A COMPARATIVE ANALYSIS OF QUANTUM NEURAL NETS ON IMBALANCE DATASET

A thesis
presented to the Faculty of the University of North Georgia
in partial fulfillment of the requirements for the degree of
Master of Science (M S.) in Computer Science

by

Sai Kundan Suddapalli

Dahlonega, Georgia
July 2024

Approval of the Thesis

A COMPARATIVE ANALYSIS OF QUANTUM NEURAL NETS ON IMBALANCE DATASET

This Thesis by Name of Candidate has been approved by the committee members below, who recommend it be accepted by the faculty of the University North Georgia partial fulfillment of requirements for the degree of

Master of Science in Computer Science

Thesis Committee:

Dr. Tamirat Abegaz, Ph.D., Chair Date

Dr. Bryson Payne, Ph.D. Date

Dr. Denise McWilliams, Ph.D. Date

Abstract

A COMPARATIVE ANALYSIS OF QUANTUM NEURAL NETS ON IMBALANCE DATASET

Sai Kundan Suddapalli

University of North Georgia

Traditional approaches to training neural networks often emphasize the necessity of relatively balanced datasets to ensure accurate and reliable outcomes. Despite the importance of balanced data, real-world scenarios are filled with imbalanced datasets. Many domains, from medical diagnostics to fraud detection, naturally produce datasets where certain classes are underrepresented. This imbalance can lead to biased models that perform well in the majority class while failing to accurately predict the minority class. The conventional methods of addressing this imbalance, such as data resampling and synthetic data generation, have their limitations and often do not fully resolve the underlying issues. Quantum computing, with its potential for computational speed-ups and unique data handling capabilities, could offer new pathways to address the limitations of Deep Neural Network (DNN) training. This research aims to explore an approach to overcoming the challenges posed by imbalanced data in machine learning by integrating quantum mechanics. The preliminary result indicates that current quantum models are struggling to match the efficiency of DNN approaches in both balanced and imbalanced datasets.

Dedication

This thesis is dedicated to my family, especially my parents, my brother and my grandparents. Their love, support, and sacrifices have been the foundation of my success.

To my father and mother, your belief in me has always been my greatest strength. To my brother, thank you for your encouragement and companionship. Your support has meant the world to me.

Thank you all for everything.

Acknowledgments

I would like to express my deepest gratitude to my committee members for their unwavering support and guidance throughout this journey:

- Dr. Tamirat Abegaz, Ph.D.
- Dr. Bryson Payne, Ph.D.
- Dr. Denise McWilliams, Ph.D.

Their expertise and encouragement have been invaluable to the completion of this work. Thank you for your commitment and dedication to my academic and personal growth.

Table of Contents

List of Tables

List of figures

CHAPTER 1: INTRODUCTION

Machine Learning and its Dependency on Data Quality

Classical machine learning algorithms, such as decision trees, support vector machines, and k-nearest neighbors, have been instrumental in solving a wide range of real-world problems (Sewak et al., 2018, Awais et al., 2021). These algorithms excel in tasks where the data is well-structured and the relationships between features are relatively simple to model. They are particularly effective in classification and regression tasks, making them valuable tools in areas like finance, healthcare, and marketing. However, classical machine learning algorithms have limitations when it comes to solving highly complex tasks that involve complex patterns.

Neural networks, particularly Deep Neural Networks (DNNs), have revolutionized machine learning by providing robust solutions for complex tasks such as image, text, and audio classification. The success of these models largely depends on the quality and quantity of the data used for training. A well-curated dataset that adequately represents all classes is crucial for the development of accurate and reliable models. However, real-world datasets often suffer from imbalances, where some classes are significantly underrepresented compared to others. This imbalance can severely affect the performance of neural networks, leading to biased predictions and reduced generalization capabilities (Goswami, 2020)

Limitation of DNNs motivate QNNs.

This limitations of DNNs in handling imbalanced datasets motivate the exploration of alternative models like Quantum Neural Networks (QNNs). (Beer, 2022). QNNs combine the power of neural networks with the principles of quantum mechanics. (Nielsen & Chuang, 2000). Nielsen & Chuang utilize quantum bits (qubits) that can exist in multiple states simultaneously (superposition), potentially offering advantages in dealing with imbalanced data.

Contributions to the Fields of AI and Quantum Computing

This research aims to explore an approach to addressing the challenges posed by imbalanced data in machine learning by integrating quantum mechanics. The primary research question guiding this thesis is: Does QNN improve accuracy when they are applied to imbalanced data? The findings indicate that QNNs, while promising, may not outperform classical neural networks in certain contexts, particularly in handling imbalanced data. Despite this, the study provides valuable insights into the potential and limitations of quantum computing in real-world machine learning tasks. This research highlights the importance of continued exploration in quantum machine learning to fully uncover its practical applications and benefits.

The thesis research is organized as follows: the introduction provides background information. A literature review is conducted to identify existing work on handling imbalanced data in DNNs. A methodology and implementation plan are developed, outlining the research methods, model development theory, hypothesis, and research question. The DNN and QNN models are defined and implemented, incorporating data preprocessing techniques and training procedures using appropriate frameworks and data processing methods. The performance of both models on imbalanced datasets is analyzed, and the results are interpreted to highlight the advantages and limitations of QNNs compared to DNNs. Finally, the findings are summarized, implications for future research discussed, and the study concluded.

