Universality of neural networks describes the ability to approximate arbitrary function, and is a key ingredient to keep the method effective. The established models for universal quantum neural networks (QNN), however, require the preparation of multiple copies of the same quantum state to generate the nonlinearity, with the copy number increasing significantly for highly oscillating functions, resulting in a huge demand for a large-scale quantum processor. To address this problem, we propose a new QNN model that harbors universality without the need of multiple state-duplications, and is more likely to get implemented on near-term devices. To demonstrate the effectiveness, we compare our proposal with two popular QNN models in solving typical supervised learning problems. We find that our model requires significantly fewer qubits and it outperforms the other two in terms of accuracy and relative error.

1 Introduction

As an important subfield in machine learning (ML), neural networks (NNs), especially deep NNs, have generated a series of impactful results in many application scenarios [1–4]. One of the most striking features of NNs is their ability to learn the hidden patterns of a given data set and to make reliable predictions based on these patterns [5]. Such feature originates from the capability to approximate any continuous function, and is known as the universality. Most NNs proposed in literature are proved to be universal and such results are called universal approximation theorems [6, 7]. Due to the power of quantum computation, the idea of quantum machine learning (QML) is proposed to implement ML on quantum circuits, in order to achieve computational advantage compared to the classical counterparts. Such advantage has been shown for many QML algorithms [8], including quantum support vector machine [9], k-means clustering [10], quantum principle component analysis [11], quantum data-fitting algorithm [12] and quantum Boltzmann machine [13], but not for QNNs. It is shown that certain QNNs have a distinctive prediction advantage on certain designed data sets [14], but less is known for the general case. In fact, research on the advantage of QNNs is still in progress, partly due to the reason that even the complexity of classical NN algorithms has not been addressed without controversy.

Besides the quantum advantage issue, universality is also crucial to keep QNNs effective. The universality of classical NNs is determined by the nonlinearity of neurons. When it comes to the QNNs, how to generate nonlinearity is one of the biggest impediments to achieve the universal QNN. To address the problem, different QNN models have been proposed such as the continuous variable quantum neural network [15], the quantum neuron [16, 17], the circuit-centric quantum classifiers algorithm [18] and the quantum circuit learning algorithm [19], but not all of them have been rigorously proved to be universal. In some of these proposals, nonlinearity relies on using multiple copies of the quantum data, resulting in a rapid increase of the size of the quantum register. In order to solve this problem, in this work, we propose a duplication-free QNN structure which also guarantees the universality of the neural network.

In this work, we aim to construct a universal QNN model without the need of multiple duplications of quantum data. We design the duplication-free quantum neural network (DQNN) whose nonlinearity is generated by the
where \( \tilde{\omega} \) is determined by the target function with the data noise, \( y_i = f(x_i) + \epsilon_i \). The goal of the DQNN is to approximate the function \( f \) using \( D \). To achieve this goal, the DQNN uses the structure as shown in Fig. 1(a). It consists of three parts, a quantum processor (QP), a classical processor (CP) and a classical optimizer (CO). QP part contains some parameterized sigmoid function and a linear transformation. CO minimizes a loss function by using the gradient of the parameters in QP and CP.

Before implementing the loop of the three parts, we need to encode the classical data into quantum system using the amplitude encoding method [20]. We firstly find a continuous injection, \( F \), mapping \( x_i \) to the \( 2^n \)-dim quantum state Hilbert space \( \mathbb{C}^{2^n}_2 \) with \( d < 2^n \). If \( 0 \notin G \), we can transform the input \( x \) into \( |\tilde{x}\rangle \in \mathbb{S}_0 \subset \mathbb{C}^{2^n}_2 \) as

\[
F : x \in G \rightarrow |\tilde{x}\rangle \equiv \frac{1}{\gamma(x_1, \ldots, x_d, \tilde{x}, 0, \ldots, 0)^T}
\]

where \( \tilde{x} = \frac{|x|}{1+|x|^2} \) and \( \gamma = (|x|^2 + \tilde{x}^2)^{\frac{1}{2}} \); if \( 0 \in G \), we can perform a shift transformation, \( x \rightarrow x + \alpha \), such that \( 0 \notin G \) which is a ring domain \( \{x \in \mathbb{R}^d | 0 < \kappa_1 \leq |x| \leq \kappa_2 \} \). It is worthwhile to mention that \( 0 < \kappa_1 \leq |x| \leq \kappa_2 \) implies \( \frac{\kappa_1}{1+\kappa_2} \leq \tilde{x} \leq \frac{\kappa_2}{1+\kappa_1} \) and

