SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
BANGALORE-562157

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

QUANTUM MACHINE LEARNING

Report

Submitted by

ABHIJITH C.	**1MV14CS004**
RAGHAVA G. DHANYA	**1MV14CS077**
SHASHANK S.	**1MV14CS131**

Abstract

Recently, the increase in computational power and available data, as well as algorithmic advances in machine learning techniques, have led to impressive results in regression, classification, data generation, and reinforcement learning. Despite these successes, the proximity of the physical boundaries of chip manufacturing as well as the growing size of data sets is driving an increasing number of researchers to explore the possibility of harnessing the power of quantum computing to accelerate classical machine learning algorithms

Machine learning tasks often involve problems with manipulation and classification of a large number of vectors in large spaces. Conventional algorithms for solving such problems typically take a polynomial time in the number of vectors and the size of the space. Quantum computers are good for handling high-dimensional vectors in large tensor product spaces

The field of quantum machine learning explores how to design and implement quantum algorithms that makes machine learning faster. Recent work has produced quantum algorithms that could serve as building blocks for machine learning programs, but the hardware and software challenges are still considerable.

Contents

INTRODUCTION

Machine learning is very difficult. It's what mathematicians call an "NP-hard" problem. That's because building a good model is really a creative act. As an analogy, consider what it takes to design a house. You're balancing lots of constraints – budget, usage requirements, space limitations, etc. – but still trying to create the most beautiful house you can. A creative architect will find a great solution. Mathematically speaking, the architect is solving an optimization problem and creativity can be thought of as the ability to come up with a good solution given an objective and constraints. Classical computers aren't well suited to these types of creative problems. Machine learning algorithms are tasked with extracting meaningful information and making predictions about data. In contrast to other techniques, these algorithms construct and/or update their predictive model based on input data. The applications of the field are broad, ranging from spam filtering to image recognition, demonstrating a large market and wide societal impact.

In recent years, there have been a number of advances in the field of quantum information theory showing that particular quantum algorithms can offer a speedup over their classical counterparts. Execution time is just one concern of learning algorithms. Quantum optimizations are capable of finding the global minimum of a non convex objective function in a discrete search space. Storage capacity is also of interest. Quantum associative memories store exponentially more patterns than a classical Hopfield network. In addition to supplying exponential speed-ups in both number of vectors and their dimensions, quantum machine learning allows enhanced privacy: in a quantum clustering algorithm only $O(log(MN))$ calls to the quantum data-base are required to perform cluster assignment, while $O(MN)$ are required to uncover the actual data [5].

Deep Learning: The boom

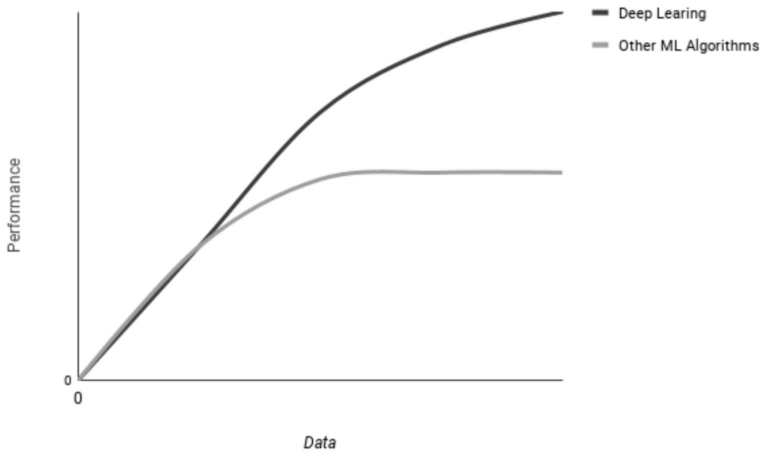

Figure 1.1: Deep learning vs Traditional Machine Learning

Now the main question, why is this field getting traction these days? The first Convolutional Neural Network was developed by Yann LeCun in 1995 [3], why did it take so long for deep learning to take off? One answer, the Scale. Scale drives deep learning progress, scale in both data-set sizes and computational power. But the same scale could act as bottleneck for deep learning.

