Lab Session 3

Introduction

In this lab session you'll be introduced to:
1) Value iteration backup.
2) Gym's API
3) Q-Learning.

The lab is divided in 2 parts, the first part will not be graded. For the second part of the lab, we have included 2 small examples of gym:-
gym_intro.py:- a small example of how to load 2 different envs. you are encouraged to explore them.
gym_taxi.py :- a taxi grid world that randomly choses an action.

Q-Learning

This part will be graded. The submission deadline will be Sunday midnight(8th March 2020).

It's mandatory to use numpy for all your work.

You'll work with **q_learning_deterministic.py** and implement the agent(all functions) in the frozen lake environment of the gym. FrozenLake environment is a grid world environment. The goal of this exercise is to make you familiar with q-learning. You'll code the agent class that will learn from the Environment.

Short description of the environment. (play with gym_taxi.py to understand the representation)

Grid elements
H -> Hole
F ->Frozen

Action mapping
0 -> Left
1 -> Down
2 -> Right
3 -> Up

SFFF (S: starting point, safe)
FHFH (F: frozen surface, safe)
FFFH (H: hole, fall to your doom)
HFFG (G: goal, where the frisbee is located)

short description of the functions to be implemented:-

act()

function to decide weather an agent should explore the environment or exploit from his previous knowledge. The function should decide on an action.

learn()

Function where you will have to update q-table.
Q-Learning:- The goal of Q-learning is to learn a policy, which tells an agent what action to take under what circumstances. It does not require a model of the environment, and it can handle problems with stochastic transitions and rewards, without requiring adaptations.

update_epsilon()

The function is already implemented. It decreases the exploratory beahiour of the agent.

TODO

Implement the Q-learning for 4x4 grid world, Currently all the hyperparameters have been set to solve 4x4 gridworld
Try training an agent on 8x8 grid world with different hyperparameters like epsilon decay_rate, learning rate, max_steps and number of episodes. Submit a report explaining the behaviour.
To run the 8x8 environment **uncomment** the line
kwargs = {'map_name': '8x8', 'is_slippery': True}

For more information on environment please visit
(https://gym.openai.com/envs/FrozenLake-v0/)