QTML 2021

Quantum Techniques in Machine Learning

Booklet of Extended Abstracts

POSTERS - Part II

QuTorch: A Quantum Torch Framing Quantum-Classical Hybrid Neural Network for Chip-Based Quantum Computing Accelerator

Junjie Hu[1,2†], Yuxin Jin[1†], Xiaojun Xu[1,2†], Lina Dong[3†], Xiangyu Li[1†], Zehui Tian[1,4†], Zhicheng Luo[1†], Shiyu Wang[1†], Yingji Sun[1], Dandan Liu[1], Zhiheng Yang[1], Shisheng Chen[1], Zheyong Zhang[2], Bo Fang[1], Fangyan Zhang[1], Shunyang Shi[1], Hao Tang[1,2*], Binju Wang[3*] and Xianmin Jin[1,2*]

[1]Algorithm&Application Department, TuringQ, Shanghai, 100190, China.

[2*]Center for Integrated Quantum Information Technologies (IQIT), School of Physics and Astronomy and State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai, 200240, China.

[3*]State Key Laboratory of Physical Chemistry of Solid Surfaces and Fujian Provincial Key Laboratory of Theoretical and Computational Chemistry, College of Chemistry and Chemical Engineering, Xiamen University, Xiamen, 360015, China.

[4]School of Naval Architecture, Ocean & Civil Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China.

*Corresponding author(s). E-mail(s): haotang2015@sjtu.edu.cn; wangbinju2018@xmu.edu.cn; xianmin.jin@sjtu.edu.cn;

†These authors contributed equally to this work.

Abstract

QuTorch is an open-source classical-quantum hybrid AI package developed by TuringQ. This Torch-based framework is designed for chip-based photonic accelerator system, which includes modules of parameterized-quantum-circuits, quantum-intermediate-representations, AI-inspired quantum neural networks, and

interdisciplinary hybrid algorithm applications. Based on quantum circuits and PyTorch, QuTorch can be extented as you imagine.

Keywords: Torch-based hybrid classical, AI-inspired quantum neural networks, Interdisciplinary applications, QuTorch

1 Extend Abstract

QuTorch is established in the aim of combining quantum computing with machine learning and developing a powerful tool for constructing hybrid quantum-classical models efficiently. Through investigation, numbers of open-source frameworks like Qutip, Qiskit, Paddle Quantum, Tensorflow Quantum invoke quantum circuits and build basic structures including gates, circuits and measurement. However, these communities may confront challenges such as difficulty in interfacing with machine learning, insufficiency of existing classical algorithm and inconveniently available hardware. To inherit advantages of quantum circuits and fix aforementioned problems as much as possible, QuTorch is designed with following characteristics: Quantum Circuits-based; Capability to quantum machine learning(QML); Conveniently Available Hardware; Simplicity and Multifunction. In actual execution, the CPU is dominant to distribute computing tasks, and QPU is the quantum accelerator of the CPU.

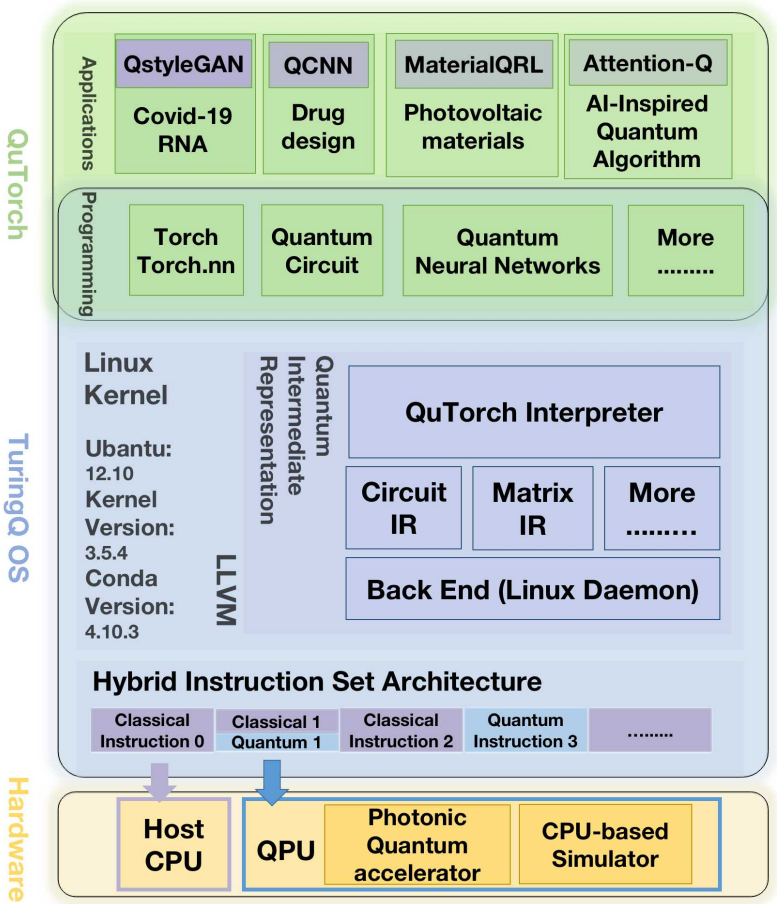
With the help of QuTorch which is friendly to classical AI developer, QStyleGAN development becomes more efficient and meanwhile it naturally achieves a classic-quantum mixing training process. Taking advantage of these, we focus on developing the quantum part of QStyleGAN. To predict variation structure of COVID-19 and map to other strains, we first encode several important COVID-19 variants to quantum state in aim of adapting quantum circuits. Then, we build a brand new QDiscriminator with equivalent network as the classical one. Noticeably, during building QDiscriminator, we create a new quantum method called quantum-inspired blur convolution, which achieves using a little parameters and extracting features. Also, we succeed to downsample density matrix by using partial trace and therefore QStyleGAN is capable of progressive training. In the process of training, it shows that QStyleGAN is more stable and faster convergent than StyleGAN with multiple loss functions.

Compared with experiments, computer-aided drug design can accelerate the process of drug development and save time and cost. Drug screening is one of the most important tasks of computer-aided drug design, which is of great significance for the discovery of new drugs and the new use of old drugs. The combination between drug and target needs to be specific and stable in order to give better play to its efficacy. This requires an accurate description of the interaction between proteins and ligands. This interaction is defined as binding affinity. Thus, accurately predicting the binding affinity between protein ligands is the key to drug screening. With the development of artificial intelligence, the application of deep learning to the binding affinity of protein ligand complexes is booming. Based on the quantum Convolutional Neural Network(QCNN) method, the parametrized-quantum-circuits(PQC)

were used to improve model performance. The various hybrid algorithm applications will be developed for protein-protein interaction, protein ligand interaction and protein nucleic-acid interaction in drug design.

The model of reinforcement learning(RL) is a Markov process, which can be described as the evolution of the state under the transition matrix. In this application, the state is designed as the different representations of various structures in the feature space of disorder crystal, and the action is designed to select the structure within the similarity for phase transition. For enabling the agent to find the optimal path, this method sets the agent to receive a reward of -1 for each step without exceeding its threshold. While reaching the end point, it will get a larger positive reward. The field of Quantum RL aims to take advantage of this improvement by designing the PQC-based agents, which is based in the Qutorch framework and designed to search the process of phase transition of photovoltaic materials.

Attention mechanisms are firstly proposed by Bahdanau, seeking to tackle the bottleneck in encoder-decoder architectures that a fixed-length vector can not contain all input information. Recently, attention mechanisms are wildly used in a variety of tasks, not only in natural language processing such as machine translation, speech recognition, text summarization, but also employed gradually in computer vision. We propose a quantum-based attention mechanism, transferring the classic self-attention mechanism to the quantum circuit to improve its parallel computing quality. Due to the great parallel computing capability, implementing classic deep learning models on quantum circuits can accelerate the computing process. In our coming research, Attention-Q will be employed and evaluated on different models in different tasks.



Variational quantum amplitude amplification eigensolver

Kyunghyun Baek,[1] Jaehee Kim,[2] Jaewan Kim,[1]
Jinhyoung Lee,[3] Joonsuk Huh,[1,4,*] and Jeongho Bang[5,†]

[1] *School of Computational Sciences, Korea Institute for Advanced Study, Seoul, 02455, Republic of Korea*
[2] *SKKU Advanced Institute of Nanotechnology (SAINT), Sungkyunkwan University, Suwon 16419, Korea*
[3] *Department of Physics, Hanyang University, Seoul, 04763, Republic of Korea*
[4] *Department of Chemistry, Sungkyunkwan University, Suwon 16419, Korea*
[5] *Electronics and Telecommunications Research Institute, Daejeon 34129, Korea*

(Dated: October 25, 2021)

We develop a quantum algorithm that finds the ground states of a Hamiltoians as an extension of variational quantum eigensolver (VQE) by combining with the quantum amplitude amplification (QAA), where the ansatz parameters are updated through QAA(-like) process—thus, we named "variational quantum amplitude amplification eigensolver (VQAAE)." VQAAE allows us to alleviate the landscape problem (often-called barren-plateau), saving the classical computational resource.

Introducetion Computing the ground state is of fundamental and applied interest in physics and quantum chemistry. However, solving the ground-state eigenproblem is usually intractable since the requirement of the computational resource exponentially scales with respect to the system size. Quantum computer is expected to solve the problem more efficiently. Herein, the variational quantum eigvensolver (VQE) has actively been studied [1–6]. On the other hand, quantum amplitude amplification (QAA) is one of the celebrated quantum kernels in quantum computation. For example, QAA can be casted to find a solution from unstructured database, which is known as Grover's algorithm [7]. QAA provides a quadratic speedup to find a target state from unstructured database, which is adapted in many applications such as cryptography [8, 9] as a subroutine. In this work, we develop an extension of VQE by combining the QAA(-like) process, where the ansatz parameters are updated via QAA without using any classical optimization. In this sense, the proposed scheme is named variational quantum amplitude amplification eigensolver (VQAAE).

Variational quantum amplitude amplification eigensolver (VQAAE) Our VQAAE consists of two main steps. In the first step, we apply QAA to the ansatz state designed by a set of controllable parameters. After a single trial of the aforementioned process, we can obtain a state which is closer to the ground state compared to the previous ansatz. We give more specifics below. For any N-qubit system, we let the ansatz state be $|\varphi(\vec{\theta})\rangle = u(\vec{\theta})|0\rangle^{\otimes N}$ where $u(\vec{\theta})$ is a controllable unitary and $\vec{\theta}$ is a vector whose components are the control parameters. Here, we additionally adopt an ancilla qubit in $|+\rangle_C = (|0\rangle_C + |1\rangle_C)/\sqrt{2}$; hence, the trial state is $|\psi(\vec{\theta})\rangle \equiv |+\rangle_C \otimes |\varphi(\vec{\theta})\rangle$. We then introduce the conditional Hamiltonian evolution such that $U = \sum_{\tau=0,1} |\tau\rangle\langle\tau|_C \otimes i^\tau \exp((-1)^\tau i\frac{\pi}{4}H)$. Here, we assume that: 1) the Hamiltonian H is normalized, and 2) its lowest eigenvalue is not degenerate. These two assumptions are simply expressed as $0 < \lambda_0 < \lambda_1 \leq ... \leq \lambda_{2^N-1} \leq 1$, where λ_i's are eigenvalues of the Hamiltonian H. This unitary acts like an oracle operation executing phase marking onto the target state in quantum amplitude amplification. Another important operation in our VQAAE is a kind of inversion operation (often-called Householder reflection), defined as $R(\vec{\theta}) = I - 2|\psi(\vec{\theta})\rangle\langle\psi(\vec{\theta})|$. By applying U and $R(\vec{\theta})$ on the trial state $|\psi(\vec{\theta})\rangle$ alternatively and measuring the ancilla qubit, one can obtain the conditional states which is closer to the ground state. Namely, the conditional states $|\varphi'_{out|\tau}(\vec{\theta})\rangle = \langle i|_C R(\vec{\theta})U|\psi(\vec{\theta})\rangle/\sqrt{p_\tau}$ with the probability

* joonsukhuh@skku.edu
† jbang@etri.re.kr

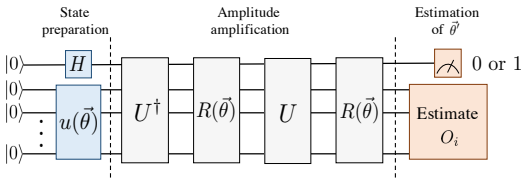


Figure 1. Quantum circuit representation of VQAAE.

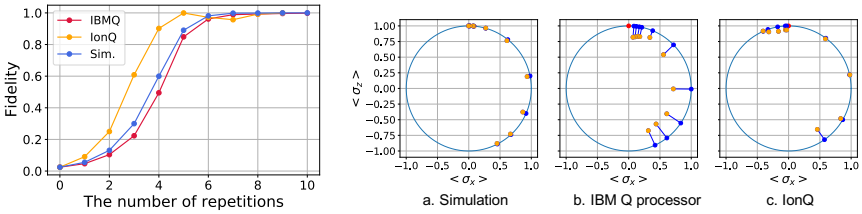


Figure 2. Left graphs is a plot of fidelity against the number repetition of VQAAE. Graphs (a)-(c) are plots of expectation values of σ_x and σ_z obtained from a simulation on IBM-QISKit based local simulator and experiments on IBM Q processors and IonQ.

