Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика, искусственный интеллект и системы управления»

Кафедра «Системы обработки информации и управления»



**Отчет по рубежному контролю №2**

**по дисциплине «Методы машинного обучения»**

Методы обучения с подкреплением .

(тема работы)

ИСПОЛНИТЕЛЬ:

Пасатюк А.Д.

группа ИУ5-23М

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.А.

Москва, 2023

**Задание**

Для одного из алгоритмов временных различий, реализованных Вами в соответствующей лабораторная работе:

* SARSA
* Q-обучение
* Двойное Q-обучение

осуществите подбор гиперпараметров. Критерием оптимизации должна являться суммарная награда.

**Выполнение**

Осуществим подбор гиперпараметров для алгоритма двойное Q-обучение для среды Toy Text / CliffWalking-v0.

Код 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:  
  
 *# Если вероятность меньше eps  
 # 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(self):  
 *'''  
 Реализация алгоритма обучения  
 '''* **pass***# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Двойное Q-обучение \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****class** DoubleQLearning\_Agent(BasicAgent):  
 *'''  
 Реализация алгоритма Double Q-Learning  
 '''  
 # Наименование алгоритма* ALGO\_NAME = **'Двойное Q-обучение'  
  
 def** \_\_init\_\_(self, env, eps=0.4, lr=0.01, gamma=0.98, num\_episodes=10000):  
 *# 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 eps* 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):  
 *'''  
 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  
 # В 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:  
 *# 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()

Начальные значения параметров: eps=0.4, lr=0.01, gamma=0.98, num\_episodes=20000

Результат работы программы для алгоритма двойное Q-обучение:

Суммарная награда: -1 364 805

Вывод Q-матриц для алгоритма Двойное Q-обучение

Q1

[[ -10.09289085 -10.05236375 -10.06661421 -10.05735569]

[ -9.71581234 -9.73462015 -9.75893961 -9.76248178]

[ -9.20171654 -9.15979638 -9.12146606 -9.11167113]

[ -8.63566668 -8.63519216 -8.66457317 -8.63401819]

[ -7.98680156 -7.94002267 -7.84642037 -7.84955342]

[ -7.367545 -7.3177254 -7.33854801 -7.33750851]

[ -6.64671873 -6.5917705 -6.57839412 -6.46176356]

[ -5.87492013 -5.81049733 -5.88691874 -5.81181292]

[ -5.09154321 -5.11917163 -5.1355987 -5.21879047]

[ -4.40171783 -4.35688049 -4.24478732 -4.29833392]

[ -3.52301773 -3.60464091 -3.56958825 -3.63954656]

[ -2.88511185 -2.91032281 -2.86508486 -2.87708437]

[ -10.48024056 -10.49745569 -10.51655902 -10.44920163]

[ -9.96535803 -9.81722866 -9.93429739 -10.00811644]

[ -9.36394517 -9.37690307 -9.3595449 -9.63528841]

[ -8.59449177 -8.62334513 -8.66033629 -8.59899088]

[ -8.01929493 -7.98289361 -8.07404167 -7.99682761]

[ -7.26646017 -7.19817033 -7.25095966 -7.20935484]

[ -6.49400171 -6.51214251 -6.46780122 -6.54922636]

[ -5.53491379 -5.64586131 -5.62541019 -5.52589125]

[ -4.78415057 -4.69002989 -4.76720792 -4.97131056]

[ -4.03705047 -3.84944952 -3.83877637 -4.24189274]

[ -3.13561396 -2.91389716 -2.92618607 -3.15328547]

[ -2.656748 -2.22634054 -1.97981038 -2.59211793]

[ -11.00557409 -10.76416381 -11.85701092 -11.29618101]

[ -10.49072883 -9.96343246 -110.55624232 -10.98223467]

[ -9.71875505 -9.14635966 -109.98786692 -10.18519696]

[ -9.07175399 -8.31261189 -108.93028503 -9.43290233]

[ -8.22365029 -7.46184887 -107.72317206 -8.6161662 ]

[ -7.60964924 -6.59372334 -106.30573095 -7.79251125]

[ -6.74440505 -5.70788096 -106.13930259 -6.98440691]

[ -5.94309344 -4.80396016 -104.08830981 -6.12166608]

[ -5.17961435 -3.881592 -102.96398032 -5.32232631]

[ -4.34524825 -2.9404 -104.00078648 -4.4418403 ]

[ -3.41063132 -1.98 -102.83486016 -3.53742476]

