MACHINE LEARNING
Science and Technology

An end-to-end trainable hybrid classical-quantum classifier

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
31 January 2021

REVISED
22 June 2021

ACCEPTED FOR PUBLICATION
30 June 2021

PUBLISHED
8 September 2021

PAPER

An end-to-end trainable hybrid classical-quantum classifier

Samuel Yen-Chi Chen[1], Chih-Min Huang[2,3], Chia-Wei Hsing[2,3] and Ying-Jer Kao[2,3,4,*]

[1] Computational Science Initiative, Brookhaven National Laboratory, Upton, NY 11973, United States of America
[2] Department of Physics, National Taiwan University, Taipei 10617, Taiwan
[3] Center for Theoretical Physics, National Taiwan University, Taipei 10607, Taiwan
[4] Physics Division, National Center for Theoretical Science, Taipei 10607, Taiwan
[*] Author to whom any correspondence should be addressed.

E-mail: yjkao@phys.ntu.edu.tw

Keywords: quantum machine learning, tensor network, variational quantum circuit, principal component analysis

Abstract

We introduce a hybrid model combining a quantum-inspired tensor network and a variational quantum circuit to perform supervised learning tasks. This architecture allows for the classical and quantum parts of the model to be trained simultaneously, providing an end-to-end training framework. We show that compared to the principal component analysis, a tensor network based on the matrix product state with low bond dimensions performs better as a feature extractor for the input data of the variational quantum circuit in the binary and ternary classification of MNIST and Fashion-MNIST datasets. The architecture is highly adaptable and the classical-quantum boundary can be adjusted according to the availability of the quantum resource by exploiting the correspondence between tensor networks and quantum circuits.

1. Introduction

Quantum computing (QC) has demonstrated superiority in problems intractable on classical computers [1, 2], such as factorization of large numbers [3] and searching in an unstructured database [4]. Recent growth of the quantum volume in noisy intermediate-scale quantum (NISQ) [5] devices has stimulated rapid development in circuit-based quantum algorithms. Due to the noise associated with the quantum gates and lack of quantum error correction on NISQ devices, performing quantum computation with large circuit depth is impossible currently. It is therefore highly desirable to develop quantum algorithms that are resilient to noise with moderate circuit depth. Variational quantum algorithms [6] are a class of algorithms currently under rapid development in many fields. In particular, quantum machine learning (QML) [7–9] using variational quantum circuits (VQCs) shows great potential in surpassing the performance of classical machine learning (ML). One of the major advantages of VQC-based QML compared to its classical counterpart is the drastic reduction in the number of model parameters, potentially mitigating the problem of the overfitting common in the classical ML. Moreover, it has been shown that under certain conditions, QML models may learn faster or achieve higher testing accuracies than its classical counterpart [10, 11]. A modern QML architecture typically includes a classical and a quantum part. Famous examples in this hybrid genre include quantum approximate optimization algorithm [12], and quantum circuit learning [13], where the VQC plays a crucial role as the quantum component with the circuit parameters updated via a classical computer. Various architectures and geometries of VQC have been suggested for tasks ranging from binary classification [11, 13–15] to reinforcement learning [16–18].

One of the key challenges in the NISQ era is that available quantum hardware has limited quantum volume and is only capable of executing quantum operations with small circuit depth. That means that most of the dataset commonly used for classical ML tasks are too large for the NISQ devices. To process the data with input dimension exceeding the number of available qubits, it is necessary to apply dimensional reduction techniques to first compress the input data. For example, in [19], pre-trained a classical deep convolutional neural network is used to compress the high-resolution images into a low-dimension representation. However, since the pre-trained model there has a huge number of parameters, it is not clear what is the contribution of the quantum circuit in the whole workload.

On the other hand, a major challenge in building a QML model is how to encode high-dimensional classical data into a quantum circuit efficiently. With the limitation imposed by NISQ in mind, the encoding process should be designed to consume as few gate operations as possible. *Amplitude encoding* is one of the encoding methods which can provide significant advantage in terms of the number of qubits required to handle the input data. For an N-dimensional vector, amplitude encoding requires only $\log_2 N$ qubits; however, the quantum circuit depth to prepare such encoded state exceeds the current limits of NISQ devices. Other approaches like single-qubit rotations require only a shallow circuit but it is unclear how to employ such encoding schemes to load high-dimensional data into a quantum circuit. This can be potentially mitigated by preprocessing the input data with classical methods to perform dimension reduction. Principal component analysis (PCA) is a simple dimension reduction method and has been widely used in the QML research; yet it lacks the representation power to retain enough information. More powerful and expressive models such as neural networks are not commonly utilized due to the requirement of pre-training and the significant number of parameters involved. Therefore, it is necessary to devise a data compression scheme which can be naturally integrated with VQC.

In this paper, we propose a hybrid framework where a matrix product state (MPS) [20, 21], the simplest form of tensor networks (TN) [22], is used as a feature extractor to produce a low-dimensional feature vector. This information is subsequently fed into a VQC for classification. Unlike other QML schemes where the classical neural network has to be pre-trained, our framework is trained as a whole. This end-to-end training indicates the quantum–classical boundary can be adjusted based on the available quantum resource. Furthermore, since a MPS can be realized precisely by a quantum circuit [23], it is possible to replace the classical component with a quantum circuit, making the scheme highly adaptable. Our scheme has shown to be superior in the binary classification task for the MNIST dataset [24]. Here, we apply the scheme to more difficult tasks such as the ternary classification of MNIST and the classification tasks of Fashion-MNIST.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to tensor networks and their application in classical ML. Section 3 describes the VQC used in this study. Section 4 introduces the hybrid TN-VQC architecture. The performance of the model is shown in section 5. Finally we conclude in section 6.

