Evalutaing the Performance of Hybrid Quantum-Classical Convolutional Neural Networks on NISQ Devices

Author: Tristan K. Nagan (1484020)
Supervisor: Dr. Ritesh Ajoodha
School of Computer Science and Applied Mathematics
The University of the Witwatersrand, Johannesburg

Declaration
2020

I, ***Tristan Kyde Nagan***, (Student number: ***1484020***) am a student registered for Introduction to Research Methods in 2020.

This declaration applies to the ***Research Report*** document of Introduction to Research Methods.

I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that the work submitted for assessment is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand, Johannesburg may take disciplinary action against me if there is a belief that this is not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Signature: ______________________________ Date: ______________________________

Evalutaing the Performance of Hybrid Quantum-Classical Convolutional Neural Networks on NISQ Devices

Tristan Kyde Nagan
School of Computer Science and Applied Mathematics
The University of the Witwatersrand
Johannesburg, South Africa
Email: 1484020@students.wits.ac.za

Ritesh Ajoodha
School of Computer Science and Applied Mathematics
The University of the Witwatersrand
Johannesburg, South Africa
Email: ritesh.ajoodha@wits.ac.za

***Abstract*—The increase in quantum algorithm development in recent years has spurred the development of new algorithms which outsource classically intractable portions of an algorithm to a quantum computer. These hybrid quantum-classical algorithms must be thoroughly studied and analyzed to determine their viability. One such algorithm is the hybrid quantum-classical convolutional neural network.**

In this paper we compared different implementations of hybrid quantum-classical convolutional neural networks and their fully classical counterparts. In particular, we compared their accuracy and loss performance over time. To do this, we implemented both the classical and hybrid quantum-classical algorithms with varying convolutional layers and ran them on the MSL cluster and IBM's cloud quantum computers using the low-dimensional Tetris dataset. We found that hybrid quantum-classical convolutional neural networks do not outperform their classical counterparts in terms of accuracy and loss over time when tested on current-state quantum computers. This could be attributed to the noise present in current quantum computers affecting the results of the circuit.

***Index Terms*—Machine Learning, Quantum Computing, Convolutional Neural Networks.**

I. Introduction

The recent improvements in quantum computing hardware have sparked a surge of funding into the research and development of quantum algorithms. The eventual goal of these algorithms is to one day be used to solve real-world problems in chemistry, finance, and machine learning. The problem is that we do not know how different variations of these algorithms perform when compared to each other and their classical counterparts in terms of accuracy and loss over time. So, the question we must ask is what is the difference in accuracy and loss over time between hybrid quantum-classical convolutional neural networks, and a classical convolutional neural network?

In this paper we perform an empirical analysis of hybrid quantum-classical convolutional neural networks on current quantum computing hardware and compare them to their fully classical counterparts. The goal is to determine whether these algorithms show any potential of outperforming their classical counterparts (regarding accuracy and loss mentioned earlier), and if so, under what conditions.

The first proposed approaches to hybrid quantum convolutional neural networks [1], [2] implemented only the convolutional layer on a quantum computer. The rest of the network is implemented on a classical computer. Data passes from the input layer on the classical computer to a quantum computer where the convolution takes place. Once the convolution has completed the data is then passed from the quantum computer back to the classical computer for pooling. The data can then pass back to the quantum computer to perform the convolution operation again, or the data can continue to the fully connected layer or a classical convolutional layer.

They proposed a convolutional layer made of variational quantum circuit. This circuit encodes the input data onto several qubits and then performs either a random quantum circuit [1] or a structured circuit [2] on them before taking a measurement. Both approaches are implementable on current quantum hardware and are scalable.

This paper focuses on the random and structured circuit approaches for quantum convolution. We compare the accuracy and loss performance of these approaches against each other and a classical convolutional neural network. We implemented the classical and hybrid quantum-classical algorithms and ran them on the Wits University MSL Cluster and the IBM cloud quantum computers and test them on a low-dimensional synthetic dataset known as the Tetris dataset.

