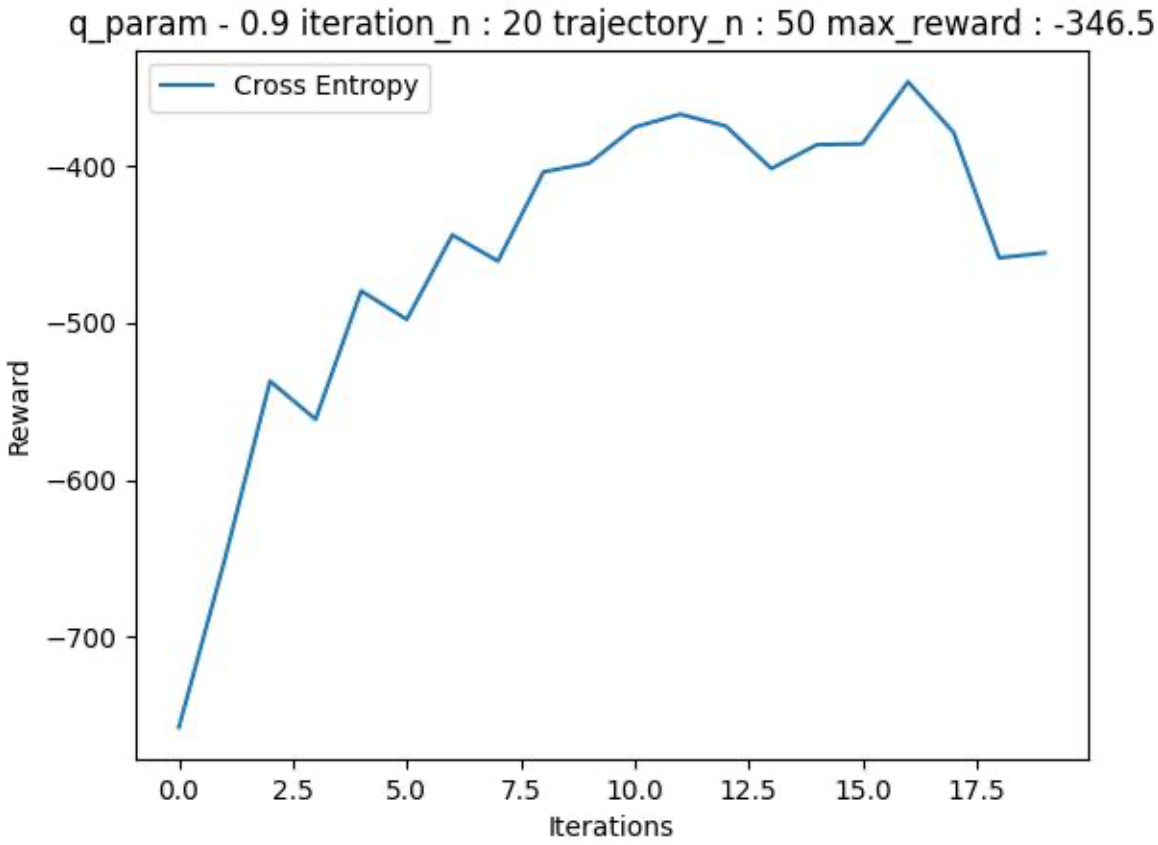
Отчет по домашнему заданию.

Задание 1: Пользуясь алгоритмом Кросс-Энтропии обучить агента решать задачу Taxi-v3 из Gym. Исследовать гиперпараметры алгоритма и выбрать лучшие.

