

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления» Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Методы машинного обучения»

Отчет по лабораторной работе №5 «Обучение на основе временны'х различий»

Выполнила: студент группы ИУ5-22М Павловская А.А. 12.05.2023

Проверил: преподаватель каф. ИУ5 Гапанюк Ю.Е.

Цель работы: ознакомление с базовыми методами обучения с подкреплением на основе временных различий.

Задание:

На основе рассмотренного на лекции примера реализуйте следующие алгоритмы:

- SARSA
- Q-обучение
- Двойное Q-обучение

для любой среды обучения с подкреплением (кроме рассмотренной на лекции среды Toy Text / Frozen Lake) из библиотеки Gym (или аналогичной библиотеки).

Ход работы

Для реализации алгоритмов SARSA, Q-обучение и двойное Q-обучение была выбрана среда обучения с подкреплением Cliff Walking из библиотеки Gym. Агент может находиться в 48 состояниях и осуществлять 4 действия.

Текст программы

```
import numpy as np
import matplotlib.pyplot as plt
import gym
from tqdm import tqdm
# ****** БАЗОВЫЙ АГЕНТ
************
class BasicAgent:
   Базовый агент, от которого наследуются стратегии обучения
   # Наименование алгоритма
   ALGO NAME = '---'
   def __init__(self, env, eps=0.1):
       # Среда
       self.env = env
       # Размерности Q-матрицы
       self.nA = env.action space.n
       self.nS = env.observation space.n
       #и сама матрица
       self.Q = np.zeros((self.nS, self.nA))
       # Значения коэффициентов
       # Порог выбора случайного действия
       self.eps=eps
       # Награды по эпизодам
       self.episodes_reward = []
```

```
def print_q(self):
        print('Вывод Q-матрицы для алгоритма ', self.ALGO_NAME)
        print(self.Q)
    def get_state(self, state):
        Возвращает правильное начальное состояние
        if type(state) is tuple:
            # Если состояние вернулось с виде кортежа, то вернуть только номер
состояния
            return state[0]
        else:
            return state
    def greedy(self, state):
        <<Жадное>> текущее действие
        Возвращает действие, соответствующее максимальному Q-значению
        для состояния state
        return np.argmax(self.Q[state])
    def make action(self, state):
        Выбор действия агентом
        if np.random.uniform(0,1) < self.eps:</pre>
            # Если вероятность меньше ерѕ
            # то выбирается случайное действие
            return self.env.action space.sample()
        else:
            # иначе действие, соответствующее максимальному Q-значению
            return self.greedy(state)
    def draw_episodes_reward(self):
        # Построение графика наград по эпизодам
        fig, ax = plt.subplots(figsize = (15,10))
        y = self.episodes_reward
        x = list(range(1, len(y)+1))
        plt.plot(x, y, '-', linewidth=1, color='green')
        plt.title('Награды по эпизодам')
        plt.xlabel('Номер эпизода')
        plt.ylabel('Награда')
        plt.show()
```

```
def learn():
       Реализация алгоритма обучения
       pass
# ******* SARSA
************
class SARSA Agent(BasicAgent):
   Реализация алгоритма SARSA
   # Наименование алгоритма
   ALGO_NAME = 'SARSA'
   def __init__(self, env, eps=0.4, lr=0.1, gamma=0.98, num_episodes=20000):
       # Вызов конструктора верхнего уровня
       super().__init__(env, eps)
       # Learning rate
       self.lr=lr
       # Коэффициент дисконтирования
       self.gamma = gamma
       # Количество эпизодов
       self.num_episodes=num_episodes
       # Постепенное уменьшение ерѕ
       self.eps decay=0.00005
       self.eps_threshold=0.01
   def learn(self):
       Обучение на основе алгоритма SARSA
       self.episodes_reward = []
       # Цикл по эпизодам
       for ep in tqdm(list(range(self.num_episodes))):
           # Начальное состояние среды
           state = self.get_state(self.env.reset())