We found that hybrid quantum-classical convolutional neural networks do not out-perform their purely classical counterparts when tested on current-state quantum computers. This conflicts with the results obtained in previous research which showed hybrid quantum-classical convolutional neural networks out-performing their classical counter-parts. This can be attributed to previous work using ideal quantum simulators and training and testing their models in these ideal quantum simulators instead of current-state quantum computers.

This research will make the following contributions: (a) an analysis of the accuracy and loss over time for hybrid quantum-classical convolutional neural networks with a varying number of convolutional layers and varying number of labels in the training and testing data, (b) an analysis of the structure of these quantum convolutional layers which will determine why a structure or random quantum circuit outperforms the other, (c) an explanation of the use of a quantum circuit as a generalized feature map and how it may extract different high-level spatial features then a classical convolutional layer, (d) an analysis which highlights the flaws in these algorithms, if any, and hopefully leads to the development of better algorithms which address these flaws.

Having introduced the problem area and briefly outlined the literature, methodology and results, the rest of this work will provide a more detailed discussion on each of these topics. In section 2 will provide context to this research by giving a more in-depth discussion on the background and related literature within this field. In the background we will cover current quantum hardware, hybrid quantum-classical algorithms, and quantum-enhanced machine learning. In the related work we will cover quantum convolutional neural networks, data, and metrics. Section 3 covers the methodology used in this research. We will outline the data used in this work, the models and their architecture, the metrics used to compare performance and the platforms and environments used to train and test our models. Finally, we will present our results along with a discussion and conclusion.

II. Background and Related Work

We will now provide context to this work by briefly going into the background and related work available within this field. The background will give a brief introduction into hybrid quantum-classical algorithms and quantum-enhanced machine learning, which is the focus of this work. This leads to the related work section, which summarizes the existing work in quantum convolutional neural networks. We will then outline the data, and environments used to test these algorithms in the past, and the performance metrics used to evaluate these algorithms.

A. Background

1) Hybrid Quantum-Classical Algorithms: Quantum computers exist as noisy intermediate-scale quantum (NISQ) devices. This means that the quantum circuits they run are susceptible to incurring noise from their environment. Quantum circuits comprise quantum bits or "qubits" and quantum gates. Qubits are the quantum analogue of classical bits and are the basic unit of quantum information [3]. Quantum gates are, in some ways, the quantum analogue of classical gates in that they are the building blocks of quantum circuits, just as classical gates are the building blocks of classical circuits. These quantum gates, however, differ from classical gates. They perform operations on one or more qubits represented as unitary matrices.

The width (the number of qubits) and the depth (the number of gates) of the circuit also influence how much noise the system will incur. The more qubits, and gates a quantum circuit has, the more noise it will incur which will ultimately break down the quantum system and the circuit will fail to run. While improvements in noise-cancelling on a hardware side have offered some help, work on noise modelling on a software side has also been effective in combatting noise [4]. These improvements are steps towards so-called "fault-tolerant" quantum computers, but NISQ devices are still very limited in their capabilities.

Despite their limitations as stand-alone computers, NISQ devices can be used in conjunction with classical computers to form hybrid quantum-classical algorithms. The idea here is to outsource portions of a classical algorithm which either can benefit from quantum phenomena or is classically intractable or both to a quantum computer.

Algorithms such as the quantum approximate optimization algorithm [5] and the variational quantum eigensolver algorithm [6] were born from this school of thought have proved to be successful and scalable. Researchers in the fields of chemistry [7], finance [8] and machine learning [9], [10], [11] are already developing hybrid quantum-classical algorithms.

2) Quantum-Enhanced Machine Learning: Hybrid quantum-classical algorithms have provided an avenue for NISQ devices to assist classical computers in various fields, including machine learning. By utilizing shallow circuits known as parameterized quantum circuits (PQC), as part of an otherwise classical algorithm, we may provide improvements to our classical algorithm. PQCs can encode data onto qubits to perform computations in higher dimensional space [12], as variational circuits that can act as learning models [13], or as an optimizing function for variational circuits [14].

