

---

---

# Machine Learning HW12

ML TAs

[mlta-2023-spring@googlegroups.com](mailto:mlta-2023-spring@googlegroups.com)

---

---

# HW Content

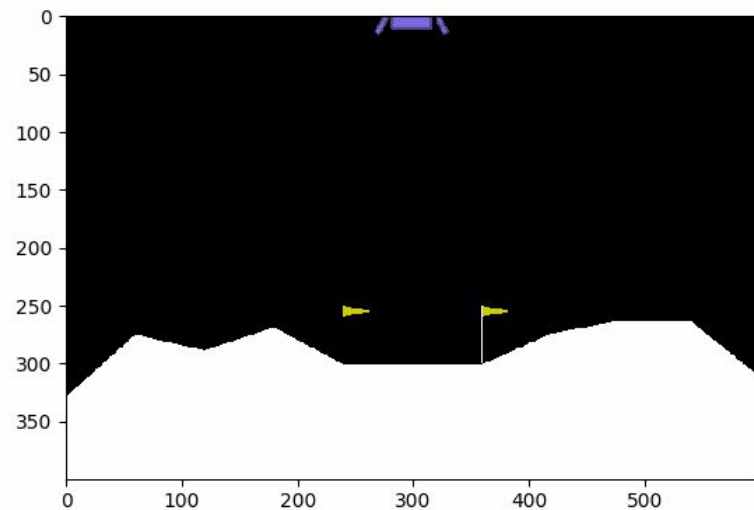
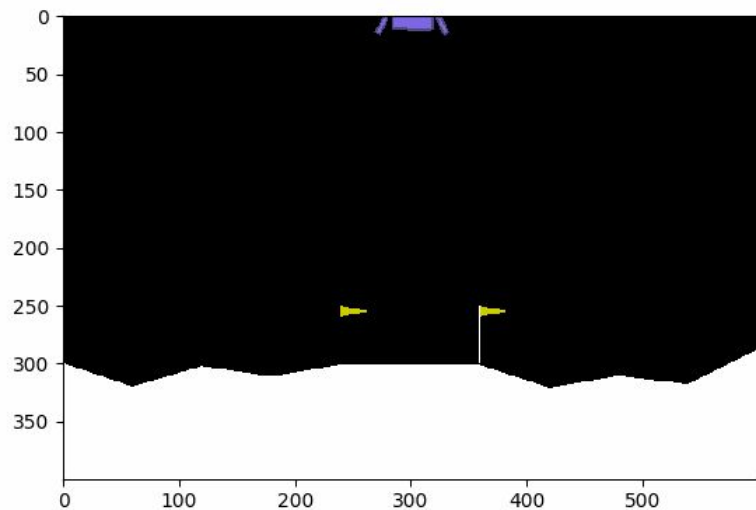
In this HomeWork, you can implement some Deep Reinforcement Learning methods by yourself:

- Policy Gradient
- Actor-Critic ( Implement by yourself to get high score !)

The environment of this HW is [Lunar Lander](#) in gym of OpenAI.

Other details can be found in the sample code.

