Tensor Ring Parametrized Variational Quantum Circuits for Large Scale Quantum Machine Learning

Dheeraj Peddireddy
Department of Industrial Engineering
Purdue University
West Lafayette, IN 47907
dpeddire@purdue.edu

Vipul Bansal
Indian Institute of Technology Roorkee
Roorkee, India
vbansal@me.iitr.ac.in

Zubin Jacob
Department of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907
zjacob@purdue.edu

Vaneet Aggarwal
Department of Industrial Engineering
Purdue University
West Lafayette, IN 47907
vaneet@purdue.edu

Abstract

Quantum Machine Learning (QML) is an emerging research area advocating the use of quantum computing for advancement in machine learning. Since the discovery of the capability of Parametrized Variational Quantum Circuits (VQC) to replace Artificial Neural Networks, they have been widely adopted to different tasks in Quantum Machine Learning. However, despite their potential to outperform neural networks, VQCs are limited to small scale applications given the challenges in scalability of quantum circuits. To address this shortcoming, we propose an algorithm that compresses the quantum state within the circuit using a tensor ring representation. Using the input qubit state in the tensor ring representation, single qubit gates maintain the tensor ring representation. However, the same is not true for two qubit gates in general, where an approximation is used to have the output as a tensor ring representation. Using this approximation, the storage and computational time increases linearly in the number of qubits and number of layers, as compared to the exponential increase with exact simulation algorithms. This approximation is used to implement the tensor ring VQC. The training of the parameters of tensor ring VQC is performed using a gradient descent based algorithm, where efficient approaches for backpropagation are used. The proposed approach is evaluated on two datasets: Iris and MNIST for the classification task to show the improved accuracy using more number of qubits. We achieve a test accuracy of 83.33% on Iris dataset and a maximum of 99.30% and 76.31% on binary and ternary classification of MNIST dataset using various circuit architectures. The results from the IRIS dataset outperform the results on VQC implemented on Qiskit, and being scalable, demonstrates the potential for VQCs to be used for large scale Quantum Machine Learning applications.

1 Introduction

Quantum computing has been demonstrably proven to be superior than classical computing when solving problems such as searching an unstructured database [10] or factorization of large numbers [22]. Due to the rising interest in the computation capabilities of noisy intermediate scale quantum (NISQ) computers and the availability of experimentation platforms, there has been a rapid growth in

the development of circuit based algorithms. Variational Quantum Circuits or Parametrized Quantum circuits are one such class of the circuit based algorithms that has been studied extensively [1], specifically in context of their capabilities in solving various combinatorial optimization problems and intrinsic energy problems of molecules which were either intractable or computationally expensive using classical devices. These studies have also been extended to use VQCs as a replacement for artificial neural networks (ANN) in discriminative and generative tasks. VQCs have been studied for their potential to achieve quantum advantage over their classical counterparts by reducing the number of model parameters and faster learning time for comparable performance. Despite its infancy, there are numerous works exploring the algorithmic aspects and applications of Variational Quantum Circuits. VQCs have first been introduced in [19] which theoretically and numerically proved that quantum circuits can approximate non-linear functions similar to neural networks. Several works have developed Quantum Approximate Optimization Algorithm and its generalizations to train parametrized quantum circuits using classical optimizers [30, 9, 18]. Circuit based algorithms have been developed to emulate several methods from the classical machine learning literature such as Long Short-Term Memory [3] and Convolutional Networks [4, 13]. VQCs have also replaced the neural networks in Reinforcement Learning tasks [2, 14] and Generative Algorithms [20, 11]. It has also been shown that that the VQCs have a better expressive power compared to neural networks [7].

