Performance Analysis of Quantum Machine Learning Classifiers

Tonni Das
Department of Computer Science,
School of Engineering and Computer Science.
Baylor University, Waco, TX, USA
Tonni_Jui1@baylor.edu

Olawale Ayoade
Department of Physics,
College of Arts & Sciences.
Baylor University, Waco, TX, USA
Olawale_Ayoade1@baylor.edu

Pablo Rivas
Department of Computer Science,
School of Engineering and Computer Science.
Baylor University, Waco, TX, USA
Pablo_Rivas@baylor.edu

Javier Orduz
Department of Computer Science,
School of Engineering and Computer Science.
Baylor University, Waco, TX, USA
Javier_OrduzDucuara@baylor.edu

Abstract

In recent years, researchers have started looking into data transformations in quantum computation. They want to see how quantum computing affects the robustness and performance of machine learning methods. Quantum mechanics succeed in explaining some phenomena where classical formulas failed in the past. Thus, it expanded in analytical research fields such as Quantum Machine Learning (QML) over the years. The developing QML discipline has proven solutions to issues that are equivalent (or comparable) to those addressed by classical machine learning, including classification and prediction problems using quantum classifiers. As a result of these factors, quantum classifier analysis has become one of the most important topics in QML. This paper studies four quantum classifiers: Support Vector Classification with Quantum Kernel (SVCQK), Quantum Support Vector Classifier (QSVC), Variational Quantum Classifier (VQC), and Circuit Quantum Neural Network Classifier (CQNNC). We also report case study outcomes and results analysis utilizing linearly and non-linearly separable datasets generated. Our research is to explore if quantum information may aid learning or convergence.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

1 Introduction

Quantum computation and Quantum information are fields within Computer science. Since we introduced the Quantum word, we should call it modern computer science, but we will not focus on that issue. However, physicists and computer scientists introduced concepts of Quantum Mechanics to computers due to Feynman's ideas [7]. Today we call quantum computation to the area that operates with quantum bits; in this context, we have different models or paradigms to represent these operations; one of the most popular of models is the quantum logic gate (universal quantum gates [11], this model implements the tensor diagram notation, see refs. [2, 3, 18]). One more popularized model is Quantum Machine Learning (QML), which is a paradigm that leverages two things: **a.** Machine Learning algorithms implemented on different libraries (modules) and **b.** quantum computing modules, in particular in Python programming language. Quantum computers can simulate these algorithms, and, therefore, we call this area QML.

A quantum computer (QC) is a machine that uses quantum physics principles like superposition and entanglement to operate. A QC can outperform an ordinary (classical) computer using these concepts. Quantum algorithms enable QCs to achieve significant improvements and speedups over classical algorithms. [15].

Quantum Computing features as the process of manipulating quantum systems, particularly, superposition of quantum states, namely: it is possible to express any vector state in terms of the basis and known coefficients, which respects a particular distribution [1]. It explains the significant calculation speedups [25]. Quantum computing utilized in machine learning compares the performance of algorithms that use quantum circuits to algorithms that use classical computation [1, 22]. For example, the black box system; predicting a binary value of n bits in classical computing will take n steps in the best-case scenario, while Quantum mechanics uses qbits to store information and accurately predict the number in one step. Bernstein-Vazirani is one such algorithm [20]. It outperforms both classical computing costs and accuracy [13, 17].

Researchers currently analyze some models with classifiers in the quantum context. For example, the Support-vector classifier (SVC) is a supervised learning method that distinguishes between classes by evaluating the hyperplane [27]. SVC can also be implemented with modified kernel set up [14]. Also, there is a quantum neural network classifier that uses a layer of qubits [16]. Shah *et al.* analyzed classifiers on quantum hardware and stated that conventional computers fail to operate on datasets with large dimensional spaces. Still, quantum computers can efficiently handle those computations [26].

In [28], Yano *et al.* introduced Quantum Random Access Coding (QRAC) to map high dimensional discrete features and showed that the training time for a variational classifier is faster compared to the conventional technique. In 2018, a quantum-classical technique that can do k-fold cross-validation with linear speedup was proposed, and these classifications were validated [5]. Quantum computing and kernel methods have an evident similarity in their mathematical frameworks [23]. Diversified and intensive research development has been witnessed in the field of quantum optimization and machine learning with several tools serving as a choice: Variational quantum circuits [24], Quantum neural networks [9, 10], as well as quantum support vector machine for big data classification [19].

