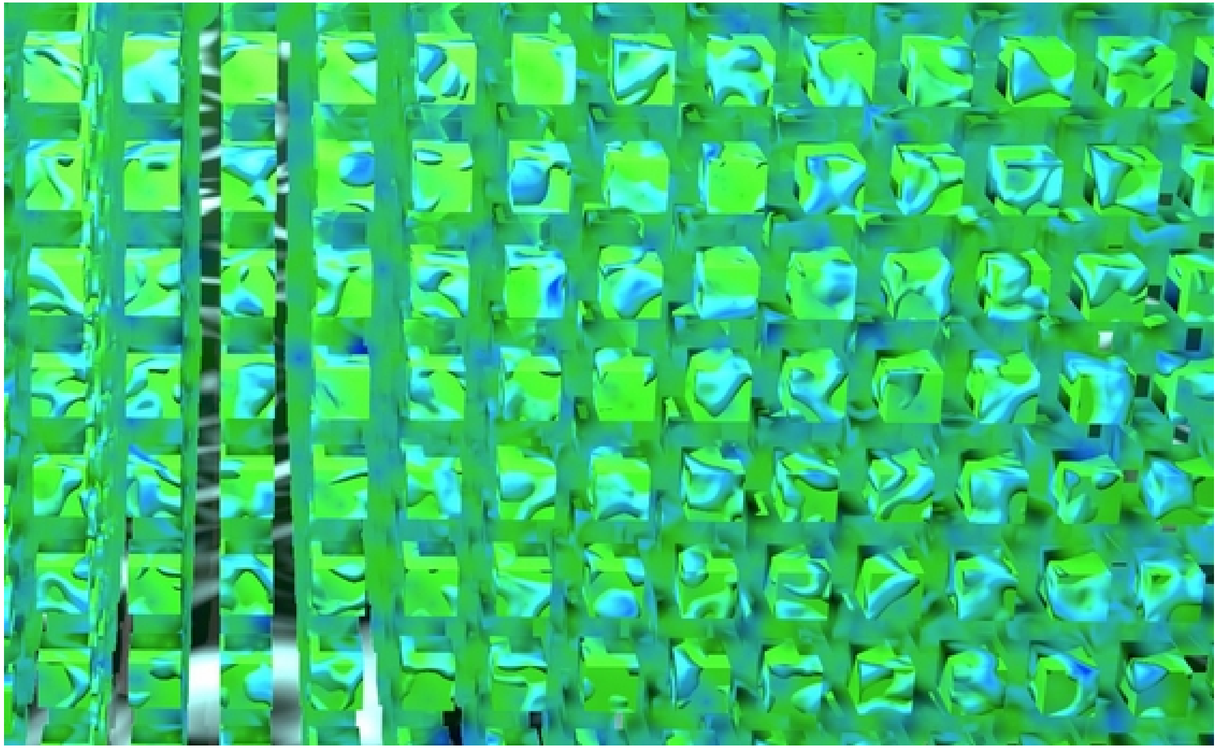
Quantum computing for quantum materials



May 26th 2023

Today’s plan

- Entanglement in many-body wavefunctions
- Basics of quantum circuits
- The variational quantum eigensolver
- Quantum machine learning

Previously, in session 8: Two magnetic Hamiltonians

The Heisenberg dimer

$$\mathcal{H} = \vec{S}_0 \cdot \vec{S}_1$$

$$|GS\rangle = \frac{1}{\sqrt{2}}[|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle]$$

Entangled

The Ising dimer

$$\mathcal{H} = S_0^z S_1^z$$

$$|GS\rangle = |\uparrow\downarrow\rangle$$

Not entangled

How do we distinguish between the two in general?

Entangled states

If a state can be written as

$$|\Psi\rangle = |\Psi_A\rangle \otimes |\Psi_B\rangle$$

Then we say that it is not entangled

Exercise: is the following state entangled or not?

$$|\Psi\rangle = |\uparrow\uparrow\rangle + |\uparrow\downarrow\rangle$$

Entangled states

If a state can be written as

$$|\Psi\rangle = |\Psi_A\rangle \otimes |\Psi_B\rangle$$

Then we say that it is not entangled

Exercise: is the following state entangled or not?

$$|\Psi\rangle = |\uparrow\uparrow\rangle + |\uparrow\downarrow\rangle + |\downarrow\downarrow\rangle + |\uparrow\uparrow\rangle$$

