

Homework #5

Reinforcement Learning



Wen-Huang Cheng (鄭文皇)

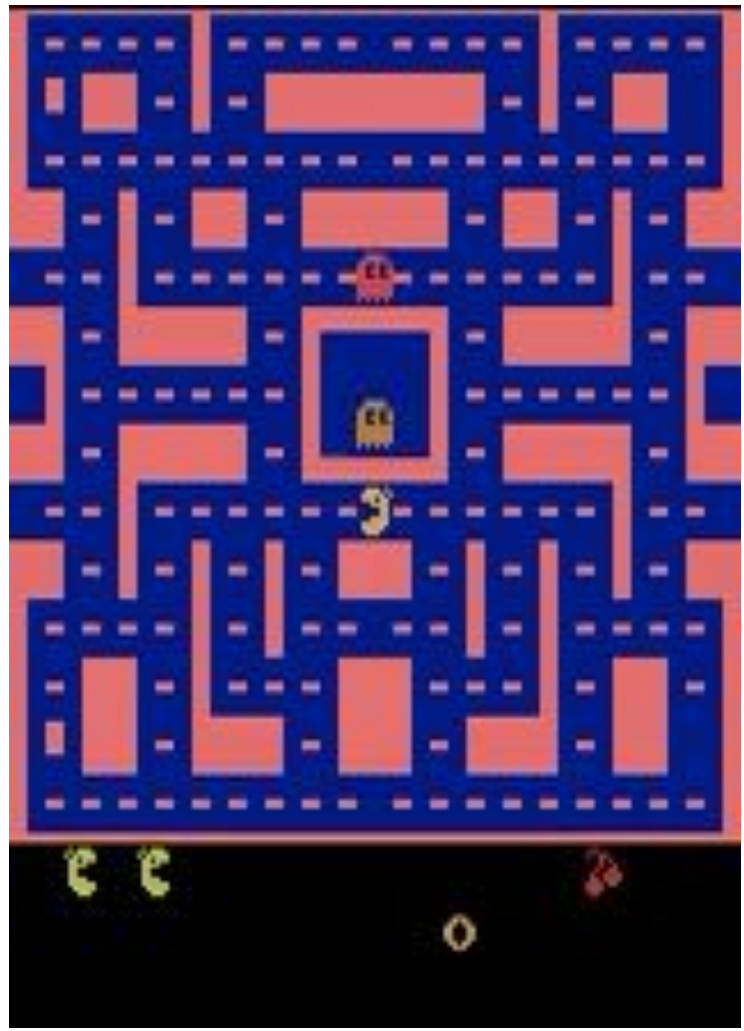
National Taiwan University

wenhuang@csie.ntu.edu.tw

- **This assignment aims to use the OpenAI Gymnasium platform to create a Pacman game environment and apply reinforcement learning techniques to play the game.**



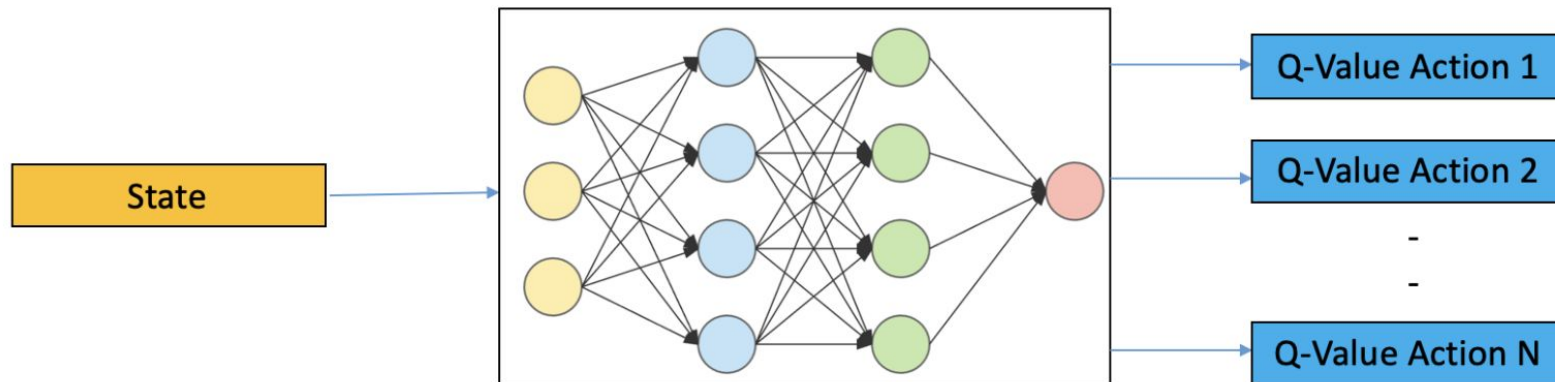
[*Announcing The Farama Foundation - The future of open source reinforcement learning | The Farama Foundation*](#)





