

Daily Log

Monday January 14

I browsed through the multitude of openAI environments and found 3 Atari Ram environments (Bowling, space invaders, and Pong) which all have the same observation (128 dim vector) and same action space (6 discrete actions).

Wednesday January 8

I modified my old DDQN code to now take inputs (train/test) and env name. In addition to cart pole, there is now a triathlon environment that I made where the agent has to play Bowling, then Pong, then Space invaders. Agents race to see who can learn all 3 the quickest (win pong 21-0, consistently get strikes, and clear all space invaders.) The agents participating are DDQN, Discrete SAC, 100* skills discovered by Diayn combined with a meta controller, and my 2 stage Diayn (90* skills discovered by Diayn and 9* super-skills discovered by Diayn).

*subject to change

Friday January 10

I finished the training portion of the triathlon and also included a method to track progress over time. I discovered an implementation for discrete soft actor critic. The author also happened to have a Diayn implementation. Unfortunately the instructions for using it weren't updated or something so I have to investigate the code more deeply

Timeline

Date	Goal	Met
Today minus 2 weeks	No, I downloaded the papers implementation of DIAYN, but haven't tested it because I was working on HAC and soft actor-critic	
Today minus 1 week	Make the project more realistic to complete	Yes, I reduced the number of things that I have to do.
Today	Get DIAYN	No, I created the triathlon environment instead. Also found something that will help me with this goal next week
Today plus 1 week	Get DIAYN Running	
Today plus 2 weeks		

Reflection

The 3 small hurdles from last week are half way done 1) Get DIAYN to work –¿ still in progress 2) Put each algorithm in a docker file –¿ no longer needed 3) Set each of the algorithms up for completing Atari RAM environments –¿ The environments are set up

Understanding a whole repository takes quite a while because each choice the repo owner made is something you have to understand, even if its not ideal. It can be overwhelming intitally, but slowly and surely I learned to piece together how each file interacts with others. and the structure of the code. There's one part for the agents, one for training, one for just user parameters, one for logging and testing. Even then, if theres a runtime error like cuda device expected but cpu instance found, and you know cuda is available because `torch.cuda.isavailable()` returned True, it gets difficult to find whats going wrong. You start doubting your package versions and don't know what reliably works anymore, because its not your code.