Solutions for week 6
Variants of TD-learning methods and eligibility traces

Exercise 1: Eligibility traces

Last week, we applied the SARSA algorithm to the case of a linear track with actions 'up' and 'down'. We found that it takes a long time to propagate the reward information through state space. The eligibility trace is introduced as a solution to this problem.

Reconsider the linear maze from Figure 1 in exercise 2, but include an eligibility trace: for each state s and action a, a memory $e(s,a)$ is stored. At each time step, all the memories are reduced by a factor $\lambda < 1$: $e(s,a) = \lambda e(s,a)$, except for the memory corresponding to the current state s^* and action a^*, which is incremented:

$$e(s^*, a^*) = \lambda e(s^*, a^*) + 1\,. \tag{1}$$

Now, unlike the case without eligibility trace, all Q-values are updated at each time step according to the rule

$$\forall(s,a) \quad \Delta Q(s,a) = \eta \left[r - \big(Q(s^*, a^*) - Q(s', a')\big)\right] e(s,a)\,. \tag{2}$$

where s^*, a^* are the current state and action, and s', a' are the immediately following state and action.

We want to check whether the information about the reward propagates more rapidly. To find out, assume that the agent goes straight down in the first trial. In the second trial it uses a greedy policy. Calculate the Q-values after two complete trials and report the result.

Hint: Reset the eligibility trace to zero at the beginning of each trial.

Solution:

The table below shows the evolution of the Q values for each relevant state action pair during the first 2 trials, starting at the first step when there is a non-zero update $\Delta = \eta\,[r - (Q(s^*, a^*) - Q(s', a'))]$. We assume that the agent goes straight down in the first trial, that it always picks the best action, and that the eligibility traces are reset when the agent picks the reward, and the agent is put back to the starting position.

trial	transition		(s, a_1)	(s', a_1)	(s'', a_1)	Δ
1	$s'' \to s'''$	Q	0	0	0	η
		e	λ^2	λ	1	–
2	$s \to s'$	Q	$\eta\lambda^2$	$\eta\lambda$	η	$\eta^2(\lambda - \lambda^2)$
		e	1	0	0	–
2	$s' \to s''$	Q	$\eta\lambda^2 + \eta^2(\lambda - \lambda^2)$	$\eta\lambda$	η	$\eta^2(1 - \lambda)$
		e	λ	1	0	–
2	$s'' \to s'''$	Q	$\eta\lambda^2 + 2\eta^2\lambda - 2\eta^2\lambda^2$	$\eta\lambda + \eta^2(1-\lambda)$	η	$\eta - \eta^2$
		e	λ^2	λ	1	–
3	$s \to s'$	Q	$2\eta\lambda^2 + 2\eta^2\lambda - 3\eta^2\lambda^2$	$2\eta\lambda + \eta^2 - 2\eta^2\lambda$	$2\eta - \eta^2$	$\cdots$
		e	1	0	0	–

$\cdots$

Although the Q-values for s'' are the same as without the eligibility trace (see exercise from last week), the Q-values for s and s' already start to approach their asymptotical value (i. e. 1) in the first trial.

Exercise 2: Eligibility traces in continuous space

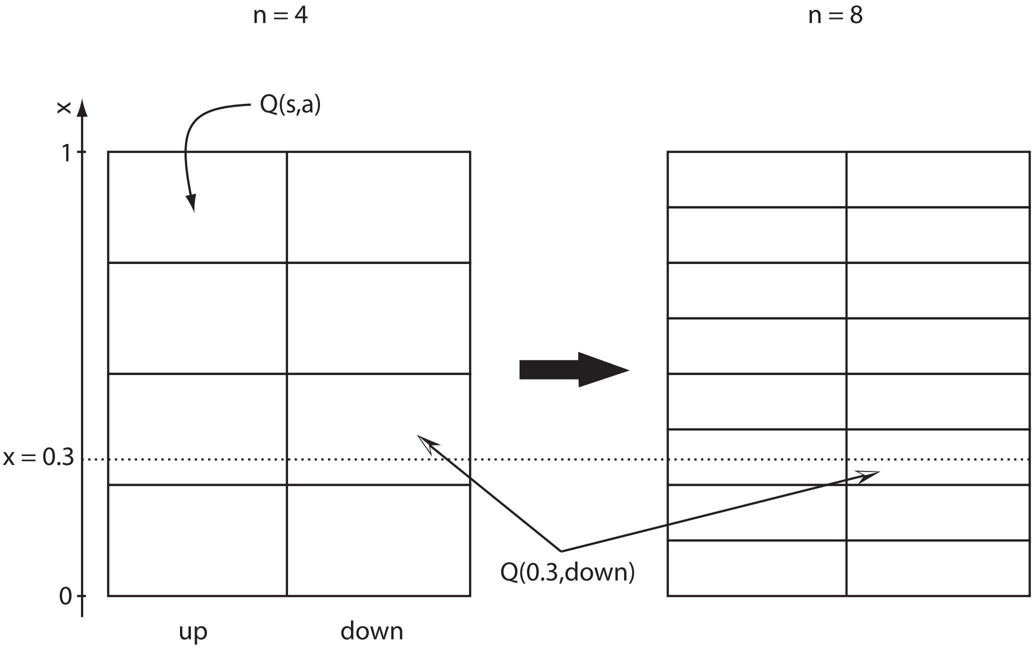

Figure 1: Figure for Exercise 2

The left part of Figure 1 shows a different representation of last week's "linear track" exercise: the vertical divisions represent different states, and the two column correspond to the two possible actions available to the agent: go up or down. Each square represents a possible state-action combination, and thus a Q-value. (Note that the uppermost "up" action and the lowermost "down" action should be "greyed out": they are impossible. But this is not relevant to the rest of this exercise.)