2. Tensor network

Tensor networks are efficient representation of data residing in high-dimensional space. Originally developed in the context of condensed matter physics, TNs have gained attention in the deep learning community, for both theoretical understanding and computationally efficiency [25]. They have provided new inspiration for machine learning algorithms and showed encouraging success in both discriminative [26–30] and generative learning tasks [31–33]. In addition, the quantum entanglement inherent in the formulation of tensor networks points to a new direction in understanding the mechanism of deep neural networks and may provide a better way to design new network architectures [27, 34, 35].

It is common to use graphical notation to express tensor networks. A tensor is represented as a closed shape, typically a circle, with emanating lines representing tensor indices (figure 1). The joined line indicates the corresponding index is contracted, as in the Einstein convention where repeated indices are summed over. MPS, also known as Tensor Train, has been widely used in physics to study low-dimensional quantum systems. In an MPS, tensors are contracted through the 'virtual' indices (α's in figure 1(d)). The dimension of these virtual indices are called bond dimensions and is indicated by χ. In the MPS representation of a quantum wave function, the bond dimension indicates the amount of quantum entanglement the MPS can represent in the bond. In the context of ML, this corresponds to the representation power of the MPS.

In the realm of machine learning, MPS also has found numerous applications. For example, MPS is used for feature extraction of multidimensional data modeled as high-order tensors in [36], In [37], it has been used to compress the weight matrices in the deep neural networks to perform classification tasks. Stoudenmire and Schwab [26] and Efthymiou *et al* [29] use MPS directly to perform supervised learning tasks for the classification of MNIST and Fashion-MNIST datasets. The MPS has also been used for generative modeling [31], probabilistic modeling [38] and sequence modeling [39]. Finally, it is shown that the restricted Boltzmann machine, an important building block of deep learning, is equivalent to an MPS [40].

In addition to MPS, other examples of TNs with distinct entanglement structures exist, such as a tree tensor network (TTN), multi-scale entanglement renormalization ansatz (MERA) and projected entangled pair state (PEPS). The successful application of a specific TN can also give insights into the hidden correlations in the data. The quantumness inherent in the TN gives it great advantage over other

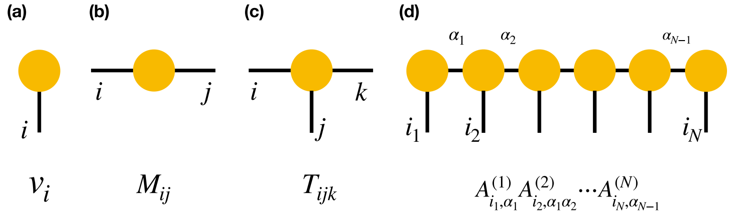
(a)
(b)
(c)
(d)
i
i j
i k j
α_1 α_2 α_{N-1}
i_1 i_2 i_N
v_i
M_{ij}
T_{ijk}
$A^{(1)}_{i_1,\alpha_1} A^{(2)}_{i_2,\alpha_1\alpha_2} \cdots A^{(N)}_{i_N,\alpha_{N-1}}$

Figure 1. Graphical notation for tensors and tensor networks. Graphical tensor notation for (a) a vector, (b) a matrix, (c) a rank-3 tensor and (d) a MPS. Here we follow the Einstein convention that repeated indices, represented by internal lines in the diagram, are summed over.

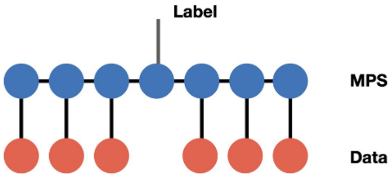

Figure 2. MPS as a feature extractor. Data is encoded into a product state (red nodes) which is contract with an MPS (blue nodes) and a class label or output is generated. The input data is first transformed via the feature map shown in equation (3) and then loaded into the MPS according to the equation (2).

architectures in the application of QML. In particular, since each TN can be mapped to a quantum circuit, it means that although in the current scheme, the TN is treated classically, it is possible to replace the whole or part of the TN by an equivalent quantum circuit when a bigger quantum volume is available. This gives our scheme the flexibility to move the quantum–classical boundary based on the available resources.

We will use the MPS as a feature extractor to compress the input data. Following [26], we approximate a feature extractor by the MPS decomposition as

$$T^{l}_{i_1 i_2 \cdots i_N} = \sum_{\{\alpha\}} A^{(1)}_{i_1,\alpha_1} A^{(2)}_{i_2,\alpha_1\alpha_2} A^{(3)}_{i_3,\alpha_2\alpha_3} \cdots A^{(j)}_{i_j,\alpha_{j-1},\beta} B^{l}_{\beta,\alpha_j} \cdots A^{(N)}_{i_N,\alpha_{N-1}}, \tag{1}$$

illustrated in figure 2. The input data is then loaded into the MPS via the following operation,

$$f^{(l)}(\mathbf{x}) = \sum_{i_1,i_2,\ldots,i_N=0}^{1} T^{l}_{i_1 i_2 \ldots i_N} \Phi(p_1)_{i_1} \Phi(p_2)_{i_2} \cdots \Phi(p_N)_{i_N} \tag{2}$$

