Algorithm

Similar with the last Reacher project, I use Deep Deterministic Policy Gradient (DDPG) as my learning algorithm for this project. DDPG is a model-free, off-policy algorithm and it's belong to the category of Actor-Critic algorithms. And it has already shown promising performance regarding the problems with continuous action spaces.

Since the environment for this project involving multiple agents, it requires two agents to play with each other. So I made some adjustment to my original DDPG algorithm which had been used for the last project.

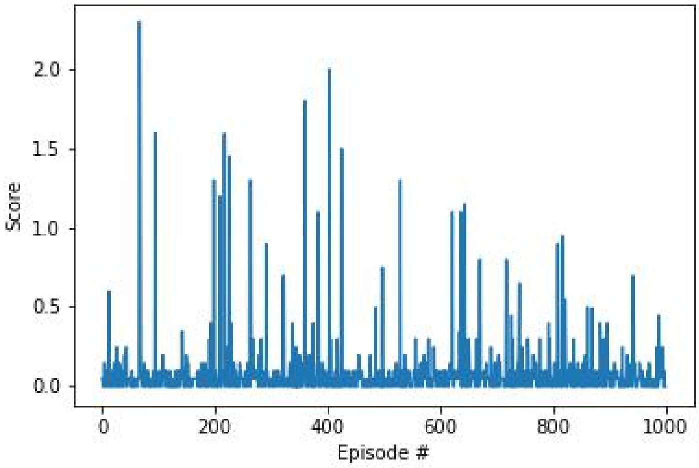
Fist of all, I created tow different DDPG agents, they both contained a actor and critic, so there's two actor and two critic network in my program. when interact with the environment, the two actors with the same network structure act separately and make their own actions every time step. But the crucial design of my algorithm is the shared replay buffer, the two agents shared a same memory , and they update their parameters at the same time. At every time step, the actors add a new experiences to their shared memory respectively, and when it's time to learn, the critics sample batches of experiences from memory to update their parameters. The important thing here is that both agent sample from the memory randomly, without knowing if these experiences coming from itself or it's opponent.

Work and Results

In my work, I implement two ddpg agent with their own actor and critic network. Both the actors and critics are 3-layer MLP neural networks with 256 units at each hidden layer. I also use batch moralization for both the actor and critic.And for the update of local actor and local critic, I use Adam optimizer with learning rate of 10^{-4} . and like my former work for the Reacher project, I also use a Uhlenbeck-Ornstein noise and a soft update strategy with a tau of 10^{-3} . I let the agents to learn every 5 episodes, and when it's time to learn, it will call the learn() function for 10 times in a row. In addition, my discount factor I set is 0.99.

After training, the agent can get +0.5 accumulated reward scores (the average score of the two agents) over 100 episodes. I trained it for over 2000+ episodes and the agent solved the environment after 2055 episodes with a average reward of 0.83.

Then I test the trained agents for the same environment for 1000 episodes, and the corresponding plot of reward scores is shown below.



Plans for Future Work

My current work is mainly about making some adjustment of DDPG to make it applicable to the multiple agent scenario. Next, I want to use try the Multi Agent DDPG approach (MADDPG).