ECE433/COS435 Introduction to RL
Assignment 5: Value Iteration and Policy Gradient
Spring 2024

Fill me in

Your name here.

Due March 25, 2024

Collaborators

Fill me in

Please fill in the names and NetIDs of your collaborators in this section.

Instructions

Writeups should be typesetted in Latex and submitted as PDFs. You can work with whatever tool you like for the code, but **please submit the asked-for snippet and answer in the solutions box as part of your writeup. We will only be grading your write-up.** Make sure still also to attach your notebook/code with your submission.

We recommend you collaborate in pairs if you find the workload of this assignment overwhelming.

Question 1. Implementation of Advantage Actor-Critic (A2C)

For details of A2C, please refer to the accompanying .ipynb file. We also recommend you to read the introduction of [Lil+19] for better insight into its motivation.

Question 1.a

Paste the entire cell implementing the `Actor` and `Critic` classes below.

Solution

Your solution here...

Question 1.b

Paste the entire cell implementing the `compute_returns()` method below.

Solution

Your solution here...

Question 1.c

Paste the code for the entire `training()` method below.

Solution

Your solution here...

Question 1.d

Train your A2C agent on the Gym environment `CartPole-v1`. Insert your plot of the cumulative reward curve in the training process below.

Solution

Your solution here...

Question 2. Implementing DDPG and TD3

In the question, we introduce a useful actor-critic type algorithm: the Deep Deterministic Policy Gradient (DDPG). For a detailed introduction, please refer to the accompanying .ipynb file.

Question 2.a

Paste the entire cell implementing the `Actor_DDPG` and `Critic_DDPG` classes below.

Solution

Your solution here...

Question 2.b

Paste the entire code of `DDPG.train` below.

Solution

Your solution here...

Question 2.c

Insert the reward v.s. episode curve below. (Hint: The optimal reward for `MountainCarContinuous` is around 100.) (Hint2: If the reward is stuck at some local minimum throughout the training process, e.g. $r = -0.1$, try restarting the training process. Being stuck in a local minimum can be caused by an insufficient starting exploration.)

Solution

Your solution here...

Question 2.d

Try out DDPG on some new tasks (apart from `MountainCarContinous`). Please indicate the name of this environment (if you define it yourself, please specify its reward and transition probability), and plot the reward curve against the training episodes below. Does it work out?

Solution

Your solution here...

Question 3. Implementation of TD3 [Optional]

In this question, we introduce a variant of DDPG: the Twin Delayed DDPG (TD3). For a detailed introduction, please refer to the accompanying .ipynb file. For a better understanding of how TD3 reduce the overestimation bias and the variance, we refer you to read [FHM18, Chapt. 4 and 5]

Question 3.a

Paste the code for `Actor_TD3` and `Critic_TD3` networks below.

Solution

Your solution here...

Question 3.b

Paste the entire code of `TD3.train` below.

Solution

Your solution here...

Question 3.c

Insert the reward v.s. episode curve below. Do both algorithms (DDPG, TD3) converge to the same cumulative reward? (Hint: The optimal reward for `MountainCarContinuous` is around 100.)

Solution

Your solution here...

References

[FHM18] Scott Fujimoto, Herke Hoof, and David Meger. "Addressing function approximation error in actor-critic methods". In: *International conference on machine learning*. PMLR. 2018, pp. 1587–1596.

[Lil+19] Timothy P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2019. arXiv: `1509.02971 [cs.LG]`.