Differentiable Quantum Circuits

Jin-Guo Liu, May 4, 2019
cacate0129@gmail.com

Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China

Differentiable quantum circuits are the basics of quantum software 2.0 and many quantum-classical algorithms.

CONTENTS

I. THE GRADIENT OF A QUANTUM CIRCUIT

A quantum circuit can be parametrized in different ways. In this note, we focus on the simplest case, which tunes the length of a control pulse (or evolution time) as shown in Fig. 1

$$|\psi_N\rangle = U_N \dots U_k \dots U_2 U_1 |\psi_0\rangle \tag{1}$$

where the k-th gate $U_k(t_k) = e^{-iH_k t_k}$, with H_k the driving Hamiltonian and t_k a tunable parameter.

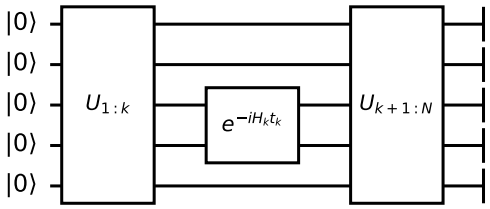


FIG. 1. Parametrize a quantum circuit with real numbers, the evolution time.

Let's first consider an observable as the loss function

$$\begin{aligned} \langle B \rangle &= \langle \psi_N | B | \psi_N \rangle \\ &= \langle \psi_{k-1} | U_k^\dagger \tilde{B}_{k+1} U_k | \psi_{k-1} \rangle \end{aligned} \tag{2}$$

where we have introduced two shorthands

$$|\psi_k\rangle = U_k \dots U_2 U_1 |\psi_0\rangle \tag{3}$$

$$\tilde{B}_k = U_k^\dagger U_{k+1}^\dagger \dots U_N^\dagger B U_N \dots U_{k+1} U_k \tag{4}$$

Differentiating $\langle B \rangle$ over t_k gives

$$\begin{aligned} \frac{\partial \langle B \rangle}{\partial t_k} &= \langle \psi_k | \tilde{B}_{k+1} (-iH_k) | \psi_k \rangle + \langle \psi_k | (iH_k) \tilde{B}_{k+1} | \psi_k \rangle \\ &= 2\Re \left[-i \langle \psi_k | \tilde{B}_{k+1} H_k | \psi_k \rangle \right] \end{aligned} \tag{5}$$

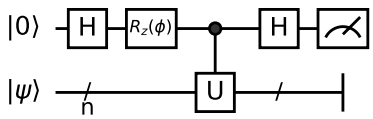
Then it comes to the question, for a general operator G, is it possible to efficiently estimate Eq. (5) using a quantum computer? We will show in the following subsection that at least when G is a unitary operator, it is true. For a general non-unitary G, as long as G can be decomposed into multiple unitary gates, e.g. in the form of weighted summation of finite Pauli strings, this gradient can be evaluated.

A. Expectation value of a unitary gate

Given a unitary gate U and a wave function $|\psi\rangle$, Hadamard test [1] is an algorithm to estimate

$$\Re[e^{i\phi} \langle \psi | U | \psi \rangle]. \tag{6}$$

The circuit representation of a Hadamard test is



The top qubit that gets measured in the final stage is an ancilla qubit. The expectation value of Z operator on this ancilla qubit gives Eq. (6).

Proof. Although Hadamard test is easily verifiable through circuit analysis, it is also interesting to prove from the tensor network contraction perspective. The following proof considers a Hadamard test in which case the gate is not only unitary but also Hermitian. This result will be used in the swap test algorithm in the following section where SWAP gates apparently falls into this category. Tensor representation of gates used in this proof can be found in Appendix B.

(a) reformulate this algorithm into a tensor network contraction problem,

(b) apply rule HZH = X,

(c) apply rule R3 in Appendix A,

(d) evaluate the first line of tensors. The final result is equivalent to $\cos(\phi)\langle\psi|U_H|\psi'\rangle$.