3. VQC

VQCs originate from the *variational quantum eigensolver* [41], a family of quantum algorithms used to compute ground states. This family of algorithms have recently drawn significant attention and numerous efforts have been made to extend their applications [6]. VQCs have been successfully applied to function approximation [10, 13], classification [11, 13–15, 19, 42–46], generative modeling [47–51], metric learning [52, 53], deep reinforcement learning [16, 17, 54] , sequential learning [10, 55], speech recognition [56] and transfer learning [19]. It has been shown that VQCs are more expressive than conventional neural networks [57–60] with respect to the number of parameters or the learning speed. It is demonstrated that with similar number of parameters, VQC-based models outperform classical models on testing accuracies [11], and achieve optimal accuracy in function approximation tasks with fewer training epochs than their classical counterparts [10]. Of particular interest for NISQ applications, it has been shown that such circuits are potentially resilient to noises in quantum hardware [12, 61, 62], and such robustness has been demonstrated empirically on either noisy simulators or real quantum hardware [16, 53]. This strongly suggests that VQC-based architectures are suitable for building ML applications on NISQ devices.

The VQC used in this work consists of three parts (figure 3): the first part is the *encoding* part, which consists of Hadamard gate H and single qubit rotation gates $R_y(\arctan(x_i))$ and $R_z(\arctan(x_i^2))$, representing rotations along y-axis and z-axis by the given angle $\arctan(x_i)$ and $\arctan(x_i^2)$, respectively. The Hadamard gate H is is used to create an unbiased initial state as described in appendix A. Notice the rotation angles

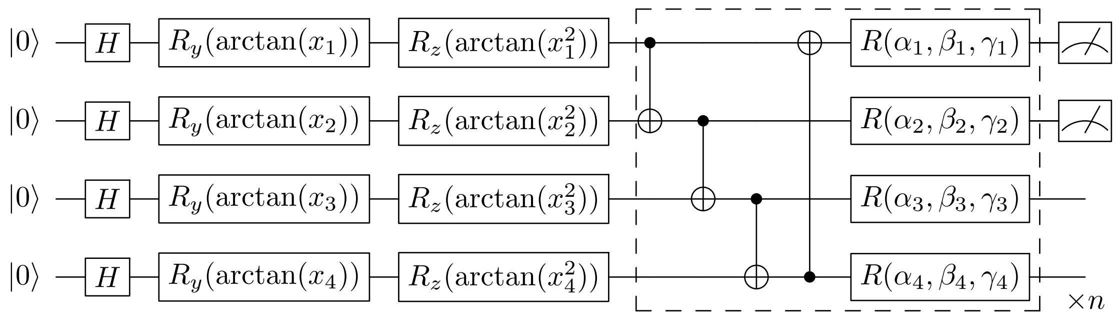

Figure 3. Generic variational quantum circuit architecture. The VQC component used in this work consists of three parts: the encoding part, the variational part with parameters to be learned and the measurement part which will output the Pauli-Z expectation values via multiple runs of the quantum circuit. The quantum measurement would be performed on first k qubits where k is the number of classes. The dashed block is repeated n times to increase the expressivity of the VQC. In the current work, $n = 4$ unless otherwise specified.

$\arctan(x_i)$ and $\arctan(x_i^2)$ are for state preparation and come directly from the input classical data. The data encoding part should be designed with respect to the problem of interest and plays a crucial role in the overall architecture [63]. Potential quantum advantage depends heavily on the encoding scheme together with the hardware limitations incorporated in the design.

The second part is the *variational* part, which consists of CNOT gates used to entangle quantum states from each qubit and $R(\alpha, \beta, \gamma)$ representing the general single qubit unitary gate with three parameters α_i, β_i and γ_i to be learned (Dashed block in figure 3). These circuit parameters can be regarded as the weights in the classical neural networks. In principle, one can repeat this variational block as many times as desired depending on the model complexity required for the machine learning problem of concern. In our experiments for image classification, we find that a total of four blocks is sufficient to produce optimal results.

The final part is the *measurement* part which will output the Pauli-Z expectation values via multiple runs of the quantum circuit. The retrieved values (logits) will go through classical processing such as softmax to generate the *probability* of each possible class. The quantum measurement would be performed on first k qubits where k is the number of classes.

Finally, we note that with a single variational block ($n = 1$), the VQC can be efficiently simulated classically through the addition of ancilla and post-selection. Therefore, for the VQC to be a quantum model, instead of a quantum-inspired model, the number of the variational blocks must be larger than one ($n > 1$).

4. Hybrid TN-VQC architecture

Figure 4 shows the architecture of the hybrid TN-VQC model. The input image of $N = 28 \times 28 = 784$ pixels from MNIST or Fashion-MNIST is flattened into a 784-dimensional vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, and each component is normalized such that $x_i \in [0, 1]$. The vector is mapped to a product state using the feature map [26]

$$\mathbf{x} \to |\Phi(\mathbf{x})\rangle = \begin{bmatrix} x_1 \\ 1 - x_1 \end{bmatrix} \otimes \begin{bmatrix} x_2 \\ 1 - x_2 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} x_N \\ 1 - x_N \end{bmatrix}, \tag{3}$$

and further processes by the MPS to generate a compressed representation. The feature vector is then encoded into the quantum circuit using the variational encoding (see appendix A). At the end of the VQC, the quantum measurement would be performed to generate the logits for classification. Both the TN and VQC have tunable parameters, labeled as θ_1 and θ_2 respectively in figure 4, which are optimized via gradient descend methods. Gradients of the quantum circuit parameters are calculated using the *parameter-shift* method (see appendix B), which avoids the use of finite difference calculation. This method is similar to the computation of gradients in neural networks; therefore, the end-to-end training of this TN-VQC model follows the standard backpropagation method as in the training of deep neural networks, and no pre-trained classical model is needed.

