Deep Reinforcement Learning Nanodegree Program

Collaboration and Competition

Atauro Chow

Deep Reinforcement Learning Nanodegree Program

Collaboration and Competition

Introduction.

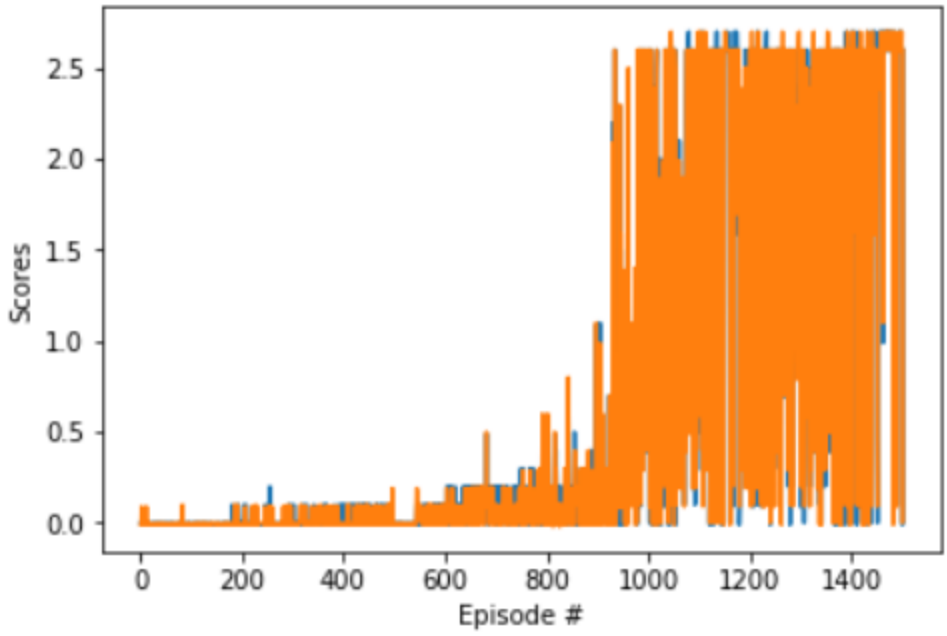
In this Tennis environment, two agents control rackets to bounce a ball over a net. If an agent hits the ball over the net, it receives a reward of +0.1. If an agent lets a ball hit the ground or hits the ball out of bounds, it receives a reward of -0.01. Thus, the goal of each agent is to keep the ball in play. The resolved criteria is the agent must receive an average reward (over 100 episodes) of at least 0.5 scores.

Algorithm

We use deep deterministic policy gradient (DDPG) as our fundamental model. DDPG is widely use in solving the problems with high-dimensional observation spaces. The model applies two neutral network, Actor and Critic. Both actor and critic have two hidden layers with 400 and 300 units respectively. They are full-connected layers. An experience replay is implemented, and its buffer size is 500,000 with mini-batch size is 128. The agent will pick 128 entries from buffer randomly for training. The training update actor and critic networks every 10 steps. The learning rate of actor and critic are 0.001 and 0.0001. Batch Normalization function is applied to second hidden layer of actor.

Results

The average score was 2 at 1500 episodes. For the agent training, the problem was solved (over score 0.5) after 1000 episodes.



Discussions

The following hyperparameters are same as Continuous Control project

- Learning rate of actor and critic
- Number of units of hidden layers for actor and critic.
- Discount factor (Gamma): 0.99
- Soft update of target parameter: 0.001
- Experience Replay buffer size: 500,000
- Mini-batch size: 128

The result proved that hyperparameters used can be applied to different continuous space problem.

Batch normalization (BN)

Batch normalization played a key role because this environment is symmetric. Without applied BN, the training is in extreme slow. The reason was there are several different distributions but can result as good behavior. It caused agent to learn slowly. After applied, the learning rate of agent was speed up and solved in 1000 episodes. BN made distribution was same for each of minibatch. As a result, it minimized the variety of distribution.

Idea for future work

We believed the following implementation can improve the current result.

- Prioritized Experience Replay: Currently, the agent picked records from replay buffer randomly. It would be benefit if good behavior will be picked in more frequently.
- Proximal Policy Optimization (PPO): The paper of Proximal Policy Optimization Algorithm proven that have better learning rate than A2C.
- Inverse Model: The paper of Using Deep Reinforcement Learning for the Continuous Control of Robotic Arms show that inverse model can achieve mean score 98.2% while DDPG baseline achieve 49.4% only.

References

1. CONTINUOUS CONTROL WITH DEEP REINFORCEMENT LEARNING (https://arxiv.org/pdf/1810.06746.pdf)
2. Sample code from Bipedal project of Deep Reinforcement Learning nanodegree program.