МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №5
по дисциплине «Методы машинного обучения»

Тема: «Обучение на основе временных различий»
Cliff Walking

ИСПОЛНИТЕЛЬ: ___Очеретная С.В.__
ФИО
группа ИУ5-25М _____
подпись
"__"_____2024 г.

ПРЕПОДАВАТЕЛЬ: _____Гапанюк Ю.Е.____
ФИО
_____
подпись
"__"_____2024 г.

Москва - 2024

Цель лабораторной работы: ознакомление с базовыми методами обучения с подкреплением на основе временных различий.

Задание

На основе рассмотренного на лекции примера реализуйте следующие алгоритмы:

- SARSA
- Q-обучение
- Двойное Q-обучение

Для любой среды обучения с подкреплением (кроме рассмотренной на лекции среды Toy Text / Frozen Lake) из библиотеки Gym (или аналогичной библиотеки).

Ход работы

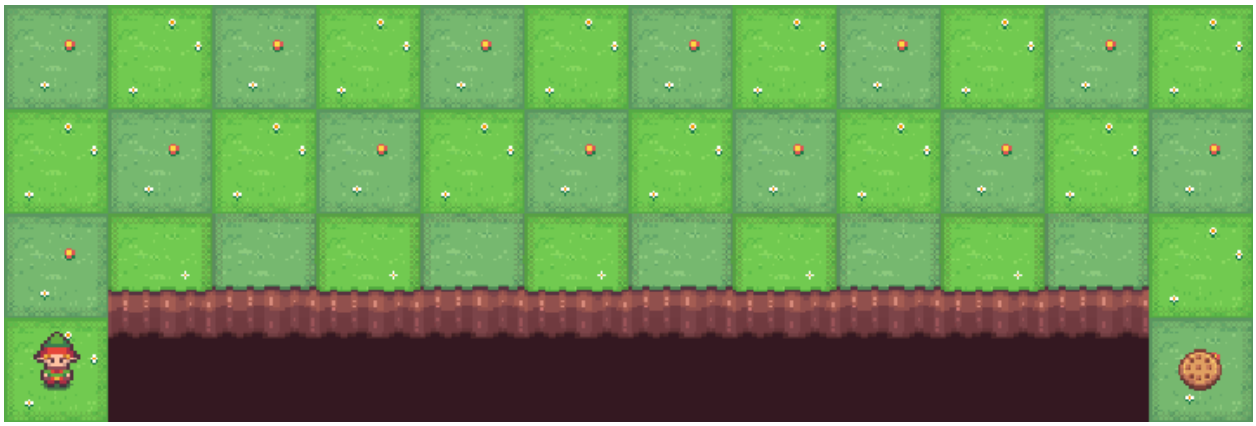
1. Подготовка

Продолжили работать в виртуальном окружении из предыдущей лабораторной (4). Переход в виртуальное окружение:

```
env/Scripts/activate
```

2. Описание среды

Как и в предыдущей лабораторной, будем работать со средой Cliff Walking:



Поле представляет собой матрицу 4x12. Агент начинает проходить карту с ячейки [3, 0] (левый нижний угол). Ему необходимо достичь ячейки [3, 11], т.е. цель размещена в правом нижнем углу. Также агенту нельзя наступать на обрыв – это ячейки [3, 1...10] (внизу, по центру). Если агент наступит на обрыв, он вернется к началу. Эпизод заканчивается, когда агент достигает цели.

Агент может совершить 4 действия:

0: переместиться вверх;

1: передвинуться вправо;

2: передвинуться вниз;

3: передвинуться влево.

За каждый шаг полагается -1 награда, а за шаг в обрыв — -100.

Подробнее данную среду описали в предыдущей лабораторной.

3. Реализация алгоритмов

На этот раз будем производить обучение с помощью алгоритмов SARSA, Q-обучения и двойного Q-обучения.

При обучении в предыдущей лабораторной (policy iteration) нам был известен граф переходов с вероятностями перехода и вознаграждениями. Однако граф и вероятности могут быть неизвестны. В этом случае можно учиться только на основе проб и ошибок, наблюдая среду, выполняя действия и получая вознаграждения. При этом предполагается, что агент может многократно «тренироваться» выполняя действия в среде в течение нескольких эпизодов, и отдельные эпизоды могут завершаться неудачно.

