QOSF Cohort 7 - Task 3 - QSVM

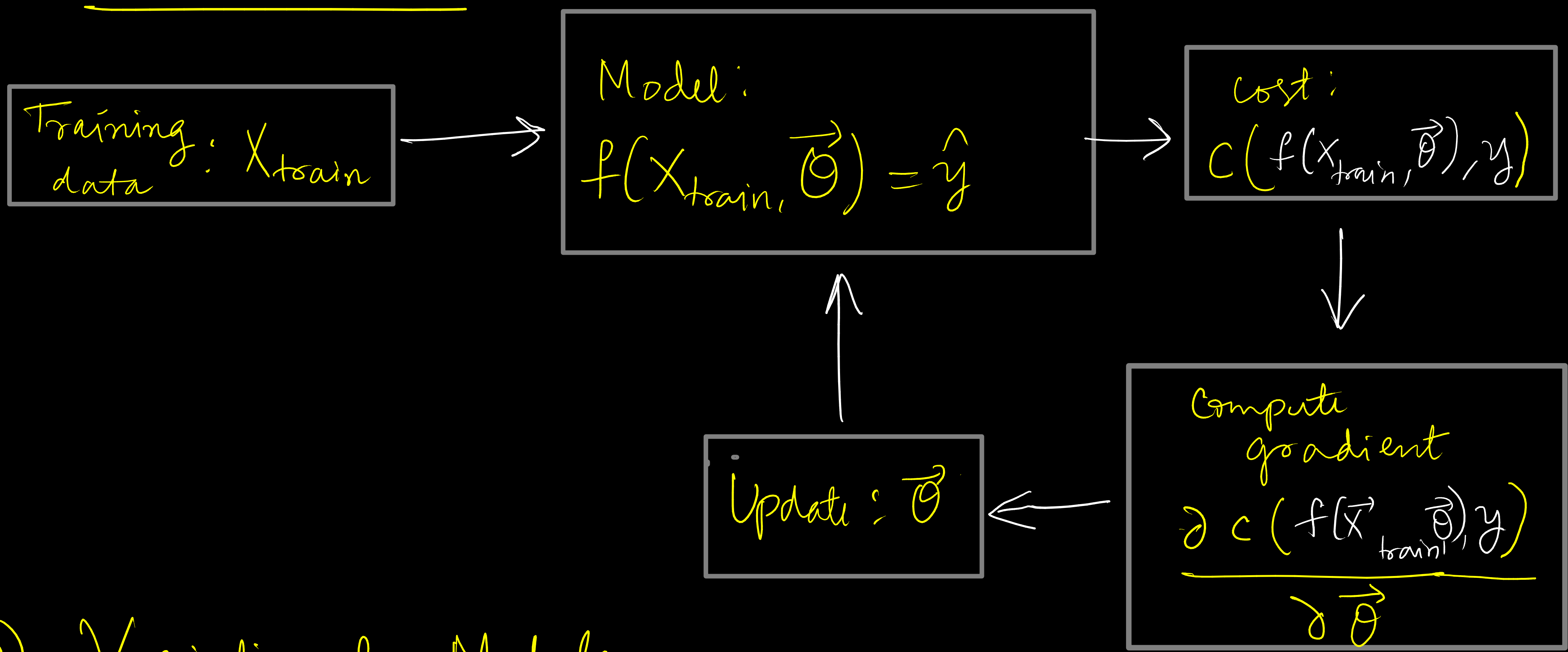
Submission by Harishankar P V
Dated: 5 March 2023

