Final Report: Expressive Power of Parametrized Quantum Circuits

Jingling Li, Qingyang Tan, Xuchen You

December 13, 2019

Abstract

Parametrized Quantum Circuits provide a near-term plausible opportunity to utilize the power of quantum computing. [1] provides an overview of parametrized quantum circuits applied to machine learning. However, as stated in the paper, the main purpose of the survey is pedagogical and non-technical. For our course project, we aim at a more thorough review with more technical details with a focus on expressive power of parametrized quantum circuits.

1 Introduction

The main purpose of our course project is to provide a thorough and technical survey on the application of parametrized quantum circuit to machine learning.

The purpose of machine learning is to identify a hypothesis h from a hypothesis set $\mathcal{H}$ that is close to the target concept $c : \mathcal{X} \to \mathcal{Y}$, which is a mapping from the input space $\mathcal{X}$. A typical input to a machine learning algorithm is a set of samples S with size m ($S := \{(x_i, y_i))\}_{i=1}^m \subset \mathcal{X} \times \mathcal{Y}$, where $y_i = c(x_i)$.

A parametrizatoin of quantum circuit defines a hypothesis class in the machine learning literature. A desirable hypothesis set posesses the following two properties: (1) it need to be expressive enough to contain a hypothesis that is close to the concept and (2) given the sample set, it should allow efficient search over the hypothesis set.

Our survey will mainly focus on known design of parametrized circuits (parametrization), and survey what kind of expressivity argument has been provided (expressive power).

Our mid-term report is organized as following:In Section 2, we describe a pioneering work in parameterized quantum circuits and list some existing work discussing the power and optimization of such formulation. In Section 3.1 we describe a line of work that can be considered as a generalization of quantum approximation optimization algorithm. In Section 3.3 we describe a few works that introduce non-linearity in a different way. In the last section (Section 5), we summarize some known emiprical results of parametrized quantum circuits.

2 Quantum Approximation Optimization Algorithms

In this section, we survey the prototype of quantum parametrized circuits, quantum approximation optimization algorithm [2].

QAOA was first proposed as a general algorithm for solving maximum satisfiability problems (MaxSat). Each instance of such a problem is specified by a set of constraints (or more specifically, boolean functions):

$$f_l : \{0,1\}^n \to \{0,1\}, l \in [m]$$

The goal is to find an optimal assignment of the boolean variables such that the number of satisfied constraints is maximized.

$$\max_{z \in \{0,1\}^n} \sum_{l \in [m]} f_l(z)$$

The following quantum procedure was proposed to provide an approximate solution to the combinatorial problem:

1. **Workspace.** For a combinatorial optimization problem with dimension-n, we represent the input, intermediate steps and output with n-qubits. The input to the quantum parametrized circuit is fixed to be $|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle$

2. **Encoding of the problem instance.** the given instance is encoded in to a p-layered quantum circuit parameterized with $\boldsymbol{\gamma}, \boldsymbol{\beta} \in (0, 2\pi)^p$ as following:

   (a) Encode each constraint into a localized (in the sense that only the boolean variables that are relevant to the constraints are involved) Hamiltonian: $\{H_{C_l}\}$. The set of local Hamiltonian specifies the problem Hamiltonian as $H_C = \sum_{l \in [m]} H_{C_l}$.

   (b) The mixing Hamiltonian is defined as $H_B = \sum_{l \in [m]} \sigma_x^l$

   (c) The parametrized quantum circuit is then defined such that for the output state $|t\rangle$, we have:

   $$|t\rangle_{\boldsymbol{\gamma},\boldsymbol{\beta}} = \Pi_{j=1}^{p} e^{-i\beta_j H_B} e^{-i\gamma_j H_C} |s\rangle$$

3. **Optimization.** The optimization problem is formulated as:

   $$\max_{\boldsymbol{\gamma},\boldsymbol{\beta}} \langle t|_{\gamma,\beta} H_C |t\rangle_{\gamma,\beta}$$

4. **Decoding of results** Given the optimized output state $|t\rangle_{\hat{\boldsymbol{\gamma}},\hat{\boldsymbol{\beta}}}$, sampling boolean vectors by measuring the output state with computational basis.

