Aleksandra Mulewicz

Is quantum the new oil?

The story of quantum machine learning fighting classical models.

Nowadays, the impact of nature on multiple areas of our lives is constantly increasing. One of the objectives of this essay is to find out if the contribution of quantum computers can increase the efficiency of machine learning models. This study highlights two types of quantum advantage - comparison of classical algorithms with a quantum equivalent and ideas derived primarily in the context of quantum computing. The research was based on multiple papers from relevant sources such as original IBM documentations or university professors' papers - I compared different views on the topic and found a lot of surprising facts. What is emphasized however is the critics of quantum machines supported mostly by the cost and time taken to evaluate these algorithms. While acknowledging the impact of quantum machine learning, my study discovers the power of optimization in simulators that are not actual quantum computers that is caused by advanced mathematics solutions.

Quantum Annealing

Quantum and classical computers follow the candle-lightbulb analogy - lightbulb isn't just a better candle as it is working completely differently. Some systems are completely different from the ones we are used to know as they aren't based on commands but logical rules that are run simultaneously.[1] These programs are called quantum annealers and adiabatic quantum computers. The whole point of these algorithms is to find an optimal solution to a given problem with a use of the tunnel effect.

[1] "*Quantum Annealing - Quantum Technology." Quantum Flagship,*

The process starts with big cumulation of qubits in a state of superposition that are independent in this environment[2]. Then we introduce rules which are actually biases and couples we need for our task. To introduce bias to our qubits we apply an external magnetic field that minimizes the energy of one of the values - 0 or 1 that lets us manipulate our quantum information. The next thing we can do at this stage is to link them together and by lowering the energy of combinations decide if we want the coupler to have opposite or the same values in these units of information. In the end of annealing what you are left with is the solution to your problem with the minimum energy state possible. It all happens in twenty microseconds.

When it comes to implementing it for classification tasks, the model that can be used for generalized and interpretable data is quantum annealing classifier[3]. In this model, two hamiltonians (mathematical description of energy of the system) are introduced - known and the one who represents our problem. To find a solution the second one needs to evolve to the first state[4].

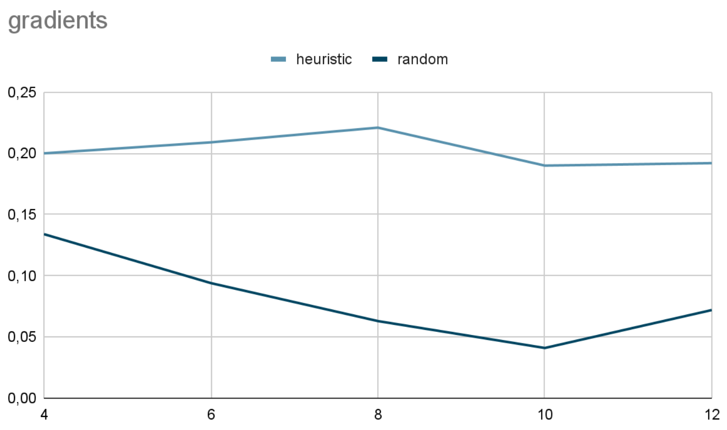
Variational Quantum Classifiers.

One of the best advantages that quantum devices hold is a better way of storing and stimulating chemical reactions as every new equation increases cost function exponentially - every new qubit however doubles the space in the model. The variational quantum eigensolver uses this fact to estimate the system's ground state energy by finding a guess that represents the wave function - it represents the molecule. So this algorithm is a hybrid model that uses the quantum part to calculate hamiltonian and classical to optimize the whole circuit.

These algorithms, to deal with quantum circuits, have to find an optimizer based on gradient manipulation to avoid the phenomena called a barren plateau - which is an exponentially vanishing gradient. One of the reasons for the big plateau is randomness - we can't just simply guess random ansatz.

An interesting approach is a heuristic introduced by Grant. It lets us start really close to random but still not quite. The so called identity block remains purposeful - chosen such that initial transformation is left undone so the identity will remain. This idea doesn't guarantee that our model won't fall into a barren plateau - it just lets us start without them.

