< Project 3 – Collaboration and Competition >

1. Learning algorithm – DDPG(Multi Agent)

In this project, we used the method of training two agents separately. The basic algorithm followed DDPG, but it was important to create two agents because the two agents had different state. The network of actor and critical itself is the same, but each agent is trained. The reason for using the DDPG algorithm is that it performs well in selecting actions in a continuous environment. DDPG also selects and evaluates actions through two networks: actor and critical, allowing them to be more sophisticated.

2. Chosen parameters

Hyperparameters	Value
Buffer size	1,000,000
Batch size	128
Discount factor	0.99
Soft update of target parameter	0.001
Learning rate of the actor	0.0001
Learning rate of the critic	0.0001

3. Model architecture for neural network

Actor	Critic
(fc1): Linear(in_features=33, out_features=256, bias=True)	(fc1): Linear(in_features=33, out_features=256, bias=True)
(fc2): Linear(in_features=256, out_features=128, bias=True)	(fc2): Linear(in_features=258, out_features=128, bias=True)
(fc3): Linear(in_features=128, out_features=2, bias=True)	(fc3): Linear(in_features=128, out_features=1, bias=True)

4. Plot of reward per episode

A plot of each Episode

Environment solved in 100 episodes!   Average Score: 1.06
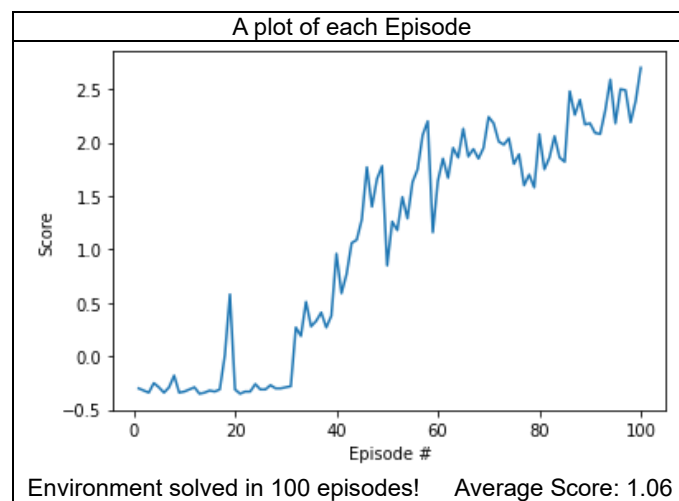
5. Idea for Future Work

- Learn more episodes

- Changing the Network layers

- Using a different learning model