The Effects of Noise on Hybrid Quantum Machine Learning Models for Image Classification

Aditya Rajeev

Abstract

Quantum machine learning is a field in computer science that meets at the intersection of two emerging fields - quantum computing and machine learning. Quantum computing is immensely powerful, especially as it pertains to manipulating and understanding big data sets and has extensive potential in its applications with machine learning. Quantum computers, in comparison to classical computers, can use quantum entanglement and super positioning while processing data, giving quantum computers the ability to outperform classical algorithms, which is known as quantum advantage. The potential applications of quantum machine learning include natural language processing, medical diagnoses, and image processing. The purpose of this study is to analyze the efficacy of quantum machine learning models, as it pertains to image classification on supervised learning models. I show that qubit gate noise limits the effectiveness of quantum deep learning networks in quantum systems with 1 or 2 qubits, slightly affect the ability of the model at error values of 0.01 and 0.2 but impairing the model to a substantial degree as the depolarizing error approaches 0.5.

Introduction

Machine learning is a field in computer science which tries to determine patterns in sets of data like images, text, or numbers. The model will iterate through the initial data as many times as specified, as it is trying to "learn", with each iteration being known as an epoch. Machine learning tasks are scalable and are used in all sorts of technology, like in the Google search engine or in social media recommendation algorithms [1].

A quantum layer can use the power of quantum physics to process data, giving it the ability to improve machine learning models and achieve quantum advantage. The quantum layer uses qubits, or quantum bits. When a quantum layer is added to a machine learning model, it combines the fields of quantum computing and machine learning [2, 3]. These qubits are incredibly powerful because they can harness properties of the quantum realm, like with super positioning, during which a bit can enter multiple states at the same time until it is measured. This means qubits are exponentially more powerful than classical bits, with n qubits having the power of 2^n classical bits. This makes quantum incredibly useful, with the ability to store and manipulate significantly larger sets of data than classical computers [4]. Its application in machine learning has significant benefits – since machine learning algorithms require a significant amount of storage and computing power, the quantum layer can improve the efficiency of this process [5].

I will start by explaining the layout of both the machine learning model and the quantum layer to show how they interact. I will then illustrate the problems and benefits of hybrid quantum machine learning based on these results.

Background

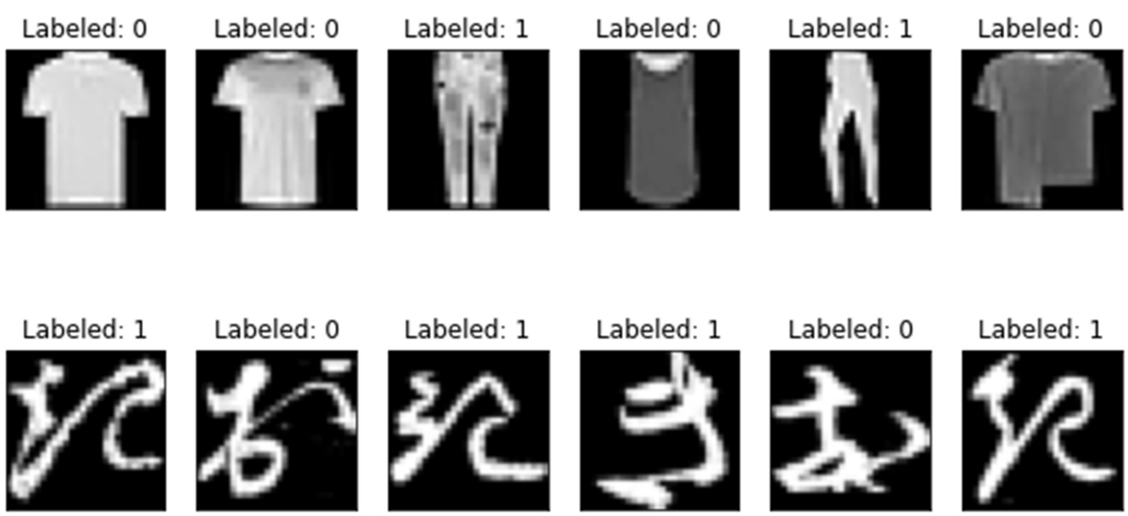
Machine learning includes the field of deep learning, which uses neural networks that imitate the structure of neurons in the brain. These neural network systems start with an input layer of neurons that feeds into hidden layers. Each neuron in these hidden layers has a weight and bias that are adjusted when the modeling is training to make the model more accurate. These neural networks are typically feedforward networks, meaning that the data only flows in one direction. However, for the model to improve, the model will need be evaluated and all the weights and biases need to be adjusted. To do this, the models will undergo backpropagation – using gradient descent and a loss function, the models will decide how to adjust the weights and biases. The loss function defines how the model evaluates itself and how it will calculate error, giving a numeric representation of how “correct” the current model is. Some machine learning systems, like those doing work in the field of medicine, will need to be more accurate, meaning that the loss function will need to evaluate error more critically to influence the model into making more drastic changes during backpropagation.