$p_\tau = 1/2$ to obtain an outcome τ satisfy

$$|\langle\lambda_0|\varphi_{out|\tau}(\vec{\theta})\rangle|^2 \geq |\langle\lambda_0|\varphi(\vec{\theta})\rangle|^2, \tag{1}$$

where $|\lambda_0\rangle$ is the ground state corresponding to the lowest eigenvalue λ_0. This relation is guaranteed for any ansatz $|\varphi(\vec{\theta})\rangle$. However, if the initial ansatz has no overlap with the ground state, the amplitude amplification does not work. The schematic of this process is illustrated in Fig. 1.

The second step is to update the control parameters as $\vec{\theta} \to \vec{\theta}'$, so that we prepare a new trial state $|\psi(\vec{\theta}')\rangle$. According to the relation (1), the conditional states are located closer to the ground state in the state space of the system. This fact implies that one can extract parameter information that give rise to $|\varphi(\vec{\theta}')\rangle$ closer to the ground state. Hence, we update the parameters by estimating them from the output state $|\varphi_{out|\tau}(\vec{\theta})\rangle$. Here, the estimation of $\vec{\theta}'$ is accomplished by the measurements with the pre-determined set of observables $\{O_i\}$.

Example 1: We first perform a proof-of-principle demonstration of VQAAE on IBMQ processor, by considering a relatively simple Hamiltonian,

$$\mathcal{H}_1 = \frac{1}{4}|0\rangle\langle 0| + \frac{3}{4}|1\rangle\langle 1|. \tag{2}$$

The ansatz is set to be $|\varphi(\theta)\rangle = R_y(\theta)|0\rangle$, i.e.,

$$\rho(\theta) = \frac{1}{2}(I + \sin\theta\sigma_x + \cos\theta\sigma_z). \tag{3}$$

The set of observables related to the ansatz parameter θ is given as $\{\sigma_x, \sigma_z\}$. In such a setting, the ground state is obtained for $\theta = 0$.

We start our proof-of-principle experiments with an initial parameter $\theta = 0.9\pi$, at which the fidelity between $\rho(0.9\pi)$ and the ground state is $\simeq 0.024$. As a result, we observe that the more iterations of our VQAAE, the closer updated ansatz to the ground state. We plot the data of fidelity vs number of repetitions in Fig. 2. The experimental data from IBM Q processor are good agreement with the data yielded from a IBM-QISKit based local simulator, while the data from IonQ are not. This is owing to the white noise immunity that we described above. We illustrate this feature in Fig. 2(a), 2(b) and 2(c), where orange dots denote the expectation values obtained from the Qiskit simulators, IBM Q processors and IonQ, respectively;

ACKNOWLEDGEMENT

We acknowledge the support of Samsung Advanced Institute of Technology for the use of IBM quantum systems.

---

[1] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, Nature Communications **5**, 4213 (2014).

[2] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New Journal of Physics **18**, 023023 (2016).

[3] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis, Phys. Rev. X **6**, 031007 (2016).

[4] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature **549**, 242 (2017).

[5] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017).

[6] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos, Phys. Rev. X **8**, 031022 (2018).

[7] L. K. Grove, In *Proc. of the 28th ACM STOC* , 212 (1996).

[8] A. Chailloux, M. Naya-Plasencia, and A. Schrottenloher, in *Advances in Cryptology – ASIACRYPT 2017*, edited by T. Takagi and T. Peyrin (Springer International Publishing, Cham, 2017) pp. 211–240.

[9] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, in *Post-Quantum Cryptography*, edited by T. Takagi (Springer International Publishing, Cham, 2016) pp. 29–43.

Machine Learning Optimization of Quantum Circuit Layouts

Alexandru Paler[1,2,5], Lucian M. Sasu[2,3], Adrian-Cătălin Florea[2], Răzvan Andonie[2,4]

[1] *Aalto University, Espoo, Finland;* [2] *Transilvania University of Brașov, Romania;* [5] *University of Texas at Dallas, USA;* [3] *Xperi Corporation, Brașov, Romania;* [4] *Central Washington University, Ellensburg, USA*

The quantum circuit layout problem is to map a quantum circuit such that the constraints of the device are satisfied. We introduce a heuristic and its machine learning version. We use a Gaussian function to estimate the depth of the circuits, and to focus on the circuit region that influences most the depth. Our methods open the path to feasible large scale compilation. The following is an extended abstract of [1].

I. INTRODUCTION

A quantum circuit is adapted to the device's register connectivity during a procedure called quantum circuit layout (QCL). QCL takes as input a circuit C_{in} incompatible with a device's register connectivity and outputs a circuit C_{out} that is compatible. *Additional gates* are used to overcome the connectivity limitations and compile a device-compatible circuit C_{out}. QCL includes two steps [2]: 1) *initial placement* (e.g., [3]), where a mapping of the circuit qubits to device registers is computed; 2) *gate scheduling*, where the circuit C_{in} is traversed gate-by-gate. We define the *Ratio* function between the depths of two circuits, where $|C| > 0$ is the depth of circuit C. We formalize the QCL optimization problem as *the minimization of the function.* $Ratio(C_{in}, C_{out}) = \frac{|C_{out}|}{|C_{in}|}$.

Fast and scalable QCL allows laying out a circuit with multiple QCL parameter values and selecting the best compiled circuit. This is not possible with current state of the art QCL methods.

The core of our work is based on machine learning techniques. The novelties of our work are: a) **automatic feature selection** - focusing on the important sub-circuit using a configurable Gaussian function; b) **automatic QCL configuration** – we use weighted random search to optimize QXX parameter values; c) **demonstrating QCL learnability** - the machine learning version QXX-MLP works as an approximation method for the circuit layout depth; d) **scalability** - the almost instantaneous layout performance. We demonstrate it by showing that QCL execution time can be shortened and at the same time maintaining the *Ratio* optimality.

II. METHODS

QXX is a fast search algorithm that uses an estimation function to generate a C_{out} with minimum depth. QXX is called by the subsequent gate scheduler to compute a good qubit allocation/mapping/placement. QXX uses a Gaussian-like function called *GDepth* to estimate the resulting depth (cost) of the layed out circuit. The *GDepth* function is a sum of a Gaussian functions whose goal is to model the importance of CNOT sub-circuits from C_{in}. The qubit mapping is found using the minimum estimated value of a function called *GDepth*.

QXX is a combination of breadth-first search and beam search. The search space is a tree. Each tree node has an associated *GDepth* cost. The level in the tree equals the length of the mapping for which the cost was computed. New depth estimation values are computed using the *GDepth* function, each time leaves are added to the tree. QXX uses *three types of parameters*: 1) for configuring the search space; 2) for adapting *GDepth* to the circuit C_{in}; 3) for adapting QXX to the second step of QCL.

The parameters of the normal QXX method are optimized using weighted random search (WRS) [4]. The WRS method starts from an initial set of parameter values and adapts the values in order to minimize the obtained *Ratio*. This procedure forms a feedback loop between WRS and QXX.

We developed QXX-MLP in order to learn the QXX algorithm. We considered multiple candidate models, and found MLP to be the best model. We used it during the parameter optimization stage as an approximator for the functionality of QXX. Running WRS multiple times for a set of benchmark circuits is equivalent to *almost* executing an exhaustive search of the parameter space. The exhaustive search data is collected and used to train QXX-MLP.

III. RESULTS

To benchmark overall performance we use synthetic benchmarking circuits, QUEKO[5] which capture the properties of Toffoli+H (e.g. arithmetic, quantum chemistry, quantum finance etc.) circuits without being specific for a particular application, and have known optimal depths $|C_{out}| = |C_{in}|$, meaning that the input circuit and the layout circuit have equal depths.

We use QXX for the initial placement (first QCL step) and the Qiskit `StochasticSwap` gate scheduling for the second QCL step. The results depend on both the initial placement as well as the performance of the `StochasticSwap` scheduler. We do not configure the latter and use the same randomization seed. QXX achieves *Ratio* values around 50% lower (which means better – best *Ratio* is 1) than Qiskit on the low depth (up to 15 gates) QUEKO TFL circuits. In general, the performance of QXX is between Qiskit and tket [6].

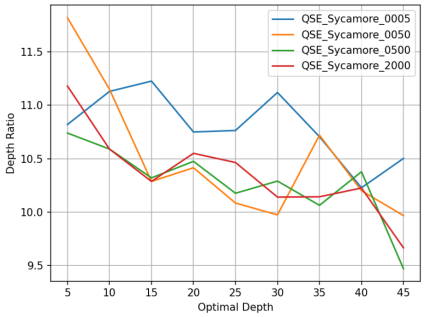
We use a combination of WRS and QXX-MLP in an attempt to minimize the time required to identify an optimal configuration. QXX-MLP and WRS are almost instantaneous. We do not report times, because these were under 2 seconds for the entire batch of 90 circuits. In contrast, the execution time of WRS using the normal QXX was about 3 hours to find the optimal parameters for the 90 circuits (cf. Table 1).

The performance of QXX-MLP is more than encouraging: it had the performance of a timedout WRS optimization. The WRS parameter optimization did not timeout, because MLP inference is a very fast constant time operation. Our results support the thesis that WRS can adapt the parameters of the QXX Gaussian curve in order to select the region of the circuit that influences the total cost of laying it out. The *GDepth* function is relevant with respect to resulting layout optimality, as well the speed of the layout method. For shallow circuits, WRS prefers Gaussian centers close to the end of the benchmark circuits, meaning that the last gates are more important than the others. For deeper circuits, WRS sets the Gaussian center to be closer to the beginning of the circuit.

The WRS and QXX-MLP optimization takes a few seconds compared to the hours (cf. Table **??**) necessary for WRS and QXX. This is a great advantage that comes on the cost of obtaining a trained MLP, which is roughly the same order of magnitude to a WRS parameter optimization. MLP training is performed only once. In the case where the MLP model is not used WRS parameter optimization has to be repeated. In a setting where quantum circuits are permanently layed out and executed (like in quantum computing clouds) incremental online learning is a feasible option.

Timeout	TFL Depth	Duration	Average *Ratio*
0.05s	45	3h 10m 38s	4.465
0.5s	45	6h 03m 05s	4.328
5s	45	7h 55m 25s	4.138
20s	45	9h 46m 10s	4.093
0.05s	25	1h 08m 22s	4.537
0.5s	25	1h 48m 17s	4.279
5s	25	2h 15m 10s	3.966
20s	25	2h 56m 23s	4.043

a)



b)

Figure 1. a) WRS time and average optimum Depth *Ratio*s for two types of QUEKO TFL circuits: with depth 45 and depth 25. The average *Ratio* is obtained for the non-MLP QXX method applied to the 90 benchmark circuits; b) Scalability – Depth Ratios are not affected by timeout durations: Laying out QUEKO QSE circuits using parameters learned from QUEKO TFL circuits. Each parameter evaluation executed by WRS was timed out after 5, 50, 500 and 2000 1/100*th* seconds (10 milliseconds).

[1] A. Paler, L. M. Sasu, A. Florea, and R. Andonie, Machine learning optimization of quantum circuit layouts, arXiv preprint arXiv:2007.14608 (2020).

[2] M. Y. Siraichi, V. F. d. Santos, C. Collange, and F. M. Q. Pereira, Qubit allocation as a combination of subgraph isomorphism and token swapping, Proceedings of the ACM on Programming Languages **3**, 1 (2019).

[3] A. Paler, On the influence of initial qubit placement during NISQ circuit compilation, in *International Workshop on Quantum Technology and Optimization Problems* (Springer, 2019) pp. 207–217.

[4] R. Andonie, Hyperparameter optimization in learning systems, J. Membr. Comput. **1**, 279 (2019).

[5] B. Tan and J. Cong, Optimality study of existing quantum computing layout synthesis tools, IEEE Transactions on Computers **early access**, 1 (2020).

[6] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, t| ket>: A retargetable compiler for nisq devices, Quantum Science and Technology (2020).

Development of a QNN based binary classifier and comparison of its performance on different Classical Processors

Rita Abani[1], Amlan Chakrabarti[2]

[1]Department of Electrical Engineering and Computer Science, Indian Institute of Science Education and Research, Bhopal , India

[2]A. K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India

Extended abstract

In this work, we investigate the feasibility of a binary Quantum Neural Network-based classifier developed using the integration of Cirq and TensorFlow Quantum and tested on a variety of hardware accelerators to see if parallel computing can provide an advantage. Numerous amounts of data have been generated, particularly in the last few years, which has witnessed a growth in the Big-data sector, and research is underway to identify strong and favourable techniques to analyse and understand data in faster, more efficient, and less chaotic ways. Moore's Law's recent developments haven't been encouraging, owing to the curve's progressive saturation, which has prompted us to look into various pathways of unorthodox computing methods that could help alleviate the current growing computing needs of industry and academia. Applied physics, biomedical engineering, chemical simulations, cryptography, and finance are just a few of the fields that require advanced data management and machine learning techniques. Several milestones have been neared, especially concerning Quantum Hardware with advances in the manufacture of qubits, scalability, error, and noise reduction increasingly convincing academia of the purported power of Quantum Computing in addressing the current shortage of computational power and the challenges of computational complexity.TensorFlow Quantum, a quantum machine learning library that enables quick prototyping of hybrid quantum classical ML models, is used in our research. It integrates quantum computation algorithms and logic designed in Cirq, a Python software library for manipulating, writing, and optimising quantum circuits as well as running them on quantum computers and quantum simulators with quantum computation primitives that are compatible with existing TensorFlowAPIs as well as high-performance quantum circuit simulators.

