Building a TD3 Agent for Continuous Control

CS 461 Artificial Intelligence
Fall 2024
Department of Computer Engineering, Bilkent University

Release Date: 10.12.2024
Due Date: 05.01.2025

Introduction

In this project, you will be implementing and evaluating the Twin Delayed Deep Deterministic policy gradient (TD3) algorithm, a state-of-the-art algorithm for solving continuous action space problems in reinforcement learning.

Background

TD3 is an algorithm that builds upon the DDPG (Deep Deterministic Policy Gradient) framework by addressing its overestimation bias. TD3 does this by using a pair of critics to reduce value overestimation and delaying policy updates to reduce per-update error.
For a detailed explanation of the TD3 algorithm, please refer to the original paper by Scott Fujimoto, Herke van Hoof, and David Meger: TD3: Twin Delayed DDPG.

Task

Your task is to implement the TD3 algorithm and test your agent in the Lunar Lander Continuous environment provided by Gymnasium. You will run your experiments using six different random seeds and average the results.
For more information on the Lunar Lander Continuous environment, please refer to the following link: Lunar Lander Continuous Environment Documentation, check the Arguments section.

Why Multiple Random Seeds?

Using multiple random seeds for the initial conditions and the environment's stochastic elements allows us to assess the robustness and reliability of the algorithm. Averaging the results helps in mitigating the effects of outlier performances due to particularly fortunate or unfortunate initializations, leading to more generalizable performance metrics.

Requirements

- It is highly recommended to use a Linux or macOS system to ensure full compatibility and to be able to take advantage of all Gym features, including rendering. If you are unable to access a Linux or macOS machine, please ensure that your code can run without rendering on Windows.
- Your implementation should be done in Python 3.x.

- You are allowed to use machine learning libraries such as PyTorch or TensorFlow for the neural network components of TD3.
- Make sure to disable rendering during training as it significantly increases the runtime.
- Document your code appropriately and include instructions on how to run your implementation.

Evaluation

Your implementation will be evaluated based on the following criteria:

- **Correctness of Implementation:** Your code should correctly implement the TD3 algorithm as described in the provided resources.
- **Code Clarity and Documentation:** Your code should be well-organized and properly documented, making it easy to understand your implementation approach.
- **Performance:** Your agent should achieve a minimum average reward of **200 per episode** in the Lunar Lander Continuous environment to be considered successful. This performance should be consistent across multiple random seeds to demonstrate the robustness of your solution.

Please be aware that reaching this level of performance is essential for the assessment phase of this project. The average reward is to be calculated over a minimum of 100 consecutive episodes to ensure statistical significance. It is not sufficient to reach this threshold in a single episode or to average this performance over the initial phase of learning; your agent must demonstrate the ability to sustain this level of performance.

Reporting Your Results

Submit your code files along with a report detailing your implementation and findings. Ensure your report includes:

1. A brief explanation of the TD3 algorithm.
2. A summary of your implementation.
3. Detailed graphs or tables that show the reward per episode over time for each of the random seeds tested.
4. Provide an analysis of the results, discussing the stability and reliability of the learned policy.
5. Discuss any variations observed between different seeds and attempt to explain why these may have occurred.