Artificial Intelligence and Expert Systems
Project 3 - Reinforcement Learning (updated)

Instructors:
Dr. Esmaeil Najafi
Armin Ghanbarzadeh
Reza Behbahani Nezhad

Due Date: 28 Tir, 1403

Project Task 1: Maze Environment

In this task, you will implement reinforcement learning algorithms in the "Frozen Lake" environment from OpenAI Gym (Fig. 1). The top-left block serves as the default starting position. The blue blocks represent holes, and if the agent reaches these blocks, it will receive no reward (+0 Reward). The goal block is depicted by a present box, which grants the agent a reward of +1. The permissible actions in this environment are; `0`: Move left; `1`: Move down; `2`: Move right; `3`: Move up.
For additional information regarding this environment, please refer to the Frozen Lake website, Github code.
The environment setup is provided in the file `RL_Project_TASK1.ipynb`.

Figure 1: Custom Frozen Lake environment for Task 1.

Minimum requirements of this task:

(a) Create a Q-Learning agent to find a policy for reaching a goal.

(b) Create a SARSA agent to find a policy for reaching a goal.

(c) Check the effects of changing the discount factor.

(d) Check the effects of adding uncertainty in the environment. (`is_slippery=True`).

(e) Plot accumulated reward and report the policy matrix for each experiment.

(f) Compare the results and write your own conclusion.

Project Task 2: Mobile Robot Control

In this task, you will design reinforcement learning algorithms to control a unicycle mobile robot. The objective is to reach the `goal` state. The environment setup is provided in the file `RL_Project_TASK2.ipynb`.

Robot Dynamics - Discrete Space

The robot moves in a 20×20 grid in 8 discrete directions corresponding to horizontal, vertical, and diagonal movements. The robots direction θ is between 0-7 corresponding to 45 deg increments:

$\theta = 3$	$\theta = 2$	$\theta = 1$
$\theta = 4$	(x_t, y_t)	$\theta = 0$
$\theta = 5$	$\theta = 6$	$\theta = 7$

Figure 2: Robot movement based on current direction θ

The dynamics of this robot are:

$$\theta_{t+1} = \theta_t + \omega_t$$

$$(x_{t+1}, y_{t+1}) = \begin{cases} (x_t + v_t, y_t) & \text{if } \theta_{t+1} \equiv 0 \\ (x_t + v_t, y_t + v_t) & \text{if } \theta_{t+1} \equiv 1 \\ (x_t, y_t + v_t) & \text{if } \theta_{t+1} \equiv 2 \\ (x_t - v_t, y_t + v_t) & \text{if } \theta_{t+1} \equiv 3 \\ (x_t - v_t, y_t) & \text{if } \theta_{t+1} \equiv 4 \\ (x_t - v_t, y_t - v_t) & \text{if } \theta_{t+1} \equiv 5 \\ (x_t, y_t - v_t) & \text{if } \theta_{t+1} \equiv 6 \\ (x_t + v_t, y_t - v_t) & \text{if } \theta_{t+1} \equiv 7 \end{cases}$$

The goal state can also be calculated relative to the robot moving frame. For more information, refer to the environment code.

Environment Design - Discrete Space

The unicycle mobile robot is modeled in a 2D plane with the following state and action spaces:

- **Observation Space (s):** The observation of the robot is represented as a dictionary containing:
  - `states`: A list $[0, 0, 0] \leqslant [x, y, \theta] < [20, 20, 8]$ representing the position (x, y) and orientation θ of the robot in the global frame.
  - `relative_states`: A list $[-40, -40, 0] \leqslant [e_x, e_y, e_\theta] < [41, 41, 8]$ representing the relative position (e_x, e_y) and orientation e_θ with respect to the `goal` state. (Note that the position errors can be both positive and negative. Also, since the robot is moving in a discrete grid, the relative distance could be 0.5 for cases when the robot is facing diagonal. Because of this, we multiply the distances by 2 to make them discrete again. For more information, look at the code)
- **Action Space (a):** The action space is a list $[v, w] \in \{-1, 0, +1\}^2$ of velocity and angular velocity.

Functions to Implement

First, complete the environment by implementing the `get_reward()` and `is_done()` functions based on your knowledge from the course.

