Artificial Intelligence Evolution
http://ojs.wiserpub.com/index.php/AIE/

UNIVERSAL WISER PUBLISHER

Research Article

Investigating the Generalization Ability of Parameterized Quantum Circuits with Hierarchical Structures

Runheng Ran, Haozhen Situ*

College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
E-mail: situhaozhen@gmail.com

Received: 13 March 2021; **Revised:** 12 April 2021; **Accepted:** 27 April 2021

Abstract: Quantum computing provides prospects for improving machine learning, which are mainly achieved through two aspects, one is to accelerate the calculation, and the other is to improve the performance of the model. As an important feature of machine learning models, generalization ability characterizes models' ability to predict unknown data. Aiming at the question of whether the quantum machine learning model provides reliable generalization ability, quantum circuits with hierarchical structures are explored to classify classical data as well as quantum state data. We also compare three different derivative-free optimization methods, i.e., Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Constrained Optimization by Linear Approximation (COBYLA) and Powell. Numerical results show that these quantum circuits have good performance in terms of trainability and generalization ability.

***Keywords*:** generalization ability, parameterized quantum circuit, classification

1. Introduction

The suitable application of quantum computing in the near future is hybrid quantum classical algorithm, namely parameterized quantum circuit (PQC) [1]. The main reason for its popularity is the emergence of noisy intermediate-scale quantum (NISQ) [2] devices. PQC has become a hot spot of academic research in the field of quantum machine learning (QML) [3] in recent years. Benedetti et al. [1] introduced this hybrid algorithm and its application in quantum machine learning in detail, and also listed several new feature encoding techniques. In this hybrid method, the quantum computer prepares the quantum state and evolves the state based on the parameters generated by the classical computer. The measurement result of the final state is used by the classical computer to update the parameters. After successful training, the optimal parameters are found and the PQC can make accurate prediction. Killoran et al. [4] proposed to construct a continuous quantum neural network in a hierarchical structure, and showed the ability and adaptability of the model to perform machine learning tasks. Du et al. [5] studied the expressive ability of parameterized quantum circuits and showed that highly entangled quantum circuits have stronger expressive ability than neural networks and provide polynomial parameter complexity better than classical neural networks. Facts have shown that PQC-based machine learning models perform well, such as low-depth quantum circuits for supervised learning tasks [6], hierarchical quantum circuits for classical and quantum data classification [7], quantum convolutional neural networks for quantum phase recognition (QPR) and quantum error correction (QEC) [8].

Nevertheless, there are still many unanswered questions regarding the advantages that quantum computing can bring to machine learning. In the classical field, people put forward some evaluation indicators, such as empirical risk,

DOI: https://doi.org/10.37256/aie.212021826

generalization ability, etc [9]. For supervised learning tasks, it is required to achieve a balance between minimizing empirical risk and maximizing generalization ability, that is, it can learn training data well while ensuring that it has the same good predictive ability for unknown data. In machine learning (ML), especially neural networks (NN), learners with too strong learning ability often lead to over-fitting problems due to excessive capture of irrelevant features [10]. Early stopping and regularization methods are usually used to alleviate this issue, but it cannot be completely solved [11-12]. Jiang et al. [13] proposed a quantum neural network composed of parametric single-qubit rotation gates and controlled-NOT (CNOT) gates, and found that it has better generalization ability than neural networks with similar structures on multiple datasets. However, research on the generalization ability of PQC on quantum state data and the influence of different layouts of quantum circuits remain to be carried out.

In recent years, tensor networks have been widely used in many-body physics to simulate strongly correlated quantum systems, and can be used to represent quantum states and quantum circuits. A tensor network with a hierarchical structure has many similarities to a neural network, and in some cases, it has been shown that machine learning based on PQCs is equivalent to a machine learning algorithm based on tensor networks [14-15]. Thus, it becomes the natural choice for exploring quantum machine learning.

This article explores quantum circuits with hierarchical structures, compares quantum circuits with different hierarchical structures, performs classification tasks on classical data as well as quantum state data, and compares three derivative-free optimization methods for parameter training. The numerical results show that the quantum circuits with hierarchical structures provide reliable generalization ability on classical and quantum data. Moreover, the quantum circuit with stronger expressive ability has faster training speed and higher generalization ability.

The structure of this paper is as follows: In Section 2, we introduce the necessary processes involved in the training of quantum circuits. In Section 3, we introduce the classification datasets and the indicators for evaluating generalization capability covered in this paper. In Section 4, we introduce and discuss numerical simulation on classical data and quantum data respectively. In Section 5, we summarize and look into the future direction of this work.

2. Hierarchical parameterized quantum circuit

Parameterized quantum circuits consist of quantum gate operations with tunable parameters, e.g., the rotation angles of rotation gates. A sample is first encoded into a quantum state through feature mapping. Then the quantum state becomes the input of the circuits. The output of the quantum circuit can be extracted from the classical post-processing function applied to the measurement result of the final state. The tunable parameters are optimized by minimizing the loss function, which is defined according to the specific task. The quantum circuit and the classical optimizer work alternately until the optimal parameters are found. Generally, the hybrid quantum-classical algorithm consists of the following five steps:

1. Encode the classical data into the quantum state, if the data is quantum, this step is ignored;
2. Evolve the quantum state through the parameterized quantum circuit;
3. Measure the target qubit of the final quantum state;
4. Quantify the loss function of the difference between the predicted value and the true value;
5. Optimize tunable parameters;

These steps repeat until the optimal parameters are found.

