Hybrid Quantum-Classical Neural Networks

Davis Arthur
University of Florida
Gainesville, Georgia, USA
Email: davisarthur@ufl.edu

Prasanna Date
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
Email: datepa@ornl.gov

Abstract—Deep learning is one of the most successful and far-reaching strategies used in machine learning today. However, the scale and utility of neural networks is still greatly limited by the current hardware used to train them. These concerns have become increasingly pressing as conventional computers are soon expected to approach the physical limitations that will slow their performance improvements in the near future. For these reasons, scientists have begun to explore alternative computing platforms, like quantum computers, for training neural networks. In recent years, variational quantum circuits have emerged as one of the most successful approaches to quantum deep learning on noisy intermediate scale quantum devices. We propose a hybrid quantum-classical neural network architecture where each neuron is a variational quantum circuit. We empirically analyze the performance of this hybrid neural network on a series of binary classification data sets using a simulated IBM universal quantum computer and a state-of-the-art IBM universal quantum computer. On the simulated hardware, we observe that the hybrid neural network achieves around 10% higher classification accuracy and 20% better minimization of the cost function than an individual variational quantum circuit. On the quantum hardware, we observe that each model only performs well when the qubit and gate count is sufficiently small.

I. INTRODUCTION

Machine learning has revolutionized the modern world. Today, machine learning models are leveraged for nearly every imaginable task ranging from medical diagnoses [1] to fraud detection [2] to marketing [3]. This widespread use of machine learning is largely due to the recent accessibility of relatively powerful computers. In accordance with the Moore's Law, computer hardware has improved exponentially in scale and speed over the past 60 years. Unfortunately, despite all of the recent success, modern hardware still greatly restricts the practicality of certain machine learning models. Deep learning in particular can be very computationally expensive, sometimes requiring hours, days, or even months of training time on today's computers [4]. Moreover, conventional computers are beginning to approach physical limitations that will slow their improvements in years to come [5]. For these reasons, many are beginning to research alternative computing platforms for training machine learning models. Among these platforms, quantum computers have emerged as a particularly interesting candidate.

The appeal of a quantum computer is largely due to the properties of quantum entanglement and quantum superposition which cannot be efficiently simulated on a classical computer. These properties can be extremely useful as illustrated by Shor's prime factorization algorithm [6] and

Grover's search algorithm [7] which offer an exponential and polynomial speed up respectively over their best existing classical counterparts. These two algorithms give a sense of what large, high-fidelity quantum computers may offer to the field of computer science in years to come. Today we are still in the era of noisy intermediate scale quantum (NISQ) computers, but already a number of quantum machine learning algorithms have been proposed [8], [9]. We have previously studied quantum approaches to linear regression [10], support vector machines [11], and balanced k-means clustering [12]. Additionally, we proposed a quantum machine learning model called quantum discriminator that can be used for binary classification on universal quantum computers [13]. In this paper, we propose a hybrid quantum-classical neural network architecture and emperically analyze its performance on several binary classification data sets.

Artificial neural networks have proven to be an extremely successful approach to learning on conventional computers. Nevertheless, they do have some serious limitations. Neural networks are often prone to overfitting [14], and training even an extremely small neural network is an NP-complete problem [15]. These limitations have inspired many to propose quantum approaches to deep learning [16], [17], [18], [19], [20], [21], [22]. Unfortunately, many of these approaches cannot be implemented on current quantum hardware, and those that can, do not have a one-to-one correspondence with conventional artificial neural networks. For this reason, there is an increased interest in deep learning approaches that perform well on near-term quantum computers. To this end, variational quantum circuits (VQCs) have proven to be a promising quantum analogue to artificial neurons [23], [24], [25], [26], [27], [28], [29].

A variational quantum circuit is comprised of three key components. First, a feature map F maps a real-valued classical data point $\boldsymbol{x}$ into a d qubit quantum state $|\psi\rangle$:

$$|\psi(\boldsymbol{x})\rangle = F(\boldsymbol{x}) |0\rangle^{\otimes d} \qquad (1)$$

Next, an ansatz A manipulates the prepared quantum state through a series of entanglements and rotation gates. The angles of the ansatz's rotations are parameterized by a vector $\boldsymbol{\theta}$.

$$|\phi(\boldsymbol{x}, \boldsymbol{\theta})\rangle = A(\boldsymbol{\theta}) |\psi(\boldsymbol{x})\rangle \qquad (2)$$

Finally, an observable O is measured, and the eigenvalue corresponding to the resultant quantum state is recorded. In

most machine learning applications, a variational quantum circuit is run many times using a particular input $\boldsymbol{x}$ and parameter vector $\boldsymbol{\theta}$ so that the circuit's expectation value, denoted by f, can be approximated.

