Qiskit Fall Fest Kolkata Chapter Hackathon 2022

Organizers : IBM Quantum
Mr. Ritajit Bhowmik, ISI
Ms. Debasmita Bhowmik, ISI

Team QBosons

Members: Vismay Joshi
Tanjin Adnan Abir
Abhishek Rawat
Suprabhat Sinha
Rajat Lakhera

Contents

Task Description

The objective of this project is to determine the best test accuracy using one or more quantum machine learning algorithms, with or without the influence of noise.

- The dataset will first be trained using traditional machine learning algorithms to figure out the best test accuracy.
- Then two QML algorithms VQC and QSVM will be used to obtain best accuracy without the effect of noise.
- And finally the target is to improve the test accuracy under noisy scenario.

At the end of our task, we will summarize the results and mention our findings.

For this project “Iris dataset” will be used.

The Iris Data Set

This data set contains classification of three different species of Iris flower.

It has a total of 150 instances - 50 for each class/species..

There are four different attributes: sepal length, sepal width, petal length, petal width.

Iris Versicolor **Iris Setosa** **Iris Virginica**

Exploratory Data Analysis

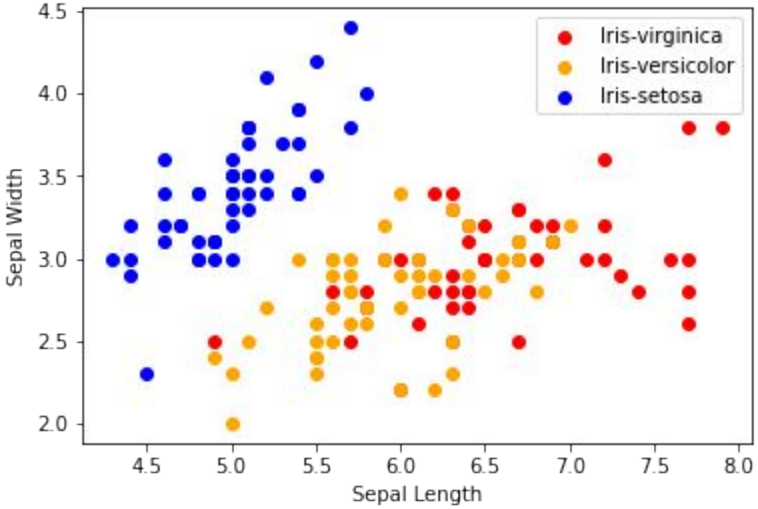


Figure-1

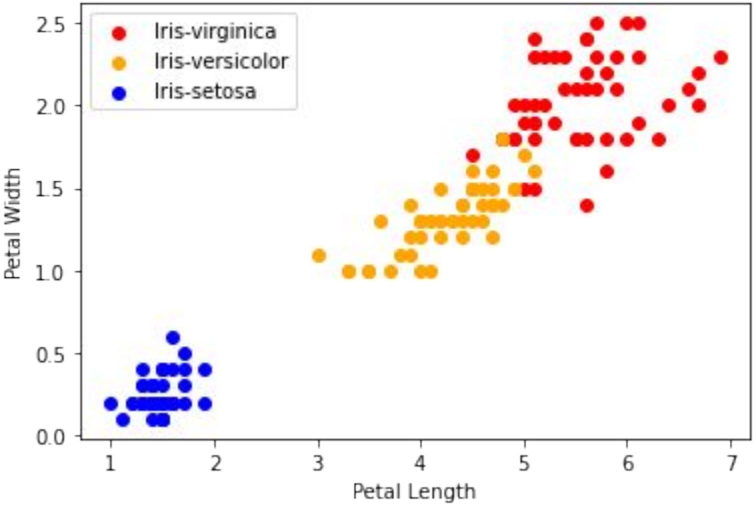


Figure-2

We have drawn scatter-plot diagrams to visualize relation between numerous features of the iris dataset. Here, in first diagram (figure-1), sepal length vs sepal width features were visualized. In figure-2, petal length vs petal width relation was shown.

