Cross Entropy Method. Отчет

Содержание

Cross Entropy Method. Отчет ***1***

1. Пользуясь алгоритмом Кросс-Энтропии обучить агента решать задачу Taxi-v3 из Gym. Исследовать гиперпараметры алгоритма и выбрать лучшие. **2**

Вывод: 5

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. **6**

LAPLACE SMOOTHING 6

Вывод: 8

POLICY SMOOTHING 8

Вывод: 12

3. Реализовать модификацию алгоритм Кросс-Энтропии для стохастических сред, указанную в лекции 1. Сравнить ее результат с алгоритмами из пунктов 1 и 2. **13**

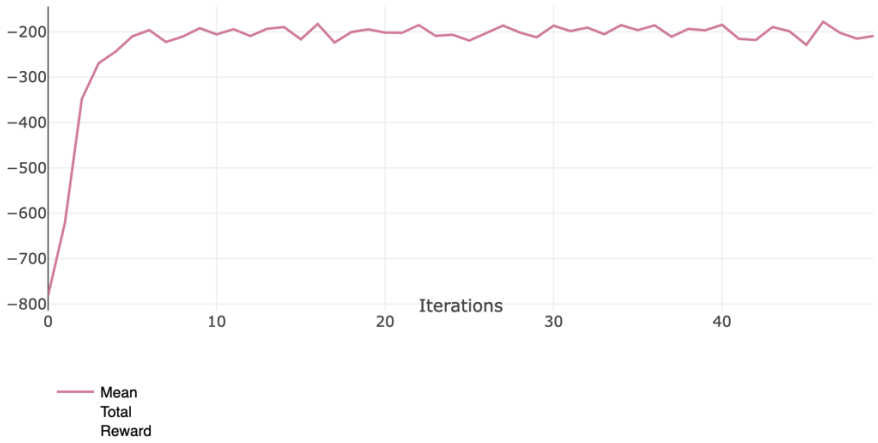
Вывод: 14

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

Были проведены эксперименты с гиперпараметрами ITERATION_N, TRAJECTORY_N, Q_PARAM.

1 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/f0c330bbc75e4b0598fd17131ee8d25d/output/execution

- ITERATION_N: 50
- TRAJECTORY_N: 300
- Q_PARAM: 0.9

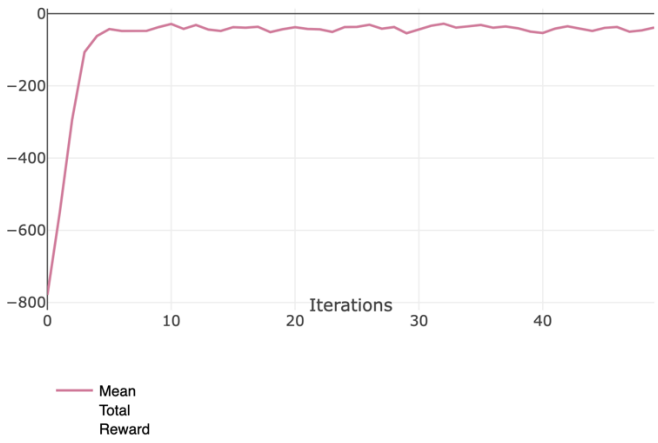


На тесте:

- Total reward: 6
- Кол-во действий на завершение игры: 14

2 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/76da9cc99de04e22968a64689e862346/output/execution

