Dynamic quanvolutional neural network

Vu Tuan Hai

June 2022

Abstract

This proposal explores the merging of computer vision and quantum computing. Combining these two disciplines yields a model known as the quanvolutional neural network. This new model has the potential to improve the accuracy of image processing task and still have many aspects that need to explore.

Introduction

Computer vision [6, 25], more specifically the convolutional neural network (CNN) [19, 24] is known as an efficient technique that constructs models from sequences of images. But nowadays, it has limited due to the curse of dimensionality. Meanwhile, quantum computing provides potentially exponential speedups due to its ability to perform massively parallel computations on the superposition of quantum states. Furthermore, this research proves that using quantum computing will help neural network models achieve higher accuracy [2]. To applying quantum computing in convolutional neural network, several model have proposed such as: quantum neural network [18], quantum convolutional neural network (QCNN) [7, 26, 17] or quanvolutional neural network (QNN) [14, 20]

Notice that there are no means of quantum computing to replace classical computing, they combine each other [3]. Instead, quantum computing will be a powerful component in a computer vision application and classical computing still conduct almost previous tasks [15]. Because the current quantum hardware called NISQ devices has a small number of qubits and high error rates [21], and limited simulation resources, the QNN model will be focused on research in this proposal.

Background

Convolutional neural network

CNN has become the standard for many computer vision applications. CNN can be visualized as a series of transformations that convert the image to some

data types. These transformations are applied in an attempt to extract useful features in the dataset which can be leveraged for classification purposes. The convolutional layers in a CNN stack are each composed of n convolutional filters. Each filter in a convolutional layer iteratively convolves subsections of the image to produce feature maps, and the output of the convolutional layer will be a tensor of n feature maps which contain information about different local patterns in the data.

Quantum convolutional neural network

A QCNN is the model that is motivated by CNN architecture. It uses $\mathcal{O}(\log(N))$ variational parameters for input sizes of N, allowing for its efficient training and implementation on NISQ but only use for small data sets or experiments.

Quanvolutional neural network

To bring a quantum application to real life, a hybrid quantum-classical model called QNN is proposed in Ref. [14], Not like QCNN, this model only replaces the convolutional layer with a new type of transform layer: the quantum convolutional (or quanvolutional) layer. Quanvolutional layers are made up of a group of N quantum filters, which operate much like their classical convolutional layer counterparts by producing features maps through locally transforming input data. The key difference is that quanvolutional filters extract features from input data by transforming spatially-local subsections of data using random quantum circuits. They prove that features produced by random quanvolutional circuits could increase the accuracy of machine learning models for classification purposes

The proposed quanvolutional layer includes three components: encoder $\mathcal{E}$, quantum filter $\mathcal{Q}$ and decoder $\mathcal{D}$.

Encoder

The encoder $\mathcal{E}$ turn an image (classical data) to a quantum data by flatting x_i as $d^2_{x_i}$ – dimensional vector. This is also called quantum embedding which conducts via a quantum feature map. The feature map here is simply a set of gate parameters in a quantum circuit. The research in [20] has proposed two encoders.

The first is "threshold encoding" which uses one qubit to encode one pixel, and the i^{th} qubit is set to the $|1\rangle$ if the value of $I_i > t$ with the threshold t and vice versa. Hence, the number of required qubits is linear to the number of pixels. Although this way can be implemented easily, it turns any image into a binary image and makes us lose important information.

The second is called Flexible representation of quantum images (FRQI). FRQI transforms a quantum register from $|0\rangle$ state into the state $|I\rangle$ encoding the intensity of a monochrome image via a sequence of CRY gates by angles $\theta_i \in [0, \frac{\pi}{2}]$. Meanwhile, the first encoder requires $\mathcal{O}(d^2_{x_i})$ qubits, the second only

needs $\mathcal{O}(\log(d_{x_i}^2))$ because of the advantages of superposition, but this ways also used more gates.

Some new research in quantum embedding help us in there are three popular feature map:

- Amplitude encoding [23, 27] and quantum random access memory (QRAM) [11, 10] with $\mathcal{O}(\log(d_{x_i}))$ qubits and $\mathcal{O}(2^{d_{x_i}})$ circuit depth.

- Divide and conquer strategy [4] with $\mathcal{O}(d_{x_i})$ qubits with $\mathcal{O}(\log^2(d_{x_i}))$ circuit depth.

- With τ is the number of inputs in dataset D that loaded a time, While Amplitude encoding and Divide and conquer strategy operate on $\tau = 1$, QRAM has a big advantage in that it can load the entire dataset, or $\tau \leq |D|$ inputs at the same time so we can apply the mini-batch technique in the future. Besides, currently no commercial hardware implementations for QRAM, and it's hard to simulate.

Quanvolutional filter

Quanvolutional filter ($\mathcal{Q}$) is denoted as $U(\boldsymbol{\theta})$ which is a randomly constructed quantum circuit includes 2 - qubits gate: $CNOT$, $SWAP$, ... and a random number of 1 - qubit gate: R_X, R_Y, R_Z, H, ... These gates also have random parameters (if it have). The order of selected gates in the circuit is shuffled and the final ordering of gate operations becomes one quanvolutional filter. This way is not based on any theorem so the high probability is not the optimized structure. Some proposed structures will be present in the Methodology section. The quantum circuit is applied to the initialized state $|\psi(x_i)\rangle$, now is $U(\boldsymbol{\theta})|\psi(x_i)\rangle$.

