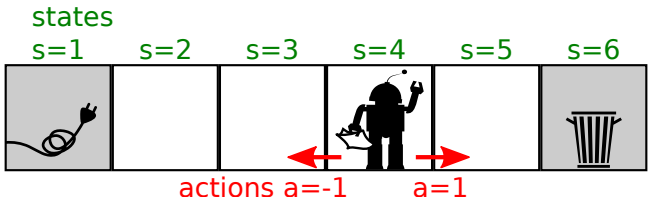
IN4320 Machine Learning Exercise

Exercises Reinforcement Learning

In this exercise you will apply various basic reinforcement learning methods to a toy example. Exercises 1-5 can be done based on the lecture slides, for the other questions either use the resources provided on Blackboard or literature you found on your own. Cite all the references you employed.

Exercise 1 (10 points) Proof that the return $R_t = \sum_{h=0}^{\infty} \gamma^h r_{t+h+1}$ is bounded for $0 \leq \gamma < 1$ and for bounded rewards $-10 \leq r_{t+h+1} \leq 10 \; \forall h \in \{0, 1, \ldots, \infty\}$. Note: $0^0 = 1$.

Scenario Our robot found a new job as cleaning robot. The robot only has the actions "left" and "right". It is working in a corridor with 6 states, the two end-states are terminal, i.e., the episode ends immediately once the robots reaches them. It gets a reward of 1 when reaching the left state (charger) and a reward of 5 when reaching the right state (trash bin).



$\mathcal{S} = \{1, 2, 3, 4, 5, 6\}$ $\mathcal{A} = \{-1, 1\} = \{\text{left, right}\}$

$$s_{t+1} = \begin{cases} s_t & \text{if } s_t \text{ is terminal } (s_t = 1 \text{ or } s_t = 6) \\ s_t + a_t & \text{otherwise, where } a_t \in \mathcal{A} \end{cases}$$

$$r_{t+1} = \begin{cases} 5 & \text{if } s_{t+1} = 6 \text{ and } s_t \neq 6 \\ 1 & \text{if } s_{t+1} = 1 \text{ and } s_t \neq 1 \\ 0 & \text{otherwise} \end{cases}$$

Exercise 2 (10 points) Implement Q-iteration. For $\gamma = 0.5$ the optimal Q-function is given in the table below. Show the values after each iteration. What is the optimal policy π^*?

action \ state	1	2	3	4	5	6
left	0.0	1.0	0.5	0.625	1.25	0.0
right	0.0	0.625	1.25	2.5	5.0	0.0

Exercise 3 (10 points) Show the optimal value functions Q^* for $\gamma = 0$, $\gamma = 0.1$,$\gamma = 0.9$, and $\gamma = 1$. Discuss the influence of the discount factor γ. Explain why $\gamma = 1$ will work here in contrast to Exercise 1.

Exercise 4 (15 points) Implement Q-learning. For the rest of the exercises use $\gamma = 0.5$. Try different values for the exploration ε and the learning rate α. Plot the difference (2-norm) between the value function estimated by Q-learning and the true value function (table above) over the number of interactions with the system. Provide plots for different values of ε and α. Describe and explain the differences in behavior.

Exercise 5 (15 points) Now the robot is partially broken and stays at the same state with a probability of 30%, else it works correctly. Test this scenario with Q-iteration and Q-learning. What happens? Can you still use the same Q-learning parameters?

Exercise 6 (40 points) Somebody tried to fix the robot. At least it is not stopping any longer but now we have a different problem: Rather than moving $s' = s + a$ the robot now moves $s' = s+a+\mathcal{N}(0, 0.01)$, where $\mathcal{N}(0, 0.01)$ is a Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 = 0.01$. Now we need to consider continuous states and non-deterministic transitions. The terminal states are reached for $s < 1.5$ and $s \geq 5.5$ respectively. Implement value function approximation with radial basis functions (we still have 2 discrete actions) for Q-learning or another algorithm of your choice. Provide pseudo-code. How many basis functions do you need? Do the widths have an influence on the performance? What happens to the learning performance compared to the other approaches? Illustrate with plots and discuss.

Hints: As you cannot use Q-iteration any longer to find the ground truth you will need to show how quickly the learning converges and how the performance evolves (e.g., expected return). Start with 6 basis functions centered at $\{1, 2, 3, 4, 5, 6\}$ and test your implementation with $\sigma^2 = 0$. Often normalizing the RBF network[1] is a good idea.

Exercise 7 (max 40 points) Bonus points[2]: implement one additional RL algorithm (e.g., policy iteration, TD(λ), SARSA, actor critic, policy search) or an additional variant of the above algorithms (e.g., state discretization, different function approximator, eligibility traces). Provide pseudo-code and illustrate the performance with plots. Discuss the the advantages and disadvantages of the chosen approach compared to the other algorithms you implemented in these exercises.

[1] See, e.g., http://en.wikipedia.org/wiki/Radial_basis_function_network#Normalized

[2] You can get a maximum of 100 points (which corresponds to a grade of 10) for Exercises 1-6. Points you missed in Exercises 1-6 can be compensated for by Exercise 7. The overall maximum grade is 10.