[ -2.75113889 -1.86781992 -1. -2.73993195]

[ -11.54888054 -110.94993464 -12.09385316 -12.10577493]

[ 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. 0. 0. 0. ]]

Q2

[[ -10.08725201 -10.12520674 -10.1195806 -10.11902158]

[ -9.71225524 -9.68958033 -9.66908317 -9.67130278]

[ -9.18152206 -9.21987742 -9.25975732 -9.2689325 ]

[ -8.59901649 -8.59776443 -8.57436691 -8.60525873]

[ -7.92750984 -7.97279723 -8.07464665 -8.06597641]

[ -7.25584046 -7.30401695 -7.28732536 -7.2989119 ]

[ -6.51751331 -6.57148497 -6.59175349 -6.7033622 ]

[ -5.84858482 -5.9082715 -5.83290626 -5.9153179 ]

[ -5.12142191 -5.09302443 -5.08228751 -5.00239973]

[ -4.28001894 -4.31715554 -4.42759652 -4.3873417 ]

[ -3.64950912 -3.56744837 -3.60515033 -3.53928069]

[ -2.8666946 -2.83678597 -2.88052962 -2.91558382]

[ -10.43087191 -10.41097794 -10.40213938 -10.46437916]

[ -9.97858795 -10.12237541 -10.01065767 -9.94380216]

[ -9.28857852 -9.27584111 -9.29802879 -9.02624789]

[ -8.78931355 -8.76093983 -8.72711492 -8.79930542]

[ -7.94182926 -7.97707646 -7.88998857 -7.96616803]

[ -7.20800997 -7.26848581 -7.21814166 -7.26869932]

[ -6.43181951 -6.40954264 -6.4540916 -6.38100473]

[ -5.7043716 -5.58265138 -5.60338121 -5.74290532]

[ -4.81478883 -4.84343715 -4.76589759 -4.8147614 ]

[ -3.76967598 -3.8300691 -3.84104215 -3.82656552]

[ -3.1775973 -2.93630598 -2.92410092 -3.34788014]

[ -2.31075539 -2.38671612 -1.97983451 -2.35762701]

[ -11.0430105 -10.76416381 -11.86581843 -11.31649696]

[ -10.42761154 -9.96343246 -110.10299523 -11.00761557]

[ -9.83474167 -9.14635966 -109.2817179 -10.24362237]

[ -8.95959349 -8.31261189 -108.58168375 -9.33229126]

[ -8.25563215 -7.46184887 -108.10547576 -8.59900418]

[ -7.58326826 -6.59372334 -106.74402458 -7.72472949]

[ -6.72891051 -5.70788096 -106.36275122 -6.94088559]

[ -5.85231886 -4.80396016 -102.38917141 -6.16923341]

[ -5.21306041 -3.881592 -103.56646553 -5.18198297]

[ -4.54895799 -2.9404 -101.52384438 -4.46791789]

[ -3.52607338 -1.98 -103.68426844 -3.58669366]

[ -2.7474357 -1.87122489 -1. -2.74728571]

[ -11.54888054 -111.00242428 -12.09113087 -12.09010116]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

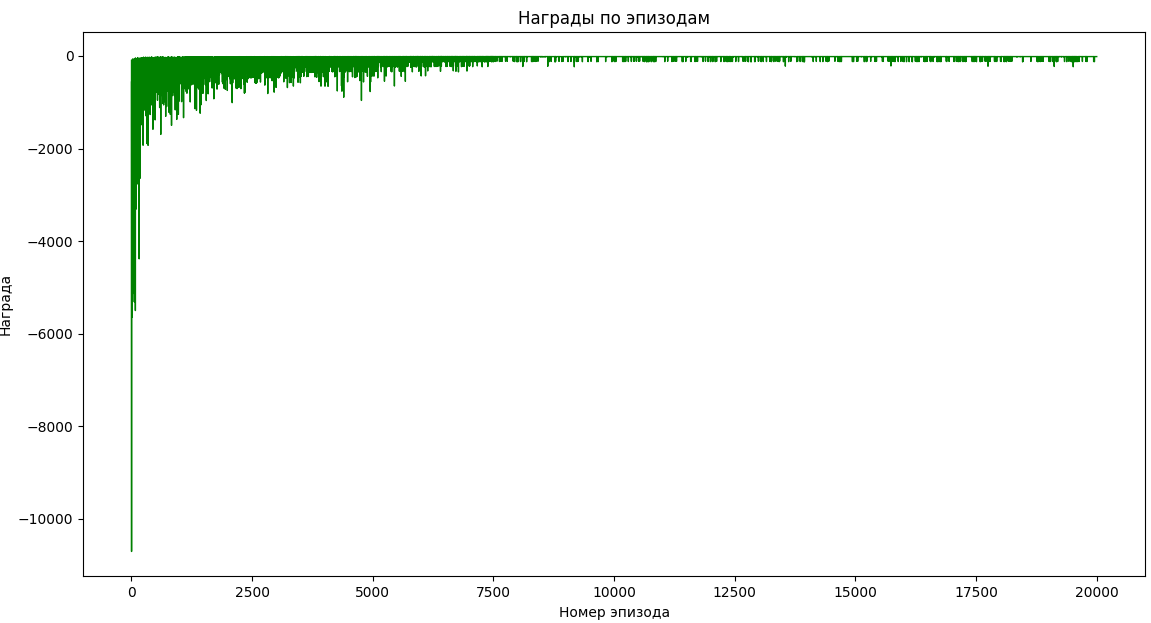
[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]]



