

Multi-Agent Reinforcement Learning

Algorithm –

Multi-agent reinforcement learning is a challenging task as multiple agents need to be trained in an environment. In this environment we will be training 2 agents in a competitive setting. Both the agents were trained using the Deep Deterministic Policy Gradients algorithm. This algorithm is an extension of actor-critic methods to continuous action space. These models directly map input states to actions instead of outputting a probability distribution over discrete actions spaces.

Unity ML Agents –

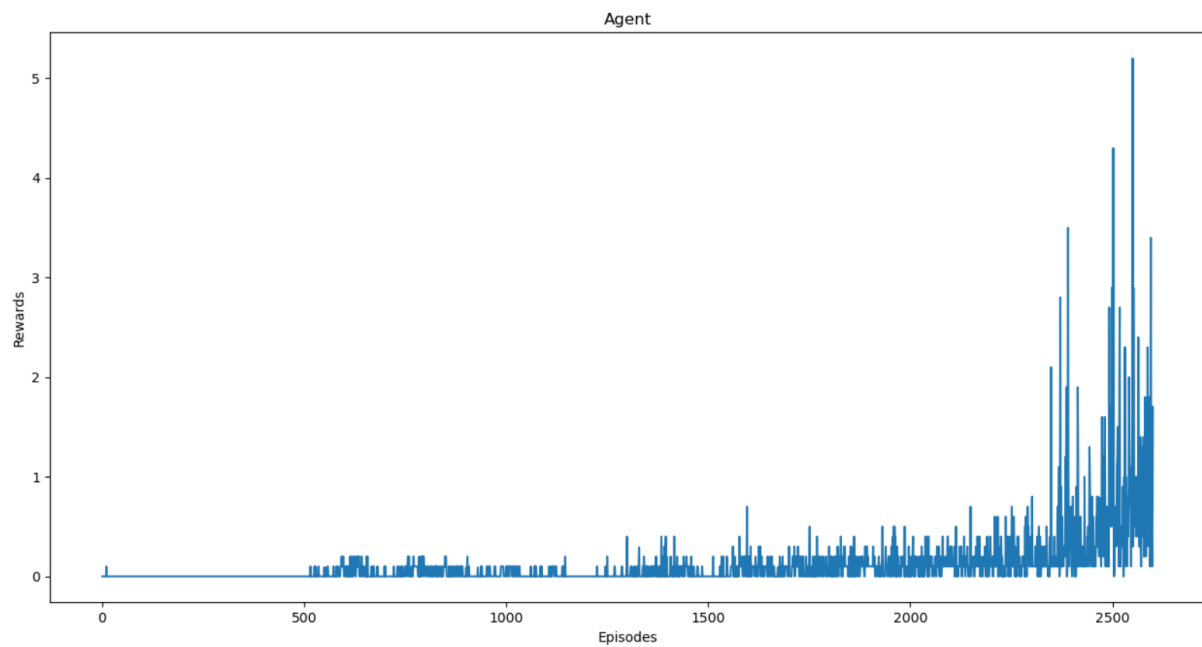
Unity Machine Learning Agents were developed by unity to provide game and simulation environments as training grounds for reinforcement learning. It is an open source plugin which contains already built environments along with the capacity to allow users to create their own environments. In this repository, an environment made by Udacity is used. There won't be a need to download Unity for this project. The implementation won't work with the latest Unity ML-Agents. To reproduce the results, you will need to use this environment.

Implementation –

This algorithm is implemented using PyTorch library. The neural network used for both the actors consists of 3 layers with 600, 400, and 200 neurons respectively. The critic network consisted of 2 layers with 400 and 300 neurons. The critic network had a weight decay factor of 0.00001. The agent was trained using Adam optimizer with a learning rate of $1e-4$ for both the actors as well as the critic network. The buffer length was $1e5$ and batch size was 256. Tau used for soft update was $1e-3$. The discount factor was set to 0.995. Every episode was run for a maximum of 1000 timesteps. The agent took approximately 1 hour on GPU to train.

Results –

The environment is considered solved if any one of the agents get an average reward of 0.5 or more over 100 episodes. My agents solved the episode in 2600 episodes.



Future Work –

I would like to solve the tougher 'Soccer' environment as my future work for this project.