

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, искусственный интеллект и с	истемь	і управления
КАФЕДРА _	Системы обработки информации и управления		
	Рубежный контроль №2		
	«Методы обучения с подкреплени	ием»	
	ИСПОЛНИТЕЛЬ:		<u>Козинов О.И.</u> _{ФИО}
	группа ИУ5-25М		подпись
		""_	_2023 г
	ПРЕПОДАВАТЕЛЬ:		<u>Гапанюк Ю.Е</u> _{ФИО}
		" "	подпись 2023 г

1. Задание

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

- SARSA
- О-обучение
- Двойное Q-обучение

осуществите подбор гиперпараметров. Критерием оптимизации должна являться суммарная награда.

2. Код программы

```
import numpy as np
import matplotlib.pyplot as plt
import gym
from tqdm import tqdm
class BasicAgent:
    1.1.1
    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):
   1.1.1
    Возвращает правильное начальное состояние
    if type(state) is tuple:
            # Если состояние вернулось в виде кортежа, то вернуть
 только номер состояния
       return state[0]
    else:
       return state
def greedy(self, state):
    1.1.1
    <<Жадное>> текущее действие
    Возвращает действие, соответствующее максимальному Q-значению
    для состояния state
    111
    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():
       1.1.1
       Реализация алгоритма обучения
        1.1.1
       pass
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
                   # 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)
                # 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)
def play_agent(agent):
    1.1.1
   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 q learning():
   env = gym.make("CliffWalking-v0")
    epsilons = [0.3, 0.4, 0.5]
    learning_rates = [0.05, 0.1, 0.2]
   gammas = [0.95, 0.98, 0.99]
   num episodes = 20000
   best reward = float('-inf')
   best hyperparams = None
   best agent = None
   for eps in epsilons:
        for lr in learning rates:
            for gamma in gammas:
                         agent = QLearning Agent(env, eps=eps, lr=lr,
gamma=gamma, num episodes=num episodes)
```

```
agent.learn()
                total reward = sum(agent.episodes reward)
                       print(f'Гиперпараметры: epsilon={eps}, learning
rate={lr}, gamma={gamma}, num episodes={num episodes}')
                print(f'Суммарная награда: {total reward}\n')
                if total reward > best reward:
                    best_reward = total_reward
                    best hyperparams = (eps, lr, gamma, num episodes)
                    best_agent = agent
        print(f'Лучшие гиперпараметры: epsilon={best_hyperparams[0]},
learning rate={best hyperparams[1]}, gamma={best hyperparams[2]}, num
episodes={best_hyperparams[3]}')
   print(f'Суммарная награда: {best reward}\n')
   best agent.print q()
   best_agent.draw_episodes_reward()
   play agent(best agent)
def main():
   run q learning()
if___name__== '__main__':
   main()
```

3. Результаты подбора гиперпараметров

3.1. Гиперпараметры: epsilon = 0.3, learning rate = 0.05, gamma = 0.95, num episodes = 20000

Суммарная награда: -734106

ооо»| шиерпараметры: epsilon=0.3, learning rate=0.05, gamma=0.95, num episodes=20000 уммарная награда: –734106 | 20000/20000 [00:11<00:00, 1707.11it/

3.2. Гиперпараметры: epsilon = 0.3, learning rate = 0.05, gamma = 0.98, num episodes = 20000

Суммарная награда: -719783

лого» Гиперпараметры: epsilon=0.3, learning rate=0.05, gamma=0.98, num episodes=20000 | 20000/20000 [00:11<00:00, 1807.87it/s]

3.3. Гиперпараметры: epsilon = 0.3, learning rate = 0.05, gamma = 0.99, num episodes = 20000

Суммарная награда: -724730

s| ерпараметры: epsilon=0.3, learning rate=0.05, gamma=0.99, num episodes=20000 20000/20000 [00:42<00:00, 469.41it/s]

3.4. Гиперпараметры: epsilon = 0.3, learning rate = 0.1, gamma = 0.95, num episodes = 20000

Суммарная награда: -738206

0%| mepmapameтры: epsilon=0.3, learning rate=0.1, gamma=0.95, num episodes=20000 | 20000/20000 [2:00:12<00:00, 2.77it/s]

