LAWS: Look Around and Warm-Start Natural Gradient Descent for Quantum Neural Networks

Zeyi Tao, Jindi Wu, Qi Xia and Qun Li
Department of Computer Science
William & Mary
Williamsburg, VA, USA
{ztao, jwu21, qxia01, liqun}@cs.wm.edu

2023 IEEE International Conference on Quantum Software (QSW) | 979-8-3503-0479-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/QSW59989.2023.00019

***Abstract*—Variational quantum algorithms (VQAs) have recently received much attention due to their promising performance in Noisy Intermediate-Scale Quantum computers (NISQ). However, VQAs run on parameterized quantum circuits (PQC) with randomly initialized parameters are characterized by barren plateaus (BP) where the gradient vanishes exponentially in the number of qubits. In this paper, we proposed a Look Around Warm-Start (LAWS) quantum natural gradient (QNG) algorithm to mitigate the widespread existing BP issues. LAWS is a combinatorial optimization strategy taking advantage of model parameter initialization and fast convergence of QNG. LAWS repeatedly reinitializes parameter search space for the next iteration parameter update. The reinitialized parameter search space is carefully chosen by sampling the gradient close to the current optimal. Moreover, we present a unified framework (WS-SGD) for integrating parameter initialization techniques into the optimizer. We provide the convergence proof of the proposed framework for both convex and non-convex objective functions based on Polyak-Lojasiewicz (PL) condition. Our experiment results show that the proposed algorithm could mitigate the BP and have better generalization ability in quantum classification problems.**

***Index Terms*—Variational Quantum Algorithms, Natural Gradient Descent**

I. Introduction

A Quantum Neural Network (QNN) implements a neural network on a quantum computer. In QNN, a task of interest is prepared and evaluated via a parameterized quantum circuit (PQC) on a quantum computer, with iteratively updating the parameters by a classical optimizer to find the optimum for the objective function [1]–[3]. However, a recently discovered phenomenon, so-called barren plateaus (BP) [4], where gradients of the cost functions vanish exponentially with the size of the system, dramatically limits the application of QNNs to practical problems. BP prevents QNN's parameter update from gradient changes when using gradient-based optimizers. To acquire the gradient information, exponential resources might be used for sampling errors in quantum measurements.

To address the BP issue, gradient rescaling [5], [6], QNN's parameter initialization [7], [8], and gradient-free optimizations [9] have been studied. Our work is also motivated by addressing the BP issue. In this paper, we first review the gradient-based method, particularly the quantum natural gradient (QNG), from the viewpoint of mirror descent [10]. Then, we proposed a look around the warm-start QNG algorithm as a primary instrument to mitigate the BP issue. The proposed algorithm is based on two observations: First, the QNG can consistently find a global optimum and requires significantly fewer epochs than other optimizers [11]. This outperformance holds even for large system sizes (40 qubits), indicating that using QNG to solve the QNN problem is suitable. Second, the success of applying parameter initialization in QNN demonstrates a potential direction for mitigating the BP issue [7], where it withstand the possible failure of using the gradient-based [12] or gradient-free [13] algorithm. Based on the above, the intuition behind LAWS is that we repeatedly reinitialize the QNN's parameter while in training. We call this reinitialization during the training as warm-start. In this way, the fast convergence speed of QNG is adopted, and the BP could be mitigated via multiple parameter reinitializations.

The contributions of this paper are fourfold: (1) we propose a new derivation of QNG by using a classical first-order optimization scheme known as mirror descent; (2) we proposed a new algorithm named LAWS for solving QNN in general. Our experiment results show that the proposed algorithm could mitigate the BP issue and have better generalization ability in quantum classification problems; (3) based on LAWS, we propose a unified framework WS-SGD for the warm-start gradient descent algorithm that is easy to implement and compatible with the most current quantum learning libraries; (4) Lastly, we provide the convergence proof of the proposed framework for both convex and non-convex objective functions.

II. Related Work

The barren plateau (BP) phenomenon in the cost function landscape was originally discovered in [4] where it was shown that deep (unstructured) parameterized quantum circuits exhibit BPs when randomly initialized. Many works have been studied to mitigate BP, and they can be roughly categorized into two directions. The first type of approach uses problem-inspired ansatzes because problem-agnostic ansatzes, such as deep hardware efficient ansatzes, could exhibit barren plateaus due to their high expressibility [14], [15]. The approach, for example [14], relaxes search space during the optimization to a smaller space that contains the solution to the problem or that at least contains a good approximation to the solution while maintaining a low expressibility. Another line of study focuses on QNN initialization [7], [8]. Parameter initialization has been proven to be helpful in classical machine learning.

979-8-3503-0479-4/23/$31.00 ©2023 IEEE
DOI 10.1109/QSW59989.2023.00019

In [7], the proposed method uses the identity block strategy to limit the effective depth of the circuits used to calculate the first parameter update to avoid the QNN being stuck in a barren plateau at the start of training.

