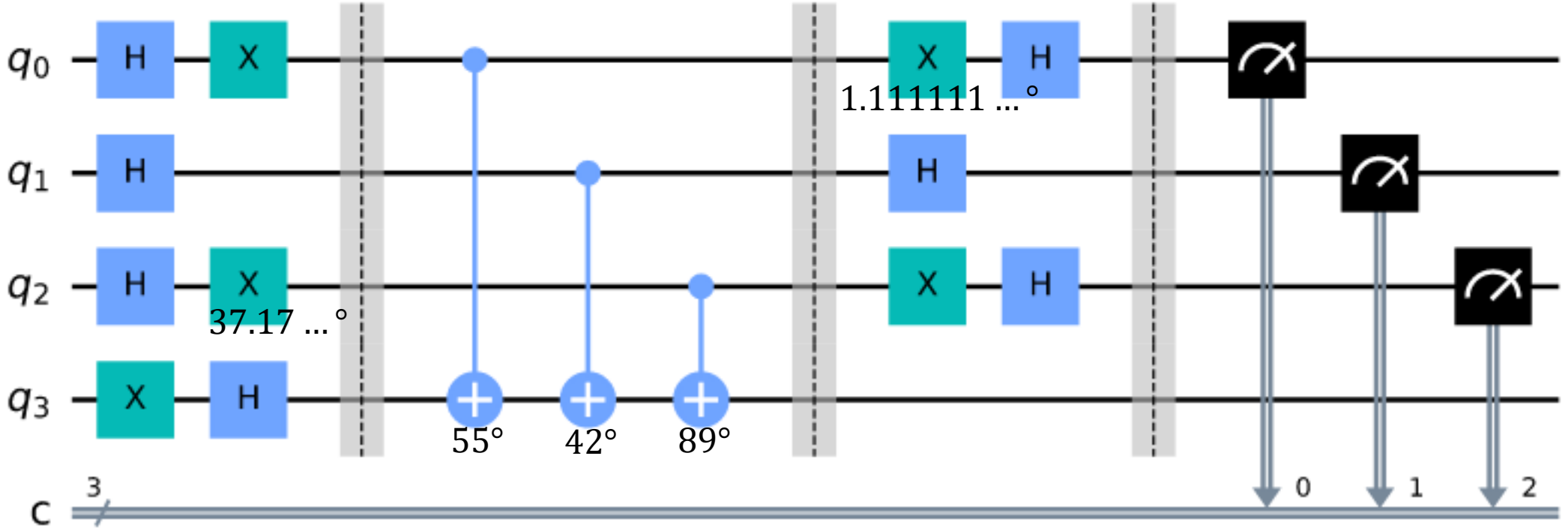


https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

The Deutsch-Jozsa algorithm

We learned in the last class that we can do arbitrary interaction times by tuning qubit frequencies, so we have access to a continuum of gates.

This is one way and some algorithms like this also exist, but continuous gates make the design of these algorithms non-intuitive. CNOT and iSWAP have clear meanings, but with arbitrary angles it becomes harded to conceptualise what's going on.

