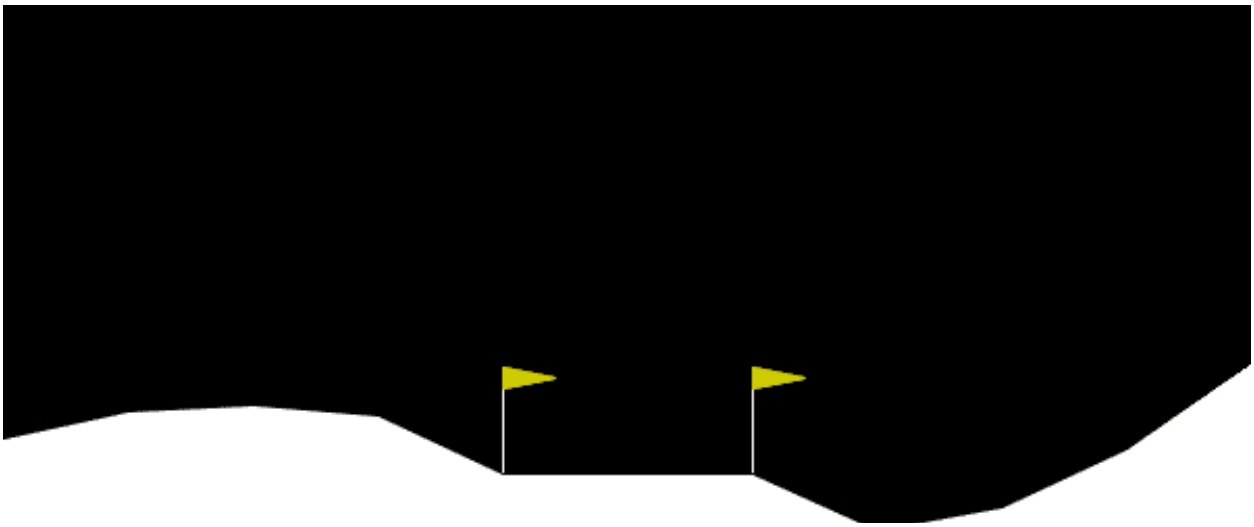


Решение домашнего задания 2

Задание 1

В качестве задания 1 я выбрал LunarLander-v2.

Анализ задачи

Описание среды

Action Space	Discrete(4)
Observation Shape	(8,)

Награды

Успешная посадка дает 100-140 очков.

Модуль разбит	-100
Модуль сел	+100
Нога коснулась земли	+10
Использование центрального двигателя	-0.3
Использование бокового двигателя	-0.03

Задание выполнено	+200

Завершение взаимодействия

Эпизод завершается при условии:

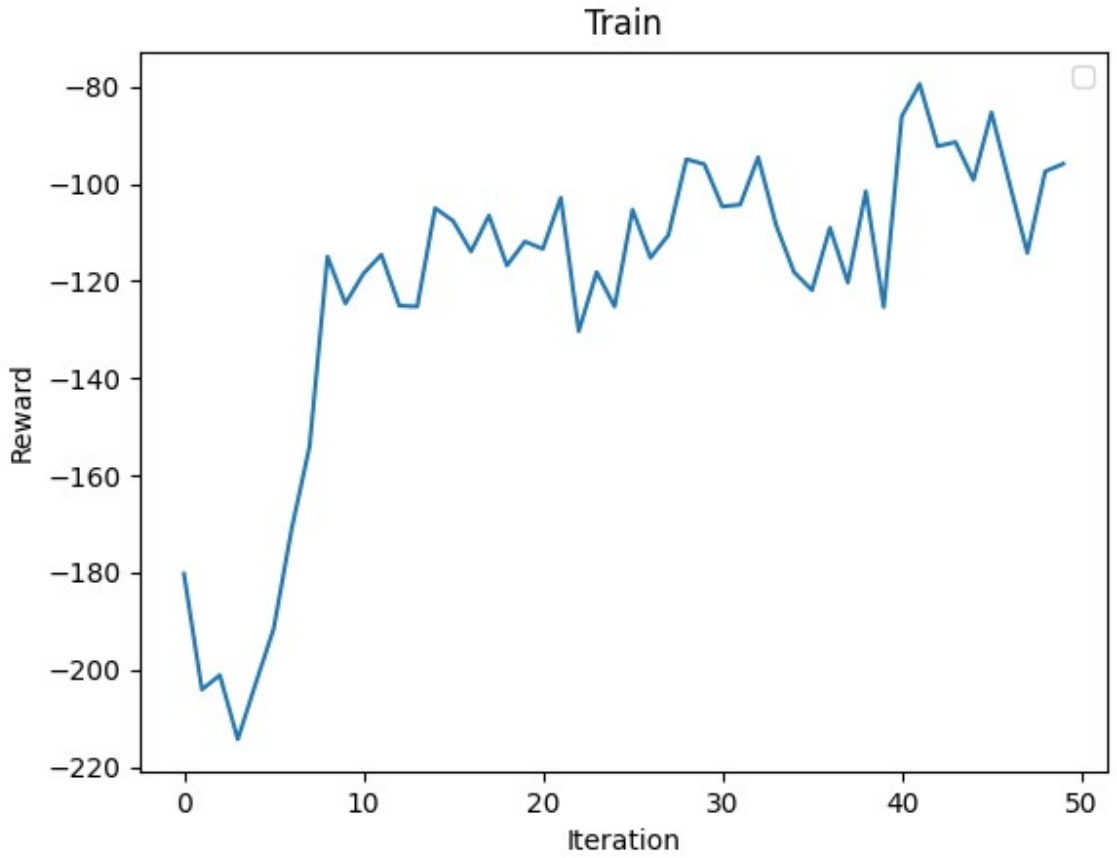
1. Модуль разбит
2. Модуль вышел за границы зоны видимости
3. Модуль не отвечает

Решение

Протестируем алгоритм из практического занятия со следующими гиперпараметрами:

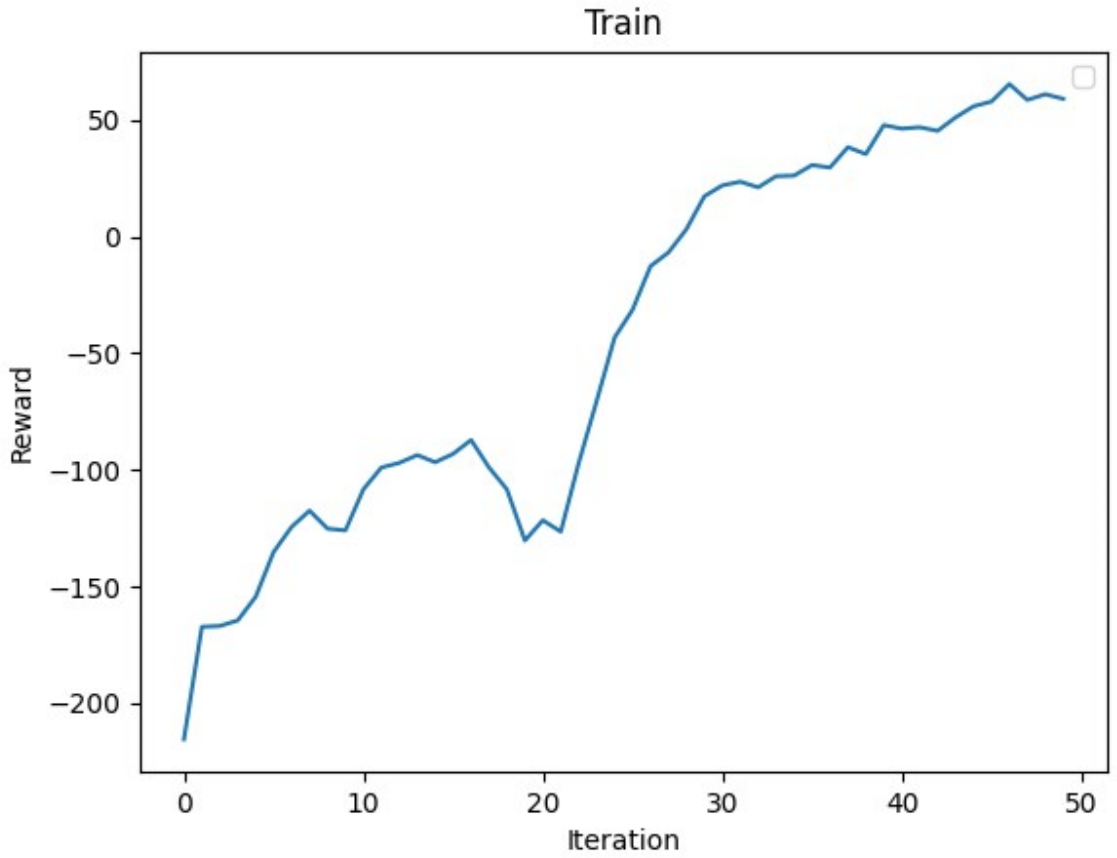
```
episode_n = 50
trajectory_n = 20
trajectory_len = 500
q_param = 0.8
```

