AI 3603 Artificial Intelligence: Principles and Techniques
Homework 2 Due Oct. 26th 18:00

Adhere to the Code of Academic Integrity. You may discuss background issues and general strategies with others and seek help from course staff, but the implementations that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is never OK for you to see or hear another student's code and it is never OK to copy code from published/Internet sources. Moss (Measure Of Software Similarity) will be used for determining the similarity of programs to detect plagiarism in the class (`https://theory.stanford.edu/~aiken/moss/`) If you encounter some difficulties or feel that you cannot complete the assignment on you own, discuss with your classmates in Discussion forum on Canvas, or seek help from the course staff.

You can complete this homework ***individually*** or ***in a group of two***. Please submit your assignment following the instructions summarized in Section. 6.

1 Reinforcement Learning in Cliff-walking Environment [50 pts]

In this assignment, you will implement Reinforcement Learning agents to find a safe path to the goal in a grid-shaped maze. The agent will learn by trail and error from interactions with the environment and finally acquire a policy to get as high as possible scores in the game.

1.1 Game Description

Suppose a 12×4 grid-shaped maze in Fig. 1. The purple rectangle represents the starting point and the green rectangle represents the exit. The yellow circle denotes your current position. You can move upward, downward, leftward and rightward in this game. You will stay in place if you try to move outside the maze. You are asked to reach the goal through the safe region(gray) and avoid falling into the cliff(black). Reaching the exit gives a reward +10 and falling into the cliff gives a reward -100, and both of the two cases terminate the current episode. Otherwise, a living cost(-1) is given.

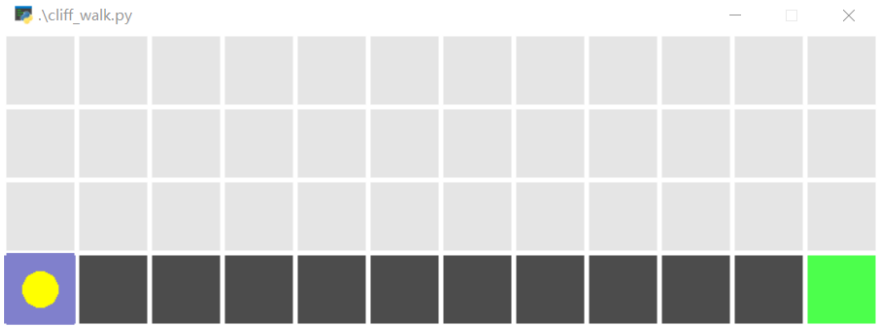


Figure 1: The cliff-walking environment

The environment is implemented in **gym_gridworld.py**. Please *do not* modify it. The state space and action space are briefly described as follows and you may learn more details in the code.

State(Integer): The state s_t is an integer, which represents the current coordinate (x, y) of the agent, i.e. $s_t = x + 12 * y$. More details are given in **gym_gridworld.py** (read to _state_to_xy and _xy_to_state functions for more details).

Action(Integer): A discrete variable, where 0, 1, 2, 3 represent move leftward, rightward, upward, downward respectively.

1.2 Game with Sarsa and Q-Learning

You are asked to implement agents based on Sarsa and Q-Learning algorithms. Please implement the agents in **agent.py** and complete the training process in **cliff_walk_sarsa.py** and **cliff_walk_qlearning.py** respectively. An agent with a random policy is provided in the code. You can learn how to interact with the environment through the demo and then write your own code.

Hint: Take **cliff_walk_sarsa.py** as an example:

- Line 29: more parameters need to be utilized to construct the agent, such as learning rate, reward decay γ, ε value, and ε-decay schema.
- Line 49: the agent needs to be provided with some experience for learning.

Hint: In **agent.py**:

- You need to implement ε-greedy with ε value decay in the **choose_action** function.
- Functions given in the template need to be completed. You can also add other utility functions as you wish in the agent classes.

Hint: You should keep a balance between exploration and exploitation by tuning ε value carefully. In the early stage of the training, exploration should be encouraged to get familiar with the environment. With the advancement of training, exploration may be reduced for the convergence of the policy.

1.3 Result Visualization and Analysis

Result Visualization: You are required to visualize the training process and the final result according to the following requirements:

1. Plot the episode reward during the training process.
2. Plot the ε value during the training process.
3. Visualize the final paths found by the intelligent agents after training.