3.5. Гиперпараметры: epsilon = 0.3, learning rate = 0.1, gamma = 0.98, num episodes = 20000

Суммарная награда: -723993

параметры: epsilon=0.3, learning rate=0.1, gamma=0.98, num episodes=20000

20000/20000 [59:15<00:00, 5.63it/s]

3.6. Гиперпараметры: epsilon = 0.3, learning rate = 0.1, gamma = 0.99, num episodes = 20000

Суммарная награда: -747135

00%| unepnapamerpы: epsilon=0.3, learning rate=0.1, gamma=0.99, num episodes=20000

20000/20000 [2:00:14<00:00, 2.77it/s]

3.7. Гиперпараметры: epsilon = 0.3, learning rate = 0.2, gamma = 0.95, num episodes = 20000

Суммарная награда: -690261

100%| Гиперпараметры: epsilon=0.3, learning rate=0.2, gamma=0.95, num episodes=20000 Соммариав награда: —600761 | 20000/20000 [1:47:10<00:00, 3.11it/s]

3.8. Гиперпараметры: epsilon = 0.3, learning rate = 0.2, gamma = 0.98, num episodes = 20000

Суммарная награда: -695174

100%| Гиперпараметры: epsilon=0.3, learning rate=0.2, gamma=0.98, num episodes=20000 Суммарная награда: —695174 | 20000/20000 [00:11<00:00, 1695.33it/s]

3.9. Гиперпараметры: epsilon = 0.3, learning rate = 0.2, gamma = 0.99, num episodes = 20000

Суммарная награда: -697318

00%| иперпараметры: epsilon=0.3, learning rate=0.2, gamma=0.99, num episodes=2000 | 20000/20000 [00:11<00:00, 1747.98it/s]

3.10. Гиперпараметры: epsilon = 0.4, learning rate = 0.05, gamma = 0.95, num episodes = 20000

Суммарная награда: -1187325

1008| при на paragraphi 1187325 (1975) (197

| 20000/20000 [00:13<00:00, 1495.20it/s]

3.11. Гиперпараметры: epsilon = 0.4, learning rate = 0.05, gamma = 0.98, num episodes = 20000

Суммарная награда: -1187912

100%| Гиперпараметры: epsilon=0.4, learning rate=0.05, gamma=0.98, num episodes=20000 | 20000/20000 [00:13<00:00, 1502.06it/s]

3.12. Гиперпараметры: epsilon = 0.4, learning rate = 0.05, gamma = 0.99, num episodes = 20000

Суммарная награда: -1187358

иперпараметры: epsilon=0.4, learning rate=0.05, gamma=0.99, num episodes=20000

20000/20000 [00:13<00:00, 1506.70it/s]

3.13. Гиперпараметры: epsilon = 0.4, learning rate = 0.1, gamma = 0.95, num episodes = 20000

Суммарная награда: -1067977

мперпараметры: epsilon=0.4, learning rate=0.1, gamma=0.95, num episodes=20000

| 20000/20000 [00:12<00:00, 1573.96it/s]

3.14. Гиперпараметры: epsilon = 0.4, learning rate = 0.1, gamma = 0.98, num episodes = 20000

Суммарная награда: -1073649

Гиперпараметры: epsilon=0.4, learning rate=0.1, gamma=0.98, num episodes=20000

| 20000/20000 [00:13<00:00, 1537.58it/s]

3.15. Гиперпараметры: epsilon = 0.4, learning rate = 0.1, gamma = 0.99, num episodes = 20000

Суммарная награда: -1085515

unepnapaметры: epsilon=0.4, learning rate=0.1, gamma=0.99, num episodes=20000

| 20000/20000 [00:13<00:00, 1532.51it/s]

3.16. Гиперпараметры: epsilon = 0.4, learning rate = 0.2, gamma = 0.95, num episodes = 20000

Суммарная награда: -1085515

раметры: epsilon=0.4, learning rate=0.2, gamma=0.95, num episodes=20000

| 20000/20000 [00:12<00:00, 1604.99it/s]

3.17. Гиперпараметры: epsilon = 0.4, learning rate = 0.2, gamma = 0.98, num episodes = 20000

Суммарная награда: -1001796

| 20000/20000 [00:12<00:00, 1600.89it/s]