Exploratory Data Analysis

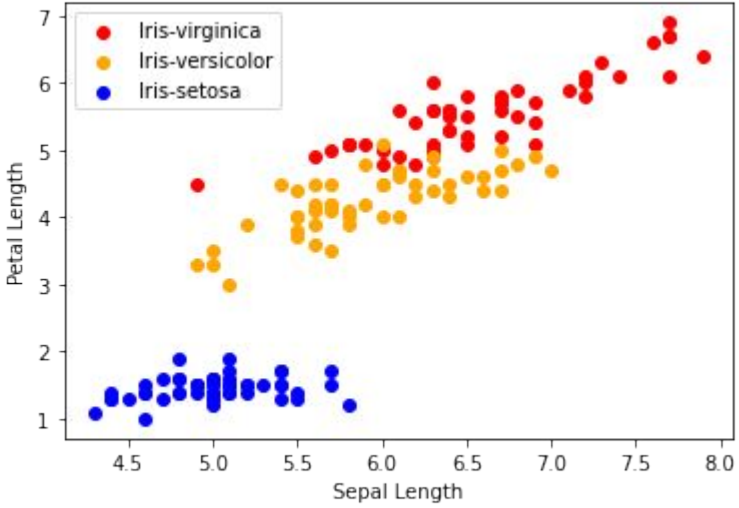


Figure-3

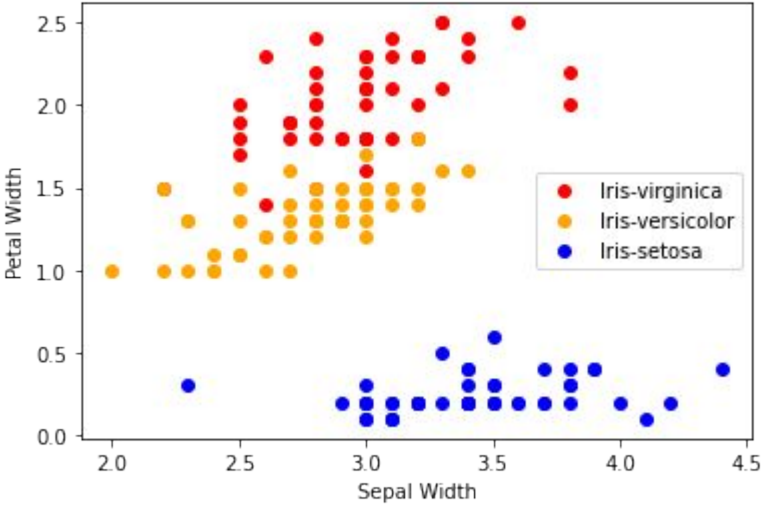


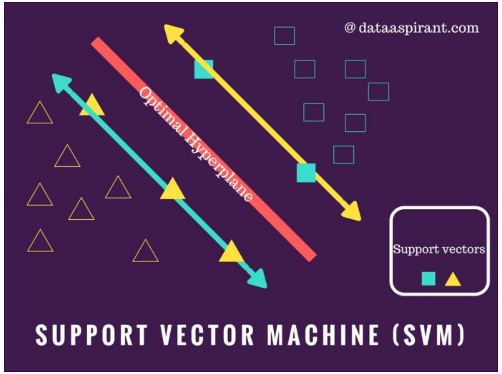
Figure-4

In figure-3, sepal length vs petal length relation was shown, on the contrary, in figure-4, sepal width vs petal width relation was visualized.

Classical ML Algorithms

Support Vector Machine: To Identify a hyperplane in an N-dimensional space to clearly classify the input points. The size of the hyperplane is determined by the number of features. Using statistical frameworks, it generates the optimum decision boundary to categorize distinct classes.

K-Nearest Neighbour: To identify the k-nearest neighbors, it uses the shortest distance between the query instance and the training examples. After obtaining the k-nearest neighbors, the forecast of the query instance requires a simple majority of these k-nearest neighbors.



Source: https://dataaspirant.com/support-vector-machine-algorithm/

Source: https://shubham-agnihotri.medium.com/quantum-machine-learning-102-qsvm-using-qiskit-731956231a54

Classical ML Algorithms

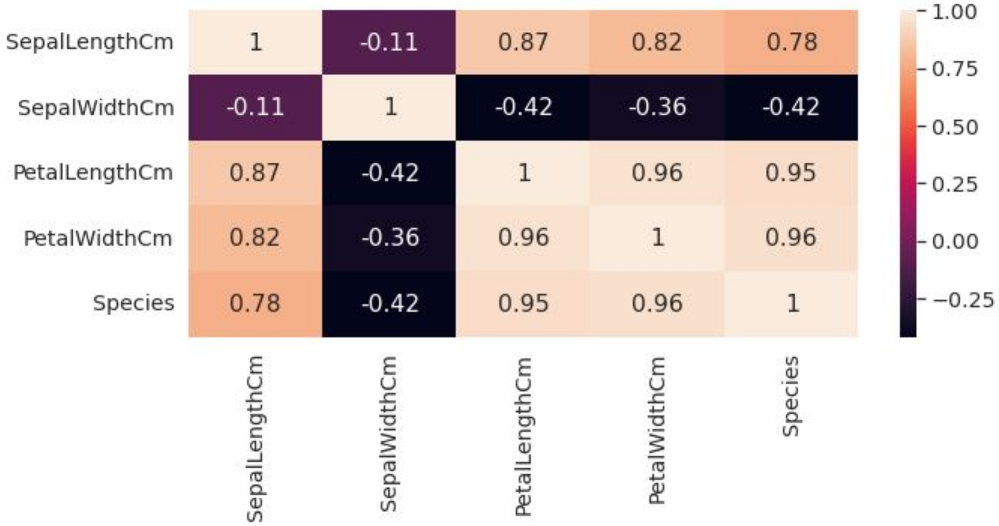
Decision Tree: A supervised ML approach that uses a data flow diagram tree structure to represent decisions, outcomes, and predictions. An algorithmic procedure (a series of if-else expressions) that discovers ways to partition, categorize, and depict a dataset depending on different circumstances is used to create such a tree.