The natural gradient [16] (NG) automatically chooses gradient step size and moves in the steepest descent direction with respect to the Fisher information. The pioneering work [17] proposes QNG as part of a general-purpose optimization framework for variational quantum algorithms. QNG's computation is expensive; hence it becomes an obstacle for applying in both classical learning and VQA.

III. Background

Let's first introduce some notations we will use in the paper. For $\theta, \mu \in R^d$, let $\sqrt{\theta}, \theta \odot \mu$, and θ/μ denote the element-wise square root, multiplication, and division of the vectors. The $\|\theta\|_2^2$ is l_2-norm. We denote θ_k^t for parameter θ at t-th iteration k-th step.

A. First-order Optimization

Globally optimizing the objective function $\mathcal{C}(\theta)$ is impractical due to the nonconvexity. To this end, practitioners search for local optima by solving the following dynamical system

$$\theta_{t+1} = \arg\min_{\theta} \left\{ \langle \theta, \nabla \mathcal{C}(\theta_t) \rangle + \frac{1}{2\eta} \|\theta - \theta_t\|_2^2 \right\} \tag{1}$$

which is equivalent to the gradient descent in the form $\theta_{t+1} = \theta_t - \eta \nabla \mathcal{C}(\theta_t)$. Notice, the stochastic gradient descent (SGD) is obtained when $g_t = \nabla \mathcal{C}(\theta_t, \xi)$ where ξ is a sample drawn from dataset $\mathcal{D}$ such that $\mathbb{E}[g_t] = \nabla \mathcal{C}(\theta_t)$ is an unbiased estimator of $\nabla \mathcal{C}(\theta_t)$ and

$$\theta_{t+1} = \theta_t - \eta g_t \tag{2}$$

Optimization problem Eq.(1) or Eq.(2) is well-suited to assumptions regarding the objective function $\mathcal{C}$ which involve the Euclidean norm. The intuition behind optimization in Eq.(1) is objective function $\mathcal{C}$ is replaced by its linearization at θ_t plus a Euclidean distance term $\frac{1}{2\eta}\|\theta - \theta_t\|_2^2$, which prevents the next iterate θ_{t+1} from being too far from θ_t.

Instead of using Vanilla SGD above, recent studies tackle the optimization problem

$$\mathcal{C}(\theta) = \mathrm{Tr}(P_\psi H) = \langle \psi | H | \psi \rangle$$

by using natural gradient descent, where we update the parameter as

$$\theta_{t+1} = \theta_t - \eta F(\theta)^{-1} g_t \tag{3}$$

Here, $F(\theta) = \Re[G(\theta)]$ is Fubini-Study metric tensor a $P \times P$ matrix recently identified as the (classical) Fisher information matrix. We define quantum geometric tensor $G(\theta)$ as

$$G_{i,j} = \left\langle \frac{\partial \psi}{\partial \theta_i}, \frac{\partial \psi}{\partial \theta_j} \right\rangle - \left\langle \frac{\partial \psi}{\partial \theta_i}, \psi \right\rangle \left\langle \psi, \frac{\partial \psi}{\partial \theta_j} \right\rangle \tag{4}$$

Seminal work [17] demonstrates the block-wise Fubini-Study metric tensor can be evaluated in terms of quantum expectation values of Hermitian observables which is thus experimentally realizable.

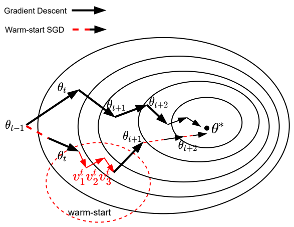


Fig. 1. A demonstration of gradient trajectory of gradient descent and warm-start QNG. The circle in red indicates the parameter re-initialization for the next step of parameter update.

B. Barren Plateau Problem

The gradient ($\nabla C(\theta)$) or stochastic gradient (g) plays an essential role in the parameter optimization process via the gradient-based method. Here, we consider the following generic definition of a barren plateau without loss of generality.

Definition 1. *(Barren Plateau). Consider the cost function $\mathcal{C}(\theta)$ defined in*

$$\min_{\theta} \mathcal{C}(\theta).$$

This cost exhibits a barren plateau if, for all $\theta_i \in \theta$, the expectation value of partial derivative $\partial_i \mathcal{C}(\theta) = \partial \mathcal{C}(\theta)/\partial \theta_i$ respect to the cost function is zero i.e., $\mathbb{E}[\partial_i \mathcal{C}(\theta)] = 0$. The variance of the above partial derivative vanishes exponentially with the number of qubits, i.e,

$$Var_\theta[\partial_i \mathcal{C}(\theta)] \in \mathcal{O}(p^{-n}) \tag{5}$$

for some $p > 1$.