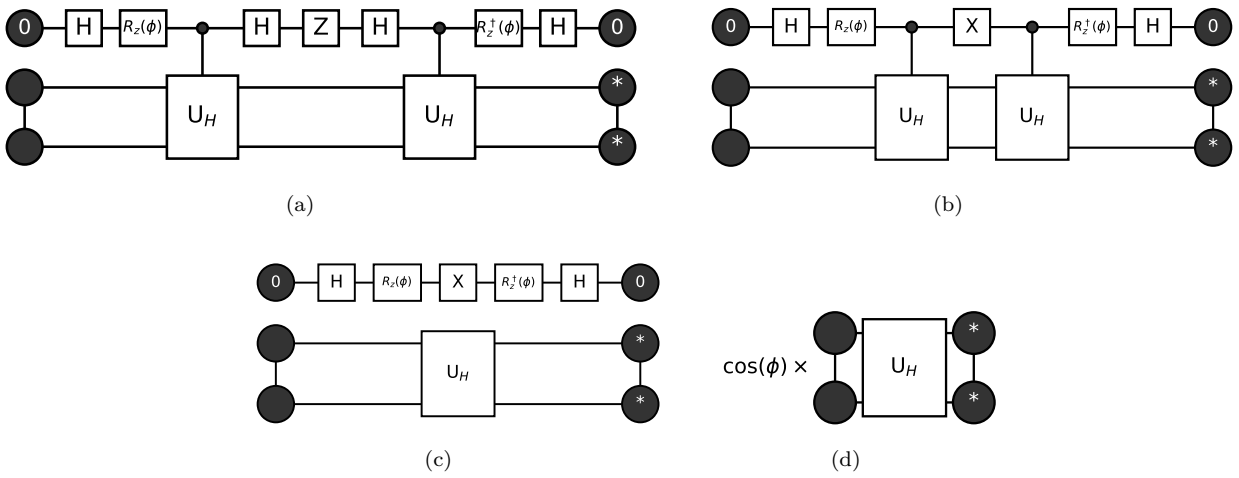


FIG. 2. Circles are tensors in the MPS representation of $|\psi\rangle$, the one with label "$*$" is the conjugate tensor, U_H is a Hermitian unitary gate, e.g. the SWAP gate in the swap test.

□

The following is a Julia implementation of this algorithm

```julia
# using Yao @ v0.4.1
using Yao, Yao.ConstGate
using Test

"""
Hadamard Test Circuit. `U` is a general unitary gate and `??` is the phase.
"""
function hadamard_test_circuit(U::AbstractBlock{N}, ??::Real) where N
    chain(N+1, put(N+1, 1=>chain(H, Rz(??))),
        control(N+1, 1, 2:N+1=>U),
        put(N+1, 1=>H)
    )
end

nbits = 3
ϕ = -π/2
reg = rand_state(nbits)
U = matblock(rand_unitary(1<<nbits))

circuit = hadamard_test_circuit(U, ??)
reg_ancilla = join(reg, zero_state(1))
res = expect(put(nbits+1, 1=>Z), reg_ancilla |> circuit)

@test real(expect(U, reg)*exp(im*ϕ)) ≈ res
```

A caveat of Hadamard test algorithm is the controlled-unitary gate of a quantum circuit is not easy to prepare in general, even when the unitary gate itself is easy to prepare.

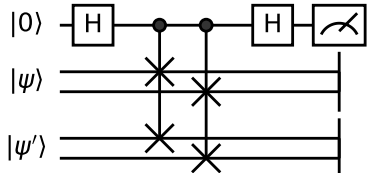
Exercise *Show the Hadamard test circuit for evaluating* $\langle\psi_N|\frac{\partial}{\partial t_k}|\psi_N\rangle$*, where* $U_k(t_k) = e^{-i\sigma_k t_k/2}$*. Hints are in Ref. 2.*

B. State overlap algorithms

[**JG: Needs polish.**] State overlap is about calculating the (squared) overlap of two state $|\psi\rangle$ and $|\psi'\rangle$, which is $|\langle\psi|\psi'\rangle|^2$. Related techniques are widely used in quantum machine learning algorithms like generative modeling [3], kernel methods [4] and quantum wave function learning. In this section, we introduce two widely used algorithms swap test and another algorithm "discovered" by machine learning [5].

1. *Swap test*

Swap test is a special kind of Hadamard test, where the unitary operation is SWAP gates that exchange two states. The following circuit is an example of a swap test circuit for two random two-qubit states $|\psi\rangle$ and $|\psi'\rangle$



The swap test algorithm is first proposed as an approach of quantum fingerprinting [6]. It was later used in some quantum machine learning algorithms [3]. This algorithm is directly verifiable by replacing the unitary gate in Hadamard test by a SWAP operation

$$\begin{aligned}\Re\left[{}_A\langle\psi|\otimes{}_B\langle\psi'|\mathrm{SWAP}_{A,B}|\psi\rangle_A\otimes|\psi'\rangle_B\right] &= {}_A\langle\psi|\psi'\rangle_A{}_B\langle\psi|\psi'\rangle_B \\ &= |\langle\psi|\psi'\rangle|^2,\end{aligned} \tag{7}$$

where A, B are two input registers. Although this algorithm is demonstrated with pure states, it can be generalized to mixed states.