The `get_reward()` function should return a reward with each action. The goal should be to encourage the robot to reach a target position (x_{goal}, y_{goal}) or a target position-orientation $(x_{goal}, y_{goal}, \theta_{goal})$.

```
def get_reward(self):
    # return: float (reward)
    x, y, theta = self.state
    ex, ey, etheta = self.relative_state
    ...
    return reward
```

The `is_done()` function should determine whether the episode has ended. This can be based on whether the robot has reached the target position or other conditions like time limits.

```
def is_done(self):
    # return: bool (True if done, False otherwise)
    x, y, theta = self.state
    ex, ey, etheta = self.relative_state
    ...
    return done
```

Minimum requirements of this task:

(a) Explain your reasoning behind defining functions `get_reward()` and `is_done()`. What other choices did you have?

(b) Implement an RL algorithm to solve the control problem for the unicycle mobile robot. Explain the approach and considerations in your solution. (The choice between Q-learning and SARSA is up to yourselves!)

(c) Explain and discuss the reason for choosing the algorithm and the functions.

(d) Save the average reward of the algorithm in each episode, plot the diagram, and discuss the results.

(e) What changes in the observation-space, `get_reward()` function and `is_done()` functions do you suggest for obstacle avoidance capability (no implementation needed - implementation has bonus score). Provide reasons to support your ideas.

Bonus Score Section: Deep Reinforcement Learning in Continuous space

For the bonus part of the assignment, you can implement deep RL algorithms for the unicycle robot in continuous spaces using popular frameworks like Stable-Baselines3. The environment setup is provided in the file `RL_Project_TASK2_Bonus.ipynb`.
Some of the algorithms you can consider are: Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG) and Trust Region Policy Optimization (TRPO). You are free to choose any deep RL algorithm and framework of your choice. Make sure to include the implementation details and learning curves.

Robot Dynamics - Continuous Space

The dynamics of the unicycle robot are given by the following equations:

$$\dot{\theta} = w$$
$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$

Where v is the linear velocity and w is the angular velocity. The discrete-time equations can be derived from the continuous-time equations using a time step of T_s:

$$\theta_{t+1} = \theta_t + wT_s$$
$$x_{t+1} = x_t + vT_s\cos(\theta_t)$$
$$y_{t+1} = y_t + vT_s\sin(\theta_t)$$

The goal state can also be calculated relative to the robot moving frame with the transformation (Fig. 3):

$$\begin{bmatrix} e_\theta \\ e_x \\ e_y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \cos(\theta_t) & \sin(\theta_t) & 0 \\ -\sin(\theta_t) & \cos(\theta_t) & 0 \end{bmatrix} \begin{bmatrix} \theta_{goal} - \theta_t \\ x_{goal} - x_t \\ y_{goal} - y_t \end{bmatrix}$$

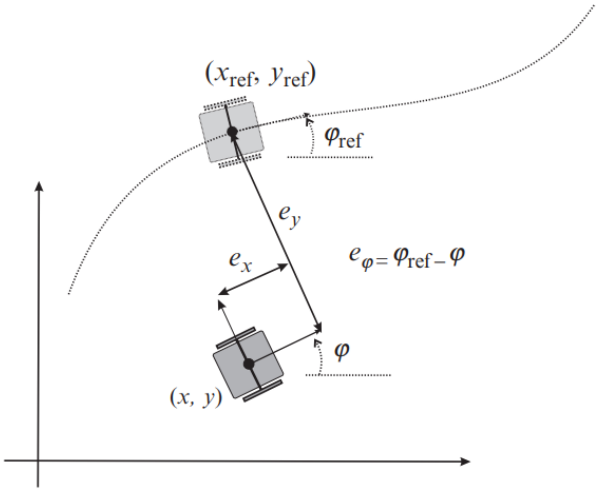

Figure 3: The illustration of the error in relative coordinates.

Environment Design - Continuous Space

The unicycle mobile robot is modeled in a 2D plane with the following state and action spaces:

- **Observation Space (s):** The observation of the robot is represented as a dictionary containing:
  - `states`: A list $[x, y, \theta]$ representing the position (x, y) and orientation θ of the robot in the global frame.
  - `relative_states`: A list $[e_x, e_y, e_\theta]$ representing the relative position (e_x, e_y) and orientation e_θ with respect to the `goal` state.
- **Action Space (a):** The action space is a list $[v, w]$ of velocity and angular velocity in $[-1, 1]^2$.

Important Notes

- Please submit your homework assignment as a zip file containing **(a)** a PDF report (analysis, results, methodology, ...), and **(b)** code files necessary to reproduce your results.
- All grading will be based on the content of the PDF report. Make sure to include and explain your code in the report.
- Please make sure to submit your solutions by the due date. No late submissions will be accepted.
- Assignments are to be completed individually. Any similarities between assignments will be subject to reduced grades.
- If you have any questions, feel free to ask.

Good Luck!