Multi-class quantum classifiers with tensor network circuits for quantum phase recognition

Marco Lazzarin  
CNR-IFN, Milan, Italy  
Università degli Studi di Milano, Milan, Italy

Davide Emilio Galli  
Università degli Studi di Milano, Milan, Italy

Enrico Prati  
CNR-IFN, Milan, Italy

Abstract—Hybrid quantum-classical algorithms based on variational circuits are a promising approach to quantum machine learning problems for near-term devices, but the selection of the variational ansatz is an open issue. Recently, tensor network-inspired circuits have been proposed as a natural choice for such ansatz. Their employment on binary classification tasks provided encouraging results. However, their effectiveness on more difficult tasks is still unknown.

Here, we present numerical experiments on multi-class classifiers based on tree tensor network and multiscale entanglement renormalization ansatz circuits.

We conducted experiments on image classification with the MNIST dataset and on quantum phase recognition with the XXZ model by Cirq and TensorFlow Quantum. In the former case, we reduced the number of classes to four to match the aimed output based on 2 qubits. The quantum data of the XXZ model consist of three classes of ground states prepared by a checkerboard circuit used for the ansatz of the variational quantum eigensolver, corresponding to three distinct quantum phases. Test accuracy turned out to be 59%-93% and 82%-96% respectively, depending on the model architecture and on the type of preprocessing.

Index Terms—quantum machine learning, tensor networks, multi-class classification

I. INTRODUCTION

We implemented some multi-class variational classifiers based on tensor network circuits and we tested them on both classical and quantum data.

A lot of effort has been made recently to investigate exploitation of quantum computing [1], [2] to the field of machine learning [3], [4]. Currently, a major research direction focuses on applications of noisy intermediate-scale quantum (NISQ) [5] devices i.e. near-term hardware characterized by a small number of noisy qubits. Variational quantum classifiers [6] trained in a hybrid quantum-classical setup are one of them. They are classification algorithms that can natively process quantum data. Therefore, they provide a strategy to manage future quantum datasets which may be classically intractable. Nevertheless, they can also process classical datasets, once encoded in quantum states. However, the choice of the variational ansatz of such algorithms requires further investigations.

Tensor networks are mathematical objects used as variational ansätze to represent quantum states, initially developed in the field of condensed-matter physics [7]–[10]. However, their field of application turned out to be broader. For example, tensor networks have been applied to machine learning tasks, e.g. supervised [11]–[13] and unsupervised [14] learning. Then, they have been proposed as ansätze for variational quantum circuits applied to machine learning tasks [15], in both discriminative [16] and generative [17] learning. Previous works [15], [16] demonstrated promising results in binary classification problems, but the capabilities on more difficult tasks are still to be explored.

In this work we tested some circuit architectures based on tensor networks on multi-class classification tasks. In particular, we trained the circuits for digit recognition with a subset of the MNIST dataset [18] and for quantum phase recognition with the 1D XXZ model [19], [20]. In the former case, we compare the results with classical benchmarks. We examined circuits based on both tree tensor networks (TTN) [21] and multiscale entanglement renormalization ansätze (MERA) [22].

We adopted two different approaches to implement the multi-class setup, in the following referred to as amplitude decoding and qubit decoding with binary labels, respectively. The former retrieves the prediction from measurements in the computational basis. The latter uses binary labels and retrieves the prediction by computing the expectation value of single-qubit observables in some readout qubits [11], [23]. This research exploits circuits based on eight qubits.

In the case of digit recognition, the input data were compressed by dimensionality reduction methods. In order to reduce the scale of the classification task, we filtered the dataset to include only four out of ten classes. In the case of the XXZ model, we worked with a system of eight spins and with three different classes.

Test accuracy turned out to be 59%-93% for digit recognition and 82%-96% for quantum phase recognition, depending on the model architecture and on the type of preprocessing.

