Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

«Методы машинного обучения в автоматизированных системах обработки информации и управления»

Рубежный контроль №2

«Методы обучения с подкреплением»

ИСПОЛНИТЕЛЬ:

Демирев Н.К.
Группа ИУ5-21М

_____

"__"_____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. Листинг

2.1. BasicAgent.py

```
import numpy as np
import plotly.express as px
import pandas as pd
import os
import pygame


os.environ['SDL_VIDEODRIVER']='dummy'
pygame.display.set_mode((640,480))


class BasicAgent:

    ALGO_NAME = 'Base'

    def __init__(self, env, eps=0.1):
        self.env = env
        self.nA = env.action_space.n
        self.nS = env.observation_space.n
        self.Q = np.zeros((self.nS, self.nA))
        self.eps=eps
        self.episodes_reward = []
        self.all_reward = 0

    def print_q(self):
        self.all_reward = np.sum(self.Q)

    def get_state(self, state):

        if type(state) is tuple:
            return state[0]
        else:
```

```
            return state

    def greedy(self, state):
        return np.argmax(self.Q[state])

    def make_action(self, state):

        if np.random.uniform(0,1) < self.eps:
            return self.env.action_space.sample()
        else:
            return self.greedy(state)

    def draw_episodes_reward(self):
        y = self.episodes_reward

        df = pd.DataFrame(data={
            'Номер эпизода': list(range(1, len(y)+1)),
            'Награда': y
        })

        fig = px.line(df, x="Номер эпизода", y="Награда", title='Награды по
эпизодам', height=400, width=600)
        fig.show()

    def learn(self):
        pass
```

2.2. SARSA_Agent.py

```
import os
import pygame
from BasicAgent import BasicAgent


os.environ['SDL_VIDEODRIVER']='dummy'
pygame.display.set_mode((640,480))


class SARSA_Agent(BasicAgent):
    ALGO_NAME = 'SARSA'

    def __init__(self, env, eps=0.4, lr=0.1, gamma=0.98):
        super().__init__(env, eps)
        self.lr=lr
        self.gamma = gamma
        self.eps_decay=0.00005
        self.eps_threshold=0.01

    def learn(self, num_episodes=20000):
        self.episodes_reward = []
```

```
        for ep in list(range(num_episodes)):
            state = self.get_state(self.env.reset())
            done = False
            truncated = False
            tot_rew = 0

            if self.eps > self.eps_threshold:
                self.eps -= self.eps_decay

            action = self.make_action(state)

            while not (done or truncated):
                next_state, rew, done, truncated, _ = self.env.step(action)

                next_action = self.make_action(next_state)

                self.Q[state][action] = self.Q[state][action] + self.lr * \
                    (rew + self.gamma * self.Q[next_state][next_action] -
self.Q[state][action])

                state = next_state
                action = next_action
                tot_rew += rew
                if (done or truncated):
                    self.episodes_reward.append(tot_rew)
```

2.3. main.py

```
import gymnasium as gym
import os
import pygame
from tabulate import tabulate
import time
import numpy as np
from tqdm import tqdm
from SARSA_Agent import SARSA_Agent


os.environ['SDL_VIDEODRIVER']='dummy'
pygame.display.set_mode((640,480))


def run_sarsa():
    all_rewards = []
    parameters = []

    lr_list = np.linspace(0.0005, 0.005, num=5)
    gamma_list = np.linspace(0.9, 1, num=5)
    eps_list = np.linspace(0.05, 0.9, num=9)
```

