Quantum Machine Learning Algorithms Applied to Skin Pixel Segmentation

A Report Submitted in Partial Fulfillment of the Requirements for the

SN Bose Internship Programme (Mode of Internship ONLINE)

By

Shuvra Neel Roy 1915045

Under the Guidance of

Dr. Partha Pakray

Assistant Professor

Computer Science and Engineering Department

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Assam

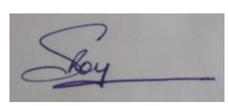
Dec-Feb 202

DECLARATION

Quantum Machine Learning Algorithms Applied to Skin Pixel Segmentation

I declare, that the works presented in this report bears full credit to name of the authors mentioned in the reference section. This report is a collective re-render of the works done by the authors referred and presented in a diluted manner to any and all to engage. Some pieces of information may have directly been rewritten without much edit for the sake of proper flow of meaning. The dataset used in this report is cited in the reference section. The collective ideation of the report falls under myself and I pledge to have adhered to all principles of academic honesty and integrity.

Shuvra Neel Roy 1915045



Department of Computer Science and Engineering

National Institute Of Technology Silchar, Assam

ACKNOWLEDGEMENT

First of all it is a great pleasure to express our gratitude to our project supervisor Dr. Partha Pakray , who has given us this internship opportunity to work under his guidance. It is a great learning experience that would definitely benefit us in our upcoming endeavours. His constant guidance has granted us the path to complete this internship with utmost sincerity.

Thank you

ABSTRACT

Quantum Computing is paving a new field for complex and parallel computation of data. With the Noisy Intermediate Scale Quantum (NISQ) technology, it is possible to simulate 4-12 qubits on real time quantum devices and with new companies entering this field, the growth is suspected to be rapid in the coming years. In this report we take a gloss over some of the aspects of quantum domain of machine learning and use it to classify viz segment skin colored pixel from other randomized pixels. The accuracy result from four types of algorithms, two classical and two quantum is presented and the conclusion drawn from my experience with the report could be summarized as ‘more amount of research and work is required for quantum machine learning algorithms to mature and outperform present day core classical Machine learning and Deep learning techniques, but that day is not nigh’.

LIST OF FIGURES

Figure **Page No.**

LIST OF ABBREVIATIONS

1. K	(Kelvin)
2. ANN	(Artificial Neural Network)
3. QML	(Quantum Machine Learning)
4. d	(feature dimension)
5. SVM	(Support Vector Machines)
6. ML	(Machine Learning)
7. VQC	(Variational Quantum Classifier)
8. NISQ	(Near-term Intermediate Scale Quantum)
9. QSVM	(Quantum support vector machine)

TABLE OF CONTENTS

1. Introduction

The idea of classification between abstract data points has now been a well endeavored supervised learning problem. Supervised Machine Learning algorithms such as Logistic Regression, Support Vector Machines have been well a established use case for Classification tasks with SOTA performances, until the data was no more simple. Along the lines, a different branch of machine learning grew which had relied on the biological structure of brain cells viz neurons. The idea was to imitate the natural flow of information within brain cells. With increasing compute power, this design of neuron like networks that relied on a non-linear activation function and backpropagation of gradients to learn complex decision boundary was a success. Artificial Neural Networks became the go-to product for complex problems. Except training such a model with more than 1 billion parameters is itself an expensive and time taking process.

On the other hand, Quantum technologies were growing at a rapid pace. In 2012, Peter Shor came up with the quantum algorithm for Prime Factor which showed to have a polynomial runtime [3]. Thus, proving the advantage of quantum computing routines for classically difficult problems. Companies quickly acted on this potential to rapidly advance their quantum research. With each new paper, new hardware and the power of cloud, it is now a matter of internet connectivity and simple programming knowledge for any person to try to find their own quantum algorithm and run it on actual devices. Companies like IBM, Amazon are some of the cloud providers for quantum compute.

