Reinforcement Learning Assignment 5

Yiheng Jiang (TA), Prof. Zou

2024-04-27

1 Introduction

Actor-Critic method reduces the variance in Monte-Carlo policy gradient by directly estimating the action-value function. The goal of this assignment is to do experiment with two improved AC methods – Asynchronous Advantage Actor-Critic (A3C) and Deep Deterministic Policy Gradient (DDPG). Due to the inherent advantage of PG method, AC method is able to tackle the environment with continuous action space. However, a naive application of AC method with neural network approximation is unstable for challenging problem. In this assignment, you're required to train the agent with continuous action space and have some fun in some classical RL continuous control scenarios.

2 Actor-Critic Algorithm

In original policy gradient $\bigtriangledown_\theta log \pi_\theta(s_t, a_t) v_t$, return v_t is the unbiased estimation of expected long-term value $Q^{\pi_\theta}(s, a)$ following a policy $\pi_\theta(s)$ (Actor). However, original policy gradient suffers from high variance. Actor-Critic Algorithm uses Q value function $Q_w(s, a)$, named Critic, to estimate $Q^{\pi_\theta}(s, a)$.
DDPG borrows the success from DQN by introducing experience replay and target Q network into deterministic policy gradient method (DPG). DPG deterministically maps states to specific action instead of outputting a stochastic policy over all actions.
In A3C, we maintain several instances of local agent and a global agent. Instead of experience replay, we asynchronously execute all the local agents in parallel. The parameter of the global agent is updated by all the local experience.
The pseudo code is listed at the end of the article. For more details of these two algorithms, you can refer to the original papers in the following.

- Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. 2016: 1928-1937.
- Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.

3 Experiment Description

- Programming language: python3
- You are required to implement A3C and DDPG algorithms.
- You should test your agent in a classical RL control environment–Pendulum. OPENAI gym provides this environment, which is implemented with python (`https://www.gymlibrary.dev/environments/classic_control/pendulum/`). You can also refer to gym's source code at (`https://github.com/openai/gym`).

4 Report and Submission

- Your report and source code should be compressed and named after “studentID+name+assignment5”.
- The file should be submitted on Canvas on May 11, 2024.