Quantum Computing based Generative Adversarial Network for Time-Series forecasting

Aadittya Tiwari1, Bradley S Fulk1,2†

1,2 Lynbrook High School, 1280 Johnson Ave, San Jose, 95129, California

\*Corresponding author; E-mail: AadiTiwari1223@gmail.com   
†Project Mentor; E-mail: [brad\_fulk@fuhsd.org](mailto:brad_fulk@fuhsd.org)

Abstract

Generative Adversarial Networks (GANs) are a popular machine learning model that can generate synthetic data by training a generator to produce samples that a discriminator can't distinguish from real data. GANs have shown promise for time-series forecasting, where synthetic values can be used to predict future trends, such as stock prices. However, this can be taken one step further with the usage of Quantum Computers. In this project, I developed a novel Quantum GAN (QGAN) architecture that combined a Quantum variation of a Long Short Term Memory generator with a classical Convolutional Neural Network discriminator. The QGAN outperformed its classical counterpart in number of convergence epochs required and had a slightly higher prediction accuracy across a range of stock datasets, despite using a low qubit count. These results provide evidence of quantum supremacy in the domain of deep learning.

**Keywords:** Quantum Computing, Qubits, Machine Learning, Time-series Forecasting

# Introduction

# 1.2 Introduction to Quantum Machine Learning

Quantum Machine Learning (QML) is an emerging interdisciplinary field combining quantum computing with machine learning algorithms to enhance data processing. Quantum Computers utilize “qubits” (quantum-bits) which exist simultaneously in multiple states, unlike classical bits that can only exist as either a 0 or 1, called quantum superposition. By leveraging the power to be in multiple states at once and perform parallel calculations faster through this ability, machine learning algorithms can be optimized for Quantum Computers to process and analyze vast amounts of data faster and obtain more precise results. As QML develops as a field, new algorithms are found frequently that demonstrate “Quantum Supremacy,” cases where Quantum Computing algorithms have advantages over Classical Computing algorithms. QML has the potential to transform research fields such as disease detection, physical simulations, but for the context of this project, time-series forecasting, by providing more accurate and faster solutions for these complex problems. This paper describes a novel quantum machine learning model that does indeed demonstrate Quantum Supremacy in the field of Generative Adversarial Networks.

# Quantum Computing and Machine Learning

## Simple introduction to Quantum Computing

Quantum computing is an emerging field of computer science that promises to revolutionize how we solve complex problems. Unlike classical computers, which rely on bits that can only exist in one of two states, quantum computers use quantum bits, or qubits, which can exist in multiple states simultaneously. This allows quantum computers to perform certain calculations much faster than classical computers, making them particularly well-suited for tasks such as optimization, cryptography, and in our paper, machine learning.

One of the key features of quantum computing which makes it so powerful is the phenomenon of superposition, which allows a qubit to exist in a combination of both 0 and 1 at the same time. In pop culture it's often said that an n number of qubits correlate to the power of 2^n classical its. While this isn't entirely true, it gives a rough idea of the computational advantage quantum computers provide. Another important feature is entanglement, where two or more qubits can be correlated in such a way that their states are linked together so that affecting one qubit will affect its entangled qubit as well. This paired with superposition is what provides such powerful exponential speed-ups in time complexities.

Quantum computing is still in its early stages, and many of the practical applications of this technology have yet to be realized. However, there has been significant progress in recent years, with the development of increasingly powerful quantum computers and the occasional demonstration and realization of quantum supremacy over classical computers for cherry-picked scenarios. However, Quantum Machine Learning as of late has been receiving a lot of attention in the last few years due to it being a conjunction of 2 of the most revolutionizing fields in Computer Science - Quantum Computing, and Machine Learning.

## Simple introduction to Machine Learning

Machine learning is a branch of artificial intelligence that allows machines to learn from data without being explicitly programmed. It involves training algorithms to identify patterns in data and make predictions or classifications based on those patterns. The process of machine learning consists of three main stages: data preparation, model training, and model evaluation. During the data preparation stage, data is collected, cleaned, and transformed into a format suitable for analysis. In the model training stage, machine learning algorithms are used to train models on the prepared data. Finally, in the model evaluation stage, the trained models are tested on new data to ensure their accuracy and effectiveness.