Yet, the current devices themselves are not perfected. Quantum computers with 50-100 qubits are able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates limits the size of quantum circuits that can be executed reliably [4]. Many error correcting techniques have been developed to cope with the present day Noisy Intermediate Scale Quantum computers and can be referenced here [32]. Many algorithms such as Grover's Search Algorithm [5] , quantum-enhanced feature space [6], have shown to have theoretical advantages in terms of performance but are not practically feasible at scale. Yet there is much vested interest in the quantum domain of computation, partly due to its potential application in Machine Learning [7-8]. In particular, Quantum Machine Learning (QML), could potentially have greater advantages over classical algorithms in terms of computational speed and complexity.

This report focuses on the performance of QML techniques versus classical machine learning algorithms for pixel level classification. The dataset used is the Skin Pixel Segmentation dataset (more about in section 4). It was found that the generalizing ability of QML algorithms are 'not significantly worse' than some of the well established classical ML models (including ANNs). This was already thoroughly well studied in the paper [15].

2 Materials and Methodologies

2.1 Classical Computer and Machine Learning

In the realm of classical computation, the unit of calculation is called a bit. A bit can be in one of its two discreet states at any given time, both of which is described in terms of presence or absence of charge. Now, a sequence of bits is known as a bit stream. With each new bit, the capacity of the bit stream to represent any combination of binary sequence in the range of 2^n. One unique combination at a time. A string of four bits is called a *nibble* and a bit string of length 8 is called a *byte.* These bits are manipulated using gates such as AND, OR, NOT etc., to perform any sequence of operation one desires. This is the crux of the classical computation. Any computer algorithm can be broken down to a sequence of operations of some combination of these gates. One may be reminded here that these operations are irreversible, i.e. we cannot know what the inputs to an AND gate were, once we perform the operation and get an output. This can also be viewed as a 'black-box' operation, represented in fig2.1

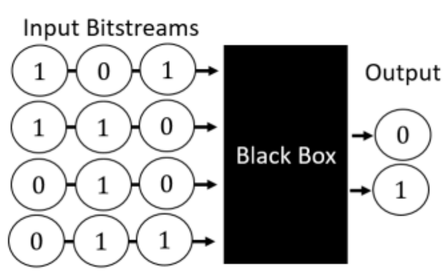


Fig 2.1: 'Black-box' analogy for a sequence of gate computations.

Coming to ML algorithms, a typically relatable method to the 'Black Box' analogy of gates, would be ANN [2], with weighted branches and a constrained non-linear activation function. It is said an ANN can approximate any Black Box [13] gate model. Given any gate circuit, first pose a classification problem to classify the final state (output) of the gate model G(x) corresponding to random valid inputs 'x'. The sampled set S:{X,Y}, where X is randomly sampled input datapoints 'x', and Y is the collection of output of G(x). The sampled set S , is then divided into training set and testing set. The ANN model is trained to predict the output $p(y = class\_i \mid x, w)$. The network is trained using backpropagation and a suitable loss function. The testing accuracy is measured against the test set to validate out of sample performance. This process of repeated optimization of weights to satisfy loss constraint is shown to open the doors to new and more powerful function approximation models.

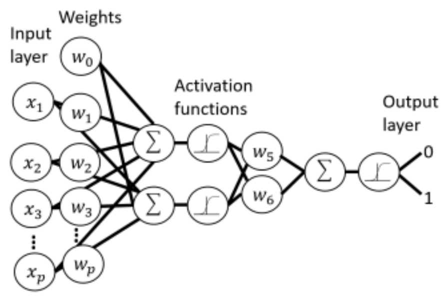


Fig 2.2: Neural network architecture. Activation functions are similar to logic gates as they take a number of inputs to generate output.

2.2 Quantum Computing

Quantum Computers are built differently than classical computers. The fundamental unit of these computers are *Qubits*. In order for these qubits to exhibit any useful quantum behavior, they must be isolated from any and all forms of natural radiations present in the surrounding. Hence the devices themselves are kept in supercooled containments that maintain a temperature of almost about 0.015 K and “this is still noisy”. More on this in [14].