Variational quantum algorithm (VQA) trains a parameterized quantum circuit using a classical optimizer [4]. VQA can be used as a classification technique as well as a dimension reducer. *Liang et al.* presented two VQA's: quantum local discriminant embedding and quantum neighborhood preserving embedding for classification and dimensionality reduction [12].

This paper implements four quantum classifiers, i.e. SVCQK, QSVC, VQC, and CQNNC, on four simulated datasets. We are interested in exploring the potential relations or correlation among classifiers after adding quantum concepts. Our analysis includes training time, prediction time, and accuracy for different algorithms and datasets. This paper shows interesting results for those parameters.

This paper is organized as follows: the next section, Methodology, describes the datasets and their processing. The section Result and Analysis shows our results and exposes results and discusses an analysis. Finally, we leave our conclusions.

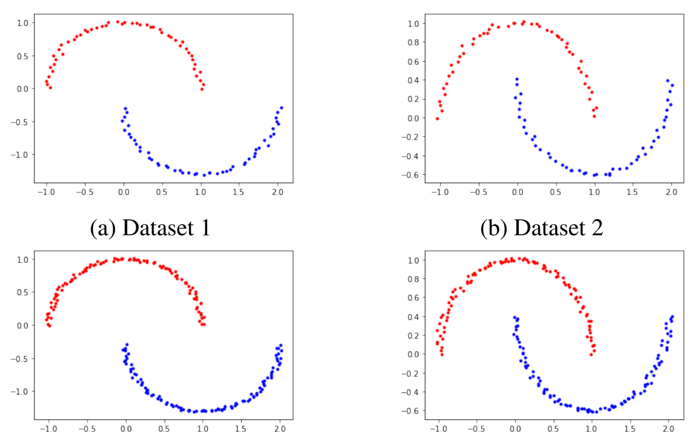
[1] More information about quantum mechanics concepts can be found in refs. [8, 21]

Methodology

This section describes how two half-moons datasets were generated and the implementation of the algorithms mentioned above. The section is divided into two parts: the first describes datasets generation, and the second illustrates the categorization procedure.

Generating Dataset

For our tests, we created four semi-circle datasets. Figure 1 depicts how the data points are dispersed throughout those datasets in a graphical representation. Dataset 1 & 3 consists of 100 and 200 data points, respectively. These datasets are linearly separable. Similarly, Dataset 2 & 4 are 100 and 200 data points, respectively, but these datasets are non-linearly separable.



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

(d) Dataset 4

Figure 1: Graphical representation of the four datasets.

Classification

SVCQK, QSVC, VQC, and CQNNC were implemented in this study. We used the Qiskit library $(0.15.0)$ for quantum implementation, scikit-learn for classical dataset techniques, and matplotlib for visualization. We generated a feature_map which converts classical features into quantum bits. A higher number of classical features in the dataset can be converted into any lower number greater or equal to two-qubit features. However, there is no way to transform lower classical feature numbers to more significant qubit feature numbers. We used the `feature_map` and quantum backend as the parameter to the kernel for SVCQK, VQC, and QNN, while the Qiskit library creates the quantum kernel for QSVC.

Our open-source implementation of the Bernstein-Vazirani (BV) algorithm and four quantum classifiers is available online[2].

This algorithm is maps a function, $f(x)$, with n-qbits to one qbit, namely, $f : \{0,1\}^n \rightarrow \{0,1\}$; where n bits to one bit. In this algorithm $f(x) = a \cdot x$. We are aware about the different algorithms, implementation and details; our criteria to choose this algorithm is the sparse literature BV algorithm as a classifier. We found classification techniques with other quantum algorithms in [6].

Results and Analysis

Table 1 presents the findings of this experiment. For each experiment, there are three measures of results: training time in seconds, prediction time in seconds, and accuracy as a percentage for each algorithm versus each dataset. Horizontally, the table is divided into six divisions. For one method specified in the first column, each horizontal part contains training time (how much time is required

[2] `https://anonymous.4open.science/r/Quantum-Classifiers-5EC4`

for training), prediction time (how much time is required for predicting), and accuracy. Table 1 has a header in the first row (Algorithm, Result, and the Datasets). Row 1, 2 & 3 inside each classifier's horizontal boundary tells the amount of time it took that algorithm to train the datasets, how long the method took to categorize the same testing data from the datasets, and the accuracy rate respectively.