CHAPTER 2: LITERATURE REVIEW

Balanced datasets are essential for neural network training as they ensure that the model learns to recognize and predict all classes accurately. When a dataset is balanced, each class contributes equally to the learning process, allowing the model to develop a comprehensive understanding of the problem domain. In contrast, imbalanced datasets can cause the model to become biased towards the majority class, ignoring or misclassifying the minority classes. This imbalance not only diminishes the accuracy of the model but also undermines its reliability and fairness, particularly in applications where accurate classification of minority classes is critical, such as medical diagnosis or fraud detection. (Chawla et al., 2002)

Underrepresented classes in imbalanced datasets pose a significant challenge for neural network models. The disparity in class representation can lead to several issues, including: (1) biased learning, where the model becomes biased towards the majority class, resulting in poor performance on the minority classes; (2) misclassification, where underrepresented classes are more likely to be misclassified, as the model lacks sufficient examples to learn their characteristics effectively; and (3) overfitting, which occurs when the model is overly specialized to the majority class, failing to generalize well to new, unseen data (Blohmke & Fetahu,. 2021)

Quantum Neural Networks (QNNs)

Liu and colleagues explored how incorporating randomness into the parameters of QNNs can significantly enhance their expressivity (Liu et al., 2023). Expressivity, in this context, refers to the ability of a QNN to approximate a wide variety of functions, crucial for solving complex computational problems. The study demonstrates through theoretical analysis and numerical simulations that QNNs with random initial parameters can produce a greater diversity of quantum states compared to deterministic ones. This increased diversity allows the QNNs to capture more intricate patterns within the data, leading to more robust and efficient quantum algorithms. The findings suggest that leveraging randomness in QNNs can help overcome limitations faced by classical neural networks, such as avoiding local minima and reducing the need for extensive training data, thereby advancing the field of quantum computing. (Liu et al.,2023). This could potentially help in improving the performance of learning algorithms in handling imbalanced datasets.

Current Challenges and Potential of QNN on imbalanced Data

While DNNs have achieved remarkable success, they struggle with imbalanced datasets, often exhibiting bias towards the majority class. This highlights the need for alternative models that can handle imbalanced data more effectively. However, a critical approach is necessary when evaluating claims about QNNs outperforming DNNs. According to Schetakis and colleagues research on the model, a Quantum Machine Learning (QML) architecture, suggests robustness to data noise. This noise tolerance could potentially contribute to effective classification even in scenarios with imbalanced datasets. (Shetakis et al.,2023*)*

A recent study by Arute and colleagues claims that QNN outperforms classic machine on small-scale benchmarks and may not translate to real-world scenarios (Arute et al., 2024). The report indicated that some reported improvements focus on specific aspects of QNNs (e.g., optimization algorithms) rather than overall model performance. Very few studies have demonstrably shown QNNs outperforming classical models, often citing limitations in current hardware as a reason for the gap. this research will examine the intersection of quantum mechanics and deep learning to overcome the limitations imposed by imbalanced data. While QNNs hold promise for overcoming challenges faced by DNNs, further research is needed to validate the potential advantages of QNNs for training imbalanced datasets.

CHAPTER 3: METHODOLOGY

To investigate the effectiveness of QNN in handling imbalanced data, this research will employ text, image, and audio classification tasks. Classification using quantum frameworks like PennyLane with TensorFlow Quantum (TFQ) or Qiskit. This research involves data preprocessing tailored for quantum deep learning, followed by the development, and testing of quantum neural network models. The primary research question guiding this investigation is whether Quantum Deep Neural Networks can effectively address the challenges of imbalanced data. The research hypothesis is that QNNs may offer a solution to the limitations of traditional neural network approaches when it comes to handling imbalanced datasets.

As shown in Figure 1, a classification task will be performed on balanced datasets for text, image, and audio data using both DNNs and QNNs. Since DNNs can struggle with imbalanced data, the performance of QNNs will be evaluated on imbalanced datasets with two split ratios: 80% majority class and 20% minority class, and 90% majority class and 10% minority class. In all these experiments, the focus will be on classifying images, text, and audio data, with accuracy as the primary dependent variable.

Table 1 summarizes the planned evaluation of classification accuracy for DNNs and QNNs across text, image, and audio data. It compares performance on both balanced datasets (equal class representation) and imbalanced datasets (uneven class distribution) with two split ratios (80%/20% and 90%/10%). Accuracy, represented by question marks ("?"), will be measured for each data type and model combination to determine which performs better under balanced and imbalanced conditions.

Figure 1

Implementation Map

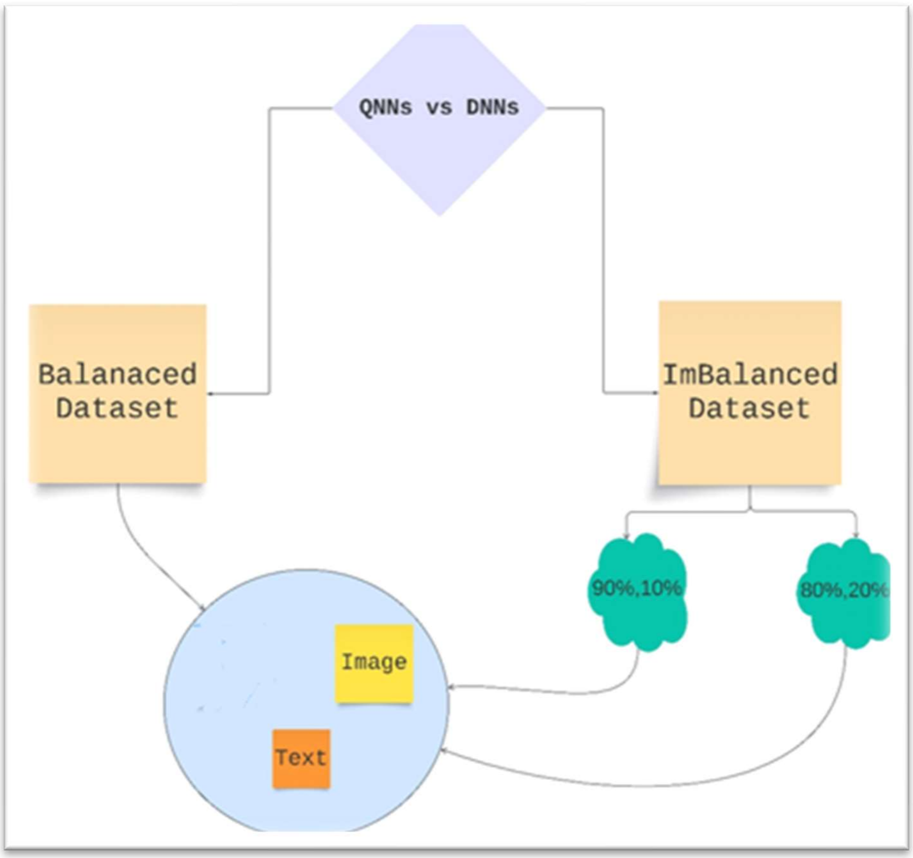


Table 1

The Research Design for Measuring Accuracy

Data type	DNN (Balanced)	QNN (Balanced)	QNN 80:20	QNN 90:10
Text	?	?	?	?
Image	?	?	?	?

