CS5180 Project Milestone Report

Zhida Zhang, Meishan Li

As of now, we are in the first half of the project, we have completed the basic research on the project, and we have roughly understood the four possible algorithms that we plan to use as follows:

1. **Natural Evolution Strategies Algorithm:** a numerical optimization algorithm that iteratively updates the continuous parameters of the search distribution by following a natural gradient towards a higher expected fitness.
2. **Genetic Algorithm**: an algorithm designed to search for optimal solutions by simulating the natural evolutionary process. Finding locally optimal solutions of parameters by generating perturbations of a fixed distribution
3. **Augmented Random Search Algorithm**: an algorithm uses a perceptron instead of a deep neural network. It randomly adds tiny values to the weights along with the negative of that value to figure out if they help the agent get a bigger reward.
4. **Advantage Actor Critic Algorithm:** an algorithm consists of two networks (actors and critics) that work together to solve a particular problem. At a high level, the dominance function calculates the TD error or prediction error of the agent and measures the value of the selected action and how the average of all actions

We have also created and tested our environment using Pytorch. We modified the environment used in ex6 as the environment for our project, which allows us to focus more on implementing the algorithm and training the agent:

And we tried to understand how to use GPUs on Discovery so that we could train our agent in a shorter time, which will make our work to be more efficient.

We have done most of what we planned to do, but we have not finished testing and comparing these potential algorithms, we still have not determined which algorithms we will eventually apply in our project. Learning these algorithms has taken more time than we thought. And we spent some extra time trying to understand how to use GPUs on Discovery, because our computers are not powerful enough and we want to shorten the training time to make our testing process more efficient.

Based on our progress so far and the knowledge that we have acquired, the upcoming week-by-week plan is shown below. The immediate next step will be testing and comparing the potential algorithms and deciding one that we will eventually apply in our project, then start coding the main part of our project. The implementation of the algorithm and training process may be more challenging than we thought, we may need more time on that part.

Week-by-Week Plan:

Week 1 11/17-11/21	Test, compare and decide algorithm we plan to use, start coding.
Week 2 11/22-11/28	Test, debug, compare results
Week 3 11/29-12/6	Test, debug, compare results Develop advanced environment and algorithm (if possible)
Week 4 12/7	Final Presentation