Project 2 Continuous Control Report

Udacity Deep Reinforcement Learning NanoDegree

Author: Denis O’Connor

Implementation

The base of this implementation was taken from the Udacity Deep Reinforcement Learning ddpg-pendulum project. I chose the first option of using a single agent. I ran into many difficulties during this process. When retraining or executing an existing cell that reset the environment, I was getting python Errno 32 Broken Pipe errors. Also, I was experiencing errors where the code would hang on the env_info.reset code. In addition to the base code used, I had to play around with several other elements. The early training would seldomly get over 0.5 for the average score. I also had several instances where I would continue to get 0.0 for average score. After much experience with hyperparameters and adding a call to pytorch’s clip_grad_norm function to the learn method (per udacity knowledge), I was finally able to solve the environment.

Learning Algorithm

Per Medium.com, I used this explanation for the basis of my learning algorithm. I used a Deep Deterministic Policy Gradients approach. Also per Medium.com, “The network architecture is comprised of two fully connected hidden layers of 128 units each with ReLU activations. In order to help speed up learning and avoid getting stuck in a local minimum, batch normalization was introduced to each hidden layer. The hyperbolic tan activation was used on the output layer for the actor-network as it ensures that every entry in the action vector is a number between -1 and 1. Adam was used as an optimizer for both actor and critic networks.” However, through my troubleshooting I commented out the batch normalization after the second layer’s activation. At the time, I was hitting problems where training was resulting in an average score of 0.0. Although, I did not try uncommenting it out after solving some of my other workspace issues.

Hyperparameters

Batch size: 128

Replay buffer size: 1e5

Gamma (discount factor): 0.99

TAU: 1e-3

Actor learn rate: 2e-4

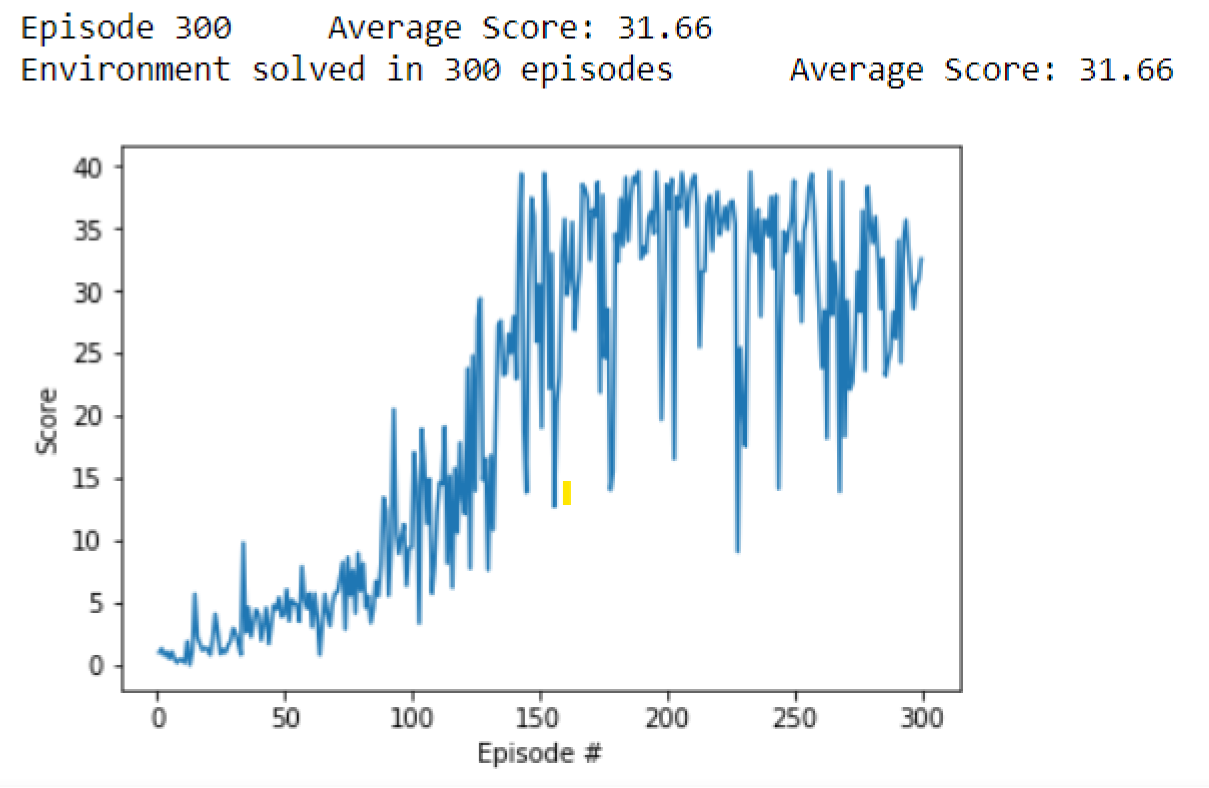
Critic learn rate: 2e-4

Weight decay: 0

OUNoise theta: 0.15

OUNoise sigma: 0.1

Results



```
Episode 300     Average Score: 31.66
Environment solved in 300 episodes      Average Score: 31.66
```

Future Improvements

First improvement I would implement is to reintroduce the batch normalization to the second hidden layer. Second, I would utilize the Option 2 of the Reacher environment (20 Agents).