Decoder

There are many way to decode information from the state $U(\boldsymbol{\theta})|\psi(x_i)\rangle$. In classic, the scalar value after applying convolution can be achieved via the sum operation. Meanwhile, we can take it via a finite number of measurements. An example value $p_{(00...0)}$, which is the probability of all qubits in-state $|0\rangle$.

The total transformation from $\mathcal{E}$, $\mathcal{Q}$ and $\mathcal{D}$ are known as the "quanvolutional transformation". It is clear that the number of computations required is $\mathcal{O}(d_{x_i}^2)$, which comes from mainly the complexity of the quantum circuit $U(\boldsymbol{\theta})$, encoding and decoding can be performed efficiently on classical devices.

Research questions

There are many problems that need to do to exploit the QNN model, some questions that are raised in the first paper about QNN [14] include:

(1) "Are there particular structured quanvolutional filters that seem to always provide an advantage over others?"

(2) "How data-dependent is the ideal "set" of quanvolutional filters?"

(3) "How much do encoding and decoding approaches influence overall performance?"

(4) "What are the minimal quanvolutional filter gate depths that lead to some kind of advantage?"

The question (1), (3), and (4) will be expected to resolve in this proposal through stages 3, 2, and 1 respectively. Question (2) is answered by surveying various datasets, it will be considered in the future because of the need for large computation resources. Meanwhile, research in Ref. [20] answered a part of the question (1) by showing that the trainable circuits have lower error values on average.

Proposed methodology

Because of the unknown where and when should we put the quanvolutional instead of the convolutional layer so the first stage is determining: the number of filters, number of quanvolutional layers, position of the quanvolutional layer, the patch size, ... The second stage helps us know some suitable hyperparameters in $\mathcal{E}$ and $\mathcal{D}$. The final stage is defining the structure of the quanvolutional layer, not randomly. Due to the unpopularity of quantum hardware and the limited simulation, two tiny datasets includes MNIST and MNIST Fashion will be used in these experiments.

Stage 1

Currently, the QNN simply replaced the first convolutional layer based on the hypothesis that the quantum layer extract features more efficiently than the classical layer. Thus, I want to know exactly where and when we should put the quanvolutional layers. With n_{conv} is the number of convolutional layers and n_{quanv} is the number of quanvolutional layers, we define the location bit-string:

$$s = i_0 i_1 ... i_{n_{\text{conv}}+n_{\text{quanv}}} \tag{1}$$

with $i_j = \{0, 1\}$, 1 value means quantum layer and 0 value means classical layer.

The cost function $C(s, \boldsymbol{\theta}^*, D)$ is proposed as the accuracy of certain dataset where $\boldsymbol{\theta}^*$ is the optimized parameter and D is the used dataset. The object of stage 1 is finding:

$$s^* = \arg\min_s (C(s, \boldsymbol{\theta}^*, D)) \tag{2}$$

This leads to the combinatorial optimization problems involving finding an optimal object out of a $\mathcal{O}(2^{n_{\text{conv}}+n_{\text{quanv}}})$ set of objects. The easiest way is trying all possible cases and getting troubles with a more deep neural network, or a high value of $n_{\text{conv}} + n_{\text{quanv}}$.

Another efficient way is Quantum Approximate Optimization Algorithm (QAOA) [9] was successful in many combinatorial optimization problems such

as MaxCut which reduce the complexity from exponential to linear. It is a variational algorithm that uses a unitary $U(\beta, \gamma)$ to prepare the quantum state $|\psi(\beta, \gamma)\rangle$. The goal of the algorithm is to find optimal parameters (β^*, γ^*) such that the quantum state $|\psi(\beta^*, \gamma^*)\rangle$ encodes the solution to the problem.

Stage 2

After determining the optimized bit-string s. In all quanvolutional layers, I propose an enhanced method in encoder and decoder. Currently encoding and decoding methods require lots of measurement and can quickly become problematic; referencing again and "quantum speedup" will disappears. Ideally, these methods are required which to produce beneficial machine learning features and require minimal resources.

One main difference between the the encoded algorithms from Sec. Encoder is that FRQI encodes the data into the amplitudes while NEQR and the Threshold encoding encode the data into the basis states. I hypothesize that amplitude encoding is more suitable for variational circuits than basis state encoding. But since I only tested one amplitude encoding algorithm it needs further investigation and more experiments with other encoding algorithms (such as the divide-and-conquer strategy) to support this hypothesis.

Other experiments will be conducted to determine the minimal number of measurements, the learning rate, ...