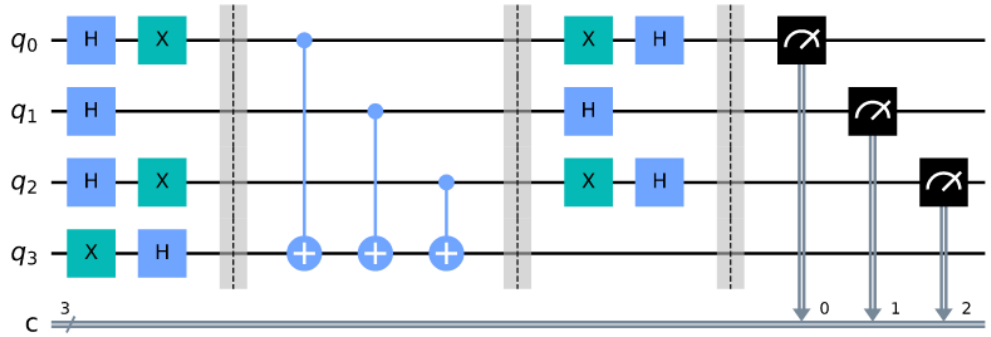


IQM https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

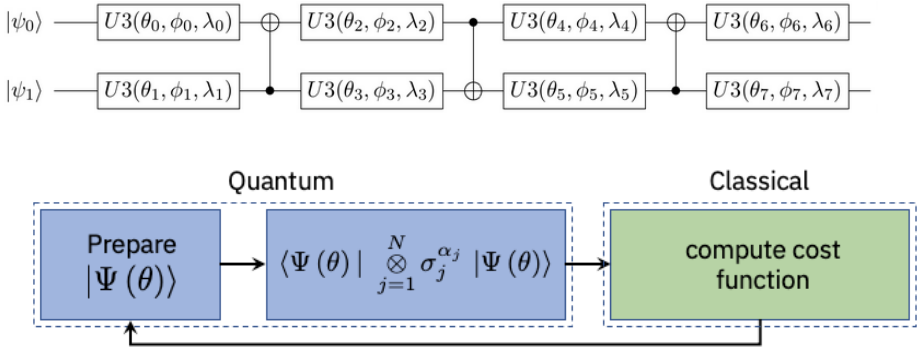
Agenda for today

11. Quantum algorithms
    a. Deutsch-Josza Algorithm
    b. Parameterised circuits and VQE

a.



b.



IQM

The Rayleigh-Ritz variational principle

Almost a trivial statement: Any state will have an energy greater or equal to the ground state (by definition).

The variational principle is is the method we have learnt for solving the helium atom spectrum

$$\langle \Psi(\theta) | \, H \, | \Psi(\theta) \rangle = \lambda_\theta \geq \lambda_{gs}$$

Procedure:
1. Pick a trial state (Ansatz) with parameters θ
2. Minimize the energy analytically

Let’s remove the human

Thinking in terms of continuus values of angles is a tough job for a algorithm designers, so we get rid of the humans and get a machine instead

The variational quantum eigensolver (VQE)

The VQE is a modern quantum-classical hybrid algorithm that attempts to combine the best parts of both classical and quantum computing.

The VQE computes the ground state energy of a Hamiltonian, just like in the case of Helium.

On a quantum computer we only have access to measurements of Pauli strings, so we must first prepare our Hamiltonian (such as the Helium one) as a sum of Pauli strings.

We then get the energy of the state as the sum of the expectation values of the Pauli strings.

This ‘qubitization’ of the Hamiltonian can be done for all Hamiltonians, but some are a more natural fit than others.

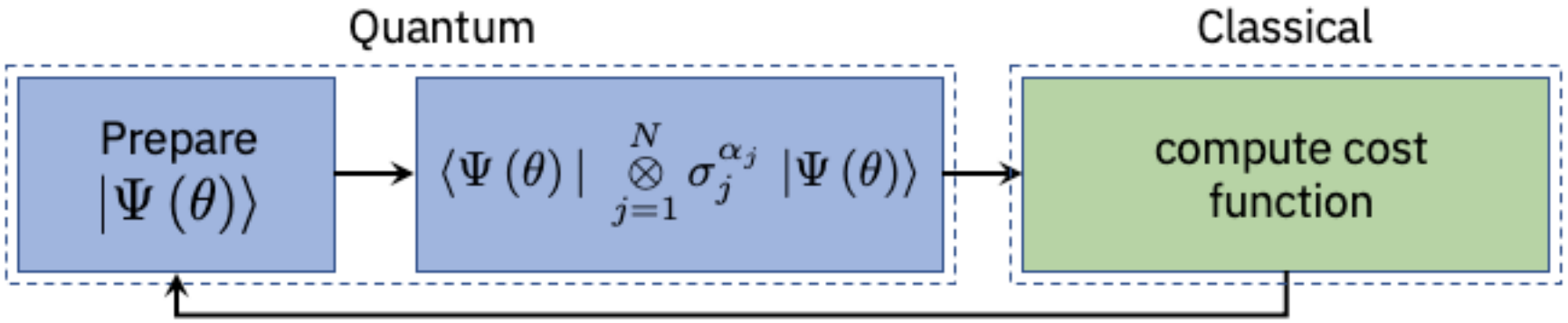
Spin problems are natively in this form. For example: Transverse field Ising model:

$$H = -J \sum^{N} \left(\hat{\sigma}_j^z \hat{\sigma}_{j+1}^z + g \hat{\sigma}_j^x \right)$$

 Short intro to VQE: https://medium.com/@lana.bozanic/the-variational-quantum-eigensolver-efb8fab14c85

The variational quantum eigensolver (VQE)

- VQE Procedure:
  1. Pick a circuit Ansatz
  2. Initialize the parameters
  3. Compute the energy of the state
  4. Minimize the energy by varying the parameters
  5. ???
  6. Profit!