           # Флаг штатного завершения эпизода
           done = False
           # Флаг нештатного завершения эпизода
           truncated = False
           # Суммарная награда по эпизоду
           tot_rew = 0
           # По мере заполнения Q-матрицы уменьшаем вероятность случайного
выбора действия
           if self.eps > self.eps_threshold:
               self.eps -= self.eps_decay
```

```
# Выбор действия
           action = self.make_action(state)
           # Проигрывание одного эпизода до финального состояния
           while not (done or truncated):
               # Выполняем шаг в среде
               next_state, rew, done, truncated, _ = self.env.step(action)
               # Выполняем следующее действие
               next_action = self.make_action(next_state)
               # Правило обновления Q для SARSA
               self.Q[state][action] = self.Q[state][action] + self.lr * \
                   (rew + self.gamma * self.Q[next_state][next_action] -
self.Q[state][action])
               # Следующее состояние считаем текущим
               state = next_state
               action = next action
               # Суммарная награда за эпизод
               tot rew += rew
               if (done or truncated):
                   self.episodes_reward.append(tot_rew)
# ******* 0-обучение
*************
class QLearning_Agent(BasicAgent):
   Реализация алгоритма Q-Learning
   # Наименование алгоритма
   ALGO_NAME = 'Q-обучение'
   def __init__(self, env, eps=0.4, lr=0.1, gamma=0.98, num_episodes=20000):
       # Вызов конструктора верхнего уровня
       super().__init__(env, eps)
       # Learning rate
       self.lr=lr
       # Коэффициент дисконтирования
       self.gamma = gamma
       # Количество эпизодов
       self.num_episodes=num_episodes
       # Постепенное уменьшение ерѕ
       self.eps decay=0.00005
       self.eps_threshold=0.01
```

```
def learn(self):
       Обучение на основе алгоритма Q-Learning
       self.episodes_reward = []
       # Цикл по эпизодам
       for ep in tqdm(list(range(self.num_episodes))):
           # Начальное состояние среды
           state = self.get_state(self.env.reset())
           # Флаг штатного завершения эпизода
           done = False
           # Флаг нештатного завершения эпизода
           truncated = False
           # Суммарная награда по эпизоду
           tot_rew = 0
           # По мере заполнения Q-матрицы уменьшаем вероятность случайного
выбора действия
           if self.eps > self.eps_threshold:
               self.eps -= self.eps_decay
           # Проигрывание одного эпизода до финального состояния
           while not (done or truncated):
               # Выбор действия
               # B SARSA следующее действие выбиралось после шага в среде
               action = self.make action(state)
               # Выполняем шаг в среде
               next_state, rew, done, truncated, _ = self.env.step(action)
               # Правило обновления Q для SARSA (для сравнения)
               # self.Q[state][action] = self.Q[state][action] + self.lr * \
                     (rew + self.gamma * self.Q[next state][next action] -
self.Q[state][action])
               # Правило обновления для Q-обучения
               self.Q[state][action] = self.Q[state][action] + self.lr * \
                   (rew + self.gamma * np.max(self.Q[next_state]) -
self.Q[state][action])
               # Следующее состояние считаем текущим
               state = next state
               # Суммарная награда за эпизод
               tot_rew += rew
               if (done or truncated):
                   self.episodes reward.append(tot rew)
# ****** Двойное Q-обучение
************
```

```
class DoubleQLearning_Agent(BasicAgent):
    Peaлизация алгоритма Double Q-Learning
    # Наименование алгоритма
    ALGO_NAME = 'Двойное Q-обучение'
    def __init__(self, env, eps=0.4, lr=0.1, gamma=0.98, num_episodes=20000):
