CS269Q: Quantum Programming **Spring 2019**

Notes on VQE and Adiabatic State Preparation

May 5, 2019

1 VQE and Ansatzes

Recall that the variational quantum eigensolver takes as input some $N \times N$ Hermitian matrix H and solves for the smallest eigenvalue λ_0 and corresponding eigenvector $|E_0\rangle$. When H corresponds to the Hamiltonian of a physical system, λ_0 is the ground state energy and $|E_o\rangle$ is the ground state of that system. The algorithm proceeds according to the variational principle which says that for any arbitrary states $|\psi\rangle$:

$$\frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \geq \lambda_0 \tag{1}$$

and that this equation is an equality if $|\psi\rangle = |E_0\rangle$.

To run this algorithm on a quantum computer we need to prepare states $|\psi\rangle$ in our quantum memory in order to calculate its expectation value. We also need to parameterize a family of states $|\psi(\vec{\theta})\rangle$ so that we can minimize $\langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle = \langle H\rangle(\vec{\theta})$ over a choice of $\vec{\theta}$. In order for this algorithm to be efficient we want to choose $|\psi(\vec{\theta})\rangle$ so that

- $\text{len}(\vec{\theta}) \leq \mathcal{O}(\text{poly}(N))$ which means the number of parameters to variationally optimize over stays small as the system grows, and
- $U(\vec{\theta})$, i.e. the preparation the states given by $|\psi(\vec{\theta})\rangle = U(\vec{\theta})|0\rangle$, can be done efficiently on the quantum computer. This means $U(\vec{\theta})$ needs a decomposition into at most a number of quantum operations that is polynomial in N.

1.1 Hardware efficient ansatzes

One way to do this is to use *hardware efficient ansatzes* see V.B.2.a of [5]. The idea behind these ansatzes is that we pick a short depth circuit that parameterizes a set of entangled states. We then repeat that circuit in some number of layers. We know that as the number of layers grows large these ansatezes approach randomly sampling unitary transformations (see [2] and [1]) and so this approach can be used on any generic H. However it is not guaranteed to do a good job of finding some $|\psi(\vec{\theta}) \approx |E_0\rangle$ for a small number of layers and a small number of parameters in theta.

2 A physically inspired ansatz: adiabatic state preparation

In order to try and pick a better parameterization $|\psi(\vec{\theta})\rangle$ we can be inspired by knowing something about our particular H, for example since H is Hermitian we can imagine that it corresponds to some Hamiltonian of some system. This may actually be the case, e.g. when we are using VQE to

solve for the grounds state energy of a molecule, but even when H comes from some other problem we are trying to solve we can imagine that it corresponds to some Hamiltonian.

Our goal is that we choose a parameterization such that our preparation of $|\psi(\vec{\theta})\rangle$ will allow it to be close to the ground state of H for some choice of $\vec{\theta}$. Given that we are now imagining that H is the Hamiltonian for some system we can imagine that our preparation of $|\psi(\vec{\theta})\rangle$ on a quantum computer is a "simulation" of a physical process that evolves into the ground state $|E_0\rangle$ of H.

In the physical world there is a theorem - the adiabatic theorem - that says that if we start in the ground state of some Hamiltonian H_0 and then slowly change H_0 into some other Hamiltonian H then we will end up in the ground state of H, assuming we have some appropriate definition of slowly enough. In the real world, quantum mechanics says that this "adiabatic" evolution from H_0 to H is a unitary transformation. Thus we can "simulate" this unitary transformation using quantum operations on our quantum computer. These notes will include a few pointers on how this works.

2.1 The Adiabatic Theorem

To model the transition from H_0 to H we introduce an adiabatic Hamiltonian $H_A(t)$ which is time-dependent. We want $H_A(t=0) = H_0$ and $H_A(t=T) = H$, where T is some final time that describes the total duration of the change. We let $H_A(t)$ be a continuous function of t that descibes how to move between the staring and finishing Hamiltonians.

The adiabatic theorem then says that in the limit $T \to \infty$, i.e. in the case of an infinitely slow (also called adiabatic) path if the system initially in the ground state of H_0 then it will end up in the ground state of H. It is beyond the scope of our course to prove this theorem but you can find a proof in [7] or take a course in quantum theory.

Generally the notion of "slow enough" has to do with making sure that you don't ever have enough energy in the $H_A(t)$ physical system to jump from the ground state to any excited states. Let the gap $\Delta(t) = |\lambda_1(t) - \lambda_0(t)|$ be the energy difference between the ground state energy $\lambda_0(t)$ and the first excited (next lowest) state energy $\lambda_1(t)$ at any time in the path. Quantum mechanics says that the probability of jumping between any two states decays exponentially with the gap between their energies. The jumping (or transistion probability) is also a function of how fast $H_A(t)$ is changing. This to remain adiabatic one must go slow relative to $\Delta(t)$.

