Artificial Neural Networks (Gerstner). Exercises for week 2

Reinforcement Learning: Q-value and SARSA

Exercise 1. Iterative update

We consider an empirical evaluation of $Q(s,a)$ by averaging the rewards for action a over the first k trials:

$$Q_k = \frac{1}{k}\sum_{i=1}^{k} r_i.$$

We now include an additional trial and average over all $k+1$ trials.

a. Show that this procedure leads to an iterative update rule of the form

$$\Delta Q_k = \eta(r_k - Q_{k-1}),$$

(assuming $Q_0 = 0$).

b. What is the value of η?

c. Give an intuitive explanation of the update rule. *Hint: Think of the following: If the actual reward is larger than my estimate, then I should ...*

Exercise 2. Greedy policy and the two-armed bandit

In the "2-armed bandit" problem, one has to choose one of 2 actions. Assume action a_1 yields a reward of $r = 1$ with probability $p = 0.25$ and 0 otherwise. If you take action a_2, you will receive a reward of $r = 0.4$ with probability $p = 0.75$ and 0 otherwise. The "2-armed bandit" game is played several times and Q values are updated using the update rule $\Delta Q(s,a) = \eta[r_t - Q(s,a)]$.

a. Assume that you initialize all Q values at zero. You first try both actions: in trial 1 you choose a_1 and get $r = 1$; in trial 2 you choose a_2 and get $r = 0.4$. Update your Q values ($\eta = 0.2$).

b. In trials 3 to 5, you play greedy and always choose the action which looks best (i.e., has the highest Q-value). Which action has the higher Q-value after trial 5? (Assume that the actual reward is $r = 0$ in trials 3-5.)

c. Calculate the expected reward for both actions. Which one is the best?

d. Initialize both Q-values at 2 (optimistic). Assume that, as in the first part, in the first two trials you get for both actions the reward. Update your Q values once with $\eta = 0.2$. Suppose now that in the following rounds, in order to explore well, you choose actions a_1 and a_2 alternatingly and update the Q-values with a very small learning rate ($\eta = 0.001$). How many rounds (one round = two trials = one trial with each action) does it take *on average*, until the maximal Q-value also reflects the best action?
Hint: For $\eta \ll 1$ we can approximate the actual returns r_t with their expectations $E[r]$.

Exercise 3. SARSA algorithm

In the lecture, we introduced the SARSA (state-action-reward-state-action) algorithm, which (for discount factor $\gamma = 1$) is defined by the update rule

$$\Delta Q(s,a) = \eta\left[r - \left(Q(s,a) - Q(s',a')\right)\right] , \tag{1}$$

where s' and a' are the state and action subsequent to s and a. In this exercise, we apply a greedy policy, i.e., at each time step, the action chosen is the one with maximal expected reward, i.e.,

$$a_t^* = \arg\max_a Q_a(s,a)\,. \tag{2}$$

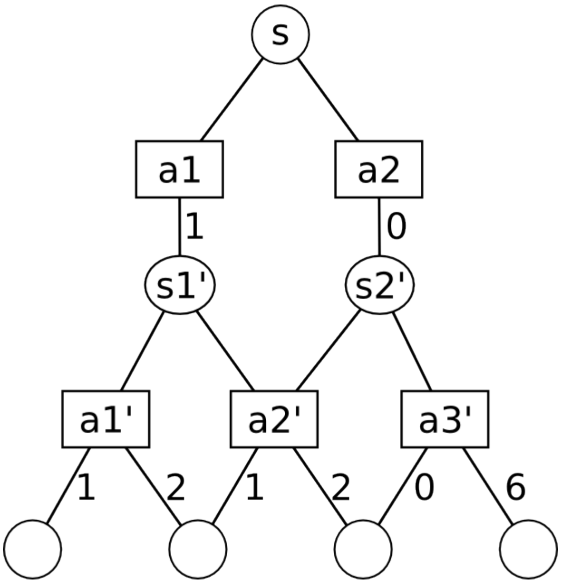

Figure 1: A tree–like environment.

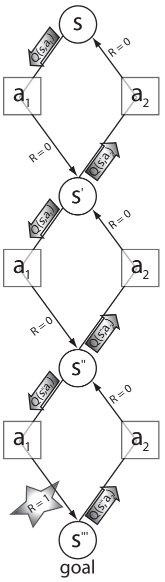

Figure 2: A linear maze.

If the available actions have the same Q-value, we take both actions with probability 0.5.

Consider a rat navigating in a 1-armed maze (=linear track). The rat is initially placed at the upper end of the maze (state s), with a food reward at the other end. This can be modeled as a one-dimensional sequence of states with a unique reward ($r = 1$) as the goal is reached. For each state, the possible actions are going up or going down (Fig. 2). When the goal is reached, the trial is over, and the rat is picked up by the experimentalist and placed back in the initial position s and the exploration starts again.

a. Suppose we discretize the linear track by 6 states, $s_1, \ldots, s_6$ where s_1 is the initial state and s_6 is the goal state. Initialize all the Q-values at zero. How do the Q-values develop as the rat walks down the maze in the first trial?

b. Calculate the Q-values after 3 complete trials. How many Q-values are non-zero? How many trials do we need so that information about the reward has arrived in the state just 'below' the starting state?

c. What happens to the learning speed if the number of states increases from 6 to 12? How many Q-values are non-zero after 3 trials? How many trial do we need so that information about the reward has arrived in the state just 'below' the starting state?

Exercise 4. Bellman equation

Use the Bellman equation to calculate $Q(s, a1)$ and $Q(s, a2)$ for the scenario shown in Figure 1. Consider two different policies:

- Total exploration: All actions are chosen with equal probability.
- Greedy exploitation: The agent always chooses the best action.

Note that the rewards/next states are stochastic for the actions $a1'$, $a2'$ and $a3'$. Assume that the probabilities for the outcome of these actions are all equal, and the discount factor γ is 1.

Exercise 5. 3-step SARSA algorithm

In class we have discussed the SARSA algorithm (for discount factor $\gamma \leq 1$) and shown that, after convergence, the resulting Q-values solve (in expectation) the Bellman equation for *neighboring* states. Your friend claims that a 3-step SARSA for

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

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 3-step SARSA algorithm converges in expectation. Proceed as during the lecture to show that $\langle \Delta Q(s_t, a_t) \rangle = 0$ 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

$$B_1(s', a') = \sum_{s''} p^{a'}_{s' \to s''} \left[R^{a'}_{s' \to s''} + \sum_{a''} \pi(s'', a'') B_2(s'', a'') \right]$$
$$B_2(s'', a'') = \sum_{s'''} p^{a''}_{s'' \to s'''} \left[R^{a''}_{s'' \to s'''} + \sum_{a'''} \pi(s''', a''') Q(s''', a''') \right]$$

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