        # Вызов конструктора верхнего уровня
        super().__init__(env, eps)
        # Вторая матрица
        self.Q2 = np.zeros((self.nS, self.nA))
        # Learning rate
        self.lr=lr
        # Коэффициент дисконтирования
        self.gamma = gamma
        # Количество эпизодов
        self.num episodes=num episodes
        # Постепенное уменьшение ерѕ
        self.eps_decay=0.00005
        self.eps threshold=0.01
    def greedy(self, state):
        <<Жадное>> текущее действие
        Возвращает действие, соответствующее максимальному Q-значению
        для состояния state
        temp q = self.Q[state] + self.Q2[state]
        return np.argmax(temp_q)
    def print q(self):
        print('Вывод Q-матриц для алгоритма ', self.ALGO_NAME)
        print('Q1')
        print(self.Q)
        print('Q2')
        print(self.Q2)
    def learn(self):
        1.1.1
        Обучение на основе алгоритма Double Q-Learning
        self.episodes_reward = []
        # Цикл по эпизодам
        for ep in tqdm(list(range(self.num episodes))):
            # Начальное состояние среды
            state = self.get_state(self.env.reset())
```

```
# Флаг штатного завершения эпизода
            done = False
            # Флаг нештатного завершения эпизода
            truncated = False
            # Суммарная награда по эпизоду
            tot_rew = 0
            # По мере заполнения Q-матрицы уменьшаем вероятность случайного
выбора действия
            if self.eps > self.eps threshold:
                self.eps -= self.eps_decay
            # Проигрывание одного эпизода до финального состояния
            while not (done or truncated):
                # Выбор действия
                # B SARSA следующее действие выбиралось после шага в среде
                action = self.make_action(state)
                # Выполняем шаг в среде
                next_state, rew, done, truncated, _ = self.env.step(action)
                if np.random.rand() < 0.5:</pre>
                    # Обновление первой таблицы
                    self.Q[state][action] = self.Q[state][action] + self.lr * \
                        (rew + self.gamma *
self.Q2[next_state][np.argmax(self.Q[next_state])] - self.Q[state][action])
                else:
                    # Обновление второй таблицы
                    self.Q2[state][action] = self.Q2[state][action] + self.lr * \
                        (rew + self.gamma *
self.Q[next_state][np.argmax(self.Q2[next_state])] - self.Q2[state][action])
                # Следующее состояние считаем текущим
                state = next_state
                # Суммарная награда за эпизод
                tot rew += rew
                if (done or truncated):
                    self.episodes_reward.append(tot_rew)
def play_agent(agent):
   Проигрывание сессии для обученного агента
   env2 = gym.make('CliffWalking-v0', render_mode='human')
   state = env2.reset()[0]
   done = False
   while not done:
        action = agent.greedy(state)
        next_state, reward, terminated, truncated, _ = env2.step(action)
```

```
env2.render()
        state = next_state
        if terminated or truncated:
            done = True
def run_sarsa():
    env = gym.make('CliffWalking-v0')
    agent = SARSA_Agent(env)
    agent.learn()
    agent.print_q()
    agent.draw_episodes_reward()
    play_agent(agent)
def run_q_learning():
    env = gym.make('CliffWalking-v0')
    agent = QLearning Agent(env)
    agent.learn()
    agent.print_q()
    agent.draw_episodes_reward()
    play_agent(agent)
def run double q learning():
    env = gym.make('CliffWalking-v0')
    agent = DoubleQLearning_Agent(env)
    agent.learn()
    agent.print_q()
    agent.draw_episodes_reward()
    play_agent(agent)
def main():
    #run_sarsa()
    #run q learning()
    run_double_q_learning()
if __name__ == '__main__':
    main()
```