The qubits at ideally 0 Kelvin have unique properties. Quantum bits can be thought of as a magnetic spin of an electron. If the electron spin is in “up” position, then it can be takes as equal to 1. Whereas, if the electron is in spin ‘down’ position, it can be taken as 0. However, we have no idea weather the qubit is in state $|1\rangle$ or $|0\rangle$ position, or can also be in a superposition of many states and only by measurement can we identify its state. However, measurement of the qubit will lead to waveform collapse onto the measurement basis which is usually either $|1\rangle$ or $|0\rangle$. Hence the process of measurement of a qubit can be thought of as a random variable. It is only by repeated shots of the measurement can we find the expectation over a certain state of measurement. More on this in [16]. To compare bits to qubits visually, a figure is given below, and is referred to as the ‘Bloch’ sphere.

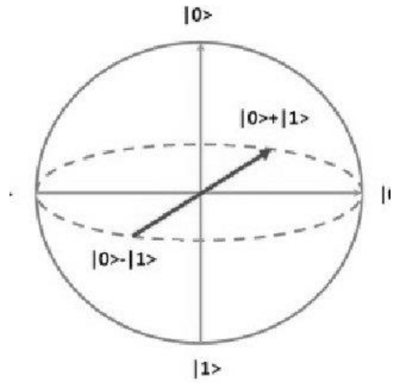


Fig 2.3: A ‘bloch’ sphere representation of qubit. The head of the arrow represents the state of the qubit, with the surface of the sphere being a potential state the qubit can take.

Operations on these qubits are done using gates but these gates are fundamentally different than the classical gates. Quantum logic gates are commonly referred to as quantum gates and they are essentially transformation on one or more qubits. In table 2.1, few of the gates used in quantum computations are listed. A circuit with quantum gates that operates on one or many qubits are called Q*uantum Circuits.* Readers may follow [17] for a detailed understanding of the computational methods quantum computers use.

Quantum gate	Input	Classical gate	Short description
Hadamard gate	1 qubit	None	Creates superposition
Pauli-X gate	1 qubit	Not gate	Creates x rotation
Pauli-Y gate	1 qubit	None	Creates y rotation
Controlled not gate	2 or more qubits	Not gate	Used to measure 2^{nd} qubit

Table 2.1: Quantum Gate examples

Quantum circuits, much like classical gate circuits operate and establish a relation between the qubits. Also, the qubits themselves can be in a superposition, hence, for a *n* qubit system, one can think of quantum operations as, acting on 2^n different states, simultaneously. A more interesting and unique property of multiple qubit systems can be observed when they are entangled. [17] describes this behavior as “deep connections between spatially separated entities”. When 2 qubits are entangled, measurement on one of the qubit leads to collapse of the wave function for both the qubits. [18] In other words, there exists a deep correlation in the measurement of two entangled qubits, even if one cannot tell at what state ($|1\rangle$ or $|0\rangle$) either one of the qubits were in.

Algorithms that take advantage of these unique properties of the quantum systems ([3-4-5-6]) are immensely more powerful for certain type of problems. I say only 'certain' because it is how one uses these tools to achieve a better solution faster. Classical computers are deterministic and hence can achieve a deterministic task faster, that includes all the background processes running inside a computer. Another interesting and almost crucial fact for quantum systems is reversibility. The algorithm for Teleportation of information with quantum bits [19] is based on this property. One can get back the initial qubit state by means of only reversing the circuit and re applying it on the qubit state. For e.g. applying X gate to state 1, we get $X|1\rangle = |0\rangle$ and reapplying the X gate $XX|1\rangle = X|0\rangle = |1\rangle$.

For purpose of this report, two quantum machine learning algorithms are discussed here. Namely the Variational Quantum Classifier (VQC) and the Quantum Support Vector Machines (QSVM). But there are other machine learning algorithms and they are thoroughly discussed in [20]. [6]

2.3 Variational Quantum Classifiers (VQC)

Variational Quantum Classifier is an algorithm synonymous to present day neural networks that can give experimental results in NISQ devices without the need for error correction techniques. This method comprises of a variational (parametrized) quantum circuit whose parameters are tuned with approaches similar to how neural networks are trained.