2.1 *Data encoding*

A simple method to realize feature mapping of classical data is product coding [16], that is, each element of the classical data vector is encoded with the amplitude of a single qubit. This encoding method requires N qubits to encode an N-dimensional data vector, so the spatial efficiency is low. However, from the point of view of time, state preparation is obviously effective because it only requires single-qubit gates. The specific method is to scale each feature of the data vector to $[0, \pi/2]$, and then for each scaled sample x, each attribute is rotated through y-axis rotation to obtain the input state that can be used in the quantum circuit

$$|x\rangle = R_y(2x_1)|0\rangle \otimes ... \otimes R_y(2x_n)|0\rangle \quad (1)$$

In the case of quantum state data, there is no need to do any encoding, and the quantum state can be fed into the quantum circuit directly.

2.2 *Quantum circuit*

In this section, we will introduce several hierarchical quantum circuits. The first circuit structure is a tree tensor network (TTN) ansatz, which is inspired by a binary tree. This circuit structure first applies unitary operations to two adjacent input qubits in turn, and then discards one of the output qubits, so the number of qubits in the next layer is halved. In the next layer, the same method is applied. This process is repeated until only one qubit remains.

The second circuit structure we consider is the multi-scale entanglement renormalization ansatz (MERA). This circuit structure is characterized by adding an additional unitary operation layer before each layer of the TTN circuit. The application of additional layers can better capture the quantum correlation, and make the circuit has a better expressive ability.

The third circuit structure we consider is the matrix product state (MPS) ansatz. The characteristic of this circuit is that each layer has only one unitary operation acting on two qubits, and discards one of the output qubits. Layers are piled up until only one qubit remains. Figures 1a, 1b, and 1c are circuit diagrams of 4-qubit TTN, MERA and MPS circuits respectively. Each rectangle is a unitary operation, including the y-axis rotation of each qubit, and then the CNOT gate.

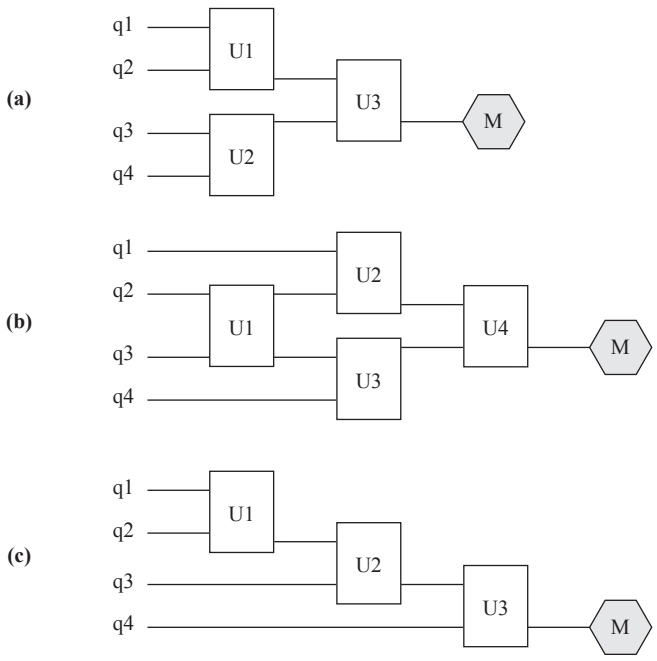


Figure 1. Four-qubit quantum circuits with hierarchical structures. a) TTN; b) MERA; c) MPS

2.3 *Measurement*

The projection measurement is performed on the remaining qubit $|\psi^d\rangle$. The measurement output probabilities are denoted as P(0) and P(1). If P(0) < 0.5, the output label is set to 0, otherwise the output label is set to 1. Block M in Figure 1 marks the measured qubit.

2.4 *Loss function*

We use binary cross entropy as the training objective function

$$L(\vec{\theta}) = -(y_i \ln(\mathrm{P}(1)) + (1 - y_i) \ln(P(0))) \tag{2}$$

The loss is calculated as the average binary cross entropy over a batch of samples.

2.5 *Data encoding*

The training of parametric quantum circuits relies on classical optimization. The optimizer tries to find the optimal or approximate optimal parameters to minimize the loss function. There are optimization methods that use gradients, but for these methods there is a training landscape of barren plateau [17], and the calculation of gradients is often time-consuming. In this work, we explore the effect of three popular derivative-free optimization methods.

The first is CMA-ES, the covariance matrix adaptation evolution strategy, which is mainly used to solve continuous optimization problems, especially continuous optimization problems under ill-conditioned conditions. It will dynamically adjust the step size to prevent the algorithm from falling into the local optimum. The evolution strategy will initially generate candidate solutions for the model parameters based on the initial standard deviation. We set the initial standard deviation to 0.5 in the numerical experiment.

The second is COBYLA. It's a sequential trust zone algorithm, which uses a linear approximation to the objective function and the constraint function, where the approximation is formed by linear interpolation at n + 1 points in the variable space, and tries to be in the iterative process simplex that maintains regular shapes. This method keeps iterating along the estimated slope as much as possible, and keeps it within the lower limit of the radius of the trust zone. When the algorithm detects that it has stopped iterating, the lower limit of the trust zone size is reduced, thus converging. The final accuracy of the lower limit of the trust zone size is set to 10^{-12}.

