Variational Quantum Eigensolver

Cody M. Grant

(Dated: February 16, 2022)

I. INTRO

Many problems in quantum physics, quantum chemistry, etc. require us to find the minimum eigenvalue of a matrix. This is usually in the form of finding the ground state energy of a particular problem. However, as problems become more complicated, it is really difficult to find exact solutions/eigenvalues to the matrix. VQE estimates the ground state energy using shallower circuits.

Given a Hermitian matrix H with unknown minimum eigenvalue λ_{min}, VQE provides a bound λ_θ such that:

$$\lambda_{min} \leq \lambda_\theta \equiv \langle \psi(\theta)| H |\psi(\theta)\rangle \tag{1}$$

where $|\psi(\theta)\rangle$ is the eigenstate associated with λ. The algorithm is iteratively optimized to change θ and finding a close (enough) estimate. Note: this reminds me of Neural Network learning and error minimization!

One can see that if $H = \sum_{i=1}^{N} \lambda_i |\psi_i\rangle \langle\psi_i|$, then it can be seen:

$$\lambda_{min} \leq \langle H \rangle_\psi = \langle \psi| H |\psi\rangle = \sum_{i=1}^{N} \lambda_i |\langle \psi_i|\psi\rangle|^2 \tag{2}$$

This is the variational method you can learn about in a QM textbook or class. Initially, you arbitrarily selection an ansatz for the wave function, calculate expectation value, and then iteratively update the wavefunction to get closer to the actual value of the ground state energy.

A systematic approach is needed to implement the variational method on a computer. VQE does this by having a parametrized circuit with a fixed form (called a *variational from*). A fixed variational form with a polynomial number of params can only generate transforms in a huge Hilbert space. Therefore, many variational forms do exists, some more generic and some utilize domain specific knowledge.

II. SIMPLE VARIATIONAL FORMS

Two opposing goals we must balance: 1. our n qubit variational form would generate any possible state ψ, but 2. we would like to use as few parameters as possible. Let us start by satisfying our first goal and ignoring the second.

Considering $n = 1$, the $U3$ gate takes three parameters:

$$U3(\theta, \phi, \lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{-i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\lambda+\phi)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{3}$$

Any possible single qubit transformation can be implemented by correctly setting these params. This universal var form only has 3 params and can be efficiently optimized. Less trivially, a two qubit form has entanglement and the circuit form is more complicated:
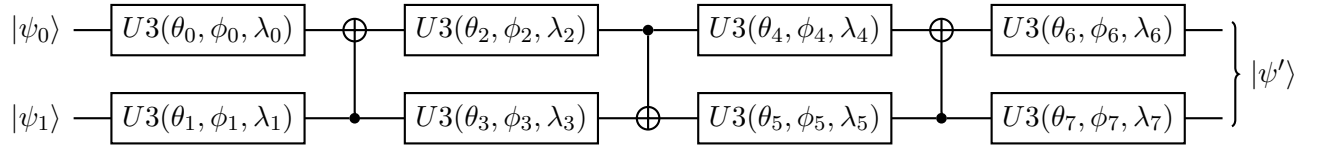
This version has 24 parameters, it isn't hundreds, but it still isn't a little amount, and it can still take some time to figure out how to optimized the set.

III. PARAMETER OPTIMIZATION

Parameter Optimization has various challenges: QHardware has various types of noise, etc.

Like in artificial neural networks, a popular optimization strat is gradient descent. A consequence of this strategy is the number of evaluations to perform for error calculation and parameter adjustment increases with the number of parameters present. Another problem is if the algorithm gets stuck in a poor local optima, then it can become expensive in terms of number of circuit evals performed and time, and you could get a poor answer at the end of it all. Therefore, it is not recommended for VQE.

For opt a noisy obj func, we can use *Simultaneous Perturbation Stochastic Approx* (SPSA). It approx the grad of the objective func with only two measurements. It does this by concurrently perturbing all params in random fashion, whereas grad descent perturbs one param at a time. Noise is the important factor here, if there were no noise present in the cost function eval, then many classical optimizers could be used.

IV. STRUCTURE OF COMMON VARIATIONAL FORMS

Two realistic categories for variational forms: 1. Uses domain or application specific knowledge to limit set of possible output states. 2. Uses heuristic circuit without prior domain or app knowledge.

In the first category, prior knowledge is used to restrict the set of transformations required. For example, finding the ground state energy of a molecule, the number of particles is known. Then limiting the starting state and var form to only be the correct number of particles used, can greatly reduce the new transformation subspace.

The second cat, gates are layered so that good approx on a wide range of states may be obtained. Qiskit Aqua supports three types: RyRz, Ry, and SwapRz.