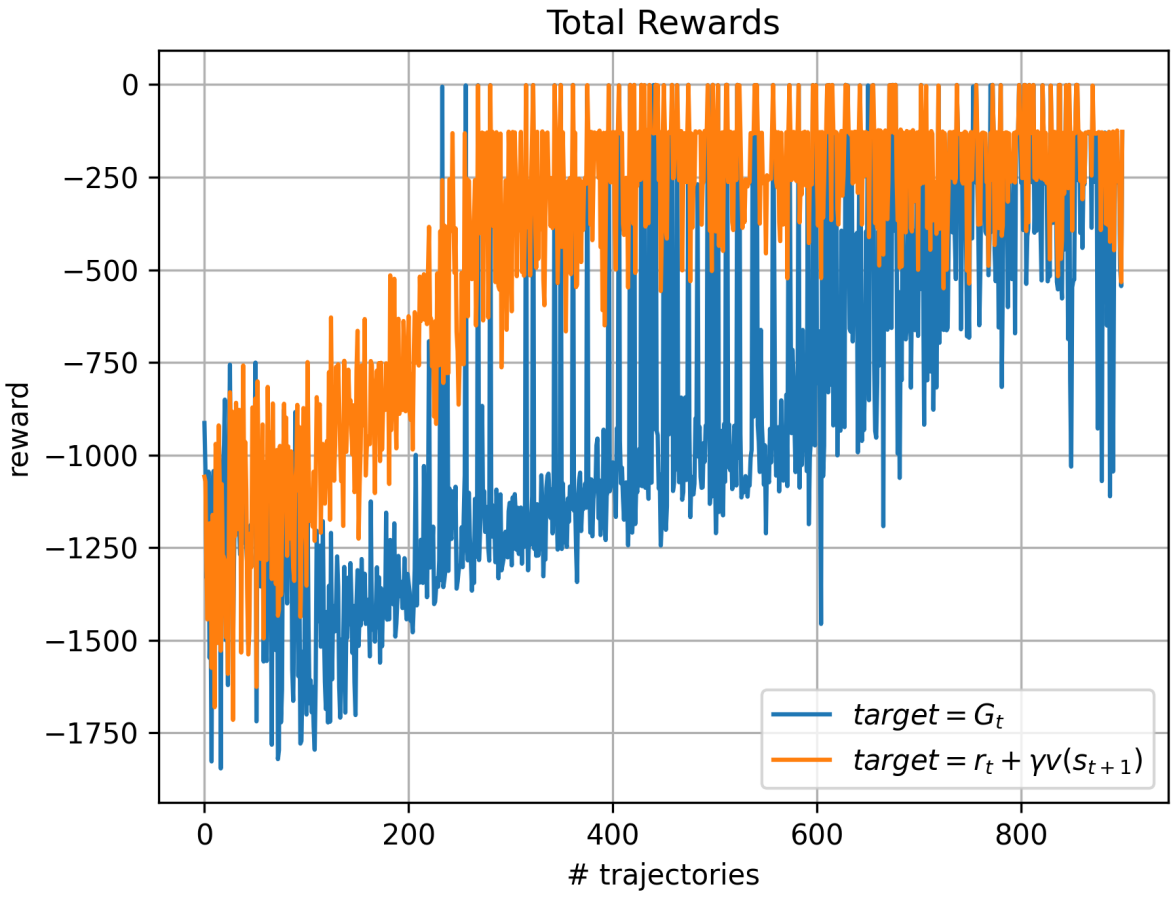
1.

В рамках первого задания был реализован метод PPO с использование Advantage функции следующего вида:

$$A = R_t + \gamma v(s_{t+1}) - v(s)$$

Вариант с использованием наград G оказался стабильнее и производительнее (обучение происходило быстрее). Я думаю это связано с теми же проблемами, которые возникали в прошлом дз и решались различными модификациями по типу Soft Target, Hard Target и DDQN.

