Exercise 7

Marco Di Francesco - 100632815
ELEC-E8125 - Reinforcement Learning

November 21, 2022

Task 1

Look at plot 1.

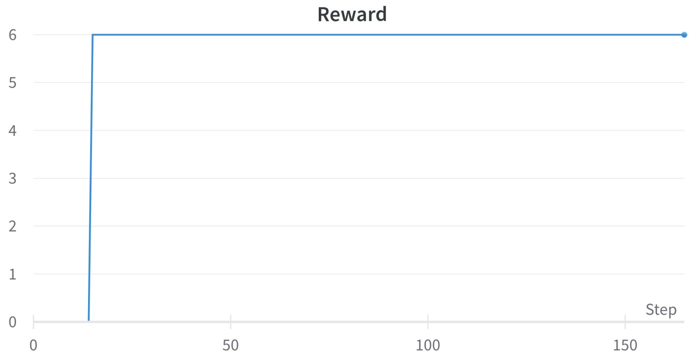

Figure 1: Exercise 1 plot

Question 1.1

The number of samples set the balance between exploration and exploitation, if we have a low number of sample we might not explore enough, if we have a high number of samples the algorithm is going to be computationally innefficient. So if the algorithm has a high number of sample we will be certain that the top-k values are going to very good representatives for the next mean and std, on the other it is going to slow down the training.

Question 1.2

Pros of **CEM with a learned dynamics model**:

- CEM with a learned dynamics model is significantly more sample efficient compared to Model-free RL algorithms because they learn from complete episodes, while Model-free RL algorithms methods use elementary steps of the system as samples, and thus exploit more information.

- CEM with a learned dynamics model compared to Model-free RL algorithms are known to be stable and is not sensitive to hyper-parameter setting.

  (https://arxiv.org/pdf/1810.01222.pdf)

Cons:

- Is has very limited implementations in the live environments because it is very computationally expensive to compute compared to model free RL algorithms.
- If we have noisier data in real environments would learn biased model thus learning lan error that would backpropagate in the model making it inacurate.

Task 2

Look at plot 2.

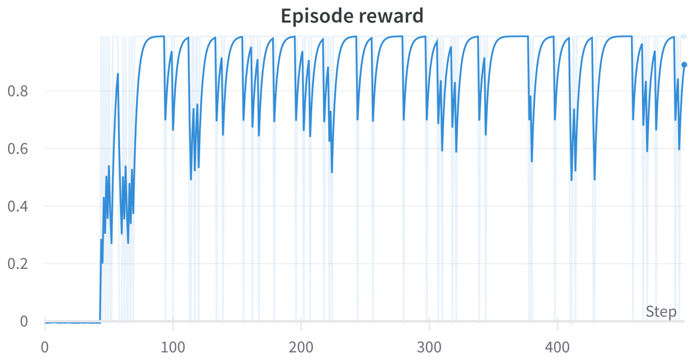

Figure 2: Exercise 2 plot

Question 2.1

1) What is the probability of reaching the goal state (a function of N)?

The probability of reaching the goal state is a function of N, and because for each state we have 2 actions the probability is $\frac{1}{2^{N-1}}$. The -1 is given that we have N tree height but N-1 steps.

2) If N is large, DQN (with the -greedy policy) usually fail to reach the goal state (in fact, N=10 is already challenging for DQN). In this case, which strategy will DQN converge to?

DQN with the -greedy policy if N is going to prefer the left during the exploration phase, and will never get the +1 reward. In this case the model-free algoithm uses greedy policy to explore, but because it will never experience the +1 (given that it is very very unlikely to go always right given the cost) it will tend to choose going left given the better reward and will never get to the +1 experience.

Question 2.2

- Selection: we move down from the root optimal child until we reach a leaf, we choose a child based on a tradeoff between exploration and exploitation
- Expansion: if we are not in a terminal node create a child according to the available actions in the current state
- Rollout/Simulation: run a rollout from the root until a terminal state is found
- Backpropagation: for each rollout we compute the new vaule starting from the leaf and backpropagaring to the parent, grandparent etc. . We do this by averageing the values of the children nodes with a weight based upon the number of visits.

Question 2.3

Actor Critic is beneficial in AlphaZero compared to using Monte Carlo because it learns a policy thanks to the Actor, and is able to use past experience to explore efficiently during the simulation phase. Monte Carlo estimates on the other hand uses a random policy which does learn past experience.

Task 2.1

Look at code.

Task 2.2

Look at code.

Task 2.3

Look at code.

Task 2.4

Greedy policy compared to PUCT search policy is not able to find the reward +1 and tends over time while learning to keep choosing going left such that has higher returns (that will tend to 0), thus never reaching the goal always right.