WSI - ćwiczenie 6.
Reinforcement learning

26 maja 2026

1 Sprawy organizacyjne

1. Ćwiczenie realizowane jest samodzielnie, w języku Python.
2. Ćwiczenie powinno zostać wysłane do prowadzącego najpóźniej w dniu 13-tych zajęć. W ramach oddawania ćwiczenia należy zademonstrować prowadzącemu działanie kodu oraz wysłać na maila kod oraz dokumentację.
3. Dokumentacja powinna być w postaci pliku .pdf albo być częścią notebooka Jupyter. Powinna zawierać opis eksperymentów, uzyskane wyniki wraz z komentarzem oraz wnioski.
4. Na ocenę wpływają poprawność oraz jakość kodu i dokumentacji.
5. W rozwiązaniu należy wykorzystywać pakiet `Gymnasium`.
6. Implementacja algorytmów powinna być ogólna.

2 Ćwiczenie

FrozenLake to klasyczne środowisko testowe w uczeniu ze wzmocnieniem. Agent porusza się po zamarzniętym jeziorze, próbując dotrzeć z punktu startowego (S) do prezentu (G), unikając dziur w lodzie (H). Lód jest śliski - agent zamierza pójść w jedną stronę, ale z pewnym prawdopodobieństwem ślizga się w bok. Dla środowiska `FrozenLake-v1` proszę zaimplementować i porównać dwa algorytmy: Q-learning oraz SARSA, dla planszy o wymiarach 8x8 z włączonym ślizganiem.

W ramach ćwiczenia należy:

1. Zaimplementować oba algorytmy w wersji tabularnej, z eksploracją ϵ-zachłanną z malejącym ϵ. Sugerowane hiperparametry początkowe: $\alpha = 0.1$, $\gamma = 0.99$, ϵ malejące liniowo od 1.0 do 0.05, liczba epizodów treningu $\geqslant 20,000$.
2. Przeprowadzić testy dla klasycznej funkcji nagrody (+1 za dotarcie do mety, 0 w pozostałych przypadkach). Każdy eksperyment powtórzyć co najmniej 5 razy z różnymi ziarnami losowymi i raportować średnią i odchylenie. Po treningu wykonać ewaluację z wyłączoną eksploracją ($\epsilon = 0$) na 1000 epizodów.
3. Zaproponować dwie autorskie funkcje nagrody (np. kara za każdy krok, kara za wpadnięcie w dziurę, nagroda zależna od odległości do celu) i powtórzyć eksperymenty dla obu algorytmów. Uwaga: metryka sukcesu (odsetek epizodów zakończonych w mecie) musi być liczona na podstawie oryginalnej, niezmodyfikowanej nagrody środowiska.
4. Przygotować raport zawierający: opis implementacji, opis przeprowadzonych eksperymentów, uzyskane wyniki w formie tabelarycznej i wykresów (krzywe uczenia z pasmami odchylenia, wizualizacja nauczonej polityki, porównanie końcowych success rate) oraz wnioski. W raporcie należy w szczególności odpowiedzieć na pytania:
   - Który algorytm uczy się szybciej, a który osiąga wyższy końcowy success rate?
   - Który algorytm ma wyższy odsetek wpadnięć w dziury podczas treningu? Jak można wyjaśnić uzyskane wyniki?
   - Czy zaproponowane modyfikacje funkcji nagrody przyspieszają uczenie? Czy zmieniają jakość końcowej polityki? Czy zaobserwowano zjawisko reward hackingu?