Notice, we have the following conclusion by using Chebyshev's inequality

$$P(|\partial_i \mathcal{C}(\theta)| \geq c) \leq \frac{\mathrm{Var}_\theta[\partial_i \mathcal{C}(\theta)]}{c^2} \tag{6}$$

for some constant c. The definition and above inequality tell that the probability of finding a $\partial_i \mathcal{C}(\theta)$ that is larger than c decreases exponentially when the variance of the partial derivative establishes an exponential decay. The presence of BPs exists in both deep unstructured PQC with randomly initialized parameters [4] and QNNs [7]. Ref. [7] theoretically analysis the BP based on the fact that when ansatzes become unitary 2-designs [18], the expected number of samples required to estimate $\partial C(\theta)$ is exponential in the system size which often refers the number of qubits n. BP is fatal in gradient-based optimization because it might halt the parameter update and quickly converge to some sub-optimal solution.

IV. Main Results

In this section, we show that QNG corresponding to quantum probability space can be implemented as a classical first-order optimization known as mirror descent. Then we show the proposed LAWS algorithm and a general WS-SGD framework.

A. Quantum Information Geometry of Mirror Descent

In the seminar work [10], the Euclidean distance term $\frac{1}{2\eta}\|\theta - \theta_t\|_2^2$ in Eq.(1) has been replaced with a general distance function $D_\Phi(\cdot, \cdot)$, i.e.,

$$D_\Phi(\theta_1, \theta_2) = \Phi(\theta_1) - \Phi(\theta_2) - \langle \nabla\Phi(\theta_2), \theta_1 - \theta_2\rangle \tag{7}$$

where $\Phi(\cdot)$ is a carefully chosen continuously differentiable, strictly convex proximity function defined on some convex set. Notice, $D_\Phi(\theta_1, \theta_2) \geq 0$ with $D_\Phi(\theta_1, \theta_1) = 0$. $D_\Phi(\cdot, \cdot)$ defined above is also known as *Bregman divergence*, which is widely used in statistical inference, optimization, machine learning, and information geometry. As a result, a generalization of stochastic iterative optimization Eq.(1) has following

$$\theta_{t+1} = \arg\min_\theta \left\{\langle \theta, g_t\rangle + \frac{1}{\eta} D_\Phi(\theta, \theta_t)\right\} \tag{8}$$

The above optimization is known as mirror descent (MD) [19] with proximity function D_Φ. Note, if $\Phi(\theta) = \frac{1}{2}\|\theta\|_2^2$ convex, then $D_\Phi(\theta, \theta_t) = \frac{1}{2}\|\theta - \theta_t\|_2^2$ yields the standard gradient descent update Eq.(2). In addition, many modern machine learning optimizations such as Vanilla SGD, AdaGrad and Adam [20] fall into MD 8 point view. For example, given Mahalanobis distance $\Phi(\theta) = \theta^\top A\theta$ where $A \succ 0$ is a positive (semi)definite matrix, i.e., $A = \sqrt{\sum_{i=1}^t g_i^2}$ a sum of all gradients for $t = 1$ to t. We have AdaGrad

$$\theta_{t+1} = \arg\min_\theta \left\{\langle \theta, g_t\rangle + \frac{1}{2\eta}(\theta - \theta_t)^\top A(\theta - \theta_t)\right\} \tag{9}$$

which is equivalent to

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\sum_{i=1}^t g_i^2} + \epsilon} \odot g_t \tag{10}$$

where ϵ is a small number, typically set as 10^{-8}, $\odot$ indicates the element-wise product. Moreover, if

$$A = \sqrt{(1 - \beta_2)\sum_{i=1}^t \beta_2^{t-i} g_i^2} \tag{11}$$

and set $m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t$ as exponential moving average (EMA) of stochastic gradient g_t with $\beta_1, \beta_2 \in \mathbb{R}$ (typical values are $\beta_1 = 0.9$ $\beta_2 = 0.999$). We recover the Adam optimizer

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{(1 - \beta_2)\sum_{i=1}^t \beta_2^{t-i} g_i^2} + \epsilon} \odot m_t \tag{12}$$

In VQA, consider a parametric family of strictly positive probability distributions $p_\theta(x)$ parametrized by $\theta \in \mathbb{R}^d$ where $x \in [N]$ is a set of probability distributions on N elements $[N] = \{1, \cdots, N\}$ and satisfies the normalization condition

$$\int p_\theta(x)dx = 1 \text{ for all } \theta \tag{13}$$

Assuming sufficient regularity, the derivatives of such densities satisfy the identity

$$\forall t > 0 \quad \int \frac{\partial^t p_\theta(x)}{\partial\theta^t}dx = \frac{\partial^t}{\partial\theta^t}\int p_\theta(x)dx = \frac{\partial^t 1}{\partial\theta^t} = 0 \tag{14}$$

