Draft version December 16, 2020
Typeset using LaTeX **twocolumn** style in AASTeX63

Quantum Machine Learning

[1]

[1] *University of Washington, Department of Comprehensive Physics*

ABSTRACT

Quantum machine learning is a rapidly developing and very promising technology that will allow for large datasets to be able to train quickly. It also allows us to train on quantum data and recognize patterns that may not otherwise be recognizable. This is done by using a variety of quantum computational methods such as qSVM, qPCA, qBLAS and many more. There are still several challenges that need to be solved but the future of quantum machine learning looks promising and seems to be what will eventually become common-place.

Keywords: quantum computing, machine learning, quantum machine learning, machine learning, quantum circuits

1. INTRODUCTION

Our rapid developments in technology has fueled the ubiquity of Machine Learning. We are collecting more data than ever, and machines are able to find patterns in data sets very rapidly. This technology is used in everything from predicting stock markets, to improving Netflix recommendations, to track reconstruction at the LHC. Another technology that has seen a lot of success and has exponential promise is quantum computing. Quantum computing takes advantage of the quantum mechanical phenomena of superposition and entanglement. It allows us to do computation on multiple states at once. In 2019, Google achieved quantum supremacy, this is the idea that quantum devices can solve problems that classical computers cannot in a reasonable time frame.

Quantum machine learning (QML) is the integration of both of these technologies to be able to train larger and more complex algorithms, to better extract patterns from data sets, and speed up computing so we can solve more difficult problems. QML has huge potential and it is believed that this may be a step towards AGI (artificial general intelligence). Currently the computational cost of classical machine learning is a looming roadblock ahead in learning on datasets that are growing ever larger. The polynomial cost of computation makes it so we have to spend more time preparing, shaping, and making the data sparse, than we do actually training it. Quantum machine learning will provide the necessary computation speedup to allow for the largest of datasets to be trained without worry about computation cost.

2. MACHINE LEARNING ALGORITHMS

There are two major types of machine learning, supervised learning, and unsupervised learning. In supervised learning, we know how many labels there are for classification. In unsupervised learning we don't know the number of labels. Deep learning builds upon these methods and adds extra layers to create a neural network which can recognize patterns.

To apply any of these methods using quantum algorithms we first have to convert classical data into quantum data. Given a data set Ω with d input features

$$\overrightarrow{x_i} \in \mathbb{R}^d \tag{1}$$

We can map the data into a quantum state with amplitudes that are equal to the features for a data point. And this is done for the entire data set

$$|\Phi(\overrightarrow{x_i})\rangle \langle\Phi(\overrightarrow{x_i})| \tag{2}$$

This data is then stored in qRAM. This process of converting classical data to quantum data is one of the key factors that causes speedup. A data point with d features requires $O(d)$ bits of precision while a quantum data point requires $O(\log_2(d))$ qubits for precision. This exponential compression is significant in the speedup.

Under the hood of all machine learning algorithms als lies FFT (fast Fourier transform), finding eigenvalues and eigenvectors with Gaussian elimination, and matrix inversion. Classically, these algorithms take $O(d\log(d))$, $O(d^3)$, $O(d\log(d))$ time respectively. The quantum version of these algorithms take $O(\log(d)^2)$, $O(\log(d)^2)$, $O(\log(d)^3)$. These basic linear algebra subroutines, BLAS, are exponentially faster on a quantum