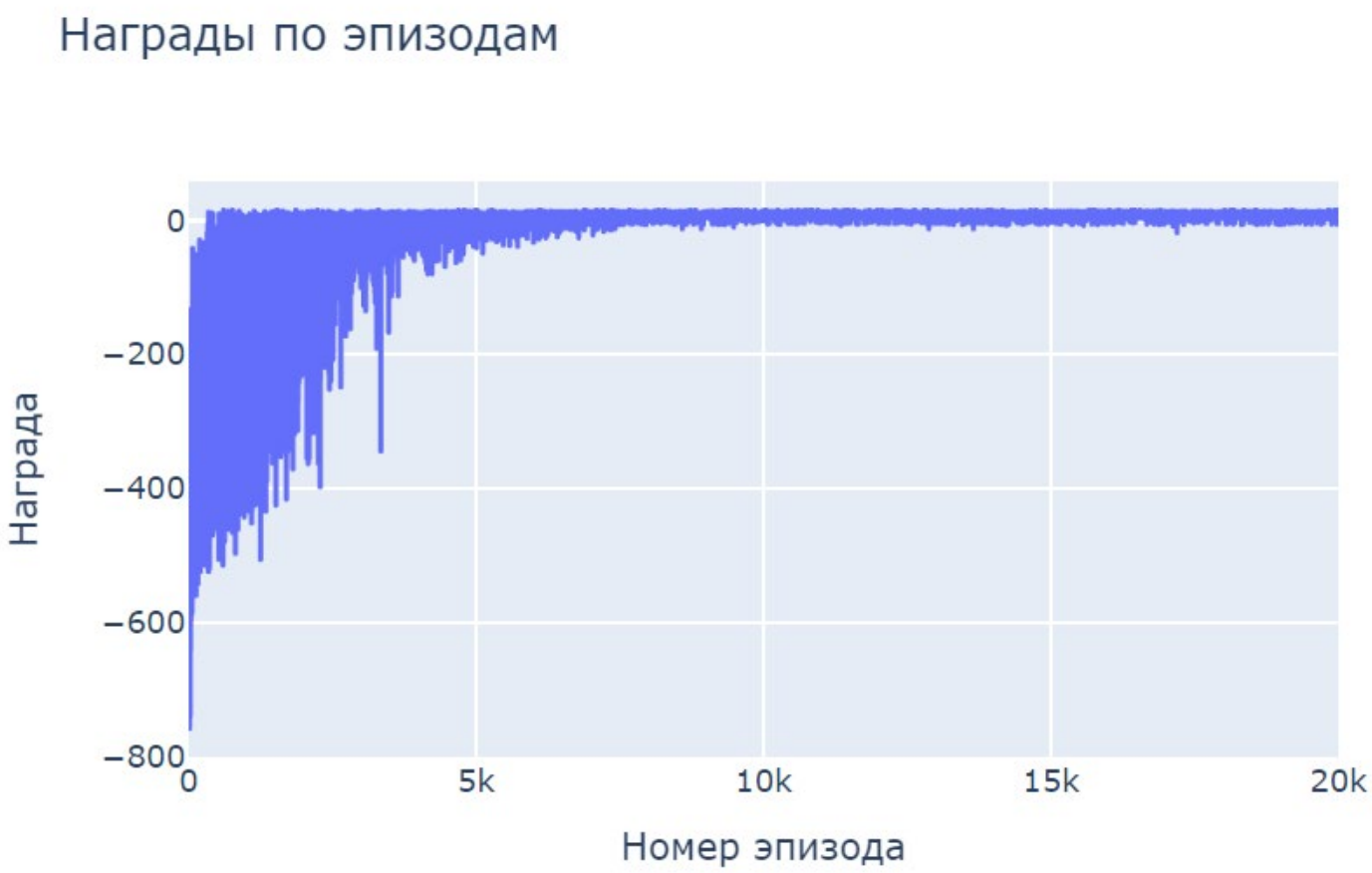
```
    env = gym.make('Taxi-v3')
    for lr in tqdm(lr_list, bar_format=' {l_bar}{bar:20}{r_bar}{bar:-10b}',
colour='CYAN'):
        for gamma in gamma_list:
            for ep in eps_list:
                agent = SARSA_Agent(env, lr=lr, gamma=gamma, eps=ep)
                agent.learn(100)
                agent.print_q()
                all_rewards.append(agent.all_reward)
                parameters.append([lr, gamma, ep])

    return all_rewards, parameters


def main():
    all_rewards, parameters = run_sarsa()

    print(tabulate(
        {
            'Максимальная награда:' : [np.max(all_rewards)],
            'Значения гиперпараметров' :
parameters[np.argmax(np.max(all_rewards))]
        },
        headers='keys',
        tablefmt='psql'))
    print(f"Закончено за {time.process_time():.3f}")


if __name__ == '__main__':
    main()
```

