Project work

Jari Mattila - 35260T
ELEC-E8125 - Reinforcement Learning

November 28, 2021

Introduction

To cite works, put them in the template.bib file and use [1].

Describe the chosen method

To embed code snippets in the report, you can use the `pycode` environment.

Part I

Answers to the questions in part 1.

Algorithm 1: Twin Delayed Deep Deterministic Policy Gradient (TD3)

Question 1

What are the problems with the deep deterministic policy gradient (DDPG) algorithm? How does TD3 solve these problems?

Question 2

For policy gradient methods seen in Exercise 5, we update the agent using only on-policy data, while in TD3 we can use off-policy data. Why is this the case?

Question 3

Finish the implementation of the TD3 algorithm and train the agent with both InvertedPendulumBulletEnvv0 and HalfCheetahBulletEnv-v0 environments.

Train your agents with three random seeds for both environments.

Include the training plots in the project report and attach the agents' weights to your submission, with filenames ending with td3.pth containing the run ID (1, 2, and 3, each with a different random seed) and the environment name.

Training performance of TD3 algorithm in InvertedPendulumBullet environment is presented in Figure. 1.

Training performance of TD3 algorithm in HalfCheetahBullet environment is presented in Figure. 2.

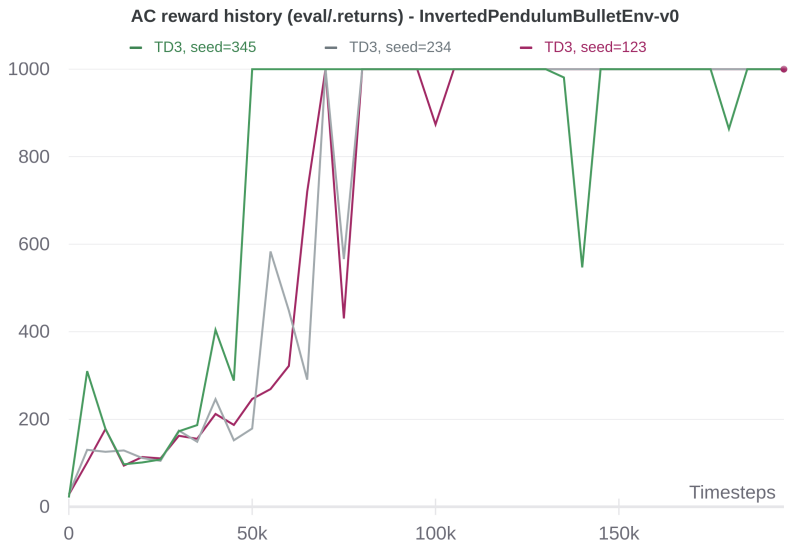

Figure 1: Training performance of TD3 algorithm in InvertedPendulumBullet environment.

Figure 2: Training performance of TD3 algorithm in HalfCheetahBullet environment.

Question 4

Now let's analyze the sensitivity of TD3 to hyperparameters. Choose one hyperparameter, e.g., target action noise, exploration noise, or policy update frequency, that you think heavily influence the training and explain why. Then, train your agent with the modified hyperparameter on both InvertedPendulumBulletEnv-v0 and HalfCheetahBulletEnv-v0 environments (3 random seeds). Show the training plots and submit the trained model of HalfCheetah with name "td3_q4.pth".

Question 5

After playing with the TD3 algorithm, could you find any aspect that could be improved? Please list three of them. Also, please propose a potential solution to one of the problems you listed. You can answer this question by providing a paper link and explaining in your own words how the proposed approach solves/mitigates the problem.

Algorithm 2: Proximal Policy Optimization Algorithms (PPO)

Question 1

Why does clipping the $\frac{\pi_\theta(a|s)}{\pi_{old}(a|s)}$ ratio stabilize the training? What is the relationship between TRPO [7] and PPO?

Question 2

Please finish the implementation of the PPO algorithm.

Similar to Question 3, train the agent on both the InvertedPendulumBulletEnv-v0 and the HalfCheetahBulletEnv-v0 environments.

Train your agents with three random seeds for both environments.

Include the training plots in the project report and attach the agents' weights to your submission, with filenames ending with ppo.pth containing the run ID (1, 2, and 3, each with a different random seed) and the environment name.