The Julia implementation of swap test is

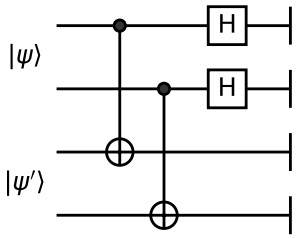
```
"""
Quantum circuit for swap test,
`nbits` is the number of qubits for one of the quantum states to compare,
`nstate` is the number of input states, for state overlap, it should be 2.
"""
function swap_test(nbits::Int, nstate::Int)
    N = nstate*nbits + 1
    chain(N, put(N, 1=>H),
        chain(N, [chain(N, [control(N, 1, (i+(k*nbits-nbits)+1, i+k*nbits+1)=>SWAP)
            for i=1:nbits]) for k=1:nstate-1]),
        put(N, 1=>H)
    )
end

nbits = 3
reg1 = rand_state(nbits)
reg2 = rand_state(nbits)
overlap_exact = abs2(reg1'*reg2)

circuit = swap_test(nbits, 2)
reg_ancilla = join(reg1, reg2, zero_state(1))
reg_ancilla |> circuit
@test expect(put(2nbits+1, 1=>Z), reg_ancilla) ≈ overlap_exact
```

2. *State overlap without introducing ancilla qubits*

It is possible to get the overlap between two states without introducing ancilla qubits, The algorithm in Ref. 5 gives the following circuit



After applying this circuit two quantum registers of n qubits, we measure the outcome on computational basis and get bit-strings $s_1, s_2 \dots s_n$ and $t_1, t_2 \dots t_n$.

$$\langle\psi|\psi'\rangle = \mathop{\mathbb{E}}_{\mathbf{s},\mathbf{t}}\left[(-1)^{s_1 t_1 + s_2 t_2 \dots s_n t_n}\right], \tag{8}$$

Which is equivalent to measuring $\prod_{i=1}^{n} \mathrm{CZ}(i, i+n)$ operator.

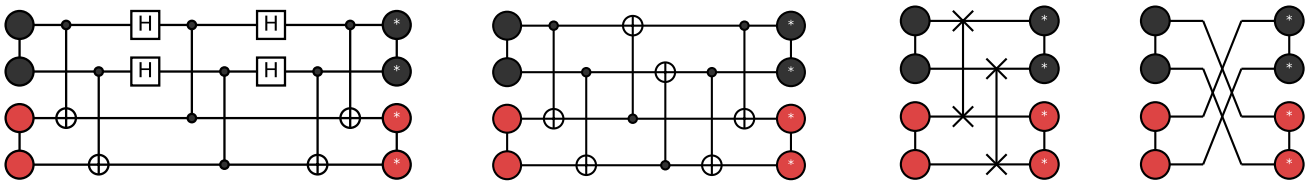
Proof. We can prove it from the tensor network perspective, tensors used in this proof can be found in Appendix B.

(a) reformulate this algorithm into a tensor network contraction problem,

(b) apply rule R1 in Appendix A,

(c) apply rule R2 in Appendix A,

(d) apply swap operations, the final result is equivalent to $|\langle\psi|\psi'\rangle|^2$.



(a) (b) (c) (d)

FIG. 3. Black (Red) circles are tensors in the MPS representation of $|\psi\rangle$ ($|\psi'\rangle$), tensors with label "$*$" are conjugate tensors.

□

The Julia implementation of this algorithm is missing...

```julia
function state_overlap_algo(nbits::Int)
    # write your implementation here.
end

nbits = 3
reg1 = rand_state(nbits)
reg2 = rand_state(nbits)
overlap_exact = abs2(reg1' * reg2)

circuit = state_overlap_algo(nbits)
observable = chain(control(2nbits, i, i+nbits=>Z) for i=1:nbits)
@test expect(observable, join(reg1, reg2) |> circuit) ≈ overlap_exact
```

Exercise *If the above test cannot be passed, the continuous integration may fail. Wish someone can help us fill the* `swap_test_simple` *function to pass the test.*

C. The landscape and gradients of an observable

In the start of this section, we introduced the general form a gradient Eq. (5), and find it hard to obtain in general. Ref. 7 shows when a quantum gate $U_k(t_k)$ is parameterized in a special form, the landscape of observable expectation value is a sine function.

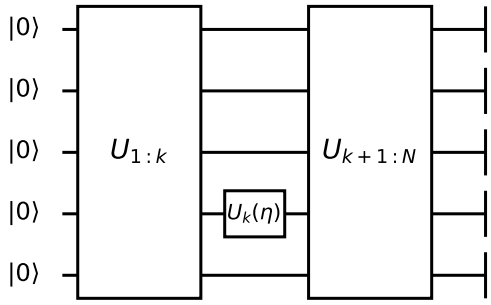


FIG. 4. A circuit containing a differentiable parameter η.