2.1 The power of QAOA

Realizability The observable H_C is designed such that it is diagonal under the computational bases. Thus each computational basis $|z\rangle, z \in \{0,1\}^n$ is an eigen-vector of the system, and the corresponding the eigenvalue is the number of constraints c_z satisfied by the assignment of boolean variables z:

$$H_C = \sum_{z \in \{0,1\}^n} c_z |z\rangle \langle z|$$

Consider the expansion of arbitrary state with respect to the computational basis $|\psi\rangle = \sum \alpha_z |z\rangle$, we have the $\langle \psi| H_C |\psi\rangle = \sum_z \|\alpha_z\|^2 c_z = \mathbb{E}_{z\sim p}[c_z]$, from which we can see that the optimal solution is achievable for some $|\psi\rangle$ in the eigen-subspace of the optimal value of the combinatorial optimization problem. This observation provides a justification for a series of quantum formulation of optimization problems based on ground state finding [3, 4, 5]. It is proven in [6] that it is not possible to sample from the distribution of the output state efficiently with classical devices as long as some common belief in complexity theory holds (more emphasis will be put on this for the second half of the semester.) Another work illustrating the strength of QAOA [7] empirically calculated the approximation ratio of QAOA solutions to MAXCUT and observe (empirically, the strength of the evidence may be doubtful) that it achieves better approximation ratio compared with the classicial 0.878 achieved through the SDP relaxation.

2.2 Optimization of QAOA

Although optimization is not the central topic for our survey, we include a brief review of existing optimization methods for QAOA. A naive brute-force grid-search method was proposed when the method was first proposed [2]. [8] empirical studies the parameter distribution and proposed FOURIER based on the observation that $\boldsymbol{\gamma}^*$ and $\boldsymbol{\beta}^*$ are smooth with respect to indexes. The FOURIER heuristics involves the following reparametrization:

$$\gamma_i = \sum_j u_j \sin\left[\left(j - \frac{1}{2}\right)\left(i - \frac{1}{2}\right)\frac{\pi}{p}\right] \tag{1}$$

$$\beta_i = \sum_j v_j \cos\left[\left(j - \frac{1}{2}\right)\left(i - \frac{1}{2}\right)\frac{\pi}{p}\right] \tag{2}$$

and iteratively stack the layers by using the outcome of p-layered optimization outcome as the initialization of the $(p+1)$-layered formulation. Other proposed optimization method include the classical gradient descent methods as described in [9, 7].

3 Formulations: linear and non-linear QNNs

3.1 Linear formulations

While the measurement of QAOA is associated with an instance of a combinatorial optimization problem and the input is a fixed state (a uniform superposition of all different assignments), such formulation can be generalized for machine learning tasks, by associating the input state with a data set and generalized the objective function to be a function of the observable.

3.2 Parametrization and Optimization

The parametrization of quantum circuit determine how efficiently a search can be conducted on the parameter space, and partially determine how expressive is the family of circuits. A natural and prevailing parametrization is a sequence of parametrized rotation matrices ([10, 11]) and non-tunable entanglement gates, which generalizes the QAOA setting. Each layer is a unitary operator $U_j(\theta_j) = e^{-i\frac{\theta}{2}P_j}$, $j \in \{1, \cdots, J\}$, with fixed $P_j \in \{I, X, Y, Z\}^{\otimes n}$, where ρ is the density matrix of the input state. The output is a function of the measurement of the final state specified by M:

$$\mathsf{tr}\left(MU_{J:1}(\boldsymbol{\theta})\rho U^{\dagger}_{J:1}(\boldsymbol{\theta})\right) \tag{3}$$

Alternatively, in [12], the circuit is parameterized as the following:

$$y \cdot \mathsf{tr}\left(|x, 1\rangle\langle x, 1|U^{\dagger}(\boldsymbol{\theta})MU(\boldsymbol{\theta})\right) \tag{4}$$

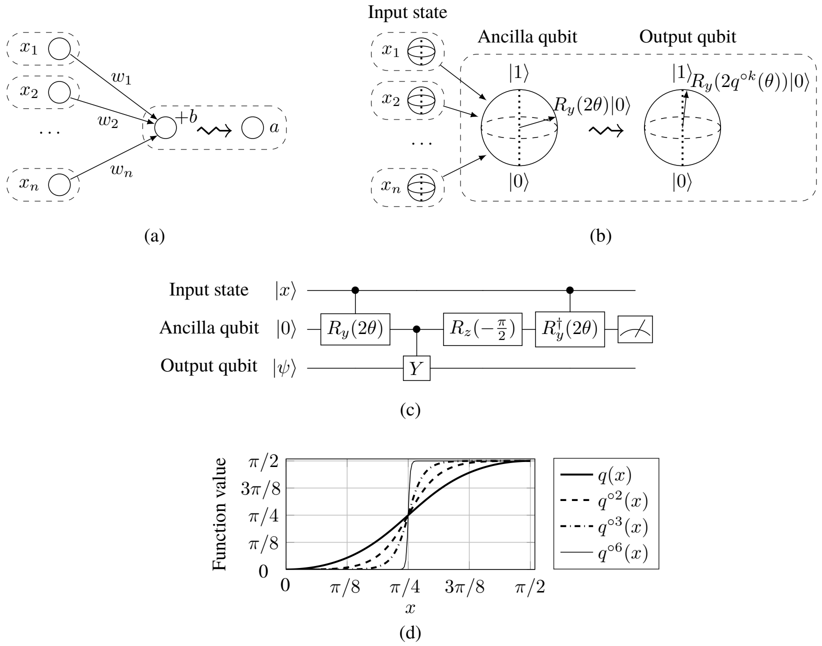
Specifically the measurement allows a decomposition: $M = I_{N\times N} \otimes Y$.