Since the classical and quantum parts of the model can be trained simultaneously, it allows for more flexibility in terms of implementation on the quantum hardware. When more qubits are available, one simply increases the dimension of the feature vector out of the MPS to match the input of the VQC and retrain the model. On the other hand, the modular architecture also has the advantage that the classical and

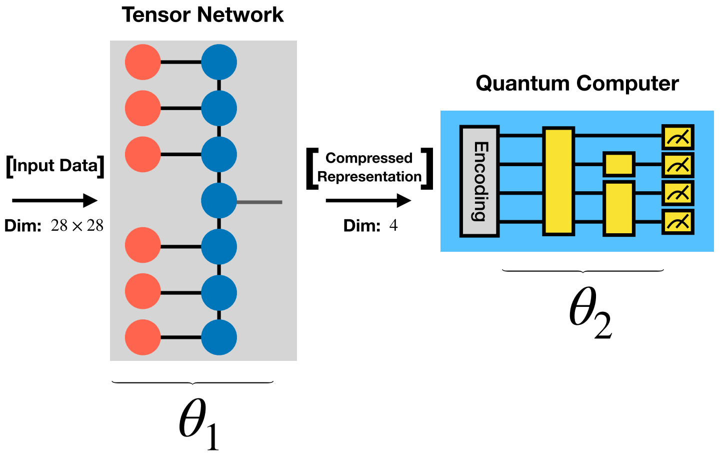

Figure 4. Hybrid TN-VQC architecture. The tunable parameters labeled with θ_1 and θ_2 are the parameters for the MPS and VQC respectively. See text for details.

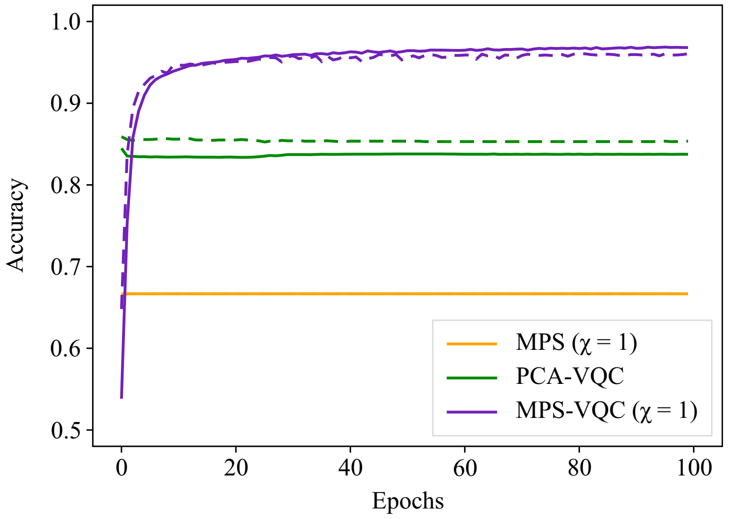

Figure 5. Binary classification of class 5 and 7. Results of the binary classification of Fashion-MNIST (class 5 vs 7). Solid line: training accuracy. Dashed line: test accuracy. For the MPS, the training and test accuracy are very close.

quantum parts can be reused independently. For example, it is possible to perform transfer learning by freezing the parameters in the MPS/VQC and training the other part to tackle different types of problems.

5. Experiments and results

We study the capabilities of the hybrid TN-VQC architecture by performing classification tasks on the standard benchmark dataset MNIST [64] and Fashion-MNIST [65]. Results for the binary classification of MNIST have been presented in [24]. Here, we perform ternary classification for MNIST and both binary and ternary classifications for Fashion-MNIST. As a baseline, we perform the same tasks on a hybrid PCA-VQC model, where the PCA serves as the simple feature extractor and the VQC as the classifier. As a comparison, we also present results using the MPS as a classifier to demonstrate the role of VQC in the workload. The computational tools we use for the simulation of variational quantum circuits and tensor networks are PyTorch [66], PennyLane [67] and Qulacs [68]. We include four variatioanl blocks ($n = 4$) in the VQC used in the experiments. Details of the simulations such as the hyperparameters and optimizers for each experiment are summarized in appendix C.

5.1. Binary classification

We perform binary classification of the Fashion-MNIST dataset (class 5 vs 7), which is a more difficult task than the binary classification of MNIST performed in [24]. The results from different models are shown in

Table 1. Block dependence of the PCA-VQC and the MPS-VQC models. Subscripts indicate the number of variational blocks in the VQC. See text for details.