Основой TD-методов является Q-матрица (количество состояний x количество действий). Q-матрица выполняет роль аналогичную стратегии (политике).

SARSA

Алгоритм состоит из следующих действий:

1. Инициализация Q-функции произвольными значениями.
2. Выбор действия из состояния с использованием эпсилон-жадной стратегии ($\varepsilon > 0$) и переход в новое состояние.
3. Обновление Q предыдущего состояния по следующему правилу:

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma Q(s', a') - Q(s, a)),$$

где a' — действие, выбранное по эпсилон-жадной стратегии ($\varepsilon > 0$).

Коэффициент альфа часто также обозначают как learning rate.

Эпсилон-жадная стратегия заключается в выборе **случайного** действия в случае, если случайное число r < эпсилон или в выборе **лучшего** действия, если r >= эпсилон.

На основе данного алгоритма произвели обучение для среды Cliff-Walking. Получили следующий график наград:

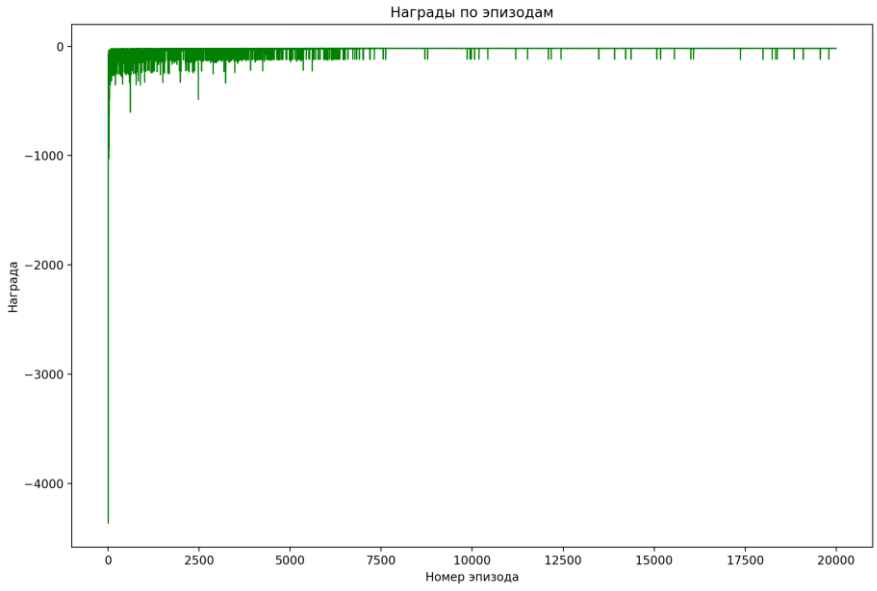


Рис. 1.1. Награды для алгоритма SARSA

Как видим, общий штраф для модели постепенно снижается (т.е. награда растет). При попадании в обрыв получаем награду -100, при каждом шаге -1. В начале графика можно видеть, что было много попаданий в обрыв и награда была сильно отрицательной. Постепенно значение награды

приблизилось к нулю, что означает минимальный штраф за шаги до достижения цели.

При запуске обученной модели получили хороший результат, который сильно лучше, чем для policy iteration из предыдущей лабораторной. Агент достиг цели очень быстро и без попадания в обрыв (однако выбрал проход по верхним ячейкам). Результат:

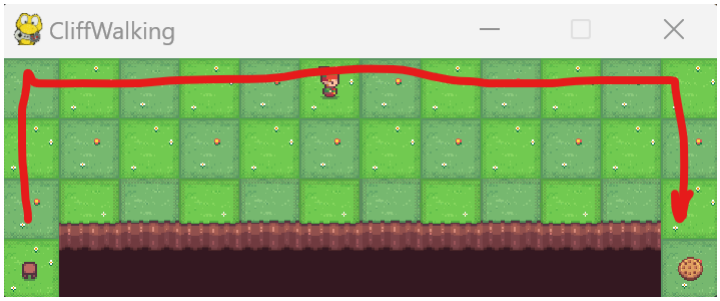


Рис. 1.2. Результат работы модели при алгоритме SARSA

Q-обучение

Алгоритм состоит из следующих действий:

1. Инициализация Q-функции произвольными значениями.
2. Выбор действия из состояния с использованием эпсилон-жадной стратегии ($\varepsilon > 0$) и переход в новое состояние.
3. Обновление Q предыдущего состояния по следующему правилу:

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \max Q(s', a') - Q(s, a)).$$

4. Повторение шагов 3 и 4 до достижения завершающего состояния.

