

CSCI 4511W Writing Assignment 4

Brian Cooper
coope824@umn.edu
University of Minnesota

November 19, 2019

1 Problem Overview

For my CSCI 4511W final project, I would like to analyze various quantum machine learning algorithms (machine learning algorithms designed to operate on quantum computers) and compare them to their classical counterparts. I plan on working by myself in this project. Using my knowledge from previous classes at University of Minnesota (Introduction to Machine Learning, two bioinformatics classes, and Introduction to Data Mining) as well as what I've learned in this class (Introduction to Artificial Intelligence), I am hoping to expand my knowledge of artificial intelligence and quantum computing through this research and am thoroughly excited to work with this material.

Two machine learning tasks are my primary focus: classification (assigning labels to unlabeled data) and clustering (discovering structure among data by grouping). As such, I am planning on using algorithms that fall into these task categories, such as support vector machines for classification and k-means for clustering.

I am planning on applying these quantum machine learning methods to problems in bioinformatics based on my previous class experience. However, the methods can be expanded to other domains if bioinformatics problems serve to be too difficult to implement and work with. Assuming bioinformatics works as a strong domain for this project, there is lots of public, free-to-use data from the UCI Machine Learning Repository [7] and the NCBI Gene Expression Omnibus [9].

2 Software

Open-source technology is a key component in constructing a software stack for this project. Although software-based quantum computing frameworks (such as APIs) are relatively young, there are still some usable, open-source projects. I am planning on using IBM's Qiskit [5], which is a Python framework for quantum computing algorithms. Qiskit offers both direct quantum computer interaction and simulation support, although I am planning on using the simulations since I would have to purchase credit to access their real quantum computers. As Qiskit is open-source, if I need to make any modifications to the algorithms or view the source code, that is entirely possible. The Qiskit API reference and premade methods are not extremely mature yet, so I anticipate having to write and/or modify some of the code myself. Beyond Qiskit, I may use methods from other Python libraries such as scikit-learn [3], which includes several methods for working with machine learning tasks.

3 Domain Reference Material

Despite its infancy, quantum machine learning is composed of two well-established fields (quantum physics and machine learning), so there are ample resources for each. I am planning on using references that combine

the fields such as *Quantum machine learning* by Biamonte, et al. [6], *Quantum-enhanced machine learning* by Dunjko, et al. [8], and *An introduction to quantum machine learning* by Schuld, et al. [10], however there are many resources of each composite field as a backup. Each of the aforementioned papers proposes methods and applications within the quantum machine learning domain.

Should the need arise, I may also look at software documentation such as Google's TensorFlow [4] for machine learning and quantum software frameworks such as Microsoft's Q# [2] programming language documentation.

4 Experimentation and Analysis

To run the experiments and analyze my results, I plan on using Jupyter Notebooks (within Jupyter Lab) to develop a reproducible framework for the experiments. Jupyter Notebooks have support for both code and Markdown within the same environment, so they can be used to develop nice documents that can tell a story with code. In this way, the Qiskit source code can be bundled together with text information to explain relevant code pieces, quantum phenomena, and personal decisions made. For the experimental setup, I will construct models in Python source code using the Qiskit API, and operate these models alongside analogous classical variants on various training and test data. The training and test data will come from a split of the initial sample (for example, 80% training and 20% testing). Scikit-learn has methods for performing this kind of data splitting.

As machine learning is my primary focus, I plan on using model accuracy as a metric for analyzing the effectiveness of the quantum algorithms (benchmarked against similar, classical ones). Since the experiments will run on a classical computer (my PC) and not on physical quantum hardware, runtime and memory complexity will not be entirely accurate nor useful for assessing experimental quality. Beyond accuracy, other machine learning model-assessment metrics can be used such as area under curve (for receiver operating characteristic curves) and F1 score. An interesting thing to consider is error analysis; since the experiments will simulate quantum results, error will be an important part (caveat) of my analysis.

I have not used Qiskit before, but I have experience with machine learning techniques from previous classes (mentioned in the *Problem Overview* section above) and some self-study through course offerings at sites such as Coursera [1]. As far as concrete programming goes, I have only used MATLAB previously (in Introduction to Machine Learning). I enjoy Python very much and am grateful for this opportunity to sharpen my Python programming proficiency and learn more about a field that I find very intriguing.

5 Timeline

The project is due in about one month, on 18 December 2019. An estimated timeline breakdown for working on the project by week is as follows:

- **November 17-23:** Set up working environment (Qiskit modules, Jupyter Lab structure)
- **November 24-30:** Begin experiments on computer, develop Jupyter notebook(s), and write code
- **December 1-7:** Begin writing paper report, wrap up experiments
- **December 8-14:** Finish writing paper report and experiments
- **December 15-18:** Wrap up project, verify integrity of results and deliverables

References

- [1] Coursera: World-class learning for anyone, anywhere. <https://www.coursera.org/>.
- [2] Microsoft Quantum. <https://www.microsoft.com/en-us/quantum>.
- [3] scikit-learn: Machine Learning in Python. <https://scikit-learn.org/>.
- [4] Tensorflow: An end-to-end open source machine learning platform. <https://www.tensorflow.org/>.
- [5] IBM Q: Quantum Starts Here. <https://qiskit.org/>, 2019.
- [6] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
- [7] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [8] Vedran Dunjko, Jacob M Taylor, and Hans J Briegel. Quantum-enhanced machine learning. *Physical review letters*, 117(13):130501, 2016.
- [9] R. Edgar. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, Jan 2002.
- [10] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.