Introduction

Since the discovery of Quantum Mechanics, people have found the behavior of the laws of probability in Quantum Mechanics counterintuitive. Quantum Mechanical phenomena behave quite differently than the classical physics that we are used to. Feynman [Simulating physics with computers] was the one of the first people to ask questions about what effect these quantum mechanical laws will have on computation. He gave arguments to simulate quantum mechanics of classical computers (Von Neuman). He also mentioned the possibility of using a computer based on quantum mechanical phenomena to avoid this problem. With this he raised a converse question: ‘by using quantum mechanics in a computer can you compute more efficiently than on a classical computer.’ Along with this there has been parallel work by Benioff [turing Hamiltonian Benioff] in the field of quantum mechanics and computing. He showed that a Turing machine can be simulated by unitary evolution of quantum process. After which Deutsch [Deutsch algorithm] was the first to give an explicit quantum model for searching algorithm in database. He defined the quantum circuits and Turing machine and investigated some of their properties. Followed by the Shor’s algorithm [Shors Algorithm paper] for discrete log and factorizing considerable number on Quantum Computers with polynomial time complexity

There has been significant research and advancement in quantum computing, including efforts to simulate quantum systems on classical machines and develop quantum algorithms with practical applications. Developments in quantum hardware, such as superconducting qubits, ion traps, neutral atoms, and quantum dots, have fueled these advances and led to more robust quantum processors. These improvements have paved the way for exploring how quantum algorithms can be applied to solve complex computational problems in new and more efficient ways, including those in the realm of machine learning.

Machine Learning Algorithms are tasked with extracting meaningful information from the given set of data and making predictions on the data. Machine learning algorithms construct and/or update their predictive model based on the input data. One such algorithm in the field of machine learning is SVM’s. SVM finds a fitting hyperplane between the data classes and maximizes separations between classes to enhance the model's predictive performance. In classical Machine learning, SVM’s are widely used for their effectiveness, especially into higher dimensions.

The intersection between machine learning and quantum computing has given rise to new fields of quantum machine learning and has been researched in the last few years. In classical machine learning, ‘kernel methods’ are well established field, which is responsible for increasing the dimensionality of the data. In short, the idea of kernels is to embed data into higher dimensional feature space to analyze it easily. However, computations performed on quantum computers are computed in an intractably higher dimension. Hence the data embedding circuits in quantum computers can be treated as Feature maps in kernel methods. The idea is to implement the so called ‘Kernel Functions’ on quantum computers, which are used to obtain the distance/similarity metrics between the datapoints.

For complex and non-linear datasets, there is a need to maximize the alignment between data classes to simplify the task for SVMs to find an optimal hyperplane. Which can be achieved by popular kernel training algorithms like Target Kernel Alignment, Centroid Kernel Alignment etc. Now for training the quantum kernels a kernel matrix, which is a square matrix representing the distance metrics between each datapoint must be computed in each iteration of training. The complexity of training set goes to O(n²) which can be an overhead for these many numbers of circuit executions.

There have been studies to reduce the number of circuit executions using batched random sampling, kernel training with Pegasus and stochastic gradient descent, QUACK centroids-based kernel alignment, that reduces the circuit execution overhead to greater extent without compromising on the accuracy of models.

Our assignment in the study is to study different kernel alignment techniques and optimize the circuit executions overhead further than the algorithms mentioned above. To achieve this task a study has been carried out on two different models of Greedy Sampling and Clustered based Centroid Kernel Alignment (CCKA) that reduces the circuit executions to O(c) where c is the number of clusters selected for the dataset. The mentioned algorithms keep the complexity consistent for increasing size of datasets and provide similar accuracies as other algorithms with much less complexity.