The wide range of uses of PQCs can provide improvements in computational complexity [15], [16], [17], sample complexity [18] and even supervised computational learning [19] which ultimately leads to improvements in runtime and accuracy levels. To identify tasks within a classical algorithm that may benefit from quantum phenomena, [11] developed a paradigm for creating quantum-enhanced machine learning algorithms.

B. Related Work

1) Quantum Convolutional Neural Networks: The first quantum algorithm to adapt the classical convolutional neural network (CNN) was developed to solve quantum many-body problems which are complex systems that are too difficult to solve theoretically [20]. Past attempts to solve these systems involved using classical machine learning algorithms. This provided mixed levels of success thus [20] proposed this algorithm to address the shortcomings of past approaches by taking the convolutional and pooling layers from a classical CNN and implementing them in the quantum space to reduce the complexities of these problems. This algorithm has two main problems: first, it has no image processing uses like a classical CNN; and second, to solve a problem of significant complexity it requires more qubits than are feasible on current quantum hardware.

The second approach to quantum CNNs was developed with image processing in mind [21]. This algorithm is a complete translation of the classical CNN into the quantum realm with convolutional layers, pooling layers, an activation function and a fully connected layer with backpropagation. This approach has two main problems: first, is the need for quantum RAM (QRAM) which has no reliable implementation; and second, the number of qubits needed to implement this algorithm efficiently on current hardware being too high.

The authors did, however, run a numerical simulation of their algorithm against a classical CNN of similar architecture and found that in terms of speed the quantum CNN could run significantly faster than its classical counterpart and provide a similar accuracy. These simulations show that a quantum CNN has the potential to outperform a classical CNN but until quantum hardware improves enough for these algorithms to be efficiently implemented and tested on real data we cannot definitively say that the quantum algorithm is better.

To address the problems of scalability for current quantum hardware and bypass the need for QRAM, researchers turned to hybrid approaches using variational quantum circuits to replace only the convolutional layer of an otherwise classical CNN. Thus the first hybrid algorithm was proposed [1]. The authors outline a quantum analogue for the classical convolutional layer which they call a "quanvolutional" layer. The purpose of this layer is the same as the classical convolutional layer because it extracts high-level spatial-features from the input image. The layer consists of a variational quantum circuit (VQC) which encodes window data on $n \times n$ qubits (where n represents the window size), performing a random quantum circuit on these qubits and then measuring them to produce a matrix.

The simplest use of this layer is replacing the convolutional layer with the quantum analogue in a CNN composed of a single convolutional layer. We can extend this use by replacing 1 out of multiple convolutional layers with this quanvolutional layer, replacing multiple layers with quanvolutional layers in varying orders or replacing the convolutional layers entirely with quanvolutional layers. The latter options put considerable strain on current hardware, and an increase in layer numbers and layer complexity may hit a limitation in hardware. This, however, is still more practically feasible then a full quantum algorithm.

When simulated with multiple layers the quantum CNN showed the behaviour of a classical CNN, with adding more layers and more training iterations resulting in a higher accuracy (95% and higher) until it reaches a level of saturation and the rate of improvement levels off to a constant. When compared to its classical counterpart, the quantum CNN showed an improvement in both accuracy and loss even in architectures that replaced just one layer with a quantum analogue. The improvements were not significant and thus did not prove a clear quantum advantage for this task over classical models. It does, however, show that quantum feature processing can aid classical computers even in a limited capacity to detect features which classical computers cannot.

The second hybrid algorithm proposed [2] is fairly similar to the first one. The key difference between them is that latter focused exclusively on a structured circuit for their VQC.

When simulated, and compared to its classical counterparts the quantum CNN consistently outperformed the classical CNN across several layer architectures with the single-layer quantum CNN outperforming a single layer CNN by almost 60% in accuracy and 25% in loss for five label classification and even outperformed a two-layer CNN by 20% in accuracy and 10% in loss for five label classification. This may show that a structured quantum circuit may provide an advantage over a random quantum circuit. This contradicts the initial findings of [1], who found that there is no difference in performance between a random and structured quantum circuit.