Starting from Eq. (2), we parameterize the k-th gate as $U_k(\eta) = e^{-i\Xi\eta/2}$, with $\Xi^2 = 1$. Then it is straightforward to have $U_k(\eta) = \cos(\frac{\eta}{2}) - i\sin(\frac{\eta}{2})\Xi$, insert it into Eq. (2)

$$\langle B\rangle = \langle\psi_{k-1}|\left[\cos\frac{\eta}{2} + i\sin\frac{\eta}{2}\Xi\right]\tilde{B}_{k+1}\left[\cos\frac{\eta}{2} - i\sin\frac{\eta}{2}\Xi\right]|\psi_{k-1}\rangle \tag{9}$$

$$= \cos^2\frac{\eta}{2}\langle\psi_{k-1}|\tilde{B}_{k+1}|\psi_{k-1}\rangle + \sin^2\frac{\eta}{2}\langle\psi_{k-1}|\Xi\tilde{B}_{k+1}\Xi|\psi_{k-1}\rangle + i\sin\frac{\eta}{2}\cos\frac{\eta}{2}\langle\psi_{k-1}|\left[\Xi,\tilde{B}_{k+1}\right]|\psi_{k-1}\rangle \tag{10}$$

$$= \cos^2\frac{\eta}{2}\left(\langle\psi_{k-1}|\tilde{B}_{k+1} - \Xi\tilde{B}_{k+1}\Xi|\psi_{k-1}\rangle\right) + i\frac{\sin\eta}{2}\langle\psi_{k-1}|\left[\Xi,\tilde{B}_{k+1}\right]|\psi_{k-1}\rangle + \langle\psi_{k-1}|\Xi\tilde{B}_{k+1}\Xi|\psi_{k-1}\rangle \tag{11}$$

$$= \frac{\cos\eta}{2}\left(\langle\psi_{k-1}|\tilde{B}_{k+1} - \Xi\tilde{B}_{k+1}\Xi|\psi_{k-1}\rangle\right) + i\frac{\sin\eta}{2}\langle\psi_{k-1}|\left[\Xi,\tilde{B}_{k+1}\right]|\psi_{k-1}\rangle + \frac{1}{2}\langle\psi_{k-1}|\tilde{B} + \Xi\tilde{B}_{k+1}\Xi|\psi_{k-1}\rangle \tag{12}$$

$$= \alpha\cos\eta + \beta\sin\eta + \gamma \tag{13}$$

$$= r\cos(\eta - \phi) + \gamma \tag{14}$$

In Eq. (13), we have introduced

$$\alpha = \frac{1}{2}\left(\langle\psi_{k-1}|\tilde{B}k + 1 - \Xi\tilde{B}k + 1\Xi|\psi_{k-1}\rangle\right), \tag{15}$$

$$\beta = i\frac{1}{2}\langle\psi_{k-1}|\left[\Xi,\tilde{B}k + 1\right]|\psi_{k-1}\rangle, \tag{16}$$

$$\gamma = \frac{1}{2}\langle\psi_{k-1}|\tilde{B} + \Xi\tilde{B}k + 1\Xi|\psi_{k-1}\rangle. \tag{17}$$

Finally, we obtained a sine function like

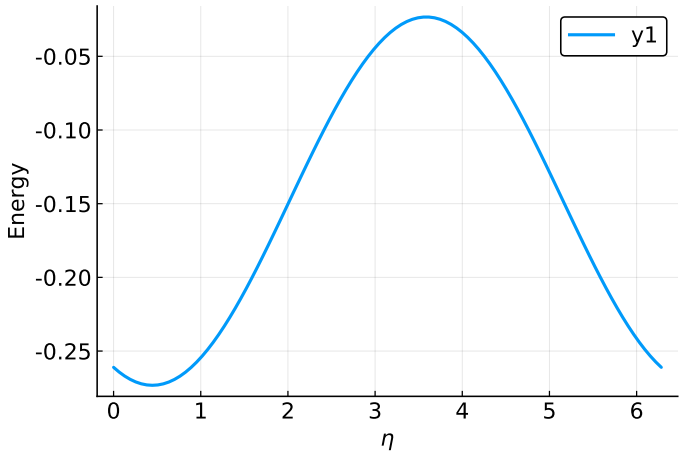


FIG. 5. Observable expectation value as a function of circuit parameter η in a random circuit.