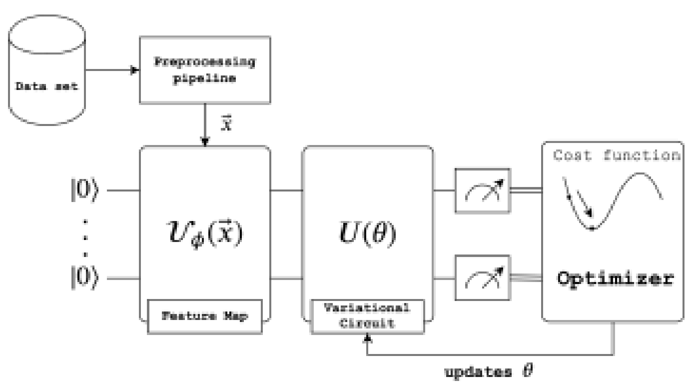


Fig 2.4: Variational Quantum Classifier schematic

The circuit comprises a feature map, that essentially maps the input to the potentially higher-dimensional Hilbert Space. The encoding of the inputs to the quantum feature space is discussed in [22] and can be achieved in many different ways. One way is to encode the input datapoints as rotational arguments to qubits and create interactions among all the qubit pairs possible. In [25-26], there is a detailed study in the work of feature embedding in quantum circuits. In short, apply the unitary $U_{\Phi(x)}$ that maps $x \colon \Phi(x) \in \mathcal{H}^n$ on the $|0_n\rangle$ state that realizes the state $|\Phi(x)\rangle$. The unitary provided in [25] creates a nonlinear embedding into the feature space of qubits. The unitary is given by

$\mathcal{U}_{\Phi(x)} = \mathbb{U}_{\Phi(x)} H^{\otimes n} \mathbb{U}_{\Phi(x)} H^{\otimes n}$, where H is the Hadamard gate , and $\mathbb{U}_{\Phi(x)}$ denotes a diagonal gate in the Pauli-Z basis as follows:

$$\mathbb{U}_{\Phi(x)} = \exp\left(i \sum_{S \subseteq [n]} \phi_s(x) \prod_{k \in S} Z_k \right) \tag{1}$$

Where the coefficients $\phi_s(x) \in R$ are fixed to encode the data x. The mapping function is later discussed in the report. In general, [25] tells us that $\mathbb{U}_{\Phi(x)}$ can be any diagonal unitary that can be efficiently realizable with short-depth circuits. In total one needs at least $n \geq d$ qubits to construct an efficient quantum-enhanced feature map, where d is the dimension of the feature vector of the dataset.

Then, a small depth quantum circuit $W(\theta)$ is applied to the quantum state $|\Phi(x)\rangle$, where θ is the vector of parameters that will be learned from the training data. Finding the circuit parameters is parallel in terms of finding the separating hyperplane in case of a support vector machine, with input features mapped to a n-dimensional Hilbert space. The binary decision rule can be given as:

$$f_\theta(x) = \langle \Phi(x) | W_\theta^\dagger \, \mathrm{Z} \, W_\theta | \Phi(x) \rangle + \mathrm{b} \tag{2}$$

Where,

$$Z = \sum_{z \in \{0,1\}^2} g(z) \langle \mathrm{z} | \mathrm{z} \rangle \text{ , where g(.)} \in \{-1,1\} \tag{3}$$

And ,

$$Label(\, x) = sign(f_\theta(x)). \tag{4}$$

The probability of measuring z is then given by

$$|\langle z | W_\theta | \Phi(x) \rangle|^2 = \langle \Phi(x) | W_\theta^\dagger | \mathrm{z} \rangle \langle z | W_\theta | \Phi(x) \rangle \tag{5}$$

Learning the best θ can be achieved by minimizing the binary cross entropy loss H(θ) or the empirical risk R(θ) with respect to the training data S.

$$R(\theta) = \frac{1}{|S|} \sum_{i \in [m_s]} |f(x_i) - y_i| \tag{6}$$

$$H(\theta) = -\frac{1}{|S|} \sum_{i \in [m_s]} \left(\frac{1 - y_i}{2} \log\left(p_i^{(-1)}\right) + \frac{1 + y_i}{2} \log\left(p_i^{(1)}\right) \right) \tag{7}$$

Where,

$$p_i^{(y)} = \sum_{i \in [m_s] | g(z) = y} |\langle z | W_\theta | \Phi(x) \rangle|^2 \tag{8}$$

