Is estimating the gradient make quantum machine learning model worse?

Vu Tuan Hai
University of Information Technology, Vietnam National University - Ho Chi Minh City (VNUHCM), Ho Chi Minh City 700000, Vietnam
E-mail: haivt@uit.edu.vn

Phan Hoang Chuong
University of Information Technology, Vietnam National University - Ho Chi Minh City (VNUHCM), Ho Chi Minh City 700000, Vietnam
E-mail: chuongph@uit.edu.vn

Pham The Bao
Saigon University, Ho Chi Minh City 700000, Vietnam
E-mail: ptbao@sgu.edu.vn

Keywords: quantum computing; machine learning; parameter-shift rule

Received: June 15, 2022

The parameter-shift rule is an approach to evaluating gradients in many variational quantum algorithms. Although the general parameter-shift rule has been proposed, it needs at least linear quantum evaluations regarding the number of eigenvalues of the generator of the quantum gate. In the paper, we discuss an approximate method named the pseudo-parameter-shift rule, this method can reduce the number of quantum evaluations to constant.

Povzetek:

1 Introduction

Variational quantum algorithms [1] (VQA) have come to the popular tool in the noisy intermediate-scale quantum (NISQ) era [2]. It can be used to solve various problems from quantum tomography, quantum measurement, [3, 4], quantum compilation [5], ... to quantum machine learning [6, 7, 8, 9, 10]. The idea behind VQA follows a strategy: an ansatz, or parameterized quantum circuit is initialized. When it's executed by many measurements, we get the value of cost function from expected value. After that, some quantum / classical optimization is used to minimize that cost function. Initially, the optimization task is solved by some gradient-free optimizers such as Nelder - Mead or COBYLA. However, like classical machine learning field, [11, 12], the first-order gradient optimizers include: SGD, Momentum, Adagrad, Adadelta, Adam, ... [13] provides significant advantages. So, we also implement these algorithm in VQA via a technique named parameter-shift rule [14, 15, 16, 17, 20].

Although gradient-based methods are better but it also require more computation resources because the parameter-shift rule need some quantum evaluations. The parameter-shift rule formula is originally used only the case of quantum gate has two distinct eigenvalues [14] and just developed for R number of distinct eigenvalues by discrete Fourier transform, so-called general parameter-shift rule [17]. It require at least $\mathcal{O}(2R+1)$ quantum evaluation with increase linearly with regards to the size of quantum gate. In this paper, we proposed an approximate method that reduce the number of evaluation.

2 Background

Consider a variational circuit $U(\boldsymbol{\theta})$ parametrized by $[\theta_0, \theta_1, \theta_2, ..., \theta_m]^T$ that has n qubits. This circuit contains a sequence of fixed and parametrized layers. We can rewrite $U(\boldsymbol{\theta})$ as:

$$\begin{aligned} U(\boldsymbol{\theta}) :=& U_L(\boldsymbol{\theta}_L)V_L U_{L-1}(\boldsymbol{\theta}_{L-1})V_{L-1} \\ & \dots U_1(\boldsymbol{\theta}_1)V_1 U_0(\boldsymbol{\theta}_0)V_0 \end{aligned} \tag{1}$$

All $U_\ell(\boldsymbol{\theta}_\ell) = \{U_j\}$ and $V_\ell(\boldsymbol{\theta}_\ell) = \{V_j\}$ are the combination of the below common parameterized gates and fixed gates, respectively.

$$\begin{aligned} U_j \in \{&R_X, R_Y, R_Z, CR_X, CR_Y, CR_Z, \\ &R_{XX}, R_{YY}, R_{ZZ}, \dots\}, \\ V_j \in \{&X, Y, Z, CX, CY, CZ, H, \dots\}. \end{aligned}$$

V_j is fixed so it does not affect to computing gradient process, we can ignore V_j in computing gradient process. Besides, each parameterized gate can be represented at exponential form: $U_j(\theta_j) = e^{-i\theta_j G} = \cos(\theta_j)\mathbb{I} - i\sin(\theta_j)G$ with G as the generator. Consider the cost function of $U(\boldsymbol{\theta})$, or U in short-handed from:

$$E(\boldsymbol{\theta}) := \langle\psi|U^\dagger \hat{B} U|\psi\rangle, \tag{2}$$

where $\hat{B}$ is the measurement operator and ψ is the initial quantum state (simply as $|0\rangle^{\otimes n}$). The target here is to finding $\boldsymbol{\theta}^*$ such as $E(\boldsymbol{\theta})$ achieve the minimum value:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} E(\boldsymbol{\theta}). \tag{3}$$