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \langle \phi(\boldsymbol{x}, \boldsymbol{\theta}) | \, O \, | \phi(\boldsymbol{x}, \boldsymbol{\theta}) \rangle \qquad (3)$$

When a variational quantum circuit is used for machine learning, this approximated expectation value is typically treated as the output of the model.
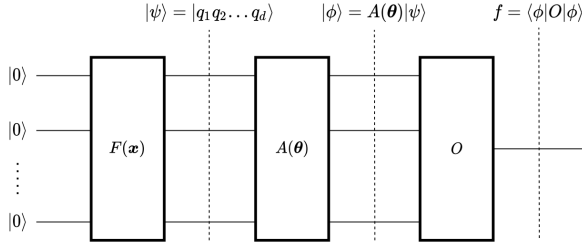

Fig. 1. Schematic of a variational quantum circuit.

The feature map of a variational quantum circuit is known to play a key role in the expressiveness of the model [30]. In general, data should be encoded in such a way that the value of a feature can be extracted from the prepared quantum state through some combination of qubit rotations and measurements. This ensures each possible input $\boldsymbol{x}$ has a unique qubit encoding before being passed to the ansatz. On modern hardware, it is also important to use a feature map with limited depth, since each additional gate introduces noise to the quantum state. We satisfy both requirements by scaling each feature x_i to fit within the interval $[0, \pi]$ and then encoding its value into the relative amplitude of a corresponding qubit:

$$|q_i\rangle = \cos\left(\frac{x_i}{2}\right)|0\rangle + \sin\left(\frac{x_i}{2}\right)|1\rangle \qquad (4)$$

It is worth noting that more sophisticated feature maps exist [31], [32], [33]. However, for the data sets analyzed in this manuscript, this straightforward feature map achieves high accuracy while avoiding many of the complications introduced by more complex methods.

The IBM Qiskit circuit library contains several ansatzes consisting of two qubit entanglements and parameterized single qubit rotations. We chose the *RealAmplitudes* ansatz (with one repetition and full entanglement) for each variational quantum circuit studied in Section II. This is the default ansatz used by Qiskit's variational quantum circuit implementation (*TwoLayerQNN*). It has also been used in a variational quantum circuit with proven advantages over traditional feedforward neural networks in terms of both capacity and trainability [24]. We also chose the default observable used by Qiskit's variational quantum circuit implementation. Mathematically, this observable can be described as the tensor product of d

Pauli-Z matrices (σ_z), where d is the number of qubits in the quantum state:

$$O = \sigma_z^{\otimes d} \qquad (5)$$

This observable has the interesting property that if the measured quantum state has odd parity, the recorded eigenvalue is -1, and if the measured quantum state has even parity, the recorded eigenvalue is 1. This means that the expectation value of the circuit will always be within the interval $[-1, 1]$.

A number of studies have used a variational quantum circuit for binary classification [34], [35], [36], [37], [38]. This can be done by relating the expectation value of the circuit to the probability that a point belongs to a given class. Consider a binary classification problem in which each data point $\boldsymbol{x}$ is labeled $y = 1$ or $y = -1$. We use the following equation to relate the expectation value of the parity observable to the probability that a point $\boldsymbol{x}$ is labelled y:

$$P(y|\boldsymbol{x}) = \frac{yf(\boldsymbol{x}, \boldsymbol{\theta}) + 1}{2} \qquad (6)$$

Training the variational quantum circuit classifier amounts to determining a parameter vector $\boldsymbol{\theta}$ that minimizes the negative log-likelihood of the probability distribution over the training data set. The exact cost function used by our binary classifier is given by the equation below:

$$\text{Cost} = -\frac{1}{N} \sum_{i=1}^{N} \log\left(P(y_i|\boldsymbol{x}_i)\right) \qquad (7)$$

where N is the number of points in the training data set, $\boldsymbol{x}_i$ is the ith data point in the training set, and y_i is the label of the ith point.