harishankarpv9@gmail.com

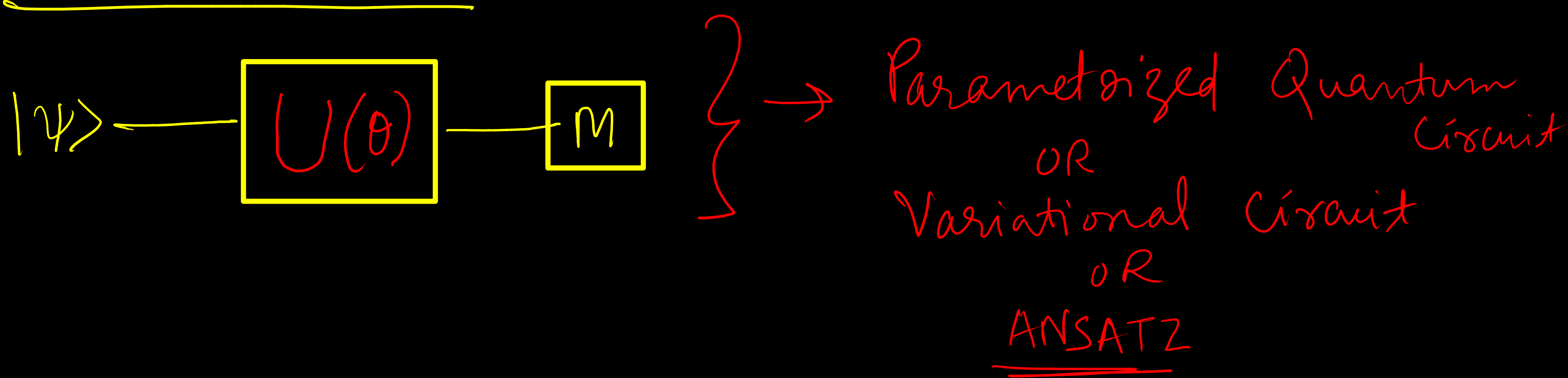
Generate a Quantum Support Vector Machine (QSVM) using the iris dataset and try to propose a kernel from a parametric quantum circuit to classify the three classes(setosa, versicolor, virginica) using the one-vs-all format, the kernel only works as binary classification. Identify the proposal with the lowest number of qubits and depth to obtain higher accuracy. You can use the UU† format or using the Swap-Test.

NOTE: The content below is part of my Quantum Machine Learning notes, originally taken from Qiskit Summer School 2021 lectures. It's purpose is to given an idea to the mentors of my current understanding of QML and QSVM in particular. I have reproduced my handwritten notes as pdf specifically for this purpose.

1.1) Classical ML model:

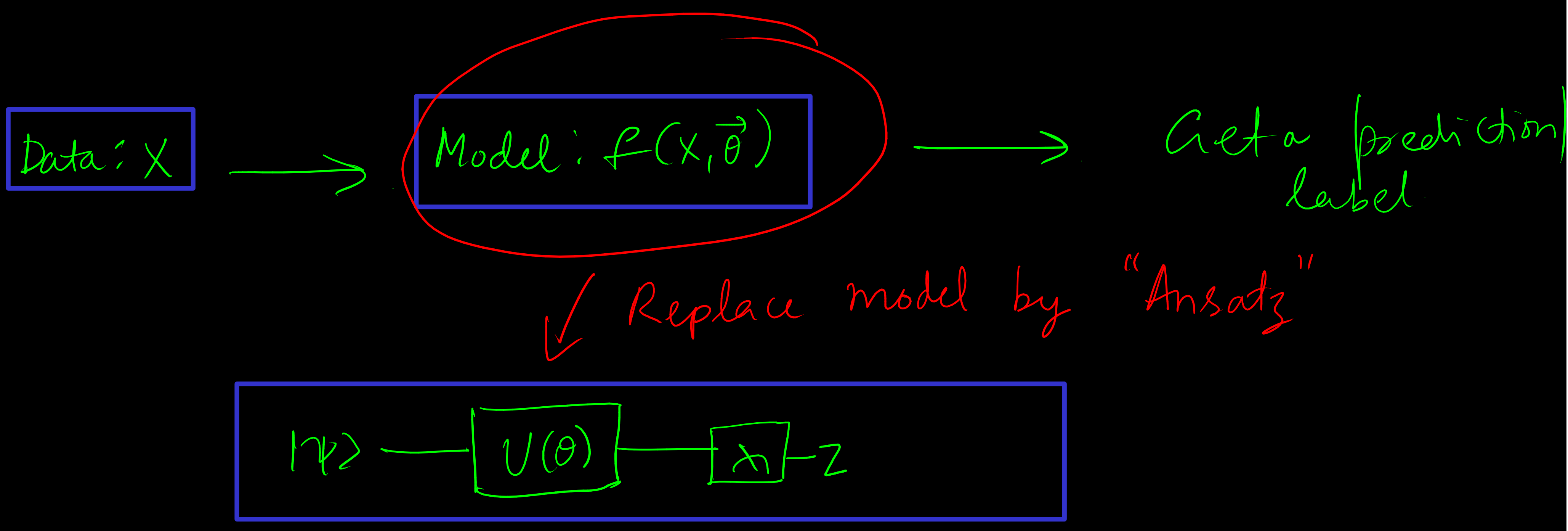


1.2) Variational Models



Parametrized Quantum Circuit / Ansatz is a quantum circuit parametrized by a parameter 'θ', which can be trained and optimized.

