Manuscript Title:
with Forced Linebreak

Peiyong Wang
School of Computing and Information Systems
Faculty of Engineering and Information Technology
University of Melbourne

(Dated: July 21, 2021)

An article usually includes an abstract, a concise summary of the work covered at length in the main body of the article.

Usage: Secondary publications and information retrieval purposes.

Structure: You may use the `description` environment to structure your abstract; use the optional argument of the `\item` command to give the category of each item.

I. INTRODUCTION

A. Brief Introduction of Quantum Error Correction

B. Brief Introduction to Neural Architecture Search and Quantum Architecture Search

C. Brief Summary of Previous Research Attempt on Finding Quantum Error Correction Codes with Machine Learning

This subsection will include a brief review on:

- QVECTOR [1]

- Thomas Fösel's paper on reinforcement learning for discovering quantum error correction circuits with non-variational gates [2]

- Simon Charles Benjamin's paper on encoding specific logical states from 5-bit or 7-bit QECC code words with variational circuits under depolarizing noises [3]

- SCB's paper on experimental re-compiling of the five-bit quantum error correction code on a superconducting processor with variational circuits [4]

- Decoding circuit for the toric code with reinforcement learning [5]

- Optimizing a family of surface code with reinforcement learning [6]

II. BACKGROUNDS

This section will briefly review relevant properties of the five-bit quantum error correction code, or other kinds of quantum error correction codes we would like to experiment on.

III. METHODS

In this research, we will employ probability-based quantum architecture methods, first developed in [7], to search for circuit structures that could provide high-fidelity encoded logical states under the influence of device noises, with a set of pre-defined quantum operations. We assume that the structure of the desired quantum circuit follows a probability distribution $P(\mathbf{k}, \alpha)$ characterized by α, which can be optimized through gradient-based classical optimization methods. $\mathbf{k}$ is the parameter corresponding to the structure of the variational circuit.

A. Circuit Representation

The circuit in this research is a product of different unitary operations selected from a pre-defined operation pool according to the structure parameter $\mathbf{k}$:

$$U(\mathbf{k}, \theta) = \prod_{i=p}^{1} V_{k_i}(\theta_i) = V_{k_p} V_{k_{p-1}} \cdots V_{k_i} \cdots V_{k_2} V_{k_1} \quad (1)$$

Where p is the number of operations (layers) in the circuit, V_{k_i} is an operation selected from the predefined operation pool

For example, if $\mathbf{k} = [1, 2, 4]$, then we have $k_1 = 1, k_2 = 2, k_3 = 4$ and $p = 3$, then we have $U = V_4 V_2 V_1$. The inverse order corresponds to the order of operations in the diagram of a quantum circuit.

B. Circuit Performance Evaluation

To evaluate the performance of the searched quantum circuit, we calculate the fidelity between the states encoded by the searched circuit and the states encoded by the ideal encoding circuit for quantum error correction codes like the five five-qubit error correction code. The single-qubit states to be encoded (which serves as the "input" data to the model) are six eigenstates of the Pauli operators and the T state: $|\varphi\rangle \in X =$

$\{|0\rangle, |1\rangle, |+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}, |-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}, |+i\rangle = \frac{|0\rangle+i|1\rangle}{\sqrt{2}}, |-i\rangle = \frac{|0\rangle-i|1\rangle}{\sqrt{2}}, |T\rangle = \frac{|0\rangle+e^{\pi/4}|1\rangle}{\sqrt{2}}\}$.

The target state $\rho \in Y = \{E(|\phi\rangle \otimes |0\rangle^{\otimes(n-1)})[E(|\phi\rangle \otimes |0\rangle^{\otimes(n-1)})]^\dagger \mid |\phi\rangle \in X\}$, where E is the ideal encoding unitary for the quantum error correction code, and n is the number of physical qubits needed to encode a logical state.

The state encoded by the searched circuit can be represented as $\sigma \in \hat{Y} = \{U(\mathbf{k},\theta)(|\phi\rangle \otimes |0\rangle^{\otimes(n-1)})[U(\mathbf{k},\theta)(|\phi\rangle \otimes |0\rangle^{\otimes(n-1)})]^\dagger \mid |\phi\rangle \in X\}$. During the experiment, the encoded state will be obtained by saving the density matrix of the state after performing operations in the searched circuit.

The cost function we are trying to optimize is based on the fidelity between the encoded states and the target states:

$$L(\mathbf{k},\theta) = 1 - \frac{1}{7} \sum_{\rho_i \in Y, \sigma_i \in \hat{Y}} F(\rho_i, \sigma_i) \tag{2}$$

Where

$$F(\rho_1, \rho_2) = \operatorname{Tr}\left[\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right]^2 \tag{3}$$

is the fidelity between two quantum states ρ_1 and ρ_2.

C. Objective of Quantum Architecture Search

With equation 2, we can construct the objective of our quantum architecture search task, as depicted in [7]:

$$\underset{\alpha,\theta}{\arg\min}\, \mathcal{L}(\alpha,\theta) \tag{4}$$

where

$$\mathcal{L}(\alpha,\theta) = \sum_{\mathbf{k}\sim P(\mathbf{k},\alpha)} L(\mathbf{k},\theta) \tag{5}$$

In our experiment, the total number of operations (layers) in the circuit is denoted as p, the number of different operations in the predefined operation pool is denoted as c, the maximum number of parameters in the operations is denoted as l. Then the shape of θ is $p \times c \times l$ and α is $p \times c$.

