Домашнее задание 2

Пьянков Денис

I. Пользуясь алгоритмом Кросс-Энтропии для конечного пространства действий обучить агента решать LunarLander-v2. Исследовать гиперпараметры алгоритма.

1. **Введение**

   В данном отчете рассматривается обучение агента для решения задачи **LunarLander-v2** с помощью алгоритма Кросс-Энтропии. LunarLander-v2 — это среда симуляции посадки космического аппарата, в которой агент должен контролировать ракетный двигатель, чтобы безопасно приземлиться на платформу.

2. Описание экспериментов

   Алгоритм Кросс-Энтропии

   Алгоритм основан на выборке множества траекторий, выполненных агентом, и отборе лучших из них (элитных траекторий) на основе квантили общего вознаграждения. Агент обновляет свою политику, обучаясь на этих элитных траекториях. Процесс повторяется несколько итераций до достижения определенного уровня производительности.

   Гиперпараметры

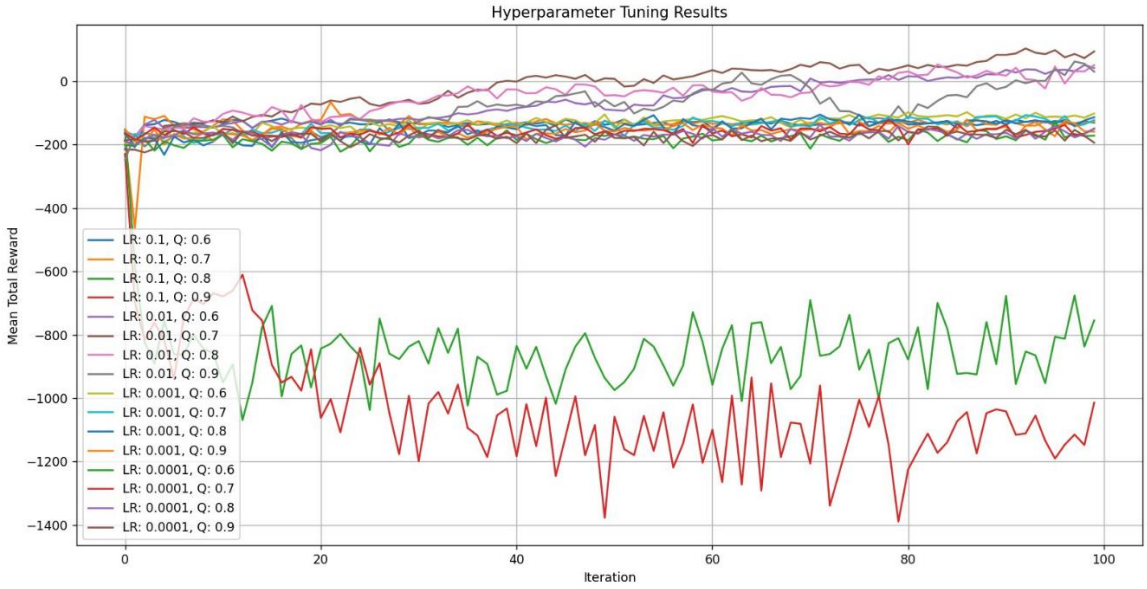
   Исследуемые гиперпараметры аналогичны тем, которые были использованы в предыдущем дз (iteration_n, trajectory_n, q_param).

   Новый гиперпараметр:

   Скорость обучения (lr): Скорость, с которой агент обновляет свои параметры в процессе обучения. Этот гиперпараметр определяет, насколько сильно изменяется политика агента на каждой итерации. Если скорость обучения слишком низкая, обучение будет слишком медленным. Если скорость обучения слишком высокая, агент может не успевать адаптироваться к изменениям в среде.