To elucidate the geometry of the probability space P, we measure the density p_θ changes when one adds a small quantity $d\theta$ to its parameter. It can be achieved in a statistically meaningful way by using the Kullback-Leibler (KL) divergence [21]. Interestingly, KL-divergence is also an instance of Bregman divergence mentioned in Eq.(7) by letting proximity function $\Phi(\theta) = \sum_i \theta_i \log(\theta_i)$ result in

$$D_\Phi(\theta, \theta + d\theta) = KL(\theta\|\theta + d\theta) = \mathbb{E}_{p_\theta}\left[\log\left(\frac{p_\theta(x)}{p_{\theta+d\theta}(x)}\right)\right] \tag{15}$$

where $\mathbb{E}_{p_\theta}$ denotes the expectation with respect to the distribution p_θ. Further, we can approximate the divergence with a second-order Taylor expansion such as

$$\begin{aligned} &KL(\theta\|\theta + d\theta) = \mathbb{E}_{p_\theta}\left[\log(p_\theta(x)) - \log(p_{\theta+d\theta}(x))\right] \\ &\approx -d\theta^\top \mathbb{E}_{p_\theta}\left[\frac{\partial \log(p_\theta(x))}{\partial\theta}\right] + \frac{1}{2}d\theta^\top \mathbb{E}_{p_\theta}\left[\frac{\partial^2 \log(p_\theta(x))}{\partial\theta^2}\right]d\theta \end{aligned} \tag{16}$$

Applying the fact that first-order term is 0 shown in Eq.(14), we have

$$D_\Phi(\theta, \theta + d\theta) = KL(\theta\|\theta + d\theta) \approx \frac{1}{2}d\theta^\top F(\theta)d\theta \tag{17}$$

$F(\theta)$ is defined by the Fisher information matrix (FIM)

$$F(\theta) = \mathbb{E}_{p_\theta}\left[\left(\frac{\partial \log(p_\theta(x))}{\partial\theta}\right)\left(\frac{\partial \log(p_\theta(x))}{\partial\theta}\right)\right] \tag{18}$$

We notice the second equality of $F(\theta)$ is often preferred because it makes clear that the $F(\theta)$ is symmetric and always positive semidefinite, though not necessarily positive definite. Finally, we plug the Bregman divergence defined on information entropy $\Phi(\theta) = \sum_i \theta_i \log(\theta_i)$ in MD optimization Eq.(8), and we have

$$\theta_{t+1} = \arg\min_\theta \left\{\langle \theta, g_t\rangle + \frac{1}{2\eta}(\theta - \theta_t)^\top F(\theta - \theta_t)\right\} \tag{19}$$

The iterative solution of the above optimization problem Eq.(19) is

$$\theta_{t+1} = \theta_t - \eta F^{-1} g_t \tag{20}$$

where F^{-1} is the pseudo-inverse of the Fisher information matrix, which recovers the natural gradient descent in Eq.(3). In the over-parameterized classical deep learning model, F is singular. To make it invertible, one often adds a non-negative damping term δ such that $\theta_{t+1} = \theta_t - \eta(F + \delta I)^{-1}g_t$.

B. Look Around Warm-start Natural Gradient

The presence of BP becomes one of the major bottlenecks in optimizing VQA, such as deep QNN. Notably, this does not preclude VQA, allowing for efficient gradient-based optimization. In section II, we discuss two mainstream techniques to mitigate BP. This work focuses on the optimization solution combined with the QNN parameter initialization strategy.

1) Motivation: Intuitively, a good parameter initialization (i.e., the distribution of initialized parameter close to optimal) requires a large number of empirical studies, hyper-parameter tuning, and possibly human intervention, which is unproduc-

Algorithm 1: Look Around and Warm-Start Natural Gradient Descent

```
1  Input: Objective function $\mathcal{C}(\theta)$, learning data $\mathcal{D}$
2  Initialization: $\theta_0$, learning rate $\eta_0$, warm-start learning
     rate $\mu_0$, warm-start iteration $K$ ($K = 5, 3,$ or 2)
3  for $t = 1, \cdots, T$ do
4      $v_0^t = \theta_{t-1}$
5      for $k = 1, \cdots, K$ do
6          Draw sample from batch data $\xi \sim \mathcal{D}_b$
7          $v_k^t = v_{k-1}^t - \mu_k \nabla C(\theta; \xi)$
8      end
9      $\theta^t_{\text{warm-start}} = v_k^t$
10     Compute natural gradient $F_t = \text{FisherIM}(\theta^t_{\text{warm-start}})$
11     Compute new gradient $g_t = \theta^t_{\text{warm-start}} - \theta_{t-1}$
12     $\theta_t = \theta^t_{\text{warm-start}} - \frac{\eta_t}{K} F_t^{-1} g_t$
13 end
```

tive. Therefore, a natural question is raised: *can we perform efficient and effective parameter initialization for QNNs?* This is the primary motivation behind our approach. Second, the one-shot model initialization strategy initializes the model only at the beginning of the training process. However, as the training process proceeds, BP appears again when we use the gradient-based method to train the model. Besides finding a suitable initialization strategy, we also consider the algorithm's efficiency since computing quantum Fisher information in QNG is expensive, as discussed in section III-A.