# Illustraion



# Policy Gradient(to get 3 points)

---

## Algorithm 1 Policy Gradient

---

**function** REINFORCE

Initialize policy parameters  $\theta$

**for** each episode  $\{s_1, a_1, r_1, \dots, s_T, a_T, r_T\} \sim \pi_\theta$  **do**

**for**  $t = 1$  to  $T$  **do**

    Calculate discounted reward  $R_t = \sum_{i=t}^T \gamma^{i-t} r_i$

$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t | s_t) R_t$

**end for**

**end for**

**return**  $\theta$

**end function**

---

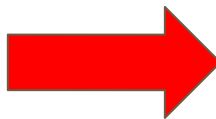
Agent



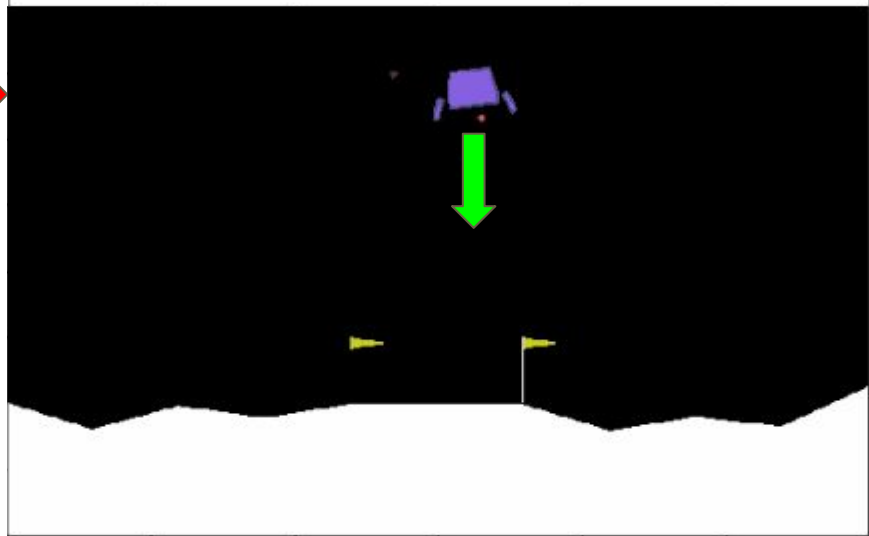
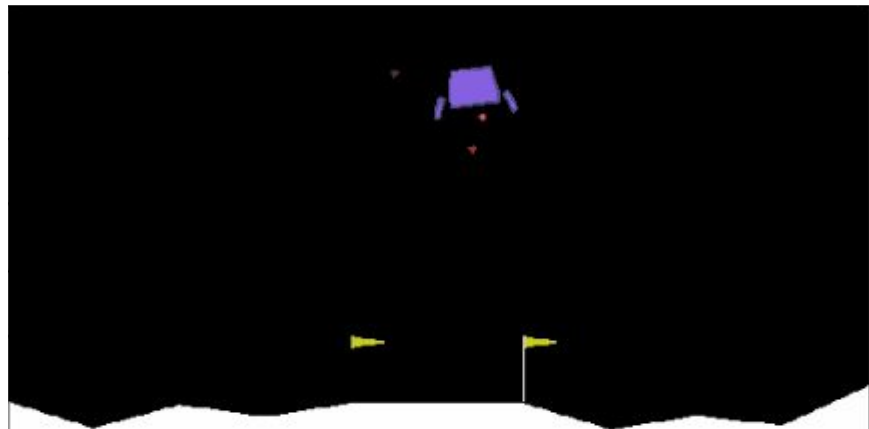
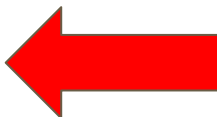
state