The data volumes are exploding: more data has been created in the past two years than in the entire previous history of the human race. Data is growing faster than ever before and by the year 2020, about 1.7 megabytes of new information will be created every second for every human being on the planet. By then, our accumulated digital universe of data will grow from 4.4 zettabyets today to around 44 zettabytes, or 44 trillion gigabytes. The demand for storage has grown more than 50% annually in recent years, a rate faster than the rapidly decreasing unit cost of storage can handle. The growth of data is profound and shows no signs slowing. There is an inconvenient truth in technology: the amount of data keeps growing exponentially, while the increases in the power of computers are slowing down. Computers today can carry out those instructions in nanoseconds, but they still do it one step at a time, which has become a liability known as the von Neumann bottleneck. If individual chips can no longer be made faster, and the amount of work computers have to do continues to grow, the only way to attack ever larger data problems with von Neumann programmable computers will be to build more computers and ever bigger data centers. It seems as though we have hit a wall in growth of computational capacities. How can we meet this demand for data storage and computational power?

One of most promising solution is Quantum computers and Quantum information processing.

How can Quantum Computers Help?

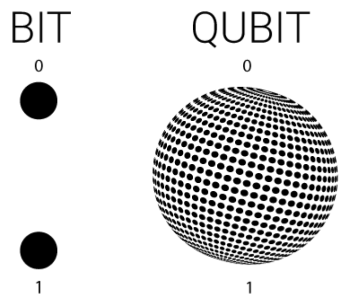

Figure 1.2: Bit vs Qubit

The basic idea is to use the atypical behavior of quantum physics such as quantum tunneling and quantum entanglement to get atoms to make calculations. A dozen atoms in a quantum computer would be more powerful than the world's biggest supercomputer. While regular computers symbolize data in bits, 1s and 0s expressed by flicking tiny switch-like transistors on or off, quantum computers use quantum bits, or qubits, that can essentially be both on and off, enabling them to carry out two or more calculations simultaneously. In principle, quantum computers could prove extraordinarily faster than normal computers for certain problems because they can run through every possible combination at once. In fact, a quantum computer with 300 qubits could run more calculations in an instant than there are atoms in the universe. Also qubits can store more data than classical bits, n qubits can carry about 2^n classical bits of information due to superposition.

Technical Description

Quantum computing can help machine learning in two ways- data and computation. Based on this, Quantum Machine Learning can be divided into 3 approaches.

1. Quantum enhanced Machine Learning (classical data with quantum algorithms)
2. Classical learning applied to quantum systems (classical algorithm with quantum data)
3. Fully quantum machine learning (both data and algorithm is quantum based)

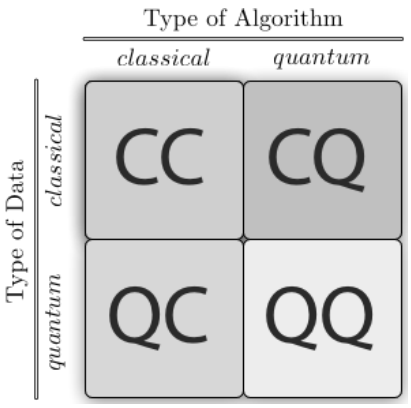

Figure 2.1: Approaches in Quantum Machine Learning

Quantum Enhanced Machine Learning

Quantum Enhanced Machine Learning is the most promising of these methods on early quantum computers. Such algorithms typically require one to encode the given classical dataset into a quantum computer so as to make it accessible for quantum information processing. Machine learning techniques use mathematical algorithms and tools to search for patterns in data. The most important and time taking parts of machine learning algorithms are searching, sampling and optimization. We explain how these can be improved by Quantum Computing in the following sections.

