Reinforcement Learning

Semester A - Final Project - 2025

Objective

The main goal of this project is to summarize and integrate the key theoretical and practical topics covered in the second part of the course. You will apply deep reinforcement learning concepts to solve the **MultiRoom** environment from the **MiniGrid** library.
You can find a demonstration of how to use the environment Here.

Background

MiniGrid and Gymnasium

MiniGrid is a library providing a variety of 2D environments designed to benchmark reinforcement learning agents. For this project, you will use **Gymnasium** (the updated and maintained fork of Gym) instead of Gym. Gymnasium introduces several improvements, including:

- Better support for type annotations.
- Enhanced error handling.
- Improved compatibility with modern libraries.

For more information, refer to the Gymnasium API documentation.

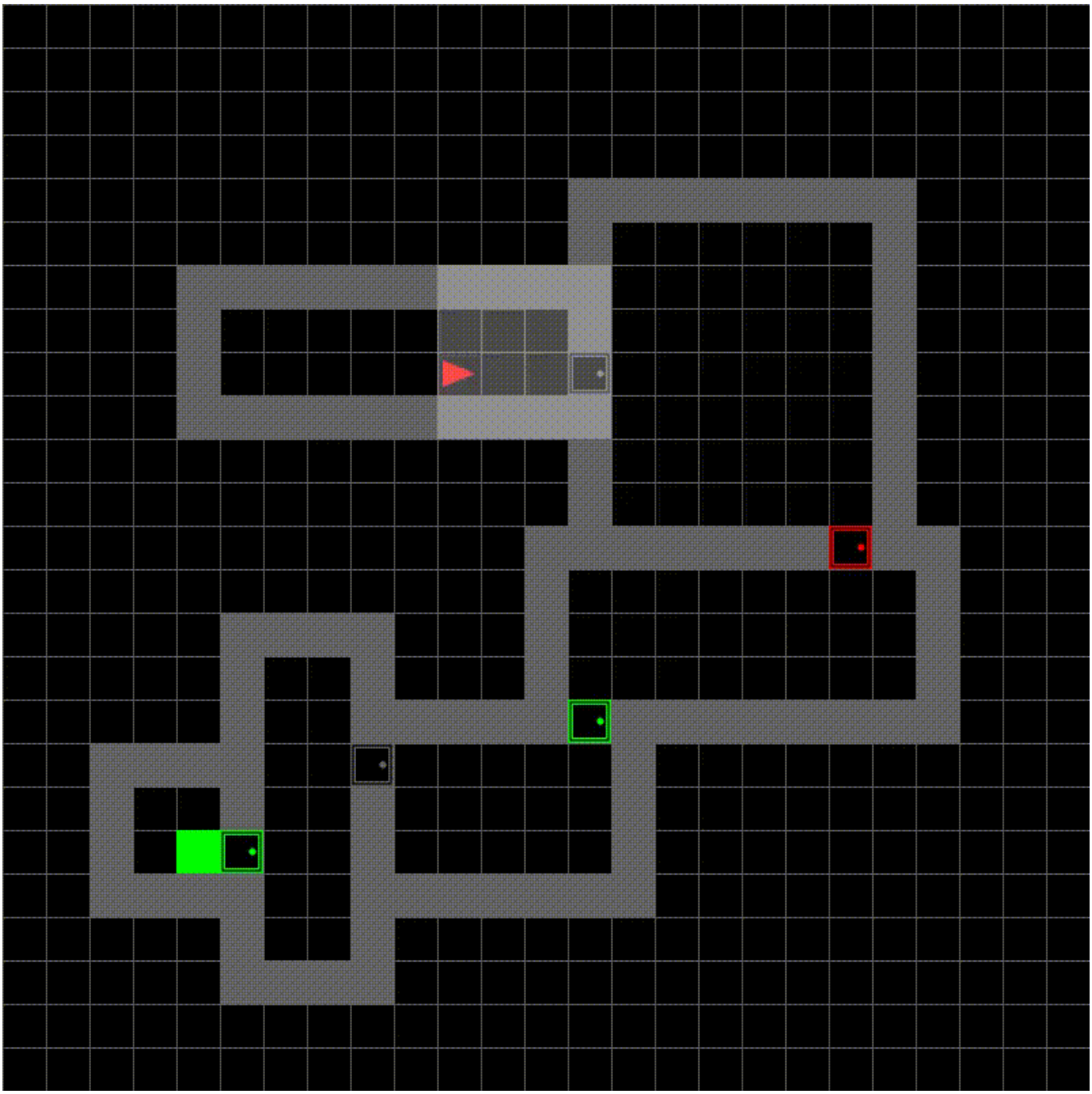
MultiRoom Environment

The **MultiRoom** environment challenges the agent to navigate through a series of connected rooms to reach a designated goal. In this task, the environment operates in Partial Observation mode, meaning the agent cannot see the full layout of the environment at once. Instead, its view is limited to the current room and the area visible from its position and orientation. The **RGBImgPartialObsWrapper** handles this partial observation dynamically, updating the agent's view as it moves or turns. For more details, refer to the Wrappers for Observations section in the provided notebook.
For detailed information about the action space, observation space, rewards, and other environment properties, refer to:

- **MiniGrid Documentation**
- **MultiRoom Environment Details**

Assignment Task

Your primary task is to solve the **MiniGrid-MultiRoom-N6-v0** environment, which includes navigating through **six large** connected rooms.



