Binary Classification with QNNs

Leveraging Qiskit's Abstraction Barrier

ALBERTO HOJEL[1], HARRISON RESNICK[1], AND OWEN SLEIGH[1]

[1] *UC Berkeley, equal contribution*

Compiled August 4, 2023

Quantum Machine Learning (QML) is an interdisciplinary field that aims to leverage the power of quantum computing to enhance machine learning algorithms. In this research paper, we explore the use of QML for classification tasks using the Variational Quantum Classifier (VQC). The VQC is a quantum machine learning algorithm that learns to classify data by encoding it into the amplitudes of a quantum state and then training an ansatz circuit to optimize a cost function that minimizes the difference between the predicted and true labels of the data. To implement the VQC algorithm in our research, we use the Qiskit framework, an open-source platform for developing quantum software that provides pre-built components for VQC, among other algorithms. Our study uses a simple two-dimensional dataset to test the performance of the VQC, where each sample is labeled as either 1 or -1 based on whether the sum of its input features is positive or negative. We construct a quantum circuit using Qiskit, which consists of a quantum feature map and an ansatz circuit. We use the Zero-Pi-Pulse (ZZ) feature map and the Real Amplitudes circuit as our quantum feature map and ansatz circuit, respectively. We use the EstimatorQNN class in Qiskit to define our quantum neural network model and the NeuralNetworkClassifier class to train and evaluate our VQC model. Finally, we analyze the weights of the VQC model using the weights attribute of the NeuralNetworkClassifier class. Our study demonstrates the feasibility of using QML for classification tasks and highlights the potential of the VQC algorithm for machine learning applications.

1. INTRODUCTION

Quantum computing has emerged as a revolutionary paradigm, with the potential to reshape the landscape of computational power and efficiency. Its inherent ability to exploit quantum effects such as superposition, entanglement, and quantum tunneling enables it to address complex and computationally demanding problems that remain intractable for classical computers. Among the plethora of applications that can benefit from quantum computing, machine learning (ML) stands out as a promising domain that could witness significant advancements.

Machine learning, a subfield of artificial intelligence, has seen tremendous growth in recent years, owing to its ability to extract pat-

terns and make predictions from large datasets. Classical machine learning algorithms, however, are hindered by scalability issues and increasing data dimensions. Quantum Machine Learning (QML), an interdisciplinary field that merges the power of quantum computing and machine learning, aims to overcome these challenges and redefine the frontiers of ML applications.

QML's potential to outperform classical ML techniques has sparked considerable interest in developing quantum-based algorithms and models, such as Quantum Neural Networks (QNNs). QNNs offer a quantum analogue to classical neural networks, relying on quantum circuits for processing and optimization. The advent of Noisy Intermediate-Scale Quantum (NISQ) systems and their associated programming interfaces has enabled researchers to explore QNNs and their applicability to real-world ML tasks.

This research paper aims to delve into the experimental aspect of QML by leveraging the abstraction barrier provided by Qiskit[1], an open-source quantum software development platform. The primary focus of this study is to implement a binary classification task using a Variational Quantum Classifier (VQC) algorithm. VQC is a QML algorithm that encodes data into the amplitudes of a quantum state and trains an ansatz circuit to optimize a cost function that minimizes the difference between predicted and true labels of the data.

Our experiment uses a simple two-dimensional synthetic dataset, where each data point is labeled as either 1 or -1 based on the sum of its input features. By employing the Zero-Pi-Pulse (ZZ) feature map and the Real Amplitudes circuit as our quantum feature map and ansatz circuit respectively, we construct a quantum circuit using Qiskit. We then use the EstimatorQNN class in Qiskit to define our quantum neural network model and the NeuralNetworkClassifier class to train and evaluate our VQC model.

In doing so, this paper seeks to contribute to the understanding of QML's feasibility for classification tasks and shed light on the potential of VQC algorithms in machine learning applications. By examining the performance of the VQC model on the synthetic dataset, we hope to provide insights into the training process of QNNs and contribute to the ongoing discussion surrounding quantum advantage in machine learning.

As we embark on this exciting journey into the realm of QML, we invite the reader to delve deeper into the paper's methodology and results sections, where we meticulously describe the im-

plementation details and analyze the outcomes of our experiment. Through this investigation, we aspire to enrich the scientific community's knowledge of QML and stimulate further research in this nascent yet promising field.