The last is Powell, which is a derivative-free direct search method formed by using the conjugate direction to accelerate the convergence rate. This method uses a one-dimensional search instead of a jump probe step. In this work, we use an improved Powell, which helps to escape the degraded, pathological situation that is linearly related to the search direction. In addition, an initial search vector set needs to be chosen, and it's usually the coordinate directions in the parameter space.

3. Classification tasks

Common machine learning tasks include regression and classification. This article focuses on the latter, especially the binary classification problem. In this section, we will first introduce the binary classification problem and the dataset we use. Then we propose an indicator for evaluating the generalization ability of a parameterized quantum circuit.

3.1 *Binary classification*

Classification is supervised learning, and the training dataset of the binary classification problem is a finite set of size m:

$$S_{train} \coloneqq \{x_i, y_i\}_{i=1}^{i=m} \tag{3}$$

where x_i is the N-dimensional input vector, $y_i \in \{0, 1\}$. The goal is to find a model f that predicts y_i with the highest accuracy on the training set S_{train}. At the same time, generalization ability requires good prediction on the so-called invisible data, that is, on the test set.

3.2 *Datasets*

For classical data, the typical machine learning dataset Iris is used. The dataset contains a total of 150 samples, divided into 3 categories y = 1, 2, 3. Each category has 50 samples, and each sample contains 4 attributes. Another classical dataset is Hayes-Roth [18]. This dataset has a total of 160 samples, which are divided into 3 categories. Each sample has 4 attributes, namely hobbies, age, education level and marital status. Noise is added by default in this dataset.

For quantum state data, we follow the synthesis method in [7], that is, we define a quantum circuit with N = 4 qubits, and generate 10 types of samples by increasing the depth of the circuit. The parameters of the circuit are initialized randomly, and 500 data samples are generated for each class. Finally, we synthesize two meaningful classification tasks: type-2 vs type-5 and type-1 vs type-10, that is, to distinguish between the second and fifth categories, and the first and tenth categories. Figure 2 shows the maximum bipartite entanglement entropy distribution of these two classification tasks. It can be seen from Figure 2 that the overlapping part of the task "1 or 10" is less than that of "2 or 5", and it is expected that the classification accuracy may be higher accordingly.

For each classification task, 80% of the data is used as the training set to train the hierarchical quantum circuits, and 20% of the data is used as the test set to evaluate the generalization ability. The samples of the training set and the test set are obtained by random sampling.

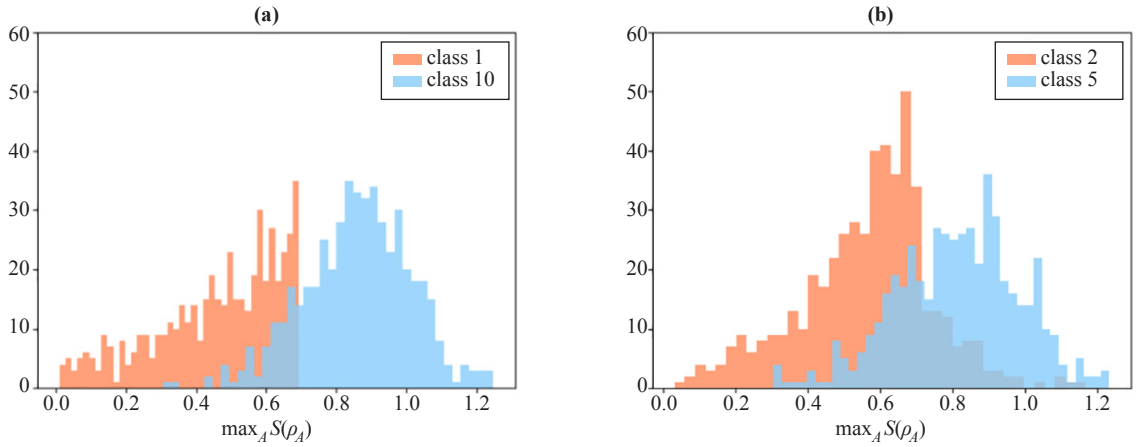


Figure 2. The maximum bipartite entanglement entropy distribution of the quantum state dataset: (a) class 1 vs class 10; (b) class 2 vs class 5

3.3 *Generalization ability*

Usually, the test set is used to evaluate the discriminative ability of the learning model for new samples, and the generalization error is approximated by the test error on the test set to reflect the generalization ability. Model complexity may also affect its performance. The receiver operating characteristic (ROC) curve is a probability curve, which comprehensively considers the quality of the "expected generalization performance" of the learning model under different tasks. The horizontal axis of the ROC is the false positive rate (*FPR*, that is, the probability that a negative sample is classified to a positive class), and the vertical axis is the true class rate (*TPR*, the probability that a positive sample is classified to a positive class).

$$FPR = \frac{FP}{N} \tag{4}$$

$$TPR = \frac{TP}{P} \tag{5}$$

where N is the number of negative samples, P is the number of positive samples, FP is the number of negative samples that are judged as positive, and TP is the number of positive samples that are judged as positive.

