Quantum Machine Intelligence (2025) 7:57
https://doi.org/10.1007/s42484-025-00281-5

RESEARCH ARTICLE

Tensor ring optimized quantum-enhanced tensor neural networks

Debanjan Konar[1] · Dheeraj Peddireddy[1] · Bijaya K. Panigrahi[2] · Vaneet Aggarwal[1]

Received: 16 September 2024 / Accepted: 10 April 2025

Abstract

Quantum machine learning researchers often rely on incorporating tensor networks (TN) into deep neural networks (DNN) and variational optimization. However, the standard optimization techniques used for training the contracted trainable weights of each model layer suffer from the correlations and entanglement structure between the model parameters in classical implementations. To address this issue, a multi-layer design of a tensor ring optimized variational quantum learning classifier (Quan-TR) comprising cascading entangling gates replacing the fully connected (dense) layers of a TN is proposed, and it is referred to as tensor ring optimized quantum-enhanced tensor neural networks (TR-QNet). TR-QNet parameters are optimized using the stochastic gradient descent algorithm on qubit measurements. The proposed TR-QNet is evaluated on three distinct datasets, namely Iris, MNIST, and CIFAR-10, to demonstrate the enhanced precision achieved for binary classification. In quantum simulations, the proposed TR-QNet achieves promising precision of 94.5%, 86.16%, and 83.54% on the Iris, MNIST, and CIFAR-10 datasets. Benchmark studies have been conducted on state-of-the-art quantum and classical implementations of TN models to show the efficacy of the proposed TR-QNet. Moreover, the scalability of TR-QNet highlights its potential for exhibiting in deep learning applications on a large scale. The PyTorch implementation of TR-QNet is available on Github https://github.com/konar1987/TR-QNet/.

Keywords Quantum computing · Tensor networks · IBM quantum computer · Qubit

1 Introduction

Deep learning is a very effective and widely used machine learning method, which has shown remarkable performance in various tasks, including recognition, classification, regression, and clustering (Lathuiliére et al. 2020; Li et al. 2019; He et al. 2016; Peng et al. 2020). Recent years have witnessed the exploitation of quantum machine learning (Biamonte et al. 2017), a new computational paradigm that blends quantum computing and machine learning. It employs quantum parallelism and non-classical connections, such as quantum entanglement, to possibly speed up or revolutionize existing classical algorithms (Arute et al. 2019). Notably, the convergence of these disciplines can result in synergistic improvements and new views on a wide range of difficult challenges (Xiao et al. 2022). Combining physics principles and classical machine learning approaches has shown significant promise in tackling quantum computing issues (Qiu et al. 2022). The researchers demonstrated that the trainable weights of neural networks strongly correlate with many-body wave functions (Gutiérrez and Mendl 2022; Pescia et al. 2022). Furthermore, ideas for identifying phase transitions in quantum many-body systems using fully connected neural networks (FCNN) and convolutional neural networks (CNN) have been examined, with encouraging results (Carrasquilla and Melko 2017; Rem et al. 2019; Zhang et al. 2019).

Deep neural networks (DNN) have extremely high spatial and temporal complexity levels owing to densely stacked layers containing large-scale matrix multiplications. Hence, DNN often suffers from a long training time, which requires

✉ Debanjan Konar
konar.debanjan@ieee.org

Dheeraj Peddireddy
dpeddire@purdue.edu

Bijaya K. Panigrahi
bkpanigrahi@ee.iitd.ac.in

Vaneet Aggarwal
vaneet@purdue.edu

[1] School of Indusrtrial Eng. and Purdue Quantum Science & Engineering Institute, Purdue University, 315 Grant St,, West Lafayette, IN 47907, USA

[2] Department of Electrical Eng., Indian Institute of Technology Delhi, IIT Delhi Main Rd, New Delhi 110016, Delhi, India

Published online: 22 April 2025

considerable memory to infer. Furthermore, substantial weight redundancy in DNN has been demonstrated (Zeiler and Fergus 2014), demonstrating the possibility of condensing DNN while preserving performance. As a result, a variety of compression approaches have been devised, including pruning (Molchanov et al. 2019), quantization (Yang et al. 2019), and low-rank decomposition (Pan et al. 2019). Applying tensor networks (TN) to DNN to generate tensor neural networks (TNN) is one of them since TNN retains outstanding potential to approximate original weights with fewer parameters (Panagakis et al. 2021), particularly involving reconstruction of convolutional and fully connected layers using a range of tensor decomposition (TD) formats (Hayashi et al. 2019). However, the scalability of DNN is hindered when a substantial number of neurons are taken into account, thereby restricting the feasible number of layers. This is primarily due to the time-consuming training process and the need for a lot of memory to store the large-weight matrices. The accuracy and effectiveness of the DNN model will suffer with an increase in the hidden layers if the parameters for such large-weight matrices are not optimized. Therefore, decreasing the number of model parameters is imperative to maintain accuracy. However, the present hardware used to train neural networks significantly restricts their scale and usefulness. These concerns have gained significance due to the imminent approach of physical limitations to impede the progress of performance enhancements in deep classical neural networks.

In contemporary times, a correlation has been established between TN and FCNN, by which the former serves as an effective ansatz for representing quantum many-body wave functions (Verstraete et al. 2008; Orús 2014). As a result, it is possible to substitute TN for these weights and rely on variational optimization techniques to train them (Stoudenmire and Schwab 2016). In recent times, a multitude of efficient TN-based algorithms for classification (Convy and Whaley 2022), anomaly detection (Fanaee-T and Gama 2016; Li et al. 2011), segmentation (Konar et al. 2024), and clustering (Stoudenmire 2018) have been proposed. In addition to their capacity for effective expression, TN offers streamlined methodologies to compress data through tensor factorization techniques (Panagakis et al. 2021; Bahri et al. 2019). For example, it is possible to significantly reduce the number of parameters in neural network models by retaining only the most significant degrees of freedom and discarding those that exhibit lower correlations. TNN (Konar et al. 2024) and variational tensor deep neural networks (Hayashi et al. 2019; Phan et al. 2020) are instances of neural networks that rely on the structures of the tensor network to replace the weight tensors of the hidden layers. This is achieved by applying singular value decomposition (SVD) methods. Recent research studies have validated that TNN exhibits superior performance and precision despite having a limited parameter space compared to conventional artificial neural networks (ANN) (Panagakis et al. 2021; Novikov et al. 2015). The low-rank tensor approximation has been extensively studied in the literature for effective model reduction, low generative error, and high prediction speed (Peddireddy et al. 2023; Wang et al. 2018; Song et al. 2020).

Recently, quantum neural networks (QNN) have emerged as a potential contender to circumvent problems and facilitate the training of DNN (Konar et al. 2016; Zhao and Gao 2021; Rebentrost et al. 2018; Konar et al. 2020; Li et al. 2020; Konar et al. 2022a, b; Cerezo et al. 2021; Abbas et al. 2021; Beer et al. 2020). Quantum states are mathematical entities of quantum systems compatible with higher-order tensors (Orús 2019). Thus, traditional computers may use TNN as simulators to emulate genuine quantum circuits (Pan and Zhang 2022; Huggins et al. 2019). Some particular TNN can be realized on compact, near-term quantum devices using quantum computing's parallelism (Cohen and Shashua 2016). Rather than the more broad paradigm of TN-based quantum circuit modeling, quantum circuit simulation on TNN focuses on the functions of TN as bridges between traditional ANN and QNN. The authors recently proposed a similar tensor ring parameterized variational quantum circuit (TR-VQC) (Peddireddy et al. 2023). However, TR-VQC suffers from directly reduced input features due to a few available qubits and the limited entanglement between the parameters, resulting in barren plateaus (McClean et al. 2018).

1.1 Motivation

Most contemporary advances in tensor neural networks (TNN) involve tensorization solely at the level of hidden layers about trainable weights (Kossaifi et al. 2017; Hayashi et al. 2019; Phan et al. 2020; Huang et al. 2021; Jahromi and Orus 2023). Training a model typically involves optimizing each layer's contracted trainable weights using established optimization techniques like gradient descent (Konar et al. 2024; Wang et al. 2018). The outcome of this is an adaptable architecture for TNN that can be effectively trained for a substantial number of neurons and layers. Furthermore, in TNN, the standard optimization techniques used to train the contracted trainable weights of each model layer suffer from correlations and the structure of entanglement between model parameters in classical implementations. To address this issue, a multi-layer design of a tensor ring optimized variable quantum learning classifier (Quan-TR) is proposed comprising cascading entangling gates that replace the fully connected (dense) layers of a TN. The variational algorithm employs a method of local gradient descent, incorporating tensor gradients. This motivates us to propose hybrid TNN models incorporating both the tensor and quantum layers.

The training algorithm used in our study offers valuable insight into the entanglement structure of trainable weights for fully connected layers of TNN. Nevertheless, it helps to clarify the expressive power of a quantum neural state.

1.2 Primary contributions and novelty

Considering the neural network's entanglement structure, a novel multi-layer design of a tensor ring-optimized variational quantum learning classifier (Quan-TR) with cascading entanglement gates is introduced in the proposed hybrid quantum-enhanced TNN model (TR-QNet). Furthermore, the accuracy and efficiency of our TR-QNet model are evaluated using numerical data and image classification on various datasets. The present study exhibits a tripartite novelty, summarized as follows:

1. Our study presents for the first time a novel tensor-ring approximated quantum-enhanced tensor neural network (TR-QNet) comprising classical tensor layers followed by TR-approximated quantum layers for data and image classification. The proposed TR-QNet incorporates the novel multi-layer design of Quan-TR, replacing the fully connected Soft-max layers in TNN, distinguishing it from the state-of-the-art TN models (Hayashi et al. 2019; Phan et al. 2020; Huang et al. 2021; Jahromi and Orus 2023).
2. In addition, the quantum layers (Quan-TR) of the proposed TR-QNet model incorporate a cascading of quantum entangling gates, leading to the elimination of barren plateaus. This is demonstrated by the convergence of the training loss of the proposed TR-QNet model.
3. Compared with the classical TN model, the binary class classification accuracy of TR-QNet is improved by 10.53%, 7.28%, and 12% on the Iris (Fisher 1936), MNIST (LeCun et al. 1998), and CIFAR-10 (Krizhevsky et al. 2012) datasets, respectively. This approach presents a distinctive and innovative effort towards expediting advancements in resolving computer vision issues through deep quantum learning.

The subsequent sections of this manuscript are organized as follows: The basic notations of TN are introduced in Section 2. Section 3 explains the proposed quantum-enhanced tensor neural network architecture, which includes an overview of classical TNN and tensor ring optimized variational quantum circuit (Quan-TR). Section 4 contains the datasets, simulation settings, and simulation results. Section 5 elucidates the efficacy of the TR-QNet model and underscores its constraints. Finally, the concluding remarks and future research directions are discussed in Section 6. Appendix section provides the convergence of the proposed TR-QNet.

2 Tensor networks: notations and definitions

Tensor networks are a series of tensor factorization models that use several sparsely linked lower-order tensors, such as the TT decomposition (Oseledets 2011; Dian et al. 2019), TR decomposition (Asante-Mensah1 et al. 2021; Yuan et al. 2020; Yu et al. 2021), and the Hierarchical Tucker (HT) decomposition (Uschmajew and Vandereycken 2013; Grasedyck and Hackbusch 2011), to represent a high-order tensor. The topics most pertinent to this article, TR decomposition and the contraction operator, are briefly introduced in the following subsections.

2.1 Tensor ring

The circular multi-linear product over a succession of third-order core tensors, as shown in Fig. 1, is defined as follows:

$$\begin{aligned}\mathcal{T}(y_1, y_2, \cdots, y_\tau) &= T_R\left(\prod_{\upsilon=1}^{\tau} \mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}}\right)\\ &= \sum_{x_1,\cdots,x_\tau=1}^{\mathcal{L}_1,\cdots,\mathcal{L}_\tau} \prod_{\upsilon=1}^{\tau} \mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}}\\ &= \sum_{x_1,\cdots,x_\tau=1}^{\mathcal{L}_1,\cdots,\mathcal{L}_\tau} \mathcal{X}(1)^{y_1}_{x_1x_2}\cdots\mathcal{X}(\tau)^{y_\tau}_{x_\tau x_1} \qquad (1)\end{aligned}$$

where $\mathcal{X}(\upsilon) \in \mathbb{R}^{\mathcal{L}_\upsilon \times \mathcal{L}_{\upsilon+1} \times \mathcal{I}_\upsilon}$, $\upsilon \in [\tau]$ and $\mathcal{L}_{\tau+1} = \mathcal{L}_1$. The TR decomposition expresses a τ^{th} order tensor $\mathcal{T} \in \mathbb{R}^{\mathcal{I}_1 \times \cdots \times \mathcal{I}_\tau}$ (Qiu et al. 2022).

Using τ tensors, each with bond dimension $\mathcal{B}$, indicated by $\mathcal{X}(\upsilon)$, the parameterization of the tensor ring describes a quantum state $|\psi\rangle$ as follows Wang et al. (2018):

$$\begin{aligned}|\psi\rangle = &\sum_{y_1,\cdots,y_\tau=0}^{1} \sum_{x_1,\cdots,x_\tau=1}^{\mathcal{B}} \mathcal{X}(1)^{y_1}_{x_1x_2}\mathcal{X}(2)^{y_2}_{x_2x_3}\\ &\cdots\mathcal{X}(\tau)^{y_\tau}_{x_\tau x_1}|y_1,\cdots y_\tau\rangle \qquad (2)\end{aligned}$$

The physical indices $y_\upsilon \in \{0, 1\}$ span the $2^{\mathcal{T}}$-dimensional Hilbert space, while the bond indices $x_\upsilon \in \{1, \cdots \mathcal{B}_\upsilon\}$ control the maximum amount of entanglement captured by the tensor ring, also known as tensor rank.

2.2 Tensor ring contraction

Tensor ring (TR) contraction, or the multi-linear product, is a basic operation in the TR decomposition process represented by a linked edge connecting two nodes. More specifically, the contraction of two TR-cores, $\mathcal{X}(\upsilon)$ and $\mathcal{X}(\upsilon+1)$, with a size

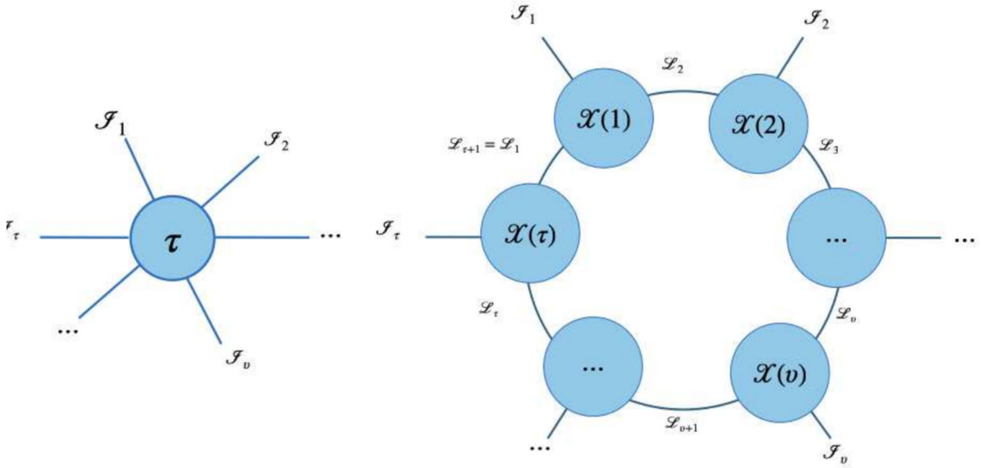


Fig. 1 Illustration of tensor ring (TR) structure

$x_{\upsilon+1}$ common mode, may be computed using Yu et al. (2021) as follows:

$$\mathcal{Q}(\upsilon, \upsilon+1)^{y_\upsilon y_{\upsilon+1}}_{x_\upsilon x_{\upsilon+2}} = \sum_{x_{\upsilon+1}=1}^{\mathcal{L}_{\upsilon+1}} \mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} \mathcal{X}(\upsilon+1)^{y_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} \tag{3}$$

where $\mathcal{Q}$ is the contracted tensors of $\mathcal{X}(\upsilon)$ and $\mathcal{X}(\upsilon+1)$.