A direct proposition is

$$\frac{\partial\langle B\rangle_\eta}{\partial\eta} = \frac{1}{2}(\langle B\rangle_{\eta+\frac{\pi}{2}} - \langle B\rangle_{\eta-\frac{\pi}{2}}). \tag{18}$$

This proposition has been derived independently in Ref. 8 and 9, it plays a central role in gradient-based optimization of quantum circuits.

Exercise

1. *Given an observable* $\langle B\rangle$*, obtain the gradient of a parameterized* CPHASE *gate in a circuit, the matrix representation of a* CPHASE *gate is*

$$\text{CPHASE}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{pmatrix} \tag{19}$$

2. Obtain the second derivative of θ.

This kind of parameterization is easy to implement experimentally when Ξ is a single qubit rotation gate or a SWAP gate, while does not put too many restrictions to symmetries and representation power of a quantum circuit. The gradient can also be estimated directly by a quantum device.

But one has to conquer the following two facts to really show the advantage of a quantum neural network. First, the complexity of this quantum gradient algorithm scale as $O(N^2)$ with N is the number of differentiable parameters in this circuit. This complexity may be the lower bond. A simple argument would be, the information we get from a single run is a constant, if we increase the number of parameters without introducing more qubits, to get N independent gradient information, the number of runs must also scales as N. On the other side, the time of a single run is proportional to the depth of a circuit, which is proportional to N. The other fact is the exponentially large Hilbert space not only provides possibilities of quantum advantage but also incurs a fundamental issue for training, the gradients vanish exponentially fast as the number of qubits increases, given the circuit structure is randomly designed. [10]

II. APPLICATIONS

A. Variational Quantum Eigensolver: Two-Site Hubbard Model

Obtaining the ground state of a Hamiltonian play an important role in condensed matter physics and quantum chemistry [11**?**]. In some classical algorithms, we first parameterize a wave function as an ansatz and obtain the ground state through optimization. This ansatz can be a tensor network, a computational graph (such as a Boltzmann machine) and more. The loss function for obtaining the ground state is usually defined as

$$E = \frac{\langle\psi(\theta)|H|\psi(\theta)\rangle}{\langle\psi|\psi\rangle} \tag{20}$$

As a toy example, we use a vector as an ansatz to show the differential programming approach to get the ground state energy of a two-site Hubbard model

$$H = t\sum_{\sigma=\uparrow,\downarrow}(c^\dagger_{1\sigma}c_{2\sigma} + h.c) + U\sum_{i=1,2}(n_{i\uparrow} - \frac{1}{2})(n_{i\downarrow} - \frac{1}{2}) \tag{21}$$

Apply the Jordan-Wigner transformation

$$c_j = \left(\prod_{k=1}^{j-1} Z_k\right) P^-_j, \tag{22}$$

we obtain a Bosonic model

$$H = t(\mathrm{P}^+_{1\uparrow}\mathrm{Z}_{1\uparrow}\mathrm{Z}_{1\downarrow}\mathrm{P}^-_{2\uparrow} + \mathrm{P}^+_{1\downarrow}\mathrm{Z}_{1\downarrow}\mathrm{Z}_{2\uparrow}\mathrm{P}^-_{2\downarrow} + h.c.) + U\sum_{i=1,2}(\mathrm{P}^1_{i\uparrow} - \frac{1}{2})(\mathrm{P}^1_{i\downarrow} - \frac{1}{2}) \tag{23}$$

$$= \frac{t}{2}(X_{1\uparrow}Z_{1\downarrow}X_{2\uparrow} + Y_{1\uparrow}Z_{1\downarrow}Y_{2\uparrow} + X_{1\downarrow}Z_{2\uparrow}X_{2\downarrow} + Y_{1\downarrow}Z_{2\uparrow}Y_{2\downarrow}) + \frac{1}{4}U\sum_{i=1,2}(Z_{i\uparrow}Z_{i\downarrow}) \tag{24}$$

Here we have used the relation $\mathrm{P}^\pm = \frac{1}{2}(\mathrm{X} \pm i\mathrm{Y})$, and $\mathrm{P}^{0/1} = \frac{1}{2}(\mathbb{1} \pm \mathrm{Z})$ to obtain the second line. Four flavors $1\uparrow, 1\downarrow, 2\uparrow$ and $2\downarrow$ can be simulated by four qubits. The second line decomposes a Hamiltonian into a weighted summation of Pauli strings, the expectation values of these Pauli strings can be estimated easily on a quantum device. As long as a Hamiltonian contains finite (means polynomial as the degree of free) terms, the energy can be estimated. Is this always true? Yes, since in condensed matter physics and quantum chemistry, interactions are always two-body $c^\dagger_i c^\dagger_j c_k c_l$, one can always represent a Hamiltonian with n^4 terms with n the number of flavors.

