Task Statement:

Please comment on quantum computing or quantum machine learning. You can also comment on one quantum algorithm or one quantum software you are familiar with. You can also suggest methods you think are good and you would like to work.

My comments:-

Quatum Computing

From the first time that I heard about qubits, I had always wanted to see their workings fisthand. I got this opportuninty when I participated in the Qiskit India Challenge. Our problem statement was rather straight forward: we had to build a **Variational Quantum Classifier** that classified between the digits 4 and 9.

This was the time when I was first introduced to the paradigms of Quantum computing. As interesting and exciting it all seemed, it was made clear to us that Quantum computers could not replace classical ones.

With Google's constant claims about Quantum supermacy and the fact that they were able to solve a problem that would take classical computers roughly 10,000 years, I never quite understood why we coudn't start using Quantum computers instead. It was soon made clear to me why this was the case. The difficulty of maintaining and managing qubits at room temperature is far more difficult than it seems.

Then I wondered that if this was the case and quantum computers in every office and household was going to take time, why were we learning Quantum algorithms and computing now? It took me 2 weeks of constant brainstorming through the study materials to answer my questions. The reason was that there were specific tasks that were much more suited to Quantum way of doing things. An example would be the good old **convolution** operation we use almost everywhere in computer vision. Using Quantum circuits and methods this operation could be done more efficiently and faster than on a classical computer.

The ability of a qubit to be put into superposition can be harnessed in many ways that can if not replace, but improve the ways in which we perform certain computations.

VARIATIONAL QUANTUM CLASSIFIER AND QISKIT

Variational Quantum Circuits or VQCs were introduced in the paper:- Supervised learning with quantum enhanced feature spaces.

In this paper the authors show that a classification problem can be solved by representing the feature space of the problem by a quantum state, taking advantage of the large dimensionality of the Hilbert space, to obtain a more enhanced solution.

They propose SVM type classifiers that process data provided purely classically and use the quantum state space as the feature space to still obtain a quantum advantage. This is done by mapping the data nonlinearly to a quantum state $\Phi: x \in \Omega \rightarrow |\Phi(x)\rangle\langle\Phi(x)|$.