Despite several studies pointing towards the promise of quantum superiority, a key challenge in the the current era of quantum computing is the limited quantum resources which in turn limits the number of qubits and the circuit depth. The other challenge is the inherent tendency of quantum devices towards decoherence, random gate errors and measurement errors which might severely limit the training of the variational circuits. To circumvent this issue, VQCs are often trained on classical devices before deploying them on quantum devices in order to avoid the compounding effect of decoherence noise throughout the numerous iterations required for training. However, the simulation of quantum computations on a classical device grow exponentially harder in both the number of qubits and the circuit depth. These shortcomings have limited the studies thus far to shallow circuits with small number of qubits on small scale applications. One of the well studied approaches to compress the information in the quantum state is using the matrix product states (MPS) [21]. MPS and its generalizations have been used extensively in the classical simulation of quantum circuits [24, 16] owing to their capability of accurately capturing low to moderate entanglements in many-qubit quantum states. Several approximate techniques have been developed using the MPS parametrization such as Density Matrix Renormalization Group (DMRG) [29], Projected entangled pair states (PEPS) [23] and multi-scale entanglement renormalization ansatz (MERA) [25]. A recent work [31] has constructed an approximation technique that effectively simulates a real quantum device by inducing a noise to represent the decoherence which limits the amount of entanglement that can be built into a quantum state. However, they (i) consider MPS structure, while we consider an extension of that, and (ii) do not consider the problem of training VQCs.

We note that the MPS (also called tensor train (TT) representation) has been used for scalability of quantum networks. However, such a decomposition suffers from the following limitations: (i) TT model requires rank-1 constraints to the border factors, (ii) TT ranks are typically small for near-border factors and large for the middle factors, and (iii) the multiplications of the TT factors are not permutation invariant. In order to alleviate these issues, researchers have started to use tensor ring (TR) representation in the classical machine learning [26, 27, 15]. TR decomposition removes the unit rank constraints for the boundary tensor factors and utilizes a trace operation in the decomposition. The multilinear products between factors also have no strict ordering and the factors can be circularly shifted [26]. This allows for all ranks to be the same. Indeed, TR representation has been shown to significantly outperform TT representation for data completion [26], compression of classical neural networks [27], etc. This motivates the use of TR representation for compressing the quantum state in the VQC in this paper.

In this paper, we consider the VQC architecture, where the parameters can be trained for classification. We assume that the initial data is prepared in a known state, which is represented in a tensor ring format. Then, the features are encoded using single qubit gate rotations to this tensor ring structure. We note that any single qubit gate maintains the tensor ring structure. Further, the key component of parametrized circuit in each layer of VQC consists of cyclic entanglement of qubits followed by parametrized single qubit rotations. We note that the two qubit gates (e.g., CNOT gate) do not preserve the tensor ring representation, where an approximation is performed for this gate which is based on a singular value thresholding. We further note that the structure of tensor ring allows for

easy computations of 2-qubit gates for the neighboring qubits. Due to the use of a cyclic structure of tensor ring, we are able to perform a CNOT from the last qubit to the first qubit too. While other operations can be done, the complexity will be higher than performing the 2-qubit gates on neighboring qubits (cyclically). We further note that since the tensor ring format is used, same rank r can be used for each decomposition, which may not be the case for the standard MPS format. Using this approximation, all the forward pass computations are linear in the number of gates. Further, we note that the state vector is never stored, and only the decomposed tensors of the tensor ring structure are stored. Thus, the storage complexity for n qubits reduces from $O(2^n)$ to $O(nr^2)$, where r is the rank of the tensor ring. Similarly, we can compute the gradients with an efficient use of backpropagation to learn the parameters in the VQC.

To the best of authors' knowledge, this is the first work on scaling VQCs using tensor rings. We note that the approximation of two-qubit gates using singular value decomposition brings non-linearity in the operations, and is inherently like adding the ReLU or soft-max operation. in classical neural networks. We believe that this non-linearity further helps in the selection of parameters of VQCs, and thus helping achieve improved performance. We evaluate the performance of the proposed approach on Iris and MNIST datasets. The performance is significantly better than the implementation of VQCs in Qiskit [5]. This demonstrates that the proposed architecture using TR representation outperforms not using approximation. This is likely because of introduction of non-linearity and efficient computation of gradients in the compressed TR representation. Further, due to linear scaling of computation and storage, this structure allows for scaling in the number of qubits, which has a potential to demonstrate quantum supremacy in machine learning.