Результаты выполнения программы

Результаты выполнения алгоритма SARSA представлены на рис.1 – рис.2.

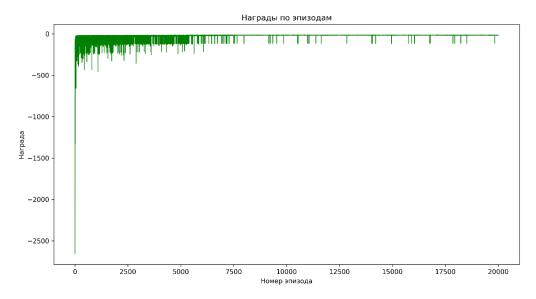


Рис.1. Награды по эпизодам для SARSA

```
20000/20000 [00:09<00:00, 2190.48it/s]
100%
                               SARSA
Вывод Q-матрицы для алгоритма
[[ -13.25582922
                               -14.17583316
                                              -13,277820981
                -12.46163893
   -12.44632902
                -11.71800849
                               -13.31377036
                                             -13.42574227
   -11.78157216
                 -10.94705462
                               -12.82689432
                                             -12.66068894]
                               -12.78838645
   -10.98077049
                 -10.16240312
                                              -11.94304846
   -10.11944569
                  -9.38137596
                               -11.12845859
                                             -11.19863056]
                  -8.37829523
                               -10.16316747
   -9.27669495
                                             -10.302386041
   -8.40533649
                  -7,49223496
                                -9.3198174
                                               -9.516390031
   -7.70579897
                  -6.61162226
                                -8.45453782
                                               -8.58616608]
    -6.64835534
                  -5.73182591
                                -7.54453378
                                               -7.71650124]
   -5.80229543
                  -4.84381419
                                -5.14539325
                                               -6.938888791
   -4.83658592
                  -3.93321176
                                -4.24038294
                                               -5.97106441
    -3.92221338
                  -3.94899549
                                -2.98346832
                                               -5.04741363]
   -13.18996671
                 -13.72531009
                               -14.87722968
                                              -13.94440274]
   -12.5132878
                 -16.46536352
                               -17.19955602
                                              -16.88536676]
                                             -17.75783804]
   -11.76635568
                -16.12533228
                               -26.21904325
   -10.86762207
                 -15.4836821
                               -25.2166053
                                              -18.64855154]
                                              -17.31070584]
   -10.11637945
                 -13.83347301
                               -18.72651819
   -9.26088323
                 -13.69390836
                               -20.57168235
                                             -15.99612363]
   -8.40473034
                 -12.23141527
                               -18.78851112
                                              -13.771024371
   -10.67299141
                  -7.52398553
                               -32.14211492
                                             -13.7851238
   -6.63659247
                  -8.74664086
                               -11.3922803
                                              -14.98931734]
    -7.69294958
                  -3.96989538
                               -19.21817286
                                               -9.82536994]
                  -2.94380558
    -7,47701689
                               -14,49592358
                                               -8.630219361
   -4.11649942
                  -2.9740145
                                -1,99719673
                                              -3.92397895
   -13.91158914
                 -14.60965823
                               -15.81912313
                                              -14.70280972]
   -13.30578242
                -19.4650053 -123.67736699
                                             -20.6192555 ]
   -16.07086469
                 -23.54777759 -113.061171
                                              -19.69449061]
   -16.34270314
                -27.17552878 -113.00142024
                                             -21.615561591
   -15.08799885
                 -18.6236901 -103.25057644
                                             -18.1289854
   -14.77278507
                 -15.45345158 -105.11053505
                                             -15.30135648
                 -14.46477813
   -13.22951743
                              -90.65674021
                                             -17.31356205]
   -15.28998759
                 -12.27386921 -125.77577708
                                              -19.69403395]
   -9.43457999
                 -14.74287458 -98.17945055
                                             -12.97431465]
    -8.03185582
                 -20.90870451 -107.5210636
                                              -18.559274361
    -5.87569736
                  -2.01851744 -128.01423467
                                              -16.89168122]
    -3.19506231
                  -2.01973396
                                               -3.03961934]
                               -1.
                                              -15.40362451]
   -14.62815318 -114.54231619
                               -15.43955938
     0.
                   0.
                                 0.
                                                0.
                   0.
                                 0.
     0.
                   0.
                                 0.
                                                0.
     0.
                   0.
                                 0.
                                                0.
     0.
                   0.
                                 0.
                                                0.
     0.
                   0.
                                 0.
                                                0.
                   0.
                                                0.
     0.
                   0.
                                 0.
     0.
                   0.
                                 0.
                                                0.
     0.
                   0.
                                 0.
                                                0.
     0.
                   0.
                                 0.
                                                0.
                                                          ]]
```

Рис.2. Q-матрица для алгоритма SARSA

Результаты выполнения алгоритма Q-обучение представлены на рис.3 – рис.4.