Получаем следующую кривую обучения:

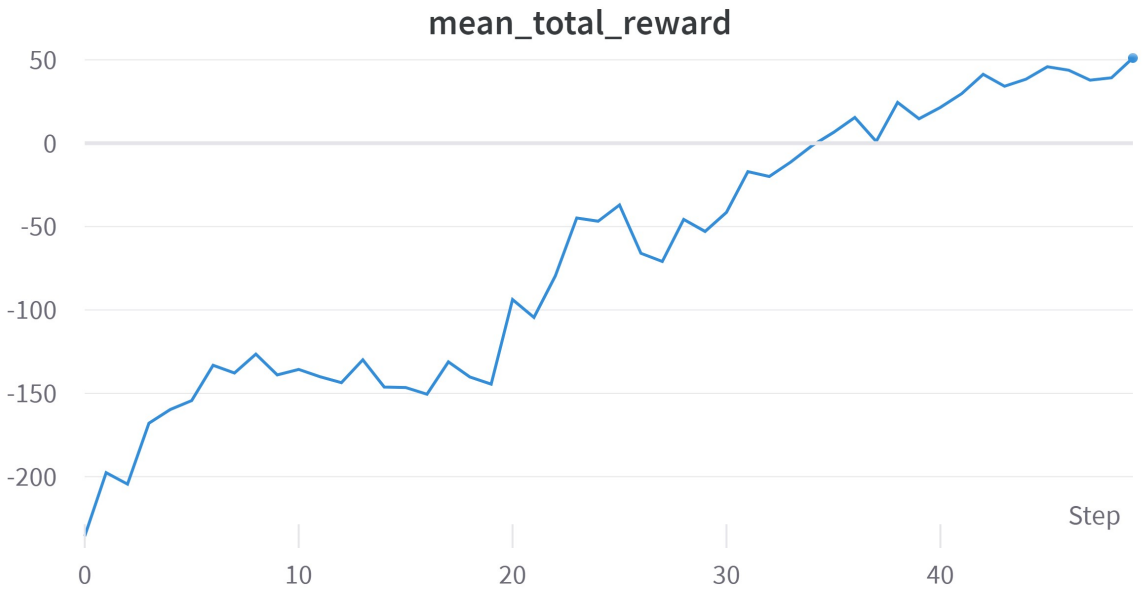


По кривой видно, что сходимость не плавная. Можно предположить, что это происходит из-за малого количества обучающих траекторий, т.е. обновление политики достаточно стохастическое. Увеличим количество траекторий до 200.