Изменим следующие параметры: lr=0.5, num\_episodes=20000

Суммарная награда: -1 439 135

Вывод Q-матриц для алгоритма Двойное Q-обучение

Q1

[[ -11.9657858 -11.87042023 -11.86732539 -11.83872938]

[ -11.41332339 -11.39771521 -11.4175467 -11.74805093]

[ -10.68288564 -10.71451528 -10.67901651 -10.91284192]

[ -9.89879522 -9.86997088 -9.93874431 -10.33062944]

[ -9.51627825 -9.23844394 -9.13758822 -9.6380229 ]

[ -8.48133609 -8.5216577 -8.28248331 -8.823221 ]

[ -7.61497493 -7.87381806 -8.1313079 -7.66800254]

[ -7.23893188 -6.72829976 -6.41499778 -7.0540342 ]

[ -5.40910339 -6.24544729 -6.52080154 -5.86225381]

[ -5.34673589 -4.73160967 -4.20692848 -4.83008464]

[ -3.90981407 -3.925385 -4.52694793 -4.34713804]

[ -3.44020774 -3.27828083 -2.93761498 -3.54764939]

[ -12.46313118 -11.61054848 -11.54888054 -12.19742999]

[ -11.91560349 -10.75625267 -10.76416381 -12.22470474]

[ -11.23176454 -9.97286426 -9.96343246 -11.53577846]

[ -10.55222519 -9.17263827 -9.14635966 -10.6015 ]

[ -9.8356837 -8.31490101 -8.31261189 -9.80883516]

[ -9.44160891 -8.18802267 -7.46184887 -8.93985243]

[ -8.25889616 -6.52998622 -6.59372333 -8.07883231]

[ -7.77040861 -6.76957049 -5.70788099 -7.66870773]

[ -6.23967876 -4.45823857 -4.80395985 -6.14192221]

[ -5.92363029 -4.77522461 -3.88159231 -6.30472527]

[ -4.25804498 -2.93969628 -2.93976962 -3.72663587]

[ -3.55458229 -2.72681421 -1.98 -3.9549554 ]

[ -12.31790293 -10.76416381 -12.31790293 -11.54888054]

[ -11.57873915 -9.96343246 -111.31790293 -11.54888054]

[ -10.76416381 -9.14635966 -111.31790293 -10.76416381]

[ -9.96343246 -8.31261189 -111.31790293 -9.96343246]

[ -9.14635966 -7.46184887 -111.31790293 -9.14635966]

[ -8.31261189 -6.59372334 -111.3179029 -8.31261189]

[ -8.31503634 -5.70788096 -111.31790282 -7.46184887]

[ -6.59372283 -4.80396016 -111.31790276 -6.59372322]

[ -6.80705805 -3.881592 -111.31789958 -5.70788098]

[ -4.80395568 -2.9404 -111.31789651 -4.80395934]

[ -3.88234228 -1.98 -111.31788262 -3.88159162]

[ -2.94039874 -1.97999827 -1. -2.94039945]

[ -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. 0. 0. 0. ]]

Q2

[[ -11.85627492 -11.94110353 -11.9363446 -11.99527241]

[ -11.36027433 -11.37766876 -11.36031981 -11.43012568]

[ -10.86007128 -10.67902143 -10.71419049 -10.98906902]

[ -10.1923441 -9.98868682 -9.91805686 -10.12927801]

