March 23, 2021

Task V: Open Task Part

Ameya Thete

Quantum computing, today is arguably one of the most upcoming and exciting fields at the intersection of quantum mechanics and classical computational theory. What perhaps seems like another buzzwordy idea thrown about by scientists and engineers, is in fact one of the most ingenious byproducts of the desire of physicists to control and study quantum systems. The 70s brought about a revolution in Atomic and Molecular physics — allowing us to finally control single quantum systems. Although it may seem distant and also unrelated, quantum computing fits rather neatly into this new scheme of things. First, it allows scientists to devise better and better methods to manipulate these single quantum systems, and secondly, better control over these quantum systems enables us to effectively harness quantum mechanics for the purposes of quantum information and computing.

It can be hard to see at first how computation has anything to do with quantum computing, after all, we have been getting better at building and theorising efficient models of computation, so surely there is no need to look elsewhere right? While this is generally true, two events provide a justification for introducing quantum mechanics into computational theory: Moore's law and the ramifications of the Solovay-Strassen test. Moore's law is pretty well known — the computational power available to us roughly doubles every two years for a fixed cost, but there's a problem: we're soon going to reach a plateau in the rapid strides we're making because of quantum effects that are no longer negligible at the length scales of newer and newer hardware devices. The Solovay-Strassen test showed at a computer with a random number generator could efficiently perform tasks with no efficient solution on a deterministic Turing machine, which lead to a modification of the Church-Turing hypothesis. Later on, David Deutsch sought to explore if the Church-Turing hypothesis could be made even stronger using quantum mechanics, which was inherently probabilistic in nature, and if a computational device based on quantum mechanics could be developed. These quantum-analogues of Turing machines lead to the emergence of quantum computing as we know it today.

So what exactly is quantum computing then? To put it succintly, it is another paradigm of computing that seeks to exploit purely quantum phenomena, such as that of entaglement and superpositions to perform, and furtheron ameliorate computational tasks. While quantum computing has shown to be rather promising, even though our current technological limitations do not allow for very complex computational tasks, there still is a lot for us to understand about how to develop new algorithms, because developing new algorithms, especially quantum ones, that demonstrate the so-called "quantum advantage", is hard, because our thinking is still rooted in the classical world.

The advancements show by quantum computing has prompted machine learning scientists to consider of quantum computing can be leveraged to improve on existing statistical learning algorithms — and there's a good argument for why this might be true. Since machine learning involves extensive linear algebra, and quantum mechanics is literally descibed using linear algebra, it might be worthwhile to explore if there really can be a quantum advantage in machine learning. One quantum algorithm I really like is the Harrison-Hassidim-Lloyd algorithm for solving linear systems. This was the first algorithm to demonstrate quantum speedups in linear algebraic tasks and lead to the emergence of quantum machine learning. Given a linear system $A\mathbf{x} = \mathbf{b}$, where A is Hermitian, the algorithm can solve for the vector $\mathbf{x}$ (There's also this caveat that the algorithm provides the expectation value of an operator in the state $|\mathbf{x}\rangle$, but that is a pedantic detail). The algorithm provides an exponential speedup over the fastest classical algorithm, and considering how linear systems form a part of every discipline, the algorithm has widespread applicability.

Regarding quantum softwares, I am familiar with Qiskit and Cirq, and I have been learning TensorFlow Quantum on the side, so I'm comfortable with using these tools. As a physics and computer science major, I would really like to focus on the application of quantum ML algorithms to high energy physics data and explore if quantum computing can be an alternative paradigm to solve problems in physics and HEP.