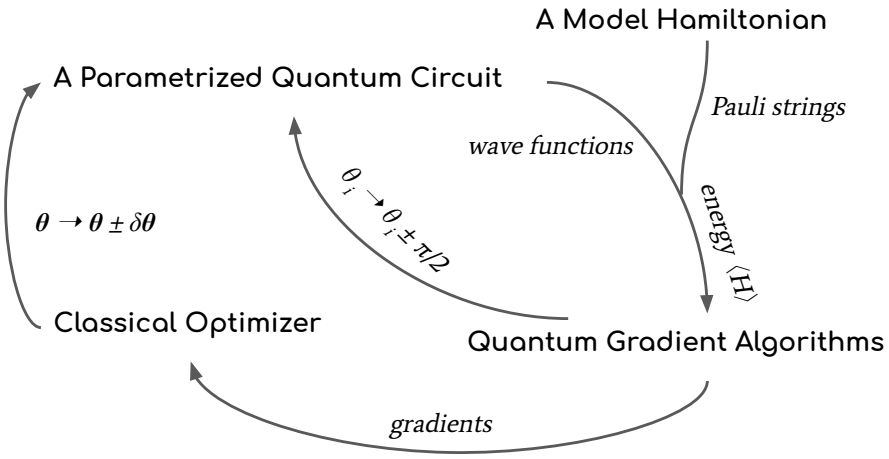


FIG. 6. A schetch of VQE algorithm, nodes are objectives and lines indicates information flow.[**JG: Maybe bit string is better than wave function, and the top right line is not elegant.**]

The gradient-based VQE algorithm has four basic components, a Hamiltonian, a quantum circuit as wave function ansatz, the algorithm to obtain the gradients and a classical optimizer to utilize these gradients for training efficiently. We run the circuit to generate wave functions repeatedly, with the Hamiltonian information, we are able to estimate the energy expectation value. Gradient algorithm will request the circuit to generate samples to produce gradients one using the relation Eq. (18). This gradient information is then fed into a classical optimizer such as `stochastic gradient descent`, `Adam` [12], `RMSProp` and `RProp` [13], refer Ref. 14 for a review of these algorithms. Also, one can use recently developed sequential optimization technic. The optimizer will decide how to tune the parameters to decrease the loss efficiently.

Let's open this notebook

https://github.com/QuantumBFS/SSSS/blob/master/4_quantum/variational_qantum_circuit.ipynb

and make your hands dirty. Contents in this notebook

1. Solving the ground state of a two-site Hubbard model, a differentiable programming approach,
2. The landscape of a quantum circuit,
3. The variational quantum eigensolver approach,
4. The gradient vanishing problem.

B. Quantum Circuit Born Machine

The measurement of a quantum circuit is a natural sampler $p(x) = |\langle x|\psi\rangle|^2$ according to the Born's rule. A generative model making use of Born's rule is nicknamed Born Machine. [**?** **?**] The complexity theory of a quantum circuits guarantees the "supremacy" of a quantum sampler, that there exists a class of quantum circuits, its distribution is #P-hard in average case [**?**], i.e. believed to requires exponentially many classical resources to simulate the same distribution for a classical sampler. This founds the advantage of using a quantum sampler.

The central problem of using a quantum circuit as a generative model is constructing a loss function to measure the distance between the training set and generation set. One of the approaches is two sample test.

$$\begin{aligned}\mathcal{L} &= \left\| \sum_x p_\theta(x)\phi(x) - \sum_x \pi(x)\phi(x) \right\|^2 \\ &= \mathop{\mathbb{E}}_{x\sim p_\theta, y\sim p_\theta}[K(x,y)] - 2\mathop{\mathbb{E}}_{x\sim p_\theta, y\sim \pi}[K(x,y)] + \mathop{\mathbb{E}}_{x\sim \pi, y\sim \pi}[K(x,y)].\end{aligned} \tag{25}$$

In the following, we derive the gradient

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta_l^\alpha} = &\sum_{x,y} K(x,y)\left(p_{\boldsymbol{\theta}}(y)\frac{\partial p_{\boldsymbol{\theta}}(x)}{\partial \theta_l^\alpha} + p_{\boldsymbol{\theta}}(x)\frac{\partial p_{\boldsymbol{\theta}}(y)}{\partial \theta_l^\alpha}\right)\\ &- 2\sum_{x,y} K(x,y)\frac{\partial p_{\boldsymbol{\theta}}(x)}{\partial \theta_l^\alpha}\pi(y).\end{aligned} \tag{26}$$

