Patryk Bandyra

WSI

Zadanie 7

Uczenie ze wzmocnieniem

1. **Opis zadania**
   Zaimplementować algorytm Q-Learning i użyć go do wyznaczenia polityki decyzyjnej dla następującego zadania:

   Dysponujemy planszą NxN (domyślnie N=8), na której znajdują się dziury. Mamy dwie myszy – Pinky, który błądzi losowo po planszy oraz Brain, który uczy się przechodzić nasz labirynt (używa naszego algorytmu). Obie myszy zaczynają w tym samym zadanym punkcie planszy i wygrywają, jeśli dotrą do sera, który jest w innym punkcie planszy. Istnieje co najmniej jedna ścieżka od startu do końca. Myszy do swojego poruszania się wykorzystują metrykę Manhattan (lewo, prawo, góra, dół). Jeżeli mysz natrafi na dziurę to kończy bieg i przegrywa, analogicznie jak wejdzie na ser to wygrywa i nie kontynuuje swojej trasy. Celem agenta jest minimalizacja pokonywanej trasy.

2. **Opis algorytmu**
   Stany – pola na mapie (w przypadku mojej implementacji – macierzy NxN)
   Stan terminalny – pole z dziurą (przegrana) lub pole z serem (wygrana)
   Akcje – ruch agenta po mapie zgodnie z metryką Manhattan (lewo, prawo, góra, dół)
   Nagroda – wyznaczana dla każdego pola na mapie:
   - -100 pkt -> pole z dziurą
   - -1 pkt -> zwykłe pole
   - +100 pkt -> pole z serem

   Polityka – maksymalizacja nagrody poprzez wybór następnej akcji

   Do wyznaczania polityki użyłem zachłannego algorytmu Epsilon.

3. **Opis implementacji**
   W programie umożliwiłem wczytanie mapy z pliku csv oraz losową generację mapy o zadanych parametrach. Mapa losowo generowana jest następnie sprawdzana przez algorytm przeszukiwania grafu w głąb (DFS), czy istnieje przejście z zadanego punktu startowego do zadanego punktu końcowego. Jeśli takowa ścieżka nie istnieje, to program generuję mapę do skutku. Czasami warto uruchomić program ponownie, gdy cały czas pojawia się informacja o braku ścieżki na mapie, w celu zrestartowania ziarna w generatorze liczb losowych, co za zwyczaj rozwiązuje problem.

Za zbadanie zachowania Pinkiego na mapie odpowiada osobna klasa i funkcja, która dla zadanej mapy oraz liczby epizodów, zwraca informacje, czy i kiedy Pinkiemu udało się dojść do sera. Wyliczany jest stosunek liczby odniesionych sukcesów (dojście do sera) do całkowitej liczby epizodów.

Za zbadanie zachowania myszy o imieniu Brain (implementacja algorytmu Q-Learnig) dla zadanej mapy i parametrów odpowiada osobna funkcja, która zwraca nauczoną najkrótszą ścieżkę od startu do sera oraz macierze nagród i q wartości (oraz wyświetla mapę).

Wszelkie inne informacje odnośnie szczegółów implementacyjnych dostępne w dokładnie skomentowanym kodzie.

4. Wyniki działania programu

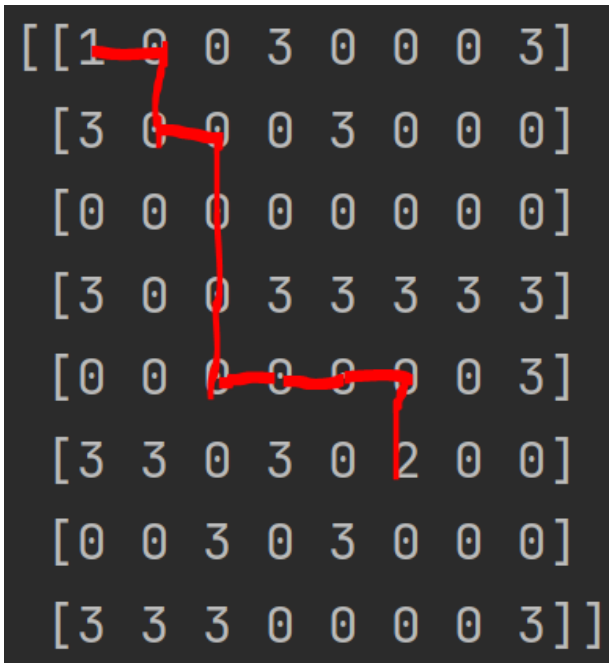
Zachowanie Braina dla mapy z pliku parametrów:

EPISODES = 10000
EPSILON = 0.9
DISCOUNT_FACTOR = 0.9
LEARNING_RATE = 0.9

```
[[1 0 0 3 0 0 0 3]
 [3 0 0 0 3 0 0 0]
 [0 0 0 0 0 0 0 0]
 [3 0 0 3 3 3 3 3]
 [0 0 0 0 0 0 0 3]
 [3 3 0 3 0 2 0 0]
 [0 0 3 0 3 0 0 0]
 [3 3 3 0 0 0 0 3]]
[[  -1.   -1.   -1. -100.   -1.   -1.   -1. -100.]
 [-100.   -1.   -1.   -1. -100.   -1.   -1.   -1.]
 [  -1.   -1.   -1.   -1.   -1.   -1.   -1.   -1.]
 [-100.   -1.   -1. -100. -100. -100. -100. -100.]
 [  -1.   -1.   -1.   -1.   -1.   -1.   -1. -100.]
 [-100. -100.   -1. -100.   -1.  100.   -1.   -1.]
 [  -1.   -1. -100.   -1. -100.   -1.   -1.   -1.]
 [-100. -100. -100.   -1.   -1.   -1.   -1. -100.]]
Training complete!
Shortest path from X:0, Y:0 to X:5, Y:5
[[0, 0], [0, 1], [1, 1], [1, 2], [2, 2], [3, 2], [4, 2], [4, 3], [4, 4], [4, 5], [5, 5]]
```

