Uczenie ze wzmocnieniem

Opis i implementacja algorytmu Q-Learning

Wstęp

Uczenie ze wzmocnieniem (*ang. Reinforcement Learning*) jest dziedziną uczenia maszynowego zajmującą się problemami podejmowania decyzji przez agenta działającego w środowisku. W przeciwieństwie do uczenia nadzorowanego, agent nie otrzymuje poprawnych odpowiedzi dla kolejnych sytuacji, lecz uczy się na podstawie nagród oraz kar uzyskiwanych podczas interakcji ze środowiskiem. Jednym z najpopularniejszych algorytmów uczenia ze wzmocnieniem jest właśnie Q-Learning, który pozwala agentowi na naukę optymalnej strategii działania bez konieczności posiadania modelu środowiska.

1. Opis algorytmu

1.1 Model środowiska

W uczeniu ze wzmocnieniem problem decyzyjny najczęściej opisuje się za pomocą procesu decyzyjnego Markowa (*ang. Markov Decision Process, MDP*). W modelu tym agent w każdej chwili znajduje się w pewnym stanie $s \in S$, wykonuje akcję $a \in A$, po czym przechodzi do nowego stanu s' i otrzymuje nagrodę r.

Celem agenta jest maksymalizacja zdyskontowanej sumy przyszłych nagród:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

gdzie $\gamma \in [0, 1]$ jest współczynnikiem dyskontowania określającym znaczenie przyszłych nagród. Im większa wartość γ, tym bardziej agent uwzględnia długoterminowe konsekwencje swoich decyzji.

1.2 Funkcja wartości akcji

Podstawą działania algorytmu Q-Learning jest funkcja wartości akcji, oznaczana jako $Q(s, a)$. Określa ona oczekiwaną sumę przyszłych nagród możliwą do uzyskania po wykonaniu akcji a w stanie s oraz dalszym postępowaniu zgodnie z optymalną strategią.

Dla środowisk o dyskretnej przestrzeni stanów i akcji funkcja ta przechowywana jest zwykle w postaci tablicy Q (*Q-table*), której elementy odpowiadają poszczególnym parom stan-akcja:

$$Q = \begin{bmatrix} Q(s_1, a_1) & Q(s_1, a_2) & \dots \\ Q(s_2, a_1) & Q(s_2, a_2) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

1.3 Reguła aktualizacji Q-Learning

Q-Learning wykorzystuje metodę iteracyjnej aktualizacji wartości funkcji Q. Po wykonaniu akcji agent obserwuje otrzymaną nagrodę oraz nowy stan środowiska, a następnie modyfikuje odpowiedni wpis w tablicy zgodnie z regułą aktualizacji opartą na równaniu Bellmana [1]:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

gdzie:

- α - współczynnik uczenia (*ang. learning rate*),
- γ - współczynnik dyskontowania,
- r - otrzymana nagroda,
- s' - stan osiągnięty po wykonaniu akcji,
- $\max_{a'} Q(s',a')$ - najlepsza przewidywana wartość możliwa do uzyskania w nowym stanie.

Wielkość znajdująca się w nawiasie kwadratowym nazywana jest błędem TD (*ang. Temporal Difference Error*) i określa różnicę pomiędzy aktualnym oszacowaniem wartości akcji a nową wiedzą uzyskaną po wykonaniu kroku w środowisku.

1.4 Strategia eksploracji

Aby agent miał szansę odnaleźć optymalną strategię działania, konieczne jest zachowanie równowagi pomiędzy wykorzystywaniem już zdobytej wiedzy (*eksploatacja*) a odkrywaniem nowych możliwości (*eksploracja*).