While the aforementioned two formulations assume fix choice and rotation angle (determined by the pauli matrices) and depend on gradient descent for optimization, [13] provides a close-form coordinate descent update when the goal is to minimize Equation (3). In addtion, they illustrated through simulation the benefit of an enumeration in the choice of the Pauli matrices.

3.3 Introducing non-linearity

The drawback of the previous parametrization is that they are linear in terms of the input state. The non-linearity needed for the expressive power were introduced through a non-linear encoding. In contrast to the aforementioned formulations ([14, 15, 16]) introduces non-linearity through different methods. [14] presents a realization of a quantum neuron with arctan as the nonlinear activation function using the RUS (repeat-until-success) circuit as a building block. Figure 1(c) demonstrates how we can use the the RUS circuit to realize a rotation with angle $q(\varphi) = \arctan(\tan^2\varphi)$. The nonlinear function $\arctan(\tan^2\varphi)$ is plotted in Figure 1(d) given various input values of the angle. Figure 1(a)-(b) compares a classical neuron with the quantum neuron they introduce. [15] presents a quantum model of a perceptron, which can be used as an elementary nonlinear classifier of simple patterns. The nonlinearity in [15] is achieved by performing a quantum measurement of the ancilla qubit in computational basis. [16] presents a quantum perceptron on a qubit with a nonlinear activation function formulated as equation (5)

$$\hat{U}_j(\hat{x}_j; f)\,|0_j\rangle = \sqrt{1 - f(\hat{x}_j)}\,|0_j\rangle + \sqrt{f(\hat{x}_j)}\,|1_j\rangle \tag{5}$$



(a)

(b)

(c)

(d)

Figure 1: Basic setup of the quantum neuron model. Figure from [14]

4 Expressive Power

4.1 Expressive Power: Overview

The expressive power of neural network is a heated research topic in the field of classical machine learning [17][18]. It describes a set of function that can be represent by a hypothesis set. Similar question can be asked for parameterized quantum circuits. For quantum circuit parametrized as in Equation (3)([10]), since the circuit is linear in the input state, the circtuit can only ouput a state affine to the input state. [12] discussed the expressive power of the quantum circuit in Equation (4) in terms of families of boolean functions. [19] discussed the expressive power of parameterized circuits for generative tasks.

4.2 Classical results

The potential of multiplayer perceptrons being universal approximator was first studies in [20]. Let I_n denote the n-dimensional unit cube, $[0,1]^n$ and let $C(I_n)$ denote the space of continuous functions on I_n. The main result of [20] is stated as follows:

Theorem 4.1. *the function class consisting multiplayer perceptrons with sigmoid activation function on n-dimensional input is dense in $C(I_n)$.*

[21]'s work can be summarized by the following two theorems (Theorem 4.2 and Theorem 4.3):

Theorem 4.2. *If the activation function is unbounded and nonconstant, then the function class consisting multiplayer perceptrons with n hidden units on k-dimensional input is dense in $L^p(\mu)$ for all finite measures μ on $\mathbb{R}^k$*

Theorem 4.3. *If the activation function is continuous, bounded and non-constant, then the function class consisting multiplayer perceptrons with n hidden units on k-dimensional input is dense in $C(X)$ for all compact subsets X of $\mathbb{R}^k$, where $C(X)$ is the space of all continuous functions on X.*

[21] In summary, Hornik's results in [21] extended the universal approximation theorem to a broad class of activation functions beyond sigmoid function.

