Maastricht Science Programme

Research Proposal for the Experimental Implementation of Quantum Machine Learning Algorithms

Mark Fingerhuth

September 12, 2016

In partial fulfillment of the requirements for the degree Bachelor of Science (BSc) at Maastricht University.

In collaboration with the Quantum Research Group at the University of KwaZulu-Natal Durban. Under supervision of Prof. Petruccione, Quantum Research Group at UKZN, and internal supervisor, Dr. Fabrice Birembaut, Maastricht Science Programme, Maastricht University.

Abstract

Mixtape tote bag quinoa, deep v ramps organic pabst. Cliche trust fund twee lo-fi, lumbersexual sustainable skateboard brunch keytar edison bulb. Try-hard blue bottle meggings fashion axe, gentrify freegan PBRB. Squid retro viral, shoreditch sriracha salvia kogi chia. Celiac tumblr thundercats, williamsburg literally etsy man braid franzen flannel chambray raw denim. Try-hard woke retro intelligentsia. Af actually synth coloring book hoodie tumeric, knausgaard paleo butcher.

Contents

1 Introduction

The ability to understand spoken language, to recognize faces and to distinguish different types of fruit comes naturally to humans, even though these processes of pattern recognition and classification are inherently complex. Machine learning (ML), a subtopic of artificial intelligence, is concerned with the development of algorithms that mimic these mechanisms, thereby enabling computers to find and recognise patterns in data and classify unknown inputs based on previous training with labelled inputs. Such algorithms paved the way for e.g. human speech recognition, recommendation engines as used by Amazon and prediction algorithms that can predict heart disease from real-time electrocardiograms (Acharya et al., 2015).

According to IBM (2016), every day approximately 2.5 quintillion (10^{18}) bytes of digital data are created. This growing number implies that every area dealing with data will eventually require advanced algorithms that can make sense of data content, retrieve patterns and reveal correlations. However, most ML algorithms involve the execution of computationally expensive operations and doing so on large data sets inevitably takes a lot of time (Bekkerman, Bilenko, & Langford, 2011). Hence, it becomes increasingly important to find efficient ways of dealing with big data and/or reduce the computational complexity of the algorithms.

A promising solution is the use of quantum computation which has been researched intensively in the last decades. Quantum computers (QCs) use quantum mechanical systems and their special properties to manipulate and process information in ways that are impossible to implement on classical computers. The quantum equivalent to a classical bit is called a quantum bit (or qubit) and additionally to being in either state they can be in a linear superposition of $|0\rangle$ and $|1\rangle$. This peculiar property gives rise to so called quantum parallelism, which enables the execution of certain operations on many quantum states at the same time. However, despite this obvious advantage the real difficulty in quantum computation lies in the retrieval of the computed solution since a measurement of a qubit collapses it into a single classical bit and thereby destroys information about its previous superposition. Several quantum algorithms have been proposed that provide exponential speed-ups when compared to their classical counterparts with Shor's prime factorization algorithm being the most famous (Shor, 1994). Hence, quantum computation bears the potential to vastly improve computational power, speed up the processing of big data and solve certain problems that are practically unsolvable on classical computers.

Considering these advantages, the combination of quantum computation and classical ML into the new field of quantum machine learning (QML) seems almost natural. However, since most ML algorithms rely on solving some system of linear equations a corresponding quantum algorithm is required for QML to become achievable. Harrow, Hassidim, and Lloyd (2009) were first to describe such an algorithm (referred to as HHL-algorithm) which since has become a subroutine in many QML algorithms. There are currently two main ideas on how to merge quantum computation with ML, namely a) running the classical algorithm on a classical computer and 'outsourcing' only the computationally intensive task to a QC or b) executing the quantum version of the entire algorithm on a QC. Current QML research mostly focusses on the latter by developing quantum algorithms that tab into the full potential of quantum parallelism.