W tym celu można zastosować strategię ε-zachłanną. Z prawdopodobieństwem ε agent wybiera losową akcję, natomiast z prawdopodobieństwem $1 - \varepsilon$ wybierana jest akcja o największej wartości funkcji Q:

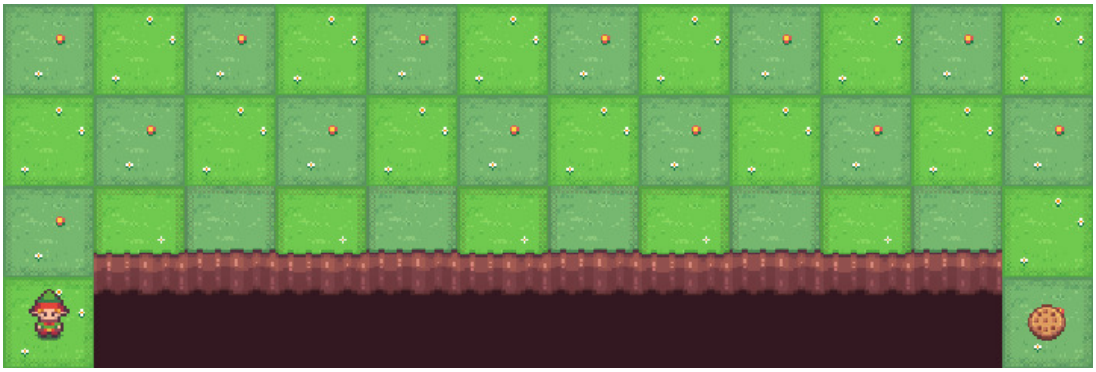
$$a = \begin{cases} \text{losowa akcja}, & \text{z prawdopodobieństwem } \varepsilon \\ \arg\max_a Q(s,a), & \text{z prawdopodobieństwem } 1-\varepsilon \end{cases}$$

Takie podejście pozwala agentowi stopniowo zdobywać informacje o środowisku, jednocześnie coraz częściej wykorzystując najbardziej opłacalne działania. Zastosowanie strategii ε-zachłannej pozwala połączyć proces eksploracji środowiska z wykorzystaniem już zdobytej wiedzy. Dzięki temu agent może odkrywać nowe ścieżki prowadzące do celu, jednocześnie stopniowo zwiększając prawdopodobieństwo wyboru akcji uznawanych za najbardziej korzystne.

[1] Wiki - Bellman Equation

2. Algorytm w działaniu

W celu oceny skuteczności zaimplementowanego algorytmu przeprowadzono serię eksperymentów w środowisku Cliff Walking [2]. Celem agenta było nauczenie się strategii prowadzącej od pola startowego do pola docelowego (ciastka) przy jednoczesnym unikaniu obszaru przepaści, którego odwiedzenie skutkuje wysoką karą.



Rysunek 1: Schemat środowiska Cliff Walking

2.1 Opis środowiska Cliff Walking

Środowisko Cliff Walking posiada dyskretną przestrzeń stanów oraz dyskretną przestrzeń akcji. Każde pole planszy odpowiada jednemu stanowi $s \in S$, natomiast zbiór dostępnych akcji definiowany jest jako:

$$A = \{\text{góra}, \text{dół}, \text{lewo}, \text{prawo}\}$$

Agent rozpoczyna każdy epizod w ustalonym stanie początkowym s_{start}, a jego celem jest osiągnięcie stanu końcowego s_{goal}. Wykonanie akcji powoduje przejście do nowego stanu zgodnie z dynamiką środowiska.

Funkcja nagrody zdefiniowana została w następujący sposób:

$$R(s, a, s') = \begin{cases} -100, & s' \in S_{cliff} \\ -1, & \text{w przeciwnym przypadku} \end{cases}$$