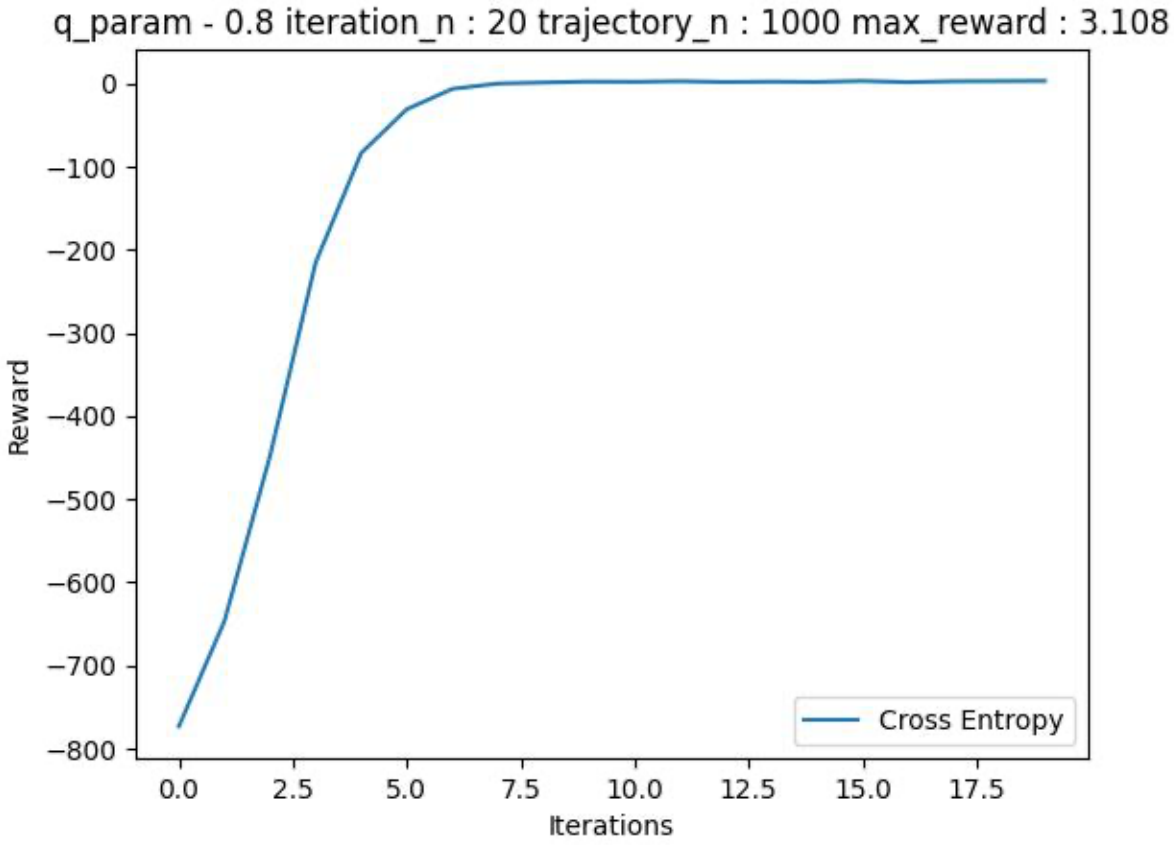
По сути, в первом задании нужно было прочитать внимательно мануал к задаче “Taxi” и поменять некоторые параметры, которые отличаются от игры “Maze”. Игра “Taxi” – сложнее, по сравнению с “Maze”, но для начала я использовал стандартные параметры:

