

Презентація до курсової роботи

Тема: Розробка та впровадження веб-інтерфейсу для досліджень у сфері Reinforcement Learning з використанням Gymnasium

Виконав: Лошак Максим Іванович, ПЗ-3

Науковий Керівник: Медвідь Сергій Олександрович



Тема: Розробка та впровадження веб-інтерфейсу для досліджень у сфері Reinforcement Learning з використанням Gymnasium

Мета проєкту:

- Підвищити зручність і доступність використання RL-досліджень.
- Розробити механізм автоматичного перехоплення викликів Gymnasium API
- Створити інтерактивний веб-інтерфейс для керування експериментами та візуального аналізу

Відомі підходи для досягнення поставленої мети

- Стандартизовані середовища (Gymnasium тощо)
- Графічні й візуалізаційні інструменти
- Хмарні/розподілені платформи
- Low-code рішення для конфігурації експериментів

Визначені прогалини:

- Відсутність динамічного перехоплення викликів API без змін у кодї користувача
- Нестача інтегрованої платформи з єдиним бекендом і клієнтом для real-time візуалізації



Amazon SageMaker



kubernetes

Аспектне перехоплення викликів

- Динамічне перехоплення викликів
Gymnasium API

З використанням аспектного перехоплювача, завдяки обгортанню середовища, наданого Gymnasium в створений клас GymAspectInterceptor, було отримано можливість перехоплювати всі виклики до API, (рисунок 1.) зокрема виклики функцій step() та reset(), щоб отримати з них сирі дані про стан тренування агенту.

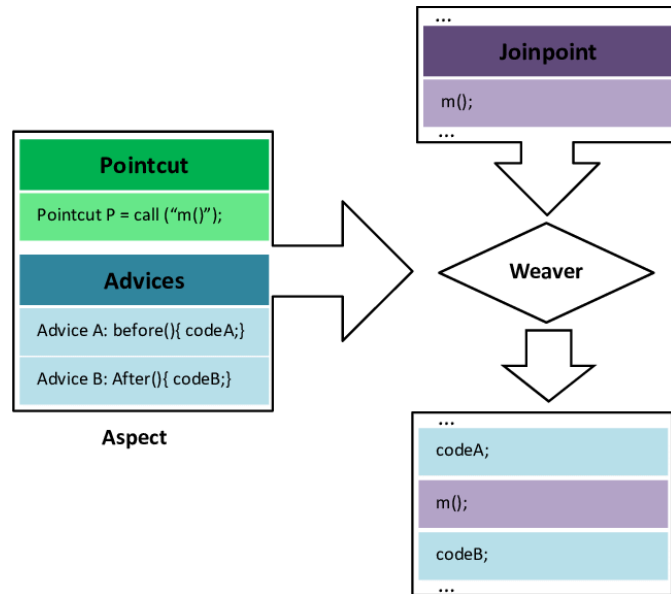


Рисунок 1.

Автор [Hanine Tout](#),
джерело: ResearchGate



Модульна серверна архітектура Django + REST / WebSocket

- REST на Django

Відповідає за розшифрування та структурування інформації, надісланої перехоплювачем.

Надає ендпоінти для надсилання, отримання та вивантаження статистики

- WebSocket на Django

Дозволяє налаштувати канал для оновлення даних про поточний стан агенту в реальному часі (рисунок 3.)

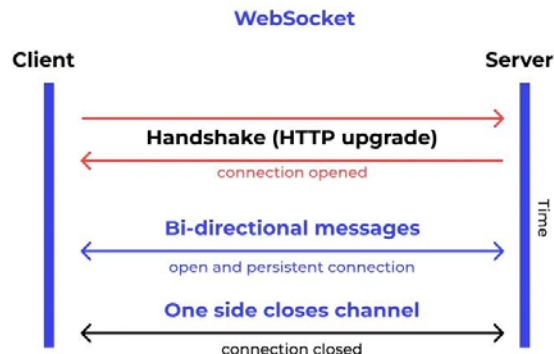
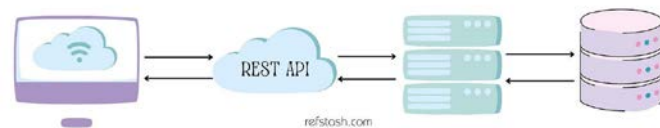


Рисунок 2, 3.
Автор [elijah samson](#),
джерело: Medium



Потокова передача даних через WebSocket

Щойно агент завершує епізод (або оновлює проміжні метрики), сервер надсилає повідомлення клієнту по WebSocket, що дозволяє будувати графіки “на льоту” без перезавантажень.

