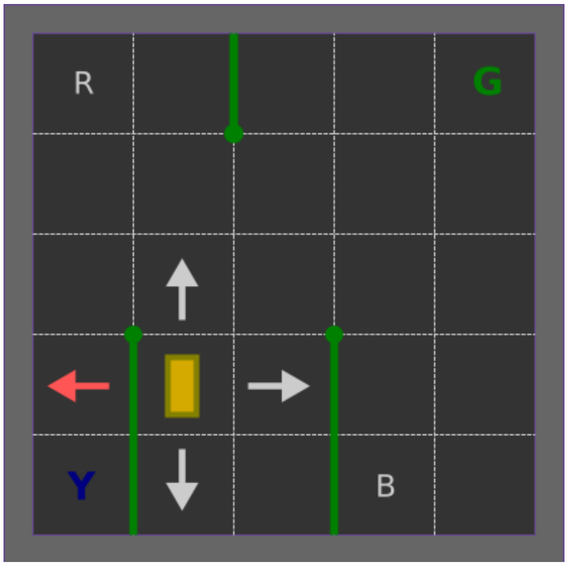
COMP 4200.201/COMP 5430.201
Artificial Intelligence

Homework 3 (25 points)

Due April 1st , Tuesday, 11:59 PM

(upload jupyter notebook with proper description)

Open Gym Taxi Game



Please read the article carefully: https://www.gocoder.one/blog/rl-tutorial-with-openai-gym/

You drive a taxi-cab. There are four locations at the four corners of the table. A passenger is waiting for the taxi at one location and you have to drive him to the designated location. You are rewarded for each move, with a low penalty when you are on travel, with a large penalty if you pick-up or drop-off the passenger at the wrong location, but you earn a big reward if you succeed.

States

In the taxi problem, a state is described by the location on the grid (a row and a column number between 0 and 4), a location to drop-off the passenger from four choices, and the passenger which can be in one of the four locations or inside the taxi. If you count well, we then have 5x5x5x4=500 possible states.

Actions:

There are 6 discrete deterministic actions:
- 0: move south
- 1: move north
- 2: move east
- 3: move west
- 4: pickup passenger
- 5: dropoff passenger

Passenger possible locations:

0: R(ed)
1: G(reen)
2: Y(ellow)
3: B(lue)
4: in taxi

Possible Destinations:
0: R(ed)
1: G(reen)
2: Y(ellow)
3: B(lue)

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

Please implement a Q-learning agent using OpenAI Gym and solve the above Taxi driving problem in python (Jupyter Notebook) where a taxi can pick up a passenger from one of the 4 (R, G, Y or B) locations and drop-off to one of the 4 (R, G, Y and B) locations.

$$\overset{\text{Q-value (for a state (S) and action(A))}}{\boxed{Q(S_t, A_t)}} \leftarrow \boxed{Q(S_t, A_t)} + \underset{\text{Learning rate}}{\boxed{\alpha}} \left[\overset{\text{Reward}}{\boxed{R_{t+1}}} + \underset{\text{Discount factor}}{\boxed{\gamma}} \overset{\text{Maximum expected future reward}}{\boxed{\max_a Q(S_{t+1}, a)}} - \boxed{Q(S_t, A_t)} \right]$$

Following the class lecture, please use the above Q-value update rule, where

Q-value for a state S_t and action A_t	$Q(S_t, A_t)$
Learning rate	α
Current transition reward	R_{t+1}
Discount factor	γ
Maximum expected future reward on the future state S_{t+1} among all possible future actions ‘a’	$\max_a Q(S_{t+1}, a)$
Epsilon: Exploration-Exploitation tradeoff	ε

Solve develop a Q-learning agent and solve the above Taxi driving problem using OpenAI Gym and Python, display (render) the solution and print final reward using:

Question 1 (5 points):

Learning rate = 0.6
Discount factor = 0.9
Design an exploration function with epsilon = 0.8, which means, 80% of time the agent will act randomly and 20% of the time, the agent will act on current policy while taking actions

Question 2 (10 points):

Learning rate = 0.9
Discount factor = 0.8
Design an exploration function with epsilon = 1, but, every episode, the epsilon will be decreasing with a rate of 0.01 (decay rate)

Question 3 (10 points):

Learning rate = 0.9
Discount factor = 0.8
Design an exploration function where each episode can be visited maximum 10 times (n=10)