action



reward



# Policy Gradient(to get 3 points)

---

## Algorithm 1 Policy Gradient

---

**function** REINFORCE

Initialize policy parameters  $\theta$

**for** each episode  $\{s_1, a_1, r_1, \dots, s_T, a_T, r_T\} \sim \pi_\theta$  **do**

**for**  $t = 1$  to  $T$  **do**

    Calculate discounted reward  $R_t = \sum_{i=t}^T \gamma^{i-t} r_i$

$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) R_t$

**end for**

**end for**

**return**  $\theta$

**end function**

---

$$\gamma = 0.99, t=1, T = 3$$

$$R_1 = r_1 + 0.99 * r_2 + 0.99^2 * r_3$$

$$R_2 = r_2 + 0.99 * r_3$$

$$R_3 = r_3$$

# Actor-Critic(to get 4 points)

---

**Algorithm 2** Actor-Critic

---

**function** REINFORCE WITH BASELINE

Initialize policy parameters  $\theta$

Initialize baseline function parameters  $\phi$

**for** each episode  $\{s_1, a_1, r_1, \dots, s_T, a_T, r_T\} \sim \pi_\theta$  **do**

**for**  $t = 1$  to  $T$  **do**

    Calculate discounted reward  $R_t = \sum_{i=t}^T \gamma^{i-t} r_i$

    Estimate advantage  $A_t = R_t - b_\phi(s_t)$

    Re-fit the baseline by minimizing  $\|b_\phi(s_t) - R_t\|^2$

$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t | s_t) A_t$

**end for**

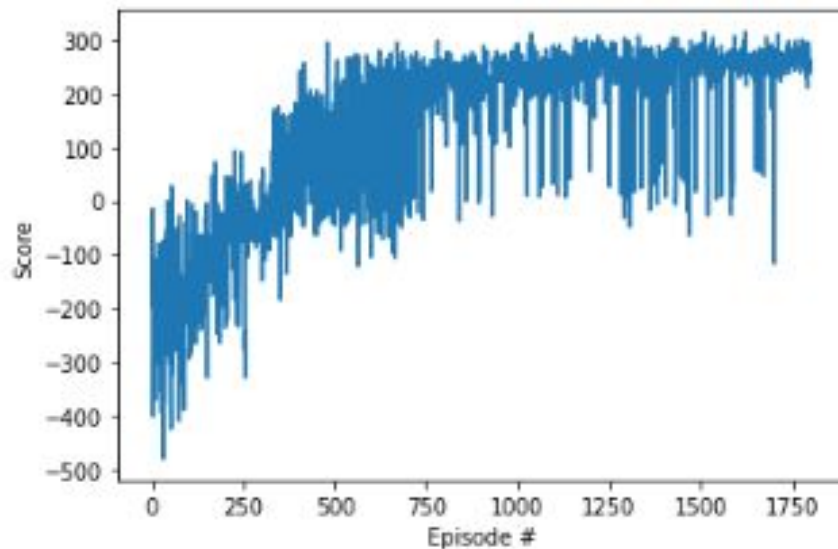
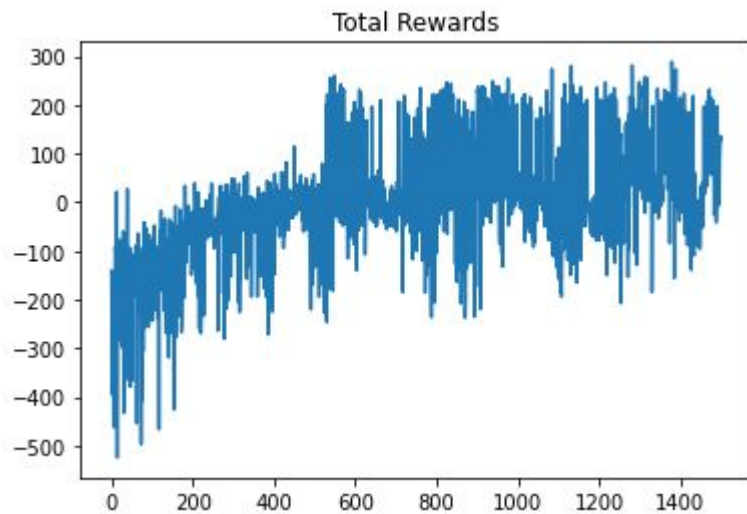
**end for**

**return**  $\theta$

**end function**

---

# Sample Result





# What you need to submit & Grading

1. Python file ( **2 points**) ( Submit on NTU COOL)
2. Action List ( **4 points**) (On JudgeBoi, no private set, **the highest one is automatically selected**)
3. Report ( **4 points**) (The questions are on **gradescope**)

Points	Intervals	
0	No valid Submission or $< 0$	
1	0-110	
2	110-180	
3	180-275	
4	$> 275$	



# What you need to submit & Grading

## More on a "valid submission ":

Your agent should output done after the last input of your action list, action list with mismatched length will be rejected。

