Collaboration and Competition

Sudhir Kumar Suman [16D070027]
Saarthak Kapse [16D070041]

Goal:
The goal of the project is to train two agents to play tennis, the agents must bounce ball between one another while not dropping or sending the ball out of bounds.

Introduction:
We will work with the Tennis[1] environment where two agents control rackets to bounce a ball over the net. If an agent hits the ball over the net, it will receive a reward of +0.1. If an agent lets a ball hit the ground or hits the ball out of bounds, it will receive 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 a ball and racket. Each agent receives its own, local observation. Two continuous actions are available, corresponding to the movement toward (or away from) the net and jumping.

The task is episodic and in order to solve the environment, the agent must get an average score of +0.5(over 100 consecutive episodes, after taking the maximum over both agents). Specifically, After each episode, all the rewards received by each agent are added up (without discounting), to get a score for each agent and then the maximum of these two scores is taken. This yields a single score for each episode.

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

Proposed Approach:
Since the environment requires the training of two separate agents and under some situation, the agents need to collaborate(don't let the ball hit the ground) and will compete(gather more points) with each other in other situations.

We are planning to solve this environment by **Multi-Agent Deep deterministic policy gradient algorithm (MADDPG)**[2] where each agent is modeled as a Deep Deterministic Policy Gradient(DDPG)[3]. A super agent called **MADDPG** will handle the training of two agents. Each one of the agents will have

a. Actor watching the state corresponding to its player and performing an action for it
b. Critic watching the state of both agents, and estimating the action-value function

That is the actor decides which action to take, and the critic tells the actor how good its action was and how it should adjust. We will use different neural network for the Actor and the Critic and will also try with some **Convolutional Layers rather than just using Multi-Layer Perceptron** along with **Batch normalization**[3] to see its impact on learning

The major challenge of learning in continuous action spaces is exploration. The advantage of MADDPG algorithm is that we can treat the problem of exploration independently from the learning algorithm. To solve this environment we are planning to use the **Ornstein-Uhlenbeck Process** discussed in DDPG paper[3] and we will also try the **exploration algorithm taught in class like e-greedy, UCB**, etc and see how our trained agents perform using these algorithms.

The Ornstein Uhlenbeck process adds a certain amount of noise to action values at each time step, this noise is correlated to the previous noise and therefore tends to stay in the same direction for longer durations without canceling itself out. This noise allows the agents to maintain their velocity and explore the action space with more continuity. We have planned to add **different types of noise** to see its effects on learning(like **Normal, Poisson noise**, etc). There are few hyperparameters that determine noise characteristics and magnitude in the Ornstein Uhlenbeck process, we will **tune these parameters manually in order to get the best result and high score**.

To learn from past experiences we have planned to use **Prioritized Experience Replay**. When the agent will be interacting with the environment, its experience will be stored in a replay buffer. These experiences will then be utilized by the Critic which allows the agents to learn from their past experiences. **Prioritized Replay selects experiences based on priority value that is correlated with the error magnitude**. Due to an increase in probability because of the fact that important experience vector is sampled, it might play in good role in better learning.

References:

[1] Tennis Environment
[2] MADDPG Paper
[3] DDPG Paper
[4] Related Work1 Related Work2