Отличие от предыдущего алгоритма в том, что берем максимум maxQ(s’, a’).

На основе алгоритма Q-обучение произвели обучение для среды Cliff-Walking. Получили следующий график наград:

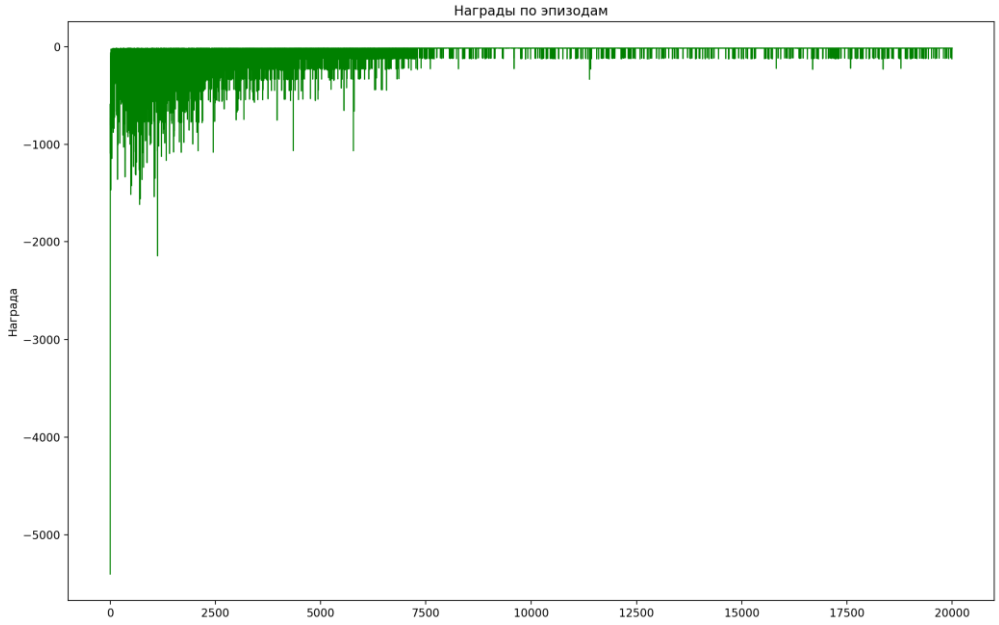


Рис. 2.1. Награды для алгоритма Q-обучение

Обученный агент выполнил следующий путь до цели:

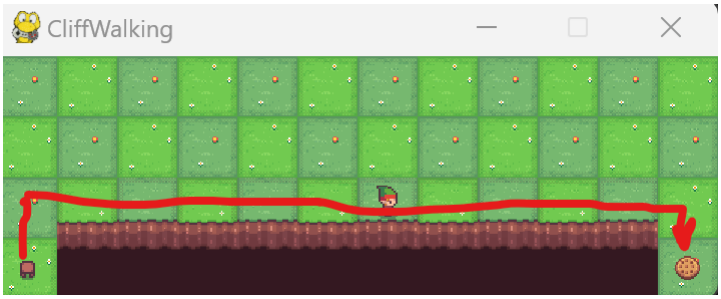


Рис. 2.2. Результат работы модели при алгоритме Q-обучение

На этот раз результат оказался еще лучше, т.к. теперь путь агента до награды – самый короткий из возможных. Но при этом в начале агент делал много действий с большим штрафом (т.е. приближении награды к нулю на этот раз было за большее число итераций).

Двойное Q-обучение

В Q-обучении используется одна и та же выборка как для определения действия, доставляющего максимум, так и для оценки его ценности.

Разобьем все множество игр на два подмножества и будем использовать их для обучения двух независимых оценок, $Q1(a)$ и $Q2(a)$ истинной ценности q(a) для всех действий a. Тогда можно было бы взять одну оценку $Q1$ для определения доставляющего максимум действия

$A* = argmax_a Q1(a)$, а другую, $Q2$, для оценки ценности этого действия: $Q2(A*) = Q2(argmax_a Q1(a))$. Тогда эта оценка будет несмещенной в том смысле, что $\mathbb{E}[Q2(A^*)] = q(A^*)$. Этот процесс можно повторить, поменяв обе оценки ролями, и получить тем самым вторую несмещенную оценку, Q1.