2) Proposed Method: We present our proposed algorithm shown in Algorithm 1. The key step in the proposed algorithm, in short, is that we perform the initialization after every parameter update instead of only initializing the PQC one at a time. The intuition behind this algorithm is that we try to warm-start the natural gradient descent for each iteration. Every time the optimizer finds a sub-optimal solution, say θ_t, we utilize this θ_t and re-initialize the model around θ_t within a small region. Later, we generalized the LAWS to accommodate all existing gradient-based methods in Algorithm 2.

There are two major advantages when using LAWS. First, LAWS could mitigate the BP issue by repeatedly performing the parameter re-initialization, where our empirical results also support this observation. Second, LAWS adopts a fast convergency speed, and it is more computationally efficient than the QNG. Third, we empirically find that LAWS achieves better generalization ability in the classification learning task.

3) Implementation Details: The implementation of LAWS is simple and is compatible with all existing gradient-based optimization frameworks. Therefore, how to effectively and efficiently perform warm-start (parameter re-initialization) is the key challenge in LAWS's design. To this end, the design of warm-start is based on a stochastic procedure, where the re-initialized parameter is sampled from a set of stochastic gradients. Fig 1 demonstrates the optimization trajectory of LAWS compared to the original QNG. We search gradients for fewer steps around the current optimal θ_t and then perform a natural gradient descent step (F_t^{-1}) on the accumulation of the previous gradient ($v_k^t - \theta_t$) at the re-initialized parameter point v_k^t.

We present two different warm-start strategies. The first one uses a K-step (K usually small, such as $K = 5$) inner loop (as the Algorithm 1 shows) to compute a set of K consecutive gradients such as $\mathcal{G}_K^t = \{g_1^t, g_2^t, \cdots, g_K^t\}$. Then, we compute a weighted average of gradients in $\mathcal{G}_K^t$ as a warm-start point of the next iteration

$$\theta_{\text{warm-start}} = \theta_{t-1} + \frac{1}{K}\sum_{k=1}^{K} g_k \tag{21}$$

for all $g_k^t \in \mathcal{G}_K^t$ such that

$$\mathcal{G}_K^t = \left\{\nabla\mathcal{C}(v_k^t, \xi) | \xi \sim \mathcal{D}_b\right\} \tag{22}$$

where each v_k^t is computed as line 9 in Algorithm 1. The second one also uses a K-step inner loop to sample gradient candidates. But one significant difference compared to the first method is that sample gradient candidates are computed with respect to the same model parameter at the current step $t-1$, say θ_{t-1}. Mathematically, we have

$$\theta_{\text{warm-start}} = \theta_{t-1} + \frac{1}{K}\sum_{k=1}^{K} \nabla\mathcal{C}(\theta_{t-1}, \xi) \text{ where } \xi \sim \mathcal{D}_b \tag{23}$$

We empirically evaluate the above-mentioned warm-start strategies. More detailed results and analysis are shown in section V.

We notice that the proposed LAWS belongs to a certain first-order optimization in modern classical learning regime so-called *Lookahead optimizer*. Based on their extraordinary work, we propose a general warm-start framework for VQA in the next section.

4) Generalized Warm-start Algorithm: In the classical machine learning study, [22] proposed a new optimization algorithm named Lookahead. Lookahead is orthogonal to the aforementioned approaches [20] due to the different parameter update settings. The core idea of Lookahead is to maintain two kinds of model parameters, i.e., "fast parameter" v_k^t and "slow parameter" θ_t, and jointly update them. Specifically, the inner loop takes the slow weights (θ_{t-1}) as initial point and updates the fast weights (v_k^t) K times to receive v_K^t; while the outer loop updates the slow weights as

$$\theta_t = (1-\alpha)\theta_{t-1} + \alpha v_K^t,\ \alpha \in (0, 1) \tag{24}$$

Any standard optimizer, e.g., Vallina SGD, AdaGrad, and Adam, can serve as the inner-loop optimizer. In our speech, the inner-loop act as a warm-start initialization. In this way, the Lookahead optimizer achieves remarkable performance improvement over the standard optimizer. Further, due to its simplicity in implementation, negligible computation and

Algorithm 2: Generalized Warm-start Stochastic Gradient Descent (WS-SGD)

```
Input: Objective function $\mathcal{C}(\theta)$, learning data $\mathcal{D}$,
  warm-start optimizer $\mathcal{W}$, reparameterization
  coefficient function $\Delta_t$
Initialization: $\theta_0$, warm-start learning rate $\mu_0$,
  warm-start iteration $K$
for $t = 1, \cdots, T$ do
    $v_0^t = \theta_{t-1}$
    for $k = 1, \cdots, K$ do
        $v_k^t = \mathcal{W}\big(\mathcal{C}(\theta), v_0^t, \mu_0, \mathcal{D}_m\big)$
    end
    $\theta^t_{\text{warm-start}} = v_k^t$
    $\theta_{t+1} = \Delta_t \theta_t + (1 - \Delta_t)\theta^t_{\text{warm-start}}$
end
```

memory cost, and compatibility with almost current ML libraries, Lookahead has been widely adopted.