This task can be done iteratively by some classical optimizers, such as gradient descent:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha \nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}^t), \tag{4}$$

where $\alpha \in \mathbb{R}_{\geqslant 0}$ is the fixed learning rate and $\nabla E(\boldsymbol{\theta}^t) = \langle\psi|\nabla(U(\boldsymbol{\theta}^t)^\dagger \hat{B} U(\boldsymbol{\theta}^t))|\psi\rangle = [\partial_{\theta_0} E(\boldsymbol{\theta}^t)\ \partial_{\theta_1} E(\boldsymbol{\theta}^t)\ \dots]^T$.

Or Adam optimizer:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{5}$$

where:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla_{\boldsymbol{\theta}} \mathcal{C}(\boldsymbol{\theta}), \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2)\nabla^2_{\boldsymbol{\theta}} \mathcal{C}(\boldsymbol{\theta}), \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t}, \end{aligned} \tag{6}$$

with the hyper-parameters $\beta_1 = 0.8$ or 0.9, $\beta_2 \approx 1$ and $\epsilon \approx 0$. This gradient has the exact same form as the single-gate case. In terms of the circuit, this means we can leave all other gates as they are, and only modify gate $U_j(\theta_j)$ when we want to differentiate with respect to the parameter θ_i.

2.1 Parameter-shift rule

Taking first-order partial derivative of $E(\boldsymbol{\theta})$ with respect to θ_j:

$$\begin{aligned} \partial_{\boldsymbol{\theta}_j} E(\theta) &= \partial_{\theta_j}\langle\psi|U^\dagger \hat{B} U|\psi\rangle \\ &= \langle\psi|(\partial_{\theta_j} U)^\dagger \hat{B} U|\psi\rangle + \langle\psi|U^\dagger \hat{B}(\partial_{\theta_j} U)|\psi\rangle \\ &= \langle\psi|(-iGU)^\dagger \hat{B} U|\psi\rangle + \langle\psi|U^\dagger \hat{B}(-iGU)|\psi\rangle \\ &= -i\langle\psi|(U^\dagger[\hat{B}, G]U|\psi\rangle, \end{aligned} \tag{7}$$

where $[\hat{B}, G] = \hat{B}G - G\hat{B}$ is the commutator. For a generator that has two distinguish eigenvalues, such as the standard rotation gates based on Pauli matrices, e.g. $G \in \{\sigma_X, \sigma_Y, \sigma_Z\}$, the Eq. 7 return the two-term parameter-shift rule [14]:

$$\partial_{\theta_j} E(\boldsymbol{\theta}) = \frac{1}{2\sin(s)}[E(\boldsymbol{\theta} + s\boldsymbol{e}_j) - E(\boldsymbol{\theta} - s\boldsymbol{e}_j)], \tag{8}$$

where $\boldsymbol{e}_j = [a_0\ a_1 \dots a_m]^T$ ($a_i = 0\ \forall i \neq j$) and s is an freely shift value, $s \in (0, \pi)$. Popularly, if $s = \frac{\pi}{2}$, we obtain the parameter-shift rule already used in Pennylane [18] and TensorFlow Quantum [19]. Because Eq. 8 is the analytic gradient, not approximate gradient so we can apply it again to get higher-order derivative, such as second-order derivative with respect to any parameters θ_j, θ_k.

$$\begin{aligned}\partial_{\theta_j\theta_k}E(\boldsymbol{\theta}) &= \frac{1}{2\sin(s_1)}[\partial_{\theta_k}E(\boldsymbol{\theta}+s_1\boldsymbol{e}_j)\\ &- \partial_{\theta_k}E(\boldsymbol{\theta}-s_1\boldsymbol{e}_j)]\\ &= \frac{1}{4\sin(s_1)\sin(s_2)}[E(\boldsymbol{\theta}+s_1\boldsymbol{e}_j+s_2\boldsymbol{e}_k)\\ &- E(\boldsymbol{\theta}+s_1\boldsymbol{e}_j-s_2\boldsymbol{e}_k)\\ &- E(\boldsymbol{\theta}-s_1\boldsymbol{e}_j+s_2\boldsymbol{e}_k)\\ &+ E(\boldsymbol{\theta}-s_1\boldsymbol{e}_j-s_2\boldsymbol{e}_k)].\end{aligned} \tag{9}$$