Model Development and Theory

In this section, a theoretical framework for the development and evaluation of QNNs for handling imbalanced data. The theoretical foundation of this research lies in the principles of quantum mechanics and their potential application in deep learning. Quantum computing offers unique advantages, such as superposition, entanglement, and interference, which can potentially enhance the processing power and data handling capabilities of neural networks. By harnessing these quantum properties, QNNs may exhibit superior performance in handling imbalanced datasets, mitigating biases, and improving predictive accuracy.

Hypothesis

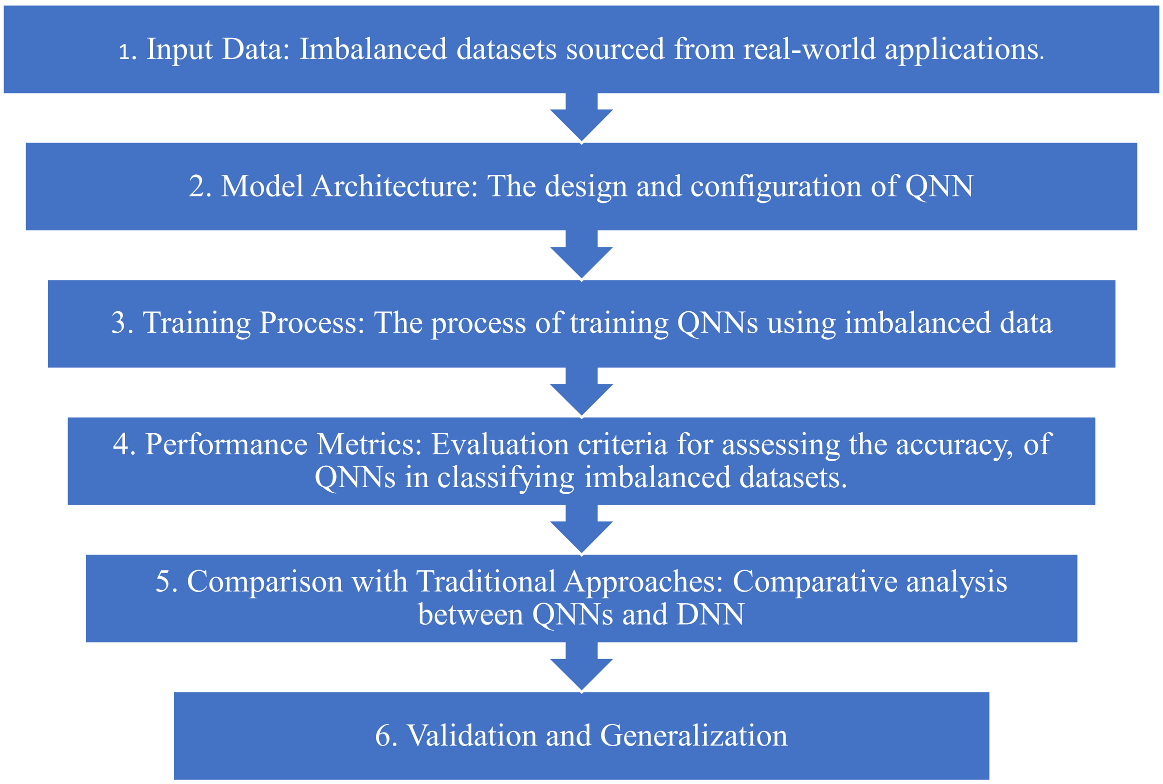
As stated, Quantum Neural Networks could potentially address the challenges posed by imbalanced data, resulting in improved accuracy and performance compared to traditional neural network approaches.

Conceptual Research Model

As shown in Figure 2, the conceptual research model outlines the key components and relationships underlying the development and evaluation of QNNs for imbalanced data. This flowchart explains using QNNs for real-world imbalanced data classification. In summary, QNN will be designed and trained on the data, then evaluated and compared to traditional models. Finally, its generalizability will be assessed across datasets.

Figure 2

Research Model Flow Chart



Hardware and Software requirement

To conduct a hybrid experiment on an 8-qubit simulator, The research setup requires specific hardware and software requirements. The hardware included an AMD Ryzen™ 9 5900X × 24 processor and 32GB of RAM, providing robust computational power to handle the demanding tasks involved in the experiment. The software environment was based on the Ubuntu operating system, which offered a stable and flexible platform for development and execution.

For the frameworks, utilized in research are mini-conda to manage the Python environments efficiently. The core libraries and frameworks included PyTorch for deep learning tasks, Qiskit 1.1.1 for quantum computing simulations, Pandas for data manipulation, Matplotlib for data visualization, and NumPy for numerical computations. These tools were essential in building and running the hybrid quantum-classical models used in the experiment.

All the setup details and the complete code for the experiment are available on my GitHub repository: https://github.com/Ironman20121/Quantum-on-imbalanced-data-.git. This repository includes comprehensive documentation and instructions to replicate the experiment, making it a valuable resource for anyone interested in exploring hybrid quantum-classical approaches to machine learning on imbalanced data.