1.1 Motivation

Classical ML is a very practical topic since it can be directly tested, verified and implemented on any commercial classical computer. So far, QML has been of almost entirely theoretical nature since the required computational resources are not in place yet. QML algorithms often require a relatively large number of error-corrected qubits and some sort of quantum data storage such as the proposed quantum random access memory (qRAM) (Giovannetti, Lloyd, & Maccone, 2008). However, to date the maximum number of superconducting qubits reportedly used for calculation is nine, the D-Wave II quantum annealing device delivers 1152 qubits but can only solve a narrow class of problems and a qRAM has not been developed yet (D-Wave, 2015; O'Malley et al., 2016). Furthermore, qubit error-correction is still a very active research field and most of the described preliminary QCs deal with non error-corrected qubits with short lifetimes and are, thus, impractical for large QML implementations.

Until now there has been only three experimental verifications of QML algorithms that provide proof-of-principle. Li, Liu, Xu, and Du (2015) successfully distinguished a handwritten six from a nine using a quantum support vector machine on a four-qubit nuclear magnetic resonance test bench. In addition, Cai et al. (2015) were first to experimentally demonstrate quantum machine learning on a photonic QC and showed that the distance between and the inner product of two vectors can indeed be computed quantum mechanically. Lastly, Ristè et al. (2015) solved a learning parity problem with five superconducting qubits and found that a quantum advantage can already be observed in non error-corrected systems.

Considering the large gap between the number of proposed QML algorithms and experimental realisations of scaled-down QML problems, it remains important to find QML problems which can already be implemented on currently available quantum technology. Thus, the purpose of this study is to provide proof-of-principle implementation of selected QML algorithms on small datasets. This is a step in the attempt to shift QML from a purely theoretical research area to a more applied field such as classical ML. Furthermore, this can also lead to verification or falsification of the claims and assumptions made in the field of QML.

1.2 Research Question

In light of the theoretical nature of current QML research and the small number of experimental realizations, this research will address the following question:

How can theoretically proposed quantum machine learning algorithms be implemented on state-of-the-art quantum technology?

The following sections will outline the steps required and the tools used in order to answer this research question.

1.3 Research Objectives

The main objective of this research is to demonstrate that QML algorithms can already be used for solving small problems on currently available quantum technology. Even though this might seem trivial at first, there are many problems that are often not addressed in the proposals of QML algorithms such as the encoding of data, the influence of quantum noise and the restrictions on the data type (e.g. uniformly distributed or sparse data sets only). All theses issues have to be addressed when implementing QML algorithms experimentally and thus consitute the major challenges during this study. Ideally, the outcome will already demonstrate observable quantum advantages over the classical algorithms, provide supporting evidence for the claim that QML can indeed be used to solve ML problems and that the increasing number of qubits will bring the expected speed-ups in computation.

2 Research Methods

The proposed research will be solely based on the two QML algorithms described in Schuld, Sinayskiy, and Petruccione (2014, 2016). Firstly, Schuld et al. (2014) is a quantum version of the distance weighted k-nearest neighbour (KNN) algorithm. For clarification, let us consider a training data set D_T consisting of five vectors $v_0, v_1, ...$ that are each either assigned to class A or B. Classical KNN is a non-parametric classifier that given an unclassified input vector x considers the k nearest neighbours (using a predefined measure of distance) and classifies x, based on a majority vote, as either A or B. Thereby, k is a positive integer and is usually chosen to be small. In the case of $k = all$, input vector x would simply be assigned to the class with the most members. In this case, the training vectors can be given distance-dependent weights (such as $1/distance$) in order to increase the influence of closer vectors over more distant ones. The advantage of the quantum version is the parallel computation of the distance between each training vector and the input vector as well as contracting distance computation and the weighing into one computational step.

Next, Schuld et al. (2016) details a quantum algorithm for a linear regression model for supervised pattern recognition.

The first step towards their experimental implementation will be the identification of one or several small ML problems that can be executed on maximally nine qubits. For example, this might include the characterization of colours or differentiation of digits or letters. It thereby plays an important role if the respective ML problem can be approached using a very small dataset such as the average pixel brightness or the ratio of pixels above and below the bisector of the image. Ideally, the data should be representable as a 2-D vector such that it requires only a few qubits to encode the information quantum mechanically.