For different classifiers, the closer the ROC curve is to the upper left, the better the generalization effect. When two ROC curves cross each other, it is impossible to directly determine which classifier is better. The AUC (Area Under Curve) is defined as the area under the ROC curve. The larger the AUC value of the classifier, the better the generalization ability. The calculation method of AUC is as follows:

$$AUC = 1 - \sum_{x^+ \in D^+} \frac{1}{2} \cdot \frac{1}{m^+} \left(\frac{2}{m^-} \sum_{x^- \in D^-} \mathrm{II}(f(x^+) < f(x^-)) + \frac{1}{m^-} \sum_{x^- \in D^-} \mathrm{II}(f(x^+) = f(x^-)) \right) \tag{6}$$

where D^+ and D^- represent the set of real positive and real negative samples respectively, m^+ and m^- are the number of samples in the D^+ and D^+ sets, respectively, II represents the index function that returns 1 when the condition is met, f is the probability value corresponding to positive for the prediction of the sample.

4. Numerical results

4.1 *Iris experiment*

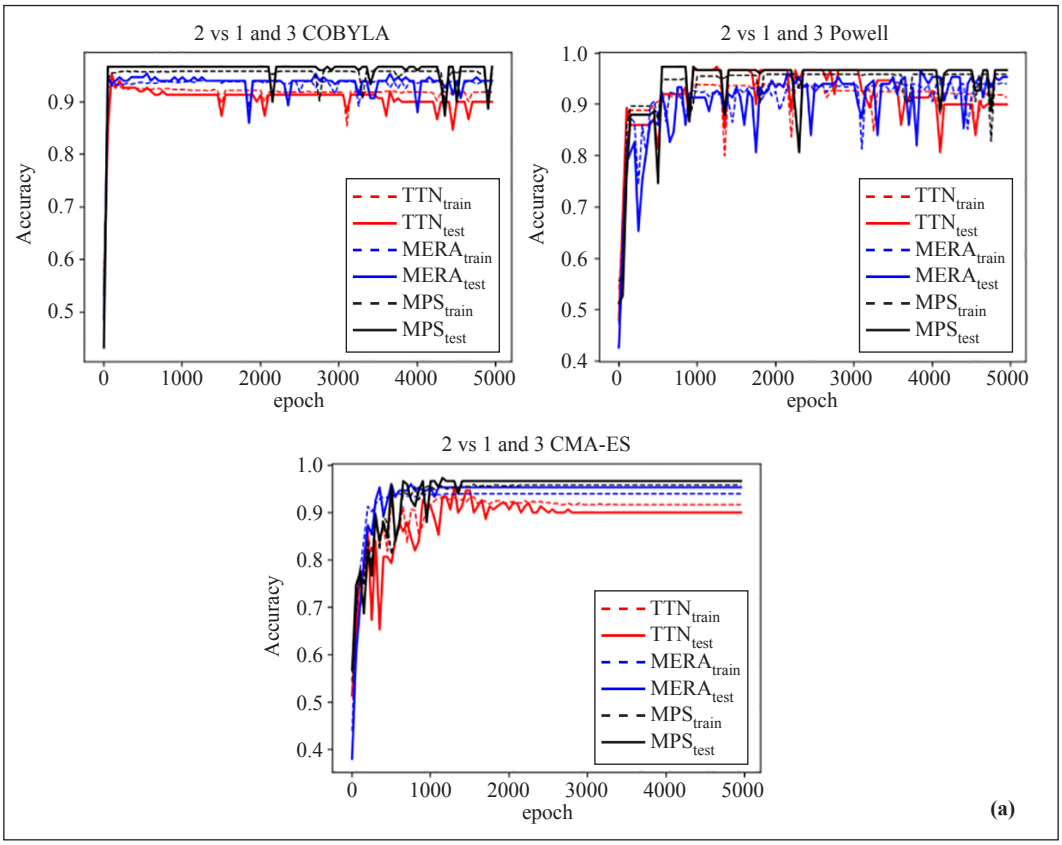
Each sample of the Iris dataset is encoded into a 4-qubit quantum state by the product encoding method of formula (1) introduced in Section 2.1. We create three binary classification tasks, namely 1 vs 2 and 3, 2 vs 1 and 3, and 3 vs 1 and 2. For example, 1 vs 2 and 3 indicate whether it is class 1.

Table 1. Binary classification accuracy of Iris dataset (unit: %)

Classifier	Optimizer	1 vs 2 and 3	2 vs 1 and 3	3 vs 1 and 2
TTN	CMA-ES	100.00	90.00	100.00
	Powell	100.00	90.00	100.00
	COBYLA	100.00	90.00	100.00
MERA	CMA-ES	100.00	95.33	100.00
	Powell	100.00	95.33	100.00
	COBYLA	100.00	94.00	100.00
MPS	CMA-ES	100.00	96.67	96.67
	Powell	100.00	96.67	96.67
	COBYLA	100.00	96.67	96.67

Table 1 shows the average accuracy of five random initializations of three quantum circuits of TTN, MERA and MPS. For each task, the three derivative-free optimization methods in Section 2.5 are used for comparison. The numerical values are average classification accuracies on the test set.