Cost can be minimized using a classical optimizer such as gradient descent. When computing the gradient of the cost function, the derivative of the expectation value of the variational quantum circuit with respect to each parameter of the ansatz is computed using parameter shift rule [39]:

$$\frac{\mathrm{d}f}{\mathrm{d}\theta_i} = \frac{f(\theta_i + s) - f(\theta_i - s)}{2} \qquad (8)$$

where s is a macroscopic shift determined by the eigenvalues of the gate parameterized by θ_i. For all of the rotation and phase gates available in the IBM Qiskit library, $s = \pi/2$.

In many ways, the aforementioned variational quantum circuit classifier resembles a logistic unit used in a conventional neural network. The circuit has an input vector $\boldsymbol{x}$ and a set of classically optimizable parameters $\boldsymbol{\theta}$. Additionally, the output (expected value) of the variational quantum circuit is continuously differentiable and bound to a small range of real values. These similarities motivated us to construct a small hybrid quantum-classical feedforward neural network using variational quantum circuits as individual neurons. To achieve reasonable training times on modern quantum hardware, we restricted the neural network architecture to contain only a single hidden layer and a single output unit.
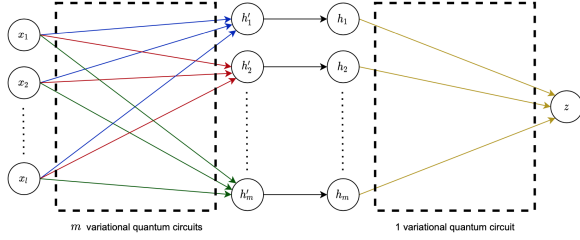
Fig. 2. Architecture of a single hidden layer quantum neural network. The inputs and corresponding output of a particular variational quantum circuit are connected by arrows of the same color. The input data point $\boldsymbol{x}$ has dimension l. The output of the entire network is denoted by z.

Using the architecture shown in Figure 2, our hybrid neural network contains $m+1$ variational quantum circuits. The first m circuits comprise the hidden layer of the feedforward network. Each of these circuits has its own parameter vector $\boldsymbol{\theta}_i^{(1)}$, and they all share the same input $\boldsymbol{x}$. The output of each circuit in the hidden-layer is stacked to create an m dimensional vector $\boldsymbol{h}' \in [-1, 1]^m$. Collectively, we denote the m circuits of the hidden layer as a function $f_1 : [0, \pi]^d \to [-1, 1]^m$.

$$\boldsymbol{h}' = f_1(\boldsymbol{x}, \boldsymbol{\theta}^{(1)}) \tag{9}$$

Before $\boldsymbol{h}'$ is passed to the feature map of the final variational quantum circuit, a transformation is applied so that each value is within the interval $[0, \pi]$:

$$\boldsymbol{h} = \frac{\pi}{2}(\boldsymbol{h}' + 1) \tag{10}$$

The last variational quantum circuit is run with input $\boldsymbol{h}$ and parameter vector $\boldsymbol{\theta}^{(2)}$. We denote this quantum circuit as a function $f_2 : [0, \pi]^m \to [-1, 1]$. All together, the hybrid neural network is expressed by the following composite function:

$$f_{\text{NN}}(\boldsymbol{x}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) = f_2\left(\frac{1}{2}(f_1(\boldsymbol{x}, \boldsymbol{\theta}^{(1)}) + 1), \boldsymbol{\theta}^{(2)}\right) \tag{11}$$

The output of the hybrid neural network can be used to learn the probability distribution that a point $\boldsymbol{x}$ is labeled y using the same method described by Equation 6:

$$P(y|\boldsymbol{x}) = \frac{y f_{\text{NN}}(\boldsymbol{x}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) + 1}{2} \tag{12}$$

Training the hybrid neural network on binary classification problems is similar to training the individual variational quantum circuit classifier. We minimize the same cost function (given by Equation 7), and we still use parameter shift rule to compute the gradient of each circuit. However, now we must compute the gradient of the cost function with respect to the output layer's parameter vector $\boldsymbol{\theta}^{(2)}$ as well each parameter vector in the hidden layer $\boldsymbol{\theta}_i^{(1)}$. This is performed most efficiently using backpropagation.