Similarly, the contraction of three TR cores, $\mathcal{X}(\upsilon-1)$, $\mathcal{X}(\upsilon)$, and $\mathcal{X}(\upsilon+1)$, with common modes x_υ and $x_{\upsilon+1}$, may be computed as follows:

$$\begin{aligned} &\mathcal{Q}(\upsilon-1, \upsilon, \upsilon+1)^{y_{\upsilon-1} y_\upsilon y_{\upsilon+1}}_{x_{\upsilon-1} x_{\upsilon+2}} \\ &= \sum_{x_\upsilon, x_{\upsilon+1}=1}^{\mathcal{L}_\upsilon, \mathcal{L}_{\upsilon+1}} \mathcal{X}(\upsilon-1)^{y_{\upsilon-1}}_{x_{\upsilon-1} x_\upsilon} \mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} \mathcal{X}(\upsilon+1)^{y_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} \end{aligned} \tag{4}$$

Moreover, TR-core contraction along common modes x_1, x_2, $\cdots$, x_τ can also depict TR decomposition as follows:

$$\begin{aligned} \mathcal{T}(y_1, y_2, \cdots, y_\tau) &= \mathcal{Q}(1, 2, \cdots, \tau)^{y_1, y_2, \cdots, y_\tau}_{x_1, x_2, \cdots, x_\tau} \\ &= \sum_{\upsilon=1}^{\tau} \sum_{x_\upsilon=1}^{\mathcal{L}_\upsilon} \mathcal{X}(1)^{y_1}_{x_1 x_2}, \cdots, \mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}}, \\ &\quad \cdots, \mathcal{X}(\tau)^{y_\tau} x_\tau x_1 \\ &= \sum_{x_\upsilon=1}^{\mathcal{L}_\upsilon} \prod_{\upsilon=1}^{\tau} \mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} \end{aligned} \tag{5}$$

Three states, namely super-critical ($\mathcal{I}_\upsilon < \mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1}$), critical ($\mathcal{I}_\upsilon = \mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1}$), and sub-critical ($\mathcal{I}_\upsilon > \mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1}$), can be distinguished via TR decomposition. The full Tucker rank tensor data have critical or supercritical TR states (Qiu et al. 2022).

Definition 2.1 (Canonical mode-υ unfolding of a tensor $\mathcal{T}$ Kolda and Bader 2009) Suppose $\mathcal{T} \in \mathbb{R}^{\mathcal{I}_1 \times \cdots \times \mathcal{I}_\tau}$ is a τ^{th}-order tensor. Its canonical mode-υ unfolding is represented as $\mathscr{T}_{(\upsilon)} \in \mathbb{R}^{\mathcal{I}_\upsilon \times \prod_{j \neq \upsilon}^{\tau} \mathcal{I}_j}$, matrixed the tensor $\mathcal{T}$ by placing the υ^{th} mode in the rows of the matrix and the other modes in the columns, keeping the original order as follows:

$$\mathscr{T}_\upsilon(x_\upsilon, \overline{x_1, x_2, \cdots, x_{\upsilon-1} x_{\upsilon+1}, \cdots, x_\tau}) = \mathcal{T}(x_1, x_2, \cdots, x_\tau) \tag{6}$$

where $\overline{x_1, x_2, \cdots, x_\tau} = x_1 + (x_2-1)\mathcal{I}_1 + (x_3-1)\mathcal{I}_1\mathcal{I}_2 + \cdots + (x_\tau - 1)\mathcal{I}_1, \cdots, \mathcal{I}_{\tau-1}$ and $x_\upsilon \in [\mathcal{I}_\upsilon]$, $\upsilon \in [\tau]$.

Definition 2.2 (First-υ mode unfolding of a tensor $\mathcal{T}$ Oseledets 2011) Suppose $\mathcal{T} \in \mathbb{R}^{\mathcal{I}_1 \times \cdots \times \mathcal{I}_\tau}$ is a τ^{th}-order tensor. Its first-υ mode unfolding is represented as $\mathscr{T}_{[\upsilon]} \in \mathbb{R}^{\prod_{i=1}^{\upsilon} \mathcal{I}_i \times \prod_{j=\upsilon+1}^{\tau} \mathcal{I}_j}$ as follows:

$$\begin{aligned} &\mathscr{T}_\upsilon(\overline{x_1, x_2, \cdots, x_\upsilon}, \overline{x_{\upsilon+1}, \cdots, x_\tau}) \\ &= x_1 + (x_2 - 1)\mathcal{I}_1 + (x_3 - 1)\mathcal{I}_1\mathcal{I}_2 + \cdots + (x_\upsilon - 1) \prod_{i=1}^{\upsilon-1} \mathcal{I}_i \\ &\quad + x_{\upsilon+1} + (x_{\upsilon+2} - 1)\mathcal{I}_{\upsilon+1} + (x_{\upsilon+3} - 1)\mathcal{I}_{\upsilon+1}\mathcal{I}_{\upsilon+2} \\ &\quad + \cdots + (x_\tau - 1) \prod_{j=\upsilon+1}^{\tau-1} \mathcal{I}_j = \mathcal{T}(x_1, x_2, \cdots, x_\tau) \end{aligned} \tag{7}$$

Definition 2.3 (Twofold unfolding: left and right Holtz et al. 2012) Assuming $\mathcal{T}_{(3)} \in \mathbb{R}^{\mathcal{L}_\upsilon \times \mathcal{L}_{\upsilon+1} \times \mathcal{I}_\upsilon}$ to be a third-order tensor, the matrix resulting from assigning row indices to the first two modes and column indices to the third modes is the left unfolding.

$$L(\mathcal{T}) = [\mathcal{T}_{(3)}]^T \in \mathbb{R}^{(\mathcal{L}_\upsilon \times \mathcal{I}_\upsilon) \times \mathcal{L}_{\upsilon+1}} \tag{8}$$

Similarly, right unfolding is defined as follows:

$$R(\mathcal{T}) = \mathcal{T}_{(1)} \in \mathbb{R}^{\mathcal{L}_\upsilon \times (\mathcal{I}_\upsilon \times \mathcal{L}_{\upsilon+1})} \tag{9}$$

Definition 2.4 (Circular Mode Unfolding Yuan et al. 2020) Suppose $\mathcal{T} \in \mathbb{R}^{\mathcal{I}_1 \times \cdots \times \mathcal{I}_\tau}$ is a τ^{th}-order tensor. Its circular mode-(υ, i) unfolding is a matrix designated as $\mathscr{T}_{(\upsilon,i)} \in \mathbb{R}^{\prod_{k=t+1}^{\upsilon-1} \mathcal{I}_i \times \prod_{j=\upsilon}^{t} \mathcal{I}_j}$ and

$$t = \begin{cases} \upsilon + i - 1, & \text{if } \upsilon + i \leqslant \tau \\ \upsilon + i - 1 - \tau, & \text{otherwise.} \end{cases} \tag{10}$$

Also,

$$\mathscr{T}_{(\upsilon,i)}(\overline{x_{t+1}, x_{t+2}, \cdots, x_{\upsilon-1}}, \underbrace{\overline{x_\upsilon, \cdots, x_t}}_{i \text{ indexes}}) = \mathcal{T}(x_1, x_2, \cdots, x_\tau) \tag{11}$$

Lemma 2.5 *Full-Tucker-rank tensor data have critical or supercritical TR approximated quantum states.*

Proof of Lemma 2.5: Considering a given τ^{th}-order tensor, $\mathcal{T} \in \mathbb{R}^{\mathcal{I}_1 \times \cdots \times \mathcal{I}_\tau}$ with TR decomposition as follows: $\mathcal{T}(y_1, y_2, \cdots, y_\tau) = T_R(\mathcal{X}(1)^{y_1}_{x_1 x_2} \cdots \mathcal{X}(\tau)^{y_\tau}_{x_\tau x_1})$.

If $\mathcal{I}_\upsilon > \mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1}$, SVD is performed on a resultant tensor $\mathcal{Q}$ and on reshaping it as $(y'_\upsilon \times x_\upsilon) \times (y'_{\upsilon+1} \times x_{\upsilon+1})$ as follows:

$$SVD(\mathcal{Q}(\upsilon, \upsilon + 1)^{y_\upsilon y_{\upsilon+1}}_{x_\upsilon x_{\upsilon+2}}) = \sum_{x_\upsilon} \mathcal{V}^{y'_\upsilon}_{x_\upsilon x_{\upsilon+1}} \mathcal{S}_{x_{\upsilon+1}} \mathcal{M}^{y'_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} \tag{12}$$

where $\mathcal{V}$ and $\mathcal{M}$ comprise orthogonal vectors, and $\mathcal{S}_{x_{\upsilon+1}}$ is composed of singular values of matrix $\mathcal{Q}$. The resultant matrix is called $\mathcal{S}'_{x_{\upsilon+1}}$. We then trim the $\mathcal{S}_{x_\upsilon}$ matrix to retain just the N biggest singular values out of $2N$. The sole reason for truncating $\mathcal{V}$ and $\mathcal{M}$ is to preserve the orthogonal vectors corresponding to the N biggest singular values. We can reconstruct the core tensors using a canonical mode-2 folding operation $fold_2(.)$ (Qiu et al. 2022) as follows:

$$\mathcal{Q}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} = fold_2(\mathcal{V}^{y'_\upsilon}_{x_\upsilon x_{\upsilon+1}} \mathcal{S}'_{x_{\upsilon+1}}) \tag{13}$$

Here, $\mathcal{Q}(\upsilon) \in \mathbb{R}^{\mathcal{L}_\upsilon \times \mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1} \times \mathcal{L}_{\upsilon+1}}$, $\upsilon \in [\tau]$.

If $y_\upsilon \leqslant \mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1}$, let us assume $\mathcal{M}^{y'_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} = I^{y'_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}}$. Consequently, one can write each core tensor equivalently as follows:

$$\mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} = \mathcal{Q}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} \times_2 \mathcal{M}^{y'_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}}, \upsilon \in [\tau]. \tag{14}$$

and $\mathcal{T}'(y_1, y_2, \cdots, y_\tau) = T_R(\mathcal{Q}(1)^{y_1}_{x_1 x_2}, \cdots, \mathcal{Q}(\tau)^{y_\tau}_{x_\tau x_1}) \in \mathbb{R}^{D_1 \times \cdots D_\upsilon \cdots \times D_\tau}$. Here, $D_\upsilon = \mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1}$ if $\mathcal{L}_\upsilon \mathcal{L}_{\upsilon+1} < \mathcal{I}_\upsilon$; $\mathcal{I}_\upsilon$ otherwise. Also,

$$\begin{aligned} \mathcal{T}(y_1, y_2, \cdots, y_\tau) = \mathcal{T}'(y_1, y_2, \cdots, y_\tau) \times_1 \mathcal{M}^{y'_1}_{x_1 x_2} \\ \cdots \times_\tau \mathcal{M}^{y'_\tau}_{x_\tau x_1} \end{aligned} \tag{15}$$

Since the factor matrix $\mathcal{Q}^{y'_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}}$ represents the identity matrix, the tensor data with a supercritical or critical quantum state estimated by TR is a full Tucker rank tensor.

3 Quantum-enhanced tensor neural network architecture

The TR-QNet model is a novel proposed framework that combines classical TN and quantum layers (Quan-TR) with tensor ring-parameterized inputs and cascading of entangling gates. Our hybrid TR-QNet model exhibits a relationship between the TNN and VQC frameworks optimized by a TR structure, as shown in Fig. 2, allowing the full input features to be fed through TNN layers with minimal loss of information. Our TR-QNet model architecture incorporates a TNN with multiple hidden layers. It introduces a multi-layer tensor-ring optimized variational quantum learning classifier with cascading entangling gates to efficiently address quantum entanglement among model parameters. This approach replaces the conventional soft-max layer typically employed at the end of TNN models. A classical pooling layer is incorporated in integrating the TNN model and Quan-TR of the proposed TR-QNet architecture to match the dimension of the input of TNN and the input of Quan-TR.

The VQC-based training algorithm resembling DMRG (White 1992) enables a straight-forward entanglement of the entanglement spectrum of the MPO's (Panagakis et al. 2021) trainable weights, thereby facilitating a lucid comprehension of the correlations within the parameters of our TR-QNet model. One can evaluate the MPOs' entanglement structure and capacity as a quantum neural state through standard quantum information measures.

3.1 Tensor neural networks

A tensor neural network (TNN) is obtained after the tensorization of an ANN, enabling it to align with the size and dimensions of MPO weights (Verstraete et al. 2008; Orús 2014). The hidden layers of the proposed TR-QNet model can be reshaped into a rank-d_T tensor, possessing a dimension of $\mathcal{N}_T$, which can subsequently be contracted to form a TN

