Deep Reinforcement Nano Degree Project 3: Collaboration and Competition

Peng Zhao

December 19, 2019

Abstract

This report briefly summarizes the deep reinforcement nano degree project 3, which is training two agent to to control rackets to bounce a ball over a net using Multi Agent Deep Deterministic Policy Gradient (MADDPG) algorithm. The problem is solved after 1412 episodes in this experiment.

1 Multi-Agent Deep Deterministic Policy Gradient Algorithm

The Multi-Agent Deep Deterministic Policy Gradient (DDPG) algorithm is used for two agents to learn the environment. As shown in Figure 1. Each agent has separate actor-critic networks, which is similar to single agent DDPG. However the critic network of each agent is slightly different, which is it has fully visibility on the environment. It not only takes in the observation and action of that particular agent, but also observations and actions of all other agents as well. The critic network is active only during training time, and will be absent in running time.

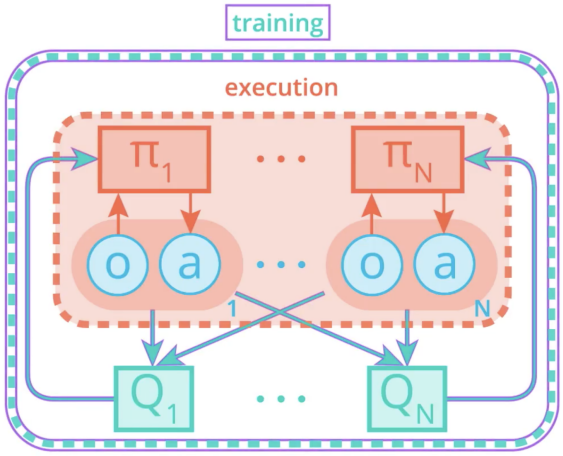

Figure 1: Illustration of MADDPG algorithm

2 Network Architecture

Figure 2 and Figure 3 illustrate the architecture of the actor critic and critic network respectively. The data size propagation for both class of network is: $inputsize -> 256 -> 128 -> outputsize$

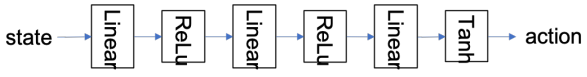

Figure 2: Actor Network Architecture

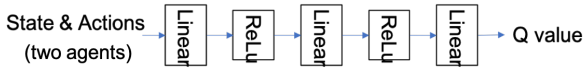

Figure 3: Critic Network Architecture

3 Hyper Parameters

The hyper parameters used in this experiment are shown in Table 1.

Parameter term	**Value**
replay buffer size	int(1e6)
minibatch size	128
learning rate of the actor	1e-3
learning rate of the critic	1e-3
L2 weight decay	0
learning timestep interval	1
number of learning passes	5
discount factor	0.99
for soft update of target parameters	8e-3
Ornstein-Uhlenbeck noise parameter, volatility	0.2
Ornstein-Uhlenbeck noise parameter, speed of mean reversion	0.15
initial value for epsilon in noise decay process in Agent.act()	5.0
episode to end the noise decay process	300
final value for epsilon after decay	0

Table 1: Hyper parameters

4 Result

The environment has been solved in this experiment, which is shown in the following figure. As shown, after an initial training, the agent began to obtain higher scores after about 1300 episodes. The required score (0.5) is finally reached after 1412 episodes.

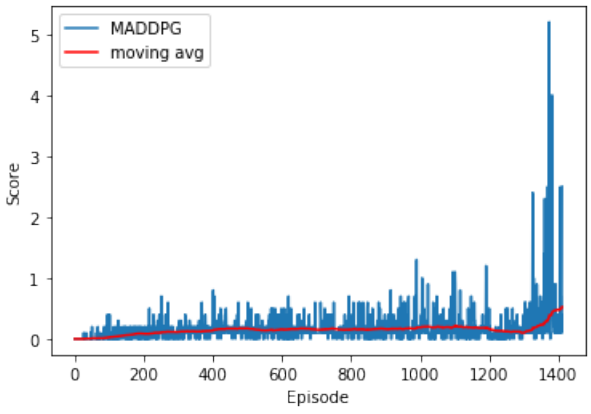

Figure 4: Result

5 Future Work

I observed the required episodes to solve the environment varies with the training run. Even in some experiments, the required score cannot be reached within 2000 episodes. I observed that this situation

happens when the two agents lost their balance in training, which means one agent learned extremely better than the other one. Method that can balance the multi-agent training may reduce and stabilize the required episodes.