This formula has original shown in Refs. [20, 21, 14, 15] and extended to the Hessian. For k-order derivative form with $k > 0$:

$$\partial_{\theta_0\theta_1\ldots\theta_k}E(\theta) = \frac{\Sigma_{\boldsymbol{s}\in\mathbf{S}_{\pm 0,\pm 1,\ldots,m}}P(\boldsymbol{k})E(\boldsymbol{\theta}+\boldsymbol{s})}{2^k(\prod_{j=1}^{k}\sin(s_j))}, \tag{10}$$

where:

$$\mathbf{S}_{\pm 0,\pm 1,\ldots,\pm k} = [\pm s_0 e_0 \ \pm s_1 e_1 \ \ldots \ \pm s_k e_k]^T$$
$$\text{and } P(\boldsymbol{s}) = \operatorname{sgn}\left(\prod_{j=0}^{k} s_j e_j\right).$$

Follow the Eq. 10, we know that the necessary number of expectation values to evaluate the n-order derivative that is the number of elements in the sum, or 2^k. As the analysis in Ref. [20], the derivative tensor of order k is symmetric, so the real number of quantum evaluation equals to the number of distinct elements, which is less than $2^k\binom{m+k-1}{k}$, this value equal $\mathcal{O}(m^k)$ in case $m \gg k$.

In this section, we know that when we want to take high order derivative, the computation resources will increase exponentially with respect to the order k.

2.2 General parameter-shift rule

Let analysis the $U(\theta)$ again, here θ is the single parameter. In Ref. [17], they consider the eigenvalues of $U(\theta)$ are given by $\{e^{i\omega_j\theta}\}_{j\in[d]}$ where $[d] \coloneqq \{1,\ldots,d\}$, and expanded B and $|\psi\rangle$ in the eigenbasis of U.

$$\begin{aligned}E(\theta) &\coloneqq \langle\psi|U^\dagger\hat{B}U|\psi\rangle\\ &= \sum_{j,k=1}^{d}\overline{\psi_j e^{i\omega_j\theta}}b_{jk}\psi_k e^{i\omega_k\theta}\\ &= \sum_{\substack{j,k=1\\ j<k}}^{d}\Big[\overline{\psi_j}b_{jk}\psi_k e^{i(\omega_k-\omega_j)\theta}\\ &+\psi_j\overline{b_{jk}\psi_k e^{i(\omega_k-\omega_j)\theta}}\Big] + \sum_{j=1}^{d}|\psi_j|^2 b_{jj},\end{aligned} \tag{11}$$

After that, they define the coefficients $c_{jk} \coloneqq \overline{\psi_j}b_{jk}\psi_k$ and introduce the R unique positive differences $\{\Omega_\ell\}_{\ell\in[R]} \coloneqq \{\omega_k-\omega_j \mid j,k\in[d],\omega_k > \omega_j\}$. Note that the differences are not necessarily equidistant, and that for $r = \left|\{\omega_j\}_{j\in[d]}\right|$ unique eigenvalues of the gate generator, there are at most $R \leqslant \frac{r(r-1)}{2}$ unique differences. However, many quantum gates will yield $R \leqslant r$ equidistant differences instead. In the following, we implicitly assume a mapping between the two indices $j,k \in [d]$ and the frequency index $\ell \in [R]$ such that $c_\ell = c_{\ell(j,k)}$ is well-defined.

By defining $c_\ell =: \frac{1}{2}(a_\ell - ib_\ell)\,\forall\ell \in [R]$ with $a_\ell, b_\ell \in \mathbb{R}$, and $a_0 \coloneqq \sum_j |\psi_j|^2 b_{jj} \in \mathbb{R}$, we can then rewrite the expectation value $E(\theta)$ as a finite-term Fourier series [22]:

$$E(\theta) = a_0 + \sum_{\ell=1}^{R} a_\ell\cos(\Omega_\ell\theta) + b_\ell\sin(\Omega_\ell\theta) \tag{12}$$

Here $\{a_\ell\}_{\ell=0}^{R}, \{b_\ell\}_{\ell=1}^{R}$ are unknown. This task can be solved easily through $2R+1$ quantum evaluation at $2R+1$ point $\{\hat{\theta}_\mu = \frac{2\pi\mu}{2R+1}\}$ where $\mu = -\{R,\ldots,R\}$.