The document is organized as follows. In Sec. II we describe the implementation, while in Sec. III we present the numerical experiments. Finally, in Sec. IV we draw the conclusions.

II. METHODS

The variational classifiers are built upon those proposed in [15], [16] and modified to account for the multi-class embodiment. The core object of the algorithm is the variational ansatz i.e. a parametric circuit that takes the role of the model in a classical deep learning setup. We employed two different
circuit, based on a TTN (Fig. 1a) and on a MERA (Fig. 1b), respectively. The inference step consists of applying the circuit to the input quantum state, then computing the expectation value of some observables on the final state. In order to fit the input quantum state, classical data require an encoding. Then, the learning process involves a hybrid quantum-classical iteration in which a batch of input data is fed into the circuit, a loss function is computed by comparing the predictions with the exact labels, and finally a classical optimization algorithm updates the parameters of the circuit in order to improve the performance of the classifier [6]. Besides the nature of the parametric model, this procedure is similar to the training process of classical deep learning [25].

In order to implement a multi-class setup, a method to decode the predictions from the final states is required. As anticipated, we compare two different approaches, referred to as qubit decoding and amplitude decoding, respectively.

When the qubit decoding method is applied, we select $N$ readout qubits and we measure the expectation value of the operator $\hat{\sigma}_z$ for each readout qubit independently. The operator $\hat{\sigma} = \frac{1}{2} (1 + \hat{\sigma}_z)$ would be an equivalent choice. In this way, we obtain $N$ output values $\in [0,1]$ that can be fed into a Softmax layer in order to transform them into a probability mass distribution for $N$ classes. Instead, when the amplitude decoding method is applied, we select $n$ readout qubits and compute the probabilities of obtaining the $2^n$ elements of the computational basis by repeated measurements. Then, we obtain directly a probability mass distribution for $N = 2^n$ classes. We still apply a Softmax layer, so that the result is normalized even if the number of classes is not a power of two.

The former strategy requires a number of readout qubits that scales linearly with the number of labels, while it scales as a logarithm for the latter. Both strategies are compatible with one-hot encoded labels and with the categorical cross entropy loss function.

The linear scaling of the qubit decoding was troublesome for the scale of our circuits, therefore in our experiments we implemented a modified version of qubit decoding. Here, the $N$ output values $\in [0,1]$ are considered as the approximated prediction of a binary number. Then, by converting the labels of the dataset to binary numbers, we can obtain a logarithmic scaling of the readout qubits with respect to the number of labels. Notice that the binary labels are not equally distant (in terms of both the Hamming distance and the Euclidean distance), so a loss function like mean squared error will weight classification mistakes differently.

We report numerical experiments by using both qubit decoding with binary labels and amplitude decoding. We obtained comparable results, without a strong evidence of advantage of one approach with respect to the other. The details about the experiments and the results are discussed in the next Section.

III. EXPERIMENTS

The variational classifiers were built with Cirq [26] and trained with TensorFlow Quantum [27]. Data preprocessing and tests with other machine learning algorithms were performed with Scikit-learn [28] and Keras [29].

The choice of the parametric unitaries that implement the nodes of the networks was inspired by [16]. In particular, we used two simple unitaries that use only one CNOT, plus a general $SO(4)$ gate and a general $SU(4)$ gate. We compiled the unitaries with CNOTs and parametrized single-qubit rotations. The corresponding circuits are described in Fig. 2. When we used one of the two simple unitaries, we added single-qubit rotations to the readout qubits at the end of the circuit. Such single-qubit rotations were the same of those used in the unitaries.

In Sec. III-A we describe the tests that we conducted on the task of digit recognition with the MNIST dataset, while in Sec. III-B we describe those aimed to quantum phase recognition referring to XXZ model.