Entangled states

If a state can be written as

$$|\Psi\rangle = |\Psi_A\rangle \otimes |\Psi_B\rangle$$

Then we say that it is not entangled

Exercise: is the following state entangled or not?

$$|\Psi\rangle = |\uparrow\uparrow\rangle - |\uparrow\downarrow\rangle$$

The entanglement entropy

Define the density matrix $$\rho = |\psi\rangle\langle\psi|$$

Trace over one subsystem (reduced density matrix) $$\rho_A = \mathrm{Tr}_B(\rho)$$

Defining the entropy of the state $$S_A = -\mathrm{Tr}(\rho_A \log \rho_A)$$

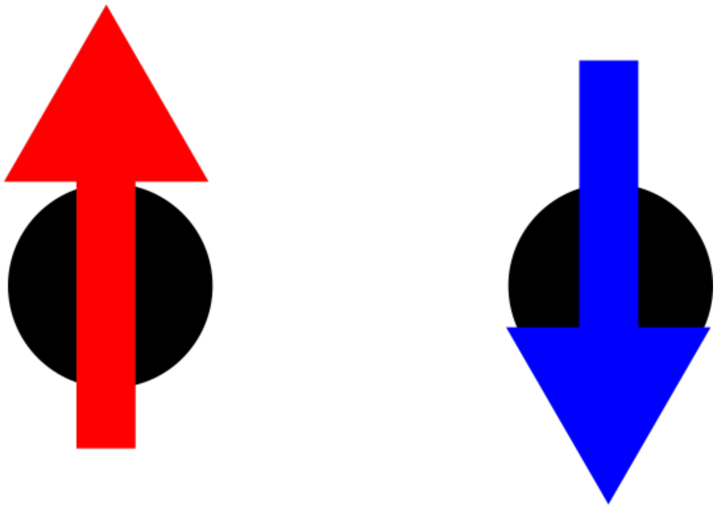
Exercise: what is the reduced density matrix of

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow_A\uparrow_B\rangle + |\uparrow_A\downarrow_B\rangle \qquad |\Psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow_A\downarrow_B\rangle - |\downarrow_A\uparrow_B\rangle$$

Quantum gates

Real pairs of spins

Imagine that you have real systems with spins you can control



What are the fundamental unitary operators you could use?

Pauli X gate

- Acting on a general qubit state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$
$$X|\psi\rangle = \alpha|1\rangle + \beta|0\rangle = \beta|0\rangle + \alpha|1\rangle$$

- It is its own inverse

$$XX = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Hadamard gate

- Acts on a single qubit

Dirac notation

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Unitary matrix

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

No classical equivalent

- One of the most important gates for quantum computing

Pauli Y gate

- Acts on a single qubit

Dirac notation Matrix representation circuit representation

$$|0\rangle \to i|1\rangle, \quad |1\rangle \to -i|0\rangle \quad \boldsymbol{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Gate with no classical equivalent