Model	Test accuracy	Test loss
PCA-VQC$_2$	82.10%	0.5882
PCA-VQC$_4$	85.35%	0.4806
MPS-VQC$_2$	95.55%	0.3580
MPS-VQC$_4$	96.05%	0.3561

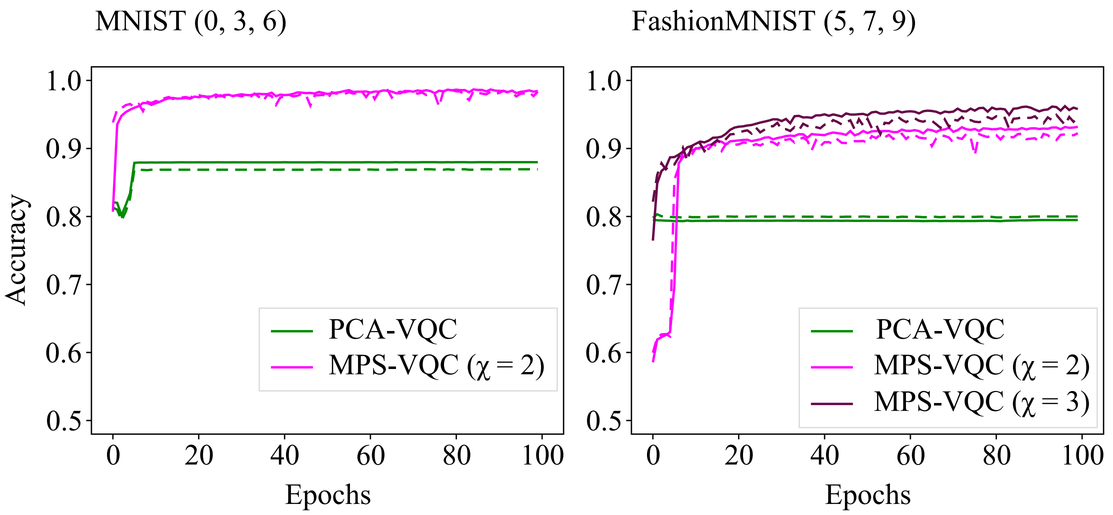

Figure 6. Ternary classification of MNIST and Fashion-MNIST. Results of the ternary classification of MNIST (class 0, 3, 6) and Fashion-MNIST (class 5, 7, 9). Solid line: training accuracy. Dashed line: test accuracy.

figure 5. For the MPS classifier and the MPS-VQC model, bond dimension $\chi = 1$ is demonstrated. At $\chi = 1$, the bond dimension is too small and the MPS classifier can only reach a test accuracy of 67%. However, the small bond dimension is enough for the MPS-VQC hybrid model to learn properly and reach a test accuracy 96.05%. As the number of parameters of the VQC part is far fewer than that of the MPS, this suggests that our VQC possess greater power in classification and dominates the workload. It is also clear that the MPS, compared to the PCA, serves as a better feature extractor for a VQC discriminator.

To further demonstrate this point, we compare the performance in test accuracy between the PCA-VQC and MPS-VQC model with a different number of variational blocks in the VQC (figure 4). The results are shown in table 1. For the PCA-VQC model, doubling the number of blocks from two to four leads to an noticeable increase in the test accuracy from 82.10% to 85.35%. On the other hand, for the MPS-VQC model, the increase is marginal, from 95.55% to 96.05%. We note that the reason why the MPS-VQC model can already achieve higher accuracy with two variational blocks compared to the PCA-VQC model is because the MPS contains trainable parameters, which is trained together with the VQC, whereas the capacity of PCA is fixed. Even when the VQC blocks are increased to four, the PCA-VQC$_4$ model still under-performs the MPS-VQC$_2$ model, again showing the MPS is a better feature extractor.

We note that in the case of $\chi = 1$ MPS-VQC, the overall effect of the MPS feature extractor is a simple rescaling of the input image; in contrast, for the case of $\chi > 1$, each tensor node in the MPS part is a matrix. In principle, increasing bond dimensions should provide more degrees of freedom. However, we find that $\chi = 2$ does not significantly improve the accuracy for this task, and for $\chi = 3$, the training fails to converge. This suggests that the classification task might be too simple for the proposed architecture. For a more difficult task, as demonstrated in the following, a higher bond dimension will be needed.

5.2. Ternary classification

In the ternary classification, we consider both the MNIST (class 0, 3, 6) and Fashion-MNIST (class 5, 7, 9) datasets. The results for the MPS-VQC model and the baseline model PCA-VQC are shown in figure 6. Since ternary classification is a more difficult task, a larger bond dimension is required for the MPS part. In terms of performance, all show that the MPS is superior to the PCA as a feature extractor. With $\chi = 2$, MPS-VQC is able to reach a test accuracy over 98% in MNIST and 92% in Fashion-MNIST. Furthermore, the representation power of an MPS feature extractor is tunable via χ, which is an advantage absent in PCA. In the case of Fashion-MNIST, we observe a 2% increase in test accuracy as the bond dimension increases, indicating a better data compression capability due to the increased representation power of the MPS. It is clear that for more complex classification problems, the performance of PCA-VQC should further deteriorate while that of MPS-VQC can be increased by increasing χ.

6. Conclusion

In this work, we present a hybrid quantum–classical classifier by integrating a quantum-inspired tensor network and a variational quantum circuit. Such a hybrid TN-VQC architecture enables researchers to build QML applications capable of dealing with larger dimensional inputs and potentially to implement these QML models on NISQ devices with a limited number of qubits and shallow circuit depth. We further demonstrate the superiority of this framework by comparing it with the baseline study of a PCA-VQC model on ternary classification tasks on the MNIST and Fashion-MNIST dataset as well as a binary classification task of the Fashion-MNIST dataset. One clear advantage is that the representation power of the trainable MPS feature extractor is tunable with the bond dimension of the tensors. Our results point to the future application of the hybrid TN-VQC model in different quantum machine learning scenarios and potentially implementation on NISQ devices. The extension of this architecture to more complicated datasets such as CIFAR-10 should further test the robustness and capability of the model. We note this requires more computing resources and better optimized simulators.