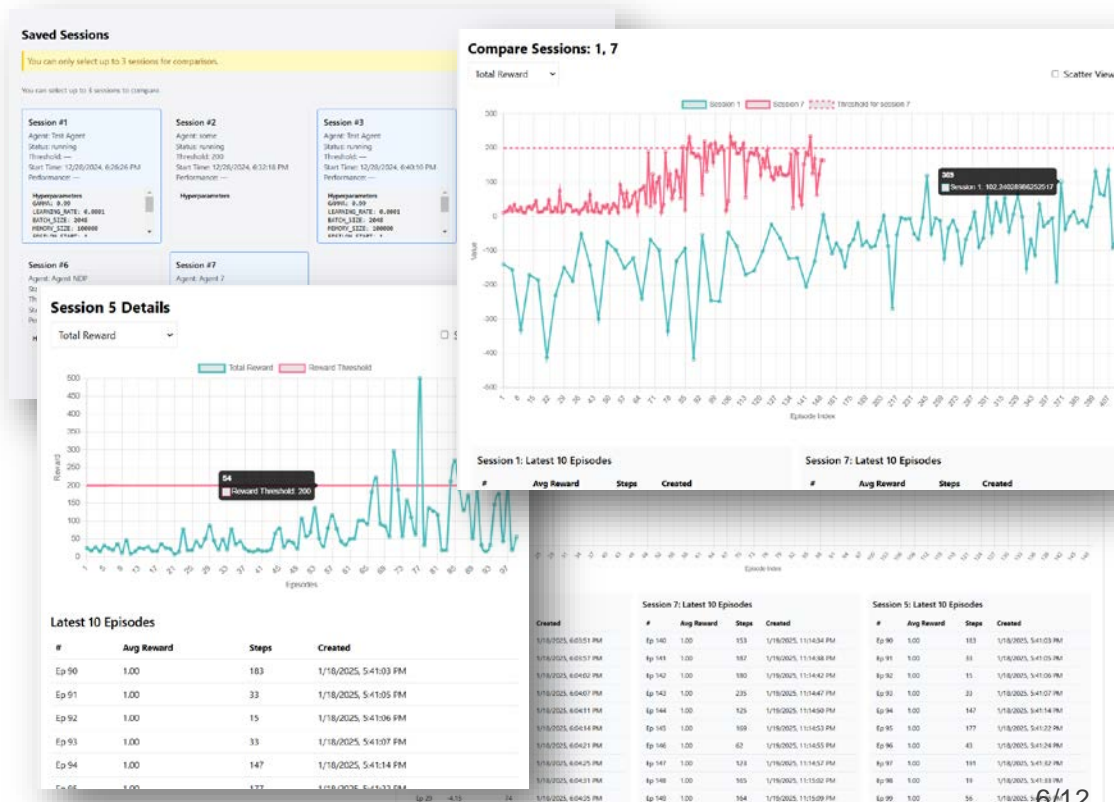
Для цього використовується ASGI інтерфейс, що й надає асинхронні можливості Django серверу.

```
1 class StatsConsumer(AsyncWebSocketConsumer):
2     def __init__(self, *args, **kwargs):
3         super().__init__(*args, **kwargs)
4         self.session_id = None
5         self.group_name = None
6
7     async def connect(self):
8         self.session_id = self.scope['url_route']['kwargs']['session_id']
9         self.group_name = f"session_{self.session_id}"
10
11         await self.channel_layer.group_add(
12             self.group_name,
13             self.channel_name
14         )
15         await self.accept()
16
17     async def disconnect(self, close_code):
18         await self.channel_layer.group_discard(
19             self.group_name,
20             self.channel_name
21         )
22
23     async def send_stats_update(self, event):
24         data = event['data']
25
26         await self.send(text_data=json.dumps({
27             'data': data
28         })))
```

Інтерактивна візуалізація з Chart.js

Клієнтська частина на React використовує бібліотеку Chart.js для побудови лінійних та точкових графіків.

Графіки оновлюються при надходженні нової епізодичної статистики через WebSocket, а користувач може фільтрувати, масштабувати та порівнювати декілька сесій, як за графіками, так і за таблицями з даними.

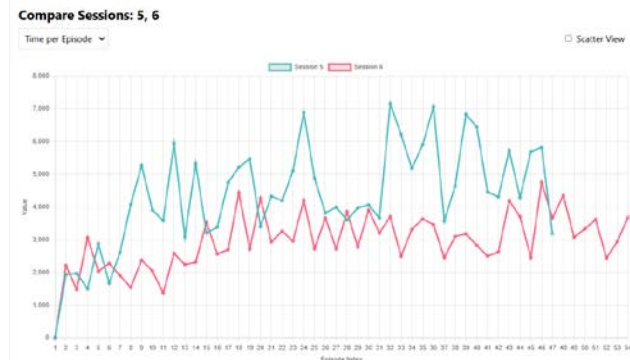
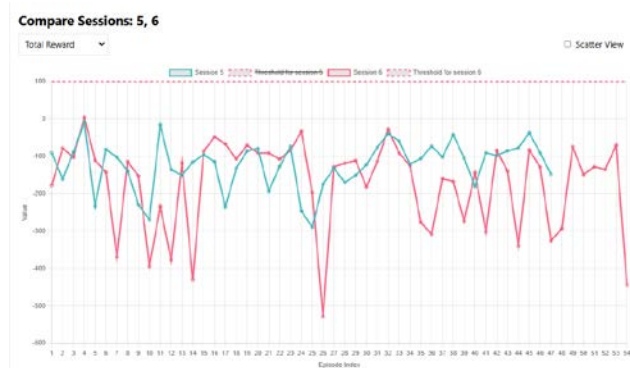


Приклад застосування методу до розв'язку модельної чи реальної задачі

Для демонстраційних цілей було взято код з репозиторію NDP (власник: [@enajx](#)) для тренування агентів в середовищах з дискретним простором дій. Мною було проведено тренування агенту на ≈ 50 епізодах в середовищі LunarLander-v2.

