WSI zad5

Michał Mokrzycki 324874

December 2023

1 Zadanie

Celem zadania była implementacja algorytmu Q-learning z ϵ - zachłanną strategią losowania akcji. Algorytm został zastosowany do wytrenowania agenta rozwiązującego problem Cliff Walking , który dostępny jest w pakiecie gymnasium. Zostanie również zbadany współczynnik dyskontowania (λ) i szybkości uczenia (β) na algorytm.

2 Zasady gry

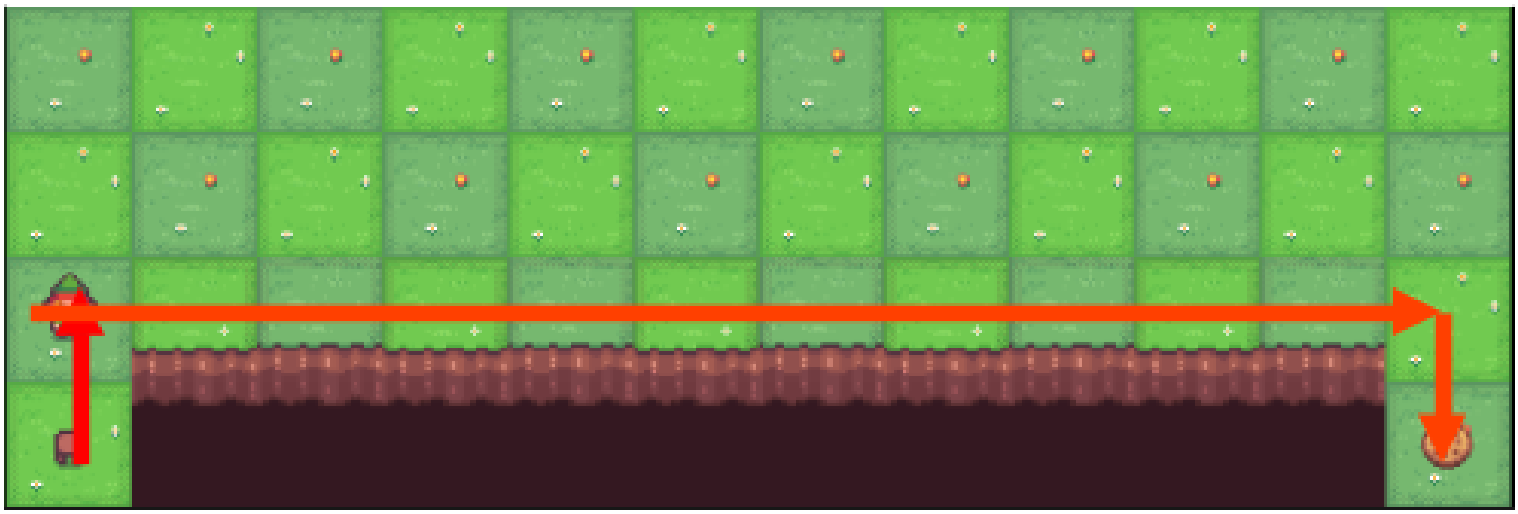
Zadanie polega na wytrenowaniu agenta który przejdzie z lewego dolnego rogu planszy do prawego dolnego. Gdy podczas ruchu natrafi na klif, powraca do miejsca docelowego. Możliwe są ruchy w 4 kierunki: w górę, w dół, w prawo oraz w lewo.

2.1 Funckja nagrody

W implementacji naszego algorytmu korzystamy z biblioteki gymnasium w której funkcja kary zaimplementowana jest w następujący sposób: Agent otrzymuję karę równą:

- **-1** za każdym razem gdy wykonuje ruch
- **-100** gdy "spadnie"z klifu

Nie jest trudno zauważyć najbardziej optymalną drogę dla agenta, jest ona następującej postaci:



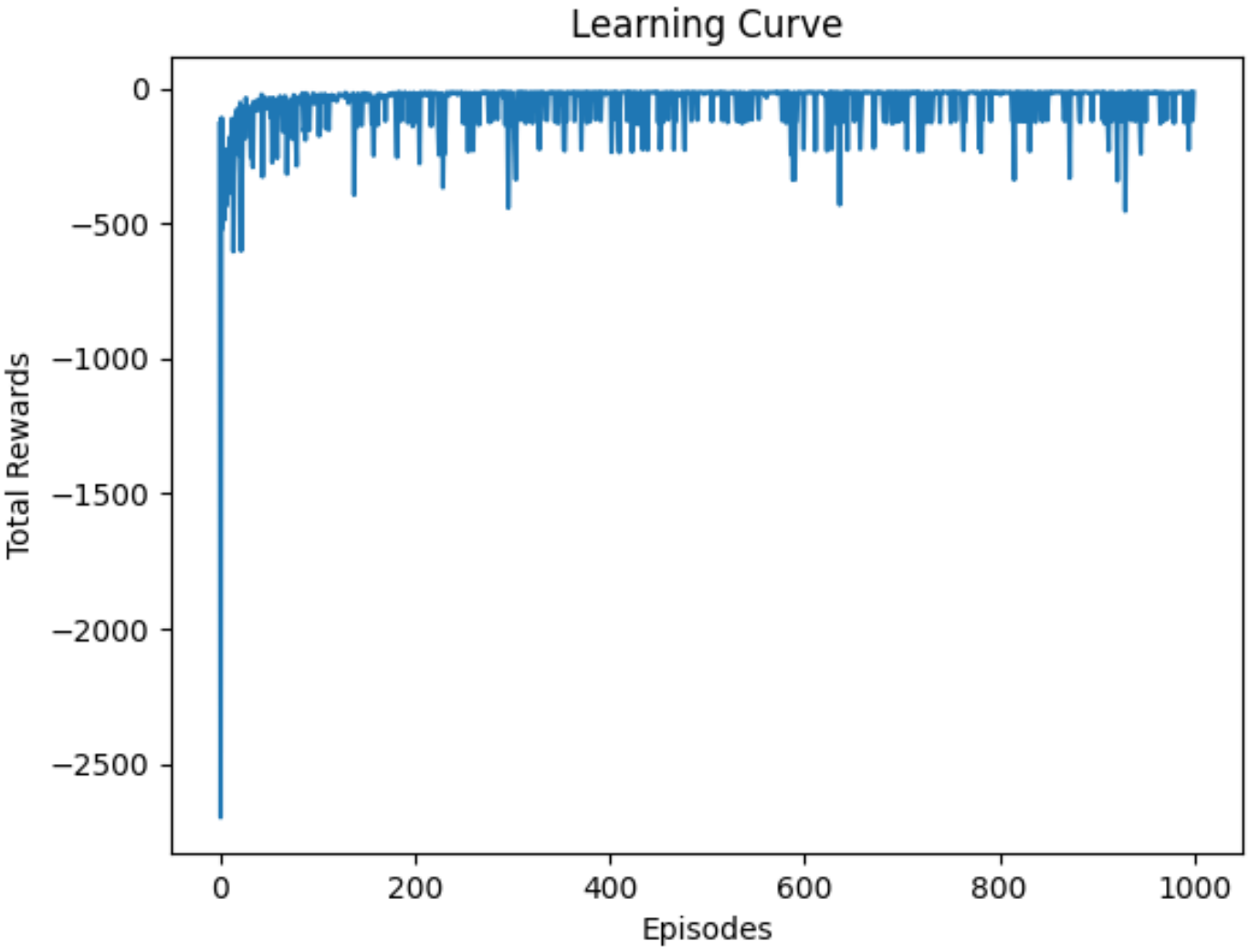
Po przejściu tej nagrody agent otrzyma tylko -13 punktów kary.

3 Q-learning z ϵ-zachłanną strategią - stały ϵ

W tym przypadku zaimplementowana została strategia, która dla każdego epizodu, korzysta ze stałej wartości ϵ przy decydowaniu czy następny krok, będzie losowym ruchem czy też zostanie wybrany ten o najwyższej ocenie. Sprawdźmy jak zachowuje się nasz algorytm dla następujących wartości hiperparametrów:

- Współczynnik dyskontowania $\lambda = 0.9$
- Szybkość uczenia $\beta = 0.1$
- Liczba epizodów: 1000