Q-param = 0.9
Iteration number = 20
Trajectory number = 50



Итог: Max_mean_reward = -346.5

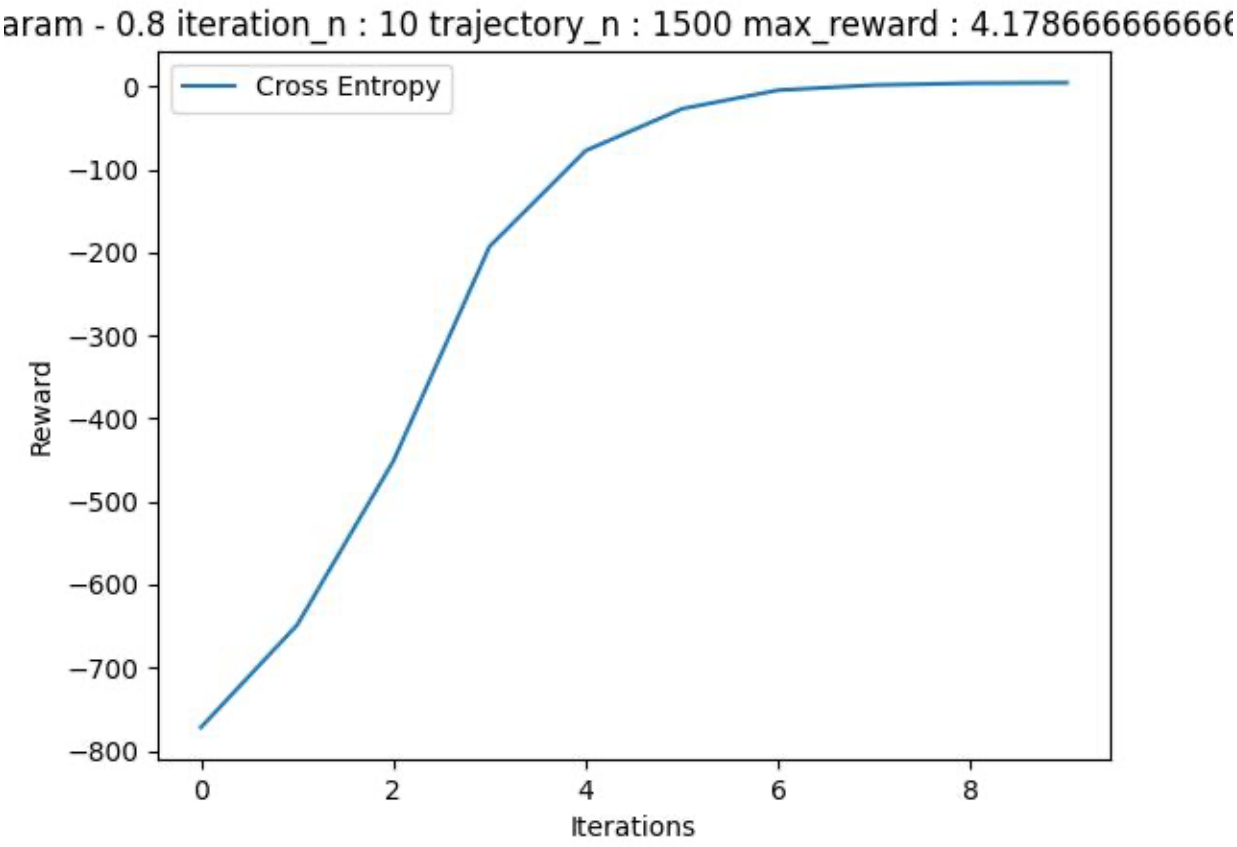
Дальше, я решил увеличивать количество траекторий, что улучшило результат, также уменьшил значение Q-param’a. Параметры:
Q-param = 0.8
Iteration number = 20
Trajectory number = 1000



Итог: Max_mean_reward = 3.108

Видно, что модель после 8-9 итерации стоит на месте, поэтому я решил уменьшить количество итераций до 10, а также попытаться поднять максимальную награду, увеличением траекторий.
Параметры:

Q-param = 0.8
Iteration number = 10
Trajectory number = 1500



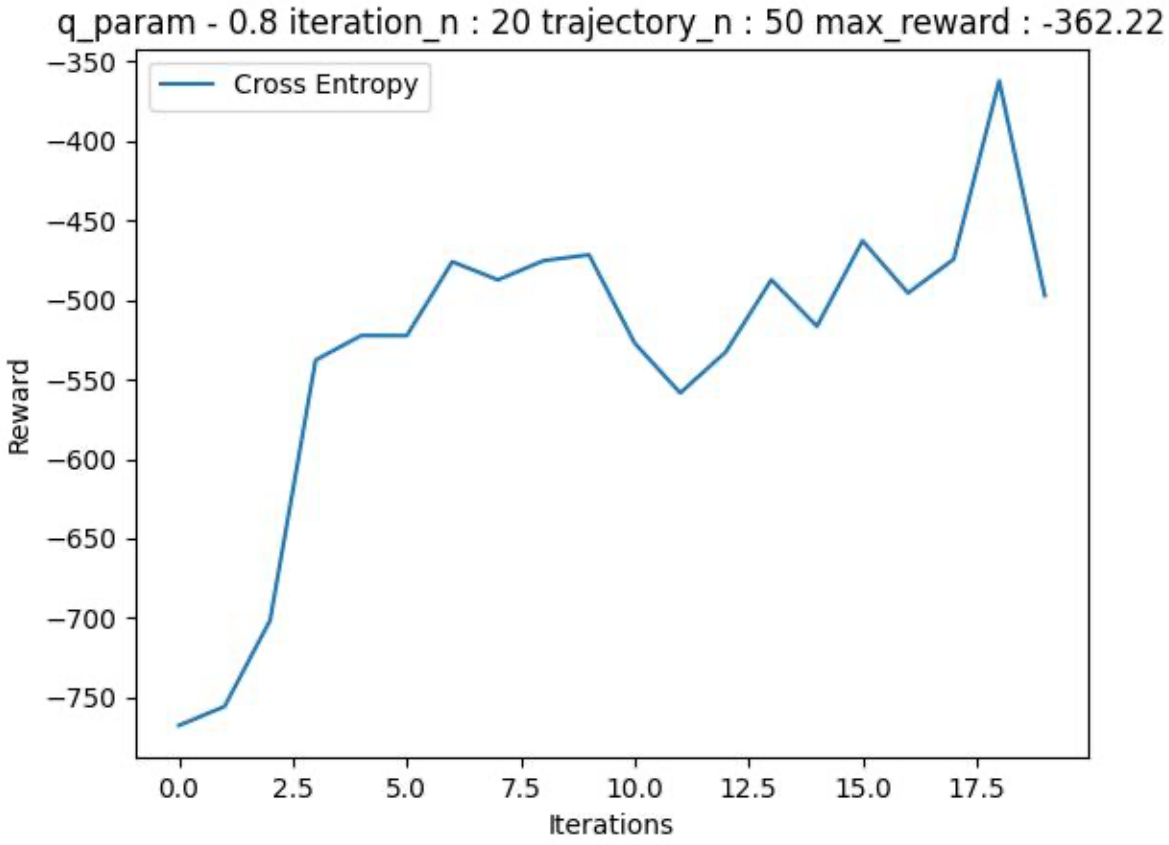
Итог: Max_mean_reward – 4.17

2) Реализовать алгоритм Кросс-Энтропии с двумя типами сглаживания, указанными в лекции 1. При выбранных в пункте 1 гиперпараметров сравнить их результаты с результатами алгоритма без сглаживания.

Хотелось бы начать с метода **Laplase smoothing**.

Тут я тоже решил идти от маленьких/стандартных значений из игры лабиринта. Появился новый параметр lambda, для начала установил его равным 1. Параметры:

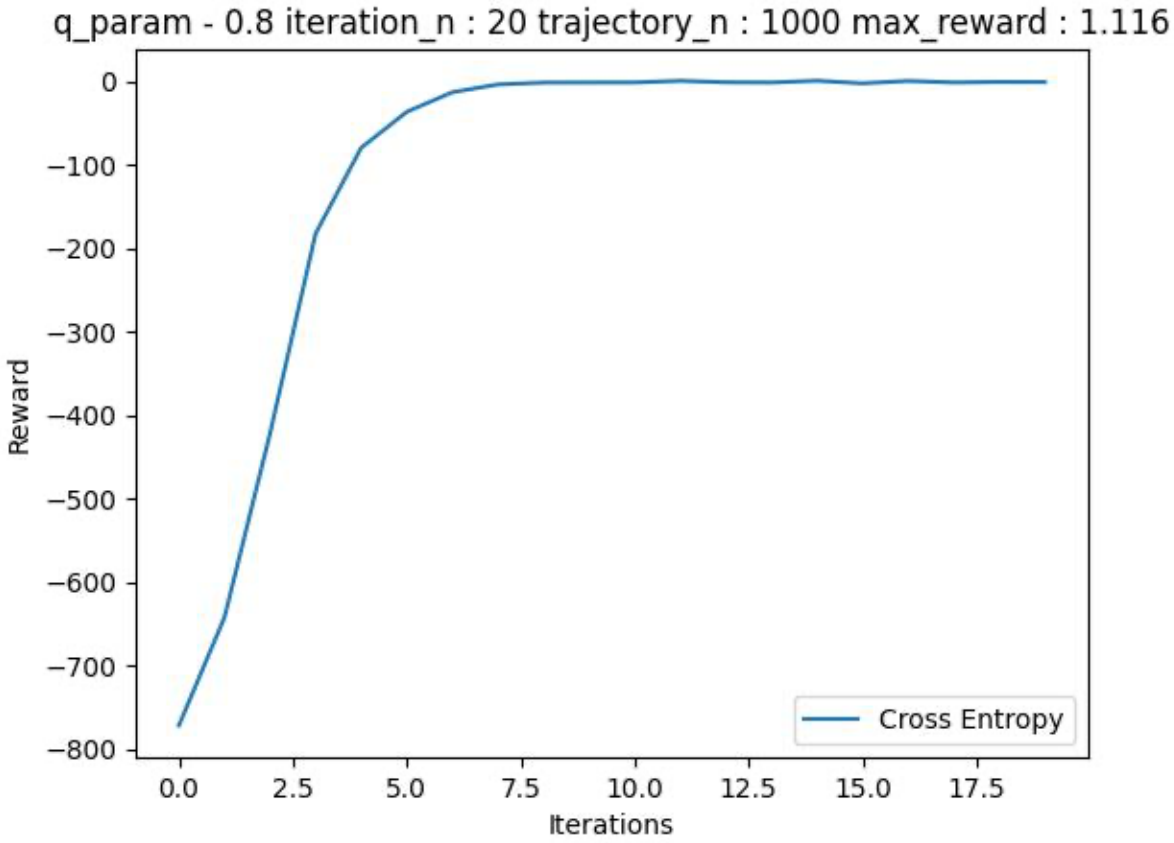
Q-param = 0.8
Iteration number = 20
Trajectory number = 50
Lambda = 1



Итог: max_mean_reward = -362.22

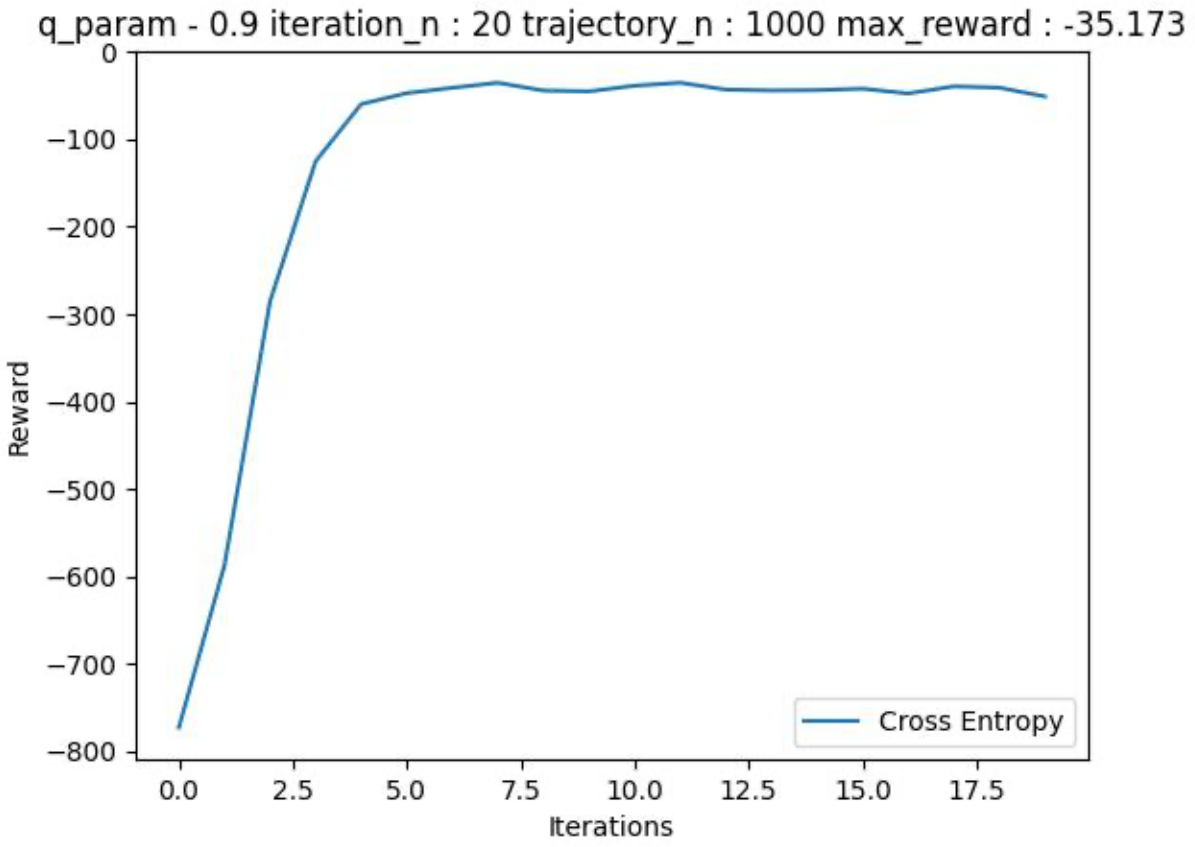
Исходя из опыта предыдущих испытаний Кросс Энтропии, увеличиваю количество траекторий, также параллельно экспериментирую с параметром Q-param, изменение его в большую или меньшую сторону негативно влияло на разультат. Повысил Lambda до 2х