- **Deep Q-Network (DQN)**

- https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html
- <https://huggingface.co/tasks/reinforcement-learning>
- <https://huggingface.co/learn/deep-rl-course/unit3/deep-q-network>



Deep Q Learning

| Category | Files | Description |
|---------------------------------|-----------------------------------|---|
| Files you'll edit | <code>`pacman.py`</code> | Use this file to train the model and validate its performance. |
| | <code>`rl_algorithm.py`</code> | Implements the DQN (Deep Q-Network) agents. |
| Files you might want to look at | <code>`pacman-intro.ipynb`</code> | Provides a basic introduction to the MsPacman environment in OpenAI Gym. |
| | <code>`custom_env.py`</code> | Wraps the image data for the MsPacman environment to facilitate interaction with the learning algorithms. |
| Supporting files you can ignore | <code>`utils.py`</code> | Auxiliary functions and utilities that support the main application files but are not essential for understanding the core functionality. |

- The code for this project is organized into several Python files. You will need to read and understand some of these files to successfully complete the assignment.
- In **`pacman.py`**, you are required to implement the **`train`, `validation`, and `evaluate` functions** to train the model.
- In **`rl_algorithm.py`**, you will need to complete the **`PacmanActionCNN`** to predict actions based on image inputs. Additionally, you must implement the initialization and other functions in the **`DQN`** to facilitate model training.

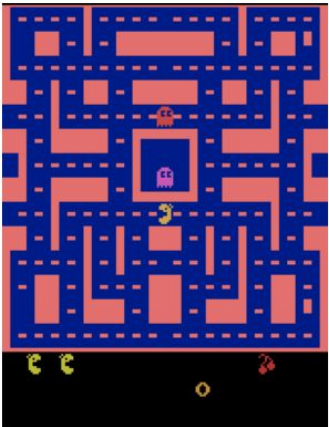
- For more detailed documentation, see the [Gymnasium](#) .
- We also provide a simple introduction in the provided code, named `pacman_intro.ipynb`.

| | |
|-------------------|--|
| Action Space | Discrete(9) |
| Observation Space | Box(0, 255, (210, 160, 3), uint8) |
| Import | <code>gymnasium.make("ALE/MsPacman-v5")</code> |

Observations

Atari environments have three possible observation types: `"rgb"`, `"grayscale"` and `"ram"`.

- `obs_type="rgb" -> observation_space=Box(0, 255, (210, 160, 3), np.uint8)`



Actions

MsPacman has the action space of `Discrete(9)` with the table below listing the meaning of each action's meanings. To enable all 18 possible actions that can be performed on an Atari 2600, specify `full_action_space=True` during initialization or by passing `full_action_space=True` to `gymnasium.make`.

| Value | Meaning | Value | Meaning | Value | Meaning |
|-------|---------|-------|-----------|-------|----------|
| 0 | N00P | 1 | UP | 2 | RIGHT |
| 3 | LEFT | 4 | DOWN | 5 | UPRIGHT |
| 6 | UPLEFT | 7 | DOWNRIGHT | 8 | DOWNLEFT |

■ **Coding (80%)**

- Deep Q-Network
- README.md
- Submission folder
- The score for each problem is detailed in p7-8.

■ **Report (20%)**

- Must be submitted in PDF format only.
- The score for each problem is detailed in p9.

■ Deep Q-Network

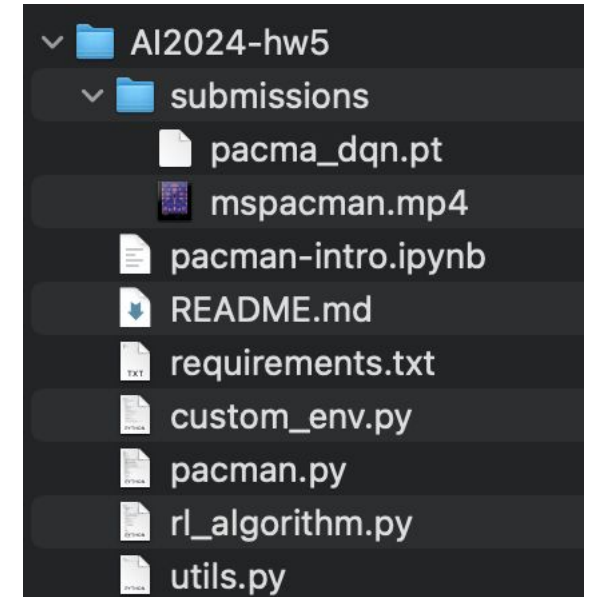
- Implement the Deep Q-Network, train and evaluate your agent in OpenAI Gymnasium ALE/MsPacman-v5.
- To receive full points, the evaluated reward must exceed 2000.
- No partial points will be given.
- *Grading: `python pacman.py --eval --eval_model_path "MODEL_PATH"`*

■ README.md

- Please provide a detailed description of how your environment is set up, as well as how to execute the training and evaluation of your code.
- The submission for reproducibility will be run according to the README. If it fails to run, it will receive a zero score. Minor modifications to the path are allowed for resubmission, but will be subject to a 10% penalty.

