Exercise 5

Jari Mattila - 35260T
ELEC-E8125 - Reinforcement Learning

October 30, 2021

1 Task 1

Implement policy gradient for the CartPole environment with continuous action space. Use agent.py for implementing the reinforcement learning algorithm itself (for example, the agent and policy classes). Use these classes to implement the main training loop in the cartpole.py file, similarly to how it was done in Exercise 1.

Use constant variance $\sigma^2 = 25$ for the output action distribution throughout the training.

(a) basic REINFORCE without baseline,

The training performance plots for each of the tasks (Task 1a, b and c, Task 2 - for different sigma values)

sgdhdjyfjghjfgjfghjf

(b) REINFORCE with a constant baseline $b = 20$,

The training performance plots for each of the tasks (Task 1a, b and c, Task 2 - for different sigma values)

fgfhdhjfhjfgh

(c) REINFORCE with discounted rewards normalized to zero mean and unit variance

The training performance plots for each of the tasks (Task 1a, b and c, Task 2 - for different sigma values)

rgdthdhgfdghdfhggf

using Figure. **??**

Question 1.1

How would you choose a good value for the baseline? textbfJustify your answer.

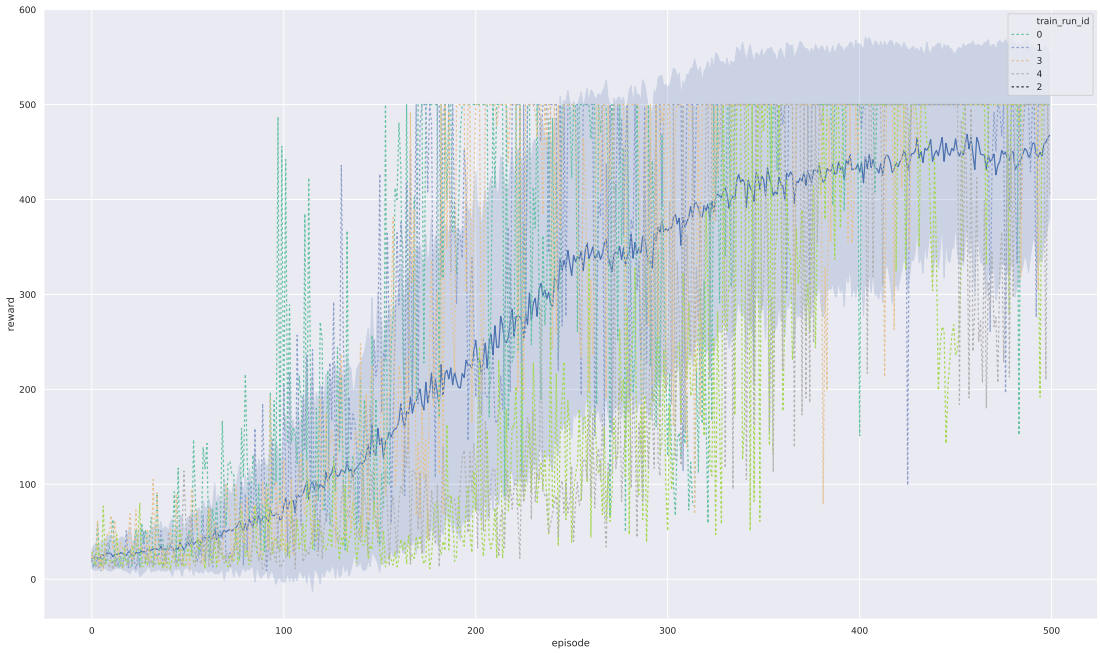

Figure 1: This is a sample figure.

Question 1.2

How does the baseline affect the training, and textbfwhy?

2 Task 2

Implement two cases of adjusting variance during training: exponentially decaying variance $\sigma^2 = \sigma_0^2 \cdot \mathrm{e}^{-c \cdot k}$ (where $c = 5 \cdot 10^4$ and k is the number of episode) and variance learned as a parameter of the network (in both cases set the initial value σ_0^2 to 100). Use REINFORCE with normalized discounted returns for this task. The expected results for the learned variance case are shown in Figure 1.

The training performance plots for each of the tasks (Task 1a, b and c, Task 2 - for different sigma values)

Question 3.1

Compare using a constant variance, as in Task 1, to using exponentially decaying variance and to learning variance during training. **Please explain** what the strong and weak sides of each of those approaches are.

Question 3.2

In case of learned variance, what's the impact of initialization on the training performance? **Please explain.**

Question 4.1

Could the method implemented in this exercise be **directly** used with experience replay? **Why/why not?**

Question 4.2

Which steps of the algorithm would be problematic to perform with experience replay, if any? **Explain your answer.**

Question 5.1

What could go wrong when a model with an unbounded continuous action space and a reward function like the one used here (+1 for survival) were to be used with a physical system?

Question 5.2

How could the problems appearing in Question 5.1 be mitigated without putting a hard limit on the actions? **Explain your answer.**

Question 6

Can policy gradient methods be used with discrete action spaces? Why/why not? Which steps of the algorithm would be problematic to perform, if any? **Explain your answer.**