The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

Toward Physically Realizable Quantum Neural Networks

Mohsen Heidari, Ananth Grama, Wojciech Szpankowski

Department of Computer Science, Purdue University, West Lafayette, IN, USA
{mheidari, ayg, szpan}@purdue.edu

Abstract

There has been significant recent interest in quantum neural networks (QNNs), along with their applications in diverse domains. Current solutions for QNNs pose significant challenges concerning their scalability, ensuring that the postulates of quantum mechanics are satisfied and that the networks are physically realizable. The exponential state space of QNNs poses challenges for the scalability of training procedures. The no-cloning principle prohibits making multiple copies of training samples, and the measurement postulates lead to non-deterministic loss functions. Consequently, the physical realizability and efficiency of existing approaches that rely on repeated measurement of several copies of each sample for training QNNs are unclear. This paper presents a new model for QNNs that relies on band-limited Fourier expansions of transfer functions of quantum perceptrons (QPs) to design scalable training procedures. This training procedure is augmented with a randomized quantum stochastic gradient descent technique that eliminates the need for sample replication. We show that this training procedure converges to the true minima in expectation, even in the presence of non-determinism due to quantum measurement. Our solution has a number of important benefits: (i) using QPs with concentrated Fourier power spectrum, we show that the training procedure for QNNs can be made scalable; (ii) it eliminates the need for resampling, thus staying consistent with the no-cloning rule; and (iii) enhanced data efficiency for the overall training process since each data sample is processed once per epoch. We present a detailed theoretical foundation for our models and methods' scalability, accuracy, and data efficiency. We also validate the utility of our approach through a series of numerical experiments.

Introduction

With rapid advances in quantum computing, there has been increasing interest in quantum machine learning models and circuits. Work at the intersection of deep learning and quantum computing has resulted in the development of quantum analogs of classical neural networks, broadly known as quantum neural networks (QNNs) (Mitarai et al. 2018; Farhi and Neven 2018; Schuld et al. 2020; Beer et al. 2020). These networks are comprised of quantum perceptrons (QPs) (Lewenstein 1994), and the corresponding circuits (either layers of QPs or variational circuits) are physically realized through technologies such as trapped ions, quantum dots, and molecular magnets. QNNs have been applied to both classical and quantum data in the context of diverse applications. They have also inspired the development of novel classical neural network architectures (Garg and Ramakrishnan 2020), which have demonstrated excellent performance in applications such as image understanding and natural language processing.

Our effort focuses on developing QNNs for quantum data within the constraints of physically realizable quantum models (e.g., no-cloning, measurement state collapse). Among many applications, quantum state classification is of significance (Chen et al. 2020) and has been studied under various specifications such as *separability* of quantum states (Gao et al. 2018; Ma and Yung 2018), integrated quantum photonics (Kudyshev et al. 2020), and dark matter detection through classification of polarization state of photons (Dixit et al. 2021). In yet other applications, it has been shown that data traditionally viewed in classical settings are better modeled in a quantum framework. In an intriguing set of results, Orus et al. demonstrate how data and processes from financial systems are best modeled in quantum frameworks (Orús, Mugel, and Lizaso 2019).

The problem of constructing accurate QNNs for the direct classification of quantum data is a complex one for a number of reasons: (i) the state of a QNN is exponential in the number of QPs. For this reason, even relatively small networks are hard to train in terms of their computation and memory requirements; (ii) the no-cloning postulate of quantum states implies that the state of an unknown qubit cannot be replicated into another qubit; (iii) measurement of the state of a qubit corresponds to a realization of a random process, and results in state collapse; (iv) the non-deterministic nature of measurements implies that computation of classical loss functions itself is non-deterministic; and (v) existing solutions that rely on replicated data and oversampling for probabilistic bounds on convergence suffer greatly from lack of data efficiency. While there have been past efforts at modeling and training QNNs, one or more of these fundamental considerations are often overlooked. We present a comprehensive solution to designing quantum circuits for QNNs, along with provably efficient training techniques that satisfy the aforementioned physical constraints.

Main Contributions

In this work, we propose a new QNN circuit, its analyses, and a training procedure to address key shortcomings of prior QNNs.

We make the following specific contributions: (i) we present a novel model for a QP whose Fourier spectrum is concentrated in a fixed sided band; (ii) we demonstrate how our QPs can be used to construct QNNs whose state and state updates can be performed in time linear in the number of QPs; (iii) we present a new quantum circuit that integrates gradient updates into the network, obviating the need for repeated quantum measurements and associated state collapse; (iv) we demonstrate data efficiency of our quantum circuit in that it does not need data replication to deal with stochastic loss functions associated with measurements; and (v) we show that our quantum circuit converges to the exact gradient in expectation, even in the presence of the stochastic loss function without data replication. We present detailed theoretical results, along with simulations demonstrating our model's scalability, accuracy, and data efficiency.

Related Results

Quantum neural networks have received significant recent research attention since the early work of Toth et al. (Toth et al. 1996), who proposed an architecture comprised of quantum dots connected in a cellular structure. Analogous to conventional neural networks, Lewenstein (Lewenstein 1994) proposed an early model for a quantum perceptron (QP) along with its corresponding unitary transformation. Networks of these QPs were used to construct early QNNs, which were used to classify suitably preprocessed classical data inputs. Since then, there has been significant research interest in the development of QNNs over the past two decades (Schuld, Sinayskiy, and Petruccione 2014; Mitarai et al. 2018; Farhi and Neven 2018; Torrontegui and Garcia-Ripoll 2018; Schuld et al. 2020; Beer et al. 2020). Massoli et al. (Massoli et al. 2021) provide an excellent survey and classification of various models and technologies for QNNs.

