Journal Report 8

10/28/19-11/11/19
Pawan Jayakumar
Computer Systems Research Lab
Period 5, White

Daily Log

Monday October 28

I worked on my quarter 1 presentation.

Wednesday October 30

Research: I thought I found a way to decide when to make an option in the same layer, but my initial solution ended up also solving the problem of trying to decide when to increase the depth of the hierarchy because making a "child" option for the most popular option is either increasing the depth of the hierarchy or the number of options at a level. I sent this idea to a Phd student who was the author of the paper on hierarchical actor critic. He seemed interested enough to reply and ask about the case where I may have to expand upward rather than increasing the depth (the current "root" is shouldn't be the root). It was something I didn't think of. I now make a slight change: whenever I decide to add an option, initialize an option with fresh weights and feed it into the parent option. buckets. Other: I presented and made note to have a script and a slide with my proposal on it (which seems to change slightly every week).

Friday November 1

The biggest problem in HRL is partitioning the state-space into sub-regions and dividing those into smaller sub-regions and create an option for each of these regions. I initially thought this was something that had been done before, but there aren't many papers on this topic. Either the hierarchies are 2 level, or have a predefined number per level, which goes against the spirit of my proposal. My fist question to help address this issue is what really is the point of having a hierarchy. Options were developed because theoretical and empirical evidence support the idea that temporally extended actions are better than controlling one move at a time. This resulted in the meta-controller networks which choose the sub-policy to execute for the next k time-steps. Hierarchies emerge when the meta-controller is itself a sub-policy for another, higher up meta-controller. The meta controller at the top of this 3 level hierarchy chooses a sub-policy to execute every k*n time-steps. Each layer in the hierarchy is responsible for an increasingly longer period. This means I either work my way up from the ground, or I search for options and try to force them into time-based buckets.

Wednesday November 6

Another aspect I have to keep in mind is that I'm working in a partially observable Markov Decision Problem (POMDP). Unlike in environments like atari where I have knowledge of the entire state (the game is simple enough that all information is displayed on the screen), in minecraft, I only receive observations from the camera attached to the player model meaning I can't see everything at all times. This throws a wrench in my plans because if I can't know what state I am in, how am I supposed to know what to do? I did some research on solving POMDP's. The gist is you have to work with what you believe to be the state you are in. Each observation changes the belief distribution of what the agent's current state is. I'm not too worried anymore because this has been solved many times. Life is a POMDP; we should learn to embrace this problem. I feel like good models would be beneficial, but research tends to shy away from model based methods because they believe the environment is too complex to model. I don't think there is anything wrong with simple models that show useful patterns in the state-space to the agent, especially because it doesn't know where it is.

Friday November 8

I said earlier one of the options was to discover a list of options and then try to organize them. Sergey Levine wrote a paper on discovering options that seemed to have really good results. They use the same old meta-controller idea, but I think the way the discovered the options is interesting. The key to discovering useful options is diversity. The less state-space two options share, the better it will be. The paper's main contribution is maximizing this diversity by maximizing the entropy of the options and the mutual information between the option and the states the option visits. Highest entropy is achieved when every outcome is equally likely. This means each option is equally useful. If there are two random variables that are dependent, mutual information is the information gained about the outcome of one event if the outcome of the other event is known. Maximizing the mutual information between states the option visits and the option itself lends itself to options that are heavily associated with a particular region in state-space. Since I'm working in a POMDP, this isn't as effective.

Timeline

Date	Goal	Met
Today minus 2 week	Iron out the details of my new internal reward function	No, I changed my goal again
Today minus 1 week	Figure out when to make new options	I have an interesting idea,but I'd have to test it, so I guess met for now.
Today	Figure out when to make a new layer in the hierarchy	Yea because I revised my method of expansion so it does both now.
Today plus 1 week	Research more on state abstraction	
Today plus 2 weeks	Find a method for determining when to expand	

Reflection

I figured out how to expand–that wasn't hard–the hard part is when. I became too invested into this line of research so I once again changed my goals for the next 2 weeks. I still have faith I am heading in the right direction. I may not have been able to do Regeneron, but that won't bother me too much. I put a lot of my thoughts into the daily logs because my work is literally thinking, so there's not much else to say.