# Collaboration and Competition Project report

## The environment

In this environment, two agents control rackets to bounce a ball over a net. If an agent hits the ball over the net, it receives a reward of +0.1. If an agent lets a ball hit the ground or hits the ball out of bounds, it receives a reward of -0.01. Thus, the goal of each agent is to keep the ball in play.

The observation space consists of 8 variables corresponding to the position and velocity of the ball and racket. Each agent receives its own, local observation. Two continuous actions are available, corresponding to movement toward (or away from) the net, and jumping.

The task is episodic, and in order to solve the environment, your agents must get an average score of +0.5 (over 100 consecutive episodes, after taking the maximum over both agents). Specifically,

After each episode, we add up the rewards that each agent received (without discounting), to get a score for each agent. This yields 2 (potentially different) scores. We then take the maximum of these 2 scores.

This yields a single score for each episode.

The environment is considered solved, when the average (over 100 episodes) of those scores is at least +0.5.

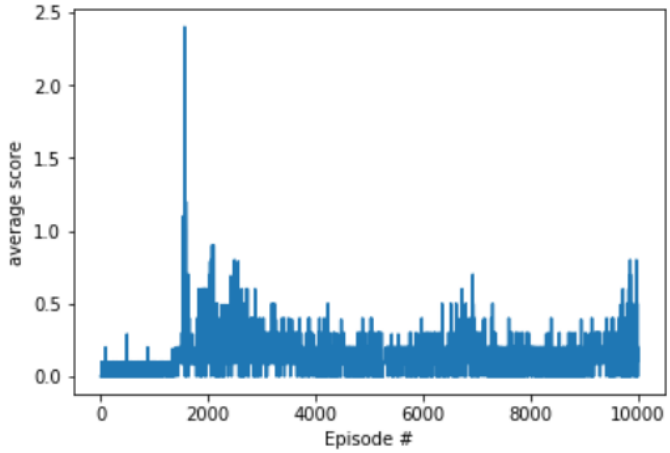
## Solution

I have used ddpg algorithm for solving this problem. The ddpg agent has got an actor and critic represented by a neural network. There is a replay buffer to store the experiences between agent and environment. The experiences are sampled from the buffer to adjust the neural network weights.

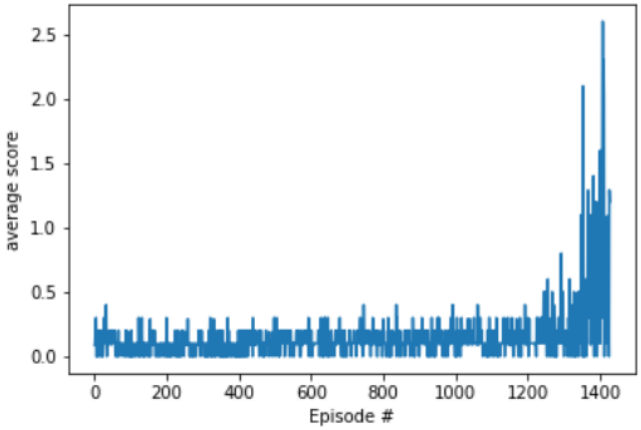
Hyperparameters used for training the agents are:

```
episode_length = 80                 #max length of any episode
BUFFER_SIZE = 1e5                   # replay buffer size
BATCH_SIZE = 512                    # minibatch size
GAMMA = 0.99                        # discount factor
TAU = 0.05                          # for soft update of target parameters
LR_ACTOR = 1.0e-4                   # learning rate of the actor
LR_CRITIC = 5.0e-4                  # learning rate of the critic
episode_per_update = 5              # how often to update the network
```

The below plots shows agent's score as episodes progress. First it ran for 10000 episodes:




Then I reran it after loading the saved weights:




```
Episode 500     Average Score: 0.10     Max Score: 0.10
Episode 1000    Average Score: 0.13     Max Score: 0.10
Environment solved in 1327 episodes (average score 0.51).
```

## Next steps:

To continue improving the learning of the agents, we can try the following:

- Changing the actor and critic networks hidden layers size and tweaking the hyperparameters.
- Prioritize experience replay: https://arxiv.org/abs/1511.05952
- Using multi agent ddpg algorithm
- Train using raw pixels of the environment state
- Also, we can apply this algorithm to the soccer environment and see how it performs