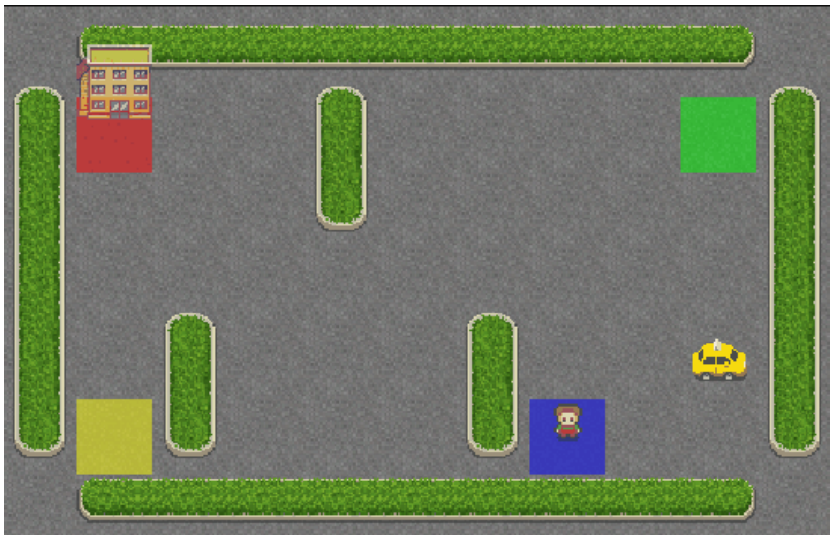
Assignment 2

Reinforcement Learning Programming - CSCN 8020

October 19, 2023

Introduction

You will work with the Taxi environment and implement QLearning and Deep Q-Learning. The way you interact with the environment will be very similar to the gym environments described in class. Therefore, most of the code we discussed is directly applicable.



The environment has 6 discrete actions, 500 discrete states, and the following rewards:

- -1 per step unless other reward is triggered.
- +20 delivering passenger.
- -10 executing "pickup" and "drop-off" actions illegally.

Action space

- 0: Move south (down)
- 1: Move north (up)
- 2: Move east (right)
- 3: Move west (left)
- 4: Pickup passenger
- 5: Drop off passenger

Observation Space

There are 500 discrete states since there are 25 taxi positions, 5 possible locations of the passenger (including the case when the passenger is in the taxi), and 4 destination locations. Destination on the map are represented with the first letter of the color.
Passenger locations:

- 0: Red
- 1: Green
- 2: Yellow
- 3: Blue
- 4: In taxi

Destinations:

- 0: Red
- 1: Green
- 2: Yellow
- 3: Blue

An observation is returned as an int() that encodes the corresponding state, calculated by $((taxi\_row * 5 + taxi\_col) * 5 + passenger\_location) * 4 + destination$

Helper Utility

To assist you in understanding the environment further, you were provided with a file (*assignment2_utils.py*) that has a few methods to allow loading the environment and printing some basic information about it. It also allows you to switch to the detailed observation description using the state scalar value.

Q-Learning [100]

Problem Statement

Implement the Q-Learning algorithm on the Taxi environment from OpenAI Gym. Train an agent to efficiently navigate and pick up passengers. Use the following hyperparameters:

- Learning Rate α: 0.1
- Exploration Factor ϵ: 0.1
- Discount Factor γ: 0.9

Tasks

- Implement the Q-Learning algorithm and train an agent on the Taxi environment.
- Report the following metrics after training:
  1. Total episodes
  2. Total steps taken per episode
  3. Average return per episode
- Make a deliberate change to the following parameters (*separately*) and use each value once.
  - Learning Rate $\alpha = [0.01, 0.001, 0.2]$
  - Exploration Factor $\gamma = [0.2, 0.3]$

Deliverables

- Python code implementing Q-Learning and running it for teh different hyperparameters. [30]
- A report (in PDF) containing the metrics, observations, and comments on the parameter change. Make sure you include appropriate plots for the previously stated metrics to support your arguments. [50]
- Based on your findings, choose what you think would be the best combination of learning rate and exploration factor and re-run the training. Report and comment on the differences observed. [20]