Random Forest Classifier: The random forest is an ensemble learning approach for classification, regression, and other problems that works by building a large number of decision trees during training. For classification problems, the random forest output delivers the class chosen by the majority of trees.

Logistic Regression: Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. It is used to calculate the connection between one or more independent variables and a dependent (target) variable. The dependent variable's output is expressed by discrete numbers such as 0 and 1.

Correlation Matrix

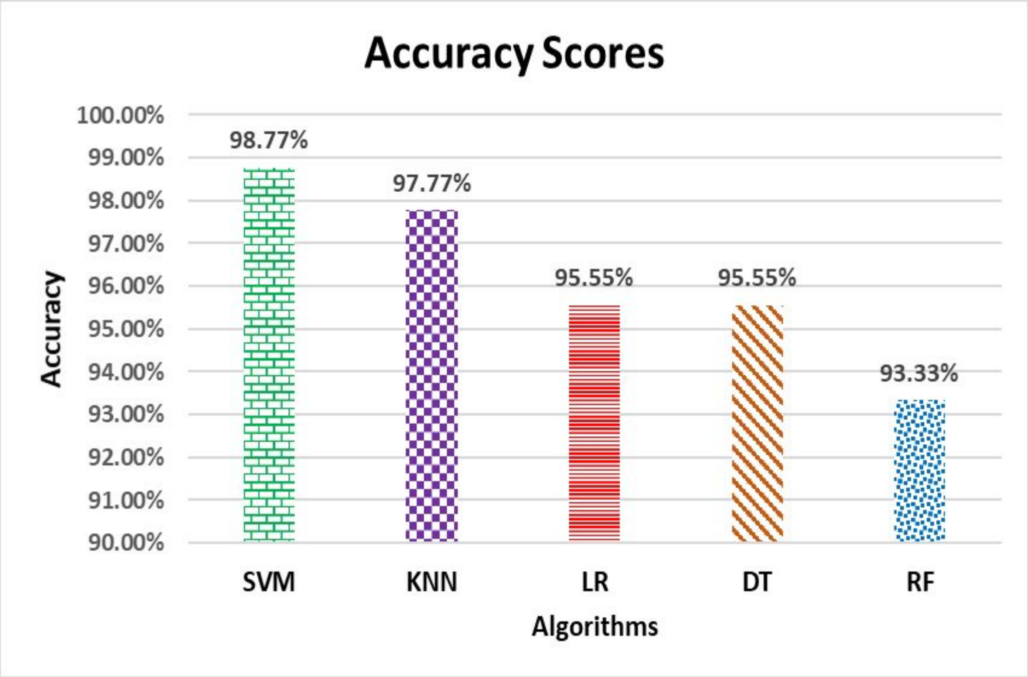
- A correlation matrix is a table showing correlation coefficients between variables.
- Each cell in the table shows the correlation between two variables.
- The value is in the range of -1 to 1.
- If two variables have high correlation, we can neglect one variable from those two.



Training and Accuracy (Classical ML)

We trained the iris dataset with several classical machine learning algorithms and got accuracies for each algorithms after training.

Here, we got best accuracy with SVM method and the least accuracy with random forest classifier.



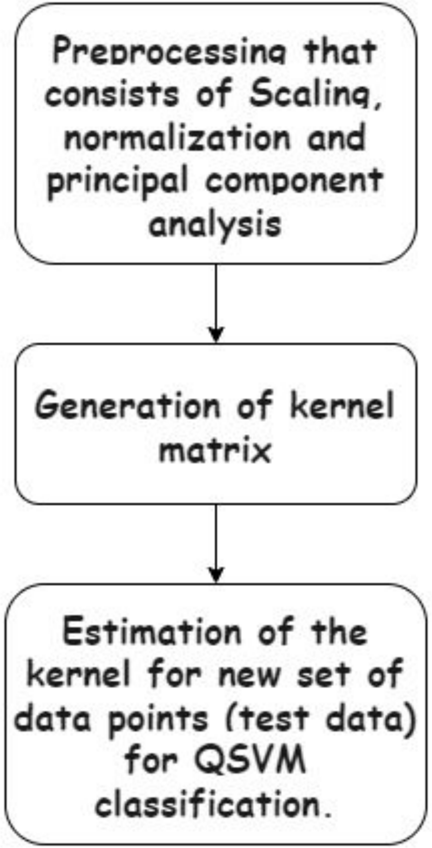
Quantum Support Vector Machine (QSVM)

- A quantum version of the **Support Vector Machine** algorithm which uses quantum laws to perform calculations.
- **Kernel** trick is used for classifying the dataset into various classes using SVM, which transforms the data to find an optimal boundary between the possible outputs.
- Data that seems hard to separate in its original space by a simple hyperplane can be separated by applying a non-linear transformation function (known as **feature map**) in a space known as **feature space.**
- The **Inner product** for each pair of data points in the set is computed to assess the similarity between them and this, in turn, is used to classify the data points in this new feature space (the higher the value of the inner product more similar they are to each other) and this collection of inner products is called the kernel.