CNOT gate

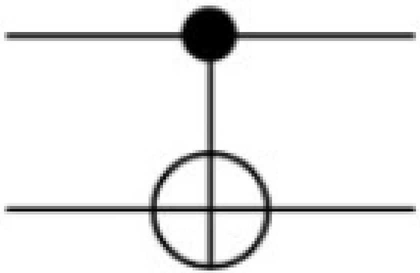
Controlled NOT gate

Acts in two qubits

Matrix representation

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Circuit representation



Approximating unitaries

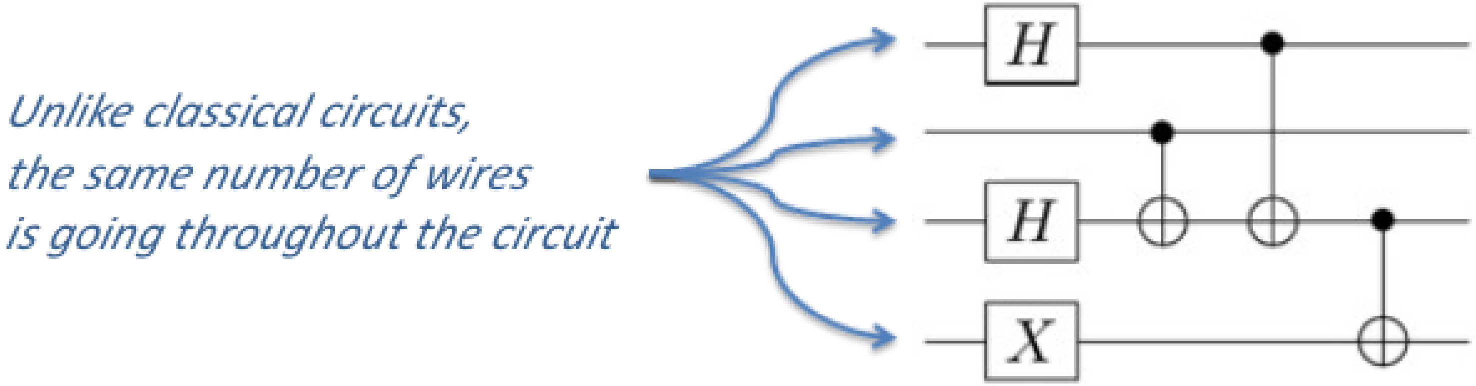
Generic unitaries can be approximated to arbitrary precision with a set of gates

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Exercise: how do you write the X gate in terms of H and S?

Quantum circuit

Gates can be arranged to form a quantum circuit



The variational quantum eigensolver

The quantum many-body problem

Let us go back to a simple many-body problem

$$\mathcal{H} = \sum_{ij} J_{ij} \vec{S}_i \cdot \vec{S}_j$$

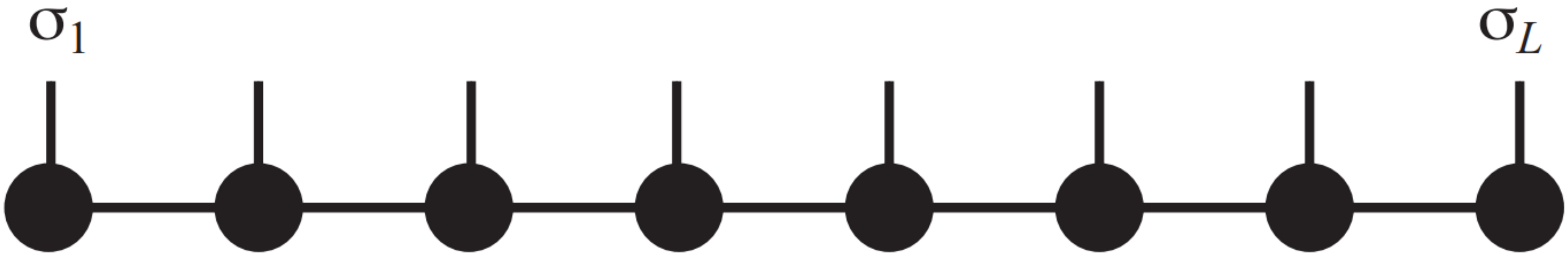
A typical wavefunction is written as

$$|\Psi\rangle = \sum c_{s_1, s_2, \ldots, s_L} |s_1, s_2, \ldots s_L\rangle$$

We need to determine in total 2^L coefficients

Is there an efficient way of storing so many coefficients?

Storing wavefunctions with matrix product states



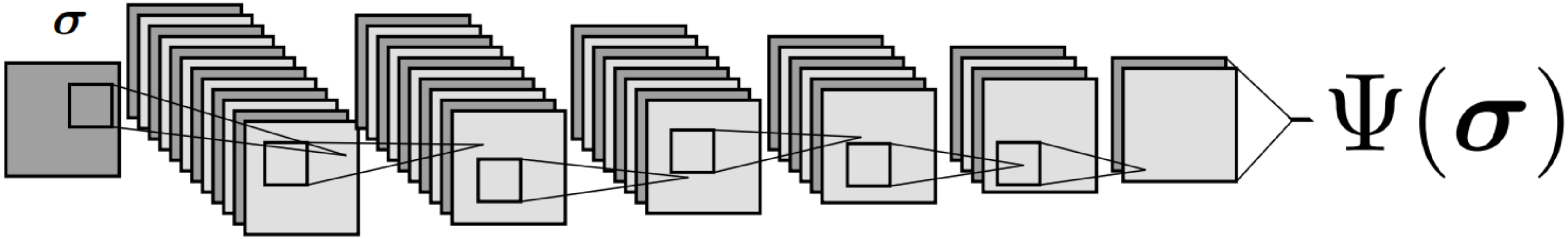
$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} M^{\sigma_1} \dots M^{\sigma_L} |\boldsymbol{\sigma}\rangle$$

