Journal Report 6
10/7/19-10/14/19
Pawan Jayakumar
Computer Systems Research Lab
Period 5, White

Daily Log

Monday October 7

Getting the Atari emulator was harder than expected. I ran into an error that I was missing some dll. I tried searching on the internet, but it seems that gym's atari environments aren't well supported on windows.

Wednesday October 9

I talked to you about a short term goal for my project that I can try to finish in the next few weeks so that I can apply for Regeneron. In the end I decided that I wanted to research more on internal reward functions that encourage the agent to learn skills that aren't necessarily usefull for solving the current task, but are usefull for later tasks.

Friday October 11

I reviewed the papers on internal curiosity (1) and empowerment (2). The former encourages the agent to be able to predict the consequences of its actions while the latter encourages the agent to gain more agency over the environment by discovering more options.

Timeline

Date	Goal	Met
Today minus 2 week	Implement my own CNN that achieves atleast 99 percent on MNIST	Yes it reached 99.4 percent
Today minus 1 week	Finish implementing the DQN for cartpole	yes and was also able to implement DDQN
Today	Adapt the DDQN to play atari breakout	No. I changed my goal midweek
Today plus 1 week	Iron out the details of my new internal reward function	
Today plus 2 weeks	Set up an environment and begin testing my new function	

Reflection

My original project idea was creating a hierarchical reinforcement learning (HRL) algorithm to be able to solve multiple tasks. There countless papers out there about different implementations of HRL and their various strengths and weaknesses, but their usage is still for solving one task very well. I wanted to tackle the issues that arise in trying to learn to solve multiple tasks concurrently. I am confident that HRL can provide the right framework to store skills in the most efficient manner (a dense tree-like structure). It is just a matter of how the agent builds up this tree of skills.

I am going to make the assumption that there is a single enviornment and there are n different reward functions to maximize. But really, this just means the agent has to learn to do n things. I think a pratical one is a robot maid. The home is a single environment, but the maid has to do many things like clean dishes, fold clothes, sweep the floor, etc. I think it is easier to learn all of these things at the same time because you can exploit skills that are usefull for many tasks. Walking around without crashing into things is something that is applicable to all of the tasks mentioned.

Rather than trying to directly try and solve all n tasks, the agent should first learn fundamentals that apply to many of the tasks. The agent can then build off these fundamentals to solve the task. In my scenario above, the agent would first learn things like how to pick up various objects, how to move around. Then it can learn more specific things. I think the main take away is that the order in which you learn things is very important. It is much more efficient to learn beginner skills and work your way up rather than stumbling upon the best way to do something, which is what current RL does. This is what will make or break my project. I have to figure out a system to make the agent prioritize skills that are more universal and low level.