Stage 3

In the quanvolutional filter $\mathcal{Q}$, I want to make the training $\mathcal{Q}$ better. Instead of calculating by zero-order method, I can update the parameter on $\mathcal{Q}$ by gradient-based and the gradient can be calculated via general parameter shift rule [28]:

$$\frac{\partial C}{\partial \theta_j} = \sum_{\mu=1}^{R} d_\mu (C(\theta_j + s_\mu e_j) - C(\theta_j - s_\mu e_j)) \tag{3}$$

Here θ_j, e_j is the j^{th} parameter and unit vector, s_μ are the shift values and d_μ are the values associated with s_μ. Classical layers are still be updated by back-propagation.

Instead of using random ansatz, I have some theorem: First, the position's relationship between the pixel was ignored after encoding meanwhile this information are very important. So I treat each qubit as a node in a graph and create a connection between each qubit, this problem can be solved by graph ansatz, where the qubits are fully connected and ordered. Besides, I will try some famous ansatz that is proven successful in many variational quantum eigensolver (VQE) problems. Some candidates are layered gate [22], alternating operator [9], tensor network [16], ...

Additional experiments

The bellows are additional experiments that need to conduct when proposing any quantum neural network model.

(1) When faced with training the quantum neural network, we must check if the barren plateaus phenomenal is exist, that means can this model be applied on a bigger scale or not.

(2) Currently, designing an ansatz is still manual based on experience because the ansatz space is very large and increase exponentially with the number of qubit or number of layer. I expected that I can reduce the dimensions of this space by applying the constraints from $\mathcal{Q}$. Some traditional methods that can be used to solve this problem include: local search [1], spatial search [8], variable neighborhood search [13], simulated annealing [5], evolution algorithm, ... Some gradient-based methods such as generative adversarial networks (GAN) [12] have a lot of potential but it's also required we have a dataset about good ansatzes.

Research Plan

The expected results of two years of research are two research papers with the progress is described below:

- Fall 2022: Proposed dissertation. Survey and conduct experiments at Stage (1).
- Winter 2022: Conduct Stage (2) and write the first report paper.
- Spring 2023: Develop a trainable quanvolutional filter and try various famous ansatz as described in Stage (3).
- Summer 2023: Conduct additional experiments and write the second report paper.

Conclusion

In this proposal, I contribute to an emergency of current quantum computing and machine learning by improving QNN - the new hybrid quantum-classical computer vision model.

References

[1] Emile Aarts, Emile HL Aarts, and Jan Karel Lenstra. *Local search in combinatorial optimization*. Princeton University Press, 2003.

[2] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.

[3] Leonardo Alchieri, Davide Badalotti, Pietro Bonardi, and Simone Bianco. An introduction to quantum machine learning: from quantum logic to quantum deep learning. *Quantum Machine Intelligence*, 3(2):1–30, 2021.

[4] Israel F Araujo, Daniel K Park, Francesco Petruccione, and Adenilton J da Silva. A divide-and-conquer algorithm for quantum state preparation. *Scientific Reports*, 11(1):1–12, 2021.

[5] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[6] Junyi Chai, Hao Zeng, Anming Li, and Eric WT Ngai. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

[7] Samuel Yen-Chi Chen, Tzu-Chieh Wei, Chao Zhang, Haiwang Yu, and Shinjae Yoo. Quantum convolutional neural networks for high energy physics data analysis. *Physical Review Research*, 4(1):013231, 2022.

[8] Andrew M Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Physical Review A*, 70(2):022314, 2004.

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

[10] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Physical Review A*, 78(5):052310, 2008.

[11] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[13] Pierre Hansen, Nenad Mladenović, Jack Brimberg, and José A Moreno Pérez. Variable neighborhood search. In *Handbook of metaheuristics*, pages 57–97. Springer, 2019.

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

[15] Essam H Houssein, Zainab Abohashima, Mohamed Elhoseny, and Waleed M Mohamed. Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision. *Expert Systems with Applications*, page 116512, 2022.

[16] William Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and E Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and technology*, 4(2):024001, 2019.

[17] Tak Hur, Leeseok Kim, and Daniel K Park. Quantum convolutional neural network for classical data classification. *arXiv preprint arXiv:2108.00661*, 2021.

[18] SK Jeswal and S Chakraverty. Recent developments and applications in quantum neural network: a review. *Archives of Computational Methods in Engineering*, 26(4):793–807, 2019.

[19] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[20] Denny Mattern, Darya Martyniuk, Henri Willems, Fabian Bergmann, and Adrian Paschke. Variational quanvolutional neural networks with enhanced image encoding. *arXiv preprint arXiv:2106.07327*, 2021.

[21] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[22] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

[23] Maria Schuld and Francesco Petruccione. *Supervised learning with quantum computers*, volume 17. Springer, 2018.

[24] Andrei-Alexandru Tulbure, Adrian-Alexandru Tulbure, and Eva-Henrietta Dulf. A review on modern defect detection models using dcnns–deep convolutional neural networks. *Journal of Advanced Research*, 35:33–48, 2022.

[25] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[26] ShiJie Wei, YanHu Chen, ZengRong Zhou, and GuiLu Long. A quantum convolutional neural network on nisq devices. *AAPPS Bulletin*, 32(1):1–11, 2022.

[27] Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. Data encoding patterns for quantum computing. In *Proceedings of the 27th Conference on Pattern Languages of Programs*, pages 1–11, 2020.

[28] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, 2022.