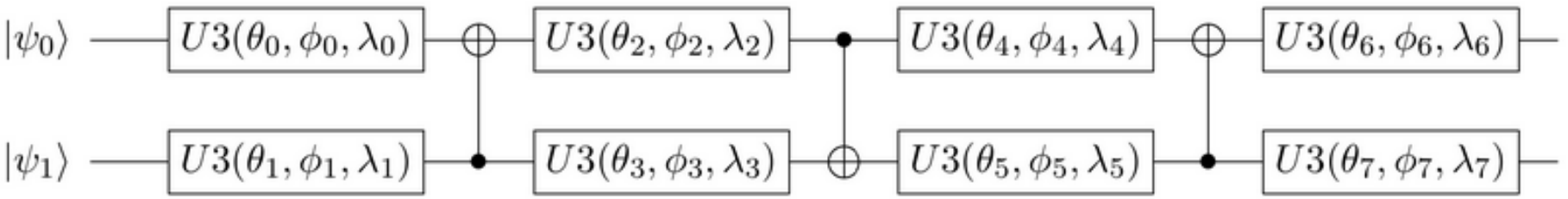


The variational quantum eigensolver (VQE)

Here we only prepare initial states with parameters that a classical computer can vary. The parameters are the angles in the gates that we can tune by controlling the qubit frequencies.

In the example circuit only the single qubit rotations are parameterised, but also the two qubit gates can be parameterised.

Superconducting qubit platforms enable us to implement parameterised gates directly in hardware.