QNNs are generally trained using Variational Quantum Algorithms (Cerezo et al. 2020; Guerreschi and Smelyanskiy 2017), which use quantum analogs of QNN parameters, initial qubits state (input data), and trail state (output of the QNN). These are used in conjunction with optimization procedures such as gradient descent to update network parameters. Among the best known of these procedures is the Quantum Approximate Optimization Algorithm of Farhi et al. (Farhi, Goldstone, and Gutmann 2014), which prescribes a circuit and an approximation bound for efficient combinatorial optimization. This method was subsequently generalized to continuous non-convex problems for finding the ground state of a circuit through a suitably defined evolution operator. A slew of recent results have used optimizers such as L-BFGS, Adam, Nestorov, and SGD to train different QNN models. An excellent survey of these methods and associated tradeoffs is presented by Massoli et al. (Massoli et al. 2021). The basic problem of exponential network state and associated cost of optimization procedures is not directly addressed in these results. In contrast, our focus in this work is on reducing the cost of training QNNs through flexible and powerful constraints on Fourier expansions of QPs.

A second issue relating to the convergence of these prior QNNs arises from the stochastic nature of quantum measurements – i.e., measuring the quantum state (in particular, the trial state) to compute a loss function is inherently stochastic. For this reason, the convergence of prior QNNs relies on repeated processing of the same data sample until the ground state is achieved with prescribed high probability. There are two challenges for these prior solutions: (i) repeated reads of the same state are problematic because of the no-cloning rule for quantum states; and (ii) the cost of the training procedure is significantly higher due to more extensive data volumes. In contrast, our method does not require data resampling and therefore does not have the drawbacks above.

Our work formally demonstrates how we can compute gradients in effective and efficient procedures (Theorem 1) and achieve asymptotically faster convergence in terms of data processing (Theorem 2). These two results provide the formal basis for our models and methods compared to prior results.

Model

Quantum Machine Learning Model

The focus of this paper is on the classification of quantum states. Following the framework in (Heidari, Padakandla, and Szpankowski 2021), our quantum machine learning model inputs a set of n quantum states corresponding to quantum training data samples, each paired with a classical label $y \in \mathcal{Y}$. We write these training samples as $\{(\rho_i, y_i)\}_{i=1}^n$. Each ρ_i is a density operator acting on the Hilbert space H of d-qubits, i.e., $\dim(H) = 2^d$. The data samples with the labels are drawn independently from a fixed but unknown probability distribution D. A quantum learning algorithm takes training samples as input to construct a predictor for labels of unseen samples. The prediction is in the form of a *quantum measurement* that acts on a quantum state and outputs $\hat{y} \in \mathcal{Y}$ as the predicted label.

To test a predictor $\mathcal{M} := \{M_y : y \in \mathcal{Y}\}$, a new sample (ρ_{test}, y_{test}) is drawn randomly according to D. Without revealing y_{test}, we measure ρ_{test} with $\mathcal{M}$. In view of the the postulate of quantum measurements, the outcome of this predictor is a random variable $\hat{Y}$ that together with the true label Y form the joint probability distribution given by

$$\mathbb{P}\{Y = y_{test}, \hat{Y} = \hat{y}\} = D_Y(y_{test}) \operatorname{tr}\{M_{\hat{y}} \rho_{test}\},$$

where D_Y is the marginal of D. Thus, one can use an existing loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ (such as the 0-1 loss or square loss) to measure the accuracy of the predictors.

Quantum Neural Networks

Figure 1 shows a generic model for QNNs considered in our work. The QNN consists of a *feed-forward* network of interconnected QPs, each of which is a parametric unitary operator. At the output of the last layer is a quantum measurement

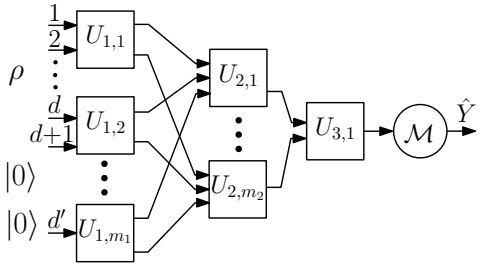

Figure 1: A generic feed-forward QNN with one hidden layer and parametric unitary operators as QPs.

acting on the readout qubits to produce a classical output as the label's prediction.

The input to the QNN (or the first layer) is a state of d' qubits consisting of the original d-qubit sample ρ padded with additional auxiliary qubits, say $|0\rangle$. The padding allows the QNN to implement a more general class of operations (quantum channels) on the samples [1]. The input to the l^{th} layer is the output of the previous layer $l-1$.

Band Limited QPs

We use an abstraction of a QP that generalizes the model of Farhi et al. (Farhi and Neven 2018). We consider QPs of the form $U = e^{iA}$, where A is a Hermitian operator acting on a small subsystem of the d'-qubit system; hence it is band-limited. This notion stems from quantum Fourier expansion using Pauli operators.

Quantum Fourier Expansion: Tensor products of Pauli operators together with identity have been used to develop a quantum Fourier expansion (Montanaro and Osborne 2010) analogous to the Fourier expansion on the Boolean cube (De Wolf 2008). We denote these operators by $\{\sigma^0, \sigma^1, \sigma^2, \sigma^3\}$. Furthermore, as a shorthand, given any vector $\mathbf{s} \in \{0,1,2,3\}^d$, we denote by $\sigma^{\mathbf{s}}$ the tensor product $\sigma^{s_1} \otimes \sigma^{s_2} \otimes \cdots \otimes \sigma^{s_d}$. With this notation we present the notion of quantum Fourier expansion:

Remark 1. *Any bounded operator A on the Hilbert space of d qubits can be uniquely written as: $A = \sum_{\mathbf{s}\in\{0,1,2,3\}^d} a_{\mathbf{s}}\ \sigma^{\mathbf{s}}$, where $a_{\mathbf{s}} \in \mathbb{C}$ are the Fourier coefficients of A and are given by $a_{\mathbf{s}} = \frac{1}{2^d}\operatorname{tr}\{A\sigma^{\mathbf{s}}\}$. If A is Hermitian then $a_{\mathbf{s}} \in \mathbb{R}$.*

Consider a QP acting on the subsystem corresponding to a subset of coordinates $\mathcal{J} \subseteq [d']$. Since A acts on qubits, then from Remark 1, it decomposes in terms of Pauli operators as

$$A = \sum_{\mathbf{s}: s_j = 0, \forall j \notin \mathcal{J}} a_{\mathbf{s}}\ \sigma^{\mathbf{s}}.$$

Note that the Fourier coefficients are zero outside of the coordinate subset $\mathcal{J}$, leading to a band-limited power spectrum in the Fourier domain. Moreover, these Fourier coefficients are used to fine-tune the QP. Intuitively, these coefficients can be viewed as quantum analogs of weights in classical NNs.