[22] studies the expressive power of that ReLU networks on n-dimensional input. First, it defines $w_{\min}(d_{in}, d_{out})$ to be the minimal value of w such that for every continuous function $f : [0, 1]^{d_{in}} \to \mathcal{R}^{d_{out}}$ and every $\varepsilon > 0$ there is a ReLU net $\mathcal{N}$ with input dimension d_{in}, hidden layer widths at most w, and output dimension d_{out} that $\varepsilon-$approximates f :

$$\sup_{x \in [0,1]^{d_{in}}} \|f(x) - f_{\mathcal{N}}(x)\| \leq \varepsilon.$$

The main result of [22] is an estimation regarding $w_{\min}(d_{in}, d_{out})$, which is stated as follows:

Theorem 4.4. *For every $d_{in}, d_{out} \geq 1$,*

$$d_{in} + 1 \leq w_{min}(d_{in}, d_{out}) \leq d_{in} + d_{out}.$$

4.3 Representing boolean function

[12] considers representing boolean functions $f : \{0, 1\}^n \to \{0, 1\}$ with the ansatz in (4) that involves $(n+1)$ qubits:

1. **Workspace.** A feature vector x from space $\{0, 1\}^n$ is encoded into the computational bases; the auxiliary qubit is initialize as $|1\rangle$;

$$|\boldsymbol{z}, 1\rangle := |z_1\rangle \otimes \cdots \otimes |z_n\rangle \otimes |1\rangle$$

2. **Ansatz.** The ansatz is composed of $(n + 1)$-qubit with the input encoded into the first n-qubits and use the last qubit as auxiliary read-out qubit;

$$U(\boldsymbol{\theta}) = U_L(\boldsymbol{\theta}_L) \cdots U_l(\boldsymbol{\theta}_l) \cdots U_1(\boldsymbol{\theta}_1)$$

3. **Prediction.** The output of the circuit is the measurement on the auxiliary qubit with Pauli-Y matrix Y_{n+1}:

$$f(\boldsymbol{z}; \boldsymbol{\theta}) = \langle \boldsymbol{z}, 1| U(\boldsymbol{\theta})^\dagger Y_{n+1} U(\boldsymbol{\theta}) |\boldsymbol{z}, 1\rangle$$

It is shown that, for any boolean function on $\{0, 1\}^n$ there exists an $(n + 1)$-qubit ansatz and a set of parameters that represents the boolean function. In order to see that, we first need the following results:

Fact 1.

$$e^{\pm i \frac{\pi}{4} X} |1\rangle = |\pm i\rangle$$

Remark 4.5. This fact allows us to represent the function by performing $e^{\pm i \frac{\pi}{4} X}$ on the auxiliary qubit based on the input state.

Another important result is the following representing theorem of boolean function:

Fact 2 (Reed-Muller Representation)**.** *Any boolean function f has the following expansion that resembles Taylor expansion, there exists a real function a on the power set of $[n]$ such that:*

$$\begin{aligned} f(z_1, z_2, \cdots, z_n) &= a_\emptyset \oplus a_{\{1\}} z_1 \oplus \cdots \oplus a_{\{1,2\}} z_1 z_2 \oplus \cdots \oplus a_{[n]} z_1 \cdots z_n \\ &= \oplus_{S \subseteq [n]} a_S \prod_{i \in S} z_i \end{aligned}$$

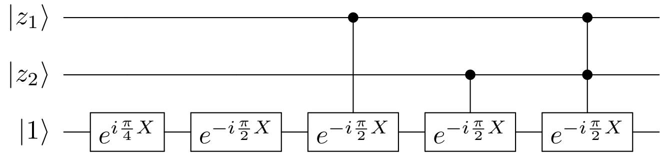
Example 1. Consider a boolean function f on $\{0, 1\}^2$:

$$f(z_1, z_2) = \neg(z_1 \vee z_2)$$

The Reed-Muller Representation of this function is :

$$f(z_1, z_2) = 1 \oplus z_1 \oplus z_2 \oplus z_1 z_2$$

Therefore the function can be represented by the following ansatz:



Combining the first two facts, one can construct the following ansatz given the Reed-Muller coefficients:

$$U(\boldsymbol{\theta}) = \exp(i\frac{\pi}{4}X_{n+1})\exp(-i\frac{\pi}{2}(\sum_{S\subseteq[n]} a_S \otimes_{i\in S} Z_i) \otimes X_{n+1})$$

Additionally, the following fact allows a construction of ansatz with 2-qubit gates:

Fact 3 (Representing N-control qubit gates with 2-qubit gates [23]). *$(n+1)-$ controlled gate with n control qubit can be represented by n^2 2-qubit gates.*

Remark 4.6. A relative result of the above fact can be found in [24], where it is shown that a $d \times d$ unitary matrices can be represented as multiplication of d $(d-1) \times (d-1)$ qubit gates.

Remark 4.7. Since the number of terms in the Reed-Muller representation is exponential in the number of qubits, in order to represent arbitray boolean functions, the number of two-qubit gates is exponential in terms of n.

