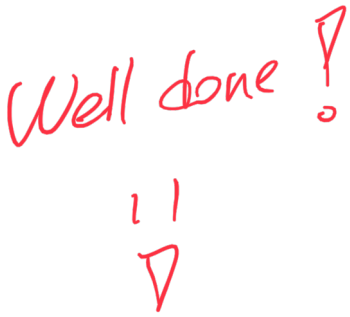
Reinforcement Learning: Assignment #6

Due on Sunday, June 11, 2023



Group 4 - Abu El Komboz, Tareq 3405686 | Jain, Likhit 3678905 | Wurm, Marcel 3695946

Task 1

Planning and Learning (4P)

(a) Why did the Dyna-Q+ (i.e., with exploration bonus) perform better in the first phase as well as in the second phase of the blocking and shortcut experiments (see figure below)? (2P)

In the first phase the exploration bonus leads to a higher cumulative reward because in relation to the size of the environment, the path from the start to the goal is rather long. So the exploration bonus biases states which are further away (as they are less likely visited due to the long path to them) and leads to more quickly finding the goal (which is indicated by the Dyna-Q+ algorithms cumulative rewards increasing after less time steps than the Dyna-Q algorithm).
In the second phase, the exploration bonus leads to a more likely exploration of the states near the new shortcut as these have likely not recently been visited (because they are not near the previously optimal path). Thus the Dyna-Q+ algorithm can capitalize on the newly introduced shortcut.

(b) Consider the tabular Dyna-Q algorithm shown on slide 9. How could you modify this algorithm in order to handle stochastic environments? Would your modification still perform well on changing environments? If not, how could you handle stochastic and changing environments? (2P)

The algorithm can be modified by changing the model to be able to handle multiple (R, S')-Pairs per (S, A)-Pair. The model would have to store a list of all previously observed (R, S')-Pairs from a given (S, A)-Pair (which are added in the line "$Model(S, A) \longleftarrow R, S'$") and a count of how often each (R, S')-Pair has been observed.
When retrieving a (R, S')-Pair from the model in the planning step, the pair is selected according to the probability of each pair for the given (S, A)-Pair over the previously observed distribution.
The model would probably perform noticeably worse on changing environments, because after a large amount of previously run time steps, the number of observations for the previously more likely pairs would be rather high. This results in the changed environment taking a very (very) long time to be properly reflected in the model.
One possible modification to the previous modification for handling stochastic and changing environments is to (instead of tracking the total number of each observed (R, S')-Pair in the model) keep a rolling list of only the last couple (a fixed number) of observations and derive the distribution only from these few pairs. This leads to new observations having a significant impact on the distribution much quicker (but could lead to the distribution being less precise overall).

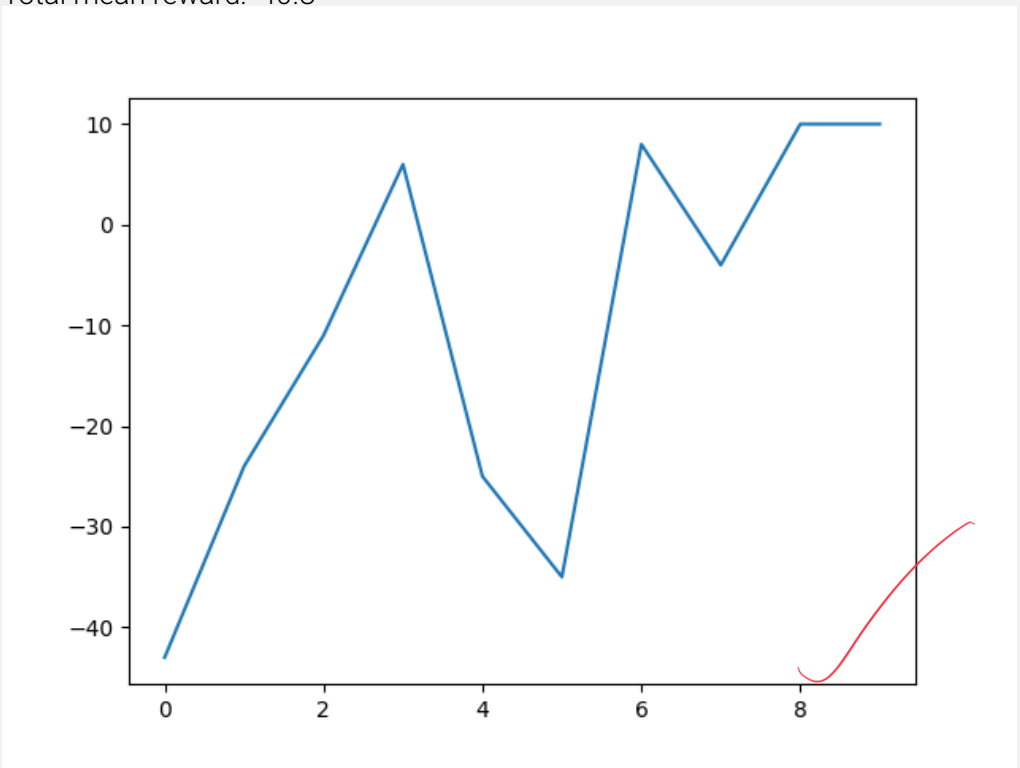
nice idea

Task 2

Monte Carlo Tree Search on the Taxi environment (6P)