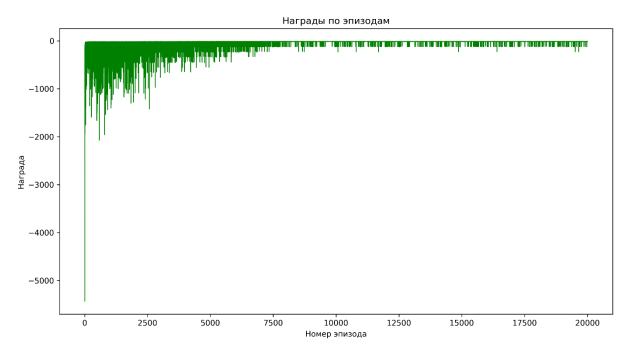


Рис.3. Награды по эпизодам для алгоритма Q-обучение

	1	, ,	, , , , ,		1	•	
100%					20000/20000	[00:08<00:00,	2274.21it/sl
Вывод Q-матрицы	для алгоритма	Q-обучение				[,	,,
[[-12.65921328	-12.30096258	-12.30221927	-12.56695728	3]			
[-12.00171097	-11.5484309	-11.54843835	-12.28046965	i			
-11.45454642	-10.76414372	-10.76414403	-12.02586187	-			
[-10.69694291	-9.96343061	-9.96343045	-11.47599036	5]			
-9.92779078	-9.1463594	-9.14635943	-10.70808049)			
[-9.12742885	-8.31261184	-8.31261184	-9.92688952	2]			
[-8.30948577	-7.46184886	-7.46184886	-9.10570262	<u>.</u>			
[-7.43610708	-6.59372334	-6.59372334	-8.2882669]			
[-6.58746673	-5.70788096	-5.70788096	-7.45216032	2]			
[-5.702146	-4.80396016	-4.80396016	-6.58501788	3]			
[-4.77747881	-3.881592	-3.881592	-5.69588611	.]			
[-3.87740964	-3.87406827	-2.9404	-4.7674234]			
[-13.03731736	-11.54888054	-11.54888054	-12.31773022	!]			
[-12.31643898	-10.76416381	-10.76416381	-12.31789487	']			
[-11.54881444	-9.96343246	-9.96343246	-11.54888022	2]			
[-10.76415658	-9.14635966	-9.14635966	-10.76416381	.]			
[-9.96343142	-8.31261189	-8.31261189	-9.96343246	5]			
[-9.14635895	-7.46184887	-7.46184887	-9.14635965	-			
[-8.31261184	-6.59372334	-6.59372334	-8.31261188	-			
[-7.46184884	-5.70788096	-5.70788096	-7.46184881	-			
[-6.59372331	-4.80396016	-4.80396016	-6.59372334	-			
[-5.70788082	-3.881592	-3.881592	-5.70788093	-			
[-4.80396013	-2.9404	-2.9404	-4.80396015	5]			
[-3.881592	-2.9404	-1.98	-3.881592]			
[-12.31790293	-10.76416381	-12.31790293	-11.54888054	-			
[-11.54888054		-111.31790293	-11.54888054	-			
[-10.76416381		-111.31790293	-10.76416381	-			
[-9.96343246		-111.31790293	-9.96343246	-			
[-9.14635966		-111.31790293	-9.14635966	-			
[-8.31261189		-111.31790293	-8.31261189	-			
[-7.46184887		-111.31790293	-7.46184887	-			
[-6.59372334		-111.31790293	-6.59372334	-			
[-5.70788096 [-4.80396016	-3.881592 -2.9404	-111.31790293 -111.31790293	-5.70788096 -4.80396016	-			
[-4.80396016 [-3.881592	-1.98	-111.31790293	-3.881592)]			
[-2.9404	-1.98	-111.31/90293	-2.9404	1			
-	-111.31790293	-12.31790293	-12.31790293	1			
[0.	0.	0.	0.	1			
[0.	0.	0.	0.	1			
[0.	0.	0.	0.	1			
[0.	0.	0.	0.	í			
[0.	0.	0.	0.	i			
[0.	0.	0.	0.	i			
[0.	0.	0.	0.	j			
[0.	0.	0.	0.	j			
[0.	0.	0.	0.	j			
[0.	0.	0.	0.	j			
[0.	0.	0.	0.]]			
	_						

Рис.4. Q-матрица для алгоритма Q-обучение

Результаты выполнения алгоритма двойное Q-обучение представлены на рис.5 – рис.7.

