Quantum Machine Learning Project

This project brief represents 100% of the assessment for COMP47950. It is expected that students spend in the region of 60-80 hours of independent study, research, and development time on this project. The module is as a whole designed to support different aspects of the project, its implementation, and the concluding analysis.

Project Motivation and Introduction

Quantum Computing has shown a lot of promise for many domains as it exhibits properties like entanglement and superposition that do not exist in classical scenarios. One area that is expected to profit greatly from this is machine learning. Current and near term Noisy Intermediate-Scale Quantum (NISQ) era quantum computers are, however, still very error prone, and current approaches to quantum machine learning are still quite immature. The class of quantum machine learning models based on variational principles is currently one of the more developed approaches and is also somewhat resilient to the presence of noise [14, 5, 7]. These approaches are often referred to as quantum neural networks (QNNs). Yet this class of approaches still has some challenges that will add complexity to this project (these are discussed below).

In general, quantum machine learning (QML) aims to leverage the unique features of quantum computers to enable new algorithms and approaches for machine learning tasks [3, 19]. The potential of QML to enhance the performance of machine learning algorithms and solve certain problems faster and more efficiently than classical methods has attracted significant attention in recent years [8, 21, 1]. A key resource in the development of quantum programs (typically called circuits due to their visual representation as operations, or gates, being performed upon one or more qubits in a manner that mimics classical logic circuits) is IBM's Qiskit [2], a python library that provides the means to define, simulate and deploy quantum programs. Qiskit is not the only available software library for these purposes, other options include PennyLane, Tensorflow Quantum, Google's Circ, Microsoft's Q#, and several others.

The development and implementation of QML models is a challenge due to the limited availability of quantum hardware and the noise and errors that can occur in quantum systems [12, 20]. As a result, researchers typically begin with quantum simulators [17, 16] and train their models on classical architectures as was performed in [20]. Similarly, QML models (especially those belonging to the class of variational methods) often make use of a classical optimisation algorithm (or similar) to learn key parameters (often for Pauli rotations). There are a number of additional challenges that efforts in this project will need to navigate, and while the following list is not exhaustive, it will cover the majority of challenges:

1. Access to quantum computers is still limited, yet, many of the cloud providers provide free access (for a limited time). Access, however, is not the main challenge you will face in this project. Instead, the time for jobs to run, i.e. how long they are in the queue, can be excessive. Thus, it is important to plan a considerable amount of time for jobs to queue.

2. Classically training a QML algorithm (i.e., in simulation) is expensive, as the problem complexity scales exponentially in the number of qubits and (parameterized) operations added. However, the class of functions that can be represented with a relatively small (i.e. $n = 7 - 10$) number of qubits is sufficient for the types of problems we will see in this module on a relatively decent personal computer / laptop, in the region of a few hours training time.

3. The variational class of QML methods, while relatively simple in concept, has a few key challenges. The first is the *ansatz lottery*: this implies it may not be trivial to find the best ansatz (here general circuit composition or architecture) for your ML task.

4. The second is the quantum equivalent of the vanishing gradient problem in deep learning. This is referred to as the barren plateau problem (see: [13]); it occurs due to the unitary nature of quantum programs (matrices of arbitrary sum to 1) and the fact that adding additional operations (gates) exponentially scale these matrices, have a non-trivial relationship with performance and often no discernibly observable impact with respect to gradients. As a consequence, the gradient becomes flat (a plateau) with no path of "escape" for the learning algorithm.

In this project, your aim is to compare and contrast the use of quantum vs. classical approaches for **simple** machine learning tasks. A key focus of the project is the explicit comparison in the performance of classical machine learning algorithms with modern approaches to quantum machine learning. The goal is to objectively benchmark a quantum machine learning approach and understand how it compares to one or more classical algorithms in terms of **cost, performance, speed, and interpretability**. Thus, the focus of the project is to facilitate a discussion that focuses on a metric-based comparison between classical and quantum approaches to machine learning for one or more specific datasets.

IMPORTANT: The project doesn't strictly need QML models to perform "well". Your goal is to be in a position to comment on the quality of the QML solution(s) you have developed and how they compare to a classical machine learning solution to the same problem(s).