[ -9.27187704 -9.12713996 -9.14280062 -9.41085218]

[ -8.65797006 -8.60423479 -8.82751155 -8.91765225]

[ -7.97369461 -7.55849936 -7.27363464 -8.16736263]

[ -6.60225054 -7.0606398 -7.38036484 -7.18632466]

[ -6.37431163 -5.54550214 -5.27335876 -5.96152807]

[ -4.58692606 -5.18077096 -5.72128062 -5.07707573]

[ -4.4184448 -3.87579005 -3.29689175 -3.75823006]

[ -3.27503977 -3.44707523 -2.97224921 -4.02028411]

[ -12.45459829 -11.54111113 -11.54888054 -12.31752148]

[ -11.9839398 -10.79408928 -10.76416381 -12.31187581]

[ -11.33397228 -9.97643058 -9.96343246 -11.35295927]

[ -10.58371301 -9.17226326 -9.14635966 -10.69708238]

[ -9.7791513 -8.92942703 -8.31261189 -9.75464004]

[ -8.99759174 -7.43786749 -7.46184887 -9.03609304]

[ -8.54176048 -7.46532353 -6.59372335 -8.75105883]

[ -7.25428594 -5.46389192 -5.70788095 -7.05279609]

[ -6.62521953 -5.92414795 -4.80396035 -7.09797239]

[ -5.19738991 -3.50014841 -3.88159183 -4.76958849]

[ -4.66393968 -2.94055435 -2.94053608 -5.60003678]

[ -3.52231315 -2.76888829 -1.98 -3.26747307]

[ -12.37932653 -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 -111.31790292 -9.96343246]

[ -9.14635966 -7.46184887 -111.31790293 -9.14635966]

[ -9.0213795 -6.59372334 -111.31790289 -8.31261189]

[ -7.46184875 -5.70788096 -111.31790291 -7.46184883]

[ -7.63384947 -4.80396016 -111.31789907 -6.59372339]

[ -5.7078732 -3.881592 -111.31789988 -5.70788031]

[ -5.68231557 -2.9404 -111.31789954 -4.80396006]

[ -3.88055672 -1.98 -111.31775528 -3.8815909 ]

[ -2.94045074 -1.97999764 -1. -2.94039934]

[ -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. ]

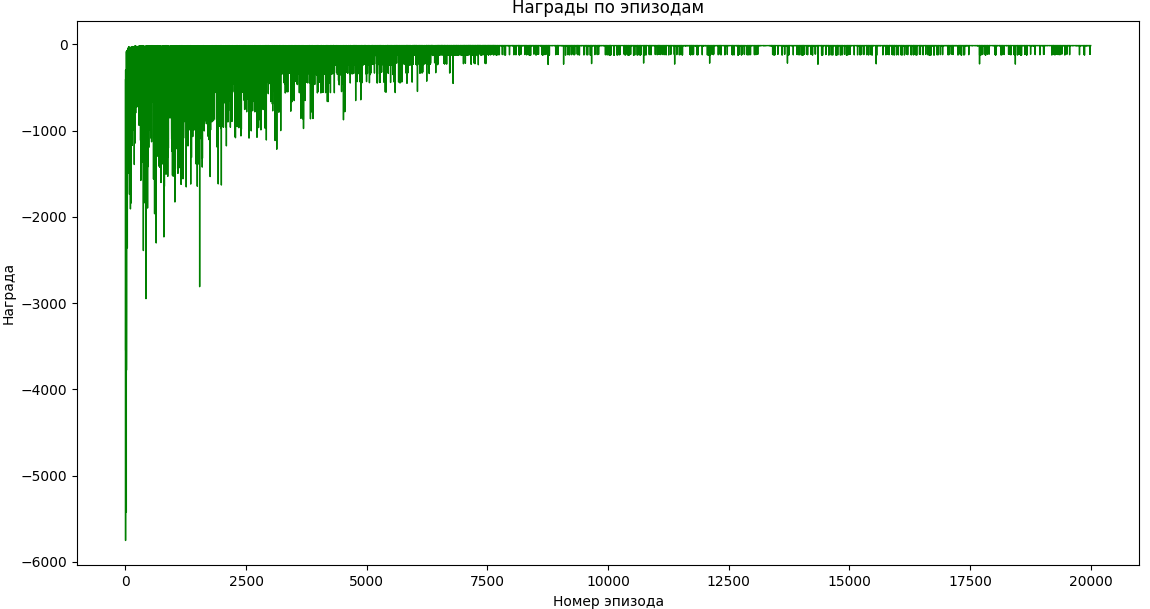
[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]