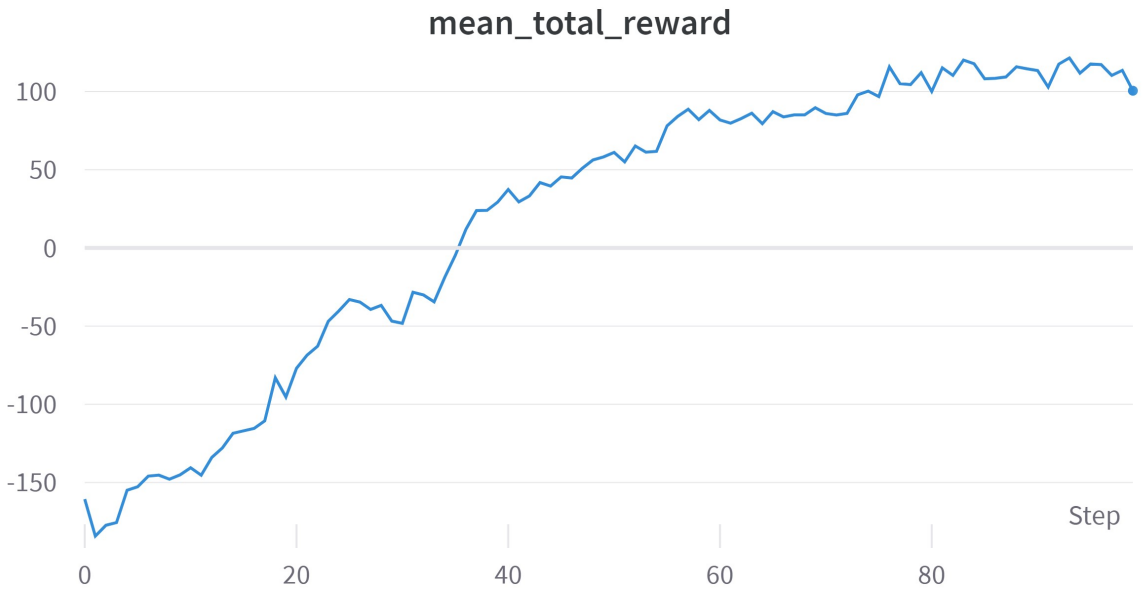
Увеличение количества траекторий дало следующий результат:



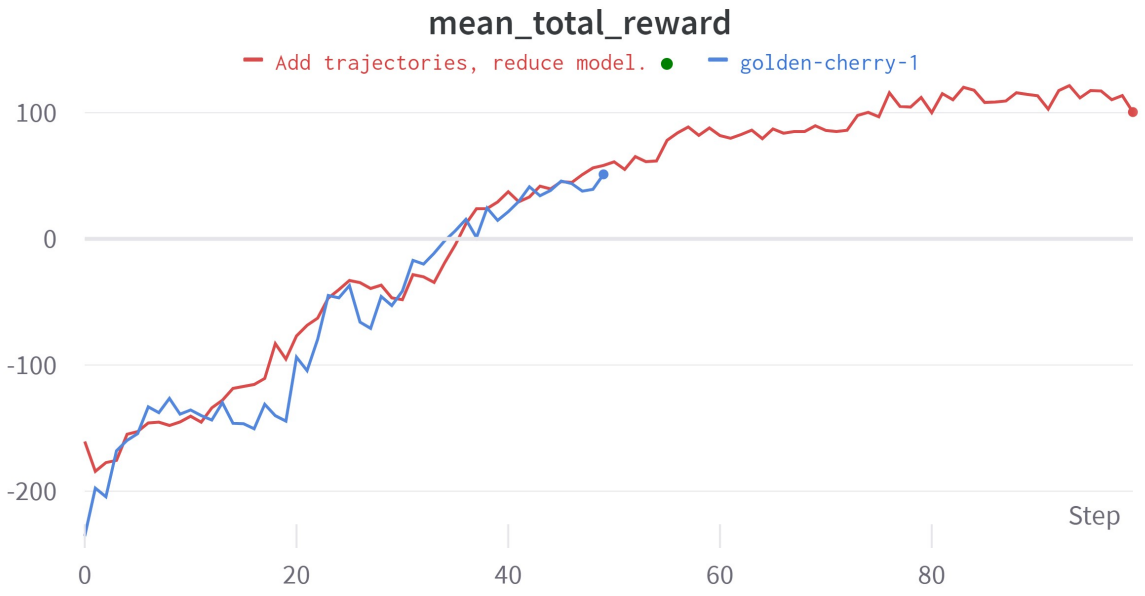
Обучение стало сходиться более плавно и качество улучшилось, но все равно заметны пики, значит сходимость еще не идеальная. Попробуем добавить шум, чтобы агент не переобучался на траектории на старте обучения, когда он еще ничего не знает о задаче.



