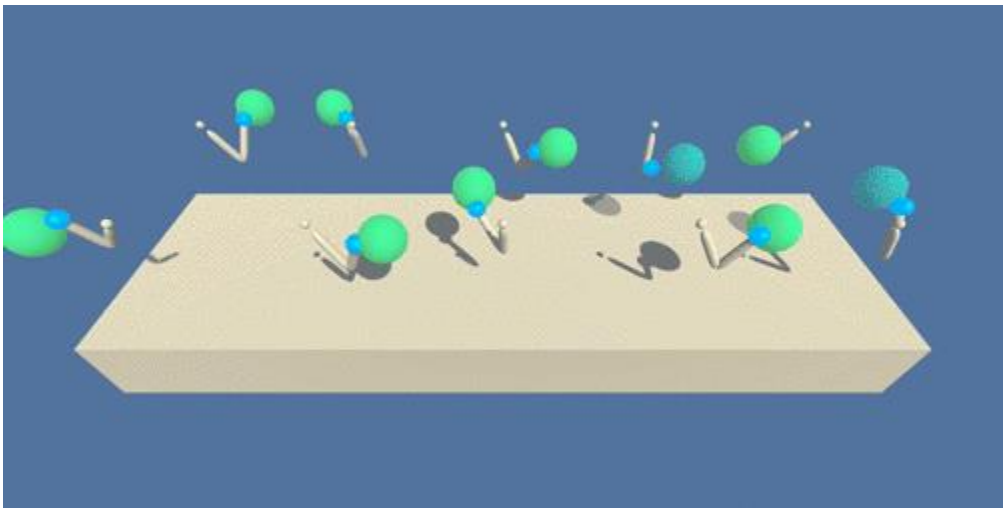
Project 2. Continuous Control

1. Project Introduction

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.

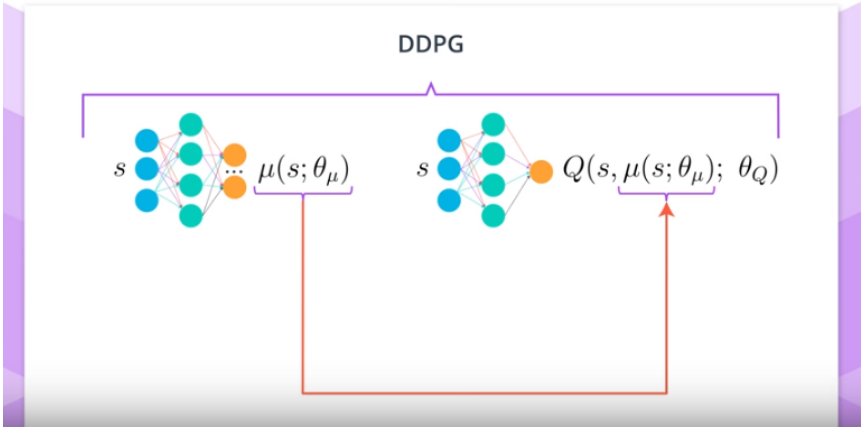


The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm. Each action is a vector with four numbers, corresponding to torque applicable to two joints. Every entry in the action vector should be a number between -1 and 1.

For this project, we will provide with the Unity environment which contains 20 identical agents, each with its own copy of the environment.

2. Learning Algorithm

To train the agent in this project, we use DDPG algorithm which is introduced as an "Actor-Critic" method. DDPG is a very successful method and it's good for us to gain some intuition. The picture below shows the model architectures for the actor and critic networks



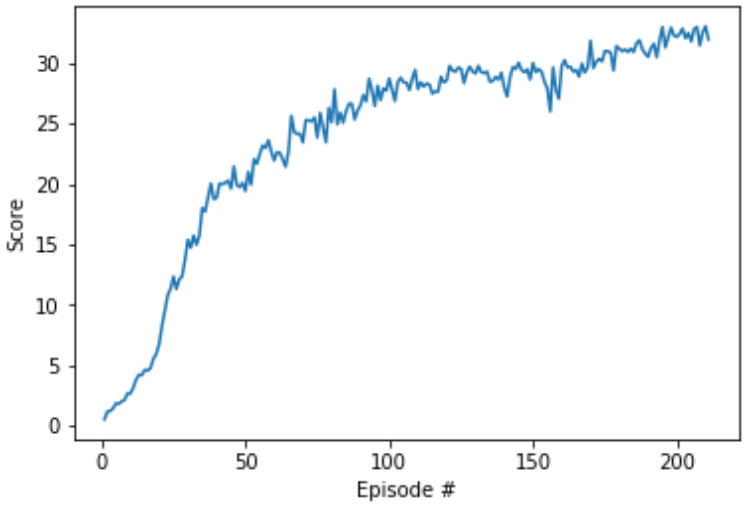
Following are the value of the hyperparameters used for my training on DDPG algorithm.

```
BUFFER_SIZE = int(1e6)  # replay buffer size
BATCH_SIZE = 128        # minibatch size
GAMMA = 0.99            # discount factor
TAU = 1e-3              # for soft update of target parameters
LR_ACTOR = 1e-5         # learning rate of the actor
LR_CRITIC = 5e-5        # learning rate of the critic
WEIGHT_DECAY = 0        # L2 weight decay
```

I used the Unity environment containing 20 different agents. At each step of the environment, (state, action, reward, next_state, done) tuples are computed for each of the 20 agents and added to the replay buffer.

3. Plot of Rewards

The plot is as below, where we have plotted the **average score** (over all 20 agents) obtained with each episode.



The environment is considered solved, when the average (over 100 episodes) of those **average scores** is at least +30. In the case of the plot above, the environment was solved at episode 111, since the average of the **average scores** from episodes 112 to 211 (inclusive) was greater than +30.

4. Ideas for Future Work

In this project, we implemented several reinforcement learning algorithms, and presented them in the context of general policy parameterizations. Results show that the DDPG algorithms are effective methods for training deep neural network policies. Still, the poor performance on the proposed hierarchical tasks calls for new algorithms to be developed. Implementing and evaluating existing and newly proposed algorithms will be our continued effort.