Sprawozdanie

WSI - lab 6
Q-learning algorithm

Mikołaj Taudul

1. Treść ćwiczenia

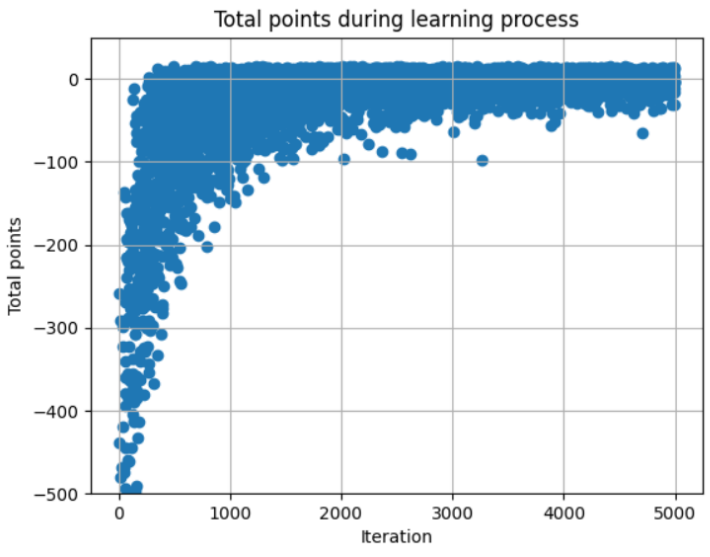
Celem ćwiczenia jest implementacja algorytmu Q-learning.
Następnie należy stworzyć agenta rozwiązującego problem Taxi
(https://gymnasium.farama.org/environments/toy_text/taxi/).

2. Założenia

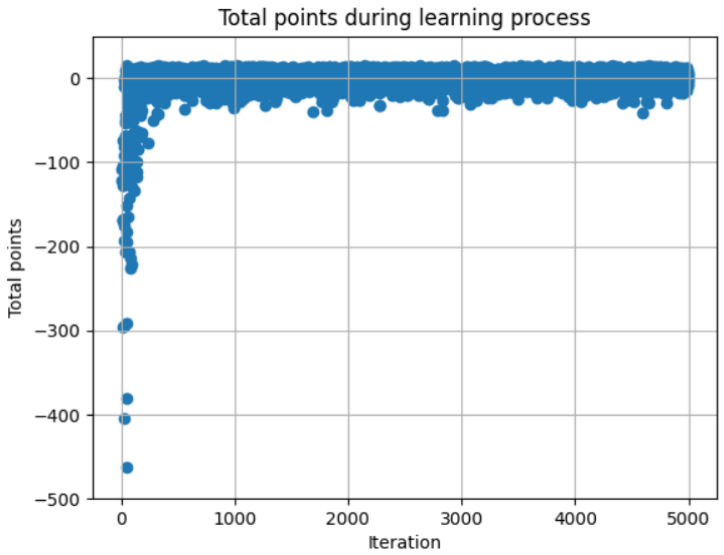
- Schemat nagradzania jest taki sam jak w bibliotece gym:
  - -1 za każdy krok, który nie kończy się wyższą nagrodą,
  - +20 za pomyślny przejazd pasażerem,
  - -10 za wykonanie operacji "pickup" lub "dropoff" w niedozwolonymi miejscu.
- Wykorzystywana jest strategia ϵ-zachłanna.
- Wykresy przedstawiają proces uczenia się algorytmu, czyli sumę nagród w każdej iteracji.
- Uczenie się będzie przebiegało przez 5000 iteracji.
- Funkcja evaluate() zwraca średni wynik dla 100 przypadków “gry” agenta po nauczeniu algorytmu.

