Deep Cross Entropy Method. Отчет

Содержание

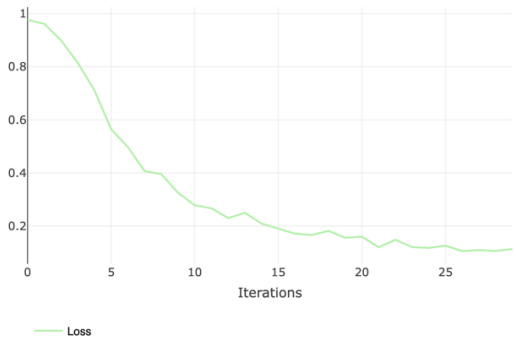
1. Пользуясь алгоритмом Кросс-Энтропии для конечного пространства действий обучить агента решать Acrobot-v1 или ~~LunarLander-v2~~ на выбор. Исследовать гиперпараметры алгоритма и выбрать лучшие.

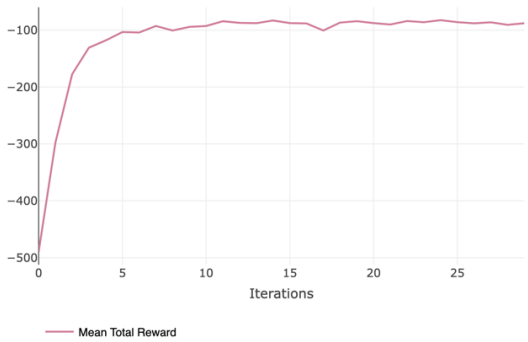
Была выбрана игра **Acrobot-v1**.
Были проведены эксперименты с гиперпараметрами EPISODE_N, Learning rate, Q_param, Trajectory_n.

- Trajectory_len: 500
- Loss function: Cross Entropy Loss

1 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/d137fc8a49434957a2fa9ba5004fd3aa/output/execution

- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.6
- Trajectory_n: 50



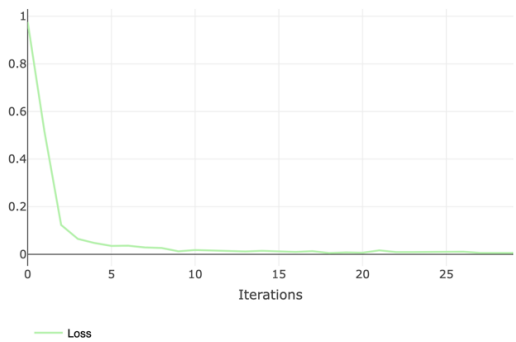


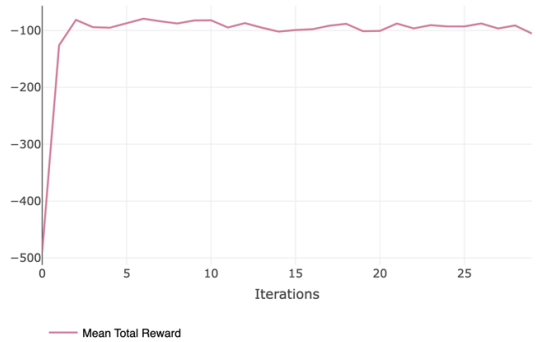
На тесте:

- Total reward: -97

2 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/ddd30da248b54c239fbc8ff2f3fc2b74/output/execution

- EPISODE_N: 30
- Learning rate: 0.1
- Q_param: 0.6
- Trajectory_n: 50



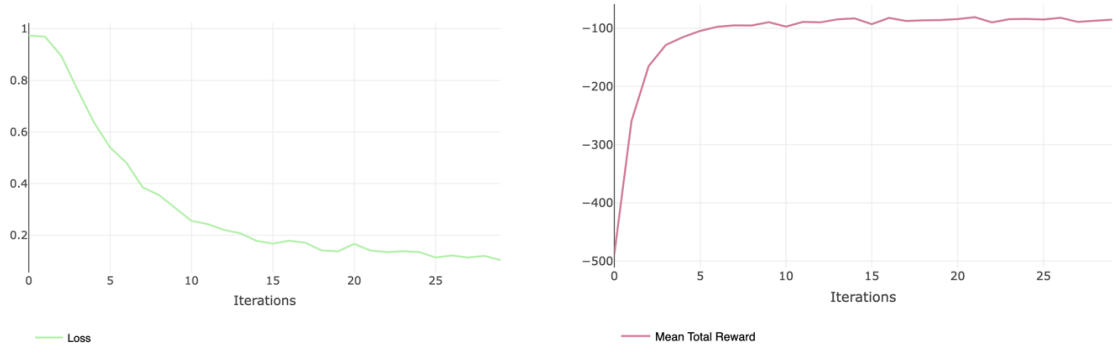


На тесте:

Total reward: -124

3 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/e9190b7938d945ff845f70d18fc077e5/output/execution

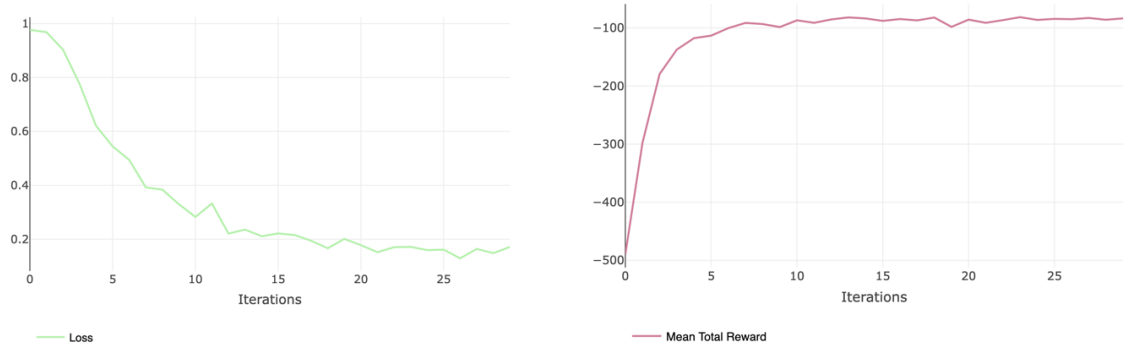
- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.6
- Trajectory_n: 100



На тесте:
Total reward: -92

4 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/76fbe14ed3154ad5aa367a898b5f4cf6/output/execution

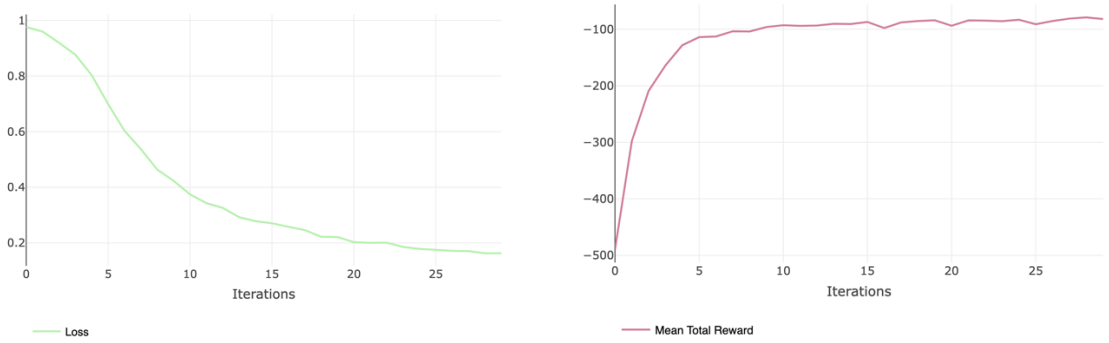
- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.8
- Trajectory_n: 50



На тесте:
Total reward: -96

5 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/cc2603a6da094abeaf46e58e5dbb1493/output/execution

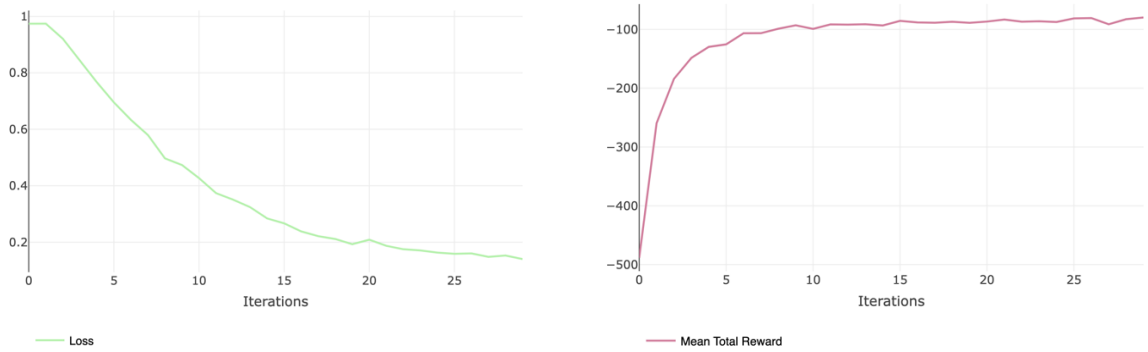
- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.2
- Trajectory_n: 50



На тесте:
Total reward: -71

6 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/899dcf596ba64d6b843f2f7e7024960a/output/execution

- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.2
- Trajectory_n: 100

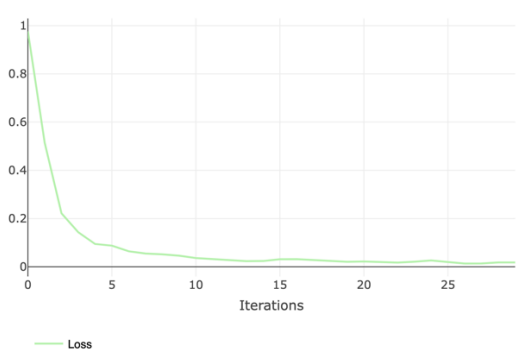


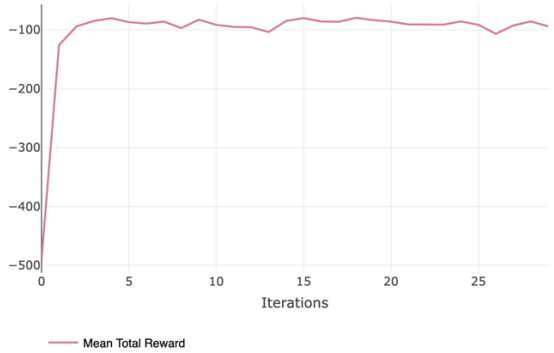
На тесте:

- Total reward: -79

7 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/f91f08c20b4f40e3bb14eb118e25c6b6/output/execution

- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.1
- Trajectory_n: 50





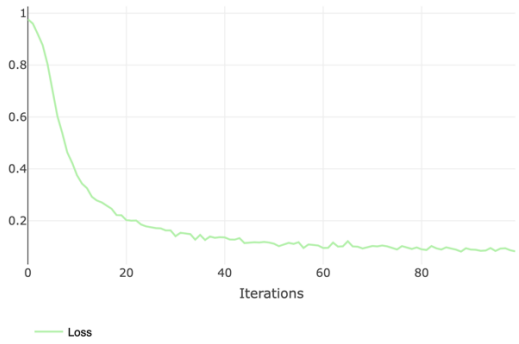
На тесте:

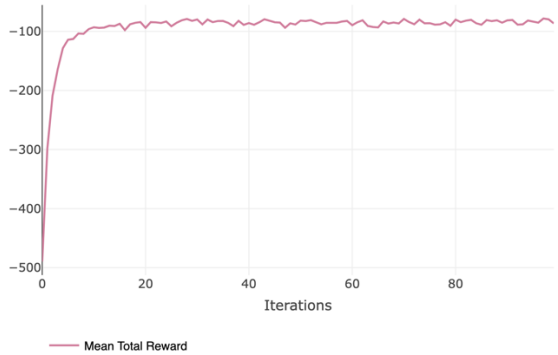
- Total reward: -136

8 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/f91f08c20b4f40e3bb14eb118e25c6b6/output/execution

- EPISODE_N: 100
- Learning rate: 0.01
- Q_param: 0.2
- Trajectory_n: 50





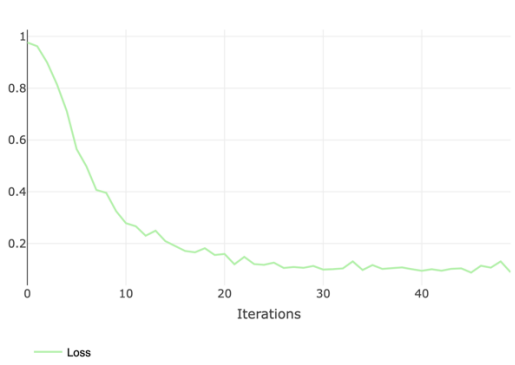
На тесте:

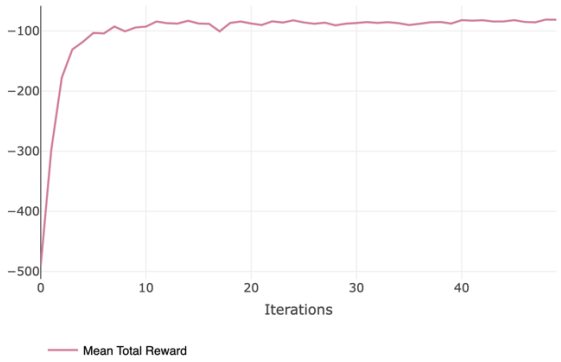
- Total reward: -93

9 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/ecff3e3cfc924ee7b2892c9f82d19d5d/output/execution