The model can be evaluated on testing data for accuracy as well, but this will not be used for backpropagation. It is important to remember that loss and accuracy are two separate metrics, and that one does not influence the other. Models may consist of high loss and high accuracy, low loss and low accuracy, or any combination of results. The loss of the function is simply how far away the estimated results are from the expected results, while the accuracy is just how often the model gives the correct output on a set of data. The loss may be arbitrarily high because it significantly underperformed on one result, skewing the average value of the loss function, but this will not affect the accuracy in the same way.

Machine learning models may become too accurate, however, and may only work on the input data. This phenomenon is known as overfitting, where the machine learning model has weights that are too specific to the input data that will not work on any other similar data. To fix this, machine learning models will have dropout layers which forget data to ensure that the model does not overfit.

Another important concept to understand is quantum error. When a quantum system is converting classical bits into quantum bits or vice versa, there may be some error in the translation. This error is known as readout error. These quantum bits will also go through gates in the quantum layer, which also have error associated with them as well, known as gate error. The gate error varies depending on the type of gate the qubit is going through. Actual quantum systems typically have thermal relaxation errors as well, which occur when a qubit is changing state. For the purposes of this study, a depolarizing error is introduced into the quantum system to simulate what a system with higher error values would behave like. This error applies either a uniform Pauli error channel or a depolarizing channel depending on the parameter entered.

Methods



(a)

(b)

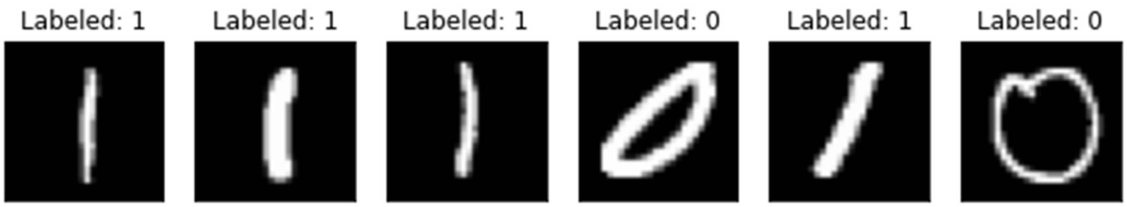
(c) 

Figure 1. Examples of the different data used to train the machine learning model. Each figure is labeled with a 0 or a 1, and the model tried to predict which of those two categories the image represented.

In this study, three different image datasets were used to train and test the model. The first set of image data was a set of clothes, out of which two were used – images with a t-shirt, and images with trousers. T-shirts were classified with the label 0, while trousers were classified with the label 1 [Figure 1(a)] [6]. The second set of image data used was a set of Japanese characters with Hiragana characters [7]. Once again, two distinct classes of Hiragana characters were chosen and classified accordingly. [Figure 1(b)]. The last set of data were images that contained hand drawn zeroes and ones, which were labeled with zeroes and ones respectively [Figure 1(c)] [8]. Every image was of size 28x28 pixels and were greyscale. 200 images were taken from each dataset for training, with half the images being of one classification and the other half being of the other classification. This was done to prevent bias in the training. These sets of data were trained independently, but all shared a common neural network and quantum circuit structure, and they all ran on the same simulated error. 100 images were taken from each dataset as well for testing, with an equal amount of each class to prevent bias. This testing data was then used to determine the accuracy of the model.