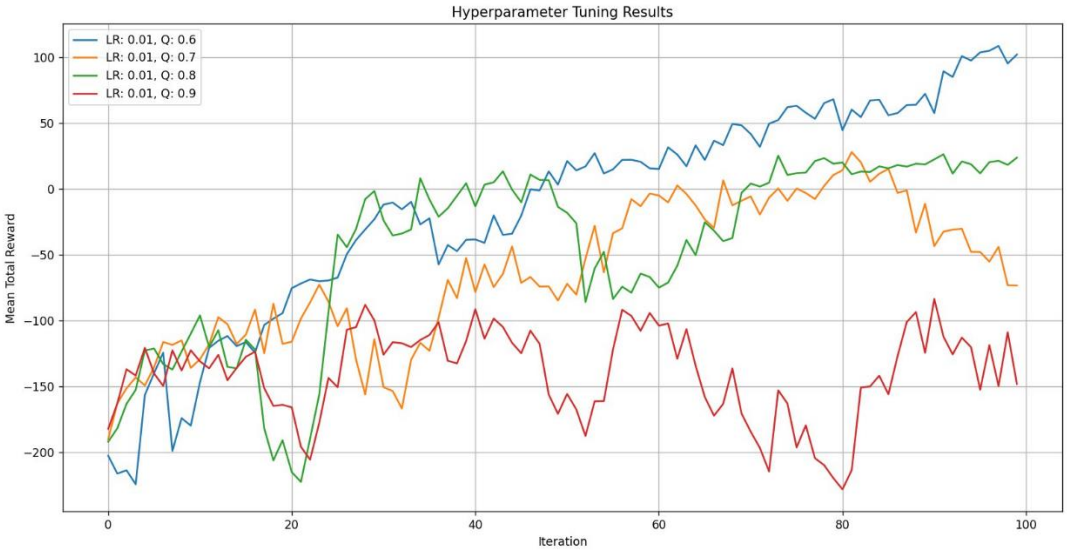
3. Результаты

В ходе экспериментов было протестировано несколько комбинаций гиперпараметров.

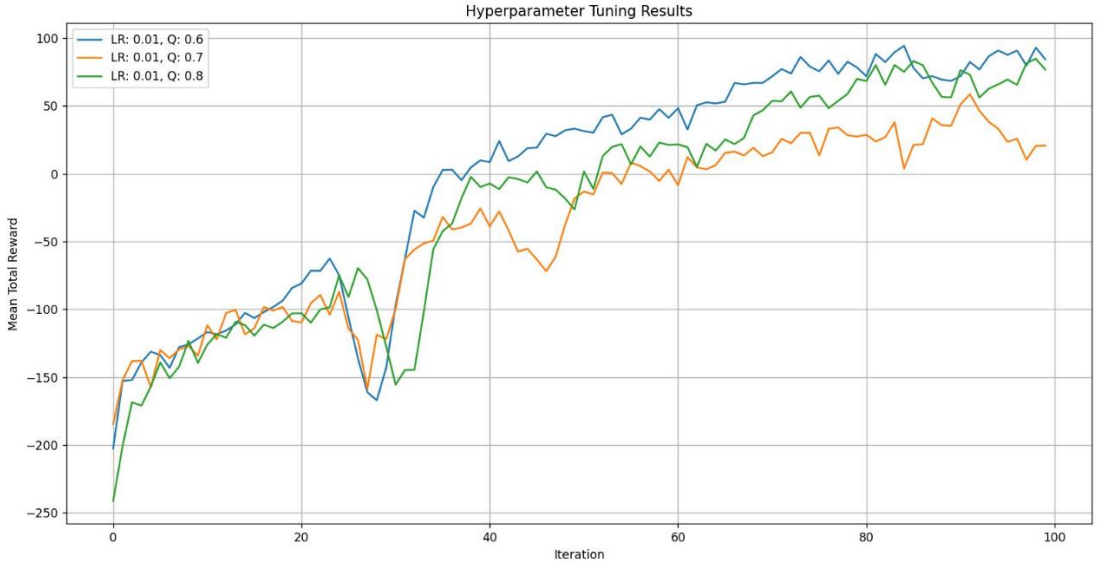


Для каждого набора гиперпараметров проводилось по 100 итераций обучения и 50 траекторий, и результаты сравнивались по средней сумме награды, полученной агентами за каждую итерацию.