Interestingly, we find LAWS also falls into this line of research. In algorithm 1, let $\lambda_t = \eta_t / K$, we compute the θ_t as

$$\begin{aligned}\theta_t &= \theta^t_{\text{warm-start}} - \lambda_t F_t^{-1} g_t \\ &= \lambda_t F_t^{-1} \theta_{t-1} + (1 - \lambda_t F_t^{-1})\theta^t_{\text{warm-start}}.\end{aligned} \tag{25}$$

The last equality is due to $v_k^t = \theta^t_{\text{warm-start}}$. From the above derivation, we see that the mathematical difference between LAWS and Lookahead is: we replace α in Lookahead Eq.(24) to some value such as

$$\alpha = 1 - \lambda_t F_t^{-1}$$

where is not a fixed real coefficient but a Fisher information related quantity.

To this end, we propose our unified framework WS-SGD for a warm-start stochastic gradient descent algorithm for QNNs as shown in Algorithm 2. We first employ a generalized optimizer $\mathcal{W}$ for the warm-start inner loop. The choice of such an optimizer heavily affects re-initialization and model performance. As reported in [22], WS-SGD may benefit from a larger learning rate in the inner loop. In other words, we could use a larger step size μ. We also propose the general form for reparameterization coefficient Δ_t as a function of gradient, for example, $\Delta_t = 1 - \alpha$ (Lookahead), $\Delta_t = 1 - \lambda_t F_t$ (WS-SGD), and $\Delta_t = 1 - \lambda_t \sqrt{\sum_k \nabla C(v_k^t, \xi)^2}$ (Adam-like SGD). Our empirical results are present in section V, we conclude that WS-SGD achieves faster convergence rates (i.e., smaller optimization error), and enjoys smaller generalization errors.

C. Convergence Analysis

In this section, we present the convergence and generalization analysis of the proposed algorithm. We first provide some useful definitions and assumptions which have been widely adopted in classical machine learning. We provide the analysis of the convergence for both convex and non-convex objective functions $\mathcal{C}(\theta)$. We start by showing the proof of convergence on the convex problem to give some intuition first and then give the proof on a more realistic non-convex problem.

1) Assumption: The assumptions we are making are

Assumption 1. *(Bounded gradient). The function $\mathcal{C}(\theta)$ has bounded (stochastic) gradients, i.e., for any $\theta \in R^d$ we have*

$$||\nabla \mathcal{C}(\theta, \xi)||_2 \leq G \text{ for all } \xi \sim \mathcal{D} \tag{26}$$

Assumption 2. *(L-Lipschitz smooth). The function $\mathcal{C}(\theta)$ is L-Lipschitz smooth i.e.*

$$||\nabla \mathcal{C}(\theta) - \nabla \mathcal{C}(\mu)||_2 \leq L||\theta - \mu|| \text{ for all } \theta, \mu \in R^d \tag{27}$$

Assumption 3. *(M-Lipschitz continuous). The function $\mathcal{C}(\theta)$ is L-Lipschitz continous i.e.*

$$||\mathcal{C}(\theta) - \mathcal{C}(\mu)||_2 \leq M||\theta - \mu|| \text{ for all } \theta, \mu \in R^d \tag{28}$$

We also define Polyak-Lojasiewicz (PL) condition as

Definition 2. *(PL Condition) Let $\theta^* \in \arg\min_\theta \mathcal{C}(\theta)$. We say a function $\mathcal{C}(\theta)$ satisfies σ-PL condition if*

$$\sigma(\mathcal{C}(\theta) - \mathcal{C}(\theta^*)) \leq \|\nabla \mathcal{C}(\theta)\|^2 \tag{29}$$

with some constant σ.

The above assumptions and definitions can be easily obtained and verified in VQE. Now, we show our theoretical results below in the Theorem 1 and Theorem 2.

2) Convex Objective function: Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where (x_i, y_i) is is drawn from an unknown distribution, one often minimizes the empirical risk $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^n \mathcal{C}(\theta, x_i, y_i)$ via a randomized algorithm, e.g. SGD, to find an estimated optimum $\theta_T \in \arg\min_\theta L(\theta)$. However, this empirical solution $\hat{\theta}$, differs from the desired optimum θ^* of the population risk

$$\theta^* \in \arg\min_\theta L(\theta, \mathcal{D}) = \mathbb{E}_{x,y\sim\mathcal{D}}[\mathcal{C}(\theta, x)]. \tag{30}$$

To begin with, we first investigate the convergence performance of WS-SGD when its warm-start optimizer $\mathcal{W}$ is SGD. We summarize our main results in Theorem 1 below.