II. RESULTS

We tested the hybrid neural network on three binary classification data sets. As a point of comparison, we also trained an individual variational quantum circuit classifier on each of these data sets. We trained both models on a simulated IBM universal quantum computer and a state-of-the-art IBM universal quantum computer. To achieve reasonable training times, we restricted the hybrid neural network to use only $m = 2$ hidden neurons. On all three data sets, 10 simulated quantum trials and 3 actual quantum trials were performed for both quantum models. In each trial, all ansatz parameters were randomly initialized using a uniform distribution with a range of $[-\pi, \pi]$.

A. Bars and Stripes

Bars and Stripes is a synthetic data set of $n \times m$ binary black and white images. Each image in the data set is either a "bar" or a "stripe." A "bar" has 1 to $m - 1$ horizontal rows highlighted in black, and a "stripe" has 1 to $n - 1$ vertical columns highlighted in black. In some variations of Bars and Stripes, an entirely white image and an entirely black image is also included. We do not include these two images since their classification is ambiguous. Overall, the data set contains $N = 2^n + 2^m - 4$ images. Of these images, $N_s = 2^n - 2$ are stripes and $N_b = 2^m - 2$ are bars. We trained the variational quantum circuit classifier and hybrid quantum classical neural network on the 2×2 Bars and Stripes data set using 20 epochs of batch gradient descent with a learning rate of 0.5. All 4 points in the data set were used for training. The results of the simulated trials are reported in Table I and Figure 3. The results of the quantum trials are reported in Table I.

B. Real Valued Data

We also trained the individual variational quantum circuit and hybrid quantum-classical neural network on two real-valued data sets. The first data set is a two dimensional, linearly separable data set generated using Sci-Kit Learn's *make_blobs()* function. The second data set is a subset of the Iris benchmark data set consisting of all points belonging to the versicolor and virginica classes, which are non-linearly separable and four dimensional. Both data sets consist of 100 data points split evenly between each class. In each experiment, 80 of the 100 data points were chosen at random to be used for training. Training consisted of 10 epochs of mini-batch gradient descent using a batch size of 16 points and a learning rate of 0.1. The results of the simulated trials are reported in Table I and Figure 3. The results of the quantum trials are reported in Table I.

On the two dimensional data set, it is possible to visualize the classification line and probability distribution learned by each quantum model. In Figure 4, we have plotted this information for one of the variational quantum circuit trials and one of the hybrid neural network trials. The two examples chosen were selected because their final accuracy and final cost value were reflective of other trials of the same model type. Additionally, both examples had roughly 50% accuracy before training.