2   Methodology

2.1   Variational Quantum Circuits

A Variational Quantum Circuit or Parametrized Quantum Circuits are quantum circuits with unitary rotation gates parametrized by trainable parameters. Several variational quantum algorithms have been developed to train the aforementioned VQCs for tasks like optimization, approximation, and classification. These variational quantum algorithms are usually quantum-classical hybrid algorithms where the forward pass through the quantum circuits are evaluated using quantum transformations and the parameters of the quantum circuit are trained using classical optimization algorithms. VQCs are used in similar applications as that of neural networks which primarily include function approximation. Quantum circuits use two qubit gates to encode entanglement into the quantum state which acts analogous to the activation functions in neural networks. Note that the quantum circuits (which do not include measurement) are linear reversible transformations and multitude of classical machine learning literature has established the importance of non-linearity in function approximation. This will be one of the key changes in our approximation, which will bring in non-linearity into the approximation.

We demonstrate how a VQC works along with a pictorial representation in Figure 1. Initially, the classical data is encoded into the qubits using rotations parametrized by the input data. The prepared qubit state is then transformed through a series of unitary rotation parametrized by trainable weights and entanglements. A subset of the transformed qubits are then measured to obtain the output in the form of expected value. The expected values are decoded into the appropriate class labels. A loss is computed akin to the neural networks using the decoded output and the labels from the dataset. The free parameters are then updated using well established classical optimization algorithms such as gradient descent, Adam, etc. The trained circuit can then be used to predict the labels of the test data.

2.2   Tensor Ring Parametrization of VQC

Quantum circuits become too difficult to simulate on classical computers for large number of qubits since the storage and computation is exponential in the number of qubits. Tensor networks can be thought of as a graphical representation of tensors, where each node in the graph is a tensor with a finite rank (a rank-0 tensor is a scalar, rank-1 tensor is a vector, rank-2 tensor is a matrix etc.). A tensor ring is a tensor network which aims to represent a higher order tensor by a sequence of 3rd order tensors that are multiplied circularly as shown in Figure 2. A higher order tensor τ of rank d is decomposed into d tensors denoted by $M(k)$ each of rank 3 spanned by the indices $\{n_k, r_k, r_{k+1}\}$ where r_k and r_{k+1} represent the bond dimension (or elements of the rank vector).
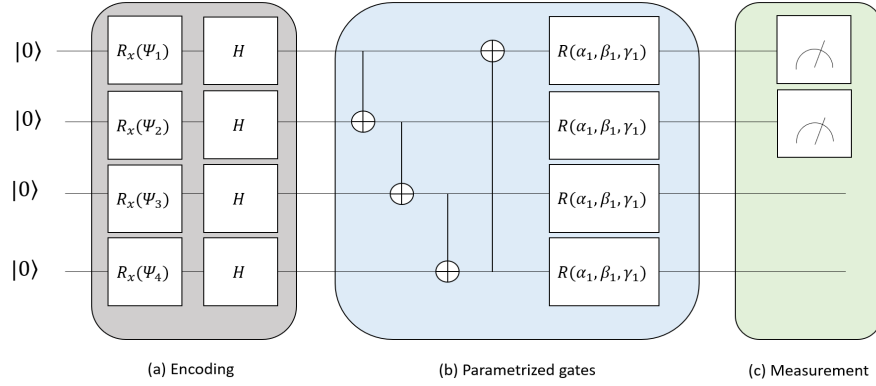
Figure 1: **Illustration of a generic 4-qubit variational quantum circuit** $R_x(\cdot)$ gates denote a rotation about the x-axis on Bloch sphere by a certain angle. H represents the Hadamard gate. (See Appendix for the matrix representations)(a) The first part of the circuit encodes the classical features ψs onto the qubits, (b) the rotation gates with free parameters transform the qubits are denoted by $R(\alpha, \beta, \gamma)$ and (c) a subset of the qubits are measured to obtain the output
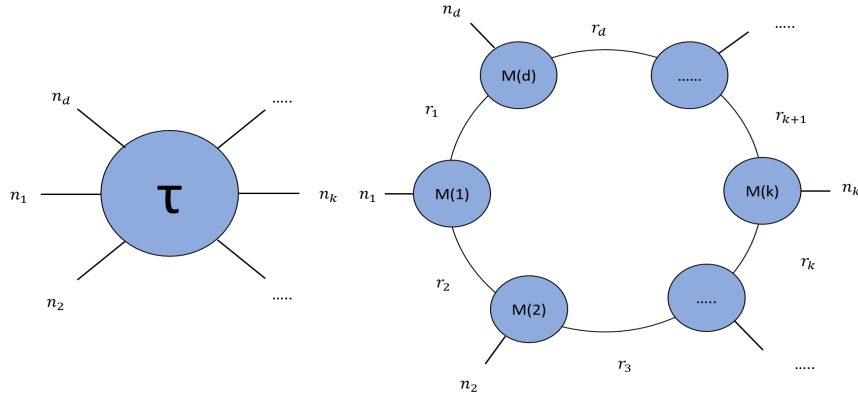