1.3) Variational Model as a classifier



Task: Train a quantum circuit on labelled samples in order to predict labels for new data.

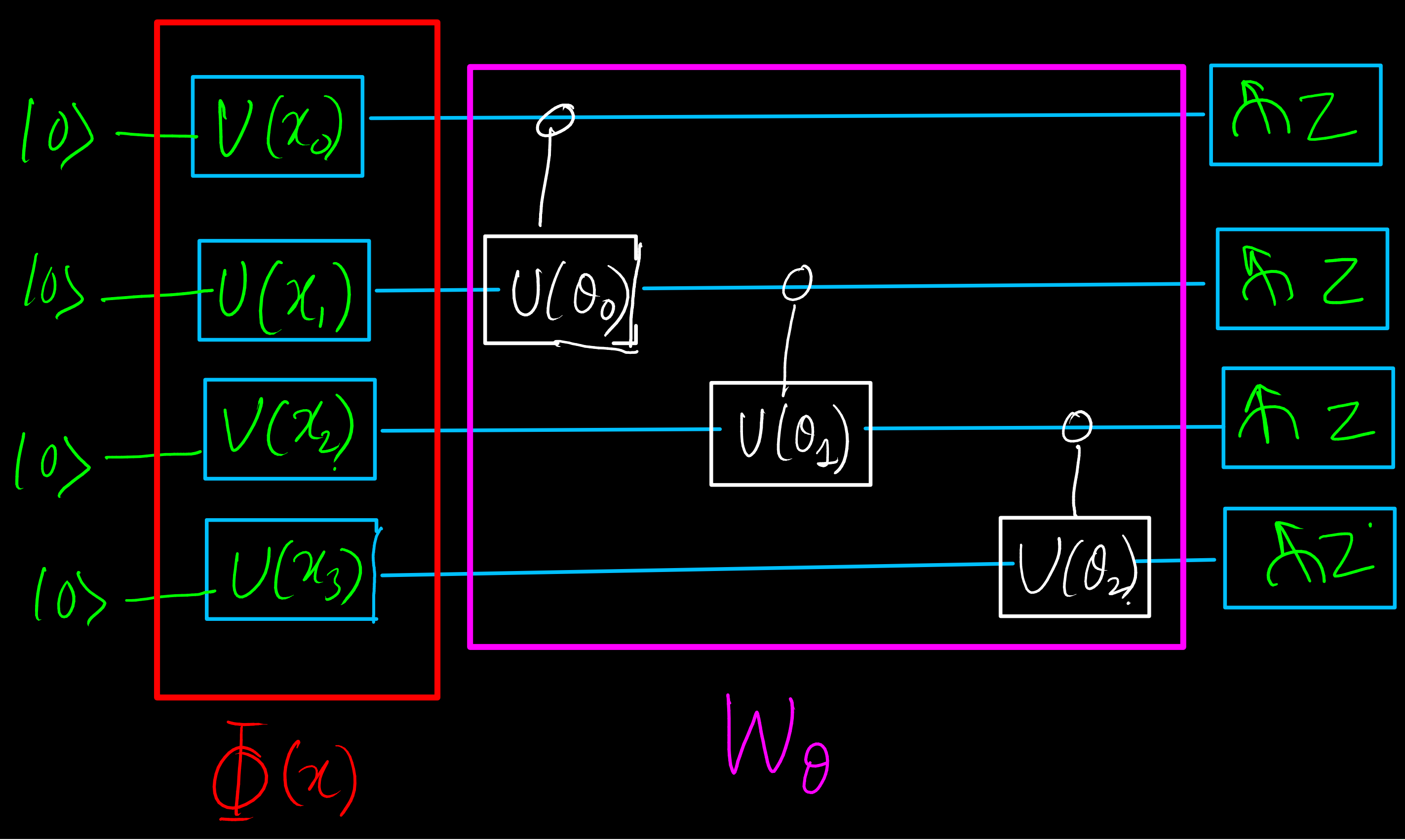
Step 1: Encode classical data into a quantum state.
This can be done by
- Basis encoding
- Amplitude encoding
- Angle encoding etc.