Since estimated gradients from sampling often suffer from high variance, in our experiment, a baseline is added to the calculation of losses:

$$L = L - \bar{L}_{prev} \tag{6}$$

Where $\bar{L}_{prev}$ is the average loss (without baseline) of the previous batch.

D. Gradient Evaluation

To optimize the parameters of the circuit and probabilistic model with gradient-based optimizers, first we need to evaluate the gradients with respect to θ and α. The gradient of θ is straightforward to calculate with parameter-shift methods:

$$\nabla_\theta \mathcal{L} = \sum_{\mathbf{k}\sim P(\mathbf{k},\alpha)} \nabla_\theta L(\mathbf{k},\theta) \tag{7}$$

As depicted in [7], the evaluation of gradients of α is based on sampling a batch of circuits from the probabilistic distribution:

$$\begin{aligned}\nabla_\alpha \mathcal{L} =& \sum_{\mathbf{k}\sim P} \nabla_\alpha \ln P(\mathbf{k},\alpha) L(\mathbf{k},\theta) \\ &- \sum_{\mathbf{k}\sim P} L(\mathbf{k},\theta) \sum_{\mathbf{k}\sim P} \nabla_\alpha \ln P(\mathbf{k},\alpha)\end{aligned} \tag{8}$$

For normalized probability distributions, the expectation of $\nabla_\alpha \ln P$ is zero. Therefore the second term in Eqn. 8 can be ignored.

E. Probabilistic Model

In this research, we choose the independent categorical probabilistic model to approximate the distribution of the operations in each layer of the circuit. This model assumes that the probabilities of selecting a certain operation for each layer are independent with each other. For a single layer, the probability of selecting a certain operation follows the categorical distribution:

$$P(\mathbf{k},\alpha) = \prod_i^p \pi_i(k_i,\alpha_i) \tag{9}$$

Where p is the total number of layers in the circuit, and

$$\pi(k_i = j, \alpha_i) = \frac{e^{\alpha_{ij}}}{\sum_k e^{\alpha_{ik}}} \tag{10}$$

is the categorical distribution of each layer.

The gradient of such a model can be evaluated as:

$$\nabla_{\alpha_{ij}} \ln P(k_i = m, \alpha) = -P(k_i = m, \alpha) + \delta_{jm} \tag{11}$$

Since the selection of operations in each layer is independent from each other, we have

$$P(k_i = m, \alpha) = \pi_i(m, \alpha_i) \tag{12}$$

Where π_i is the categorical probability distribution of operations of layer i. However, such property may not hold for more complex probabilistic models.

F. Differentiable Quantum Architecture Search

1. Workflow of Architecture Search

With the probabilistic model and gradient evaluation methods at hand, we can co-optimize the parameters for the circuit and probabilistic model together:

1. Randomly initialize the $p \times c \times l$ shared parameter tensor θ. Initialize a $p \times c$ all-zero matrix for α;
2. Repeat the following steps until stop criteria is met:
   (a) Sample a batch of different $\mathbf{k}$ from the probabilistic model $P(\mathbf{k}, \alpha)$ of size K;
   (b) Compute the loss of each sample in the batch according to Eqn. 2;
   (c) Compute the gradients of α and θ according to Eqn. 8 and 7;
   (d) Update θ and α with gradient-based optimizers.
3. After the previous step meets its stopping criteria, obtain the circuit structure parameter $\mathbf{k}^*$ that has the highest probability from $P(\mathbf{k}, \alpha)$, and fine-tune the circuit parameter to reach the optimal θ^*.

2. Improvements

When the number of layers and number of qubits increase, the total number of possible circuits grows exponentially, and the loss landscape for the probability model could be very rough, leading to being trapped in some local optimal. To deal with this, we'll need to increase the batch size for each iteration. Also, after obtaining the parameter for probabilistic model, we could adopt beam search to find top-k best circuit structures from the probabilistic model, and train all of them to find the best one. Although this could increase the time we need to find the best circuit, it would still be considerably faster than [3], in which the authors searched among a very large number of different ansätze (order of 10000) for their tasks.

To ensure the circuits are sampled unbiased in the first iteration, we initialize α as an all-zero matrix. However, to prevent being trapped in some local minima, we could adopt parallel training on different starting points of θ.

IV. EXPERIMENTS AND RESULTS

V. CONCLUSION

[1] P. D. Johnson, J. Romero, J. Olson, Y. Cao, and A. Aspuru-Guzik, Qvector: an algorithm for device-tailored quantum error correction (2017), arXiv:1711.02249 [quant-ph].

[2] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Reinforcement learning with neural networks for quantum feedback, Physical Review X **8**, 10.1103/physrevx.8.031084 (2018).

[3] X. Xu, S. C. Benjamin, and X. Yuan, Variational circuit compiler for quantum error correction, Physical Review Applied **15**, 10.1103/physrevapplied.15.034068 (2021).

[4] M. Gong, X. Yuan, S. Wang, Y. Wu, Y. Zhao, C. Zha, S. Li, Z. Zhang, Q. Zhao, Y. Liu, and et al., Experimental exploration of five-qubit quantum error correcting code with superconducting qubits, National Science Review 10.1093/nsr/nwab011 (2021).

[5] P. Andreasson, J. Johansson, S. Liljestrand, and M. Granath, Quantum error correction for the toric code using deep reinforcement learning, Quantum **3**, 183 (2019).

[6] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, Optimizing Quantum Error Correction Codes with Reinforcement Learning, Quantum **3**, 215 (2019).

[7] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Differentiable quantum architecture search (2020), arXiv:2010.08561 [quant-ph].