Storing wavefunctions with neural networks

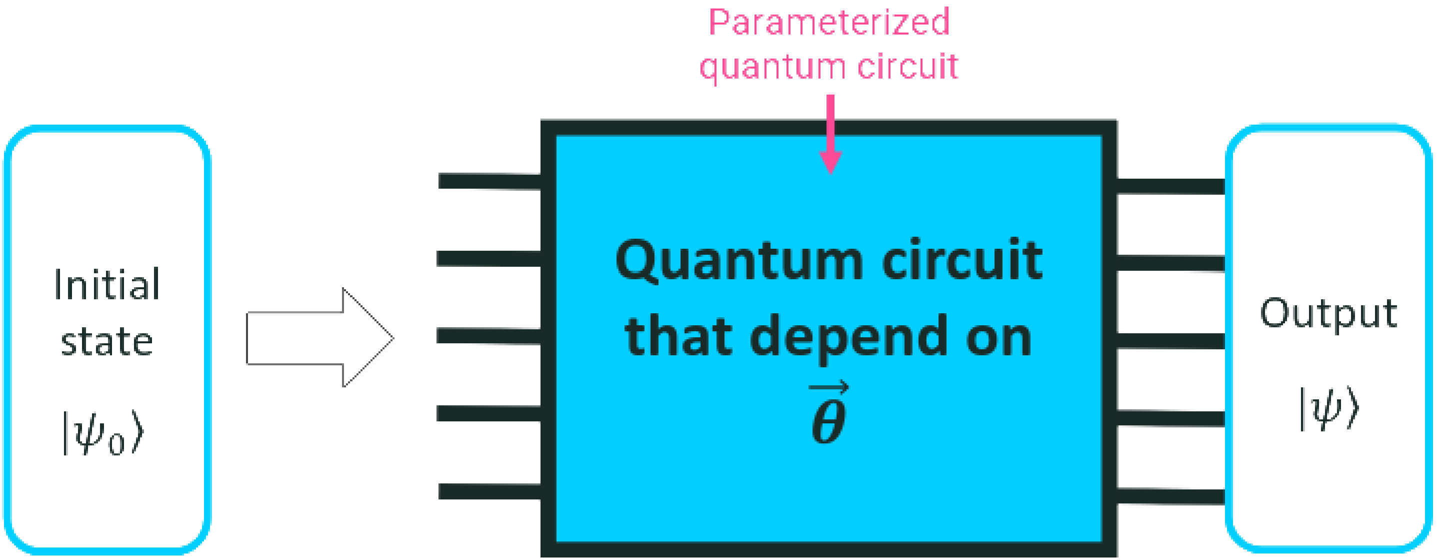
Do not store the coefficient, but find the right function that generates them