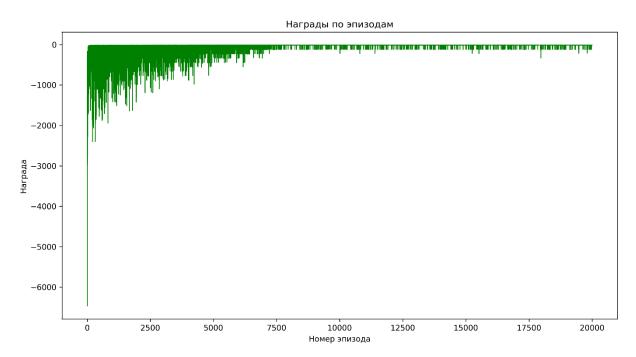


Рис. 5. Награды по эпизодам для алгоритма двойное Q-обучение

```
100%|
Вывод Q-матриц для алгоритма Двойное Q-обучение
[[ -13.26802076 -13.08688301 -12.32003079 -13.28602916]
  -12.76173276 -12.39008676 -11.54941937 -13.25528846]
  -11.83872158 -12.23704987 -10.78349991 -12.45482904]
 [ -11.43865788 -10.68082469
                             -9.96746322 -11.6858199 ]
  -10.16180866 -10.08517639
                              -9.14800047 -11.20989426]
   -9.86379006
                -9.17698199
                              -8.31636658
                                           -9.52365216]
   -8.60963749
                -7.61789831
                             -8.64066182 -9.51511561]
                -7.44747372 -6.59544358
   -7.67871466
                                           -8.44065716]
   -7.22313026
                 -6.5863032
                               -5.82992635
                                            -7.20791629]
   -6.34746906
                -6.02084794
                              -4.94934655
                                           -6.99064998]
   -4.7019478
                 -3.99264969
                              -5.74428784 -5.14188031]
   -4.03967381
                -4.15750827
                              -2.94144683
                                           -5.15181658]
  -13.08089144 -11.55076358 -11.54888054 -12.31868954]
  -12.33562005 -10.77936506 -10.76416381 -12.32199885]
  -11.60855737
                -9.98211475
                              -9.96343246 -11.55278601]
  -10.83156993
                 -9.14635966
                              -9.1919895
                                           -10.77429067]
   -9.96684149
                -8.31452341
                              -8.31261189
                                           -9.96383124]
   -9.21996275
                 -7.4943311
                              -7.46184887
                                          -9.15699451]
   -8.52431506
                 -6.63772183
                              -6.59372334
                                            -8.31077918]
   -7.58530005
                 -5.77057032
                              -5.70788096
                                           -7.50876088]
                              -4.80396016 -6.57715618]
   -6.76126252
                -5.11053449
                 -4.5067658
                              -3.881592
   -5.88664857
                                            -5.655096991
   -5.04260268
                 -2.9404
                              -2.86379982
                                           -4.78362613]
   -3.89799321
                -2.94323619
                                            -4.46736371]
                             -1.98
  -12.31790293 -10.76416381 -12.31790293 -11.54888054]
  -11.54888054
                 -9.96343246 -111.31790293 -11.54888054]
                -9.14635966 -111.31790293 -10.76416381]
  -10.76416381
   -9.96343246
                -8.31261189 -111.31790293 -9.96343246]
   -9.14635966
                -7.46184887 -111.31790293 -9.14635966]
   -8.31261189
                -6.59372334 -111.31790293
                                           -8.31261189]
   -7.46184887
                 -5.70788096 -111.31790293
                                            -7.46184887]
                -4.80396016 -111.31790293
   -6.59372334
                                            -6.59372334]
   -5.70788096
                -3.881592 -111.31790293
                                            -5.70788096]
                 -2.9404
   -4.80396016
                            -111.31790293
                                            -4.80396016]
   -4.09901429
                -1.98
                            -111.31790293
                                            -3.881592
   -2.9404
                 -1.98
                                            -2.9404
                              -1.
  -11.54888054 -111.31790293 -12.31790293 -12.31790293]
                  0.
                               0.
                                             0.
    0.
                  0.
                               0.
                                             0.
    0.
                  0.
                               0.
                                             0.
    0.
                  0.
                               0.
                                             0.
                              0.
    0.
                  0.
                                             0.
                                             0.
    0.
                  0.
                                             0.
    0.
                  0.
                               0.
    0.
                  0.
                               0.
                                             0.
    0.
                  0.
                               0.
                                             0.
    0.
                  0.
                               0.
                                             0.
 [
                                                       ]]
                  0.
                               0.
                                             0.
```