To control the power spectrum of QPs, we bound $|\mathcal{J}| \leq k$, where k is a bandwidth parameter. This constraint ensures that $a_{\mathbf{s}} = 0$ for any $\mathbf{s}$ with more than k non-zero components. In addition, it provides the basis for controlling the cost of gradient computation by limiting the number of coefficients to optimize over. In this paper, we typically set $k = 2$.

Let $\overrightarrow{a}_{l,j}$ denote the vector of Fourier coefficients for the jth QP in the l^{th} layer and $U_{l,j}(\overrightarrow{a}_{l,j})$ denote the unitary operator of this QP. With this setup, each layer consists of several QPs acting on different subsystems of $\dim \leq 2^k$. By $U_l(\overrightarrow{a}_l)$, denote the overall unitary operator of the l^{th} layer, which is given by $U_l(\overrightarrow{a}_l) = \prod_{j=1}^{m_l} U_{l,j}(\overrightarrow{a}_{l,j})$, where m_l is the number of QPs in this layer. Building on this formulation, a QNN with L layers is itself a unitary operator that factors as:

$$U_{QNN}(\overrightarrow{a}) = U_L(\overrightarrow{a}_L)U_{L-1}(\overrightarrow{a}_{L-1})\cdots U_1(\overrightarrow{a}_1),$$

where $\overrightarrow{a}$ is the vector of all the parameters of all the layers. Since, each layer is characterized by $m_l 4^k$ coefficients, the QNN needs at most $c_{QNN} = 4^k m$ coefficients to be optimized over, where m is the total number of QPs.

Computing the Loss

To the output of the final layer, one applies a quantum measurement $\mathcal{M} = \{M_y : y \in \mathcal{Y}\}$ to produce the prediction for the classical label. For a padded sample $\rho' = \rho \otimes |0\rangle\langle 0|$, measuring the output qubit of the QNN results in an outcome $\hat{y}$ with the following probability:

$$\mathbb{P}(\hat{Y} = \hat{y}|\,\overrightarrow{a}, \rho) = \operatorname{tr}\Big\{M_{\hat{y}} U_{QNN}(\overrightarrow{a})\rho' U^{\dagger}_{QNN}(\overrightarrow{a})\Big\}.$$

Let $\ell : \mathcal{Y}\times\mathcal{Y} \mapsto \mathbb{R}$ be the loss function. Then, conditioned on a fixed sample (ρ, y), the sample's expected loss taken w.r.t. $\hat{Y}$ is given by

$$\begin{aligned} L(\overrightarrow{a}, \rho, y) &= \mathbb{E}_{\hat{Y}}[\ell(\hat{Y}, y)] \\ &= \sum_{\hat{y}} \ell(y, \hat{y}) \operatorname{tr}\Big\{M_{\hat{y}} U_{QNN}(\overrightarrow{a})\rho' U^{\dagger}_{QNN}(\overrightarrow{a})\Big\}. \quad (1) \end{aligned}$$

Taking the expectation over the sample's distribution D gives the generalization loss:

$$L(\overrightarrow{a}) = \mathbb{E}_D[L(\overrightarrow{a}, \rho, Y)].$$

While it is desirable to minimize the expected loss, this goal is not feasible, since the underlying distribution D is unknown. Alternatively, given a sample set $\mathcal{S}_n = \{(\rho_j, y_j)\}_{j=1}^n$, one aims to minimize the average per-sample expected loss:

$$\frac{1}{n}\sum_j L(\overrightarrow{a}, \rho_j, y_j). \quad (2)$$

However, unlike classical supervised learning, exact computation of this loss is not possible as ρ_j's are unknown in general. One can approximate this loss, but under the condition that several exact copies of each sample are available.

[1] It is well-known that any quantum channel (CPTP map) is a partial trace of a unitary operator (Stinespring dilation) in a larger Hilbert space.

Typically, this is infeasible in view of the no-cloning principle. This restriction makes training more challenging than its classical counterpart. We argue that even if several copies of the samples are available, such a strategy is not desirable due to its increased data and computational cost. In the next section, we present our optimization approach based on a randomized variant of SGD to address this problem.

Quantum Stochastic Gradient Descent

In this section, we present our approach to training QNNs using gradient descent. Ideally, if the t^{th} sample's expected loss $L(\overrightarrow{a}, \rho_t, y_t)$ was known, one would apply a standard gradient descent method to train the QNN as follows:

$$\overrightarrow{a}^{(t+1)} = \overrightarrow{a}^{(t)} - \eta_t \nabla L(\overrightarrow{a}, \rho_t, y_t).$$

Typically, $L(\overrightarrow{a}, \rho_t, y_t)$ is unknown hence the above update is not practical. One approach to deal with this issue is to use several exact copies of each sample, enabling one to approximate the gradient (Farhi and Neven 2018). The drawback of this approach is two-fold: (i) multiple exact copies of each sample are needed; and (ii) the associated utilization complexity for training is high. The latter drawback derives from the fact that approximating the derivative of the sample's expected loss with error up to ε requires the processing of $O(1/\varepsilon^2)$ exact copies. Therefore, training a QNN with the total of c_{QNN} parameters and using T iterations of SGD requires $O(T\frac{c_{QNN}}{\varepsilon^2})$ uses of the QNN.

