Reinforcement Learning

**Homework 3: Deadline November 1, 2021**

The environment is modeled by a grid of 5x5 cell size as shown in figure below, but you are welcome to try a larger grid size (e.g., 10 x 10). The cells of the grid correspond to the states of the environment. Assume that the robot has four actions (up, down, right, left) to select at each time/iteration. You would need to define the reward for the robot to learn to find an optimal/sub-optimal way to go to the goal. Optimal here means less/minimum number of actions taken by the robot

Suggested reward (you are encouraged to define your own reward):

* Action that makes the robot tend to go out of the grid will get a reward of -1 (when the robot is in the border cells)
* Action that makes the robot reach the goal will get a reward of 100
* All other actions will get a reward of 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Starting** |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | **Goal** |

Using Q learning (Off-Policy Control) presented in lecture 10, slides# 10-11, to train the robot for this task. This Q learning technique is in Chapter 6 of the Sutton’s book.

**Requirement:**

1. Plot the action selection of the initial learning episode and the last learning episode. Something is similar to this table:

Every blue square represents a square that was visited, every move is considered a -1 reward while a move that reaches the final square will receive 100 reward. So you can calculate how many times each square was traversed along with the q-values at any given point.

Text

Description automatically generated

By the end of the learning episodes, you can see the object got a lot better at finding the optimal path with minimal moves.

Qr code

Description automatically generated

1. Show the Q table of the last learning episode

Qr code

Description automatically generated

This is also the q-values of the last episode.

1. Plot the reward of all learning episodes

Chart, line chart

Description automatically generated

1. Submit your homework with source code to Webcampus/Canvas