Assignment No. 8
EECS 658
Introduction to Machine Learning
Due: 11:59 PM, Thursday, December 9, 2021
Submit deliverables in a single zip file to BlackBoard
Name of the zip file: FirstnameLastname_Assignment8 (with your first and last name)
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment8

Deliverables:

1. Copy of Rubric8.docx with your name and ID filled out (do not submit a PDF)
2. Python source code.
3. Screen print showing the successful execution of your Python code. (Copy and paste the output from the Python console screen to a Word document and PDF it).
4. Answer to Part 1, Question 1.
5. Answer to Part 2, Question 2.
6. Answer to Part 3, Question 3.
7. Answer to Part 3, Question 4.
8. Answer to Part 3, Question 5.
9. Answer to Part 4, Question 6.
10. Answer to Part 4, Question 7.
11. Answer to Part 4, Question 8.

Assignment:

- For all Parts, we are going to use the same modified version of the Gridtask World, which we used for Assignment 7:

▨	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	▨

- The goal of the Gridworld Task is for a robot, starting at any square in the grid, to move through the grid and end up in a termination state (grey squares) which ends the game.
- Each grid square is a state (s).
- The actions (a) that can be taken are up, down, left, or right.
- The rewards (r) will be different for each Part.
- Assume for all 4 Parts:
  - γ (Gamma) = 0.9
- Separate the output for each Part by printing out a line identifying which Part the output is from.

Part 1: RL Monte Carlo First Visit Algorithm

- Write a Python program that uses the RL Monte Carlo First Visit algorithm to develop an optimal policy (π*).
- The rewards (r) are:
    - -1 on all transitions
    - 0 terminal state
    - -1 when it hits a wall (no transition, but still a reward)
- The program should print out the N(s), S(s), and V(s) arrays with the epoch number for epoch 0 (the initial values), epoch 1, epoch 10, and the final epoch.
- The program should also print out an array showing k, s, r, γ, and G(s) for all values of k for epoch 1, epoch 10, and the final epoch.
- Determine a method for deciding when the Monte Carlo algorithm has converged.
- Question 1: Explain the convergence method and why you picked it. There is no wrong answer. You will get credit for any method you pick as long as it converges and you provide a reasonable explanation of why you picked it.

Part 2: RL Monte Carlo Every Visit Algorithm

- Write a Python program that uses the RL Monte Carlo Every Visit algorithm to develop an optimal policy (π*).
- The rewards (r) are:
    - -1 on all transitions
    - 0 terminal state
    - -1 when it hits a wall (no transition, but still a reward)
- The program should print out the N(s), S(s), and V(s) arrays with the epoch number for epoch 0 (the initial values), epoch 1, epoch 10, and the final epoch.
- The program should also print out an array showing k, s, r, γ, and G(s) for all values of k for epoch 1, epoch 10, and the final epoch.
- Use the same method for deciding when the Monte Carlo algorithm has converged as you did in Part 1.
- Question 2: Did Part 1 and Part 2 converge in the same number of epochs? Why? There is no right or wrong answer as long as you make a reasonable attempt to explain why.

Part 3: RL Q-Learning Algorithm

- Write a Python program that uses the RL Q-Learning algorithm to develop an optimal policy (π*).
- The rewards (r) are:
    - 100 on transitions to the terminal state
    - 0 on all other transitions
- Question 3: Draw a Q-Learning State Diagram for the Gridtask World above.
- Your program should first display the Q-Learning Rewards Matrix (R) as a Python array.
- The program should print out the Q-Learning Value Matrix (Q) with the episode number for episode 0 (the initial values), episode 1, episode 10, and the final episode.

- Determine a method for deciding when the Q-Learning algorithm has converged.
- Question 4: Explain the convergence method and why you picked it. There is no wrong answer. You will get credit for any method you pick as long as it converges and you provide a reasonable explanation of why you picked it.
- Question 5: Based on your final Value Matrix (Q) what is the optimal path from square 7 to the upper left-hand termination state.

Part 4: RL SARSA Algorithm

- Write a Python program that uses the RL SARSA algorithm to develop an optimal policy (π*).
- Use the same Q-Learning State Diagram for the Gridtask World that you developed for Part 3.
- Your program should first display the SARSA Rewards Matrix (R) as a Python array.
- The program should print out the SARSA Value Matrix (Q) with the episode number for episode 0 (the initial values), episode 1, episode 10, and the final episode.
- Determine a method for deciding when the SARSA algorithm has converged.
- Question 6: Explain the convergence method and why you picked it. There is no wrong answer. You will get credit for any method you pick as long as it converges and you provide a reasonable explanation of why you picked it.
- Question 7: Based on your final Value Matrix (Q) what is the optimal path from square 7 to the upper left-hand termination state.
- Question 8: Did Part 3 and Part 4 converge in the same number of episodes? Why? There is no right or wrong answer as long as you make a reasonable attempt to explain why.

Remember:

- Your Programming Assignments are individual-effort.
- You can brainstorm with other students and help them work through problems in their programs, but everyone should have their own unique assignment programs.