Хотя мы обучаем две оценки, при каждой игре обновляется только одна. Двойное обучение требует двойного объема памяти, но не увеличивает объем вычислений на каждом шаге. Обновление производится по правилу:

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha\left[R_{t+1} + \gamma Q_2(S_{t+1}, \operatorname*{argmax}_a Q_1(S_{t+1}, a)) - Q_1(S_t, A_t)\right]$$

На основе двойного Q-обучения произвели обучение для среды Cliff-Walking. Получили следующий график наград:

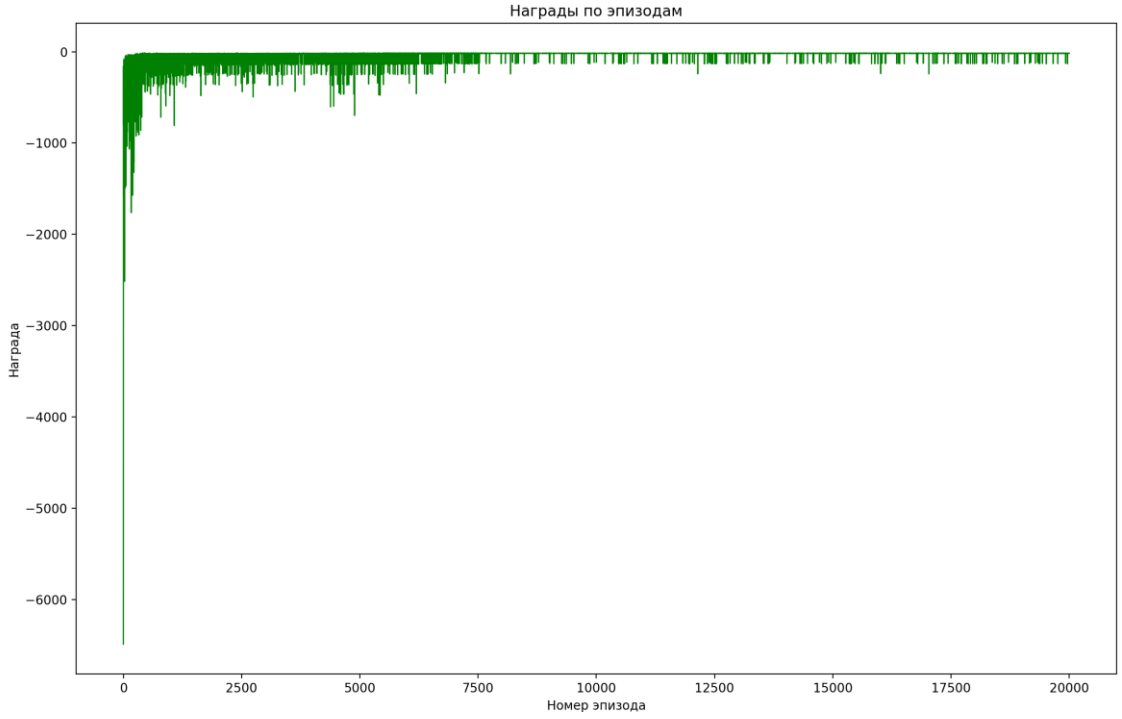


Рис. 3.1. Награды для алгоритма двойное Q-обучение

Путь агента до цели:

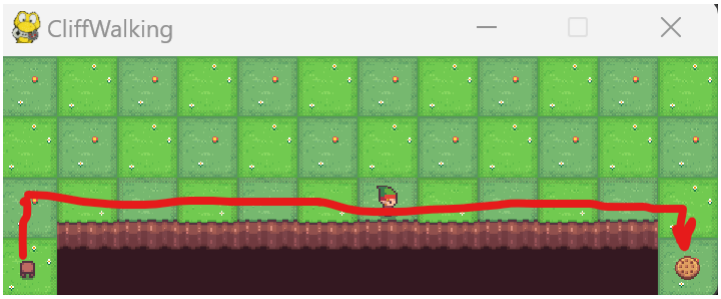


Рис. 3.2. Результат работы модели при алгоритме двойное Q-обучение

Обученный агент дошел до цели также хорошо, как и для Q-обучения, однако штраф уменьшился гораздо быстрее (т.е. быстрее произошло приближении награды к нулю).