Datasets

For the classification tasks, two distinct datasets used. The image data came from a multi-class fruit dataset available on Kaggle. This dataset includes a variety of fruit images used for developing and testing image classification models. For text classification SMS spam vs. ham classification dataset is also from Kaggle. This dataset contains SMS messages labeled as either spam or ham (non-spam), making it ideal for natural language processing tasks aimed at detecting spam messages.

Data Prepossessing for Text classification

The CSV data set was initially converted to UTF-8 encoding to ensure compatibility and proper handling of text characters. Subsequently, Label Encoding was applied to transform categorical labels into numerical representations, a necessary step for most machine learning algorithms. The dataset comprised 747 instances labeled as 'spam' and 4825 instances labeled as 'ham'. For the sake of balanced data, 700 samples were used for each class. For the imbalanced data, a ratio of 80:20 was used, resulting in 560 ham samples and 140 spam samples. For the 90:10 ratio, there were 630 ham samples and 70 spam samples.

Figure 3

Text Data First 10 Rows

[5]:

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

Data Prepossessing for Image classification

The dataset in this study is derived from the Kaggle Multi-Class Fruits dataset, encompassing ten distinct fruit categories: apple, banana, avocado, cherry, kiwi, mango, orange, pineapple, strawberries, and watermelon. The training partition consists of 254 images per fruit category, totaling 2,540 images. Conversely, the testing partition comprises 10 images per category, amounting to 100 images overall. Both traditional Neural Networks (NNs) and Quantum Neural Networks (QNNs) will be evaluated on this balanced dataset.

To further investigate the impact of class imbalance, additional experiments were conducted with varying ratios of apple images to other fruit categories. Specifically, two imbalance ratios were tested: 90:10 and 80:20. In both cases, the apple class was overrepresented, comprising 90% and 80% of the dataset, respectively, while the remaining nine fruit categories were equally distributed with 10% and 20% of the total images. These imbalanced datasets were processed using both QNN and NN models to assess the algorithms' performance under skewed class distributions.

Quantum circuits for convolution and pooling layers, combining them into an ansatz circuit, and transforming classical image data into a quantum format using the ZFeatureMap.

To transform classical image data into a quantum format, the ZFeatureMap is used. This quantum circuit encodes classical data into the amplitudes of a quantum state by applying parameterized gates based on the input features. In this code, the ZFeatureMap(8) maps classical data into an 8-qubit quantum state, preparing it for further quantum processing. The ansatz circuit, which combines the convolutional and pooling layers, is applied to the quantum state produced by the ZFeatureMap. This ansatz circuit is designed to extract and process the features encoded in the quantum state, like how classical neural networks process data. Figure 4 is a Radom Images from dataset which consist of 10 fruit classes.

Figure 4

Random Images From All 10 Classes

Architecture of 8Q bit classifier Circuits

The Z Feature Map is a technique used to convert classical data into a quantum format suitable for processing by quantum circuits. This is crucial because quantum algorithms operate on quantum bits (qubits) rather than classical bits. The Z Feature Map transforms classical data points into a superposition of quantum states, encoding the information in a way that leverages the principles of quantum mechanics, such as superposition and entanglement.

In this experiment used an 8-qubit Z Feature Map, meaning that the classical data is encoded into a quantum state using 8 qubits. This allows for a higher-dimensional representation of the data, potentially capturing more complex relationships within the data compared to classical representations.

Figure 5: This figure illustrates the structure of an 8-qubit feature map circuit. It shows how classical data is mapped onto a quantum state using 8 qubits. The figure should depict the quantum gates and operations involved in this transformation, demonstrating the process of encoding classical information into a quantum system.

An Ansatz in quantum computing is a trial wavefunction used as an initial guess for the quantum state in quantum algorithms, including quantum neural networks (QNNs). The choice of Ansatz is critical because it influences the efficiency and accuracy of the quantum algorithm. In the context of QNNs, the Ansatz serves as a starting point for the optimization process, where the goal is to find the best quantum state that represents the solution to a given problem.

Figure 6: This figure represents an 8-qubit Ansatz for a text classifier. It should show the configuration of quantum gates and the initial quantum state used to classify text data. The figure helps visualize how the Ansatz sets the stage for the quantum neural network to process and classify textual information.

