Artificial Neural Networks (Gerstner). Exercises for week 3

Markov Decision Processes

Exercise 1. Optimal policies for finite horizon.

Create a Markov Decision Process where the optimal horizon-T policy depends on the time step, i.e. there is at least one state s and one pair of timesteps t and t' such that $\pi^{(t)}(a|s) \neq \pi^{(t')}(a|s)$.

Hint: You can choose $T = 2$ for simplicity.

Exercise 2. Shortest path search.

Let $\mathcal{S} = \{s_1, s_2, s_3, \ldots\}$ denote a set of vertices (think of cities on a map) and let the vertices be connected by some edges $e_{s_i,s_j} \in (0, \infty]$ (think of distances between cities), where $e_{s_i,s_j} = \infty$ indicates that there is no direct connection between s_i and s_j. Dijkstra's algorithm for finding the shortest paths to some goal vertex g can be written in the following way (we show the lenght of the shortest path from vertex s to g by $V(s)$):

- For each vertex $s \in \mathcal{S}$, initialize all distances from g by $V(s) \leftarrow \infty$.
- Initialize the distance of g from itself by $V(g) \leftarrow 0$.
- Define and initialize $\tilde{\mathcal{S}} \leftarrow \mathcal{S}$.
- While $\tilde{\mathcal{S}}$ is not empty
  - $s_i \leftarrow \arg\min_{s \in \tilde{\mathcal{S}}} V(s)$
  - Remove s_i from $\tilde{\mathcal{S}}$
  - For each neighbor s_j of s_i still in $\tilde{\mathcal{S}}$: $V(s_j) \leftarrow \min(V(s_j), V(s_i) + e_{s_i,s_j})$.
- Return $V(s)$ for all $s \in \mathcal{S}$.

The output $V(s)$ of Dijkstra's algorithm is equal to the lenght of the shortest path from s to g. In this exercise, we formulate the problem of finding the shortest path as a dynamic programming problem.

a. What is the equivalent Markov Decision Process for the problem of finding the shortest paths to some goal state?
Hint: Define the goal state as an absorbing state and describe the properties of r_s^a and $p_{s_i \to s_j}^a$.

b. Compare the value iteration algorithm on the MDP of part a with Dijkstra's algorithm.

Exercise 3. Bellman operator.

Proof that the Bellman operator is a contraction.

Hint: Show the contraction with the infinity norm, i.e.

$$\|T_\gamma[X] - T_\gamma[Y]\|_\infty = \max_s |T_\gamma[X]_s - T_\gamma[Y]_s| \leq \gamma \|X - Y\|_\infty,$$

where the last inequality is to be proven. You can use the notation $Q_{sa}^X = r_s^a + \gamma \sum_{s' \in \mathcal{S}} p_{s \to s'}^a X_{s'}$ and the facts that $|\max_a Q_{sa}^X - \max_{a'} Q_{sa'}^Y| \leq \max_a |Q_{sa}^X - Q_{sa}^Y|$ and $\sum_{s' \in \mathcal{S}} p_{s \to s'}^a = 1$.

Exercise 4. Coding exercise: Value and policy iteration.

Implement value and policy iteration in python to solve the MDP from Example 1 from the lecture.