First, we separate the left-hand in Eq. 12 as:

$$\begin{aligned}E_{\text{odd}}(\theta) &= \sum_{\ell=1}^{R} b_\ell\sin(\Omega_\ell\theta)\\ &= \frac{1}{2}(E(\theta)-E(-\theta))\end{aligned} \tag{13}$$

and:

$$\begin{aligned}E_{\text{even}}(\theta) &= a_0 + \sum_{\ell=1}^{R} a_\ell\cos(\Omega_\ell\theta)\\ &= \frac{1}{2}(E(\theta)+E(-\theta))\end{aligned} \tag{14}$$

Let define $\Omega_{j,\mu} := \sin(\Omega_j \hat{\theta}_\mu)$, the odd part need $2R$ evaluation to create the R variables system of linear equations:

$$\begin{cases} b_1\Omega_{1,1} + b_2\Omega_{2,1} + \ldots + b_R\Omega_{R,1} = E_{\text{odd}}(\hat{\theta}_1) \\ b_1\Omega_{1,2} + b_2\Omega_{2,2} + \ldots + b_R\Omega_{R,2} = E_{\text{odd}}(\hat{\theta}_2) \\ \vdots \\ b_1\Omega_{1,R} + b_2\Omega_{2,R} + \ldots + b_R\Omega_{R,R} = E_{\text{odd}}(\hat{\theta}_R) \end{cases}$$

Similarly, we can re-use $2R$ values of $E_{\text{odd}}(\hat{\theta}_\mu)$ to compute the coefficients $\{a_\ell\}_{\ell=1}^R$ and one more special point, e.g: 0, for a_0. The total is $2R+1$ evaluations.

After known the value of all coefficients $\{a_\ell\}_{\ell=0}^R, \{b_\ell\}_{\ell=1}^R$, the derivative of $E(\theta)$ can be calculated the $E_{\text{odd}}(\theta)$ or $E_{\text{even}}(\theta)$ at any θ without more evaluation:

$$\begin{aligned} E_{\text{odd}}(\theta) &= \sum_{k=1}^{R} E_{\text{odd}}\left(\hat{\theta}_k\right) \tilde{D}_{(\text{odd})k}(\theta) \\ E_{\text{even}}(\theta) &= \sum_{k=0}^{R} E_{\text{even}}\left(\hat{\theta}_k\right) \tilde{D}_{(\text{even})k}(\theta) \end{aligned} \tag{15}$$

Here $\{E_{\text{odd}}(\hat{\theta}_k)\}, \{E_{\text{even}}(\hat{\theta}_k)\}$ are known and $\{\tilde{D}_{(\text{odd})k}(\theta)\}, \{\tilde{D}_{(\text{even})k}(\theta)\}$ are the modified Dirichlet kernel which can be computed analytically. Note that the derivative of E can also expressed as:

$$\partial_\theta E(\theta) = \sum_{\ell=1}^{R} \Omega_\ell \left(b_\ell \cos(\Omega_\ell\theta) - a_\ell \sin(\Omega_\ell\theta)\right) \tag{16}$$

In the case $\theta = 0$, it only depends on the odd part of E, we arrive at the general parameter-shift rule [17]:

$$\begin{aligned} \partial_\theta E(0) &= \sum_{\ell=1}^{R} b_\ell\Omega_\ell \\ &= \sum_{\ell=1}^{R} E_{\text{odd}}\left(\hat{\theta}_\ell\right) \tilde{D}'_{(\text{odd})\ell}(0) \\ &= \sum_{\ell=1}^{R} E_{\text{odd}}\left(\hat{\theta}_\ell\right) \frac{(-1)^{\ell-1}}{2R\sin^2\left(\frac{2\ell-1}{4R}\pi\right)} \end{aligned} \tag{17}$$

Consider the differences:

$$\begin{aligned} &E(\theta+s) - E(\theta-s) \\ &= \sum_{\ell=1}^{R} (a_\ell \left(\cos\left(\Omega_\ell\theta + s\right) - \cos\left(\Omega_\ell\theta - s\right)\right) + \\ &b_\ell(\sin\left(\Omega_\ell\theta + s\right) - \sin\left(\Omega_\ell\theta - s\right))) \\ &= \sum_{\ell=1}^{R} (a_\ell(-2\sin(\Omega_\ell\theta)\sin(\Omega_\ell s) + \\ &b_\ell(2\cos(\Omega_\ell\theta)\sin(\Omega_\ell s))) \end{aligned} \tag{18}$$

