An unsupervised feature learning for quantum-classical convolutional network with applications to fault detection

Tong Dou\textsuperscript{1}, Zhenwei Zhou\textsuperscript{2}, Kaiwei Wang\textsuperscript{2}, Shilu Yan\textsuperscript{1}, Wei Cui\textsuperscript{1}

1. School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China
2. The Fifth Institute of Electronics, Ministry of Industry and Information Technology, Guangzhou 511370, China

E-mail: zhouzw_ceprei@163.com, wangkaiwei82@163.com, aucuiwei@scut.edu.cn

Abstract: Combining the advantages of quantum computing and neural networks, quantum neural networks (QNNs) have gained considerable attention recently. However, because of the lack of quantum resource, it is costly to train QNNs. In this work, we presented a simple unsupervised method for quantum-classical convolutional networks to learn a hierarchy of quantum feature extractors. Each level of the resulting feature extractors consist of multiple quanvolution filters, followed by a pooling layer. The main contribution of the proposed approach is to use the $K$-means clustering to maximize the difference of quantum properties in quantum circuit ansatz. One experiment on the bearing fault detection task shows the effectiveness of the proposed method.

Key Words: Quantum Computing, Feature learning, K-means, Fault Detection

1 Introduction

Quantum computers are devices that harness the laws of quantum physics, such as superposition and entanglement, to perform computation. Benefiting from super parallel computing power in principle, quantum computers are expected to solve certain problems that classical computers either cannot solve, or not solve in any reasonable amount of time. The growth of computing power and the rapidly increasing volume of data make a great progress of machine learning (ML) techniques that build complex models for finding patterns in data. As the data processing ability of classical computers is approaching the limit, quantum computing is believed to promote the development of machine learning because of its powerful parallel computing power. The intersection between machine learning and quantum computing, called quantum machine learning (QML)\textsuperscript{[1]}, has attracted more and more attention in recent years. The goal of quantum machine learning is to take advantages of quantum properties to achieve better performance than the classical machine learning techniques in terms of computational complexity or pattern recognition. This has led to a number of quantum machine learning algorithms\textsuperscript{[2–4]}, such as qSVM, qPCA, quantum Boltzmann machine. Some of these algorithms are shown to be more efficient than their classical counterparts. However, it is hard to implement them in noisy intermediate scale quantum (NISQ)\textsuperscript{[5]} devices which may include a few tens to hundreds of qubits without error correction capability. Recently, several NISQ algorithms which are based on parameterized quantum circuits (PQCs), such as variational quantum eigensolvers (VQE)\textsuperscript{[6–8]} for ground states, quantum approximate optimization algorithm (QAOA)\textsuperscript{[9]} for combinatorial problems and quantum kernel methods\textsuperscript{[10, 11]} for classification, have been developed.

In a way, PQCs offer a promising path for NISQ era. Based on PQCs, quantum neural networks(QNNs)\textsuperscript{[12, 13]} have been proposed. As an important kind of model of classical neural networks, convolutional neural networks (CNNs), which are designed for processing data that has a known grid-like topology, are good at computer vision tasks, such as image recognition, image segmentation and objection detection. Utilizing the thoughts of CNNs, quantum convolutional neural networks (QCNNs) models\textsuperscript{[14, 15]} are proposed. In \textsuperscript{[15]}, authors replaced the convolution operation with the unitary transformation of a random quantum circuit by introducing a new quanvolutional layer which consists of quanvolutional filters. Similar to the conventional convolutional layer, a quanvolutional layer can be considered as a feature extractor. And the pooling layers and fully connected layers remain classically. Quanvolutional layers can easily integrate into classical CNNs to form a quantum-classical hybrid model, which can be implemented in near-term NISQ devices. However, using random quantum circuits with parameters unchange, the properties of circuits, such as expressibility and entangling capability\textsuperscript{[16]}, will become more and more similar as the number of quanvolutional filters grows. On the other hand, quantum resources are scarce and expensive nowadays. A major drawback of many feature learning based QML algorithms is their complexity and expense because they need to be run many times to estimate the gradients. Thus we need a method to initialize the structure and the parameters of quanvolutional filters.