$$c_{s_1,s_2,\ldots,s_L} = f(s_1, s_2, \ldots, s_L)$$

Deep neural network

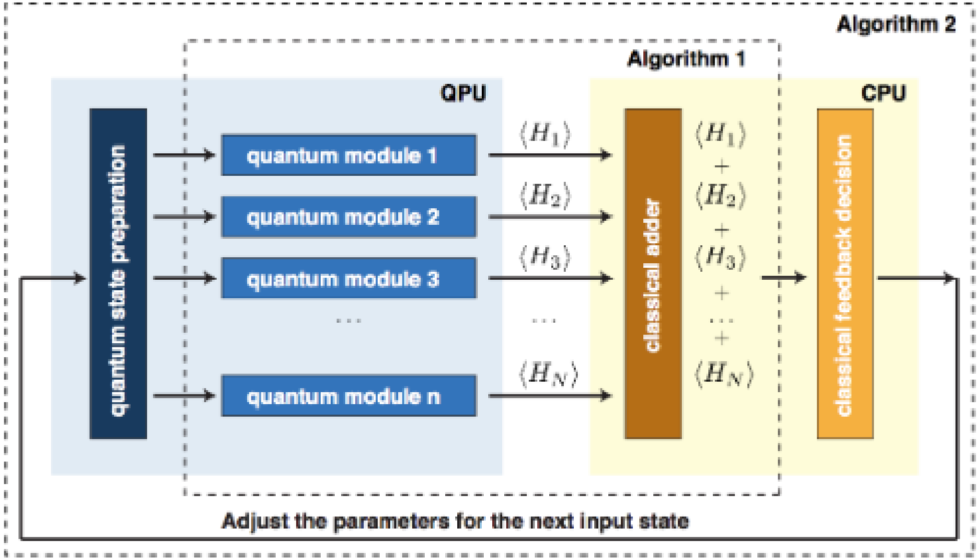


Storing wavefunctions with quantum circuits



The architecture of the circuit depends on the parameters

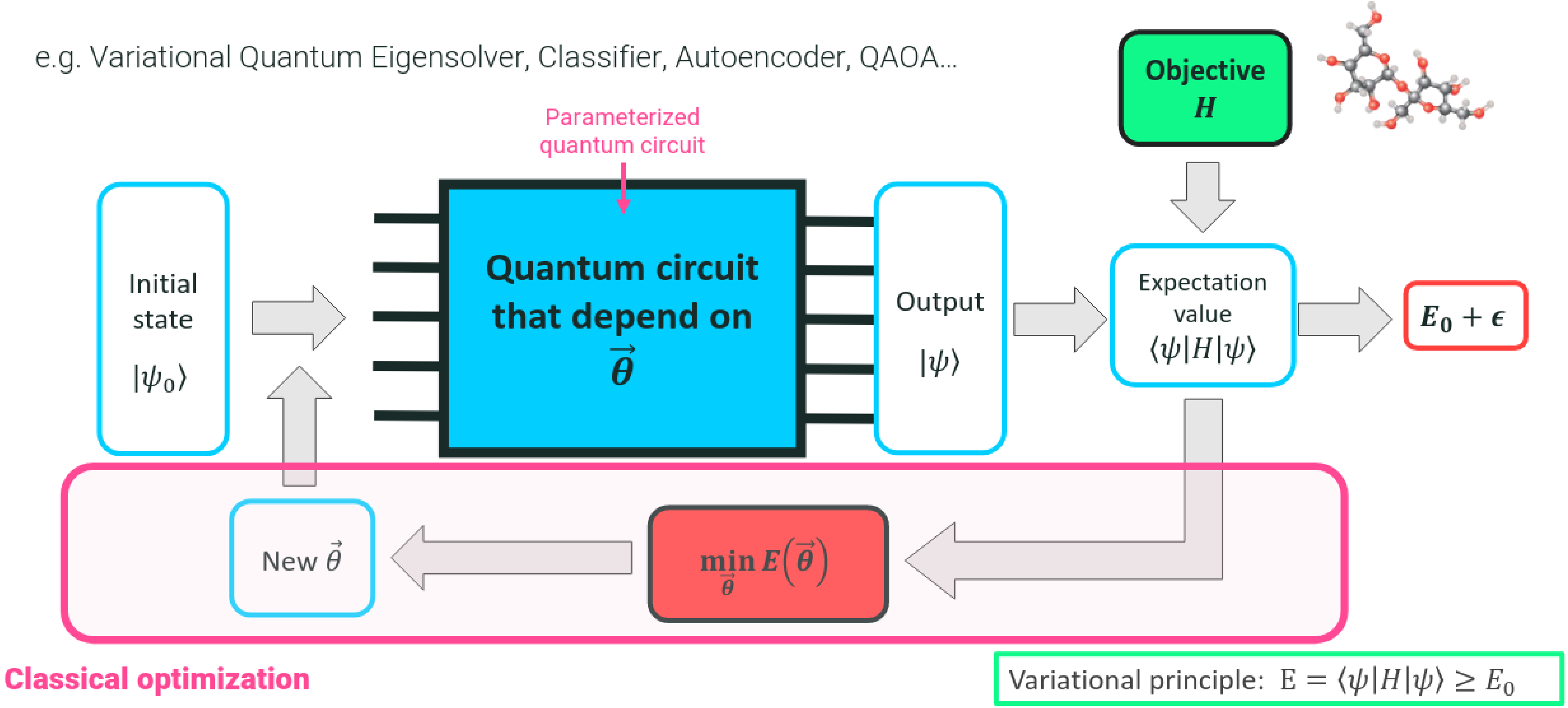
How to obtain the ground state energy with a quantum circuit



Execute the circuit that generates your trial wavefunctions
Get the expectation value of the Hamiltonian
Update the gate parameters to minimize energy
Repeat until convergence

Quantum circuit as a ground state generator

e.g. Variational Quantum Eigensolver, Classifier, Autoencoder, QAOA...