2.2 Simulating the Adiabatic Evolution

Now we have a time dependant Hamiltonian $H_A(t)$ that describes how to prepare a ground state of H. Our task is to implement this Hamiltonian on our quantum computer so that the ground state ends up stored in our quantum memory using discrete quantum operations.

In general, it is possible to simulate time dependant Hamiltonians as circuits using a series of advanced methods that are beyond the scope of this course. For a modern example see [8] and [3]. One approach to this is using the Trotter formulas that we discussed in class. In general the longer the simulation time T the longer the quantum circuit. This is unfortunate for our adiabatic simulation because shorter adiabatic simulation times mean less accuracy in the answer.

2.2.1 Efficient Simulation of Hamiltonians

That we can simulate any Hamiltonian should not be surprising. It comes from the universality of quantum computation. What is interesting is to ask what Hamiltonians can be efficiently simulated.

Note that what is described in the following section are complexity theoretic arguments and apply to cases where we have very low noise quantum computation. They are still good to know about though.

Definition 1 *A Hamiltonian H can be* efficiently simulated *if for any $t > 0$, $\epsilon > 0$, there is a quantum circuit U consisting of poly$(n, t, 1/\epsilon)$ gates such that $||U - e^{-iHt}|| \leq \epsilon$.*

There are some interesting facts about what Hamiltonians meet this condition [4]:

- (Local Hamiltonians) If H acts on $\mathcal{O}(1)$ qubits then it can be efficiently simulated.
- (Rescaling) If H is efficiently simulable then cH is efficiently simulable for any $c = \text{poly}(n)$.
- (Addition) If H_1 and H_2 are efficiently simulable than $H_1 + H_2$ is as well.
- (Unitary Conjugation) If H can be efficiently simulated and U can be efficiently implemented then $UHU^\dagger$ can be efficiently simulated.
- (Diagonal Hamiltonians) If H is diagonal in the computational basis and any diagonal element can be efficiently computed then H can be efficiently simualted.
- (Sparse Hamiltonians) Suppose that for any a, one can efficiently compute all the values of b for which $langlea|H|b\rangle$ is nonzero. Then H can be efficiently simulated.

2.3 Applying Adiabatic Ansatzes to VQE

What this means is that there is a way to simulate the adiabatic evolution and that in many cases this is an efficiently simulation.

However: Efficient simulation still may require a lot of gates and we have lots of noise. So this means that instead we can

1. Choose a relatively small T value for our adiabatic state preparation.
2. Compile the simulation of this adiabatic state preparation into a relatively short quantum circuit U. This circuit is short because T was made small.
3. Pick some way of parameterizing $U(\vec{\theta})$ by tweaking some of the gate parameters from the original U.
4. Variationally optimize $|\psi(\vec{\theta})\rangle = U(\vec{\theta})|0\rangle$ to compensate for the fact that we chose a small T.

It is an open question as to how much we can compensate for short T by using variational optimization.

3 Other physically inspired ansatzes

There are other ways to make inspired choices of parameterization. Another example is the Unitary Couple Cluster Ansatz, which is a generalization of the couple cluster ansatz. The coupled cluster ansatz forms the gold standard for classical computing approximations for ground state calculations and we would expect - for reasons from the domain of quanum chemistry - that the unitary coupled cluster ansatz does better. Simulating the UCC ansatz requires only a polynomial number of gates on a quantum computer and is described in more detail in [6] and [5]. These references also contain pointers to other kinds of physically inspired ansatzes.

References

[1] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. *arXiv preprint arXiv:1612.05903*, 2016.

[2] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595, 2018.

[3] Andrew M Childs, Dmitri Maslov, Yunseong Nam, Neil J Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.

[4] Andrew Macgregor Childs. *Quantum information processing in continuous time*. PhD thesis, Massachusetts Institute of Technology, 2004.

[5] Sam McArdle, Suguru Endo, Alan Aspuru-Guzik, Simon Benjamin, and Xiao Yuan. Quantum computational chemistry. *arXiv preprint arXiv:1808.10402*, 2018.

[6] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[7] Albert Messiah. *Quantum Mechanics, volume II*. Dover, 1999.

[8] Nathan Wiebe, Dominic W Berry, Peter Høyer, and Barry C Sanders. Simulating quantum dynamics on a quantum computer. *Journal of Physics A: Mathematical and Theoretical*, 44(44):445308, 2011.