. [21] presents the method of 'parameter shift rule' for finding the gradients of these parameters with respect to some loss function. The rule lets us find the gradients of the parameters by taking the average of the expectation value of the circuit calculated with a shift of $\theta \pm \pi$. After this, it is a job of classical gradient based optimizer to optimize the parameters. However, author Amira Abbss mentions that VQCs are merely linear classifiers and it only parameterizes a subset of the possible hyperplanes for classification. This means Quantum Kernel based classifiers should strictly more powerful than parameterized variational circuits. There is also the overhead of barren plateaus landscape of the loss function which leads to almost non-existent gradients in some part and steep loss curves in others leading to difficulty in training with standard gradient descent based optimization techniques. [31]

2.4 Quantum Support Vector Machines

In classical SVM [1], we are given the training set S = {(x_1,y_1), (x_2,y_2), (x_3,y_3)…….. (x_{mS},y_{mS})}, where $x_i \in R^d$ and $y_i \in \{-1,1\}$. The binary classification problem can be posed from S to construct a function $f(x)$ such that $f(x_i)y_i > 0$ for every *i*. There can be many classifiers that separates the training example exactly, the simplest form of such a function is $f(x) = w^T x + b$, where $(w, b) \in \mathcal{R}^{d+1}$. But a reasonable classifier is one that maximizes the distance with respect to the closest points of the two classes. This is a quadratic optimization problem also called hard-SVM. When the hard SVM is relaxed to produce a classifier that predicts the training set with some accepted outlier (or slack) to fit the new problem $f(x_i)y_i \geq 1 - \epsilon_i$, for all i. The slack variable $\{\epsilon_i > 0\}$ determines the quality of the classifier. Closer to zero is considered good. Hence, mapping the data to a higher feature space greatly increases the predicting power of our model. Finding a map $x : \Phi(x) \in \mathcal{R}^n$ for n > d. The classifier $f(x)$ is now defined as $f(x) = w^T \Phi(x) + b$. Now, when $\Phi(x)$ is an embedding of data nonlinearly to the quantum state $|\Phi(x)\rangle$, we can use the quantum-enhanced classifier f. This is also the core idea behind VQCs. Much of theory written above is taken from [25].

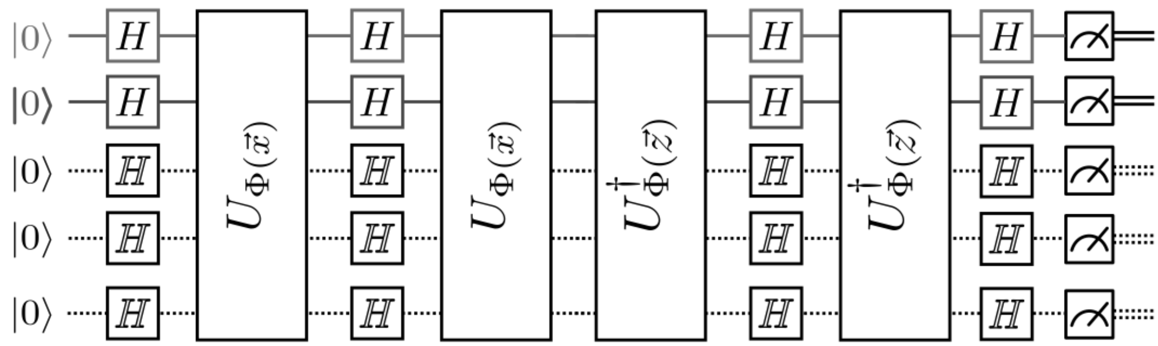


Fig 2.5: Quantum Support Vector Machine. (quantum kernel estimator circuit)(source)

The Kernel estimator circuit, gives us the correlation between feature vectors from n dimensional Hilbert space. The core idea behind the quantum circuit is to find the Hilbert Schmid Inner product between all the training pairs i,j in the training set S. First prepare the state $|\Phi(x)\rangle$ and $|\,\Phi(x_j)^{\dagger}\rangle$. Prepare state $\Phi(x_j)^{\dagger}.\Phi(x)\,|0\rangle$. The values of the Kernel matrix $K_{i,j} = \left|\langle 0|\,\Phi(x_j)^{\dagger}.\Phi(x)\,|0\rangle\right|^2$, which is nothing but the probability to measure state $|0\rangle^n$. This matrix is then used to solve the dual problem to the SVM optimization problem. The resultant kernel matrix can directly be used in classical support vector machine algorithms for classification.