- ITERATION_N: 50
- TRAJECTORY_N: 500
- Q_PARAM: 0.9

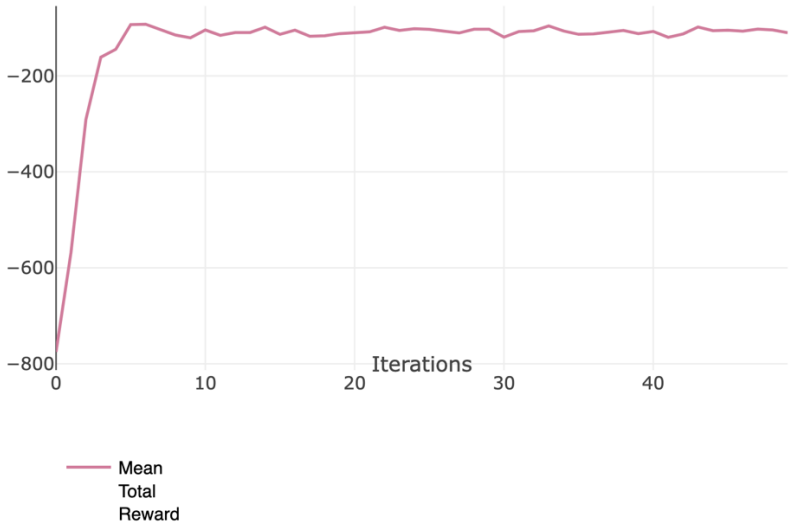


На тесте:

- Total reward: 4
- Кол-во действий на завершение игры: 16

3 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/3bf6b2531de64c7180a0d1b177f014c5/output/execution

- ITERATION_N: 50
- TRAJECTORY_N: 750
- Q_PARAM: 0.9

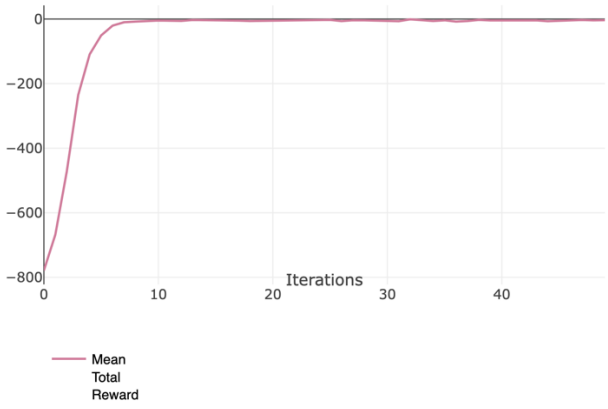


На тесте:

- Total reward: 11
- Кол-во действий на завершение игры: 9

4 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/44c9991934ed4455b21480f67cc3a681/output/execution

- ITERATION_N: 50
- TRAJECTORY_N: 500
- Q_PARAM: 0.75



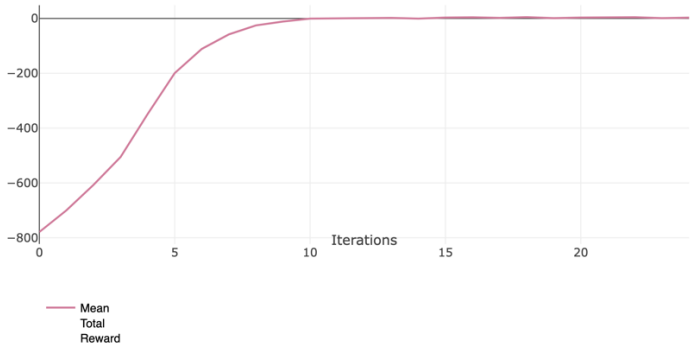
На тесте:

- Total reward: 3

- Кол-во действий на завершение игры: 17

5 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/41f614913e364d7bb93076caeec0f59b/output/execution

- ITERATION_N: 25
- TRAJECTORY_N: 500
- Q_PARAM: 0.6

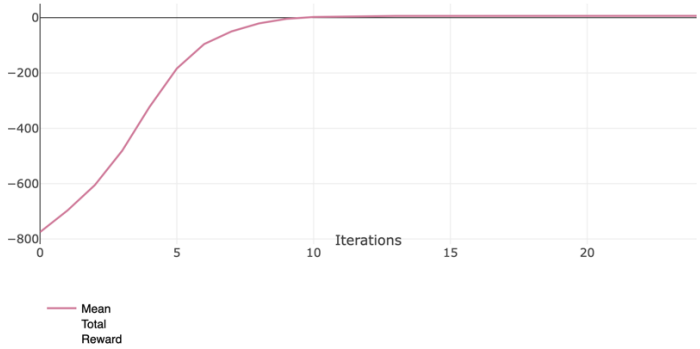


На тесте:

- Total reward: 13
- Кол-во действий на завершение игры: 7

6 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/55a59f4bc39a4c5caf1f0fdc9f99f4d6/output/execution

- ITERATION_N: 25
- TRAJECTORY_N: 750
- Q_PARAM: 0.6

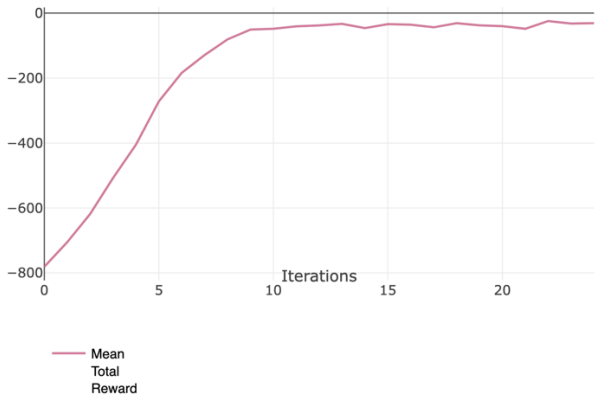


На тесте:

- Total reward: 11
- Кол-во действий на завершение игры: 9

7 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/2df5827e44984fa89540f88903acb76e/output/execution

- ITERATION_N: 25
- TRAJECTORY_N: 300
- Q_PARAM: 0.6



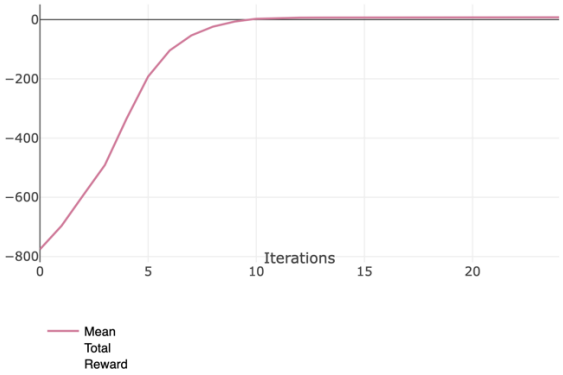
На тесте:

- Total reward: 6
- Кол-во действий на завершение игры: 14

8 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/6b2d84cdc29e43fa892534085a57106c/output/execution

- ITERATION_N: 25
- TRAJECTORY_N: 1000
- Q_PARAM: 0.6



На тесте:

- Total reward: 10
- Кол-во действий на завершение игры: 10

Вывод:

- **Оптимальное количество итераций**, которые необходимы для хорошего изучения окружения, **примерно равно 20 – 2**5, так как дальше нет улучшения по rewards.
- **Оптимальное количество траектории ~500**, так как если меньше, то агент ведет себя нестабильно и не выходит по rewards в положительное значение.
- **Оптимальный Q ~0.6**, что меня удивило, так как я ожидал, что оно будет в районе 0.9 и выше, но это не так. Я это связываю с тем, что начальное состояние такси и пассажира случайны, кроме того, кол-во оптимальных траекторий, которые позволяет завершить игру также является достаточно большим.

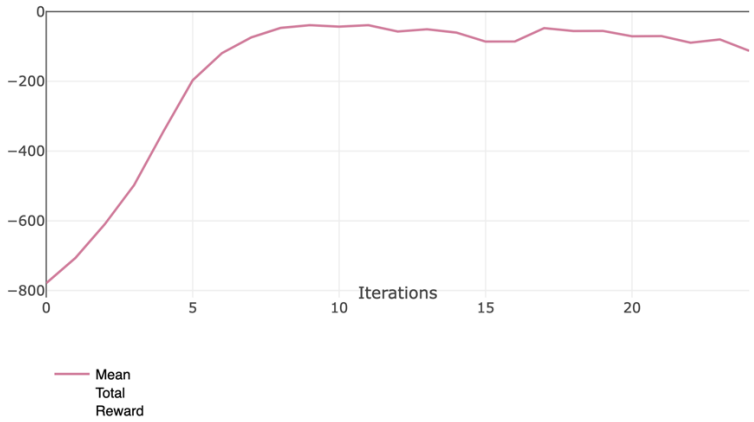
2. Реализовать алгоритм Кросс-Энтропии с двумя типами сглаживания, указанными в лекции 1. При выбранных в пункте 1 гиперпараметров сравнить их результаты с результатами алгоритма без сглаживания.