Figure 5

ZFeature map

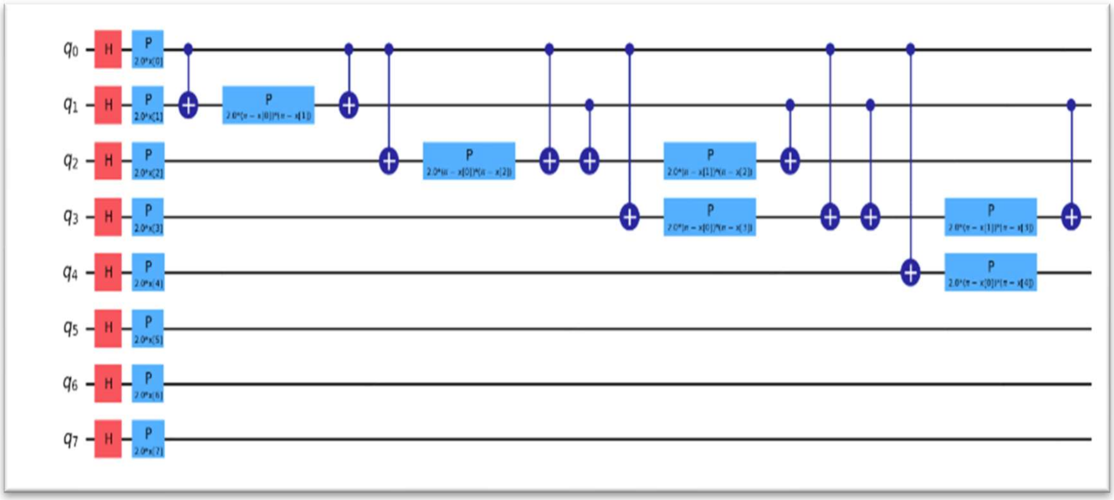


Figure 6
Ansatz

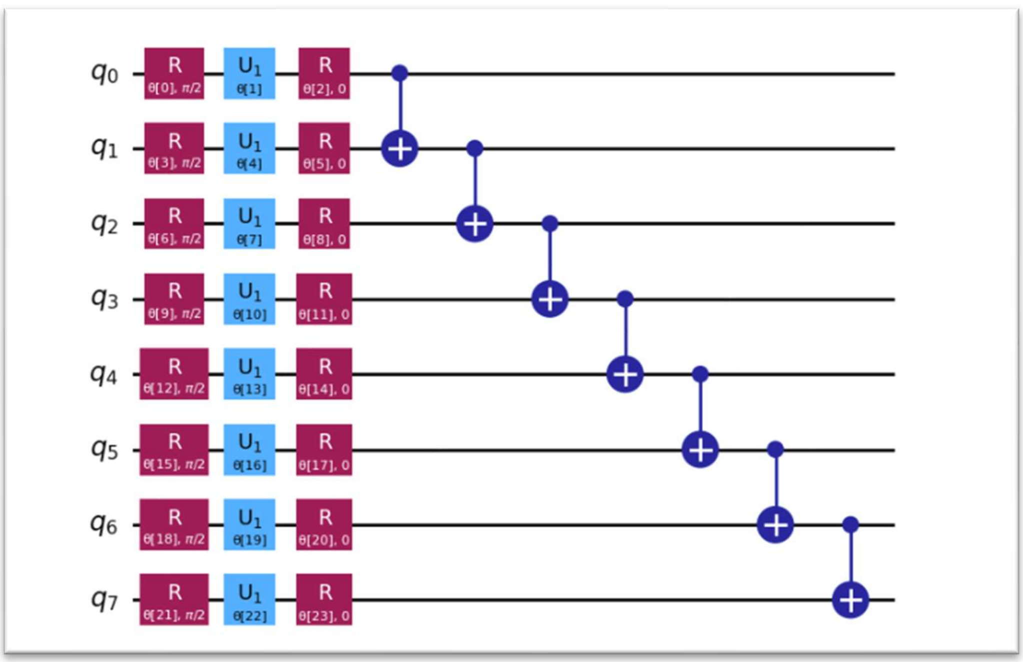
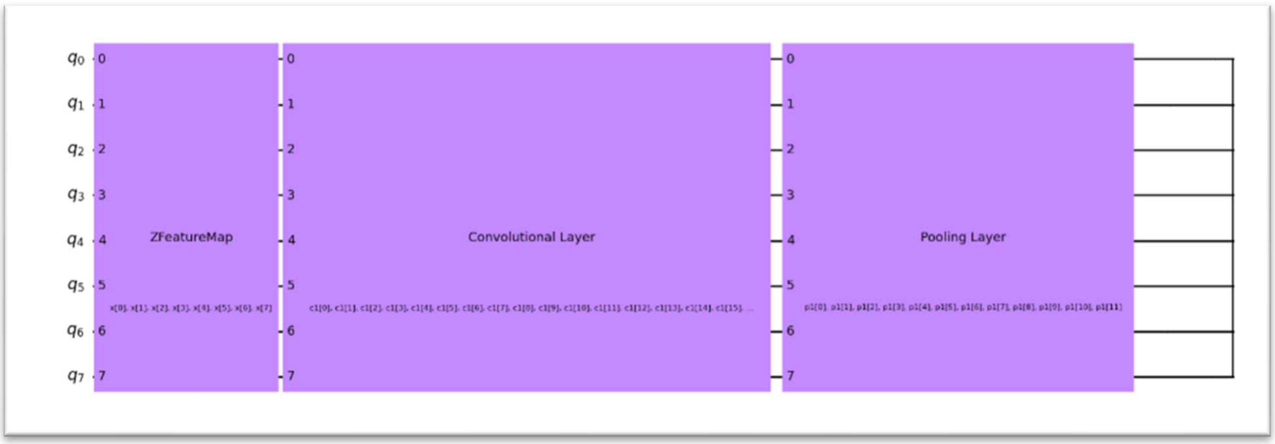


Figure 7

Multi Image Classifier Z Feature, Convolution layer and Pooling layer



Implementation

Text Classification using Quantum Neural Nets

This circuit is built to help classify emails as spam or ham (not spam). The circuit has two main parts: a feature map that converts your data into a quantum-friendly format, and an ansatz, which is a flexible quantum circuit that can be trained to make predictions.

In Figure 8, lines 51 and 52 depict the creation of a feature map and ansatz. Lines 53-57 introduce parameterized gates (ry, rz, rx) to add variability and flexibility to quantum operations during computation. Lines 58-60 technically create entanglement, which involves rotations or quantum spins around specific axes (X, Y, and Z). Finally, lines 60-63 combine all circuits to construct a quantum text classifier.

Figure 8

Quantum Circuit code

```
def create_circuit(self):
    feature_map = ZZFeatureMap(self.num_qubits)
    ansatz = QuantumCircuit(self.num_qubits)
    params = ParameterVector('θ', length=self.num_qubits * 3)
    for i in range(self.num_qubits):
        ansatz.ry(params[i*3], i)
        ansatz.rz(params[i*3+1], i)
        ansatz.rx(params[i*3+2], i)
    for i in range(self.num_qubits - 1):
        ansatz.cx(i, i+1)
    circuit = QuantumCircuit(self.num_qubits)
    circuit.compose(feature_map, inplace=True)
    circuit.compose(ansatz, inplace=True)
    return circuit, feature_map, ansatz
```

In Figure 9 create classifier function line 70 -74 Sets up a quantum neural network (QNN) using EstimatorQNN with the previously defined quantum circuit (circuit). . Initializes a NeuralNetworkClassifier using the qnn and a COBYLA optimizer, with a callback function (callback_graph) for plotting the objective function during training. In figure 9, line 63 observable one of the basic quantum operations, which measures the state of a qbit. When applied to all qbits, it helps in determining the output state of the entire quantum system, and this can help us drawing training graphs. Line 80 COBYLA (Constrained Optimization BY Linear Approximations) is a numerical optimization method which can be used for multi classification problems, maxiter=100 specifies the maximum number of iterations for the optimizer like epoch in classical machine learning