Search: Grover's algorithm

When you compute Grover's Algorithm[2] you are simultaneously testing every input value to see which is the correct input value. Using quantum superposition you can get qubits in a state that represents all possible inputs. Then run that superposition of states through some function to get each input together with its associated output. You are given a list of n elements, and you know that one element satisfies a certain condition, while the other n-1 elements don't. Basically, this is an algorithm for finding a specific element in an unordered list. A classical computer would not be able to exploit any structure in this problem and therefore needs to scan up to $n-1$ elements to find the one needed. Prepare n qubits in a uniform state so that all 2^n numbers are in a uniform superposition, each with a coefficient of $1/\sqrt{n}$ If we would measure

the n qubits now, all 2^n results would be equally likely.
Then run the Grover iteration k times, which consists of two steps (which can be merged into 1 step):

1. Negate the coefficient of the sought element. This is a unitary operation, which leaves all elements not satisfying the condition as they are and only negates the intended one.
2. Reflect all quantum states at the arithmetic mean of all quantum states. This also is a unitary operation.

Each iteration amplifies the coefficient of the correct solution while damping the coefficient of all $n-1$ incorrect solutions, however only to a certain point. If you choose the number of iterations k correctly, you have maximized the coefficient for the correct solution. This means that the sought element's probability is now (almost) 1, so if we measure the qubits, we are very likely to get the correct answer. If we don't get it, we can repeat the algorithm from the beginning until we get the right answer. Doing more or less than the optimal k iterations reduces the probability of finding the correct solution - however, the algorithm periodically reaches the maximum coefficient. It can be shown that $k = O(\sqrt{n})$. This is remarkable, because a classical computer needs $O(n)$ steps for solving this problem. This is only a quadratic speedup (rather than an exponential speedup as for other quantum algorithms), but the Grover algorithm has quite a large number of useful applications and is fairly simple. The Grover algorithm can be extended to support a predefined number of $m(0 \leq m \leq n)$ elements which satisfy the condition, instead of 1. m need not be known in advance. Quantum period finding can be used to determine m before starting the extended Grover algorithm.