Figure 2: Illustration tensor ring decomposition of a tensor of rank d

It has been shown that the tensor rings are a better and more effective compression algorithm than tensor trains (or MPS format) [26, 27], which motivates our choice. The key is that the tensor train representation has lower border ranks and larger interior ranks which limit an efficient representation, while all ranks could be made similar in tensor ring format and thus decrease of ranks helps achieve better representation at the same number of parameters. We also note that the typical format of VQC has a cyclic entanglement, where the CNOT from the last qubit to the first qubit will not be directly possible using the MPS representation, which further motivates the use of tensor ring format in this paper. We represent the quantum state $|\phi\rangle$ using N tensors, denoted by $M(n)$, as below:

$$|\phi\rangle = \sum_{i_1 \ldots i_N} \sum_{\mu_1 \ldots \mu_N} M(1)^{i_1}_{\mu_N \mu_1} M(2)^{i_2}_{\mu_1 \mu_2} \ldots M(N)^{i_N}_{\mu_N \mu_1} |i_1 i_2 \ldots i_N\rangle \tag{1}$$

where $i_n \in \{0, 1\}$ are the physical indices spanning the 2^N dimensional Hilbert space while $\mu_n \in \{1, \ldots, \chi_n\}$ are the bond indices of the tensors which control the maximum amount of entanglement captured by the tensor ring. If the χ_n is allowed to grow large, the tensor ring captures the exact quantum state information which comes at a computational cost. When designing the training algorithm, χ_n is considered to be one of the hyperparameter choices along with batch size, learning
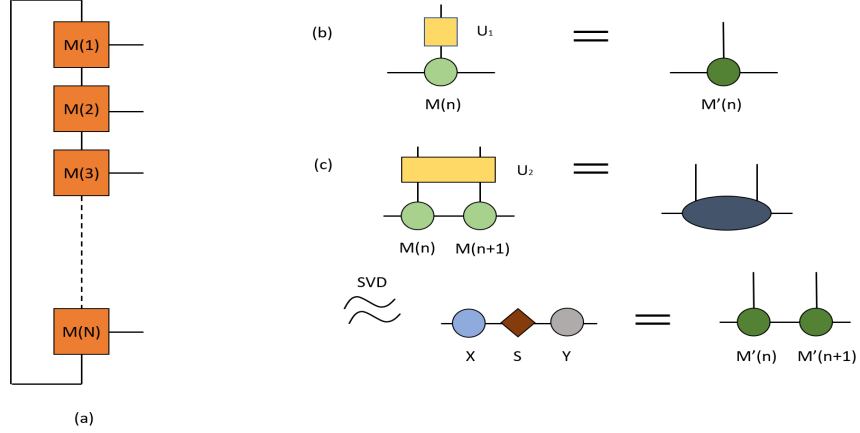
Figure 3: (a) A tensor ring structure representing the N-qubit state with bond dimension χ (b) Applying a single qubit gate to the n-th qubit of a tensor ring (c) Applying a two qubit gate to n-th and n+1-th qubits of a tensor ring and using the truncated SVD to decompose the resultant matrix into new tensors

rate, etc. We assume $\chi_n = \chi$ for all n, reducing the hyperparameters following [26]. This bond-dimension χ is called the tensor ring rank. A tensor ring parametrization of an N-qubit state is illustrated in Figure 3(a). Each square in the ring is a tensor of dimension $\chi \times \chi \times 2$. The open bonds of the tensor ring index the 2-dimensional Hilbert space of the qubit corresponding to the site.