3. Dataset, Preprocessing and Metrics

The dataset selected for this report is the Skin Segmentation Data Set [9], with B,G,R pixel values generated using skin textures from face images of diversity of age, gender and race people. The classification problem at hand was to classify weather the given pixel value was a region of skin or not skin. Relevant papers [10-11]. The pixel values range from 0-255. It is shown that preprocessing of data for use in quantum algorithms is of utmost importance. Qubit amplitudes range from 0 to 1 in the real domain. Hence the pixel values were renormalized by dividing by 255. I kept 100 datapoints from both the classes with final shape being {'skin' : (100,3), 'not_skin' : (100,3)} as training set against a test set of 1000 example of each class.

The metric used for this experiment was the accuracy score. The general idea behind this experiment was to impress upon the generalization power of the quantum algorithms against some of the current machine learning algorithms.

<u>4. Experimentation</u>

For the experiment, two classical ML algorithms and two QML algorithms were set up. I used Support Vector Machine from scikit-learn with C = 1.0, gamma set to 'scale', and 'rbf' kernel. Second, I used MLPClassifier from scikit-learn neural networks library. The hidden layers for the Multi-layer Perceptron (MLP) are were kept in the sequence [[64,128,64,2]] with 'ReLU' activation after each layer and a 'Sigmoid' activation at the final layer. The network was trained with 'Adam' optimizer constrained to Binary Cross-entropy loss. The batch size was kept at 16 and trained for some 50 epochs.

For Quantum Support Vector Machine, I used the qiskit provided QSVM library to build the classifier which takes in feature mapping circuit as one of the arguments. The feature map used was 'ZZFeatureMap' (discussed below). It was

For the variational quantum classifier, I applied a second-order Pauli-Z evolution circuit 'ZZFeatureMap'[24-25] as the input encoder, which maps the input as follows:

$$\phi_s = \begin{cases} x_i & for\ S = \{i\} \\ (x_i - \pi)(x_j - \pi) & for\ S = \{i,j\} \end{cases}$$

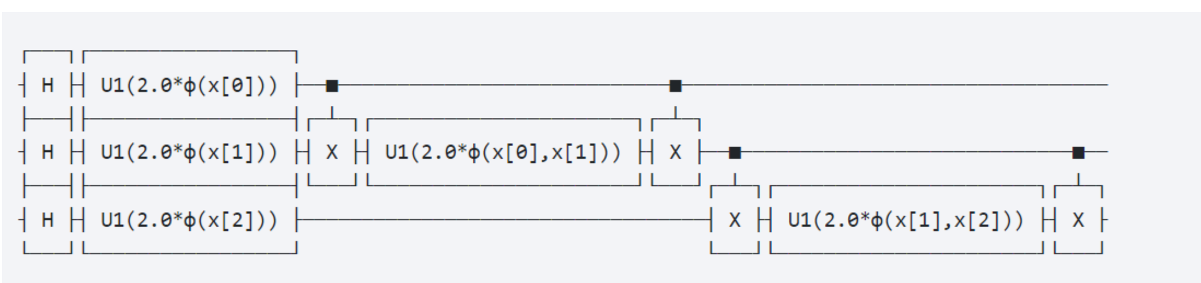


Fig 3.1: ZZ Feature Map for 3 qubits and 1 repetition with 'linear' entanglement. (source:[25])

For the variational circuit (aka ansatz), we leveraged the custom built qiskit EfficientSU2 circuit which consists of two layers of single qubit operations spanned by SU(2) which stands for special unitary group of degree 2, and CX entanglement gates. A figure of the circuit is given below:

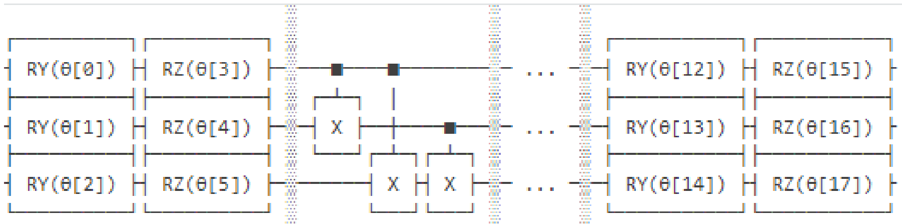


Fig 3.2: EfficientSU2 ansatz. The dotted portion represents k number of repetitions of the Ry,Rz,CX circuit. (source: [27])

The qiskit VQC library takes in a feature map and an ansatz as arguments and builds the whole circuit. The COBYLA [28-30] optimizer was chosen as the classical optimizer for the VQC model for an iteration of 100.

Experiment code + notebook can be found in the following link :
(https://colab.research.google.com/drive/1gWrRT-34kjq5MDypKRPMDHH6yOX3G2hA?usp=sharing)

5. Results & Conclusions:

The results drawn from the experiment is listed in the table below. Due to high job queues on the IBM quantum computers, the 'qasm_simulator' (Qiskit) was used to simulate the Quantum Circuit. The training was left unregularized to observe the steady state generalization of the model, hence higher training iterations.

Table 5.1: Accuracy Score For SVM,MLP,QSVM,VQC

Model	Number of parameters	Training Time	Test Accuracy
Classical SVM	1(bias)	18 secs	98.67%
MLP	16960	59 secs	98.78%
QSVM	1 (bias)	5min47sec	92.85%
VQC	18	4min7secs	98.56%

It is observed that classical algorithms outperformed QSVM but the VQC had a relatively close accuracy considering only 18 parameters as opposed to many ANNs. This could directly be related to imbalance in the distribution of the R, G, B values taken into consideration during training. Also, training on larger training sample is necessary to consider ahead. Such performance is also suspected to be a result dependent upon the choice of the feature mapping. I plan to work ahead on exploring other combination of parameters with the models. A figure of the kernel matrix estimated by the QSVM is given below. Other learning avenues would be to use the quantum-kernel matrix as hidden embeddings for various tasks with the leverage of a hybrid quantum-neural network. Related works have been done previously been done on this by [29] and others.

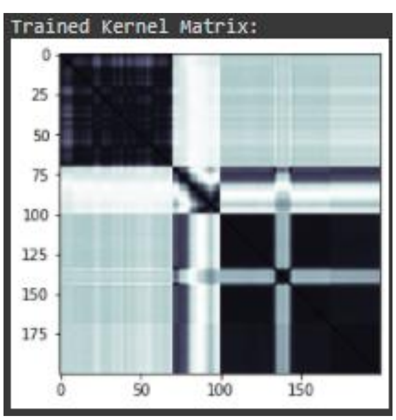


Fig 5.1: Learned kernel matrix. QSVM

6. References

[1] Massimiliano Pontil and Theodoros Evgeniou, Support Vector Machines: Theory and Applications. https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications

[2] Juergen Schmidhuber, Deep Learning in Neural Networks: An Overview. https://arxiv.org/abs/1404.7828

[3] Peter W. Shor (AT&T Research) , Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, https://doi.org/10.1137/S0097539795293172

[4] John Preskill, Quantum Computing in the NISQ era and beyond, https://doi.org/10.22331/q-2018-08-06-79

[5] A fast quantum mechanical algorithm for database search Lov K. Grover (Bell Labs, Murray Hill NJ) , https://arxiv.org/abs/quant-ph/9605043v3

[6] V. Havlícek, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, https://www.nature.com/articles/s41586-019-0980-2

[7] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress. https://doi.org/10.1088/1361-6633/aab406

[8] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification. https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.113.130503

[9] Rajen Bhatt, Abhinav Dhall, IIT Delhi., Skin Segmentation Data Set. https://archive.ics.uci.edu/ml/datasets/skin+segmentation

[10] Rajen B. Bhatt, Gaurav Sharma, Abhinav Dhall, Santanu Chaudhury, â€œEfficient skin region segmentation using low complexity fuzzy decision tree modelâ€• , IEEE-INDICON 2009, Dec 16-18, Ahmedabad, India, pp. 1-4.