First of all, we can see the difficulty of the three tasks from Table 1, 1 vs 2 and 3 < 3 vs 1 and 2 < 2 vs 1 and 3. The three classifiers under each task performed very well, and the average accuracy exceeds 90%, and reaches 100% in the simple task of 1 vs 2 and 3. It can be seen from Table 1 that there is almost no difference in the accuracy achieved by different optimization methods. Secondly, the MERA classifier performs better than the TTN classifier in the case of more complex tasks (2 vs 1 and 3), and can provide better generalization ability, which reveals the powerful function of the additional unitary blocks.



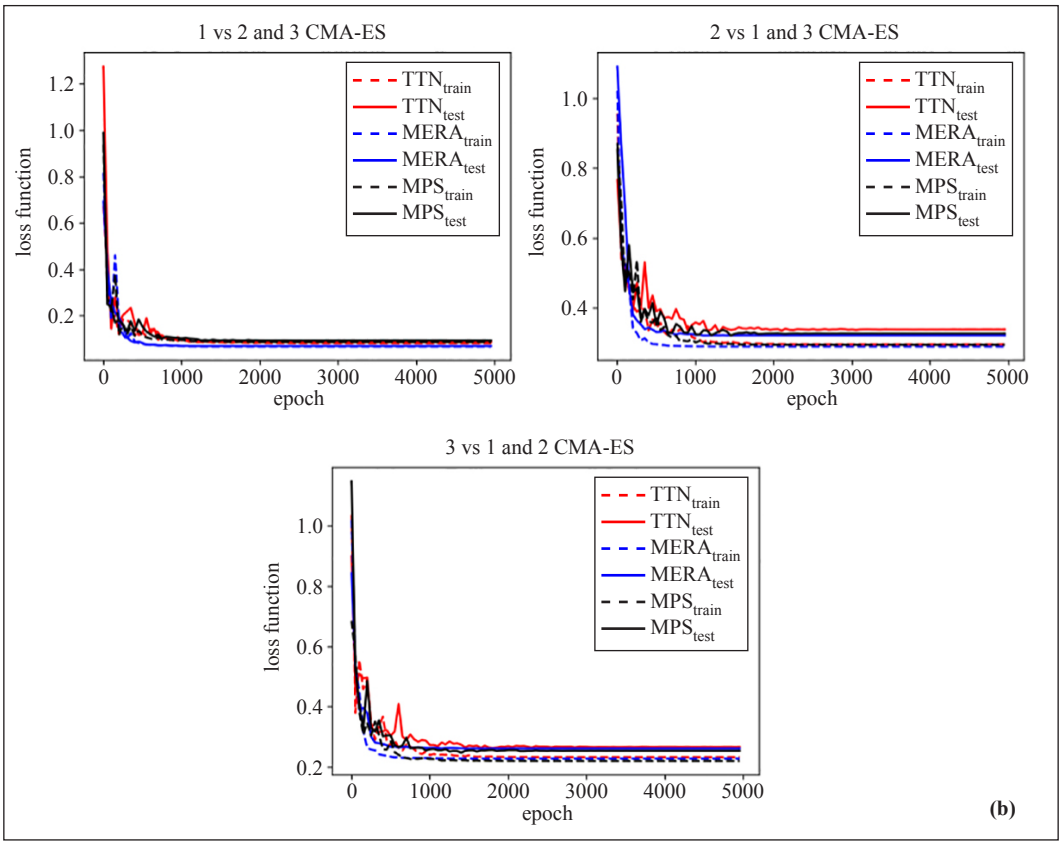


Figure 3. Training process on Iris dataset: the variation of (a) the classification accuracy and (b)the loss function

Figure 3 depicts the average accuracy and loss variation during the training process of five random initializations. Some interesting results were found. First, from Figure 3a, we can find that as the training progresses, all the hierarchical classifiers have similar fitting processes on the training set and the test set, there is no problem of overfitting. This shows that the hierarchical classifiers have good generalization ability, even different optimizers can achieve similar classification accuracy. Second, in Figure 3a, different classifiers can converge very well and have similar convergence speeds. It is worth noting that under different tasks, the convergence speed of the MERA classifier is faster than that of TTN, which reveals that the quantum circuit with better expressive ability has faster training speed. Figure 3b shows the training performance of the CMA-ES optimizer on three tasks. It can be seen that the loss function drops and converges very quickly.

In order to compare the generalization ability of the classifiers more explicitly, we use the test set to draw the average ROC curve of five random initializations, and calculate the AUC value. Figure 4 shows the ROC and AUC results of the hierarchical classifiers for the three tasks. It can be inferred from Figure 4 that under the three tasks, the AUC value of the MERA classifier is greater than TTN, and the ROC curve of the MERA classifier in tasks 2 vs 1 and 3 encloses TTN and MPS, revealing the generalization ability of the MERA classifier better.