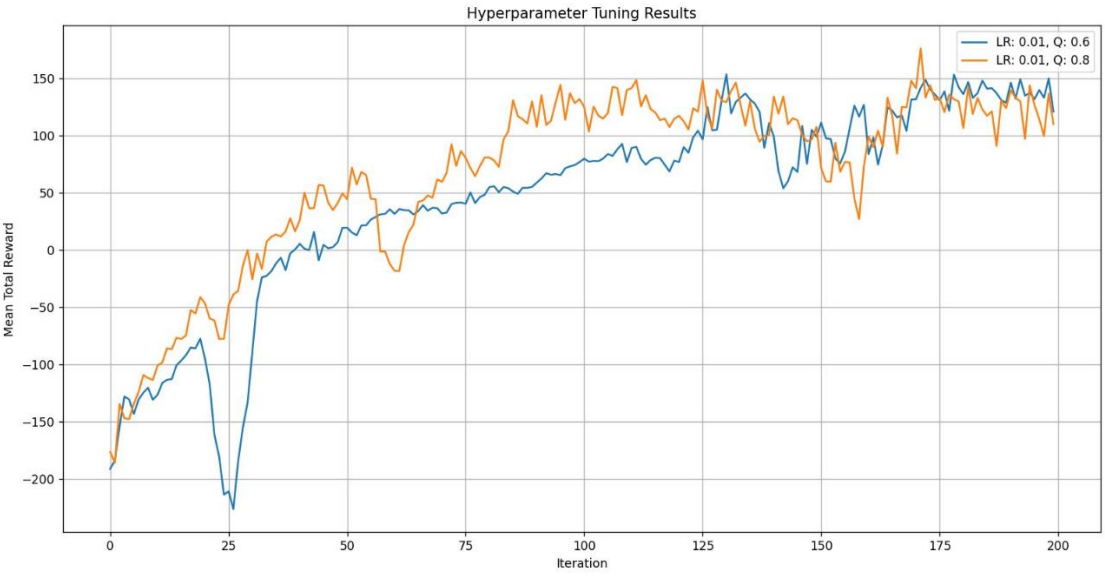
Скорость обучения (lr = 0.01): Показала лучшие результаты среди всех протестированных значений скорости обучения.



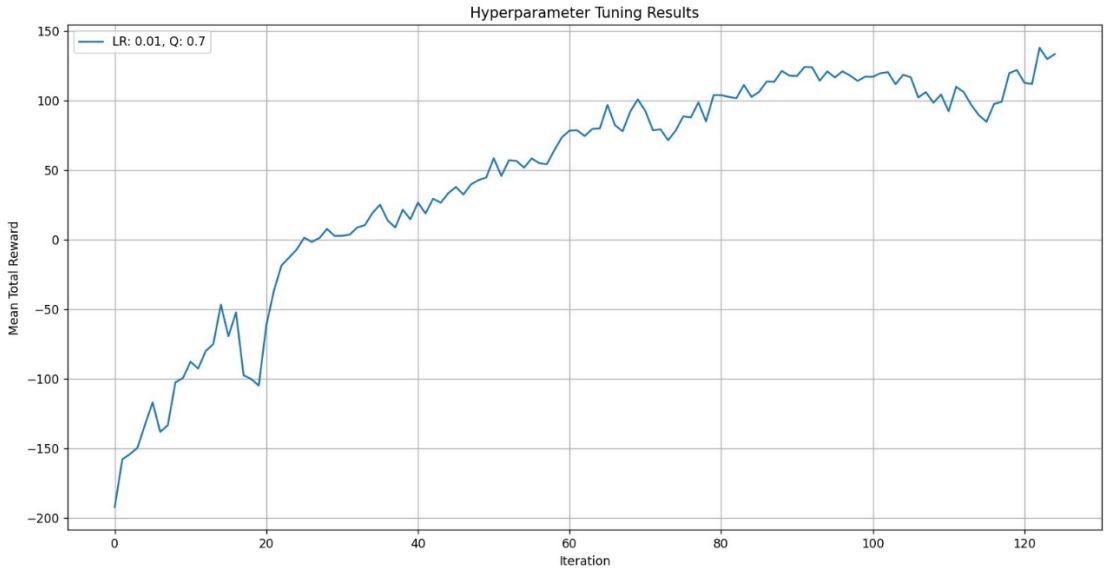
Значения **q = 0.6, 0.7, 0.8** показали хорошие результаты, при количестве итераций **iteration_n = 100** и количестве траекторий **trajectory_n = 50**.



