Individual Capstone Assessment

Throughout a college education, students' abilities are measured frequently, often multiple times per week, through assessments such as homework, tests, quizzes, etc. However, these earlier mentioned types of assessments differ greatly from the types of assessments students will face upon finishing college. For instance, a student might finish a homework assignment in a day and forget about it. But at work projects rarely take days to complete and we usually can't forget about assignments once we complete them. At work, projects can take years or decades. To offer students an assignment that looks more like something students see outside of work UC requires a senior project. The senior project is a yearlong assessment that students complete as the culmination of classwork, which provides students the opportunity to demonstrate the ability to work on a longer project more similar to what they will see outside of college.

The senior project is open-ended and consequently, it will be heavily shaped by the interests and earlier class work of students. I think that I learned valuable skills in my classes and outside my classes through self-study, which will help me in my senior project. The area that I have been most interested in for the past few years has been artificial intelligence and specifically deep reinforcement learning. I was first introduced to the area of artificial intelligence through an edX course, which I took after my freshman year titled "6.86x: Machine Learning with Python-From Linear Models to Deep Learning". After taking this edX course I went through online coursework from Berkeley and Stanford during my sophomore and pre-junior year. Namely, I went through Berkeley's material for CS188 (artificial intelligence), CS189 (machine learning), CS182 (deep learning), and EECS127 (convex optimization); and Stanford's material for CS224N (natural language processing). During my junior year, I started taking some of the artificial intelligence classes at UC. Namely, I took CS4033 (artificial intelligence), CS5073 (deep learning), and special topics. I had already learned the material in CS4033 and CS5073 through my self-studying of Berkeley's material, but the special topics course was mostly new to me. I took that course with Prof. Atluri and went through the material from Berkeley's CS285 (deep reinforcement learning), which is available online. While I had some experience will deep reinforcement learning prior to taking this course from my co-ops, this was my first experience studying the material formally. I thought that reinforcement learning was, out of all the different learning approaches, the most powerful and the most similar to how learning happens in the real world.

I co-opted at two different companies: Siemens PLM Software and Northrop Grumman. My first two co-ops were at Siemens, where I worked as a software engineer. At Siemens, I learned how to work on a large project with a team of people through my work on Siemens' NX CAD program. I enjoyed my time at Siemens, but after two rotations I decided that I wanted to work on something with a substantial machine learning component. So, I switched to working as a machine learning engineer at Northrop Grumman, where I completed my last three co-ops. At Northrop, worked with a small team on projects focused on using machine learning to support national defense. I major project that I worked on involved using reinforcement learning to autonomously control military assets, i.e., planes, tanks, ships, submarines, etc. I hope that the skills I picked up at Northrop and through my studies will serve me well on my senior project.

For my senior project, I will be focusing on designing and testing a new offline deep reinforcement learning (DRL). I find this interesting first because I find DRL interesting. I would like to study DRL in graduate school and eventually make a career working in DRL. So, any project with a DRL aspect to it is interesting to me. But this project is particularly interesting to me because it has a number of applications to using DRL in the real world. One of the problems with running DRL in the real world is that we have to train our agent by interacting with the world. As our agent start out performing poorly, this can be dangerous and expensive. One possible solution is to train the agent offline on a collection of trajectories from some prior behavioral policy and the run/fine-tune the agent in the real world.

Preliminarily, I plan to base my offline RL algorithm off deep q-learning. The basic idea will be the constrain the actions of my learned policy to be similar to the behavioral policy the collected the offline dataset. This would probably require running behavioral cloning to get a copy of the behavioral policy. Then I can just ignore action for which the cloned policy is unlikely to sample. To me, success in this project would mean training an agent that performs much better than random. If I could beat some of other offline RL approaches of my tests, I would consider that a great success.