3.18. Гиперпараметры: epsilon = 0.4, learning rate = 0.2, gamma = 0.99, num episodes = 20000

Суммарная награда: -992852

| 20000/20000 [00:12<00:00, 1612.75it/s]

3.19. Гиперпараметры: epsilon = 0.5, learning rate = 0.05, gamma = 0.95, num episodes = 20000

Суммарная награда: -1753563

100%| Гиперпараметры: epsilon=0.5, learning rate=0.05, gamma=0.95, num episodes=20000 | 20000/20000 [00:15<00:00, 1275.97it/s]

3.20. Гиперпараметры: epsilon = 0.5, learning rate = 0.05, gamma = 0.98, num episodes = 20000

Суммарная награда: -1753563

100%| Гиперпараметры: epsilon=0.5, learning rate=0.05, gamma=0.98, num episodes=20000 Суммарная награда: -1703565 | 20000/20000 [00:15<00:00, 1281.49it/s]

3.21. Гиперпараметры: epsilon = 0.5, learning rate = 0.05, gamma = 0.99, num episodes = 20000

Суммарная награда: -1767028

00%|| иперпараметры: epsilon=0.5, learning rate=0.05, gamma=0.99, num episodes=20000 | 20000/20000 [00:15<00:00, 1265.42it/s]

3.22. Гиперпараметры: epsilon = 0.5, learning rate = 0.1, gamma = 0.95, num episodes = 20000

Суммарная награда: -1553120

อยจะ บทepnapamerpы: epsilon=0.5, learning rate=0.1, gamma=0.95, num episodes=200 เคลงลายลง และกลกละ -1553170 | 20000/20000 [00:14<00:00, 1342.04it/s]

3.23. Гиперпараметры: epsilon = 0.5, learning rate = 0.1, gamma = 0.98, num episodes = 20000

Суммарная награда: -1593516

.00%| иперпараметры: epsilon=0.5, learning rate=0.1, gamma=0.98, num episodes=20000 | 20000/20000 [00:14<00:00, 1333.72it/s

3.24. Гиперпараметры: epsilon = 0.5, learning rate = 0.1, gamma = 0.99, num episodes = 20000

Суммарная награда: -1573441

|00%| | Unepnapametpu: epsilon=0.5, learning rate=0.1, gamma=0.99, num episodes=20000 | 20000/20000 [00:15<00:00, 1330.83it/s]

3.25. Гиперпараметры: epsilon = 0.5, learning rate = 0.2, gamma = 0.95, num episodes = 20000

Суммарная награда: -1419773

100%| Гиперпараметры: epsilon=0.5, learning rate=0.2, gamma=0.95, num episodes=20000 | 20000/20000 [00:14<00:00, 1380.07it/s]

3.26. Гиперпараметры: epsilon = 0.5, learning rate = 0.2, gamma = 0.98, num episodes = 20000

Суммарная награда: -1418196

100%| Гиперпараметры: epsilon=0.5, learning rate=0.2, gamma=0.98, num episodes=2000| Суммарная награда: -1418196 | 20000/20000 [00:14<00:00, 1384.42it/s]

3.27. Гиперпараметры: epsilon = 0.5, learning rate = 0.2, gamma = 0.99, num episodes = 20000

Суммарная награда: -1418465

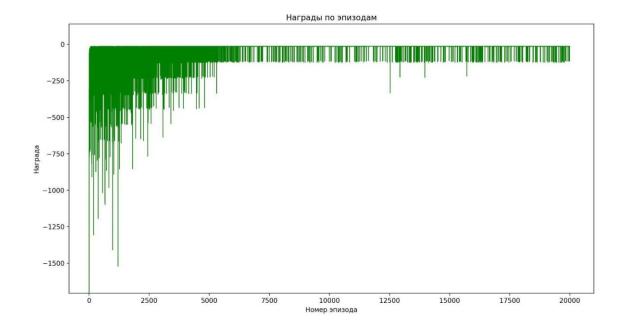
90%| иперпараметры: epsilon=0.5, learning rate=0.2, gamma=0.99, num episodes=20000 уммарная награда: —1418465 | 20000/20000 [00:14<00:00, 1396.09it/s]