Action list 的長相

```
1 print("Action list looks like ", action_list)
2 print("Action list's shape looks like ", np.shape(action_list))
```

```
Action list looks like [[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3
Action list's shape looks like (5,)
```

# Submission & Grading - JudgeBoi Rules (1/2)

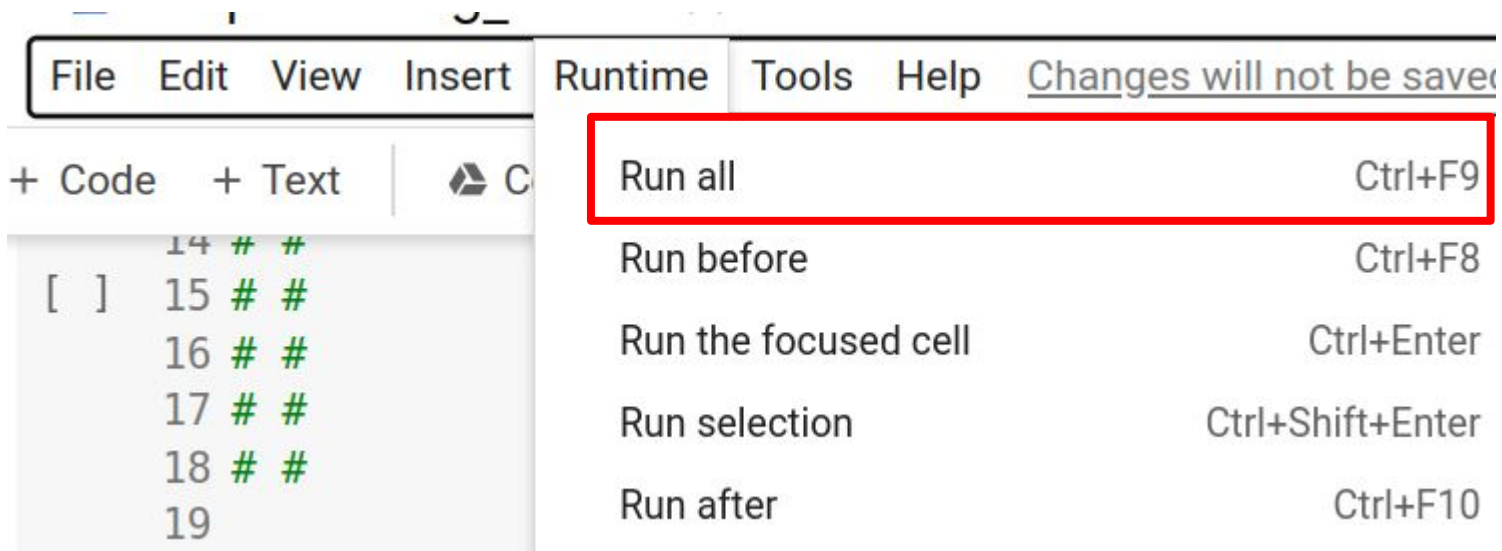
- We do limit the number of connections and request rate for each IP.
  - If you cannot access the website temporarily, please wait a moment.
- The system can be very busy as the deadline approaches.
  - If this prevents uploads, we do not offer additional submission opportunities.
- Please do not attempt to attack JudgeBoi.
- Every **Saturday** from **6:00 to 9:00** is our system maintenance time.
- For any JudgeBoi issues, please post on NTUCOOL discussion.
  - Discussion Link: [https://cool.ntu.edu.tw/courses/24108/discussion\\_topics/182915](https://cool.ntu.edu.tw/courses/24108/discussion_topics/182915)

# Submission & Grading - JudgeBoi Rules (2/2)

- 5 submission quota per day, reset at **midnight**.
  - Guest users have no quota.
- Only \*.npy file is allowed, file size should be smaller than **2MB**.
- You do not have to select submission since there is no private score
- JudgeBoi should complete the evaluation within one minute.
  - You do not need to wait for the progress bar to finish

