Performance Comparison and Analysis Between Q-Learning, A2C, A2C with Generalized Advantage Estimation, and PPO in BipedalWalker-v2

Minh Nguyen

Problem Statement / Abstract:

In the beginning of our class, students were introduced to one of the long-standing challenges of reinforcement learning (RL) which is learning to control agents directly from high-dimensional sensory input such as vision in autonomous driving cars. Several methods had been introduced in classes with the latest lesson being Q-Learning. However, policy gradients methods which try to optimize the policy function directly, in contrast with Q-Learning where the policy manifests itself as maximizing the value function [6]. Influenced by the performance of Q-Learning with a convolutional neural network in Atari game, OpenAI provided an open-source Gym to allow researchers to compare the performance of RL algorithms with benchmark problems [2][9]. In this project, I will experience Q-Learning and two online policy gradient algorithms, which learn the optimal value function V and policy π, to solve for the continuous BipedalWalker-v2 [3][8]. In addition, I will also experience A2C with Generalized Advantage Estimation (GAE) [4]. Later, I will compare and explain the performance for the three algorithms.

Experiment:

I will first explain the three policy gradient algorithms Q-Learning, A2C, A2C with GAE, and PPO in mathematical terms. Then, I introduce the BipedalWalker-v2, which is a simple, but bug-free continuous Gym environment to test the performance of these three algorithms. Moreover, I will implement the three algorithms from scratch in Python to work on the BipedalWalker-v2. I will then fine tune the parameters of each algorithm to get better performance. Furthermore, I will compare the all three algorithms with different tuning parameters in terms of "Episode Length over Time", "Episode Reward over Time", and "Episode per Time Step". Besides the result comparison mentioned above, I will also compare my results with other top algorithms in the OpenAI Gym Leaderboard [5]. Lastly, I will explain the best performing algorithms in the BipedalWalker-v2 continuous environment.

Results:

I expect that the convergence of PPO will be faster than other policy gradient algorithms in the BipedalWalker-v2 environment.

Additional Note:

The decision to choose this project is influenced by "Helen (Mengxi) Ji Blog", "Alexander Van de Kleut Blog", and partially my experience with RL since A2C and PPO are the regularly-used algorithms in the industry [1][7]. This class has provided me with a strong foundation in RL, thus I am ready to deepen my knowledge in these two algorithms to benefit my future research and job.

Reference:

[1] Alexander Van de Kleut, "Beyond vanilla policy gradients: Natural policy gradients, trust region policy optimization (TRPO) and Proximal Policy Optimization (PPO)," Alexander Van de Kleut, 18-Jul-2020. [Online]. Available: https://avandekleut.github.io/ppo/. [Accessed: 28-Oct-2021].

[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai Gym," arXiv.org, 05-Jun-2016. [Online]. Available: https://arxiv.org/abs/1606.01540. [Accessed: 28-Oct-2021].

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv.org, 28-Aug-2017. [Online]. Available: https://arxiv.org/abs/1707.06347. [Accessed: 28-Oct-2021].

[4] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation,” arXiv.org, 20-Oct-2018. [Online]. Available: https://arxiv.org/abs/1506.02438. [Accessed: 28-Oct-2021].

[5] Leaderboard - openai/gym wiki. [Online]. Available: https://github-wiki-see.page/m/openai/gym/wiki/Leaderboard. [Accessed: 28-Oct-2021].

[6] “Papers with code - an overview of policy gradient methods,” An Overview of Policy Gradient Methods | Papers With Code. [Online]. Available: https://paperswithcode.com/methods/category/policy-gradient-methods. [Accessed: 28-Oct-2021].

[7] “Proximal policy optimization,” Helen(Mengxin) Ji, 15-Apr-2019. [Online]. Available: https://mengxinji.github.io/Blog/2019-04-15/Proximal-Policy-Optimization/. [Accessed: 28-Oct-2021].

[8] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” arXiv.org, 16-Jun-2016. [Online]. Available: https://arxiv.org/abs/1602.01783v2. [Accessed: 28-Oct-2021].

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing Atari with deep reinforcement learning,” arXiv.org, 19-Dec-2013. [Online]. Available: https://arxiv.org/abs/1312.5602. [Accessed: 28-Oct-2021].