A. Digit recognition with MNIST

We apply the classifiers to the task of digit recognition on the MNIST dataset [13]. The dataset consists of 70000 grayscale images of 28x28 pixels depicting handwritten digits from zero to nine, subdivided between training and test set with a ratio 6:1. In order to use circuits with a small number of qubits, we reduced the dimensionality of the data from 784 to 8 by using two alternative methods. First, we applied Principal Component Analysis (PCA) [16], [32], by keeping only the components with higher variance. Second, we performed the dimensionality reduction with a convolutional autoencoder [33] to see whether a deep learning algorithm could improve the performance of the classifiers. Autoencoders are machine learning algorithms based on neural networks that are able to learn representations of the input data. Autoencoders are made of two submodels: an encoder that maps the data into a latent space and a decoder that reconstructs the input data from the latent space. The autoencoder is trained by optimizing the quality of the reconstruction i.e. a distance between the original example and its reconstruction. Once that the model is trained, the encoder can be used to encode the examples into the latent space, which is supposed to have a lower dimensionality than the original input space, so that a dimensionality reduction is achieved. An illustration of a convolutional autoencoder is presented in Fig. 3. while a description of the specific configuration we used is given in Table I.

We encoded the classical data in quantum circuits by using the qubit encoding i.e. by constructing a factorized state in which each predictor is encoded in the amplitude of a single-qubit wavefunction. This type of encoding requires a qubit for each predictor. In particular, we used the approach introduced in [11]. Let $X$ be the design matrix of the training set, with examples in the first index and predictors in the second index. At first, we scale the predictors to be $\in [0,1]$ within the training set. Then, we encode each example $X_{ix}$ in a quantum state using $R_Y$ rotations:

$$|\psi_i\rangle = R_Y(\pi x_{i1}) \otimes R_Y(\pi x_{i2}) \otimes ... \otimes R_Y(\pi x_{is}) |0\rangle^\otimes s$$

(1)
(a) Variational classifier based on a TTN.

(b) Variational classifier based on a MERA.

Fig. 1. Circuits used for the quantum classifiers. The nodes labelled by capital letters are parametric unitaries. In our experiments, we used those illustrated in Fig. 2. The input can be either an encoded classical state or a quantum state. The variational quantum circuit is applied to such input state. Then, some readout qubits are measured repeatedly to obtain the prediction. The number of readout qubits depends on the number of classification labels to predict and on the strategy used to decode the prediction from the final state of the circuit. Two readout qubits are used in the experiments of Sec. III. The representation of these circuits and of the following ones is made with Quantikz [24].

---

Fig. 2. Quantum circuits that implement the nodes of the tensor networks of our experiments. Circuit 2a and Circuit 2b are taken from [16], while the other two are derived in [31]. For the first two, the CNOT gates may be reversed to follow the causal structure of the network e.g. gates $B$, $D$ and $F$ of Fig. 1.
The gradient is computed with the default differentiator of decoding and qubit decoding with binary labels. In particular, we kept the digits ‘0’, ‘1’, ‘2’ and ‘3’ so the dimension two. They natively support at most two readout without using the elements of the test set.

The preprocessing transformations (dimensionality reduction and scaling) are applied to both the training and the test set, but the parameters of these transformations are computed again by a neural network to reconstruct the input.

![Fig. 3. Illustration of a convolutional autoencoder.](image)

**Table I**

| Layer                  | Output shape | Activation |
|------------------------|--------------|------------|
| Input                  | 28x28x1      |            |
| Convolutional layer    | 14x14x32     | ReLU       |
| Convolutional layer    | 7x7x64       | ReLU       |
| Fully connected layer  | 100          | ReLU       |
| Fully connected layer  | 8            | ReLU       |
| Fully connected layer  | 7x7x64       | ReLU       |
| Transposed conv. layer | 14x14x64     | ReLU       |
| Transposed conv. layer | 28x28x32     | ReLU       |
| Transposed conv. layer | 28x28x1      | Sigmoid    |

Configuration of convolutional layers

| Same padding | Filter size: 3x3 |
|--------------|------------------|
| Optimizer: Adam [34] | Epochs: 100 Batch size: 128 |
| Loss function: Binary Cross-entropy |

where \(x_{ij}\) are the elements of the design matrix \(X\). The encoding requires a number of one-qubit rotations that scales linearly with the number of predictors.