\[
(1+(1+\kappa_2)^2)^{\frac{1}{2}} < \tilde{x}_{d+1} < (1+(1+\kappa_1)^2)^{\frac{1}{2}} \tag{2}
\]

with \( \tilde{x}_{d+1} = \tilde{x}_d \). After obtaining the new data set \( \{\tilde{x}_i, y_i\} \), we implement the loop of QP, CP and CO to approximate \( f \).

The specific structure of QP and CP are shown in Fig. 1(c). In this paper, the QP part uses a specific circuit ansatz which is presented in [18]. As shown in Fig. 1(b), the circuit ansatz represents the \( U(\theta) \) in Fig. 1(c). It contains \( n \) parameterized single-qubit gates, \( G(\theta_1, \theta_2, \theta_3) \), and \( n \) parameterized two-qubit control gates, \( CG(\theta_1, \theta_2, \theta_3) := |0\rangle \langle 0| \otimes I + |1\rangle \langle 1| \otimes G(\theta_1, \theta_2, \theta_3) \) where \( G \) is written as

\[
G(\theta_1, \theta_2, \theta_3) = \begin{pmatrix}
  e^{i\theta_2} \cos(\theta_1) & e^{i\theta_3} \sin(\theta_1) \\
  -e^{-i\theta_3} \sin(\theta_1) & e^{-i\theta_2} \cos(\theta_1)
\end{pmatrix}
\]

QP outputs the measurement results, \( \langle B_i \rangle = \text{Tr}(U(\theta)|\tilde{x}\rangle \langle \tilde{x}|U(\theta)^\dagger) \) with \( N \) observables. The number, \( N \), depends on the problem itself, and \( \{B_i\}_{i=1}^N \) is a subset of the Pauli basis \( \{P_i\}_{i=1}^4 \).

The CP part applies the parameterized sigmoid function \( \sigma^{(i)} \) to each of \( \langle B_i \rangle \) where the sigmoid function is defined as

\[
\sigma^{(i)}(\langle B_i \rangle) \equiv \frac{1}{1+\exp\{-a^{(i)}(\langle B_i \rangle) - c^{(i)}\}} \tag{4}
\]

with \( a^{(i)} > 2 \) and \( c^{(i)} \in [0, 1] \). Then CP feeds \( \{\sigma^{(i)}(\langle B_i \rangle)\}_{i=1}^N \) into a classical linear node and

2

Figure 1: (a) The framework of the DQNN. The measurement results of the quantum processor are the inputs of the classical processor. The parameters are updated by a classical optimizer. (b) The circuit ansatz used in the numerical simulation. (c) The circuit structure of the DQNN. The directed line represents the classical information, and the undirected line represents the quantum information.
Theorem 1. The universality of DQNN is guaranteed by the

\[ Q(\bar{x}) := \sum_{j=1}^{N} a_j \sigma(a_j(\langle B_j(\bar{x}, \theta) \rangle - c_j)), \]

where \( \alpha = \{a_j\} \) are trainable.

The final part of the DQNN, \( \text{CO} \), minimizes a loss function \( L(\theta, \alpha, a, c) := \sum_{k=1}^{m} \|Q(\bar{x}_k) - y_k\| \) by using some gradient-based methods such as SGD [21], ADAM [22] and BFGS [23], and obtains the optimal parameters, \( (\theta^*, \alpha^*, a^*, c^*) = \arg \min L(\theta, \alpha, a, c) \). The gradient of each parameter is analytically given below:

\[
\begin{align*}
\frac{Q(\bar{x})}{\partial \theta_j} &= \frac{1}{2} \sum_i \alpha_i a_i \sigma^{(i)}(1 - \sigma^{(i)})(\langle B_i \rangle_j^+ - \langle B_i \rangle_j^-), \\
\frac{Q(\bar{x})}{\partial a_j} &= \alpha_j \sigma^{(j)}(1 - \sigma^{(j)})(\langle B_j \rangle - c_j), \\
\frac{Q(\bar{x})}{\partial c_j} &= -\alpha_j \sigma^{(j)}(1 - \sigma^{(j)}) a_j, \\
\frac{Q(\bar{x})}{\partial \alpha_j} &= \sigma^{(j)}.
\end{align*}
\]

