CP8319 Assignment 3

Marko Vukovic

April 14, 2021

1

a)

Assuming $\gamma = 1$ we will find the best possible result given by a path in the given environment. Since the rewards are only received after arriving in a new state, we will start with the best 2 length path and extend it to the best 5 length path. To start, using the property of $S(2)$ which causes us to get $* - 10$ reward upon leaving we should go from $S(2) \rightarrow S(1)$ as it has the largest negative reward giving us a total of $R = 2$. As we have a length of 5 to work with, we can assume that the optimal path will repeat this process again. Now there are 3 sequences to order including going from $S(2) \rightarrow S(1)$ twice and visiting some other $S(N)$ once. As there is no value in going to state two, it does not make a difference if the other state occurs in the middle or end of the sequence. Finally we you compare the remaining two choices $S(?) \rightarrow S(2) \rightarrow S(1) \rightarrow S(2) \rightarrow S(1)$ or $S(2) \rightarrow S(1) \rightarrow S(2) \rightarrow S(1) \rightarrow S(?)$. In the first case it does not matter which state you start with as you do not receive the reward from that state, giving a best value of $R = 4$. In the second case the best possible option is to go to $S(0)$ in the last state giving a reward of $R = 4.1$ showing that this is the optimal path.

$$S(2) \rightarrow S(1) \rightarrow S(2) \rightarrow S(1) \rightarrow S(0)$$

2

a)

If the state space is large, several problems arise when trying to calculate the exact Q-values. One major benefit of utilizing an approximation function is that this allows generalization to new states. The agent will then perform similar behaviour in similar states as the q function approximation is similar. This also allows the agent to handle continuous state space, as with a traditional table approach continuous spaces have too many potential states to evaluate.

b)

All three tests in `q2_schedule.py` passed.

c)

N/A

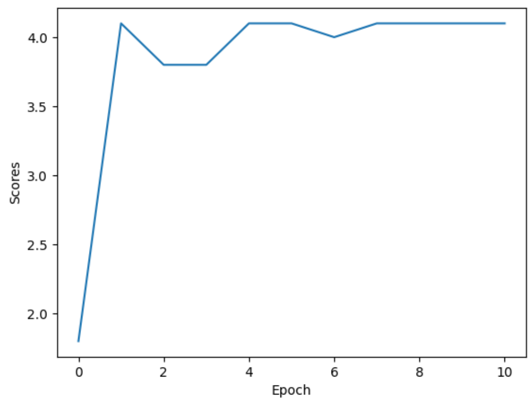
d)

N/A

3

a)

The model reached the optimal reward after training.



b)

This model also reached the optimal reward after training. The end policies from both models return $R = 4.1$, however from the graph you can see that the deep Q-network took less epochs to converge to the optimal reward.