[2] *D-Wave Systems. "How The Quantum Annealing Process Works." YouTube*
[3] *"Quantum Annealing - Quantum Technology." n.d. Quantum Flagship.*
[4] *"Quantum Classifiers." n.d. Qu&Co.*



The signals in the heuristic model are significantly stronger than the ones in a truly random algorithm. We lost this holistic randomness but our system's work is valid[5].

However after I tried coding this classifier, it took 5 hours and the kernel broke. It is really possible that the problem was caused by my dataset but still I couldn't prove the theoretical quantum advantage.

Support Vector Machine.[6]

In this type of classification what is abstracted behind the model is the graph plotted with features and their labels (training data). After collecting data it plots a hyperplane - a line (or plane) that separates points from each other in N(number of x's)-dimensional space. The margin between the hyperplane and data points needs to be significantly big so we can identify labels effectively for future inputs.

Referring to this fact, we can identify how our cost function works - it increases the margin size. Support vectors are the whole dynamics of this system - these points are closest to the margin, therefore, they are responsible for the position of the hyperplane. Consequently to maximise our space between points and the line, we need to simply remove those vectors. Even if it is said to be one of the best classifiers it has some disadvantages. Large problem arises when attributes of two classes overlap and then the label won't be clear and accuracy may desperately drop.

[5] "Barren plateaus." 2022. TensorFlow.
[6]"Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi."

Quantum SVM[7].

The whole process starts with encoding classical data into a quantum state - in the case of this algorithm the feature maps were used, the labels take values one and minus one. Then before training our model - that works really similarly to the classical way as this system is hybrid, we need to calculate the kernel matrix[8]. The main quantum advantage we can see here is with the data that isn't easily stimulated by classical machines - such as chemical reactions or molecules.

Not all of the quantum models are perfect however - the really important point is that when we implement SVM to quantum, there are more than possible ideas. Some of them are better, some worse but still exactly like with classical ones every one of them has some advantages in different cases. One of the more faulty implementations is the one of least-squares support vector machines - the more features, the programme becomes less stable. Even though quantum properties can obviously deal with this problem, in the meantime there are some new inaccuracies such as:

- There is no solution if the inverse matrix does not exist and the whole program collapses.

- The whole kind of these operations is susceptible for mistakes and errors - in this field it makes it mostly invalid.

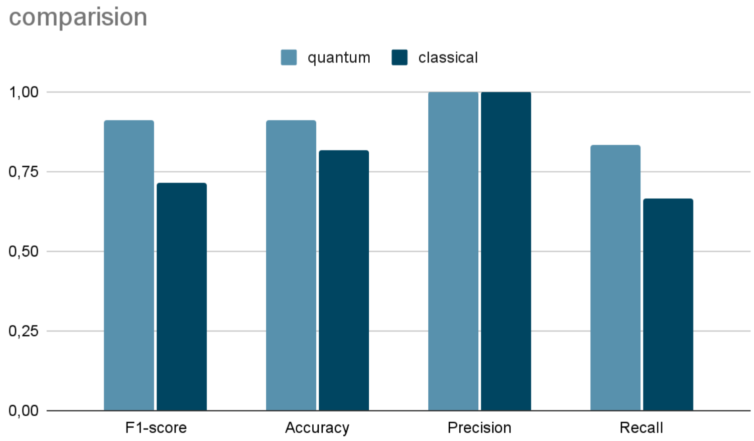
- If our parameters aren't high in quality then, the possibility of errors significantly increases as they are a really important part of this model.

This is the optimization problem and with classical classifiers we have lots of algorithms and equations that help us with these errors. In the quantum world, as it is relatively new, there are only a few of them. One of our options is working with a gradient and using it to increase efficiency of our model along with using quantum annealing. It was proved that this idea is really efficient but what it does is decrease the number of iterations that we use to find the minimum of the function.

After comparing the classifiers, it was observed that quantum classifiers get much better results than the classical one in this particular case.