The core concept in the current design is to leverage classical computing resources (the MPS part) to assist the existing or near-term quantum computers in processing high-dimensional data. We use the cases of binary and ternary classification simply to demonstrate that such an algorithm indeed works. In these cases, we find that four input qubits are enough to yield good results. For more difficult tasks such as 10-class classification, it would be necessary to scale up the VQC with more than ten input qubits. Since the number of CNOT gates scales linearly with the number of qubits, the circuit depth would also increase accordingly. Obviously, simulating such a large non-Clifford circuit is not classically efficient. In contrast, a quantum device could readily handle such a calculation. In short, our hybrid algorithm is designed for quantum devices to perform multi-class classification on high-dimensional data that is classically hard to simulate, and the number of input qubits depends on the number of classes to be discriminated, not on the number of pixel inputs.

The proposed hybrid model can be integrated with various kinds of TNs or VQCs, given the suitable encoding methods. For example, one can replace the MPS by other TNs with a different entanglement structure such as TTN, MERA and PEPS, whose potential in the supervised learning context has been demonstrated [30, 69, 70]. They may serve also good feature extractors for datasets that contain special structure and correlations. Another way to build a more sophisticated feature extractor is to stack multiple TN layers. It has been shown that stacking more layers in classical deep neural networks can increase the model performance [71, 72]. Attempts to build a deep convolutional tensor network [73] do not show the same improvement generally observed in classical deep neural networks. We can also replace the simple VQC in our architecture with novel VQC architectures such as the quantum convolutional neural networks [11, 74–80]. How to apply these ideas to improve the performance of the current model is worth further investigation.

Data availability statement

The data that supports the findings of this study is available from the corresponding author upon reasonable request.

Acknowledgment

This work is supported (in part) by the U.S. DOE under Grant No. DE-SC-0012704 and the BNL LDRD No. 20-024 and Ministry of Science and Technology (MOST) of Taiwan under Grant Nos. 108-2112-M-002-020-MY3 and 107-2112-M-002-016-MY3. We acknowledge the anonymous referee for the comment on the simulability of the VQC.

Appendix A. Encoding into quantum states

In our hybrid framework, the outputs from the classical parts need to be encoded such that they can be used by the quantum circuit. A general N-qubit quantum state can be represented as:

$$|\psi\rangle = \sum_{(q_1,q_2,\ldots,q_N)\in\{0,1\}^N} c_{q_1,\ldots,q_N} |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle \otimes \cdots \otimes |q_N\rangle, \tag{A.1}$$

where $c_{q_1,\ldots,q_N}$ are complex numbers. They are *amplitudes* of each quantum state and $q_i \in \{0,1\}$. The square of the amplitude $c_{q_1,\ldots,q_N}$ represents the *probability* of measurement results in $|q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle \otimes \cdots \otimes |q_N\rangle$, and the total probability should sum to 1, i.e.

$$\sum_{(q_1,q_2,\ldots,q_N)\in\{0,1\}^N} ||c_{q_1,\ldots,q_N}||^2 = 1. \tag{A.2}$$

In this work, we choose the *variational encoding* method to encode our classical data into the quantum states. The initial quantum state $|0\rangle \otimes \cdots \otimes |0\rangle$ first undergoes the $H \otimes \cdots \otimes H$ operation to create the unbiased state $|+\rangle \otimes \cdots \otimes |+\rangle$, where H is the Hadamard gate. Consider a n-qubit system, the corresponding unbiased initial state is,

$$\begin{aligned}
(H|0\rangle)^{\otimes n} &= \underbrace{H|0\rangle \otimes \cdots \otimes H|0\rangle}_{n} \\
&= \underbrace{|+\rangle \otimes \cdots \otimes |+\rangle}_{n} \\
&= \underbrace{\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right] \otimes \cdots \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right]}_{n} \\
&= \left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right]^{\otimes n} \\
&= \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n} \\
&= \frac{1}{\sqrt{2^n}}(|0\rangle \otimes \cdots \otimes |0\rangle + \cdots + |1\rangle \otimes \cdots \otimes |1\rangle) \\
&= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle.
\end{aligned} \tag{A.3}$$

This initial quantum state will first go through the encoding part, which consists of R_y and R_z rotations. These rotation operations are parameterized by the input vector $\vec{x} = (x_1, x_2, \ldots, x_n)$. On the ith qubit with $i = 1, \ldots, n$, R_y rotates the state by an angle of $\arctan(x_i)$ and R_z by $\arctan(x_i^2)$. The encoded state is then processed with the variational quantum circuits with optimizable parameters, as shown in the dashed box in figure 3.

Appendix B. Calculation of gradients of quantum functions

Here the models are trained via gradient-descent methods widely used in training the deep neural network. To calculate the gradients with respect to the parameters of quantum circuits, we employ the *parameter-shift* method [13, 67, 81]. Given the knowledge of computing the expectation values of an observable $\hat{P}$ on quantum function,

$$f(\mathbf{x};\theta_i) = \left\langle 0 \middle| U_0^\dagger(\mathbf{x}) U_i^\dagger(\theta_i) \hat{P} U_i(\theta_i) U_0(\mathbf{x}) \middle| 0 \right\rangle = \left\langle x \middle| U_i^\dagger(\theta_i) \hat{P} U_i(\theta_i) \middle| x \right\rangle, \tag{B.1}$$

where $\mathbf{x}$ is the classical input vector (e.g. the output values from the PCA or MPS parts), $U_0(x)$ is the quantum encoding routine to prepare the classical value x into a quantum state, i is the circuit parameter index for which the gradient is to be evaluated, and $U_i(\theta_i)$ represents the single-qubit rotation generated by the Pauli operators X, Y, Z. It can be shown [13] that the gradient of this quantum function f with respect to the parameter θ_i is

$$\nabla_{\theta_i} f(x;\theta_i) = \frac{1}{2}\left[f\left(x;\theta_i + \frac{\pi}{2}\right) - f\left(x;\theta_i - \frac{\pi}{2}\right)\right]. \tag{B.2}$$

Appendix C. Computational details

Here we summarize the optimizer and hyperparameters in our experiments.