III. CONCLUSION

On simulated hardware, the hybrid quantum-classical neural network always outperformed the individual variational quantum circuit in terms of both accuracy and cost. Specifically, the
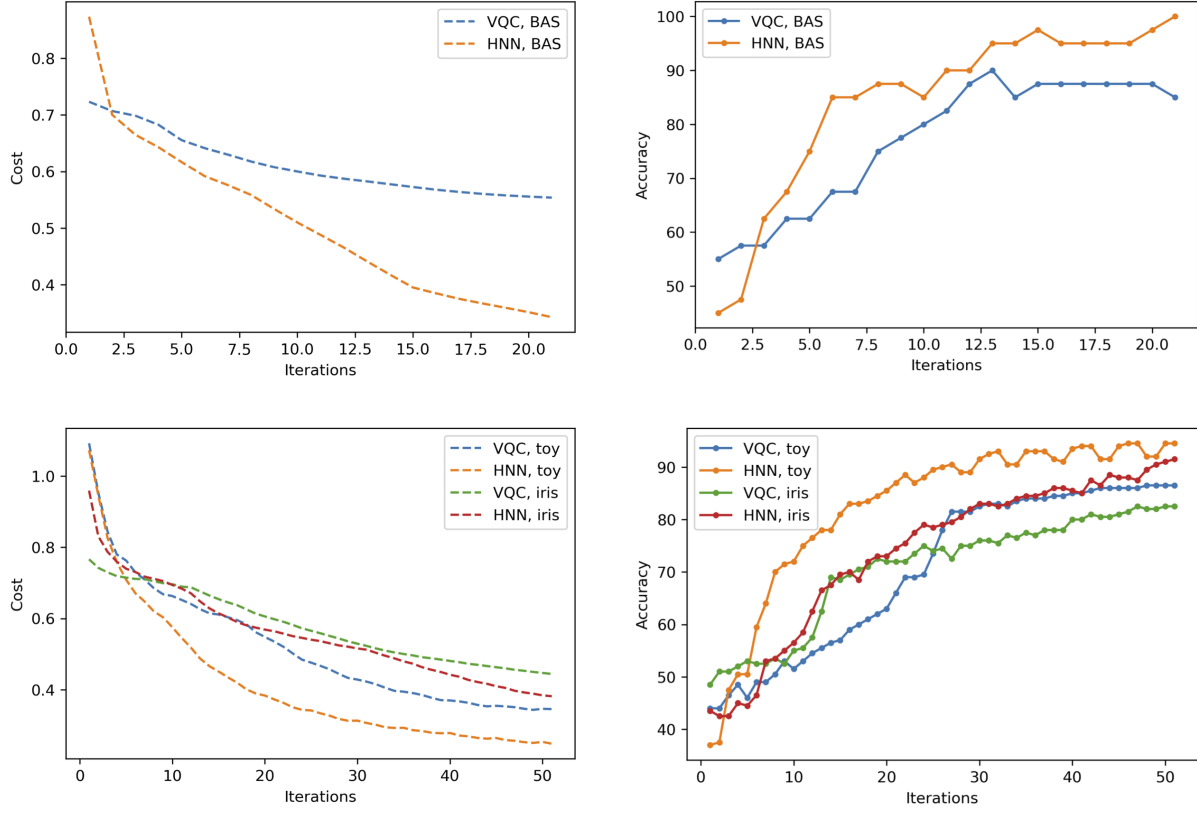
Fig. 3. Average cost and accuracy achieved by the variational quantum circuit (VQC) and hybrid neural network (HNN) during training on the 2×2 Bars and Stripes data set (BAS), a synthetic two-dimensional data set (toy), and a subset of the Iris data set (Iris). All illustrated trials were performed on simulated quantum hardware.

hardware	data	model	parameters	in sample accuracy			in sample cost			out of sample accuracy			out of sample cost		
				median	avg.	std.	median	avg.	std.	median	avg.	std.	median	avg.	std.
simulated	BAS	VQC	8	100.0	88.89	12.42	0.55	0.54	0.04	N/A	N/A	N/A	N/A	N/A	N/A
		HNN	20	100.0	100.0	0.0	0.33	0.35	0.07	N/A	N/A	N/A	N/A	N/A	N/A
simulated	toy	VQC	4	97.5	85.5	18.34	0.37	0.46	0.14	100.0	86.5	20.13	0.35	0.43	0.15
		HNN	12	97.5	93.88	9.0	0.29	0.33	0.13	97.5	94.5	8.79	0.25	0.29	0.14
simulated	Iris	VQC	8	88.12	81.5	14.37	0.45	0.48	0.12	87.5	82.5	17.92	0.44	0.48	0.12
		HNN	20	91.25	89.88	4.24	0.37	0.39	0.09	95.0	91.5	9.23	0.38	0.39	0.10
quantum	BAS	VQC	8	50.0	50.0	0.0	0.71	0.71	0.01	N/A	N/A	N/A	N/A	N/A	N/A
		HNN	20	25.0	33.33	11.79	0.71	0.72	0.11	N/A	N/A	N/A	N/A	N/A	N/A
quantum	toy	VQC	4	96.25	82.92	20.65	0.38	0.46	0.13	95.0	90.0	10.8	0.35	0.4	0.1
		HNN	12	96.25	95.0	3.68	0.26	0.31	0.07	100.0	95.0	7.07	0.23	0.27	0.06
quantum	Iris	VQC	8	50.0	50.0	0.0	0.72	0.72	0.0	55.0	55.0	0.0	0.72	0.72	0.0
		HNN	20	48.75	48.75	0.0	0.71	0.71	0.0	55.0	55.0	0.0	0.7	0.7	0.0