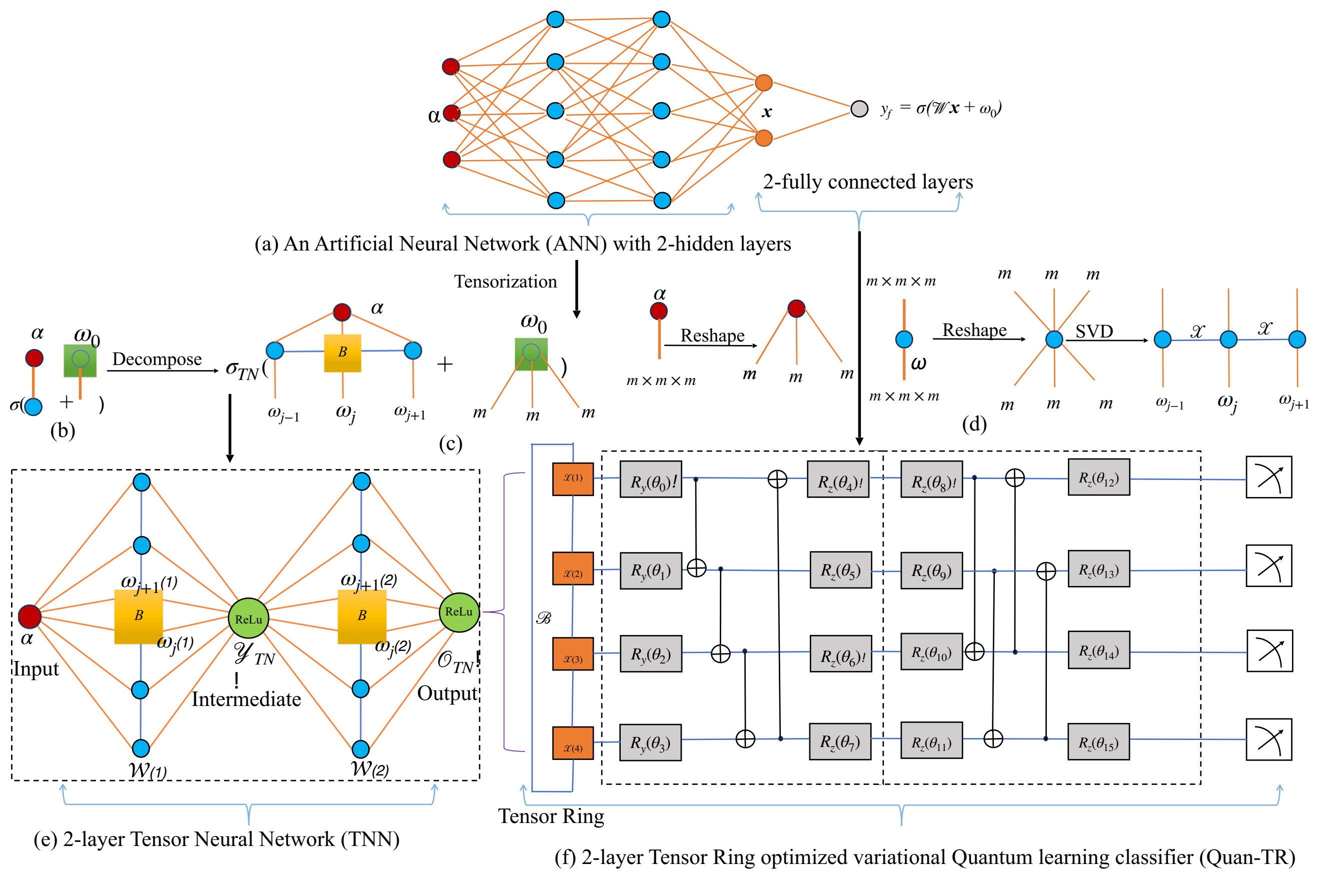


Fig. 2 A tensor ring optimized quantum-enhanced tensor neural network (TR-QNet) architecture with 4-qubits Quan-TR for tensorizing an **a** ANN with two hidden layers and two fully connected dense layers. The network prediction, y_f, is derived by feeding the model the input feature vector α as $y_f = \sigma(\mathcal{W}^2\sigma(\mathcal{W}^1\sigma(\mathcal{W}^0\alpha+w_0)+w_1)+w_2)$, where w_0, w_1, w_2 are bias vectors, and σ is the *ReLU* activation function; **b** the ANN's TN representation using matrix product states (MPS) and matrix product operators (MPO); **c** the resulting TNN with MPO trainable weights; **d** MPO decomposition of the weight matrix $\mathcal{W}$ performing singular value decomposition (SVD) and truncating the inconsequential singular values includes MPO factorization for a matrix $\mathcal{W}_{m^3\times m^3}$ followed by reshaping $\mathcal{W}$ into a rank-6 tensor and using a suitable SVD, matrix $\mathcal{W}$ may be represented as a 3-site MPO; **e** 2-layer TNN tensorizing the ANN using part **b**, **c**, and **d**; and **f** low-rank Quan-TR component employed in this proposed TR-QNet has three parts: TR encoding ($\mathcal{X}$), variational learning parameters, and quantum measurement. The cascading CNOT gates are preserved through the TR approximation relying on SVD. $\mathcal{R}_y(\omega)$ and $\mathcal{R}_z(\omega)$ are used for data encoding and measurements

layer. This TN layer comprises six matrix product operators (MPO) (Panagakis et al. 2021) weights, each having an input size of m. Features that cannot be factored to align with MPOs in the TN layers are transformed during the preprocessing stage of the training TNN to conform to the input size of the TN layer. A dense, non-trainable layer is added as a connecting layer preceding the tensor ring optimized variational quantum learning classifier (Quan-TR) layer to address the issue of reduction in the size of input data in the classical TNN model. The length of the input feature vector of Quan-TR is denoted by $\mathcal{N}_q$, while the output size of the contracted TNN layer is represented by $\mathcal{N}_t$. The contraction of two rank-2 tensors, $\mathcal{S}_{ij}$ and $\mathcal{V}_{jk}$, can be represented schematically by connecting the two tensors along their shared index j. Mathematically, the contraction operation is described as follows:

$$\mathcal{T}_{ik} = T_R(\mathcal{S}_{ij}\mathcal{V}_{jk}) = \sum_j \mathcal{S}_{ij}\mathcal{V}_{jk} \tag{16}$$

Here, T_R designates the trace over shared indices j.

A viable approach to intelligent data compression techniques that rely on TN and MPO decomposition to enhance the representation of weight matrices involves substituting weights with MPOs. The MPO form of the weight matrix of a hidden layer can be derived from the $\mathcal{W}$ matrix by applying SVD consecutively, as demonstrated in Fig. 2. The TN layers comprise a set of trainable weights denoted w_i represented by MPO. A bond tensor $B_{j,j+1}$ is obtained by contracting a pair of neighboring MPO tensors w_j and w_{j+1}, along their shared virtual dimension. Adjusting the input feature vector α to align with the dimensions of the MPO allows the network output $\mathcal{O}_{TN}$ to be derived through contraction of the resulting tensor network. The activation function σ_{TN} is applied to the result of a tensor contraction operation (Jahromi and Orus 2023) as follows:

$$\mathcal{O}_{TN} = \sigma_{TN}(T_R(\alpha_i, \alpha_j, \alpha_k, \cdots w_i, w_j, w_k, \cdots) + w_0) \tag{17}$$

The tensor contraction operation between the input tensor α and the weight tensor w and i, j, k, represents the tensor indices. It may be noted that the activation function σ_{TN} is applied element-wise to the matrix obtained from the tensor contraction, and it cannot be directly applied to individual MPO tensors separately due to the non-linearity introduced by the activation function. In the proposed TR-QNet, one approach involves contracting the features and tensor network layers (MPOs) before applying the activation function and reshaping the resulting tensor to match the inputs of the next layer. This process is repeated until the entire TNN network is contracted.

The concept of a TN layer involves using tensorization to adapt feature vectors to match MPO weights. When the feature vector can be factorized into integers, it allows for reshaping into a tensor that matches the dimensions required by the MPO weights, facilitating the creation of a TN layer with the appropriate input sizes. An illustration of a feature vector α with 64 entries is depicted in Fig. 2, showcasing its potential to be transformed into a rank-3 tensor each with a dimension of 4 that can be contracted to a TN layer with 3 MPO weights, each with an input size of 4. This process aligns the feature vector with the requirements of the TN layers, enabling seamless integration within the network architecture. In cases where features are not directly factorizable to align with the MPOs, preprocessing steps have been employed to adjust their sizes to match the TN layer input requirements. We introduce a non-trainable dense layer with dimensions $\mathcal{N}_F \times \mathcal{N}_T$ preceding the TNN. Here, $\mathcal{N}_F$ represents the size of the feature vector, while $\mathcal{N}_T$ corresponds to the input size required by the contracted TN layers. By incorporating this dummy dense layer, the network can effectively compensate for any size mismatch, ensuring compatibility and smooth operation within the TR-QNet model architecture.

3.2 Tensor ring optimized variational quantum learning classifier

The proposed TR-optimized variational quantum learning classifier (Quan-TR) with TNN is a hybrid classical-quantum algorithm combining tensor network elements and variational quantum circuits for data and image classification. The proposed Quan-TR introduces a multi-layer TR-optimized variational quantum learning classifier with cascading entangling gates to address quantum entanglement among model parameters efficiently, which is the primary distinction with our previously published TR-VQC (Peddireddy et al. 2023). The proposed Quan-TR framework consists of three main components: TR encoding, variational learning parameters, and measurement. TR encoding represents the quantum states in a compressed format. It leverages the TR structure, a TN with a specific hierarchical ring-like connectivity pattern. The TR approximation uses SVD to compress the quantum states while preserving important features. This approximation allows for efficient representation and manipulation of quantum states within Quan-TR, as shown in Fig. 3. In the proposed Quan-TR, single-qubit rotation gates $\mathcal{R}_y(\omega)$ and $\mathcal{R}_z(\omega)$ are used to represent rotations along the Y and Z axes, respectively. These rotation angles (ω) are learned during training to find optimal values that minimize the objective function. By combining TR encoding with variational learning parameters and measurement, the proposed TR-QNet architecture enables the training of quantum circuits for data and image classification. Within the proposed Quan-TR framework, using τ tensors, each with bond dimension $\mathcal{B}$, indicated by $\mathcal{X}(\upsilon)$, the tensor-ring approximation describes a quantum state $|\psi\rangle$ as follows Wang et al. (2018):

$$|\psi\rangle = \sum_{y_1,\cdots,y_\tau} \sum_{x_1,\cdots,x_\tau} \mathcal{X}(1)^{y_1}_{x_1x_2} \mathcal{X}(2)^{y_2}_{x_2x_3} \cdots \mathcal{X}(\tau)^{y_\tau}_{x_\tau x_1} |y_1, \cdots y_\tau\rangle \tag{18}$$

Here, the 2^τ-dimensional Hilbert space is spanned by the physical indices $y_\upsilon \in \{0, 1\}$, and the maximum amount of entanglement recorded by Quan-TR is controlled by the bond indices $x_\upsilon \in \{1, \cdots \mathcal{B}_\upsilon\}$. The entanglement captured by a tensor ring is controlled by the parameter $(\mathcal{B}_\upsilon)$. Allowing $\mathcal{B}_\upsilon$ increases enables the TR to capture more information about the quantum state, although at a computational cost. When designing the training algorithm, $\mathcal{B}_\upsilon$ is a critical hyperparameter choice along with parameters such as batch size and learning rate. We assume for all $\upsilon, \mathcal{B}_\upsilon = \mathcal{B}$, reducing the number of hyperparameters following Wang et al. (2018). This bond dimension is called the tensor ring rank.

The parameterization of the tensor ring of a 4-qubit state is illustrated in Fig. 2. In Quan-TR, each $\mathcal{X}(\upsilon)$ in TR represents a tensor of dimension $\mathcal{B} \times \mathcal{B} \times \mathcal{D}$, signifying the connections between the tensors in the tensor ring. The tensor, $\mathcal{X}(\upsilon)$, has three indices, two of which have a bond dimension $\mathcal{B}$, and the third index has a dimension $\mathcal{D}$. Subsequently, the input characteristics are encoded through the utilization of single-qubit rotation gates $(\mathcal{R}_y(\omega))$, which preserve the tensor ring configuration. The fundamental element of the parametrized

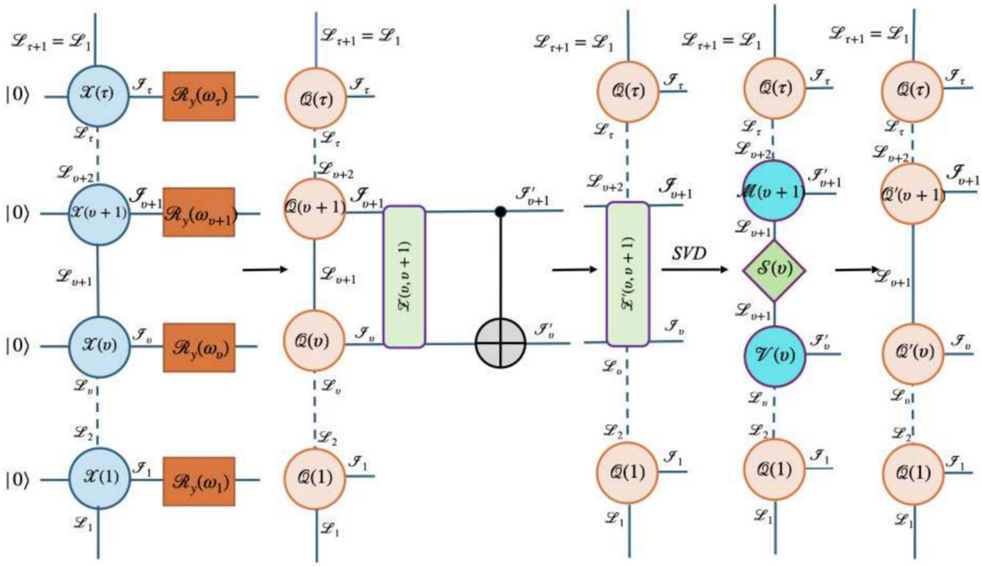

$\mathscr{L}_{\tau+1} = \mathscr{L}_1$
$|0\rangle$
$\mathscr{X}(\tau)$
$\mathscr{X}(\upsilon+1)$
$\mathscr{X}(\upsilon)$
$\mathscr{X}(1)$
$\mathscr{R}_y(\omega_\tau)$
$\mathscr{R}_y(\omega_{\upsilon+1})$
$\mathscr{R}_y(\omega_\upsilon)$
$\mathscr{R}_y(\omega_1)$
$\mathscr{J}_\tau$
$\mathscr{J}_{\upsilon+1}$
$\mathscr{J}_\upsilon$
$\mathscr{J}_1$
$\mathscr{J}'_{\upsilon+1}$
$\mathscr{J}'_\upsilon$
$\mathscr{L}_\tau$
$\mathscr{L}_{\upsilon+2}$
$\mathscr{L}_{\upsilon+1}$
$\mathscr{L}_\upsilon$
$\mathscr{L}_2$
$\mathscr{L}_1$
$\mathscr{Q}(\tau)$
$\mathscr{Q}(\upsilon+1)$
$\mathscr{Q}(\upsilon)$
$\mathscr{Q}(1)$
$\mathscr{Z}(\upsilon,\upsilon+1)$
$\mathscr{Z}'(\upsilon,\upsilon+1)$
SVD
$\mathscr{M}(\upsilon+1)$
$\mathscr{S}(\upsilon)$
$\mathscr{V}(\upsilon)$
$\mathscr{Q}'(\upsilon+1)$
$\mathscr{Q}'(\upsilon)$

Fig. 3 The initial tensor $\mathcal{X}(\upsilon)$ is contracted with the unitary rotation matrix $\mathcal{R}_y$. The common bond index is contracted between $\mathcal{X}(\upsilon)$ and $\mathcal{X}(\upsilon+1)$ to apply a two-qubit controlled rotation operation. SVD is applied on the resultant tensor $\mathcal{Z}'$ after reshaping it as $(\mathcal{I}'_\upsilon \times \mathcal{L}_\upsilon) \times (\mathcal{I}'_{\upsilon+1} \times \mathcal{L}_{\upsilon+1})$

circuit in every layer of the proposed Quan-TR model is the cascading entanglement of qubits, which is subsequently followed by parametrized single-qubit rotations. The two-qubit gates, such as the CNOT gate, do not preserve the TR representation. An approximation technique based on singular value thresholding (SVD) addresses this issue at this gate. The TR structure facilitates the computation of 2-qubit gates for adjacent qubits. A CNOT operation from the ultimate qubit to the initial qubit becomes feasible by employing a cascading configuration of the tensor ring. It should be noted that using the TR format allows utilization of the same rank $\mathcal{B}$ in each decomposition, which may not be feasible with the conventional MPS format. Using this approximation, all calculations for the forward pass exhibit linearity concerning the number of gates.