Remark 4.8. This result also provide a characterization of family of boolean functions that can be represented efficiently (*i.e.* boolean functions with sparse coefficients). As mentioned in [12], an example is the subset parity function, where the expansion can be written as:

$$f^*_{\mathcal{S}}(\boldsymbol{z}) = \oplus_{i\in\mathcal{S}} z_j$$

4.4 Representing generative models

Another important application of quantum circuit is sampling.

Fixing an input state (e.g. $|0^{\otimes n}\rangle$), measurement with computational bases on the output state naturally induces a distribution over $\{0,1\}^n$.

It is shown in [19], parametrized quantum circuit can efficiently represent instantaneous quantum polynomial (with $O(\mathsf{poly}(N))$ single qubit gates and CNOT gates),

Remark 4.9. This result does not have direct practical meaning, but an illustration of achieving something might not be efficiently achieved by classical models.

4.5 Representing unitary matrices

Whether a given gate sets can efficiently represent all unitary matrices is an important question in the study of quantum circuits. We can ask the same question about parametrized quantum circuits:

Question 1. Consider the parametrized circuit in 3, how many parameter it needs to represent all unitary gates?

While being a native in the quantum realm, unitary matrices have been introduced in classicial models ([25, 26]) to avoid the issue of vanishing and exploding gradients in recurrent models, where norm is proportional to the power of singular values of the weight matrices[27]. The classical paper [26] provides a necessary condition for a parametrized circuit to be able to represent all unitary matrices:

Theorem 4.10 (Necessary condition for representing unitary matrices with real parameters). *For any real parameterization with number of parameters P less than N^2, it can not represent all $N \times N$ unitary matrices.*

Remark 4.11. We have the following two remarks on the theorem:

1. This theorem does not provide sufficient condition of parametrized representation;

2. This theorem also does not provide sufficient conditions for approximating arbitrary unitary matrices.

5 Empirical Observations

For completeness, we summarized the emipirical performance of aforementioned parametrized circuits:

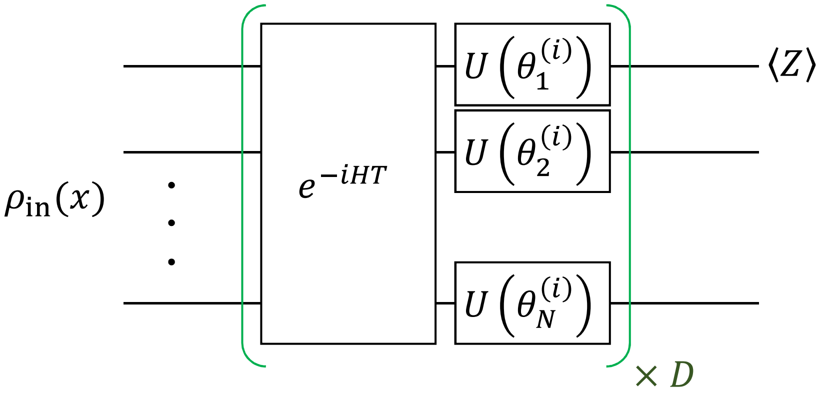

$\rho_{\text{in}}(x)$
e^{-iHT}
$U\left(\theta_1^{(i)}\right)$
$U\left(\theta_2^{(i)}\right)$
$U\left(\theta_N^{(i)}\right)$
$\langle Z \rangle$
$\times D$

Figure 2: Quantum circuit used in numerical simulations of QCL.

1. Quantum Circuit Learning (QCL) [10]

   In the simulations of QCL, the authors utilized a quantum circuit as shown in Fig. 2 with the number of qubit $N = 6$, and the depth of network $D = 6$. Each circuit needed to be optimized is formulated as:

   $$U(\theta_j^{(i)}) = R_j^X(\theta_{j1}^{(i)}) R_j^Z(\theta_{j2}^{(i)}) R_j^X(\theta_{j3}^{(i)}). \tag{6}$$

   We use $\boldsymbol{\theta}$ to denote the parameter of all the arbitrary unitaries $U(\theta_j^{(i)})$. In all the experiments, $\boldsymbol{\theta}$ is initilized with random numbers uniformly distributed on $[0, 2\pi]$.

   In the paper [10], three problems are used to show the power of QCL. First, the authors fit the circuit to functions $(f(x) = x^2, e^x, \sin x, |x|)$, to demonstrate the power of representing nonlinear functions. Input state of the circuit is prepared by operating $U_{\text{in}}(x) = \prod_j R_j^Z(\cos^{-1} x^2) R_j^Y(\sin^{-1} x)$ on $|0\rangle$. The output is taken from Z expectation value of the first qubit shown in Fig. 2 and they use the normal quadratic loss as the cost function and For each function, they fed 100 training samples. The results are shown in Fig. 3. The authors pointed out that QCL has the ability to bring out the high order terms of $\sin x$ and nonanalytical function $|x|$, which are hidden in the input teacher samples.