How QSVM Works

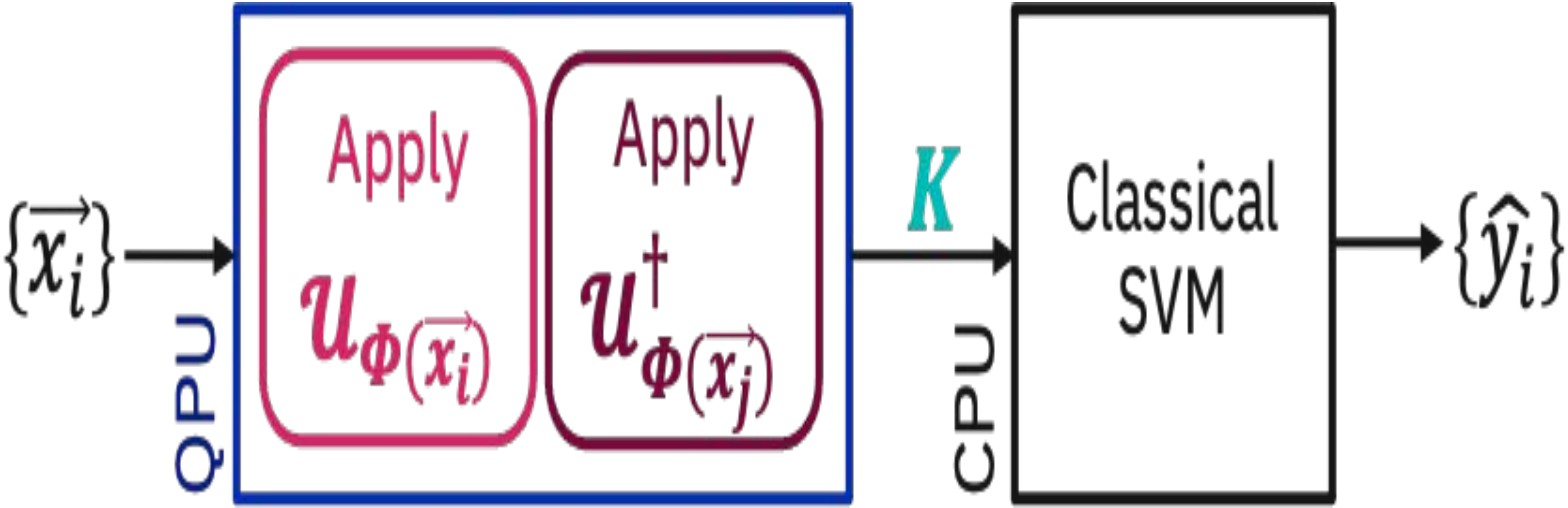
The implementation consists of three basic steps:

- In the QSVM classification phase, classical SVM is used to generate the separating hyperplane rather than using a quantum circuit and here the quantum computer is used twice.
- First, the kernel is estimated for all pairs of training data, and the second time the kernel is estimated for a new datum (test data).
- Least-squares reformulation of the support vector machine is used to change the quadratic programming problem of SVM, into a problem of solving a linear equation system.



How QSVM Works

Here is a block diagram representation of our working.

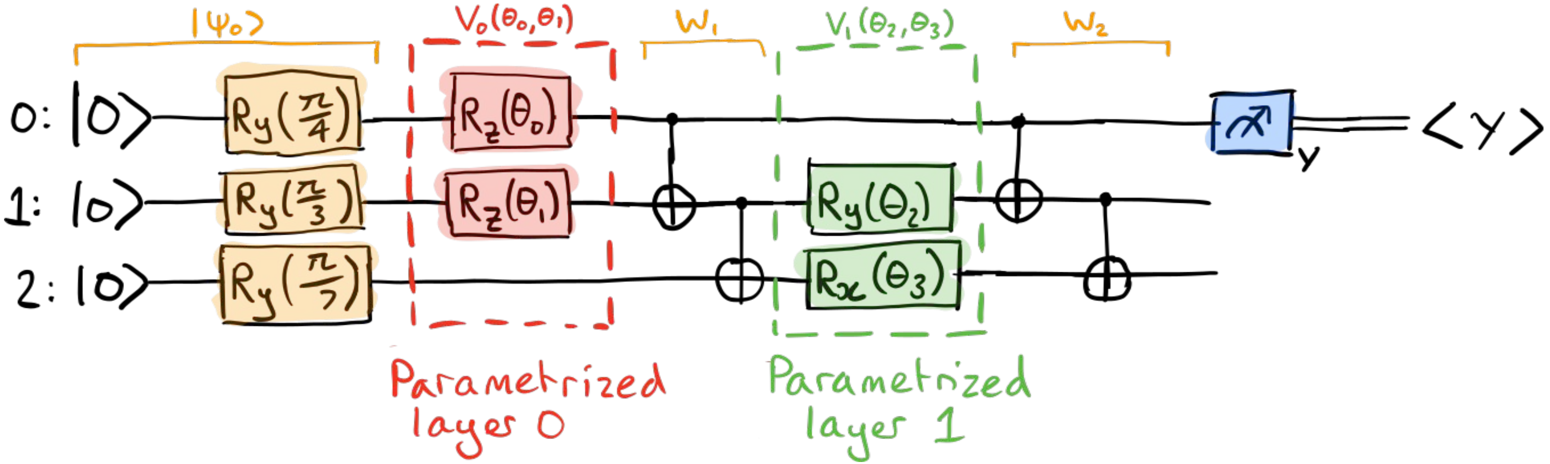


Variational Quantum Classifier (VQC)