We first consider a one qubit rotation which is a tensor contraction of a 2×2 matrix with the tensor corresponding to the qubit. A one qubit rotation (represented by a unitary matrix U) on n^{th} qubit is given by the following equation:

$$M'(n)^{i'_n}_{\mu_{n-1}\mu_n} = \sum_{i_n} U_{i'_n i_n} M(n)^{i_n}_{\mu_{n-1}\mu_n} \qquad (2)$$

In order to perform a two qubit gate transformation on qubits n and $n+1$, we first transform the tensor ring into an orthogonal form centered around the qubits of interest. A series of operations are performed followed by a singular value decomposition as shown below. The two tensors at n and $n+1$ are first contracted along the shared bond index which is given by:

$$T^{i_n i_{n+1}}_{\mu_{n-1}\mu_{n+1}} = \sum_{\mu_n} M(n)^{i_n}_{\mu_{n-1}\mu_n} M(n+1)^{i_{n+1}}_{\mu_n\mu_{n+1}} \qquad (3)$$

The two qubit gate (U, reshaped into $U_{i'_n i'_{n+1} i_n i_{n+1}}$) is then applied on the two qubit tensor computed in the previous equation:

$$(T')^{i_n i_{n+1}}_{\mu_{n-1}\mu_{n+1}} = \sum_{i_n i_{n+1}} U_{i'_n i'_{n+1} i_n i_{n+1}} T^{i_n i_{n+1}}_{\mu_{n-1}\mu_{n+1}} \qquad (4)$$

Finally, we reshape the tensor T' into a matrix of shape $(i'_n + \mu_{n-1}) \times (i'_{n+1} + \mu_{n+1})$, and perform singular value decomposition of the matrix:

$$(T')^{i_n i_{n+1}}_{\mu_{n-1}\mu_{n+1}} = \sum_{\mu_n} X^{i'_n}_{\mu_{n-1}\mu_n} S_{\mu_n} Y^{i'_{n+1}}_{\mu_n\mu_{n+1}} \qquad (5)$$

where the X and Y matrices are composed of the orthogonal vectors and S_μ contains the singular values of the matrix T'. The matrix has 2χ singular values irrespective of the two qubit gate structure where χ denotes the bond dimension of the tensor ring. We then truncate the S_μ matrix to keep only the χ largest singular values and the resulting matrix is denoted by S'_μ. X and Y are truncated to

5

only keep the orthogonal vectors corresponding to the χ largest singular values. The new tensors corresponding the two qubits are given by:

$$M'(n)^{i_n}_{\mu_{n-1}\mu_n} = X^{i_n}_{\mu_{n-1}\mu_n} S'_{\mu_n} \tag{6}$$

$$M'(n+1)^{i_{n+1}}_{\mu_n\mu_{n+1}} = Y^{i_{n+1}}_{\mu_n\mu_{n+1}} \tag{7}$$

Without the approximation, the operation of 2-qubit gate will be exponential in the number of qubits, the above operations are $O(1)$ in the number of qubits, and thus helps scalability of the approach. The single qubit and two qubit transformations are demonstrated in Figure 3.

2.3 Training

We first explain the overall VQC architecture. We prepare the initial N qubit state as $|00\cdots0\rangle$, which is then converted to a TR representation as $M(i)$ is a $\chi \times \chi \times 2$ tensor with only $(1,1,1)$ element as 1 and rest as zeros. Then, the data features are used to perform single qubit rotations as illustrated in the **encoding** step of Fig. 1 (as one possible approach, while any other encoding mechanism can be used). The **parametrized gate circuit** is chosen as a modification of that illustrated in Fig. 1. This gate circuit is repeated d times to illustrate the depth of the circuit. Finally, some of the qubits are **measured**, based on which the loss function is determined for training.

We evaluate the "forward pass" of the circuit by computing the single qubit and two qubit transformations as discussed in the previous sub-sections to obtain the final output obtained by measuring certain number of qubits. The loss is then computed using both the final output and the ground truth, and the gradients are calculated by backpropagating through the tensor ring structure similar to neural networks. The details of backpropagation through quantum gates and singular value decomposition can be found in [28, 12]. The gate parameters are then updated using a classical optimizer like Adam owing to its promising performance with neural networks. Given the compression obtained by the tensor ring representation and the the singular value decomposition used at the two qubit gates, the training of the proposed method would be much faster than training the full VQC. Once trained, the parameters can be deployed on a real quantum device to evaluate the performance on unseen data.

2.4 Complexity

We first note that we never store the quantum state, unlike in standard quantum circuit. So, at each step of the operation, we only store the N tensors, giving an overall storage complexity of $O(N\chi^2)$, where the tensor-ring rank χ is a hyper-parameter.

