For this project, I worked with the environment in the *https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Learning-Environment-Examples.md#reacher*.

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 my agent is to maintain its position at the target location for as many time steps as possible. My agent has been trained using the Udacity Workspace.

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, I was provided with two separate versions of the Unity environment:

The first version contains a single agent.

The second version contains 20 identical agents, each with its own copy of the environment.

The second version is useful for algorithms like PPO, A3C, and D4PG that use multiple (non-interacting, parallel) copies of the same agent to distribute the task of gathering experience.

Since my project submission need only solve one of the two versions of the environment. In this project I used the first version. In the first version,the task is episodic, and in order to solve the environment, my agent must get an average score of +30 over 100 consecutive episodes. My agent was able to achieve this target at 686 episodes.

In future I will like to see the result for 20 agents using DDPG. In addition I will like to consider what other algorithms like A3C, D4PG and PPO will yield.