- VQC is one of the methods with possible quantum advantage in using quantum-enhanced features that are hard to compute by classical methods.
- Its performance depends on the mapping of classical features into a quantum-enhanced feature space.
- It tries to learn patterns from sets of data, then sort new data into those sets (e.g. given some labelled photos of cats and dogs, try to identify whether a new photo is a cat or a dog).
- The variational quantum classifier does this using parameterized quantum circuits.

Variational Quantum Classifier

VQC uses parameterized circuits (also known as ansatz) to train a quantum model and predict labels for new data using classical labelled data for training.



How does VQC work?

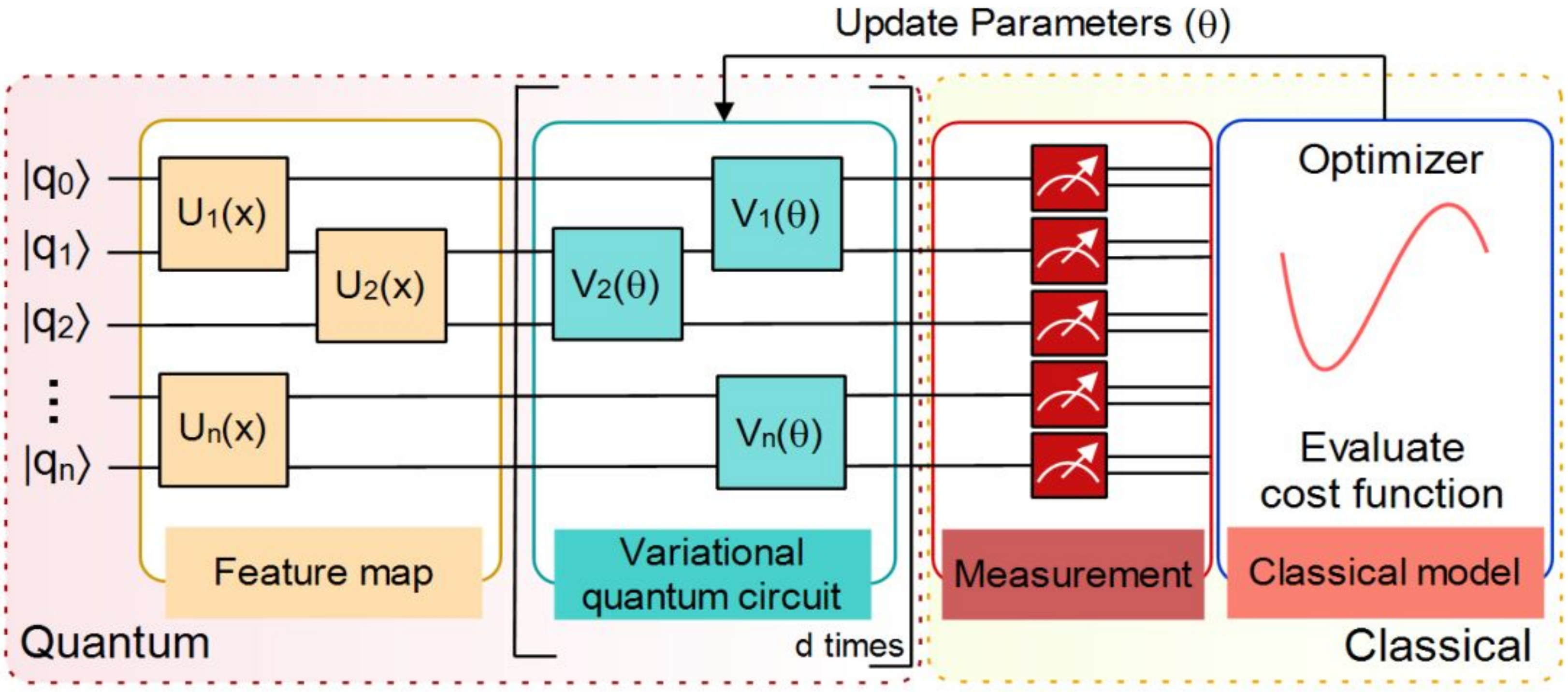
VQC usually involves four steps:

Step 1: Encoding the classical data into quantum circuits

Step 2: Applying parameterized gates to create a working ansatz

Step 3: Measure the circuit and extract the output in form of labels

Step 4: Optimize the circuit using classical optimizers.