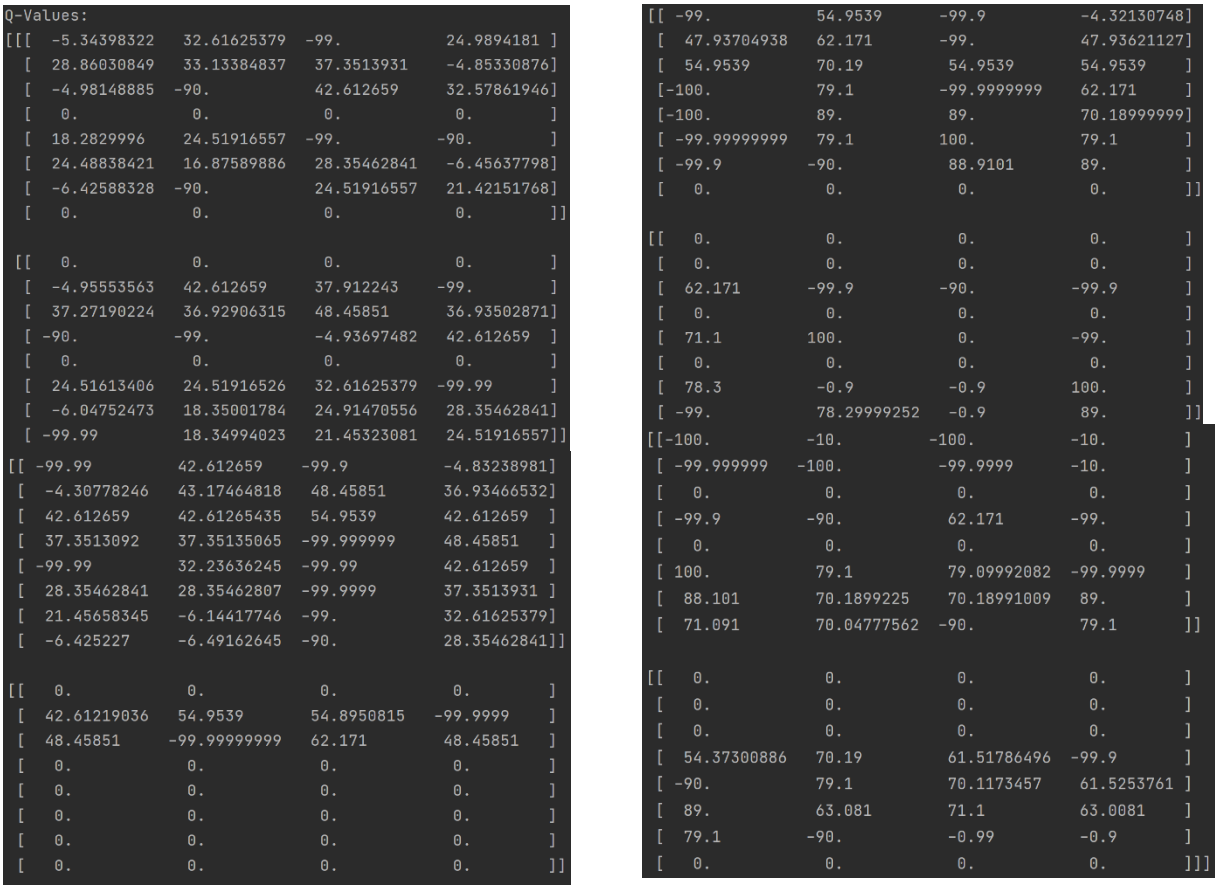
Mapa

Macierz nagród



Mapa z naniesioną najkrótszą trasą, którą nauczył się Brain

Macierz 3D q-wartości, która służy do wyboru następnej akcji dla danego stanu



```
Q-Values:
[[[  -5.34398322   32.61625379  -99.           24.9894181 ]
  [  28.86030849   33.13384837   37.3513931    -4.85330876]
  [  -4.98148885  -90.           42.612659     32.57861946]
  [   0.            0.            0.            0.        ]
  [  18.2829996    24.51916557  -99.          -90.        ]
  [  24.48838421   16.87589886   28.35462841   -6.45637798]
  [  -6.42588328  -90.           24.51916557   21.42151768]
  [   0.            0.            0.            0.        ]]

 [[   0.            0.            0.            0.        ]
  [  -4.95553563   42.612659     37.912243    -99.        ]
  [  37.27190224   36.92906315   48.45851      36.93502871]
  [ -90.          -99.           -4.93697482   42.612659  ]
  [   0.            0.            0.            0.        ]
  [  24.51613406   24.51916526   32.61625379  -99.99      ]
  [  -6.04752473   18.35001784   24.91470556   28.35462841]
  [ -99.99         18.34994023   21.45323081   24.51916557]]
[[ -99.99         42.612659    -99.9          -4.83238981]
 [  -4.30778246   43.17464818   48.45851      36.93466532]
 [  42.612659     42.61265435   54.9539       42.612659  ]
 [  37.3513092    37.35135065  -99.999999     48.45851   ]
 [ -99.99         32.23636245  -99.99         42.612659  ]
 [  28.35462841   28.35462807  -99.9999       37.3513931 ]
 [  21.45658345   -6.14417746  -99.           32.61625379]
 [  -6.425227     -6.49162645  -90.           28.35462841]]

[[   0.            0.            0.            0.        ]
 [  42.61219036   54.9539       54.8950815   -99.9999    ]
 [  48.45851     -99.99999999   62.171        48.45851   ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]]
```

```
[[ -99.           54.9539      -99.9          -4.32130748]
 [  47.93704938   62.171       -99.           47.93621127]
 [  54.9539       70.19         54.9539       54.9539    ]
 [-100.           79.1         -99.9999999    62.171     ]
 [-100.           89.           89.           70.18999999]
 [ -99.99999999   79.1         100.           79.1       ]
 [ -99.9         -90.           88.9101       89.        ]
 [   0.            0.            0.            0.        ]]

[[   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [  62.171       -99.9         -90.          -99.9       ]
 [   0.            0.            0.            0.        ]
 [  71.1         100.            0.          -99.        ]
 [   0.            0.            0.            0.        ]
 [  78.3          -0.9          -0.9         100.        ]
 [ -99.          78.29999252    -0.9          89.        ]]
[[-100.          -10.         -100.          -10.        ]
 [ -99.999999   -100.          -99.9999      -10.        ]
 [   0.            0.            0.            0.        ]
 [ -99.9         -90.           62.171       -99.        ]
 [   0.            0.            0.            0.        ]
 [ 100.           79.1          79.09992082  -99.9999    ]
 [  88.101        70.1899225    70.18991009   89.        ]
 [  71.091        70.04777562  -90.           79.1       ]]

[[   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [  54.37300886   70.19         61.51786496  -99.9       ]
 [ -90.           79.1          70.1173457    61.5253761 ]
 [  89.           63.081        71.1          63.0081    ]
 [  79.1         -90.           -0.99         -0.9       ]
 [   0.            0.            0.            0.        ]]]
```

Zachowanie Pinkiego w tym samym środowisku (te same parametry i mapa):

```
Win! Pinky got to X:5, Y:5 at episode 3080
Win! Pinky got to X:5, Y:5 at episode 7394
Win-Failure Ratio: 0.0002 wins
```

Z przeprowadzonych badań wynika, że ratio nie przekracza wartości 0.001 (dla 10000 epizodów).

Natomiast Brain po ponad około 120-150 + epizodach jest już nauczony najkrótszej ścieżki i za każdym razem jest w stanie dojść do celu.

Przykładowe wyniki długości ścieżek nauczonych się przez Brainiego dla danej liczby epizodów:

Liczba epizodów	100	120	10000+
Długość ścieżki	12	11	11

Badanie wpływu parametru EPSILON (czyli jak często wybieramy najlepszą możliwość przy wyborze kolejnego stanu):

Z badania wynik, że im mniejszy parametr EPSILON, tym dłużej (więcej epizodów) zajmuje znalezienie najlepszej ścieżki, tak samo jak w przypadku zbyt dużej wartości tego parametru. Optimum znalazłem dla wartości ~ 0.9.

Badanie wpływu tempa uczenia (parametru LEARNING_RATE) dla mapy z pliku:

Dla mapy z pliku dla 1000 epizodów nie zauważyłem żadnych różnic. Prawdopodobnie byłaby potrzebna większa mapa dla takiej ilości epizodów, aby zaobserwować znaczące różnice w tempie wyznaczania najkrótszej ścieżki.

Badanie wpływu parametru DISCOUNT_FACTOR:

Znowu dla mapy z pliku 8X8 nie zauważyłem wpływu na jakość ścieżki dla 1000 epizodów.