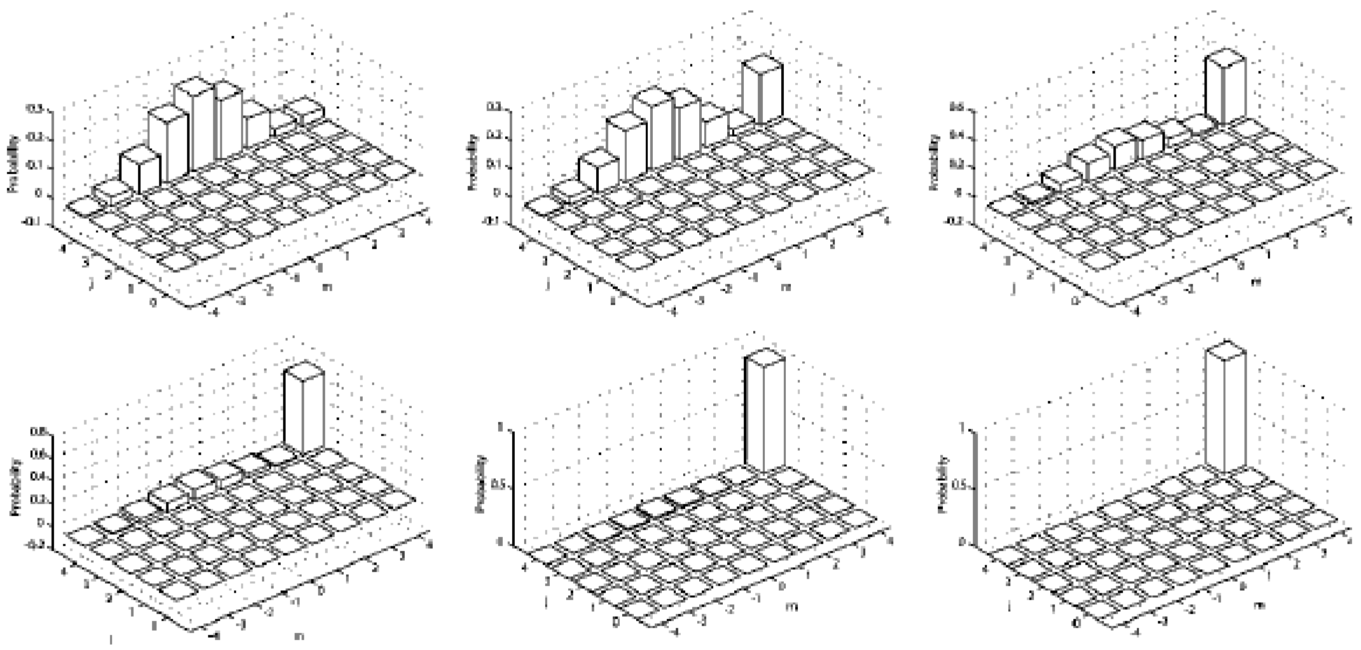

Figure 2.2: Grover's Search

Optimization: Quantum Annealing

Classical computing might use what's called "gradient descent": start at a random spot on the surface, look around for a lower spot to walk down to, and repeat until you can't walk downhill anymore. But all too often that gets you stuck in a "local minimum" – a valley that isn't the very lowest point on the surface. That's where quantum computing comes in. It lets you cheat a little, giving you some chance to "tunnel" through a ridge to see if there's a lower valley hidden beyond it. This gives you a much better shot at finding the true lowest point – the optimal solution. We'll see how that works using Quantum annealing.

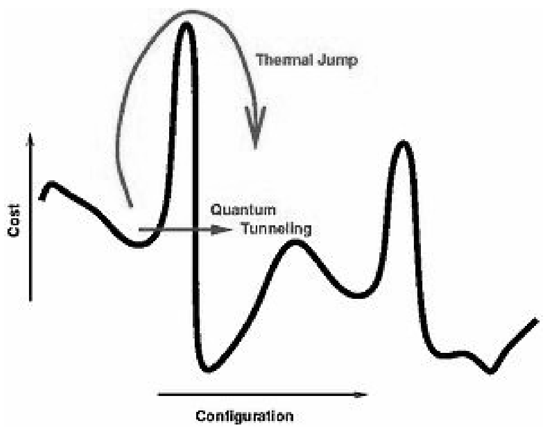

Figure 2.3: Quantum annealing

Quantum annealing[1] can be compared to simulated annealing, whose "temperature" parameter plays a similar role to Quantum Annealing's tunneling field strength. In simulated annealing, the temperature determines the probability of moving to a state of higher "energy" from a single current state. In quantum annealing, the strength of transverse field determines the quantum-mechanical probability to change the amplitudes of all states in parallel. The tunneling field is basically an energy term that does not commute with the classical potential energy part. Analytical and numerical evidence suggests that quantum annealing outperforms simulated annealing under certain conditions. Quantum annealing is also used in sampling from high-dimensional probability distributions.

Quantum algorithm for linear systems of equations

Solving a system of linear equations can be thought of as a Matrix inversion problem.The best known classical algorithm to solve matrix inversion takes $O(n^2 log(n))$(best proven lower bound) time. Using Quantum algorithms[4] we can perform matrix inversion in logarithmic time complexity of N. This is because using superposition we can find the eigenvectors and eigenvalues of the matrix by eigen decomposition of a matrix in a logarithmic time. This algorithm can be used to improve Least Square fitting , Principal Component Analysis and Support Vector Machines, which is a large margin optimized linear or non-linear classifier. Due to the prevalence of linear systems in virtually all areas of science and engineering, the quantum algorithm for linear systems of equations has the potential for widespread applicability.

Other than these basic algorithms that are used in almost all machine learning systems there are more complex ones which are applicable for specific systems such ANN, HMM etc. There are quantum algorithms which can provide exponential speedup for these systems.

Classical learning applied to quantum systems

This approach uses classical learning techniques to process large amounts of experimental quantum data in order to characterize an unknown quantum system.

