ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Unsupervised and reinforcement learning in neural networks
Week 7

Lecturer: Marc-Oliver Gewaltig

Assistants: Dane Corneil
Berat Denizdurduran

Reinforcement learning

Exercise 1 (in class): Iterative update

We consider an empirical evaluation of $Q(s, a)$ by averaging the rewards for action a over the first k and later $k + 1$ trials:

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

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

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

(assuming $Q_0 = 0$).

1.2. What is the value of η?

1.3. Give an intuitive explanation of the update rule.

Exercise 2: Greedy policy and the two-armed bandit

In the "2-armed bandit" problem, one has to choose one of 2 actions at every time step. 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 $r = 0.7$ with probability $p = 0.5$ and 0 otherwise.

2.1. Assume that you initialize all Q values at zero. In the first round you try both actions. You choose a_1 and get $r = 1$, then you choose a_2 and get $r = 0.7$. Update your Q values ($\eta = 0.2$). Which action would you choose in the next time step if you were following an *action-greedy* policy, i. e. choosing the action with the highest associated Q-value?

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

2.3. Initialize both Q-values at 2 (optimistic). Assume that, as in in the first part, in the first round you get for both actions the reward. Update your Q values once with $\eta = 0.2$. Suppose now that in the following rounds, 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 does it take *on average*, until the maximal Q-value also reflects the best action? *(Hint: Transform the discrete online update rule for the two Q-values into differential equations for the expected Q-values after each time step.)*

Exercise 3: 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.

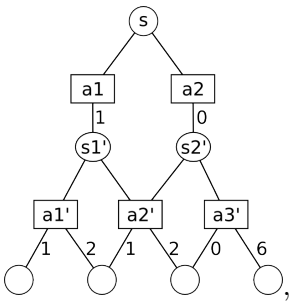

Figure 1.

Exercise 4: SARSA algorithm

In the lecture, we introduced the SARSA (state-action-reward-state-action) algorithm, which is defined by the update rule

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

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.,

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

Consider a rat navigating in a 1-armed maze. 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 linear sequence of states with a unique reward 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 rat is placed back in the initial position s and the exploration starts again.

1. Initialize all the Q-values at zero. How do the Q-values develop as the rat walks down the maze?
2. Calculate the Q-values after 3 complete trials. What happens to the learning speed if the number of states increases?

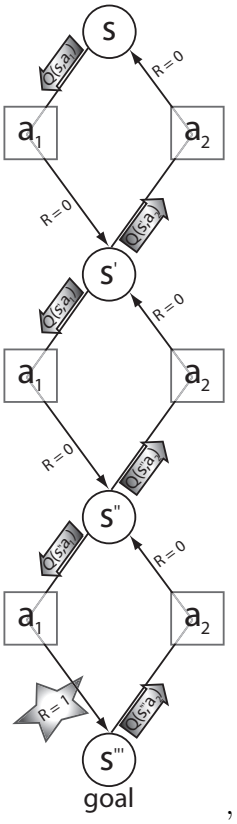

Figure 2.