Видно, что обучение хоть и идет не достаточно плавно, однако не замедляется. Также предположим, что сходимость прерывистая из за переобучения модели под конкретный батч траекторий. Исходя из всего вышесказанного попробуем увеличить количество итераций обучения и уменьшить модель.



Для сравнения отобразим оба графика обучения на одной координатной плоскости:



Видно, что агент действительно продолжает обучаться и достигает средней награды > 100. В документации к среде сказано, что успешное решение получает от 100 до 140 очков. Поэтому можно сказать, что с данными гиперпараметрами задача решена:

```
episode_n = 100
trajectory_n = 200
trajectory_len = 500
q_param = 0.8
```

Задание 2

В качестве второго задания я выбрал MountainCarContinuous-v0.

Анализ задачи

Описание среды

Action Space	Box(-1.0, 1.0, (1,), float32)
Observation Shape	(2,)
Observation High	[0.6 0.07]
Observation Low	[-1.2 -0.07]

Действие представляет собой одно действительное число в интервале от -1 до 1, умноженное на 0.0015.

Награды

До достижения вершины	-0.1*action^2
При достижении вершины	+100

Завершение взаимодействия

1. Машина достигла координаты 0.45 (вершины холма)
2. Превышено количество шагов (999)

Решение

Для модификации решения необходимо учесть следующие ограничения:

1. Действие представляет собой одно действительное число от -1 до 1
2. Максимальная длина траектории 999

В связи с этим модифицируем код агента следующим образом:

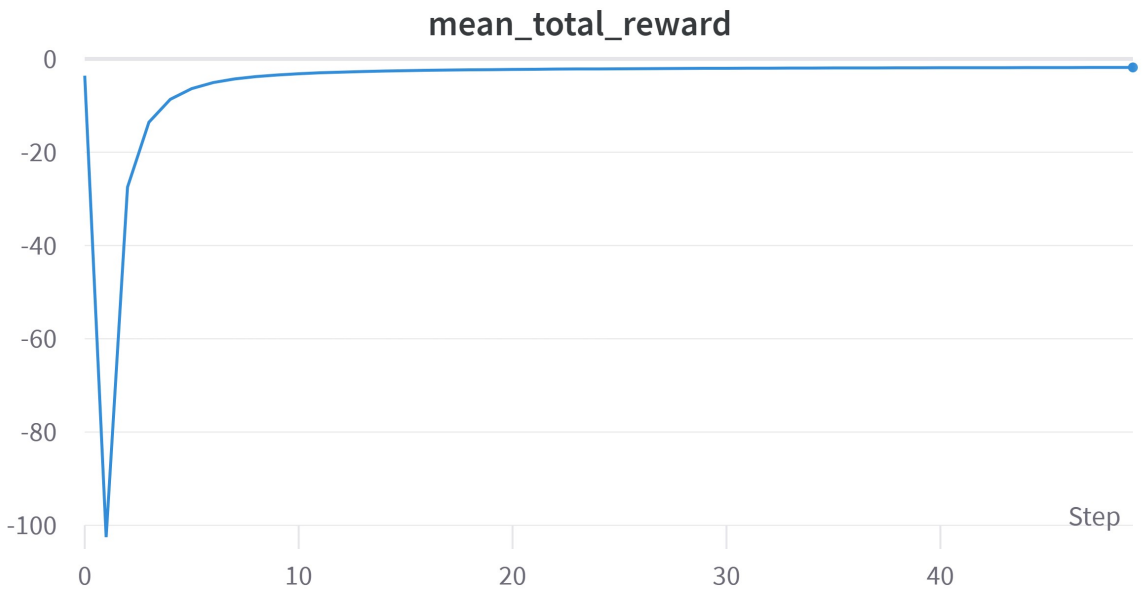
```
self.network = nn.Sequential(
  nn.Linear(self.state_dim, hidden_dim),
  nn.ReLU(),
  nn.Linear(hidden_dim, self.action_n),
)

self.activation = nn.Tanh()
self.optimizer = None
```

```
self.scheduler = None
self.loss = nn.MSELoss()
```

И зададим в качестве бейзлайна следующие гиперпараметры обучения:

```
episode_n = 500
hidden_dim = 32
optim: SGD
trajectory_n = 100
trajectory_len = 1000
q_param = 0.8
```