TABLE I

FINAL BINARY CLASSIFICATION RESULTS OF THE VARIATIONAL QUANTUM CIRCUIT (VQC) AND HYBRID NEURAL NETWORK (HNN) ON THE 2×2 BARS AND STRIPES DATA SET (BAS), A SYNTHETIC TWO-DIMENSIONAL DATA SET (TOY), AND A SUBSET OF THE IRIS DATA SET (IRIS). AVERAGE VALUES ARE DENOTED BY AVG. AND THE CORRESPONDING STANDARD DEVIATION IS DENOTED BY STD.
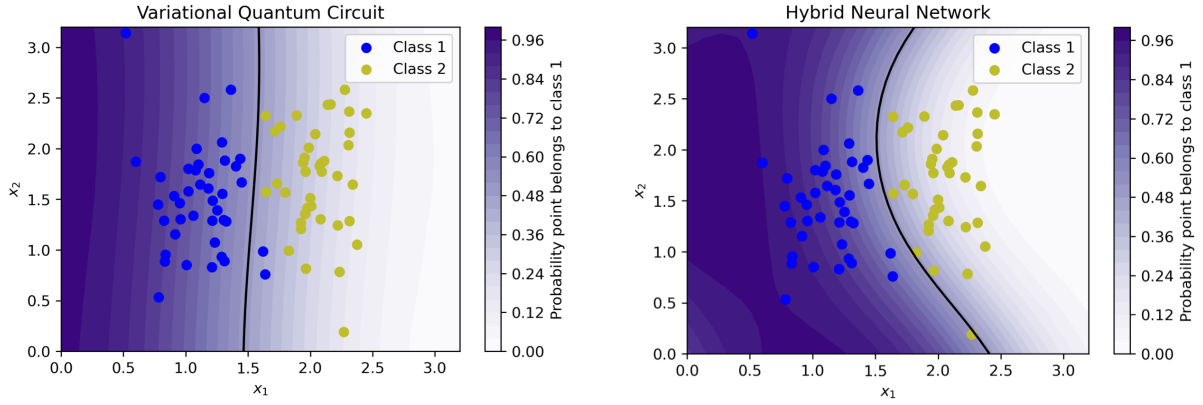
Fig. 4. Final classification line and probability distribution of a variational quantum circuit and a hybrid neural network trained on a two dimensional, linearly separable data set.

average accuracy was 8 to 11 percent higher, and the average cost was 20 to 40 percent lower. Notably, the advantages achieved by the hybrid neural network were observed on both the training data set and test data set. This suggests that they were not a product of overfitting. The learned Bernoulli distributions illustrated in Figure 4 give some indication of why the hybrid quantum-classical neural network achieves better performance. The neural network is able to produce a probability distribution with a much steeper gradient near the classification line. This enables the neural network to classify points with greater certainty than the individual variational quantum circuit, which in turn helps minimize cost. On the flip side, the hybrid neural network is more expressive than the variational quantum circuit classifier since it has more than twice as many parameters. Nevertheless, increasing the number of parameters of a machine learning model does not always guarantee better results, especially when data points outside of the training set are considered. The proposed hybrid neural network architecture illustrates one effective way to add parameters to a quantum machine learning model, especially on NISQ-era quantum computers.

Notably, when quantum hardware was used, the variational quantum circuit classifier and the hybrid neural network both performed extremely poorly on the Iris data set and the Bars and Stripes data set. This is likely because the number of qubits and number of required gates is proportional to the dimension of the data set. Increasing the number of qubits or increasing the number of gates adversely impacts the fidelity of modern quantum computation. Future research may investigate using a more sophisticated feature map or ansatz within the variational quantum circuits used by the neural network. Additionally, a more in depth study into hyper-parameter optimization of the learning rate and batch size may prove useful for improving results on modern quantum hardware. Finally, larger and more complex hybrid neural network architectures may be investigated on more challenging real-world classification problems.

IV. METHODS

A. Amplitude Encoding Feature Map

The amplitude encoding feature map is implemented by first initializing a quantum register of d qubits, each in the $|0\rangle$ state. Next, a single qubit parameterized RY gate is applied to each qubit. The parameters of each gate are chosen such that the ith qubit is rotated by an angle x_i. The probability p of measuring the ith qubit in the $|0\rangle$ state after the feature map is applied is given by the following equation:

$$p(x_i) = \cos^2\left(\frac{x_i}{2}\right) \tag{13}$$

Note that if x_i is restricted to the interval $[0, \pi]$, $p(x_i)$ is a one-to-one function with a range spanning all possible probabilities from 0 to 1. This feature map guarantees that each unique input $\boldsymbol{x}$ will have a unique quantum encoding without requiring a large number of quantum gates.