References

Cong, Iris & Choi, Soonwon & Lukin, Mikhail. (2018). Quantum Convolutional Neural Networks. https://arxiv.org/pdf/1802.06002.pdf

Farhi, Edward & Neven, Hartmut. (2018). Classification with Quantum Neural Networks on Near Term Processors.

Aliyev, Vagif. "A definitive explanation to the Hinge Loss for Support Vector Machines." *Towards data science*, Medium, https://towardsdatascience.com/a-definitive-explanation-to-hinge-loss-for-support-vector-machines-ab6d8d3178f1.

Beer, K., Bondarenko, D., Farrelly, T. *et al.* Training deep quantum neural networks. *Nat Commun* 11, 808 (2020). https://doi.org/10.1038/s41467-020-14454-2

Chowdhury, Kunal. "Understanding loss functions : Hinge loss." *Analytics Vidhya*, Medium, https://medium.com/analytics-vidhya/understanding-loss-functions-hinge-loss-a0ff112b40a1.

Laurel, Carlos & Dong, Shihai & Cruz-Irisson, M.. (2015). Equivalence of a Bit Pixel Image to a Quantum Pixel Image. Communications in Theoretical Physics. 64. 501-506. 10.1088/0253-6102/64/5/501. https://www.researchgate.net/publication/284750787_Equivalence_of_a_Bit_Pixel_Image_to_a_Quantum_Pixel_Image

Other websites:

- https://towardsdatascience.com/a-definitive-explanation-to-hinge-loss-for-support-vector-machines-ab6d8d3178f1
- https://serverguy.com/comparison/cpu-vs-gpu-vs-tpu/
- https://www.geeksforgeeks.org/intuition-of-adam-optimizer/

Bounds on approximating Max kXOR with quantum and classical local algorithms

Kunal Marwaha* and Stuart Hadfield

Hybrid quantum-classical methods offer a new paradigm for tackling challenging learning and optimization tasks. While quantum approaches to combinatorial optimization such as the quantum approximate optimization algorithm (QAOA) [2] have attracted much recent attention, relatively few results are known concerning their potential for quantum advantage. To clarify this, a recent line of work has sought to compare the performance of low-depth realizations of QAOA with *local* classical algorithms, where in each case the algorithm's output for each variable depends only on a fixed neighborhood around that variable. We continue this research direction on the combinatorial optimization problem Max kXOR, a generalization of two problems previously studied with quantum algorithms (MaxCut and Max E3LIN2).

We start by analyzing the depth-1 QAOA and a natural generalization of the threshold algorithm [4] on regular, triangle-free Max kXOR instances. This comparison is motivated by the algorithms' similar performance on MaxCut [3, 5]. On instances with either *random signs* or *no overlapping clauses* and $D+1$ clauses per variable, we calculate the average satisfying fraction of both algorithms. Notably, the quantum algorithm outperforms the threshold algorithm for $k > 4$.

On the other hand, we highlight potential difficulties for the QAOA to achieve computational quantum advantage on this problem. We first identify the optimal satisfying fraction of large random instances for all k. We do this by conducting numerical calculations of the ground state energy density $P(k)$ of a mean-field k-spin glass [6] for all $k < 35$. Both algorithms perform far from optimally as k increases. We observe the value $P(k)$ converges to $\sqrt{2\log 2}$ (its large-k limit) by $k = 15$.

We also identify a new obstruction result for low-depth quantum circuits (including the QAOA) when $k = 3$, generalizing a result of [1] when $k = 2$. Obstructions rule out low-depth circuits from providing quantum advantage for optimization on some problems. We extend the NLTS result from quantum circuits with global bit-flip $X^{\otimes|V|}$ ($\mathbb{Z}_2$) symmetry, such as QAOA for MaxCut, to those where the circuit unitary commutes with an operator $X^{\otimes|V_+|}$ acting on a large, fixed subset $V_+ \subsetneq V$ of qubits. We call this *partial $\mathbb{Z}_2$ symmetry*:

Theorem 1 (informal)**.** *There exists a family of Hamiltonians with partial $\mathbb{Z}_2$ symmetry that have no low-energy states accessible from low-depth quantum circuits with the same partial $\mathbb{Z}_2$ symmetry.*

The Hamiltonians we construct are instances of the Max 3XOR problem. The QAOA unitary inherits the symmetry from the problem instance. This implies the following:

Corollary 1 (informal)**.** *There exists a family of fully-satisfiable Max 3XOR instances where the QAOA can only satisfy* 99% *of clauses until depth at least logarithmic in the problem size.*

We conjecture this *partial $\mathbb{Z}_2$ symmetry* technique can be extended to show a constant-fraction obstruction to QAOA at every k.

[1] Bravyi, S., Kliesch, A., Koenig, R., Tang, E.: Obstacles to state preparation and variational optimization from symmetry protection. arXiv preprint (2019), https://arxiv.org/abs/1910.08980

[2] Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint (2014), https://arxiv.org/abs/1411.4028

[3] Hastings, M.B.: Classical and quantum bounded depth approximation algorithms (2019), https://arxiv.org/abs/1905.07047

[4] Hirvonen, J., Rybicki, J., Schmid, S., Suomela, J.: Large cuts with local algorithms on triangle-free graphs. The Electronic Journal of Combinatorics pp. P4–21 (2017), https://arxiv.org/abs/1402.2543, preliminary version on arXiv, 2014

[5] Marwaha, K.: Local classical MAX-CUT algorithm outperforms $p = 2$ QAOA on high-girth regular graphs. Quantum **5**, 437 (Apr 2021). https://doi.org/10.22331/q-2021-04-20-437, https://arxiv.org/abs/2101.05513

[6] Sen, S.: Optimization on sparse random hypergraphs and spin glasses (2017), https://arxiv.org/abs/1606.02365

*NSF GRFP DGE-1746045, kmarw@uchicago.edu

Quantum Feature Maps for Binary Classification

Dr. Manjunath Ramachandra Iyer[1], Kunal Sunil Kasodekar[2], and Kiran Venkata Narasimha Naga[3]

[1,2]*Wipro Limited, Bangalore*
[1,2]{*manjunath.iyer, kunal.kasodekar, kiran.naga*}*@wipro.com*

Abstract

In this paper, we explore the feasibility and expressibility of using Quantum feature maps for binary classification problems to leverage their rich data representation in Hilbert space. We classify Quantum feature maps generated on synthetic datasets using QSVM and propose a two-stage pipeline for binary classification of real images.

1 Introduction

Machine learning models inherently learn probability distributions of the target labels given a set of tunable parameters. However, the intrinsic nature of Quantum Computing results represents real-world probability distributions making it an ideal candidate to bolster current Machine learning models. Quantum Computing in principle can provide exponential speedup over classical systems by utilizing quantum properties like entanglement, superposition, and, interference [1]. Quantum systems like variational circuits can access higher-dimensional space (Hilbert Space) to produce highly expressive machine learning models [2]. Although many quantum systems are not useful in this Noisy Intermediate-Scale Quantum (NISQ) era some of them inherently mitigate error [3]. These properties of a quantum system coupled with its ability to solve intractable problems make it a lucrative field for research and investigating practical quantum-inspired applications. Generally, the accuracy of QML models is lower than their classical counterparts but its compensated by speed and scalability. Hybrid approaches are being followed to exploit the strength of both worlds.

Quanvolutional neural networks are quintessential ConvNets with an essential modification for richer feature extraction [4]. Quantum transformations are employed as filters at the convolution layers. These transformations can provide a qualitative advantage and can be used on NISQ devices due to their noise-resilience from inherent shallow circuits. This hybrid quantum-classical solution is also easier to integrate into current deep learning frameworks. We use a Quanvolutional Neural Net to generate the feature map for our experiment. We also use a Quantum Support Vector Machine (QSVM) for binary classification by utilizing a quantum kernel estimator to estimate the kernel in the feature space [5].

2 Experiment and Results

A Quantum optimiser, due to inherent parallelism, can find the global minima accurately. As a result, feature maps generated are accurate and able to find decision boundaries precisely. In previous works, PCA is used to limit the data samples ingested to the Quantum algorithm. Instead of PCA, the most relevant samples representing the class generated at the Penultimate layer of a CNN/Quanvolutional encoder or Autoencoder can be used. It provides a better and explainable representation apart from making the decisions relevant to the data.

In this initial experiment, a two-class classifier is proposed over both quantum only and quantum-classical hybrid approaches. The usage of the most relevant features to segregate the

classes instead of Principle Components is proposed to make the decisions relevant and explainable. Here the synthetic random data for a two-class classifier is considered. Instead of PCA, the two most relevant samples of the penultimate layer of the 28 layers AlexNet are taken and ingested to the variational quantum circuit running over a Two QBit system. The feature maps generated by conventional and quantum algorithms are provided are illustrated in Figure 1 and Figure 2. We can see that the quantum feature map can distinguish the classes very well. The Implementation was carried out over an IBM Quantum computer. This generated feature map is then fed to a Quantum SVM classifier. The performance was close to a classical SVM classifier.

Afterwards, we found an interesting use case to demonstrate the real-world significance of Quantum Computing. We started working on a binary classification problem of Surface crack detection on concrete surfaces with real-world data [4][6].

2.1 Surface defect detection

Surface defect detection is one of the key aims of the quality control process in any large manufacturing setup like the automotive industry, Basic Metal Production industry, etc. Detecting defects such as hairline cracks, scratches, abrasions etc is not easy and any human errors in detecting this, can shorten the serviceability of the product and at times might lead to safety issues. Automatic defect detection can not only avoid human errors but also adapts to an unsuitable environment thus improving production efficiency and product quality. We experimented with a Quanvolutional neural network for this problem [4]. Even after multiple efforts to tune the number of layers, channels, qubits, rotations, encoding we found discouraging results i.e very low test accuracy [Figure 3]. It seemed to encounter barren plateaus or overfit in most cases. To find a solution to this problem in our ongoing experiment we have proposed a cascaded pipeline that utilizes Quantum feature maps as follows:

- Pass the image through an initial Quanvolutional Layer (Fine-tuned).
- Obtain the feature map generated at the output of the Quantum Convolution Layer.
- Feed this feature map to our QSVM for classifying the images as cracked or not cracked.

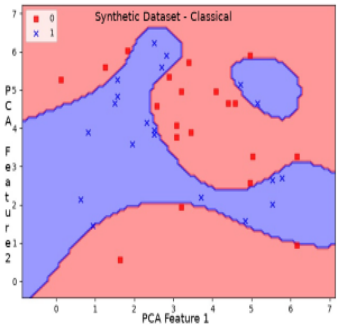


Figure 1: Classical

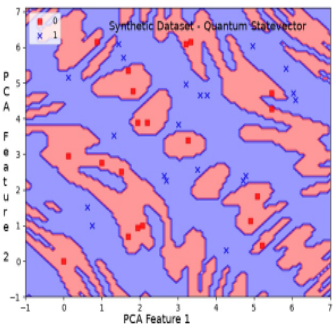


Figure 2: Quantum

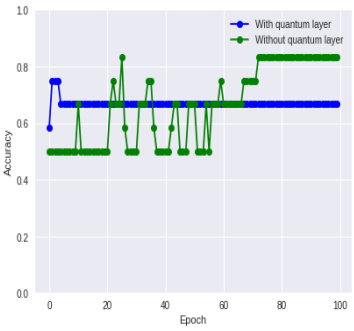


Figure 3: QCNN

3 Conclusion

With the advent of large noise-free Quantum systems near, it is important to assess, experiment with the feasibility and capacity of current Quantum machine algorithms on real-world problems. We experimented with the feasibility of using Quantum Feature Maps for binary classification using a QSVM on synthetic data and found lower but comparable accuracy with a classical SVM. We are currently experimenting with using a two-stage quantum pipeline on a real-world binary classification problem.

References

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

[2] Sim, Sukin and Johnson, Peter D. and AspuruGuzik, Aln. (2019) Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid QuantumClassical Algorithms. In *Adv. Quantum Technol. 2 (2019) 1900070*

[3] Preskill, John. (2018) Quantum Computing in the NISQ era and beyond. In *Quantum 2, 79*

[4] Havlek, Vojtch and Crcoles, Antonio D. and Temme, Kristan and Harrow, Aram W. and Kandala, Abhinav and Chow, Jerry M. and Gambetta, Jay M. (2019) Supervised learning with quantum-enhanced feature spaces. In *Nature. vol. 567, pp. 209-212*

[5] Maxwell Henderson and Samriddhi Shakya and Shashindra Pradhan and Tristan Cook. (2019) Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits. In *Quantum Machine Intelligence*

[6] Arun Pandian R. (2019) *Surface Crack Detection Dataset*. Retrieved from https://www.kaggle.com/arunrk7/surface-crack-detection

Universal and optimal coin sequences for high entanglement generation in 1D discrete time quantum walks

Aikaterini Gratsea [1,2] Friederike Metz [1] and Thomas Busch [1]

[1] *Quantum Systems Unit, Okinawa Institute of Science and Technology Graduate University, 1919-1 Tancha, Onna, Okinawa 904-0495, Japan*
[2] *ICFO - Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, Av. Carl Friedrich Gauss 3, 08860 Castelldefels (Barcelona), Spain*

We generate highly entangled states in a discrete time quantum walk irrespective of the initial state using a deterministic sequence and a reinforcement learning algorithm. Our methods are useful when the initial state is not fully known or entanglement has to be generated universally for different initial states.