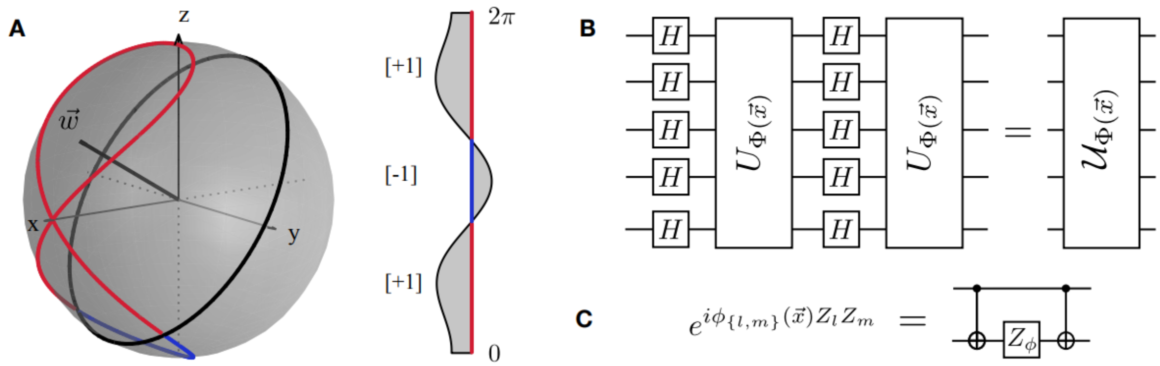

Figure 1. Quantum Kernel Functions: (A) Feature map of a single qubit. A classical dataset can be mapped onto the Bloch sphere using (b). The mapped data can be separated by a hyperplane $\vec{w}$. (B) Circuit to map features onto the Bloch sphere. (C) Implementation of the parameterized diagonal two-qubit operations using CNOTs and Z-gates. (Havlíček et al. 2019)

computer, called qBLAS. This is because the quantum data has exponentially fewer bits than the classical data.

2.1. *Supervised Learning*

The most common supervised machine learning algorithm is SVM (support vector machine). The goal of classification is to correctly classify data and to do that SVM constructs the best hyperplane to divide the data most accurately. To do that we can find the distance between two data points

$$\Delta(\overrightarrow{x_i}, \overrightarrow{x_j}) \tag{3}$$

this is be used to construct the optimal hyperplane between groups of data. The goal is to maximize the margins between the data and the plane. This is a linear algebra problem in a high dimensional vector space.

We can use quantum computing to speed up this process and one such method to do so is by using quantum variational classification (Havlíček et al. 2019). We can then use a quantum circuit to generate the hyperplane Fig 1. This hyperplane is then used to label data points. However in this case we take the expectation value of the hyperplane and we can classify binary labels with positive or negative expectation values Fig 2. (Havlíček et al. 2019) There are countless other methods that have been implemented and are being researched to translate the functionality of SVMs to quantum systems, some other methods include quantum least-squares SVM (Rebentrost et al. 2014) and quantum kernel estimator (Havlíček et al. 2019).

2.2. *Unsupervised Learning*

The most common unsupervised learning method is principal component analysis. From the dataset we can construct a co-variance matrix

$$C = \sum_i \overrightarrow{x_i}\overrightarrow{x_i}^\dagger \tag{4}$$

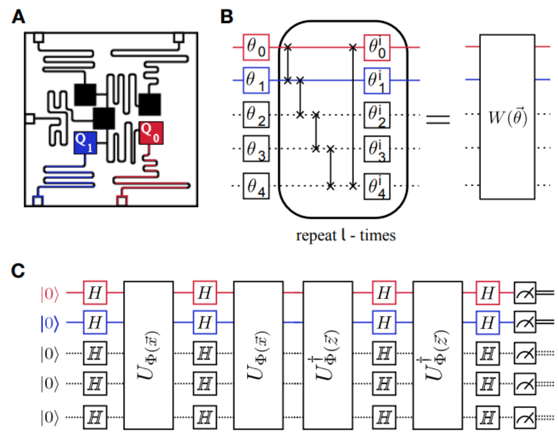

Figure 2. Circuit Implementation: (a) Schematic of the 5-qubit quantum processor. (b) Circuit used for optimization.(c) Circuit used to measure fidelity. (Havlíček et al. 2019)

afterwards we diagonlize C and we get

$$C = \sum_k P_k \overrightarrow{w_k}\overrightarrow{w_k}^\dagger \tag{5}$$

where $\overrightarrow{w_k}$ are the principal components when $P_k \gg 0$. This can be seen as compressing the data into values or patterns that matter. These same calculations can be done using quantum algorithms because these vectors can be written as states of a high dimensional quantum system. In a quantum system we can write the co-variance matrix as a density matrix

$$\rho = \frac{1}{\sqrt{N}} \sum_i |\overrightarrow{x_i}\rangle \langle\overrightarrow{x_i}| \tag{6}$$