Были проведены эксперименты с гиперпараметром LAMBDA, гиперпараметры ITERATION_N = 25, TRAJECTORY_N = 500, Q_PARAM ~0.6 в соответствии с лучшими из 1 пункту

LAPLACE SMOOTHING

1 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/4709037ad42543e297287cf8bc47b006/output/execution

- LAMBDA: 0.1

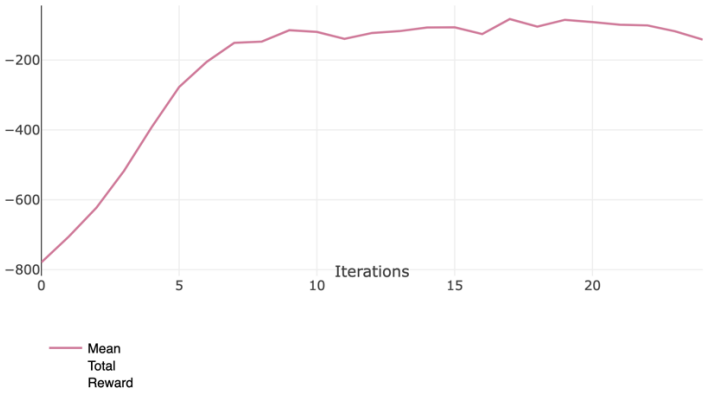


На тесте:

- Total reward: -10
- Кол-во действий на завершение игры: 21

2 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/e8ade66046ce4af7a7730b06f8fd71e6/output/execution

- LAMBDA: 0.5

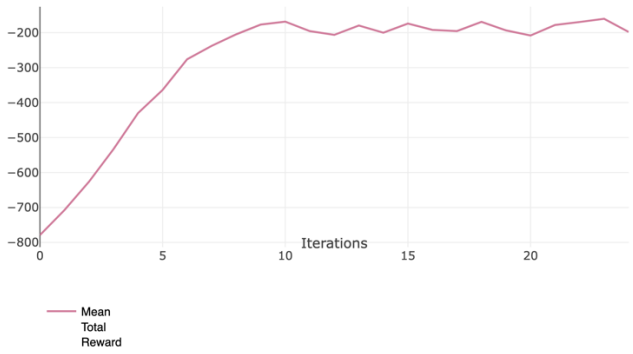


На тесте:

- Total reward: -96
- Кол-во действий на завершение игры: 44

3 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/defe373997d8438ab136188e60dcbd3c/output/execution

- LAMBDA: 1

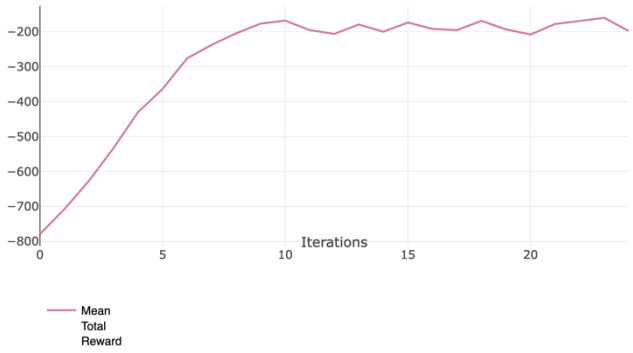


На тесте:

- Total reward: 3
- Кол-во действий на завершение игры: 17

4 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/4bc320ef8f734728b7f66523050e5592/output/execution

- LAMBDA: 10

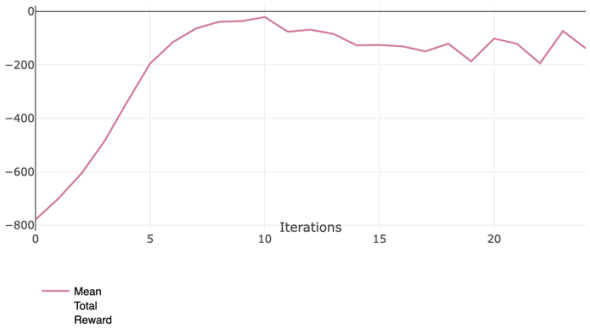


На тесте:

- Total reward: -650
- Кол-во действий на завершение игры: 199

5 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/94b7858a728b4c249fb8cc6e8e05fdbf/output/execution

- LAMBDA: 1e-5



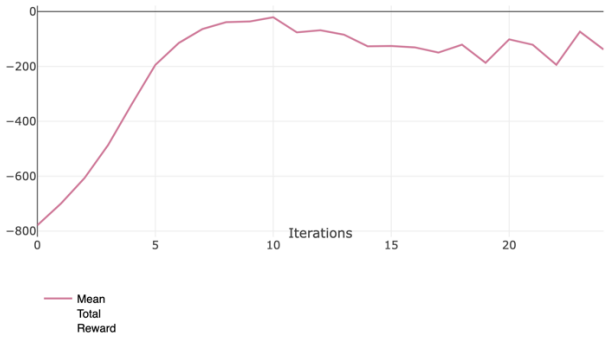
На тесте:

- Total reward: 10
- Кол-во действий на завершение игры: 10

6 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/c4ea8419ee614c86908b7d2573715599/output/execution

- LAMBDA: 1e-9



На тесте:

- Total reward: 10
- Кол-во действий на завершение игры: 10

Вывод:

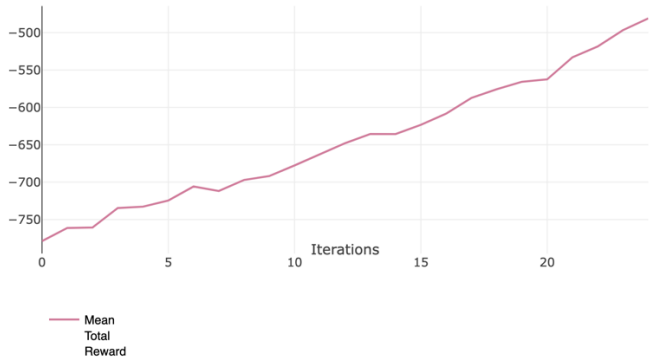
При применении данного вида сглаживания **оптимальным является LAMBDA <= 1e-2**. Если LAMBDA больше этого значения, то средний reward на обучении практически не поднимается выше -150, кроме значения LAMBDA 1, но на тесте, результат не оптимальный. Можно сделать вывод, нужно **аккуратно поднимать LAMBDA** для данного вида сглаживания, так как **график reward на обучении выглядит весьма хаотично и начинает снижаться где-то с 10 эпохи**, но при это **reward на тесте и кол-во шагов для завершения игры выглядят достаточно хорошими** при LAMBDA <= 1e-2, практически также как и без сглаживания.

POLICY SMOOTHING

1 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/1983abb9139440f8b03f69837fd1ebfc/output/execution

- LAMBDA: 0.1



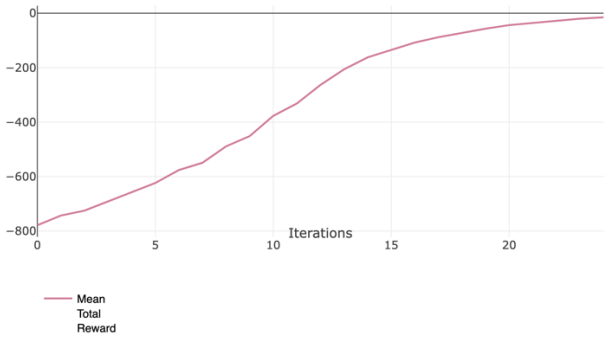
На тесте:

- Total reward: -327
- Кол-во действий на завершение игры: 95

2 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/4ba8b1cd98dd4351a2bc24260f6cbbe2/output/execution