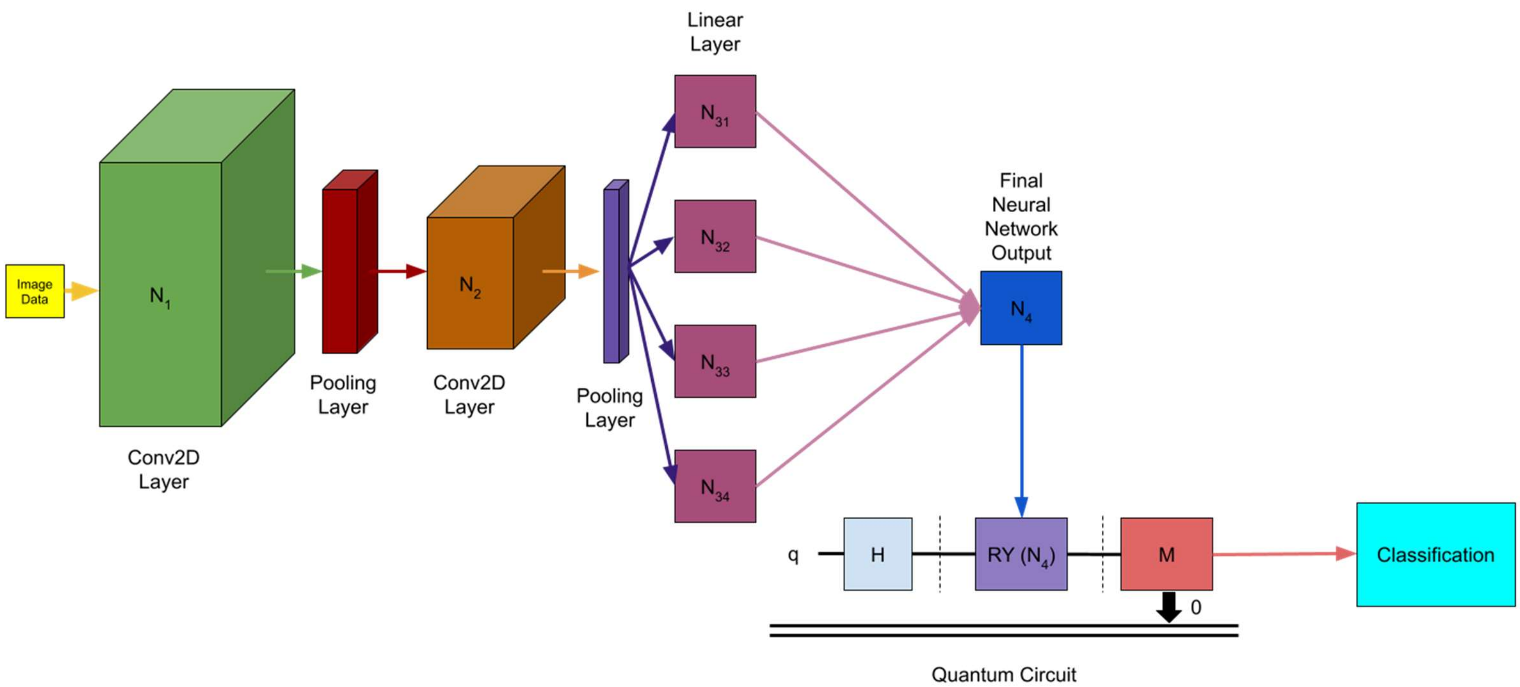


Figure 2. A diagram of the hybrid quantum machine learning model.

The model used in this experiment was adapted from the Qiskit textbook [9]. The hybrid model that the data was trained on involves 3 parts – the classical machine learning neural network, the quantum circuit, and the final optimization algorithm. As seen in Figure 2, the data is initially fed into a classical machine learning model [10]. This model consists of two Convolutional 2D layers with a pooling layer between each, with a dropout layer after those. The data is then reorganized and fed into two fully connected layers, producing one output value.