Theorem 1. *(Convex) Suppose the objective function $\mathcal{C}(\theta)$ is gamma-strongly convex, M-Lipschitz continuous, and L-Lipschitz smooth w.r.t., θ. Let $\theta^* = \arg\min_\theta \mathcal{C}(\theta)$. Let warstart learning rate $\mu_k^t = \frac{c_0}{((t-1)k+K+2)}, c_0 \in (0, 1]$, the optimization error of the output θ_T of WS-SGD satisfies*

$$\begin{aligned}\mathbb{E}[\mathcal{C}(\theta_T) - \mathcal{C}\theta^*)] \leq{}& \frac{e^{2\Delta} L(k+2)^{2\Delta}}{2((T+1)K+2)^{2\Delta}}\|\theta_0 - \theta^*\|^2 \\ &+ \frac{16LG^2}{c_0^2((T+1)K+2)^{2(1-\Delta)}(2\Delta - 1)}\end{aligned} \tag{31}$$

3) Non-Convex Objective function: To prove the non-convex objective function, we use the Polyak-Lojasiewicz condition defined in 2.

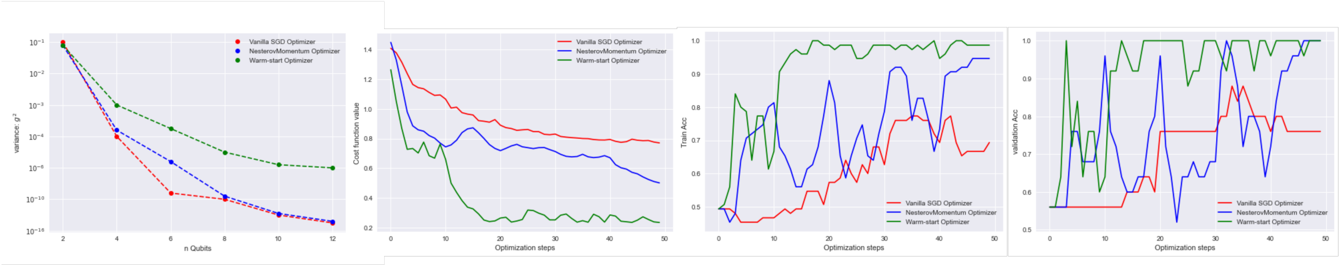


Fig. 2. Variational classifier for Iris classification task: SGD vs. Nesterov vs. LAWS

Theorem 2. *(Non-Convex) Suppose the objective function* $\mathcal{C}(\theta)$ *is M-Lipschitz continuous, and L-Lipschitz smooth w.r.t.,* θ*. In addition, suppose* $\mathcal{C}(\theta)$ *satisfies* σ*-PL condition. Let* $\mu_k^t = \frac{1}{tK+t+1}$*, we have*

$$\begin{aligned}\mathbb{E}[\mathcal{C}(\theta_T) - \mathcal{C}(\theta^*)] \leq & \frac{4}{(TK+1)^{2\Delta}}\mathbb{E}\big[\mathcal{C}(\theta_0) - \mathcal{C}(\theta^*)\big] \\ & + \frac{2\Delta M G^2 C_0}{(TK+1)^{2\Delta-1}}\end{aligned} \tag{32}$$

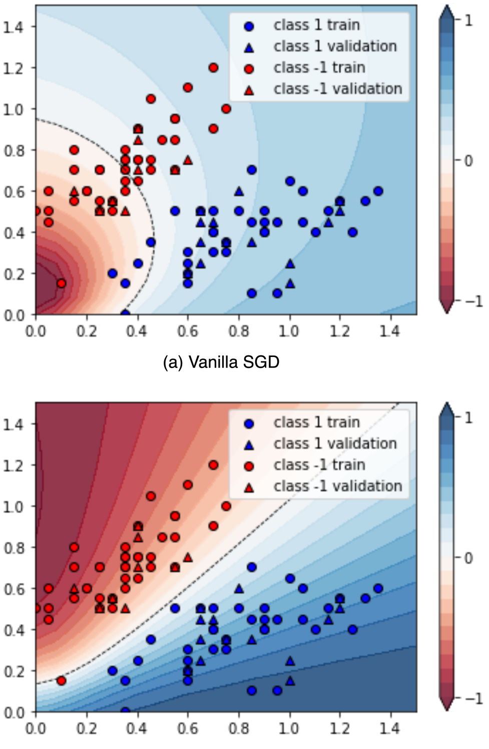
where $C_0 = \Delta + (1-\Delta)(K-1)$.