Machine learning has already achieved remarkable results in various areas of physics and different machine learning approaches have been combined with quantum walks for exploring quantum speed-up [1, 2] and graph structures [3]. Reinforcement learning (RL) has also been successfully applied to challenging problems in quantum physics including quantum state preparation [4, 5], quantum optimal control [6, 7], and quantum error correction [8, 9].

We use a RL technique to tackle the optimization of entanglement in quantum walks. Entanglement is a key resource in many quantum information applications and achieving high values independently of the initial conditions is an important task. While most often entanglement between different states of the same degree of freedom is considered, more recently the so-called hybrid entanglement between different degrees of freedom has been of interest [10]. If these degrees of freedom belong to the same particle, more information can be encoded at the single particle level, which helps to reduce required resources.

We discuss the generation of hybrid entangled states that can be created in discrete time quantum walks, where the motion of a particle that moves in a high-dimensional discrete space depends on an internal, two-dimensional coin degree of freedom [11]. The evolution itself consists of the recurrent application of a coin and shift operator which in general leads to entanglement between the walker and coin. To be experimentally realistic we restrict to Hadamard and Fourier coins only and aim at a minimal number of steps.

In contrast to previously employed optimization schemes [12–14], RL allows us to not only find the optimal sequence of coin operators for a specific initial state but also for classes of initial states. The resulting optimized coin sequences achieve on average higher values of entanglement than the deterministic (universal) sequence of coin operators. While the amount of entanglement produced by the deterministic sequence is fully independent of the initial states considered, the optimized sequences achieve in general higher average values of entanglement that do however depend on the initial state parameters. Our proposed sequence and optimization algorithm are especially useful in cases where the initial state is not fully known or entanglement has to be generated in a universal manner for a range of initial states.

Our work extends existing results that either achieve state independent entanglement only in the asymptotic limit of an infinite quantum walk [12] or that achieve maximal entanglement in a short sequence, but not independently of the initial state [13, 14]. Furthermore, our RL scheme can be useful in a variety of other, experimentally relevant settings. For example, one could restrict the class of initial states to match the experimental problem, such as a fixed initial state with noise. The RL objective can also be altered in different ways. One could for example choose to maximize the fidelity between the final state and a given target state. Another option is to apply the techniques to higher dimensional quantum walks [15], quantum walks on graphs [16], or quantum walks involving more than one particle [17]. Additionally, one could move to the continuous case and use deep reinforcement learning to directly optimize the parameters in the coin operator.

The full article can be found here: https://iopscience.iop.org/article/10.1088/1751-8121/abb54d.

[1] A. A. Melnikov, L. E. Fedichkin, and A. Alodjants, (2019), arXiv:1901.10632 [quant-ph] .
[2] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Phys. Rev. X **4**, 031002 (2014).
[3] S. Dernbach, A. Mohseni-Kabir, S. Pal, M. Gepner, and D. Towsley, Appl. Netw. Sci. **4**, 2364 (2019).
[4] M. Bukov, Phys. Rev. B **98**, 224305 (2018).
[5] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Phys. Rev. X **8**, 031086 (2018).
[6] Z. T. Wang, Y. Ashida, and M. Ueda, (2019), arXiv:1910.09200 [quant-ph] .
[7] M. Niu, S. Boixo, V. Smelyanskiy, and H. Neven, Npj Quantum Inf. **5**, 33 (2019).
[8] R. Sweke, M. S. Kesselring, E. P. L. van Nieuwenburg, and J. Eisert, (2018), arXiv:1810.07207 [quant-ph] .
[9] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Phys. Rev. X **8**, 031084 (2018).
[10] F. Flamini, N. Spagnolo, and F. Sciarrino, Rep. Prog. Phys. **82**, 016001 (2018).
[11] Y. Aharonov, L. Davidovich, and N. Zagury, Phys. Rev. A **48**, 1687 (1993).
[12] R. Vieira, E. P. M. Amorim, and G. Rigolin, Phys. Rev. Lett. **111**, 180503 (2013).
[13] L. Innocenti, H. Majury, T. Giordani, N. Spagnolo, F. Sciarrino, M. Paternostro, and A. Ferraro, Phys. Rev. A **96**, 062326 (2017).

[14] A. Gratsea, M. Lewenstein, and A. Dauphin, Quantum Sci. Technol. **5**, 025002 (2020).
[15] T. D. Mackay, S. D. Bartlett, L. T. Stephenson, and B. C. Sanders, J. Phys. A: Math. Gen. **35**, 2745 (2002).
[16] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani (2001) p. 50–59.
[17] Y. Omar, N. Paunković, L. Sheridan, and S. Bose, Phys. Rev. A **74**, 042304 (2006).

NON-PARAMETRIC MULTIVARIATE CONTINUOUS DENSITY ESTIMATION IN A QUANTUM COMPUTER

EXTENDED ABSTRACT SUBMITTED TO QTML2021

FABIO A. GONZÁLEZ[1], **VLADIMIR VARGAS-CALDERÓN**[2], **ALEJANDRO GALLEGO**[1], **AND HERBERT VINCK-POSADA**[2]

[1]MindLab Research Group, Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia, Bogotá, Colombia

[2]Grupo de Superconductividad y Nanotecnología, Departamento de Física, Universidad Nacional de Colombia, Bogotá, Colombia

ABSTRACT

This work presents a method to perform non-parametric density estimation in $\mathbb{R}^d$ in a quantum computer. The method uses a quantum feature map based on random Fourier features to build a density matrix, ρ_{train}, that summarizes the training data set. A quantum circuit prepared with ρ_{train} estimates the density of a new sample x^* by calculating the amplitude of the projection of ρ_{train} on the quantum state representing x^*.

Introduction.— The problem of density estimation is given a sample $X = \{x_1, \ldots, x_N\}$, find a function $\hat{f}$ that approximates the true probability density function (pdf) of the distribution from which X was generated. Many tasks in machine learning can be approached as density estimation problems: supervised learning can be seen as the problem of estimating the joint probability of inputs and labels $P(\mathcal{X}, \mathcal{Y})$, while unsupervised and generative learning can be addressed by estimating the probability $P(\mathcal{X})$. A quantum density estimation routine may serve as a building block for developing different quantum machine learning algorithms, e.g. anomaly detection [1] and generative models [2].

The quantum formalism can be seen as a generalization of classical probability. A density matrix in an d-dimensional Hilbert space can be viewed as a catalog of categorical distributions on the finite set $\{1 \ldots d\}$. If we want to efficiently model continuous probability distributions in $\mathbb{R}^d$, a direct application of this fact is not very useful. In [3] we show that it is possible to model arbitrary probability distributions on $\mathbb{R}^d$ using finite-dimensional density matrices together with random Fourier features (RFF) [4]. In this work, we use this idea to build a quantum circuit that is able to perform multivariate continuous density estimation.

The method can be seen as a quantum implementation of the very well known Kernel Density Estimation (KDE) [5, 6], also known as Parzen-Rossenblat window density estimation. The method overcomes one of the main weaknesses of KDE namely the fact that it is a memory-based method that requires storing the entire data set to make a prediction. This is done by summarizing the training dataset as a density matrix. The density matrix is the average of a quantum feature map based on RFF [4]. Thanks to this, the associated quantum kernel [7] approximates the Gaussian kernel, and performing a measuring over a particular quantum state approximates the estimation of a Gaussian-kernel-based KDE.

The method.— During training we calculate a density matrix that summarizes the training dataset, $X = \{x_i\}_{i=1\ldots N}$ with $x_i \in \mathbb{R}^d$, as follows:

1. Given a RFF embedding, $\phi_{\text{rff}} : \mathbb{R}^d \to \mathbb{R}^D$, with the property that $\forall x, y \in \mathbb{R}^d,\ k(x,y) \approx \langle \phi_{\text{rff}}(x), \phi_{\text{rff}}(y) \rangle$, where k is the Gaussian kernel, define the quantum feature map $\left|\bar{\phi}_{\text{rff}}(x)\right\rangle = \sum_j \frac{\phi_{\text{rff},j}(x)}{||\phi_{\text{rff}}(x)||} \left|j\right\rangle$, where $\left|j\right\rangle$ is the bit string multi-qubit state corresponding to the binary representation of j.
2. Calculate $\left|\Psi_{\text{train}}\right\rangle = \frac{1}{N} \sum_{i=1}^{N} \left|\bar{\phi}_{\text{rff}}(x_i)\right\rangle$

During prediction, the density of a new sample x is calculated by the Born rule, $\hat{f}_{\rho_{\text{train}}}(x) = \frac{1}{M_\gamma} \left\langle \bar{\phi}_{\text{rff}}(x) \right| \rho_{\text{train}} \left| \bar{\phi}_{\text{rff}}(x) \right\rangle$, where $M_\gamma = (\pi/\gamma)^{\frac{d}{2}}$, with γ the spread parameter of the Gaussian kernel, and $\rho_{\text{train}} = \left|\Psi_{\text{train}}\right\rangle\left\langle\Psi_{\text{train}}\right|$.

The circuit.— In [3] we showed how to implement a more general version of the above method, which uses a mixed density matrix, in a classical computer as a differentiable model that can be also trained using gradient descent. Here, we demonstrate that we can use a quantum computer [1] (IBM Lima) to realize probability density estimation.

The density estimation circuit must compute $\left\langle \bar{\phi}_{\text{rff}}(x) \right| \rho_{\text{train}} \left| \bar{\phi}_{\text{rff}}(x) \right\rangle$. This is equivalent to measuring the amplitude of the projection of $\left|\Psi_{\text{train}}\right\rangle$ onto $\left|\bar{\phi}_{\text{rff}}(x)\right\rangle$. This can be done with a gate U_d that prepares $\left|\Psi_{\text{train}}\right\rangle$ for a given data set, followed by the inverse of a gate $U_\star$ that prepares the state of a single sample point where the density is to be estimated. Therefore, density estimation is achieved by estimating, through multiple shots, the quantity $\left|\left\langle 0^n \right| U_d(X) U_\star^\dagger(x) \left| 0^n \right\rangle\right|^2$, where $n = \lceil \log_2 D \rceil$ is the number of qubits that encode the D-dimensional random Fourier features space. In other words, the density is estimated by measuring the frequency of the 0 bit string after applying $U_d(X)$ and $U_\star^\dagger(x)$ consecutively to the state $\left|0^n\right\rangle$.

Results.— We tested our framework with a simple dataset distributed on two overlapping Gaussians, as shown in the rug-plots in fig. 1. A total of three qubits were used to encode 8 random Fourier features, which are sufficient to generate a high-quality density estimation, close to the classical Gaussian kernel density estimation shown in the orange lines in the panels of fig. 1. Our density matrix kernel density estimation method was simulated on an ideal quantum circuit, on a noisy quantum circuit[2], and run on the IBM Lima quantum computer, as shown in the left, central and right panels of fig. 1, respectively.

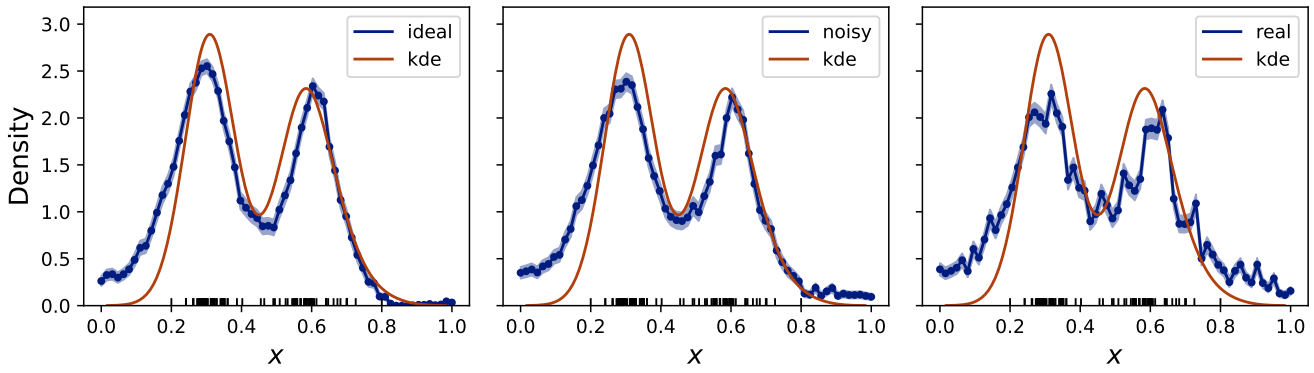


Figure 1: Density estimation (blue points) of bi-Gaussian-distributed data with exact circuit simulation (left), noisy circuit simulation (middle) and run on the IBM Lima quantum computer. Orange lines are computed through regular Gaussian kernel density estimation. 1024 shots were used to estimate every point on a (simulated or real) quantum computer. Confidence intervals are computed with the asymptotic normal approximation of the Bernoulli distribution from which measurements are sampled.

[1] A qubit-based quantum computer. However, we have also shown that the method can be implemented in qudit-based quantum computers [8]

[2] The noisy quantum circuit is modeled in qiskit [9] by considering initialization, readout, depolarizing and thermal relaxation errors measured on each device, as reported by IBM.

References

[1] Nana Liu and Patrick Rebentrost. Quantum machine learning for quantum anomaly detection. *Physical Review A*, 97(4):042315, 2018.

[2] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *Physical review letters*, 121(4): 040502, 2018.

