Report for the Continuous Control Project

1. Introduction

This report introduces the algorithm used for this continuous control project. Detailed algorithm is shown in the next session. Results are shown and discussed. Future improvement ideas are discussed at the end of the report.

2. Approach

Deep Deterministic Policy Gradient (DDPG) [1] is used in this continuous control problem. It is an actor-critic based method. There are mainly two parts in this algorithm: the critic and the actor. The policy that the robot arm follows is parametrized by a neural network which is the actor. The critic is used to guide the update of the actor by estimating an action-value function which is also parametrized by a neural network. The update of the actor uses the results derived in [2] which is the Deterministic Policy Gradient (DPG) method. The update of the critic mainly follows the idea in [3], using target network and experience replay to stabilize the training.
The detail algorithm is as followed.

```
------------------------------------------------------------------------------------------------
Initialize the critic network $Q(s, a|\theta^Q)$ with random parameters $\theta^Q$
Initialize the target critic network $Q'(s, a|\theta^{Q'})$ with $\theta^{Q'} = \theta^Q$
Initialize the actor network $\mu(s|\theta^\mu)$ with random parameters $\theta^\mu$
Initialize the target actor network $\mu'(s|\theta^{\mu'})$ with $\theta^{\mu'} = \theta^\mu$
Initialize the replay memory $D$
For episode = 1, M do
    Get initial state $s_1$ from environment
    For t = 1, T do
        Select action $a_t = \mu(s_t|\theta^\mu)$ according to the current policy
        Execute action $a_t$ on the environment and observe next state $s_{t+1}$ and reward $r_t$
        Store the transition $(s_t, a_t, r_t, s_{t+1})$ in $D$
        Sample random minibatch (size $N$) of transitions $(s_i, a_i, r_i, s_{i+1})$ from D.
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
        Update critic network $Q(s, a|\theta^Q)$ by minimizing the loss: $L = \frac{1}{N}\sum_{i=1}^{N}(y_i - Q(s_i, a_i|\theta^Q))^2$
        Update the actor network $\mu(s|\theta^\mu)$ using the sampled policy gradient:
            $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$
        Update target networks:
            $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$
            $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
    End For
End For
------------------------------------------------------------------------------------------------
```

3. Results

In this work, I use the same structure for the actor network and the critic network. There are 2 hidden layers. The first hidden layer has 256 units and the second hidden layer has 128 units. The activation function used in the hidden layers are the Rectified Linear Unit (ReLU). The activation function for output layer of the actor network is tanh which will limit the output in the range of (-1,1). There is no activation function for the output unit of the critic network. No L1, L2 regularization is used. Also, techniques like dropout and batch normalization are also not used.

Same optimizer and learning rate are used for the actor and critic network due to simplicity. The optimizer used is Adam. Learning rate is 0.00015. Batchsize is set as 64 and the capacity of the replay buffer is set as 100,000. The hyperparameters described here are not tuned in a systematic way due to computation power.

Better performance can be guaranteed by tuning these parameters as well as the structure of the neural network. The learning curve of running 500 episodes is shown in Fig. 1. It can be seen that the score is unstable and crashed at the end of the training.

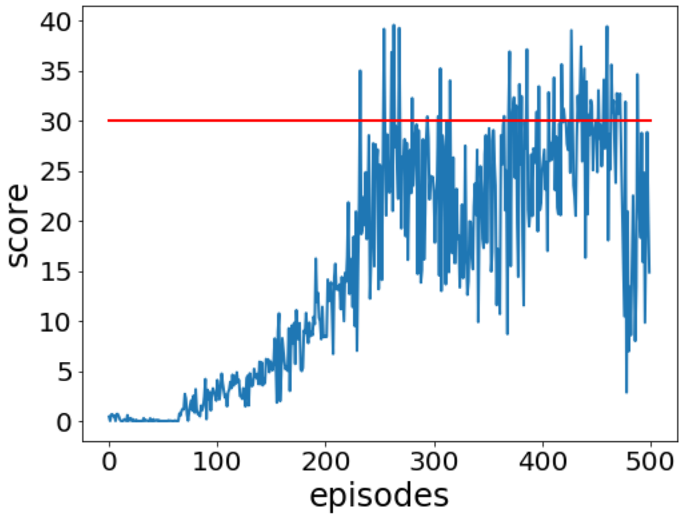

Fig. 1. Learning curve with learning rate 0.00015

Gradient clipping is used when training the critic network to stabilize the training and the results are shown in Fig. 2 (left). It can be shown that the performance is improved. Further, decreasing the learning rate from 0.00015 to 0.00010 makes the learning process smoother (right). An average of 30+ score over 100 episodes can be got from the results of Fig.2 using about less than 300 episodes.

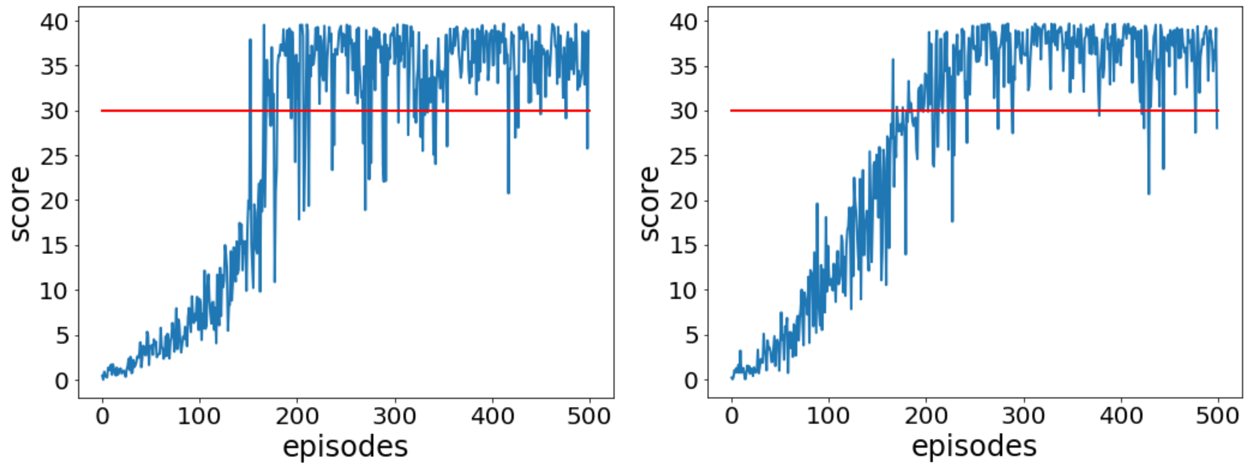

Fig. 2. Learning curve with learning rate 0.00015 (left) and 0.00010 (right) with gradient clipping for critic network

4. Future work

First, the hyperparameters in the algorithm can be tuned and they can be different for the critic and actor network. Second, other methods like TRPO [4] and PPO [5] can be tried. Also, recent work related to the DDPG like D4PG should be investigated.

Reference

[1]. Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971*(2015).

[2]. Silver, David, et al. "Deterministic policy gradient algorithms." *ICML*. 2014.

[3]. Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves et al. "Human-level control through deep reinforcement learning." *Nature* 518, no. 7540 (2015): 529.

[4]. Schulman, John, et al. "Trust region policy optimization." *International Conference on Machine Learning*. 2015.
[5]. Schulman, John, et al. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347* (2017).
[6]. Barth-Maron, Gabriel, et al. "Distributed Distributional Deterministic Policy Gradients." *arXiv preprint arXiv:1804.08617* (2018).