# Report of Deep reinforcement learning Nano-degree Project 3: Collaboration and Competition

2019-2-24

This report summarizes the brief detail of my implementation of this project.

## 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.

## Algorithm

I borrowed the typical Deep Deterministic Policy Gradients (DDPG) framework of the former lessons, and Multi DDPG agent's framework are built based on it.

Two agents have separate networks and same actor-critic structures. As suggested in the benchmark, they shared the same replay buffer in this project.

Parameters:

Actor:
Full connected hidden layer 1: (input states, 256)
ReLU function
Batch Normalization of layer 1
Full connected hidden layer 2: (256, 256)
ReLU function
Output layer (256, actions)
Tanh function

Critic:
Full connected hidden layer 1: (input states, 256)
ReLU function
Batch Normalization of layer 1

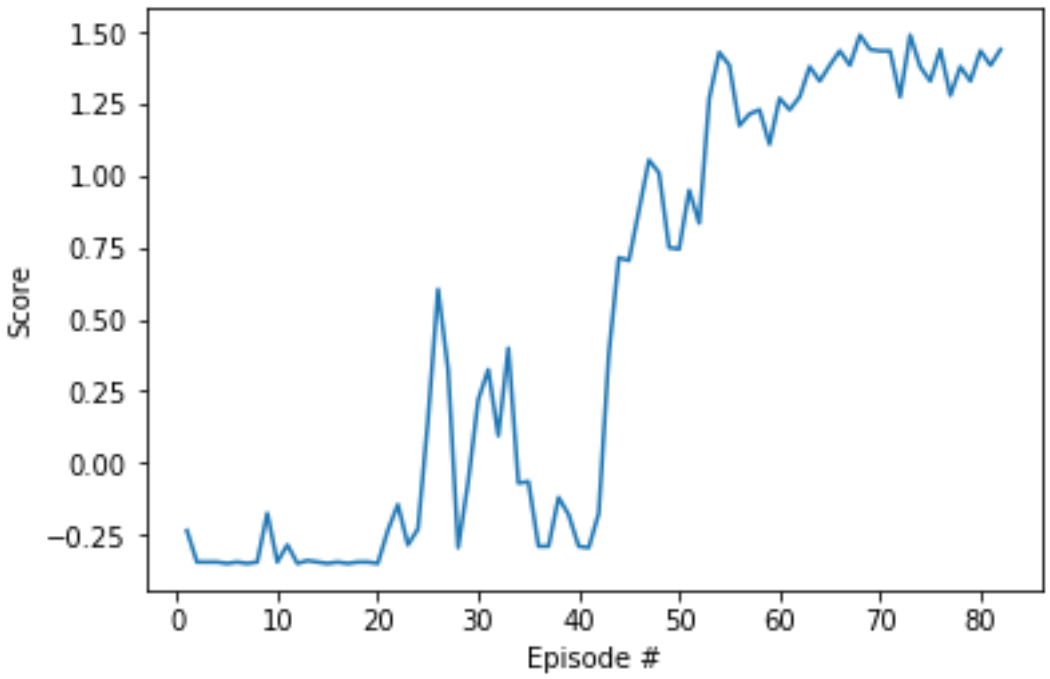
Full connected hidden layer 2: (256+action_size, 256)
ReLU function
Output layer (256, 1)
Linear function

Hyperparameters of the agent:

```
BUFFER_SIZE = int(1e5) # replay buffer size
BATCH_SIZE = 256            # minibatch size
GAMMA = 0.99                  # discount factor
TAU = 1e-3                    # for soft update of target parameters
LR_ACTOR = 1e-4               # learning rate of the actor
LR_CRITIC = 1e-4            # learning rate of the critic
WEIGHT_DECAY = 0              # L2 weight decay
OUNoise: mu=0., theta=0.15, sigma=0.2
Update_frequecy=2
```

## Reflection and Results

In this project, the main effort is to re-construct the framework which could train multi-agents with DDPG in this environment. The basic DDPG framework without changing in noise (small noise may cause slow learning in this case) or adding grad norm was applied.
The plot of training result is follows:




It seems that a larger Batch size for the shared replay buffer would be useful. Since these two agents could be considered to face very similar experiences, it would be helpful for them to

learn from each other.

## Perspective

It would be interesting to see what happens if these two agents did not share replay buffers or both agents share one DDPG network. Also, I would like to check out other algorithms used by others for this project.