Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика, искусственный интеллект и системы управления»

Кафедра «Системы обработки информации и управления»



**Рубежный контроль № 6**

**по дисциплине «Методы машинного обучения»**

Обучение на основе глубоких Q-сетей

ИСПОЛНИТЕЛЬ:

студентка ИУ5-23М

Морозевич М.А.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

\_\_\_ "\_\_\_\_\_\_\_\_" 2024 г.

Москва, 2024

# **Задание**

В рамках работы предполагается ознакомление с базовыми методами обучения с подкреплением на основе глубоких Q-сетей.

На основе рассмотренных на лекции примеров реализуйте алгоритм DQN. В качестве среды можно использовать классические среды (в этом случае используется полносвязная архитектура нейронной сети) или игры Atari (в этом случае используется сверточная архитектура нейронной сети).

# **Описание среды и алгоритма**

Обучение с подкреплением на основе глубоких Q-сетей использует DQN. Это алгоритм обучения с подкреплением, который использует нейронные сети для обучения агента принимать решения в среде с дискретным пространством действий. Его особенности включают:

1. Использование нейронных сетей: обеспечивает обучение агента на основе большого объема данных и сложных состояний среды;
2. Experience Replay: агент хранит и повторно использует предыдущие переживания (experience) для обучения;
3. Целевая сеть (Target Network): использование двух нейронных сетей: одну для оценки Q-значений и другую для вычисления целевых Q-значений;
4. Использование метода Q-обучения: позволяет агенту принимать решения на основе оценки ценности действий в конкретных состояниях.

В качестве среды была выбрана среда Acrobot из библиотеки Gym. Система состоит из двух звеньев, соединенных линейно и образующих цепочку, один конец которой фиксирован. Соединение между двумя звеньями приводится в действие. Цель состоит в том, чтобы повернуть свободный конец внешнего звена, чтобы достичь заданной высоты, приложив крутящий момент к приводу.

Действие предоставляет собой крутящий момент, приложенный к приводимому в действие соединению между двумя звеньями. Их набор:

1. применить -1
2. применить 0
3. применить 1

Наблюдение представляет собой 6 параметров о двух углах поворота шарнира, а также об их угловых скоростях:

|  |  |  |
| --- | --- | --- |
| **Наблюдение** | **Мин значение** | **Мин значение** |
| Косинус theta1 | -1 | 1 |
| Синус theta1 | -1 | 1 |
| Косинус theta2 | -1 | 1 |
| Синус theta2 | -1 | 1 |
| Угловая скорость theta1 |  |  |
| Угловая скорость theta2 |  |  |

Награды:

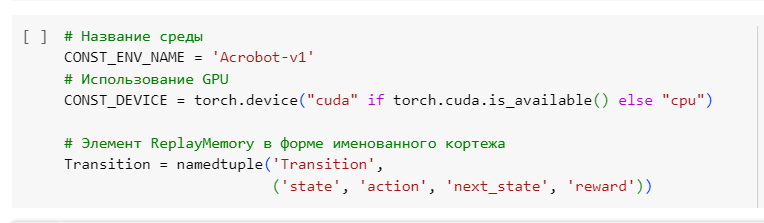
* За все шаги, которые не достигают цели, начисляется вознаграждение в размере -1. Порог составляет -100
* Достижение заданной высоты приводит к завершению с вознаграждением в размере 0.

Эпизод заканчивается, если происходит одно из следующих событий:

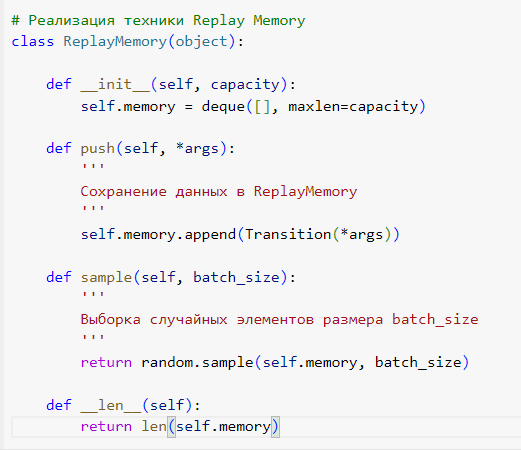
* Завершение: Свободный конец достигает заданной высоты, которая определяется как: -cos(theta1) - cos(theta2 + theta1) > 1,0
* Сокращение: длина эпизода превышает 500 (200 для версии 0)

# **Реализация программы**

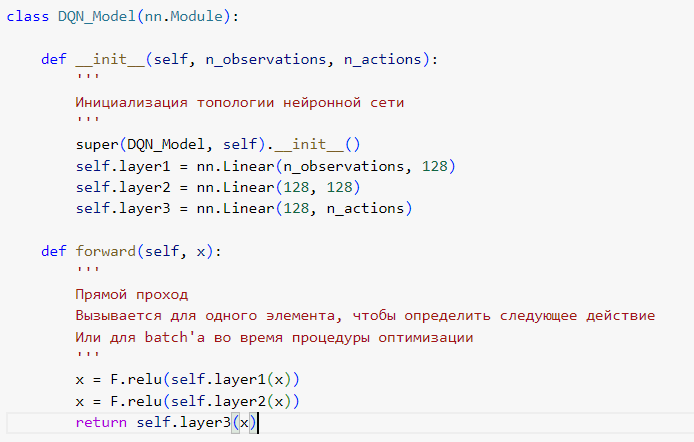
Определим параметры, которые будут необходимы для работы программы.



Добавим реализацию Replay Memory



Добавим модель нейронной сети, которая будет использована для обучения агента.



Определим класс DQN\_Agent, который будет эмулировать работу агента, с методами:

* select\_action – выбирает действие;
* plot\_durations – выводит результаты обучения;
* optimize\_model – обучение НС с улучшением стратегии;
* play\_agent – рендеринг игры для обученного агента;
* learn – обучение агента.

Запустим программу, обучим агента и выведем данные об эпизоде.