B. Real Amplitudes Ansatz

The *RealAmplitudes* ansatz from the Qiskit library consists entirely of single qubit RY rotation gates and two qubit CX entanglement gates. First, a parameterized RY gate is applied to each qubit. The parameters of these RY gates are the first d parameters of the ansatz. Next, two qubit CX gates are applied to each possible combination of qubits in the quantum state. The least significant qubit is used as the control bit each time. Finally, each qubit is subject to another parameterized RY gate. The parameters of these gates are the second d parameters of the ansatz. Additional rounds of CX entanglements and RY rotations can be added to the ansatz by adjusting the number of repetitions. However, to avoid long training times, we used one repetition for each variational quantum circuit. With this specification, the *RealAmplitudes* ansatz always has exactly $2d$ parameters.

C. Preprocessing Real Valued Data

As mentioned in Section IV-A, we require each feature to be within the interval $[0, \pi]$ before passing it to the feature map. For Bars and Stripes, this is not an issue since all features have
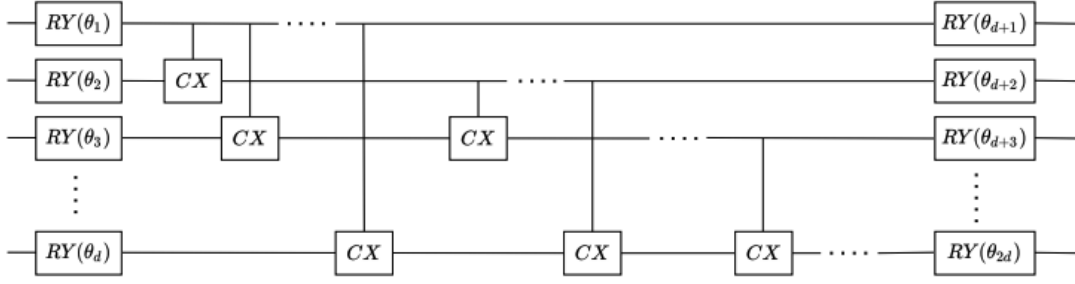
Fig. 5. *RealAmplitudes* ansatz from the Qiskit library with one repetition and full entanglement.

a binary value. Alternatively, the two real-valued data sets must be modified before training since both include many points with feature values outside of the desired range. We prepared these data sets for the variational quantum circuit using the following procedure:

1) Scale the data set so that each feature has a mean of 0 and a variance of 1 using Sci-Kit Learn's *StandardScalar* class.
2) Divide the modified data set by its feature with the largest absolute value. Now all features in the data set have a value between -1 and 1.
3) Multiply the modified data set by $\pi/2$.
4) Add $\pi/2$ to each feature of the modified data set.

After this procedure is performed, each feature in the modified data set will fall within the interval $[0, \pi]$, ensuring that every unique input x will have a unique quantum encoding.

D. Hardware and Job Specifications

During the simulated quantum trials, each quantum circuit was run using the IBM QASM simulator. During the actual quantum trials, each quantum circuit was run using the IBM Mumbai quantum computer. This computer has 27 qubits and a quantum volume of 128. The IBM Mumbai machine uses CX, ID, RZ, SX, and X gates as its basis gates. When our results were compiled, the average CNOT error of IBM Mumbai was 8.572×10^{-3}, and the average readout error was 3.834×10^{-2}. To determine the output (expectation value) of a variational quantum circuit on a particular input, we ran the circuit 1024 times and then averaged the result of each run. Each job was initialized and sent to the quantum computer using a personal laptop with a 2.7 GHz Dual-Core Intel i5 processor and 8 GB 1,867 MHz DDR3 memory. This laptop was also used to process the results of each job and optimize model parameters accordingly.

REFERENCES

[1] M. A. Myszczynska, P. N. Ojamies, A. M. B. Lacoste, D. Neil, A. Saffari, R. Mead, G. M. Hautbergue, J. D. Holbrook, and L. Ferraiuolo, "Applications of machine learning to diagnosis and treatment of neurodegenerative diseases," *Nature Reviews Neurology*, vol. 16, pp. 440–456, 2020.

[2] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 2017, pp. 1–9.

[3] J. Sterne, *Artificial intelligence for marketing: practical applications.* John Wiley & Sons, 2017.