[3] Fabio A González, Alejandro Gallego, Santiago Toledo-Cortés, and Vladimir Vargas-Calderón. Learning with density matrices and random features. *arXiv preprint arXiv:2102.04394*, 2021. URL `https://arxiv.org/abs/2102.04394`.

[4] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1177–1184, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

[5] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, 27(3): 832–837, 09 1956. doi: 10.1214/aoms/1177728190. URL `https://doi.org/10.1214/aoms/1177728190`.

[6] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[7] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.

[8] Diego H Useche, Andres Giraldo-Carvajal, Hernan M Zuluaga-Bucheli, Jose A Jaramillo-Villegas, and Fabio A González. Quantum measurement classification with qudits. *arXiv preprint arXiv:2107.09781*, 2021. URL `https://arxiv.org/abs/2107.09781`.

[9] MD SAJID ANIS, Héctor Abraham, AduOffei, Rochisha Agarwal, Gabriele Agliardi, Merav Aharoni, et al. Qiskit: An open-source framework for quantum computing, 2021.

On Improving the Performance of Quantum Classifiers Through Feature Engineering

Sonai Biswas[†1], Sayantan Pramanik[+2], M Girish Chandra[*3] and Kriti Kumar[*4]

[†]*Indian Institute of Science, Education and Research, Kolkata* [+]*TCS Incubation* [*]*TCS Research*

[1]*sb17ms124@iiserkol.ac.in* [234]{*sayantan.pramanik, m.gchandra, kriti.kumar*}*@tcs.com*

Abstract

The present and future Quantum Machine Learning architectures involve a synergetic combination of quantum and classical computations. In this poster, through a set of representative results, how classical preprocessing in terms of appropriately engineered features can influence the performance of quantum classifiers for a given number of qubits, is brought out. The study is carried out through simulations using Gearbox and Bearing Fault Data set.

1. Quantum Machine Learning

Quantum Machine Learning (QML) is at the crossroads of two of the most exciting current areas of research: quantum computing and classical machine learning [1]. In QML, the proposition is to enhance the performance of machine learning tasks by using quantum computers, like, improving the accuracy, reducing the number of training samples, speeding up the process of training and inference, among others. In the present Noisy Intermediate Scale Quantum (NISQ) computing era, the hybrid quantum-classical processing has established itself as an essential combination, and this "cooperation" will continue for a long time.

2. Gearbox Data set and Classification

One of the well studied task in QML is the supervised learning, in particular, classification. In this work we have considered the Gearbox and Bearing Fault Data set (see [2] and the references therein), where, the data is collected through Drivetrain Dynamic Simulator. In the existing classical Machine Learning/Signal Processing works, different time domain and frequency domain features are used to classify four different faults for both gearbox and bearings, along with the healthy state.

3. PCA and Feature Selection

A standard procedures in QML is to reduce the feature vector size (dimensionality reduction) through classical Principal Component Analysis(PCA) to map the data into fewer qubits for both quantum simulators and NISQ implementation. Instead of invoking PCA preprocessing, we explored selecting the right smaller subset of features. The rationale is, since classical computing is already an integral part of the quantum classifiers, either the variational or support vector machine versions, we can use it for feature engineering as well. For the fully classical results, the root mean square value and variance have been identified as discriminating features for the task at hand (see [2]). The selection can be carried out by considering recursive feature elimination, for example.

4. Implementation of Multi-class Classification

Fig. 1 portrays the variational quantum classification scheme employed in this work, specifically, R_Z encoding was used, followed by an ansatz of strongly entangling layers. We implemented multi-class classification by considering one-versus-one binary classifiers (see Fig. 2). For this poster, we started with the 6 extracted time-domain features; Root Mean Square value, variance, maximum value, kurtosis, skewness and peak-to-peak value. Since categorization of the features has to be done into 5 different classes, we need $\binom{5}{2} = 10$ binary variational quantum classifiers for each comparison scenario. The five possible categories for the Gear Box are - chipped tooth, missing tooth, root fault, surface fault and healthy. Further, we have two different sets of classifiers - one for a subset of original features and the other for PCA-reduced modified features. Needless to say, for testing or inference, appropriately trained classifiers are utilized for either the selected features or PCA-reduced vectors.

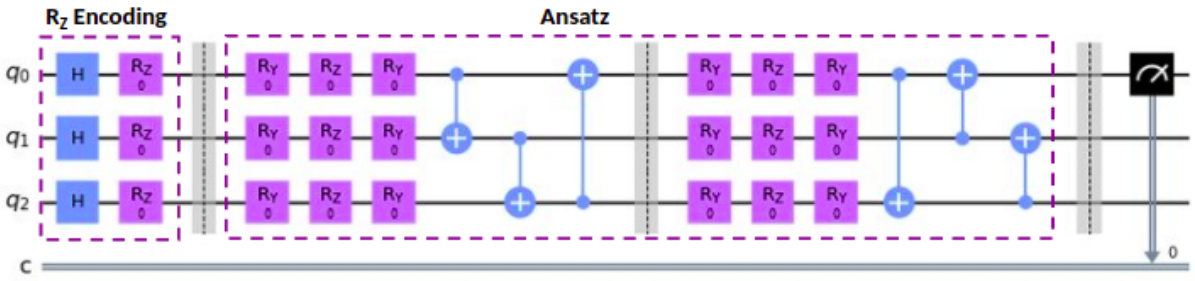


Fig. 1: Quantum Circuit with 3 qubits (for illustration purposes) used for binary classification. The parameters to the R_Z gates in the encoding layer accommodate the inputs, while the parameters in the ansatz are optimizable.

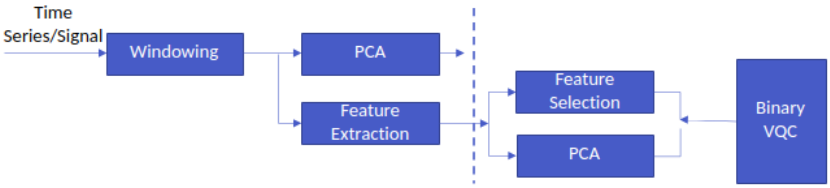


Fig. 2: Block Diagram of the training phase.

5. Results and Discussion

As a preliminary examination, we carried out simulation studies by considering all the six time-domain features and the aforementioned two engineered features. In order to compare with the standard pipeline, we also considered reducing the 6 features to two Principal Component values. With the variational classifier involving only simple Rz rotations for data encoding (to quantum states), we found that the engineered features gave 100% accuracy for nine one-vs-one classifiers out of 10. The results are captured in Table 1 and in all the cases, two properly engineered features outperformed the PCA scenario. The trained models were tested against a new batch of 255 data points and the accuracy achieved for the multi-class classification was 85.49% for the representative features, 92.16% when all the features were considered, but reduced to an abysmal 45.49% for the PCA-reduced feature vectors; and the misclassifications were between classes 3 and 5, as per expectation. This demonstrates the efficacy of using data set specific feature-engineering over generic techniques such as PCA. Further experimentation can be carried out to improve the accuracy by considering different data encoding strategies (say, *ZZ* feature maps), on which work is underway. One can also consider building an ensemble of classifiers on a couple of subsets of (engineered) features and using them for the final inference. More systematic construction quantum multi-class classifiers with engineered features rather than primitive one-vs-one and one-vs-rest can be another useful thread.

	1-vs-2	**1-vs-3**	**1-vs-4**	**1-vs-5**	**2-vs-3**	**2-vs-4**	**2-vs-5**	**3-vs-4**	**3-vs-5**	**4-vs-5**
All Features	100	100	100	100	100	100	100	100	69.23	100
2 Features	100	100	100	100	100	100	100	100	61.53	100
PCA	57.69	69.23	84.61	80.76	61.54	57.69	76.92	69.23	57.69	46.15

Table 1: Accuracy metrics obtained from VQC using all the 6 features, the 2 best representative features, and PCA-reduced feature vectors.

References

1. J. Allcock and S. Zhang, Quantum machine learning, National Science Review, 2019, Vol. 6, No. 1, pp. 26-28.
2. Kriti Kumar, Saurabh Sahu, Angshul Majumdar and M Girish Chandra, AutoFuse: A Semi-supervised Autoencoder based Multi-Sensor Fusion Framework, The International Joint Conference on Neural Networks (IJCNN), July 2021.

LEARNING WITH DENSITY MATRICES

EXTENDED ABSTRACT SUBMITTED TO QTML2021

FABIO A. GONZÁLEZ[1], **ALEJANDRO GALLEGO**[1], **SANTIAGO TOLEDO-CORTÉS**[1], **AND VLADIMIR VARGAS-CALDERÓN**[2]

[1]MindLab Research Group, Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia, Bogotá, Colombia

[2]Grupo de Superconductividad y Nanotecnología, Departamento de Física, Universidad Nacional de Colombia, Bogotá, Colombia

ABSTRACT

This work explores how density matrices can be used as a building block for machine learning models exploiting their ability to straightforwardly combine linear algebra and probability. One of the main results of this work is to show that density matrices coupled with random Fourier features could approximate arbitrary probability distributions over $\mathbb{R}^n$. Based on this finding the work builds different models for density estimation, classification and regression. These models are differentiable, so it is possible to integrate them with other differentiable components, such as deep learning architectures and to learn their parameters using gradient-based optimization. In addition, the work presents optimization-less training strategies based on estimation and model averaging. The models are evaluated in benchmark tasks and the results are reported and discussed. This extended abstract is based on arXiv:2102.04394 [1].

Introduction.— The formalism of density operators and density matrices was developed by von Neumann as a foundation of quantum statistical mechanics [2]. From the point of view of machine learning, density matrices have an interesting feature: the fact that they combine linear algebra and probability, two of the pillars of machine learning, in a very particular but powerful way.

The main question addressed by this work is how density matrices can be used in machine learning models. One of the main approaches to machine learning is to address the problem of learning as one of estimating a probability distribution from data, joint probabilities $P(x, y)$ in generative supervised models or conditional probabilities $P(y|x)$ in discriminative models.

The central idea of this work is to use density matrices to represent these probability distributions tackling the important question of how to encode arbitrary probability density functions in $\mathbb{R}^n$ into density matrices. This is accomplished by using random Fourier features (RFF) [3] as a quantum feature map. The RFF method builds an embedding $\phi_{\text{rff}} : \mathbb{R}^d \to \mathbb{R}^D$ given a shift-invariant kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that $\forall x, y \in \mathbb{R}^d,\ k(x, y) \approx \langle \phi_{\text{rff}}(x), \phi_{\text{rff}}(y) \rangle$. Given a training dataset $X = \{x_i\}_{i=1...N}$ with $x_i \in \mathbb{R}^d$, a training density matrix is built as the average of the density matrices of each training sample, $\rho_{\text{train}} = \frac{1}{N} \sum_{i=1}^{N} |\phi_{\text{rff}}(x_i)\rangle \langle \phi_{\text{rff}}(x_i)|$. The density of a new sample is calculated by applying the Born rule, $\hat{f}_{\rho_{\text{train}}}(x) = \frac{1}{M_\gamma} \langle \phi_{\text{rff}}(x)| \rho_{\text{train}} |\phi_{\text{rff}}(x)\rangle$.

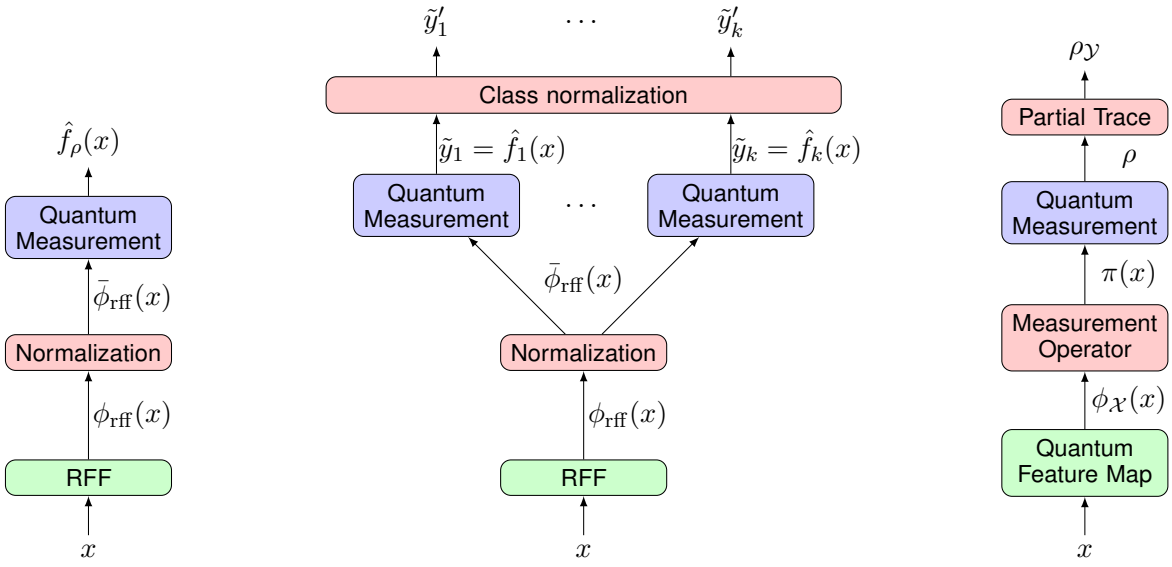
Methods.— Based on the idea described above, different models for density estimation, classification and regression are defined. The models can be trained using two different strategies: one, using averaging of the training density matrices; and two, thanks to the fact that the models are differentiable, using stochastic gradient descent over the

parameters to minimize an appropriate loss function. The models can be integrated as layers in conventional deep neural networks and can be easily implemented in GPU-accelerated machine learning frameworks. Figure 1 shows three of the models, which are briefly explained next.