We want to find the eigenvalues and eigenvectors of this to determine the principal components. If we apply our state to some 0, then we get

$$|\Psi\rangle\,|0\rangle \longrightarrow \sum_k \Psi_k\,|k\rangle\,|a_k\rangle \tag{7}$$

where $|k\rangle$ is an eigenvector of A and $|a_k\rangle$ is an eigenvalue of A. A is the Hamiltonian

$$e^{-iAt}\,|\Psi\rangle \tag{8}$$

This is called the quantum phase algorithm.

Now to actually find the eigenvalues and eigenvectors we want

$$e^{-i\rho t}\,|\Psi\rangle \tag{9}$$

This means that we need several copies of ρ to be able to exponentiate and we can do that using this

$$\mathrm{tr}_A e^{-i\mathrm{SWAP}\Delta t}\rho_A \otimes \sigma\rho_B^{i\mathrm{SWAP}\Delta t} = e^{-i\rho\Delta t}\sigma e^{i\rho\Delta t} + O(\Delta t^2) \tag{10}$$

this entire process is known as qPCA and it computes in $O((\log d)^2)$ time while classical data takes $O(d^2)$ time. (Biamonte et al. 2017) New circuits are continually being researched to improve the speed at which all of this can occur experimentally. SVD, a classical method used to compute PCA in $O(MN^2)$ for an $M \times N$ matrix, can be computed using a quantum method, qSVD in $O(\mathrm{poly}(r)\mathrm{poly}(\log(MN)))$ time, where r is the rank of the matrix.

2.3. *Deep Learning*

Deep learning is a a machine learning algorithm where there are several layers connected to one another and between the nodes of each layer Fig 3, the connection have specific weights to them. This is called a neural network

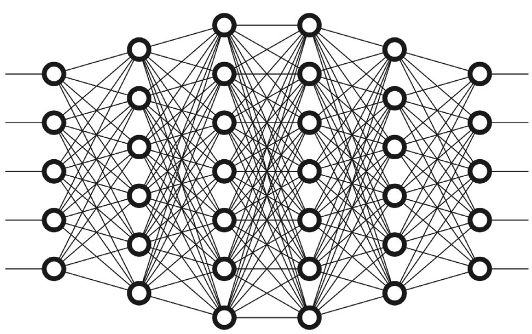

Figure 3. Neural Network: 1 input layer, 4 hidden layers, and 1 output layer

and it was inspired by how our brain is understood to function. Deep learning is one of the best methods used today for any ML task. A machine that can do deep learning is a Boltzmann machine an Ising model where the Hamiltonian

$$H = \frac{\hbar}{2}\sum_i w_i\sigma_z^i + \Gamma_{ij}\sigma_z^i \otimes \sigma_z^j \tag{11}$$

and the goal is to tune the weights w_i and the energies and couplings Γ_{ij} to recreate the statistics of the data. Specifically, the goal is to create a physical system who's thermal states

$$G = \frac{1}{Z(\beta)}e^{-\beta H} \tag{12}$$

match the statistics of the data. This can easily be made quantum by adding $\alpha_i\Delta_x^i + \kappa_{ij}\Delta_x^i \otimes \Delta_x^j$ to the Hamiltonian. This is known as deep quantum learning. One of the benefits to deep quantum learning is that it doesn't need a big quantum computer to function, instead it can be run on quantum annealers. Quantum annealing is a procedure for finding the global minimum of a function using quantum fluctuations. This allows us to reach the ground state of an Ising model much quicker than when done classically. Fig 4