where \( \sigma^{(i)} \) represents \( \sigma(a_i(\langle B_i(\bar{x}, \theta) \rangle - c_i)) \) and \( \langle B_i \rangle_j^+ \) and \( \langle B_i \rangle_j^- \) denotes the expectation value \( \langle B_i \rangle \) inserting \( \pm \frac{\pi}{2} \) into the \( j \)-th quantum parameter \( \theta_j \) according to the parameter-shift rule [19].

The basic idea to design such a structure is whenever the QNN is designed by using the variational quantum circuit, the classical part containing in it can generate and enhance the nonlinearity of the hybrid system and further decrease the number of required qubits to achieve the quantum neural network. In this way, the universality can be proved and the complexity can be reduced.

3 Universality of DQNN

The universality of DQNN is guaranteed by the following theorem:

\[ \int_{\mathcal{S}_0} \left| \sum_{i=1}^{N} \alpha_i \sigma(a_i(\langle B_i(\bar{x}, \theta) \rangle - c_i) - f(\bar{x}) \right|^2 d\mu(\bar{x}) < \epsilon. \]

Proof. Denoting the quantum circuit of the DQNN in Fig. 1(c) as \( U(\theta) \), which maps the quantum data \( |\bar{x}\rangle \) into \( |\bar{x}\rangle = U(\theta)|\bar{x}\rangle \), the output \( y \) of DQNN is derived through measuring a set of observables \( \{B_i\}_{i=1}^{N} \) on the final state \( |\bar{x}\rangle \). Based on measurement statistics, \( y = \sum_{i=1}^{N} \alpha_i \sigma(a_i((\langle B_i(\bar{x}, \theta) \rangle - c_i)) \) is found according to:

\[ \langle B_i(\bar{x}, \theta) \rangle = \langle \bar{x}|B_i|\bar{x}\rangle = \sum_{j} \lambda_{i,j}|\bar{x}||\xi_{i,j}\rangle^2 \]

where \( \lambda_{i,j} \) and \( |b_{i,j}\rangle \) are the eigenvalues and the eigenvectors of \( B_i \), and \( |\xi_{i,j}\rangle \equiv U(\theta)|b_{i,j}\rangle \). In the following, we make two assumptions: the number of observables \( N \) is sufficiently large, and the circuit ansatz, represented as \( U(\theta) \), should have enough expressibility to approximate any arbitrary unitary evolution. The former is guaranteed if the local Pauli operators on each qubit can be measured, and the latter is guaranteed if the set of physically-implementable gates form a universal gate set. Without loss of generality, we will restrict ourselves to the case of \( \lambda_{i,1} = 1 \) and \( \lambda_{i,j} = 0 \) (\( j \neq 1 \)) where \( \langle B_i \rangle = |\langle \bar{x}|\xi_i\rangle|^2 \). The output of the DQNN is the function

\[ q(\bar{x}) \alpha, a, \xi_1, \ldots, \xi_N, c = \sum_{i=1}^{N} \alpha_i \sigma(a_i(\langle |\bar{x}|\xi_i\rangle^2 - c_i)). \]

Denoted by \( Q(S_0) \), the function space is composed of the finite linear combination of the sigmoid-type functions:

\[ Q(S_0) = \{q(\bar{x}) \alpha, a, \xi_1, \ldots, \xi_N, c : N \in \mathbb{N}, \alpha \in \mathbb{R}^N, \]

\[ a \in \mathbb{R}_+^N, c \in [0, 1]^N, \{\xi_i\}_{i=1}^{N} \subset S_{CN} \}. \]

To prove the universality, it suffices to verify that \( Q(S_0) \) is dense in \( L^2(S_0) \). We assume that the closure \( \overline{Q(S_0)} \neq L^2(S_0) \). The contradiction is shown in supplemental material. □

One can see that the two above assumptions are crucial for the validity of the proof. In order to satisfy them, the circuit can become very long and is required to be repeated many times to derive the measurement outcomes for all \( B_i \). In theory, the classical NN requires infinite neurons in one layer to be universal [6]. However, in practice, the classical case only uses a finite number of neurons and has obtained excellent results in various aspects. Analogously, for a given
data set, a chosen DQNN structure, a finite number of observables in DQNN are often sufficient to approximate the target function well, and this will be demonstrated in the following examples. In addition, the length of the DQNN circuit depends on the given data set, and it may become very long for a special data set. In fact, it is an important unsolved question on how exactly the complexity of the DQNN circuit depends on the data set.