Key Difference from Mid-Semester Project

In this project, the agent's observations are **image-based (pixel observations)**, unlike the mid-semester project, which utilized structured feature-based observations.

TIP: Use the following simpler environments as stepping stones:

- **MiniGrid-MultiRoom-N2-S4-v0**: Two small rooms.
- **MiniGrid-MultiRoom-N4-S5-v0**: Six small rooms.

These simpler environments are designed to help you progressively refine your agent. Start small, debug, and optimize your training process before scaling up to the more challenging **MiniGrid-MultiRoom-N6-v0** environment.

Example initialization code for the two rooms environment:

```
import gymnasium

# Example for MiniGrid-MultiRoom-N2-S4-v0
env_small = gymnasium.make("MiniGrid-MultiRoom-N2-S4-v0", render_mode=render_mode,
highlight=highlight) # fo the smaller environment
env_mid = gymnasium.make("MiniGrid-MultiRoom-N4-S5-v0", render_mode=render_mode,
highlight=highlight) # fo the intermediate environment
```

Project Requirements

1. Algorithm Selection and Explanation

You are required to demonstrate knowledge from the second part of the course. Implement multiple reinforcement learning algorithms, compare their differences, and analyze the results. For each algorithm, include:

- **Algorithm Overview**:
  - Provide a brief explanation of the algorithm, its theoretical foundation, and why it is suitable for the task.
  - Highlight how the algorithm addresses the environment's challenges.
- **Hyperparameters**:
  - Discuss the key hyperparameters (e.g., learning rate, epsilon, replay buffer size, target network update rate) and how they were selected/tuned for the environment.
  - Explain the impact of these hyperparameters on performance.
- **Deep Learning Architecture**:
  - Describe the architecture used (e.g., layers, activation functions) and justify your design choices.
  - Include details on how you validated your architecture during experimentation (e.g., metrics, visualizations).
- **Comparison of Algorithms**:
  - Compare algorithms based on performance, learning efficiency, and convergence.
  - Discuss their differences in handling exploration, exploitation, and partial observability.

2. Performance and Analysis

- Solve the **MiniGrid-MultiRoom-N6-v0** environment in as few episodes as possible.
- Discuss the advantages and disadvantages of each approach relative to the MiniGrid mission.
- Provide detailed analyses and highlight key insights and lessons learned.

3. Visualization and Graphs

- Include a variety of graphs to illustrate the learning process, such as but not limited to :
  - Rewards over episodes for each algorithm.

- Training curves showing key metrics (e.g., loss, reward convergence).
- Comparative graphs for algorithms and setups.
- Graphs illustrating the exploration-exploitation trade-off.
- Graph showing the average number of steps to complete an episode over the last 25 episodes during inference.

4. Preprocessing and Exploration-Exploitation

- You can include preprocessing steps applied to the input image and justify their necessity.
- Discuss how you managed the exploration-exploitation trade-off for each algorithm and environment.

5. Evaluation and Reporting

- Provide a balanced evaluation of your approaches, highlighting both strengths and weaknesses.
- Discuss challenges faced during implementation and how you addressed them.

Note: While implementations of these algorithms exist online, focus on demonstrating your thought process, considerations, and problem-solving strategies.

Submission Guidelines

Code Submission

1. **Notebooks**: Submit two Colab notebook links:
    - **Notebook 1**: Includes all your work related to training, such as preprocessing, model setup, and training loops.
    - **Notebook 2**: Focused on evaluation. It must:
        - Demonstrate how to load the trained weights and initialize the environment.
        - Evaluate your policies and save a video from a random episode.
2. **Code Requirements**:
    - Write clean, modular code with proper comments and formatting.
    - Include videos showing the agent's behavior during training and after convergence.

Report Guidelines

1. **Quality Matters**: The majority of your grade will be based on the quality of your report. Ensure it is concise, accurate, and well-structured.
2. **Page Limit**: The body of the report must be maximum of 4 pages (excluding graphs).
3. **Content Requirements**:
    - Reference notebook content in your report. Unreferenced content in notebooks will not be graded.
    - Place all graphs at the end of the report, properly labeled and referenced within the text (e.g., "As shown in Figure 1...").
4. **File Naming**: The report must be submitted as a PDF file with the format: `report_ID1_ID2.pdf`.

Additional Submissions

In the submission box, include:

1. **details.txt**: A text file containing:
    - Names and IDs of both team members.
    - Links to the Colab notebooks.
2. **report_ID1_ID2.pdf**: Your report.
3. **explainer.txt**: Instructions on how to operate your notebooks.

Important Notes

- **Team Size**: The project must be completed in teams of two students.
- **No Prebuilt RL Libraries**: All algorithms must be implemented from scratch.
- **Academic Integrity**: Plagiarism or unauthorized sharing of code will result in penalties. If you use any external resources, give proper attribution.

Deadlines

- **Final Submission**: February 25 at 23:59.
- **Late Submissions**: Will not be accepted.

Good luck, and we look forward to seeing your results!