Update Parameters (θ)
|q0⟩
|q1⟩
|q2⟩
|qn⟩
U1(x)
U2(x)
Un(x)
Feature map
V2(θ)
V1(θ)
Vn(θ)
Variational quantum circuit
d times
Measurement
Optimizer
Evaluate cost function
Classical model
Quantum
Classical

Note on QML Algorithms

- Unlike in VQC where we can use different types of optimizers like COBYLA, ADAM, SPSA, etc., in QSVM we cannot change the optimizer. The default optimizer for QSVM is SPSA.

- The Multiclass extension of QSVM is currently deprecated. On previous version of Qiskit, when the value of the multiclass extension parameter was " AllPairs() ", an accuracy of 1 could be achieved on a pure model model devoid of noise elements.

- Increasing the depth of Ansatz in VQC, does not increase the convergence of model substantially.

- In QSVM, when the noise parameters are increased from (0.01, 0.01) to (0.1), we see that the model will start to perform badly.

Noise in Quantum Circuitry

Noise is a common issue in every circuitry. In quantum circuitry there are several types of noise some of them are described below

- **Coherent noise** is described by unitary operations that maintain the purity of the output quantum state. A common source are systematic errors originating from imperfectly-calibrated devices that do not exactly apply the desired gates, e.g., applying a rotation by an angle $\phi+\epsilon$ instead of ϕ.
- **Incoherent noise** is more problematic: it originates from a quantum computer becoming entangled with the environment, resulting in mixed states — probability distributions over different pure states. Incoherent noise thus leads to outputs that are always random, regardless of what basis we measure in.

Noise in Quantum Circuitry

- **Bit flip:** When measured in the computational state, the qubit flips from $|0\rangle$ to $|1\rangle$ or vice versa with some specified probability. Bit flip errors are modeled by applying an X gate. The Kraus representation is $\mathcal{N}(\rho) = (1\text{-}s)\rho + sX\rho X$, where s is the probability of a bit flip error.
- **Phase flip:** This error affects the phase of the qubit, rotating around the Z axis, mapping $|1\rangle$ to $-|1\rangle$. Phase flip errors are modeled by applying a Z gate. The Kraus representation is $\mathcal{N}(\rho) = (1\text{-}s)\rho + sZ\rho Z$, where s is the probability of a phase flip error.
- **Depolarizing:** This type of error describes the process in which a 1-qubit quantum state loses its quantum information and decays into an incoherent mixture of the computational basis states, $|0\rangle\langle 0|$ or $|1\rangle\langle 1|$. The quantum state loses the phase as well

Noise in Quantum Circuitry

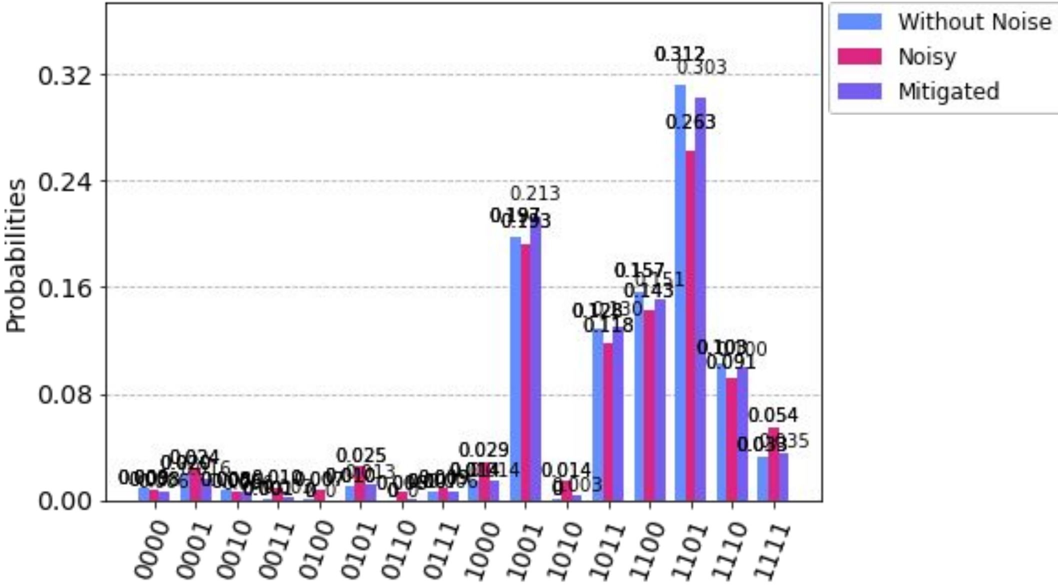
as the superposition between basis states. The Kraus representation is $\mathcal{M}(\rho) = (1-s)\rho + s/3\,(X\rho X + Y\rho Y + Z\rho Z)$, where s controls the strength of the noise.

- **Measurement Noise** generate when the circuit is to extract bit strings as an output. For n qubit final measurement, it extract n bit strings out of the 2^n possible bit strings. It can imagine that the measurement first selects one of these outputs in a perfect and noiseless manner, and then noise subsequently causes this perfect output to be randomly perturbed before it is returned..

