НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Навчання з підкріпленням

Практична робота # 1

Перевернутий маятник (Cartpole), понг (Pong) та MountainCar

16 вересня 2021 р.

1 Вступ

1.1 Навчання з підкріпленням

За останні 5 років глибинне навчання з підкріпленням (deep RL) стало одним з найінтенсивніших напрямків досліджень у сфері штучного інтелекту. Глибинне навчання з підкріпленням поєднує нейронні мережі та навчання з підкріпленням – набір методів навчання, які вивчають оптимальну стратегію, метою якої є максимізація загальної винагороди, отриманої внаслідок взаємодії агента з навколишнім середовищем [1]. Сьогодні глибинне навчання з підкріпленням дозволяє досягати надлюдської продуктивності в ряді завдань: відео ігри [2], покер [3], а також в настільних іграх, включаючи ґо (go) та шахи (chess) [4, 5, 6, 7].

2 Завдання

Мета цієї роботи – познайомитися на практиці як можна створити середовище та навчити агента дотримуватись певної стратегії в цьому середовищі. Вам пропонується файл-розв'язок до лабораторної роботи #3, яка була запропонована для слухачів курсу `MIT 6.S191 Introduction to Deep Learning`. У цьому файлі-розв'язку Ви познайомитесь з прикладами моделювання двох середовищ різної складності на основі інструментарію OpenAI Gym та навчите агента приймати правильні рішення на основі взаємодій з середовищем.

- Файл-розв'язок (англійською): ФАЙЛ 1.
- Файл-розв'язок (українською): ФАЙЛ 2.

Основне Ваше завдання – розібратися з принципом побудови середовищ на основі інструментарію OpenAI Gym та процесом навчання агента у цих середовищах.

Додаткові завдання:

1. Дати коротке власне обґрунтування на питання, які пропонуються в останньому розділі `файла-розв'язку` – **3.10 Conclusion**.
2. Створити за аналогією попередніх прикладів середовище `MountainCar-v0` та навчити агента досягати поставленої мети у цьому середовищі.

Агент (автомобіль) у середовищі ***`MountainCar-v0`*** *знаходиться на одновимірній трасі між двома "горами". Мета агента – заїхати на гору праворуч до прапорця; проте двигун агента недостатньо потужний, щоб піднятися на гору просто рухаючись вперед. Тому єдиний спосіб досягти успіху – це рухатися вперед-назад, щоб набрати обертів.*

2.1 Оцінювання

Ваша оцінка за виконання завдання буде залежати від:

- 30% – подано у кінці `файла-розв'язку` коротке власне обґрунтування (відповіді) на питання
- 50% – створено середовище `MountainCar-v0` та навчено агента досягати поставлену мету у цьому середовищі
- 20% – звіт: описано процес створення середовища `MountainCar-v0` та процес навчання агента у цьому середовищі

Шаблон LaTeX

Шаблон за яким потрібно підготувати звіт можна звантажити ТУТ. Якщо не бажаєте установлювати додаткове програмне забезпечення – можете скористатися для підготовки звіту цим онлайн-ресурсом: www.overleaf.com.

Структура цього шаблону:

```
...
├── code     # Директорія для файлів з кодом (лістинг коду у звіті)
│   ├── test.py
│   └── hello.c
├── images    # Основна директорія для зображень (.pdf, .jpg, .png)
│   └── NN_.png
├── logo      # Директорія для логотипу
│   └── LogoKPI.jpg
├── main.pdf       # Файл отриманий внаслідок компіляції main.tex
├── main.tex       # TeX  файл звіту
├── ref.bib        # Бібліографія (BibTeX)
└── report.cls    # Файл налаштування стилів звіту
```

Відправлення роботи на розгляд

Дедлайн: Четвер, 30 вересня 2021 року о 23:59

- **Звіт проєкту**. Створіть архів `Прізвище Ім'я_група.zip` у який повинні бути включені:
  - Ваш звіт (.pdf файл) та решта файлів LaTeX
  - `Файл-розв'язок` з відповідями на питання, реалізованим та навченим агентом у середовищі `MountainCar`: `RL_Solution_Прізвище Ім'я_група.ipynb`

Файл звіту потрібно відправити на перевірку сюди: https://cloud.comsys.kpi.ua/s/LiMQjdBERoNx9GK

Література

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018. [Online]. Available: http://incompleteideas.net/book/RLbook2020.pdf

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] M. Moravčík, M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling, "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, 2017.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[5] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[6] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[7] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.