

WSI - ćwiczenie 6: Uczenie się ze wzmocnieniem

22 grudnia 2025

1 Sprawy organizacyjne

1. Ćwiczenie realizowane jest samodzielnie.
2. Ćwiczenie wykonywane jest w języku Python.
3. Ćwiczenie powinno zostać wysłane do prowadzącego najpóźniej w terminie 12.01.2026r (zademonstrowane i omówione może być na późniejszych zajęciach). W ramach oddawania ćwiczenia należy zademonstrować prowadzącemu działanie kodu oraz utworzyć pull request (z kodem oraz raportem), który prowadzący będzie mógł komentować.
4. Kod oraz raport powinien być umieszczony na repozytorium gitlab-stud.elka.pw.edu.pl, a pull request utworzony do konta prowadzącego: @djagodzi (dla nazwisk A-N) i @pzawist2 (dla nazwisk O-Z)
5. Dokumentacja powinna być w postaci pliku .pdf albo być częścią notebooka jupyterowego. Powinna zawierać opis eksperymentów, uzyskane wyniki wraz z komentarzem oraz wnioski.
6. Na ocenę wpływa poprawność oraz jakość kodu i dokumentacja.
7. Można korzystać z pakietów do obliczeń numerycznych, takich jak *numpy*
8. Można skorzystać ze środowiska z pakietu *minigrid*
9. Implementacja powinna być ogólna, w szczególności algorytm powinien nadawać się do adaptacji do rozwiązywania innych problemów uczenia ze wzmocnieniem.

2 Ćwiczenie

Celem ćwiczenia jest implementacja algorytmu **Q-learning** z epizodami oraz ϵ -zachłanną strategią wyboru akcji. Algorytm należy zastosować do wytrenowania agenta rozwiązującego problem Four Rooms (<https://minigrid.farama.org/environments/minigrid/FourRoomsEnv/>), który dostępny jest w pakiecie *minigrid*. Wykonaj przy tym eksperymenty dla różnych zestawów parametrów, uwzględniając: współczynnik dyskontowania γ , szybkość uczenia β , parametr eksploracji ϵ .

2.1 Zadania dla chętnych

- Dostosuj rozwiązanie, aby działało również dla losowych pozycji agenta i celu.
- Zaimplementuj algorytm **SARSA** dla tego samego środowiska.

2.2 Wskazówki

- Warto pomyśleć nad odpowiednią reprezentacją stanu przekazywanego przez środowisko, aby nie było problemu z wielkością tabeli Q.
- Można przyjąć założenie, że trenujemy agenta w środowisku, w którym jego pozycja i pozycja celu są stałe:

```
env = gym.make("MiniGrid-FourRooms-v0", max_steps=max_steps,
               goal_pos=(2, 2), agent_pos=(7,7))
```