Exploration of Quantum Machine Learning and AI Accelerators for Fusion Science

Computational Science Division
About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory’s main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY


Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: orders@ntis.gov

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: reports@osti.gov

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.
Exploration of Quantum Machine Learning and AI Accelerators for Fusion Science

prepared by
Minzhao Liu\textsuperscript{1,5}, Ge Dong\textsuperscript{4}, Kyle Gerard Felker\textsuperscript{3}, Matthew Otten\textsuperscript{6},
Prasanna Balaprakash\textsuperscript{2,3}, William Tang\textsuperscript{4}, and Yuri Alexeev\textsuperscript{1,3}
1. Computational Science Division, Argonne National Laboratory
2. Mathematics and Computer Science Division, Argonne National Laboratory
3. Leadership Computing Facility, Argonne National Laboratory
4. Princeton Plasma Physics Laboratory
5. Department of Physics, The University of Chicago
6. HRL Laboratories

October 23, 2021
Exploration of Quantum Machine Learning and AI Accelerators for Fusion Science

Minzhao Liu\textsuperscript{1,5}, Ge Dong\textsuperscript{4}, Kyle Gerard Felker\textsuperscript{3}, Matthew Otten\textsuperscript{6}, Prasanna Balaprakash\textsuperscript{2,3}, William Tang\textsuperscript{4}, and Yuri Alexeev\textsuperscript{1,3}

\textsuperscript{1}Computational Science Division, Argonne National Laboratory
\textsuperscript{2}Mathematics and Computer Science Division, Argonne National Laboratory
\textsuperscript{3}Leadership Computing Facility, Argonne National Laboratory
\textsuperscript{4}Princeton Plasma Physics Laboratory
\textsuperscript{5}Department of Physics, The University of Chicago
\textsuperscript{6}HRL Laboratories

January 14, 2022

Abstract
The dawn of the noisy intermediate-scale quantum (NISQ) era sparked rapid development in variational quantum algorithms. These algorithms, utilizing parameterized quantum circuit optimized by classical computers with feedback, are practical under the constraints of the current hardware and can potentially show quantum advantage. In recent years, these variational circuit are applied to neural networks, hoping to boost the triumphant success of deep learning. We simulate various of quantum-classical hybrid neural networks applied to scientific tasks, and seek to understand their capabilities and limitations. Specifically, we devise quantum convolutional layers and apply them to deep neural networks used for predicting plasma disruption in fusion reactors.

1 Introduction
Parameterized quantum circuits (PQCs) have been shown to possess better expressibility than their classical counterparts for generative models \cite{9}, which states that quantum models can approximate distributions better. For quantum neural networks (QNNs), the expressibility can be captured by its effective dimension, and it was shown that quantum models are advantageous in this respect as well \cite{3}. This kind of advantage is tied to the exponential size of the Hilbert space. In the setting of kernel methods, if the geometric difference between embeddings generated by classical and quantum approaches is large, one might be able to achieve quantum advantage because it is classically hard to approximate these quantum embeddings \cite{11}. Besides theoretical discussions, concrete quantum neural networks have been proposed and tested. For example, Cong \textit{et al.} \cite{7} proposed a translationally invariant quantum circuit to perform quantum convolution for quantum phase recognition on quantum many-body systems, and demonstrated detection 1D symmetry-protected topological phases with superior sample complexity and generalization.

In most QNNs that deal with classical data, a data dependent variational circuit is used to embed the data. A data independent circuit with learnable parameters processes the data and measurement on one or more qubits is the output. One the other hand, deep learning uses deep layers with nonlinearity in between, whereas data manipulation in QNNs is linear (except encoding). To mimic the classical deep learning framework more, multiple of these quantum layers are stacked together. Classical values are obtained between layers and re-encoded into quantum states. With this approach, numerous quantum-classical hybrid neural networks are developed \cite{4, 5, 6}.

In this work, we apply this scheme of quantum deep learning to prediction of plasma disruption activities in nuclear fusion reactors. Specifically, we implement quantum convolution in the spatial and temporal domain. We explore various schemes for data encoding, circuits for multichannel convolution, and study the parameter dependence of model performance for quantum and classical models. We show that although previous works using this approach demonstrated superior performance for restricted tasks (simulated and noise-free data, small problem size, etc.), claiming advantage for real world complex tasks such as plasma disruption prediction is tricky. Finally, we discuss possible directions of future research.
2 Quantum Convolutional Neural Network

Figure 1: An illustration of quantum convolution. In this picture, a window with a kernel size of $k = 3$ slides through in input. Slices are fed to the same quantum circuit independently.