Step 2: Apply a parametrized Model

Step 3: Measure the circuit to extract labels.

Step 4: Use optimization techniques to update model parameters.

1.4) Variational Quantum classifier (VQC)



- Feed data into parametrized gates
- Measure a Pauli Observable (Z) to assign label
- Optimize the remaining parameters to improve performance.

NOTE: - Eigen values of Pauli matrices are ± 1, hence their expectation value always lies b/w -1 & $+1$

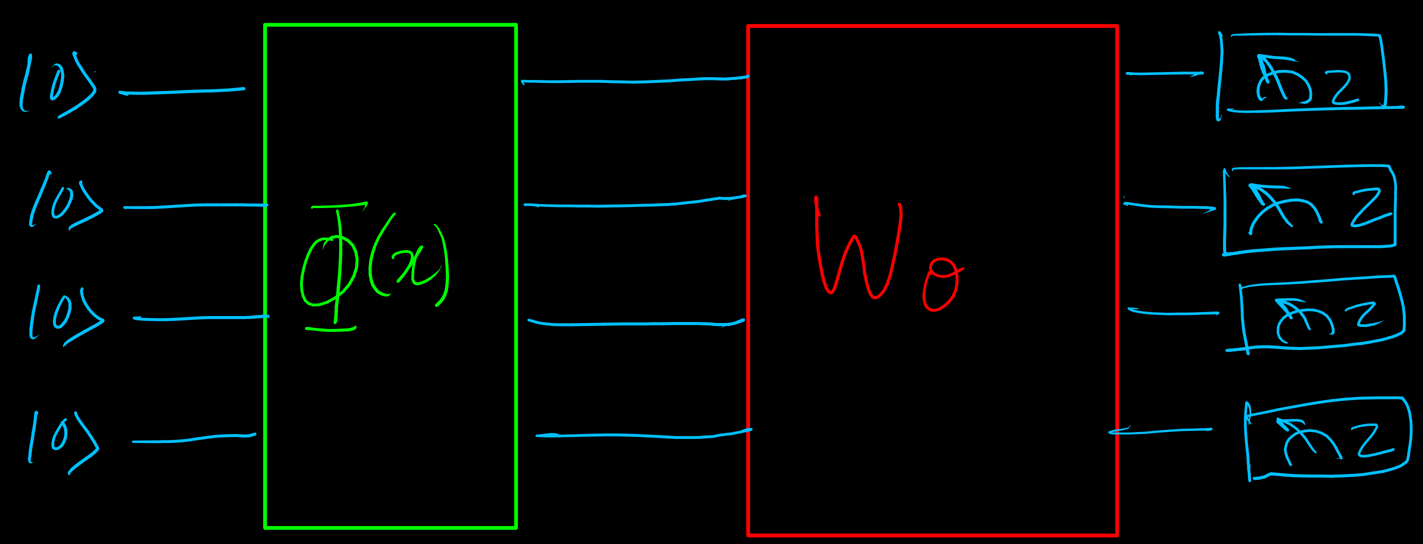
$$\langle \psi | P_i | \psi \rangle \in [-1, 1], \text{ where } P_i \in \{I, X, Y, Z\}$$

- Any observable (Hermetian matrices) can be decomposed in the Pauli basis

$$H = \sum_i c_i P_i \quad \text{where} \quad P_i \in \{I, X, Y, Z\}^{\otimes n}$$

$$\langle \psi | H | \psi \rangle = \sum_i c_i \underbrace{\langle \psi | P_i | \psi \rangle}_{\text{These are easy to compute!}}$$

Redrawing the VQC circuit as:



$$f_\theta(x) = \langle \phi(x) | W_\theta^\dagger Z W_\theta | \phi(x) \rangle \in [-1, 1]$$

Choose a threshold $b \in [-1, 1]$

$$\text{label}(x) = \begin{cases} +1 & \text{if } f_\theta(x) \geq b \\ -1 & \text{if } f_\theta(x) < b \end{cases}$$

Here, $\phi(x)$ is the "Quantum Feature Map" & W_θ is optimizing over our choice of Hamiltonian.