The preprocessing transformations (dimensionality reduction and scaling) are applied to both the training and the test set, but the parameters of these transformations are computed without using the elements of the test set.

The circuits of Fig. 1 are TTN and MERA circuits with bond dimension two. They natively support at most two readout qubits at the last node of the circuit. Therefore, we reduced the number of different labels of the dataset from ten to four. In particular, we kept the digits ‘0’, ‘1’, ‘2’ and ‘3’ so the required number of readout qubits is two for both amplitude decoding and qubit decoding with binary labels.

Concerning the training, we use the Adam [34] optimizer. The gradient is computed with the default differentiator of TensorFlow Quantum. We used batches of 20 examples, epochs of 10 batches and default learning rate (0.001). Moreover, we further split the training set between training and validation set (ratio 11:1) to use the Early Stopping method [25].

Let us first define the task. Consider a quantum system with a Hamiltonian \(H(\Delta)\) which presents different phases at different values of \(\Delta\). We can use the Variational Quantum Eigensolver algorithm (VQE) [36] to obtain a circuit that prepares a state \(|\psi(\Delta)\rangle\) which approximates the ground state of \(H(\Delta)\). Then, we can build a dataset of ground state circuits \{\(|\psi(\Delta)\rangle, y(\Delta)\)\}_{\Delta \in \Lambda}\), where \(|\psi(\Delta)\rangle\) is a variational approximation to the ground state wave function of \(H(\Delta)\), and \(y(\Delta)\) is a label for the quantum phase of the system and \(\Lambda\) is the set of different values of the parameter \(\Delta\) with which we want to generate the dataset.

We selected the one-dimensional XXZ model as our benchmark system. It has the following Hamiltonian, at least in absence of an external magnetic field:

\[
H(\Delta) = J \sum_{i=1}^{N} [\sigma_i^{x} \sigma_{i+1}^{x} + \sigma_i^{y} \sigma_{i+1}^{y} + \Delta \sigma_i^{z} \sigma_{i+1}^{z}] 
\]

Let us consider a spin chain of length \(N = 8\) with periodic boundary conditions i.e. \(\sigma_{i+N} = \sigma_i\). Let \(J = 1\) so that the antiferromagnetic order is favoured along the \(x-y\) plane. The parameter \(\Delta\) regulates the intensity of the \(z\)-axis anisotropy with respect to the planar \(x-y\) term and discriminates between axial (\(|\Delta| > 1\)) and planar (\(|\Delta| < 1\)) regimes. Regarding the axial regimes, for \(\Delta > 1\) we have a gapped antiferromagnet...
TABLE II
EXPERIMENTAL RESULTS WITH THE MNIST DATASET

| Classifier     | Unitaries | PCA                  | Test accuracya (%) | Convolutional autoencoder |
|----------------|-----------|----------------------|--------------------|---------------------------|
|                |           | Qubit decoding w/ binary labels | Amplitude decoding | Qubit decoding w/ binary labels | Amplitude decoding |
| TTN            | Simple SO(4) | 64 ± 5               | 59 ± 10            | 76.1 ± 1.2                | 76 ± 3           |
| TTN            | Simple SU(4) | 64 ± 7               | 69 ± 10            | 75 ± 4                    | 74 ± 6           |
| TTN            | General SO(4) | 77 ± 7               | 69 ± 5             | 81 ± 4                    | 80 ± 4           |
| TTN            | General SU(4) | 81.0 ± 1.6           | 68 ± 10            | 83 ± 3                    | 82 ± 5           |
| MERA           | General SO(4) | 82.8 ± 1.4           | 81 ± 7             | 90 ± 3                    | 91 ± 3           |
| MERA           | General SU(4) | 84.8 ± 1.6           | 85.0 ± 1.9         | 91.3 ± 1.0                | 93 ± 2           |
| Logistic regression |           | 94                   |                    |                          |                 |
| Neural network (186 parametersb) |           | 95                   |                    |                          |                 |

*aThe results of the quantum classifiers are averaged over five trials, with different random seeds. The measurement of uncertainty is the standard deviation.