2. PREVIOUS WORKS

The field of Quantum Machine Learning (QML) has garnered substantial interest in recent years, as researchers explore the potential of quantum computing to enhance traditional machine learning algorithms. Numerous studies have focused on the development and application of Quantum Neural Networks (QNNs) and Variational Quantum Classifiers (VQCs) to solve various machine learning problems. In this section, we provide an overview of the current state of QML research, with a focus on the contributions made by different works in the field.

Several studies have concentrated on the development and analysis of quantum classifiers. Li and Deng [2] offered a comprehensive review of recent advances in quantum classifiers, covering various algorithms such as quantum support vector machines, quantum kernel methods, quantum decision tree classifiers, quantum nearest neighbor algorithms, and quantum annealing-based classifiers. They also discussed the architectures of variational quantum classifiers and the barren plateau problem, as well as the vulnerability of quantum classifiers in adversarial learning and recent experimental progress. In a similar vein, Blance and Spannowsky [3] introduced a novel hybrid variational quantum classifier that combines quantum gradient descent with steepest gradient descent for parameter optimization. Their work demonstrated superior learning outcomes compared to classical neural networks and quantum machine learning methods that employ non-quantum optimization methods.

In the realm of QNNs, Markidis et al. [4] presented an extensive overview of programming QNNs on Noisy Intermediate-Scale Quantum (NISQ) systems. Their work surveyed state-of-the-art high-level programming approaches for QNN development, addressing target architectures, critical QNN algorithmic components, hybrid workflows, QNN architectures, optimizers, gradient calculations, and applications. This study provides a valuable understanding of existing programming QNN frameworks, their software architecture, and associated quantum simulators.

Another group of studies has focused on applying quantum classifiers to specific problems or datasets. Maheshwari et al. [5] employed a

Variational Quantum Classifier (VQC) for binary classification using real and synthetic datasets. They introduced a pre-processing method to enhance the prediction rate when applying the VQC method, including feature selection and state preparation. The authors reported considerable improvements in classification accuracy by utilizing amplitude encoding-based VQC compared to the standard VQC model. Similarly, Sierra-Sosa et al. [6] applied VQC to predict dementia in elderly patients and demonstrated that the VQC implemented in IBM's Qiskit framework outperformed a classical Support Vector Machine (SVM) with a linear kernel in terms of consistency when using different numbers of features.

Chen [7] explored Deep Reinforcement Learning using Hybrid Quantum Neural Networks. They designed a parameterized quantum circuit (PQC) to address a model-free reinforcement learning problem using the deep-Q learning method. The research compared the performance of a PQC-based model with classical deep neural networks (DNNs) with and without integrated PQCs, providing insights into the potential of quantum advantage on current quantum computers and the prospects for developing deep quantum learning in reinforcement learning problems.

In conclusion, the literature on QML has witnessed rapid progress, particularly in the areas of QNNs and VQCs. Researchers have investigated various quantum classification algorithms, optimization techniques, and hybrid approaches to leverage the power of quantum computing for machine learning tasks. These studies contribute to an improved understanding of the current limitations and potential advantages of QML, setting the stage for further advancements in the field. Our work aims to build on these achievements and explore new possibilities in the domain of quantum machine learning.

3. METHODOLOGY

Quantum Machine Learning (QML) is a rapidly growing interdisciplinary field that leverages the power of quantum computing to enhance machine learning algorithms. QML offers a novel approach to solving machine learning problems by exploiting the intrinsic properties of quantum systems to encode and process information in a fundamentally different way than classical computing. In this research paper, we explore the use of QML for classification tasks using the Variational Quantum Classifier (VQC). The VQC is a quantum machine learning algorithm that learns to classify data by encoding it

into the amplitudes of a quantum state, and then training an ansatz circuit to optimize a cost function that minimizes the difference between the predicted and true labels of the data. The VQC consists of two key components: a quantum feature map and an ansatz circuit. The quantum feature map encodes the input data into a quantum state by applying a set of unitary operations to the initial state. The ansatz circuit, on the other hand, is a parameterized quantum circuit that is trained to output the correct labels of the data.