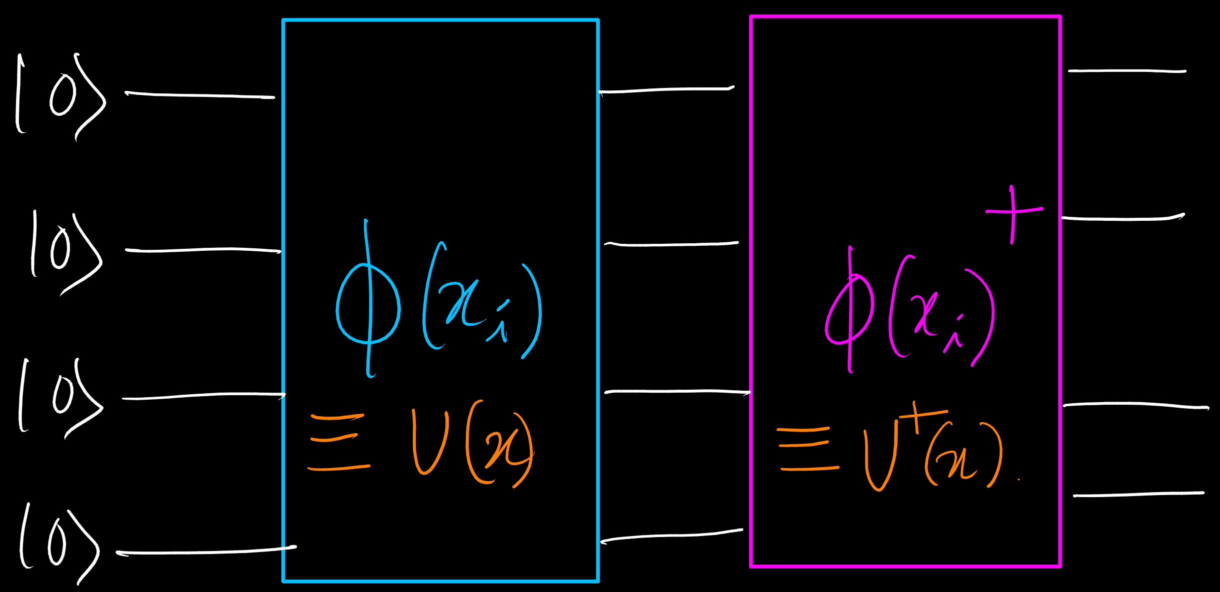
But, It can be shown that VQC is equivalent to a Linear classifier.

This is because

- W_θ parametrizes a small subset of our possible hyperplane.

This can be solved by computing the "Kernal function" for the feature map.

2) Quantum Kernal Estimator (QKE).



Start with zero, evolve forward in $U(x)$, then evolve backward in $U^\dagger(x)$ and count how many times zero is measured.

This transition amplitude i.e., the probability of measuring zero bit string on the output is exactly the Kernal value

→ other way to do it is using swap test.

Estimating K_{ij} on a Quantum Computer.

Final state after forward & back evolution is

$$\phi(x_j)^\dagger \phi(x_i)|0\rangle$$

The probability of measuring all zero-bit string:

$$Pr[|00\cdots0\rangle] = \left|\langle 0|\phi(x_j)^\dagger \phi(x_i)|0\rangle\right|^2$$

$$K_{ij} = \left|\langle \phi(x_j)|\phi(x_i)\rangle\right|^2$$

After estimating kernal entries, rest is a convex optimization problem, solutions of which will be a set of Lagrange multipliers (scalars one for each of our training data points), which is going to tell us which of our data points is a support vector.

$$Label(s) = sign\left(\sum_{i\in N_S} \alpha_i K(x_i, s) + b\right)$$

The Rest can be done on a classical machine.

$$X \in \mathbb{R}^n \xrightarrow{\text{encoding}} \phi(x) \in V \xrightarrow{\text{classification}} \{y_i\}$$

	encoding	classification
SVM	classical	classical
QSVM	Quantum	classical