QEML Project Relevant Information

Software Documentation

Links to all code: https://github.com/siddrrsh/QEML

Link to ArXiv preprint for this research: https://arxiv.org/abs/2002.10453

This research was not conducted at an RRI institution and was done using free open source software

Qiskit Textbook: (Where I learned most of the background for the project):
https://qiskit.org/textbook/preface.html

IBM Quantum Experience Platform: https://quantum-computing.ibm.com/

FAQs and Answers to any questions of interest:

Q. Why did you choose the KNN classification algorithm to make a quantum variant? Why not decision trees or random forests, etc.?

A. We can only benefit from quantum improvements in execution and performance if the classifier is a deterministic or kernel classifier. Algorithms like the decision tree and random forest rely primarily on boolean logic compared to spatial reasoning and distance comparison between data points. Thus, we cannot apply quantum parallelism and an enhanced feature space in those classifiers. I initially wanted to implement the Support Vector Machine (a kernel method), but a quantum SVM (qSVM) already exists in the public domain. (https://github.com/Qiskit/qiskit-community-tutorials/blob/master/machine_learning/qsvm_directly.ipynb)

Q. What is a kernel method/kernel classifier?

A. A kernel method is a trick used in machine learning to model a nonlinear problem with a linear classifier. Kernel classifiers require transformations of the feature space of the data (with implicit computation of the inner products). Examples include Principal Component Analysis, Ridge Regression, etc. While the KNN is not a kernel method, it does implicitly mimic that style of classification.

Q. What do you mean by a "Quantum Feature Space"?

A. This is the idea that all the data points lie in an enhanced feature map where properties like superposition and entanglement apply. This concept allows us to reduce the size of the feature map from exponential to linear. More information on the mathematical implementation can found in this paper: https://arxiv.org/abs/1804.11326

Q. What was your data and where did you get it from?

A. My data was in the form of .csv file from the Wisconsin Research Data Archives. I used *the Wisconsin Breast Cancer Dataset*, a set with 10 dimensions and features. This dataset was ideal for assessing the performance of QKNN against the KNN since it has enough dimensions such that QEML should provide a tangible benefit but is not too large in that both ML and QEML will become computationally expensive.

Q. Some of these concepts have already been articulated in the scientific community, what is your contribution?

A. My scientific contribution to the field of quantum machine learning is that I am providing concrete implementations of an algorithm which has not yet been implemented in the public domain with an innovative approach of using Hamming distance. I am also doing a direct comparison between QKNN and KNN to evaluate the proposed benefits of QEML. I'm not really reinventing the wheel here.

Q. Why are properties like Fidelity and Grover's Algorithm "crucial" to demonstrating QEML?

A. These concepts form the backbone of QEML since they are a starting point for more advanced properties like regression and classification. Traditional Machine learning is heavily dependent on pattern analysis and feature extraction. To be able to differentiate between data points (inputs) and their respective labels, we must show that properties like minimum path search and cosine similarity exist. In a quantum environment, these properties aggregate up to Grover's Algorithm and Fidelity. QKNN would not be possible with these two algorithms.

Q. How were you able to learn how to manipulate quantum software (Qiskit, IBM Q)?

A. I got started with this youtube series: (https://www.youtube.com/watch?v=M4EkW4VwhcI) and read through portions of the qiskit textbook (linked above). There is also a major Qiskit Slack community which I was able to consult for technical help.

Q. What inspired you to do this project?

A. This video kickstarted my interest: https://www.youtube.com/watch?v=-ZNEzzDcllU When I heard that Google had achieved quantum supremacy, I was super interested in learning about it since it seemed like a big deal! I slowly started to understand what quantum computing was and what it means for us. Since I was already extremely passionate about Machine Learning, I wanted to combine my interest in both fields into a groundbreaking project.

Q. How did you get the numbers for accuracy?

A. I was able to test the Scikit-Learn KNN which uses Euclidean distance with the Score metric. I was able to test the QKNN circuit by transpiling the circuit and finding the state vector output for 1,024 shots on the IBM QASM.

Q. When you implement Hamming Distance, how do you go from "a+1" to "a+d" circuit?

A. To go from $a + 1$ to $a + d$, we apply the $CNOT$ gate and X gate to move from ancillary qubit addition to a circuit where we can implement Hamming distance.

Q. If you could summarize your project in one sentence, what would it be?

A. Utilizing quantum computing and properties like superposition and entanglement to revolutionize machine learning and improve performance on higher dimensional data.

Q. Who would possibly be interested in your research and why?

A. I believe that quantum computing researchers would in particular be interested in my conclusions and experiments. Currently, researchers at Google and IBM are on a mission to build a completely fault-tolerant quantum computer and despite achieving quantum supremacy, we are still decades away from a near-perfect quantum computer. Quantum computing is developing along the same path as classical computing: 40 years ago, classical computers were large in size and had low computing power. Today the inverse has become true and we can store abundant amounts of processing power in the smallest chips. Maybe 40 years from now, people will look back at our current quantum computers are joke about how big and impractical they are. Despite the fact that quantum computing has not fully developed yet, my work shows that there are so many possible applications of quantum computing to benefit society - whether it's improving machine learning, discovering new pharmaceuticals, solving problems in robotics and medicine, etc.

<u>Q. Did you do repeated trials and how did you account for statistical variation in the dataset?</u>

A. I did 5 training runs and the results from Scikit-Learn and the IBM QASM simulator are averages. The dataset used is a common benchmark for testing Quantum ML algorithms. A similar breast cancer dataset was employed for analysis of the Quantum SVM. The dataset has unequal incidence of both malignant and benign tumors with about 350 benign and 210 malignant.

<u>Q. What do you mean by Hilbert Space?</u>

A. This is a generalization of Euclidean space. It is an abstract vector space that allows length and angle to be computed. It is useful to quantum mechanics and quantum computing since it is a space which encodes quantum properties like superposition and it is infinite in terms of dimensions.

<u>Q. What problems did you encounter/what were your difficulties?</u>

A. There were two major difficulties involved in this research. The first was that I spent almost a month figuring out how to use IBM Q and Qiskit. I already knew python and how to use jupyter notebooks, but figuring out how to use the simulator and understanding each gate and its respective transformation was a challenge. The next difficulty was the issue of encoding Hamming distance into my QKNN. I read a lot of background literature and research paper which discuss using Hamming distance, but the implementation itself was not straightforward and spent a few weeks drafting possible approaches. I solved this problem using subroutine gates (pi/2). In fact, my final circuit is not the perfect solution and isn't a complete QKNN, but more importantly demonstrates the key points of a QKNN (use superposition, use Hamming Distance, and implement a circuit). One final difficulty was part of the testing phase and involved

<u>Q. What did you learn from doing this project?</u>

A. I learned a lot about quantum computing and about the development process. I specifically learned how much effort it takes to go from an idea to a physical product. Sure, anyone can visualize a Quantum ML algorithm, but to implement one is a completely different animal.

<u>Q. What was your test criteria/hypothesis?</u>

A. I believe that Quantum KNN will be able to outperform the classical KNN due to the 2 aforementioned properties