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

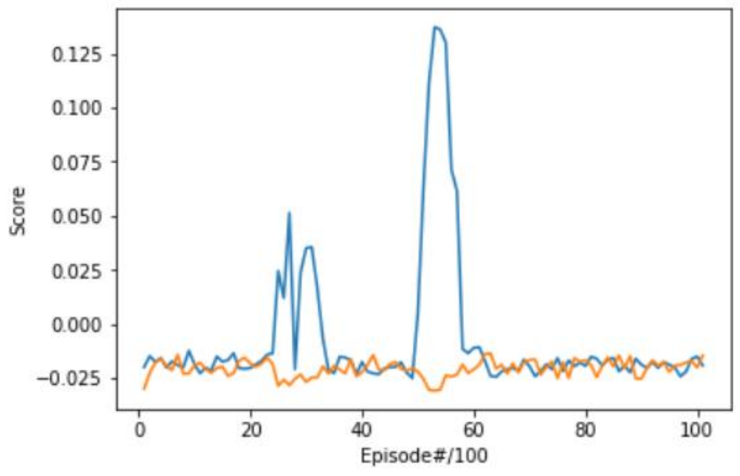
I have used multi agent ddpg (MADDPG) method for solving this problem. The paper describing this algorithm can be found here. I have used two ddpg agents, one for each player. Each ddpg agent has got and actor and critic represented by a neural network.

Hyperparameters used for training the agents are:

episode_length = 80	#max length of any episode
BUFFER_SIZE = 5000*episode_length	# replay buffer size
BATCH_SIZE = 128	# minibatch size
GAMMA = 0.95	# discount factor
TAU = 0.02	# for soft update of target parameters
LR_ACTOR = 3.0e-4	# learning rate of the actor
LR_CRITIC = 3.0e-4	# learning rate of the critic
WEIGHT_DECAY = 1.e-5	# L2 weight decay
episode_per_update = 4	# how often to update the network

Most of the code is reused from MADDPG lab as part of the nanodeegre.

The below plot shows how two agent’s average score after every 100 episode:



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
- Also, we can apply this algorithm to the soccer environment and see how it performs