In this paper, we propose an alternate procedure for performing quantum stochastic gradient descent (QSGD) without the need for exact copies, substantially reducing the utilization complexity to $O(Tc_{QNN})$. We note here that minimizing the loss without access to exact copies is a more complex problem since even the per-sample loss is unknown due to the randomness from quantum measurements. In particular, we introduce a randomized QSGD with the following update rule:

$$\overrightarrow{a}^{(t+1)} = \overrightarrow{a}^{(t)} - \eta_t Z_t,$$

where Z_t is a random variable representing the outcome of a gradient measurement that is designed in the preceding section of the paper. We present an analysis for a QNN and associated convex objective function. We note that non-convex objective functions associated with deep networks are typically formulated as convex quadratic approximations within a trust region or cubic regularization.

Unbiased Measurement of the Gradient

We now present a procedure for measuring the derivative of the expected loss (Figure 2). We start by training a single-layer QNN. Training multi-layer QNNs with this approach is done using *backpropagation* to find (local) minima for the loss function. The corresponding unitary operator of a single-layer network is of the form:

$$U_{QNN}(\overrightarrow{a}) = U_1(\overrightarrow{a}) = \exp\Big\{i \sum_{\mathbf{s}} a_{\mathbf{s}} \sigma^{\mathbf{s}}\Big\}. \tag{3}$$

In what follows, we intend to measure the derivative of the loss w.r.t. $a_{\mathbf{s}}$ for a given $\mathbf{s}$ appearing in the summation above.

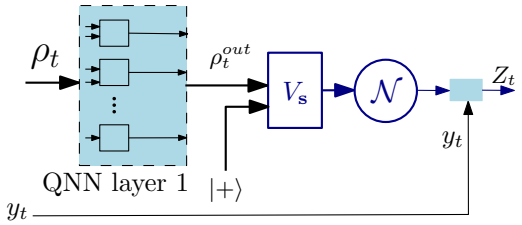

Figure 2: The procedure for measuring the derivative of the loss consisting of a unitary operator $V_{\mathbf{s}}$ followed by a quantum measurement $\mathcal{N}$ with additional classical processing.

In our method, at each time t, we apply the QNN on the t^{th} sample ρ_t, which produces ρ_t^{out}. Next, we add an auxiliary qubit to ρ_t^{out}, which creates the following state:

$$\tilde{\rho}_t = \rho_t^{out} \otimes |+\rangle\langle+|_E\,, \tag{4}$$

where $|+\rangle_E = \frac{1}{\sqrt{2}}(|0\rangle_E + |1\rangle_E)$, and E represents the auxiliary Hilbert space. Then, we apply the following unitary operator on $\tilde{\rho}_t$:

$$V_{\mathbf{s}} = e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}} \otimes |0\rangle\langle 0|_E + e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}} \otimes |1\rangle\langle 1|_E\,. \tag{5}$$

Next, we measure the state by a quantum measurement $\mathcal{N} := \{\Lambda_{b,\hat{y}} : \hat{y} \in \mathcal{Y}, b \in \{0,1\}\}$ with outcomes in $\mathcal{Y} \times \{0,1\}$ and operators:

$$\Lambda_{b,\hat{y}} = M_{\hat{y}} \otimes |b\rangle\langle b|_E\,, \qquad \forall \hat{y} \in \mathcal{Y},\ b = 0, 1. \tag{6}$$

Finally, if the outcome of this measurement is $(\hat{y}, b)$, we compute $z_t = -2(-1)^b \ell(y, \hat{y})$ as the measured gradient, where $\ell(\cdot,\cdot)$ is the loss function. Note that Z_t is a random variable depending on the current parameters of the network and the input sample (ρ_t, y_t). The following lemma, shows that Z_t is an unbiased measure of the derivative of loss.

Lemma 1. *Let Z_t be the output of the procedure shown in Figure 2 applied on the t^{th} sample. Then Z_t is an unbiased estimate of the derivative of the loss. More precisely,* $\mathbb{E}[Z_t|y_t, \rho_t] = \frac{\partial L(\overrightarrow{a}, \rho_t, y_t)}{\partial a_{\mathbf{s}}}$.

Proof. We start by taking the derivative of the sample's expected loss w.r.t. $a_{\mathbf{s}}$. Instead of the finite difference method, we compute the derivative directly. Using (3), for the t^{th} sample, we can write:

$$\frac{\partial L(\overrightarrow{a}, \rho_t, y_t)}{\partial a_{\mathbf{s}}} = \sum_{\hat{y}} i\ell(y_t, \hat{y})\, \mathrm{tr}\Big\{M_{\hat{y}}\Big(\sigma^{\mathbf{s}}$$

$$U_{QNN}(\overrightarrow{a})\rho_t U_{QNN}^t(\overrightarrow{a}) - U_{QNN}(\overrightarrow{a})\rho_t U_{QNN}^t(\overrightarrow{a})\sigma^{\mathbf{s}}\Big)\Big\}.$$

Denote $\rho_t^{out} = U_{QNN}(\overrightarrow{a})\rho_t U_{QNN}^\dagger(\overrightarrow{a})$ as the output state. Hence, the derivative is given by:

$$\frac{\partial L(\overrightarrow{a}, \rho_t, y_t)}{\partial a_{\mathbf{s}}} = \sum_{\hat{y}} i\ell(y_t, \hat{y})\, \mathrm{tr}\big\{M_{\hat{y}}\big[\sigma^{\mathbf{s}}, \rho_t^{out}\big]\big\}, \tag{7}$$

where $[\cdot,\cdot]$ is the commutator. Note that $\big[\sigma^{\mathbf{s}}, \rho_t^{out}\big]$ might not be measurable directly. However, as $\sigma^{\mathbf{s}}$ is a product of Pauli operators, as shown in (Mitarai et al. 2018), we have that:

$$\big[\sigma^{\mathbf{s}}, \rho_t^{out}\big] = i\Big(e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}} - e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\Big).$$