Figure 9

Text Quantum Deep Neural Net Classifier

```
    def create_classifier(self):
        circuit, feature_map, ansatz = self.create_circuit()
        # Define the observable
        observable = SparsePauliOp.from_list([("Z" * self.num_qubits, 1)])
        # Create the QNN
        qnn = EstimatorQNN(
            circuit=circuit,
            observables=observable,
            input_params=feature_map.parameters,
            weight_params=ansatz.parameters
        )
        # Create the classifier
        initial_point = np.random.random(len(ansatz.parameters))
        self.classifier = NeuralNetworkClassifier(
            qnn,
            optimizer=COBYLA(maxiter=100),
            callback=self.callback_graph,
            initial_point=initial_point
        )
```

Image Classification using Quantum Neural Nets

It is different from text classification need to create separate circuit for different layers, processing images creating circuits with specific bits to which overall combines uses 8 q-bits, creating a 2Q bit convolution layer In figure 10 line 21.Similary created a pooling circuit too.

In Figure 11 function conv_layer constructs a quantum circuit that mimics the behavior of a classical convolution layer. It does this by iteratively applying parametrized quantum gates to pairs of qubits, first to even-odd pairs and then to odd-even pairs, wrapping around as needed. The final quantum circuit, with barriers separating different operations, is converted into a reusable instruction and appended to a new circuit, which is then returned. This structure allows for the creation of complex quantum convolution layers useful for classifying tasks. Similarly, it is done pooling layer.

Initializing feature map and ansatz and create three convolution layers followed by pooling layer

Figure 10

Convolution Circuit Code

```
def conv_circuit(params):
    target = QuantumCircuit(2)
    target.rz(-np.pi / 2, 1)
    target.cx(1, 0)
    target.rz(params[0], 0)
    target.ry(params[1], 1)
    target.cx(0, 1)
    target.ry(params[2], 1)
    target.cx(1, 0)
    target.rz(np.pi / 2, 0)
    return target
```

Figure 11

Convolution Layer Code

```
def conv_layer(num_qubits, param_prefix):
    qc = QuantumCircuit(num_qubits, name="Convolutional Layer")
    qubits = list(range(num_qubits))
    param_index = 0
    params = ParameterVector(param_prefix, length=num_qubits * 3)
    for q1, q2 in zip(qubits[0::2], qubits[1::2]):
        qc = qc.compose(conv_circuit(params[param_index : (param_index + 3)]), [q1, q2])
        qc.barrier()
        param_index += 3
    for q1, q2 in zip(qubits[1::2], qubits[2::2] + [0]):
        qc = qc.compose(conv_circuit(params[param_index : (param_index + 3)]), [q1, q2])
        qc.barrier()
        param_index += 3

    qc_inst = qc.to_instruction()

    qc = QuantumCircuit(num_qubits)
    qc.append(qc_inst, qubits)
    return qc
```

In Figure 12 , We Entire implementation of of Quantum Convolution Neural Network other than create separate layer and circuits rest of implementation is similar to above text classifier.

Figure 12

Quantum Multi image classification model code

```python
def callback_graph(weights, obj_func_eval):…
def conv_circuit(params):…
def conv_layer(num_qubits, param_prefix):…
def pool_circuit(params):…
def pool_layer(sources, sinks, param_prefix):…
feature_map = ZFeatureMap(8)
ansatz = QuantumCircuit(8, name="Ansatz")
# First Convolutional Layer
ansatz.compose(conv_layer(8, "c1"), list(range(8)), inplace=True)
# First Pooling Layer
ansatz.compose(pool_layer([0, 1, 2, 3], [4, 5, 6, 7], "p1"), list(range(8)), inplace=True)
# Second Convolutional Layer
ansatz.compose(conv_layer(4, "c2"), list(range(4, 8)), inplace=True)
# Second Pooling Layer
ansatz.compose(pool_layer([0, 1], [2, 3], "p2"), list(range(4, 8)), inplace=True)
# Third Convolutional Layer
ansatz.compose(conv_layer(2, "c3"), list(range(6, 8)), inplace=True)
# Third Pooling Layer
ansatz.compose(pool_layer([0], [1], "p3"), list(range(6, 8)), inplace=True)
# Combining the feature map and ansatz
circuit = QuantumCircuit(8)
circuit.compose(feature_map, range(8), inplace=True)
circuit.compose(ansatz, range(8), inplace=True)
observable = SparsePauliOp.from_list([("Z" + "I" * 7, 1)])
qnn = EstimatorQNN(
    circuit=circuit.decompose(),
    observables=observable,
    input_params=feature_map.parameters,
    weight_params=ansatz.parameters,
)
initial_point = np.random.rand(len(qnn.weight_params))
classifier = NeuralNetworkClassifier(
    qnn,
    optimizer=COBYLA(maxiter=200),
    callback=callback_graph,
    initial_point=initial_point,
)

objective_func_vals = []
plt.rcParams["figure.figsize"] = (12, 6)
classifier.fit(x_train, y_train)
train_accuracy = classifier.score(x_train, y_train)
test_accuracy = classifier.score(x_test, y_test)
print(f"Accuracy from the train data: {np.round(100 * train_accuracy, 2)}%")
print(f"Accuracy from the test data: {np.round(100 * test_accuracy, 2)}%")
```

CHAPTER 4: FINDINGS OR RESULTS

Text Classification

Table 2 presents the results of balanced data classification, where DNNs achieved an accuracy of 97% in 5 minutes, while QNNs significantly underperformed with an accuracy of only 12%, taking 3 hours. Table 2 summarizes the imbalanced data classification outcomes, showing that DNNs maintained a high accuracy of 95% within 5 minutes. In contrast, QNNs achieved only 10% accuracy with an 80:20 imbalance ratio and 11% accuracy with a 90:10 imbalance ratio, both requiring 3 hours. These results highlight the inefficacy of QNNs in comparison to DNNs for both balanced and imbalanced datasets, indicating that QNNs are currently not a viable alternative for these classification tasks.