There are two types of tools available for the implementation of the QML algorithms. Firstly, since current state-of-the-art quantum technology uses maximally nine qubits, a classical computer can still be used to simulate the behaviour of the QC. Such a software architecture is provided by Microsoft Research which has released the quantum simulation toolsuite Liqui$|\rangle$ based on the programming language F#. There are many more QC simulator toolsuites available and the decision which one to use depends on the selected QML problem and dataset (Quantiki, 2016).

Secondly, earlier this year technology company IBM has enabled public access to their experimental quantum processor containing five non error-corrected superconducting qubits. Instead of only simulating on classical hardware, this opens up the possibility of executing the QML algorithm on actual quantum hardware. If it is possible to make use of IBM's QC is highly dependent on the dataset chosen. Furthermore, until this point it remains unclear if the algorithm in Schuld et al. (2016) can be executed using five qubits only.

Both algorithms assume that the classical data is readily available in the form of quantum states. Hence, the first crucial step will be translating the classical data into such states, constituting the first challenge since it is a non-trivial and still researched topic. The quantum version of the distance weighted kNN requires a binary data string of length n to be encoded one to one into n qubits. This will be referred to as *qubit encoding*. A possible algorithm for this type of data encoding was proposed by Ventura and Martinez (1999). The quantum linear regression algorithm is based on so called *amplitude encoding*, where the classical data is written into the amplitudes of quantum states which is more difficult to achieve than qubit encoding. Amplitude encoding is still a very active field of research and until now it can only be done with relatively uniform data sets. Grover and Rudolph (2002) described such an algorithm and special attention will be paid in choosing a suitable uniform dataset when using this algorithm.

Next, two separate quantum circuits consisting of quantum logic gates will be designed that accurately represent the two QML algorithms as outlined in the paper of Schuld et al. (2014, 2016). Each quantum circuit is then combined with its respective data encoding circuit. Finally, the computed solution needs to be retrieved by measuring specific qubits. By repeating the execution of the algorithms, a probability distribution over the qubit measurements is obtained that represents the solution to the given problem.

To summarize, the steps needed for the successful implementation of a QML algorithm are given below.

1. Find a small implementable ML problem
2. Generate or find a suitable dataset
3. Encode the classical data into quantum states
4. Design a quantum circuit representing the QML algorithm
5. Execute the entire quantum circuit multiple times
6. Retrieve the solution from the resulting probability distribution

3 Timeline

The projected timeline for the proposed bachelor thesis research is given below.

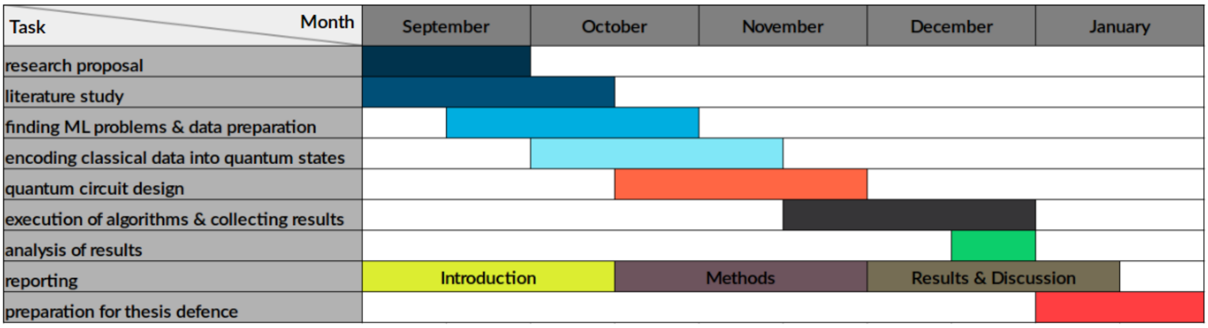

Figure 1: Projected timeline for proposed research

4 Research Impact

Successful proof-of-principle studies are crucial for further research to be funded and supported since it shows that an upscaling of quantum computational power will eventually lead to at best exponential speed ups compared to classical ML and hence has the potential to revolutionize the handling of big data.

