Project 2 Continuous Control Report

Alvin (Cheuk Hin) Li

April 2020

1 Introduction

The Reacher environment was used in this project. The goal is to use a single double-jointed arm to move and reach certain locations. +0.1 is rewarded every step the arm successfully reaches the target. The observational space consists of 33 dimensions, including position, velocity, etc.. The agent must achieve an average reward of +30 over 100 episodes, in order for it to be considered solved.

2 Learning Algorithm

Deep Deterministic Policy Gradient is used to combat the problem of continuous spaces as seen in this project. Key Features:

1. Actor-Critic

   The Actor (policy) is defined by $\mu(s|\theta^{\mu})$. The Critic is defined by Q(s,a), similar to Q-learning.
   For this specific case, the Actor network has 4 Dense layers with size 33, 512, 256 and 128 respectively. The last layer is put through a tanh activation for continuous output space.
   The Critic network has 3 Dense layers with size 33, 400, 300 respectively.

2. Soft Update

   Since directly implementing Q learning with neural networks is unstable, soft update is used to slowly merge the learned networks back into the target networks. This is achieved by updating the weights with the following equation:

   $$\theta' \leftarrow \tau\theta + (1-\tau)\theta'$$

   where τ is a very small number.

3. Replay Buffer

   The buffer is used for sampling. Once the cache is full, the old samples are discarded.

4. Noise

   Noise is added to facilitate the exploration of the continuous space. The Ornstein-Uhlenbeck process is used to generate noise and add to the actor policy.

Batch normalization was recommended to improve training speed but was found to decrease scores. Thus it was not used.

The following hyperparameters were used:

- BUFFER SIZE: $1e6$
- BATCH SIZE: 128
- GAMMA: 0.99
- TAU: $1e-3$
- LR ACTOR: $1e-3$
- LR CRITIC: $1e-3$
- WEIGHT DECAY: 0
- UPDATE EVERY: 20
- of UPDATES per UPDATE = 10

3 Plot of Rewards

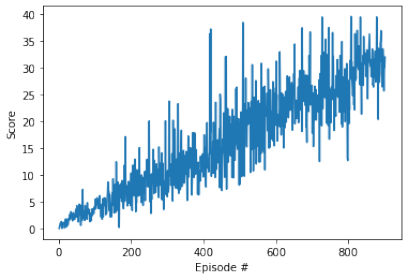

Figure 1: Rewards per Episode

The code solved the environment within 902 episodes.

4 Ideas for Future Work

In the future, prioritized experience replay can be used in conjunction with DDPG as suggested by Hou et. al. [2]. We can also consider using a bootstrapped version of DDPG as suggested by Zheng et. al. [3].

5 References

[1] https://arxiv.org/pdf/1509.02971.pdf
[2] https://www.researchgate.net/publication/332859622$_{I}mproving_{D}DPG_{v}ia_{P}rioritized_{E}xperience_{R}eplay_{R}L_{c}ours$
[3]$https://www.ijcai.org/Proceedings/2018/0444.pdf$