Result Analysis: You are required to analyze the learning process based on the experiment results according to the following requirements:

1. Analyze the learning process of Sarsa and Q-learning respectively;
2. You may find that there exists a little difference between the paths found by Sarsa and Q-learning. Please describe the difference in the report and analyze the reason in detail.

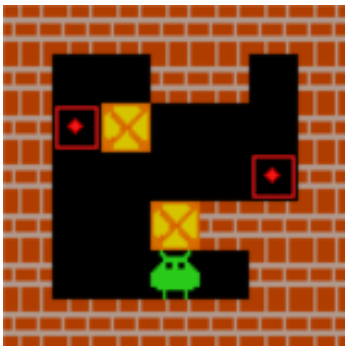
2 Reinforcement Learning in the Sokoban Game [50 pts]

In this part, you are asked to implement intelligent agents to play the Sokoban game utilizing Sarsa, Q-learning, and dyna-Q algorithms. You will have a deeper understanding on the model-based RL algorithms and the explore-exploit dilemma.

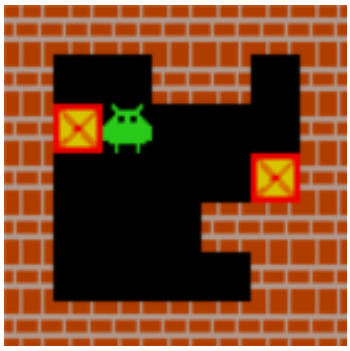
2.1 Game Description

As shown in Fig. 2, Sokoban game is a classic video game. The game is a transportation puzzle, where the player has to push all boxes in the room on the storage targets. The possibility of making irreversible mistakes makes this game a bit challenging for RL agents.

In this task, a 7×7 room with two boxes is utilized, as illustrated in Fig. 2(a). The environment utilized in this task is static, i.e. the room settings including box coordinates, target positions, and agent positions are the same after each reset. In this game, you can move upward, downward, leftward and rightward. Pushing two boxes onto the targets gives a reward +10; Pushing one box on a target gives a reward +1, while pushing



(a) Sokoban initial state



(b) Success State

Element	Element state	Graphic
Wall	Static	
Floor	Empty	
Box Target	Empty	
Box	Off Target	
Box	On Target	
Player	Off Target	
Player	On Target	

(c) Room elements

Figure 2: The sokoban environment in this task.

a box off the target gives -1. In addition, a reward of -0.1 is given for each step as living cost. The game terminates when two boxes are pushed onto the target, or 100 steps are exhausted before success.

The environment is implemented in **gym_sokoban** folder. Please *do not* modify it. The state space and action space are briefly described as follows and you may learn more details in the code.

State(Array): The state s_t is an array composed of 6 integers, where the first two integers represent current coordinate of the agent, and the last four integers represent current coordinates of two boxes in the room.

Action(Integer): A discrete variable and 0, 1, 2, 3 represent move upward, downward, leftward, rightward respectively.

2.2 Game with RL algorithms

You are asked to solve this task utilizing Sarsa, Q-Learning, and Dyna-Q algorithms respectively. Please implement the intelligent agents in **agent.py** and complete the training process in **sokoban_sarsa.py**, **sokoban_qlearning.py**, and **sokoban_dynaq.py** respectively. Actually, Sarsa and Q-Learning agents implemented in Section 1 can be utilized for this task.

Hint: Different from Q-Learning and Sarsa, Dyna-Q needs to collect experience and learn in the "imagined" world. Thus, you should construct a database in the **DynaQAgent** class to store experience. Besides, **sokoban_dynaq.py** should be modified to enable the agent to learn by sampling from historical experience.

You can refer to Section. 1.2 for more hints.

2.3 Result Visualization and Analysis

Result Visualization: You are required to visualize the training process and the final result according to the following requirements:

1. Plot the episode rewards during the training process.
2. Plot the ε value during the training process.
3. Visualize the final result of three agents. Recording videos of the final result is recommended. An instrcution to record a video with gym library is given in **gym_recoder_instruction.py**, while other recording tools or other visualization methods are also acceptable.

Result Analysis: You are required to analyze the learning process based on the experiment results according to the following requirements:

1. Analyze the learning process of three agents respectively;
2. Compare the learning speed of three agents. Please analyze the reason for the difference among them.

