



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 системы управления

КАФЕДРА Системы обработки информации и управления

**Рубежный контроль №2**  
**«Методы обучения с подкреплением»**

ИСПОЛНИТЕЛЬ:

группа ИУ5-25М

Козин О.И.

ФИО

\_\_\_\_\_

подпись

"\_\_" \_\_\_\_\_ 2023 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

ФИО

\_\_\_\_\_

подпись

"\_\_" \_\_\_\_\_ 2023 г.

Москва - 2023

## 1. 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.

## 2. Код программы

```
import numpy as np
import matplotlib.pyplot as plt
import gym
from tqdm import tqdm

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 в 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():
    '''
    Реализация алгоритма обучения
    '''
    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 eps
    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):
    '''
    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)

```



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

Суммарная награда: -724730

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:42<00:00, 469.41it/s]
Гиперпараметры: epsilon=0.3, learning rate=0.05, gamma=0.99, num episodes=20000
Суммарная награда: -724730
```

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

Суммарная награда: -738206

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [2:00:12<00:00, 2.77it/s]
Гиперпараметры: epsilon=0.3, learning rate=0.1, gamma=0.95, num episodes=20000
Суммарная награда: -738206
```

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

Суммарная награда: -723993

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [59:15<00:00, 5.63it/s]
Гиперпараметры: epsilon=0.3, learning rate=0.1, gamma=0.98, num episodes=20000
Суммарная награда: -723993
```

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

Суммарная награда: -747135

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [2:00:14<00:00, 2.77it/s]
Гиперпараметры: epsilon=0.3, learning rate=0.1, gamma=0.99, num episodes=20000
Суммарная награда: -747135
```

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

Суммарная награда: -690261

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [1:47:10<00:00, 3.11it/s]
Гиперпараметры: epsilon=0.3, learning rate=0.2, gamma=0.95, num episodes=20000
Суммарная награда: -690261
```

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

Суммарная награда: -695174

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



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

Суммарная награда: -697318

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:11<00:00, 1747.98it/s]
Гиперпараметры: epsilon=0.3, learning rate=0.2, gamma=0.99, num episodes=20000
Суммарная награда: -697318
```

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

Суммарная награда: -1187325

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:13<00:00, 1495.20it/s]
Гиперпараметры: epsilon=0.4, learning rate=0.05, gamma=0.95, num episodes=20000
Суммарная награда: -1187325
```

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

Суммарная награда: -1187912

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:13<00:00, 1502.06it/s]
Гиперпараметры: epsilon=0.4, learning rate=0.05, gamma=0.98, num episodes=20000
Суммарная награда: -1187912
```

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

Суммарная награда: -1187358

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:13<00:00, 1506.70it/s]
Гиперпараметры: epsilon=0.4, learning rate=0.05, gamma=0.99, num episodes=20000
Суммарная награда: -1187358
```

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

Суммарная награда: -1067977

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:12<00:00, 1573.96it/s]
Гиперпараметры: epsilon=0.4, learning rate=0.1, gamma=0.95, num episodes=20000
Суммарная награда: -1067977
```

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

Суммарная награда: -1073649

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:13<00:00, 1537.58it/s]
Гиперпараметры: epsilon=0.4, learning rate=0.1, gamma=0.98, num episodes=20000
Суммарная награда: -1073649
```

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

Суммарная награда: -1085515

100% | 20000/20000 [00:13<00:00, 1532.51it/s]  
Гиперпараметры:  $\epsilon=0.4$ , learning rate=0.1,  $\gamma=0.99$ , num episodes=20000  
Суммарная награда: -1085515

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

Суммарная награда: -1085515

100% | 20000/20000 [00:12<00:00, 1604.99it/s]  
Гиперпараметры:  $\epsilon=0.4$ , learning rate=0.2,  $\gamma=0.95$ , num episodes=20000  
Суммарная награда: -983916

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

Суммарная награда: -1001796

100% | 20000/20000 [00:12<00:00, 1600.89it/s]  
Гиперпараметры:  $\epsilon=0.4$ , learning rate=0.2,  $\gamma=0.98$ , num episodes=20000  
Суммарная награда: -1001796

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

Суммарная награда: -992852

100% | 20000/20000 [00:12<00:00, 1612.75it/s]  
Гиперпараметры:  $\epsilon=0.4$ , learning rate=0.2,  $\gamma=0.99$ , num episodes=20000  
Суммарная награда: -992852

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

Суммарная награда: -1753563

100% | 20000/20000 [00:15<00:00, 1275.97it/s]  
Гиперпараметры:  $\epsilon=0.5$ , learning rate=0.05,  $\gamma=0.95$ , num episodes=20000  
Суммарная награда: -1753563

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

Суммарная награда: -1753563

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

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

Суммарная награда: -1767028

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:15<00:00, 1265.42it/s]
Гиперпараметры: epsilon=0.5, learning rate=0.05, gamma=0.99, num episodes=20000
Суммарная награда: -1767028
```