For computation, each computation of single or two-qubit gate in the tensor ring VQC is $O(1)$, and thus the overall complexity of the forward pass is $O(Nd)$, or the size of the circuit/number of gates. Similarly, each backpropagation round is the same complexity. Thus, each iteration of training has complexity $O(Nd)$, thus making the overall approach scalable in the number of qubits.

To illustrate the capabilities of the stated method of tensor rings for simulating variational quantum circuits, we perform various experiments namely, iris classification [8], binary classification using MNIST data set [6] on class 0 and 1, and ternary classification using MNIST data set on class 0, 1, and 2.

2.5 Iris Classification

The Iris data set includes four features: sepal length, sepal width, petal length, and petal width. The dataset is split in a ratio of $8:2$ for training/testing data. The circuit ansatz chosen for the problem is illustrated in Figure 4. The features are normalized before being encoded into the qubits. The initially prepared 4-qubit state of $|0000\rangle$ is decomposed into a tensor ring as described before. The features ψ extracted from the data are encoded onto the qubits using a layer of R_x gates. This is followed by a sequence of CNOT gates entangling all the consecutive qubits. After this, a sequence of single-qubit gates, i.e., R_x, R_y, and R_z (See Appendix for matrix representations), are applied to each qubit with trainable parameters. Here, $R_x(\cdot), R_y(\cdot), R_z(\cdot)$ are rotation gates about the x, y and z axes respectively in the Bloch sphere. The CNOT gate is represented by a connection between two
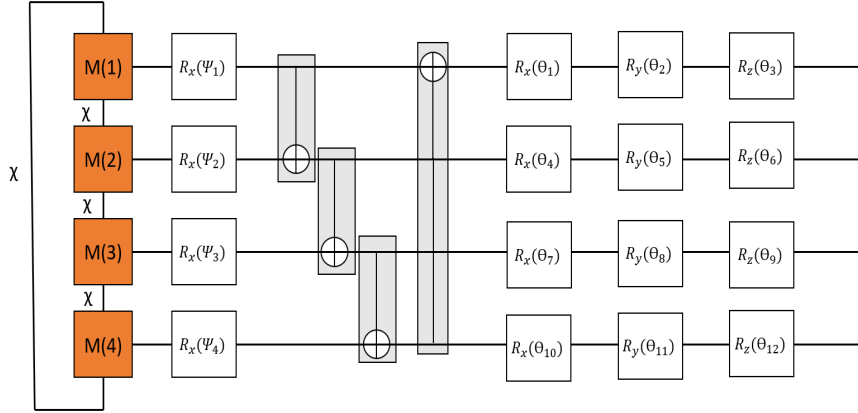
Figure 4: Circuit architecture for the Iris Classification problem

qubit "wires" as show in Figure 4.The circular end of the connection represents the target qubit and the dot end represent the control qubit. A NOT gate or Pauli gate X is applied on the target qubit if the control qubit is a $|1\rangle$.The four feature vector is fed into the VQC for training. Out of the 2^4 possible measurements obtained from the VQC, we randomly select 3 measurements for representing 3 classes of the dataset. We further apply sigmoid function to selected measures to convert them to class probabilities. Then, we utilize a cross entropy loss and back-propagate to obtain gradients for each trainable parameter. In this experiment, we use an Adam optimizer with a learning rate of 0.001 for 50 epochs for a batch size of 4 . The model achieved a final train set accuracy of 72.5% and a test set accuracy of 83.33% while the VQC implementation on Qiskit with full quantum state information results in a train accuracy of 55.03% and a test accuracy of 57.04%

2.6   Binary Classification: MNIST

We perform a binary classification MNIST dataset with classes 0 and 1. MNIST dataset consists of images of size 28×28, which is reduced to a feature vector using an autoencoder. The autoencoder consists of 2 convolution layers in encoder assembly and 3 transpose convolution layers in the decoder assembly. We generate a feature space of dimension 4, 8, 16 and 32 for multiple experiments.The cirucit structure uses the same layer as the experiment in IRIS dataset, however multiple experiments are conducted by repeating these layers multiple times. The initially prepared qubit state $|0000\rangle$ for a 4 dimension feature space is decomposed into a tensor ring as described before. We select the final measurements $|0000\rangle$ and $|1111\rangle$ as output values for 2 classes. We further apply logarithmic