Q-param = 0.8
Iteration number = 20
Trajectory number = 1000
Lambda = 2



Итог: max_mean_reward = 1.116, Q-param = 0.8

Вот пример графика rewards, при тех же параметрах, только с увеличенным Q-param = 0.9

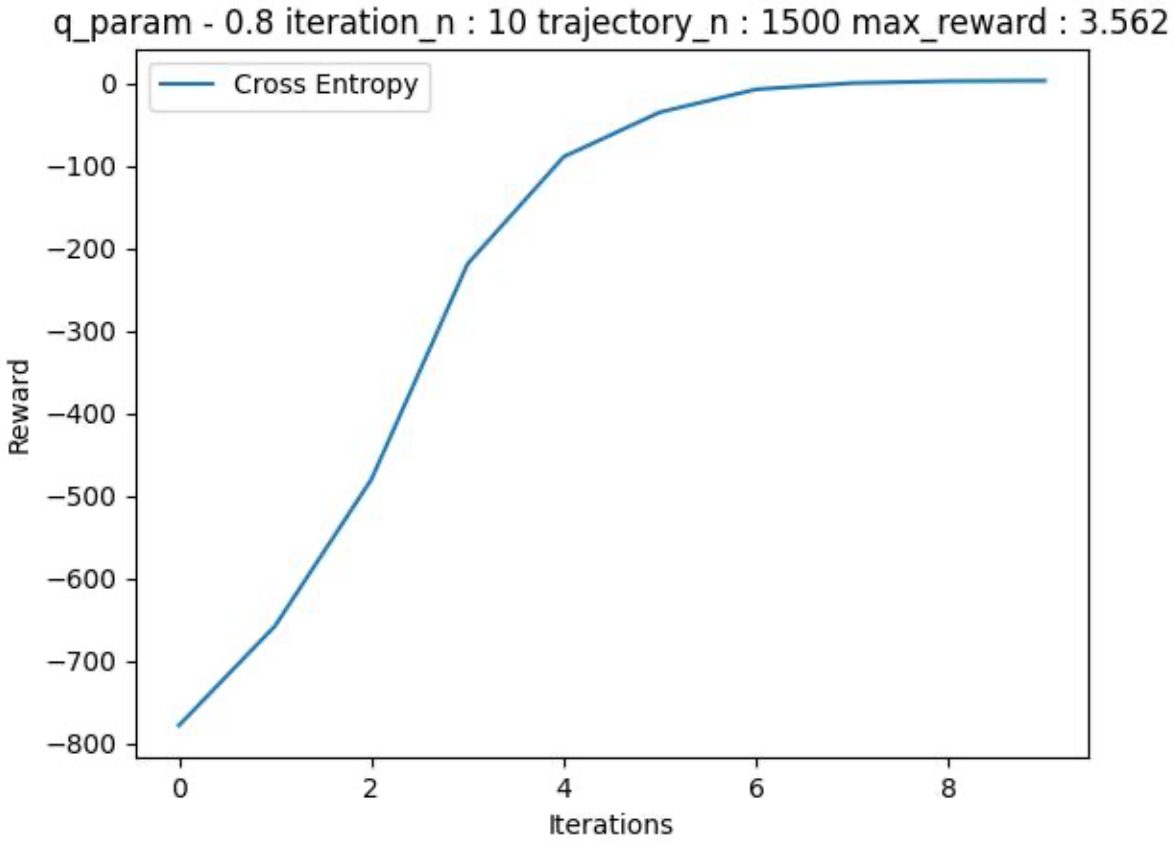


Даже если учитывать небольшой рандом, то все равно не получится повторить результаты при равных параметрах. Так как положение “Taxi” и “Passenger” для каждой модели могут быть разными.