2) Data and Testing: Due to the limitations of quantum hardware discussed in previous sections, the datasets used to test these algorithms in the past have been small and extremely low dimensional. The largest dataset used was by [21], who used the MNIST dataset when they tested their algorithm in a simulated environment. In contrast, [2] used a low dimensional synthesized dataset known as the Tetris dataset but also tested their algorithm in a simulated environment.

3) Comparison and Metrics: We measure the performance of neural network performance by the change in accuracy and loss over iterations and architectures. The general behaviour is that the accuracy starts low and steadily increases over iterations as the networks undergo more and more training, while the loss starts high and steadily decreases. This continues until the network has reached the limits of its learning potential, at which point the accuracy and loss will level off to a constant. This metric was used in all prior works[2], [1], [21], [22] to measure the performance of their neural networks.

[2] compared single-layer and two-layer quantum CNNs against single-layer and two-layer classical CNN, however, they did not test an architecture greater than two nor the effects of having alternating quantum and classical layers in one network architecture. This would allow us to see what effects quantum layers have when used in conjunction with classical layers as part of a larger system of layers.

[1] proposed several questions left unanswered by their paper as they were out of the scope of the paper but are still questions which need answers if we are to create viable quantum CNNs. These questions include, what role does different encoding, and decoding play and what structured circuits provide advantages?

III. Methodology

We will now cover the methodology used in this work. We will start by going over the data used to train and test our models, then we will cover the structure of the models and the quantum circuits. Following this we will cover the metrics used to evaluate the performance of our models and finally we will cover the version design platforms and environments we used to code and run our models.

A. Data

The data used in this research is a synthesized Tetris dataset. It comprises 3×3 grayscale images resembling the blocks in the popular arcade game Tetris. There are five basic shapes which can be in a variety of orientations. We chose this dataset for its low dimensionality. A bonus of using this synthesized dataset is that we can synthesize the amount of data we need. The goal of the convolutional neural network is to classify an image as one of the five (or two) shapes regardless of its orientation.

B. Models

The classical and hybrid quantum-classical models share the same basic network structure as seen in figure 1.

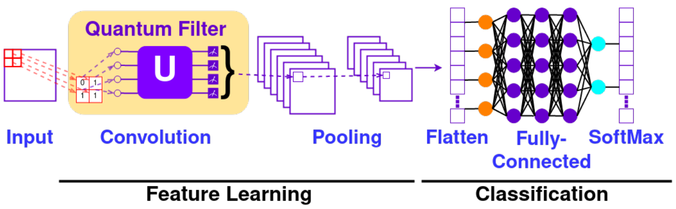

Fig. 1: An overview of the convolutional neural network structure.

The 2×2 window in the image is passed to the convolutional layer (in figure 1 the layer is quantum but it can also be classical) which is then passed to the pooling layer. The convolutional and pooling layers can be repeated. The output of the pooling layer is then flattened and passed to the fully-connected layer and then the Soft Max is taken to determine the model prediction.

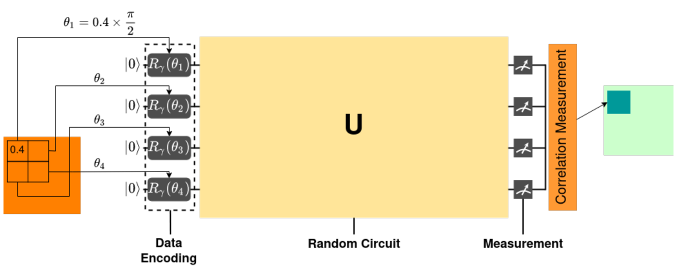

Fig. 2: The random quantum convolutional layer.

The Random convolutional layer is shown in figure 2. The 2×2 window in the image is encoded onto the qubits using an R_y gate. This is then passed to the random circuit and then measured. This result is then sent to the pooling layer and the pipeline continues as shown in figure 1.

The Structured convolutional layer is shown in figure 3. The 2×2 window in the image is encoded onto the qubits using an R_y gate similar to the random quantum convolutional layer. This is then passed to the first layer of single-qubit gates (which is another R_y gate) and then passed to the first layer of two-qubit gates (which are interlaced CNOT gates). This structured of single-qubit followed by two-qubit gates is repeated N times (where N is $n \times n$ and n is the number of qubits).

