Deep Reinforcement Learning Nanodegree Program

Continuous Control

Atauro Chow

Deep Reinforcement Learning Nanodegree Program

Continuous Control

Introduction.

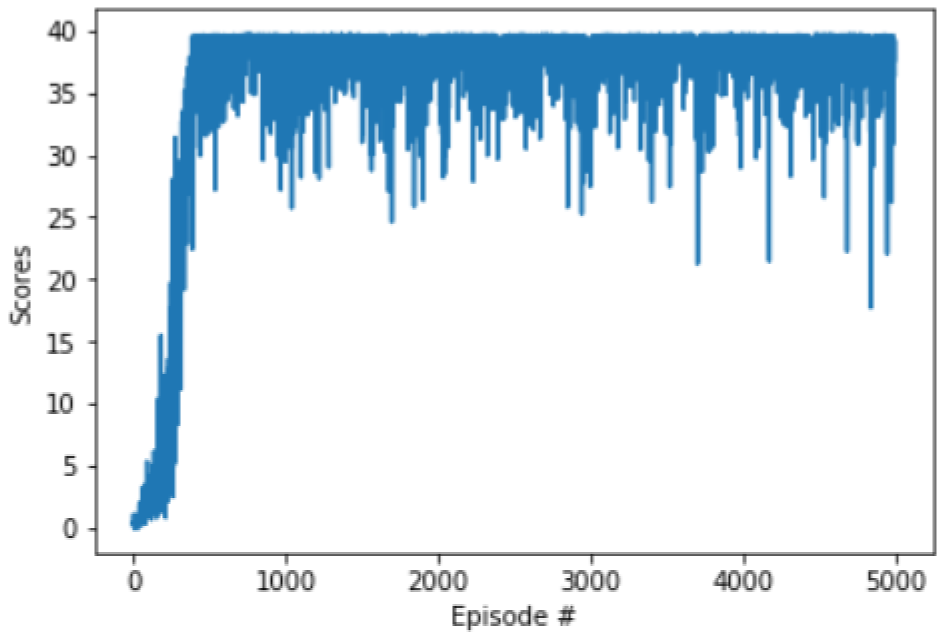
The goal of this project is to train an agent can maintain its position at the target location for as many time steps as possible. In this environment, a double-jointed arm can move to target locations. A reward of +0.1 is provided for each step that the agent's hand is in the goal location. The resolved criteria is the agent must receive an average reward (over 100 episodes) of at least 30 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 36.6. For the agent training, the problem was solved (over score 30) after 500 episodes. The agent is quite stable. It maintain average score between 36 and 39 until at the end of episodes.



Discussions

The following hyperparameters are used based on the paper of Continuous Control with Deep Reinforcement Learning.

- 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

The following hyperparameters are used based on experiments.

- Experience Replay buffer size
- Mini-batch size

Experience Replay

In my environment, single agent set to run 5000 episodes. Each episode has 1000 steps. Therefore maximum 5,000,000 entries can be stored in buffer totally. We found that 500,000 entries is an optimized size. More than or less than this numbers, the learning is not efficiency. It is because the agent may pick not-good-enough records for learning when the buffer size is too large. Also, when the buffer size is too small, the good records may fade out quickly and cannot be used by agent.

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 500 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.