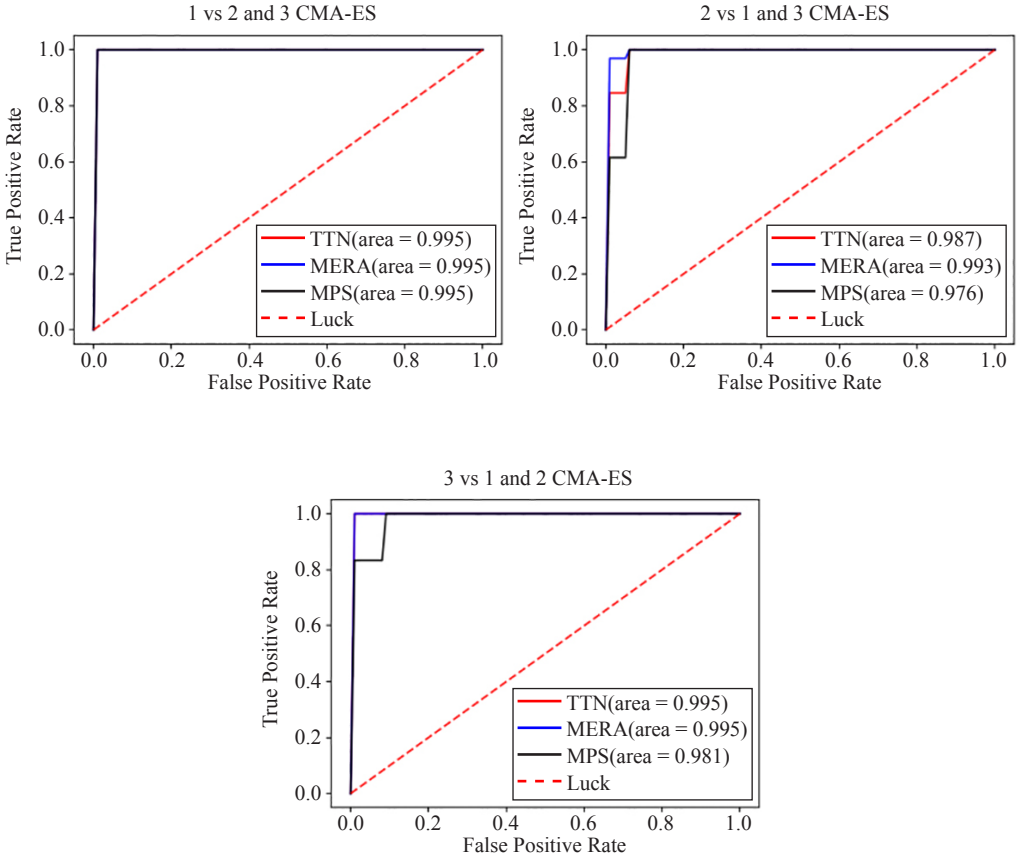


Figure 4. ROC_AUC curve of Iris dataset classification

4.2 *Hayes-Roth experiment*

In order to perform binary classification, three binary datasets are extracted from the original dataset, namely "1 or 2", "2 or 3", and "1 or 3". The samples in each class correspond to all the samples of that class in the original dataset. Since it is a dataset of classical data, the data encoding step is indispensable, and the same encoding method as the Iris dataset is used.

On the Hayes-Roth dataset, Figure 5 shows the typical training results of five random initializations optimized with CMA-ES. It can be intuitively found that all classifiers converge to a stable level and there is no over-fitting phenomenon. It is important that the MERA classifier has better classification accuracy and faster convergence speed than TTN.

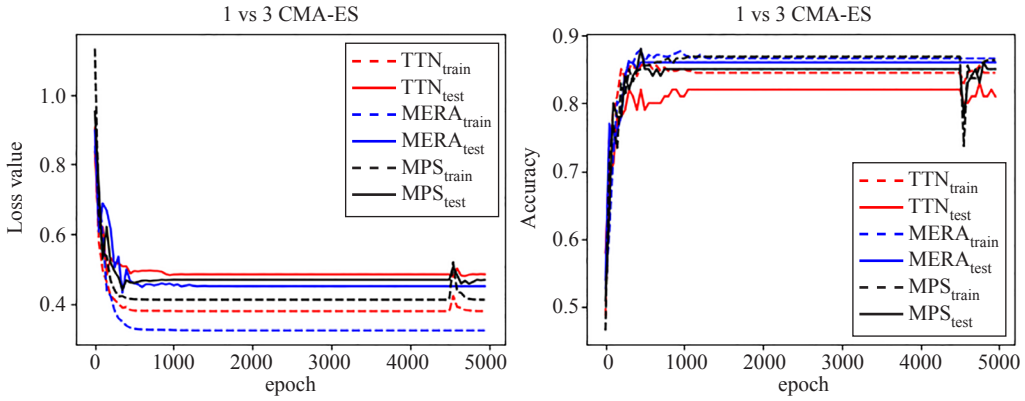


Figure 5. Training process on Hayer-Roth dataset

Table 2. Binary classification accuracy of Hayes-Roth dataset (unit: %)

Classifier	Optimizer	1 or 2	2 or 3	1 or 3
TTN	CMA-ES	69.23	73.68	82.00
MERA	CMA-ES	76.92	82.10	85.53
MPS	CMA-ES	73.08	78.95	84.21

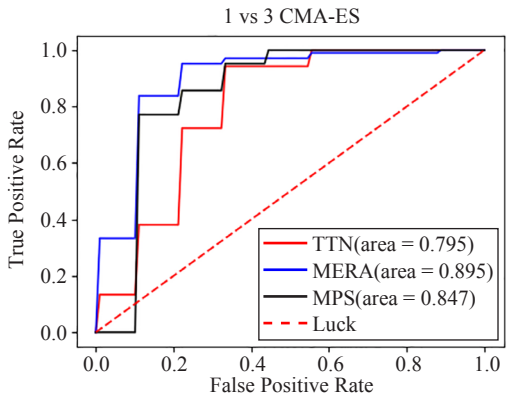


Figure 6. ROC_AUC curve of Hayes-Roth dataset classification

Finally, the average classification accuracy of the hierarchical quantum classifier on the test set is calculated. Table 2 shows these results. The numerical values are average classification accuracies of five random initializations. We can see that MERA classifier performs the best in all tasks, followed by MPS classifier and TTN classifier. The ROC-AUC curve of the task "1 or 3" is shown in Figure 6. The hierarchical quantum classifiers are also suitable for Hayes-Roth

dataset tasks, which further illustrates the reliable generalization ability of these quantum circuits. Moreover, the MERA classifier with stronger expression ability has better generalization ability, and the performance difference between the test set and the training set is smaller.