Density matrix kernel density estimation (DMKDE) (Figure 1a). The model receives an input $x \in \mathbb{R}^d$, represent it using a RFF quantum feature map, ϕ_{rff}, and estimates the density of it using the Born rule. The model can be trained by averaging the density matrices corresponding to the training samples or by using stochastic gradient descent.

Density matrix kernel density classification (DMKDC) (Figure 1b). The extension of kernel density estimation to classification is called kernel density classification [4]. Following this strategy, DMKDC uses DMKDE to estimate the likelihood of a given sample for the different classes and applies the Bayes rule to obtain the class posteriors.

Quantum measurement classification (QMC) (Figure 1c). This model generalizes DMKDC to deal with more general probability distributions of the outputs. The idea is to model the joint probability of inputs and outputs using a density matrix. A prediction is made by performing a measurement with an operator specifically prepared from a new input sample, producing a density matrix.



(a) DMKDE (b) DMKDC (c) QMC

Figure 1

Experimental evaluation.— Some experiments were performed to evaluate the performance of the proposed methods in different benchmark tasks for density estimation, classification and ordinal regression. The experimental results showed that the models exhibit a competitive performance providing evidence that density matrices are a useful component to build competitive machine learning models.

References

[1] Fabio A González, Alejandro Gallego, Santiago Toledo-Cortés, and Vladimir Vargas-Calderón. Learning with density matrices and random features. *arXiv preprint arXiv:2102.04394*, 2021. URL `https://arxiv.org/abs/2102.04394`.

[2] John Von Neumann. Wahrscheinlichkeitstheoretischer aufbau der quantenmechanik. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1927:245–272, 1927.

[3] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1177–1184, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

[4] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Quantum Classifiers for Transcriptome Analysis of SARS-CoV-2 Infection

Ju-Young Ryu[1], Gwonhak Lee[2], Minjin Choi[2], Seongmo Kang[3], Hyun Uk Kim[3], Francesco Petruccione[2,4], June-Koo Kevin Rhee[1,2]

[1] *School of Electrical Engr. & KAIST Inst. of IT Convergence, KAIST, Daejeon, S. Korea*

[2] *Qunova Computing, Daejeon, S. Korea*

[3] *Dept. of Chemical and Biomolecular Engineering, KAIST, Daejeon, S. Korea*

[4] *School of Chemistry and Physics, UKZN, Durban, Rep. of South Africa*

Short Abstract

Quantum classifier models are examined for classifying genes using RNA sequencing data whether or not they are involved in SARS-CoV-2 infection pathways, and compared with commonly used classical classifiers. Results show that quantum models have similar accuracy to classical models, thus providing a practical application of quantum classifiers.

Objective of the work

The variational quantum classifier model [1] [2] turns to be an important practical application of NISQ machines. A recent work [3] has shown that depending on the dataset and data embedding, quantum classifiers may outperform classical models. It is highly anticipated to find practical examples of variational quantum classifiers. Also, one of the most significant tasks for the scientific community since the COVID19 pandemic began is to understand what gene expression is affected by the COVID19 infection. Motivated by these circumstances, this paper aims to train a quantum model to classify whether or not a gene is involved in the SARS-CoV-2 infection pathway by using RNA sequencing data.

Methods and Results

1. RNA Sequencing Data

The dataset used in this study is Series GSE147507 (Series 6 and 16) [4] in the GEO database [5], which show transcriptome changes of A549 cells (i.e., adenocarcinomic human alveolar basal epithelial cells) and A549 cells overexpressing ACE2, both infected with SARS-CoV-2. ACE2 serves as the entry point for SARS-CoV-2. The data is presented as a table with integer counts in each cell. These values represent expression levels of each gene in each biological sample. Each biological sample was obtained from a different condition, which allows observations of how cells react to different environments. The ground truth labels for the genes was determined by using 11 different infection-related pathways in the KEGG database [6]. 51 infection related genes and 53 unrelated genes were chosen for the entire dataset, with each gene having 12 features (counts from 12 different biological samples). Training dataset and test dataset was split 4:1.

For data preprocessing, 12 features were grouped into 6 pairs, and the expression level change between them was calculated using $\log_2$ fold change. This was reduced to 3

feature dimensions using PCA, then resized so that each feature is in $[0, \pi]$.

2. Variational Quantum Classifier

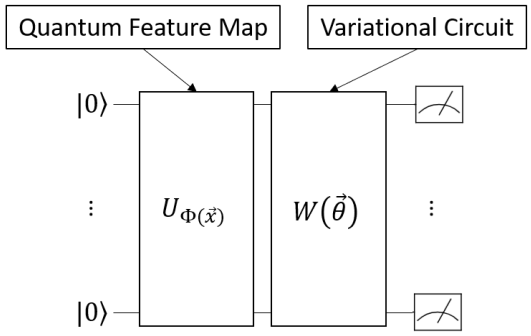


Figure 1. Variational Quantum Classifier

The quantum models used in this study is the variational quantum classifier. The classifier consists of a quantum feature map and a variational circuit. The n qubit computational basis measurement result is used as an input to a fixed function $f : \{0,1\}^{\mathrm{n}} \to \{1,-1\}$. One assigns label y to $\vec{\mathrm{x}}$ if $\Pr(f(\vec{x}) = y) > \Pr(f(\vec{x}) = -y) - yb$ for fixed constant b. The variational circuit is trained to minimize the empirical risk on the training dataset.

The IQP-style circuit proposed in [1] was used for the $\mathrm{U}_{\Phi(\vec{x})}$ circuit. The $\Phi(\vec{x})$ functions tested was the function used in [1] (denoted as ϕ_1) as well as $\phi_2(\vec{x}) = \vec{x}[\mathrm{m}] + \vec{x}[n]$, $\phi_3(\vec{x}) = \pi \cos(\vec{x}[m]) \cos(\vec{x}[n])$, $\phi_4(\vec{x}) = \pi \frac{\exp((\vec{x}[\mathrm{m}]+\vec{x}[n])^2)-1}{e-1}$ rotation angles for ZZ rotations between qubit m and n. These functions were inspired by those analyzed in [7]. For the variational circuit, 2 layers of 3-qubit versions of circuit 12 and 4 discussed in [8] were used. The function f was chosen so that if the majority is bit 0, label is 1 and if the majority is bit 1, the label is -1. b is fixed to be zero. An SPSA optimizer was used to train the circuits. 10000 shots were used to calculate the probabilities. SVC implemented in the python library *sklearn* with radial basis function kernel was the classical model used. The hyperparameter C was tuned to maximize test accuracy. Some of the training results are shown in the below table.

Classifier	Quantum Classifier					Classical Classifier
	ϕ_1,circuit4	ϕ_2,circuit4	ϕ_3,circuit4	ϕ_4,circuit4	ϕ_1,circuit12	SVC, C=32
Test Accuracy	0.6667	0.4762	0.5238	0.7619	0.6667	0.7143

Table 1. Classifier Training Results

The quantum classifier using ϕ_4 and circuit4 showed slightly better results compared to the classical classifier. Using quantum metric learning to search for better feature maps could be attempted to improve the quantum classifier, as proposed in [9].

Acknowledgements

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2018-0-01402) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). This work was supported by KAIST Institutes(KI).

Bibliography

[1] Havlíček, V., Córcoles, A.D., Temme, K. *et al.* Supervised learning with quantum-enhanced feature spaces. *Nature* **567,** 209–212 (2019)

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

[3] Huang, HY., Broughton, M., Mohseni, M. *et al.* Power of data in quantum machine learning. *Nat Commun* **12,** 2631 (2021)

[4] Blanco-Melo, D. et al. Imbalanced Host Response to SARS-CoV-2 Drives Development of COVID-19. *Cell* **181**, 1036-1045 (2020)

[5] Barret, T. et al. NCBI GEO: archive for functional genomics datasets – update. *Nucleic Acids Research* **41**, D991-D995 (2013) https://doi.org/10.1093/nar/gks1193

[6] Kanehisa, M. et al. KEGG: integrating viruses and cellular organisms. *Nucleic Acids Research* **49**, D545-D551 (2020) https://doi.org/10.1093/nar/gkaa970

[7] Suzuki, Yudai, et al. "Analysis and synthesis of feature map for kernel-based quantum classifier." *Quantum Machine Intelligence* **2**: 1-9 (2020) https://doi.org/10.1007/s42484-020-00020-y

[8] Sim, Sukin, 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)
https://doi.org/10.1002/qute.201900070

[9] Lloyd, Seth, et al. "Quantum embeddings for machine learning." *arXiv preprint* arXiv:2001.03622 (2020) https://arxiv.org/abs/2001.03622

Deep tensor networks with matrix product operators

Bojan Žunkovič

Faculty of Computer and Information science, University of Ljubljana, Ljubljana, Slovenia

Abstract

We revisit the image classification problem with matrix product states (MPS) by applying an extensive augmentation hyperparameter search. Obtained test error rates are 0.37% (MNIST), 7.6% (FashionMNIST), and 37.2% (CIFAR10). We further improve the MPS baseline models by introducing deep tensor networks based on the matrix product operator formalism.

1 Introduction

Tensor networks are ubiquitous in the field of low-dimensional many-body quantum physics due to their effectiveness in modelling short-range correlations and the locality structure of prototypical quantum Hamiltonians. Motivated by the success of describing quantum amplitudes tensor networks have been applied also to other domains (e.g. image classification [SS16, Sto18, EHL19, LRW+19, MVRL20, MZZ+20, CPD21, KLH21], generative modelling [CWXZ19, ST19, SPL+20, LLZZ21], sequence and language modelling [PV17, GJLP18, BST20, BV20], anomaly detection [WRVL20, SSL+20]) with no apparent structure, that might justify their use. Recent studies of correlations in several image datasets (MNIST, FashionMNIST, CIFAR10, TinyImages) [CCW18, MVRL20, CHLW21, LKNKC21] imply a limited utility of tensor networks (in particular matrix product states) for image classification and generation due to the existence of long-range correlations. Further, the state-of-the-art tensor networks achieve lower accuracy on simple image datasets (MNIST, FashionMNIST, CIFAR10) than the best or even benchmark neural networks. We partially fill this gap by using a tuned augmentation and deep tensor network models based on matrix product operator formalism.

We revisit the problem of image classification with matrix product states (MPS) and significantly increase the accuracy of MPS classifiers by using image augmentation (Section 2). We obtain an error rate of 0.37% on the simple MNIST dataset and 7.6% on the FashionMNIST dataset. In contrast, we achieve only 62.8% accuracy on the CIFAR10 dataset. We also evaluate the performance of transnational invariant models on the same datasets and find significantly lower performance in all cases. We also assess all models on the datasets with permuted inputs and find slightly larger error rates than in the non-permuted case. Finally, in Section 3, we introduce deep tensor networks based on the matrix product operator formalism and evaluate their performance on the MNIST dataset.

2 Baseline MPS models

We perform image classification with the MPS models in two steps. In the first, we embedded the image into a high dimensional Hilbert space via a local map $x \to \psi(x) = \begin{bmatrix} 1-x \\ x \end{bmatrix}$, where $x \in [0,1]$ denotes the scaled pixel intensity. Then a linear classification is performed as a maximum of the scalar products of the embedded vector with matrix product state vectors representing different classes. We chose the following matrix product state representation for a class c

$$\Psi_c = \frac{1}{D} \sum_{s_1,s_2,\dots s_n=1}^{n} \mathrm{tr}(A_1^{s_1} \cdot A_2^{s_2} \dots \cdot A_{n/2}^{s_{n/2}} B^c \cdot A_{n/2+1}^{s_{n/2+2}} \dots A_n^{s_n}) \psi(s_1) \otimes \psi(s_2) \otimes \dots \psi(s_n), \tag{1}$$

where $s_j \in \{0,1\}$ and $A_j^{s_j}, B^c$ are $D \times D$ matrices with a fixed bond dimension D. The logits for a given input are then given by $l_c = \Psi_c \cdot \Psi(x)$. We chose the initial condition such that the logits are close to one for any input x. In particular we chose $[A_j^{s_j}]_{ij} = I_D + R_j$, where I_D denotes a $D \times D$ identity matrix and R_j a $D \times D$ matrix with normally distributed elements with mean zero and variance ϵ. We use the standard cross-entropy loss with L2 regularization of the MPS matrices. In our final (best) setting,

we use batch size 500, AdamW optimizer, and reduce the learning rate on a plateau with $\gamma = 0.5$ and patience 20. We determine the optimizer has as a part of the hyperparameter tuning together with the augmentation.

2.1 Augmentation

We use *torchvision* to augment the dataset images and performed hyperparameter tuning for the various transformation parameters and probabilities. In particular, we use the following sequence of transformations: random pixel shift, random colour jitter, random sharpness, random Gaussian blur, random horizontal flip, random horizontal flip, random affine, random perspective, resize, crop, random elastic transformation, random erasing. We also tune the probability of applying each random transformation while keeping the order of transformations fixed. During hyperparameter tuning, we restrict the training to a maximum of 30 epochs.

2.2 Results

After finding the best hyperparameters, we train an MPS model and a translational invariant MPS model (TI-MPS) at increasing bond dimensions and standard and permuted datasets. In the following table, we report the error on MNIST and FashionMNIST test datasets. Along with mean error rates and standard deviation of the 10-fold cross-validation, we also report results for ensembles obtained by averaging logits of the obtained models. In the permuted case, we report the results for ensembles with the same permutation (Ens. Same) and different permutations (Ens. Rand) of the inputs in each model in the ensemble.

