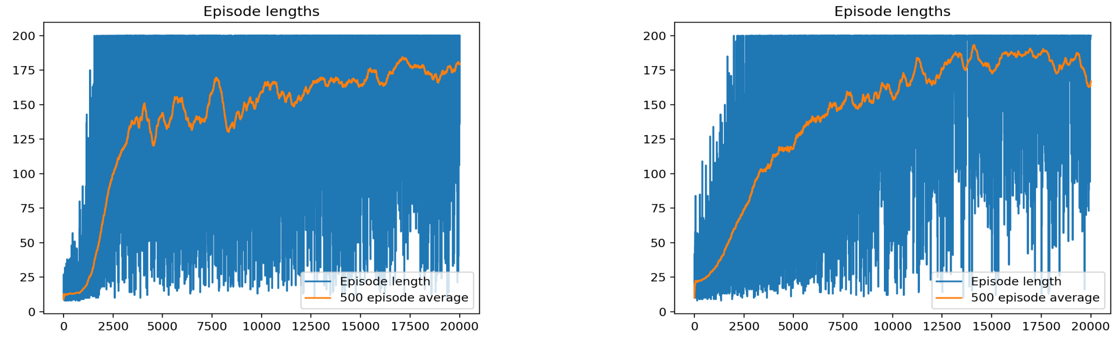
Exercise 3

Breno Aberle - 876438
ELEC-E8125 - Reinforcement Learning

October 11, 2020

1 Task 1

In task 1 the Q-Learning algorithm was implemented and run twice for 20000 episodes. In the first run with an epsilon of 0.2 and in the second epsilon decreases over time implemented with GLIE. The plots of the training performance are shown in figure 1.



(a) $\epsilon = 0.2$

(b) GLIE: $\epsilon = 0.1$ after 20000 episodes

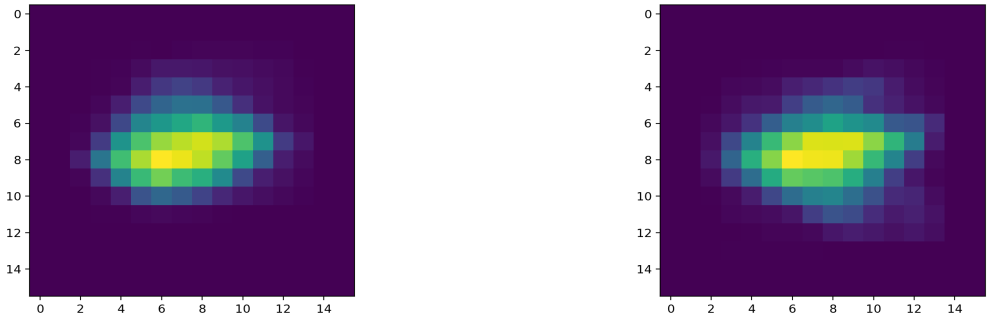
Figure 1: Training performance of Q-Learning

2 Task 2

First of all, the value function is calculated. The value function is the max values over the qgrid variable. As demanded in the task, I averaged the values of xdot and thetadot. Figure 2 shows the heatmap for constant epsilon as well as for decreasing epsilon after training.

3 Question 1

In this question, we were asked to justify how the heatmap looks before training, after one episode and after half way through. Plots were not required. However, I created them to

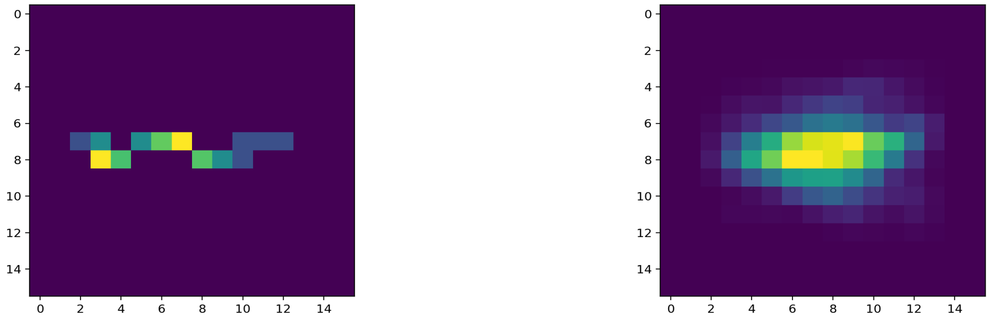


(a) $\epsilon = 0.2$

(b) GLIE: $\epsilon = 0.1$ after 20000 episodes

Figure 2: Heatmap over value function after training

prove my explanations. The heatmap before training is one colour for every cell. That makes sense because qgrid gets initialized with one value for every entry. Consequently, the heatmap must display only one colour. After one episode only the states of that episode get an update. Therefore you can see kind of a stripe in the heatmap like in figure 3a. If you compare the figure 3b with figure 2b you can see that the heatmap converged almost after halfway through training. The shapes looks almost the same, however with the upcoming updates it obviously develops and converges more.