For a quantum gate that its generator is Pauli matrix $G \in \{\sigma_X, \sigma_Y, \sigma_Z\}$. The eigenvalues of G are $\{-1, 1\}$, we have $R = 1$ and the Eq. 18 is equivalent to:

$$\begin{aligned} &E(\theta+s) - E(\theta-s) \\ &= 2\sin(\Omega_1 s)(b_\ell\cos(\Omega_1\theta) - a_\ell\sin(\Omega_1\theta)) \end{aligned} \tag{19}$$

with $\Omega_1 = 1$, it similar to the two-term parameter-shift rule in Eq. 8, in case θ is a single parameter. When facing with the gate that has three eigenvalue $\{-1, 0, 1\}$ such as control rotation gate $\{CR_X, CR_Y, CR_Z\}$, we have $R = 2$ [23] and the derivative of $E(\theta)$ now is:

$$\begin{aligned} \partial_\theta E(\theta) = \Omega_1 &\left(b_1\cos(\Omega_1\theta) - a_1\sin(\Omega_1\theta)\right) \\ + \Omega_2 &\left(b_2\cos(\Omega_2\theta) - a_2\sin(\Omega_2\theta)\right) \end{aligned} \tag{20}$$

That means we need at least four quantum evaluation with two different shift values $\{s_1, s_2\}$ to achieve the left-hand in the above equation:

$$\begin{aligned} \partial_\theta E(\theta) = d_1 &\left[E\left(\theta + s_1\right) - E\left(\theta - s_1\right)\right] \\ + d_2 &\left[E\left(\theta + s_2\right) - E\left(\theta - s_2\right)\right] \end{aligned} \tag{21}$$

where:

$$\begin{cases} d_1\sin(\frac{s_1}{2}) - d_2\sin(\frac{s_2}{2}) = \frac{1}{4} \\ d_1\sin(s_1) - d_2\sin(s_2) = \frac{1}{2} \end{cases} \tag{22}$$

where we can choose set $\{d_{1,2}, s_{1,2}\}$ which satisfy the system of linear equation 22. A particularly symmetric solution is $s_{1,2} = \pi/2 \mp \pi/4$ with coefficients $d_{1,2} = (\sqrt{2} \pm 1)/(2\sqrt{2})$, or $s_{1,2} = 3\pi/4 \mp \pi/4$ with coefficients $d_{1,2} = (...)$. This is known as the four-term parameter-shift rule formula [24].

In general, for arbitrary quantum gate that has R distinct eigenvalues, we have $2R$ - term parameter-shift rule:

$$\partial_\theta E(\theta) = \sum_{k=1}^{R} d_k \left[E\left(\theta + s_k\right) - E\left(\theta - s_k\right) \right] \quad (23)$$

In this section, we show that the computation resource will increase linearly according the number of eigenvalues of generator.

3 Method

Here, we propose a approximate method which can compute either high-order and high-term derivative, named pseudo - parameter-shift rule. Consider $E(\theta)$ with θ is a single parameter, assuming that $E(\theta)$ is differentiable at $(\theta + s)$ and $(\theta - s)$, use Taylor expansion:

$$\begin{aligned} E(\theta + s) &\approx E(\theta) + \frac{(\theta + s - \theta)}{1!}\partial_\theta E(\theta) + \dots \\ &= E(\theta) + s\partial_\theta E(\theta) + \dots \\ E(\theta - s) &\approx E(\theta) + \frac{(\theta - s - \theta)}{1!}\partial_\theta E(\theta) + \dots \\ &= E(\theta) - s\partial_\theta E(\theta) + \dots \end{aligned} \quad (24)$$

Then:

$$\begin{aligned} \frac{E(\theta + s) - E(\theta - s)}{2s} &\approx \partial_\theta E(\theta) + \frac{s^2}{6}\frac{\partial^3}{\partial\theta^3}E(\theta) + \dots \\ &= \partial_\theta E(\theta) + O(s^2). \end{aligned} \quad (25)$$

Or:

$$k\partial_\theta \hat{E}(\theta) = \frac{E(\theta + s) - E(\theta - s)}{2s}. \quad (26)$$

where $r \in [0, 1]$ is the multiplicative constant, which is used to reduce the error. The formula 26 is also known as scaled parameter-shift gradient estimator [20].