- EPISODE_N: 50
- Learning rate: 0.01
- Q_param: 0.6
- Trajectory_n: 50





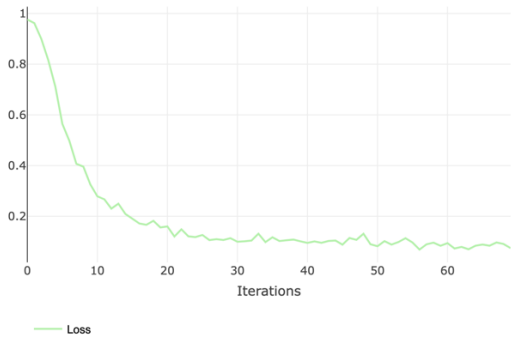
На тесте:

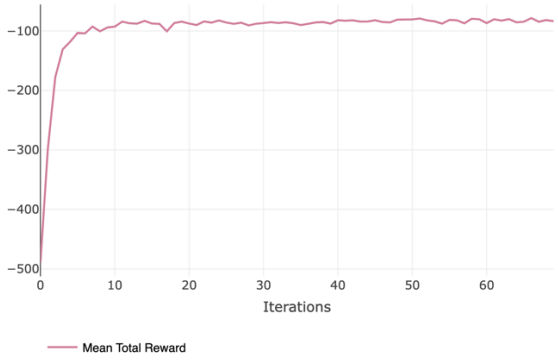
- Total reward: -109

10 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/b9257282992a48b1bc554113f52172a4/output/execution

- EPISODE_N: 70
- Learning rate: 0.01
- Q_param: 0.6
- Trajectory_n: 50





На тесте:

- Total reward: -71

Вывод:

- **Оптимальное количество итераций для обучения примерно равно 30**, так как дальше нет улучшения по rewards.
- **Оптимальное количество траектории ~50**, так как если меньше, то агент ведет себя нестабильно.
- **Оптимальный Q ~0.2**, что меня, как мне показалось закономерным. Так как игра может быть закончена большим количеством траекторий.
- **Learning rate ~0.01**

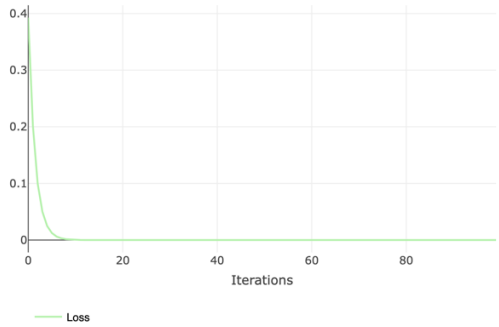
2. Реализовать алгоритм Кросс-Энтропии для непрерывного пространства действий. Обучить агента решать ~~Pendulum-v1~~ или MountainCarContinuous-v0 на выбор. Исследовать гиперпараметры алгоритма и выбрать лучшие.

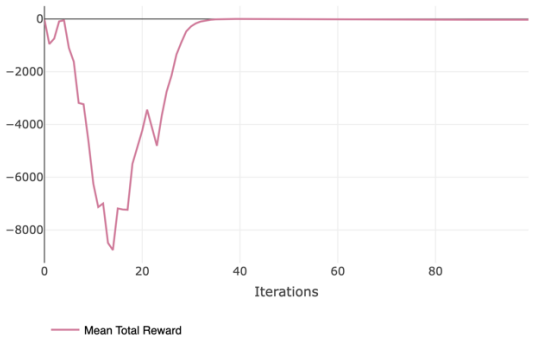
Была выбрана игра **MountainCarContinuous-v0.**
Были проведены эксперименты с гиперпараметрами EPISODE_N, Learning rate, Q_param, Trajectory_n, EPS / Noise, Loss.

- Loss исследовался MSE и MAE
- Использовались различные способы добавления шума: добавлять шум при помощи нормального распределения с std = eps, при помощи нормального распределения с генерацией mean и std из нормального распределение
- Также выбиралась функция активации между Tanh и Clip(min=-1, max=1)
- Trajectory_len: 999

1 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/3ca1f5d00e0f447ea136df4fd7d6f255/output/execution

- EPISODE_N: 100
- Learning rate: 0.1
- Q_param: 0.5
- Trajectory_n: 100
- EPS: 0.5
- Noise: Normal
- Loss: MAE