This value was then used as a trainable parameter in the quantum layer of the model [Figure 2]. For the purposes of this study, the quantum layer had either one or two qubits with an RY gate applied to each qubit. This RY gate was rotated by a certain angle, theta, which was fed as a parameter into the quantum circuit. The measured value of the qubit(s) would then serve as the final prediction for the data after passing through an optimization function. During training, the model would optimize the neural network weights using backpropagation. In order distinguish the quantum circuit from the classical machine learning model when calculating the gradient descent, the parameter shift rule [11] was used as a part of the backpropagation routine.

The models ran using an Adam optimizer [12] with a learning rate of 0.001. The cost function used was a negative log-likelihood loss function. For every trial, 3 separate models were trained and analyzed, and the average values of the results were taken. The first set of error values was directly taken from an IBM Quantum processor, *ibmq_lima* (v. 1.0.38), and these error values were mapped onto a quantum circuit simulator [13]. To test higher qubit gate error levels, a quantum circuit simulator with equal depolarizing error on one and two qubit gates was used. The error values for the circuit taken from the IBM backend for qubit 0 were:

Square Root X Gate Error; 0.0005,

T_1; 8.3 x 10^{-5} seconds,

Readout Error; 0.02.

The error values for qubit 1 were:

Square Root X Gate Error; 0.0005,

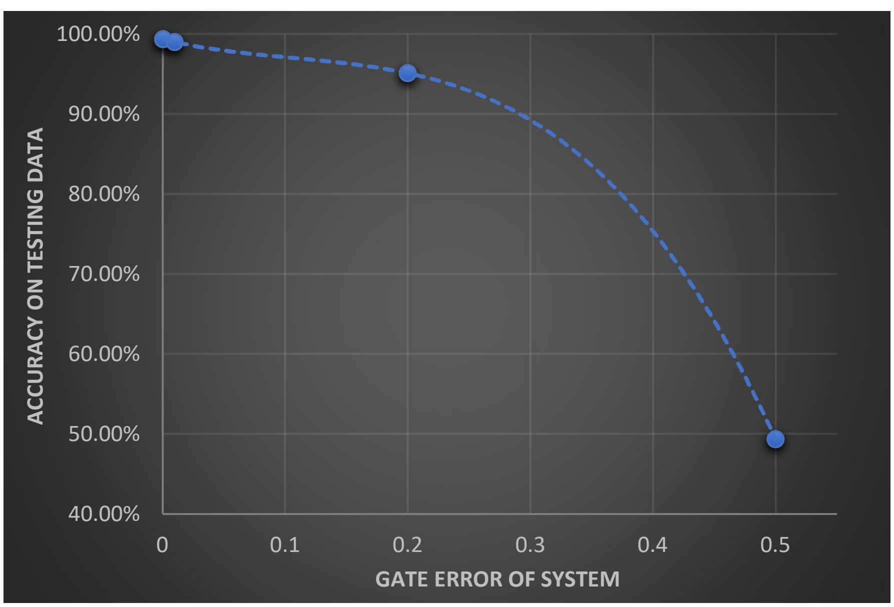
T_1; 9.3 x 10^{-5} seconds,

Readout Error; 0.01.

The error values for the following simulators were a depolarizing error of 0.01, 0.2, and 0.5 respectively on all one and two qubit gates.

Results

The first model trained was on the fashion dataset. The results for the one qubit trials are as follows.