3. Подбор гиперпараметров

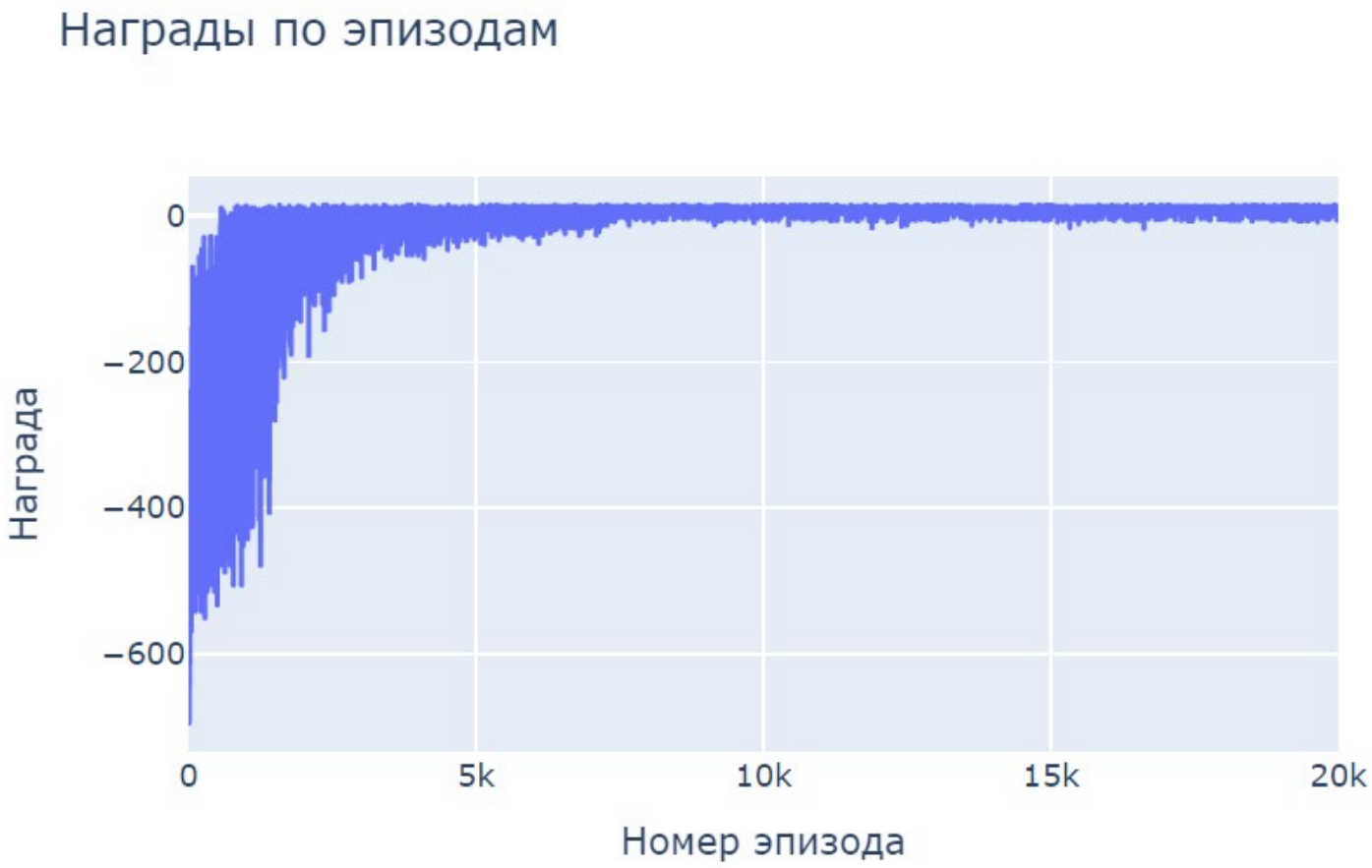
- lr от 0.0005 до 0.005 – 5 значений с равным шагом
- Gamma от 0.9 до 1 – 5 значений с равным шагом
- Eps от 0.05 до 0.9 – 9 значений с равным шагом

```
(venv) PS G:\repos\MMO\RK 2> python .\main.py
 100%|                    | 5/5 [00:59<00:00, 11.92s/it]
+-------------------------+-----------------------------+
|   Максимальная награда: |   Значения гиперпараметров  |
|-------------------------+-----------------------------|
|                -17.3628 |                      0.0005 |
|                         |                      0.9    |
|                         |                      0.05   |
+-------------------------+-----------------------------+
Закончено за 59.312
(venv) PS G:\repos\MMO\RK 2>
```

3.1. Q-обучение



3.2. SARSA


Награды по эпизодам
Награда
0
−200
−400
−600
0
5k
10k
15k
20k
Номер эпизода

3.3. Двойное Q-обучение