Dataset	#qubits	Tensor Ring VQC		
		1 layer	2 layers	3 layers
IRIS	4	83.33%	-	-
MNIST (0 & 1)	4	79.57%	98.51%	99.30%
	8	90.01%	85.70%	92.84%
	16	80.73%	95.72%	93.45%
	32	75.53%	94.84%	97.58%
MNIST (0,1&2)	4	74.08%	74.51%	54.75%
	8	76.31%	67.05%	58.92%
	16	53.88%	52.67%	70.35%

Table 1: Results from training VQCs with varying number of qubits and circuit ansatz on three datasets
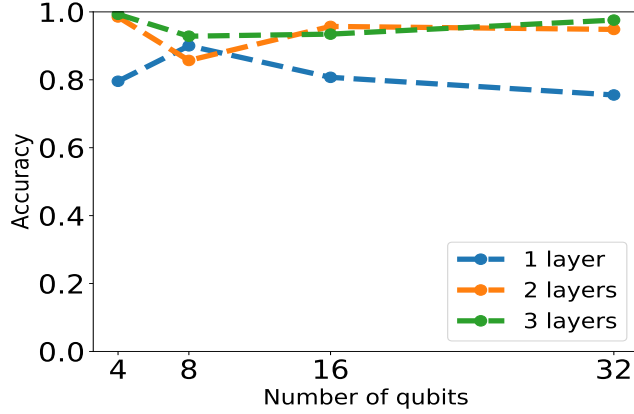
7

Figure 5: Performance of Tensor Network w.r.t number of qubits and number of layers in the circuit on binary classification of MNIST
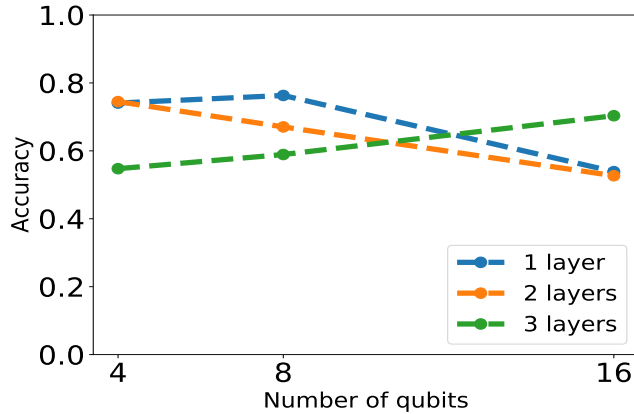

Figure 6: Performance of Tensor Network w.r.t number of qubits and number of layers in the circuit on ternary classification of MNIST

sigmoid function to selected measurements to obtain class probability and compute a negative log likelihood loss and backpropagate to obtain gradients. The accuracy curves with respect to number of qubits layers are shown in Figure 5.

2.7   Ternary Classification: MNIST

We perform a ternary classification MNIST dataset with classes 0, 1, and 2. The feature vector is generated in the same way as binary classification using autoencoders, experimenting with 4, 8 and 16 qubit states. We use a similar ansatz as binary classification with the variational layer repeating multiple times. We select 3 possible measurements as value for 3 classes. We further apply logarithmic sigmoid function to selected measurements and compute a negative log likelihood loss and back propagate. The VQC is trained with an Adam optimizer with a learning rate of 0.01 and a batch size of 10000 for 5 epochs. The accuracy curves with respect to number of qubits layers are shown in Figure 6.

2.8   Discussion

The overall results are summarized in Table 1. We observe a general pattern of increasing classification accuracy with the increasing number of qubits since a larger number of qubits indicate a larger Hilbert

space to parametrize the input data. We also notice that in the case of IRIS dataset, the tensor ring parametrized VQC significantly outperforms the vanilla implementation of VQC with full quantum state information owing to the additional non-linearity induced by the truncated singular value decomposition over the two qubit gate transformations. The tensor ring approximation also offers improved training speed as compared to the standard VQC due to linear complexity, which was observed in our evaluations for 8 qubits.