**2.3.2** **Generative Adversial Networks (GANs)**

Generative Adversarial Networks, or GANs, are a type of neural network machine learning architecture that can generate new data that is similar to the training dataset. They consist of two models - a Generator and a Discriminator - that compete to learn and generate complex data such as images, audio, and video files. The Generator creates fake data to train on the Discriminator, which in turn learns to identify real data from the fake data produced by the Generator. This adversarial game between the two models continues until the Generator produces data that is indistinguishable from the real data.

# Quantum GAN Model Architecture

## Quantum Long Short-Term Memory Architecture

The Quantum GAN (Q-GAN) that this paper proposes is essentially a Quantum-Classical hybrid machine learning model. The generator for the QGAN is a Quantum Long Short-Term Memory (QLSTM) model developed by Chen et al. The QLSTM simply replaces a layer in a classical Long Short-Term Memory (LSTM) circuit with 6 Variational Quantum Circuit (VQC) to form a QLSTM cell. This new quantum layer in the model was shown to have advantages over the classical counterpart in being able to converge quicker and have a stabler loss function graph.

Information flow in a quantum LSTM:

where denotes the sigmoid function, are classical neural networks , where represents the forget block, represents the input block, represents a new state cell candidate, and represents the output block.

Information flow in a quantum LSTM:

Where denotes a different Variational Quantum Circuit that will be used in the hybrid Quantum LSTM.

Chen et al.’s research paper demonstrated that this QLSTM model had advantages over a control LSTM in terms of epochs to converge, stabler convergence, and slight prediction accuracy for time-series forecasting which could be attributed to its ability to recognize temporal dependencies faster, which in turn is a result of using VQCs.

## Classical Convolutional Neural Network Architecture

The discriminator in our model is a classical Convolutional Neural Network (CNN) with three layers. The reason we are trying to avoid using Quantum Machine Learning models everywhere despite its proposed advantages is because it is computationally expensive to run on my hardware. Sending algorithms to a Quantum Computer or Quantum Computing simulator to run ends up taking significantly longer than running a classical algorithm on your machine and ends up taking hours longer to train. The CNN is a prebuilt model made using the PyTorch machine learning package, meaning that backpropagation and such are automatically prebuilt into the model. A one-dimensional convolution is applied to input signals. In the simplest case, the output value of the layer with input size and output can be described with the equation

# Q-GAN implementation

Moving on from the technical architecture, we begin to implement the Quantum LSTM into a time series generator. Essentially, all we do is replace the classical generator in the Classical GAN (CGAN) (which is currently a classical prebuilt PyTorch LSTM) with our custom Quantum LSTM. However, as our model is not prebuilt and the entire GAN relies on a prebuilt model, we will need to still keep the code for the prebuilt generator and instead create what is basically a second instance of the generator and call that instead, while the classical generator remains but not for the actual purpose of generation, more as a utility function. Code for the GitHub can be found at this link which contains a file named under QGAN and one under CGAN that show the generators and discriminators for each GAN which cannot fit here.

## Variables

The variables for both models were set after lots of experimentation to find optimal values to allow both functions to perform at their best but keeping as many values alike as possible to allow objective testing. The data table as follows shows what each value was set to in the following categories for the generators and the discriminators:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generator | Learning Rate | Epochs | Optimizer | Layers |
| QGAN | 0.000016 | 15 | Adam | 4 |
| CGAN | 0.004 | 100 | Adam | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Discriminator | Learning Rate | Epochs | Optimizer | Layers |
| QGAN | 0.00003 | 300 | Adam | 3 |
| GAN | 0.00003 | 300 | Adam | 3 |

As you can see, the dimensions for the discriminator are identical since we are using the exact same Convolutional Neural Network (CNN) model for both GANs.