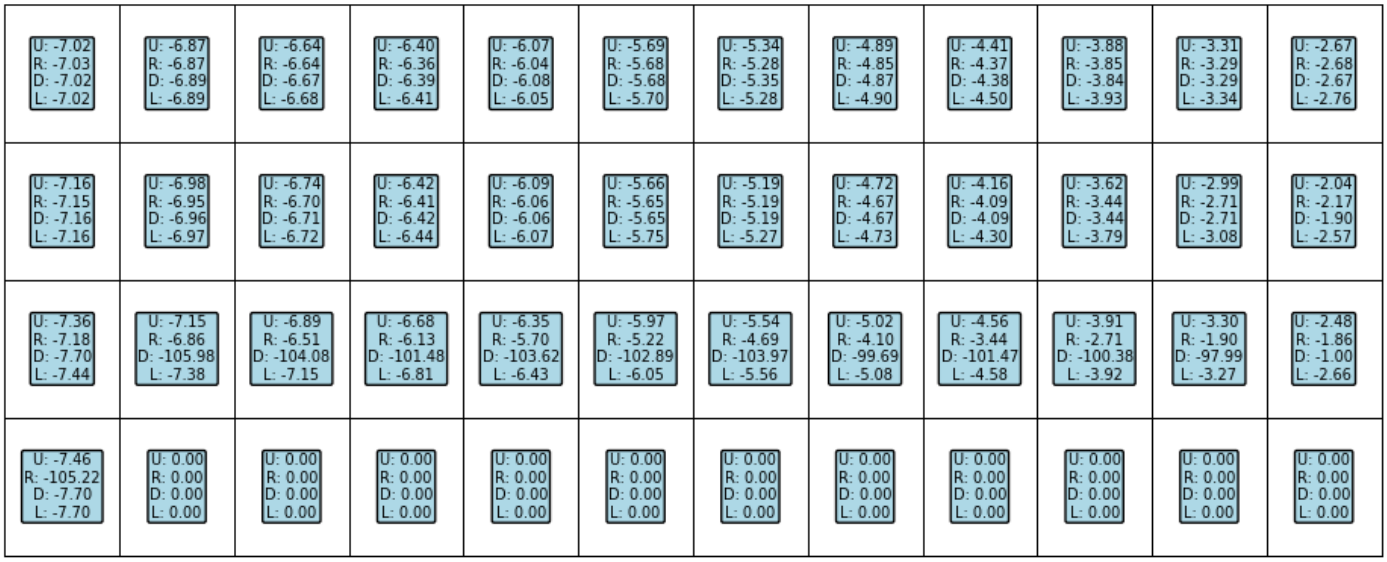
Zależność wysokości nagrody (w naszym przypadku kary) od numeru epizodu dla tych danych przestawia następujący wykres:



Widzimy, ze algorytm stosunkowo szybko sprowadza wartość tej nagrody do stosunkowo niskich wartości. Dla tego przypadku znalezienie optymalnej ścieżki waha się od 170 do 210 epizodów. Średnia wartość nagrody dla tych hiperparametrów wyniesie około -60. Można zadać sobie pytanie dlaczego przez około 800 kolejnych iteracji gdy algorytm zna już idealną ścieżkę, nie zawsze ją wybiera. Wpływa na to oczywiście parametr ϵ strategii zachłannej, który w naszym przypadku raz na 10 ruchów wybiera losową akcję zamiast tej najlepszej co prowadzi między innymi do przypadkowego wpadania na "klif"lub po prostu wybieraniu złej drogi. Wpływ na to może mieć również przypadkowe znalezienie optymalnego rozwiązania w początkowej fazie uczenia gdy nie wszystkie możliwe drogi zostały jeszcze odkryte. W naszym przypadku nagroda jest zawsze ujemna, więc tablica Q posiada tylko ujemne wartości. Wynika z tego, że dla danego stanu podczas wywołania:

```
action = np.argmax(q_table[observation])
```

gdy algorytm nie ma jeszcze obliczonej danej wartości stan-akcja, zostanie wybrana ta z wartością zero, czyli wcześniej nierozpatrywana. Wartości oceny każdego ruchu dla poszczególnych stanów (czyli wyniki naszej tablicy Q), prezentuje następująca mapa:



3.1 Badanie wpływu hiperparametrów na wynik

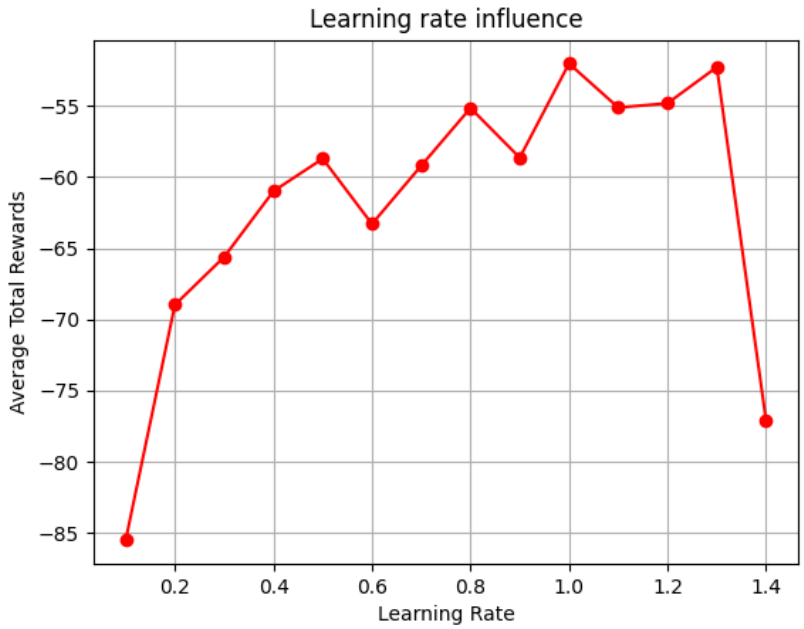
3.1.1 Tabela

Poniższa tabela przedstawia wpływ doboru hiperparametrów na efektywnośc uczenia: W celu uwiarygodnienia wyniku dla każdego zestawu hiperparametrów przeprowadzono 10 eksperymentów a otrzymane wyniki uśredniono:

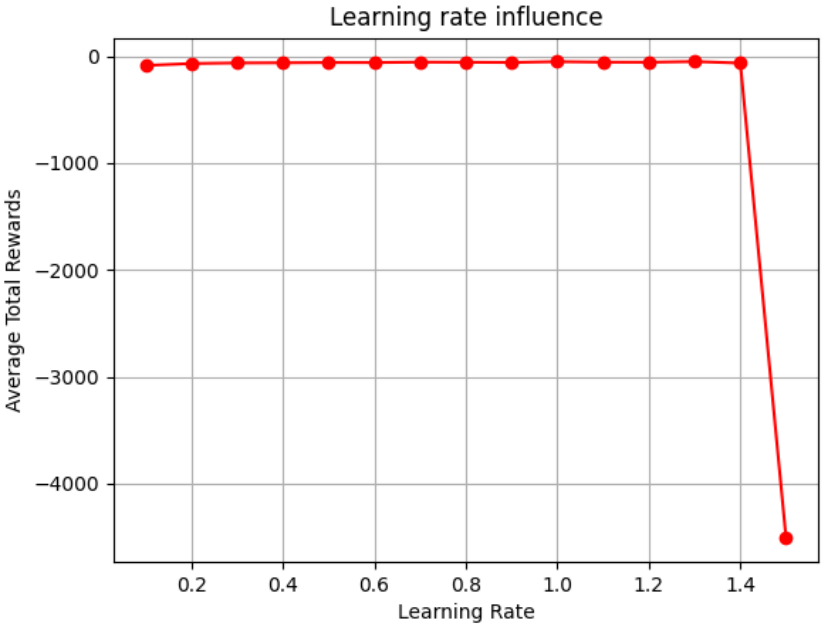
learning rate	discount factor	num of episodes	epsilon	first perfect solution	% of optimal solutions	average reward
0.1	0.9	1000	0.1	170th episode	27.66%	-60.86
0.3	0.9	1000	0.1	68th episode	33.18%	-54.09
0.5	0.9	1000	0.1	43th episode	35.07%	-52.53
0.7	0.9	1000	0.1	33th episode	34.90%	-53.46
0.9	0.9	1000	0.1	26th episode	34.79%	-50.6
0.1	0.1	1000	0.1	231th episode	25.91%	-84.73
0.1	0.3	1000	0.1	215th episode	26.70%	-69.65
0.1	0.5	1000	0.1	192th episode	26.97%	-65.58
0.1	0.7	1000	0.1	194th episode	26.65%	-62.1
0.4	0.4	1000	0.1	62th episode	33.96%	-55.2
0.7	0.7	1000	0.1	38th episode	34.44%	-51.97
1	1	1000	0.1	30th episode	36.44%	-47.25
0.3	0.9	10000	0.1	72th episode	36.22%	-47.76
0.3	0.9	50000	0.1	67th episode	36.18%	-46.23
0.3	0.9	5000	0.2	77th episode	11.87%	-92.73
0.3	0.9	5000	0.4	134th episode	0.95%	-196.6