Based on the hybrid model introduced in \textsuperscript{[15]}, we proposed an unsupervised feature learning method to adress the problem mentioned above in this paper. We use $K$-means algorithm to cluster the quantum circuits in different structures and parameters. And the quantum circuits which are closest to the cluster centers are initialized as quanvolutional filters. Once the quanvolutional filters are determined, we can then extract the features for the entire training set just once, essentially constructing a new training set for the last layer, which means that...
it is possible to use this unsupervised method to train quantum-classical models without ever using quantum resources during the training process.

This paper is organized as follows. Section II is the preliminary, in which we first provide a brief background of the framework of PQCs and the K-means clustering method. Then, the proposed unsupervised feature learning method for quanvolutional layers is described in detail in Section III. In Section IV, to verify the efficiency, case study on bearing fault detection is presented through numerical simulation. Conclusion are given in Section V.

2 Preliminaries

In this section, we will briefly introduce the concepts of parameterized quantum circuits and K-means clustering method.

2.1 The Framework of Parametrized Quantum Circuits

Parametrized quantum circuits (PQCs) are a kind of quantum circuits that have trainable parameters subject to iterative optimizations. In general, a PQC can be described as

$$U(\theta) = \prod_{j=1}^{M} U_j(\theta_j),$$

where $\theta = (\theta_1, \ldots, \theta_M)$ are tunable parameters, while $U_j = e^{-i\theta_j V_j}$ is a rotation gate of angle $\theta_j$ generated by a Hermitian operator $V_j$ such that $V_j^2 = I$. In this paper, $V_j$ is one of Pauli matrices.

Algorithms involving PQCs usually work in a hybrid quantum-classical scheme, as shown in Fig. 1. In general, a hybrid quantum-classical scheme combines quantum state preparation, variational evolution and measurement with classical optimization.

3) Then measure the circuit repeatedly on a specific observable $O$ to estimate an expectation value $\langle O(\theta) \rangle$. Based on the $\langle O(\theta) \rangle$ which is fed into a classical optimizer, we compute a cost function $L(\langle O(\theta) \rangle)$ to be minimized by updating $\theta$.

These steps need to be repeated until when an approximate solution is reached.

2.2 K-means

The K-means method is a prototype-based objective function clustering method that selects the sum of the Euclidean distances of all objects to the prototype as the objective function of the optimization. The problem is described mathematically as: Given a dataset $D = \{x_1, x_2, \ldots, x_m\}$ and a number $K$, find a partition of $K$ clusters to the dataset $D$ by optimizing the partitioning criterion:

$$\min E = \sum_{k=1}^{K} \sum_{x \in C_k} d^2(x, \mu_k),$$

where $\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$ denotes the mean vector of $C_k$, and $d^2(x, \mu_k) = ||x - \mu_k||_2^2$. To cluster all objects into $K$ classes, first select $K$ initial particles randomly, assign each object to the particle with the smallest Euclidean distance to form $K$ clusters, and calculate the mean of each cluster as the new $K$ particles. Iterate continuously until the shutdown condition is met. In this way, one can easily classify all the objects into $K$ classes. Concretely, the K-means algorithm can be described as follow.

**Algorithm 1 K-means Clustering Algorithm**

**Input:** dataset $D = \{x_1, x_2, \ldots, x_m\}$; the number of clusters $K$.

**Output:** the clusters $C = \{C_1, C_2, \ldots, C_K\}$.

1: randomly select $K$ samples from $D$ as initial mean vectors $\{\mu_1, \mu_2, \ldots, \mu_K\}$;

2: repeat

3: let $C_i = \emptyset \; (1 \leq i \leq K)$;

4: for $j = 1, 2, \ldots, m$ do

5: compute the distance between $x_j$ and each of mean vectors $\mu_i \; (1 \leq i \leq K)$: $d_{ji} = ||x_j - \mu_i||_2$;

6: determine the cluster of $x_j$ according to the distance of mean vectors: $\lambda_j = \arg \min_{i \in \{1, 2, \ldots, K\}} d_{ji}$;

7: update the cluster $C_{\lambda_j}$: $C_{\lambda_j} = C_{\lambda_j} \cup \{x_j\}$;

8: end for

9: for $i = 1, 2, \ldots, K$ do

10: compute the new mean vectors: $\mu_i' = \frac{1}{|C_i|} \sum_{x \in C_i} x$;

11: if $\mu_i' \neq \mu_i$, then

12: update the mean vector $\mu_i$ to $\mu_i'$;

13: else

14: keep the current mean vector unchanged;

15: end if

16: end for

17: until the mean vectors do not update
3 Algorithms

In this section, we recap the hybrid quantum-classical model, quanvolutional neural network, introduced in [15]. Based on the hybrid model, we describe our unsupervised feature learning method in detail.