With this relation, the derivative of the loss equals:

$$\frac{\partial L(\overrightarrow{a},\rho_t,y_t)}{\partial a_{\mathbf{s}}} = -\sum_{\hat{y}} \ell(y_t,\hat{y}) \operatorname{tr}\Big\{M_{\hat{y}}\Big(e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}} - e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\Big)\Big\}.$$

Next, we show that the procedure in Figure 2 measures this derivative. With the definition of $V_{\mathbf{s}}$ in (5) and $\tilde{\rho}_t$ in (4), we have that

$$\begin{aligned} V_{\mathbf{s}}\tilde{\rho}_t V_{\mathbf{s}}^\dagger = \frac{1}{2}\Big(& e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\otimes|0\rangle\langle 0| \\ &+ e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\otimes|0\rangle\langle 1| \\ &+ e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\otimes|1\rangle\langle 0| \\ &+ e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\otimes|1\rangle\langle 1|\Big). \end{aligned}$$

Next, we apply the measurement $\mathcal{N}$. Then, with $(\hat{y}, b)$ as the outcome, we output $z_t = -2(-1)^b\ell(y_t,\hat{y})$. It is not difficult to verify that conditioned on a fixed $\hat{y}$, the expected outcome of $(-1)^b$ equals:

$$\begin{aligned} \operatorname{tr}\big\{(\Lambda_{0,\hat{y}} - \Lambda_{1,\hat{y}})V_{\mathbf{s}}\tilde{\rho}_t V_{\mathbf{s}}^\dagger\big\} &= \frac{1}{2}\operatorname{tr}\Big\{M_{\hat{y}}\, e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\Big\} \\ &\quad - \frac{1}{2}\operatorname{tr}\Big\{M_{\hat{y}}\, e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\Big\}. \end{aligned}$$

Therefore, the expectation of Z_t is obtained by taking the expectation over $\hat{Y}$and equals to

$$\begin{aligned} \mathbb{E}[Z_t|y_t,\rho_t] &= -2\sum_{\hat{y},b}(-1)^b\ell(y_t,\hat{y})\operatorname{tr}\big\{\Lambda_{b,\hat{y}}V_{\mathbf{s}}\tilde{\rho}_t V_{\mathbf{s}}^\dagger\big\} \\ &= -\sum_{\hat{y}}\ell(y_t,\hat{y})\operatorname{tr}\Big\{M_{\hat{y}}\Big(e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}} - e^{-i\frac{\pi}{4}\sigma^{\mathbf{s}}}\rho_t^{out}e^{i\frac{\pi}{4}\sigma^{\mathbf{s}}}\Big)\Big\} \\ &= \frac{\partial L(\overrightarrow{a},\rho_t,y_t)}{\partial a_{\mathbf{s}}}. \end{aligned}$$

With that the proof is completed. □

Measuring the gradient with randomization: We have, thus far, presented a method to measure the derivative of the loss without sample reuse. Next, we present a randomized approach to measure the gradient with respect to all parameters $\overrightarrow{a}$. At each step, we randomly select a $\mathbf{s}$ as a component of $\overrightarrow{a}$ and measure the derivative of the loss with respect to it. Conditioned on a selected $\mathbf{s}$, we create a vector $\overrightarrow{Z}_t$ whose components are zeros except at the $\mathbf{s}^{th}$ component, which is the outcome of measuring the derivative with respect to $a_{\mathbf{s}}$. Consequently, by randomly selecting $\mathbf{s}$ and measuring the derivative with respect to it, we obtain an unbiased measure of the gradient. This procedure is summarized in Algorithm 1. With this argument and Lemma 1, we get the desired result which is formally stated below with the proof presented in (Heidari, Grama, and Szpankowski 2022).

Theorem 1. *Given a QNN with c_{QNN} parameters (components) in $\overrightarrow{a}$. Randomized Quantum SGD in Algorithm 1 outputs a vector $\overrightarrow{Z}_t$, whose expectation conditioned on the t^{th} sample satisfies:*

$$\mathbb{E}[\overrightarrow{Z}_t|\rho_t,y_t] = \frac{1}{c_{QNN}}\nabla L(\overrightarrow{a},\rho_t,y_t),$$

where c_{QNN} is the number of components in $\overrightarrow{a}$.

With this result, we obtain our SGD update rule as

$$\overrightarrow{a}^{(t+1)} = \overrightarrow{a}^{(t)} - \eta_t\overrightarrow{z_t},$$

where η_t is the learning rate incorporating c_{QNN}.

Algorithm 1: Randomized QSGD

Input: Training data $\{(\rho_t,y_t)\}_{t=1}^n$, Learning rate η_t
Output: Finial parameters of QNN: $\overrightarrow{a}$

Initialize the parameter $\overrightarrow{a}$ with each component selected with uniform distribution over $[-1,1]$.
for $t=1$ *to* n **do**
  Randomly select a QP in the network. Let it be the j^{th} QP of layer l.
  Pass ρ_t through the previous layers and let ρ_t^{l-1} be the output of the layer $(l-1)$.
  Randomly select a component $a_{l,j,s}$ of $\overrightarrow{a}_{l,j}$.
  Apply $V_{\mathbf{s}}$ as in (5) on $\rho_t^{l-1}\otimes|+\rangle\langle+|$ and measure the resulting state with $\mathcal{N}$ as in (6).
  With $(\hat{y},b)$ being the outcome, compute the measured derivative as $z_t = -2(-1)^b\ell(y,\hat{y})$.
  Update the s^{th} component of $\overrightarrow{a}_{l,j}$ as $a_{l,j,s} \leftarrow a_{l,j,s} - \eta_t z_t$.
return $\overrightarrow{a}$

Discussion and Analysis

Convergence Rate

We now present a theoretical analysis of our proposed approach. Specifically, we show that randomized QSGD converges to the optimal choice of parameters when the underlying objective function is bounded and convex. Moreover, we argue that randomized QSGD is sample efficient compared to the other QSGD methods, where the gradient is approximated using repeated measurements. Specifically, it has a faster convergence rate as a function of the number of samples/ copies. We note here that, similar to the classical NNs, the convexity requirement is not generally satisfied in QNNs. However, our analysis with the convexity assumption provides insights on the convergence rate even for non-convex (DNN) objective functions solved using trust region, cubic regularization, or related techniques. Our numerical results in the next section demonstrate the convergence of randomized QSGD for training QNNs.