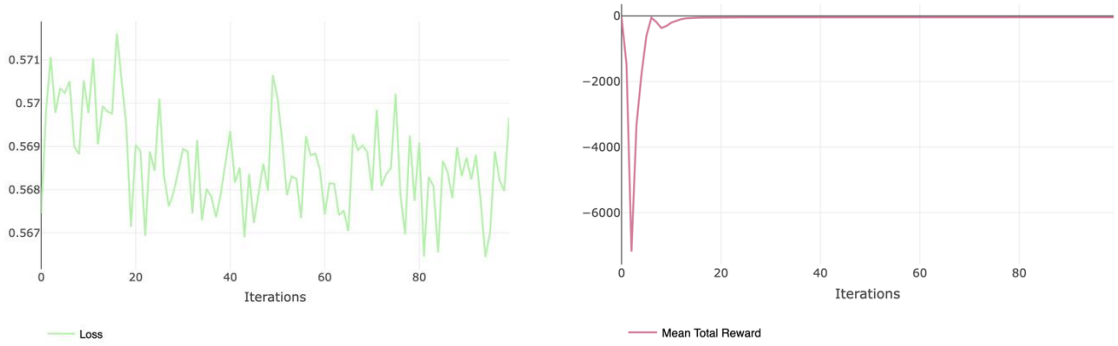


На тесте:

- Total reward: ~-30

2 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/549584e68a02453ea03f15ec62110991/output/execution

- EPISODE_N: 100
- Learning rate: 0.1
- Q_param: 0.5
- Trajectory_n: 100
- EPS: None
- Noise: Normal Mean, Normal STD, Normal Common
- Loss: MAE



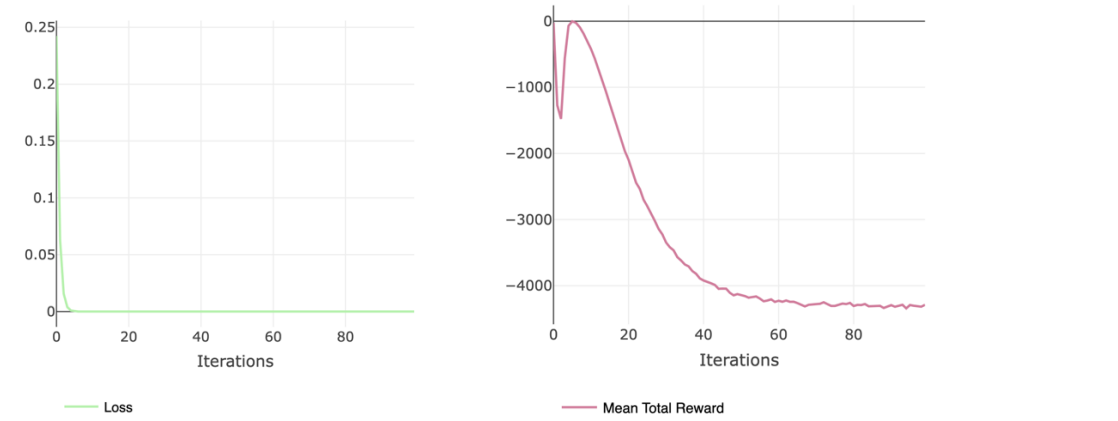
На тесте:

- Total reward: ~0

3 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/c646ce1069a74af18e734cfa0117a070/output/execution

- EPISODE_N: 100
- Learning rate: 0.1
- Q_param: 0.5
- Trajectory_n: 100
- EPS: 0.5
- Noise: Normal Mean, Normal STD, Normal Common
- Loss: MSE



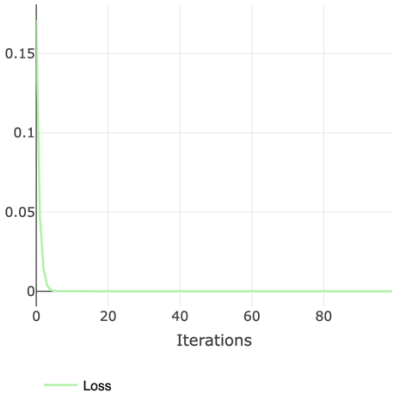
На тесте:

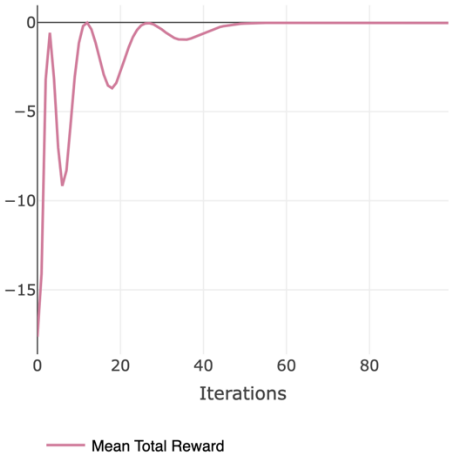
- Total reward: ~-4269

4 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/17ec8cee289e4219b060dc8f45541584/output/execution

