Project: Deep Reinforcement Learning Continuous Control

Introduction:

This project is a part of Udacity Nanodegree program on Deep Reinforcement Learning. This project is basically about controlling a Robotic arm and making it control its actions and it is solved by Reacher environment. There is a vast variety of observation space and different actions. The project is considered successful if the Robotic arm scores +100 in 33 consecutive steps. This project is the second of three projects which is a part of Deep Reinforcement Learning program.

Implementation:

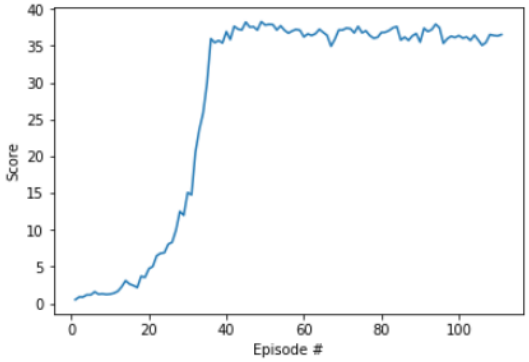
This environment was solved using a deep Reinforcement Learning agent. This deep Reinforcement Learning agent can be found in ddpg.py file. The model.py contains the neural networks used as estimators.

Learning-Algorithm:

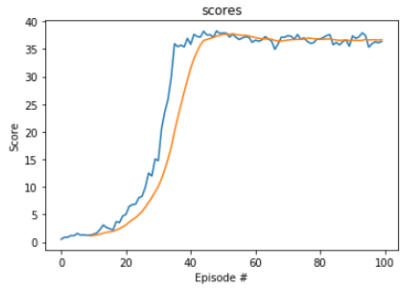
The deep deterministic gradient algorithm is an actor-critic algorithm which was utilized as a learning algorithm. It is quite same as DQN. One advantage of DDPG over DQN is that it can also be utilized to solve continuous spaces. DDPG utilizes four neural networks. A local actor, target actor, local critic, target critic. In each training step, the experience of twenty agents is stored. Then in the second training step, the agent learns from a random sample which is stored in the previous training steps. The actor tries to estimate the optimal policy from each step of actor-critic values and the critic tries to estimate the optimal q-policy and learns by using a normal q-approach. Using this approach, one gains the benefits of value-based and policy-based methods at the same time.

Results:

Following are the two graphs that depict the results of the training that the agent has undergone through this algorithm. The agent was trained on 1000 episodes and for every 100 episodes we had plotted and depicted the result.



```
Episode 100     Average Score: 26.35
Episode 111     Average Score: 30.19
Environment solved in 11 episodes!      Average Score: 30.19
```



Further Improvements:

There can be many improvements that can be done. For example, one can implement DQN improvements. Also, parallel algorithms like A3C can be tried out.