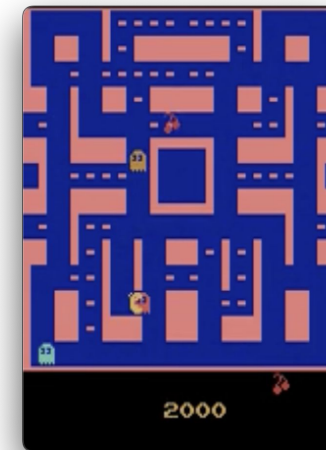
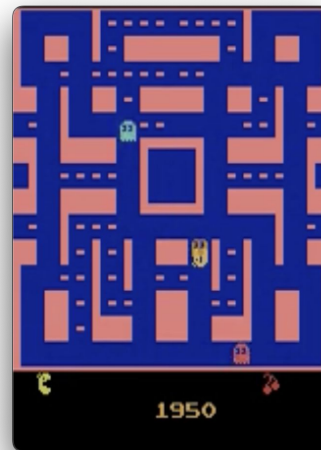
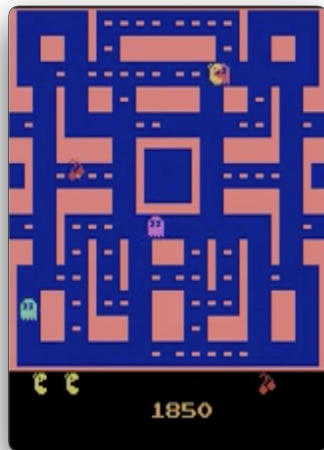
■ Submissions (folder)

- Please ensure that your "submission" folder includes the following items. Failure to include any of the required files will result in a 20-point penalty:
 - Model Weights: Include the saved model weights.
 - Evaluation Video: A video demonstrating the model in action.

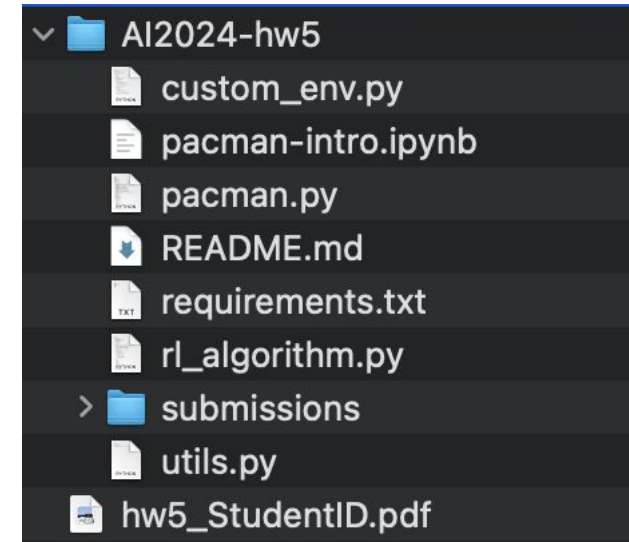


- Describe the Deep Q-Network (7%)
- Describe the architecture of your PacmanActionCNN (7%)
- Plot your training curve, including both loss and rewards. (3%)
- Show screenshots from your evaluation video (3%)

("ALE/MsPacman-v5 has a total of three chances. Display the reward (score) each time you are caught.)."



- Deadline: 2024/05/22 (Wed.) 23:59
- Submit to NTU COOL
- Your submission should include the following files:
 - hw5_<student_id>.zip
 - hw5_<student_id>.pdf
 - AI2024-hw5 (all the code we provided should be included.)
 - Do not put report.pdf into AI2024-hw5 folder
- *Note: hw5_<student_id> is an example format. For instance, if your student ID is r1234567, then the file name should be hw5_r1234567.*





```
opencv-python  
gymnasium[atari]  
gymnasium[accept-rom-license]  
numpy  
matplotlib  
opencv-python  
imageio-ffmpeg  
imageio  
torch  
torchvision
```

UPDATED:



Any Question

ai.ta.2024.spring@gmail.com