4 Circuit complexity for DQNN

The advantage of DQNN is introducing the classical sigmoid function to generate the nonlinearity and significantly reduces the circuit complexity compared with two duplication-based QNNs, quantum circuit learning(QCL) algorithm [19] and circuit-centric quantum classifiers(CCQ) algorithm [18]. The circuit complexity to implement the QNNs is given as $C := O(n_g n_b)$ where $n_g, n_b$ respectively denote the number of quantum gates and the number of measurement observables. Without loss of generality, we assume the number of gates is polynomial to the number of required qubits.

To show the difference of complexity among QNN, QCL and CCQ, we concentrate on a polynomial function approximation problem whose goal is approximating an $M$-order polynomial function of $x \in \mathbb{R}^d$. The CCQ stores $x$ into the amplitudes of data qubits as $|x\rangle = \frac{1}{||x||}\sum_{i=1}^{d} x_i |i\rangle$ and needs $O(M)$ copies of $|x\rangle$. After applying $n_g = O(poly(M|\log d|))$ gates, $O(1)$ POVM operators is used to measure the system. Its complexity is $C_{CCQ} = O(poly(M|\log d|))$. In the meanwhile, the QCL encodes $x$ as $\rho \otimes M |x\rangle = \frac{1}{2^d} \sum_{l=1}^{d} \left( \otimes_{k=1}^{M} \left[ I + x_i \sigma_x^{(k)} + \sqrt{1 - x_i^2} \sigma_z^{(k)} \right] \right)$ using $M$ copies of the data qubits $\rho(x)$. QCL uses $O(1)$ Pauli operators to measure the circuit. Therefore, its complexity is $C_{QCL} = O(poly(Md))$. Because the classical sigmoid function makes DQNN need no duplication and the encoding method is the same as CCQ, the DQNN only needs $n_g = O(poly(|\log d|))$ gates. Moreover, the number of observables is independent to the number of required qubits but a hyperparameter determined by the problem. The complexity of DQNN is $C_{DQNN} = O(poly(|\log d|))$. The comparison is summarized in Table. 1. It can be seen that DQNN efficiently reduces the number of required qubits compared with CCQ and QCL.

| Algorithm | # Duplication | # Data qubits |
|-----------|---------------|---------------|
| QCL       | $M$           | $d$           |
| CCQ       | $M$           | $\lceil \log d \rceil$ |
| DQNN      | 1             | $\lceil \log d \rceil$ |

Table 1: The number of required qubits among three proposals to approximate an $M$-order polynomial function of $x \in \mathbb{R}^d$.

5 Applications

We design two data sets, regression and classification data sets, to show the advantage of the DQNN compared with QCL and CCQ. The regression data set (Fig. 2(a)) contains 400 data samples which are randomly generated by $f = (0.715625 - 1.0125x_1^2 + x_1^4)(0.715625 - 1.0125x_2^2 + x_2^4)$ with $x_1, x_2 \in [-0.8, 0.8]$. The classification data set (Fig. 2(b)) has 800 data samples where the boundaries are generated by $x_1^2 + x_2^2 = 0.16$ and $x_1^2 + x_2^2 = 0.81$ with $x_1, x_2 \in [-1, 1]$. In the donut-like area, the data samples are labeled $y^{(i)} = [1, 0]^T$ and others are $y^{(i)} = [0, 1]^T$. We use different numbers of duplications and layers in QCL and CCQ. The DQNN uses 5 and 10 randomly generated observables respectively in the regression and classification task. The simulation results (Table. 2) show that the performances of DQNN are better than other proposals which complexity is around 5 times larger both in regression and classification. With the increasing of the number of copies and layers, CCQ in the classification task shows underfitting which leads to a lower accuracy.