[11] Abhinav Dhall, Gaurav Sharma, Rajen Bhatt, Ghulam Mohiuddin Khan, Adaptive Digital Makeupâ, in Proc. of International Symposium on Visual Computing (ISVC) 2009, Nov. 30 â€“ Dec.

02, Las Vegas, Nevada, USA, Lecture Notes in Computer Science, Vol. 5876, pp. 728-736.
[12] Andrew Nader, Kyle Oppenheimer, and Gary Tom, Exponential Advantages in Quantum Machine Learning through Feature Mapping. http://www.henryyuen.net/fall2020/projects/qml.pdf
[13] https://towardsdatascience.com/emulating-logical-gates-with-a-neural-network-75c229ec4cc9
[14] IBM Staff. Inside look: Quantum computer. https://www.research.ibm.com/ ibm-q/learn/what-is-ibm-q/images/infographic-inside.jpg.
[15] Hsin-Yuan Huang, Richard Kueng, John Preskill Just How Much Better is Quantum Machine Learning than its Classical Counterpart? https://arxiv.org/abs/2101.02464
[16] D. F. Walls , G. J. Milburn, Quantum Coherence and Measurement Theory, https://link.springer.com/chapter/10.1007/978-3-642-79504-6_16#citeas
[17] K. M. Svore and M. Troyer, "The Quantum Future of Computation," in Computer, vol. 49, no. 9, pp. 21-30, Sept. 2016, doi: 10.1109/MC.2016.293.
[18] YU. A. PASHKIN, T. TILMA, D. V. AVERIN, O. ASTAFIEV, T. YAMAMOTO, Y. NAKAMURA ,F. NORI and J. S. TSAI . ENTANGLEMENT OF TWO COUPLED CHARGE QUBITS. https://doi.org/10.1142/S0219749903000334 (paid)
[19] Yi-Han Luo, Ming-Cheng Chen, Manuel Erhard, Han-Sen Zhong, Dian Wu, Hao-Yang Tang, Qi Zhao, Xi-Lin Wang, Keisuke Fujii, Li Li, Nai-Le Liu, Kae Nemoto, William J. Munro, Chao-Yang Lu, Anton Zeilinger, and Jian-Wei Pan Quantum teleportation of physical qubits into logical code spaces. https://www.pnas.org/content/118/36/e2026250118
[20] Weikang Li1 and Dong-Ling Deng, Recent advances for quantum classifiers. https://arxiv.org/pdf/2108.13421.pdf
[21] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning https://journals.aps.org/pra/abstract/10.1103/PhysRevA.98.032309
[22] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces.
[23] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuitcentric quantum classifiers, https://journals.aps.org/pra/abstract/10.1103/PhysRevA.98.032309
[24] https://qiskit.org/documentation/stubs/qiskit.circuit.library.ZZFeatureMap.html
[25] HIROSHI YANO1 , YUDAI SUZUKI2, KOHEI M. ITOH1,3, RUDY RAYMOND3,4 , AND NAOKI YAMAMOTO1, Efficient Discrete Feature Encoding for Variational Quantum Classifier, https://arxiv.org/ftp/arxiv/papers/2005/2005.14382.pdf
[26] https://pennylane.ai/qml/glossary/quantum_embedding.html
[27] https://qiskit.org/documentation/stable/0.19/stubs/qiskit.circuit.library.EfficientSU2.html
[28] M. J. D. Powell A View of Algorithms for Optimization Without Derivatives. https://www.researchgate.net/publication/237619137_A_View_of_Algorithms_for_Optimization_Without_Derivatives
[29] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli and Stefan Woerner, The power of quantum neural networks. https://www.researchgate.net/publication/352718881_The_power_of_quantum_neural_networks

[30] https://qiskit.org/documentation/stubs/qiskit.algorithms.optimizers.COBYLA.html
[31] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, https://doi.org/10.1038/s41467-021-21728-w
[32] Joschka Roffe, Quantum Error Correction: An Introductory Guide , https://arxiv.org/abs/1907.11157