Optimizing the wavefunction

- Gradient Descent
  - Commonly chosen in classical optimization problems
  - Update parameter(s) based on largest energy change
  - Requires many circuit evaluations
  - Easy to get stuck in local minima
- Simultaneous Perturbation Stochastic Approximation (SPSA) optimizer
  - Ideal for “noisy” cost functions
  - Perturbs all parameters at once
  - Runs circuit twice, takes set of parameters that minimizes energy

Mapping fermions to spins

Naively, one might just $\mathrm{map}\, a_i \Rightarrow \sigma_i^+ = \frac{1}{2}(\sigma^x + i\sigma^y)$ and $a_i^\dagger \Rightarrow \sigma_i^- = \frac{1}{2}(\sigma^x - i\sigma^y)$ as they preserve anticommutator relations for same-site occupancy $\{\sigma_i^+, \sigma_i^-\} = 1$

However, $\left[\sigma_i^+, \sigma_j^-\right] = 0, i \neq j$ implies spins on different sites commute The Jordan-Wigner mapping gets around this by considering a string of N qubit operations:

$$a_i \Rightarrow I^{\otimes i-1} \otimes \sigma^+ \otimes \sigma^{Z \otimes N-i}$$
$$a_i^\dagger \Rightarrow I^{\otimes i-1} \otimes \sigma^- \otimes \sigma^{Z \otimes N-i}$$

Requires knowledge of the occupancy of the $N - i$ state occupations of those orbitals.

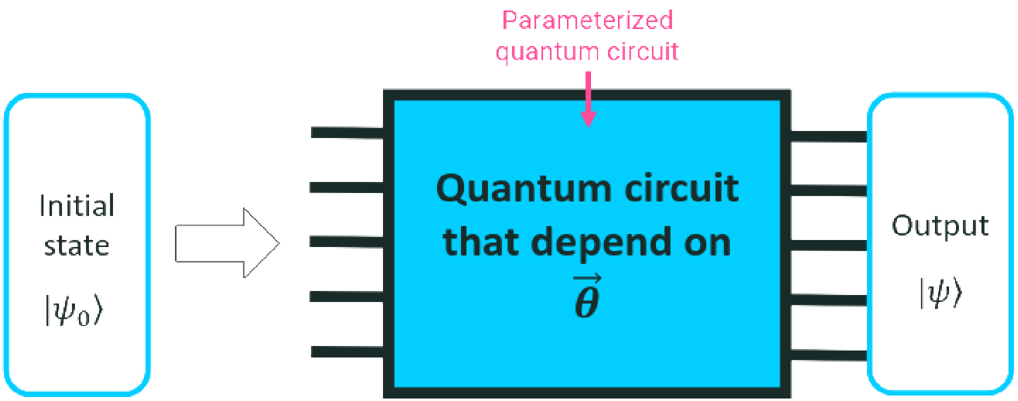
The wave function is spread out across all N qubits.

A minimal example of VQE

Take the Hamiltonian $\mathcal{H} = \vec{S}_0 \cdot \vec{S}_1$

With initial state $|\Psi_0\rangle = |\uparrow\downarrow\rangle$

What is the quantum circuit that would give you the ground state?



Quantum circuits for machine learning

Some examples of machine learning

Supervised learning

“Dog”

Labeled prediction

Generative learning

Probability Modeling

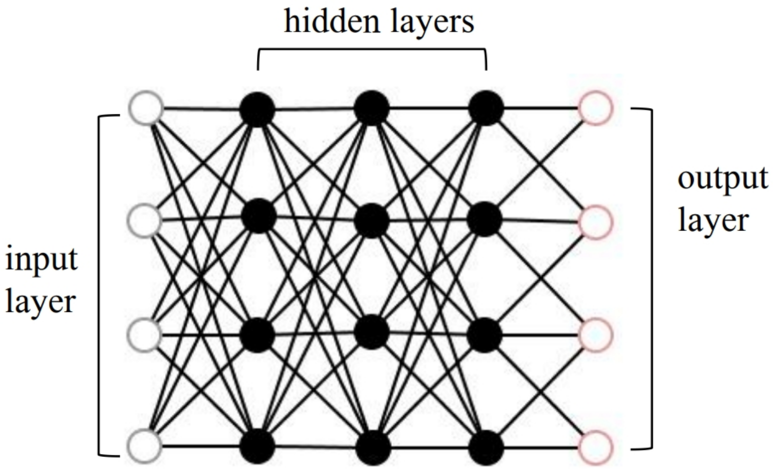
Reinforcement learning

Reward-based decision

(and others)

