Exercise 5

Marco Di Francesco - 100632815
ELEC-E8125 - Reinforcement Learning

October 25, 2022

Task 1

a) Look at 1

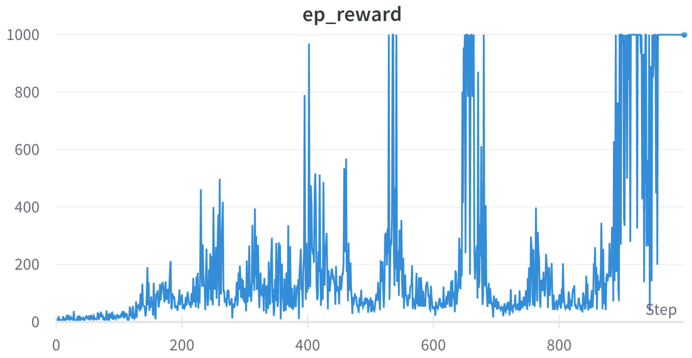

Figure 1: Exercise 1a

b) Look at 2

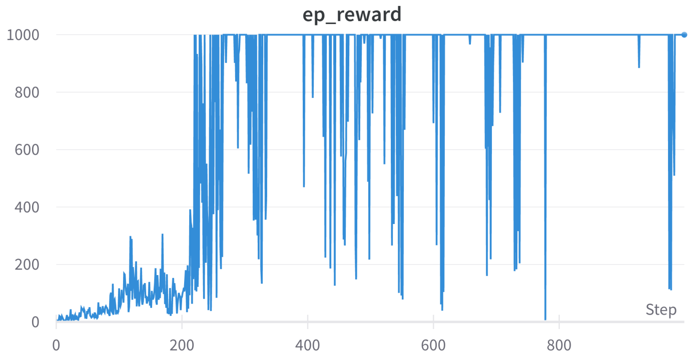

Figure 2: Exercise 1b

c) Look at 3

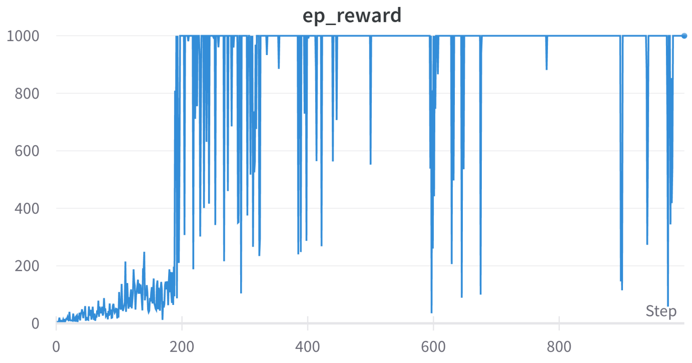

Figure 3: Exercise 1c

Question 1.1

How would you choose a good value for the baseline? Why is the training more stable when using a baseline?

The training is more stable when using a baseline. This is because it results in lower variance, while keeping an unbiased estimate.

We can get the baseline for a timestep by sampling the rewards and taking the mean of the experience rewards.

Task 2

Look at 4

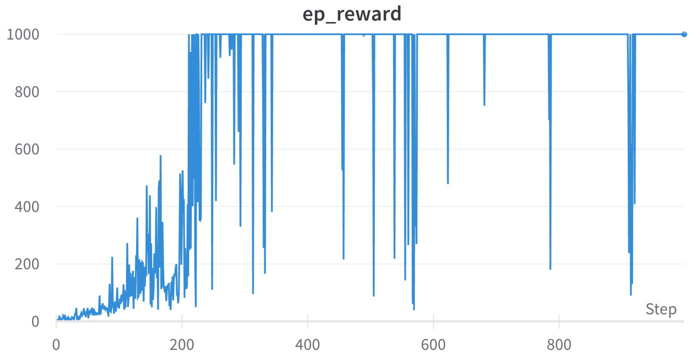

Figure 4: Exercise 2

Question 2.1

What are the strong and weak sides of using either a) constant variance during training, or b) learning the variance during training?

Pros of learning the variance is that it may converge faster. This is because learning the vairance is going to tell us how often to explot, and lower variance means we explot more, this means we may be able to learn faster. Cons of learning the variance during training is that the variance may increase instead of decreasing, and we may nto be able to learn anything is that way, because we would always be exploring instead of exploiting.

Question 2.2

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

Having an initialized lower variance might make the training faster, because we would exploit more often. On the other hand, if we are exploiting too often, we may take more time to learn, or we may not be able to learn at all.

Question 3

Why the method implemented in this exercise could not be directly used with experience replay? Which steps of the algorithm would be problematic to perform with experience replay? How the problematic steps could be resolved?

REINFORCE is on-policy method, thus using experience replay would not make sense. Taking past trajectories like we do in experience replay would not make sense because we are using past information to update the gradients with SGD (or AGD), and making derivative of past experiences would not make sense when updating the current moment.

We may solve this problem using importance sampling, this is because if the trajectories we are considering is close to the old trajectories, importance sampling lets us compute the new rewards based on the old computation by reusing the old samples to recalculate the total rewards.

Question 4.1

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?

Continuous unbounded space would allow extreme actions and thus rewarding them grately. This would be problematic because making a physical system go very fast may lead to brake it. This means that we might want to put a limit of the action.

Question 4.2

How could the problems appearing in Question 4.1 be mitigated without putting a hard limit on the actions?

We can fix that by giving negative rewards to extreme movements, penalizing them, but still allowing them such that we can explore more the space.

Question 5

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?

Yes, it's possible to make it, but because every action would be represented by its own distribution, it would make it way more expensive to compute, because it would need to learn each action on its own.