We develop a universal TR-QNet model that uses the intrinsic probabilistic behavior of qubit measurements to classify images using a hybrid classical-quantum framework. The aspects of the VQC learning framework regarding encoding, variational, and measurement are accomplished within the implementation of Quan-TR. The single-qubit rotation gate, $\mathcal{R}_y(\omega)$, encodes rotations along the Y axes in the encoding section. Quantum bits (qubits) represent the input state of Quan-TR in the proposed TR-QNet as

$$|\psi(\omega)\rangle = (\cos\omega|0\rangle + \sin\omega|1\rangle)|\mathcal{O}_{TN}\rangle \,. \tag{19}$$

In Quan-TR, the quantum states $|\psi(\omega)\rangle$ correspond to the quantum encoding of the classical inputs $\mathcal{O}_{TN}$ from the classical layer of TNN. The proposed Quan-TR is dense and constitutes parametrized single qubit gates with CNOT gates to entangle quantum states from each qubit. To encode phase information, the dressed quantum layer of TR-QNet uses the rotation gates $\mathcal{R}_y$ and $\mathcal{R}_z$. Complementary quantum states are created with the help of the CNOT gate. In the Bloch sphere projection, the $\mathcal{R}_y(\omega)$ and $\mathcal{R}_z(\omega)$ gates represent the following single-qubit rotations about the Y and Z-axes, respectively, as follows:

$$\mathcal{R}_y(\omega) = \exp\left(-jY\omega/2\right) = \begin{bmatrix} \cos\omega/2 & -\sin\omega/2 \\ \sin\omega/2 & \cos\omega/2 \end{bmatrix} \tag{20}$$

and

$$\begin{aligned} \mathcal{R}_z(\omega) &= \exp\left(-jZ\omega/2\right) \\ &= \begin{bmatrix} \exp\left(-j\omega/2\right) & 0 \\ 0 & \exp\left(j\omega/2\right) \end{bmatrix} . \end{aligned} \tag{21}$$

The initial tensor $\mathcal{X}(\upsilon)$ is contracted with the 2×2 unitary rotation matrix $\mathcal{R}_y$ to accomplish the one-qubit rotation. The rotated state of the υ^{th} qubit is represented by the resultant tensor $\mathcal{Q}(\upsilon)$ as follows:

$$\mathcal{Q}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} = \sum_{y_\upsilon} \mathcal{R}_y \mathcal{X}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} \tag{22}$$

Before conducting the two-qubit controlled rotation gate operations, the tensor ring must be converted into an orthogonal form centered around the qubits of interest, υ and $(\upsilon + 1)$ in the proposed TR-QNet. To build a new tensor, the common bond index is contracted between the tensors $\mathcal{X}(\upsilon)$ and $\mathcal{X}(\upsilon+1)$ as follows:

$$\mathcal{Z}(\upsilon,\upsilon+1)^{y_\upsilon y_{\upsilon+1}}_{x_\upsilon x_{\upsilon+2}} = \sum_{x_{\upsilon+1}=1}^{\mathcal{L}_{\upsilon+1}} \mathcal{Q}(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} \mathcal{Q}(\upsilon+1)^{y_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} \tag{23}$$

where $\mathcal{Z}$ is the contracted tensors of $\mathcal{Q}(\upsilon)$ and $\mathcal{Q}(\upsilon+1)$.

We restructure the two-qubit controlled-NOT gate ($CNOT$) into an operator acting on the joint state of qubits υ and $(\upsilon+1)$ to apply it to the two-qubit tensor derived from Eq. 23.

$$\mathcal{Z}'(\upsilon,\upsilon+1)^{y_\upsilon y_{\upsilon+1}}_{x_\upsilon x_{\upsilon+2}} = \sum_{y_\upsilon y_{\upsilon+1}} CNOT(\upsilon,\upsilon+1)^{y'_\upsilon y'_{\upsilon+1}}_{y_\upsilon y_{\upsilon+1}} \times \mathcal{Z}(\upsilon,\upsilon+1)^{y_\upsilon y_{\upsilon+1}}_{x_\upsilon x_{\upsilon+2}} \tag{24}$$

We apply singular value decomposition (SVD) on the resulting tensor $\mathcal{Z}'$ after reshaping it as $(y'_\upsilon \times x_\upsilon) \times (y'_{\upsilon+1} \times x_{\upsilon+1})$ in the following manner:

$$\mathcal{Z}'(\upsilon,\upsilon+1)^{y_\upsilon y_{\upsilon+1}}_{x_\upsilon x_{\upsilon+2}} = \sum_{x_{\upsilon+1}} \mathcal{V}(\upsilon)^{y'_\upsilon}_{x_\upsilon x_{\upsilon+1}} \mathcal{S}_{x_\upsilon} \mathcal{M}(\upsilon+1)^{y'_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} \tag{25}$$

In this context, $\mathcal{V}$ and $\mathcal{M}$ consist of orthogonal vectors, while $\mathcal{S}_{x_\upsilon}$ is formed by the singular values of matrix $\mathcal{Z}'$. The matrix always has a total of $2N$ singular values, regardless of the structure of the two-qubit gate. Here, N represents the bond dimension of the tensor ring. After truncating the matrix $\mathcal{S}_{x_\upsilon}$ to retain only the N largest singular values, we refer to the resulting matrix as $\mathcal{S}'_{x_\upsilon}$. $\mathcal{V}$ and $\mathcal{M}$ are truncated to retain only the orthogonal vectors associated with the N largest singular values and resultant matrices, and $\mathcal{Q}'(\upsilon)$ and $\mathcal{Q}'(\upsilon+1)$ are evaluated as follows:

$$\mathcal{Q}'(\upsilon)^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} = \mathcal{V}^{y_\upsilon}_{x_\upsilon x_{\upsilon+1}} \mathcal{S}'_{x_\upsilon} \tag{26}$$

$$\mathcal{Q}'(\upsilon+1)^{y_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} = \mathcal{M}^{y_{\upsilon+1}}_{x_{\upsilon+1} x_{\upsilon+2}} \tag{27}$$

The TR-approximation procedures are illustrated in Fig. 3 for better clarity and understanding.

The preprocessed data from the TNN layer, denoted as $\mathcal{O}^i_{TN}$, is transformed into a quantum state represented by $|\psi(\mathcal{O}^i_{TN})\rangle$. Subsequently, the quantum state undergoes processing through Quan-TR with parameters $\mathcal{U}(\omega_1,\omega_2,\cdots,\omega_n)$. Finally, by performing measurements on particular qubits using the Pauli-Z basis, we obtain a collection of outputs denoted as λ_j along with their corresponding probabilities as follows:

$$\mathcal{H}_{ij} = \langle\psi(\mathcal{O}^i_{TN})|\mathcal{U}^\dagger(\omega)|\lambda_j\rangle\langle\lambda_j|\mathcal{U}(\omega)|\psi(\mathcal{O}^i_{TN})\rangle \tag{28}$$

where complete operation is $\mathcal{U}(\omega)$ is defined as

$$\mathcal{U}(\omega) = \prod_{i=1}^{n} \mathcal{U}_i(\omega_i) = \mathcal{U}_n(\omega_n)\mathcal{U}_{n-1}(\omega_{n-1})\cdots\mathcal{U}_1(\omega_1). \tag{29}$$

The loss function, $\mathcal{L}(\omega)$, can be defined as follows considering the input quantum state as $|0\rangle^{\mathcal{N}_q}$.

$$\begin{aligned}\mathcal{L}(\omega) &= f(y_j(\omega), t_j) = \mathcal{H}(y_j(\omega) \neq t_j)\\ &= \sum_j^{\mathcal{N}_q} f((\langle 0|\psi^\dagger(\mathcal{O}^j_{TN})\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}^j_{TN})|0\rangle), t_j)\end{aligned} \tag{30}$$

where $\theta_j \in |\lambda_j\rangle\langle\lambda_j|$ and t_j correspond to a target output. To train the proposed Quan-TR model, the gradient of the loss function is evaluated as follows:

$$\begin{aligned}\frac{\delta\mathcal{L}(\omega)}{\delta\omega_j} &= \langle 0|\psi^\dagger(\mathcal{O}^j_{TN})\frac{\delta\mathcal{U}^\dagger(\omega)}{\delta\omega_j}\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}^j_{TN})|0\rangle\\ &\quad+\langle 0|\psi^\dagger(\mathcal{O}^j_{TN})\mathcal{U}^\dagger(\omega)\theta_j\frac{\delta\mathcal{U}(\omega)}{\delta\omega_j}\psi(\mathcal{O}^j_{TN})|0\rangle\\ &= \langle 0|\psi^\dagger(\mathcal{O}^j_{TN})\mathcal{U}_1^\dagger(\omega_1)\cdots\frac{\delta\mathcal{U}_j^\dagger(\omega_j)}{\delta\omega_j}\\ &\quad\cdots\mathcal{U}^\dagger{}_n(\omega_n)\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}^j_{TN})|0\rangle\\ &\quad+\langle 0|\psi^\dagger(\mathcal{O}^j_{TN})\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}_n(\omega_n)\cdots\frac{\delta\mathcal{U}_j(\omega_j)}{\delta\omega_j}\\ &\quad\cdots\mathcal{U}_1(\omega_1)\psi(\mathcal{O}^j_{TN})|0\rangle\\ &= \langle 0|\psi^\dagger(\mathcal{O}^j_{TN})\mathcal{U}_-^\dagger[i\psi_j]\mathcal{U}_+^\dagger\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}^j_{TN})|0\rangle\\ &\quad+\langle 0|\psi^\dagger(\mathcal{O}^j_{TN})\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}_+[-i\psi_j]\mathcal{U}_-\psi(\mathcal{O}^j_{TN})|0\rangle\end{aligned} \tag{31}$$

where $\mathcal{U}_j(\omega_j) = e^{-i\omega_j\psi(\omega^j)}$.

However, due to NISQ's limitations, classical simulators are now being utilized to optimize and update parameters and feed them back to TNN and Quan-TR separately until convergence conditions are reached. Hence, we have used the cross-entropy loss to update the parameters. The loss function ($\mathcal{L}_\omega$) is derived with the hyper-parameters ω of the proposed TR-QNet model as

$$\begin{aligned}\arg\min_{\omega} \mathcal{L}_{(\omega)} &= \sum_j^{\mathcal{N}_q}[t_j \log f(\mathcal{O}^j_{TN})\\ &\quad+(1-t_j)\log\{1-f(\mathcal{O}^j_{TN})\}]\,.\end{aligned} \tag{32}$$

where $f(\mathcal{O}^j_{TN},\omega)$ can be defined for binary optimization problem as follows:

$$f(\mathcal{O}^j_{TN},\omega) = \begin{cases} 1, & \text{if } f(\mathcal{O}^j_{TN},\omega) > 0\\ -1, & \text{otherwise}\end{cases} \tag{33}$$

The DMRG-like sweeping technique (White 1992) for training the TNN uses a stochastic gradient-based optimization strategy in which a gradient descent step with a learning rate updates the local bond tensors $B_{j,j+1}$ towards a global minimum of the loss function. To update the weights in TNN, a gradient of the bond tensors concerning the loss, $B_{j,j+1}$, is obtained by defining $f(\mathcal{O}^{j}_{TN}) = TB$, where T represents the contraction of every tensor in the TNN other than the bond tensor B.

3.3 Absence of barren plateaus in TR-QNet

To investigate the presence of barren plateaus within the TR-QNet framework, it is essential to evaluate the variance of the partial derivative of the empirical loss function.

$$\delta_\mu \mathcal{L}(\omega) = \frac{\delta_\mu \mathcal{L}(\omega)}{\delta \omega_\mu}, \tag{34}$$

where $\omega_\mu \in \omega$.

If there exists a barren plateau in the ω_μ direction, then

- $\mathbb{E}_\omega[\delta_\mu \mathcal{L}(\omega)] = 0$ and
- $Var[\delta_\mu \mathcal{L}(\omega)] \in O(c^{-\mathcal{N}_q}) \forall c > 1$ where $\mathbb{E}[\cdot]$ and $Var[\cdot]$ are designated as expected value and variance over randomly chosen parameters, respectively.

By randomly selecting parameters in TR-QNet, the probability of encountering a gradient magnitude $\delta_\mu \mathcal{L}(\omega) > \eta$ can be exponentially influenced by the qubit count $\mathcal{N}_q$. This relationship can be elucidated using Chebyshev's inequality, showcasing the exponential variation in the probability concerning the gradient magnitude threshold concerning the qubit count (Cervero Martín et al. 2023) as follows:

$$\begin{aligned} \mathcal{P}[|\delta_\mu \mathcal{L}(\omega) - \mathbb{E}_\omega[\delta_\mu \mathcal{L}(\omega)]| \geqslant \eta] &\leqslant \frac{Var[\delta_\mu \mathcal{L}(\omega)]}{\eta^2} \\ &\in O\left(\frac{c^{-\mathcal{N}_q}}{\eta^2}\right) \end{aligned} \tag{35}$$

For our local cost function for Quan-TR,

$$\mathbb{E}_\omega[\delta_\mu \mathcal{L}(\omega)] = Tr\left[\rho\left(\frac{\delta_\mu \mathcal{L}(\omega)}{\delta \omega_\mu} \mathcal{U}^\dagger(\omega)|\lambda_j\rangle\langle\lambda_j|\mathcal{U}(\omega)\right)\right], \tag{36}$$

where ρ designates the initial state density matrix. The variance of the gradient for a local cost function can be expressed as follows:

$$\begin{aligned} Var[\delta_\mu \mathcal{L}(\omega)] &\approx \sum_{j=0}^{2^{(\mathcal{N}_q - 1)}} |\langle j|\mathcal{U}^\dagger(\omega)|\lambda_j\rangle\langle\lambda_j|\mathcal{U}(\omega)|j\rangle|^2 \\ &\in O\left(\frac{1}{poly(\mathcal{N}_q)}\right) \end{aligned} \tag{37}$$

Hence, it is indicated that the variance may vanish polynomially with Quan-TR and the absence of a barren plateau. However, in our Quan-TR framework, the utilization of two-qubit CNOT gates with TR approximation leads to the establishment of a barren plateau cost landscape, with exceptions in exponentially small regions around local minima as shown in Fig. 4.

4 Results

4.1 Data sets

The Iris dataset (Fisher 1936) is often used as a benchmark to evaluate the performance of different machine learning algorithms. The Iris dataset contains a total of 150 samples, and each sample has four features: sepal length, sepal width, petal length, and petal width. We extracted three distinct binary data sets from the original Iris dataset. We added 80% samples to each training subset and the remaining 20% samples for each class as a test data set.

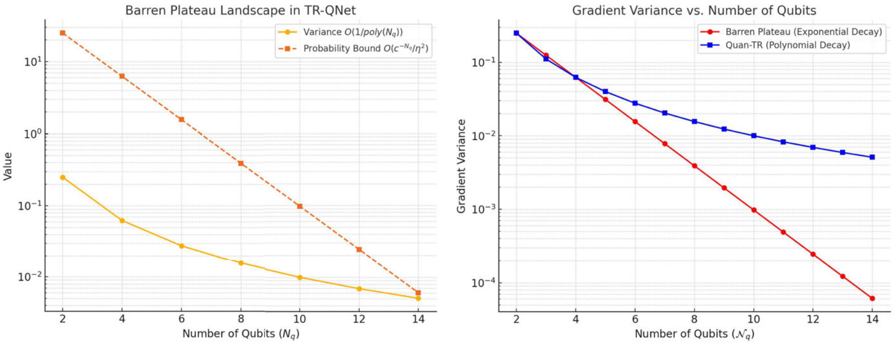
Researchers studying computer vision often rely on the MNIST dataset (LeCun et al. 1998) as a benchmark for ANN. The MNIST dataset has 70, 000 28 × 28 grayscale images (60, 000 for training and 10, 000 for testing), divided into 10 classes and each containing 7000 images.

The CIFAR-10 (Krizhevsky et al. 2012) dataset comprises a total of 60, 000 images in 10 categories (6000 for each class), with 32 × 32 color images including 50, 000 training images and 10, 000 test images. The tests, however, resize and transform CIFAR-10 images into 28 × 28 times their original size grey-scale images.

However, owing to the limited qubit available at the NISQ processor, we perform binary classification jobs using this batch of images with values 0 or 9; 1 or 8, 2 or 7, 3 or 6, and 4 or 5 and multi-class classification with values 0, 1, or 9; 2, 4, or 5; and 3, 6, or 7. We had to restrict our datasets to two randomly selected classes in our investigations, since the Qiskit quantum simulator only has access to a few qubits.

4.2 Simulation settings

We compute the mean and variance of the original input data sets. The data sets are then normalized using the zero-mean normalization procedure to have a zero-mean and unit variance before feeding into the TNN. The proposed TR-QNet comprises TN layers with six trainable MPO tensors, and a stochastic gradient-based algorithm has been employed to train MPOs (Panagakis et al. 2021), relying on a DMRG-like technique. Each MPO trainable weight on the TN layers has virtual dimension $\mathcal{V} = 4$ (Jahromi and Orus 2023) and a *ReLU* activation function. The last layer of Quan-TR in the output chain is a dense layer with softmax activation, which produces hot encoded vectors (OHE) and contains



(a) (b)

Fig. 4 **a** Visualization of the cost function landscape, revealing large flat regions indicative of barren plateaus. **b** The gradient variance decay as the number of qubits increases, demonstrating the expected exponential suppression

the predicted probabilities for the desired number of labels. The prior tensor gradient technique is appropriate for TNN models fully composed of TN layers. However, it is neither effective nor adaptable in hybrid architectures that combine TN layers and VQC models.