3.1 Notation

For notational convenience, we will make some simplify assumptions. First, here we assume that the inputs to the algorithms are \( N \times N \) data, even though there is no requirement that the inputs be square, equally sized, or even two-dimensional. And we use \( \otimes \) to denote tensor product, \( \dagger \) to denote transpose, and \( * \) to denote convolutional operation. Note that a convolutional operation of an \( N \times N \) array with an \( f \times f \) array in stride \( s \) results in an \( \left( \left\lfloor \frac{N-f}{s} \right\rfloor + 1 \right) \times \left( \left\lfloor \frac{N-f}{s} \right\rfloor + 1 \right) \) so as quantum convolutional operation.

3.2 Hybrid Quantum-Classical CNN Model

First, we briefly introduce the quanvolutional network. Intuitively, the quanvolutional network is an extension of classical CNNs with an additional quanvolutional layer, as shown in Fig. 2. Convolutional layers, pooling layers and fully connected layers are also ingredients of this hybrid model. Sparse interactions, weight sharing and equivariant representations, which are three important ideas that can help improve the performance, are kept in quanvolutional layers. A quanvolutional layer consists of a specific number of quanvolutional filters, which transforms input data using a random quantum circuit.

Formally, quanvolutional layer can expressed as:

\[
 f_x = d(q(e(u_x))),
\]

where \( u_x \) which is taken from spatially-local subsections of input, is a patch of size \( n \times n \), \( e(\cdot) \) denotes the input encoding; \( q(\cdot) \) denotes the unitary transformation applied by the random quantum circuit; \( d(\cdot) \) denotes decoding, including measurement and post-processing. \( e(\cdot) \), \( q(\cdot) \) and \( d(\cdot) \) are corresponded to quantum state preparation, variational evolution and measurement, respectively.

In this work, \( e(u_x) \) is expressed as:

\[
 e(u_x) = |u_x\rangle = \bigotimes_{i=1}^{n} R_y(x_i),
\]

where \( x_i \) is the element of \( u_x \), and \( R_y \) is the rotation operator about the \( y \) axes, defined by the equation:

\[
 R_y(x) = \begin{pmatrix}
 \cos \frac{x}{2} & -\sin \frac{x}{2} \\
 \sin \frac{x}{2} & \cos \frac{x}{2}
\end{pmatrix}.
\]

\( q(\{u_x\}) \) is expressed as:

\[
 q(e(u_x)) = U|u_x\rangle
\]

where \( |u_x\rangle \) is the output from \( e(u_x) \) and \( U \) are random selected PQCs with parameters fixed, which means we disables the learning mechanisms described in Section 2.1.

\[
d(U|u_x\rangle) \text{ is expressed as:}
\]

\[
d(U|u_x\rangle) = g(U^\dagger Z^{\otimes n \times n} U|u_x\rangle),
\]

where \( g(\cdot) \) is a nonlinear activation function which defined by equation:

\[
g(z) = \pi \times \frac{e^z - e^{-z}}{e^z + e^{-z}},
\]

and \( Z^{\otimes n \times n} \) is an observable which defined by:

\[
 Z = \begin{pmatrix}
 1 & 0 \\
 0 & -1
\end{pmatrix}.
\]

3.3 Unsupervised Quantum Feature Learning

In the early stage of classical CNNs without GPU, it was expensive for CNNs to learn the features because of the lack of the computing power. Similarly, quantum resources are scarce at present, and therefore we need a simple method to build the quantum feature extractors, that is, quanvolutional layers.

Instead of training in a purely supervised fashion, the use of unsupervised learning methods, such as Principal Component Analysis(PCA) and K-means, for obtaining convolution kernels in classical CNNs has a long history[17–19].Here, we describe a common unsupervised learning framework used for obtaining the quantum feature extractors.

Concretely, the following steps are performed to learn the feature representation:

1) Select a bunch of circuit ansatze\(^1\) with different layers and initialize the variational parameters randomly;
2) Measure the output state of ansatze repeatedly on computational basis and the probability distribution for each circuit ansatz;
3) Convert each probability distribution to a vector in \( \mathbb{R}^{2^n} \), then we construct a dataset \( X \). Given this dataset, we apply K-means clustering algorithm and choose the ansatze which are the nearest to cluster centers to be the quantum feature extractors, where \( K \) is the number we need for the quantum feature extractors.

