sliu1@uvm.edu
Sida Liu

Revised Project Proposal

Structure

The deliverable will be a piece of open-source software.

Topic

Motivation

In 2017, Yoshua Bengio submitted an arXiv article named *The Consciousness Prior* [1]. He proposed the *Consciousness Prior Theory*, which is about how, in his opinion, the *Consciousness* works and how to implement the *Global Workspace Theory* using recently developed *Deep Learning* technology. There are two major parts in his theory, the *Consciousness* c and the *Unconsciousness* h.

I will focus on the h in this course project because it is more practical and is the foundation before we get anything similar to the *Consciousness*.

Big Picture of Deep Reinforcement Learning

Our objective is to get a good h, but how can we get it.

Deep Learning is trying to do hierarchical information abstraction since its invention. CNN, RNN, Attention model, and many other Deep Learning models are very good tools to do this job.

In the setting of Reinforcement Learning, we can simply situate the agent in an environment, and use a high-level reward function to specify the goal.

At each time step, we can pass observation through the neural network model of the agent and get a value of that state (or action); we can also use the Temporal Difference (TD) method to get a value of that state (or action). We assume the TD method gives us the right value, so we adjust the parameters of the neural network model of the agent to make two values consistent.

This consistency is what we usually use to train an agent in Deep Reinforcement Learning.

Additional signal beside TD method

What if the TD method doesn't give us the perfect value function? Do we have other sources of learning signals?

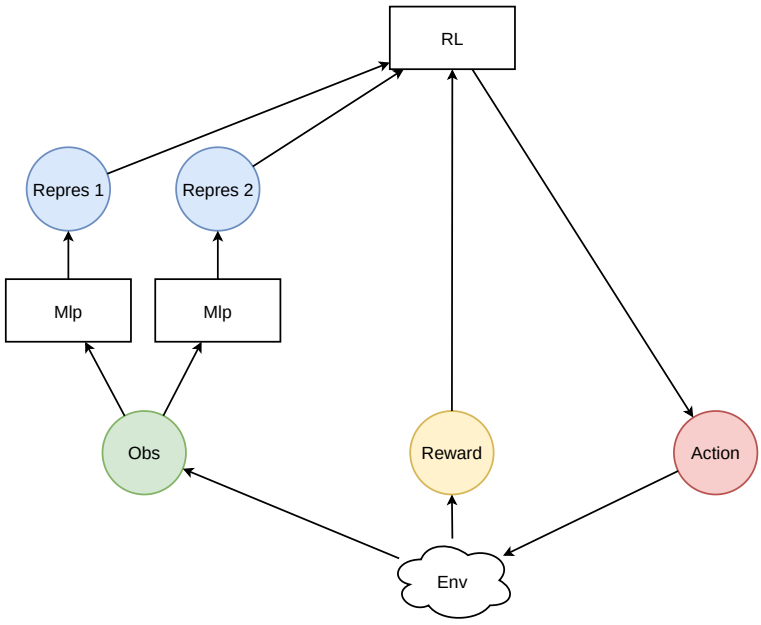
One possible learning signal can come from temporal consistency. For example, we have two observations of two consecutive time steps. We can assume the world doesn't change much in one single step, so the meaning of those two observations should be similar. So we can pass two observations through the neural network model and compare two values produced. They should be consistent. And we can train the neural network model based on this consistency.

This might be an additional source for learning besides the TD method. I mention this because I think this is interesting, but it might not be quite related to my course project.

Course Project Plan

I am planning to do Deep Reinforcement Learning.

I am not sure what the final project will look like, but I'll start the project with this:



This is a standard RL model, with two different MultiLayer Perceptron (MLP) modules. The representation (in blue) will be aggregated into the RL core algorithm.

Later in this project, my job is to produce better representations for the RL core algorithm on the left side. And in the future, maybe use modularized hierarchical RL on the right side.

I will probably start with something similar to the World Model [2], since it utilizes CNN, VAE, and RNN, which are all great tools.

Then, if possible, I'll do a Hierarchical version of the World Model maybe.

Environments

My algorithm should be general, so it can work in different environments.

One possible testbed is the Continuous Control problem. For example, locomotion is my most familiar task.

I also wish to test in some Card Games, so maybe I can ask my agent to play with someone else's agent.

References

[1] Yoshua Bengio. The Consciousness Prior. *arXiv:1709.08568 [cs, stat]*, December 2019. arXiv: 1709.08568.

[2] David Ha and Jürgen Schmidhuber. World Models. *arXiv:1803.10122 [cs, stat]*, March 2018. arXiv: 1803.10122.