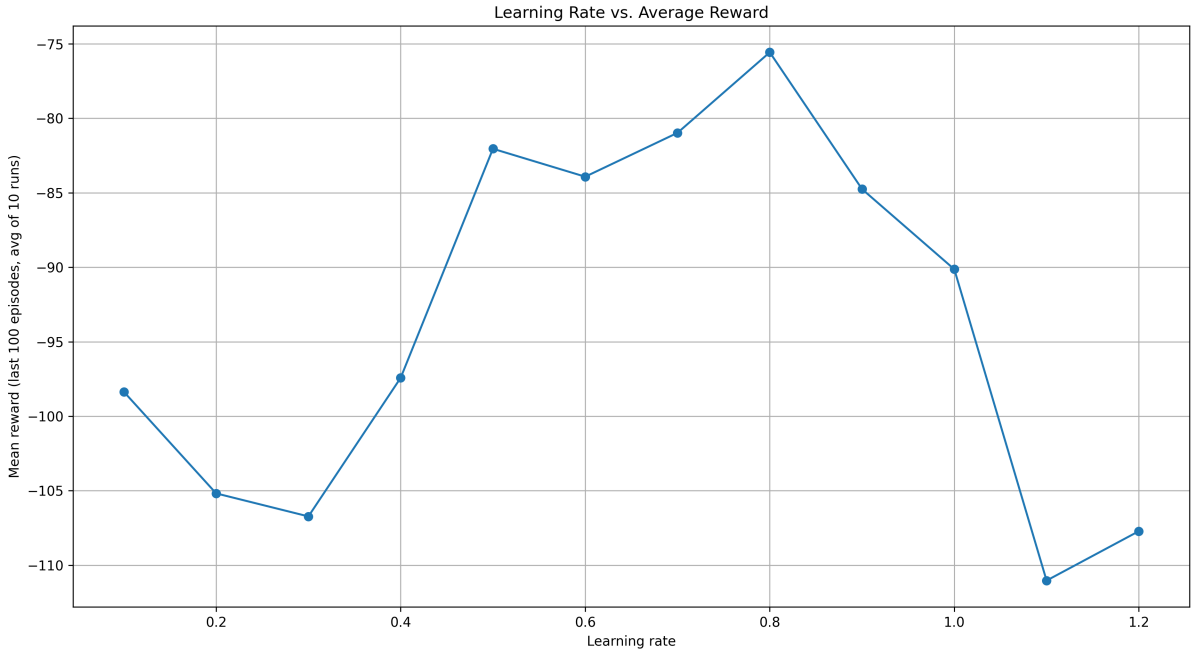
gdzie S_{cliff} oznacza zbiór stanów należących do obszaru przepaści.

Osiągnięcie stanu należącego do zbioru S_{cliff} powoduje przyznanie wysokiej kary oraz powrót agenta do stanu początkowego. Epizod kończy się po osiągnięciu stanu docelowego s_{goal}. Tak zdefiniowana funkcja nagrody zachęca agenta do odnalezienia ścieżki minimalizującej całkowity koszt przejścia pomiędzy stanem początkowym a końcowym. Jednocześnie wysoka kara za wejście na pola przepaści wymusza uwzględnienie ryzyka związanego z wyborem krótszych tras przebiegających w pobliżu obszaru karnego.

[2] Cliff Walking - Gymnasium

2.2 Dobór współczynnika uczenia

Współczynnik uczenia α w algorytmie Q-Learning określa stopień, w jakim nowe doświadczenia wpływają na aktualizację wartości funkcji $Q(s, a)$. Jako miarę jakości procesu uczenia zastosowano średnią nagrodę z ostatnich 100 epizodów treningowych, która stanowi przybliżenie stabilnego poziomu polityki po zakończeniu fazy eksploracji. Wykres przedstawiający zależność średniej nagrody od liczby epizodów dla różnych wartości współczynnika uczenia α został przedstawiony poniżej:



Na podstawie przeprowadzonych eksperymentów można sformułować następujące wnioski:

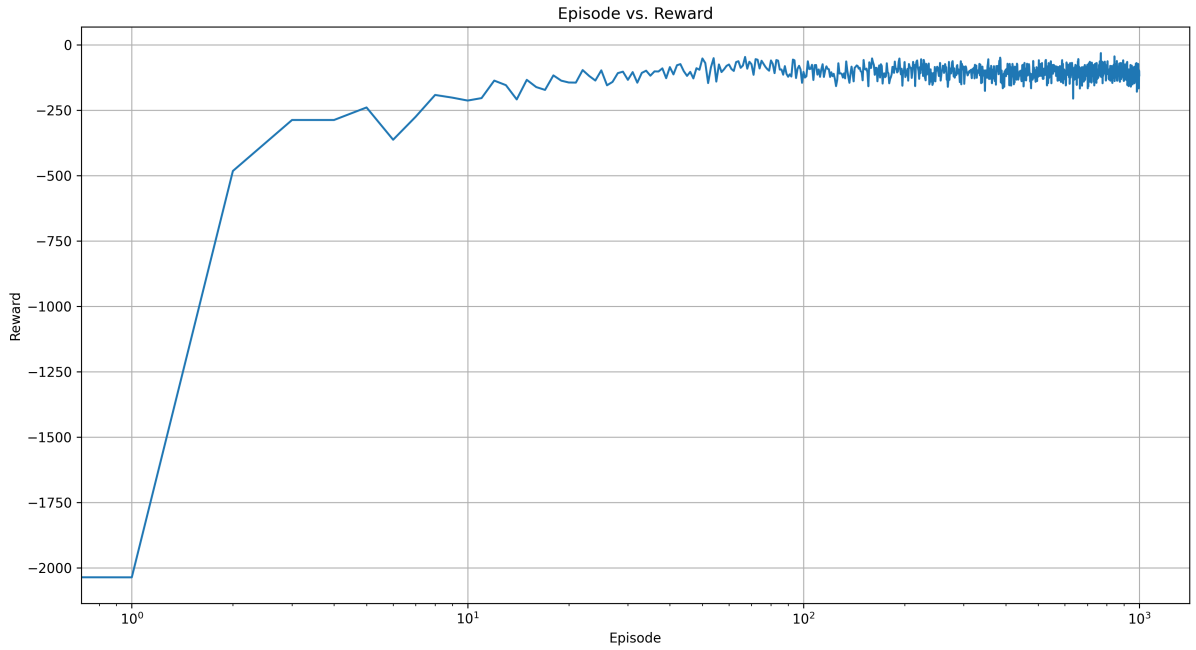
- **Niskie wartości współczynnika uczenia:** dla małych wartości α proces aktualizacji funkcji $Q(s, a)$ przebiegał zbyt wolno, co skutkowało niedostatecznym wykorzystaniem informacji zwrotnych ze środowiska oraz niższą jakością końcowej polityki.
- **Średnie wartości α:** w zakresie umiarkowanych wartości współczynnika uczenia obserwuje się najbardziej stabilny przebieg procesu uczenia, co prowadzi do szybszej konwergencji oraz lepszej jakości uzyskiwanych strategii działania.
- **Wysokie wartości współczynnika uczenia:** zbyt duże wartości α powodują nadmierną reaktywność aktualizacji Q, co może prowadzić do oscylacji wartości funkcji oraz pogorszenia stabilności procesu uczenia.
- **Dobór parametru:** eksperymentalnie wyznaczono, że wartość $\alpha = 0.8$ zapewnia najlepszy kompromis pomiędzy stabilnością a szybkością uczenia w analizowanym środowisku Cliff Walking.

Uzyskane wyniki potwierdzają, że odpowiedni dobór współczynnika uczenia ma kluczowe znaczenie dla stabilności oraz efektywności algorytmu Q-Learning w środowiskach o dyskretnej przestrzeni stanów.

2.3 Wpływ ilości epizodów treningowych

Liczba epizodów treningowych określa, jak długo agent może zdobywać doświadczenie oraz aktualizować wartości funkcji $Q(s, a)$. Parametr ten wpływa bezpośrednio na stopień poznania środowiska oraz jakość wyuczonej polityki.

W celu analizy wpływu liczby epizodów przeprowadzono serię eksperymentów dla różnych długości treningu. Dla każdej konfiguracji obliczono średnią nagrodę osiąganą podczas procesu uczenia, a następnie uśredniono wyniki z wielu niezależnych uruchomień algorytmu.