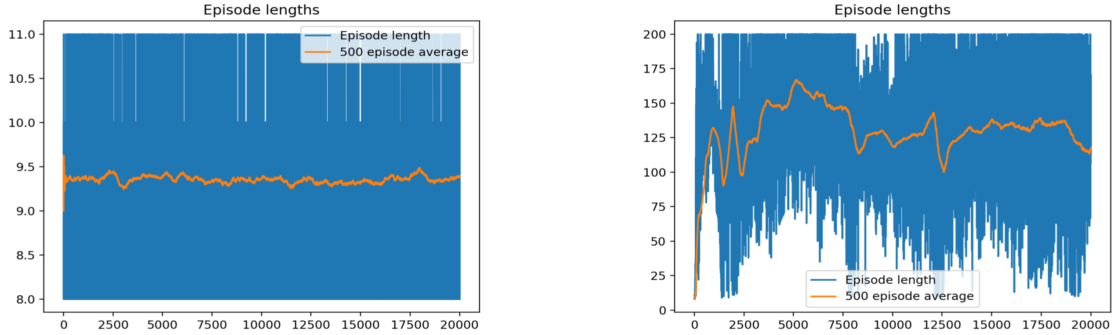
(a) After 1 episode

(b) After half way through

Figure 3: Heatmap over value function after training

4 Task 3

While letting the initial q values set to zero and setting epsilon to 0 the agent is obviously not able to learn which is shown in 4a. The average episode length is around 10 which is really bad.



(a) (b)

Figure 4: Caption for this figure with two images

5 Question 2

6 Question 2.1

However, if we set initial q values to 50 the model performs really which you can see in figure 4b. With an average of 150 time steps. Therefore, the model clearly performs better with an initialized q of 50.

7 Question 2.2

Setting epsilon to zero means eliminating exploration. Consequently, only choosing the greedy action. However, when the q values already have initialized with a higher number the model performs better. Since already values are assigned the agent visits more states. It doesn't only stick with same greedy failure as with initial q equals 0. Since values on other states are greater than zero and reward the will be more iterations.

This pre-initialization boosts learning. Since the agent explores more states and each state already has assigned state-action values not many episodes are needed till the agent converges into it's greedy pattern. We can see in figure 4b that in less than 1000 episodes the average episode length converges.

8 Task 4

Several parts needed to be modified to adapt the code to the Lunar Lander environment. First of all, we now have 4 instead of 2 actions. Furthermore, the environment gets much more complex because instead of 4 we have know 8 variables. This makes things more complicated. For each variable we need create new discretization. The initial q are set back to 0 and we are still applying GLIE. With more variables the dimension of q grid increases drastically. In the q value update function not much needed to get updated since I chose a convenient indexing way that's robust to the increase of state variables. Since the environment got much more complex the computation increased drastically. The training took almost one hour to complete.

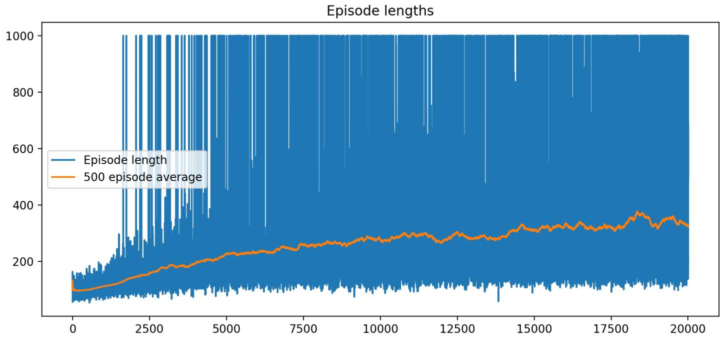


Figure 5: Performance plot Lunar Lander with GLIE.

9 Question 3.1

The lunar lander was not able to learn a useful behaviour. Already the long computation time showed how complex the environment is. Therefore, the environment is too complex to be learned by these q-learning setup. A more powerful model or approach is needed to model that environment.

References