

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



**Лабораторная работа № 6**  
**по дисциплине «Методы машинного обучения»**  
**Обучение на основе глубоких Q-сетей**

ИСПОЛНИТЕЛЬ:

студент ИУ5-25М  
Мацнев А.А.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

"\_\_" \_\_\_\_\_ 2023 г.

---

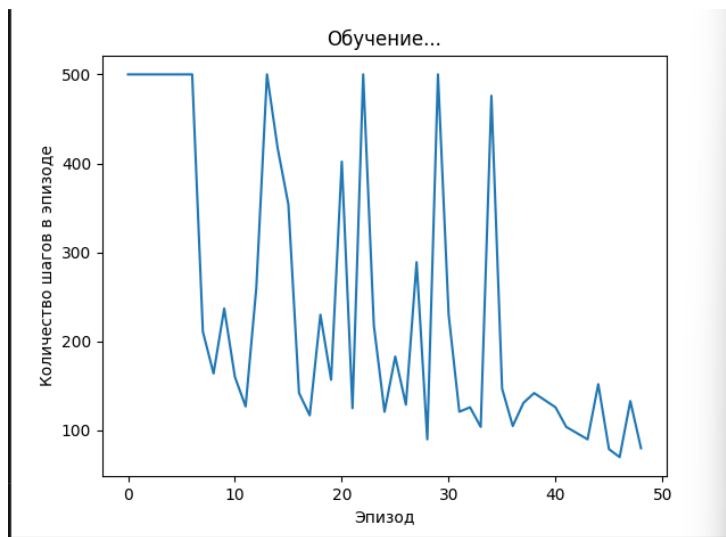
Москва, 2023

## **Задание**

1. На основе рассмотренных на лекции примеров реализуйте алгоритм DQN.
2. В качестве среды можно использовать классические среды (в этом случае используется полносвязная архитектура нейронной сети).
3. В качестве среды можно использовать игры Atari (в этом случае используется сверточная архитектура нейронной сети).
4. В случае реализации среды на основе сверточной архитектуры нейронной сети +1 балл за экзамен.

## Выполнение задания

1. На основе рассмотренных на лекции примеров реализуйте алгоритм DQN.



2. В качестве среды можно использовать классические среды (в этом случае используется полносвязная архитектура нейронной сети).

```
'''
Инициализация топологии нейронной сети
'''
super(DQN_Model, self).__init__()
self.layer1 = nn.Linear(n_observations, 128)
self.layer2 = nn.Linear(128, 128)
self.layer3 = nn.Linear(128, n_actions)
```

В качестве среды используется Acrobot-v1.

3. В качестве среды можно использовать игры Atari (в этом случае используется свёрточная архитектура нейронной сети).

```
'''
Инициализация топологии нейронной сети
'''
HIDDEN_SIZE = 64
super(DQN_Model, self).__init__(),
global_step_count = 0
self.nn_layers = nn.ModuleList()
self.conv1 = nn.Conv2d(1, HIDDEN_SIZE, kernel_size=5, stride=2, padding=2)
# self.conv2 = nn.Conv2d(HIDDEN_SIZE, HIDDEN_SIZE*2, 3, stride=1, padding=1)
self.max_pool = nn.MaxPool2d(4)
self.flatten = nn.Flatten()
self.linear1 = nn.Linear(520, 32)

self.linear2 = nn.Linear(32, n_actions)
self.nn_layers.append(self.conv1)
# self.nn_layers.append(self.conv2)
self.nn_layers.append(self.flatten)
self.nn_layers.append(self.linear1)
self.nn_layers.append(self.linear2)
```

В качестве среды используется ALE/Pong-v5.

4. В случае реализации среды на основе сверточной архитектуры нейронной сети +1 балл за экзамен.