	MNIST (%)	pMNIST (%)	FashionMNIST (%)	pFashionMNIST (%)
MPS	0.50 ± 0.03	0.73 ± 0.08	8.45 ± 0.13	9.00 ± 0.13
MPS Ens. Same	**0.37**	**0.46**	**7.57**	**8.02**
MPS Ens. Rand	/	0.49	/	8.27
TI-MPS	0.71 ± 0.07	4.02 ± 1.28	11.23 ± 0.26	16.37 ± 0.38
TI-MPS Ens. Same	0.46	2.32	10.80	15.63
TI-MPS Ens. Rand	/	1.69	/	14.04

Our best MPS ensemble models achieve comparable results to neural network and mixed (tensor + neural network) approaches on the MNIST and FashionMNIST datasets [GPC18, MPG19]. Besides, our results indicate that linear mutual information scaling with the system size might not be an excluding indicator for good classification performance, as suggested in [LKNKC21]. Namely, the accuracy of the MPS models does not drop dramatically on the permuted datasets, where we expect a volume-law scaling.

3 Deep tensor networks with matrix product operators

Although we were able to obtain good results on the simple MNIST and FashionMNIST datasets, our results on the CIFAR10 dataset are far below the best neural network approaches [DBK+20]. The latter might be due to a sub-optimal Hilbert space embedding or a small bond dimension and a thus limited amount of correlations between distant pixels. To solve these problems, we propose a non-linear transformation of the Hilbert space embedding based on the time-dependent variational principle for matrix product states [HLO+16] (MPS-TDVP). The local embedding (feature) vectors are first L2-normalized and then one or more times transformed with an MPS-TDVP determined by matrix product operators (MPOs) of fixed bond-dimension. The non-linear MPS-TDVP transformation is almost the same as in [HLO+16]. The main difference is that the calculation of the effective Hamiltonians is done for all sites (feature vectors) at the same time and not sequentially as in [HLO+16]. Parallel evaluation enables us to speed up the computation to $log(n)$ parallel time similarly to the MPS contraction. After the last MPO layer, we L1 normalize the feature vectors and classify them with an MPS layer.

The proposed deep tensor networks improve the performance with respect to the augmented MPS baseline on the MNIST dataset, especially in the translational invariant setting where the current single-model average error rate is 0.75% but with significantly less parameters than the best TI-MPS model. Further tests and evaluations are in progress.

References

[BST20] Tai-Danae Bradley, E Miles Stoudenmire, and John Terilla. Modeling sequences with quantum states: a look under the hood. *Machine Learning: Science and Technology*, 1(3):035008, 2020.

[BV20] Tai-Danae Bradley and Yiannis Vlassopoulos. Language modeling with reduced densities. *arXiv preprint arXiv:2007.03834*, 2020.

[CCW18] Song Cheng, Jing Chen, and Lei Wang. Information perspective to probabilistic modeling: Boltzmann machines versus born machines. *Entropy*, 20(8):583, 2018.

[CHLW21] Ian Convy, William Huggins, Haoran Liao, and K Birgitta Whaley. Mutual information scaling for tensor network machine learning. *arXiv preprint arXiv:2103.00105*, 2021.

[CPD21] Yiwei Chen, Yu Pan, and Daoyi Dong. Residual tensor train: a flexible and efficient approach for learning multiple multilinear correlations. *arXiv preprint arXiv:2108.08659*, 2021.

[CWXZ19] Song Cheng, Lei Wang, Tao Xiang, and Pan Zhang. Tree tensor networks for generative modeling. *Physical Review B*, 99(15):155131, 2019.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[EHL19] Stavros Efthymiou, Jack Hidary, and Stefan Leichenauer. Tensornetwork for machine learning. *arXiv preprint arXiv:1906.06329*, 2019.

[GJLP18] Chu Guo, Zhanming Jie, Wei Lu, and Dario Poletti. Matrix product operators for sequence-to-sequence learning. *Physical Review E*, 98(4):042114, 2018.

[GPC18] Ivan Glasser, Nicola Pancotti, and J Ignacio Cirac. Supervised learning with generalized tensor networks. *arXiv preprint arXiv:1806.05964*, 2018.

[HLO+16] Jutho Haegeman, Christian Lubich, Ivan Oseledets, Bart Vandereycken, and Frank Verstraete. Unifying time evolution and optimization with matrix product states. *Physical Review B*, 94(16):165116, 2016.

[KLH21] Fanjie Kong, Xiao-yang Liu, and Ricardo Henao. Quantum tensor network in machine learning: An application to tiny object classification. *arXiv preprint arXiv:2101.03154*, 2021.

[LKNKC21] Sirui Lu, Márton Kanász-Nagy, Ivan Kukuljan, and J Ignacio Cirac. Tensor networks and efficient descriptions of classical data. *arXiv preprint arXiv:2103.06872*, 2021.

[LLZZ21] Jing Liu, Sujie Li, Jiang Zhang, and Pan Zhang. Tensor networks for unsupervised machine learning. *arXiv preprint arXiv:2106.12974*, 2021.

[LRW+19] Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blázquez García, Gang Su, and Maciej Lewenstein. Machine learning by unitary tensor network of hierarchical tree structure. *New Journal of Physics*, 21(7):073059, 2019.

[MPG19] Khatereh Meshkini, Jan Platos, and Hassan Ghassemain. An analysis of convolutional neural network for fashion images classification (fashion-mnist). In *International Conference on Intelligent Information Technologies for Industry*, pages 85–95. Springer, 2019.

[MVRL20] John Martyn, Guifre Vidal, Chase Roberts, and Stefan Leichenauer. Entanglement and tensor networks for supervised image classification. *arXiv preprint arXiv:2007.06082*, 2020.

[MZZ+20] Ye-Ming Meng, Jing Zhang, Peng Zhang, Chao Gao, and Shi-Ju Ran. Residual matrix product state for machine learning. *arXiv preprint arXiv:2012.11841*, 2020.

[PV17] Vasily Pestun and Yiannis Vlassopoulos. Tensor network language model. *arXiv preprint arXiv:1710.10248*, 2017.

[SPL+20] Zheng-Zhi Sun, Cheng Peng, Ding Liu, Shi-Ju Ran, and Gang Su. Generative tensor network classification model for supervised machine learning. *Physical Review B*, 101(7):075135, 2020.

[SS16] E Miles Stoudenmire and David J Schwab. Supervised learning with quantum-inspired tensor networks. *arXiv preprint arXiv:1605.05775*, 2016.

[SSL+20] Ananda Streit, Gustavo Santos, Rosa Leão, Edmundo de Souza e Silva, Daniel Menasché, and Don Towsley. Network anomaly detection based on tensor decomposition. In *2020 Mediterranean Communication and Computer Networking Conference (MedComNet)*, pages 1–8. IEEE, 2020.

[ST19] James Stokes and John Terilla. Probabilistic modeling with matrix product states. *Entropy*, 21(12):1236, 2019.

[Sto18] E Miles Stoudenmire. Learning relevant features of data with multi-scale tensor networks. *Quantum Science and Technology*, 3(3):034003, 2018.

[WRVL20] Jinhui Wang, Chase Roberts, Guifre Vidal, and Stefan Leichenauer. Anomaly detection with tensor networks. *arXiv preprint arXiv:2006.02516*, 2020.

Pairwise Classification by Quantum Support Vector Machines with Kronecker Kernels

Taisei Nohara, Satoshi Oyama, Itsuki Noda

Hokkaido University

We investigate a possibility to apply quantum computation to pairwise classification using Kronecker kernels, and propose a way to apply HHL-based quantum support vector machine algorithm. The pairwise classification can predict relationships among data and is used for problems such as link prediction and chemical interaction prediction. However, in pairwise classification using the Kronecker kernels, it is very costly to prepare the Kronecker product of the kernel matrices when the size of the data is large. We show that the Kronecker product of kernel matrices can be represented more efficiently in time and space in quantum computation than in classical computation. We also show that we can effectively train pairwise classifier by applying the HHL-based quantum support vector machine algorithm to the Kronecker kernel matrix. In the comparison experiments between a classical algorithm and the proposed algorithm on a quantum computing simulator, the latter achieved almost the same misclassification rate as the former for the same pairwise classification problem. This indicates that the proposed algorithm can achieve equivalent accuracy with the classical algorithm more efficiently and scalably. This study paves the way for applying quantum machine learning to predicting relationships in large-scale data.

Variational Quantum approach to learn Quantum gates from Hamiltonian

Arunava Majumder [1], **Dylan Lewis** [2] and **Sougato Bose** [2]

[1] Indian Institute of Technology Kharagpur, Kharagpur 721302, India

[2] Department of Physics and Astronomy, University College London, London WC1E 6BT, United Kingdom

We present a Variational Quantum (VQA) framework to tackle the problem of finding time-independent dynamics generating target unitary evolutions. Such generators will typically contain highly non-local interactions, which can be difficult to realize in a given physical setup. The natural dynamics generating a Toffoli gate involve non-local three-qubit interactions, which are not easily implemented in experimental architectures. The method we followed is distinct from techniques such as quantum control and gate compilation and our approach can be applied in near-term devices also. For the optimization purpose, we used a very special kind of ansatz known as HVA (Hamiltonian variational ansatz) which can directly encode the parameters of a given Hamiltonian and can efficiently bypass the "barren plateaus" phenomenon. But for large circuits, the effect of "barren plateaus" is still noticeable. In the first step, we need to consider a general Hamiltonian, with the minimum number of qubit-qubit interactions (e. g. two in the case of Toffoli), which can generate the same dynamics as the original one. In the second step, we have to go through some given criteria to minimize the number of parameters or local terms in the designed Hamiltonian. Lastly, we need to perform optimization using multiple Trotterization steps such that fidelity between the natural dynamics and actual unitary remains greater than at least 0.99 to satisfy the threshold for surface code (on the order of 1%). This work has been previously done by researchers using classical supervised learning (Obtained fidelity-0.98). We have done it recently using QML and achieved greater fidelity (0.999).

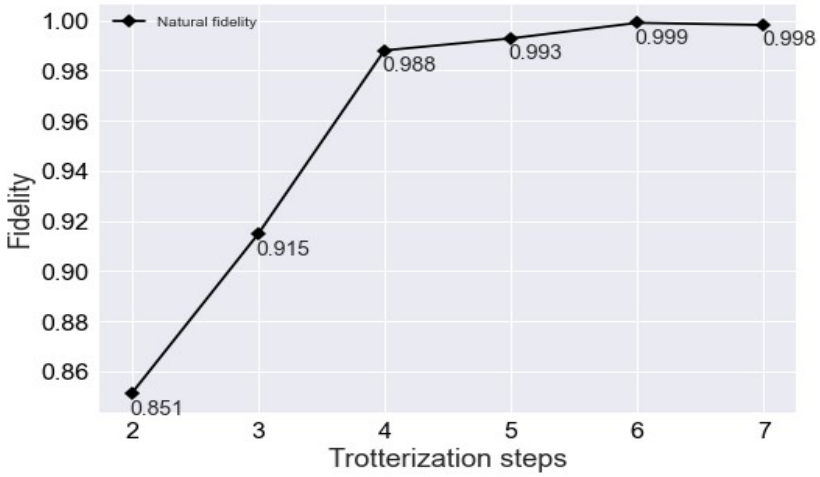
Keywords: Variational Quantum algorithms, Quantum circuits, Fidelity, HVA

Theory: The natural system dynamics of a quantum system implements the quantum operation G, with Hermitian generator H_G such that $G = \exp(-iHG)$. To avoid the highly non-local interactions we can design a subset Γ of physical interactions that can be realized relatively easily and inexpensively. Using spectral decomposition of G, then we have $H_G = -iU \log(\Lambda)U\dagger$. $\tilde{H}_G$ is defined as the generator that contains physical interactions from the subset Γ. In general $\tilde{H}_G = \tilde{H}_G(\lambda)$, λ parametrizes the set of physical interactions. The physical interactions can be a certain subset of the possible two-body and single-body interactions, like the set of coupling strengths and local magnetic fields. The following three conditions are necessary and sufficient to have the correct $\tilde{H}_G$

[ref.1]:

1. $\tilde{H}_G$ contains only physical interactions,
2. $[\tilde{H}_G, H_G] = 0$
3. $\text{Eigen}(\tilde{H}_G - H_G) = 2\pi ni$, $ni \in Z$

Results: Now using the above theory and some big algebra we can train the quantum circuit to reach a higher fidelity. The below plot shows how the fidelity increases as we increase the Trotterization steps.



So using a specific circuit and optimizer, after 6 trotterization we can achieve a high fidelity for 3-qubti toffoli gate and as stated earlier this approach can be extended to any general multi-control multi-qubit quantum gate.

[*I can give a more detail overview of the work but it would not be possible in the extended abstract. Thus, I am looking for the poster session and wish to present my work in front of all of you.*]

References:

[1] Innocenti Luca etal.: Supervised learning of time-independent Hamiltonians for gate design: https://doi.org/10.1088/1367-2630/ab8aaf

[2] Wiersema Roeland etal.: Exploring entanglement and optimization within the Hamiltonian variational ansatz: http://dx.doi.org/10.1103/PRXQuantum.1.020319

The effects of noise on quantum option pricing

Zoltán Udvarnoki* and Gábor Fáth

Department of Physics of Complex Systems, Eötvös Loránd University, Hungary

Abstract

We examined the effect of noise on financial option pricing with the Quantum Monte Carlo method using noisy simulations. The characteristics of the option payoff result in plausible price changes with noise strength and noise model type. Systematic analysis of the deviation from the expected price simultaneously with Quantum Volume estimation for the corresponding noise strength makes a Quantum Volume roadmap interpretable in terms of the option pricing error.