4.3 *Quantum state classification experiment*

The optimizer used for the quantum state classification task is the CMA-ES in Section 2.5. Figure 7 shows the training process of five random initializations for two tasks. The training is carried out for 5000 epoch iterations, and the average training accuracy (dashed line) and average test accuracy (solid line) are recorded every 50 epochs.

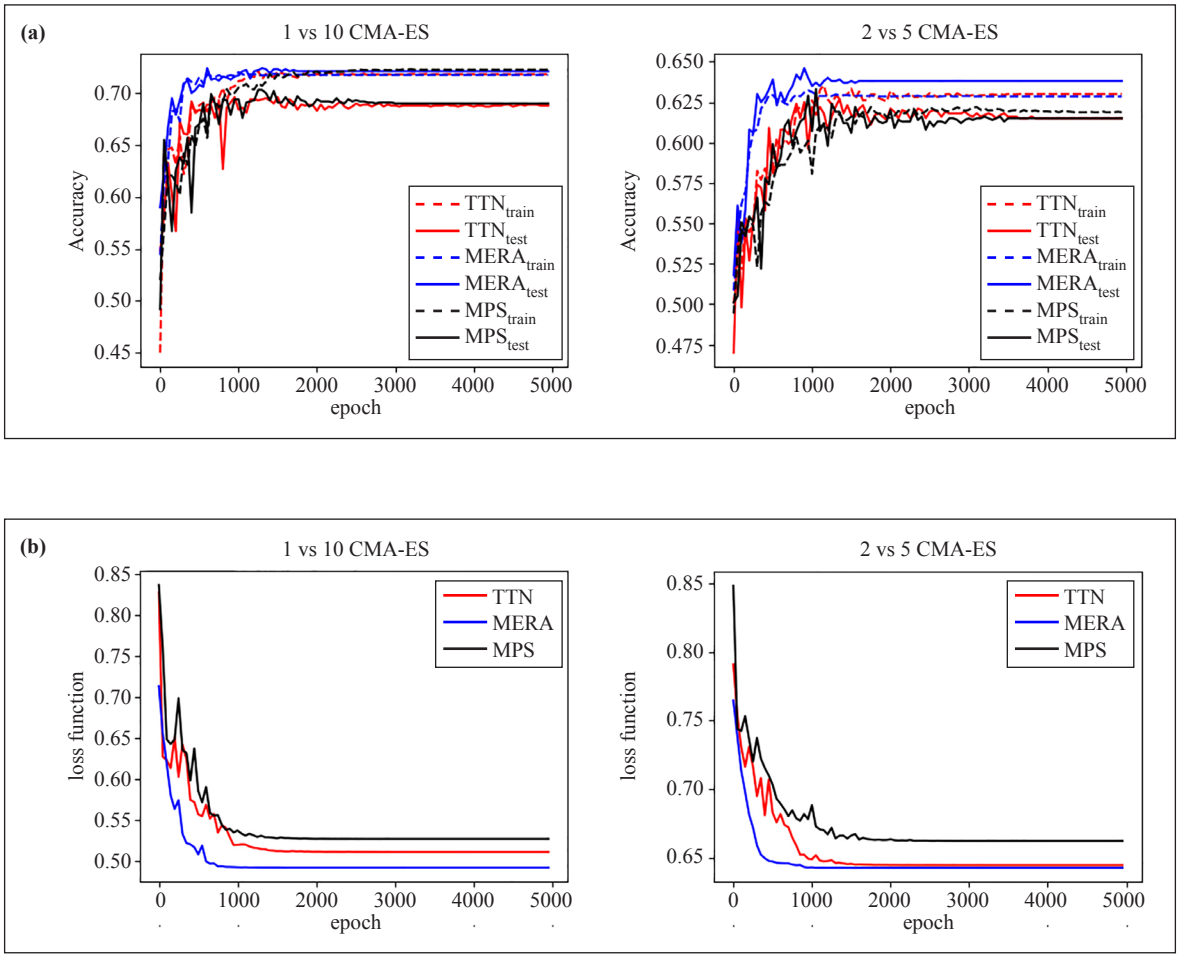


Figure 7. The training process of the quantum state dataset: the variation of (a) the classification accuracy and (b) the loss function

In Figure 7a, similar to the Iris classical dataset, the hierarchical quantum classifiers have similar classification accuracy and training convergence trend on the training set and the test set. The interesting finding is that in Figure 7a, the classification accuracy of the MERA classifier on the test set is slightly greater than that of the training set, which is not available on the TTN and MPS classifiers. Figure 7b shows the convergence of the loss function. On average, the MERA classifier requires the least training steps, and the training error is the smallest. This verifies that quantum models with better expressive capabilities provide faster training speed. The average classification accuracy of the task "1 or 10" is higher, which is also in accordance with the distribution overlap of the dataset. It can be inferred from Figure 8 that in more complex tasks (2 or 5), the performance gap between the MERA classifier and other classifiers increases.

