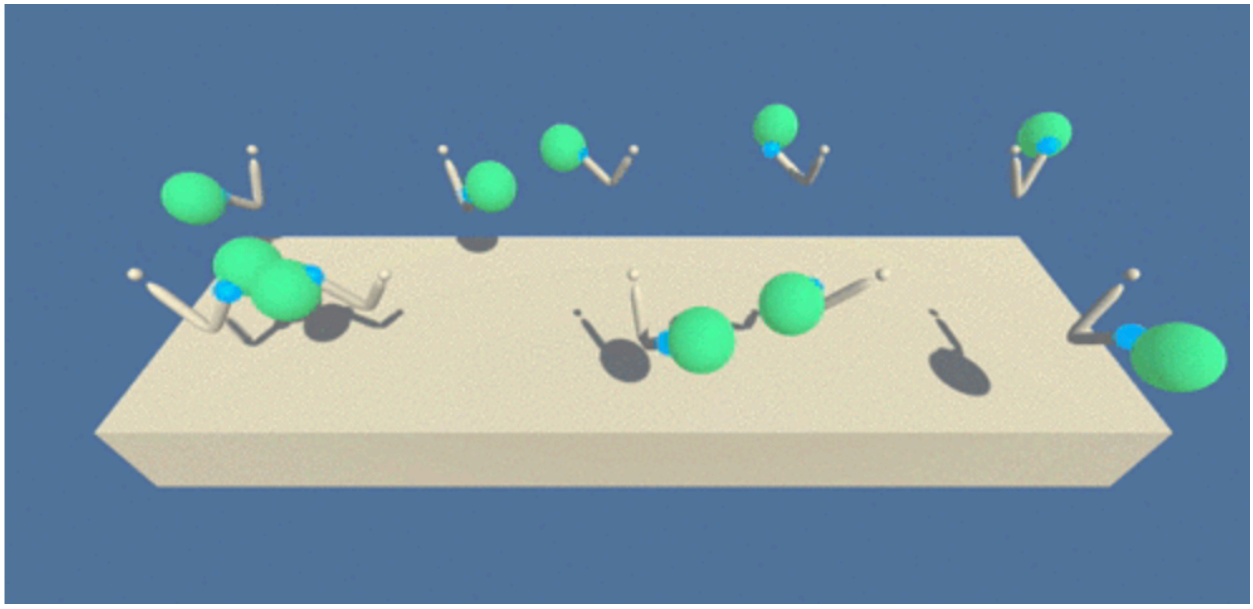
Ali Abdullah Al-Saeedi
Reinforcement Nanodegree
Continuous Control Project
April 2, 2019

Continuous Control



In this environment, a double-jointed arm can move to target locations. A reward of +0.1 is provided for each step that the agent's hand is in the goal location. Thus, the goal of your agent is to maintain its position at the target location for as many time steps as possible.

The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm. Each action is a vector with four numbers, corresponding to torque applicable to two joints. Every entry in the action vector should be a number between -1 and 1.

1.Implementation

The solution is based on the DDPG algorithm, similar to the pendulum exercise. The agent reaches (no pun intended) an average score of 30 after **186** episodes. The main change I made to the original implementation was forcing the initial network weights to be equal for the local and target networks after initializing them, for both actor and critic which helped the agent to learn much faster.

Both actor and critic networks have 2 fully-connected hidden layers of **128 nodes**, both trained with a learning rate of **0.001**, using mini-batches of **256**, a replay buffer of **100000**, and a discount of **0.9**. The Ornstein-Uhlenbeck noise has a sigma of **0.1**and the soft update is made using a tau of **0.001**. The training seems to be fairly stable for a RL task. The score kept on improving after the goal was reached.

Hyperparameters:

There are many Hyperparameters that can be edited:

Number of episodes	n_episodes=2000
Max time per episodes	max_t=1000 (ms)

2.Results

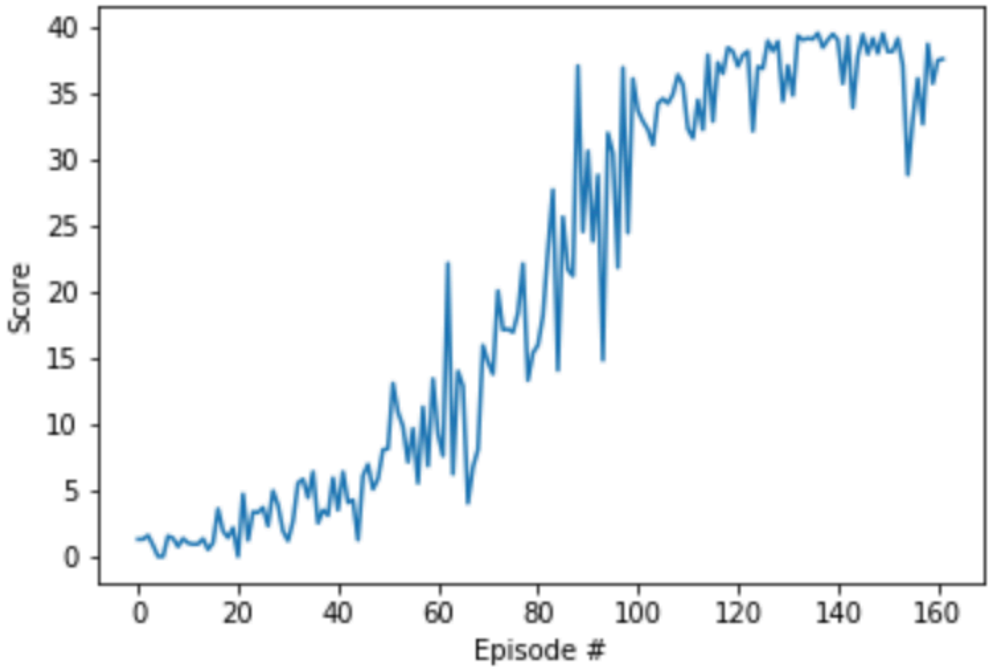
In the training we could solve the environment in 162 episodes. By solving it we mean reaching an average score of at least 30.

```
In [7]: # Training our agent

        scores = ddpg(4000, 1000)
```

```
Episode 100     Average Score: 10.29
Episode 162     Average Score: 30.30
Environment solved in 62 episodes!      Average Score: 30.30
```

This graph shows how while our agent is training we are gaining a better result.



3. Future Work

- We could use different algorithms to try if our agent would give us a better results on training. Such algorithms would be PPO.
- Keep training for a longer number of episodes and see if the average score keeps improving or would it crash.
- Try the multi-agent case.