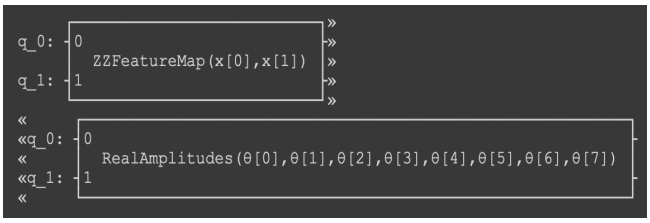

Fig. 1. The quantum circuit used within the EstimatorQNN. It includes both ZZFeatuerMap and ansatz amplitudes.

To implement the VQC algorithm in our research, we use the Qiskit framework, which is an open-source platform for developing quantum software. Qiskit provides a comprehensive set of tools and libraries that make it easy to create and simulate quantum circuits on classical computers. To use Qiskit for QML, we install the qiskit_machine_learning library which provides pre-built components for VQC, among other algorithms. To test the performance of the VQC, we use a simple two-dimensional dataset of 15 samples—we generate the data using NumPy and plot it using Matplotlib.

We then construct a quantum circuit using Qiskit that consists of a quantum feature map and an ansatz circuit. The quantum feature map we use is the Zero-Pi-Pulse (ZZ) feature map, which is a simple parameterized circuit that applies controlled-Z gates between each pair of qubits in the input state. The ansatz circuit we use is the Real Amplitudes circuit, which is a parameterized circuit that applies rotations and entangling gates to each qubit in the circuit. We combine these two circuits to form our complete quantum circuit. Next, we use the EstimatorQNN class in Qiskit to define our quantum neural network model. EstimatorQNN takes as input our quantum circuit, the parameters of the quantum feature map, and the parameters of the ansatz circuit, and returns a quantum circuit that can be used to compute the output of the neural network. To train our VQC model, we use the NeuralNetworkClassifier class in Qiskit, which takes as input our quantum neural network model, an optimizer, and a callback function for monitoring the training progress. We use the COBYLA optimizer, which is a constrained optimization algorithm that is well-suited for optimizing black-box func-

tions, and we define a callback function that plots the value of the objective function (i.e., the cost function) at each iteration of the optimization. After training our VQC model, we evaluate its performance by calculating its accuracy on the training data using the score() method of the NeuralNetworkClassifier class. We also use the predict() method to make predictions on the data and plot the results using Matplotlib. In the plot, correctly classified data points are represented by blue and gold dots, while misclassified data points are circled in red. Finally, we analyze the weights of the VQC model using the weights attribute of the NeuralNetworkClassifier class. The weights represent the parameters of the ansatz circuit that have been learned during training. We can visualize the learned parameters using Matplotlib to gain insight into how the VQC is making its predictions. By analyzing the learned parameters, we can identify patterns in the data that the VQC has learned to exploit in order to make accurate predictions.

4. RESULTS

When training the EstimatorQNN, we used a 15 point 2D dataset as seen in figure 2. Two dimensions was a natural choice so that accuracy could be easily visualized and it kept runtimes of training lower, as it was a lot less complex. Having only 15 data points was the result of similar testing, as higher data points (100-200) took a lot longer to run and were less accurate. Additionally, other runs with a non-linearly separable training set did not yield very good accuracy, showing the limitations of current QML.

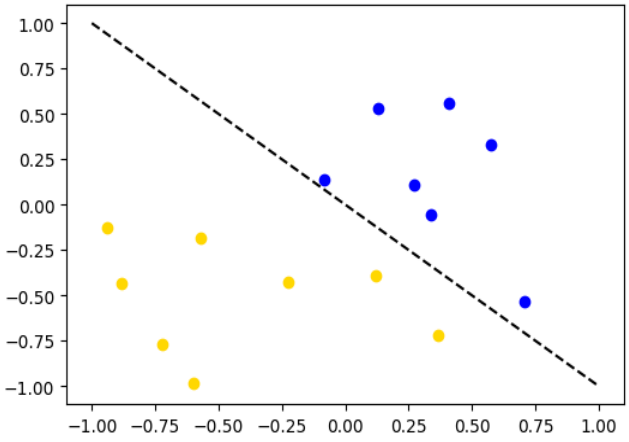

Fig. 2. The toy dataset used to train and test the EstimatorQNN. Note that it is linearly separable.

