Report

1. Introduction

This is a report file that details the steps done by us to solve the "Reacher" Environment. It describes the learning algorithm used, along with the number of steps performed to solve the task and the extensions made by our team that we plan to do in the future.

2. Learning Algorithm

We decided for complicity's sake to solve the one agent environment. As described in the README, the algorithm is based on ths Udacity's Deep Reinforcement Learning Nanodegree exercise on DDPG. Inspired by the structure in that exercise, we used two files, one to specify the model (AC_model.py) and one to run the critic and the runner (Agent_Continuous_Control.ipynb) which are described in the README. The modifications from that implementation are as follows:

- AC_Model.py: just like in that workspace, we use an agent consisting of three fully connected layers, and a batch normalization layer that acts upon the first layer. The first layer has 256 units, the second one has 128 units and the number of units of the final layer is equivalent to the number of actions spaces in the environment, in this case four. We used RELU activation units throughout the DNN.

- Agent_Continuous_Control.ipynb: After we have the model architecture, we define the DDPG agent, in order to teach him how to learn, act, get the replay buffer, etc. The hyperparameters used to run the network are as follows:

  -Buffer size: 1,000,000
  -Batch size: 128
  -Discount factor: 0.99
  -Tau = 0.001
  -Actor learning rate: 2e-4
  -Critic learning rate: 2e-4
  -Weight decay: none

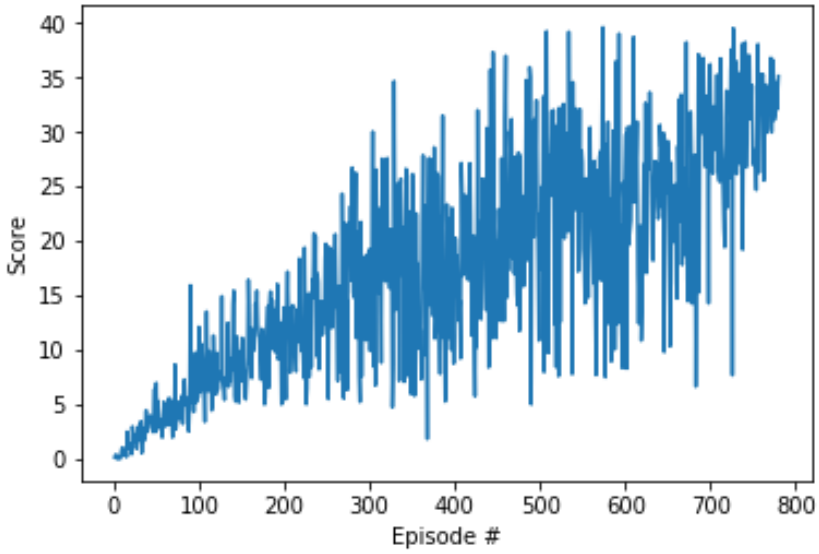
  For the OU noise, the parameters chosen were mean 0, theta is 0.15 and sigma is 0.1. The network updates every 20 episodes and updates over 10 steps.

  Another differences are that different since we are using a Unity environment, whereas we used OpenAI gym for the previous assignment. The code to make the agent act differs by the "brain" operator, among other considerations.

3. Results

After a lot of tries (more than 15 for sure), The agent was able to achieve the desired score over 680 episodes.

Training graph summary


Score
Episode #
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500
600
700
800

4. Considerations for future work

While we understand that we could run the one agent environment using other actor critic methods or other policy based methods, this exercise really drained us, so we're not doing that. Instead, for future work we intend to solve the "Crawler" environment, as well as to solve the environment with multiple agents using D4PG, PPO or other parallel processing. Despite that, you'll have to wait for those updates, since we intend to do the work with the "Bananas environment", as well as the last project before we do this.