# Model evaluation methods (Experiments & Results)

## Evaluation Methods

We plan to evaluate the efficiency of the QGAN through the following 3 steps:

a) Selecting 3 stocks with differing trends to encourage robust models

b) Using an existing classical GAN for time series forecasting

c) Developing a novel QGAN model that changes ONLY the generator of the classical GAN

d) Evaluating the performance of both models on the 3 different stocks in the categories of prediction RMSE, epochs to converge, and parameters across all 6 predictions.

## Results

The following data table shows the performance of each model across the evaluation metrics defined above:

|  |  |  |
| --- | --- | --- |
| RMSE | CGAN | QGAN |
| Intel prediction | 7.11 | 13.10 |
| Apple prediction | 7.97 | 3.61 |
| Synopsys prediction | 4.77 | 8.50 |

|  |  |
| --- | --- |
| Epochs taken to reach convergence |  |
| QGAN | 6 out of 15 total epochs |
| CGAN | 80 out of 100 total epochs |

|  |  |
| --- | --- |
| Total parameters (Generator + Discriminator) |  |
| QGAN | 233+6199041 |
| CGAN | 109429+6199041 |

# Conclusion and Outlook

QML almost always returned higher accuracy for forecasts, albeit minimal. This higher accuracy is indeed a result of the QLSTM being used and having it perform calculations through a quantum computing simulator which then gave the QLSTM an advantage in converging quicker. QML was also able to reach comparable performance to CML in a tenth of the epochs. This is related to the Qubits above again - as Quantum Bits can exist in multiple states at once, it’s able to compute different possibilities at the same time. This means it trains more in one epoch than Classical ML does in one epoch because of these parallel time computations. Another interesting result was that there were significantly less parameters in the QML models, again as a result of having qubits that make up for the lack of parameters. Finally, as a result of using a simulator (a classical computer coded to behave like a quantum computer) and not a real quantum computer, it will actually take you more time to train your models unless you use a real computer. Possible room for improvement would be to try out different QC simulators from different companies, or in the far future evaluate performance on a real quantum computer. As hardware scales with time, so will algorithms making QML models such as this one far more effective and pave the way for algorithms that aren’t constructed from existing classical machine learning algorithms and are instead unique with no classical counterparts. Overall, the QGAN has shown itself to be a competitive counterpart to classical GAN’s in terms of performance and further shows the future of Quantum Machine Learning as a viable alternative to Classical Machine Learning.

# References

1. Chen, S. Y.-C., Yoo, S., & Fang, Y.-L. L. (2022). Quantum long short-term memory. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. <https://doi.org/10.1109/icassp43922.2022.9747369>
2. Cerezo, M., & Arrasmith, A. (2021). Variational quantum algorithms. *Nature Reviews Physics*, *3*(9), 625–644. <https://doi.org/10.1038/s42254-021-00348-9>
3. Cong, I., Choi, S., & Lukin, M. D. (2019). Quantum Convolutional Neural Networks. *Nature Physics*, *15*(12), 1273–1278. <https://doi.org/10.1038/s41567-019-0648-8>
4. Goodfellow, I. J. (2023). Generative adversarial nets. *ArXiv*, 73–76. <https://doi.org/> <https://doi.org/10.48550/arXiv.1406.2661>
5. Yoon, J., Drumright, L. N., & van der Schaar, M. (2020). Anonymization through data synthesis using generative adversarial networks (ADS-gan). *IEEE Journal of Biomedical and Health Informatics*, *24*(8), 2378–2388. <https://doi.org/10.1109/jbhi.2020.2980262>
6. Lin, H. C., Chen, C., Huang, G. F., & Jafari, A. (2021). Stock price prediction using generative Adversarial Networks. *Journal of Computer Science*, *17*(3), 188–196. <https://doi.org/10.3844/jcssp.2021.188.196>
7. Dikshant Dulal, Yick Wei. (n.d.). *Dikshantdulal/softserve\_qlstm*. Retrieved January 3, 2023, from <https://github.com/DikshantDulal/SoftServe_QLSTM>
8. ChickenBenny. (n.d.). *Chickenbenny/stock-prediction-with-gan-and-WGAN: Stock prediction with Gan and WGAN*. GitHub. Retrieved April 11, 2023, from <https://github.com/ChickenBenny/Stock-prediction-with-GAN-and-WGAN>