We used automated differentiation techniques (Baydin et al. 2018) and a classical back-propagation algorithm to determine the gradient of TNN trainable weights, as our TR-QNet is a feedforward hybrid neural network that combines TN and quantum layers. We have used the *TensorLy-Torch* library to compute the automatic differentiation of TN layers in PyTorch settings. However, being a hybrid classical-quantum framework, the classical TNN model is simulated on the Qiskit simulator. The weights of the TN layers are updated using a layer-by-layer approach. The intermediate dense layer has merely been included to compensate for the size mismatch between the features in the last TN layer and Quan-TR, and it is not trainable. We set the initial qubit state as $|00\cdots0\rangle$, which is later transformed into a TR representation since $\mathcal{X}(\upsilon) \in \mathbb{R}^{\mathcal{B}\times\mathcal{B}\times 2}$ tensor with only $(0, 0, 0)^{th}$ element as 1 and the rest as $0's$. Our Quan-TR is repeated r times to illustrate the depth of Quan-TR. The tensor ring rank in Quan-TR is set to $\mathcal{B} = 4$ for all tests. This rank serves as a pivotal hyperparameter choice in TR decompositions, impacting the compressibility, flexibility, and robustness of the model. Qiskit simulations have been carried out using the varying numbers of qubits (4, 6, 8, 10, 12) and the number of TN and Quan-TR layers on an Nvidia Tesla $V100 - SXM2$ GPU Cluster with 32 GB of memory and 640 tensor cores with 8 cores of Intel(R) Xeon(R) CPU E5-2683 v4@2.1GHz.

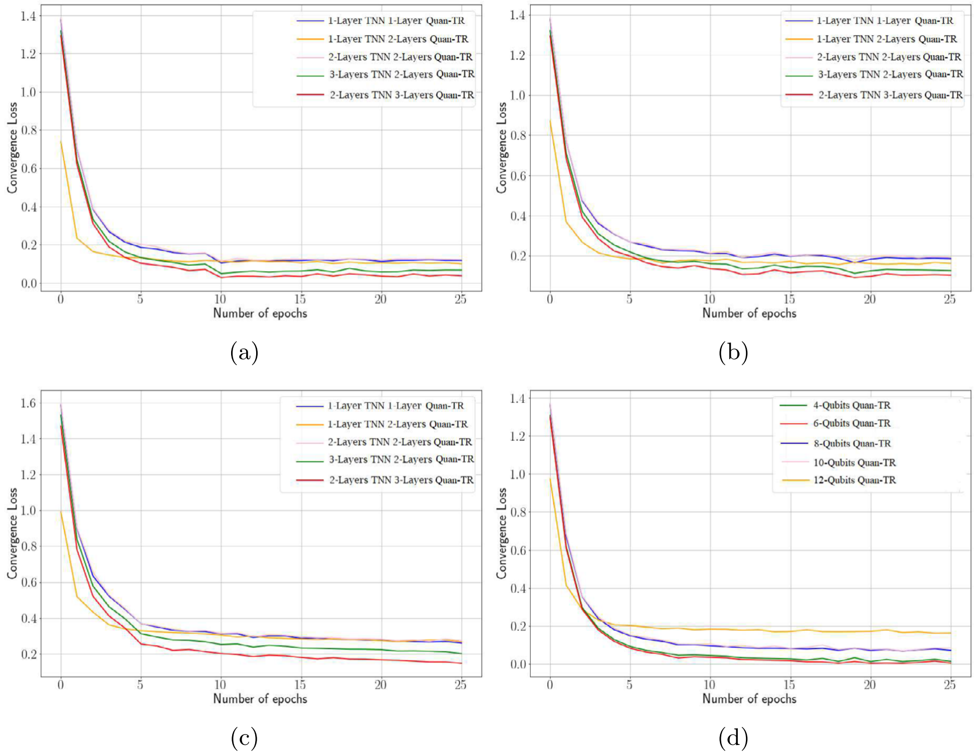
In the Iris data classification, TR-QNet model architecture involves utilizing TN layers alongside a non-trainable dense layer and TR-QNet to facilitate the classification of Iris data points. The TN layers comprise 64 input entries with 3 MPO trainable weights, each with input and output dimensions $d = 4$, virtual dimension $\mathcal{V} = 4$ (Jahromi and Orus 2023), and *ReLU* activation function. Our Quan-TR is provided with the four input feature vectors ($\mathcal{N}_q = 4$) for training from the previous TNN layer with batch size 4.

In the case of image classification, 784 input features (28 × 28) from the input images are received at the input layer of the proposed TR-QNet. The TN layers comprise 64 input entries with 6 MPO trainable weights, each with input and output dimensions $d = 2$, virtual dimension $\mathcal{V} = 4$, and *ReLU* activation function. With a maximum of 25 epochs, Quan-TR layers of the proposed TR-QNet model are rigorously trained using the Adam optimizer with an initial learning rate of 0.01 and weight decay (δ) of 0. Figures 5 and 6 show the convergence of loss during training of the proposed TR-QNet model varying the number of qubits and the TN and Quan-TR layers with 5-fold cross-validation. We chose three measures at random to represent the three classes of the dataset out of the 2^4 available measurements acquired from Quan-TR. To further transform selected measurements into class probabilities, we employ the sigmoid activation function (Softmax) and the cross-entropy loss function as given in Eq. A1. However, in the case of the MNIST and CIFAR-10 datasets, for binary classification, we choose the final measurements $|00\cdots0\rangle$ and $|11\cdots1\rangle$ as output values and batch size 32, where multiple readouts must feed the results to TR-QNet.

4.3 Simulation results

Numerical simulations have been conducted using large sets of Iris (Fisher 1936), MNIST (LeCun et al. 1998), and CIFAR-10 (Krizhevsky et al. 2012) data sets with varying numbers of qubit counts 4, 6, 8, 10, and 12 and TR ranks ($\mathcal{B}$) of 2, 3, and 4 as provided in Table 1. However, it has been found from the simulation data for the Iris dataset that the optimal result is found for 4 qubits Quan-TR model with tensor ring rank of 4 as reported in Table 1. In addition, to justify the introduction of the tensor ring (TR) in VQC, we have performed an ablation study on the proposed TR-QNet without the TR component and replacing variational Quan-TR by classical fully connected (FC) layers and referred to as TN-FC as provided in Table 2. The ablation study has been performed with varying numbers of qubit counts 4, 6, and

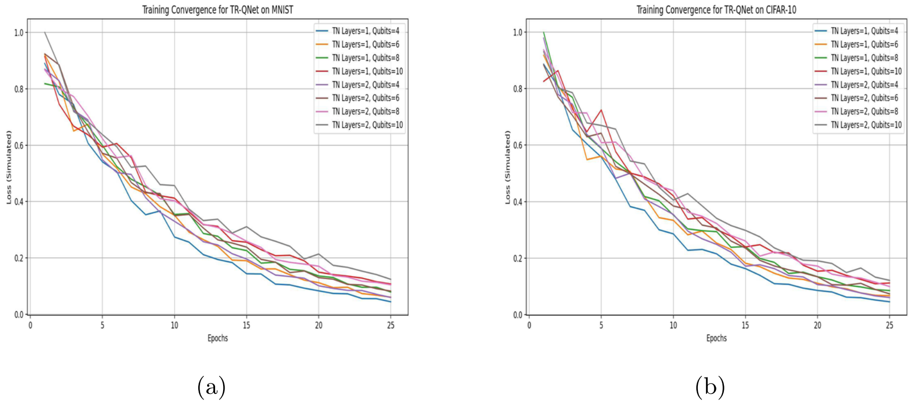
Fig. 5 TR-QNet training loss is reported on a 4 qubits systems with varying layers of TNN and Quan-TR layers for randomly selected binary classes **a** 1 or 2, **b** 2 or 3, and **c** 1 or 3, **d** varying with qubits (4, 6, 8, 10, and 12) on Iris dataset (Fisher 1936)



(a) (b)

(c) (d)

Fig. 6 TR-QNet training loss is reported with varying layers of TNN (1 and 2) and Quan-TR layers varying with qubits (4, 6, 8, and 10) for randomly selected binary classes on **a** MNIST (LeCun et al. 1998) and **b** CIFAR-10 (Krizhevsky et al. 2012)



(a) (b)

Table 1 Comparative analysis of the proposed 2-2 layers TR-QNet with varying number of qubits and tensor ranks ($\mathcal{B}$) on Iris dataset

Qubit	$\mathcal{B}$=2			$\mathcal{B}$=4			$\mathcal{B}$=6		
	$1-2$	$2-3$	$1-3$	$1-2$	$2-3$	$1-3$	$1-2$	$2-3$	$1-3$
4	0.919	0.875	0.891	0.941	0.939	0.955	0.939	0.911	0.924
6	0.801	0.794	0.789	0.882	0.865	0.880	0.829	0.817	0.829
8	0.782	0.765	0.770	0.787	0.780	0.788	0.782	0.757	0.772
10	0.728	0.765	0.760	0.773	0.743	0.760	0.769	0.750	0.763
12	0.628	0.605	0.616	0.673	0.643	0.690	0.629	0.617	0.609

Table 2 Mean accuracy of the proposed TR-QNet, QNet (without TR approximation), and TN-FC

Model	Qubits	Iris			MNIST					CIFAR-10				
		$1-2$	$2-3$	$1-3$	$0-9$	$1-8$	$2-7$	$3-6$	$4-5$	$0-9$	$1-8$	$2-7$	$3-6$	$4-5$
TR-QNet	4	**0.941**	**0.939**	**0.955**	0.817	**0.828**	0.869	0.850	0.809	0.758	0.803	0.747	0.798	0.771
	6	0.882	0.865	0.880	**0.828**	**0.836**	**0.891**	**0.863**	**0.870**	**0.819**	**0.849**	**0.833**	**0.857**	**0.809**
	8	0.787	0.780	0.788	0.667	0.684	0.669	0.650	0.671	0.658	0.603	0.647	0.618	0.624
QNet	4	0.928	0.925	0.915	0.801	0.803	0.819	0.813	0.798	0.739	0.789	0.713	0.772	0.755
	6	0.869	0.851	0.856	0.802	0.812	0.839	0.823	0.843	0.801	0.813	0.803	0.819	0.798
	8	0.769	0.754	0.771	0.647	0.663	0.628	0.623	0.645	0.634	0.588	0.612	0.601	0.607
TN-FC	N/A	0.809	0.811	0.815	0.765	0.750	0.788	0.746	0.741	0.687	0.679	0.711	0.709	0.697

8, and it has been observed that optimal accuracy has been achieved for class 2 or 3 in most cases of the Iris dataset. Our TR-QNet achieves promising accuracy of 94.5%, 86.16%, and 83.54% with 4 qubits on the Iris and with 6 qubits on the MNIST and CIFAR-10 datasets, respectively. It is noteworthy that the proposed TR-QNet and its quantum and classical counterparts, namely, variational quantum tensor networks classifier (VQTN) (Huang et al. 2021), quantum convolutional neural networks (QCNN) (Cong et al. 2019), tensor ring parametrized variational quantum circuit (TR-VQC) (Peddireddy et al. 2023), fully classically simulated variational tensor neural network (VTNN) (Jahromi and Orus 2023), and its classical counterpart TN-FC are trained on the binary and ternary pair of classes from the datasets. However, only two hidden layers are used in each network model to benchmark the simulations without any model bias.

To illustrate the resilience of the proposed model over the quantum counterpart and the models based on classical tensor neural networks, unforeseen test images are used for evaluation in the Iris (Fisher 1936), MNIST (LeCun et al. 1998), and CIFAR-10 (Krizhevsky et al. 2012) datasets. The training loss curves for the proposed TR-QNet model are demonstrated on the Iris (Fisher 1936), MNIST (LeCun et al. 1998), and CIFAR-10 (Krizhevsky et al. 2012) in Figs. 5 and 6, respectively. Our TR-QNet achieves near-optimal performance within $20-25$ epochs, demonstrating effective learning. The loss reduction follows an exponential decay pattern, stabilizing as the network approaches minima. However, convergence on CIFAR-10 is slower compared to MNIST, due to the dataset's increased complexity. The loss function exhibits fluctuations early in training but eventually stabilizes, suggesting that TR-QNet effectively learns higher-dimensional features over time. More training epochs $(30-35)$ are required to reach a lower loss plateau. The convergence analysis of the proposed TR-QNet is also provided in Appendix. Table 3 summarizes the numerical results obtained using our TR-QNet using 4, 6, and 8 number of qubits, VQTN (Huang et al. 2021), and QCNN (Cong et al. 2019) and TR-VQC (Peddireddy et al. 2023) using 4 (Iris dataset) and 8 qubits (MNSIT and CIFAR-10 datasets), fully classically simulated VTNN (Jahromi and Orus 2023), and its classical counterpart TN-FC on Iris, MNSIT, and CIFAR-10 datasets. It has been observed from the simulation results reported in Table 3 that optimal accuracy has been achieved in the case of the MNIST and CIFAR-10 datasets, class 3 or 6 reports optimal accuracy for the state-of-the-art models. However, in the case of multi-class classification, despite TR-QNet's low accuracy, as provided in Table 4, it outperforms

Table 3 Mean accuracy of the proposed TR-QNet with VQTN (Huang et al. 2021), TR-VQC (Peddireddy et al. 2023), QCNN (Cong et al. 2019), fully classically simulated VTNN (Jahromi and Orus 2023), and TN-FC on the test Iris (Fisher 1936), MNIST (LeCun et al. 1998), and CIFAR-10 (Krizhevsky et al. 2012) datasets (The bold values shed light to the two-sided paired Wilcoxon signed-rank test data (Conover 1999))

Model	Qubits	Iris			MNIST					CIFAR-10				
		$1-2$	$2-3$	$1-3$	$0-9$	$1-8$	$2-7$	$3-6$	$4-5$	$0-9$	$1-8$	$2-7$	$3-6$	$4-5$
TR-QNet	4	**0.941**	**0.939**	**0.955**	0.817	**0.828**	0.869	0.850	0.809	0.758	0.803	0.747	0.798	0.771
	6	0.882	0.865	0.880	**0.828**	**0.836**	**0.891**	**0.863**	**0.870**	**0.819**	**0.849**	**0.833**	**0.857**	**0.809**
VQTN	4/6	**0.924**	0.905	0.911	0.813	0.806	0.829	0.811	0.823	0.788	0.794	0.776	0.745	0.763
QCNN	4/6	0.871	0.852	0.861	0.772	0.736	0.740	0.742	0.755	0.721	0.714	0.717	0.732	0.746
TR-VQC	4/6	0.853	0.849	0.829	0.803	0.799	0.802	0.789	0.790	0.767	0.759	0.761	0.753	0.747
VTNN	N/A	0.838	0.839	0.842	0.797	0.788	0.798	0.778	0.780	0.701	0.698	0.734	0.727	0.715
TN-FC	N/A	0.809	0.811	0.815	0.765	0.750	0.788	0.746	0.741	0.687	0.679	0.711	0.709	0.697

Table 4 Mean accuracy of the proposed TR-QNet with tensor ranks ($\mathcal{B} = 4$), VQTN (Huang et al. 2021), VTNN (Jahromi and Orus 2023), and TN-FC for multi-class (3-class) classification (The bold numbers provide information about the two-sided paired Wilcoxon signed-rank test data)

		Iris	MNIST			CIFAR-10		
Model	Qubit	$1-2-3$	$0-1-9$	$2-4-5$	$3-6-7$	$0-1-9$	$2-4-5$	$3-6-7$
TR-QNet	4	**0.815**	**0.738**	0.723	**0.746**	0.707	0.712	0.723
	6	0.802	**0.741**	**0.736**	**0.739**	**0.719**	**0.725**	**0.731**
	8	0.785	0.709	0.717	0.716	**0.718**	0.707	0.713
VQTN	6	**0.811**	0.714	0.701	0.698	0.688	0.682	0.694
VTNN	N/A	0.774	0.689	0.684	0.679	0.613	0.639	0.657
TN-FC	N/A	0.755	0.694	0.677	0.681	0.601	0.619	0.643

VQTN, VTNN, and TN-FC. It may be noted that TR-VQC and QCNN are not feasible for multi-class classification owing to the limitations of their framework. In addition, we use a significant threshold $\gamma = 0.05$ for a two-sided paired Wilcoxon signed rank test (Conover 1999) to demonstrate the effectiveness of the proposed TR-QNet model over other methods. It is evident from the two-sided paired Wilcoxon signed-rank test that the proposed TR-QNet model yields statistically significant results using 4 and 6 qubits Quan-TR for Iris data and image (MNIST and CIFAR-10) classification, respectively. This is primarily owing to the limited Iris data feature demanding fewer qubits, whereas the larger image size requires more qubits. The subsequent increase to 8, 10, and 12 qubits resulted in a substantial decrease in accuracy for the proposed TR-QNet and the other methods, probably as a result of over-parameterization (Larocca et al. 2023) and barren plateaus (McClean et al. 2018).