В результате получается следующая кривая изменения суммарной награды:



Исходя из алгоритма получения награды можно сделать вывод, что агент обучается стоять на месте или совершать малые телодвижения, так как награда штрафует его за большие шаги. Также видно, что агент ни разу не достиг вершины или число успехов гораздо меньше числа провалов, так как иначе в элитных траекториях появились бы действия, приводящие к достижению вершины, а такие действия должны иметь большую амплитуду, т.е. автомобиль должен хоть иногда иметь большую скорость. А исходя из того, что алгоритм быстро сошелся к нулевой награде и перестал обучаться говорит о том, что агент не понял, что

нужно балансировать между низкой и высокой скоростью и принял стратегию стояния на месте.

Нужно сделать так, чтобы агент определенное количество раз достиг вершины, чтобы таких траекторий было достаточно для запуска “направления мысли” агента в нужном направления. Для этого можно попробовать увеличить количество исследуемых траекторий, усилить exploration. Также можно уменьшить модель, так как большая модель может переобучиться под стояние на месте, а маленькая способна решить задачу качественно, потому что задача простая.

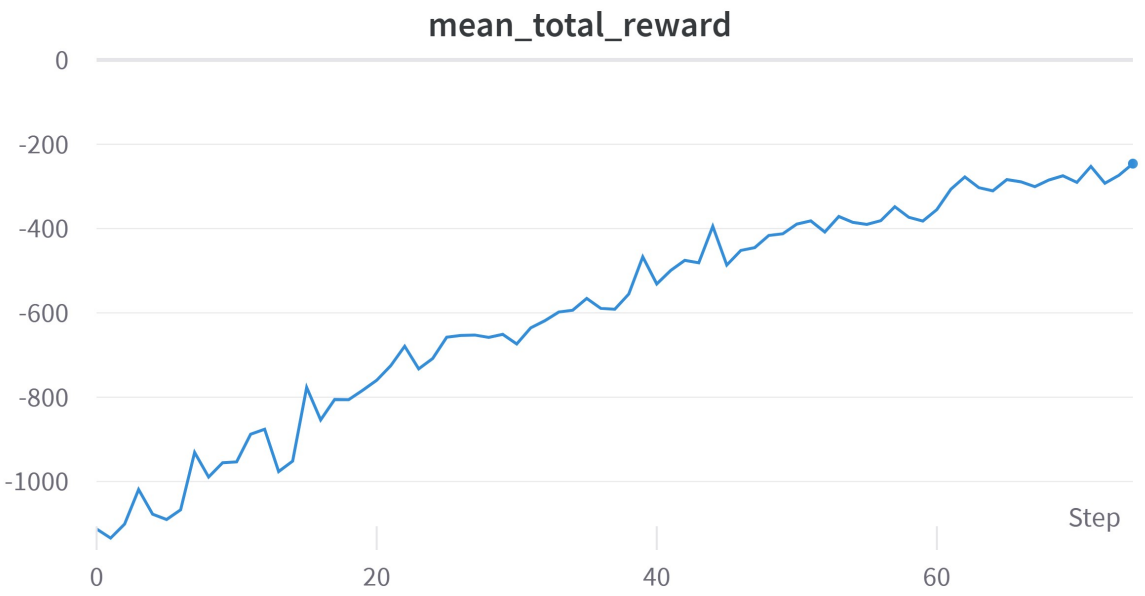
Также, исходя из результатов экспериментов коллег, можно внести следующие модификации:

- использовать MAE в качестве функции потерь
- переводить выходное значение в отрезок [1, 1] только при передаче действия в среду, а при обучении считать потери на сети без активации
- взять максимальную длину траектории 5000 (интуиция не понятна, т.к. максимальная длина траектории по условиям среды 1000, но в качестве эксперимента можно попробовать)

Берем следующие гиперпараметры:

```
episode_n = 500
hidden_dim = 128
optim: SGD
trajectory_n = 1000
trajectory_len = 5000
q_param = 0.8
```

При таких параметрах модель сходится стабильнее:



Однако так модель обучается очень долго. Темп сходимости не падает, поэтому можно предположить, что с такими параметрами модель сойдется довольно близко к оптимуму.