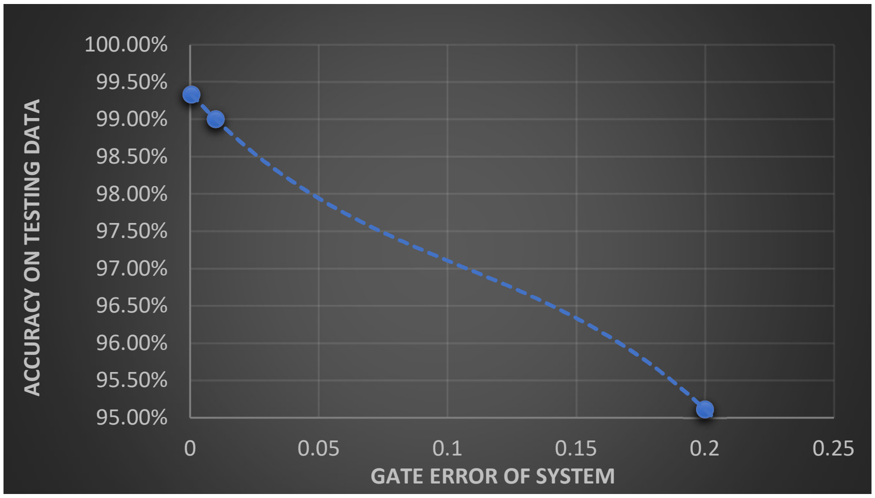
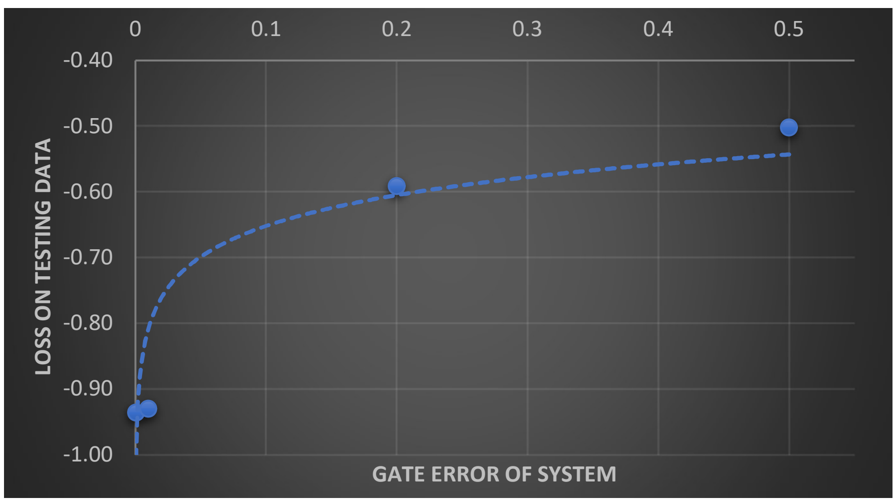


Figure 3a (top) and 3b (bottom). These graphs show the accuracy of the model in comparison to the gate error of the system for a one qubit model. 3(b) offers a closer view of the distinction between lower error values.