The Eq. 26 is equivalent the pseudo 2-term parameter-shift rule when it's divided by k:

$$\begin{aligned} \partial_\theta \hat{E}(\theta) &= \frac{1}{2sk}[E(\theta + s) + E(\theta - s)] \\ &= r[E(\theta + s) + E(\theta - s)] \\ &:= F(\theta, r, s), \end{aligned} \quad (27)$$

where $\{r, s\} \in \mathbb{R}_+$ that can be find via optimizing the absolute error between functions E and F at the fixed point θ_0:

$$\begin{aligned} \Delta F(\theta_0) &= \mathbb{E}[\|F(\theta_0) - E(\theta_0)\|] \\ &= \sum_{r,s} \mathbb{E}[|F(\theta_0, r, s) - E(\theta_0)|], \end{aligned} \quad (28)$$

such that:

$$\{r^*, s^*\} := \arg\min_{\{r,s\}}(\Delta F(\theta_0)). \quad (29)$$

In general, r, s are actually parameter vectors, re-notation as $\boldsymbol{r}, \boldsymbol{s}$, we have pseudo general parameter-shift rule:

$$F(\theta, \boldsymbol{r}, \boldsymbol{s}) = \sum_{k=1}^{|\boldsymbol{r}|} r_i \left[E(\theta + s_k) - E(\theta - s_k) \right] \quad (30)$$

Note that we can choose shift values $\{s_j\}$ (particularly symmetric) automatically and only need to find $\{r_j\}$. Here, $\boldsymbol{r}$ must to be satisfy some equations, such as $r = 2\sin^{-1}(s)$ in two-term parameter-shift rule 19, or system of linear equations 22 in four-term parameter-shift rule.

This optimization task can be conducted by some free-gradient methods, such as grid search and spatial search. Its main idea is to find a lowest point in a $|r|$ - dimensional space.

3.1 Grid search

Grid search used in hyperparameter optimization [25] which split the search space $\mathbb{S}$ into many discrete points with the size $n_R \times n_S, n_R = \frac{2R}{\Delta r}, n_S = \frac{2S}{\Delta s}$, where $R \in [0, 2\pi]$ and $S \in \mathcal{R}$ are bound of $\mathbb{S}$, smallest share Δr and Δs. The number of point can be increased by decrease the value of Δr or Δs or both. The complexity of Grid search is $\mathcal{O}(n^2)$. Obviously, the larger the value of n_R and n_S (or the smaller the value of Δr and Δs, the smaller distance between $\{r, s\}$ and $\{r^*, s^*\}$ is. But increasing the value of it means increasing the computation time, so like other optimization algorithms, we need to balance between the accuracy and existing resources. The algorithm is described as Algo. 3.1.

Algorithm 1 Grid search

Input: the bound R and S, smallest share Δr and Δs, the value $\partial_x E(0)$ and $E(x_i)$ at $\frac{4RS}{\Delta r \Delta s}$ points.
Output: $\{r^*, s^*\}$

$n_R \leftarrow \frac{2R}{\Delta r}, rs \leftarrow \{r : r \leftarrow -R + \Delta r \times n, n \in \{0, 1, \dots n_R\}\}$
$n_S \leftarrow \frac{2S}{\Delta s}, ss \leftarrow \{s : s \leftarrow -S + \Delta s \times n, n \in \{0, 1, \dots n_S\}\}$
$\Delta \leftarrow$ (empty list)
for r in rs **do**
  for s in ss **do**
    $\partial_x E_{r,s}(0) \leftarrow PSR(E, 0)$
    $\Delta_{r,s} \leftarrow |\partial_x E(0) - \partial_x E_{r,s}(0)|$
    Δ.insert($\Delta_{r,s}$)
  end for
end for
return $\arg\min_{\{r,s\}} \Delta$ =0

3.2 Spatial search

Based on the survey from Grid search, we know that $\mathbb{S}$ is a landscape, not a randomness space, so we can take advantage the decrease direction. In spatial search, we decay the complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(\log n)$. The main idea behind spatial search is taking a point as center point O and compute the distance of points around O. These point these are about $\mathcal{R}$ units away from O. After that, we choose the lowest point as new center and repeat this procedure until meet the minimum distance. If the center and the lowest point are far from each other, we double the value $\mathcal{R}$ each step, otherwise, we bipartite the value $\mathcal{R}$ each step.

In pseudo 2-term parameter-shift rule, the minimum number of point around O is 4 regarding to up, down, left and right direction. Obviously, we can increase the number of point to coverage faster, but require more computation resources. Take example: from $X = \{x_{\pm}, x^{\pm}\}$ to $X = \{x_+, x_-, x^+, x^-, x^+_+, x^+_-, x^-_+, x^-_-\}$ where $x^{\pm}_{\mp} = \{r_0 \pm \mathcal{R}, s_0 \mp \mathcal{R}\}$.