To train the data set we used the built in Qiskit COBYLA (Constrained Optimization By Linear Approximation) optimization algorithm to minimize the loss and graphed the loss vs iteration number, as seen in figure 3. COBYLA is a numerical optimization algorithm that doesn't need derivatives, making it ideal for a neural net, since 1000s of optimization steps are necessary. Different optimization algorithms could have been more effective or faster, but COBYLA is fairly standard and what Qiskit recommends.

We opted for 100 iterations of the neural net to ensure that the loss converged. Since we ended up using fewer data points, it converged a bit quicker than 100 iterations, but other tests with more data points took longer to converge. Also note that the value of the objective function spiked at around iteration 20. This occurred for almost all sets of data points/iterations. It would be interesting to further research why this occurs, and if it's a product of the optimization function, if it's unique to quantum machine learning, and if it can be resolved easily. However, as the objective value eventually did converge, we did not pay to much attention to it.

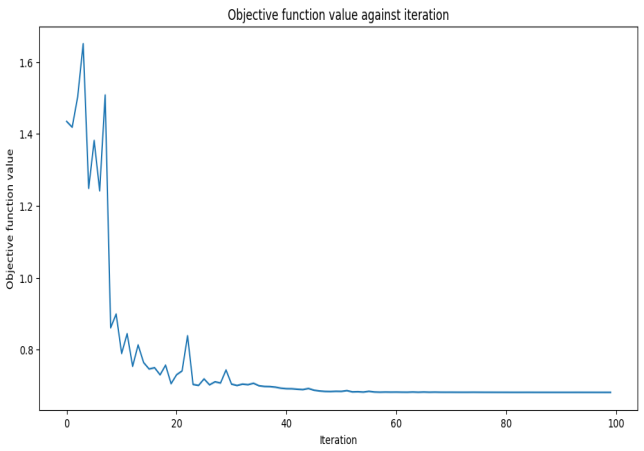

Fig. 3. Figure 2: A graph of the optimization of the loss value. Note that it converges at around iteration 50 and the spike at around iteration 20.

Testing the accuracy of the EstimatorQNN was simple, as we just had to compare predicted labels vs the actual labels that we created. We graphed the results with incorrectly labeled data points circled in red, as seen in figure 4 and found that we had a 66% accuracy, a little bit better than flipping a coin. The classifier tended to get points around the boundary wrong which makes a lot of sense, as this is where classical machine learning algorithms tend to have errors too. It was interesting to see that as the amount of data points increased, the accuracy actually went down in some cases, and the algorithm tended to misclassify a lot of points in one class, instead of misclassifying the two classes evenly. This trend can be seen even with the 15 point dataset, where only three of the gold points got classified correctly, while there are 7 blue points classified correctly. Another interesting result was that increasing the distance between the two classes didn't increase the accuracy of the classifier by that much. In tests with two distinct blobs, the classifier would still sit at around 75% accuracy. It would be the expectation that more separated groups would largely increase the accuracy, but that did not happen. It would be interesting research why this is the case. Perhaps increasing the amount of features, or changing the ansatz could yield higher accuracy, but more research is required to answer this question.

Qiskit has tested out their QNN implementations on real data sets, including a dataset classifying different species of Iris's. They found

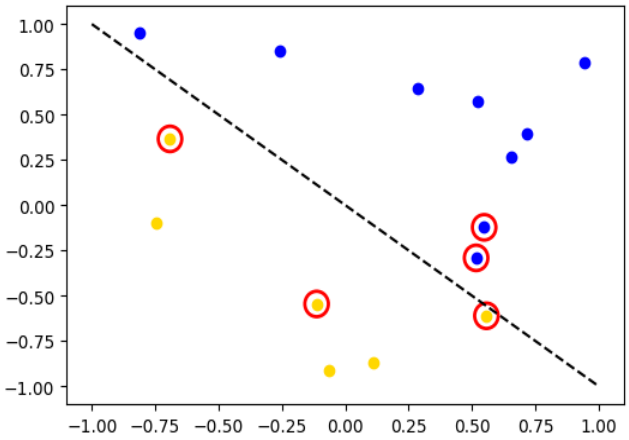

Fig. 4. Figure 3: the accuracy of classified points, incorrectly classified points are highlighted in red and tend to sit around the boundary line.

that the quantum neural net took slightly longer to train, and did not have as good of accuracy compared to its classical counterpart. The quantum version with 2 features was able to achieve a 58% accuracy, while the classical version had a 97% accuracy. They also found that increasing the amount of features, increased the quantum accuracy to 85% and the classical accuracy to 99%, however this did double the amount of qubits necessary to train the data (Qiskit, n.d.). This is impressive, as quantum machine learning is still in its early stages of development and classical machine learning has had decades of improvements to make it the powerhouse that it is today. Perhaps after quantum computers have advanced enough, they can actually leverage the speedup over classical computers, but for now classical machine learning has quantum machine learning beat.