- EPISODE_N: 100
- Learning rate: 0.01
- Q_param: 0.5
- Trajectory_n: 100
- EPS: 0.5
- Noise: Normal
- Loss: MSE





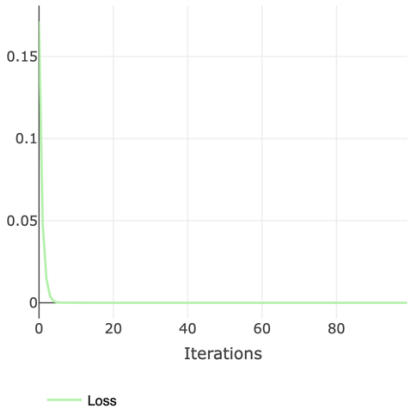
На тесте:

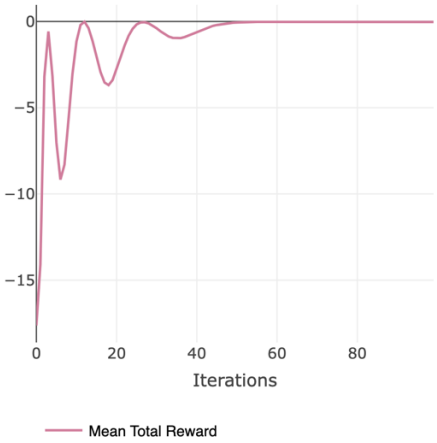
- Total reward: ~0

5 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/17ec8cee289e4219b060dc8f45541584/output/execution

- EPISODE_N: 100
- Learning rate: 0.01
- Q_param: 0.2
- Trajectory_n: 100
- EPS: 0.5
- Noise: Normal
- Loss: MSE





На тесте:

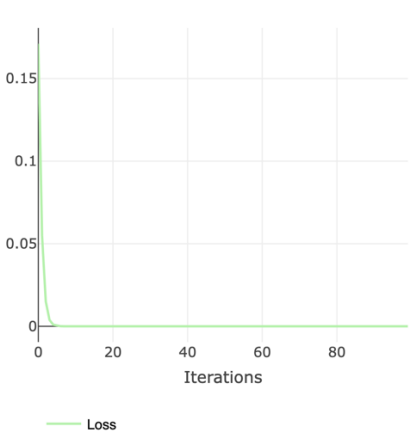
- Total reward: ~0

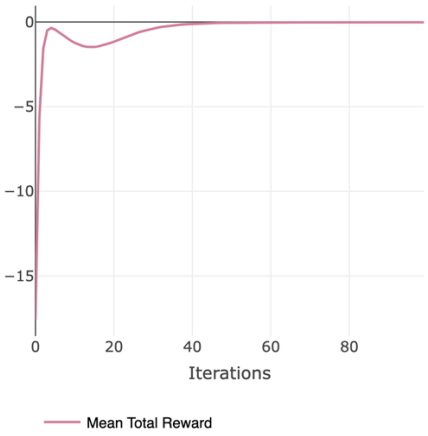
6 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/035a913c7df74908bb0a2a2cc480c7a5/output/execution

- EPISODE_N: 100
- Learning rate: 0.001
- Q_param: 0.2

- Trajectory_n: 500
- EPS: 0.5
- Noise: Normal
- Loss: MSE





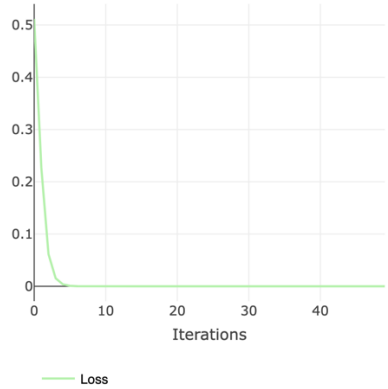
На тесте:

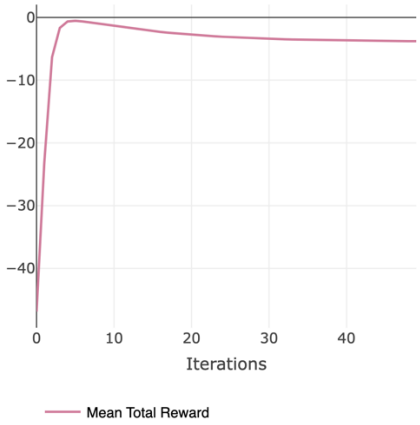
- Total reward: ~0

7 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/ef010316b48748c9885cc396ca379347/output/execution

- EPISODE_N: 50
- Learning rate: 0.001
- Q_param: 0.2
- Trajectory_n: 200
- EPS: 1
- Noise: Normal
- Loss: MSE + CLIP