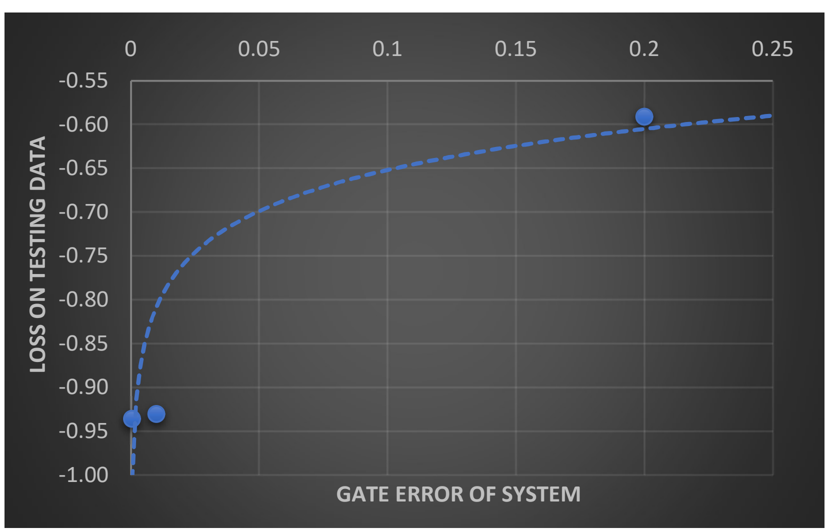
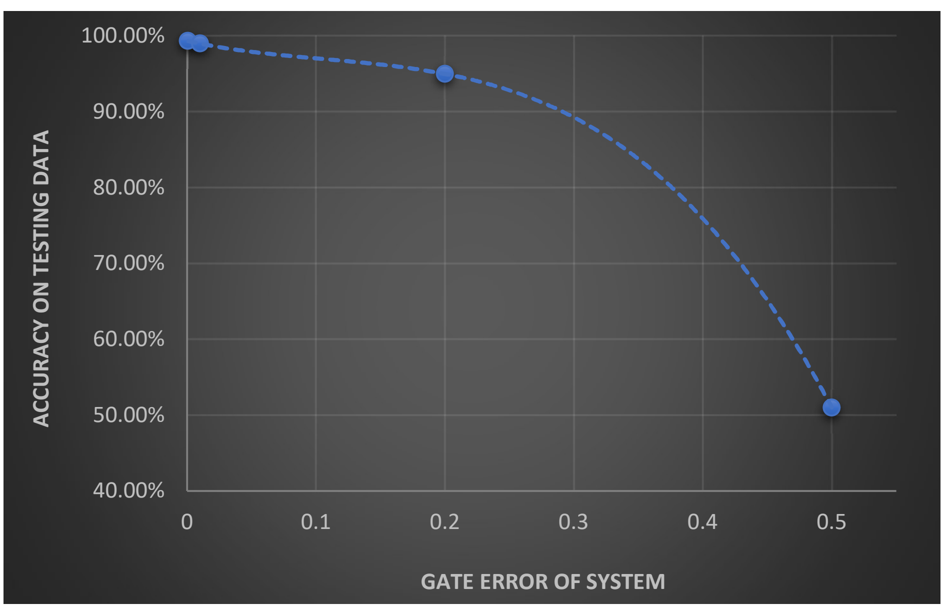


Figure 4a (top) and 4b (bottom). These graphs show the loss of the model on the testing data in comparison to the gate error of the system for a one qubit model. 4(b) offers a closer view of the distinction between lower error values.

This data shows that, in one qubit machine learning system, the average accuracy falls at a limited rate between the gate error values of 0.0005 and 0.2 [Figure 3(b)], with the accuracy of the model steeply falling until it reaches an error of 0.5 [Figure 3(a)]. An accuracy of 50% is no better than random guessing, so the model does not work at all by that point. In terms of the loss calculated by the negative log-likelihood cost function, the loss steeply decreases after an error value of 0.0005 [Figure 3(d)] and tapers off as the error value approaches 0.2 and 0.5 [Figure 3(c)].

In terms of a hybrid machine learning model with 2 qubits, the results are shown in Figure 5 and 6.