# If you can't reproduce your result on JudgeBoi

- Please use "Run all" in colab to avoid reproducibility issues



The image shows a screenshot of the Google Colab interface. The top menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A status bar at the top right indicates 'Changes will not be saved'. Below the menu bar, there are tabs for '+ Code' and '+ Text'. The main area shows a code editor with lines 14 through 19, each containing '# #' in green. The 'Runtime' menu is open, and the 'Run all' option is highlighted with a red box. The 'Run all' option is accompanied by the keyboard shortcut 'Ctrl+F9'. Other options in the menu include 'Run before' (Ctrl+F8), 'Run the focused cell' (Ctrl+Enter), 'Run selection' (Ctrl+Shift+Enter), and 'Run after' (Ctrl+F10).

Option	Keyboard Shortcut
Run all	Ctrl+F9
Run before	Ctrl+F8
Run the focused cell	Ctrl+Enter
Run selection	Ctrl+Shift+Enter
Run after	Ctrl+F10

# Note

- HW12 won't use GPU by default.
- We recommend to use Colab in HW12.
- If anyone intend to use environments other than Colab, please fix reproducibility issues by yourself. TA won't help you to fix any environment issue.
- The training of HW12 should be able to finish within 30 min.

# Submission & Grading - Report

1. (2分) Implement Advanced RL algorithm
  - a. Choose one algorithm from Actor-Critic、REINFORCE with baseline、Q Actor-Critic、A2C, A3C or other advance RL algorithms and implement it.
  - b. Please explain the difference between your implementation and Policy Gradient
  - c. Please describe your implementation explicitly (If TAs can't understand your description, we will check your code directly.

# Submission & Grading - Report

2. (2分) How does the objective function of "PPO-ptx" differ from the "PPO" during RL training as used in the [InstructGPT paper](#)? (1 point) Also, what is the potential advantage of using "PPO-ptx" over "PPO" in the [InstructGPT paper](#)? Please provide a detailed analysis from their respective objective functions. (1 point)

Note. You should answer based on [InstructGPT paper](#)

$$\text{objective}(\phi) = E_{(x,y) \sim D_{\pi_{\phi}^{\text{RL}}}} \left[ r_{\theta}(x,y) - \beta \log \left( \pi_{\phi}^{\text{RL}}(y | x) / \pi^{\text{SFT}}(y | x) \right) \right] + \gamma E_{x \sim D_{\text{pretrain}}} \left[ \log(\pi_{\phi}^{\text{RL}}(x)) \right]$$



# Submission & Grading - NTU COOL

1. Compress the code, and submit to NTU COOL, the format is show below

Ex: <student\_id>\_hw12.zip

2. Only submit the code you use, **do not submit other files (model ,data...)**
3. **Deadline: 2023/6/16 23:59**

# Regulations

- You should **NOT** plagiarize, if you use any other resource, you should cite it in the reference.(\*)
- You should **NOT** modify your prediction files manually.
- Do **NOT** share codes or prediction files with any living creatures.
- Do **NOT** use any approaches to submit your results more than 5 times a day. Do **NOT** use pre-trained models.
- Your assignment will **not be graded** and your **final grade x 0.9** if you violate any of the above rules.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

(\*) [Academic Ethics Guidelines for Researchers by the Ministry of Science and Technology \(MOST\)](#)

# Deadline

- Leaderboard: JudgeBoi
  - 2023/06/16 23:59 (UTC+8)
- Code submission: NTU COOL
  - 2023/06/16 23:59 (UTC+8)
- Report submission: Gradescope
  - 2023/06/16 23:59 (UTC+8)

# Contact us if you have problems...

- NTU COOL (Best way)
  - [link](#)
- Email
  - [mlta-2023-spring@googlegroups.com](mailto:mlta-2023-spring@googlegroups.com)
  - The title should begin with “[hw12]”
- **Submit Deadline: 2023/6/16 23:59**