For the training curve, you can reference 2.

Training performance of PPO algorithm in InvertedPendulumBullet environment is presented in Figure. 3.

Training performance of PPO algorithm in HalfCheetahBullet environment is presented in Figure. 4.

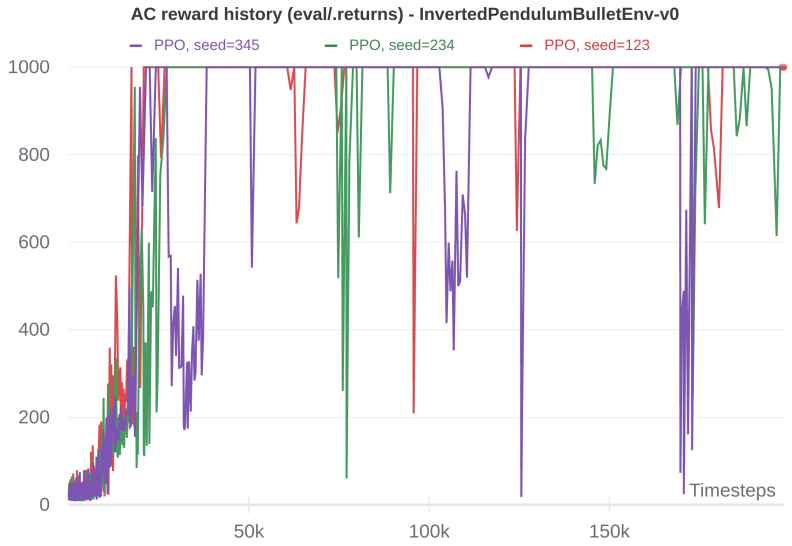

Figure 3: Training performance of PPO algorithm in InvertedPendulumBullet environment.

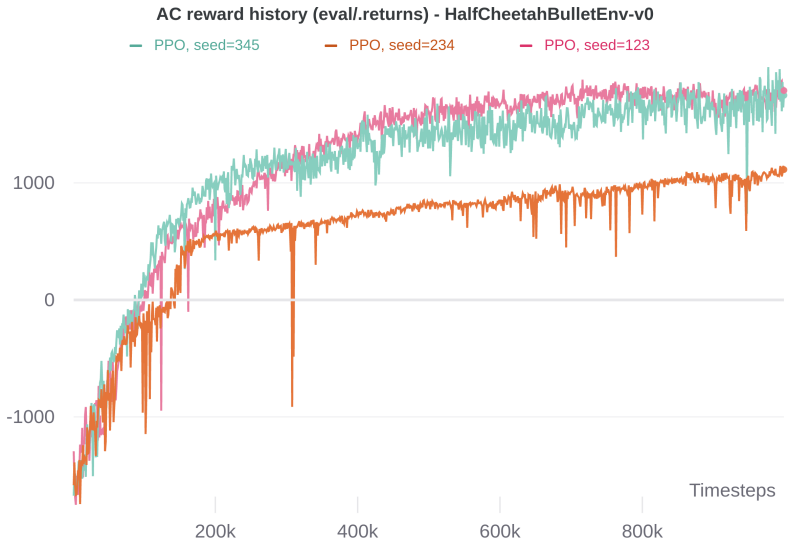

Figure 4: Training performance of PPO algorithm in HalfCheetahBullet environment.

Question 3

In PPO, the target value is calculated by generalized advantage estimation (GAE) [8], as shown in the second equation. Explain the relationship between n-step advantage and GAE. Why is GAE better than n-step advantage?

Part 2

Answers to the questions in part 2.

Question 1

Please correctly implement your algorithm and show the training plots against the TD3/PPO with three random seeds.

The plots should include both InvertedPendulum and HalfCheetah environments.

Training performance of SAC algorithm in InvertedPendulumBullet environment is presented in Figure. 5.

Training performance of SAC algorithm in HalfCheetahBullet environment is presented in Figure. 6.