На тесте:

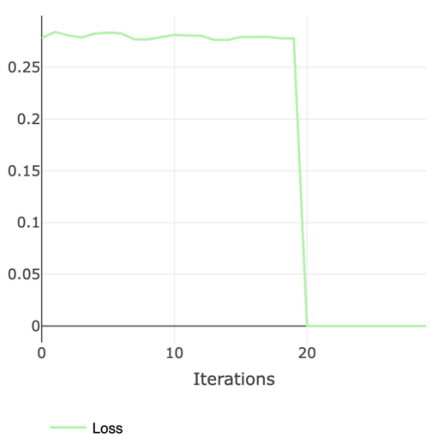
- Total reward: ~-4

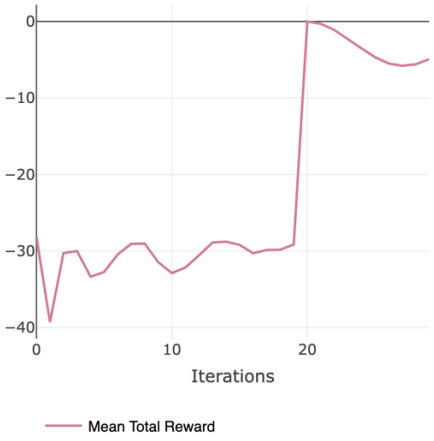
8 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/01a752836b98417aa6c55b1e3a7accbc/output/execution

- EPISODE_N: 30

- Learning rate: 0.01
- Q_param: 0.8
- Trajectory_n: 200
- EPS: 1
- Noise: Normal
- Loss: MSE + CLIP





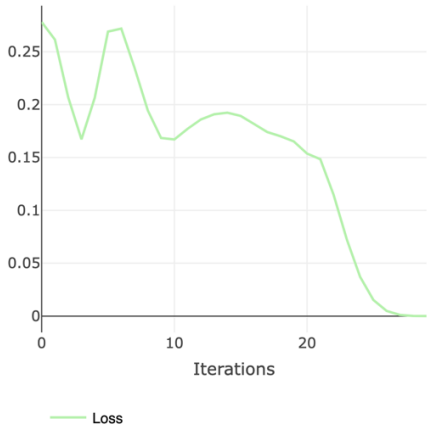
На тесте:

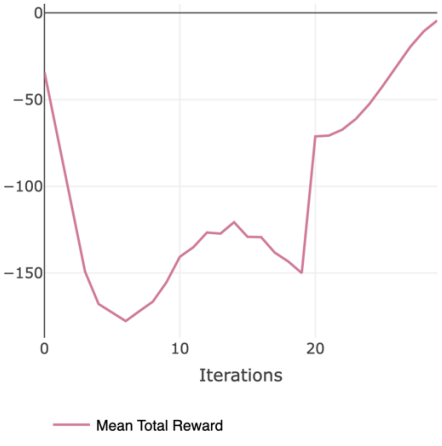
- Total reward: ~-4

9 Эксперимент:

https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/1ad97de1b54b42c5b947bb7bce60761c/output/execution

- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.8
- Trajectory_n: 200
- EPS: None
- Noise: Noise negative and positive
- Loss: MSE + Tanh



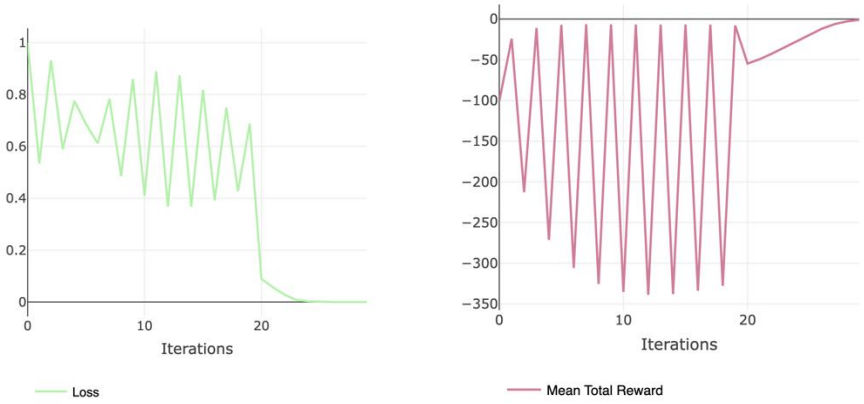


На тесте:

- Total reward: ~-1

10 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/10f1258a774f48cc93703856815c1384/output/executionv

- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.8
- Trajectory_n: 200
- EPS: None
- Noise: Noise negative and positive
- Loss: MSE + Tanh

