Reinforcement Learning Tutorial Exercises

Exercise 1

Consider the grid in Figure 1. An agent can move in either of the four directions starting from S and finishing in the goal state G.
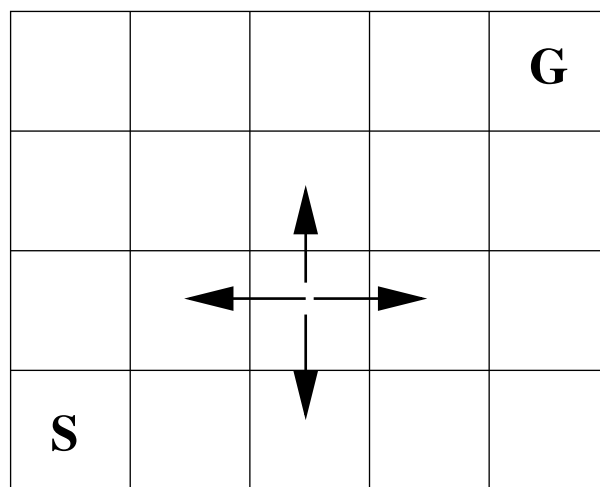

Figure 1: The grid world example for the application of reinforcement learning techniques.

1. If the reward on reaching the goal is 100, and all other rewards between state transitions are 0, write a program in a programming language of your choice (e.g. Java, Python, C++) which uses Q learning to learn the optimal policy. Assume that $\gamma = 0.9$.

2. What are the actions of the optimal policy?

Exercise 2

In Exercise 1, how does the optimal policy change if another goal state is added to the lower right corner with reward -100.

Hint: To see this, rerun your implementation with the additional goal.

Exercise 3

Consider the classic tic-tac-toe problem. The game is played with two players, **X** (user) and **O** (computer), who take turns marking the spaces in a 3×3 grid. The **X** player goes first. The

player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game (see Figure 2 below).

Figure 2: An instance of the tic-tac-toe board where the X player has won since he managed to place three 'X' in a diagonal.

We would like to develop an intelligent program which can play any human.

For this reason, we apply reinforcement learning using formula (1) in the lecture and play hundreds of games between 2 machines.

1. Write a program (Java, Python or C++) which applies formula (1) to play hundreds or thousands of games between 2 machines which learn to play tic-tac-toe by interacting with each other.

   The machines should be using an ϵ-greedy selection of an action with $\epsilon = 0.1$.

2. Test the learning of the machines by developing a program which a human can use to play against one of the "expert machines" developed in the previous task (i.e. the ones which played with each other for several hundreds/thousands of games)