We implemented support vector and neural network classifiers in both conventional and quantum versions. The first two classifiers, SVCQK and QSVC, are quantum versions of the support vector, while SVM is the classical support vector result information. Similarly, VQC and QNN are quantum variants, while Multi-layer-perceptron (MLP) is the conventional equivalent of both. Dataset 1, Dataset 2, Dataset 3, and Dataset 4 are the simulated data sets represented by the dataset columns. Datasets 1 and 2 are semi-circle datasets of size 100, with Dataset 1 being linearly separable but Dataset 2 is not linearly separable. Datasets 3 and 4 each include 200 data points. Similar to Dataset 1, Dataset 3 is linearly separable and Dataset 4 isnt. The data shows that prediction in classical machine learning takes a short time after training, but it takes longer in quantum machine learning since we use a quantum simulator in a classical computer. Quantum computers will most likely be fast when they become accessible in a few years.

Table 1: Performance of the four algorithms on the four datasets.

Algorithm	Result	Dataset 1	Dataset 2	Dataset 3	Dataset 4
	Train (s)	83.81	83.01	336.51	315.83
SVCQK	Predict (s)	42.55	42.52	168.77	160.01
	Accuracy	80	85	80	62.5
	Train (s)	84.39	83.17	335.59	320.85
QSVC	Predict (s)	40.98	42.39	169.01	159.73
	Accuracy	80	85	80	62.5
	Train (s)	0.01	0.01	0.01	0.01
SVM	Predict (s)	0.01	0.01	0.01	0.01
	Accuracy	100	100	100	90
	Train	166.19	154.16	268.37	284.70
VQC	Predict	0.58	0.60	1.18	1.13
	Accuracy	30	30	25	22.5
	Train	190.04	146.43	324.85	311.16
QNN	Predict	42.56	42.79	164.42	158.70
	Accuracy	80	85	80	62.5
	Train (s)	0.42	0.69	0.75	0.79
MLP	Predict (s)	0.04	0.04	0.05	0.06
	Accuracy	100	100	95	95

Our evaluation of the table describes accuracy rate for analogous classifiers in classical and quantum versions as little discernible change, with accuracy rates about identical for each type of classifier in both versions. In addition, Table 1 shows that SVCQK, QSVC, and QNN outperform VQC for all of these datasets, despite its substantially shorter testing time. Another interesting finding is that when we doubled the dataset size, training and testing time nearly quadrupled for both linearly separable and non-separable datasets. At the same time, performance accuracy for linearly separable datasets remains constant while it decreases for non-separable datasets.

Conclusion

We summarize the essential facts of our work as follows in this section. We have implemented four quantum classifiers: support vector classifier with a quantum kernel, quantum support vector classifier, variational quantum classifier, and quantum neural network on four simulated datasets. We focused on the problem of discovering the relationship between quantum classifiers applying in classical datasets and their performance on these different datasets.

In the quantum computing context, we found little research efforts on classifiers for different algorithms, and their impact on any particular dataset.

According to our overall experiment result, the difference in performance between quantum and classical classifiers on classical datasets is insignificant. In larger and linearly separable datasets, the training and testing time generally increased. However, the training and testing time with the classical classifier is comparatively low. The performance of accuracy stays almost the same in all datasets and classifiers except for VQC. The performance of VQC is relatively poor than others. Accuracy also drops slightly than in smaller datasets when implementing it on linearly inseparable datasets.

We summarize five critical facts about our work, as follows:

- We implement SVCQK, QSVC, VQC, and CQNNC on four types of generated datasets.
- We show the analytical result of these quantum classifiers on the datasets and discuss performance concerning overall required training and testing time, including the accuracy of each classifier on all datasets.
- We show that for both 100% linearly separable and non-linearly separable datasets, SVCQK, QSVC, and CQNNC give similar accuracy. However, VQC provides a low performance in accuracy than other models.
- We show that SVCQK, QSVC require similar training and testing time while VQC, CQNNC requires more training time than the other two for the smaller datasets and slightly less training time than the other two for the larger (double of, the smaller ones) datasets.
- We further find that, except for VQC, all classifiers take roughly a similar prediction time. VQC takes significantly less time to test and evaluate the output.