   The second experiment is a classification problem on the training dataset shown in Fig. 4 (a): Blue and red points are labeled 0 and 1 respectively, and the input data has the form $\boldsymbol{x}_i = (x_{i,0}, x_{i,1})$ and the input to the circuit is the result of applying $U_{\text{in}}(x) = \prod_j R_j^Z(\cos^{-1} x_{i,j \bmod 2}^2) R_j^Y(\sin^{-1} x_{i,j \bmod 2})$ on $|0\rangle$. The output is taken from the expectation value of the Pauli Z operator of the first 2 qubits, and then transformed by softmax function $\boldsymbol{F}$: $\boldsymbol{y}_i = (y_{i,0}, y_{i,1}) = \boldsymbol{F}(\langle Z_1(\boldsymbol{x}_i, \boldsymbol{\theta})\rangle, \langle Z_2(\boldsymbol{x}_i, \boldsymbol{\theta})\rangle)$. They use the cross-entropy loss to train the framework. Learned output is shown in Fig. 4. (b). Classically, we need to use kernel-trick to avoid directly using a large number of basis function, however, QCL can benefit from the use of a quantum computer to utilizes an exponentially large number of basis functions under certain constraints.

   The final one is a regression over quantum many-body dynamics. They demonstrated that they can use 6-qubit circuit to approximate a 10-spin system well.

2. Quantum Neural Networks (QNN) on Near Term Processors [12]

   The paper [12] performed its simulation experiment on MNIST hand-written digit dataset. To perform the experiment on a traditional simulator, they downsample the data to 4 × 4, which could be represented by a 16-bit string. Also, they limited the samples labeled as 3 and 6 to fit the 1 readout bit.

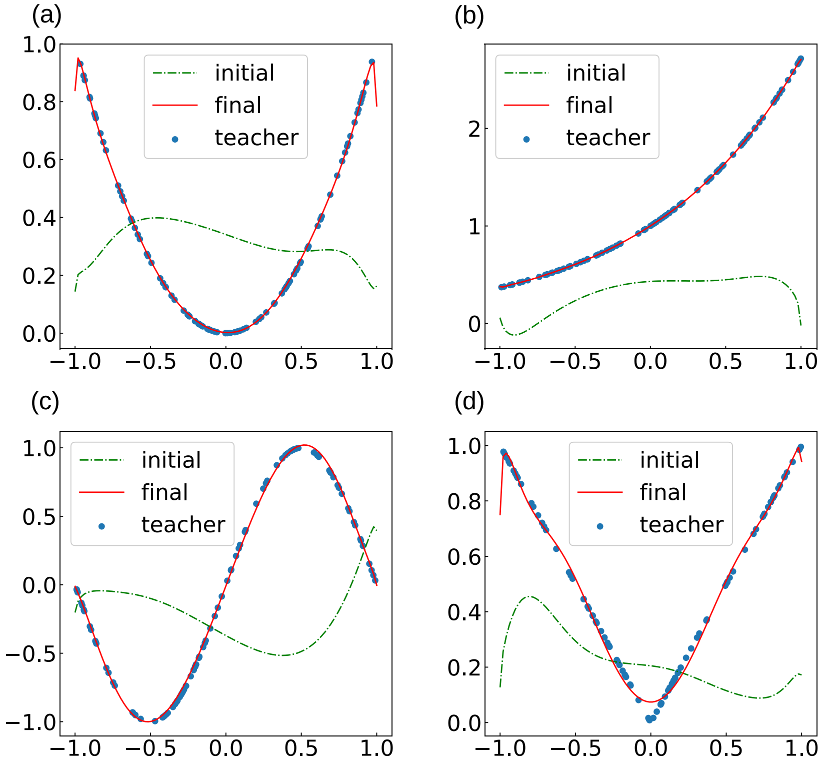


Figure 3: Function fitting results of QCL.

First, they restricted their gate set to ZX and XX, with the second bit always being the readout bit, while the first bit being one of the other 16 bits. They performed their algorithm on 3 layers of ZX and 3 layers of XX with 96 parameters in in total, which achieved two percent categorical error after seeing less than the full sample set. This verified that QNN has the power similar to traditional neural network.