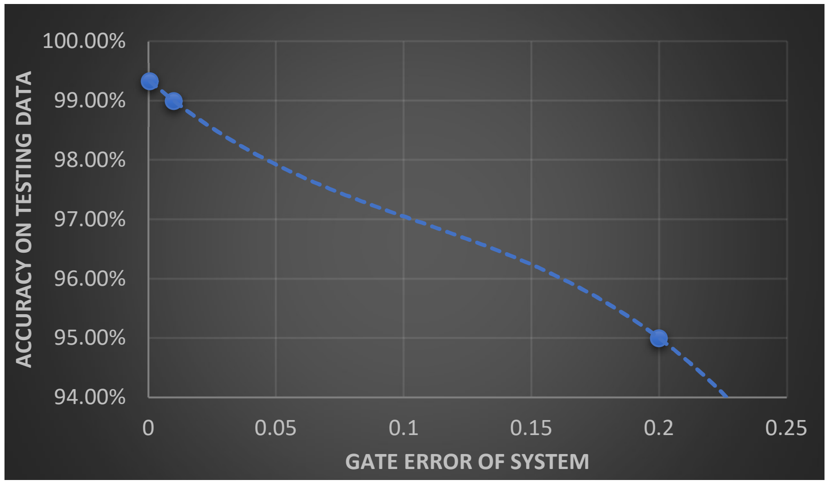
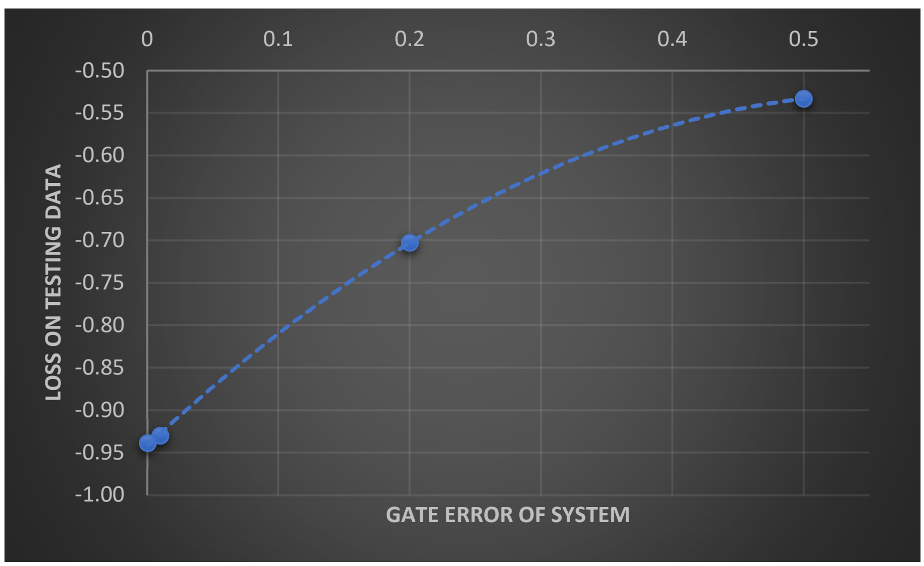


Figure 5a (top) and 5b (bottom). These graphs show the accuracy of the model on the testing data in comparison to the gate error of the system for a two-qubit model. 5(b) offers a closer view of the distinction between lower error values.



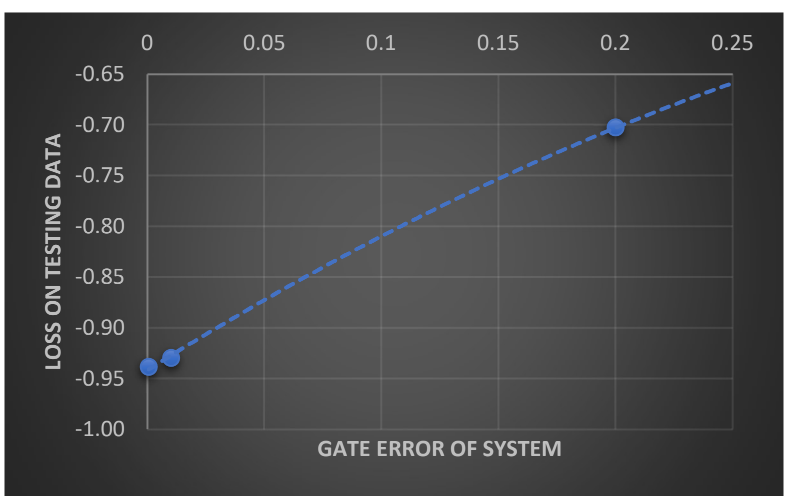


Figure 6a (top) and 6b (bottom). These graphs show the loss of the model on the testing data in comparison to the gate error of the system for a two-qubit model. 6(b) offers a closer view of the distinction between lower error values.

The two-qubit system performed just as well as the one qubit system, indicating that the addition of another qubit to the quantum layer is not necessarily beneficial. The same patterns in the accuracy graphs of the one qubit model [Figure 3(a) and 3(b)] are demonstrated in the two qubit model accuracy graphs [Figure 5(a) and 5(b)]. In the loss graphs of the two-qubit model [Figure 6(a) and 6(b)], however, there is a slight improvement in the performance of the model, with the error values being lower than the corresponding error values in the one qubit system [Figure 4(a) and 4(b)]. This discrepancy is not substantial enough to demonstrate an improvement in performance. These results may be due to the additional quantum gate error associated with the addition of another qubit, since both qubits have independent depolarizing error values. The hybrid model is not optimized for a multi-qubit quantum layer as well, which could be contributing to the negligible change in performance.

