Report

1. Introduction

This is a report file that details the steps done by us to solve the "Tennis" 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 the Udacity's Deep Reinforcement Learning Nanodegree exercise on MARL DDPG, as well as some fixes used by some users in the github community on how to work with two agents (since the MARL file was very complex). For this one we decided to use the implementation mostly used in the github community, which separates each process in the file in a separate process. We took a lot of the hyperparameters from the previous project on continuous control, as can be seen from the main jupyter file, the interesting modifications from the implementation are as follows:

-Neural network: two hidden units of size 256 and 128 for both the actor and the critic
-Buffer size: 1,000,000
-Batch size: 1024
-Discount factor: 0.995
-Tau = 0.001
-Actor learning rate: 1e-4
-Critic learning rate: 3e-4
-Weight decay: none

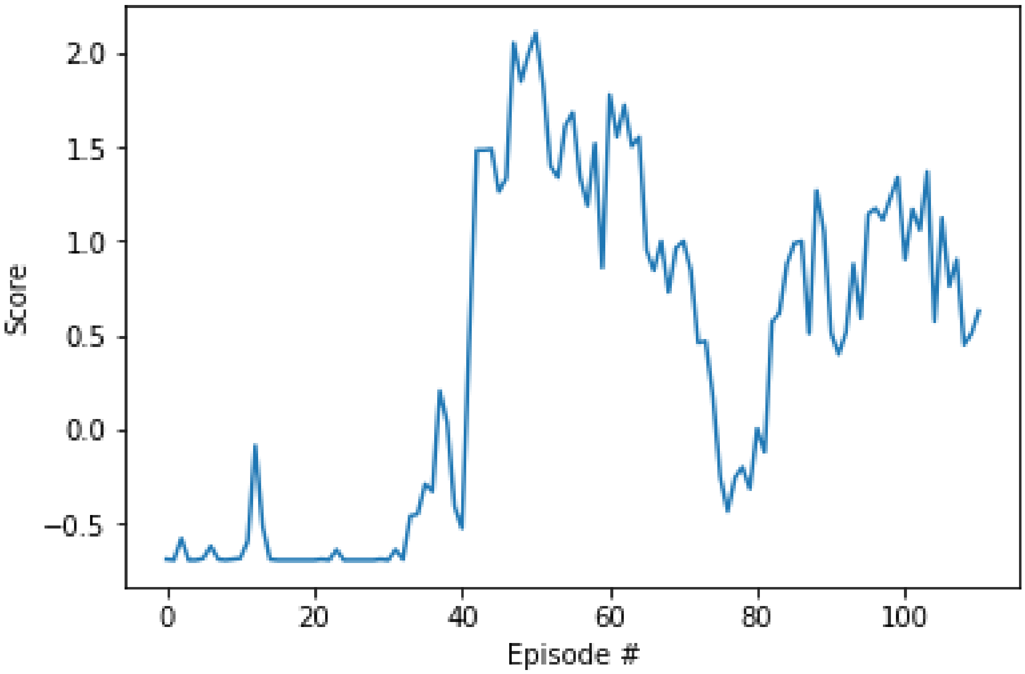
For the OU noise, the parameters chosen were mean 0, theta is 0.15 and sigma is 0.2. The network updates every 4 episodes.

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

The first thing we have to disclose here was that we didn't understand why we ought to take the maximum score of each agent, instead of the average score which is more representative of both performances. So we made a mistake and ran the algorithm using the average score. Using this metric, the agent was able to achieve the desired score over 111 episodes. If desired by Udacity, we can run the algorithm using the max score, but since the solution requires at least 100 episodes to be run and, by construction, our metric is lower than Udacity's metric, we don't foresee very different changes in performance. As you can see it has two very high peaks and very low valleys.

Training graph summary


2.0
1.5
1.0
0.5
0.0
−0.5
Score
0
20
40
60
80
100
Episode #

4. Considerations for future work

If we’re being honest, we did this exercise on a rush given the deadlines and lack of time, we look forward to using different algorithms other than the ones specified in lesson one to solve this environment, in particular using methods such as specified in the paper “Multi-agent reinforcement learning: An overview[1]”. More importantly, we look forward to expanding our knowledge of MARL solving different environments, starting with the soccer environment, which looks more fun.

[1] L. Bus¸oniu, R. Babuska, and B. De Schutter, “Multi-agent reinforcement learning: ˇ An overview,” Chapter 7 in Innovations in Multi-Agent Systems and Applications – 1 (D. Srinivasan and L.C. Jain, eds.), vol. 310 of Studies in Computational Intelligence, Berlin, Germany: Springer, pp. 183–221, 2010. Retrieved on August 9th, available here: http://www.dcsc.tudelft.nl/~bdeschutter/pub/rep/10_003.pdf