3.1.2 Heatmapa - learning rate vs discount factor

3.1.3 Wpływ wartości learning rate na wynik



Kolejny wykres przedstawia sytuację gdy learning rate jest zbyt duży:



3.2 Wnioski

Przedstawiona tabela oraz heatmapa nie odzwierciedla w pełni efektywności nauki. Uśrednioną nagrodę mocno zaburza "wpadanie"na klif. Pierwsze znalezione optymalne rozwiązanie też nie jest wiarygodną oceną ponieważ, algorytm może trafić na nie losowo, nie wiedząc że jest optymalne. Procent optymalnych rozwiązań jest dość wiarygodną statystyką jednak często zaburzaną przez ϵ. W celu lepszej oceny użyteczności parametrów, dane na heatmapie były wykonywane dla 300 epizodów. Warto też wspomnieć jak źle pracuje algorytm gdy learning rate jest zbyt duży co przedstawiał powyższy wykres. W celu zbieżności nie należy ustalać wartości tego parametru na większe od jedynki.

3.3 Jak discount i learning factor wpływają na efektywność?

Learning rate:

- Nowe informacje: Wyższy współczynnik uczenia przypisuje większą wagę nowym informacjom podczas aktualizacji wartości Q. Oznacza to, że agent szybciej się dostosowuje do zmian w środowisku. Jednak zbyt wysoki współczynnik uczenia może prowadzić do niestabilności w działaniu.
- Zbieżność i działanie algorytmu: Gdy learning rate jest większy od jedynki mogą wystąpić problemy ze zbieżnością, między innymy przepełnienie (overflow), która uniemożliwi prawidłowe działanie programu. Nawet gdy algorytm wykona się prawidłowo, będzie on dla wartości > 1 mniej efektywny.

Discount factor:

- Wpływ na Nagrody Krótko- i Długoterminowe: Wyższy rabat przypisuje większą wagę nagrodom długoterminowym. Jeśli gamma jest blisko 1, agent bardziej uwzględnia przyszłe nagrody, co podkreśla znaczenie korzyści długoterminowych. Jednak jeśli gamma jest bliższa 0, agent bardziej skupia się na nagrodach krótkoterminowych.
- Efekt na Dylemat Eksploracji i Eksploatacji: Wyższe wartości gamma mogą zachęcać agenta do większej eksploracji w celu odkrycia optymalnej strategii długoterminowej.