***Table* 2**

Balanced Data Text Classification

Models	Accuracy	Time
DNNs	97%	5 min
QNNs	12%	3 hr

Table 3
Imbalanced Data Text Classification

Models	Accuracy	Time
DNNs 90:10	79%	5 min
QNNs 80:20	10%	3 hr
QNNs 90:10	11%	3 hr

Figure 13

Quantum Training Graphs for Balanced Data

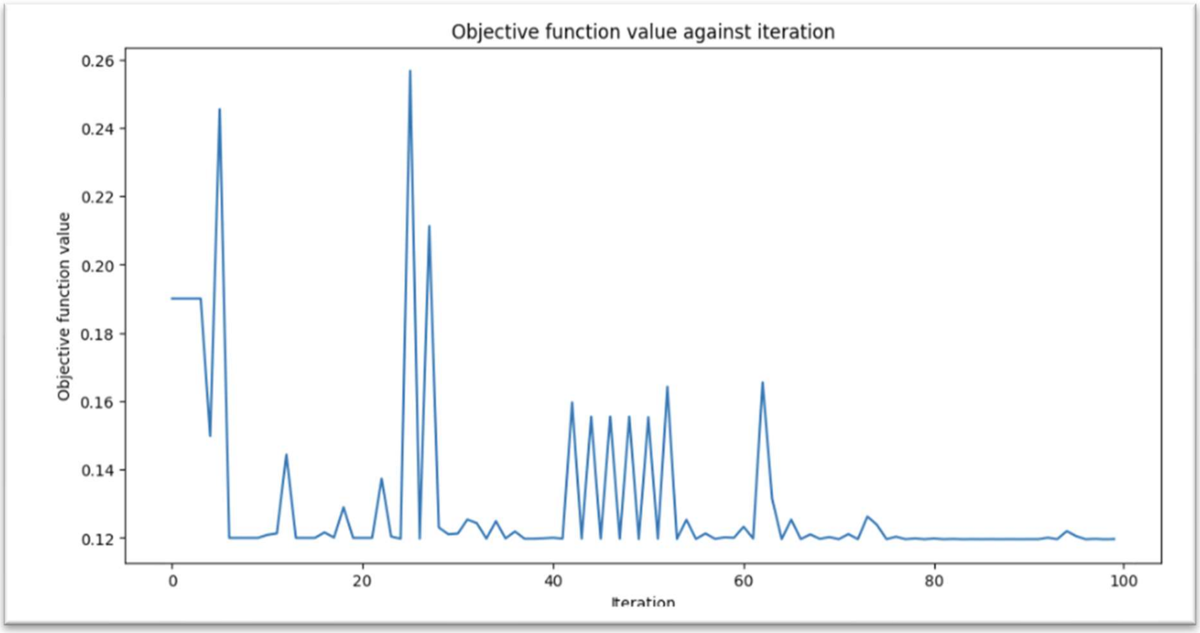


Image Classification

Tables 4 and 5 illustrate the performance disparity between Deep Neural Networks (DNNs) and Quantum Neural Networks (QNNs) in image classification tasks for both balanced and imbalanced data sets.

In Table 4, which focuses on balanced data image classification, DNNs achieved a commendable accuracy of 90% within just 5 minutes. On the other hand, QNNs managed only an 11% accuracy and took more than a day to process the same data.

Table 4

Balanced Data Image Classification

Models	Accuracy	Time
DNNs	90%	5 min
QNNs	11%	> 1 Day

Table 5 highlights the results for imbalanced data image classification. Here, DNNs still performed relatively well, with a 78% accuracy and a processing time of 5 minutes. However, QNNs showed poor performance, with just 9% accuracy for an 80:20 imbalance ratio and 10% accuracy for a 90:10 imbalance ratio, each taking over a day to complete the task.

These results clearly indicate that, despite the promise of quantum computing, QNNs are currently not effective for image classification, whether the data is balanced or imbalanced. In contrast, DNNs consistently deliver high accuracy in a fraction of the time, making them the superior choice for these tasks at present.

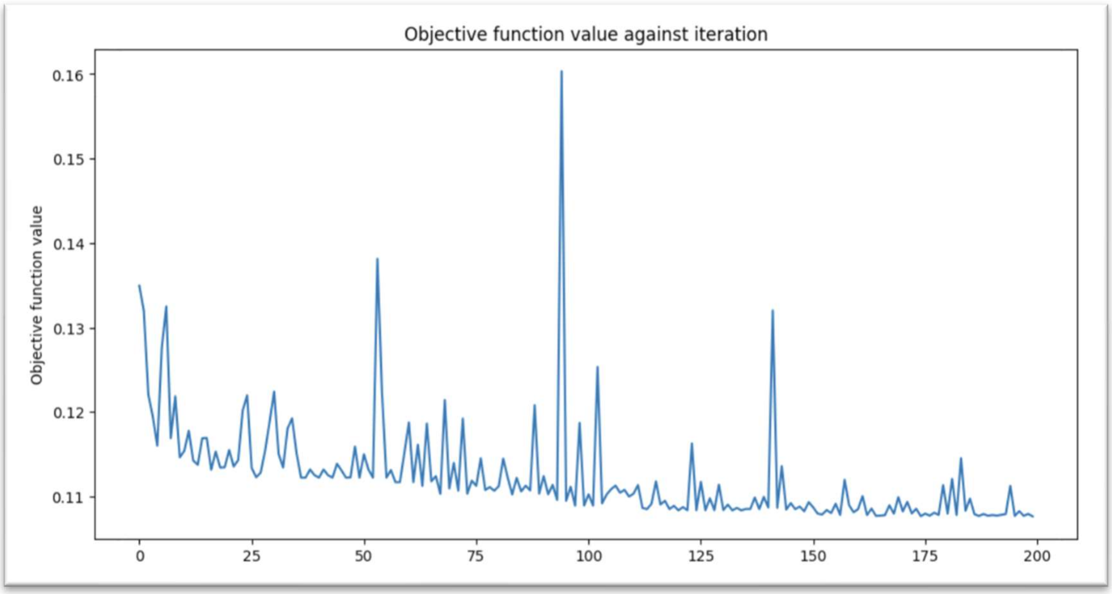
Table 5

Imbalanced Data Image Classification

Models	Accuracy	Time
DNNs 90:10	67%	5 min
QNNs 80:20	9%	> 1 Day
QNNs 90:10	10%	> 1 Day