Then, they tried to borrow the idea of feeding a 'batch' of samples from traditional neural network training. Benefiting from superposition in quantum computing, they proposed to combine different samples into single superposition states and then evaluate the gradient on a suitable loss function. For a binary classification problem, one can divide the sample space into those samples labeled as +1 and those labeled as -1:

$$|+1\rangle = N_+ \sum_{z:l(z)=1} e^{i\varphi_z} |z, 1\rangle \qquad |-1\rangle = N_- \sum_{z:l(z)=-1} e^{i\varphi_z} |z, 1\rangle \, , \tag{7}$$

where N_+ and N_- are normalization factors.

Consider the unitary operator $U(\vec{\theta})$ representing the network, the expected value of Y_{n+1} of the state obtained by having this operator act on $|+1\rangle$ is the average over all samples with the label +1 of the quantum neural network's predicted label values. Similarly for the state $|-1\rangle$. Using this information, they formulate the empirical risk of the whole sample space as:

$$1 - \frac{1}{2}\Big(\langle +1| U^\dagger(\vec{\theta}) Y_{n+1} U(\vec{\theta}) |+1\rangle - \langle -1| U^\dagger(\vec{\theta}) Y_{n+1} U(\vec{\theta}) |-1\rangle \Big). \tag{8}$$

Go back to the digit classification problem, they were able to drive the empirical loss to a value of around 0.5 and the quantum neural network had low categorical error on a test set of individual data samples. In the quantum batch case, the progress of the empirical risk smoothly decreased until it

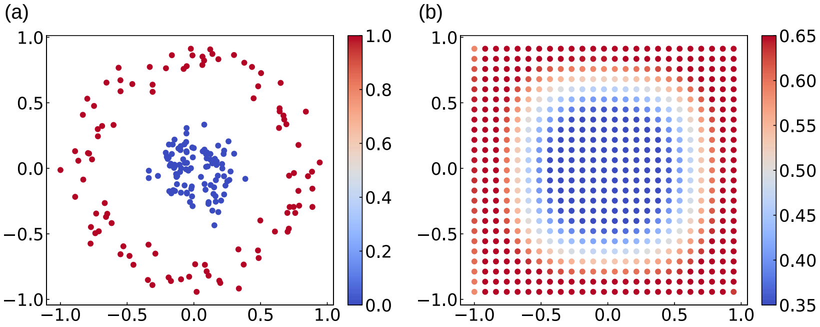


Figure 4: Classfication data and result for QCL.

settled at a local minimum. This demonstrated the benefit of using superpositions of samples than learning by presenting sequentially single strings.

3. Quantum Machine Learning with Tensor Networks [28]

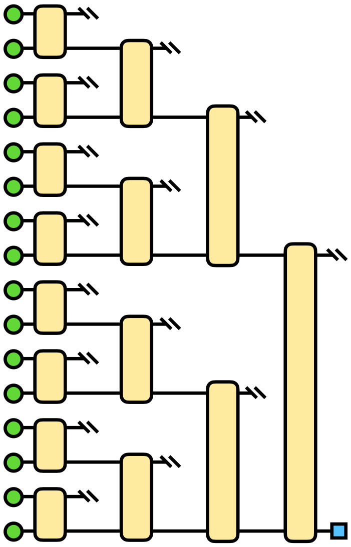


Figure 5: Model architecture used in the experiments of [28]

This paper [28] trained a discriminative model based on a tree tensor network with architecture shown in Fig. 5, for a supervised learning task, namely labeling image data. They trained circuits with a single output qubit at each node to recognize grayscale images of size 8 × 8 belonging to one of two classes using the simultaneous perturbation stochastic approximation (SPSA) optimization procedure. The data are also from the MNIST data set. Each node of the network is parameterized by $U = \exp(iH)$, where H is a Hermitian matrix. The free parameters are chosen to be the elements forming the diagonal and upper triangle of each Hermitian matrix, resulting in exactly 1008 free parameters for the 8 × 8 image recognition task. They achieved test accuracy above 95%.

4. Quantum Perceptron [15]

This paper [15] built quantum perceptron with non-linearty by performing a quantum measurement of the ancilla qubit in computational basis. They performed their experiments on IBM real quantum computer with 2 qubits. With 2 qubits, they can feed $2^2 = 4$ dimensional input as 2×2 binary image, and 2^{2^2} different patterns could be analyzed. As shown in Fig. 6a, they labeled each pixel left to right,