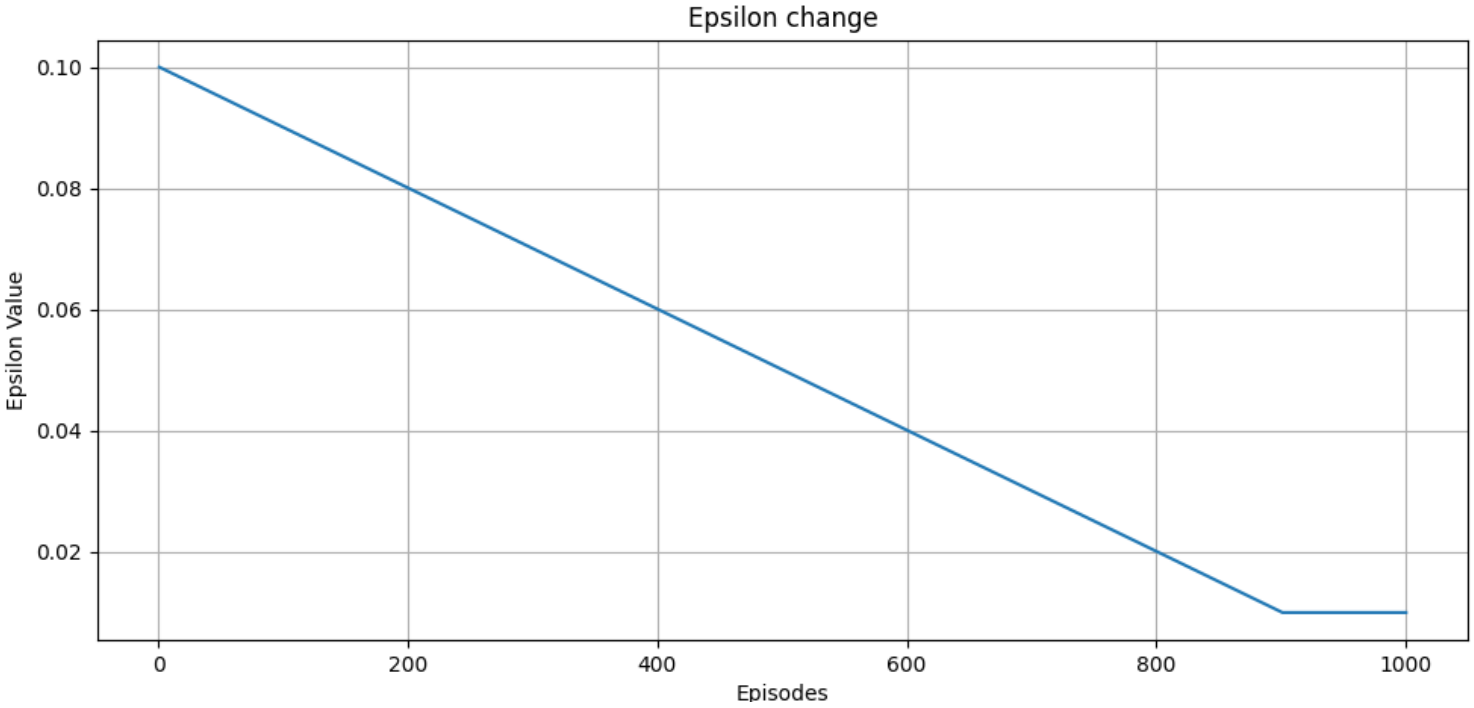
3.4 Eksploracja vs eksploatacja

Definiuje je parametr strategii zachłannej ϵ:

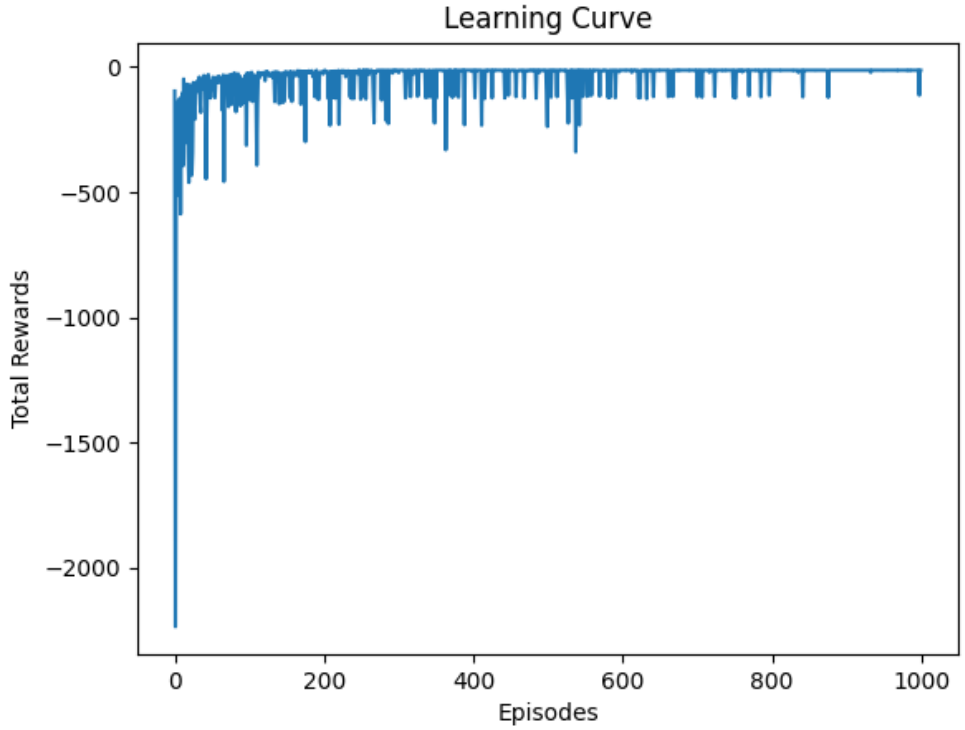
- Zachłanne wybieranie akcji, czyli takie gdzie epsilon jest bliski zeru, uniemożliwia przeszukanie przestrzeni rozwiązań.
- Całkowicie losowe wybieranie akcji,czyli epsilon bliski jedynce, nie korzysta ze zgromadzonej wiedzy. W nietrywialnych środowiskach można nie dojść do stanu akceptującego w rozsądnym czasie.