*bThe architecture of the neural network was chosen in order to have a number of free parameters which is similar to that of the most expressive quantum circuit we used (that is 165).

along the z-axis, while for $\Delta < -1$ we have a gapped z-axis ferromagnet. For $|\Delta| < 1$ the system exhibits a gapless paramagnetic phase [19], [20].

We implemented the VQE by using TensorFlow Quantum with the Adam optimizer [34]. In particular, the optimization was performed by minimizing the mean squared distance between the energy of the variational state and a target energy, which was chosen to be lower than the exact ground state energy. We partially reused the code of [35]. The variational ansatz was a checkerboard tensor network made with general SU(4) gates. It is illustrated in Fig. 4.

We generated 1000 ground states for $\Delta \in [-2, 2]$. Then, we compared the ground state energies obtained with the VQE with those computed by exact diagonalization, resulting in relative errors up to 4-5%. While our results are probably suboptimal, we found out that it was enough to our goal.

The dataset consists of quantum data that can be fed directly to the classifiers, without any data encoding process. In practice, the examples consist of quantum circuits that can be readily added in front of the classifiers. The number of required operations depends on the complexity of the variational ansatz of the VQE.

We split the dataset in training and test set with ratio 2:1, then we trained the classifiers on the training set and we evaluated their performance on the test set. Concerning the training, we used the Adam [34] optimizer and the gradient was computed with the default differentiator of TensorFlow Quantum. We used batches of 8 examples and default learning rate (0.001). Moreover, we further split the training set in training and validation set (ratio 11:1) to use the Early Stopping method [25]. We set the patience of the Early Stopping to 250, monitoring validation accuracy. The maximum number of epochs was set to 1000. We tried both the qubit decoding with binary labels and amplitude decoding approaches presented in Sec. II, respectively with mean squared error and categorical cross entropy as loss functions.

The results are presented in Table III. Overall, the classifiers were quite accurate. Those implemented with qubit decoding with binary labels achieved a test accuracy of 87%-96%, depending on the model complexity. The results for the amplitude decoding were slightly worse (82%-90%).

IV. CONCLUSIONS

We implemented variational quantum classifiers based on TTNs and MERAs and we tested them on multi-class classifi-
c The results are averaged over five trials, with different random seeds. The measurement of uncertainty is the standard deviation.

| Classifier | Unitaries | Test accuracy (%) | Qubit dec. | Amplitude dec. |
|------------|-----------|-------------------|------------|----------------|
| TTN        | Simple SO(4) | 2a | 87 ± 4 | 82 ± 5 |
| TTN        | Simple SU(4) | 2b | 91 ± 7 | 82 ± 10 |
| TTN        | General SO(4) | 2c | 94 ± 2 | 82 ± 4  |
| TTN        | General SU(4) | 2d | 91 ± 4 | 86 ± 6  |
| MERA       | General SO(4) | 2e | 95.6 ± 0.7 | 90 ± 7 |
| MERA       | General SU(4) | 2f | 96.0 ± 0.5 | 90 ± 5 |

For quantum data like the dataset of XXZ ground states obtained a classification accuracy of 59%-93% and 82%-96% respectively, depending on the class of tensor network, the choice of the unitaries, the choice of the decoding method and the type of preprocessing. MERAs perform better than TTNs. The results with four classes are promising as networks with higher bond dimension may be able to manage more complicated datasets, provided that an efficient implementation method is available. In particular, such networks require the implementation of multi-qubit parametric unitaries.