Task

In this project, you are expected to take a (simple) machine learning problem and compare and contrast 3 different ways of addressing it:

1. **Classically**: apply a traditional machine learning pipeline to establish a performance baseline (you may use one from a previous module);
2. **Simulated**: create and apply a **simulated** quantum machine learning pipeline to establish a best case scenario and compare this to the classical baseline; and
3. **Quantum Device**: take your simulated pipeline and run it on one or more (free-tier) cloud-based quantum offerings to establish a "realistic" quantum baseline and compare this to the classical and simulated solution performance.

Using these three different pipelines, compare and contrast the classical and quantum approaches for the machine learning problem you have selected. **NOTE** the classical pipeline should be quite straightforward, the simulated one will often be "well behaved" in terms of its output, **BUT** expect significant challenges in the use, interpretation and training when using a real quantum device!

Deliverables

1. **Midterm Presentation (20%):** A presentation of the classical approach results representing the baseline performance ($\approx$ 5 min), and the plan (ansatz / architecture / input encoding, cloud infrastructure, basic experiment design etc.) for the quantum part of the project ($\approx$ 5 min). **Due: week 7 – see brightspace for exact dates** See **Table 1** for details of grading criteria.
2. **Implementation Notebook & Report (70%):** Develop your project with a Jupyter notebook (or similar) and submit one notebook for the whole project (.ipynb). Use markdown cells to structure your report and discuss your findings. **Due: week 12 – see brightspace for exact dates**. See **Table 2** for details of grading criteria.
3. **Final Presentation (10%):** a 10 min presentation capturing: 1) the ML problem being addressed, 2) the classical approach and results, 3) the QML approach and results; and 4) the main points of comparison between the classical and quantum ML approaches. **Presented in week 12** See **Table 3** for details of grading criteria.

Plagiarism

The submission must be yours and yours alone. If you are unsure what is or is not plagiarism, the following is a none exhaustive list of example activities you cannot do:

- Copy the completed files of another student and submit them as your own

- Share copies, images or print outs of your code with another student (by e-mail, Social Media, Google Drive, WhatsApp etc.)
- Groups of students working on a single solution and then all submitting the same work or subset of the same work (regardless of changes made)
- Students collaborating at too detailed a level. For example, consulting each other after each line / block / segment of code and/or sharing the results.
- Using resources from the internet without appropriate citations.
- Use of AI tools (e.g. ChatGPT etc.) for significant parts of the report and/or implementation.

For more details see:

- `https://csintranet.ucd.ie/Plagiarism/`, and
- `https://www.ucd.ie/secca/studentconduct/`

Any submission suspected of plagiarism will be submitted to the School of Computer Science Plagiarism subcommittee for further investigation.

(Some) Starter Reading

- [3] gives a good impression of key ideas and the fundamental challenges in QML.
- [20] is an example of how to evaluate and compare QML models using simulation via Qiskit. It also discusses some general design principles and makes some remarks on potential datasets you could use.
- [1, 10, 12, 4, 9, 18] are all example QML papers some of which have repositories and code that could act as initial ideas for the project.
- [8, 15, 16, 6] are example papers that undertake comparisons of QML approaches that give an indication of possible metrics and forms of comparison.
- [11] is one of the Qiskit book's from IBM, note that this is also available online.

This is a very fast paced area, so papers are emerging all the time, and this list will age very fast.

References

[1] Abohashima, Z., Elhoseny, M., Houssein, E. H., and Mohamed, W. M. Classification with Quantum Machine Learning: A Survey.

[2] Aleksandrowicz, G., et al. Qiskit: An Open-source Framework for Quantum Computing.

[3] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. Quantum machine learning. *Nature* (2017).

[4] Cappelletti, W., Erbanni, R., and Keller, J. Polyadic Quantum Classifier. In *Proceedings - IEEE International Conference on Quantum Computing and Engineering, QCE 2020* (Oct 2020), Institute of Electrical and Electronics Engineers Inc., pp. 22–29.

[5] Cincio, L., Rudinger, K., Sarovar, M., and Coles, P. J. Machine Learning of Noise-Resilient Quantum Circuits. *PRX Quantum 2*, 1 (Feb 2021), 10324.