Suppose now that the agent moves in a continuous 1-dimensional space $0 \leq x \leq 1$, with the target located at $x = 0$. Separate this state space into n equal bins of width $\Delta x = 1/n$. In each time step, the agent moves by one bin. Vary the discretization by varying n: $n = 4, 8, 16 \ldots$

a. Suppose that one action (such as move down) corresponds to one time step Δt in 'real time'. How should we rescale the parameter Δt, so that the speed $v = \Delta x/\Delta t$ remains constant when we change the discretization?

b. We use an eligibility trace with decay parameter λ. How should we rescale λ, in order that the "speed of information propagation" in SARSA(λ) remains constant?

   Hint: Consider the Q-value at a fixed x, for example at $x = 0.5$, after 2 complete learning trials.

Solution:

a. If $\Delta x/\Delta t$ has to remain constant, then Δt should vary like Δx, i.e., $\Delta t \propto 1/n$.

b. A point "sitting" at x is $x/\Delta x = x \cdot n$ steps away from the reward (assuming we always choose the "down" action). As we have seen in exercise 2, on the first trial, the Q-value of a state d steps from the trial is updated proportional to λ^d. Thus, if we want the update to stay constant under rescaling, we need

$$\Delta Q(s,a) \propto \lambda^d = \lambda^{x \cdot n} = cst$$

   This holds if we use λ with the following "rescaling": $\lambda = \tilde{\lambda}^{\frac{1}{n}}$ where $\tilde{\lambda}$ is constant.

Exercise 3: 2-step SARSA algorithm

In class we have discussed the SARSA algorithm and shown that, after convergence, the resulting Q-values solve (in expectation) the Bellman equation for *neighboring* states (Variant A in the slides, fixed/non-fluctuating Q-values after convergence). Your friend claims that a 2-step SARSA for

$$\Delta Q(s_t, a_t) = \eta \left[r_t + \gamma r_{t+1} + \gamma^2 Q(s_{t+2}, a_{t+2}) - Q(s_t, a_t) \right] , \tag{3}$$

should work just as well.

To simplify the analysis, we assume that the environment has no loops (i.e., the graph is directed) so that we can consider $\gamma = 1$.

a. Assume that the 2-step SARSA algorithm converges in expectation. Proceed as during the lecture to show that $\mathbb{E}\big[\Delta Q(s_t, a_t)\big] = 0$ implies

$$Q(s_t, a_t) = \sum_{s'} P^{a_t}_{s_t \to s'} \left[R^{a_t}_{s_t \to s'} + \sum_{a'} \pi(s', a') B_1(s', a') \right]$$

where

$$B_1(s', a') = \sum_{s''} P^{a'}_{s' \to s''} \left[R^{a'}_{s' \to s''} + \sum_{a''} \pi(s'', a'') Q(s'', a'') \right]$$

$$B_2(s'', a'') = \sum_{s'''} P^{a''}_{s'' \to s'''} \left[R^{a''}_{s'' \to s'''} + \sum_{a'''} \pi(s''', a''') Q(s''', a''') \right]$$

b. Show the equivalence of the previous equation to the 1-step Bellman equation.

Solution:

a. $\mathbb{E}\big[\Delta Q(s_t, a_t)\big] = 0$ implies

$$Q(s_t, a_t) = \sum_{s'} P^{a_t}_{s_t \to s'} \left(R^{a_t}_{s_t \to s'} + \langle \gamma r_{t+1} + \gamma^2 Q(s_{t+2}, a_{t+2}) \rangle_{s_{t+1} = s'} \right) \tag{4}$$

$$= \sum_{s'} P^{a_t}_{s_t \to s'} \left(R^{a_t}_{s_t \to s'} + \gamma \sum_{a'} \pi(s', a') \left(\sum_{s''} P^{a'}_{s' \to s''} R^{a'}_{s' \to s''} + \langle \gamma Q(s_{t+2}, a_{t+2}) \rangle_{s_{t+2} = s''} \right) \right) \tag{5}$$

where we used the fact that the probability of action a' given state s' is determined by the policy $\pi(s', a')$. Continuing along the same lines we find the claimed result.

b. If we expand the 1-step Bellman equation by iteratively inserting the right-hand-side of the Bellman equation, we arrive at the 2 equations in a.

Exercise 4: Computer exercises: Environment 1 (part 2)[1]

Complete the computer exercise for environment 1.

[1] Start this exercise in the second exercise session of week 3.