5. DISCUSSION

Quantum Machine Learning (QML) has generated significant interest and investment, fueled by the promise of quantum advantage in various applications. However, it is important to recognize that much of this enthusiasm comes from individuals who may not fully comprehend the underlying technology. As a result, questions surrounding the practicality and potential of quantum advantage remain.

In our paper, we explored a QML technique on a synthetic dataset. While a classical neural network would perform well on this task, we aimed to investigate the capabilities of a Quantum Neural Network (QNN) in a simulated environment, rather than running it on actual quantum hardware. This approach raises the question of whether our findings are truly useful or representative of the potential of QNNs in real-world applications.

The current state of QML research is still evolving, with no definitive answer as to when, or even if, quantum advantage will be achieved. Our study adopted an experimental approach, leveraging the abstraction barrier of Qiskit to reduce the barrier of entry into the field of

Quantum Computing (QC). This has allowed researchers and enthusiasts to create quantum circuits, simulate their performance, and even run them on real quantum hardware with relative ease.

Although the abstraction barrier has facilitated access to QC technology, it also enables users to work with quantum circuits without fully understanding the intricacies of the underlying primitives, such as those of the EstimatorQNN. While this may simplify the process of utilizing QML techniques, it could potentially hinder the development of deeper insights and innovations that arise from a thorough comprehension of the technology.

Nonetheless, the increased accessibility to QC technology, enabled by Qiskit's abstraction barrier, has fostered a more inclusive research environment. It allows a broader range of individuals to explore their curiosity and contribute to the field, ultimately propelling the development of QML forward.

In conclusion, the future of QML and the achievement of quantum advantage remain uncertain. However, the increasing accessibility of QC technology, facilitated by tools such as Qiskit, has created a fertile ground for further exploration and experimentation. Whether or not QML will ultimately fulfill its promise, the opportunity for researchers and enthusiasts to engage with this cutting-edge technology and expand the boundaries of human knowledge is an invaluable outcome in itself.

6. CONTRIBUTION STATEMENT

As a group, Harrison, Alberto, and Owen collaborated throughout the exploration, leveraging each other's unique skillset. This supportive environment is what allowed them to complete the research paper. Harrison wrote the abstract and the methodology and helped proofread and format the research paper. Alberto focused on developing the introduction, discussion, and previous work sections. He also implemented a LaTeX template to promote readability. Owen worked on playing around with the code to explore different hyperparameters for the neural network and wrote the main results section.

REFERENCES

1. Q. D. Team, "Neural network classifier and regressor - qiskit machine learning," (2021).
2. W. Li and D.-L. Deng, "Recent advances for quantum classifiers," Sci. China Physics, Mech. & Astron. **65**, 220301 (2022).
3. A. Blance and M. Spannowsky, "Quantum machine learning for particle physics using a variational quantum classifier," J. High Energy Phys. **2021**, 1–20 (2021).
4. S. Markidis, "Programming quantum neural networks on nisq systems: An overview of technologies and methodologies," Entropy. **25** (2023).
5. D. Maheshwari, D. Sierra-Sosa, and B. Garcia-Zapirain, "Variational

quantum classifier for binary classification: Real vs synthetic dataset," IEEE Access **10**, 3705–3715 (2021).

6. D. Sierra-Sosa, J. Arcila-Moreno, B. Garcia-Zapirain, C. Castillo-Olea, and A. Elmaghraby, "Dementia prediction applying variational quantum classifier," arXiv preprint arXiv:2007.08653 (2020).
7. H.-Y. Chen, "Deep reinforcement learning using hybrid quantum neural network," arXiv preprint arXiv:2304.10159 (2023).