Figure 2: (a) The regression data set with a polynomial function. (b) The classification data set with two circular decision boundaries.
Table 2: The running result and specific setup of DQNN, QCL and QCL. $c$ is a specific value of the circuit complexity which is calculated by $c = n_g \times n_b$.

| Task             | n | $N_{train}$ | $N_{test}$ | $\epsilon_{Train}$ | $\epsilon_{Test}$ | $\epsilon_{Test}$ |
|------------------|---|-------------|------------|--------------------|--------------------|-------------------|
| MNIST$_2$        | 8 | 12665       | 2115       | 0.0047             | 0.0009             |                  |
| MNIST$_3$        | 8 | 18623       | 3147       | 0.0172             | 0.0114             |                  |
| Wine             | 4 | 143         | 35         | 0.0000             | 0.0286             |                  |
| Breast Cancer    | 4 | 560         | 39         | 0.0143             | 0.0432             |                  |

Additionally, in order to compare with the classical counterpart, we implement the classical neural network (NN) based on above two tasks to compare the performance with QNN. The results show that with similar number of parameters, the DQNN has the same power as its classical analogue. In the regression task, the training process (Fig. 3(a)) shows that both proposals have the similar rate of convergence and the mean relative error on NN is 4.19% which is similar with DQNN. As for the classification task, the training process with the accuracy shown in Fig. 3(b) demonstrates that the neural network with the similar number of parameters achieves 74.5% (NNv2). Meanwhile, the classical neural network which contains three times more parameters than DQNN can achieve the similar accuracy as DQNN.

To verify the power of DQNN on the real-world data sets, we further implement some classification tasks. Firstly, on the handwritten digits data set, MNIST [24], we choose 0 and 1 for a binary classification and 0, 1, 2 for a multi-target classification on MNIST. Each picture is reshaped into $16 \times 16$ and encoded into 8 qubits. Additionally, on the Wine and Breast Cancer data sets [25], we randomly divide the data sets into five equivalent parts and pick one of them as the test set. The result (Table 3) shows the DQNN has the ability to find the complexity pattern hidden in the real-world data sets.

Besides the classical tasks above, DQNN provides the ability to investigate the intrinsic property of quantum mechanics such as the quantum phase recognition (QPR). Specifically, we apply the DQNN to the $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry-protected topological (SPT) phase discrimination task [26]. The ground states of a parameterized spin–1/2 chain Hamiltonian,

$$H = -J \sum_{i=1}^{N-2} \sigma_x^{(i)} \sigma_x^{(i+1)} \sigma_x^{(i+2)} - h_1 \sum_{i=1}^{N} \sigma_x^{(i)}$$

where $h_1, h_2$ and $J$ are parameters, corresponds to the different topological phases. The phase diagram of the Hamiltonian is given in Fig. 3(c).
The train data takes 400 equally spaced points from $h_1 \in [0, 1.6]$ and $h_2 \in [-1.6, 1.6]$. And the test data contains 4096 equally spaced points. The ground state corresponding to each point is labeled $[1, 0]^T$ if it belongs to the SPT phase. Otherwise it is labeled $[0, 1]^T$. We numerically implement the DQNN with 15 qubits, 420 parameters and 10 observables. The accuracy on the test data achieves 99.10%. It shows the DQNN could find the relation between the ground states of the Hamiltonian and their corresponding phase.

![Figure 3](image.png)

Figure 3: (a) The MSE loss with training episodes of QNN and classical neural network. (b) The classification accuracy with training episodes of QNN and classical neural network. (c) The phase diagram of the spin−1/2 chain. The phase boundary is generated by the 2-degree polynomial regression based on some boundary points.

6 Conclusion

In this article, we present the universal duplication-free quantum neural network whose nonlinearity is generated by the classical sigmoid function. The simulation results show that the DQNN significantly reduces the number of required qubits to complete the supervised learning tasks compared with previous work and has the ability to recognize the SPT phase of a spin-1/2 chain Hamiltonian. However, how to design an appropriate circuit ansatz for a certain problem still remains an open question. Besides the scenarios discussed in this work, we expect the duplication-free quantum neural network has broad applications in other area, including natural language processing, computer version and reinforcement learning.