- LAMBDA: 0.3



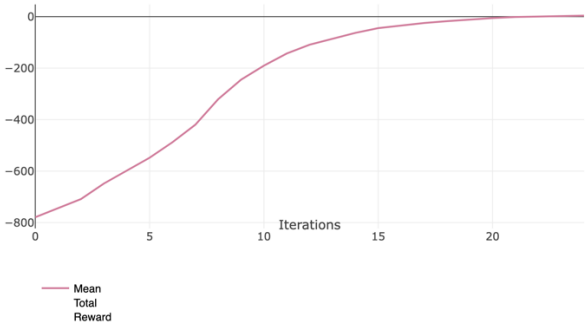
На тесте:

- Total reward: 8
- Кол-во действий на завершение игры: 12

3 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/9bd22658d4f54302a4f5c3ba91fcdffb/output/execution

- LAMBDA: 0.4



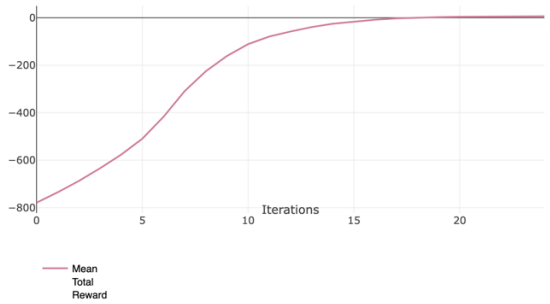
На тесте:

- Total reward: 8
- Кол-во действий на завершение игры: 12

4 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/f34bf37bc85f4a99828d5d4763a98a64/output/execution

- LAMBDA: 0.5



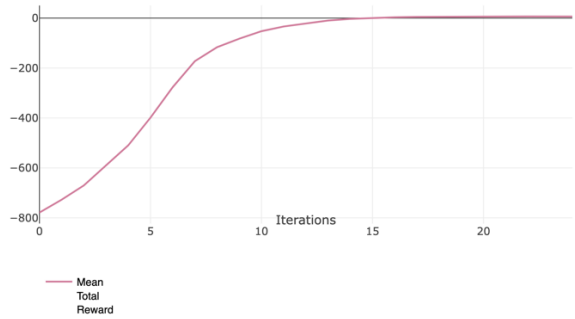
На тесте:

- Total reward: 7
- Кол-во действий на завершение игры: 13

5 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/f34bf37bc85f4a99828d5d4763a98a64/output/execution

- LAMBDA: 0.6



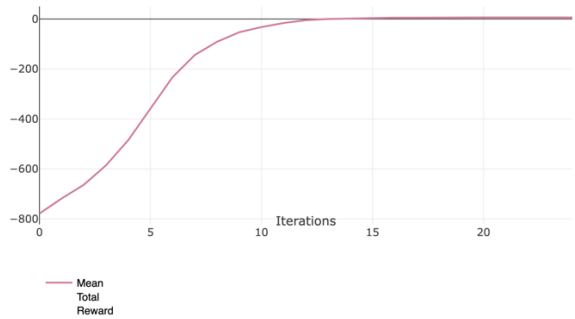
На тесте:

- Total reward: -1
- Кол-во действий на завершение игры: 21

6 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/2a172db9e7274bb3aa6d7dbb23476314/output/execution

- LAMBDA: 0.7



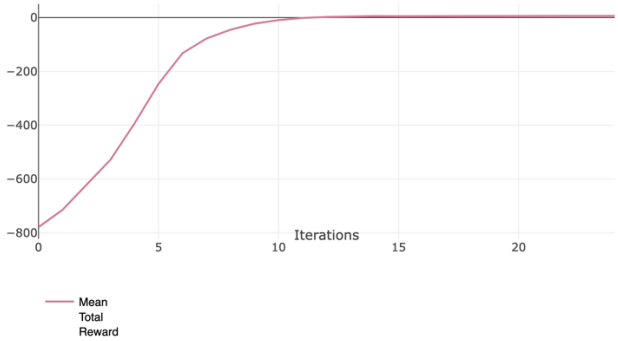
На тесте:

- Total reward: 7
- Кол-во действий на завершение игры: 13

7 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/fac5d9b181f24590a94d25183c8604df/output/execution

