Московский государственный технический университет им. Н.Э. Баумана

Факультет «Головной учебно-исследовательский и методический центр

профессиональной реабилитации лиц с ограниченными возможностями здоровья (инвалидов)»

Кафедра «Системы обработки информации и управления»



**Лабораторная работа 6**

**по дисциплине «Методы машинного обучения в АСОИУ»**

**"** **Обучение на основе глубоких Q-сетей"**

СТУДЕНТ:

студент группы ИУ5Ц-21М

Москалик А.А.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

Москва, 2024

# 

# Цель лабораторной работы: ознакомление с базовыми методами обучения с подкреплением на основе глубоких Q-сетей.

# Задание:

# - На основе рассмотренных на лекции примеров реализуйте алгоритм DQN.

# - В качестве среды можно использовать классические среды.

# - В качестве среды можно использовать игры Atari.

# Ход работы

# Импорт библиотек и настройка среды

# 

# В данном блоке кода задаются основные параметры и настройки, необходимые для обучения агента с использованием глубоких Q-сетей (DQN) на примере игры CartPole. Вот описание ключевых параметров и их значений:

# `env = gym.make('CartPole-v1')`: Эта строка создаёт экземпляр игровой среды CartPole версии 1 из библиотеки OpenAI Gym. В этой игре цель состоит в том, чтобы удерживать шест, который стоит на движущейся тележке, в вертикальном положении как можно дольше.

# `state\_size = env.observation\_space.shape[0]`: `state\_size` определяет размерность вектора состояний среды. Для CartPole это обычно 4, что соответствует таким параметрам среды, как позиция тележки, скорость тележки, угол наклона шеста и скорость вращения шеста.

# `action\_size = env.action\_space.n`: `action\_size` определяет количество возможных действий, которые агент может выбрать в данной среде. Для CartPole это 2: двигать тележку влево или вправо.

# `batch\_size = 32`: Этот параметр указывает размер мини-пакета (batch), который используется для обучения нейронной сети. Значение 32 является стандартным выбором, обеспечивая хороший баланс между временем обучения и качеством модели.

# n\_episodes = 10`: Указывает количество эпизодов, для которых будет обучаться агент. Эпизод заканчивается, когда шест упадёт или будет достигнут предел времени эпизода. В этом случае установлено значение 10, что является довольно малым количеством для демонстрации или начального тестирования.

# output\_dir = 'model\_output/cartpole/'`: Путь, где будут сохраняться файлы весов модели после обучения. Это позволяет в дальнейшем использовать обученную модель без необходимости повторного обучения.

# `os.makedirs(output\_dir)`: Этот вызов создаёт каталог `output\_dir`, если он не существует. Это гарантирует, что программа не столкнётся с ошибкой при попытке сохранить данные в несуществующую директорию.