The partial derivative on the right-hand side of this equation can be unbiasedly computed using the approach of [9, 15]. For a circuit containing a unitary gate parameterized as $U(\eta) = e^{-\frac{i}{2}\eta\Sigma}$ with $\Sigma^2 = 1$ (e.g. Σ can be Pauli operators, CNOT or SWAP), the gradient of the expected value of an observable B with respect to the parameter η reads

$$\frac{\partial \langle B\rangle_\eta}{\partial \eta} = \frac{1}{2}\left(\langle B\rangle_{\eta^+} - \langle B\rangle_{\eta^-}\right), \tag{27}$$

where $\langle\cdot\rangle_{\eta^{(\pm)}}$ represents expectation values of observables with respect to the output quantum wave function generated by the same circuit with circuit parameter $\eta^\pm \equiv \eta \pm \frac{\pi}{2}$. Note Eq. (27) is an unbiased estimate of the gradient in contrast to the finite difference approaches which are sensitive to noise of the quantum circuit.

Since the quantum circuit Born machine employs the same parameterization, we identify $|x\rangle\langle x|$ as the observable and apply Eq. (27) to the output probability of the circuit

$$\frac{\partial p_{\boldsymbol{\theta}}(x)}{\partial \theta_l^\alpha} = \frac{1}{2}\left(p_{\boldsymbol{\theta}^+}(x) - p_{\boldsymbol{\theta}^-}(x)\right), \tag{28}$$

where $\boldsymbol{\theta}^\pm \equiv \boldsymbol{\theta} \pm \frac{\pi}{2}\mathbf{e}_l^\alpha$, with $\mathbf{e}_l^\alpha$ the (l,α)-th unit vector in parameter space. Substituting Eq. (28) into Eq. (26) we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = &\frac{1}{2}\left(\sum_{x,y} K(x,y)p_{\boldsymbol{\theta}}(y)p_{\boldsymbol{\theta}^+}(x) - \sum_{x,y} K(x,y)p_{\boldsymbol{\theta}}(y)p_{\boldsymbol{\theta}^-}(x)\right)\\ &+ \frac{1}{2}\left(\sum_{x,y} K(x,y)p_{\boldsymbol{\theta}}(x)p_{\boldsymbol{\theta}^+}(y) - \sum_{x,y} K(x,y)p_{\theta}(x)p_{\boldsymbol{\theta}^-}(y)\right)\\ &- \left(\sum_{x,y} K(x,y)p_{\boldsymbol{\theta}^+}(x)\pi(y) - \sum_{x,y} K(x,y)p_{\boldsymbol{\theta}^-}(x)\pi(y)\right).\end{aligned} \tag{29}$$

Using the symmetric condition of the kernel $K(x,y) = K(y,x)$, we arrive at

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta_l^\alpha} = &\mathop{\mathbb{E}}_{x\sim p_{\boldsymbol{\theta}^+}, y\sim p_{\boldsymbol{\theta}}}[K(x,y)] - \mathop{\mathbb{E}}_{x\sim p_{\boldsymbol{\theta}^-}, y\sim p_{\boldsymbol{\theta}}}[K(x,y)]\\ &- \mathop{\mathbb{E}}_{x\sim p_{\boldsymbol{\theta}^+}, y\sim \pi}[K(x,y)] + \mathop{\mathbb{E}}_{x\sim p_{\boldsymbol{\theta}^-}, y\sim \pi}[K(x,y)].\end{aligned} \tag{30}$$

This gradient algorithm for QCBM scales as $O(d^2)$ where d is the depth of the circuit, which is less efficient compared to the linear scaling back-propagation algorithm for classical neural networks. It is still unknown whether one can reach similar scaling on a quantum computer since the back-propagation algorithm requires cached intermediate results in the forward evaluation pass.

III. MAPPING A QUANTUM CIRCUIT TO A TENSOR NETWORK

Mapping a quantum circuit to a tensor network [16–19] is straight forward. One replace gates with tensors. But is this the best approach for simulation?

As show Appendix B, a tensor in a quantum circuit can be very sparse, it may contain a lot of diagonal tensors or even δ tensors. It is more efficient for simulation to view a quantum circuit as a line graph.

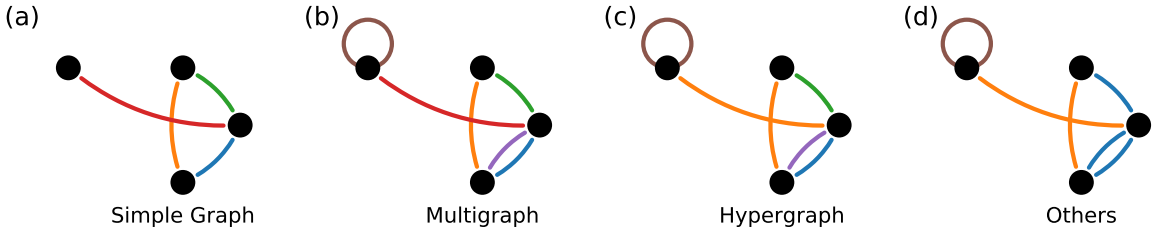


FIG. 7. Graph types that may appear in an Einstein notation.