3 Conclusion

In this work, we propose a new technique to train Variational Quantum Circuits using an approximate tensor ring representation of the quantum state. The computational time of the circuit evaluation grows linearly in the number of qubits N and the depth of the circuit d with the proposed method, as compared to the exponential growth in the simulation of ideal quantum computers. We train the tensor ring variational quantum circuits using the proposed approach on Iris and a subset of MNIST datasets. The performance is significantly better than the implementation of VQCs in Qiskit. This demonstrates that the proposed architecture using TR representation outperforms not using approximation. Thus, we see that the approximation helps in both the scalability as well as performance improvement.

We note that the quantum circuit (except measurement) are linear in the Hilbert space, while the added approximation of two-qubit gates bring non-linearity in the circuit, similar to Rectified Linear Unit(ReLU) or soft-max in the classical neural networks. One of the major challenges in training large scale quantum circuits is the barren plateau effect [17] which demands good initialization to achieve comparable results. However, it is still to be seen whether this is also a challenge in tensor network representations of variational quantum circuits for even larger quantum bits. It would also be interesting to see the performance of proposed approach replacing neural networks in tasks like Reinforcement Learning or Generative Adversarial Networks etc.

4 Appendix

The matrix representation of some of the commonly used gates in the manuscript are listed below:

$$R_x(\theta) = \begin{bmatrix} cos(\theta/2) & -isin(\theta/2) \\ -isin(\theta/2) & cos(\theta/2) \end{bmatrix}, \; R_y(\theta) = \begin{bmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{bmatrix}, \; R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} cos(\alpha/2) & -e^{i\gamma}sin(\alpha/2) \\ e^{i\beta}sin(\alpha/2) & e^{i\beta+i\gamma}cos(\alpha/2) \end{bmatrix}$$

References

[1] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, pages 1–20, 2021.

[2] Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan. Variational quantum circuits for deep reinforcement learning. *IEEE Access*, 8:141007–141024, 2020.

[3] Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang. Quantum long short-term memory. *arXiv preprint arXiv:2009.01783*, 2020.

[4] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

[5] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March Meeting Abstracts*, volume 2018, pages L58–003, 2018.

[6] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[7] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Expressive power of parametrized quantum circuits. *Physical Review Research*, 2(3), Jul 2020.

[8] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

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

[10] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

[11] Amir Khoshaman, Walter Vinci, Brandon Denis, Evgeny Andriyash, Hossein Sadeghi, and Mohammad H Amin. Quantum variational autoencoder. *Quantum Science and Technology*, 4(1):014001, 2018.

[12] Cheng Hua Li and Soon Choel Park. An efficient document classification model using an improved back propagation neural network and singular value decomposition. *Expert Systems with Applications*, 36(2):3208–3215, 2009.

[13] Junhua Liu, Kwan Hui Lim, Kristin L Wood, Wei Huang, Chu Guo, and He-Liang Huang. Hybrid quantum-classical convolutional neural networks. *Science China Physics, Mechanics & Astronomy*, 64(9):1–8, 2021.

[14] Owen Lockwood and Mei Si. Reinforcement learning with quantum variational circuit. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 245–251, 2020.

[15] Osman Asif Malik and Stephen Becker. A sampling-based method for tensor ring decomposition. In *International Conference on Machine Learning*, pages 7400–7411. PMLR, 2021.

[16] Igor L Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008.

[17] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.

[18] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

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

[20] Jonathan Romero and Alán Aspuru-Guzik. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *Advanced Quantum Technologies*, 4(1):2000003, 2021.

[21] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of physics*, 326(1):96–192, 2011.

[22] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[23] Frank Verstraete, Valentin Murg, and J Ignacio Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in physics*, 57(2):143–224, 2008.

[24] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14):147902, 2003.

[25] Guifre Vidal. Entanglement renormalization. *Physical review letters*, 99(22):220405, 2007.

[26] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5697–5705, 2017.

[27] Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal. Wide compression: Tensor ring nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] Masaya Watabe, Kodai Shiba, Masaru Sogabe, Katsuyoshi Sakamoto, and Tomah Sogabe. Quantum circuit parameters learning with gradient descent using backpropagation. *arXiv preprint arXiv:1910.14266*, 2019.

[29] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.

[30] 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. *Physical Review X*, 10(2):021067, 2020.

[31] Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What limits the simulation of quantum computers? *Phys. Rev. X*, 10:041038, Nov 2020.