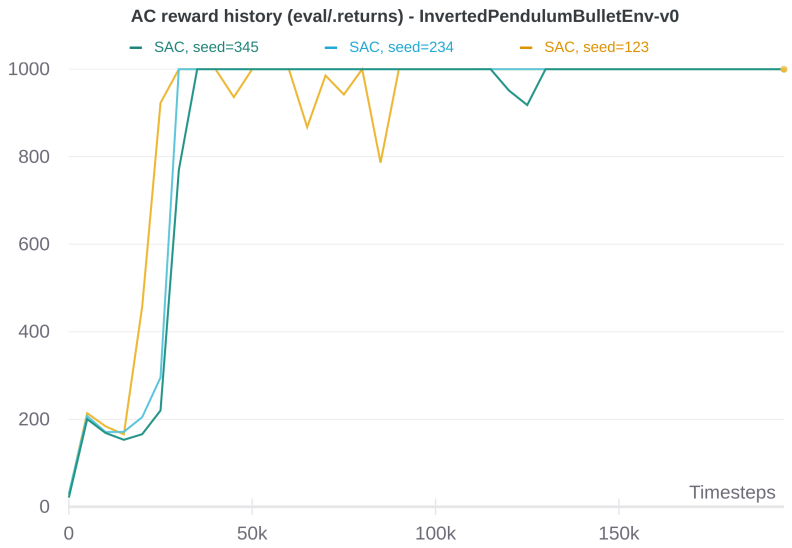

Figure 5: Training performance of SAC algorithm in InvertedPendulumBullet environment.

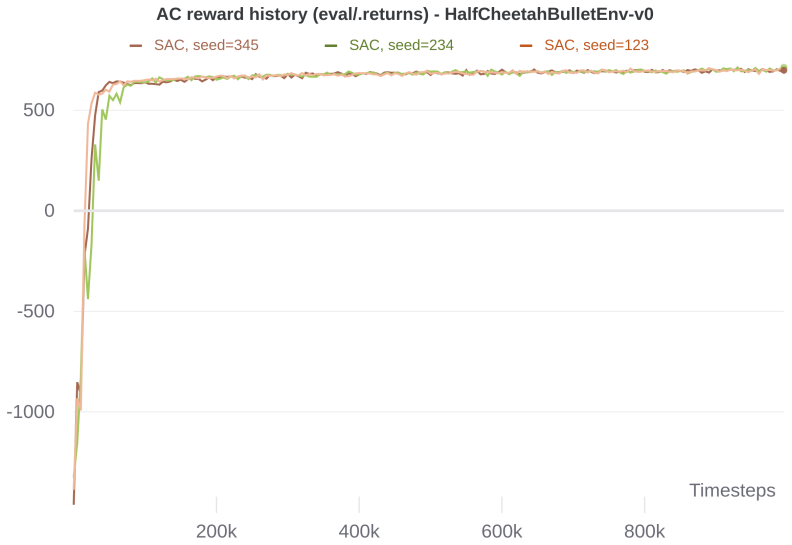

Figure 6: Training performance of SAC algorithm in HalfCheetahBullet environment.

Also, you need to describe the network structure and training procedure as well as hyperparemeters. A clear way to show the hyperparameters is using a table.

Question 2

Let's analysis your algorithm by performing an ablation study. You could modify one design option that you expect to influence the training performance.

Then train the agent using the modified code and compare the results to the original algorithm. Training your agent on HalfCheetah environment is enough but with three random seeds.

Hyperparameters

qwmuq9wu9q8wuz98q

Table 1. SAC Hyperparameters

Parameter	Value
optimizer	Adam(Kingma & Ba, 2015)
learning rate	$3 \cdot 10^{-4}$
discount (γ)	0.99
reply buffer size (γ)	10^6
number of hidden layers (γ)	2
number of hidden units per layer (γ)	256
number of samples per minibatch (γ)	256
nonlinearity	ReLU

Table 2. SAC Environment Specific Parameters

Environment	State Dimensions	Action Dimensions	Reward Scale
InvertedPendulumBullet-v0	5	1	1
HalfCheetah-v0	26	6	5

Conclusions

oekdwpodkwpod

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. Adaptive computation and machine learning series, Cambridge, Massachusetts: The MIT Press, second edition ed., 2018.

[2] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *arXiv:1802.09477 [cs, stat]*, Oct. 2018. arXiv: 1802.09477.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347 [cs]*, Aug. 2017. arXiv: 1707.06347.

[4] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *arXiv:1801.01290 [cs, stat]*, Aug. 2018. arXiv: 1801.01290.