4. Вывод

Лучшие гиперпараметры: epsilon = 0.3, learning rate = 0.2, gamma = 0.95, num episodes = 20000

Суммарная награда: -690261

```
Лучшие гиперпараметры: epsilon=0.3, learning rate=0.2, gamma=0.95, num episodes=20000
Суммарная награда: -690261
Вывод Q-матрицы для алгоритма
[[ -10.15122819 -10.16080799
                                 Q-обучение
-10.15528908
                                                 -10.2115043
    -9.78189542
                                                  -9.92690154]
                   -9.72532187
                                   -9.72634666
    -9.52241221
                   -9.19199606
                                   -9.19209058
                                                  -9.70172541]
                   -8.62394292
    -8.78334646
                                   -8.62393363
                                                  -9.44289466
                                   -8.0252482
    -8.55195742
                   -8.02524897
                                                  -9.04865249]
     -7.96616829
                   -7.3950095
                                   -7.39500978
                                                  -8.38539217
                   -6.73159123
     -7.28558023
                                   -6.7315912
                                                  -8.00852816]
                   -6.03325406
                                   -6.03325406
                                                  -7.29529639]
    -6.57247631
     -5.99462608
                   -5.29816219
                                   -5.29816219
                                                  -6.69052357
    -5.24931545
                   -4.52438125
                                   -4.52438125
                                                  -5.9616897
                   -3.709875
-3.51325354
                                   -3.709875
    -4.52246015
                                                  -5.10578582]
                                   -2.8525
                                                  -4.40984427
    -3.64362124
   -10.55568587
                                                 -10.24602962]
                                   -9.73315833
                   -9.73315833
                   -9.19279825
                                   -9.19279825
   -10.22844006
                                                 -10.24626327
     -9.7310139
                   -8.62399815
                                   -8.62399815
                                                  -9.73314855
    -9.19262235
                   -8.02526122
                                   -8.02526122
                                                  -9.19279805]
                   -7.39501181
    -8.62385584
                                   -7.39501181
                                                  -8.62399799]
                                                  -8.02526113]
    -8.02524409
                   -6.73159137
                                   -6.73159137
     -7.39501061
                   -6.03325408
                                   -6.03325408
                                                  -7.39501121]
    -6.7315913
                   -5.29816219
                                   -5.29816219
                                                  -6.73159137]
    -6.03325406
                   -4.52438125
                                   -4.52438125
                                                  -6.033254081
    -5.29816215
                   -3.709875
                                   -3.709875
                                                  -5.29816211]
                   -2.8525
                                   -2.8525
                                                  -4.5243811
    -4.52438118
     -3.709875
                   -2.8525
                                   -1.95
                                                  -3.70987483]
   -10.24650042
                   -9.19279825
                                -10.24650042
                                                  -9.73315833]
                   -8.62399815 -109.24650042
    -9.73315833
                                                  -9.73315833]
                   -8.02526122 -109.24650042
    -9.19279825
                                                  -9.19279825]
                   -7.39501181 -109.24650042
    -8.62399815
                                                  -8.62399815]
                                -109.24650042
                   -6.73159137
    -8.02526122
                                                  -8.02526122]
                   -6.03325408 -109.24650042
    -7.39501181
                                                  -7.39501181]
                   -5.29816219 -109.24650042
-4.52438125 -109.24650042
                                                  -6.73159137]
    -6.73159137
    -6.03325408
                                                  -6.03325408]
     -5.29816219
                   -3.709875
                                -109.24650042
                                                  -5.29816219
                                 -109.24650042
                                                  -4.52438125]
    -4.52438125
                   -2.8525
    -3.709875
                   -1.95
                                 -109.24650042
                                                  -3.709875
    -2.8525
                   -1.95
                                                  -2.8525
    -9.73315833 -109.24650042
                                 -10.24650042
                                                 -10.24650042]
                                   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.
```



Для исходных гиперпараметров epsilon = 0.4, learning rate = 0.1, gamma = 0.98, num episodes = 20000 суммарная награда равнялась -1073649, следовательно подбор гиперпараметров помог улучшить результаты обучения и суммарную награду. Подбор гиперпараметров является важным шагом в обучении моделей и может значительно улучшить качество работы модели.