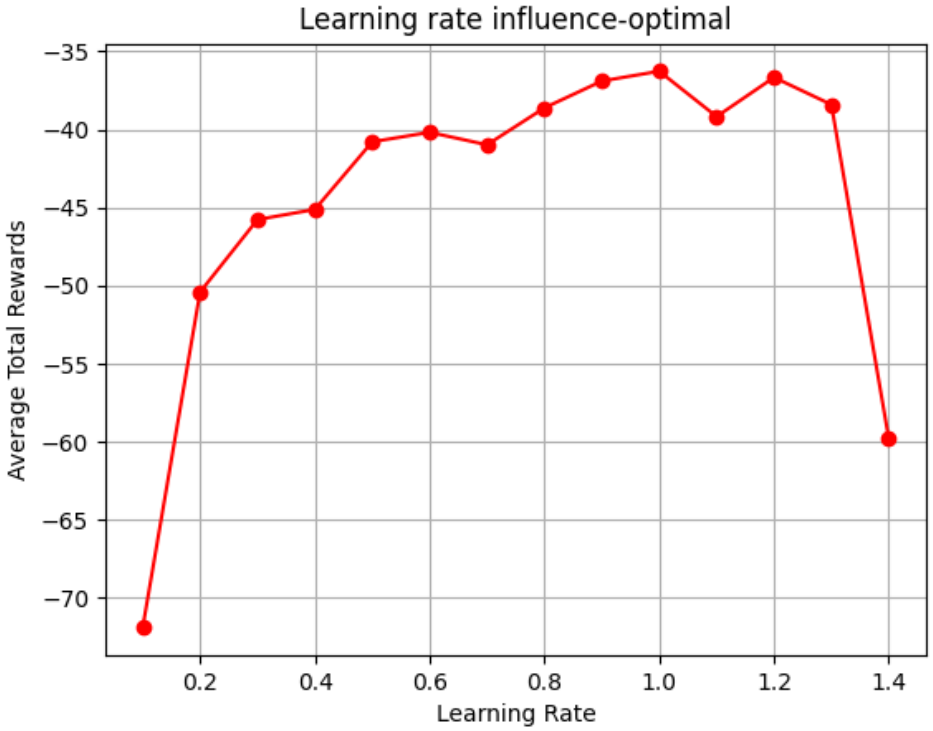
4 Strategia zachłanna - zmienny epsilon

Jak zauważono wcześniej bardzo ważnym parametrem uczenia jest epsilon. Jako, że chcielibyśmy by nasze akcje prowadziły jednocześnie do eksploracji jak i eksploatacji, zaimplementowano zmienny epsilon. Oznacza to, że w późniejszych epizodach będzie on bardziej stawiał na wybieranie najlepszego ruchu zamiast dalsze eksplorowanie poprzez wybieranie losowych ruchów. Przykładowa zmiana epsilona dla postępujących epizodów:



Zoptymalizowana krzywa uczenia się przedstawiająca wartość nagrody dla kolejnych epizodów:





wykres zależności średniej sumy nagórd od learning factor;

Oraz zwizualizowana talica Q dla tego przypadku:

Q-Values on Environment Grid

U: -6.91 R: -6.86 D: -6.96 L: -6.89	U: -6.75 R: -6.72 D: -6.77 L: -6.72	U: -6.61 R: -6.56 D: -6.53 L: -6.62	U: -6.33 R: -6.30 D: -6.38 L: -6.45	U: -6.05 R: -6.01 D: -5.98 L: -6.14	U: -5.80 R: -5.66 D: -5.69 L: -5.75	U: -5.33 R: -5.28 D: -5.32 L: -5.48	U: -4.88 R: -4.86 D: -4.88 L: -4.91	U: -4.53 R: -4.40 D: -4.41 L: -4.48	U: -3.86 R: -3.88 D: -3.88 L: -3.90	U: -3.44 R: -3.31 D: -3.31 L: -3.50	U: -2.80 R: -2.85 D: -2.68 L: -2.81
U: -7.02 R: -7.01 D: -7.13 L: -7.06	U: -6.80 R: -6.84 D: -6.88 L: -6.94	U: -6.69 R: -6.64 D: -6.66 L: -6.68	U: -6.40 R: -6.33 D: -6.38 L: -6.44	U: -6.09 R: -6.01 D: -6.03 L: -6.08	U: -5.63 R: -5.64 D: -5.65 L: -5.85	U: -5.27 R: -5.19 D: -5.19 L: -5.47	U: -4.82 R: -4.67 D: -4.67 L: -4.94	U: -4.39 R: -4.09 D: -4.09 L: -4.18	U: -3.76 R: -3.43 D: -3.43 L: -3.69	U: -3.02 R: -2.71 D: -2.71 L: -3.35	U: -2.08 R: -1.92 D: -1.90 L: -2.19
U: -7.21 R: -7.18 D: -7.56 L: -7.44	U: -7.02 R: -6.86 D: -106.07 L: -7.11	U: -6.89 R: -6.51 D: -104.45 L: -7.05	U: -6.54 R: -6.13 D: -93.51 L: -6.82	U: -6.27 R: -5.70 D: -99.79 L: -6.43	U: -5.97 R: -5.22 D: -102.13 L: -6.02	U: -5.64 R: -4.69 D: -105.95 L: -5.51	U: -5.03 R: -4.10 D: -93.82 L: -5.09	U: -4.58 R: -3.44 D: -102.18 L: -4.55	U: -3.89 R: -2.71 D: -102.01 L: -3.86	U: -2.59 R: -1.90 D: -97.71 L: -3.42	U: -2.62 R: -1.89 D: -1.00 L: -2.64
U: -7.46 R: -105.85 D: -7.70 L: -7.70	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00	U: 0.00 R: 0.00 D: 0.00 L: 0.00

Eksperymenty przeprowadzone dla parametrów:

- β: 0.1
- λ: 0.9
- Ilość epizodów: 1000
- Początkowy ϵ: 0.1
- Końcowy ϵ: 0.01
- Różnica między kolejnymi epsilonami: $\frac{Pocztkowyepsilon}{Liczbaepizodow}$

4.1 Wnioski

Testując algorytm można zauważyć, że wykorzystanie zmiennej wartości epsilona w strategii zachłannej niesie ze sobą wiele korzyści. Są to między innymi:

- Balansowanie Eksploracji i Eksploatacji: Zmienny parametr epsilon umożliwia dynamiczne dostosowywanie poziomu eksploracji i eksploatacji w trakcie uczenia się. Na początku algorytmu można zachęcać agenta do wybierania losowych akcji, a następnie stopniowo zmniejszać korzystanie z maksymalizacji tabeli oceny w miarę postępów w nauce.
- Szybsze Dostosowanie się do Optymalnej Strategii: Początkowe duże wartości epsilon skłaniają algorytm do eksploracji, co może pomóc w szybszym zlokalizowaniu optymalnej strategii. Następnie, w miarę upływu czasu, zmniejszenie epsilon pozwala na bardziej skoncentrowane eksploatowanie znanych już korzystnych akcji.
- Unikanie Zatrzymania się w Minimum Lokalnych: Zmienne epsilon pozwala uniknąć sytuacji, w której agent zbyt szybko zatrzyma się w suboptymalnej strategii, a zmienna eksploracja pozwoli na bardziej skuteczne przeszukiwanie przestrzeni akcji.