Computer Vision
hugomcp@di.ubi.pt, 2023/24

Ficha Prática 8

Reinforcement Learning

1) **Q-Learning**. Create a Pyhton script that generates random "N x N" boards, with each cell having only two possible types: "**Safe**" (penalty=1) or "**Dangerous**" (penalty=100). Generate two random cell positions (each one provided in a (row, column) format), corresponding to the current position of the agent and to the final position.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48

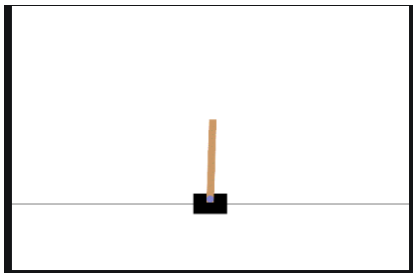
a) Using Reinforcement Learning techniques (Q-learning), design and implement a solution for moving from the current state to the final position in as few movements as possible, avoiding the Dangerous cells as much as possible.

b) Compare different strategies for defining the "state" in terms of the spatial computational cost of the algorithm (the amount of storage required) and the quality of the solutions generated (the cost of the overall path, i.e., the sum of the penalties for the cells composing the best path).

c) Observe the variations in the results, with respect to the α, γ values used in the Q-Table update formula:

$$Q_{new}(s_t, at) \leftarrow (1-\alpha)Q(s_t, at) + \alpha(Rt + \gamma\, max\, Q(S_{t+1}, a))$$

2) **Deep Q-Learning.** Consider the Cart-Pole environment from Open-AI. Suppose that a pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.



In this problem, the Action Space takes only 1 out of 2 possibilities: "0" Push cart to the left; and "1" Push cart to the right.

The observation space, that defines every possible state, is composed of four parameters.

- Cart position: [-4.8, 4.8]
- Cart velocity [-Inf, Inf]
- Pole Angle [-24º, 24º]
- Pole Angular velocity [-Inf, Inf]

Since the goal is to keep the pole upright for as long as possible, a reward of +1 for every step taken, including the termination step, is allotted.

The simulation/game ends upon one of three criteria:

1. Pole Angle is greater than ±12°
2. Cart Position is greater than ±2.4 (center of the cart reaches the edge of the display)
3. Episode length is greater than 500 (200 for v0)

Use the minimal Q-Learning implementation (due to Mike Wang, https://github.com/mswang12/minDQN/blob/main/minDQN.py), develop an automated agent that solves the Cart-Pole environment, i.e., learns to control a cart-pole device.