In general try pseudo 2^n-term parameter-shift rule has the complexity $\mathcal{O}(nk\log(2))$ with k is the average number of evaluation per updating step. k is increase linearly with the min value of $|X|$. In case of $n = 1, k = \frac{50}{9}$.

3.3 The effective

Take the full-form of $E(x)$ via DFT will take $2R+1$ evaluation. For compare, we assume that all eigenvalues are equidistant, mean $2R+1 = 2n-1$. Then:

$$2n - 1 > \frac{50 \log 2}{9} \rightarrow n > 2.42 \tag{31}$$

It mean our method will be less evaluation in case $n \geqslant 3$.

4 Application

We test on circuit ... Test on QML model, quanvolutional neural network. In quanvolutional layer, calculating gradient by normal parameter-shift rule => how resources consume?

Change to pseudo parameter-shift rule => Is resources consume less, is accuracy consume less.

5 Conclusion

In this paper, we introduced an approximate method for the general parameter-shift rule. This method is based on the assumption that $E(x)$ is differentiable at almost points in parameter space $x \in [0, 2\pi]$. In case the pseudo-2-term PSR has the distance $\Delta_{r*,s*} \not\approx 0$, we can try the 2^n-pseudo PSR ($n > 1$) which requires more number of evaluation per step.

This research mention a problem about the lowest distance that lower-term pseudo PSR can achieve. Take example: what is the gap between 4-term PSR and pseudo 2-term PSR?

Furthermore, we are developing a method based on Larange interpolation that ...

Algorithm 2 Spatial search

Input: the initial radius $\mathcal{R} \in \mathbb{R}$, the threshold $T \approx 0$, the max number of loop $t_{\max}$, the value $\partial_x E(0)$ and $E(x_i)$ at ... points.

Output: $\{r^*, s^*\}$

Set the initial center $\mathcal{O} = \{0, 0\}$,
while True **do**
  $t = 0$
  Create $X \leftarrow \{x_{\pm}, x^{\pm}\}$ based on $x_0 = \mathcal{O}^t$
  for $\{r, s\}$ in X **do**
    $\partial_x E_{r,s}(0) \leftarrow PSR(E, 0)$
    $\Delta_{r,s} \leftarrow |\partial_x E(0) - \partial_x E_{r,s}(0)|$
    Δ.insert$(\Delta_{r,s})$
  end for
  min $\leftarrow \min_{\{r,s\} \in X}(\Delta_{r,s})$
  if min $< T$ or $t = t_{\max}$ **then**
    return $\mathcal{O}$
  end if
  $\mathcal{O}^{t+1} \leftarrow$ min [1]
  if $\mathcal{O}^{t+1} \neq \mathcal{O}^t$ **then**
    $\mathcal{R} \leftarrow 2\mathcal{R}$
  else
    $\mathcal{R} \leftarrow \mathcal{R}/2$
  end if
  $t \leftarrow t + 1$
end while=0

[1] If multiple points have the same error, we will prioritize the center point first, then the points from the top left corner to the bottom right corner.

6 Acknowledgements

This work is supported by VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

[1] Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S., Endo, S., Fujii, K., McClean, J., Mitarai, K., Yuan, X., Cincio, L. & Others Variational quantum algorithms. *Nature Reviews Physics*. **3**, 625-644 (2021). https://doi.org/10.1038/s42254-021-00348-9

[2] Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum*. **2** pp. 79 (2018). https://doi.org/10.22331/q-2018-08-06-79

[3] Xin, T., Nie, X., Kong, X., Wen, J., Lu, D. & Li, J. Quantum pure state tomography via variational hybrid quantum-classical method. *Physical Review Applied*. **13**, 024013 (2020). https://doi.org/10.1103/PhysRevApplied.13.024013

[4] Carolan, J., Mohseni, M., Olson, J., Prabhu, M., Chen, C., Bunandar, D., Niu, M., Harris, N., Wong, F., Hochberg, M. & Others Variational quantum unsampling on a quantum photonic processor. *Nature Physics*. **16**, 322-327 (2020). https://doi.org/10.1038/s41567-019-0747-6

[5] Jones, T. & Benjamin, S. Robust quantum compilation and circuit optimisation via energy minimisation. *Quantum*. **6** pp. 628 (2022) https://doi.org/10.22331/q-2022-01-24-628