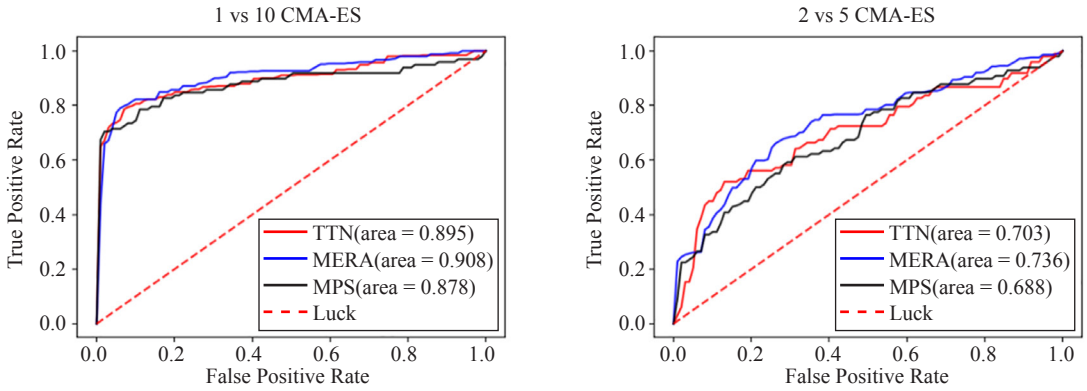


Figure 8. ROC_AUC curve of quantum state dataset classification

5. Conclusion

The quantum machine learning model that introduces the ability of quantum computing has an attractive development prospect. This paper evaluates the generalization ability of hierarchical quantum classifiers on classical data as well as quantum state data. Numerical results show that the quantum models with hierarchical structures have similar classification accuracy and similar training trends on the training set and test set data, confirming that it can provide excellent generalization ability. At the same time, a quantum model with better expressive ability can obtain better generalization ability and faster training speed. As for the influence of different derivative-free parameter learning methods, only the convergence speed is different. Finally, it can be found that in the classification of quantum state data, the upper limit of the classification accuracy achieved is less than 80%. Trying quantum circuits with more complex structures or using auxiliary qubits [7] will be the main direction of future work.

References

[1] Benedetti M, Lloyd E, Sack S, Fiorentini M. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*. 2019; 4: 043001.
[2] Preskill J. Quantum computing in the NISQ era and beyond. *Quantum*. 2018; 2: 79.
[3] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. *Nature*. 2017; 549: 195-202.
[4] Killoran N, Bromley TR, Arrazola JM, Schuld M, Quesada N, Lloyd S, et al. Continuous-variable quantum neural networks. *Physical Review Research*. 2019; 1: 033063.
[5] Du Y, Hsieh MH, Liu T, Tao D. The expressive power of parameterized quantum circuits. *Physical Review Research*. 2020; 2: 033125.
[6] Schuld M, Bocharov A, Svore K, Wiebe N. Circuit-centric quantum classifiers. *Physical Review A*. 2020; 101: 032308.
[7] Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, et al. Hierarchical quantum classifiers. *npj Quantum Information*. 2018; 4: 65.
[8] Cong I, Choi S, Lukin MD. Quantum convolutional neural networks. *Nature Physics*. 2019; 15: 1273-1278.
[9] Shalev-Shwartz S, Ben-David S. Understanding machine learning: From theory to algorithms. Cambridge University Press; 2014.
[10] Wu Y. Improvement of convolutional neural network and its application in image classification. Master's Thesis. Nanning Normal University; 2019.
[11] Liu DF, Liu JX. Neural network model for deep learning overfitting problem. *Journal of Natural Science of Xiangtan University.* 2018; 40: 96-99.
[12] Wu Y, Zhang LM. Research on the generalization ability and structure optimization algorithm of neural network. *Computer Application Research*. 2002; 19: 21-25.

[13] Jiang JZ, Zhang X, Li C, Zhao YQ, Li RG. Generalization study of quantum neural network. 2020. Available from: https://arxiv.org/abs/2006.02388.
[14] Huggins W, Patil P, Mitchell B, Whaley KB, Stoudenmire EM. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*. 2019; 4: 2.
[15] Liu D, Ran S-J, Wittek P, Peng C, García RB, Su G, et al. Machine learning by unitary tensor network of hierarchical tree structure. *New Journal of Physics*. 2019; 21: 073059.
[16] Stoudenmire EM, Schwab DJ. Supervised learning with quantum-inspired tensor networks. *Advances in Neural Information Processing Systems*. 2016; 29: 4799.
[17] McClean JR, Boixo S, Smelyanskiy VN, Babbush R, Neven H. Barren plateaus in quantum neural network training landscapes. *Nature Communications*. 2018; 9: 4812.
[18] Blake CL, Merz CJ. UCI repository of machine learning databases. 1998. Available from: http://archive.ics.uci.edu/ml/datasets/Hayes-Roth.