Figure 1 illustrates quantum convolution. For convolution with a kernel size of $k$, a classical input array is sampled by a sliding window of size $k$. The values for each slice determine the rotation angles of the embedding circuit. A trainable parameterized quantum circuit followed the encoding circuit. Finally, measurement values are used as the output of the convolutional layer. The same quantum embedding and parameterized quantum circuit is applied to the next window of values. The outputs of the values of each window form the next layer’s input array, which goes through further quantum convolution or other types of processing.

Besides the simple single qubit rotation encoding and variational circuit ansatz used by Chen et al., [4] other approaches are also explored. For example, we implement the encoding strategy proposed by Havlíček et al., [10]. This encoding strategy utilizes products of input values as rotation angles. We also test the parameterized circuit proposed by Abbas et al., [3]. The complete circuit used in this study is shown in Fig. 1. Empirically speaking, these circuit Ansatz are harder to simulate classically compared to single qubit rotation Ansatz and have greater potential to quantum advantage [3].

Another aspect of our work deals with multichannel quantum convolution. For a classical convolutional layer with a kernel size of $k$, $n_{in}$ input channels and $n_{out}$ output channels, a window of size $k$ slides through all input channels, collecting data points of size $k \times n_{in}$ for each slice. The inputs for one slice is multiplied by a weight matrix of size $k \times n_{in}$ by $n_{out}$, giving a vector of size $n_{out}$ and yielding an output for each channel. For quantum convolution, we propose three approaches. In the first approach illustrated by Fig. 2a, each quantum circuit with $k$ qubits takes input from one channel. We use $n_{in}$ circuits with independent parameterization.
to evaluate independent outputs. These outputs are then combined classically to produce a single output for a single output channel. The independent such circuit collections are used to produce multichannel outputs. In the second approach illustrated by Fig. 2b, we concatenate the classical inputs and feed them into a single circuit with $k \times n_{in}$ qubits. $n_{out}$ independently parameterized circuits are used to evaluate multichannel outputs. This schemes leads to a much larger Hilbert space size, hence more promising. In the third approach illustrated by Fig. 2c, we concatenate the classical inputs, but use amplitude encoding instead of single qubit rotation encoding. The circuit size is of $O(\log(k \times n_{in}))$. Since the Hilbert space size is linear to the classical input size instead of exponential, we add a ancilla qubits to increase the Hilber space size. Thereafter, the same kind of $n_{out}$ independently parameterized circuits are used to produce multichannel outputs.

3 Application to Nuclear Fusion Reactor Plasma Disruption Prediction

Deep neural networks have been applied to predict disruptive plasma activities in nuclear fusion reactors. The goal of such predictions is to enable interventions that mitigate or avoid the detrimental effects of such disruptions on the fusion reactors. The recent state-of-the-art architecture utilizes scalar inputs, including plasma current and internal inductance, as well as 1D electron temperature and density profiles [12, 8]. For each time step, the 1D profiles are processed using multichannel classical convolution, followed by concatenation with the scalar signals. This is followed by a network that processes the temporal relationship. The earliest work uses an long short-term memory (LSTM) architecture to propagate information from the past in order to inform the prediction [12]. A more recent iteration uses temporal convolutional networks (TCN) to again a much larger memory capacity compared to recurrent models [8].

Figure 3: An illustration of our quantum-classical hybrid neural network.

The above summary points out two ways we can insert quantum convolution: replacing spatial convolution on the 1D profiles with quantum convolution and replacing temporal convolution with quantum temporal convolution. Our hybrid model is illustrated in Fig. 3.

4 Results

4.1 Implementation

All quantum circuit simulation is achieved using native PyTorch tensor operations, which is fully differentiable and is compatible with backpropagation. Simulated quantum-classical hybrid model training is performed on the Tiger cluster at Princeton Research Computing [2]. This work primarily uses the GPU part of the cluster, which is a Dell computer cluster comprised of 320 NVIDIA P100 GPUs across 80 Broadwell nodes, each GPU processor core has 16 GB of memory. The nodes are interconnected by an Intel Omnipath fabric. The CPUs are Intel Broadwell e5-2680v4 with 28 cores per node. The largest quantum models are trained on four GPUs on a single node using PyTorch DataParallel container. The code for this project can be found at our github repository [1].
4.2 Quantum Spatial Convolution

![Quantum Spatial Convolution Graph](image)

Figure 4: Comparison of quantum and classical convolution. ROC is a metric for the prediction quality of the model, ranging from 0 to 1. Higher is better. (a) Networks utilizing both scalar signals and convolved 1D signals. (b) Networks utilizing only convolved 1D signals.