[6] Dhara, B., Agrawal, M., and Roy, S. D. Multi-class classification using quantum transfer learning. *Quantum Information Processing 23*, 2 (2024), 34.

[7] FONTANA, E., FITZPATRICK, N., RAMO, D. M. N., DUNCAN, R., AND RUNGGER, I. Evaluating the noise resilience of variational quantum algorithms. *Phys. Rev. A 104* (Aug 2021), 022403.

[8] HOUSSEIN, E. H., ABOHASHIMA, Z., ELHOSENY, M., AND MOHAMED, W. M. Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision. *Expert Systems with Applications* (2022).

[9] JURCEVIC, P., ET AL. Quantum support vector machines: A review. *arXiv preprint arXiv:2012.03037* (2020).

[10] KHAN, T. M., AND ROBLES-KELLY, A. Machine Learning: Quantum vs Classical. *IEEE Access* (2020).

[11] LOREDO, R. *Learn Quantum Computing with Python and IBM Quantum Experience: A hands-on introduction to quantum computing and writing your own quantum programs with Python*. Packt Publishing Ltd, 2020.

[12] LUBINSKI, T., JOHRI, S., VAROSY, P., COLEMAN, J., ZHAO, L., NECAISE, J., BALDWIN, C. H., MAYER, K., AND PROCTOR, T. Application-Oriented Performance Benchmarks for Quantum Computing, 2021.

[13] MCCLEAN, J. R., BOIXO, S., SMELYANSKIY, V. N., BABBUSH, R., AND NEVEN, H. Barren plateaus in quantum neural network training landscapes. *Nature Communications 9*, 1 (2018).

[14] MCCLEAN, J. R., ROMERO, J., BABBUSH, R., AND ASPURU-GUZIK, A. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics 18*, 2 (Feb 2016), 23023.

[15] MEZQUITA, Y., ALONSO, R. S., CASADO-VARA, R., PRIETO, J., AND CORCHADO, J. M. A Review of k-NN Algorithm Based on Classical and Quantum Machine Learning. In *Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference*, S. Rodríguez González, A. González-Briones, A. Gola, G. Katranas, M. Ricca, R. Loukanova, and J. Prieto, Eds. Springer International Publishing, 2021.

[16] SAVCHUK, A. O., AND SHAPOVALOVA, N. N. Classification problem solving using quantum machine learning mechanisms.

[17] SCHULD, M., AND KILLORAN, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* (2019).

[18] SCHULD, M., AND KILLORAN, N. Supervised learning with quantum enhanced feature spaces. *Quantum Machine Intelligence 1*, 1 (2019), 49–56.

[19] SCHULD, M., SINAYSKIY, I., AND PETRUCCIONE, F. An introduction to quantum machine learning. *Contemporary Physics* (2015).

[20] SELIG, P., MURPHY, N., R, A. S., REDMOND, D., AND CATON, S. A Case for Noisy Shallow Gate-based Circuits in Quantum Machine Learning. IEEE.

[21] WIEBE, N., KAPOOR, A., AND SVORE, K. M. Quantum Deep Learning, 2015.

Grading Criteria

Table 1: Midterm Presentation Rubric

<table>
<tr><th>Criteria</th><th>A+, A, A-</th><th>B+, B, B-</th><th>C+, C, C-</th><th>D+, D, D-</th><th>$\leq$ FM+</th></tr>
<tr><td>Classical ML Baseline: 10%</td><td colspan="2">An excellent (A) / very good (B) evaluation of several classical ML approaches that includes an in depth discussion of at least:
1. data prep (e.g. handling missing data, feature selection, feature engineering, etc.)
2. appropriate sampling strategy / strategies
3. expansive exploration of different performance metrics
4. discussion of the core ML methodology
5. accurate commentary on aspects of the data that will be challenging for a QML approach</td><td colspan="2">A good (C) / satisfactory (D) evaluation of at least 1 classical ML approach that explores several of the following:
1. a data preparation strategy
2. appropriate sampling strategy
3. different performance metrics
4. discussion of the core ML methodology</td><td>The ML methodology lacks detail in several aspects and in its present state does not form a meaningful baseline for comparison with QML approaches later in the project.</td></tr>
<tr><td>QML Plan: 10%</td><td colspan="2">A clear and reasonable action plan for the QML project that captures in sufficient detail:
1. data input transformation approach(es),
2. type(s) of QML models under consideration,
3. model parameter considerations (no. qubits, software libraries, computational considerations etc.),
4. general timeline for simulation and cloud deployment,
5. a discussion of key resources that will help support / facilitate the project.</td><td colspan="2">A plan (capturing the items on the left) but where one or more aspects of the plan are weak and/or require significantly more detail.</td><td>The QML plan is either missing or lacks sufficient detail to generally understand the objectives / progress / set up of the project.</td></tr>
</table>