[6] Bharti, K., Cervera-Lierta, A., Kyaw, T., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J., Menke, T. & Others Noisy intermediate-scale quantum algorithms. *Reviews Of Modern Physics*. **94**, 015004 (2022) https://doi.org/10.1103/RevModPhys.94.015004

[7] Guan, W., Perdue, G., Pesah, A., Schuld, M., Terashi, K., Vallecorsa, S. & Vlimant, J. Quantum machine learning in high energy physics. *Machine Learning: Science And Technology*. **2**, 011003 (2021) https://doi.org/10.1088/2632-2153/abc17d

[8] Wu, S. & Yoo, S. Challenges and opportunities in quantum machine learning for high-energy physics. *Nature Reviews Physics*. **4**, 143-144 (2022) https://doi.org/10.1038/s42254-022-00425-7

[9] Schuld, M. & Petruccione, F. Machine learning with quantum computers. (Springer,2021) https://doi.org/10.1007/978-3-030-83098-4

[10] Schuld, M. & Killoran, N. Is quantum advantage the right goal for quantum machine learning?. *ArXiv Preprint ArXiv:2203.01340*. (2022) https://doi.org/10.48550/arXiv.2203.01340

[11] Bottou, L. Large-scale machine learning with stochastic gradient descent. *Proceedings Of COMPSTAT'2010*. pp. 177-186 (2010) https://doi.org/10.1007/978-3-7908-2604-3$_1$6

[12] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M., Pfau, D., Schaul, T., Shillingford, B. & De Freitas, N. Learning to learn by gradient descent by gradient descent. *Advances In Neural Information Processing Systems*. **29** (2016)

[13] Ruder, S. An overview of gradient descent optimization algorithms. *ArXiv Preprint ArXiv:1609.04747*. (2016) https://doi.org/10.48550/arXiv.1609.04747

[14] Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A*. **98**, 032309 (2018,9) https://doi.org/10.1103/PhysRevA.98.032309

[15] Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*. **99**, 032331 (2019,3) https://doi.org/10.1103/PhysRevA.99.032331

[16] Mari, A., Bromley, T. & Killoran, N. Estimating the gradient and higher-order derivatives on quantum hardware. *Phys. Rev. A*. **103**, 012405 (2021,1) https://doi.org/10.1103/PhysRevA.103.012405

[17] Wierichs, D., Izaac, J., Wang, C. & Lin, C. General parameter-shift rules for quantum gradients. *Quantum*. **6** pp. 677 (2022,3) https://doi.org/10.22331/q-2022-03-30-677

[18] Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Alam, M., Ahmed, S., Arrazola, J., Blank, C., Delgado, A., Jahangiri, S. & Others Pennylane: Automatic differentiation of hybrid quantum-classical computations. *ArXiv Preprint ArXiv:1811.04968*. (2018) https://doi.org/10.48550/arXiv.1811.04968

[19] Broughton, M., Verdon, G., McCourt, T., Martinez, A., Yoo, J., Isakov, S., Massey, P., Halavati, R., Niu, M., Zlokapa, A. & Others Tensorflow quantum: A software framework for quantum machine learning. *ArXiv Preprint ArXiv:2003.02989*. (2020) https://doi.org/10.48550/arXiv.2003.02989

[20] Mari, A., Bromley, T. & Killoran, N. Estimating the gradient and higher-order derivatives on quantum hardware. *Physical Review A*. **103**, 012405 (2021) https://doi.org/10.1103/PhysRevA.103.012405

[21] Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Science And Technology*. **4**, 043001 (2019) https://doi.org/10.1088/2058-9565/ab4eb5

[22] Vidal, J. & Theis, D. Calculus on parameterized quantum circuits. *ArXiv Preprint ArXiv:1812.06323*. (2018) https://doi.org/10.48550/arXiv.1812.06323

[23] Kottmann, J., Anand, A. & Aspuru-Guzik, A. A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers. *Chemical Science*. **12**, 3497-3508 (2021) https://doi.org/10.1039/D0SC06627C

[24] Anselmetti, G., Wierichs, D., Gogolin, C. & Parrish, R. Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems. *New Journal Of Physics*. **23**, 113010 (2021) https://doi.org/10.1088/1367-2630/ac2cb3

[25] Liashchynskyi, P. & Liashchynskyi, P. Grid search, random search, genetic algorithm: a big comparison for NAS. *ArXiv Preprint ArXiv:1912.06059*. (2019) https://doi.org/10.48550/arXiv.1912.06059