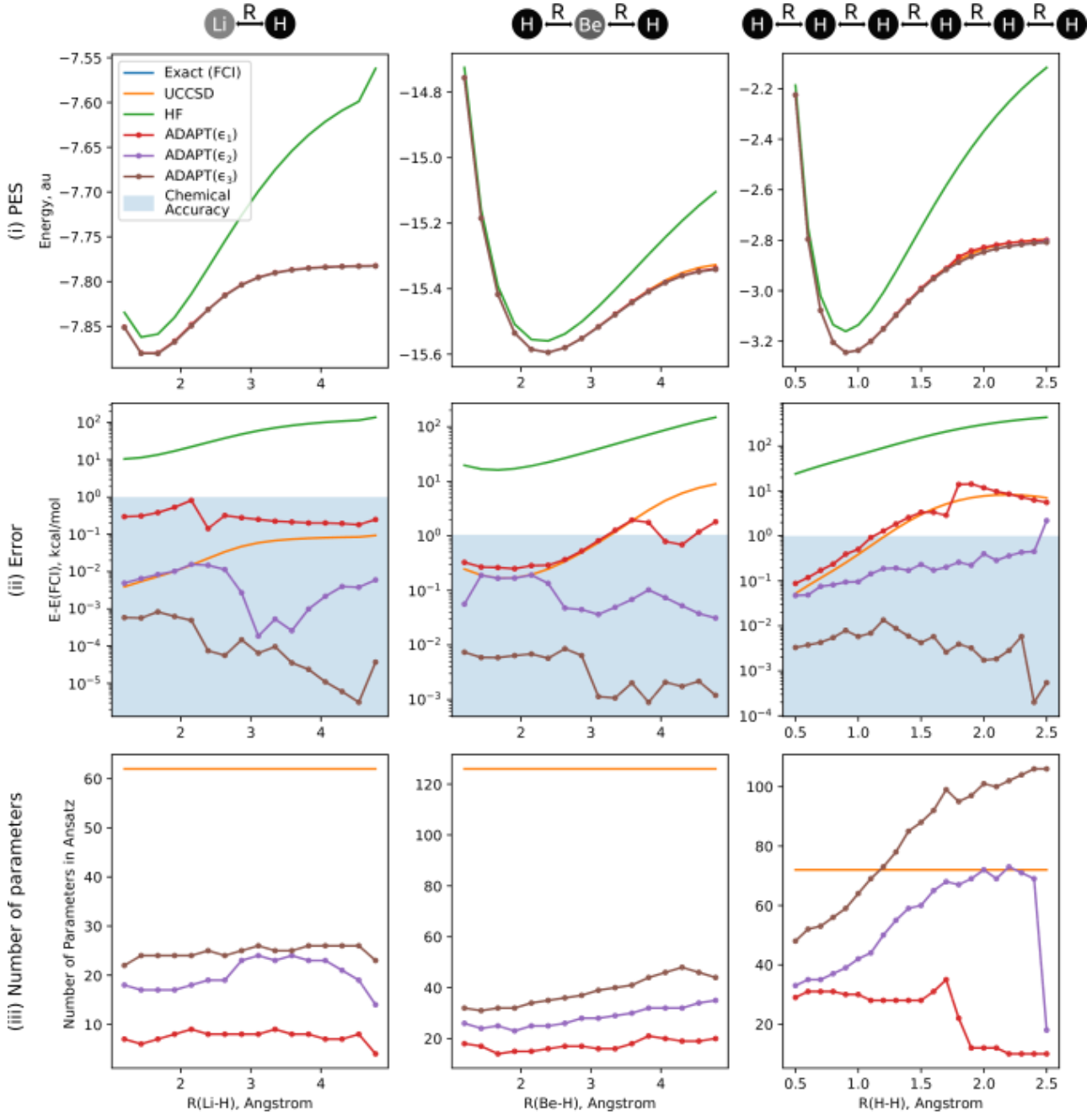


Applications of the VQE

Quantum chemistry applications are the most prominent use-case of the VQE

Often chemists are interested in the potential energy surface (PES) of molecules as they are deformed.

Top: Energy computed with different Ansatze
Middle: error from the exact solution
Bottom: Number of parameters in the Ansatz



IQM Reviews: https://arxiv.org/pdf/2101.08448.pdf, https://arxiv.org/pdf/2012.09265v1.pdf

Challenges and open questions

To use the VQE we need to pick an Ansatz, which is an arbitrary design decision

1. Problem-inspired Ansätze can be generated from the Hamiltonian in some cases
2. Hardware-efficient Ansätze are shallow circuits that aim at creating a lot of entanglement

We are trying to ‘steer’ the QC to the ground state, so if the Ansatz is poorly chosen, we might never get there

When the qubit count grows, the relative ‘size’ of the ground-state manifold becomes exponentially small. Our initial state will on average have an exponentially small overlap with the ground state, so the gradients of our parameters will be exponentially small. Then the convergence of the optimization is very slow.

This is called the Barren Plateau problem.

The search landscape might also have lots of local minima, and finding the global one can be hard. Again, good Ansatz selection is key.

Good initialization and Ansatz design are current research topics in the field.

 Reviews: https://arxiv.org/pdf/2101.08448.pdf, https://arxiv.org/pdf/2012.09265v1.pdf