Machine learning with neural networks

Neural networks are a family of parametric functions



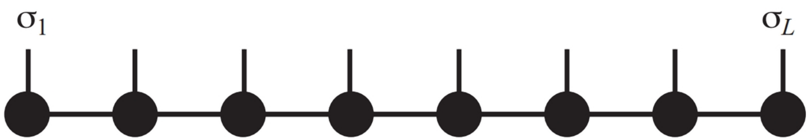
The parameters are optimized to minimize a certain functional

$$\chi = \mathrm{LOSS}[\vec{y}_{\mathrm{real}} - \vec{y}_{\mathrm{predicted}}] = \mathrm{LOSS}[\vec{y}_{\mathrm{real}} - f(\vec{x}_{\mathrm{real}})]$$

For example $\chi \sim |\vec{y}_{\mathrm{real}} - f(\vec{x}_{\mathrm{real}})|^2$

Machine learning with tensor-networks

Tensor-networks allow to parametrice high-dimensional functions

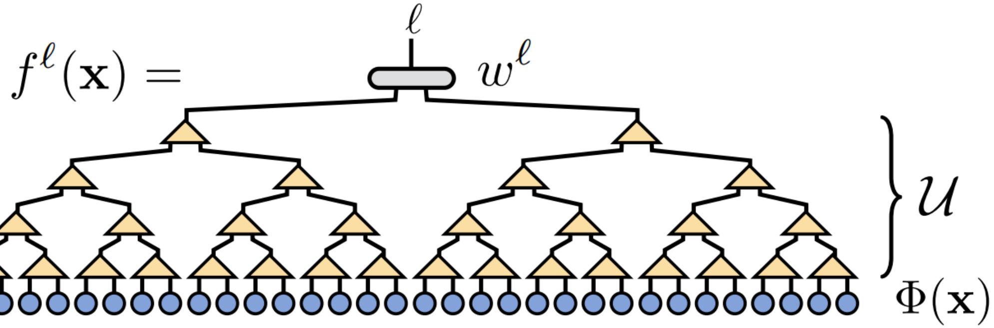


$$c_{s_1,s_2,\ldots,s_L} = M_1^{s_1} M_2^{s_2} \ldots M_L^{s_3}$$

$$|\Psi\rangle = \sum c_{s_1,s_2,\ldots,s_L} |s_1, s_2, \ldots s_L\rangle$$

Can we use tensor-network architectures “as if” they were neural networks?

Quantum many-body inspired machine learning

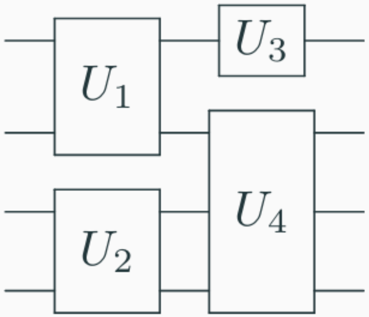


Quantum circuits as variational functions

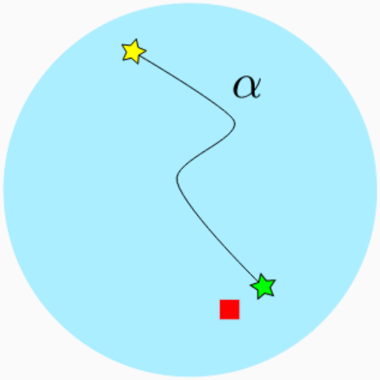
Rational:

Deliver variational quantum states $\rightarrow$ explore a large Hilbert space.