При **iteration_n = 100** и **trajectory_n = 100** обучение стало более стабильным.



iteration_n=200, trajectory_n=100



iteration_n=125, trajectory_n=150

4. Вывод

По результатам экспериментов можно сделать следующие выводы:

Скорость обучения (lr = 0.01) показала наилучшие результаты, обеспечив стабильное и быстрое обучение агента. Дальнейшее снижение или увеличение значения скорости обучения дает ухудшения.

Значение квантили (q = 0.6, 0.7, 0.8) оказались оптимальными, обеспечивая как достаточно быстрый прогресс в обучении, так и стабильные результаты.

Количество итераций (100 <= iteration_n <= 200) и **количество траекторий (100 <= trajectory_n <= 200)** были оптимальны для стабильного обучения. Увеличение этих значений до ~150-200 привело к улучшению результатов, но также к значительному увеличению времени обучения.

II. **Реализовать алгоритм Кросс-Энтропии для непрерывного пространства действий. Обучить агента решать MountainCarContinuous-v0.**

В данной задаче агент управляет непрерывным пространством действий, где он должен генерировать плавные действия для управления ускорением вагонетки. Действия представляют собой одно число, которое определяет силу движения вагонетки.

Для обучения агента используется среднеквадратичная ошибка (MSE), так как действия непрерывны и цель — приблизить выход нейросети к элитным действиям.

Из-за непрерывного пространства действий требуется добавить шум для исследовательских действий агента. Это делается с использованием параметра ε (эпсилон), который контролирует степень шума. Для успешного обучения требуется регулировать шум, чтобы агент мог преодолеть начальные негативные награды, исследуя больше состояний.

В этой среде агент часто получает отрицательные награды, особенно если не может подняться на горку. Поэтому важно контролировать параметры обучения, чтобы агент не был мотивирован останавливаться в низших состояниях.

Результаты и вывод:
В процессе обучения агента на основе алгоритма кросс-энтропии (CEM) в среде **MountainCarContinuous-v0** было проведено множество экспериментов, и результаты показали важные аспекты влияния параметров обучения, таких как шум и исследование среды, на эффективность решения задачи.

При установке низкого уровня шума и уменьшении параметров исследования среды, вагонетка не могла подняться достаточно

высоко для преодоления финишной черты. Это явление связано с тем, что при недостаточной исследовательской активности агент застревал в локальных минимумах, пытаясь максимизировать награду. В частности, он пытался оставаться в точке с наименьшей наградой, то есть в точке, близкой к нулю слева. Это приводило к тому, что вагонетка фактически не двигалась, пытаясь оптимизировать своё поведение в условиях жестких ограничений.

В данном случае недостаток исследовательских действий вынуждал агента сосредоточиться на том, чтобы избегать отрицательных наград, что в итоге и стало причиной его бездействия. Таким образом, модель не только не достигала цели, но и демонстрировала откат к неэффективным стратегиям, из-за чего обучение не давало желаемых результатов.

С учетом вышеописанных проблем становится очевидным, что для повышения исследовательской активности агента необходимо увеличить уровень шума и адаптировать другие параметры, способствующие более широкому исследованию пространства действий. Увеличение шума позволит вагонетке исследовать больше состояний, что может привести к нахождению более оптимальных действий, необходимых для достижения финиша.

Тем не менее, данная стратегия имеет свои недостатки. Повышение уровня шума может существенно увеличить время, затрачиваемое на обучение и эксперименты, так как агент будет генерировать больше неэффективных действий, прежде чем обнаружит приемлемые стратегии для достижения цели. Увеличение исследовательской активности требует большего количества итераций и, как следствие, увеличивает общее время тренировки.

После проведенных экспериментов можно сделать однозначный вывод: в рамках алгоритма CEM значительно затруднено получение оптимальной стратегии решения задачи в среде MountainCarContinuous-v0. Даже с увеличением уровня шума и корректировкой параметров, эффективность решения задачи оставалась на низком уровне. Это указывает на необходимость пересмотра выбранного алгоритма.

Рекомендуется рассмотреть использование других методов/алгоритмов/модификаций обучения с подкреплением.