Рис. 6. Матрица Q1 алгоритма двойное Q-обучение

```
Q2
ГΓ
  -13.34038444 -12.83089646
                               -12.31860787
                                              -13.32084295]
[ -12.6906184
                 -12.46142441
                               -11.55220462
                                             -12.76605139]
  -12.44055462
                 -11.45431152
                               -10.77269721
                                             -12.01969573]
  -11.39266268
                 -11.15365865
                                -9.97101674
                                             -11.69921465]
  -10.57666081
                  -9.70181225
                                -9.14667129
                                              -10.77031629]
-9.36412331
                  -9.57066797
                                -8.32769203
                                              -10.29274439]
-8.99607497
                  -7.51088933
                                -8.04066208
                                               -9.1036498
   -7.93985503
                  -7.53275796
                                 -6.59634956
                                               -8.80887463]
   -7.08306959
                  -7.45594574
                                -5.73695892
                                               -7.57625673]
Γ
   -6.03705605
                  -6.18928127
                                -4.89559184
                                               -5.72958759]
   -5.65823062
                  -3.998285
                                -3.40717133
                                               -5.29643997]
    -4.08158656
                  -4.06744601
                                -2.94221612
                                               -5.8000926 ]
  -13.08214965
                 -11.5690338
                               -11.54888054
                                              -12.31946459]
Γ
  -12.32451307
                 -10.77185267
                               -10.76416381
                                              -12.32656292]
  -11.64865692
                  -9.97354336
                                -9.96343246
                                              -11.5616919 ]
  -10.79117431
                  -9.14635966
                                -9.16314747
                                              -10.76930655]
    -9.97810861
                  -8.31300602
                                -8.31261189
                                              -9.96373313]
   -9.18295762
                  -7.48351479
                                -7.46184887
                                               -9.15226038]
                                -6.59372334
   -8.62563399
                  -6.65262791
                                               -8.36696611]
-7.48975571
                  -5.82046727
                                -5.70788096
                                               -7.46427613]
-6.90615069
                  -4.89796948
                                -4.80396016
                                               -6.62507467]
                  -3.89492067
                                -3.881592
                                               -5.70587723]
-6.16506472
Γ
    -6.52447473
                  -2.9404
                                -3.15837429
                                               -4.87003761]
                  -2.94647923
-3.88676972
                                -1.98
                                               -3.87095241]
  -12.31790293
                 -10.76416381
                               -12.31790293
                                             -11.54888054]
  -11.54888054
                 -9.96343246 -111.31790293
                                              -11.54888054]
  -10.76416381
                  -9.14635966 -111.31790293
                                              -10.76416381]
-9.96343246
                  -8.31261189 -111.31790293
                                               -9.96343246]
                                               -9.14635966]
-9.14635966
                  -7.46184887 -111.31790293
   -8.31261189
                  -6.59372334 -111.31790293
                                               -8.31261189]
   -7.46184887
                  -5.70788096 -111.31790293
                                               -7.46184887]
   -6.59372334
                 -4.80396016 -111.31790293
                                               -6.59372334]
                                               -5.70788096]
-5.70788096
                  -3.881592
                              -111.31790293
-4.80396016
                  -2.9404
                              -111.31790293
                                               -4.80396016]
-3.881592
                  -1.98
                              -111.31790293
                                               -3.881592
   -2.9404
                  -1.98
                                -1.
                                               -2.9404
  -11.54888054 -111.31790293 -12.31790293 -12.31790293]
0.
                                 0.
                                                0.
    0.
                   0.
                                 0.
                                                0.
    0.
                   0.
                                 0.
                                                0.
    0.
                   0.
                                 0.
                                                0.
                   0.
                                 0.
                                                0.
    0.
                   0.
                                 0.
    0.
                                                0.
    0.
                   0.
                                                0.
    0.
                   0.
                                 0.
                                                0.
0.
                   0.
                                 0.
                                                0.
0.
                   0.
                                 0.
                                                0.
    0.
                                 0.
                                                0.
                                                          ]]
```

Рис.7. Матрица Q2 алгоритма двойное Q-обучение

Пример нахождения агента в конечном состоянии представлен на рис. 8.

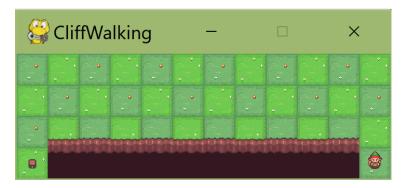


Рис. 8. Пример агента в конечном состоянии

Вывод: в ходе выполнения лабораторной работы мы ознакомились с базовыми методами обучения с подкреплением на основе временных различий.