Udacity Deep Reinforcement Learning Nanodegree

Project 3: Collaboration and Competition

Khanh Nguyen Vu

I. Approach

The **Multi-Agent Deep Deterministic Policy Gradient (MADDPG)** algorithm was adopted for this project. I reused some of the components from the first project (Navigation) and the second project since the MADDPG is pretty much similar to DQN algorithm and DDPG (it has a ReplayBuffer and the same networks updating scheme).

Network architecture

Actor network

Observation (24,)
256 nodes (ReLU, batch normalization)
128 nodes (ReLU)
2 nodes (Tanh)

Critic network

All states (24 * n_agents)
256 nodes (ReLU, batch normalization) concat with **All actions (2 * n_agents)**
128 nodes (ReLU)
1 node (Linear)

For the exploration factor, Ornstein–Uhlenbeck process was added to the actions vector at every time-step.

Hyperparameters

Networks:
- Actor optimizer: Adam, learning rate = 0.0001.
- Critic optimizer: Adam, learning rate = 0.0003.
- Soft update: TAU = 0.01.
- Gamma: 0.99

Memory buffer:

- Buffer size: 1000000 (one million).
- Batch size: 128.
- Uniformly sampling.

OUNoise:

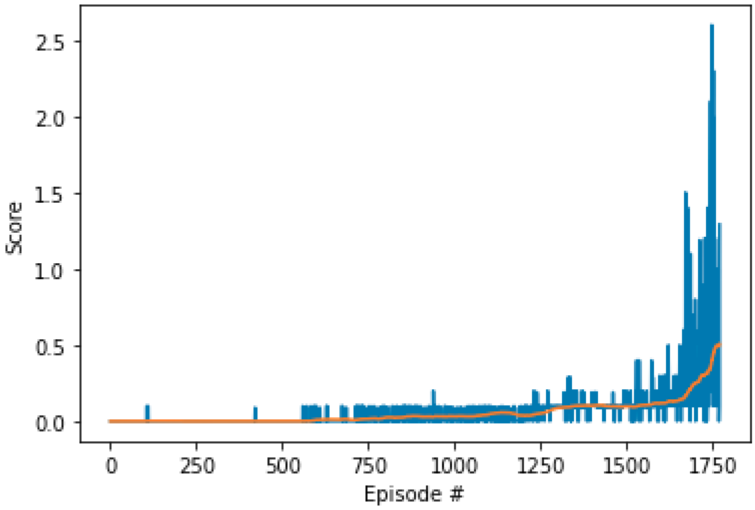
- Mu: 0
- Theta: 0.1
- Sigma: 0.2
- Sigma_min: 0.1
- Sigma_decay: 0.99 (reduce exploration rate as the agent learns)

Training:

- Max episode: 5000.
- Max steps per episode: util termination.

II. Result

```
Episode: 1772, Average score: 0.51
```



III. Ideas for improvements

- Use prioritized experience replays buffer to improve the learning speed.
- Try to adapt PPO algorithm to this multi-agent setting.
- Speed up the process of hyperparams tuning (use Optuna, Hyperopt, etc.)