DATA.ML.100 Introduction to Pattern Recognition and Machine Learning
TAU Computing Sciences
Exercise - Week 7
Reinforcement learning (OpenAI Gym)

Be prepared for the exercise sessions (watch the demo lecture). You may ask TAs to help if you cannot make your program to work, but don't expect them to show you how to start from the scratch.

1. **OpenAI Gym – Frozen Lake Environment** (50 points)

   In this exercise we will use the OpenAI Gym environment (https://www.openai.com/). You may install it via Anaconda. Launch Python and type the following commands:

   ```
   $> python
   >>> import gym
   >>> env = gym.make("FrozenLake-v1", is_slippery=False)
   >>> env.reset()
   >>> env.render()
   ```

   You should see a 4×4 map where the starting location is on the top left corner and the goal at the bottom right. Read the env. description from https://www.gymlibrary.dev/environments/toy_text/frozen_lake/ Use the commands (actions) 0-3 and solve the problem manually (render after each step):

   ```
   >>> state, reward, done, info = env.step(1)
   >>> env.render()
   ```

   Let's learn the optimal policy $\pi^\star$ using the below Q-learning algorithm.

Algorithm 1 Q-learning

```
Initialize $Q(s, a)$ arbitrarily for all $s$ and $a$
for $N$ episodes do
    for $M$ episode steps do
        Choose a random action $A$
        Take action $A$, observe $R$ and $S'$
        $Q(S, A) \leftarrow R + \gamma \max_a Q(S', a)$
        $S \leftarrow S'$
        If $S$ terminal or $M$ steps taken, break
    end for
end for
```

Your task is to implement the algorithm in Python and run it. After each E episodes (e.g. 0, 10, 20, ...) test the current policy (see the evaluation code in the lecture notebook) and store the average reward.

Produce the followging graphs:

(a) Re-run the code (use e.g. $\gamma = 0.9$) 10 times and plot all performance (average reward) graphs in the same plot. They demonstrate how the performance (hopefully) improves as more training episodes are run.

(b) Re-run the experiments for the non-deterministic version of FrozenLake (is_slippery=True) and plot the same graphs. Any difference?

(c) Use the non-deterministic update rule $Q(S, A) \leftarrow Q(S, A) + \alpha(R + \gamma \max_a Q(S', a) - Q(S, A))$ with $\alpha = 0.5$ are reproduce the graph for the non-deterministic. Are they any different?