- **State preparation noise** generates during the preparation of quantum states. When preparing states using rotation gates, some error occurs in the amount by which rotation occurs about the axis we choose. This causes inconsistency with actual data.

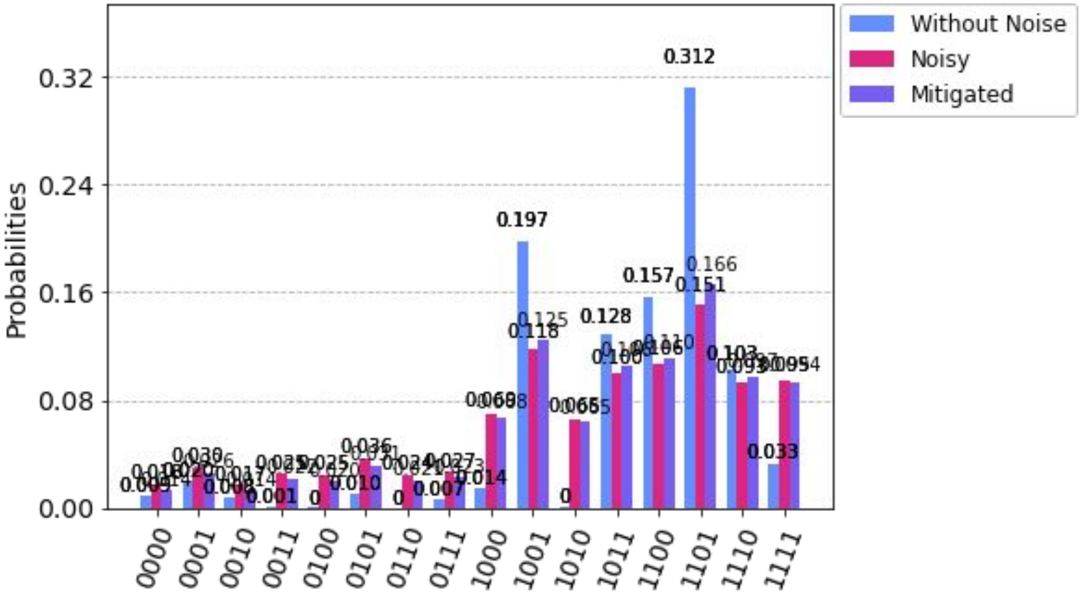
Noise in QSVM

- In QSVM, measurement error is applied to measurement and measured the output.
- To reduce the error, measurement error mitigation technique is applied and measured the output again.
- Finally both output results plotted against without noise measurement

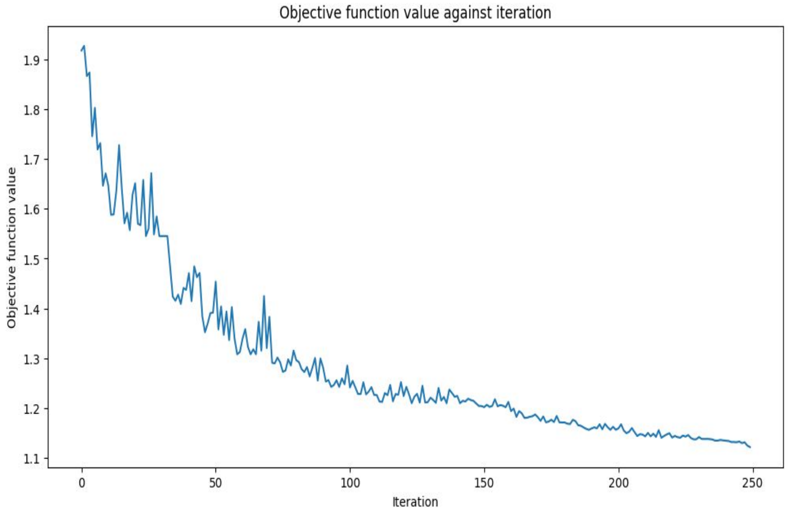


Noise in QSVM

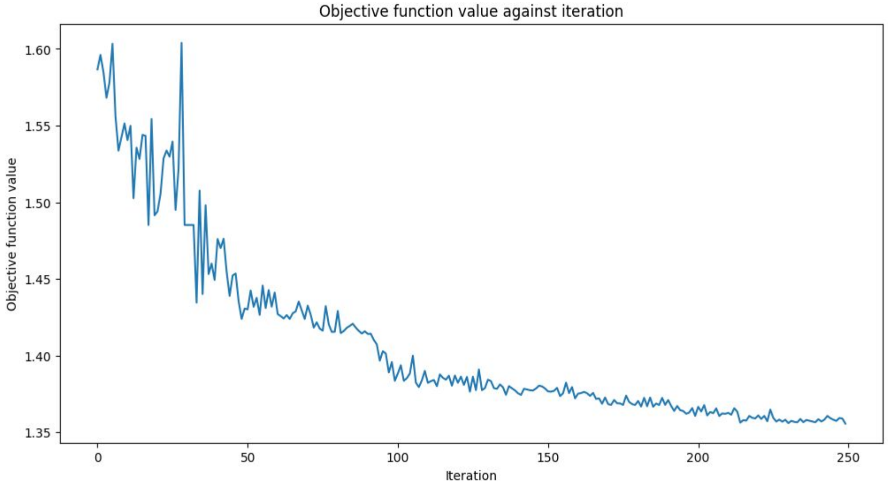
- With the measurement error depolarizing error also applied on every gates in the QSVM and measured the output.
- measurement error mitigation technique is applied to reduce the measurement error.
- Output results plotted for comparison.