With Theorem 1, and the standard analysis of the convergence rate of classical SGD, we obtain the following Theorem with the proof provided in (Heidari, Grama, and Szpankowski 2022).

Theorem 2. *Suppose the loss function $\ell(y,\hat{y})$ is bounded by $\gamma \in \mathbb{R}$. Suppose also that for any choice of (ρ,y), the*

sample's expected loss $L(\overrightarrow{a}, \rho, y)$ is a convex function of $\overrightarrow{a}$ with $\|\overrightarrow{a}\| = 1$. If our randomized SGD (Algorithm 1) is performed for T iterations and $\eta = \frac{1}{2\gamma\sqrt{T}}$, then the following inequality is satisfied

$$\mathbb{E}[L(\overrightarrow{a}_{ave}, \rho, y)] - L(\overrightarrow{a}^*) \leq \frac{2\gamma c_{QNN}}{\sqrt{T}},$$

where $\overrightarrow{a}_{ave} = \frac{1}{T}\sum_t \overrightarrow{a}^{(t)}$, c_{QNN} is the number of parameters, and $\overrightarrow{a}^* = \arg\min_{\overrightarrow{a}:\|\overrightarrow{a}\|=1} L(\overrightarrow{a})$.

As a result, the convergence rate of randomized QSGD scales with the number of parameters c_{QNN}. Next, we compare this result to the previous methods in which the gradient is approximated by repeated measurement on several copies of the samples. First, we bound the number of copies needed to approximate the gradient of $L(\overrightarrow{a}, \rho, y)$.

Lemma 2. *Given $\varepsilon, \delta \in (0, 1)$, with probability $(1 - \delta)$, the gradient of a sample's expected loss is approximated with error upto ε and using $O(\frac{1}{\varepsilon^2} c_{QNN} \log \frac{c_{QNN}}{\delta})$ copies.*

The proof of the lemma is given (Heidari, Grama, and Szpankowski 2022). Consequently, we fix the total number of samples/ copies for a fair comparison with our approach. With T samples randomized QSGD runs with T iterations, whereas the prior repeated sample method with gradient approximation runs with $O(\frac{T\varepsilon^2}{c_{QNN} \log c_{QNN}})$ iterations. Therefore, given T samples/ copies and under the convexity assumption, QSGD with gradient approximation has the following convergence rate:

$$\mathbb{E}[L(\overrightarrow{a}_{ave}, \rho, y)] - L(\overrightarrow{a}^*) \leq \mathcal{O}\Big(\frac{\sqrt{c_{QNN} \log c_{QNN}}}{\epsilon\sqrt{T}}\Big).$$

Hence, for QNNs with a moderate number of parameters (e.g., near-term QNNs), randomized SGD converges faster even when sample duplication is not an issue. Faster convergence holds as long as $\frac{\log c_{QNN}}{c_{QNN}} = O(\epsilon^2)$. For instance, with $\varepsilon \approx 0.05$, the number of parameters can be $c_{QNN} \approx 3000$. For larger networks, one might consider a hybrid approach.

The Expressive Power of QNNs

It is well known that classical neural networks are universal function approximators. For quantum analogs of this result, we can show that QNNs implement every quantum measurement on qubits.

Theorem 3. *For any quantum measurement $\mathcal{M}$ on the space of d qubits, there is a QNN on the space of $d' > d$ that implements $\mathcal{M}$. Moreover, it is sufficient to use QPs with a narrowness of $k = 2$.*

The proof is given in (Heidari, Grama, and Szpankowski 2022).

Impacts and Future Directions

We note that the randomized QSGD eliminates the need for replicated samples. Instead of approximating the gradient, it outputs a random variable $\overrightarrow{Z}_t$ as a probabilistic unbiased estimation. Although $\overrightarrow{Z}_t$ is not necessarily close to the gradient at any given data point, statistically, it points to the direction of the gradient over a time interval. As a result, the system is gradually pushed towards a local minimum by following a stochastic path around the negative of the gradient. Consequently, it is expected that the randomized QSGD can indeed be implemented on noisy quantum devices as the hardware imperfections contribute to additional randomness in the system. Our model can take such randomness into account, as it already incorporates randomness due to quantum measurements (see Theorem 1). An immediate future direction is to derive more concrete analyses of the convergence rate as a function of the device's infidelities.

Another improvement of this work is developing a measurement for finding multiple gradient components with one sample. In Algorithm 1 only one component of the gradient is measured. Improvements to multiple components rely on addressing the *measurement incompatibility* which is applicable when some of the $\sigma^{\mathbf{s}}$ terms mutually commute.

Numerical Results

We experimentally assess the performance of our QNNs trained using randomized QSGD in Algorithm 1. In our experiments, we focus on binary classification of quantum states with $\{-1, 1\}$ as the label set and with conventional 0-1 loss to measure predictor's accuracy. Numerical experiments are simulated on a classical computer.