For quantum data like the dataset of XXZ ground states considered above, the elements of the dataset are quantum states. Providing such states as input to a classical machine learning algorithm may be intractable, while the quantum classifiers presented here can process such states natively. The computations underlying this work have been simulated on classical hardware. Future advances of quantum processors and quantum algorithms may provide quantum datasets which are not classically tractable, so that development of quantum classifiers may turn out crucial.

ACKNOWLEDGMENT

Computational resources were provided by INDACO Platform, which is a project of High Performance Computing at the University of Milan.

REFERENCES

[1] D. Rotta, F. Sebastiano, E. Charbon, and E. Prati, “Quantum information density scaling and qubit operation time constraints of CMOS silicon-based quantum computer architectures,” npj Quantum Information, vol. 3, no. 1, pp. 1–14, 2017.

[2] E. Ferraro and E. Prati, “Is all-electrical silicon quantum computing feasible in the long term?” Physics Letters A, vol. 384, no. 17, p. 126352, 2020.

[3] L. Rocutto, C. Destri, and E. Prati, “Quantum semantic learning by reverse annealing of an adiabatic quantum computer,” Advanced Quantum Technologies, vol. 4, no. 2, p. 2000133, 2021.

[4] M. Maronese and E. Prati, “A continuous Rosenblatt quantum perceptron,” International Journal of Quantum Information, p. 2140002, 2021.

[5] J. Preskill, “Quantum computing in the NISQ era and beyond,” Quantum, vol. 2, p. 79, Aug. 2018, issn: 2521-327X. doi: 10.22331/q-2018-08-06-79.

[6] E. Farhi and H. Neven, “Classification with quantum neural networks on near term processors,” arXiv e-prints, Feb. 2018. arXiv:1802.06002 [quant-ph].

[7] R. Örús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” Annals of Physics, vol. 349, pp. 117–158, 2014, issn: 0003-4916. doi: 10.1016/j.aop.2014.06.013 arXiv:1306.2164 [cond-mat.str-el].

[8] F. Verstraete, V. Murg, and J. Cirac, “Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems,” Advances in Physics, vol. 57, no. 2, pp. 143–224, 2008. doi: 10.1080/14789940801912366 arXiv:0907.2796 [quant-ph].

[9] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” Annals of Physics, vol. 326, no. 1, pp. 96–192, 2011, January 2011 Special Issue, issn: 0003-4916. doi: 10.1016/j.aop.2010.09.012. arXiv:1008.3477 [cond-mat.str-el].

[10] R. Orús, “Tensor networks for complex quantum systems,” Nature Reviews Physics, vol. 1, no. 9, pp. 538–550, Aug. 2019. doi: 10.1038/s42254-019-0086-7 arXiv:1812.04011 [cond-mat.str-el].

[11] E. Stoudenmire and D. J. Schwab, “Supervised learning with tensor networks,” in Advances in Neural Information Processing Systems 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., Curran Associates, Inc., 2016, pp. 4799–4807. [Online]. Available: http://papers.nips.cc/paper/6211-supervised-learning-with-tensor-networks.pdf.

[12] S. Efthymiou, J. Hidary, and S. Leichenauer, “TensorNet for machine learning,” arXiv e-prints, Jun. 2019. arXiv:1906.06329 [cs.LG].

[13] Z.-Z. Sun, C. Peng, D. Liu, S.-J. Ran, and G. Su, “Generative tensor network classification model for supervised machine learning,” Phys. Rev. B, vol. 101, p. 075135, 7 Feb. 2020. doi: 10.1103/PhysRevB.101.075135 arXiv:1903.10742 [cs.LG].

[14] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, “Unsupervised generative modeling using matrix product states,” Phys. Rev. X, vol. 8, p. 031012, 3 Jul. 2018. doi: 10.1103/PhysRevX.8.031012 arXiv:1709.01662 [cond-mat.stat-mech].

[15] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, “Towards quantum machine