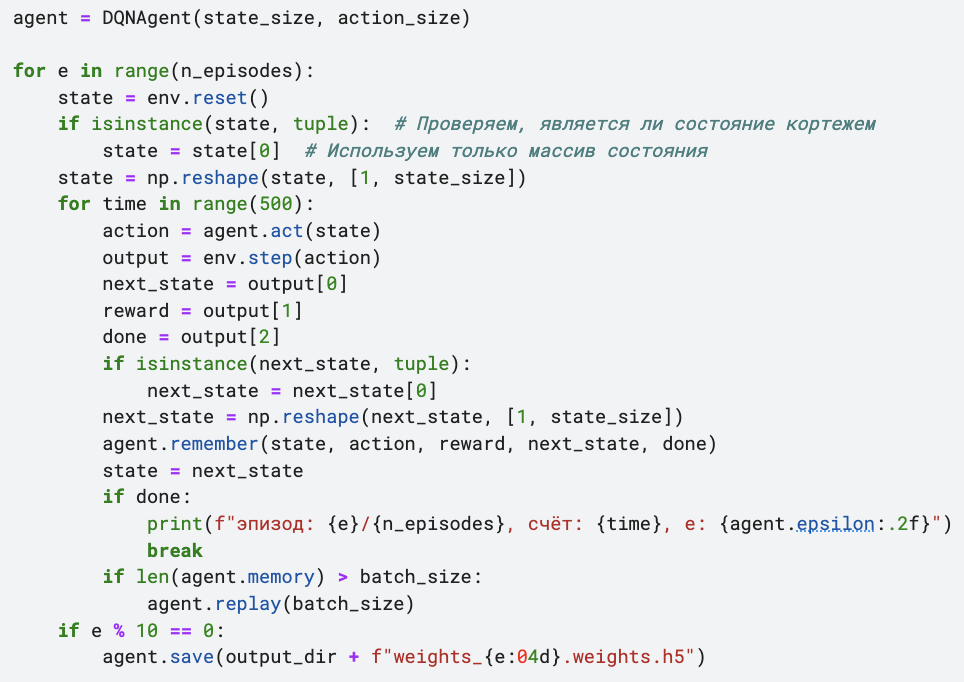
# 2. Определение класса DQNAgent

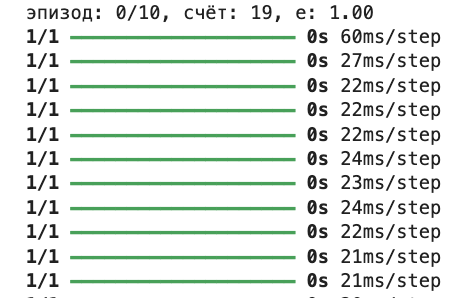
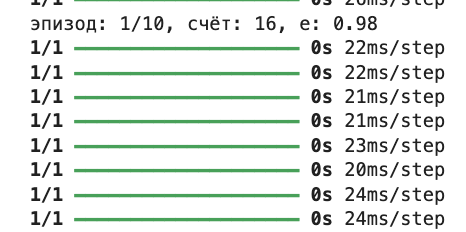
# Класс DQNAgent представляет собой реализацию агента, который использует глубокие Q-сети (Deep Q-Networks, DQN) для обучения и принятия решений в задаче обучения с подкреплением. В данном случае агент обучается управлять тележкой таким образом, чтобы шест, установленный на ней, оставался в вертикальном положении как можно дольше.

# 

# Создание и тренировка агента

# Этот этап лабораторной работы включает инициализацию и обучение агента с использованием глубокой Q-сети (DQN) на задаче CartPole. Основная цель состоит в том, чтобы агент научился поддерживать шест в вертикальном положении на движущейся тележке максимально долго.





Эти строки вывода предоставляют информацию о процессе обучения вашего агента DQN в симуляции среды CartPole.

Эпизоды и счёт:

эпизод: 0/10, счёт: 11, e: 1.00 - Это сообщение говорит о том, что первый эпизод закончился, и за это время агент смог удержать шест в вертикальном положении на тележке 11 временных шагов. Значение e: 1.00 относится к параметру ε в стратегии ε-greedy, который используется для баланса между исследованием новых действий и использованием уже известных знаний. Значение 1.00 означает, что агент на этом этапе полностью фокусируется на исследовании.

Логи обучения модели:

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step и последующие - Эти строки показывают, как происходит процесс обучения нейронной сети на данных, собранных за эпизод. 1/1 означает, что производится одна итерация обучения (одна эпоха) за каждый шаг обучения. Значения типа 0s 63ms/step показывают временные затраты на каждый шаг обучения.

**Вывод:**

В ходе выполнения данной лабораторной работы была реализована модель глубокой Q-сети (DQN) для решения задачи балансировки шеста на движущейся тележке в симулированной среде CartPole из библиотеки OpenAI Gym. Эта задача демонстрирует основные принципы обучения с подкреплением, где агент должен научиться принимать решения на основе окружающего состояния для максимизации получаемого вознаграждения.

Реализация и анализ глубокой Q-сети показали значительные возможности методов обучения с подкреплением в контролируемых условиях. Понимание и умение применять эти методы открывает широкие перспективы для исследований и разработок в области искусственного интеллекта и машинного обучения.