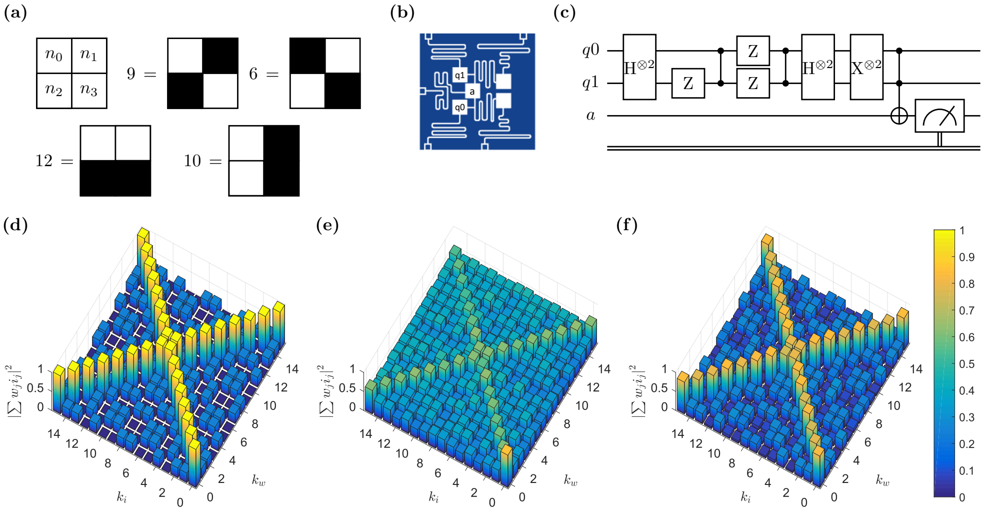


Figure 6: Results for 2-qubit quatum perceptron from [15]. (a) Scheme used to label the 2 × 2 patterns and a few examples. (b) Scheme of IBM Q-5 "Tenerife" backend quantum processor. (c) Example of the gate sequence for the 2-qubit case. (d) Ideal outcome of the quantum perceptron algorithm, simulated on a classical computer. (e) Results from the Tenerife processor using the algorithm with multi-controlled sign flip blocks. (f) Results from the Tenerife processor using the algorithm for the generation of hypergraph states.

top to bottom, and assigning a value $n_j = 0(1)$ to a white (black) pixel. Some example schemes include 6 and 9 as checkerboard-like pattern.

Fig. 6d shows the ideal outcome through simulation. The perceptron is able to recongnize different patterns for these 2 × 2 grids, e.g., vertical lines, horizontal lines, or checkerboard patterns.

Fig. 6e are the results of the non-optimized approach, while Fig. 6f are the improved results based on the hypergraph states formalism. The optimized version reach a very good quantitative agreement with the expected results plotted in Fig. 6d.

References

[1] Marcello Benedetti, Erika Lloyd, and Stefan Sack. Parameterized quantum circuits as machine learning models. *arXiv preprint arXiv:1906.07682*, 2019.

[2] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[3] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[4] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998.

[5] Sergio Boixo, Troels F Rønnow, Sergei V Isakov, Zhihui Wang, David Wecker, Daniel A Lidar, John M Martinis, and Matthias Troyer. Quantum annealing with more than one hundred qubits. *arXiv preprint arXiv:1304.4595*, 2013.

[6] Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv preprint arXiv:1602.07674*, 2016.

[7] Gavin E Crooks. Performance of the quantum approximate optimization algorithm on the maximum cut problem. *arXiv preprint arXiv:1811.08419*, 2018.

[8] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D Lukin. Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. *arXiv preprint arXiv:1812.01041*, 2018.

[9] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K Fährmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *arXiv preprint arXiv:1910.01155*, 2019.

[10] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

[11] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. Hybrid quantum-classical approach to quantum optimal control. *Physical review letters*, 118(15):150503, 2017.

[12] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[13] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Quantum circuit structure learning. *arXiv preprint arXiv:1905.09692*, 2019.

[14] Yudong Cao, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. Quantum neuron: an elementary building block for machine learning on quantum computers. *arXiv preprint arXiv:1711.11240*, 2017.

[15] Francesco Tacchino, Chiara Macchiavello, Dario Gerace, and Daniele Bajoni. An artificial neuron implemented on an actual quantum processor. *npj Quantum Information*, 5(1):26, 2019.

[16] E Torrontegui and JJ García-Ripoll. Unitary quantum perceptron as efficient universal approximator. *EPL (Europhysics Letters)*, 125(3):30004, 2019.

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[18] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2847–2854. JMLR. org, 2017.

[19] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. The expressive power of parameterized quantum circuits. *arXiv preprint arXiv:1810.11922*, 2018.

[20] George Cybenko. Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:183–192, 1989.

[21] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[22] Boris Hanin and Mark Sellke. Approximating continuous functions by relu nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

[23] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.

[24] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[25] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128, 2016.

[26] Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 4880–4888, 2016.

[27] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[28] William James Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and technology*, 2018.