The ability to experimentally control and prepare increasingly complex quantum systems brings with it a growing need to turn large and noisy data sets into meaningful information. This is a problem that has already been studied extensively in the classical setting, and consequently, many existing machine learning techniques can be naturally adapted to more efficiently address experimentally relevant problems. For example, Bayesian methods and concepts of algorithmic learning can be fruitfully applied to tackle quantum state classification, Hamiltonian learning and the characterization of an unknown unitary transformation. It has many applications such as

- Identifying an accurate model for the dynamics of a quantum system
- Extracting information on unknown states
- Learning unknown unitary transformations and measurements

As Classical learning on classical data helps us understand the data and the system that produced that data, Classical learning on Quantum data helps us understand Quantum data and the Quantum system that produced that data. This is relatively small and current research field.

Fully quantum machine learning

In the most general case of quantum machine learning, both the learning device and the system under study, as well as their interaction, are fully quantum. This goes beyond the early quantum computers and requires a much sophisticated Quantum Computer.

One class of problem that can benefit from the fully quantum approach is that of 'learning' unknown quantum states, processes or measurements, in the sense that one can subsequently reproduce them on another quantum system. For example, one may wish to learn a measurement that discriminates between two coherent states, given not a classical description of the states to be discriminated, but instead a set of example quantum systems prepared in these states. The naive approach would be to first extract a classical description of the states and then implement an ideal discriminating measurement based on this information. This would only require classical learning. However, one can show that a fully quantum approach is strictly superior in this case. (This also relates to work on quantum pattern matching.) The problem of learning unitary transformations can be approached in a similar way. A fully quantum clustering algorithm is also available - A Quantum Nearest-Centroid Algorithm for k-Means Clustering. k-means clustering is a popular machine learning algorithm that structures an unlabelled dataset into k classes. k-means clustering is an NP-hard problem. The complexity arguments on the dependence of m were rigorously confirmed using the QPCA construction for a support vector machine (SVM) algorithm. This can roughly be thought of as a k-means clustering problem with $k = 2$. A speedup is obtained due to the classical computation of the required inner products being $O(nm)$.

LIMITATIONS

Let's tackle the question of when to use quantum machine learning. It turns out that the quantum computers can be a real lifesaver for the big data people, you could query all of the data at a single time, not just sample them in batches. This is a very big improvement as most of the heavy data retrieval is done in batches and this adds to significant delay in what has to be achieved. It would also do great for those who are working with graph computing where you could factor in the complexity of many-to-many relationships that would otherwise require endless joins with relational data models. They will soon be known for breaking most encryptions as factorization will be very fast using algorithms like Shor's algorithm.

On the downside, quantum computers are not universal computers. They cannot replace classic computers. The whole idea of quantum computers is to exploit the quantum physics in performing certain complex computations very fast. For most of the mundane programs and tasks, the classic computer would actually perform better than quantum computers. But quantum computers would substantially improve the runtime in case of certain computations that would have been practically impossible with classic computers.

Let's extend this further and look into the limitation of QC. In physics, quantum noise refers to the uncertainty of a physical quantity that is due to its quantum origin. In certain situations, quantum noise appears as shot noise. To explain what shot noise is, consider a simple experiment of tossing a coin and counting the occurrences of heads and tails, the numbers of heads and tails after a great many throws will differ by only a tiny percentage, while after only a few throws outcomes with a significant excess of heads over tails or vice versa are common; if an experiment with a few throws is repeated over and over, the outcomes will fluctuate a lot.

These random fluctuations may be caused from heat in the qubits. Another reason might be that the quantum-mechanical processes will occasionally flip or randomize the state of a qubit, potentially derailing a calculation. This is a hazard in classical computing too, but it's not hard to deal with– you just keep two or more backup copies of each bit so that a randomly flipped bit stands out as the odd one out.

Researchers working on quantum computers have created strategies to deal with the noise. But these strategies impose a huge debt of computational overhead– all your computing power goes to correcting errors and not to running your algorithms. Current error rates significantly limit the lengths of computations that can be performed. The major issue in quantum error correction is that Superpositions can only be sustained as long as you don't measure the qubit's value. If you make a measurement, the superposition collapses to a definite value: 1 or 0. One ingenious scheme involves looking indirectly, by coupling the qubit to another "ancilla" qubit that doesn't take part in the calculation but that can be probed without collapsing the state of the main qubit itself. It's complicated to implement, though. To create one "logical qubit" around 10,000 of today's physical qubits is needed, which is impractical. Some researchers think that the problem of error correction will prove intractable and will prevent quantum computers from achieving the grand goals predicted for them.