In [1] the weights of the random circuit were not trained. Instead the random circuit is just used as a non-linear transformation of the input data. In this work we will be training the weights of the random circuit to see if this affects the accuracy of the model.

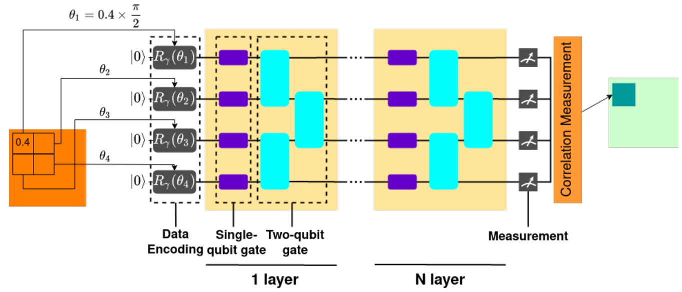

Fig. 3: The structured quantum convolutional layer.

C. Metrics

To keep testing consistent with previous studies, we measured the change in accuracy and loss of our convolutional neural networks.

D. Design Platform and Environments

The classical and hybrid quantum-classical convolutional neural network algorithms were implemented in Python using Pytorch for the Deep Learning pipeline and IBM's open-source framework for quantum algorithm development, Qiskit. We integrated them using Pennylane, a cross-platform python library for quantum machine learning.

The models will be trained on a classical computer through a Quantum Simulator. This is done to reduce training time as each convolution operation is done sequentially and could sit on a queue for up to 15 minutes before being run on an IBM quantum computer. For each image there are 4 convolutions that must be done and in the training dataset there are 804 images for 5 label classification. This means there are 3216 convolution operations that must be done which could each take 15 minutes in the worst case. This paired with calculating the derivative of the weights of the quantum circuit during backpropagation increase the training time significantly. The models were tested on IBM's 5 qubit Ourense Quantum Computer via the IBM Quantum Experience. To ensure that the simulator used to train the models is as close to a true quantum computer we applied a noise model to the simulator. The noise model was taken from IBM's 5 qubit Ourense Quantum Computer so that our training and testing environments would be as similar as possible.

IV. Results and Discussion

The results for 2 label and 5 label classification in terms of accuracy and loss on the training and testing data over time are shown in 4. From the results we see that the Classical CNN consistently outperformed both the structured and random QCCNNs, conflicting with the results obtained in previous work [1], [2]. This discrepancy between the results obtained in this work and the results obtained in previous work may be due to two factors:

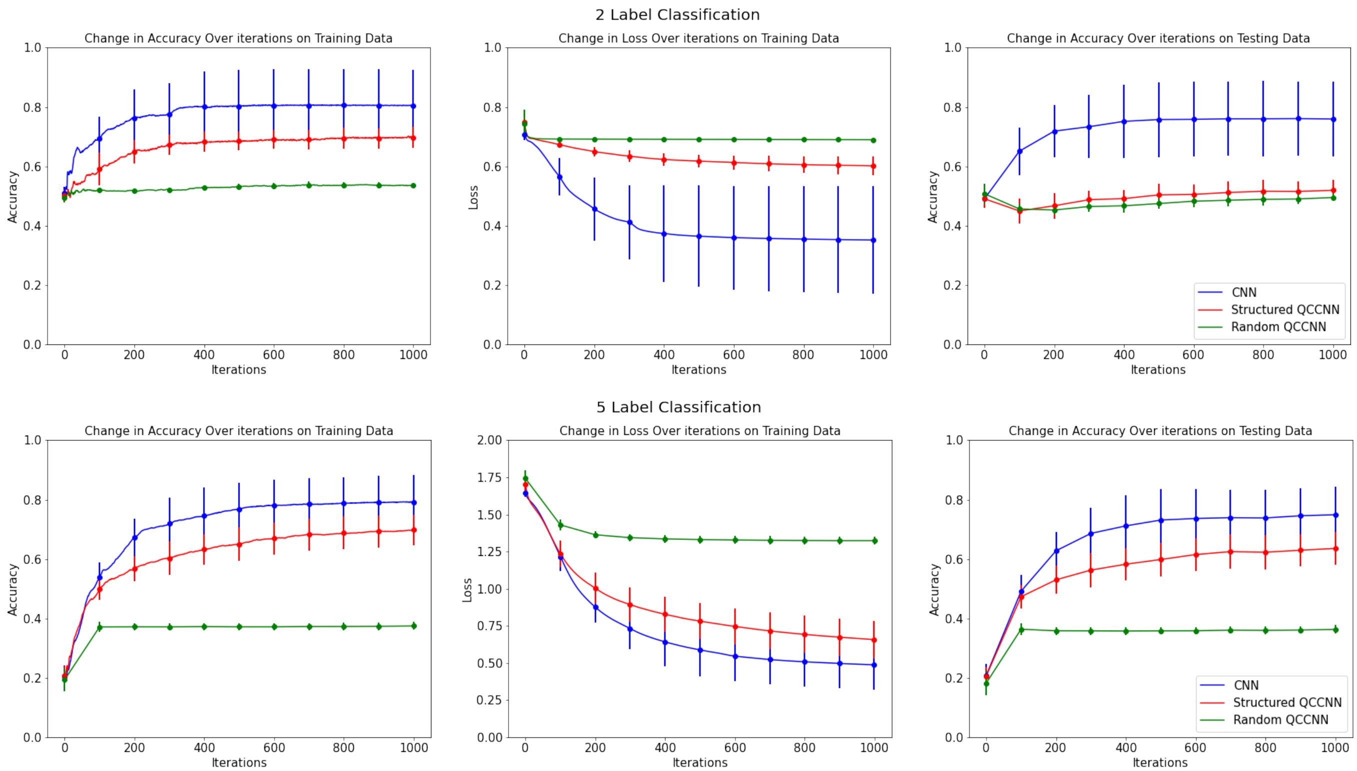

Fig. 4: Results for 2 label and 5 label classification in terms of accuracy and loss on training and testing data over time.

- The simulators used in previous works may have been analogues for perfect quantum computers (i.e. quantum computers without noise). An effort was made in this work to make sure that the simulator used to train the models was as close to a true current-state quantum computer as possible.
- The results obtained in previous works may have also been obtained by both training and testing the models in a quantum simulator.

The results of previous work may be considered the results that can be obtained in an ideal situation where quantum computers are completely devoid of noise. The results obtained in this work are a reflection of how noise in current-state quantum computers can negatively impact results. The results obtained in this work do confirm that the structured QCCNN does indeed out-perform the random QCCNN. This could be a result of making the weights of the random circuit trainable as number of trainable weights within the circuit may not be enough and as a result is underfitting the data. Increasing the number of trainable parameters in both the random and structured quantum circuits may improve the accuracy of these models.

V. Conclusion

In this paper we compared different implementations of hybrid quantum-classical convolutional neural networks and their fully classical counterparts. In particular, we compared their accuracy and loss performance over time. Our results directly conflicted with the results found in previous work outlining these algorithms. We determined that this discrepancy can be attributed to the used of ideal quantum simulators and training and testing their models in these ideal quantum simulators instead of current-state quantum computers in previous work. In this work we opted to train our models in quantum simulators with noise models in place to better simulate the conditions found in a true quantum computer where these models would be tested. Our results show the how noise can impact results and skew them away from the results found in ideal situations. Implementing a quantum convolution circuit that is robust against noise may be a possible avenue for future work.

References

[1] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quanvolutional neural networks: powering image recognition with quantum circuits," *Quantum Machine Intelligence*, vol. 2, pp. 1–9, 06 2020. [Online]. Available: https://link.springer.com/article/10.1007%2Fs42484-020-00012-y

[2] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, "Hybrid quantum-classical convolutional neural networks," *arXiv preprint arXiv:1911.02998*, 2019. [Online]. Available: https://arxiv.org/abs/1911.02998

[3] G. Brassard, I. Chuang, S. Lloyd, and C. Monroe, "Quantum computing," *Proceedings of the National Academy of Sciences*, vol. 95, no. 19, pp. 11 032–11 033, 1998. [Online]. Available: https://www.pnas.org/content/95/19/11032

[4] T. A. Brun and D. A. Lidar, *Quantum Error Correction*. Cambridge University Press, 2013.