3. Testowanie algorytmu dla różnych parametrów

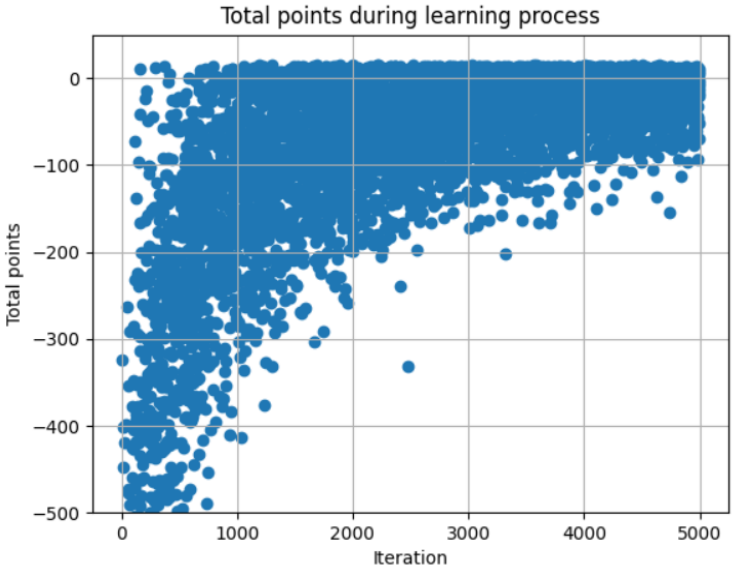
Wpływ learning rate



learning rate: 0.1
discount factor: 0.5
epsilon: 0.1



learning rate: 0.9
discount factor: 0.5
epsilon: 0.1

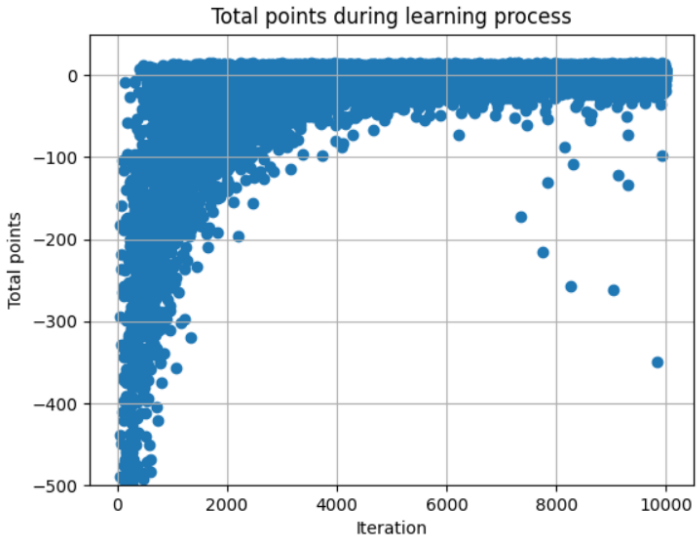


learning rate: 0.03
discount factor: 0.5
epsilon: 0.1

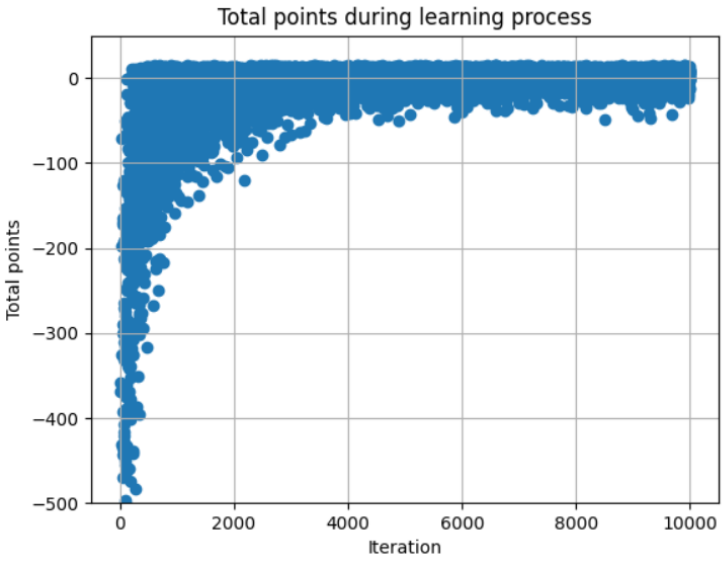
Zmniejszenie learning rate negatywnie wpływało na uzyskane wyniki.
Przy wyższym learning rate algorytm szybciej osiągał optymalne wartości.

Wpływ discount factor

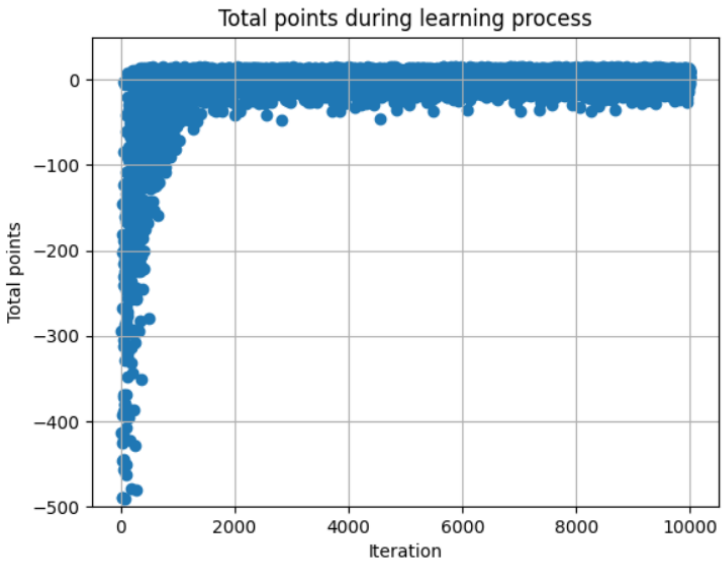
Zmniejszenie learning rate do 0.1, by lepiej zobrazować różnicę.