- LAMBDA: 0.8



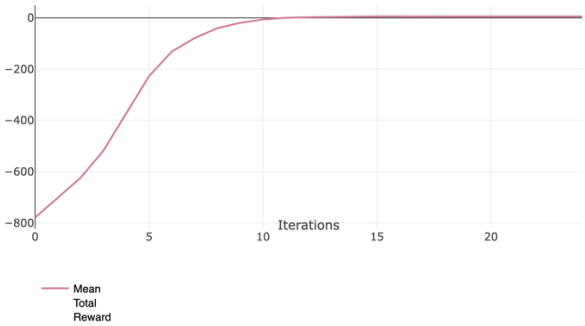
На тесте:

- Total reward: 8
- Кол-во действий на завершение игры: 12

8 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/b58cd0c499fb4eb5bb9a84c8fdc4d2f0/output/execution

- LAMBDA: 0.9



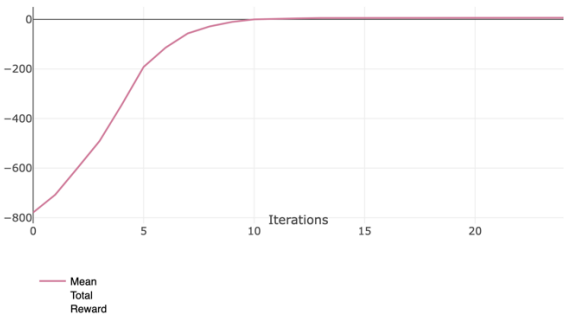
На тесте:

- Total reward: 9
- Кол-во действий на завершение игры: 11

9 Эксперимент:

https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/c604825890ea4d75a2e33cf6154fb92a/output/execution

- LAMBDA: 0.95



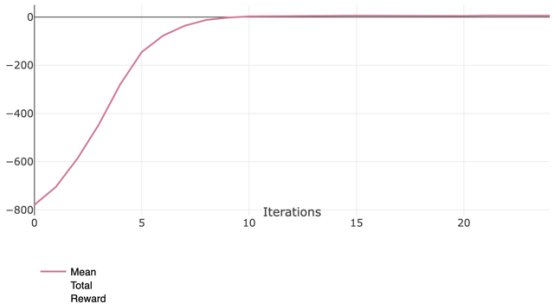
На тесте:

- Total reward: 10

- Кол-во действий на завершение игры: 10

10 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/ea63af4ca3864991b0ebec170dd7ba93/output/execution

- LAMBDA: 0.99

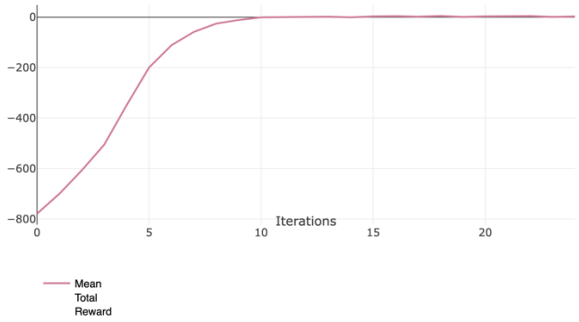


На тесте:

- Total reward: 5
- Кол-во действий на завершение игры: 15

11 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/93f5135d86f743b69d897421269651af/output/execution

- LAMBDA: 1



На тесте:

- Total reward: 7
- Кол-во действий на завершение игры: 13

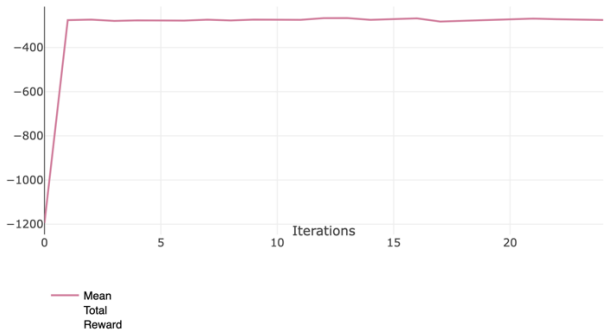
Вывод:
Оптимальный LAMBDA для данного вида сглаживания лежит **в пределах от 0.5 до 1 включительно**. Меньшие значения LAMBDA не успевают «прогреть» модель до нужно уровня, то есть 25 итераций мало для подобных значений LAMBDA, однако, если повысить кол-во итераций обучения, то и при таких значениях агент может хорошо обучиться. Можно сделать вывод, что **чем меньше LAMBDA тем большее кол-во итераций обучения нужно (обратно-пропорциональная зависимость)**. Данный факт можно объяснить тем, что **при меньшим LAMBDA большее предпочтение отдается предыдущей политике и поэтому процесс обучения замедляется.**