The known quantum algorithms for machine learning problems suffer from a number of caveats that limit their practical applicability. Classical computers and quantum computers are very different on the lowest level. Classical computers use electrons while quantum computers use photons for the same purpose. Also, all the logical gates used in a classical computer are non-reversible (except the NOT gate), while in case of quantum computers all the gates are reversible. So, classical computers and quantum computers are very different, not just in architecture, but in the way they perform their operations. This is the reason why quantum computers are good in performing calculations but can't run even our 1st and 2nd generation of computer games.

Suppose you are in the middle of a maze and want to find the exit path. In case of classical computer you will try each and every path one by one until you find the exit. And think about this in the way of algorithms used, like Branch and Bound, Backtracking, etc. Things are different and more surprising with quantum computers. In our little puzzle a quantum computer will try all the paths simultaneously in the first time. And no matter what kind of maze is given it will find the path in the first iteration. And, trying all the paths is due to the fact that quantum computer is in superposition of exponentially many states, not the processor actually computing all the paths simultaneously. It is very important that you understand the difference between the two. Quantum Computers, instead of checking possibilities one by one, create a uniform superposition over all possibilities and repeatedly and destructively interfere states that are not solutions. All quantum algorithms [can] have at-least two parts. One part is often like the classical part and be run on classical computers without any difference in output. The other part is the quantum part and a correct result can only come from quantum computer. One of the most well known algorithm is Shor's algorithm for factoring. Shor's algorithms runs exponentially faster than the best known classical algorithm for factoring. Grover's algorithm searches an unstructured database (or an unordered list) with N entries, for a marked entry, using only $O(\sqrt{N})$ queries instead of the $O(N)$ queries required classically.

Quantum computers have the potential to do some kinds of calculation with unprecedented speed, as small-scale demonstrations have confirmed. However, to perform most of these calculations effectively these machines will eventually need to access something resembling random access memory (RAM) – a large store of quantum information that can be selectively accessed. Ordinary RAM contains a large array of memory cells, each holding one bit of information – a binary 0 or 1. To check the contents of particular cell, a computer accesses it using its address – a string of bits that identifies the cell's location. A quantum computer uses not bits but qubits, which can be a blend of 0 and 1 – a quantum superposition of the two states. Therefore, in quantum RAM, the address qubits would not identify a single memory cell but a certain superposition of all possible memory cells. For large data access, large QRAM might be required which might not be implementable.

Conclusion

Machine learning and quantum information processing are two large, technical fields that must both be well understood before their intersection is explored. To date, the majority of QML research has come from experts in either one of these two fields and, as such, oversights have been made. The most successful advances in QML have come from collaborations between CML and quantum information experts, and this approach is highly encouraging. These early results highlight the promise of QML, where not only are time and space scalings possible but also are more accurate classifiers.

There are several important issues to overcome before the practicality of QML becomes clear. The main problem is efficiently loading large sets of arbitrary vectors into a quantum computer. The solution may come from a physical realization of QRAM, however, as discussed previously, it is currently unclear whether this will be possible.

Despite the potential limitations, research into the way quantum systems can learn is an interesting and stimulating pursuit. Considering what it means for a quantum system to learn could lead to novel quantum algorithms with no classical analogue.

REFERENCES

[1] A.B. Finnila, M.A. Gomez, C. Sebenik, C. Stenson, and J.D. Doll. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics Letters*, 219(5):343 – 348, 1994.

[2] Vladimir E. Korepin and Lov K. Grover. Simple algorithm for partial quantum search. *Quantum Information Processing*, 5(1):5–10, Feb 2006.

[3] Yann LeCun and Yoshua Bengio. The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258. MIT Press, Cambridge, MA, USA, 1998.

[4] S. Lloyd. Quantum algorithm for solving linear systems of equations. In *APS March Meeting Abstracts*, page D4.002, March 2010.

[5] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *ArXiv e-prints*, July 2013.

Seminar Incharge **Head of the Department**