[ 0. 0. 0. 0. ]]



Изменим следующие параметры: lr=0.8, num\_episodes=10000

Суммарная награда: -859 934

Вывод Q-матриц для алгоритма Двойное Q-обучение

Q1

[[ -13.07239812 -12.31790978 -12.31790293 -13.07240228]

[ -12.31790293 -11.54888056 -11.54888054 -13.07154487]

[ -11.54888059 -10.76416381 -10.76416381 -12.31790294]

[ -10.76416381 -9.96346368 -9.96343246 -11.54888054]

[ -9.96343246 -9.14635966 -9.14635966 -10.76416381]

[ -9.14635966 -8.31261189 -8.31261189 -9.96343472]

[ -8.31261189 -7.46184887 -7.46184887 -9.14635966]

[ -7.46184887 -6.59372334 -6.59372334 -8.31261189]

[ -6.59372334 -5.70788099 -5.70788096 -7.46184887]

[ -5.70788096 -4.80396016 -4.80396016 -6.59372334]

[ -4.80396016 -3.881592 -3.881592 -5.70788096]

[ -3.881592 -3.881592 -2.9404 -4.80396016]

[ -13.07154487 -11.54888054 -11.54888054 -12.31790293]

[ -12.31790293 -10.76416381 -10.76416381 -12.31790293]

[ -11.54888054 -9.96343246 -9.96343246 -11.54888054]

[ -10.76416381 -9.14635966 -9.14635966 -10.76416381]

[ -9.96343246 -8.31261189 -8.31261189 -9.96343246]

[ -9.14635966 -7.46184887 -7.46184887 -9.14635966]

[ -8.31261189 -6.59372334 -6.59372334 -8.31261189]

[ -7.46184887 -5.70788096 -5.70788096 -7.46184887]

[ -6.59372334 -4.80396016 -4.80396016 -6.59372334]

[ -5.70788096 -3.881592 -3.881592 -5.70788096]

[ -4.80396016 -2.9404 -2.9404 -4.80396016]

[ -3.881592 -2.9404 -1.98 -3.881592 ]

[ -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 -7.46184887 -111.31790293 -9.14635966]

[ -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 -3.881592 -111.31790293 -5.70788096]

[ -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. 0. 0. 0. ]]

Q2

[[ -13.07154492 -12.31790433 -12.31790293 -13.07154492]

[ -12.31790294 -11.54888054 -11.54888054 -13.07154488]

[ -11.54888054 -10.76416381 -10.76416381 -12.31790293]

[ -10.76416381 -9.96343873 -9.96343246 -11.5488808 ]

[ -9.96343246 -9.14635966 -9.14635966 -10.76416381]

[ -9.14635966 -8.3126119 -8.31261189 -9.96343246]

[ -8.3126119 -7.46184887 -7.46184888 -9.14636125]

[ -7.46184887 -6.59372334 -6.59372334 -8.31261189]

[ -6.59372334 -5.70788125 -5.70788096 -7.46184887]

[ -5.70788099 -4.80396016 -4.80396016 -6.59372337]

[ -4.80396016 -3.881592 -3.881592 -5.70788096]

[ -3.881592 -3.881592 -2.9404 -4.80396016]

[ -13.07154487 -11.54888054 -11.54888054 -12.31790293]

[ -12.31790293 -10.76416381 -10.76416381 -12.31790293]

[ -11.54888054 -9.96343246 -9.96343246 -11.54888054]

[ -10.76416381 -9.14635966 -9.14635966 -10.76416381]

[ -9.96343246 -8.31261189 -8.31261189 -9.96343246]

[ -9.14635966 -7.46184887 -7.46184887 -9.14635966]

[ -8.31261189 -6.59372334 -6.59372334 -8.31261189]

[ -7.46184887 -5.70788096 -5.70788096 -7.46184887]

[ -6.59372334 -4.80396016 -4.80396016 -6.59372334]

[ -5.70788096 -3.881592 -3.881592 -5.70788096]

[ -4.80396016 -2.9404 -2.9404 -4.80396016]

[ -3.881592 -2.9404 -1.98 -3.881592 ]

[ -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 -7.46184887 -111.31790293 -9.14635966]

[ -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 -3.881592 -111.31790293 -5.70788096]

[ -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. 0. 0. 0. ]]

