Schedule

April 30, 2019

1 Software Engineering for Noisy Intermediate Scale Quantum Computers

2 13 Classes and whats in them

Suggested textbook: *Quantum Computing: A Gentle Introduction* by Eleanor Rieffel and Wolfgang Polak

2.1 Class 1

2.1.1 Classical Bits and Clocks.

Class is designed to build intuition about Quantum Computers by showing how regular computations can be performed with clocks.

2.2 Class 2

2.2.1 Linear Algebra for Quantum Computers

Quantum Computers require a subset of linear algebra. Familiarity will be created in the math and notation needed for the class. Of most importance is the Bra-Ket notation.

2.3 Class 3

2.3.1 Classical Bits to Quantum Bits

We use the math learned in the last class to reimagine compute with classical bits. We then show that classical bits are simply a more constrained instance of quantum bits. With our new quantum bits we will introduce the Not, Control-Not, Toffoli and Hadamaard gates.

2.4 Class 4

2.4.1 Introduction to IBM simulator and Deutsche's Algorithm

We break open JupyterHub and show how to simulate quantum computers and gates on IBM's simulator. We will explain Deutsche's algorithm and then simulate it using IBM's Qiskit.

Take this time to clarify Quantum Entanglement and Parallelization

2.5 Class 5

2.5.1 Classical Computation $\subset$ Quantum Computation

Show how classical operations can be computed with quantum computers even if there is not necessarily a quantum advantage.

2.6 Class 6

2.6.1 Grovers and beyond!

What is grovers? What does it do? Why is it better than any classical algorithm? Is it provable? (yes) How can we use it on ONISQ Devices?

2.7 Class 7

2.7.1 Shors algorithm, Quantum Fourier Transform and its impracticalities

Actually, this is pretty self explanatory

2.8 Class 8

2.8.1 Quantum Error Correction

What is it, why do we need it and how can we implement it?

2.9 Class 9

2.9.1 VQE and QAOA - Embrace the Noise

Introduction to NISQ algorithms that don't need error correction. How do they work? What are their limitations?

We may jump into HHL if there is time, doubtful.

2.10 Class 10

2.10.1 Goeman-Williamsons take on Maxcut

Semi-definite programming is extremely reminiscent of how our Qbits work. Lets attempt to take advantage of that for a speed up on the MaxCut problem

2.11 Class 11

2.11.1 Quantum Boltzmann Machines

Quantum Computers promise a speedup along many pieces of a Machine Learning pipeline. Lets go over them and then dive into Quantum Boltzmann Machines.

2.12 Class 12

2.12.1 Quantum Chemistry

The reason Richard Feynman proposed a quantum computer in 1981 was to study quantum chemistry. So lets do it! What can we model with our NISQ computers?

2.13 Class 13

2.13.1 Quantum Computer Resistant Cryptography

Quantum Computers promise to break RSA (in the far future). What can we do to prevent that? There are Classical and Quantum encryptions that Quantum Computers would struggle to break.

2.14 Class 14

2.14.1 Portfolio Optimization

Quantum Computers could potentially change how our Financial Tech approaches problems.