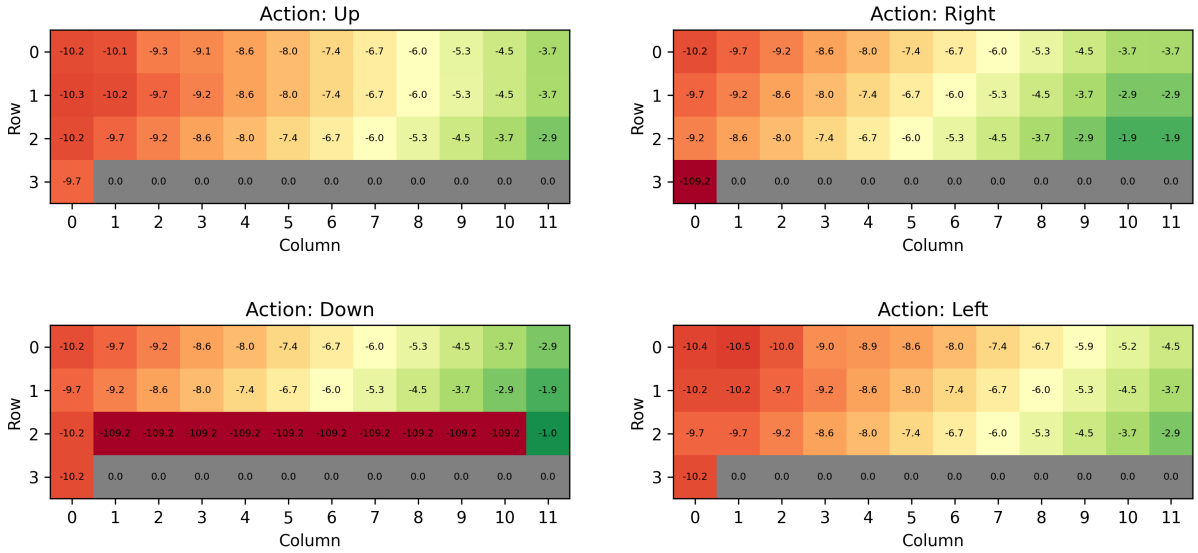
Na podstawie przeprowadzonych eksperymentów można sformułować następujące wnioski:

- **Znaczenie długości treningu:** zbyt mała liczba epizodów nie pozwala agentowi na zgromadzenie wystarczającej liczby doświadczeń potrzebnych do wyuczenia skutecznej polityki.
- **Poprawa jakości polityki:** wraz ze wzrostem liczby epizodów obserwowany jest wzrost średniej nagrody, świadczący o coraz skuteczniejszym działaniu agenta.
- **Malejące korzyści:** po przekroczeniu pewnej liczby epizodów dalsze wydłużanie treningu prowadzi do coraz mniejszych przyrostów jakości rozwiązania.
- **Stabilizacja procesu uczenia:** dla dużej liczby epizodów wartości średniej nagrody stabilizują się, co sugeruje zbieżność algorytmu do najlepszej polityki działania.

Na podstawie uzyskanych wyników można stwierdzić, że odpowiednio długa faza treningowa jest niezbędna do uzyskania skutecznej polityki działania. Jednocześnie po osiągnięciu stanu bliskiego konwergencji dalsze zwiększanie liczby epizodów przynosi jedynie ograniczoną poprawę jakości uzyskiwanego rozwiązania.

2.4 Trening i wyniki

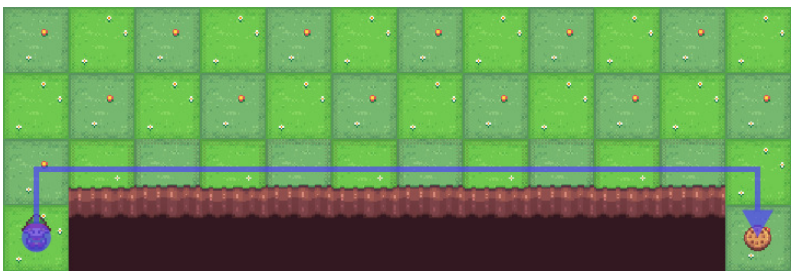
W wyniku procesu uczenia algorytmu Q-Learning uzyskano tablicę wartości akcji $Q(s, a)$ dla środowiska Cliff Walking, opisującą oczekiwaną sumę zdyskontowanych nagród dla każdej pary stan–akcja. Poniższa wizualizacja Q-table składa się z czterech map cieplnych odpowiadających możliwym akcjom: góra, prawo, dół oraz lewo. Każda z nich przedstawia wartości funkcji $Q(s, a)$ dla każdego stanu w środowisku.