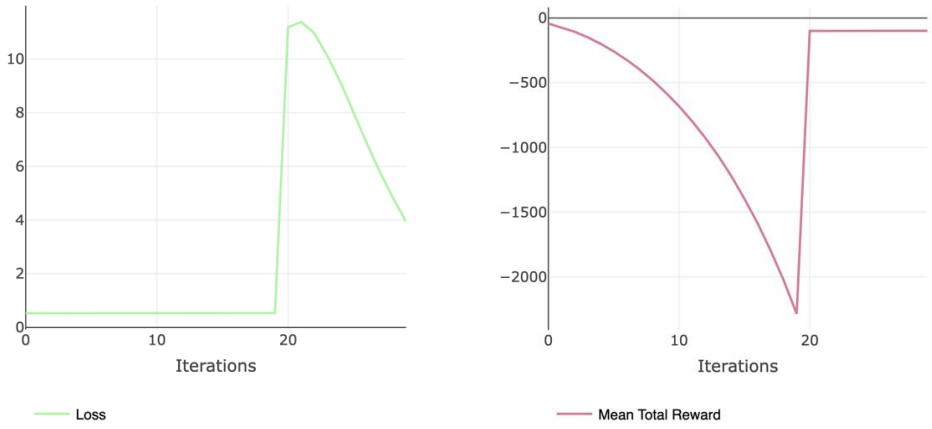


На тесте:

- Total reward: ~-0.5

11 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/e023f182e9f1411586c1edd473335b4b/output/execution

- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.8
- Trajectory_n: 200
- EPS: None
- Noise: Noise negative and positive
- Loss: MSE + Tanh

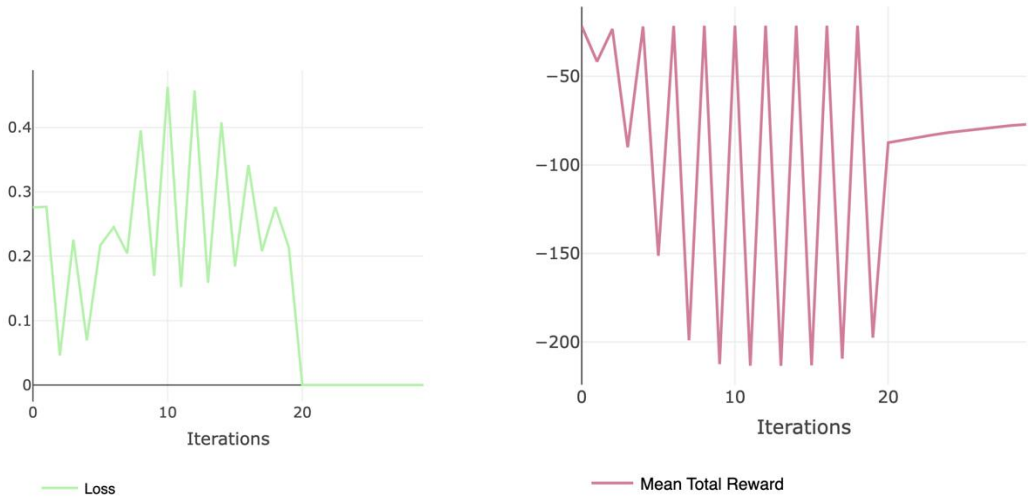


На тесте:

- Total reward: ~-98.5

12 Эксперимент:
https://app.clear.ml/projects/d9268489baca4b9e86e6a4d09878b798/experiments/366b25cf1b6646049a0b2bd143b87251/output/execution

- EPISODE_N: 30
- Learning rate: 0.01
- Q_param: 0.8
- Trajectory_n: 200
- EPS: None
- Noise: Noise negative and positive
- Loss: MSE + Tanh



На тесте:

- Total reward: ~-77.5

Вывод:
Лучшие результаты были достигнуты при следующих параметрах:

- EPISODE_N: 100
- Learning rate: 0.001
- Q_param: 0.2
- Trajectory_n: 500
- EPS: 0.5
- Noise: Normal
- Loss: MSE

Агент не может обучиться доезжать до финиша. Я это связываю с тем, что получить максимальное значения reward (100) в этой задачи дело случая и агент выгоднее не двигать машинку, так как reward при малых значениях action становится малым и близким к нулю.

Rewards

Since the goal is to keep the pole upright for as long as possible, a reward of `+1` for every step taken, including the termination step, is allotted. The threshold for rewards is 500 for v1 and 200 for v0.

Предположу, что можно применить какой-то другой шум на начальных итерациях обучения или увеличивать кол-во сэмплируемых траекторий (Trajectory_n), чтобы увеличить вероятность получения максимального reward’а.