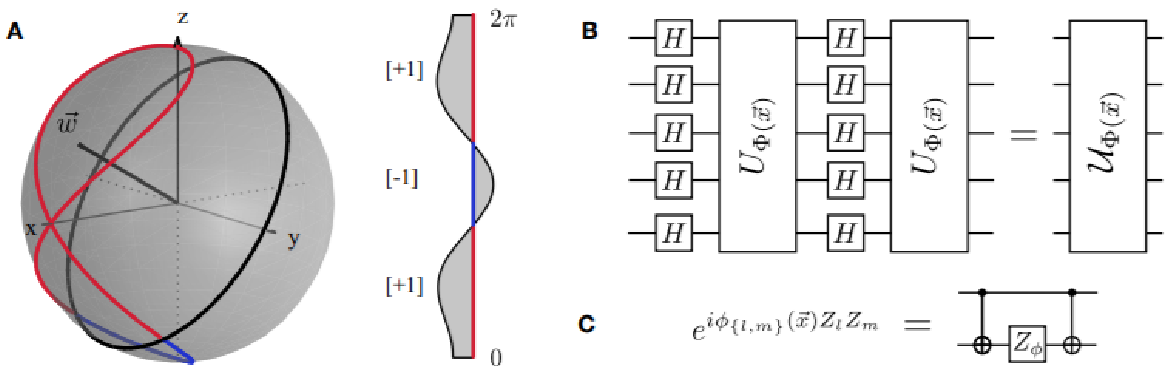


FIG. 1. **Quantum Kernel Functions:** (a) Feature map representation for a single qubit. A classical dataset in the interval $\Omega = (0, 2\pi]$ with binary labels (a, right) can be mapped onto the Bloch sphere (red / blue - lines) by using the non-linear feature map described in (b). For a single qubit $U_{\Phi(x)} = Z_x$ is a phase-gate of angle $x \in \Omega$. The mapped data can be separated by the hyperplane given by normal $\vec{w}$. States with a positive expectation value of $\vec{w}$ receive a $[+1]$ (red) label, while negative values are labeld $[-1]$(blue). (b) For the general circuit $U_{\Phi(\vec{x})}$ is formed by products of single- and two-qubit unitaries that are diagonal in the computational basis. In our experiments, both the training and testing data is artificially generated to be perfectly classifiable with the aforementioned feature map. The circuit family depends non-linearly on the data through the coefficients $\phi_S(\vec{x})$ with $|S| \leq 2$. (c) Experimental implementation of the parameterized diagonal two-qubit operations using CNOTs and $Z-$gates.

In the challenge we were told to use the Qiskit framework. My experience with Qiskit was good as it had a smooth learning curve and within days of being introduced to it I was designing my own circuits. One thing that amazed me in the Qiskit framework was their focus on visualizaton. I thing visualizing circuits helps a lot in getting what you want from the algorithm of your choice.

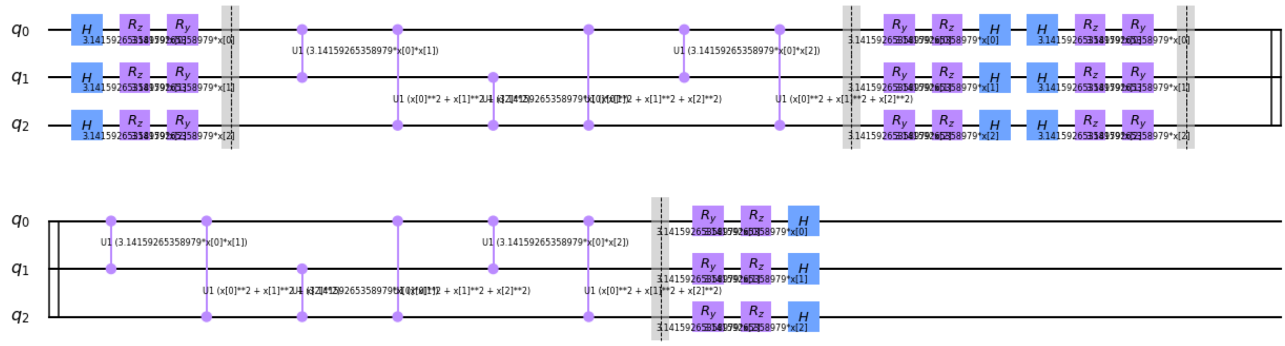
The first step in the algorithm was to build a **quantum feature map** that would encode the classical data onto the quantum computer. I invested majority of my time in this part as I believed a good feature map would result in a better classification, beacuse it was all about taking adavantage of quantum computing to represent classical data in the best way as possible. I eventually ended up making a custom feature map using the **FeatureMap** class of the **aqua** library in qiskit.

The next part was to build a circuit that classified the data. I used a short depth variational circuit on the previously created feature map. The parameters of this circuit were trained in a classical optimization loop until the data was classified. I used the **TwoLocal** class of the qiskit.circuit library for this.

Finally we had to design the classical optimization loop. I used the **COBYLA** :-Constrained Optimization By Linear Approximation optimizer for this purpose.

In the end I ended up with an accuracy of 0.805, on the test set.

The feature map used by me:-



Comments on methods I think are good and would like to work on:-

According to my experince in this field, I like Hybrid methods, which are a combination of both quantum methods as well as classical ones. For example: in Quantum Generative Adverserial Networks, designing the Generator with quantum circuits but designing the discriminator using the plain Sequential class results in good accuracy. My justification for this is the fact that more research and resources are available for the classical methods. Also the classical methods can be used to provide insight into the inner workings of the quantum circuits.