We first test multichannel quantum convolution in the first approach with single qubit rotation embedding. The performance of our quantum model is inferior than that of a comparable classical model. We therefore will not discuss single qubit rotation embedding and the first multichannel quantum convolution scheme from here on. We test the second kind of multichannel quantum convolution with the embedding scheme shown in Fig. 1. Additionally, we add another layer of CNOT gates and a parameterized measurement qubit $XYZ$ rotation with three parameters at the end. Here, we use two quantum convolutional layers with two input and output channels each. The kernel size is $k = 3$. This corresponds to having 60 quantum convolutional parameters. For a comparable classical model, we choose the first classical convolutional layer to have 2 input channels and 4 output channels, and we choose the second layer to have 4 input channels and 3 output channels. This corresponds to 60 classical convolutional parameters. As illustrated in Fig. 4a, our quantum model slightly outperforms its classical counterpart. In the hope of increasing the performance difference between the quantum model and the classical model, we perform tests using only the 1D profiles as input signals. Figure 4b shows that the tested quantum model outperforms the classical model slightly, but we fail to widen the performance gap compared to the previous approach. In this test, the classical test is optimized for a range of temporal convolutional network hyperparameters. Namely, we perform grid search with 2, 4, 6, and 8 temporal convolutional layers, 2, 4, 6, and 8 temporal convolutional output channels, and a temporal kernel size of 5. The best performing classical model has 6 temporal convolutional layers, and 2 temporal convolutional channels.

4.3 Quantum Temporal Convolution

![Quantum Temporal Convolution Graph](image)

Figure 5: Comparison of quantum and classical temporal convolution. The quantum models here all use two ancilla qubits.

The state-of-the-art classical TCN employs a very large number of channels and a large filter size. Running
multichannel quantum convolution in this regime requires too many qubits to be simulated. Therefore, we opt for the third type of multichannel quantum convolution for temporal convolution. The larger number of layers, channels and kernel sizes that amplitude encoding allows us to achieve means that we can simulate models with more quantum parameters. We exploit this and perform a performance vs number of parameters analysis for both quantum and classical models, and the results are shown in Fig. 5.

For the quantum models, we perform a grid search with 2, 4, 6, and 8 quantum temporal convolutional layers, 2, 4, and 8 quantum temporal convolutional channels, and kernel sizes of 4 and 8. For the classical models, we also search over larger numbers of layers and channels.

The largest quantum model we can simulate has 2432 temporal convolutional parameters, bounded by our computational resources. The best performing quantum model has 816 temporal convolutional parameters with an ROC of 0.7895. To find a better performing classical model, we need to go to more than 10000 parameters. The salmon shaded region in Fig. 5 shows the parameter region where the quantum model we found is superior. The blue shaded region is where classical models are better and quantum models are untested. However, the performance of both quantum and classical models is very noisy in the few parameter regime, contrasting the uniformly good performance of larger classical models in the blue shaded region. It is therefore hard to conclude whether or not quantum models are truly advantageous.

![Figure 6: Comparison of different numbers of ancilla qubits.](image)

Additionally, since the size of the Hilbert space depends on the number of ancilla qubits used, we study the dependence of model performance as we change the number of ancilla qubits. Figure 6 shows the performance of models with 1, 2 and 3 ancilla qubits vs the total number of temporal convolutional parameters. Overall, the differences between different numbers of ancilla qubits is not significant compared to the noise in the model performance.

5 Fusion Deep Neural Network on AI Accelerator

As part of the Laboratory Directed Research and Development (LDRD) expedition project, we also implement the classical TCN based fusion neural network on the SambaNova AI accelerator at Argonne National Laboratory. We build a proof-of-principle model with the accelerator API and demonstrated that it is possible to test our model architecture on this novel platform.

6 Conclusion

We explored various quantum-classical hybrid architectures based on the fusion neural network and found that these architectures have limitations on challenging scientific tasks. We also implemented the model on the AI accelerators from SambaNova. A lot of future work remains for us to effectively exploit the immense opportunities offered by novel computational platforms such as quantum computers and AI accelerators.

References


Computational Science Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 240
Argonne, IL 60439

www.anl.gov