Larger scale quantum machine learning models have some tolerance when it comes to error in the quantum system. The disparity in accuracy may not increase significantly if the error is less than 0.01, but the same is not true for this disparity in the loss. Regardless, it shows that quantum machine learning models on today's quantum computers can withstand the error that these systems may possess and allows for models to be trained effectively. These results are good for simple tasks like natural language processing or image classification, but its applications in fields like medical diagnoses, where even the slightest of error is not tolerated, are still limited.

An important note is that these results do not represent a true hybrid quantum machine learning model. The quantum layer is simulated on classical hardware, which negates some of the power of actual quantum computers. For example, the quantum layer used in this experiment generates no entanglement, a property that can only be demonstrated if run on actual quantum hardware. A model may only be able to achieve quantum speedup if run on a real quantum computer.

Conclusion

The work demonstrates that both one- and two-qubit approaches to hybrid quantum machine learning are resilient to relatively high quantum gate error. The trials have shown that quantum circuit models that incorporate depolarizing errors retain their accuracy levels at above 90% for error rates as high as 0.2 for the image datasets, especially with the easier datasets like the numbers MNIST dataset. As the depolarizing error approaches 0.5, the models show that the hybrid algorithm fails and the accuracy approaches 50%.

Through all the training, the complexity of the image data influenced performance of the model. The Japanese dataset was hardest for the model to label, with the model's loss and accuracy being lower across the board compared to the other datasets. This was due to the complexity associated with the dataset since the categorizations had two components to them due to the nature of the Hiragana characters (explained further in [7]). The numbers dataset, on the other hand, was significantly easier for the model to label since the images are qualitatively more distinguishable. The model performed well enough to notice a difference between error values, but it raises the important idea that a dataset may exist where the model fails at any error level, making it impossible to make these comparisons.

This work has many applications in the increasing growth and use of quantum computers and systems. Companies creating quantum systems with several high error qubits can be effective if their system is used for a quantum machine learning model, as the accuracy of the model is not substantially affected as the gate error values approach 0.01. Companies creating quantum computers with few qubits and low error may be less effective when creating hybrid quantum machine learning models - an approach with more qubits and higher error has the potential to be more powerful if the model is created with a multi-qubit layer in mind.

Citations:

[1] Huang, HY., Broughton, M., Mohseni, M. et al. Power of data in quantum machine learning. Nat Commun 12, 2631 (2021). https://doi.org/10.1038/s41467-021-22539-9

[2] Maria Schuld, Ilya Sinayskiy & Francesco Petruccione (2015) An introduction to quantum machine learning, Contemporary Physics, 56:2, 172-185, DOI: 10.1080/00107514.2014.964942

[3] Saini, S., Khosla, P., Kaur, M. et al. Quantum Driven Machine Learning. Int J Theor Phys 59, 4013–4024 (2020). https://doi.org/10.1007/s10773-020-04656-1

[4] Biamonte, J., Wittek, P., Pancotti, N. et al. Quantum machine learning. Nature 549, 195–202 (2017). https://doi.org/10.1038/nature23474

[5] Vedran Dunjko and Hans J Briegel 2018 Rep. Prog. Phys. 81 074001

[6] Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv. https://doi.org/10.48550/ARXIV.1708.07747

[7] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., & Ha, D. (2018, December 3). *Deep Learning for Classical Japanese Literature.*

[8] LeCun, Y., Cortes, C. and Burges, C.J.C. (1998) The MNIST Database of Handwritten Digits. New York, USA. http://yann.lecun.com/exdb/mnist/

[9] MD SAJID ANIS., Abby-Mitchell, Abraham, H., AduOffei, Agarwal, R. (2021). Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505

[10] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, & R. Garnett (Eds.), Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[11] Crooks, G. E. (2019). Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition.

[12] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization.

This source was used in the methods section to explain what the Adam optimizer was.

[13] IBM Quantum. https://quantum-computing.ibm.com/, 2021