Our work is among fewer works regarding quantum classifier analysis, ours found VQC performing differently than other quantum classifiers. The impact of our study is that the quantum support vector performs best in small classical datasets among different quantum classifiers. Therefore, the key facts that we summarize in this section are about the problem that we solved, the process that we followed, and the outcome of our experiment.

Acknowledgments and Disclosure of Funding

This work was partially supported by the Baylor University, Baylor AI Lab, https://baylor.ai/.

References

[1] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Quantum clustering algorithms. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 1–8, New York, NY, USA, 2007. Association for Computing Machinery.

[2] Jacob Biamonte and Ville Bergholm. Tensor networks in a nutshell, 2017.

[3] Jacob Biamonte, Ville Bergholm, and Marco Lanzagorta. Tensor network methods for invariant theory. *Journal of Physics A: Mathematical and Theoretical*, 46(47):475301, Nov 2013.

[4] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, pages 1–20, 2021.

[5] Priscila G.M. dos Santos, Ismael C.S. Araujo, Rodrigo S. Sousa, and Adenilton J. da Silva. Quantum Enhanced k-fold Cross-Validation. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 194–199, October 2018.

[6] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. A grover-search based quantum learning scheme for classification. *New Journal of Physics*, 23(2):023020, feb 2021.

[7] Richard P Feynman. Simulating physics with computers. In *Feynman and computation*, pages 133–153. CRC Press, 2018.

[8] Griffiths, D. *Introduction to Quantum Mechanics (3d Edition)*. Pearson Prentice Hall, 2nd edition, April 2004.

[9] Iordanis Kerenidis, Jonas Landman, and Anupam Prakash. Quantum algorithms for deep convolutional neural networks. *arXiv preprint arXiv:1911.01117*, 2019.

[10] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Physical Review Research*, 1(3):033063, 2019.

[11] Aleksei Yur'evich Kitaev. Quantum computations: algorithms and error correction. *Uspekhi Matematicheskikh Nauk*, 52(6):53–112, 1997.

[12] Jin-Min Liang, Shu-Qian Shen, Ming Li, and Lei Li. Variational quantum algorithms for dimensionality reduction and classification. *Physical Review A*, 101(3):032323, 2020.

[13] Ji Liu, Gregory T. Byrd, and Huiyang Zhou. Quantum Circuits for Dynamic Runtime Assertions in Quantum Computation. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20, pages 1017–1030, Lausanne, Switzerland, March 2020. Association for Computing Machinery.

[14] Daniele Loiacono, Andrea Marelli, and Pier Luca Lanzi. Support vector regression for classifier prediction. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 1806–1813, London, England, July 2007. Association for Computing Machinery.

[15] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):1–8, 2016.

[16] Massimo Panella and Giuseppe Martinelli. Neural networks with quantum architecture and quantum learning. *International Journal of Circuit Theory and Applications*, 39:61–77, 2011.

[17] Tirthak Patel and Devesh Tiwari. Veritas: accurately estimating the correct output on noisy intermediate-scale quantum computers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, number 15, pages 1–16. IEEE Press, November 2020.

[18] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1:221–244, 1971.

[19] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[20] C. D. Herold S D Fallek. Transport implementation of the Bernstein–Vazirani algorithm with ion qubits. *New Journal of Physics*, 18, 2016.

[21] J. J. Sakurai. *Modern Quantum Mechanics (Revised Edition)*. Addison Wesley, 1 edition, September 1993.

[22] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans): Challenges, solutions, and future directions. *ACM Comput. Surv.*, 54(3), May 2021.

[23] Maria Schuld. Quantum machine learning models are kernel methods. *arXiv e-prints*, pages arXiv–2101, 2021.

[24] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.

[25] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[26] Aditya Shah, Maulik Shah, and Pratik Kanani. Leveraging Quantum Computing for Supervised Classification. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 256–261, May 2020.

[27] Quinn Vinlove, Jens Mache, and Richard Weiss. Predicting student success in cybersecurity exercises with a support vector classifier. *Journal of Computing Sciences in Colleges*, 36(1):26–34, October 2020.

[28] Hiroshi Yano, Yudai Suzuki, Rudy Raymond, and Naoki Yamamoto. Efficient Discrete Feature Encoding for Variational Quantum Classifier. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 11–21, October 2020.