PROJECT REPORT

Topic – Collaboration and Competition using deep reinforcement learning

Name – Pranav Sivadas Menon

Environment –Tennis Environment from Unity Machine Learning Agents toolkit.

Goal – Train two agents to control rackets to bounce a ball over a net

Reward structure - 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.

State space –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.

Action space – Two continuous actions are available, corresponding to movement toward (or away from) the net, and jumping.

Stopping Criterion - The task is episodic and is considered solved when the agents get an average score of +0.5 (over 100 consecutive episodes, after taking the maximum over both agents): After each episode, we add up the rewards that each agent received to get a score for each agent. We then take the maximum of these 2 scores.

In this project I trained a single model: Multi agent deep deterministic policy gradients. MADDPG is a multi-agent version of DDPG. Each agent has its own actor and critic and for this project I utilized a common replay buffer. Each agent learns its own reward function and incorporates actions of other agents in its learning by sampling from this replay buffer.

For more details please refer this paper

Neural Network:

Actor:

• 3 linear layers. The first hidden layer has 256 units and second hidden layer has 128 units. Parameters are reset according to number of units in the layers i.e

lim = 1. / np.sqrt(f_in)

• Output size – action_size (vector of 2 numbers since it is continuous action space).Tanh activation

Critic:

• 3 linear layers. The first hidden layer has 256 units and second hidden layer has 128 units. Parameters are reset according to number of units in the layers i.e

lim = 1. / np.sqrt(f_in)

• The action is fed into the 2nd layer
• Output size – 1 (action value function)

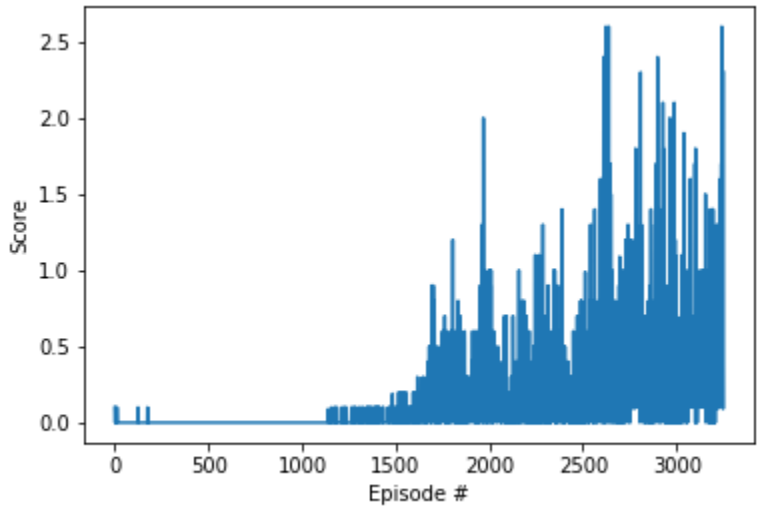
The state is batch normalized in both the actor and critic.

All hidden layers use RELU non linearity

Hyperparameters chosen:

batch_size = 128
gamma = 0.99
buffer_size = int(3e5)
TAU = 1e-3
lr_actor = 1e-3
lr_critic1 = 1e-4
Update every = 2

Plot of rewards



The environment was solved in 3152 episodes

Ideas for future work

- Try multi agent PPO on the same environment
- Try agents with different replay buffers

- Apply algorithm to the soccer environment