Table 2: QML Implementation and Report Rubric

Criteria	A+, A, A-	B+, B, B-	C+, C, C-	D+, D, D-	≤ FM+
Input Transformation & Data Prep 10%	An comprehensive exploration of different transformation approaches for the QML approach(es) and competent data preparation.	An thorough exploration of transformation approaches for the QML approach(es) with considerations of different forms of data preparation.	A well-executed approach to input transformation that also considers forms of data preparation.	An appropriate approach to input transformation and data preparation.	The approach to input transformation and data preparation lacks detail, rigour and/or accuracy of approach.
QML Model Design & Optimisation 30%	Extensive and competent application and design of several QML approaches and optimisation techniques: generates a rich setting for comparative analysis.	Thorough and accurate application and design of several QML approaches and/or optimisation techniques: generates a rich setting for comparative analysis.	Well-executed application and design of at least one QML approach with an appropriate optimisation strategy: generates a meaningful setting for comparative analysis.	Appropriate application of QML and its optimisation: generates a sufficient setting for comparative analysis.	The QML methodology has some flaws that inhibits the generation of results and/or comparative analysis.
Experimental Design 15%	A comprehensive exploration of the parameter and design space (with very extensive consideration of other effects e.g. noise, multiple architectures / ansatze etc.) is performed, such that the design of QML the model(s) to be experimented with on the cloud is very well informed.	A thorough exploration of the parameter and design space (with extensive consideration of other effects e.g. noise, multiple architectures / ansatze etc.) is performed, such that the design of QML the model(s) to be experimented with on the cloud is very well informed.	A reasonable exploration of the parameter and design space (with some consideration of other effects e.g. noise, multiple architectures / ansatze etc.) is performed, such that the design of QML the model(s) to be experimented with on the cloud is very well informed.	Some exploration of the parameter and design space (with some consideration of other effects e.g. noise, multiple architectures / ansatze etc.; this may be a little shallow) is performed, such that the design of QML the model(s) to be experimented with on the cloud is very well informed.	The experimental design is weak, contrived or otherwise significantly flawed.

Continued on next page

Table 2: QML Implementation and Report Rubric – *Continued from previous page*

Criteria	**A+, A, A-**	**B+, B, B-**	**C+, C, C-**	**D+, D, D-**	**≤ FM+**
Evaluation Methodology and Discussion of Findings 15%	The project extends well beyond simply applying models to data, and comprehensively compares and contrasts different QML and ML approaches with empirically sound analytical methods. Results are competently discussed and situated in key literature.	The project extensively analyses generated performance data of the QML and ML approach with empirically sound analytical methods to inform the comparison. Results are competently discussed and contextualised with relevant literature.	The project provides a good analytical basis in discussing generated performance data of the QML and ML approach(es) with appropriate analytical methods. Results are accurately discussed.	The project provides a fair analytical comparison of the QML and ML approach(es) with reasonable result data. Results are accurately discussed.	There are some flaws in the evaluation methodology and/or it lacks depth.

Table 3: Final Presentation Rubric

Criteria	**A+, A, A-**	**B+, B, B-**	**C+, C, C-**	**D+, D, D-**	**≤ FM+**
Presentation: 5%	Excellent presentation of all key aspects of the project	Very good presentation of all most aspects of the project	Good presentation of the project and key findings	Acceptable presentation of the project and its findings	Poor presentation
Demo: 5%	Excellent demo of QML part(s) of the project that gives a comprehensive overview of how the project results were generated	Very good demo of QML part(s) of the project that gives a comprehensive overview of how the project results were generated	Good demo of key part(s) of the project with some insight into how results were generated	Acceptable demo of QML part(s) of the project	Poor / no demo