Journal Report 16

1/29/20-2/7/20
Pawan Jayakumar
Computer Systems Research Lab
Period 5, White

Daily Log

Wednesday January 29

I got DIAYN to work. I had to examine every line with pytorch in it to make sure it was moving the array to the gpu. I tried it on MountainCarContinuous. The training was really slow though, so I have to investigate why that is.

Friday January 31

Turns out MountainCarContinuous doesn't put a cap on its episode length. This means the agent tried random moves until it beat mountain car by chance, which took like 1500 time steps. I checked online for what the "official" time step limit was and it was 200. This one change reduced the training time from a day to like 5 minutes. Of course, the number of episodes was so low it had no chance of succeeding, but I'm not really trying to use it on this environment, it was just a simple continuous environment.

Monday February 3

I re-read the DIAYN paper to see what changes I would have to make to make DIAYN work on discrete environments. Turns out it only uses SAC because it is based on entropy maximization, not because it is continuous. Since SAC has a discrete version, DIAYN should work just fine. I also planned out how to implement my triathlon environment within P. Christodolu's framework. Since I change the environment every n time steps, I need to modify the chain of methods that lead to run_n_episodes_for_agent() to handle a list of environments rather than a single one. After that it is easy because I just have to set up a simple counter and switch indexes at the appropriate time.

Wednesday February 5

I worked on my quarter 3 presentation and a good script.

Friday February 7

I listened to our class's presentations.

Timeline

Date	Goal	Met
Today minus 2 weeks	Get DIAYN running	No, I created the triathlon environment instead. Also found something that will help me with this goal next week
Today minus 1 week	Get DIAYN Running	Not yet
Today	Get DIAYN Running	yes finally
Today plus 1 week	Benchmark DIAYN, discrete SAC, and DDQN on triathlon	
Today plus 2 weeks	Make DBH (my algorithms name)	

Reflection

Having re-read the DIAYN paper actually gave me a moral boost because I realized I had a misconception about what they were doing with entropy. They weren't maximizing the entropy of the skill being selected because the skill selected already has a fixed probability of 1/(num_skills). They were maximizing the entropy of the skill itself because if it had to be as random as possible and continue to be easy do distinguish based on the states it visits, the skills have to diverge from each other in terms of the state space it covers. This is really good because given a sufficiently large number n, DIAYN is able to partition the state space for option creation. I'm still not convinced the optimal skill is random so I would bet that if you were to optimize each skill based on its ability to reach a goal, my idea could be really good. The number of skills in each layer is fixed, but later research can be done on pruning/ adding connections between layers, because that's how the brain works. Anyways, this just means my algorithm makes more sense because what it's doing is adding layers of abstraction which potentially make it easier to solve complex tasks.