3. Реализовать модификацию алгоритм Кросс-Энтропии для стохастических сред, указанную в лекции 1. Сравнить ее результат с алгоритмами из пунктов 1 и 2.

Были проведены эксперименты с гиперпараметром DETERMINISTIC_POLICY_N, гиперпараметры ITERATION_N = 25, TRAJECTORY_N = 500, Q_PARAM ~0.6 в соответствии с лучшими из 1 пункту LAMBDA 0.8 (Policy Smoothing) и 1e-5 (Laplace Smoothing).

1 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/d594589f41e6488e90489343f0009e5e/output/execution

- Simple (without smoothing)
- DETERMINISTIC_POLICY_N: 50

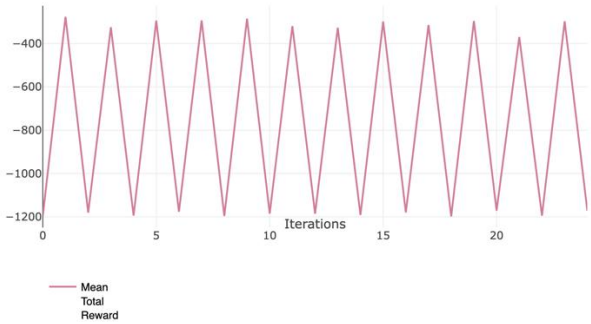


На тесте:

- Total reward: -200
- Кол-во действий на завершение игры: 199

2 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/4a61e2ebd3d34634add46a8142ae34ef/output/execution

- Laplace Smoothing
- DETERMINISTIC_POLICY_N: 50

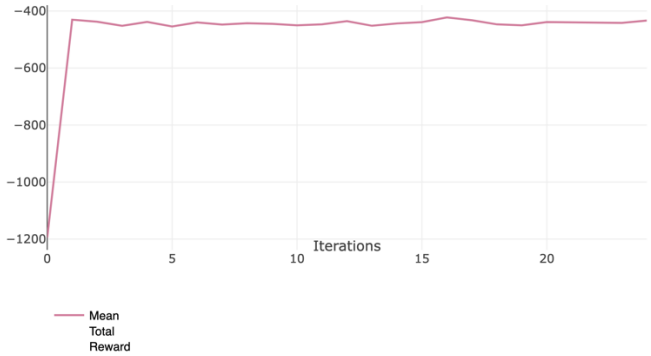


На тесте:

- Total reward: -200
- Кол-во действий на завершение игры: 199

3 Эксперимент:
https://app.clear.ml/projects/1973d7d1f894446c885f09ae225d2992/experiments/74473b4e71a24930a0e906c2f71d8047/output/execution

- Policy Smoothing
- DETERMINISTIC_POLICY_N: 50



На тесте:

- Total reward: -299
- Кол-во действий на завершение игры: 199

Вывод:

Очень долгое обучение. При семплировании детерминированных политик для простого агента и с policy smoothing, reward на обучении доходит до определенного уровня ~400 и на нем закрепляется. При lamplace smoothing график обучения выглядит хаотичным.
Я думаю, это происходит по нескольким причинам:

1. **Мы не уходим от стохастики при семплировании детерминированных политик** (это хорошо видно на первой итерации, когда у нас равномерное распределение в каждом состоянии)
2. **Агент будет сосредотачиваться на более менее хорошей, но не лучшей, траектории** при условии, что остальные будут плохими, а плохих тут может быть достаточно много