---

[7] Zhang, Rui. 2022. "Quantum support vector machine based on regularized Newton method." ScienceDirect.
[8] Huembeli, Patrick. 2019. "Introduction into Quantum Support Vector Machines | by Patrick Huembeli." Medium

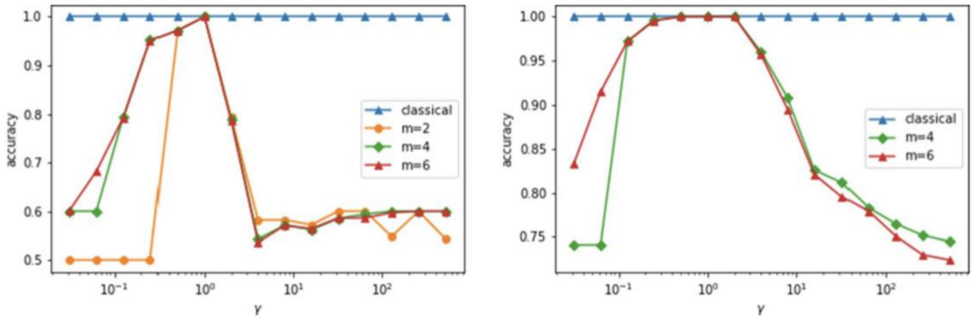


Analyzing the results we can see that with 100% precision of both algorithms the quantum SVM received much better results - for instance increasing accuracy by 11%. The F1-score[9] is the score that is responsible for the overall performance of a classifier - it is a harmonic mean of precision and recall[10].

To obtain the most correct results, the datasets used were the same for both models. It concerned the flow separation concept in aerodynamics. I feel like the type of data used had a big impact on our results as even if the data is not completely quantum it is very advanced and clearly physics related.

Following the rule, the next possible quantum implementation of SVM involves the Kronecker kernel. The data used for this experiment is a digit classification dataset and the idea follows the least-squares support vector machine. Both of these ideas combined were doomed to failure - classical computers are so well trained on this particular data manipulation that using quantum circuits is just a waste of time and resources. The LS-SVM however was proved to work for small numbers and then become really unstable for bigger numbers just as commonly known.

[9]Educative Answers Team. n.d. “What is the F1-score?” Educative.io.
[10]Quantum support vector machines for aerodynamic classification

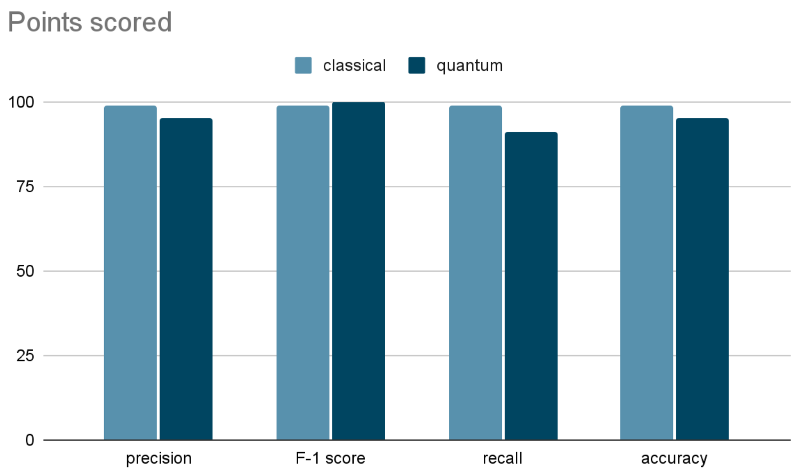


(a) Linear kernel

(b) Quadratic kernel

The next thing that makes us doubt the validity of this experiment is a perfect 100% accuracy for every data input no matter the size of the y. If you obtain these results what is there to optimize? Secondly the perfect score isn't often and when it happens it is mostly fortuity of randomness - machine learning models are never obtaining such results[11].

For my project I used pegasos quantum SVM model. Firstly I had to create a feature map of 2 dimensions and then build a quantum kernel. After changing the feature map dimension to 4, the results increased.



It can be seen that the F-1 score was the only increased value. No matter how many dimensions I used the model still couldn't predict pulsar star (1)

[11] Nohara, Taisei.. "Pairwise classification using a quantum support vector machine with Kronecker kernel."

Summary.

The evaluation of the models in practical manners proved itself to be really advanced and complicated - after hours spent on variational quantum classifiers and quantum k-nearest neighbors I decided to leave it and concentrate on developing a quantum support vector machine. What I came across doing research were some incredible optimization ideas that proved itself to be working. Qiskit models didn't prove themselves in practice but I found an optimization technique - changing dimensions. Quantum - inspired machine learning brings effects but they aren't critical therefore we need more years for developing quantum hardware. However it is really important to study the software so we can further understand the art of quantum computing.