Comparison of VQC performance without and with Noise



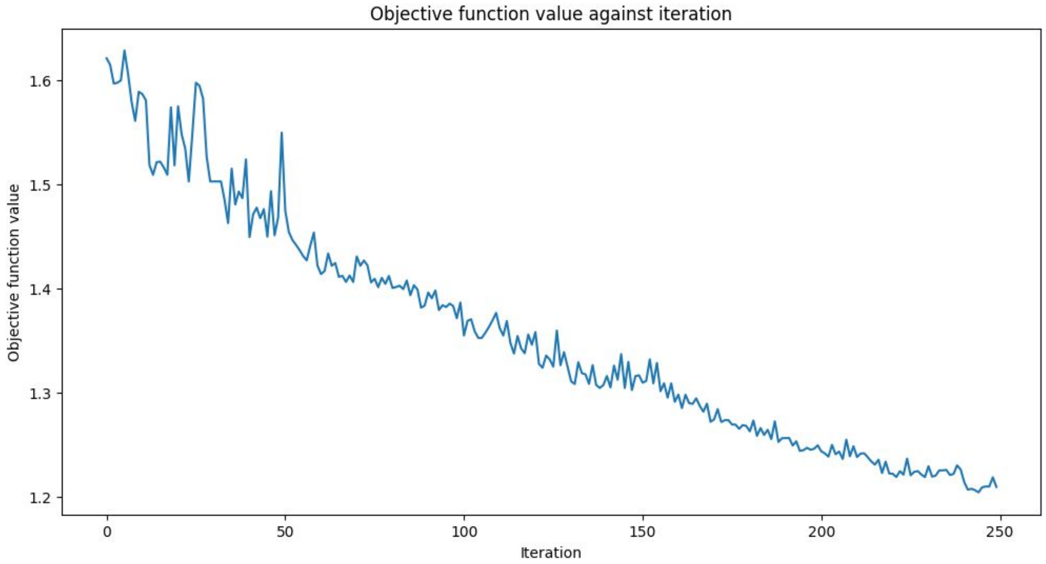
Performance without Noise



Performance with Noise

- Noisy simulation of VQC was done using noise data of ibmq_vigo
- In Noiseless simulation, objective function attains much lower value than in Noisy Simulation for a given number of iterations

VQC performance with Noise and Measurement Error Mitigation



- We used CompleteMeasFitter to mitigate the measurement error.
- After measurement error mitigation, objective function converges to much lower value even in the presence of noise

Conclusion

Quantum Machine Learning algorithms have the potential to solve certain kinds of problems much faster than classical algorithms. Some advantages are:

- Learning capacity improvements: increase of the capacity of associative or content addressable memories.
- Less training information or simpler models needed to produce the same results or more complex relations can be learned from the same data.
- In this project, we have attained accuracies by training the iris dataset with QML algorithms (VQC, QSVM) with or without the noisy scenarios.
- The best possible accuracy for VQC is 76.67% and for QSVM, it is 96.67%.
- Accuracy is not affected by noise upto a certain threshold.

References

1. Qiskit Textbook, https://qiskit.org/learn/
2. Characterizing the Reproducibility of Noisy Quantum Circuits. S. Dasgupta, T. Humble. https://www.mdpi.com/1099-4300/24/2/244
3. Maria Schuld and Francesco Petruccione, *Supervised Learning with Quantum Computers*, Springer 2018, doi:10.1007/978-3-319-96424-9.
4. Vojtech Havlicek, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow and Jay M. Gambetta, *Supervised learning with quantum enhanced feature spaces*, Nature 567, 209-212 (2019), doi.org:10.1038/s41586-019-0980-2, arXiv:1804.11326.
5. Maria Schuld and Nathan Killoran, *Quantum machine learning in feature Hilbert spaces*, Phys. Rev. Lett. 122, 040504 (2019), doi.org:10.1103/PhysRevLett.122.040504, arXiv:1803.07128.
6. Vedran Dunjko, Jacob M. Taylor and Hans J. Briegel, *Quantum-Enhanced Machine Learning*, Physical Review Letters 117 (13), 130501 (2016) doi:10.1103/PhysRevLett.117.130501 arXiv:1610.08251
7. Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik and Jeremy L. O'Brien, *A variational eigenvalue solver on a quantum processor*, Nature Communications, 5:4213 (2014), doi.org:10.1038/ncomms5213, arXiv:1304.3061.

Thank you