However, to validate our claim regarding the barren plateau in our Quan-TR framework, we have conducted additional numerical experiments. Specifically, we analyzed the gradient variance decay as the number of qubits increases, demonstrating the expected exponential suppression. Additionally, we visualized the cost function landscape, revealing large flat regions indicative of barren plateaus. These results, which align with our theoretical analysis, are now included in Fig. 4.

5 Discussions

The simulation results reported in the manuscript show that the proposed TR-QNet model outperforms its quantum and classical counterparts for binary classification and multi-class (ternary) classification in test datasets in the given simulation settings. This is due to the fact that the proposed TR-QNet is capable of modulating classification tasks by substituting the trainable weight matrices of the fully connected dense layers of standard TNN with Quan-TR, and hence, TNN acts as an efficient encoding tool, especially for large image features with minimal loss of information from the input images. The VQC-based training algorithm resembling DMRG (White 1992) enables a straightforward entanglement of the entanglement spectrum of the MPO's (Panagakis et al. 2021) trainable weights, thereby facilitating a lucid comprehension of the correlations within the parameters of TN layers. We have presented a novel entanglement-aware training technique relying on hybrid classical-quantum algorithms and stochastic gradient-descent updates to train the proposed Quan-TR model efficiently. This approach operates on a condensed parameter subspace obtained from the tensorization of trainable weights, leading to local minima-free convergence and promising results.

However, in our Quan-TR framework, we find that while the tensor-ring approximation can mitigate the extent of barren plateaus, they are not entirely eliminated due to the presence of entangling operations. This implies that Quan-TR retains a level of quantum complexity that prevents full classical simulability, distinguishing it from classical machine learning models.

Moreover, our implementation enables the creation of hybrid architectures that combine TN layers, dense layers, and Quan-TR to create valid instances of deep learning models. We have conducted an additional experiment in which we completely eliminate the quantum component from the proposed TR-QNet, effectively converting the quantum-enhanced TNN into a fully classical model and referred to as TN-FC as reported for binary classification and provided additional results for ternary classification. Our findings indicate that the fully classical system TN-FC, while still performing reasonably well, does not achieve the same level of performance as our hybrid quantum-classical TR-QNet model. Specifically, we observe that the hybrid model exhibits improved accuracy and generalization capabilities, particularly in scenarios with complex data structures. This demonstrates that the inclusion of the quantum part indeed provides an enhancement, rather than merely an alternative approach.

Nevertheless, it is worth noting that the multi-layer design of Quan-TR within our proposed TR-QNet has the potential to produce the cascading effect of entanglement between the neuronal inputs and their outputs. Moreover, quantum layers in TR-QNet exploit entanglement to capture non-classical correlations between qubits, enhancing the learning of intricate dependencies within the data. They may exhibit resistance to certain types of noise, which could lead to more robust models in specific applications (Konar et al. 2022, 2023a, b, c). Our results indicate that the classical TNNs with DMRG-like training and Quan-TR methods work accurately and efficiently for data and image classification tasks. Direct access to the singular values throughout the virtual dimensions of the trainable MPOs of TN layers provided by the DMRG-like training method and tensor ring-optimized variational learning algorithm is crucial as it enables the computation of a measure of entanglement (correlation) between the features and model parameters. Since more qubits signify a bigger Hilbert space to parametrize the input data (Biamonte et al. 2017), we see a general pattern of increasing the classification accuracy with the qubit count from 4 to 6. However, a further increase in the 8, 10, and 12 qubits resulted in a substantial decrease in accuracy for the proposed TR-QNet, probably as a result of overparameterization (Larocca et al. 2023) and barren plateaus (McClean et al. 2018). Due to the additional non-linearity caused by the truncated SVD over the MPOs and two-qubit gate transformations, we also notice that in the case of the Iris dataset, TR-QNet significantly outperforms the VQTN, QCNN, and TR-VQC with complete quantum state information and classically simulated VTNN. Eight-qubit circuit topologies are used in a series of studies utilizing different rankings to examine the impact of tensor-ring rank on the performance of TR-QNet as it yields optimal results regarding input qubit counts.

However, the proposed TR-QNet model for multi-class image classification has achieved a comparable level of precision, primarily due to the inherent challenges faced by the slow convergence of Quan-TR. Hence, even though its promising performance is exhibited on relatively more minor datasets, the proposed TR-QNet is restricted due to the inherent difficulties in scaling and time-intensive training of Quan-TR. Furthermore, TR-QNet has achieved higher accuracy in binary classification tasks than its quantum and classical counterparts. Our method paves the way for developing novel representations of a quantum state in the deep neural network. It serves as a valuable tool for investigating the expressive potential of quantum neural states. We aim to develop an efficient TR-QNet model comprising an optimized Quan-TR with fewer hyper-parameters.

5.1 Complexity analysis

Assuming that for an τ^{th} order TR $\mathcal{T}$ as $y_\upsilon = \mathcal{I}$ and $x_\upsilon = \mathcal{L} \forall \upsilon = 1, \cdots \tau$, successive SVDs on TR cores $(\mathcal{I}\mathcal{L} \times \mathcal{L})$ require $O(\mathcal{I}\mathcal{L}^3)$ for each operation, and for the TR of τ^{th} order, $\mathcal{T}$ requires $O(\tau\mathcal{I}\mathcal{L}^3)$ (Oseledets 2011). The computational complexity of Quan-TR is $O(\mathcal{N}_q\mathcal{I})$ since each calculation of a single or two-qubit rotation gate in Quan-TR is $O(1)$ (Peddireddy et al. 2023). Hence, the total computational complexity of the proposed TR-QNet is estimated as $O(\tau\mathcal{N}_q\mathcal{I} + (\tau - 2)\mathcal{I}\mathcal{L}^3)$.

6 Conclusion

In line with the impressive advances in quantum machine learning, the proposed TR-QNet framework improves over fully classical TNN. It has been developed as a proof-of-concept using hybrid classical-quantum algorithms for better training strategies for TNN. This paper investigates the benefits of the proposed Quan-TR to find a better optimization strategy for TR-QNet, which exploits the inherent entanglement between qubits. The simulation results on the test datasets using the proposed TR-QNet model show its efficiency over the quantum and classical counterparts in binary and multi-class classification. Moreover, the simulation results demonstrate the efficacy of the proposed TR-QNet in various settings, which is crucial for data classification and image recognition in noisy intermediate-scale quantum (NISQ) devices. Consequently, our TR-QNet model is a strong contender for deep learning and can revolutionize the studies in quantum machine learning.

While tensor rings (TR) and variational quantum layers (VQC) are integrated into our tensor network (TR-QNet) for classification tasks, the following benefits emerge:

1. Efficient computation: TR parameter efficiency allows the model to handle expensive two-qubit gate operations (CNOT) in polynomial time. Hence, our TR-QNet model is suitable for high-dimensional inputs without excessive computational costs.
2. Enhanced pattern recognition: The combined expressivity of TR and the powerful feature extraction of quantum layers enable the network to capture and utilize complex patterns in the image data for classification.
3. Scalable and robust model: The structural regularization of TR and the adaptive nature of variational quantum layers (Quan-TR) contribute to building robust and scalable models that generalize well to unseen data.
4. Potential quantum speedups: While still an active research area, the potential computational speedups from quantum processing of our TR-QNet could offer practical advantages in training and inference times.

However, the current TR-QNet architecture for deep convolutional neural networks and their training algorithms for regression and classification, which can be deployed immediately in near-term quantum devices, remains to be investigated. The authors are engaged in this direction.

Appendix A: Convergence analysis of TR-QNet

Due to NISQ's limitations, classical simulators are now being utilized to optimize and update parameters and feed them back to TNN and Quan-TR separately until convergence conditions are reached. Hence, we have used the cross-entropy loss to update the parameters. The loss function ($\mathcal{L}_\omega$) is derived with the hyperparameters ω of the proposed TR-QNet model as

$$\begin{aligned}\arg\min_{\omega} \mathcal{L}_\omega = \sum_{j}^{\mathcal{N}_q}[t_j \log f(\mathcal{O}_{TN}^j) \\ +(1-t_j)\log\{1-f(\mathcal{O}_{TN}^j)\}]\end{aligned} \tag{A1}$$

t_j corresponds to a target output, $\mathcal{N}_q$ is the number of qubits in Quan-T6,R and $f(\mathcal{O}_{TN}^j)$ is the average outcome on quantum measurement of a qubit j concerning the network hyperparameter set ω as evaluated in the following subsection as follows:

$$f(y_j(\omega), t_j) = \sum_{j}^{\mathcal{N}_q} f((\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}_{TN}^j)|0\rangle), t_j) \tag{A2}$$

where $\theta_j(\omega) \in |\lambda_j\rangle\langle\lambda_j|$ and t_j corresponds to a target output and the preprocessed data from the TNN layer, denoted as $\mathcal{O}_{TN}^i$, is transformed into a quantum state represented by $|\psi(\mathcal{O}_{TN}^i)\rangle$.

To train the proposed Quan-TR model, the gradient of the loss function is evaluated as follows:

$$\begin{aligned}\frac{\delta f^\iota(\mathcal{O}_{TN}^j)}{\omega^j} &= \langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\frac{\delta\mathcal{U}^\dagger(\omega)}{\delta\omega_j}\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}_{TN}^j)|0\rangle \\ &+\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}^\dagger(\omega)\theta_j\frac{\delta\mathcal{U}(\omega)}{\delta\omega_j}\psi(\mathcal{O}_{TN}^j)|0\rangle \\ &= \langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}_1^\dagger(\omega_1) \\ &\cdot\frac{\delta\mathcal{U}_j^\dagger(\omega_j)}{\delta\omega_j}\cdot\mathcal{U}^\dagger{}_n(\omega_n)\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}_{TN}^j)|0\rangle \\ &+\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}_n(\omega_n) \\ &\cdot\frac{\delta\mathcal{U}_j(\omega_j)}{\delta\omega_j}\cdot\mathcal{U}_1(\omega_1)\psi(\mathcal{O}_{TN}^j)|0\rangle\end{aligned} \tag{A3}$$

where $\mathcal{U}_j(\omega_j) = e^{-i\omega_j\psi(\omega^j)}$. The global phase does not directly affect the measurement results; hence, we disregard the global phase. Now, the rotation gates can be written as follows:

$$\begin{aligned}\frac{\delta\psi(\omega, \mathcal{O}_{TN})}{\delta\omega^j} &= \frac{1}{2}\psi(\omega+\frac{\pi}{2}, \mathcal{O}_{TN})\frac{\delta\psi^\dagger(\omega, \mathcal{O}_{TN})}{\delta\omega^j} \\ &= \frac{1}{2}\psi(\omega-\frac{\pi}{2}, \mathcal{O}_{TN})\end{aligned} \tag{A4}$$

We obtain as follows by substituting Eq. A3 by Eq. A4:

$$\begin{aligned}\frac{\delta f^\iota(\mathcal{O}_{TN}^j)}{\omega^j} &= \langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}_1^\dagger(\omega_1)\cdot\frac{\delta\mathcal{U}_j^\dagger(\omega_j)}{\delta\omega_j} \\ &\cdot\mathcal{U}^\dagger{}_n(\omega_n)\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}_{TN}^j)|0\rangle \\ &+\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}_n(\omega_n) \\ &\cdot\frac{\delta\mathcal{U}_j(\omega_j)}{\delta\omega_j}\cdot\mathcal{U}_1(\omega_1)\psi(\mathcal{O}_{TN}^j)|0\rangle \\ &= \frac{1}{2}\Bigg\{-\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}_1^\dagger(\omega_1)\cdot\frac{\delta\mathcal{U}_j^\dagger(\omega_j-\frac{\pi}{2})}{\delta\omega_j} \\ &\cdot\mathcal{U}^\dagger{}_n(\omega_n)\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}_{TN}^j)|0\rangle \\ &+\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}_n(\omega_n) \\ &\cdot\frac{\delta\mathcal{U}_j(\omega_j+\frac{\pi}{2})}{\delta\omega_j}\cdot\mathcal{U}_1(\omega_1)\psi(\mathcal{O}_{TN}^j)|0\rangle\Bigg\} \\ &= \frac{1}{2}\{\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}_-^\dagger[i\psi_j]\mathcal{U}_+^\dagger\theta_j\mathcal{U}(\omega)\psi(\mathcal{O}_{TN}^j)|0\rangle \\ &-\langle 0|\psi^\dagger(\mathcal{O}_{TN}^j)\mathcal{U}^\dagger(\omega)\theta_j\mathcal{U}_+[-i\psi_j]\mathcal{U}_-\psi(\mathcal{O}_{TN}^j)|0\rangle\} \\ &= \frac{1}{2}\Psi_{\omega_+}(\psi(\mathcal{O}_{TN}^j)) - \frac{1}{2}\Psi_{\omega_-}(\psi(\mathcal{O}_{TN}^j))\end{aligned} \tag{A5}$$

For the rotation gates $\mathcal{R}_y(\omega_y)$ and $\mathcal{R}_z(\omega_z)$ of Quan-TR, the angle of rotation [variational parameter (ω)] is ω_y and ω_z, respectively. The rotation gates $\mathcal{R}_y(\omega_y)$ and $\mathcal{R}_z(\omega_z)$ of Quan-TR operate the qubits $|\psi_y\rangle$ and $|\psi_z\rangle$ as follows:

$$|\psi_y(\iota+1)\rangle = \begin{pmatrix}\cos\triangle\omega_y(\iota) & -\sin\triangle\omega_y(\iota) \\ \sin\triangle\omega_y(\iota) & \cos\triangle\omega_y(\iota)\end{pmatrix}|\psi_y(\iota)\rangle \tag{A6}$$

$$|\psi_z(\iota+1)\rangle = \begin{pmatrix}\exp(-j\triangle\omega_z(\iota)) & 0 \\ 0 & (-j\triangle\omega_z(\iota))\end{pmatrix}|\psi_z(\iota)\rangle \tag{A7}$$

where

$$\omega_y(\iota+1) = \omega_y(\iota) + \triangle\omega_y(\iota) \tag{A8}$$

and

$$\omega_z(\iota+1) = \omega_z(\iota) + \triangle\omega_z(\iota) \tag{A9}$$

For the quantum layers in Quan-TR at epoch, ι, Eqs. A8 and A9 measure the change in the phase or angles $\triangle\omega_y(\iota)$ and $\triangle\omega_z(\iota)$, respectively. Let us consider

$$\mathcal{C}(\iota) = \omega_y(\iota) - \overline{\omega_y(\iota)} \tag{A10}$$

$$\mathcal{D}(\iota) = \omega_z(\iota) - \overline{\omega_z(\iota)} \tag{A11}$$

and

$$\mathcal{R}(\iota) = \omega_y(\iota+1) - \omega_y(\iota) = \mathcal{C}(\iota+1) - \mathcal{C}(\iota) \tag{A12}$$

$$\mathcal{S}(\iota) = \omega_z(\iota+1) - \omega_z(\iota) = \mathcal{D}(\iota+1) - \mathcal{D}(\iota) \tag{A13}$$

The optimal phases or angles are therefore $\overline{\omega}_y(\iota)$ and $\overline{\omega}_z(\iota)$ for the rotation gates $\mathcal{R}_y(\omega_y)$ and $\mathcal{R}_z(\omega_z)$, respectively.