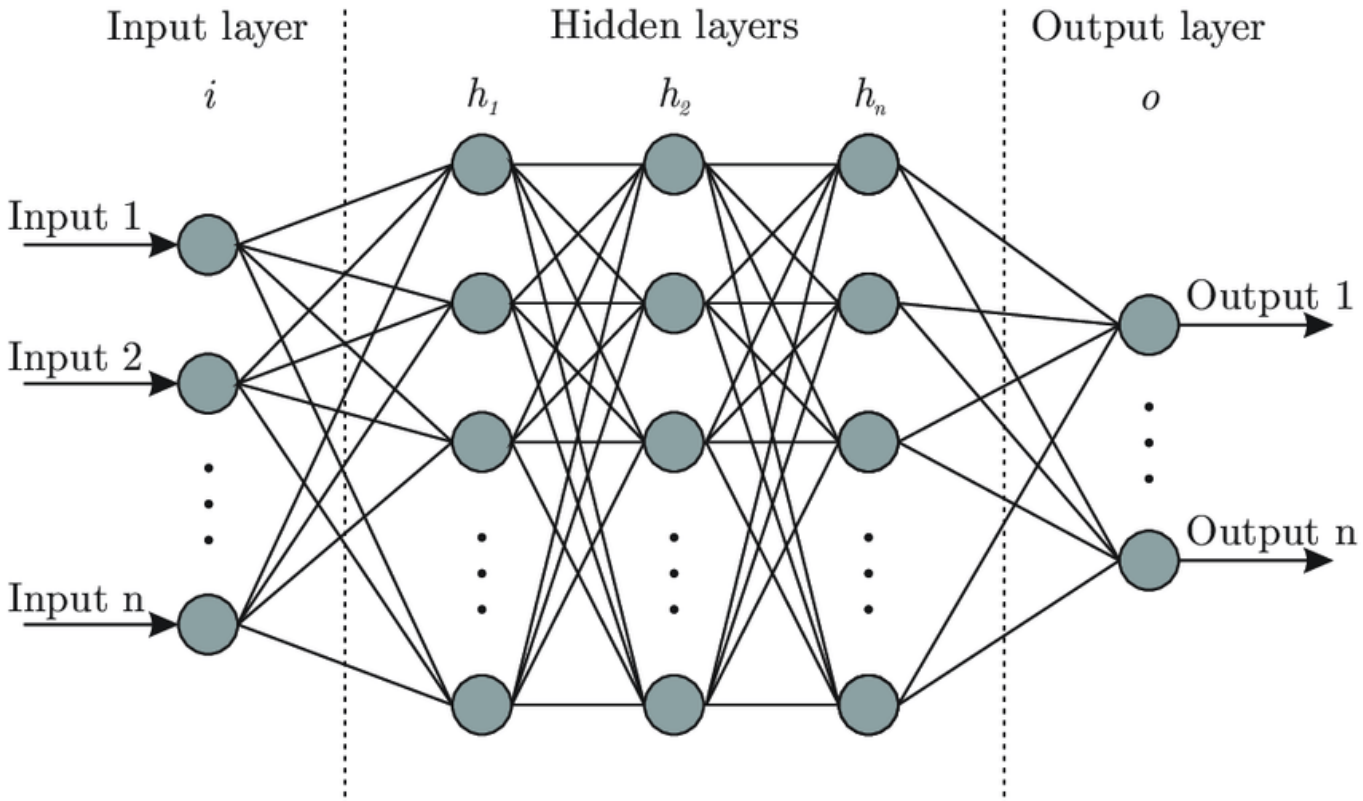
Analogue with Deep learning

The VQE can be thought of in analogue with how neural networks and deep learning work

Each gate represents a neuron
Each parameter the weight

Then we optimize the weights until the output minimizes the cost function

In fact, both deep learning and VQEs fall under the broad category of differential programming. To learn more, you can read the tutorials at www.pennylane.ai



IQM

Review: Quantum Algorithms

- A suitable combination of initialization, single- & two-qubit gates, as well as readout, compiles a quantum algorithm.

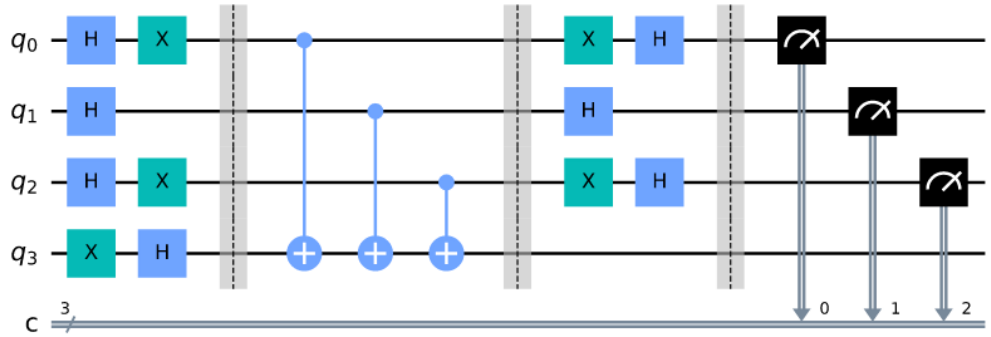
- The goal of using quantum algorithms is to use the two quantum resources superposition and entanglement to speed up the performance of classical algorithms

- There are “pure” quantum algorithms which run completely on the quantum computer and there are “hybrid” quantum algorithms which are executed partly on classical and partly on quantum computers.

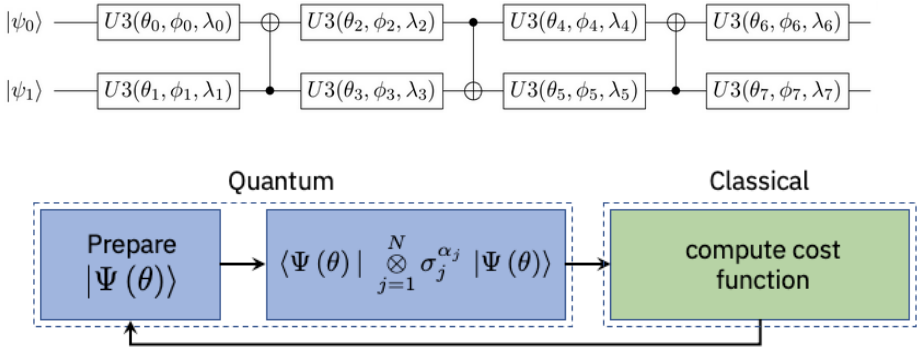
Agenda for today

11. Quantum algorithms
    a. Deutsch-Josza Algorithm
    b. Parameterised circuits and VQE

a.



b.



IQM