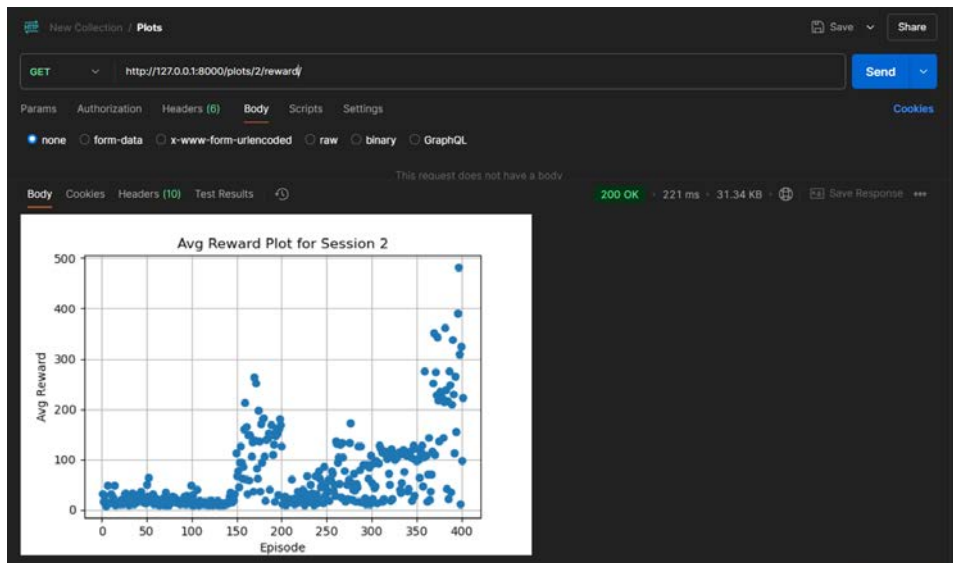
На наведених рисунках можна побачити порівняння результатів тренування

[@enajx](#) – Session 6 (червоний графік)
Мій код – Session 5 (зелений графік)



Інші приклади застосування

- Серверна частина також надає ендпоінти для отримання згенерованих графіків в форматі .jpeg
- Є можливості завантаження/вивантаження статистики в форматі .json та скрипти для роботи з консоллю, що спрощує обмін результатами досліджень.



```
-m pip install --upgrade pip
PS C:\Users\maxlo\workspace\Git\coursework\code\gym_rl_manager> python cli_upload.py my_stats.json --session-id 2
-m pip install --upgrade pip
PS C:\Users\maxlo\workspace\Git\coursework\code\gym_rl_manager> python cli_generate.py --session-id 2
```



Наукова новизна

Синтез методологічних рішень:

Поєднання динамічного перехоплення викликів Gymnasium API та автоматичного збору статистичних даних із застосуванням сучасних веб-технологій для відображення результатів у режимі реального часу.



Практична значимість

Спрощення інфраструктури проведення експериментів у сфері Reinforcement Learning

Досліднику більше не потрібно вручну інтегрувати засоби моніторингу й оцінки роботи агента: достатньо лише скористатися модифікованим середовищем, яке логує всі необхідні метрики та передає їх на сервер.

Це в свою чергу знижує ризик появи помилок, які часто виникають при реалізації збору даних дослідником, а також заощаджує час підготовки експерименту



Перспективи подальших досліджень

- Підтримка ширшого спектра сценаріїв і метрик.
- Удосконалення перехоплювача для роботи з моделями, що тренуються в кількох потоках на багатьох середовищах.
- Додання аутентифікації, розмежування доступу та засобів спільної роботи.
- Інтеграція з хмарними сервісами та платформи Container-orchestration (Docker, Kubernetes).



Висновки

- Розроблено комплексну систему для підтримки досліджень у сфері RL на базі Gymnasium
- Реалізовано аспектний перехоплювач, що забезпечує автоматичне надсилання статистичних даних
- Імплементовано серверну та клієнтську частину:
 - *Серверна частина:* багаторівнева взаємодія – через REST API, WebSocket та консоль. Надає вбудовані засоби аналізу.
 - *Клієнтська частина:* забезпечує інтуїтивно зрозуміле представлення результатів досліджень та інтерфейс для доступу до всього функціоналу застосунку.

Дякую за увагу