[5] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014. [Online]. Available: https://arxiv.org/pdf/1411.4028.pdf

[6] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023023, Feb 2016. [Online]. Available: https://iopscience.iop.org/article/10.1088/1367-2630/18/2/023023/meta

[7] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, no. 1, Jul 2014. [Online]. Available: https://www.nature.com/articles/ncomms5213

[8] F. Fontanela, A. Jacquier, and M. Oumgari, "A quantum algorithm for linear pdes arising in finance," *SSRN Electronic Journal*, Jan 2019. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3499134

[9] S. Sharma, "Qeml (quantum enhanced machine learning): Using quantum computing to enhance ml classifiers and feature spaces," *arXiv preprint arXiv:2002.10453*, 2020. [Online]. Available: https://arxiv.org/pdf/2002.10453.pdf

[10] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, D. N., E. Schuyler Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, R. S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, B. R. Johnson, M. Reagor, M. P. da Silva, and C. Rigetti, "Unsupervised machine learning on a hybrid quantum computer," *arXiv preprint arXiv:1712.05771*, 2017. [Online]. Available: https://arxiv.org/pdf/1712.05771.pdf

[11] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 043001, 2019. [Online]. Available: https://iopscience.iop.org/article/10.1088/2058-9565/ab4eb5

[12] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Phys. Rev. Lett.*, vol. 122, no. 4, p. 040504, 2019. [Online]. Available: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.122.040504

[13] C. M. Wilson, J. S. Otterbach, N. Tezak, R. S. Smith, P. M., P. J. Karalekas, S. Heidel, M. Sohaib Alam, G. E. Crooks, and M. P. da Silva, "Quantum kitchen sinks: An algorithm for machine learning on near-term quantum computers," *arXiv preprint arXiv:1806.08321*, 2018. [Online]. Available: https://arxiv.org/pdf/1806.08321.pdf

[14] K. M. Nakanishi, K. Fujii, and S. Todo, "Sequential minimal optimization for quantum-classical hybrid algorithms," *arXiv preprint arXiv:1903.12166*, 2019. [Online]. Available: https://arxiv.org/pdf/1903.12166.pdf

[15] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical Review Letters*, vol. 113, no. 13, Sep 2014. [Online]. Available: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.113.130503

[16] E. Aïmeur, G. Brassard, and S. Gambs, "Quantum speed-up for unsupervised learning," *Machine Learning*, vol. 90, no. 2, Feb 2013. [Online]. Available: https://link.springer.com/article/10.1007/s10994-012-5316-5

[17] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, "Quantum speedup for active learning agents," *Phys. Rev. X*, vol. 4, no. 3, p. 031002, Jul 2014. [Online]. Available: https://journals.aps.org/prx/abstract/10.1103/PhysRevX.4.031002

[18] S. Arunachalam and R. de Wolf, "Optimal quantum sample complexity of learning algorithms," in *Proceedings of the 32nd Computational Complexity Conference*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. [Online]. Available: http://www.jmlr.org/papers/volume19/18-195/18-195.pdf

[19] A. Atici, "Advances in quantum computational learning theory," Ph.D. dissertation, Columbia University, 2006. [Online]. Available: http://www.alpatici.net/dissertation.pdf

[20] I. Cong, S. Choi, and M. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019. [Online]. Available: https://www.nature.com/articles/s41567-019-0648-8

[21] I. Kerenidis, J. Landman, and A. Prakash, "Quantum algorithms for deep convolutional neural networks," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=Hygab1rKDS

[22] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on mnist classification task," *arXiv preprint arXiv:1804.02763*, 2018. [Online]. Available: https://arxiv.org/abs/1804.02763