- MPS classifier: In experiments where MPS alone is used as a classifier, the optimizer is Adam [82] with a learning rate of 0.001 and batch size of 100.
- PCA-VQC model: In our baseline model, PCA is used to reduce the input dimension of $28 \times 28 = 784$ into a four-dimensional vector and is implemented with the Python package scikit-learn [83]. For the VQC classifier, the optimizer is RMSProp [84] with the hyperparameters: learning rate $= 0.01$, $\alpha = 0.99$ and $\varepsilon = 10^{-8}$.
- MPS-VQC hybrid model: In the hybrid TN-VQC architecture, the optimizer is Adam [82] with a learning rate of 0.0001 and batch size of 50. The VQC consists of four variational blocks (dashed-line box in figure 3). The number of quantum circuit parameters is $4 \times 4 \times 3 = 48$ in this setting. The quantum measurement at the final part of a VQC depends on the number of classes to be classified. For binary classification tasks, the first two qubits are measured, whereas for ternary classification, we perform quantum measurement on the first three qubits.

In all cases, we use cross entropy loss as our loss function.

ORCID iDs

Samuel Yen-Chi Chen https://orcid.org/0000-0003-0114-4826
Ying-Jer Kao https://orcid.org/0000-0002-3329-6018

References

[1] Harrow A W and Montanaro A 2017 *Nature* **549** 203–9
[2] Nielsen M A and Chuang I 2002 *Quantum Computation and Quantum Information* (New York: Cambridge University Press)
[3] Shor P W 1999 *SIAM Rev.* **41** 303–32
[4] Grover L K 1997 *Phys. Rev. Lett.* **79** 325
[5] Preskill J 2018 *Quantum* **2** 79
[6] Cerezo M *et al* 2020 arXiv:2012.09265
[7] Schuld M and Petruccione F 2018 *Supervised Learning With Quantum Computers* vol 17 (Berlin: Springer)
[8] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195–202
[9] Dunjko V and Briegel H J 2018 *Rep. Prog. Phys.* **81** 074001
[10] Chen S Y C, Yoo S and Fang Y L L 2020 arXiv:2009.01783
[11] Chen S Y C, Wei T C, Zhang C, Yu H and Yoo S 2020 arXiv:2012.12177
[12] Farhi E, Goldstone J and Gutmann S 2014 arXiv:1411.4028
[13] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 *Phys. Rev.* A **98** 032309
[14] Abohashima Z, Elhosen M, Houssein E H and Mohamed W M 2020 arXiv:2006.12270
[15] Schuld M, Bocharov A, Svore K M and Wiebe N 2020 *Phys. Rev.* A **101** 032308
[16] Chen S Y C, Yang C H H, Qi J, Chen P Y, Ma X and Goan H S 2020 *IEEE Access* **8** 141007–24
[17] Lockwood O and Si M 2020 Reinforcement learning with quantum variational circuit *Proc. Conf. on Artificial Intelligence and Interactive Digital Entertainment* vol 16 pp 245–51
[18] Wu S, Jin S, Wen D and Wang X 2020 arXiv:2012.10711
[19] Mari A, Bromley T R, Izaac J, Schuld M and Killoran N 2019 arXiv:1912.08278
[20] Östlund S and Rommer S 1995 *Phys. Rev. Lett.* **75** 3537–40
[21] Schollwöck U 2011 *Ann. Phys., NY* **326** 96–192
[22] Orús R 2014 *Ann. Phys., NY* **349** 117–58
[23] Huggins W, Patil P, Mitchell B, Whaley K B and Stoudenmire E M 2019 *Quantum Sci. Technol.* **4** 024001
[24] Chen S Y C, Huang C M, Hsing C W and Kao Y J 2020 Hybrid quantum-classical classifier based on tensor network and variational quantum circuit (arXiv:2011.14651)
[25] Orús R 2019 *Nat. Rev. Phys.* **1** 538–50
[26] Stoudenmire E M and Schwab D J 2016 Supervised learning with quantum-inspired tensor networks *Advances in Neural Information Processing Systems* vol 29 p 4799
[27] Levine Y, Yakira D, Cohen N and Shashua A 2018 Deep learning and quantum entanglement: fundamental connections with implications to network design *Int. Conf. on Learning Representations*
[28] Liu D, Ran S J, Wittek P, Peng C, García R B, Su G and Lewenstein M 2019 *New J. Phys.* **21** 073059
[29] Efthymiou S, Hidary J and Leichenauer S 2019 Tensor network for machine learning (arXiv:1906.06329)
[30] Reyes J and Stoudenmire M 2020 arXiv:2001.08286
[31] Han Z Y, Wang J, Fan H, Wang L and Zhang P 2018 *Phys. Rev.* X **8** 031012
[32] Cheng S, Wang L, Xiang T and Zhang P 2019 *Phys. Rev.* B **99** 155131
[33] Sun Z Z, Peng C, Liu D, Ran S J and Su G 2020 *Phys. Rev.* B **101** 075135
[34] Cohen N, Sharir O and Shashua A 2016 On the expressive power of deep learning: a tensor analysis *Conf. on Learning Theory* pp 698–728
[35] Levine Y, Sharir O, Cohen N and Shashua A 2019 *Phys. Rev. Lett.* **122** 065301
[36] Bengua J A, Phien H N and Tuan H D 2015 Optimal feature extraction and classification of tensors via matrix product state decomposition *2015 IEEE Int. Congress on Big Data* pp 669–72
[37] Novikov A, Podoprikhin D, Osokin A and Vetrov D 2015 Tensorizing neural networks (arXiv:1509.06569)

