

Homework #5

Reinforcement Learning



Wen-Huang Cheng (鄭文皇)

National Taiwan University

wenhuang@csie.ntu.edu.tw

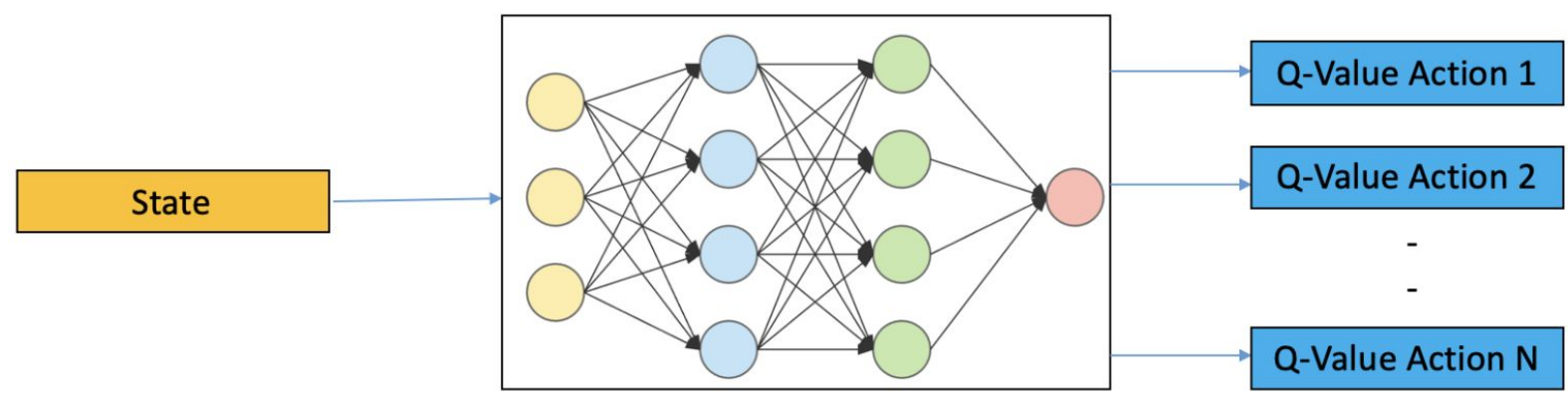
- **This assignment aims to use the OpenAI Gymnasium platform to create a Pacman game environment and apply reinforcement learning techniques to play the game.**



[*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.

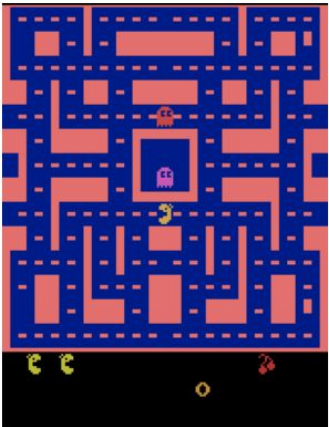
- For more detailed documentation, see the [Gymnasium](#) .
- We also provide a simple introduction in the provided code, named `pacman_intro.ipynb`.

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

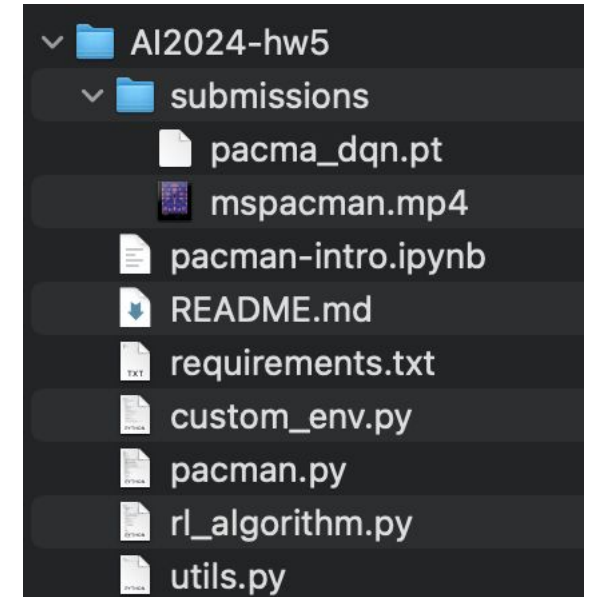
- Implement the Deep Q-Network, train and evaluate your agent in OpenAI Gymnasium ALE/MsPacman-v5.
- To receive full points, the evaluated reward must exceed **1500**.
- No partial points will be given.
- *Grading: `python pacman.py --eval --eval_model_path "MODEL_PATH"`*

■ 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.