To update the weights in TNN, a gradient of the bond tensors concerning the loss $(\mathcal{L}_\omega)$, $B_{j,j+1}$, is obtained by defining $f(\mathcal{O}^j_{TN}) = TB$, where T represents the contraction of every tensor in the TNN other than the bond tensor B.

When considering $B_j(\iota)$, $\omega^j_y(\iota)$, and $\omega^j_z(\iota)$, the loss function $\mathcal{L}_\omega(B, \omega_y, \omega_z)$ is differentiated as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_\omega(B,\omega_y,\omega_z)}{\partial B_j(\iota)} &= \sum_{j=1}^{\mathcal{N}_q} \frac{\partial f^\iota(\mathcal{O}^j_{TN})}{B_j(\iota)} \left[\frac{t_j}{f^\iota(\mathcal{O}^j_{TN})} - \frac{t_j - 1}{1 - f^\iota(\mathcal{O}^j_{TN})}\right] \\ &= \sum_{j=1}^{\mathcal{N}_q} \mathcal{T}^j(\iota) \left[\frac{t_j}{f^\iota(\mathcal{O}^j_{TN})} - \frac{t_j - 1}{1 - f^\iota(\mathcal{O}^j_{TN})}\right]\end{aligned} \tag{A14}$$

Hence, the change in the bond tensor designated as $\triangle B_j(\iota)$ is evaluated as follows:

$$\triangle B_j(\iota) = -\gamma(\iota) \frac{\partial \mathcal{L}_\omega(B,\omega_y,\omega_z)}{\partial B_j(\iota)} \tag{A15}$$

Here, $\gamma(\iota)$ is a learning rate in the gradient descent procedure for updating the bond tensors in TN layers.

$$\frac{\partial \mathcal{L}_\omega(B,\omega_y,\omega_z)}{\partial \omega^j_y(\iota)} = \sum_{j=1}^{\mathcal{N}_q} \frac{\partial f^\iota(\mathcal{O}^j_{TN})}{\omega^j_y(\iota)} \left[\frac{t_j}{f^\iota(\mathcal{O}^j_{TN})} - \frac{t_j - 1}{1 - f^\iota(\mathcal{O}^j_{TN})}\right] \tag{A16}$$

$$\frac{\partial \mathcal{L}_\omega(B,\omega_y,\omega_z)}{\partial \omega^j_z(\iota)} = \sum_{j=1}^{\mathcal{N}_q} \frac{\partial f^\iota(\mathcal{O}^j_{TN})}{\omega^j_z(\iota)} \left[\frac{t_j}{f^\iota(\mathcal{O}^j_{TN})} - \frac{t_j - 1}{1 - f^\iota(\mathcal{O}^j_{TN})}\right] \tag{A17}$$

Here, parameter shift techniques are used to evaluate the gradient of the Quan-TR parameters ω_y and ω_z (Mitarai et al. 2018; Li et al. 2017; Huang et al. 2021) as follows:

$$\frac{\partial f^\iota(\mathcal{O}^j_{TN})}{\omega^j_y(\iota)} = \frac{1}{2}\left[\Psi^{\iota+1}_{\omega_y+\frac{\pi}{2}}(\psi(\mathcal{O}^j_{TN})) - \Psi^{\iota}_{\omega_y-\frac{\pi}{2}}(\psi(\mathcal{O}^j_{TN}))\right] \tag{A18}$$

and

$$\frac{\partial f^\iota(\mathcal{O}^j_{TN})}{\omega^j_z(\iota)} = \frac{1}{2}\left[\Psi^{\iota+1}_{\omega_z+\frac{\pi}{2}}(\psi(\mathcal{O}^j_{TN})) - \Psi^{\iota}_{\omega_z-\frac{\pi}{2}}(\psi(\mathcal{O}^j_{TN}))\right] \tag{A19}$$

where with rotation angles $\omega^j_y(\iota)$ and $\omega^j_z(\iota)$, respectively, $\Psi(\iota)_{\omega_y \pm \frac{\pi}{2}}(\psi(\mathcal{O}^j_{TN}))$ and $\Psi(\iota)_{\omega_z \pm \frac{\pi}{2}}(\psi(\mathcal{O}^j_{TN}))$ are the measured qubit $\psi(\mathcal{O}^j_{TN})$. The changes in phase or angles are designated as $\triangle\omega^j_y(\iota)$ and $\triangle\omega^j_z(\iota)$, respectively, for the rotation gate used to update the qubits. The rotation angles are then modified using the formula below:

$$\triangle\omega^j_y(\iota) = -\nu(\iota)\left\{\frac{\partial f^\iota(\mathcal{O}^j_{TN})}{\omega^j_y(\iota)}\right\} \tag{A20}$$

$$\triangle\omega^j_z(\iota) = -\mu(\iota)\left\{\frac{\partial f^\iota(\mathcal{O}^j_{TN})}{\omega^j_z(\iota)}\right\} \tag{A21}$$

Here, $\nu(\iota)$ and $\mu(\iota)$ are the learning rates for updating the rotation angles in the gradient descent procedure.

Appendix B: Additional simulation results

Additional plots for the simulation results reported in Tables 1– 4 are provided in Figs. 7, 8, 9, and 10, respectively.

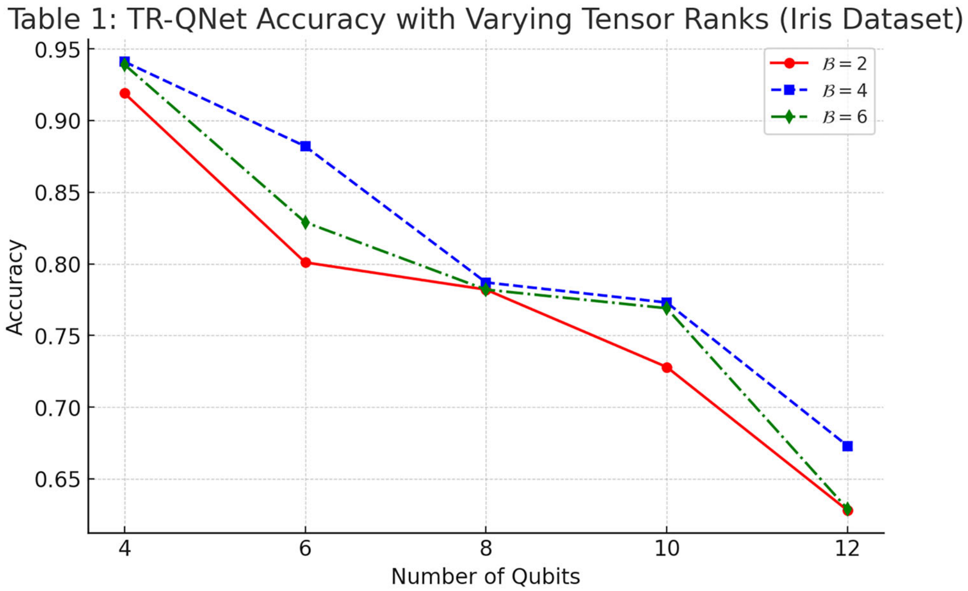


Fig. 7 Plots for analysis of the proposed 2-2 layers TR-QNet with varying number of qubits and tensor ranks ($\mathcal{B}$) on Iris dataset as reported in Table 1

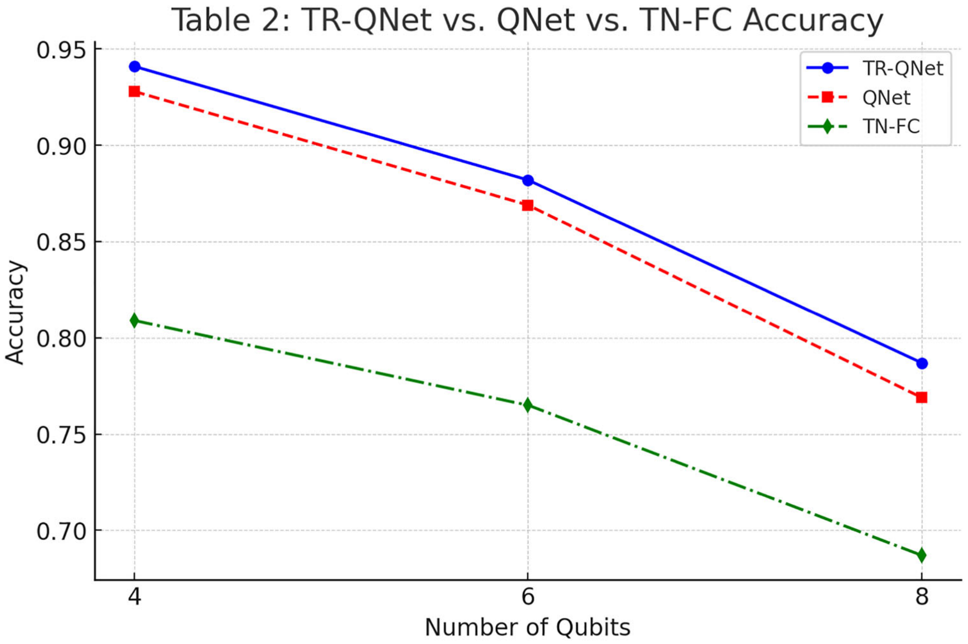


Fig. 8 Plots for mean accuracy of the proposed TR-QNet, QNet (without TR approximation), and TN-FC as reported in Table 2

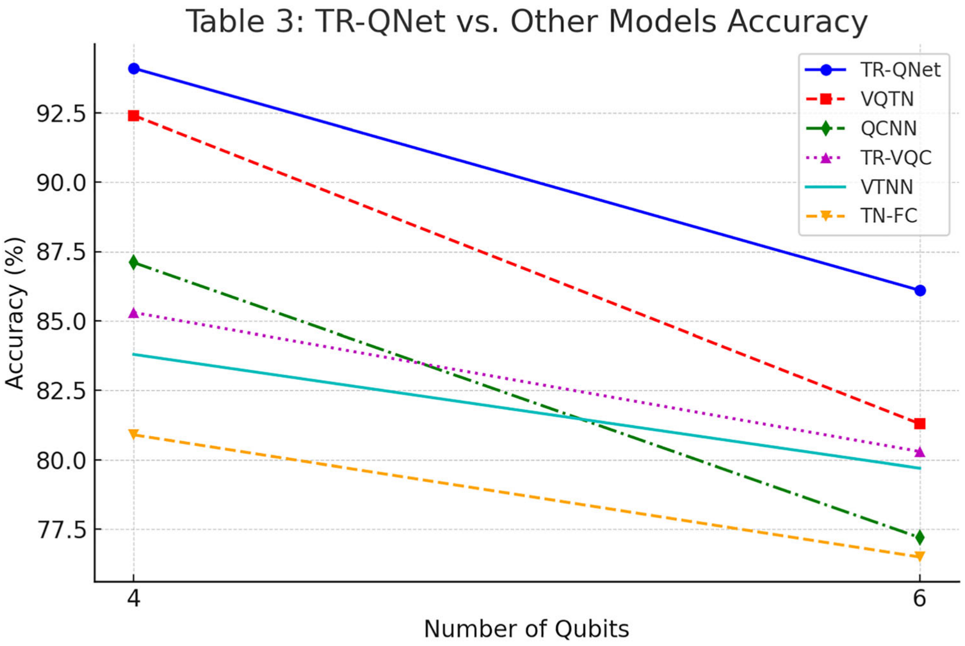


Fig. 9 Plots for mean accuracy of the proposed TR-QNet with VQTN (Huang et al. 2021), TR-VQC (Peddireddy et al. 2023), QCNN (Cong et al. 2019), and fully classically simulated VTNN (Jahromi and Orus 2023), and TN-FC on the test Iris (Fisher 1936), MNIST (LeCun et al. 1998), and CIFAR-10 (Krizhevsky et al. 2012) datasets as reported in Table 3

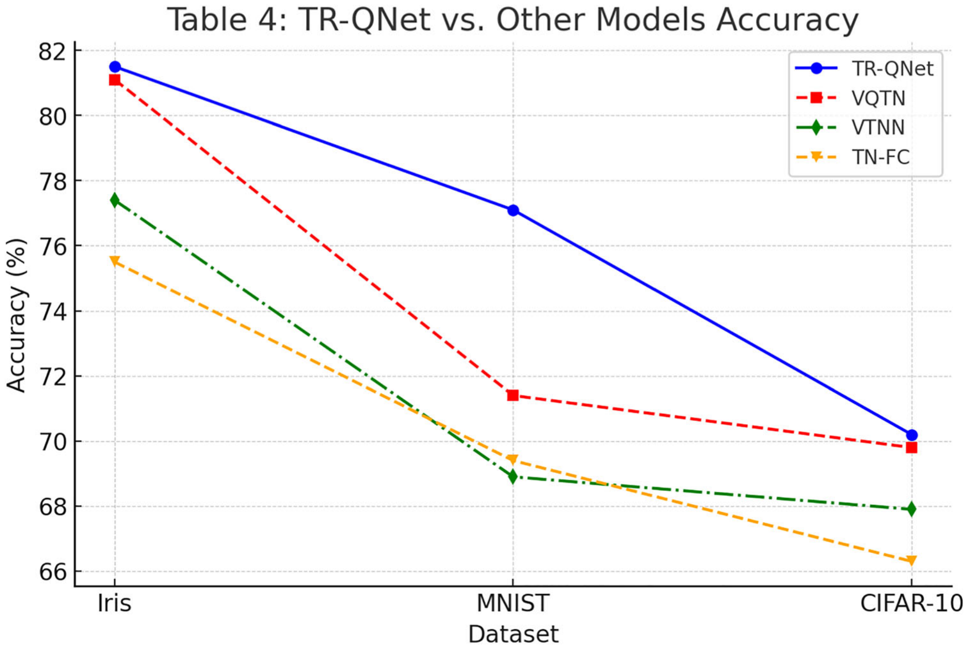


Fig. 10 Bar plots for mean accuracy of the proposed TR-QNet with tensor ranks ($\mathcal{B} = 4$), VQTN (Huang et al. 2021), VTNN (Jahromi and Orus 2023), and TN-FC for multi-class (3-class) classification as reported in Table 4

Author Contributions Debanjan Konar (design and implementation, writing the draft manuscript), Dheeraj Pedireddy (review and editing the manuscript), Bijaya K. Panigrahi (mentoring and review the manuscript), and Vaneet Aggarwal (mentoring and review the manuscript).

Funding This work was partially supported by the Fulbright-Nehru Visiting Researcher Grant #2858$FNPDR$/2022.

Data Availability The Iris dataset [53], MNIST[54], and CIFAR-10 [55] datasets can be found in the following links: https://archive.ics.uci.edu/dataset/53/iris, http://yann.lecun.com/exdb/mnist/, and https://www.cs.toronto.edu/~kriz/cifar.html, respectively.