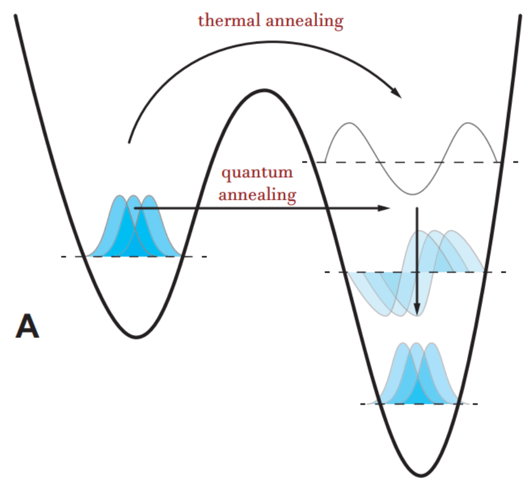

Figure 4. Quantum tunneling lets a quantum state get to the local minimum (Biamonte et al. 2017).

At D-wave, they have developed a quantum annealer for quantum systems and it has been used to perform deep learning. Currently this is only being done on a small dataset. (Biamonte et al. 2017)

2.4. *Other Algorithms*

There are many other machine learning algorithms which have their quantum counterparts which show quantum speedup: Bayesian theory, hidden Markov models, decision trees, k-nearest neighbour, etc. For all of these classical models there are quantum circuits created to apply these models to quantum data and exponentiate compute time.

3. DEVELOPMENT SOFTWARE

Within the past year, quantum machine learning software has taken a huge leap with the development of TensorFlow Quantum. TensorFlow is a machine learning library developed by Google, and TensorFlow Quantum is that same library but built for quantum algorithms. Their QML library allows for developers to be able to build models that can learn on quantum and classical data. There are also hybrid models that can be built using this library. Fig 5 This library along side other libraries such as cirq allow the development of complex quantum circuits and algorithms so that as the hardware gets better, so does the implementation. The library allows for the preparation of quantum datasets, the creating and training of both classical and quantum models, and the evaluation of cost functions. The back end systems are constantly being updates as newer and faster quantum circuits are developed to perform these tasks. In the paper they show how to create sev-

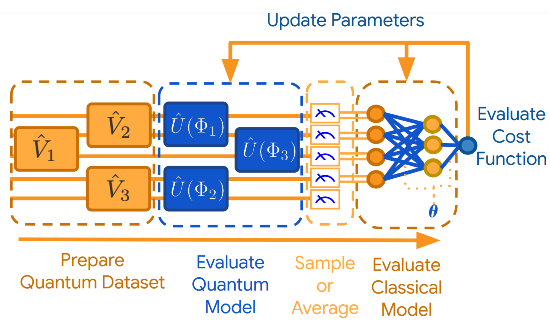

Figure 5. Pipline for hybrid model in TFQ (Broughton et al. 2020).

erl models including quantum convolusional neural networks (QCNNs) they test out three types of QCNNs and evaluate its performance. Fig 6 Many complicated ML

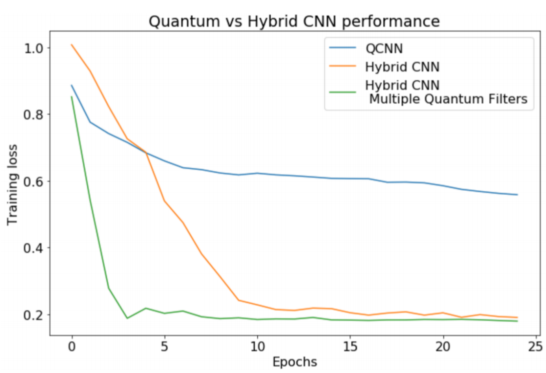

Figure 6. Performance of difference QCNN's (Broughton et al. 2020).

models such as LSTM also have a quantum implementation.