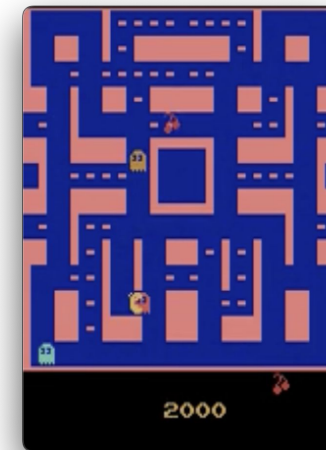
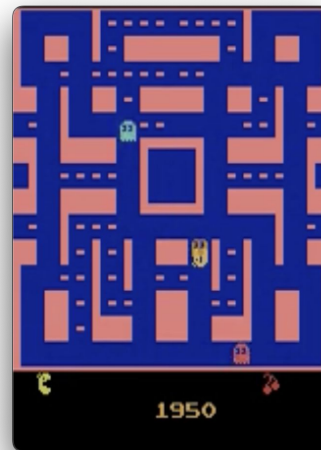
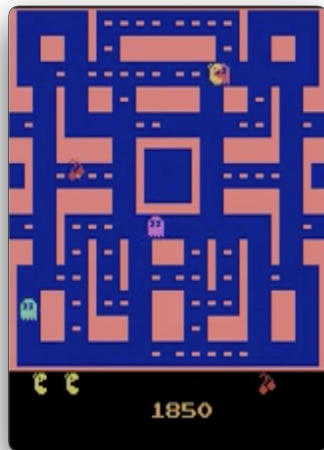
■ Submissions (folder)

- 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%)

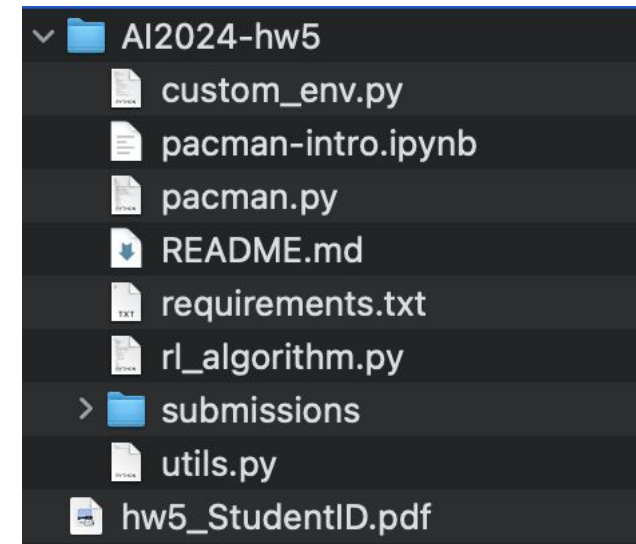
("ALE/MsPacman-v5 has a total of three chances. Display the reward (score) each time you are caught.)."





- Deadline: **2024/05/25 (Fri.) 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.*





UPDATE

- 請各位同學注意，作業五的文件中標示為 YOUR_CODE_HERE 的部分需要你們填寫有效的程式碼以使其運行。
- 你們會發現有些地方使用了 "self.optimizer.YOUR_CODE_HERE" 的格式。使用引號是為了避免在你們尚未替換 YOUR_CODE_HERE 之前編譯出錯。完成代碼後，請記得移除這些引號，使代碼能正常執行。
- 另外，除了標有 "**** YOUR CODE HERE ****" 的部分需要同學們自行填寫外，其他部分也都可以進行修改以適應作業需求。然而，請注意，實作 DQN 的部分必須自行編寫，不應使用任何現成的套件。

```
26 YOUR_CODE_HERE = "**** YOUR CODE HERE ****"
```

utils.py line 26 的 “**** YOUR CODE HERE ****” 不用理它。

“**** YOUR CODE HERE ****” 的範例：

```
# get q-values from network
next_q = YOUR_CODE_HERE
# td_target: if terminated, only reward, otherwise reward + gamma * max(next_q)
td_target = YOUR_CODE_HERE
# compute loss with td_target and q-values
loss = YOUR_CODE_HERE

# initialize optimizer
"self.optimizer.YOUR_CODE_HERE"
# backpropagation
YOUR_CODE_HERE
# update network
"self.optimizer.YOUR_CODE_HERE"
```

```
# 應移除引號並填寫正確代碼：
td_target = YOUR_CODE_HERE
# 範例：
td_target = reward
```

```
# 應移除引號並填寫正確代碼：
"self.optimizer.YOUR_CODE_HERE"
# 範例：
self.optimizer.state_dict()
```

- 為了協助不熟悉 CNN 的同學，以下是基本設定參考。
 - self.conv1 / self.conv2: <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html>
 - self.fc1 / self.fc2: <https://pytorch.org/docs/stable/generated/torch.nn.Linear.html#torch.nn.Linear>

```
class PacmanActionCNN(nn.Module):
    def __init__(self, state_dim, action_dim): ...

    def forward(self, x):
        """ YOUR CODE HERE """

        x = F.relu(self.conv1(x))
        x = F.relu(self.conv2(x))
        x = x.view(-1, self.in_features)
        x = self.fc1(x)
        x = self.fc2(x)
        return x
```



opencv-python==4.8.1.78

swig==4.2.1

gymnasium==0.29.1

gymnasium[atari, accept-rom-license]

numpy==1.26.4

matplotlib==3.8.4

imageio-ffmpeg

imageio==2.34.1

torch

tqdm

UPDATED:

chardet



Any Question

ai.ta.2024.spring@gmail.com