Dataset

We use a synthetic dataset from recent efforts (Mohseni, Steinberg, and Bergou 2004; Chen et al. 2020; Patterson et al. 2021; Li, Song, and Wang 2021) focused on quantum state discrimination. In this dataset, for each pair of parameters $u, v \in [0, 1]$, three input states are defined as follows:

$$|\phi_u\rangle = \sqrt{1 - u^2}\,|00\rangle + u\,|10\rangle\,,$$
$$|\phi_{\pm v}\rangle = \pm\sqrt{1 - v^2}\,|01\rangle + v\,|10\rangle\,.$$

Therefore, the two quantum states to be classified are:

$$\rho_1(u) = |\phi_u\rangle\langle\phi_u|\,,$$
$$\rho_2(v) = \frac{1}{2}\Big(|\phi_{+v}\rangle\langle\phi_{+v}| + |\phi_{-v}\rangle\langle\phi_{-v}|\Big).$$

We assign label y = -1 for the first state and $y = 1$ for the second state. For each sample, we generate a state ρ_1 with probability $p = 1/3$ and state ρ_2 with probability $(1 - p) = 2/3$. Furthermore, for each sample, parameters u and v are selected randomly, independently, and with the uniform distribution on $[0, 1]^2$. Hence, there are infinitely many realizations of samples without replication.

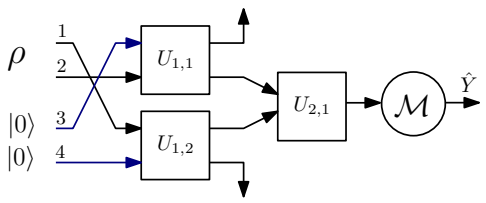

Figure 3: The QNN used in the experimental study.

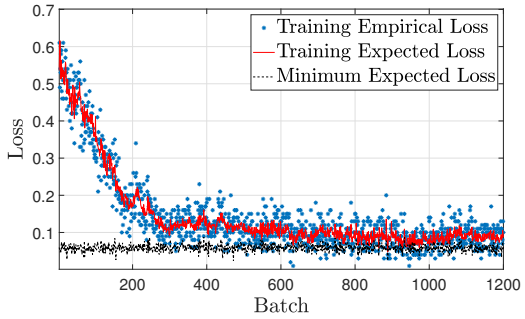

Figure 4: The training loss of the QNN in Figure 3 using randomized QSGD. The loss is averaged over each batch of size 100 samples. The figure shows the loss vs. batch number as the QNN is updated using Algorithm 1. The minimum expected loss is calculated using Lemma 3.

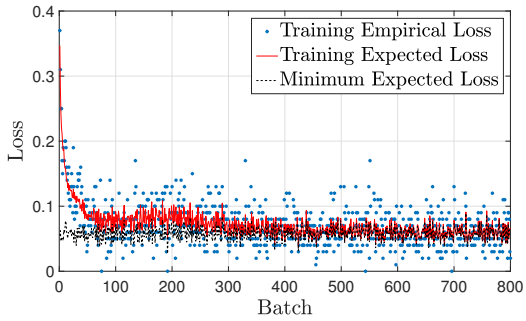

Figure 5: The training loss of the QNN in Figure 3 using exact computation of the gradient.

QNN Setup

The QNN we use is shown in Figure 3. It has two layers with two quantum QPs in the first layer and one quantum QP in the second layer. The input states are padded with two auxiliary qubits $|0\rangle \otimes |0\rangle$; hence, $d = 2$ and $d' = 4$. The bandwidth parameter is set to $k = 2$ for each QP. The measurement used in the readout qubits has two outcomes in $\{-1, 1\}$, each measured along the computational basis as

$$\mathcal{M} = \Big\{ |00\rangle\langle 00| + |11\rangle\langle 11|\,, \quad |01\rangle\langle 01| + |10\rangle\langle 10| \Big\}.$$

Experimental Results[2]

We train the QNN in Figure 3 using Algorithm 1 and with $\eta_t = \frac{\alpha}{\sqrt{t}}$ and $\alpha \approx 0.77$. Our training process has no epochs in contrast to conventional SGD, since sample replication is prohibited. Therefore, to show progress during the training phase, we group samples into multiple batches, each of 100 samples. For each batch, we compute the average empirical loss of the QNN with updated parameters. Figure 4 shows the loss during the training phase averaged over each batch. In addition to the empirical loss, we compute the average of the sample's expected loss $L(\overrightarrow{\alpha}^{(t)}, \rho_t, y_t)$ for all samples in the batch (as in (2)). Furthermore, we compare the performance of the network with the optimal expected loss within each batch. The following lemma provides a closed-form expression for the optimal loss with the proof presented in (Heidari, Grama, and Szpankowski 2022).

Lemma 3. *Given a set of m labeled density operators $\{(\rho_j, y_j)\}_{j=1}^m$ with $y_j \in \{-1, 1\}$, the minimum of the expected sample loss averaged over the m samples as in* (2) *equals: $\frac{1}{2}\Big(1 - \Big\|\frac{1}{m}\sum_j y_j \rho_j\Big\|_1\Big)$, where $\|\cdot\|_1$ is the trace norm, and the minimum is taken over all measurements applied on each ρ_j for predicting y_j.*

As seen in Figure 4, training loss converges to its minimum as more batches of samples are used, showing that randomized QSGD converges without the need for exact copies.

[2]The source codes and implementations are available at https://github.com/mohsenhdkh/RandomizedQSGD.

To assess the impact of replicated data, we repeat the experiment, this time by allowing unlimited exact copies of samples. Therefore, instead of randomized QSGD, we train the QNN by exactly computing the gradient as in (7). Figure 5 shows the training loss for each batch of samples. Comparing results in Figure 5 (unlimited replication) with Figure 4 (no replication), we observe a near-optimal training of the QNN using the randomized QSGD.

Comparison of Loss Function Values

Once the QNN is trained, we generate a new set of 100 samples to test the performance of the QNN. Table 1 shows the resulting accuracy of the network, as compared to the optimal accuracy from Lemma 3 and the variant of the experiment with replicated samples. The resulting accuracy for our randomized QSGD is observed to be close to optimum.