7 Acknowledgement

The authors gratefully acknowledge the grant from National Key R&D Program of China, Grant No. 2018YFA0306703. We also thank Chu Guo, Bujiao Wu, Yuesen Wu, Shaojun Wu, Yuhan Huang, Dingding Wen and Yi Tian for helpful discussions.

References

[1] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, volume 1, pages 655–665. Association for Computational Linguistics, 2014. DOI: https://doi.org/10.3115/v1/P14-1062.

[2] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur. Extensions of recurrent neural network language model. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5528–5531, 2011. DOI: 10.1109/ICASSP.2011.5947611.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016. URL https://arxiv.org/abs/1604.07316.

[4] Christopher M Bishop. Pattern recognition and machine learning. springer, 2006. DOI: https://doi.org/10.1007/978-1-4615-7566-5.

[5] Goodfellow Ian, Bengio Yoshua, and Courville Aaron. Deep learning, volume 1. MIT Press, 2016. URL https://books.google.co.in/books?id=Np9SDKAAQBAJ.

[6] G. Cybenko. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems, 2(4):303–314, 1989. DOI: https://doi.org/10.1007/BF02551274.
[7] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991. DOI: https://doi.org/10.1016/0893-6080(91)90009-T.

[8] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017. DOI: https://doi.org/10.1038/nature23474.

[9] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014. DOI: 10.1103/PhysRevLett.113.130503.

[10] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*, 2013. URL https://arxiv.org/abs/1307.0411.

[11] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014. DOI: https://doi.org/10.1038/nphys3029.

[12] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical review letters*, 109(5):050505, 2012. DOI: 10.1103/PhysRevLett.109.050505.

[13] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. *Phys. Rev. X*, 8:021050, May 2018. DOI: 10.1103/PhysRevX.8.021050.

[14] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1), 2021. ISSN 2041-1723. DOI: https://doi.org/10.1038/s41467-021-22539-9.

[15] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Physical Review Research*, 1(3):033063, 2019. DOI: 10.1103/PhysRevResearch.1.033063.

[16] Dan Ventura and Tony Martinez. An artificial neuron with quantum mechanical properties. In *Artificial Neural Nets and Genetic Algorithms*, pages 482–485. Springer, 1998. DOI: https://doi.org/10.1038/s41534-019-0140-4.

[17] 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. URL https://arxiv.org/abs/1711.11240.

[18] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020. DOI: 10.1103/PhysRevA.101.032308.

[19] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018. DOI: https://doi.org/10.1103/PhysRevA.98.032309.

[20] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, 83:032302, Mar 2011. DOI: 10.1103/PhysRevA.83.032302.

[21] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of COMPSTAT’2010*, pages 177–186. Physica-Verlag HD, 2010. DOI: 10.1007/978-3-7908-2604-3_16.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. DOI: 10.1080/10556788.2019.1649672.

[23] Albert Buckley and A LeNir. Algorithm 630: Bbvscg—a variable-storage algorithm for function minimization. *ACM Transactions on Mathematical Software (TOMS)*, 11(2):103–119, 1985. DOI: 10.1145/363219.363231.

[24] Yann LeCun and Corinna Cortes. Mnist handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[25] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[26] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural net-
works. *Nature Physics*, 15(12):1273–1278, 2019. DOI: 10.1038/s41567-019-0648-8.

[27] Hans Hahn. Über lineare gleichungssysteme in linearen räumen. *Journal für die reine und angewandte Mathematik*, 1927(157):214–229, 1927. DOI: https://doi.org/10.1515/crll.1927.157.214.

[28] Maurice Fréchet. Sur les ensembles de fonctions et les opérations linéaires. *CR Acad. Sci. Paris*, 144:1414–1416, 1907. DOI: https://doi.org/10.1090/S0002-9947-1904-1500687-0.
A The proof of Theorem 1

