Exercises for week 4
Reinforcement Learning: Basics

Exercise 1: Iterative update[1]

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

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

(assuming $Q_0 = 0$).

b. What is the value of η_k?

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: Batch vesrsus online learning rules: Recap

[1] The result is also used in class; the calculation is analog to a calculation that was done for online k-means clustering.

We define the mean squared error in a dataset with P data points as

$$E^{\text{MSE}}(\boldsymbol{w}) = \frac{1}{2}\frac{1}{P}\sum_{\mu}(t^{\mu} - \hat{y}^{\mu})^2 \tag{1}$$

where the output is

$$\hat{y}^{\mu} = g(a^{\mu}) = g(\boldsymbol{w}^T\boldsymbol{x}^{\mu}) = g(\sum_k w_k x_k^{\mu}) \tag{2}$$

and the input is the $\boldsymbol{x}^{\mu}$ with components $x_1^{\mu} \dots x_d^{\mu}$.

a. Calculate the update of weight w_j by gradient descent (batch rule)

$$\Delta w_j = -\eta \frac{dE}{dw_j} \tag{3}$$

Hint: Apply chain rule

b. Rewrite the formula by taking one data point at a time (stochastic gradient descent). What is the difference to the batch rule?

c. Rewrite your result in b in vector notation (hint: use the weight vector $\boldsymbol{w}$ and the input vector $\boldsymbol{x}^{\mu}$). Show that the update after application of data point μ can be written as

$$\Delta\boldsymbol{w} = \eta\delta(\mu)\boldsymbol{x}^{\mu}$$

where $\delta(\mu)$ is a scalar number that depends on μ. Express $\delta(\mu)$ in terms of $t^{\mu}, \hat{y}^{\mu}, g'$.

d. The result is a non-Hebbian rule. Nevertheless, please try to identify the 'pre' contribution and the 'post' contributions. Which term makes it non-Hebbian?

Exercise 4: Geometric interpretation of an artificial neuron: Recap

Consider the single-neuron function in 2-D with

$$y = g(\boldsymbol{x}^T \boldsymbol{w}) \tag{4}$$

where g is a strictily increasing activation function, $\boldsymbol{x} = (x_1, x_2, -1) \in \mathbb{R}^{2+1}$ is the extended 2-dimensional input (i.e., the threshold/bias value has been integrated as an extra input $x_3 = -1$), and $\boldsymbol{w} = (w_1, w_2, w_3) \in \mathbb{R}^3$ is the weight vector. The hyperplane $\boldsymbol{x}^T \boldsymbol{w} = 0$ describes the boundary between where the neuron is on, i.e., $\boldsymbol{x}^T \boldsymbol{w} > 0$, and where it is off, i.e., $\boldsymbol{x}^T \boldsymbol{w} < 0$. Consider this hyperplane in the 2-D space of (x_1, x_2) and answer the following questions:

a. The hyperplane is a line in 2-D. What is the slope of this line as a function of w_1, w_2, and w_3? Where does the line intersect with the y-axis and where with the x-axis?

b. Is it possible to have two weight vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ such that $\boldsymbol{w} \neq \boldsymbol{w}'$ but $\boldsymbol{x}^T \boldsymbol{w} = 0$ and $\boldsymbol{x}^T \boldsymbol{w}' = 0$ describe the same hyperplane? If yes, what conditions $\boldsymbol{w}$ and $\boldsymbol{w}'$ must meet?

c. For the general case of $\boldsymbol{x} = (x_1, ..., x_N, -1) \in \mathbb{R}^{N+1}$, what is the distance of the hyperplane $\boldsymbol{x}^T \boldsymbol{w} = 0$ from the origin in $\mathbb{R}^N$? Where does the hyperplane intersect with the x_n-axis for $n \in \{1, ..., N\}$?

d. Use the online learning rule you derived in Exercise 3c and describe, in words, how the separating hyperplane in $\mathbb{R}^N$ changes after each update. Make sure you consider the effects of both changing bias/threshold on one side and changing weight parameter $\boldsymbol{w}$ on the other side.