1 Introduction

Quantum computing is a fast-developing field with an extensive awaited impact on many fields; one of them is financial modeling [1]. Therefore predicting the time of practical quantum advantage is of great interest in many industries. Measures that quantum computer manufacturers indicate in their roadmaps are mostly limited to quantum volume, and the number of qubits [2, 3]. However, interpreting these in terms of error rates for a specific algorithm is not straightforward.

The quantum version of the Monte Carlo method promises a quadratic speed-up in query complexity. The estimation error scales with $\frac{1}{N}$ instead of $\frac{1}{\sqrt{N}}$. The quantum Monte Carlo algorithm applies first the quantum operation $\mathcal{A}$ on $n+1$ qubits:

$$\mathcal{A}\left|0\right\rangle_n\left|0\right\rangle = \sqrt{1-f(i)}\sqrt{p_i}\left|i\right\rangle_n\left|0\right\rangle + \sqrt{f(i)}\sqrt{p_i}\left|i\right\rangle_n\left|1\right\rangle$$

and then uses quantum amplitude estimation [4] to find the probability:

$$\mathrm{P}(\left|1\right\rangle_{last}) = \sum_i f(i)p(i) = \mathrm{E}(f)$$

European call options are financial derivatives that give the holder the right, but no obligation, to buy the underlying asset at a certain point in time (maturity) for an agreed price (strike). The payoff of a European call option is a piecewise linear function depending on the strike and the price of the underlying asset at maturity:

$$f(x) = \begin{cases} 0 & \text{if } x \leq x < K \\ x-K, & \text{if } x > K, \end{cases}$$

where x is the price at maturity, and K is the strike.

If we assume a price distribution for the underlying asset, the option's expected price can be calculated by Monte Carlo methods. Taking log-normal probability distribution for the price is analytically calculable using the Black-Sholes formula making European options an ideal subject for error analysis.

*zoltan.andras.udvarnoki@ttk.elte.hu

The reason why quantum computers are not useful yet is noise. There are several noise models which try to describe the noisy processes. The depolarizing noise model is popular because of its analytically tractable effects, while the thermal noise with thermal relaxation and dephasing is a physically motivated noise model.

We developed a method to connect noise models, quantum volume, and calculation error using noisy quantum simulation through the example of the Quantum Monte Carlo option pricing algorithm that is of particular interest for the financial industry.

We analyzed noise models with a combination of depolarizing noise and thermal relaxation and examined the effect of different transverse and longitudinal relaxation times and different one and two-qubit error rates.

Our work aims to offer a better understanding of the capabilities of NISQ devices and to give a future outlook of the technology given the announced roadmaps of leading quantum hardware companies.

2 Methods

We use the Qiskit package of IBM to simulate the behavior of quantum computers. The implementation of the option pricing algorithm followed [5].

To explain the effects of noise on the algorithm, we analyzed the four main parts of the algorithm separately: the distribution loading, the comparator, the linear function, and the amplitude estimation.

Assuming depolarizing noise on the maximum likelihood quantum amplitude estimation algorithm, a lower error bound can be calculated from the Fischer information matrix [6]. This analytical expression was used in combination with the error rate of circuit $\mathcal{A}$ to approximate the error of the quantum option pricing algorithm.

To give a reasonable estimate for the near-term applicability of quantum computers for a specific problem, we also utilize noisy quantum simulations. The quantum volume and error rate can be calculated simultaneously by simulating a quantum circuit with different noise strengths. Thus providing an estimate for the necessary quantum volume given some error bound.

3 Results

Our results show that the depolarizing noise model causes the estimated option price to increase. The reasons are twofold. First, the most costly and noisy part is the distribution loading, where the effect of noise is the broadening of the distribution. Second, the linear function is rescaled in a way that makes the analytical option price correspond to a probability less than 0.5. Since the depolarizing noise drives every probability towards 0.5, the option price increases with the noise strength.

The effect of thermal noise is basically the opposite, as the $|1\rangle$ states are decaying to $|0\rangle$ the estimated option price is decreasing similarly.

Acknowledgement

Prepared with the professional support of the Doctoral Student Scholarship Program of the Co-operative Doctoral Program of the Ministry of Innovation and Technology financed from the National Research, Development and Innovation Fund.

References

[1] Román Orús, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: Overview and prospects. *Reviews in Physics*, 4:100028, nov 2019.

[2] IBM. IBM Achieves Highest Quantum Volume to Date, Establishes Roadmap for Reaching Quantum Advantage, 2019.

[3] Petar Jurcevic, Ali Javadi-Abhari, Lev S. Bishop, Isaac Lauer, Daniela F. Bogorin, Markus Brink, Lauren Capelluto, Oktay Günlük, Toshinari Itoko, Naoki Kanazawa, Abhinav Kandala, George A. Keefe, Kevin Krsulich, William Landers, Eric P. Lewandowski, Douglas T. McClure, Giacomo Nannicini, Adinath Narasgond, Hasan M. Nayfeh, Emily Pritchett, Mary Beth Rothwell, Srikanth Srinivasan, Neereja Sundaresan, Cindy Wang, Ken X. Wei, Christopher J. Wood, Jeng Bang Yau, Eric J. Zhang, Oliver E. Dial, Jerry M. Chow, and Jay M. Gambetta. Demonstration of quantum volume 64 on a superconducting quantum computing system. *Quantum Science and Technology*, 6(2), 2021.

[4] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Quantum Information, Samuel J. Lomonaco, Jr. (editor), AMS Contemporary Mathematics 305, 53 (2002).*, pages 53–74, may 2002.

[5] Nikitas Stamatopoulos, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner. Option pricing using quantum computers. *Quantum*, 4, 2020.

[6] Tomoki Tanaka, Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tamiya Onodera, and Naoki Yamamoto. Amplitude estimation via maximum likelihood on noisy quantum computer. *Quantum Information Processing*, 20(9):3–14, 2021.

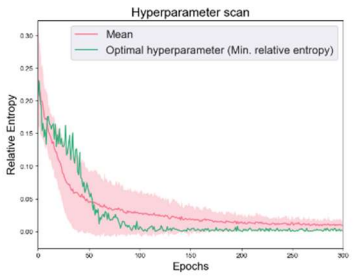
Running the Dual-PQC GAN on Noisy Simulators and Real Quantum Hardware

Su Yeon Chang[1,2], Steven Herbert[3,4], Sofia Vallecorsa[1], Elías F. Combarro[5], Ross Duncan[3,6,7], Michele Grossi[1], Edwin Agnew[3]

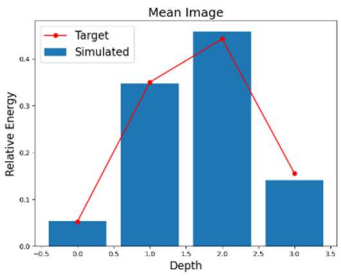
[1]OpenLab, CERN, Geneva, Switzerland
[2]Department of Physics, EPFL, Lausanne Switzerland
[3]Cambridge Quantum Computing Ltd, Cambridge, UK
[4]Department of Computer Science and Technology, University of Cambridge, Cambridge, UK
[5]Department of Computer Science, University of Oviedo, Oviedo, Spain
[6]Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK
[7]Department of Physics, University College London, London, UK

Abstract: In an earlier work [1], we introduced dual-Parameterized Quantum Circuit (PQC) GAN, an advanced prototype of quantum Generative Adversarial Networks (GAN), which consists of a classical discriminator and two quantum generators that take the form of PQCs. We have shown that the model can imitate calorimeter outputs in High-Energy Physics (HEP), interpreted as reduced size pixelated images. However, the simulation was limited to state-vector simulation in the absence of noise, even though noise due to the interaction with the environment is the major obstacle to overcome for existing quantum devices.

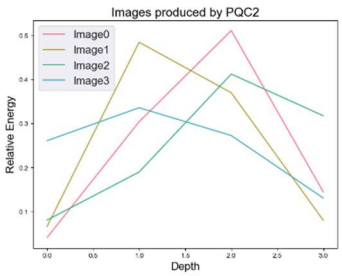
In this work, we extend the dual-PQC GAN to more practical usage by testing its performance in the presence of different types of quantum noise, including statistical fluctuations. We start by simulating noise models only with two-qubit gate errors, which are the most significant for our dual-PQC GAN architecture. In the simulation, we investigate the impact of the number of measurements and hyperparameters optimization on the performance of the dual PQC-GAN within a range of errors, covering a greater range of fidelity values than the real hardware that we will use. The results prove that we consistently get hyperparameters that reproduce the result close to the target, while more improvements are required in some areas. The hyperparameter scan also allows us to evaluate the impact of each factor on the dual-PQC training and gives us some insight as to how to tune them to improve convergence.



a) Relative entropy from a hyperparameter scan with 98% of two-qubit gate fidelity



b) Mean image at the end of the training with the optimal hyperparameters



c) Individual images reproduced by the second PQC

We then add to the noisy simulation a read-out error, which is tuned to probe the level that can be tolerated. Finally, we present the results obtained both on *superconducting* and *trapped-ion* quantum hardware with the hyperparameters that are found to be optimal in noisy simulations. The small standard deviation in the images reproduced by several inference tests on the real hardware with pre-trained parameters suggests the possibility of successful training, within a reasonable amount of time where the calibration rate is kept stable. We also propose different ansatz designs, to be efficient on different hardware architectures we use. Our work ultimately aims to provide a global overview of the effect of different types of noise in the training of dual-PQC GAN and suggest realistic solutions to lead the model to convergence.

[1] S. Y. Chang, S. Herbert, S. Vallecorsa, E. F. Combarro, R. Duncan, *Dual-Parameterized Quantum Circuit GAN Model in High Energy Physics*, arXiv:2103.15470

Estimating the degree of non-Markovianity using quantum machine learning

Hossein T. Dinani[1], Felipe Fanchini[2], Raul Coto[1]

[1] Centro de Investigación DAiTA Lab, Facultad de Estudios Interdisciplinarios, Universidad Mayor, Santiago 7560908, Chile
[2] Faculdade de Ciencias, UNESP - Universidade Estadual Paulista, Bauru, SP, 17033-360, Brazil

In many applications of machine learning in quantum systems, the system is measured and the measurement results are analyzed using classical machine learning. However, recently developed quantum machine learning models, such as variational quantum circuits, can be implemented directly on the state of the quantum system (quantum data). In this theoretical work, we propose to use variational quantum circuits (VQC) to characterize the interaction of the quantum system with its surrounding environment. We show that a VQC can classify and approximate the degree of non-Markovianity of a quantum system dynamics, with high precision, for the case of phase damping and amplitude damping.

Machine Learning for Continuous Quantum Error Correction on Superconducting Qubits

Abstract – QTML 2021

Ian Convy*, Haoran Liao*, Song Zhang, Sahil Patel, William P. Livingston, Ho Nam Nguyen, K. Birgitta Whaley

Department of Chemistry and Department of Physics, University of California, Berkeley, CA 94720, USA

An essential feature of quantum error correction (QEC) is the measurement of certain error syndrome operators, which provides information about errors on the physical qubits without collapsing the logical quantum state. In the canonical approach, quantum error correction is conducted in a discrete manner, using quantum logic gates to transfer the qubit information to ancilla qubits and subsequently making projective measurements on these to extract the error syndromes. However, in contrast to this theoretical idealization of instantaneous projections of the quantum state, experimental implementation of such measurements inherently involves performing weak measurements over finite time intervals, with the dispersive readouts in superconducting qubit architectures constituting the prime example of this in today's quantum technologies. This has motivated the development of continuous quantum error correction (CQEC), where the error syndrome operators are measured weakly in strength and continuously in time.

Previous theoretical work on CQEC has focused primarily on measurement signals that behave in an idealized manner, such that each sample is assumed to be i.i.d. Gaussian with a mean given by one of the syndrome eigenvalues. However, in real signals we observe a wide variety of "imperfections" caused by hardware limitations and post-processing effects, which can lead to more complicated syndrome dynamics or significant alterations to the noise distribution. A well-calibrated CQEC protocol should be designed to take into account any non-ideal behavior. However it is difficult to generate a precise mathematical description of the imperfections present in real measurement signals.

We propose a machine learning algorithm for CQEC that is based on the use of a recurrent neural network to identity bit-flip errors. The algorithm is designed to operate on measurement signals deviating from the ideal behavior in which the mean value corresponds to a code syndrome value and the measurement has white noise. We analyze continuous measurements taken from a superconducting architecture using three transmon qubits to identify three significant practical examples of non-ideal behavior, namely auto-correlation at temporal short lags, transient syndrome dynamics after each bit-flip, and drift in the steady-state syndrome values over the course of many experiments. Based on these real-world imperfections, we generate synthetic measurement signals from which to train the recurrent neural network, and then test its proficiency when implementing active error correction, comparing this with a traditional double threshold scheme and a discrete Bayesian classifier. The results show that our machine learning protocol is able to outperform the double threshold protocol across all tests, achieving a final state fidelity comparable to the discrete Bayesian classifier.

* These two authors contributed equally