It suffices to show that $Q(S_0)$ is dense in $L^2(S_0)$. We assume that the closure $\overline{Q(S_0)} \neq L^2(S_0)$ and show the contradiction. By the Hahn-Banach theorem [27], there exists a bounded linear functional $L$ of $L^2(S_0)$ such that $L(Q(S_0)) = 0$ and $L \neq 0$. By the Riesz representation theorem [28], there exists a function $g(\bar{x}) \in L^2(S_0)$ such that

$$L(f) = \int_{S_0} f(\bar{x})g(\bar{x})d\mu(\bar{x}) \quad \text{for all} \ f \in L^2(S_0)$$

where $L \neq 0$ implies that $g(\bar{x}) \neq 0$. Since $L(Q(S_0)) = 0$, we have

$$\int_{S_0} \sigma(a(|\langle \bar{x}|\xi \rangle|^2 - c))g(\bar{x})d\mu(\bar{x}) = 0$$

(9)

In particular, there exists an open subset $E \subset S_0$ with the measure $\mu(E) > 0$ such that $g(\bar{x}) \neq 0$ in $E$. Without loss of generality, we assume $g(\bar{x}) \geq k > 0$ in $E$. Since $E$ is open, there exists a small ball $B(\xi^*, \delta) = \{\bar{x} : |\bar{x} - \xi^*| < \delta\} \subset E$.

For $\xi^*, \bar{x} \in \mathbb{R}^N$, we have

$$|\bar{x} - \xi^*|^2 = 2 - 2\langle \bar{x}|\xi^* \rangle.$$ (10)

Therefore $|\langle \bar{x}|\xi^* \rangle|^2 > c$ is equivalent to

$$|\bar{x} - \xi^*|^2 < 2(1 - \sqrt{c}) \quad \text{or} \quad |\bar{x} - \xi^*|^2 > 2(1 + \sqrt{c}).$$ (11)

We claim that $|\bar{x} - \xi^*|^2 > 2(1 + \sqrt{c})$ is impossible to hold if $c$ is closed to 1. Or else, by using

$$|\bar{x} - \xi^*|^2 + |\bar{x} + \xi^*|^2 = 4$$

(12)

we find that $|\bar{x} + \xi^*|^2 < 2(1 - \sqrt{c})$. From Eqn. (2) and $\xi^*, \bar{x} \in S_0$, we see that $|\bar{x} + \xi^*|^2 \geq (x_{d+1} + \bar{x}_{d+1})^2 \geq 4(1 + (1 + \kappa_2)^2)^{-1}$. Hence for $c > (1 - 2(1 + (1 + \kappa_2)^2)^{-1})^2$, the latter case of Eqn. (11) makes no sense, and $|\langle \bar{x}|\xi^* \rangle|^2 > c$ is only equivalent to

$$|\bar{x} - \xi^*|^2 < 2(1 - \sqrt{c}) \quad \forall \xi^*, \bar{x} \in S_0.$$ (13)

Therefore, passing to the limit $a \to \infty$ we obtain

$$\sigma(a(|\langle \bar{x}|\xi^* \rangle|^2 - c)) \rightarrow \begin{cases} 1 & \forall \bar{x} \in B(\xi^*, \delta_1) \\ 0 & \forall \bar{x} \notin B(\xi^*, \delta_1) \end{cases}$$

(14)

with $\delta_1 = (2(1 - \sqrt{c}))^2$. By taking $c$ sufficiently close to 1 such that $\delta_1 \leq \delta$, and using Lebesgue dominate convergence theorem, from Eqn.(B13) we obtain

$$0 = \int_{S_0} \sigma(a(|\langle \bar{x}|\xi \rangle|^2 - c))g(\bar{x})d\mu(\bar{x})$$

(15)

$$\geq k \int_{B(\xi^*, \delta_1)} \sigma(a(|\langle \bar{x}|\xi \rangle|^2 - c))d\mu(\bar{x})$$

(16)

$$\rightarrow k \int_{B(\xi^*, \delta_1)} 1d\mu(\bar{x}) = k\mu(B(\xi^*, \delta_1)) > 0$$

(17)

which comes out a contradiction. Hence, we conclude that $Q(S_0)$ is dense in $L^2(S_0)$. Thus for any $f \in L^2(S_0)$ and $\epsilon > 0$, we can find a $q(\bar{x}) \in Q(S_0)$ such that

$$\|f - q(\bar{x})\|^2_{L^2(S_0)} = \int_{S_0} |f(\bar{x}) - q(\bar{x})|^2 d\mu(\bar{x}) \leq \epsilon$$

(18)

which proves the theorem.