4. DIFFICULTIES AND CHALLENGES

There are several issues that still need to be solved before quantum machine learning becomes easily doable, one of the issues is of course, decoherance, quantum machine learning requires repeated measurements and this leads to decoherance. However, this is being reduced as we develop better technologies and measurement methods. Another hardware issue is the development of qRAM, many quantum algorithms require qRAM and currently, while proof of concept pieces have been constructed, we still face difficulty constructing large arrays. qRAM need to be able to read all the data without much decoherance, this means that it has to operate quickly with minimal errors.

Another issue is that controlling quantum systems is difficult. Luckily, classical machine learning is being implemented and it has better performance than numerical tools.

While hardware challenges will remain an issue for the near future, they are quickly shrinking as time passes, it seems inevitable that the hardware for large quantum computation will be available soon, and when that happens, there are other problems that remain for QML. One of the major problems is bottle-necking in the input and output of data. Sure quantum algorithms provide speedup during training but the conversion of data from classical to quantum could take too long to provide any benefits for classical systems. Unless the data is initially quantum (stored as quantum data) there is going to be a bottleneck here until better methods for conversion are created.

5. THE FUTURE OF QML

5.1. *Current Research*

This field of research and development has taken a giant leap forward over the last few years. With the innovations at D-wave, IBM, and other hardware groups. Small quantum computers are now available, machine learning specifically can take advantage of small-scale quantum computers of around 50 to 100 qubits because they do not need quantum error correction due to the high data compression. Quantum annealing, qRAM, NV-diamond arrays are all advancing in size and complexity (Biamonte et al. 2017). Specifically, quantum annealers have gown bigger with over 2000 qubits. There is also TensorFlow Quantum which allows for any developer to contribute towards improving QML systems. Along side this there is also constant improvements being made to quantum algorithms and many different circuits are being experimentally tested to find the fastest and best performing ones. Just a few months ago there was an improved qPCA algorithm created using qSVD (quantum singular value decomposition). (He et al. 2019)

5.2. *Future use cases*

There are many potential use case scenarios for QML. The most obvious ones are the areas where we have collected the most data for very complicated problems. Chemistry and biology are two sciences that can be advanced from this technology, from the simulation and creation of different molecules and Drugs to the use in identifying diseases, their functions, and their cure. For example, the search for the corona virus vaccine used 437 petaflops of computational power. With a QML algorithm we could have done it much quicker. Another use case is for autonomous driving, am autonomous car must make thousands of decisions every second from terabytes of data.

6. CONCLUSION

Machine Learning, while still a growing and developing technology, has a major flaw which is becoming more and more apparent as we try to improve our models by training on larger and larger datasets. Quantum computing, another rapidly growing and developing technology, allows its quantum supremacy to exponentially speed up machine learning tasks and . While there is still a lot of work to be done, quantum machine learning is an inevitability that will be much superior to classical machine learning when fully developed.

REFERENCES

Biamonte, J., Wittek, P., Pancotti, N., et al. 2017, Nature, 549, 195, doi: doi.org/10.1038/nature23474

Broughton, M., Verdon, G., McCourt, T., et al. 2020, TensorFlow Quantum: A Software Framework for Quantum Machine Learning. https://arxiv.org/pdf/2003.02989.pdf

Havlíček, V., Crócoles, D., A., Temme, K., et al. 2019, Nature, 567, 209, doi: doi.org/10.1038/s41586-019-0980-2

He, C., Li, J., & Liu, W. 2019, Physics Letters A, 383, 24, doi: doi.org/10.1016/j.physleta.2019.06.026

Rebentrost, P., Mohseni, M., & Lloyd, S. 2014, Physics Review Letter, 113, 13, doi: doi.org/10.1103/PhysRevLett.113.130503