Итог: max_mean_reward = -35.173

В этом случае, также видно, что модель выходит на плато, поэтому решаю уменьшить количество итераций, при этом увеличив количество траекторий.
Параметры:

Q-param = 0.8
Iteration number = 10
Trajectory number = 1500



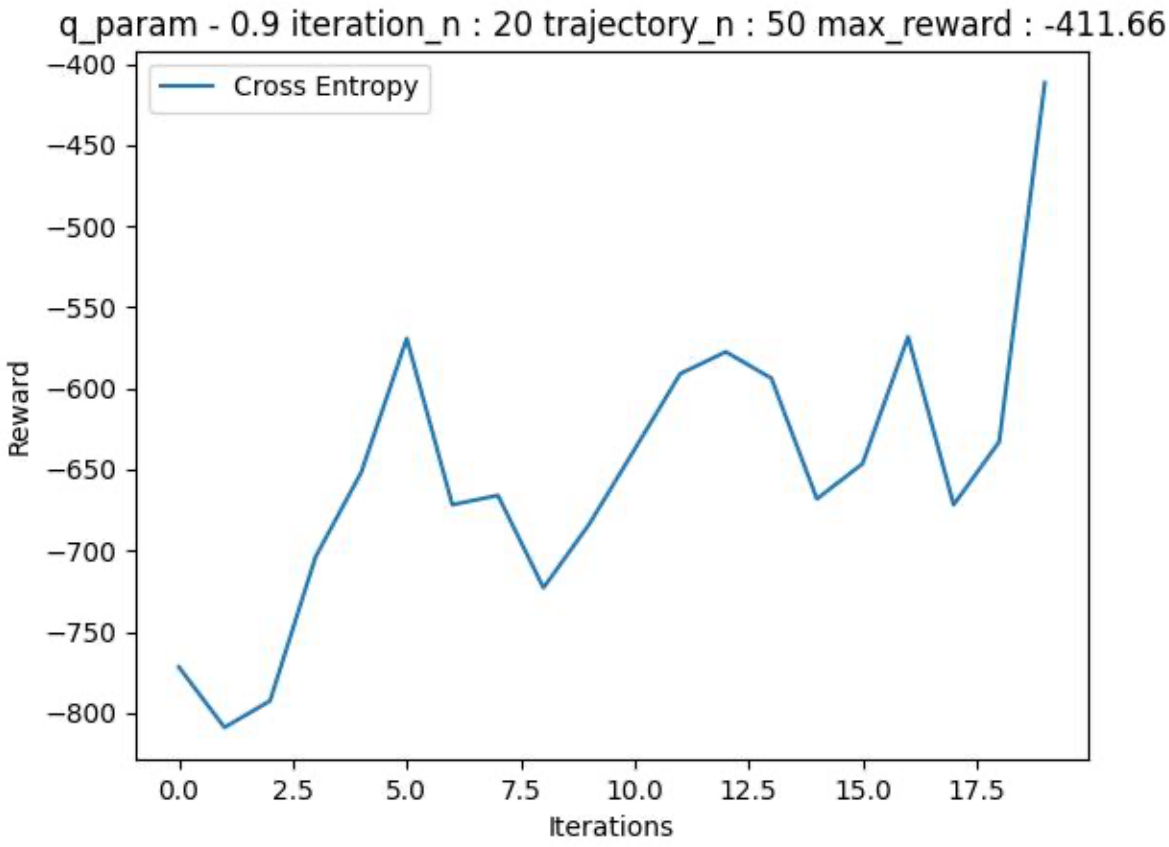
Итог: max_mean_rewards = 3.562

Метод **Policy Smoothing:**

Тут также все стандартно, начинаем с параметров:

Q-param = 0.9
Iteration number = 20
Trajectory number = 50

Также появился новый параметр lambda, для начала установил его равным 0.5

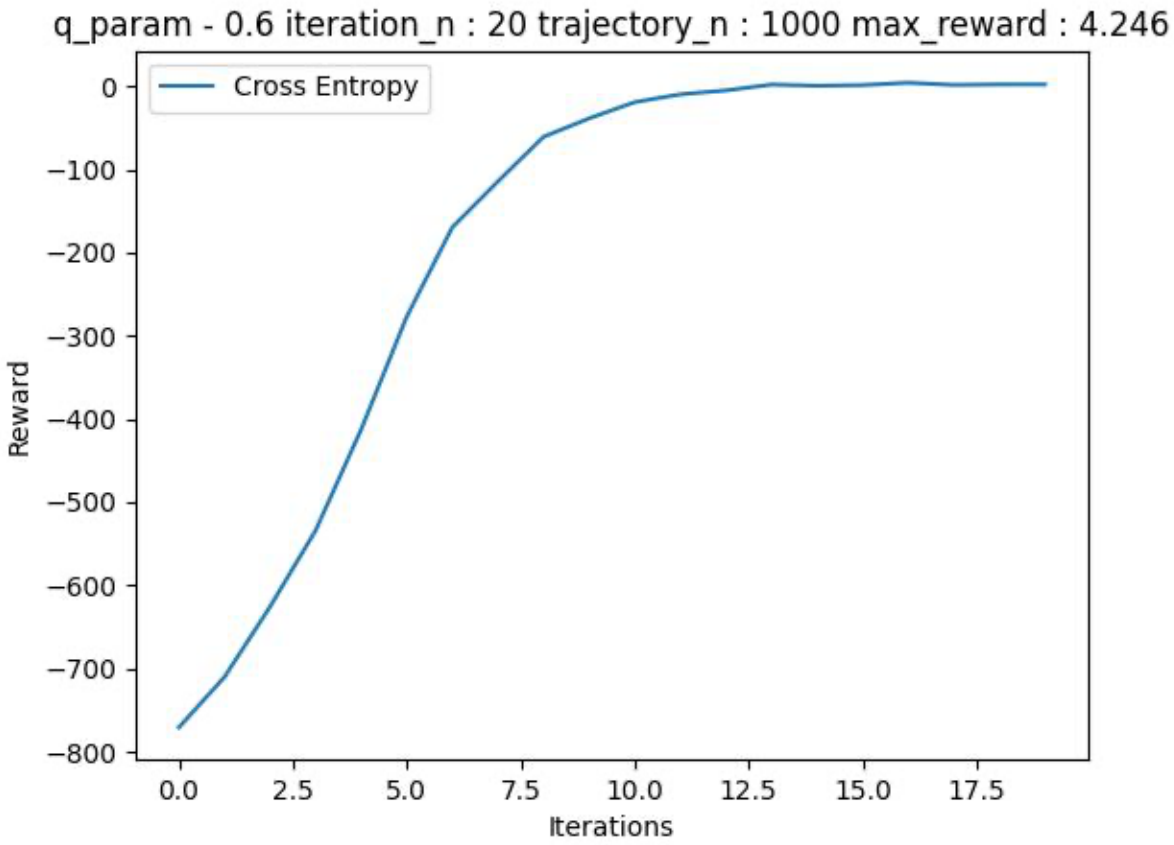


Итог: max_mean_reward = -411.66

После многочисленных попыток подбора параметров, получилось выявить best_model.

Параметры:

Q-param = 0.6
Iteration number = 20
Trajectory number = 1000
Lambda = 0.95



Итог: max_mean_reward = 4.246

Общий итог:
В этом отчете представлена лишь часть результатов, меняя было проведено более 50 итераций подборов параметров. Не всегда получалось повторить лучший результат из-за небольшого рандома в состояниях игры.