Assignment 2 - OpenAi Gym

FrozenLake-v0

SFFF S=start, safe
FHFH F=frozen surface, safe
FFFH H=hole, unsafe
HFFG G=goal, where the frisbee are

Actions state:
0: left
1: down
2: right
3: up

Exercise 2a

We chose to use a table to represent the data structure of the Q-function. In our table the index (0-3) represent an action and the values are the Q-values. Each row in the table is a state (0-15) of the environment. This makes it easy to initialize and fetch max-values, which we used in this exercise.

Exercise 2c

Without ε will the agent always pick the best Q-value. With the ε we now have a probability for exploration that will let the agent to explore and learn about the environment.

Exercise 2d

The reward quantifier and Q(s, South) are both estimates of how good the action 'South' is. The relationship between them is that high Q-values for an action will also give high long term rewards for that action.

Such quantities can help give better estimates of Q(s, South), and Q(s, a) in general, by trying each action in a state 100 times to estimate total reward of that action.

Greedy policies will not consider long term rewards (and will not try to estimate total reward), just choose the seemingly best action in the given state.

Exercise 4

By using ε-greedy instead of max(Q(s',a)), where *s'* is next state and *a* is a action the algorithm is more likely to learn from actions with lower Q-values. This will make the algorithm more willing to explore in the beginning. ε is being decreased over time and when ε is zero the learning policy will be identical to Q-learning.

Exercise 5

Off-policy	**On-policy**
Learns Q-values relative to a greedy policy	Learns Q-values relative to the policy it follows
Only takes optimal action	Sometimes takes optimal actions, and sometimes explore other actions
Compares the best action in next state with the action just executed in the current state	Compares the next action in the next state with the action just executed in the current state
Does not try to improve the policy	Attempts to evaluate or improve the policy that is used to make decisions
Will not learn to be careful in environments where exploration is costly	Will learn to be careful in an environment where exploration is costly

Table 1

When learning Q-values, we use the optimal action in the next state from a fixed Q-function. This policy is greedy (only takes optimal actions) and therefore is off-policy.

Exercise 6

In table 1 we have compared *on-* and *off-policy*. We can see the human agent as a policy that generates data, but that human might make some mistakes. If we chose to use off-policy, the optimal action would always be chosen by the policy, which might not be the correct action if the human made mistakes. By choosing on-policy we can tolerate some errors in the data because the policy will be improved during the learning process.

Exercise 7 - Taxi-v1

We chose to use a table to represent the data structure of the Q-function. In our table the index (0-5) represent an action and the values are the Q-values. Each row in the table is a state (0-499) of the environment. This makes it easy to initialize and fetch max-values, which we used in this exercise.

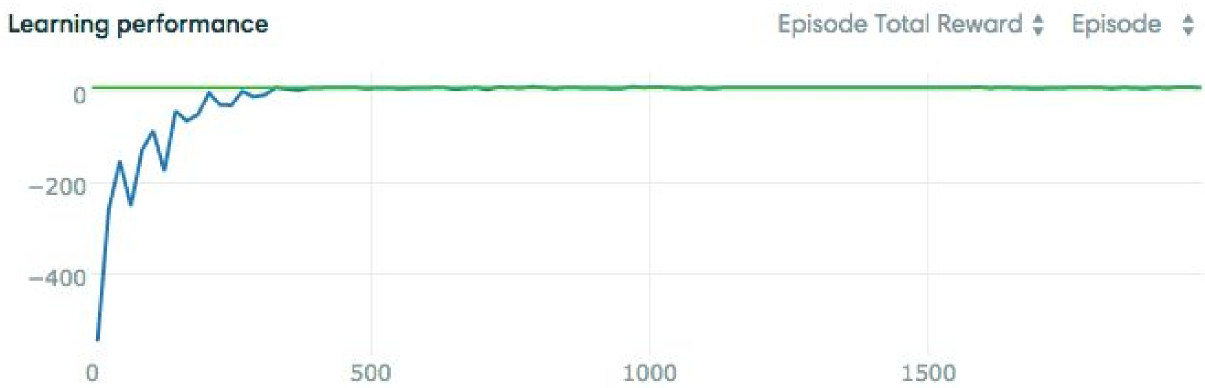
The performance of the algorithm (2000 episodes)

Q-learning:

Solved after 467 episodes. Best 100-episode average reward was 10.72 ± 0.36. (Taxi-v1 is considered "solved" when the agent obtains an average reward of at least 9.7 over 100 consecutive episodes.)

(https://gym.openai.com/evaluations/eval_uevhfH57QHWooM72xiaUg)

Q-learning with method from ex 4:



Solved after 779 episodes. Best 100-episode average reward was 10.62 ± 0.40. (Taxi-v1 is considered "solved" when the agent obtains an average reward of at least 9.7 over 100 consecutive episodes.)

(https://gym.openai.com/evaluations/eval_7HATPyW6TES2rWLjMfNVzQ)

As we see from the figures above, when applying the method from exercise 4, the learning time is slower than the regular Q-learning algorithm.