learning rate: 0.1
discount factor: 0.1
epsilon: 0.1



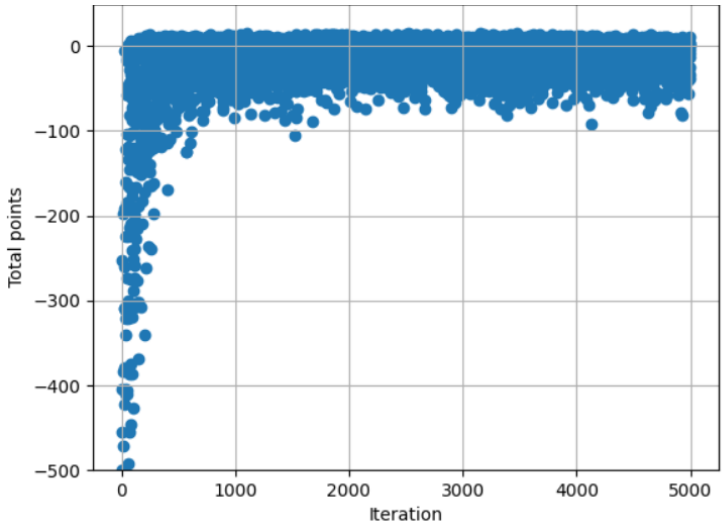
learning rate: 0.1
discount factor: 0.5
epsilon: 0.1



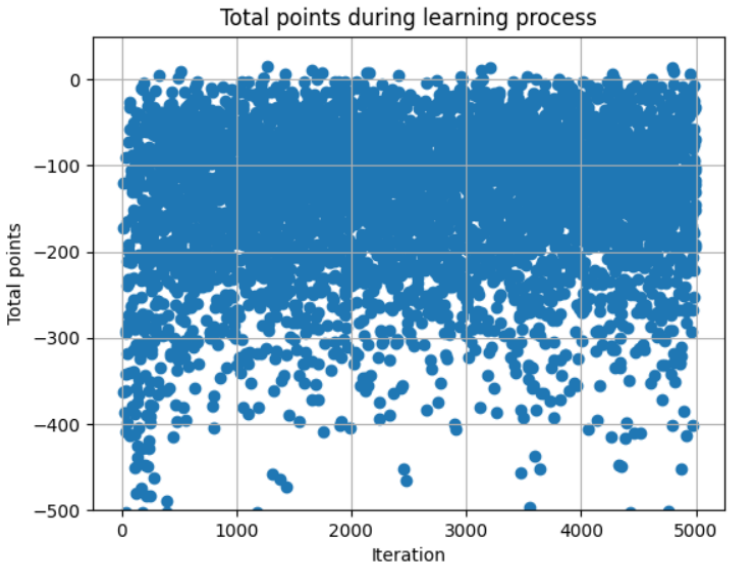
learning rate: 0.1
discount factor: 0.9
epsilon: 0.1

Zwiększanie discount factor wpływało pozytywnie na uzyskiwane wyniki. Przy zbyt niskim parametrze optymalne wyniki osiągane są wolniej i mimo praktycznie całkowitego nauczenia w późnych iteracjach momentami całkowita nagroda spada do bardzo niskich poziomów.

Wpływ epsilon



learning rate: 0.5
discount factor: 0.5
epsilon: 0.3



learning rate: 0.5
discount factor: 0.5
epsilon: 0.7

Przy zbyt wysokim epsilon, algorytm całkowicie przestawał się uczyć.

4. Wnioski

Learning rate

Algorytm przy zbyt niskich poziomach learning rate uczył się znacznie wolniej niż przy wyższych. Jest to parametr. który najbardziej ze wszystkich bezpośrednio wpływa na to jak szybko algorytm będzie uczył się przystosowywać do środowiska.

Discount factor

Jest to parametr, który przy niskim poziomie powoduje, że agent skupia się na nagrodach bieżących i ignoruje te w przyszłości, a przy wysokich wartościach odwrotnie W naszym przypadku niskie wartości powodowały delikatnie wolniejsze uczenie się agenta i pojawienie się przypadkowych niskich wartości nagrody mimo końcowego etapu uczenia się. Przy wyższych wartościach algorytm działał optymalnie.

Epsilon

Epsilon jest używany do określenia prawdopodobieństwa wyboru losowego ruchu w danej sytuacji. Przy wysokich wartościach agent bardzo często ruszał się losowo co całkowicie zachwiało proces uczenia się. Parametr ten powinien być niski, aby agent nie poruszał się ciągle losowo.