We again omit the complete proof of this theorem. The proof sketch is we first bound $\mathbb{E}\big[\|\mathcal{C}(v_K^t) - \mathcal{C}(\theta^*)\|\big]$, then we use the relation of v_K^t and θ_t defined in the Algorithm 1 line 10 to derive the final bound of $\mathbb{E}[\mathcal{C}(\theta_T) - \mathcal{C}(\theta^*)]$. In the next section, we present the numerical simulation results of LAWS, WS-SGD, and their variants.

V. Numerical Simulations

To evaluate the performance of LAWS, WS-SGD, and their variants, we use the open-source library PennyLane [23] 0.22.2 built on Python 3.7. Most of the experiments follow the open-source tutorials from the official PennyLane website. We conduct experiments on variational quantum classifiers such that the quantum circuits can be trained from labeled data to classify new data samples. The classification training data is public and can be downloaded from the PennyLane tutorial. All the experimental results and source code implementation can be found at https://github.com/taozeyi1990/LAWS.

We perform the binary Iris classification task, which is a simple but powerful QNN to show that the warm-start strategy could mitigate BP issue and has better generalization ability. The learning rate for SGD and Nesterov momentum optimizer is set to be 0.01. While the learning rate, look-around rate and look-around steps are 0.01, 0.5, and 5, respectively. We train QNN model within 50 iterations. Figure 2 show the (1) gradient variance when increasing n-qubit; (2) cost value of the objective function; (3) training accuracy, and (4) validation accuracy, respectively. As shown in each figure, the warm-start SGD in green demonstrates its superiority in this task. Figure 3 indicates the decision boundaries of the model trained with different optimizers. We observe that the two classes in the train and validation dataset are perfectly separated when using the warm-start optimizer, which indicates the WS-SGD has a stronger generalization ability. The result of the Nesterov optimizer seems to suffer from the under-fitted where the samples at the bottom left are mixed.



(a) Vanilla SGD

(b) Warm-start SGD

Fig. 3. Variational classifier for Iris classification task: decision boundary

VI. Conclusion

In this work, we propose a unified framework for QNG by using a classical first-order optimization scheme. The proposed new algorithm named WS-SGD shows its power in QVA learning. Our experiment results show that the proposed algorithm could mitigate the BP issue and have better generalization ability in quantum classification problems.

Acknowledgment

The authors would like to thank all the reviewers for their helpful comments. This project was supported in part by the US National Science Foundation grant CNS-1816399.

References

[1] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, no. 1, pp. 1–7, 2014.

[2] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[3] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," *arXiv preprint arXiv:1802.06002*, 2018.

[4] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, no. 1, pp. 1–6, 2018.

[5] Y. Suzuki, H. Yano, R. Raymond, and N. Yamamoto, "Normalized gradient descent for variational quantum algorithms," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2021, pp. 1–9.

[6] T. Haug and M. Kim, "Optimal training of variational quantum algorithms without barren plateaus," *arXiv preprint arXiv:2104.14543*, 2021.

[7] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, "An initialization strategy for addressing barren plateaus in parametrized quantum circuits," *Quantum*, vol. 3, p. 214, 2019.

[8] H.-Y. Liu, T.-P. Sun, Y.-C. Wu, Y.-J. Han, and G.-P. Guo, "A parameter initialization method for variational quantum algorithms to mitigate barren plateaus based on transfer learning," *arXiv preprint arXiv:2112.10952*, 2021.

[9] L. Franken, B. Georgiev, S. Muecke, M. Wolter, N. Piatkowski, and C. Bauckhage, "Gradient-free quantum optimization on nisq devices," *arXiv preprint arXiv:2012.13453*, 2020.

[10] A. S. Nemirovskii, D. B. Yudin, D. B. Iudin, and D. B. Iudin, *Problem complexity and method efficiency in optimization*. Wiley, 1983.

[11] D. Wierichs, C. Gogolin, and M. Kastoryano, "Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer," *Physical Review Research*, vol. 2, no. 4, p. 043246, 2020.

[12] M. Cerezo and P. J. Coles, "Impact of barren plateaus on the hessian and higher order derivatives," *arXiv e-prints*, pp. arXiv–2008, 2020.

[13] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, "Effect of barren plateaus on gradient-free optimization," *Quantum*, vol. 5, p. 558, 2021.

[14] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nature communications*, vol. 12, no. 1, pp. 1–12, 2021.

[15] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, "Connecting ansatz expressibility to gradient magnitudes and barren plateaus," *PRX Quantum*, vol. 3, no. 1, p. 010313, 2022.

[16] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural computation*, vol. 10, no. 2, pp. 251–276, 1998.

[17] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, "Quantum natural gradient," *Quantum*, vol. 4, p. 269, 2020.

[18] A. W. Harrow and R. A. Low, "Random quantum circuits are approximate 2-designs," *Communications in Mathematical Physics*, vol. 291, no. 1, pp. 257–302, 2009.

[19] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[22] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[23] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.