Figure 14

Quantum Training Graphs for Balanced Data



CHAPTER 5: DISCUSSION

Table 6 presents an updated accuracy comparison across various models and data types. For text data, the Deep Neural Network (DNN) with balanced data achieves a high accuracy of 97%, significantly outperforming the Quantum Neural Network (QNN) models, which show much lower accuracies (12% for balanced, 10% for 80:20, and 11% for 90:10 splits). Similarly, for image data, the DNN model achieves 90% accuracy, whereas the QNN models perform notably worse with accuracies ranging from 11% to 12%. This table highlights the considerable disparity in performance between traditional DNNs and QNNs across different data splits, suggesting that DNNs currently have a more effective approach for these tasks.

Table 6

Updated Accuracy Table

Data type	DNN (Balanced)	QNN (balanced)	QNN 80:20	QNN 90:10
Text	97%	12%	10%	11%
Image	90	11%	9%	12%

Problem and Reasons for QNNs Not Working Well

The limited use of only 8 qubits appears inadequate for the problem's complexity, as increasing the number of qubits could enhance performance by enabling more intricate quantum states and computations. Additionally, the current feature map and ansatz might not be effectively translating classical data into quantum data, suggesting a need for further optimization in these areas to improve quantum data representation and QNN performance. Future efforts should focus on refining these components or developing new techniques for better quantum data encoding. Moreover, direct implementation on quantum hardware might be impractical due to the expertise required and the current limitations of quantum technology. Hybrid computing, which combines classical and quantum computations, could offer a more feasible solution by utilizing classical computers for certain tasks and quantum processors for others, potentially leading to improved performance.

Conclusion

Based on the results, it is evident that Deep Neural Networks (DNNs) significantly outperform Quantum Neural Networks (QNNs) in both text and image data classification tasks. The DNN's high accuracy, particularly in text classification, highlights its effectiveness in handling complex data with balanced datasets. In contrast, the QNN's performance remains substantially lower, indicating that the current quantum models are struggling to match the efficiency of classical approaches.

Several factors contribute to the underperformance of QNNs. The limited number of qubits, currently set at 8, restricts the quantum model's ability to process and represent complex data effectively. Additionally, the feature map and ansatz used may not be optimized for translating classical data into quantum data, further impeding the QNN's accuracy. These technical limitations suggest that improvements in quantum data encoding and model design are necessary to enhance the QNN's performance.

To address these issues and improve QNN performance, future work should focus on increasing the number of qubits and optimizing the feature map and ansatz. Exploring hybrid computing approaches, which combine classical and quantum methods, could also offer a practical solution by leveraging the strengths of both computational paradigms. Such advancements are crucial for advancing quantum computing and enabling QNNs to achieve comparable or superior results to their classical counterparts.

REFERENCES

1. Sewak, M., Sahay, S. K., & Rathore, H. (2018, June). Comparison of deep learning and the classical machine learning algorithm for the malware detection. In 2018 19th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD) (pp. 293-296). IEEE.
2. Awais, M., Chiari, L., Ihlen, E. A., Helbostad, J. L., & Palmerini, L. (2021). Classical machine learning versus deep learning for the older adults free-living activity classification. Sensors, 21(14), 4669.
3. Goswami, S. (2020, January 20). *Impact of Data Quality on Deep Neural Network Training. https://arxiv.org/abs/2002.03732*
4. Beer, K. (2022, May 17). *Quantum Neural Networks. https://arxiv.org/abs/2205.08154*
5. Nielsen, M. A., & Chuang, I. L. (2000). *Quantum computation and quantum information. Cambridge, UK: Cambridge University Press.*
6. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). S*MOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357. https://arxiv.org/pdf/1106.1813*
7. Blohmke, S., & Fetahu, B. (2021, January). *Fairness in machine learning: A survey. https://arxiv.org/abs/2010.04053*
8. Rehák, M., Semančíková, J., Fiedler, J., & Železný, J. (2020). Impact of *Data Quality on Machine Learning Algorithms. https://arxiv.org/abs/2002.03732*
9. Beer, K., Bondarenko, D., Farrelly, T., Osborne, T. J., Salzmann, R., & Wolf, R. (2022). *Training deep quantum neural networks. https://arxiv.org/abs/2205.08154*
10. Schetakis, N., Boguslavsky, M., & Griffin, P. (2023). *Quantum Machine Learning for Imbalanced Data Classification. https://arxiv.org/abs/2308.10847*
11. Liu, Y., Zhang, X., & Zhang, J. (2023). *Randomness-Enhanced Expressivity of Quantum Neural Networks. https://arxiv.org/abs/2308.04740*
12. Arute, M., Babbush, R., Bardin, J. C., Barends, R., Boixo, S., Broughton, M., ... & Roushan, P. (2024). *Better than classical? The subtle art of benchmarking quantum machine learning models. https://arxiv.org/abs/2403.07059*
13. Schetakis, N., Aghamalyan, D., Boguslavsky, M., Rees, A., Raktomalala, M., & Griffin, P. (2023, August 7). *Quantum Machine Learning for Credit Card Score. https://arxiv.org/abs/2308.03575*