Code Availability The PyTorch implementation of TR-QNet is available on Github: https://github.com/konar1987/TR-QNet/.

Declarations

Conflict of interest The authors declare no competing interests.

References

Abbas A, Sutter D, Zoufal C et al (2021) The power of quantum neural networks. Nat Comput Sci 1:403–409. https://doi.org/10.1038/s43588-021-00084-1

Arute F, Arya K, Babbush R et al (2019) Quantum supremacy using a programmable superconducting processor. Nature 574:505–510. https://doi.org/10.1038/s41586-019-1666-5

Asante-Mensah1 MG, Ahmadi-Asl1 S, Cichocki A (2021) Matrix and tensor completion using tensor ring decomposition with sparse representation. Mach Learn Sci Technol 2(3):035008. https://doi.org/10.1088/2632-2153/abcb4f

Bahri M, Panagakis Y, Zafeiriou S (2019) Robust Kronecker component analysis. IEEE Trans Pattern Anal Mach Intell 41(10):2365–2379. https://doi.org/10.1109/TPAMI.2018.2881476

Baydin AG, Pearlmutter BA, Radul AA, Siskind JM (2018) Automatic differentiation in machine learning: a survey. J Mach Learn Res 18(153):1–43

Beer K, Bondarenko D, Farrelly T et al (2020) Training deep quantum neural networks. Nat Commun 11(808). https://doi.org/10.1038/s41467-020-14454-2

Biamonte J, Wittek P, Pancotti N et al (2017) Quantum machine learning. Nature 594:195–202. https://doi.org/10.1038/nature23474

Carrasquilla J, Melko R (2017) Machine learning phases of matter. Nat Phys 13:431–434. https://doi.org/10.1038/nphys4035

Cerezo M, Arrasmith A, Babbush R et al (2021) Variational quantum algorithms. Nat Rev Phys 3:625–644. https://doi.org/10.1038/s42254-021-00348-9

Cervero Martín E, Plekhanov K, Lubasch M (2023) Barren plateaus in quantum tensor network optimization. Quantum 7:974. https://doi.org/10.22331/q-2023-04-13-974

Cohen N, Shashua A (2016) Convolutional rectifier networks as generalized tensor decompositions. In: Proceedings of the 33rd international conference on international conference on machine learning (ICLR), pp 955–963. https://doi.org/10.5555/3045390.3045492

Cong I, Choi S, Lukin MD (2019) Quantum convolutional neural networks. Nat Phys 15:1273–1278. https://doi.org/10.1038/s41567-019-0648-8

Conover WJ (1999) Practical nonparametric statistics, 3rd ed

Convy I, Whaley KB (2022) Interaction decompositions for tensor network regression. Mach Learn Sci Technol 3(4):045027. https://doi.org/10.1088/2632-2153/aca271

Dian R, Li S, Fang L (2019) Learning a low tensor-train rank representation for hyperspectral image super-resolution. IEEE Trans Neural Netw Learn Syst 30(9):2672–2683. https://doi.org/10.1109/TNNLS.2018.2885616

Fanaee-T H, Gama J (2016) Tensor-based anomaly detection: an interdisciplinary survey. Knowl Based Syst 98:130–147. https://doi.org/10.1016/j.knosys.2016.01.027

Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(2):179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Grasedyck L, Hackbusch W (2011) An introduction to hierarchical (H-) rank and TT-rank of tensors with examples. Comput Methods Appl Math 11(3):291–304. https://doi.org/10.2478/cmam-2011-0016

Gutiérrez IL, Mendl CB (2022) Real-time evolution with neural-network quantum states. Quantum 6:627. https://doi.org/10.22331/q-2022-01-20-627

Gyongyosi L, Imre S (2019) Training optimization for gate-model quantum neural networks. Sci Rep 9:12679. https://doi.org/10.1038/s41598-019-48892-w

Hayashi K, Yamaguchi T, Sugawara Y, Maeda S (2019) Exploring unexplored tensor network decompositions for convolutional neural networks. In: Advances in neural information processing systems, vol 32

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on computer vision and pattern recognition (CVPR), pp 770–778. https://doi.org/10.1109/CVPR.2016.90

Holtz S, Rohwedder T, Schneider R (2012) On manifolds of tensors of fixed TT-rank. Numer Math 120:701–731. https://doi.org/10.1007/s00211-011-0419-7

Huang R, Tan X, Xu Q (2021) Variational quantum tensor networks classifiers. Neurocomputing 452:89–98. https://doi.org/10.1016/j.neucom.2021.04.074

Huggins W, Patil P, Mitchell B, Whaley KB, Stoudenmire EM (2019) Towards quantum machine learning with tensor networks. Quantum Sci Technol 4(2):024001. https://doi.org/10.1088/2058-9565/aaea94

Jahromi SS, Orus R (2023) Variational tensor neural networks for deep learning. https://doi.org/10.48550/arXiv.2211.14657

Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500. https://doi.org/10.1137/07070111X

Konar D, Bhattacharya S, Panigrahi BK, Nakamatsu K (2016) A quantum bi-directional self-organizing neural network (QBDSONN) architecture for binary object extraction from a noisy perspective. Appl Soft Comput 46:731–752. https://doi.org/10.1016/j.asoc.2015.12.040

Konar D, Bhattacharyya S, Gandhi TK, Panigrahi BK (2020) A quantum-inspired self-supervised network model for automatic segmentation of brain MR images. Appl Soft Comput 93:106348. https://doi.org/10.1016/j.asoc.2020.106348

Konar D, Bhattacharyya S, Panigrahi BK, Behrman EC (2022) Qutrit-inspired fully self-supervised shallow quantum learning network for brain tumor segmentation. IEEE Trans Neural Netw Learn Syst 33(11):6331–6345. https://doi.org/10.1109/TNNLS.2021.3077188

Konar D, Bhattacharyya S, Dey S, Panigrahi BK (2022) Optimized activation for quantum-inspired self-supervised neural network based fully automated brain lesion segmentation. Appl Intell 52:15643–15672. https://doi.org/10.1007/s10489-021-03108-5

Konar D, Sarma AD, Bhandary S, Bhattacharyya S, Cangi A, Aggarwal V (2023) A shallow hybrid classical-quantum spiking feedforward neural network for noise-robust image classification. Appl Soft Comput 136:110099. https://doi.org/10.1016/j.asoc.2023.110099

Konar D, Bhattacharyya S, Gandhi TK (2024) Other: 3-D quantum-inspired self-supervised tensor network for volumetric segmentation of medical images. IEEE Trans Neural Netw Learn Syst 35(8):10312–10325. https://doi.org/10.1109/TNNLS.2023.3240238

Konar D, Aggarwal V, Das Sarma A, Bhandary S, Bhattacharyya S, Cangi A (2023) Deep spiking quantum neural network for noisy image classification. In: 2023 International joint conference on neural networks (IJCNN), pp 1–10. https://doi.org/10.1109/IJCNN54540.2023.10191509

Konar D, Gelenbe E, Bhandary S, Das Sarma A, Cangi A (2023) Random quantum neural networks for noisy image recognition. In: 2023 IEEE international conference on quantum computing and engineering (QCE), vol 02, pp 276–277. https://doi.org/10.1109/QCE57702.2023.10240

Konar D, Gelenbe E, Bhandary S, Sarma AD, Cangi A (2022) Random quantum neural networks (RQNN) for noisy image recognition. https://doi.org/10.48550/arXiv.2203.01764

Kossaifi J, Khanna A, Lipton Z, Furlanello T, Anandkumar A (2017) Tensor contraction layers for parsimonious deep nets. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) workshops, pp 1940–1946. https://doi.org/10.1109/CVPRW.2017.243

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25

Larocca MJN, García-Martín D et al (2023) Theory of overparametrization in quantum neural networks. Nat Comput Sci 3:542–551. https://doi.org/10.1038/s43588-023-00467-6

Lathuilière S, Mesejo P, Alameda-Pineda X, Radu H (2020) A comprehensive analysis of deep regression. IEEE Trans Pattern Anal Mach Intell 42(9):2065–2081. https://doi.org/10.1109/TPAMI.2019.2910523

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791

Li YC, Zhou R-G, Xu R, Luo J, Hu W (2020) A quantum deep convolutional neural network for image recognition. Quantum Sci Technol 5. https://doi.org/10.1088/2058-9565/ab9f93

Li J, Wen JG et al (2011) Robust tensor subspace learning for anomaly detection. Int J Mach Learn Cyber 2:89–98. https://doi.org/10.1007/s13042-011-0017-0

Li J, Yang X, Peng X, Sun C-P (2017) Hybrid quantum-classical approach to quantum optimal control. Phys Rev Lett 118(15):150503

Li S, Song W, Fang L, Chen Y, Pedram G, Benediktsson JA (2019) Deep learning for hyperspectral image classification: an overview. IEEE Trans Geosci Remote Sens 57(9):6690–6709. https://doi.org/10.1109/TGRS.2019.2907932

McClean JR, Boixo S, Smelyanskiy VN et al (2018) Barren plateaus in quantum neural network training landscapes. Nat Comm 9:4812. https://doi.org/10.1038/s41467-018-07090-4

Mitarai K, Negoro M, Kitagawa M, Fujii K (2018) Quantum circuit learning. Phys Rev A 98(3):032309. https://doi.org/10.1103/PhysRevA.98.032309

Molchanov P, Mallya A, Tyree S, Frosio I, Kautz J (2019) Importance estimation for neural network pruning. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11256–11264. https://doi.org/10.1109/CVPR.2019.01152

Novikov A, Podoprikhin D, Osokin A, Vetrov DP (2015) Tensorizing neural networks. In: Advances in neural information processing systems, vol 28

Orús R (2014) A practical introduction to tensor networks: matrix product states and projected entangled pair states. Ann Phys 349:117–158. https://doi.org/10.1016/j.aop.2014.06.013

Orús R (2019) Tensor networks for complex quantum systems. Nat Rev Phys 1:538–550. https://doi.org/10.1038/s42254-019-0086-7

Oseledets IV (2011) Tensor-train decomposition. SIAM J Sci Comput 33(5):2295–2317. https://doi.org/10.1137/090752286

Pan F, Zhang P (2022) Simulation of quantum circuits using the big-batch tensor network method. Phys Rev Lett 128:030501. https://doi.org/10.1103/PhysRevLett.128.030501

Panagakis Y, Kossaifi J, Chrysos GG, Oldfield J, Nicolaou MA, Anandkumar A, Zafeiriou S (2021) Tensor methods in computer vision and deep learning. Proc IEEE 109(5):863–890. https://doi.org/10.1109/JPROC.2021.3074329

Panagakis Y, Kossaifi J, Chrysos GG, Oldfield J, Nicolaou MA, Anandkumar A, Zafeiriou S (2021) Tensor methods in computer vision and deep learning. Proc IEEE 109(5):863–890. https://doi.org/10.1109/JPROC.2021.3074329

Pan Y, Xu J, Wang M, Ye J, Wang F, Bai K, Xu Z (2019) Compressing recurrent neural networks with tensor ring for action recognition. In: Proceedings of the Thirty-Third AAAI conference on artificial intelligence and thirty-first innovative applications of artificial intelligence conference and ninth AAAI symposium on educational advances in artificial intelligence. https://doi.org/10.1609/aaai.v33i01.33014683

Peddireddy D, Bansal V, Aggarwal V (2023) Classical simulation of variational quantum classifiers using tensor rings. Appl Soft Comput 141:110308. https://doi.org/10.1016/j.asoc.2023.110308

Peng W, Hong X, Chen H, Zhao G (2020) Learning graph convolutional network for skeleton-based human action recognition by neural searching. Proc AAAI Conf Artif Intell 34(03):2669–2676. https://doi.org/10.1609/aaai.v34i03.5652

Pescia G, Han J, Lovato A, Lu J, Carleo G (2022) Neural-network quantum states for periodic systems in continuous space. Phys Rev Res 4:023138. https://doi.org/10.1103/PhysRevResearch.4.023138

Phan AH, et al (2020) Stable low-rank tensor decomposition for compression of convolutional neural network. In: Computer Vision - ECCV 2020, vol 12374. https://doi.org/10.1007/978-3-030-58526-6_31

Qiu Y, Zhou G, Zhao Q, Xie S (2022) Noisy tensor completion via low-rank tensor ring. IEEE Trans Neural Netw Learn Syst 1–15. https://doi.org/10.1109/TNNLS.2022.3181378

Qiu Y, Zhou G, Zhao Q, Xie S (2022) Noisy tensor completion via low-rank tensor ring. IEEE Trans Neural Netw Learn Syst 1–15. https://doi.org/10.1109/TNNLS.2022.3181378

Rebentrost P, Bromley TR, Weedbrook C, Lloyd S (2018) Quantum Hopfield neural network. Phys Rev A 98(4):042308. https://doi.org/10.1103/PhysRevA.98.042308

Rem BS, Käming N, Tarnowski M et al (2019) Identifying quantum phase transitions using artificial neural networks on experimental data. Nat Phys 15:917–920. https://doi.org/10.1038/s41567-019-0554-0

Song D, Zhang P, Li F (2020) Speeding up deep convolutional neural networks based on tucker-cp decomposition. In: Proceedings of the 2020 5th international conference on machine learning technologies, pp 56–61. https://doi.org/10.1145/3409073.3409094

Stoudenmire EM, Schwab DJ (2016) Supervised learning with tensor networks. In: Advances in neural information processing systems, vol 29, pp 4799–4807. https://doi.org/10.48550/arXiv.1605.05775

Stoudenmire EM (2018) Learning relevant features of data with multiscale tensor networks. Mach Learn Sci Technol 3(3):034003. https://doi.org/10.1088/2058-9565/aaba1a

Uschmajew A, Vandereycken B (2013) The geometry of algorithms using hierarchical tensors. Linear Algebra Appl 439(1):133–166. https://doi.org/10.1016/j.laa.2013.03.016

Verstraete F, Murg V, Cirac JI (2008) Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. Adv Phys 57(2):143–224. https://doi.org/10.1080/14789940801912366

Wang W, Eriksson B, Sun Y, Wang W, Aggarwal V (2018) Wide compression: tensor ring nets. In: 2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 9329–9338. https://doi.org/10.1109/CVPR.2018.00972

White SR (1992) Density matrix formulation for quantum renormalization groups. Phys Rev Lett 69(19):2863–2866. https://doi.org/10.1103/PhysRevLett.69.2863

Xiao T, Huang J, Li H et al (2022) Intelligent certification for quantum simulators via machine learning. NPJ Quantum Inf 138(8). https://doi.org/10.1038/s41534-022-00649-6

Yang J, Shen X, Xing J, Tian X, Li H, Deng B, Huang J, Hua X-s (2019) Quantization networks. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 7300–7308. https://doi.org/10.1109/CVPR.2019.00748

Yu J, Zhou G, Li C, Zhao Q, Xie S (2021) Low tensor-ring rank completion by parallel matrix factorization. IEEE Trans Neural Netw Learn Syst 32(7):3020–3033. https://doi.org/10.1109/TNNLS.2020.3009210

Yuan L, Li C, Cao J, Zhao Q (2020) Rank minimization on tensor ring: an efficient approach for tensor decomposition and completion. Mach Learn 109(3):603–622. https://doi.org/10.1007/s10994-019-05846-7

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Computer Vision – ECCV 2014, pp 818–833. https://doi.org/10.1007/978-3-319-10590-1_53

Zhang Y, Mesaros A, Fujita K et al (2019) Machine learning in electronic-quantum-matter imaging experiments. Nature 570:484–490. https://doi.org/10.1038/s41586-019-1319-8

Zhao C, Gao XS (2021) QDNN: deep neural networks with quantum layers. Quantum Mach Intell 3(15). https://doi.org/10.1007/s42484-021-00046-w