5 Conclusion

axidermy edison bulb plaid, chia swag organic roof party shabby chic raw denim tilde waistcoat. Swag everyday carry iPhone, pitchfork pop-up ethical blog small batch la croix before they sold out chartreuse chia gastropub craft beer crucifix. Occupy mustache organic tumblr, scenester cred listicle kombucha lumbersexual. Crucifix tumeric bushwick, organic unicorn ugh food truck 90's echo park freegan mumblecore chia shabby chic. Keytar actually intelligentsia mumblecore, ugh selvage schlitz tousled iPhone cray paleo wayfarers snackwave viral humblebrag. Subway tile pop-up squid church-key craft beer. Church-key la croix cornhole kitsch 8-bit gluten-free.

6 References

Acharya, U. R., Fujita, H., Sudarshan, V. K., Sree, V. S., Eugene, L. W. J., Ghista, D. N., & San Tan, R. (2015). An integrated index for detection of sudden cardiac death using discrete wavelet transform and nonlinear features. *Knowledge-Based Systems*, *83*, 149–158.

Bekkerman, R., Bilenko, M., & Langford, J. (2011). *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press.

Cai, X. D., Wu, D., Su, Z. E., Chen, M. C., Wang, X. L., Li, L., ... Pan, J. W. (2015). Entanglement-based machine learning on a quantum computer. *Physical Review Letters*, *114*(11), 1–5. doi: 10.1103/PhysRevLett.114.110504

D-Wave. (2015). *Breaking through 1000 qubits.* `http://www.dwavesys.com/blog/2015/06/breaking-through-1000-qubits`. (Accessed: 2016-09-09)

Giovannetti, V., Lloyd, S., & Maccone, L. (2008). Quantum random access memory. *Physical review letters*, *100*(16), 160501.

Grover, L., & Rudolph, T. (2002). Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*.

Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters*, *103*(15), 1–4. doi: 10.1103/PhysRevLett.103.150502

IBM. (2016). *What is big data?* `https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html`. (Accessed: 2016-09-08)

Li, Z., Liu, X., Xu, N., & Du, J. (2015). Experimental realization of a quantum support vector machine. *Physical Review Letters*, *114*(14), 1–5. doi: 10.1103/PhysRevLett.114.140504

O'Malley, P. J. J., Babbush, R., Kivlichan, I. D., Romero, J., McClean, J. R., Barends, R., ... Martinis, J. M. (2016, Jul). Scalable quantum simulation of molecular energies. *Phys. Rev. X*, *6*, 031007. Retrieved from `http://link.aps.org/doi/10.1103/PhysRevX.6.031007` doi: 10.1103/PhysRevX.6.031007

Quantiki. (2016). *List of quantum computing simulators.* `https://quantiki.org/wiki/list-qc-simulators`. (Accessed: 2016-09-12)

Ristè, D., da Silva, M. P., Ryan, C. A., Cross, A. W., Smolin, J. A., Gambetta, J. M., ... Johnson, B. R. (2015). Demonstration of quantum advantage in machine learning. *arXiv:1512.06069*, 1–12. Retrieved from `http://arxiv.org/abs/1512.06069`

Schuld, M., Sinayskiy, I., & Petruccione, F. (2014). Quantum computing for pattern classification. , 14. Retrieved from `http://arxiv.org/abs/1412.3646` doi: 10.1007/978-3-319-13560-1

Schuld, M., Sinayskiy, I., & Petruccione, F. (2016). Pattern classification with linear regression on a quantum computer. , 5. Retrieved from `http://arxiv.org/abs/1601.07823`

Shor, P. W. (1994). Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer: Proc. , 124.

Ventura, D., & Martinez, T. (1999). Initializing the amplitude distribution of a quantum state. *Foundations of Physics Letters*, *12*(6), 547–559.