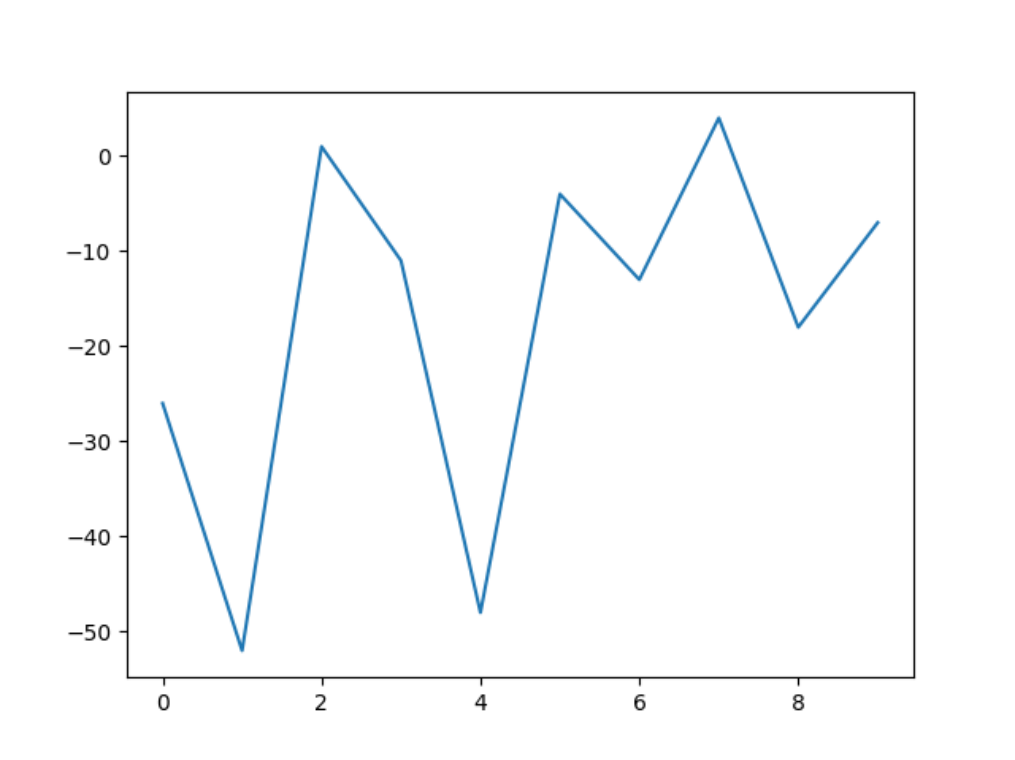
(a) Implement Monte Carlo Tree Search in order to solve the task. The agent might need some time in the first few runs but should get faster as he learns. Keep track of the mean return and plot it over the number of episodes. Report the average reward for solving the problem with MCTS. Is it better than the reward from the plain code template (which does not construct a tree and thus just performs Simple Monte Carlo Search)? (4P)

Our algorithm:
Total mean reward: -10.8

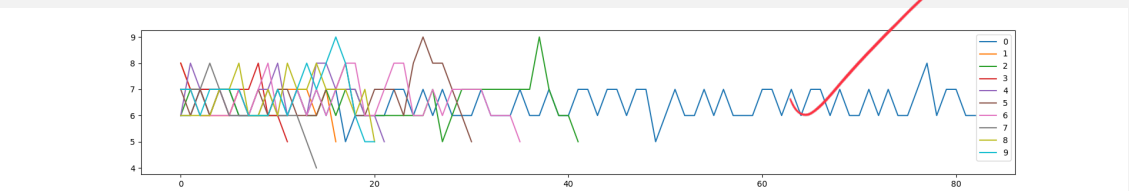


Task 2 continued on next page...

Template algorithm:
Total mean reward: -17.4



(b) How does the tree evolve? Keep track of the length of the tree and plot the length of the longest path over the number of iterations. (2P)



What's the axis?
better add labels.
Anyway good job !!.