2.4 Explore-Exploit Dilemma

You may have found the importance of exploration-exploitation dilemma in the Reinforcement Learning through previous experiments. In this section, you are required to analyze the influence of different exploration schemes on the learning speed and result.

1. Conduct experiments in the *Sokoban* environment utilizing *any one* RL algorithm with different ε values and ε-decay schemes. Try to summarize the relationship between different exploration schemes and learning speeds/results.

2. Actually, there exists lots of other exploration strategies except ε-greedy. You are asked to find and learn one new exploration method, such as Upper Confidence Bound(UCB).

3. **(Bonus +5pt)** Implement the exploration strategy you have found in the *Sokoban* environment. You can add new agent class in the **agent.py** if necessary. The corresponding training process should be implemented in **sokoban_new_exploration.py**.

3 Installation

You can follow the tutorial in this section to install the environment on Linux, Windows or macOS, and we strongly recommend you to use Linux system.

3.1 Install Anaconda

Open the address `https://www.anaconda.com/distribution/` and download the installer of **Python 3.x version**(3.6 recommended) for your system.

- For Linux

  bash Downloads/Anaconda3-2021.05-Linux-x86_64.sh

- For Windows

  Open the .exe file and follow the installer steps.

- For macOS

  Open the .pkg file and follow the installer steps.

3.2 Install Required Environment

After installing anaconda, open a terminal (Linux and macOS) or Anaconda Prompt (Windows) and create an environment for Gym:

conda create python=3.6 --name gym

Then activate the environment

conda activate gym

Install numpy

pip install numpy

Install gym and some dependencies

pip install gym

pip install pyglet

pip install imageio

Install matplotlib

pip install matplotlib

If you want to record videos utilizing gym Monitor library, ffmpeg is needed:

- For most Ubuntu and its variants: **sudo apt-get install ffmpeg**
- For Windows: You can record videos utilizing gym immediately, if you are lucky enough.
- For macOS: **brew install ffmpeg**

4 Code, Demo Video, and Report

Code: You can edit the code between "##### START CODING HERE #####" and "##### END CODING HERE #####". Please **DON'T** modify other parts of the code.

Demo Video: Videos for Section. 2.3 should be in .mp4 format and a 10MB max in total(you can compress/speed up the videos). Videos are named as **sokoban_sarsa.mp4**, **sokoban_qlearning.mp4**, and **sokoban_dynaq.mp4**. All the videos are put into a folder called **videos**.

Report: Summarize the process and results of the homework, including but not limited to:

- Description of the experiment environment, including operating system version, python version, and other python dependencies in your code;
- The description and implementation of RL algorithms;
- Visualization and analysis of the experiment result;
- Description, implementation(Optional, Bonus), and analysis of different exploration strategies.

Submission File List: You can add other files if necessary. Please describe the function of each submitted file in your report.

- **agent.py:** Implementaion of RL agents;
- **cliff_walk_qlearning.py:** train Q-Learning in Cliff-walking environment.
- **cliff_walk_sarsa.py:** train Sarsa in Cliff-walking environment.
- **sokoban_qlearning.py:** train Q-Learning in Sokoban environment.
- **sokoban_sarsa.py:** train Sarsa in Sokoban environment.
- **sokoban_dynaq.py:** train Dyna-Q in Sokoban environment.
- **sokoban_new_exploration.py:** (Optional, Bouns)train RL agent with new exploration method in Sokoban environment.
- **videos (folder):** Demo videos recorded.
- **gym_sokoban (folder):** (Don't modify)Sokoban game environment.
- **gym_gridworld.py:** (Don't modify)Cliff-Walking game environment.

5 Discussion and Question

You are encouraged to discuss your ideas, ask and answer questions about this homework. A new discussion forum is opened on Canvas. If you encounter any difficulty with the assignment, try to post your problem for help. The classmates and the course staff will try to reply.

6 Submission instructions

1. Zip all your program files, experiment result(including video files), and report file **HW2_report.pdf** to a file named as **HW2_ID1_name1_ID2_name2.zip** for a group, or **HW2_ID_name.zip** for individual. If you are working as a group, both two member names and IDs should be written on the cover of the report, and submit only one copy.

2. Upload the file to the homework 2 page on the Canvas.