Solving the Multi-Agent Reacher Environment with a Deep Deterministic Policy Gradient Method

Jason Kutarnia

January 28, 2019

1 Learning Algorithm

This repository extends the Deep Deterministic Policy Gradient method to work in a multi-agent environment. It is based on code provided in Udacity's DeepRL Nanodegree. Key features of this architecture include the definition of actor and critic networks as well as a replay memory which is uniformly sampled before each learning step. The introduction of the actor/critic architecture allows the DQN algorithm to solve continuous control problems. The actor network outputs the deterministic action the agent should take, while the critic network attempts to learn the value function. Noise is introduced to the actor network to improve convergence through increased exploration. The replay memory addresses the correlation problem between states occurring close together in time which keeps the network from reinforcing undesirable behaviors.

To solve the multi-agent reacher problem the agents share their experience via a replay memory, which is sampled during the learning step. Although there were multiple agents they shared the same actor/critic networks so the addition of more agents simply made experience gathering more efficient. After every 20 experiences the agents learned by taking a step using the ADAM optimization method. Parameters for the networks can be found in the *models.py* and *ddpg_agent.py* files.

2 Plot of Rewards

Figure 1 shows that the DDPG agent successfully solved the environment after 106 episodes, which occurs when the agent's average performance over the last 100 episodes is at least 30.0.

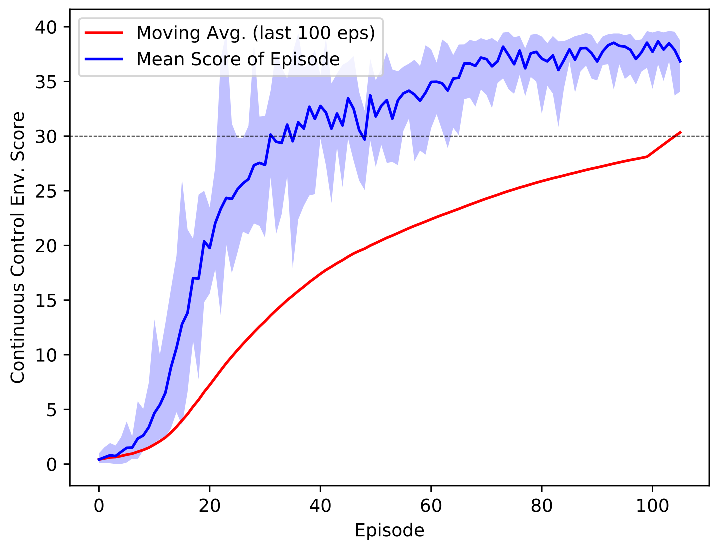

Figure 1: Plot of rewards for DDPG agent

3 Ideas for Future Work

In the paper "Benchmarking Deep Reinforcement Learning for Continuous Control" the authors discuss many additional algorithms which may be more effective for this task, including REINFORCE and Trust Region Policy Optimization.