Algorithm

First, DDPG (Deep Deterministic Policy Gradient) [1] was selected to solve this problem. Each agent (racket) uses the same Actor and Critic models to bounce the ball over the net. The Tennis Environment treats the experience (s, a, s', r) of both agent as a set. More specifically, the environment begins from a set of states of the both agents (s_1, s_2), takes actions of the both agents (a_1, a_2), and returns a set of next states and rewards (s'_1, s'_2, r_1, r_2) for the both agents (note that the subscripts 1 and 2 represent each racket). In the training of DDPG, however, the experience of each agent, (s_1, a_1, s'_1, r_1) and (s_2, a_2, s'_2, r_2), was treated as an independent experience to train the Actor/Critic models. The theory behind this approach is that, if the Actor/Critic models for a racket is trained to achieve high reward (which is to bounce the ball over the net), the rally will get longer as the consequence, resulting in a higher score.

The idea seems to work well. The Actor/Critic models were learned reasonably quickly. The environment was solved (average score of +0.5 over 100 consecutive episodes) less than 1000 episodes without making so much effort to tune the hyperparameters. However, based on the discussion in the slack channel [2], I played with the soft-update τ several times and finally got the hyperparameters with which the problem was solved at episode 317. Figure 1 shows the Actor/Critic models, and Table 1 shows the hyperparameters used for the training. Figure 2 shows the average score plot for the training.

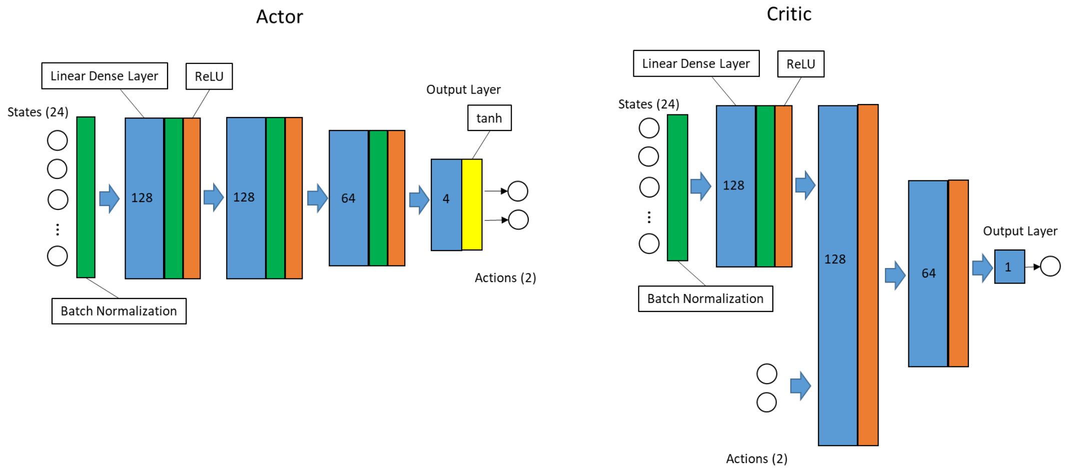

Figure 1: Actor/Critic Models

Table 1: Hyperparameters

Replay Buffer Size	10,000
Batch Size	512
Discount	0.9
Actor Learning Rate	0.001
Critic Learning Rate	0.001
Soft Update τ	0.1
OU Noise θ	0.15
OU Noise σ	0.01

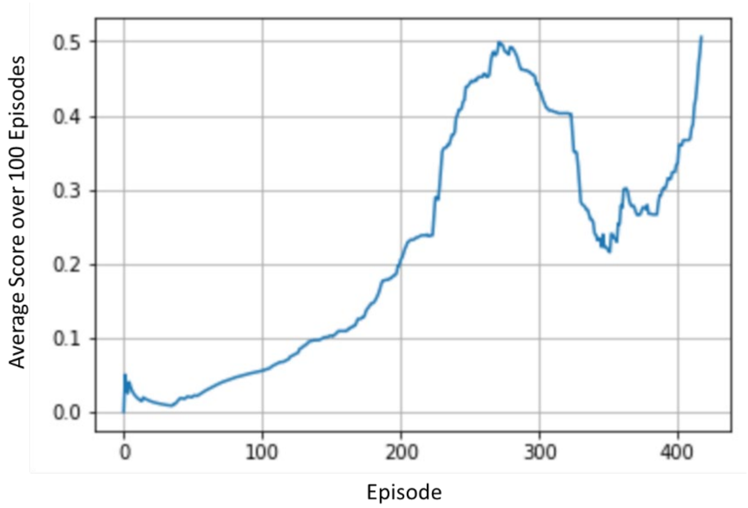

Figure 2: Score Plot for the DDPG model

Future Work

I am now examining a variety of MADDPG methods to find a model which solves the environment faster than the DDPG model. However, it was found that tuning the hyperparameters for MADDPG is quite difficult. I tried MADDPG introduced in [3] in which each agent has their own Actor/Critic models, and the Critic model takes the observations and actions from both agents. I also tried MADDPG having a centralized Critic model [4]. Despite of massive efforts to tune the hyperparameters including the model architecture of the Actor/Critic models, the learning performance is still far from that of the DDPG model. The best average score plot so far is shown in Figure 3. This score was obtained with a MADDPG method introduced in [3]. As shown, the environment is solved at episode 1537 which is much slower than the DDPG model. Hence, my future work is to pursue this MADDPG research to find a model which at least learns as quick as the DDPG model.

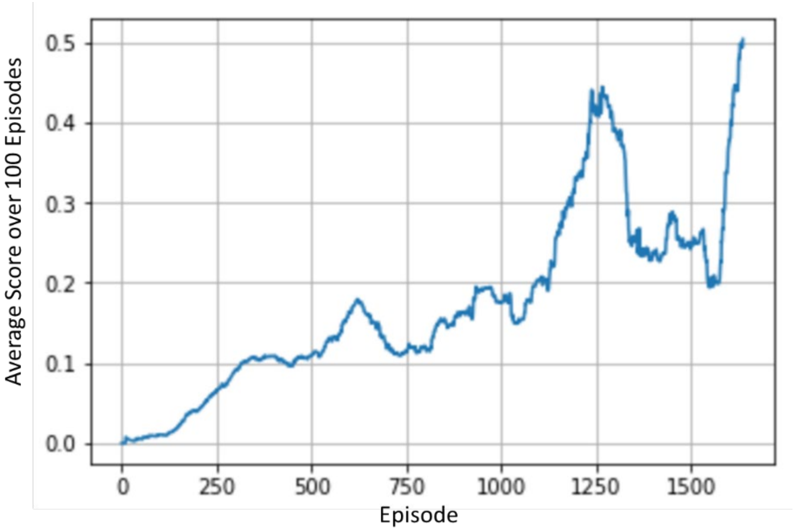

Figure 3: Score Plot for the MADDPG model

Reference

[1] Timothy P. Lillicrap, *et al*. “Continuous Control With Deep Reinforcement Learning”, ICLR, 2016.

[2] Udacity Deep Reinforcement Learning Nanodegree, Slack channel #project-3_collab-comp

[3] Ryan Lowe, *et al*. “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments”, NIPS, 2017.

[4] Jakob N. Foerster, *et al*. “Counterfactual multi-agent policy gradients”, 2017.