Na podstawie wizualizacji można zauważyć, że w większości stanów jedna z akcji osiąga wyraźnie wyższą wartość $Q(s, a)$ niż pozostałe, co wskazuje na ukształtowanie stabilnej polityki działania. W początkowej części środowiska agent preferuje ruch w górę, a następnie w prawo, co odpowiada najkrótszej ścieżce prowadzącej do stanu docelowego. Struktura wartości Q potwierdza konsekwentne kierowanie się w stronę celu w kolejnych stanach.

Wyraźnie niższe wartości funkcji Q dla akcji prowadzącej w dół w obszarze przepaści wskazują, że agent skutecznie nauczył się unikać stanów kończących epizod wysoką karą. Całość tablicy Q pokazuje stopniowe „wznoszenie się" wartości w kierunku celu oraz wyraźne rozdzielenie obszaru bezpiecznych przejść i przepaści, co potwierdza poprawne wyuczenie polityki dla środowiska Cliff Walking.

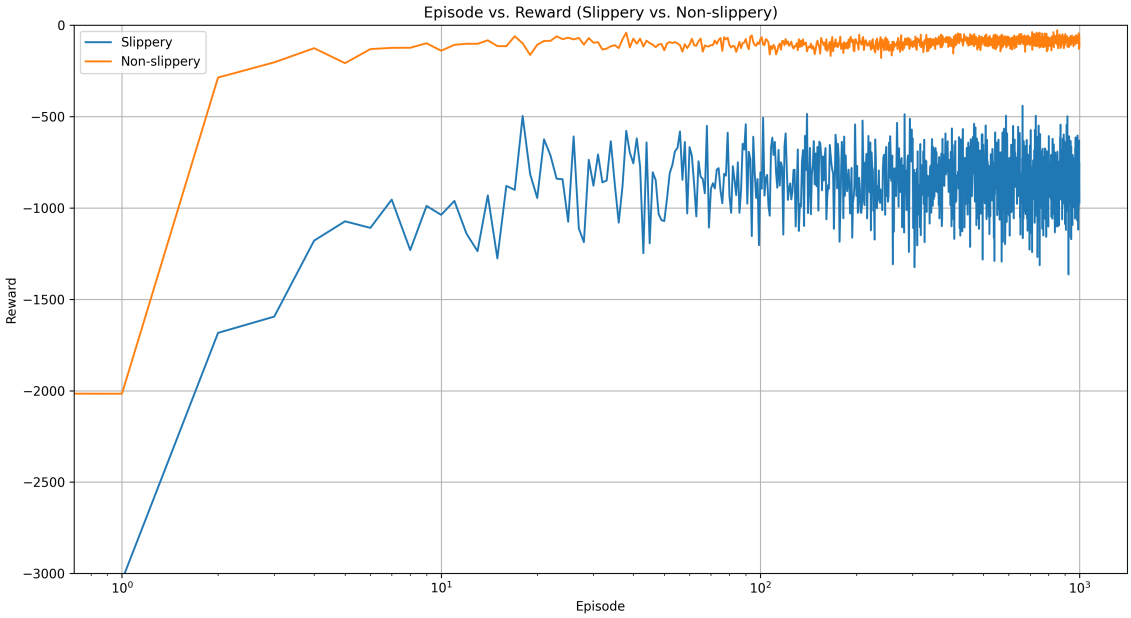
Ostateczna ścieżka wyznaczona przez politykę zachłanną znajduje się poniżej. Otrzymana trasa stanowi bezpośrednią interpretację tablicy Q i pokazuje finalną strategię agenta prowadzącą od stanu startowego do celu.



2.5 Stochastyczna modyfikacja środowiska (slippery)

Domyślnie środowisko Cliff Walking jest deterministyczne, co oznacza, że wykonanie danej akcji zawsze prowadzi do tego samego stanu. Włączenie parametru `is_slippery=True` wprowadza stochastyczność przejść pomiędzy stanami. Agent nie zawsze przemieszcza się dokładnie w wybranym kierunku, a wykonana akcja może skutkować ruchem do jednego z sąsiednich stanów.