In this way, we can maximize the difference of the quantum feature extractors. If the dimension is large, PCA can be applied for dimensionality reduction before the K-means. For multilayer architectures, each level can be trained separately.

Once trained, the quantum feature extractors produced by the above algorithm can be applied to large inputs. We can consider this quantum feature learning as a quantum preprocessing to the input data. The processed data were used to train a fully-connected neural network. This means that it is possible to train the hybrid quantum-classical model without ever using PQCs during the training process.

4 Experiments

Numerical simulations of the experiments were performed with Pennylane and Tensorflow packages.

\(^1\)In this work, we chose the same circuits that were used in [16] as ansatze.
Fig. 2: (A). Simple example of a quanvolutional layer in a full network stack; (B). An in-depth look at the processing of classical data into and out of the quanvolutional filter. (The picture is from [15] Fig.1)

Fig. 3: The architecture of the hybrid quantum-classical model considered

4.1 Dataset
We constructed a hybrid quantum-classical model and trained it on a bearing fault detection dataset for binary classification. The dataset has 299 samples. Each input is a $192 \times 1$ time series obtained from motor bearings. We randomly selected 200 samples to build a training set with the rest as a test set. The labels use one-hot encoding.

4.2 Model
We used the proposed unsupervised method to learn two-level hierarchies of local features from a dataset of bearing fault detection. In order to test the representational power of the learned features, after normalizing, we used them as the input to the classifier: a three-layer fully-connected neural network. The quantum feature extractors are composed of stacked modules, each with a quanvolutional layer followed by a max-pooling layer. The architecture of the hybrid model is shown in Fig. 3.

4.3 Results
The configurations of the model are as follow. We used stochastic gradient descent optimization with a batch size of 32. The number of the training epoch is 25. The learning rate is 0.001. The loss curve and the accuracy during training are shown in Fig. 4 and Fig. 5.

It can be seen that the proposed unsupervised feature learning method can achieve competitive results on the bearing fault detection dataset, combining the hybrid model introduced in [15].

5 Conclusion
In this work we have presented an unsupervised method for learning quantum feature extractors, and showed that our
model performs well in a bearing fault detection dataset, which is a binary classification task.

Intuitively, it seems that it is not easy to stack as many layers as needed to get useful higher-level representations because the input data are not used in this algorithm. How to combine the inputs when learning feature hierarchies. This question deserves further investigation.

References

[1] Jacob D. Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. Nature, 549(7671):195–202, 2017.

[2] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. Nature Physics, 10(9):631–633, 2014.

[3] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. Phys. Rev. Lett., 113:130503, Sep 2014.

[4] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. Phys. Rev. X, 8:021050, May 2018.

[5] John Preskill. Quantum Computing in the NISQ era and beyond. Quantum, 2:79, August 2018.

[6] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O’brien. A variational eigenvalue solver on a photonic quantum processor. Nature communications, 5(1):1–7, 2014.

[7] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. Phys. Rev. A, 92:042303, Oct 2015.

[8] Jarrod Ryan McClean, Jonathan Romero, Ryan Joseph Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. New Journal of Physics, 18(2):23023, 2016.

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

[10] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747):209–212, 2019.

[11] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. Physical Review Letters, 122(4):40504, 2019.

[12] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002, 2020.

[13] Andrea Mari, Thomas R. Bromley, Josh A. Izac, Maria Schuld, and Nathan Killoran. Transfer learning in hybrid classical-quantum neural networks. Quantum, 4:340, 2020.

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

[15] Maxwell Henderson, Samriddhi Shaka, Shashindra Pradhan, and Tristan Cook. Quantum neural networks: powering image recognition with quantum circuits. Quantum Machine Intelligence, 2(1):1–9, 2020.

[16] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies, 2(12):1900070, 2019.

[17] Adam Coutes, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, volume 15, pages 215–223, 2011.

[18] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object detection. In Proceedings of IEEE international conference on computer vision, pages 786–793. IEEE, 1995.

[19] Kevin Murphy, Antonio Torralba, Daniel Eaton, and William Freeman. Object detection and localization using local and global features. In Toward Category-Level Object Recognition, pages 382–400. Springer, 2006.