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

Суммарная награда: -1553120

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:14<00:00, 1342.04it/s]
Гиперпараметры: epsilon=0.5, learning rate=0.1, gamma=0.95, num episodes=20000
Суммарная награда: -1553120
```

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

Суммарная награда: -1593516

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:14<00:00, 1333.72it/s]
Гиперпараметры: epsilon=0.5, learning rate=0.1, gamma=0.98, num episodes=20000
Суммарная награда: -1593516
```

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

Суммарная награда: -1573441

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:15<00:00, 1330.83it/s]
Гиперпараметры: epsilon=0.5, learning rate=0.1, gamma=0.99, num episodes=20000
Суммарная награда: -1573441
```

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

Суммарная награда: -1419773

```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:14<00:00, 1380.07it/s]
Гиперпараметры: epsilon=0.5, learning rate=0.2, gamma=0.95, num episodes=20000
Суммарная награда: -1419773
```

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

Суммарная награда: -1418196

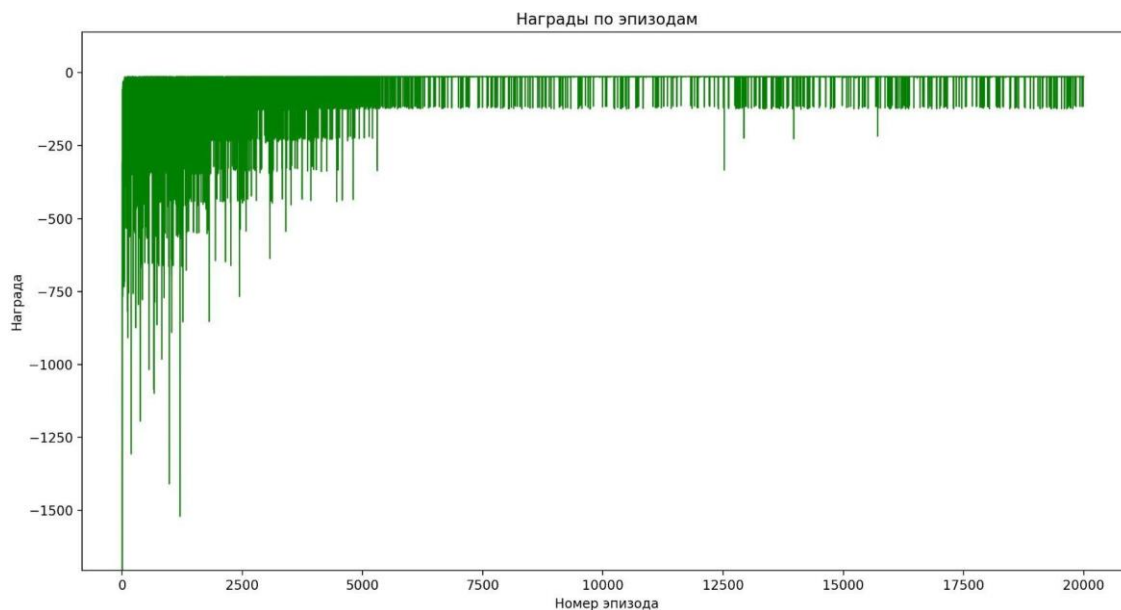
```
100%|████████████████████████████████████████████████████████████████████████████████| 20000/20000 [00:14<00:00, 1384.42it/s]
Гиперпараметры: epsilon=0.5, learning rate=0.2, gamma=0.98, num episodes=20000
Суммарная награда: -1418196
```

Суммарная награда: -1418465

## 4. Вывод

Суммарная награда: -690261

[illegible]



Для исходных гиперпараметров  $\epsilon = 0.4$ ,  $\text{learning rate} = 0.1$ ,  $\gamma = 0.98$ ,  $\text{num episodes} = 20000$  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.