Powoduje to wzrost niepewności środowiska oraz utrudnia proces uczenia, ponieważ identyczne decyzje mogą prowadzić do różnych rezultatów. W konsekwencji agent musi nauczyć się polityki odpornej na losowe zakłócenia zamiast wykorzystywać pojedynczą ścieżkę.



Na podstawie przeprowadzonych eksperymentów można sformułować następujące wnioski:

- **Wolniejsza zbieżność:** w środowisku stochastycznym agent potrzebuje większej liczby epizodów do wyuczenia skutecznej polityki działania.
- **Większa niestabilność uczenia:** przebieg procesu uczenia dla wariantu `slippery=True` charakteryzuje się większymi wahaniami osiąganych nagród pomiędzy kolejnymi epizodami.
- **Niższa jakość polityki:** nawet po długim treningu średnia nagroda pozostaje niższa niż w przypadku środowiska deterministycznego, co wynika z losowego charakteru przejść.
- **Większa trudność problemu:** agent nie może polegać wyłącznie na jednej znalezionej trajektorii, lecz musi uwzględniać możliwość przypadkowego opuszczenia zaplanowanej ścieżki.
- **Odporność algorytmu:** pomimo zwiększonej trudności środowiska algorytm Q-Learning nadal skutecznie poprawia swoją politykę wraz ze wzrostem liczby epizodów treningowych.

3. Podsumowanie

Celem zadania była implementacja algorytmu Q-Learning oraz analiza jego działania w środowisku Cliff Walking. Zaimplementowany agent uczył się wyznaczania polityki maksymalizującej sumę przyszłych nagród wyłącznie na podstawie interakcji ze środowiskiem, bez znajomości jego modelu.

Przeprowadzone eksperymenty pozwoliły ocenić wpływ najważniejszych parametrów algorytmu na jakość procesu uczenia. Uzyskane wyniki potwierdziły, że odpowiedni dobór hiperparametrów ma istotny wpływ zarówno na szybkość zbieżności, jak i końcową jakość wyuczonej polityki. Na podstawie przeprowadzonych eksperymentów można sformułować następujące wnioski:

- **Skuteczność algorytmu:** Q-Learning poprawnie odtworzył strukturę środowiska i wyznaczył politykę prowadzącą do osiągnięcia celu przy jednoczesnym unikaniu obszaru przepaści.
- **Wpływ współczynnika uczenia:** odpowiednio dobrana wartość parametru α pozwala znacząco przyspieszyć proces uczenia, natomiast zbyt małe wartości prowadzą do wolnej adaptacji agenta.
- **Znaczenie liczby epizodów:** wydłużenie treningu umożliwia dokładniejsze oszacowanie wartości funkcji $Q(s, a)$, jednak po osiągnięciu zbieżności dalsze zwiększanie liczby epizodów przynosi coraz mniejsze korzyści.
- **Wpływ losowości środowiska:** wprowadzenie mechanizmu `slippery` utrudnia proces uczenia, zwiększa wariancję uzyskiwanych nagród oraz obniża końcową jakość wyuczonej polityki.
- **Interpretowalność rozwiązania:** tablica Q umożliwia bezpośrednią analizę procesu podejmowania decyzji przez agenta, co stanowi jedną z głównych zalet klasycznych metod uczenia ze wzmocnieniem opartych na tablicach wartości.

Przeprowadzone eksperymenty potwierdzają, że Q-Learning jest skuteczną metodą rozwiązywania problemów sekwencyjnego podejmowania decyzji w środowiskach o dyskretnej przestrzeni stanów i akcji. Jednocześnie zadanie pokazuje ograniczenia podejścia tablicowego, którego zastosowanie staje się trudne dla środowisk o dużej liczbie stanów, gdzie konieczne jest wykorzystanie metod aproksymujących funkcję wartości, takich jak na przykład sieci neuronowe.