IV. MAPPING A TENSOR NETWORK TO A QUANTUM CIRCUIT

V. ACKNOWLEDGMENT

Thank Lei Wang for helpful advices.

Appendix A: Quantum Circuit Transformations

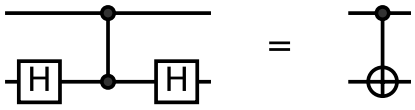


FIG. 8. Rule R1

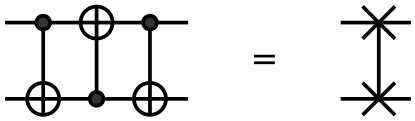


FIG. 9. Rule R2

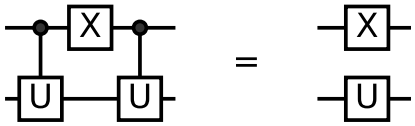


FIG. 10. Rule R3

Appendix B: Gates and their tensor network representations

For a detailed list of matrix representations of quantum gates, I would suggest referencing Chapter 2 of Ref. 20. Some special operators in the main text, like projection operators, are defined as

$$\mathrm{P}^{+} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{B1}$$

$$\mathrm{P}^{-} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{B2}$$

$$\mathrm{P}^{0} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{B3}$$

$$\mathrm{P}^{1} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{B4}$$

Some gates has convenient tensor decomposition

$$\mathrm{XOR}_{ijk} := \delta_{k(i \veebar j)} \tag{B5}$$

$$\mathrm{CNOT}_{ijkl} := \delta_{ijc}\mathrm{XOR}_{klc} \tag{B6}$$

$$\mathrm{CPHASE}(\theta)_{ijkl} := \delta_{ijc}(\delta_{dkl}\mathrm{P}(\theta)_{cd}) \tag{B7}$$

$$\mathrm{CZ}_{ijkl} := \delta_{ijc}(\delta_{dkl}\mathrm{P}(\pi)_{cd}) \tag{B8}$$

$$\mathrm{SWAP}_{ijkl} := \delta_{kj}\delta_{il} \tag{B9}$$

The above representations follows Einstein summation notations, where same indices are contracted. $a \veebar b$ returns 1 if $a \neq b$ else 0. And

$$\mathrm{P}(\theta) = \begin{pmatrix} 1 & 1 \\ 1 & e^{i\theta} \end{pmatrix} \tag{B10}$$

[1] A. K. Ekert, C. M. Alves, D. K. L. Oi, M. Horodecki, P. Horodecki, and L. C. Kwek, Phys. Rev. Lett. **88**, 217901 (2002).
[2] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017).
[3] X. Gao, Z. Zhang, and L. Duan, arXiv:1711.02038.
[4] M. Schuld and N. Killoran, Phys. Rev. Lett. **122**, 040504 (2019).
[5] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, New Journal of Physics **20**, 113022 (2018).
[6] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Phys. Rev. Lett. **87**, 167902 (2001).
[7] K. M. Nakanishi, K. Fujii, and S. Todo, arXiv:1903.12166.
[8] J. Li, X. Yang, X. Peng, and C.-P. Sun, Phys. Rev. Lett. **118**, 150503 (2017).
[9] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Phys. Rev. A **98**, 032309 (2018).
[10] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nat. Commun. **9**, 4812 (2018).
[11] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, Nat. Commun. **5**, 4213 (2014).
[12] D. P. Kingma and J. Ba, arXiv:1412.6980.
[13] M. Benedetti, E. Grant, L. Wossnig, and S. Severini, New Journal of Physics **21**, 043023 (2019).
[14] S. Ruder, arXiv:1609.04747.
[15] J. Li, X. Yang, X. Peng, and C.-P. Sun, Phys. Rev. Lett. **118**, 150503 (2017).
[16] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, and H. Neven, arXiv:1712.05384.
[17] I. L. Markov and Y. Shi, SIAM Journal on Computing **38**, 963 (2008).
[18] J. Biamonte and V. Bergholm, arXiv:1708.00006.
[19] J. Chen, F. Zhang, M. Chen, C. Huang, M. Newman, and Y. Shi, arXiv:1805.01450.
[20] C. P. Williams, "Quantum gates," in *Explorations in Quantum Computing* (Springer London, London, 2011) pp. 51–122.