Mentors: Roni Khardon, Palash Chatterjee

Exploring Reinforcement Learning for Planning Algorithm Adaptation

Henry Bobeck

Indiana University UROC Spring 2021 Research Project

Exploring modern reinforcement learning algorithms through OpenAI Gym to collect data to support development of a stochastic planning algorithm

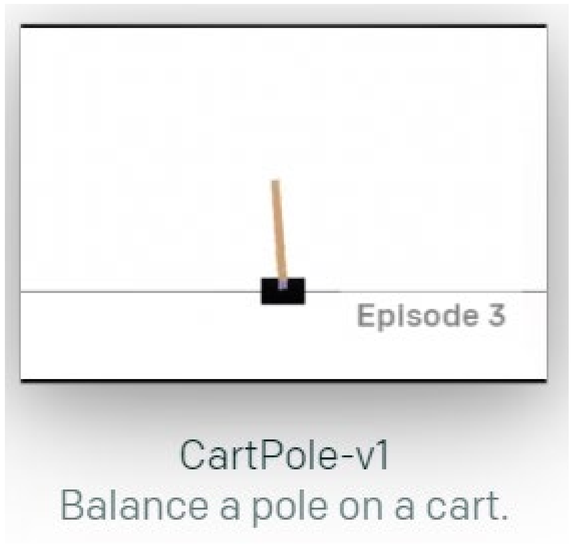
Purpose

The purpose of this research project has been twofold:

1) To learn and become acquainted with state-of-the-art reinforcement learning techniques by training models for the OpenAI Gym environments
2) To collect transition data from these models trained in different physics environments to supply to a team working on developing their advanced stochastic planning algorithm

Methodology

By first building a general workflow which included setting up a framework to merge training/evaluation with data collection, I was then able to test and collect results on increasingly complex environments by following the same few steps:



(1) Initialize an environment through OpenAI Gym

(2) Create a RL model using Stable Baselines 3

(3) Train the model for 10k to 1 million timesteps

(4) Run a modified evaluation function that also collects transition data

(5) Format transition data and send it to the team

Progress so far

After a considerable time spent debugging an error-prone setup process for all the necessary tools and licensing required, I have been able to:

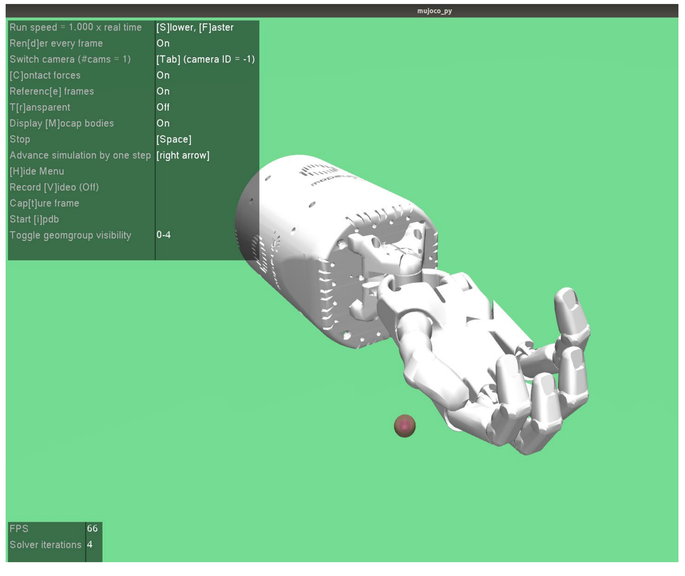
- Modify Stable Baseline's evaluation code to involve data collection procedures
- Set up a method of formatting and saving the data to be sent off to the team
- Successfully trained and evaluated from the CartPole and continuous MountainCar environments using PPO
- Implemented an epsilon-greedy approach during the evaluation process

Challenges

- Setting up a functioning Windows-based environment for Python, OpenAI Gym and MuJoCo despite only officially supporting Linux
- Modifying the model's evaluation code to include data collection of each consecutive environment state and chosen action
- Finding the ideal hyperparameters to get a successfully trained model for environments with sparse reward systems

Tools Used

- Python
- Anaconda
- JupyterLab
- OpenAI Gym
- MuJoCo Physics Simulation
- Stable Baselines 3
- NumPy, pandas



Conclusion

By making use of a sampling of state-of-the-art libraries and tools for reinforcement learning, this project has given me a very practical and hands-on introduction to this ever-evolving field, allowing me to quickly build off my previous ML knowledge. Plus, the process of working to support Roni and Palash has given me a valuable insight to the realm of AI research and how I may see myself a part of it in the future.