Multi-Agent Project

Tennis Environment

In this project, I solved the Tennis environment with DDPG (Deep Deterministic Policy Gradient) algorithm.

In this environment, two agents control rackets to bounce a ball over a net. If an agent hits the ball over the net, it receives a reward of +0.1. If an agent lets a ball hit the ground or hits the ball out of the bounds, it receives a reward of -0.01. The goal of each agent is to keep the ball in play.

The observation space consists of 8 variables corresponding to the position and velocity of the ball and racket. Each agent receives its own, local observation. Two continuous actions are available, corresponding to movement toward (or away from) the net, and jumping.

The task is episodic, and I order to solve the environmet, your agents must get an average score of +0.5 (over 100 consecutive episodes, after taking the maximum over both agents).

DDPG (Deep Deterministic Policy Gradient)[1,3]

DDPG is an algorithm that simultaneously learns a Q-function and a policy. It uses off-policy data and the Bellman equation to learn the Q-function and uses the Q-function to learn the policy.

Goal

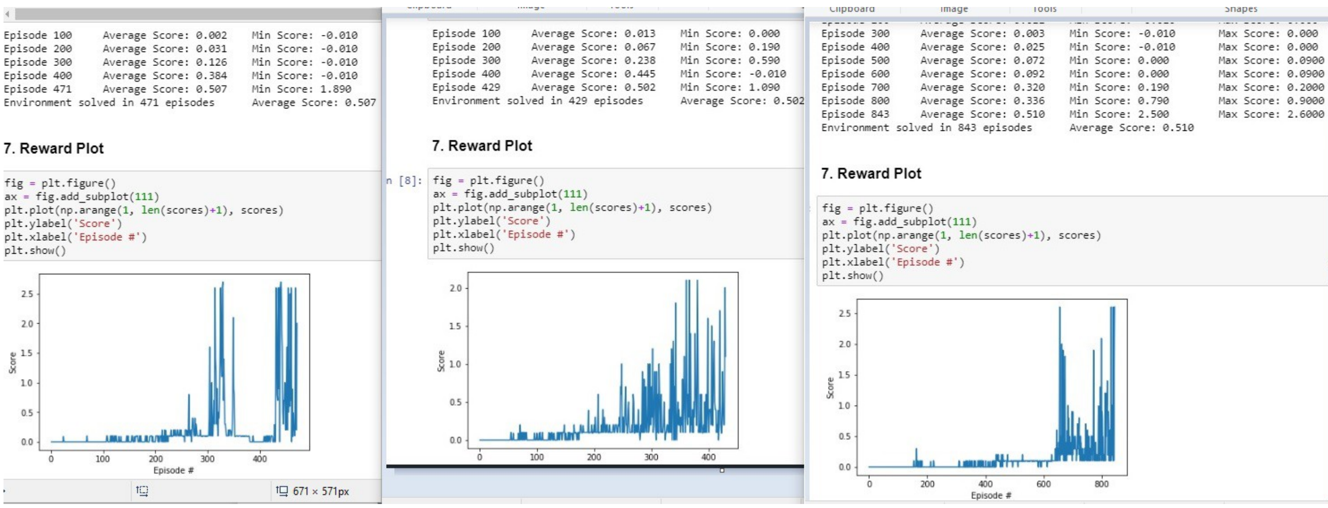
The environment is considered solved, when the average (over 100 episodes) of those scores is at least +0.5.

Learning Algorithm

I followed the code provided by Udacity repository DDPG Pendulum.[1]
My hypothesis was that I can solve this environment in 2000+ episodes based on most references I read and that DDPG is unstable. However, I found quite interesting results from my experimentation.
Below is the example of training results with hyperparameters I used in project 2.

```
Episode 100     Average Score: 0.00     Max Score: 0.00
Episode 200     Average Score: 0.00     Max Score: 0.00
Episode 300     Average Score: 0.00     Max Score: 0.00
Episode 400     Average Score: 0.00     Max Score: 0.00
Episode 500     Average Score: 0.00     Max Score: 0.09
Episode 600     Average Score: 0.04     Max Score: 0.10
Episode 700     Average Score: 0.05     Max Score: 0.00
Episode 784     Average Score: 0.05     Max Score: 0.00
```

I did some hyperparameters adjustments to improve the agents performances. The hyperparameters that improve agent training significantly are asymmetric learning rate Actor and Critic, increasing Critic learning rate and increasing TAU value. I was able to solve the environment in 843, 471, 429 episodes respectively.



<u>Hyperparameters:</u>

- BUFFER_SIZE= 3e5
  The replay buffer size
- BATCH_SIZE= 128
  Number of inputs processed per batch when running Stochastic Gradient Descent
- GAMMA= 0.99
  Discount factor of the Q-Learning Algorithm
- TAU: 3e-3
  To perform soft updates of the target network parameters
- LR_ACTOR: 2e-4
  Learning rate of the actor
- LR_CRITIC: 6e-4
  Learning rate of the critic

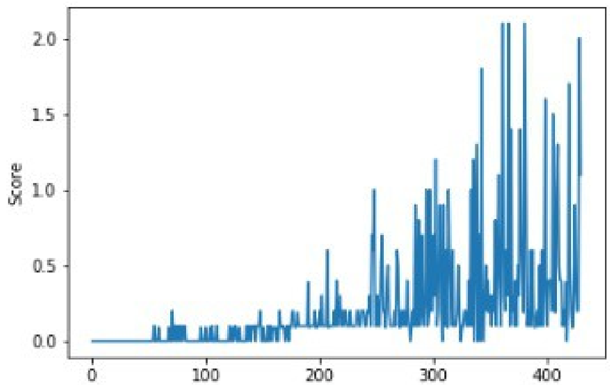
Plot of Rewards

By increasing the Critic learning rates from 2e-4 to 6e-4 and TAU value from 1e-3 to 3e-3, I solved the environment in 429 episodes.

```
Episode 100     Average Score: 0.013    Min Score: 0.000        Max Score: 0.090
Episode 200     Average Score: 0.067    Min Score: 0.190        Max Score: 0.2000
Episode 300     Average Score: 0.238    Min Score: 0.590        Max Score: 0.7000
Episode 400     Average Score: 0.445    Min Score: -0.010       Max Score: 0.100
Episode 429     Average Score: 0.502    Min Score: 1.090        Max Score: 1.1000
Environment solved in 429 episodes      Average Score: 0.502
```

7. Reward Plot

n [8]:
```
fig = plt.figure()
ax = fig.add_subplot(111)
plt.plot(np.arange(1, len(scores)+1), scores)
plt.ylabel('Score')
plt.xlabel('Episode #')
plt.show()
```



Ideas for Future Work

I need to continue this project and will do it with Multi Agent DDPG algorithm and PPO. It is interesting to see how those two algorithms in comparison with DDPG.

References

1. Deep Reinforcement Learning Nano Degree Udacity Course
2. https://github.com/udacity/deep-reinforcement-learning/tree/master/ddpg-pendulum
3. Lowe R., Wu Y., Tamar A., et.al., Multi-Agent Actor-Critic for Mixed Cooperative_Competitive Environments.
   https://arxiv.org/abs/1706.02275