**Project Design**

**Topic Selection**

For this project, I chose to pursue work in topic I had previously presented an investigation on, as I found the work quite interesting. I had researched deep reinforcement learning and various different algorithms used for it: Q-value tables, DQN, DDQN, A3C, and so forth. I decided I wanted to find a project where I could learn more about this area of work, and tried to find tutorials on Deep RL. Most resources seemed to recommend starting out with a DQN agent, and the majority of tutorials were on atari’s breakout, so that’s where I began my work.

As I learned what things to search for, however, I found better and better tutorials in the field, and found that it might be easier to start on a game that wouldn’t require a convolutional neural network, and so I switched over to working on atari’s Lunar Lander. I think this provided a better introduction to the field, as I had more resources supporting the work I was doing, and it was a slightly lower bar to set for myself (which I needed as this was my first time doing a project in the field). As such, my topic was doing deep reinforcement learning in atari’s lunar lander.

To start, I followed a few resources in TensorFlow, and built my project out from scratch. I tried a behemoth-approach, as I will call it, wherein I tried to get the whole DQN working without doing small iterations. This proved extremely difficult: all parts of the code rely on other internal connections, and when one thing wasn’t written correctly the whole thing failed. Unfortunately, more iterable approaches are hard when working with deep reinforcement learning. There are a few things I would suggest, however, that can be built independently.

DQN requires: 1) a memory bank where you save experiences, 2) a way for your agent to decide how to act, 3) a way for your agent to record in memory what it did along with information about the environment, 4) the creation of the neural networks, and 5) a way for it to grab a sample of experiences and update the neural network. The first step, a memory bank, can be built independently from the rest. For the rest of the steps, some of the functionality can be broken down to be independent modules: for step 3, for example, practice grabbing information from the environment, then practice writing things into your memory bank generally, then combine the two.

When I finished with TensorFlow, I had working code in all parts except for 5: updating my neural networks. I found TensorFlow to be quite finicky, and it was extremely hard to work with. I kept receiving error messages regarding my optimizer not being able to find connected variables, and after much debugging still couldn’t get it to work. My instructor suggested I moved over to Keras, and within a day I had code that would at least compile with a working agent.

Unfortunately, my Keras agent wasn’t getting better as it played – it in fact would get worse, and converge to a score around -600. As the deadline approached, I decided I would have to be more comfortable relying on adapting others’ code, and moved over to PyTorch. Most of the examples for the Lunar Lander were in Pytorch, so writing similar code was quite easy.

Ultimately, I’d highly suggest that someone new to the field write the agent in either Keras or Pytorch, but I’d recommend doing what I did and trying to write it from scratch. I think this exercise could be quite trivial if you simply copy and paste the code and adjust parameters, and I don’t think you’ll have as good of an understanding of what each part does.

**Tools**

Open AI’s Gym

Monitor

Numpy

TensorFlow

Keras

PyTorch

**Algorithms**

DQN, Bellman Equation

DDQN

**What I’d do differently next time**

I think I did a relatively good job structuring myself given that I was entering a whole new workspace: a new field (deep RL), new tools (tensorflow, keras, pytorch), and so forth. I think I might have tried to do a more comprehensive lit review at first, rather than attaching myself to the few tutorials I found initially, but it was hard as I didn’t know the space/search terms.