$$U(\vec{\alpha}) = U_n \dots U_2 U_1$$



Near optimal solution

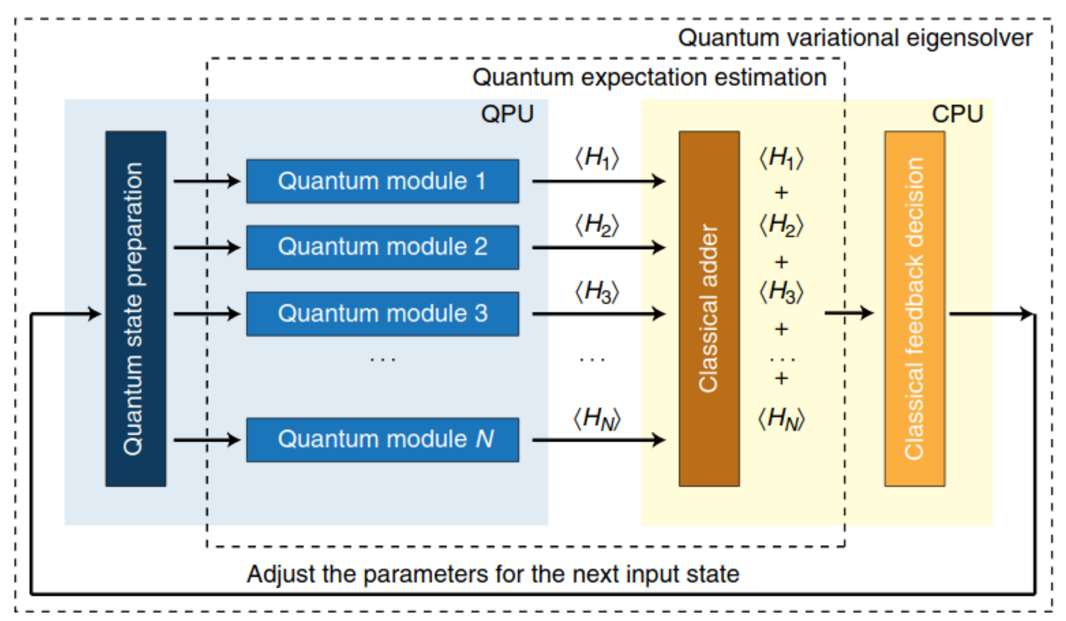


A quantum computer is, in a way, a machine that generates variational states

VQE as a quantum machine learning algorithm

VQE is hybrid classical-quantum algorithm.

1. Define an optimization problem, e.g. energy, correlations, etc.
2. Apply "machine learning" on circuit design.

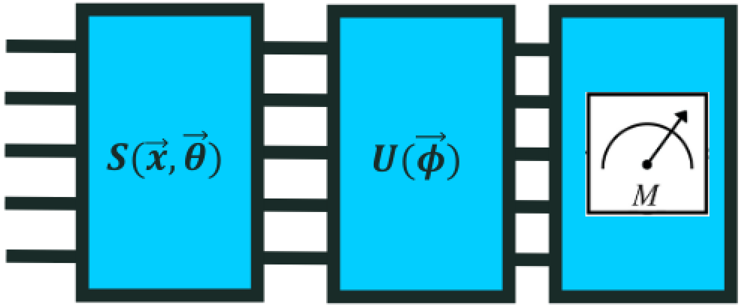


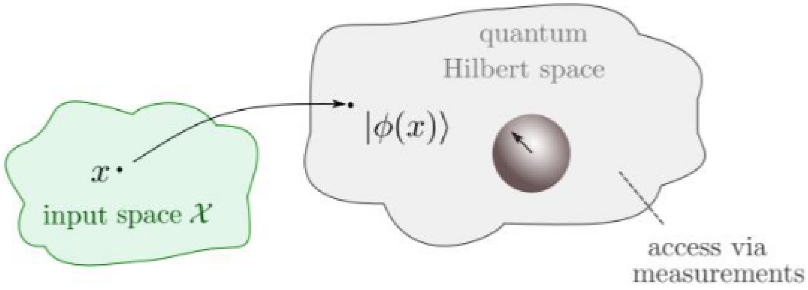
Quantum circuit for supervised learning

$$|\psi_0\rangle \rightarrow |\psi(\vec{x},\vec{\theta})\rangle \rightarrow |\psi(\vec{x},\vec{\theta},\vec{\phi})\rangle$$

Encode the data (quantum feature space)

Rotate to the correct measurement basis





We can then compute the Kernel

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \langle \Phi(\boldsymbol{x}_i) | \Phi(\boldsymbol{x}_j) \rangle$$

Or minimize the fidelity w.r.t. target states

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{\mathcal{D}} \left(1 - |\langle y_i | \Psi(\boldsymbol{x}_i, \boldsymbol{\theta}) \rangle|^2\right)$$

Take home

- Quantum circuits allows us to encode many body Hamiltonians and minimize their wavefunctions
- Remember the deadline for this (last) exercise sheet on Friday June 2nd