[38] Stokes J and Terilla J 2019 *Entropy* **21** 1236

[39] Miller J, Rabusseau G and Terilla J 2020 Tensor networks for probabilistic sequence modeling (arXiV:2003.01039)

[40] Chen J, Cheng S, Xie H, Wang L and Xiang T 2018 *Phys. Rev.* B **97** 085104

[41] Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O'brien J L 2014 *Nat. Commun.* **5** 4213

[42] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 *Nature* **567** 209–12

[43] Farhi E and Neven H 2018 arXiv:1802.06002

[44] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 *Quantum Sci. Technol.* **4** 043001

[45] Easom-McCaldin P, Bouridane A, Belatreche A and Jiang R 2020 arXiv:2008.12616

[46] Sarma A, Chatterjee R, Gili K and Yu T 2019 arXiv:1909.04226

[47] Dallaire-Demers P L and Killoran N 2018 *Phys. Rev.* A **98** 012324

[48] Stein S A, Baheri B, Tischio R M, Mao Y, Guan Q, Li A, Fang B and Xu S 2020 arXiv:2010.09036

[49] Zoufal C, Lucchi A and Woerner S 2019 *npj Quantum Inf.* **5** 1–9

[50] Situ H, He Z, Li L and Zheng S 2018 arXiv:1807.01235

[51] Nakaji K and Yamamoto N 2020 arXiv:2010.13727

[52] Lloyd S, Schuld M, Ijaz A, Izaac J and Killoran N 2020 arXiv:2001.03622

[53] Nghiem N A, Chen S Y C and Wei T C 2020 arXiv:2010.13186

[54] Jerbi S, Trenkwalder L M, Nautrup H P, Briegel H J and Dunjko V 2019 arXiv:1910.12760

[55] Bausch J 2020 arXiv:2006.14619

[56] Yang C H H, Qi J, Chen S Y C, Chen P Y, Siniscalchi S M, Ma X and Lee C H 2020 arXiv:2010.13309

[57] Sim S, Johnson P D and Aspuru-Guzik A 2019 *Adv. Quantum Technol.* **2** 1900070

[58] Lanting T *et al* 2014 *Phys. Rev.* X **4** 021041

[59] Du Y, Hsieh M H, Liu T and Tao D 2020 *Phys. Rev. Res.* **2** 033125

[60] Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A and Woerner S 2020 arXiv:2011.00027

[61] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 *Nature* **549** 242–6

[62] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 *New J. Phys.* **18** 023023

[63] Schuld M and Petruccione F 2018 Information encoding *Supervised Learning With Quantum Computers* (Cham: Springer International Publishing) pp 139–71

[64] LeCun Y 1998 (available at: http://yann.lecun.com/exdb/mnist/)

[65] Xiao H, Rasul K and Vollgraf R 2017 Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms (arXiv:1708.07747)

[66] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* pp 8026–37

[67] Bergholm V, Izaac J, Schuld M, Gogolin C, Blank C, McKiernan K and Killoran N 2018 arXiv:1811.04968

[68] Suzuki Y *et al* 2020 arXiv:2011.13524

[69] Stoudenmire E M 2018 *Quantum Sci. Technol.* **3** 034003

[70] Glasser I, Pancotti N and Cirac J I 2019 From probabilistic graphical models to generalized tensor networks for supervised learning (arXiv:1806.05964)

[71] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *Proc. Conf. on Computer Vision and Pattern Recognition* pp 1–9

[72] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition *Int. Conf. on Learning Representations*

[73] Blagoveschensky P and Phan A H 2020 arXiv:2005.14506

[74] Cong I, Choi S and Lukin M D 2019 *Nat. Phys.* **15** 1273–8

[75] Li Y, Zhou R G, Xu R, Luo J and Hu W 2020 *Quantum Sci. Technol.* **5** 044003

[76] Oh S, Choi J and Kim J 2020 arXiv:2009.09423

[77] Kerenidis I, Landman J and Prakash A 2019 arXiv:1911.01117

[78] Henderson M, Shakya S, Pradhan S and Cook T 2020 *Quantum Mach. Intell.* **2** 1–9

[79] Liu J, Lim K H, Wood K L, Huang W, Guo C and Huang H L 2019 arXiv:1911.02998

[80] Pesah A, Cerezo M, Wang S, Volkoff T, Sornborger A T and Coles P J 2020 arXiv:2011.02966

[81] Schuld M, Bergholm V, Gogolin C, Izaac J and Killoran N 2019 *Phys. Rev.* A **99** 032331

[82] Kingma D P and Ba J 2014 arXiv:1412.6980

[83] Pedregosa F *et al* 2011 *J. Mach. Learn. Res.* **12** 2825–30 (arXiv:1201.0490)

[84] Tieleman T and Hinton G 2012 Lecture 6.5—RmsProp: divide the gradient by a running average of its recent magnitude *COURSERA: Neural Networks for Machine Learning*