Concluding Remarks

In this work, we addressed two key shortcomings of conventional QNNs – the need for replicated data samples for gradient computation and the exponential state space of QNNs. To address the first problem, we present a novel randomized SQGD algorithm, along with detailed theoretical proofs of correctness and performance. To address the exponential state space, we propose the use of band-limited QPs, showing that the resulting QNN can be trained in time linear in the number of QPs. We present detailed theoretical analyses, as well as experimental verification of our results. To this end, our work represents a major step towards realizable, scalable, and data-efficient QNNs.

	QNN	QNN (gradient comp.)	Optimal
Acc.	$91\% \pm 1$	$93.48\% \pm 1$	93.51%

Table 1: Validation of accuracy of the trained QNN in Figure 3 for classifying test samples. The first column is for the QNN trained with randomized QSGD. The second column is for the QNN trained using direct gradient computation. The third is the optimal accuracy derived from Lemma 3.

Acknowledgements

This work was supported in part by NSF Center on Science of Information Grants CCF-0939370 and NSF Grants CCF-2006440, CCF-2007238, OAC-1908691, and Google Research Award.

References

Beer, K.; Bondarenko, D.; Farrelly, T.; Osborne, T. J.; Salzmann, R.; Scheiermann, D.; and Wolf, R. 2020. Training deep quantum neural networks. *Nature Communications*, 11(1).

Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S. C.; Endo, S.; Fujii, K.; McClean, J. R.; Mitarai, K.; Yuan, X.; Cincio, L.; and Coles, P. J. 2020. Variational Quantum Algorithms. arXiv:2012.09265.

Chen, H.; Wossnig, L.; Severini, S.; Neven, H.; and Mohseni, M. 2020. Universal discriminative quantum neural networks. *Quantum Machine Intelligence*, 3(1).

De Wolf, R. 2008. *A Brief Introduction to Fourier Analysis on the Boolean Cube*. Number 1 in Graduate Surveys. Theory of Computing Library.

Dixit, A. V.; Chakram, S.; He, K.; Agrawal, A.; Naik, R. K.; Schuster, D. I.; and Chou, A. 2021. Searching for Dark Matter with a Superconducting Qubit. *Physical Review Letters*, 126(14): 141302.

Farhi, E.; Goldstone, J.; and Gutmann, S. 2014. A Quantum Approximate Optimization Algorithm. arXiv:1411.4028.

Farhi, E.; and Neven, H. 2018. Classification with Quantum Neural Networks on Near Term Processors. *arXiv:1802.06002*.

Gao, J.; Qiao, L.-F.; Jiao, Z.-Q.; Ma, Y.-C.; Hu, C.-Q.; Ren, R.-J.; Yang, A.-L.; Tang, H.; Yung, M.-H.; and Jin, X.-M. 2018. Experimental Machine Learning of Quantum States. *Physical Review Letters*, 120(24): 240501.

Garg, S.; and Ramakrishnan, G. 2020. Advances in Quantum Deep Learning: An Overview. *arXiv:2005.04316*.

Guerreschi, G. G.; and Smelyanskiy, M. 2017. Practical optimization for hybrid quantum-classical algorithms. arXiv:1701.01450.

Heidari, M.; Grama, A.; and Szpankowski, W. 2022. Toward Physically Realizable Quantum Neural Networks. *arXiv:2203.12092*.

Heidari, M.; Padakandla, A.; and Szpankowski, W. 2021. A Theoretical Framework for Learning from Quantum Data. In *IEEE International Symposium on Information Theory (ISIT)*.

Kudyshev, Z. A.; Bogdanov, S. I.; Isacsson, T.; Kildishev, A. V.; Boltasseva, A.; and Shalaev, V. M. 2020. Rapid Classification of Quantum Sources Enabled by Machine Learning. *Advanced Quantum Technologies*, 3(10): 2000067.

Lewenstein, M. 1994. Quantum perceptrons. *Journal of Modern Optics*, 41(12): 2491–2501.

Li, G.; Song, Z.; and Wang, X. 2021. VSQL: Variational Shadow Quantum Learning for Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9): 8357–8365.

Ma, Y.-C.; and Yung, M.-H. 2018. Transforming Bell's inequalities into state classifiers with machine learning. *npj Quantum Information*, 4(1).

Massoli, F. V.; Vadicamo, L.; Amato, G.; and Falchi, F. 2021. A Leap among Entanglement and Neural Networks: A Quantum Survey. *arXiv:2107.03313*.

Mitarai, K.; Negoro, M.; Kitagawa, M.; and Fujii, K. 2018. Quantum circuit learning. *Physical Review A*, 98(3): 032309.

Mohseni, M.; Steinberg, A. M.; and Bergou, J. A. 2004. Optical Realization of Optimal Unambiguous Discrimination for Pure and Mixed Quantum States. *Physical Review Letters*, 93(20): 200403.

Montanaro, A.; and Osborne, T. J. 2010. Quantum boolean functions. *arXiv:0810.2435*.

Orús, R.; Mugel, S.; and Lizaso, E. 2019. Quantum computing for finance: Overview and prospects. *Reviews in Physics*, 4: 100028.

Patterson, A.; Chen, H.; Wossnig, L.; Severini, S.; Browne, D.; and Rungger, I. 2021. Quantum state discrimination using noisy quantum neural networks. *Physical Review Research*, 3(1): 013063.

Schuld, M.; Bocharov, A.; Svore, K. M.; and Wiebe, N. 2020. Circuit-centric quantum classifiers. *Physical Review A*, 101(3): 032308.

Schuld, M.; Sinayskiy, I.; and Petruccione, F. 2014. The quest for a Quantum Neural Network. *Quantum Information Processing*, 13(11): 2567–2586.

Torrontegui, E.; and Garcia-Ripoll, J. J. 2018. Unitary quantum perceptron as efficient universal approximator. *EPL 125 30004, (2019)*.

Toth, G.; Lent, C. S.; Tougaw, P.; Brazhnik, Y.; Weng, W.; Porod, W.; Liu, R.-W.; and Huang, Y.-F. 1996. Quantum cellular neural networks. *Superlattices and Microstructures*, 20(4): 473–478.