[4] N. C. Thompson, K. H. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," *CoRR*, vol. abs/2007.05558, 2020. [Online]. Available: https://arxiv.org/abs/2007.05558

[5] F. Peper, "The end of moore's law: Opportunities for natural computing?" *New Generation Computing*, vol. 35, no. 3, pp. 253–269, 2017.

[6] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[7] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, p. 195–202, 2017.

[9] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, "Quantum machine learning: a classical perspective," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2209, p. 20170551, 2018.

[10] P. Date and T. Potok, "Adiabatic quantum linear regression," *Scientific Reports*, vol. 11, no. 1, pp. 1–10, 2021.

[11] P. Date, D. Arthur, and L. Pusey-Nazzaro, "QUBO formualtions for training machine learning models," *Scientific Reports*, vol. 11, no. 10029, 2021.

[12] D. Arthur and P. Date, "Balanced k-means clustering on an adiabatic quantum computer," *Quantum Information Processing*, vol. 20, no. 9, pp. 1–30, 2021.

[13] P. Date, "Quantum discriminator for binary classification," 2020.

[14] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[15] A. L. Blum and R. L. Rivest, "Training a 3-node neural network is np-complete," *Neural Networks*, vol. 5, no. 1, pp. 117–127, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608005800103

[16] M. Schuld, I. Sinayskiy, and F. Petruccione, "Simulating a perceptron on a quantum computer," *Physics Letters A*, vol. 379, no. 7, pp. 660–663, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037596011401278X

[17] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim, "Quantum generalisation of feedforward neural networks," *npj Quantum Information*, vol. 3, no. 36, 2017.

[18] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, , and S. Lloyd, "Continuous-variable quantum neural networks," *Phys. Rev. Research*, vol. 1, 2019.

[19] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, "Training deep quantum neural networks," *Nature Communications*, vol. 11, no. 808, 2020.

[20] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Information*, vol. 5, no. 103, 2019.

[21] A. Kamruzzaman, Y. Alhwaiti, A. Leider, and C. C. Tappert, "Quantum deep learning neural networks," in *Future of Information and Communication Conference*. Springer, 2019, pp. 299–311.

[22] S. Garg and G. Ramakrishnan, "Advances in quantum deep learning: An overview," 2020.

[23] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, pp. 1–20, 2021.

[24] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nature Computational Science*, vol. 1, pp. 403–409, 2021.

[25] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Sci. Technol.*, vol. 4, no. 4, 2019.

[26] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters, M. Leib, A. Skolik, M. Streif, D. V. Dollen, J. R. McClean, S. Boixo, D. Bacon, A. K. Ho, H. Neven, and M. Mohseni, "Tensorflow quantum: A software framework for quantum machine learning," 2020.

[27] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms," *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1900070, 2019.

[28] T. Hubregtsen, J. Pichlmeier, P. Stecher, and K. Bertels, "Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability," *Quantum Machine Intelligence*, vol. 3, no. 1, pp. 1–19, 2021.

[29] G. Chen, Q. Chen, S. Long, and W. Zhu, "Quantum convolutional neural network for image classification," in *2020 8th International Conference on Digital Home (ICDH)*, 2020, pp. 116–120.

[30] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Phys. Rev. A*, vol. 103, p. 032430, Mar 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.103.032430

[31] T. Goto, Q. H. Tran, and K. Nakajima, "Universal approximation property of quantum feature map," 2020.

[32] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, "Quantum embeddings for machine learning," 2020.

[33] H. Yano, Y. Suzuki, R. Raymond, and N. Yamamoto, "Efficient discrete feature encoding for variational quantum classifier," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2020, pp. 11–21.

[34] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Phys. Rev. A*, vol. 101, p. 032308, Mar 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.101.032308

[35] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, and Y.-J. Kao, "An end-to-end trainable hybrid classical-quantum classifier," *Machine Learning: Science and Technology*, 2021. [Online]. Available: http://iopscience.iop.org/article/10.1088/2632-2153/ac104d

[36] ——, "Hybrid quantum-classical classifier based on tensor network and variational quantum circuit," 2020.

[37] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018.

[38] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A*, vol. 98, p. 032309, Sep 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.98.032309

[39] G. E. Crooks, "Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition," 2019.