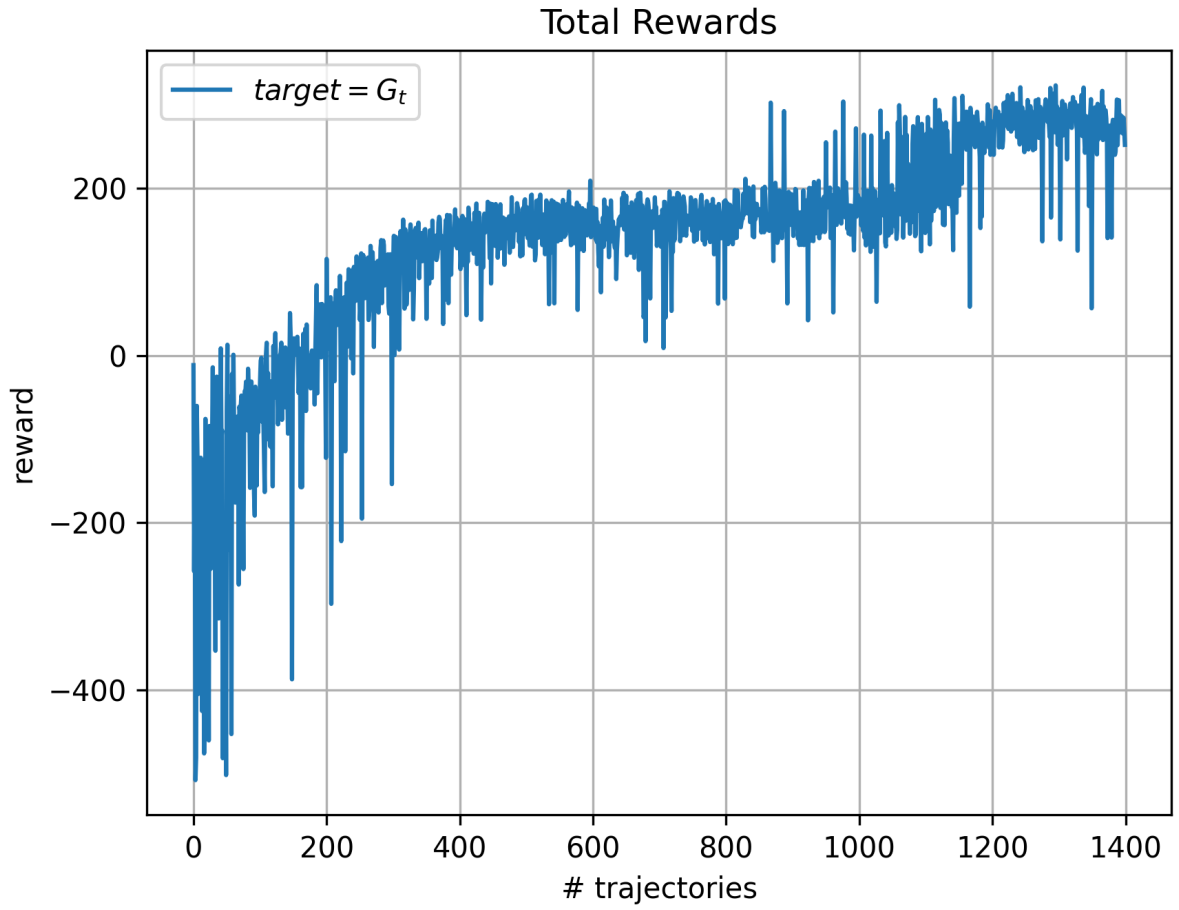
Ниже представлен график обучения этих двух алгоритмов при одинаковых параметрах. (Для варианта с G можно выбрать немного другие гипер параметры при которых скорость сходимости еще выше, однако второй алгоритм не сходится за сравнимый временной промежуток).



Однако обе этих реализации имеют общий недостаток. Он вызван тем, что мы применяем к предсказанным логарифмам дисперсий функцию активации тангенс гиперболический. Таким образом без каких либо модификаций минимальная величина дисперсии будет равно $e^{-1} \approx 0.37$. Эту проблему я решу выполняя второе задание, так как в данном случае для сравнения Advantage функций это не помеха.

2.

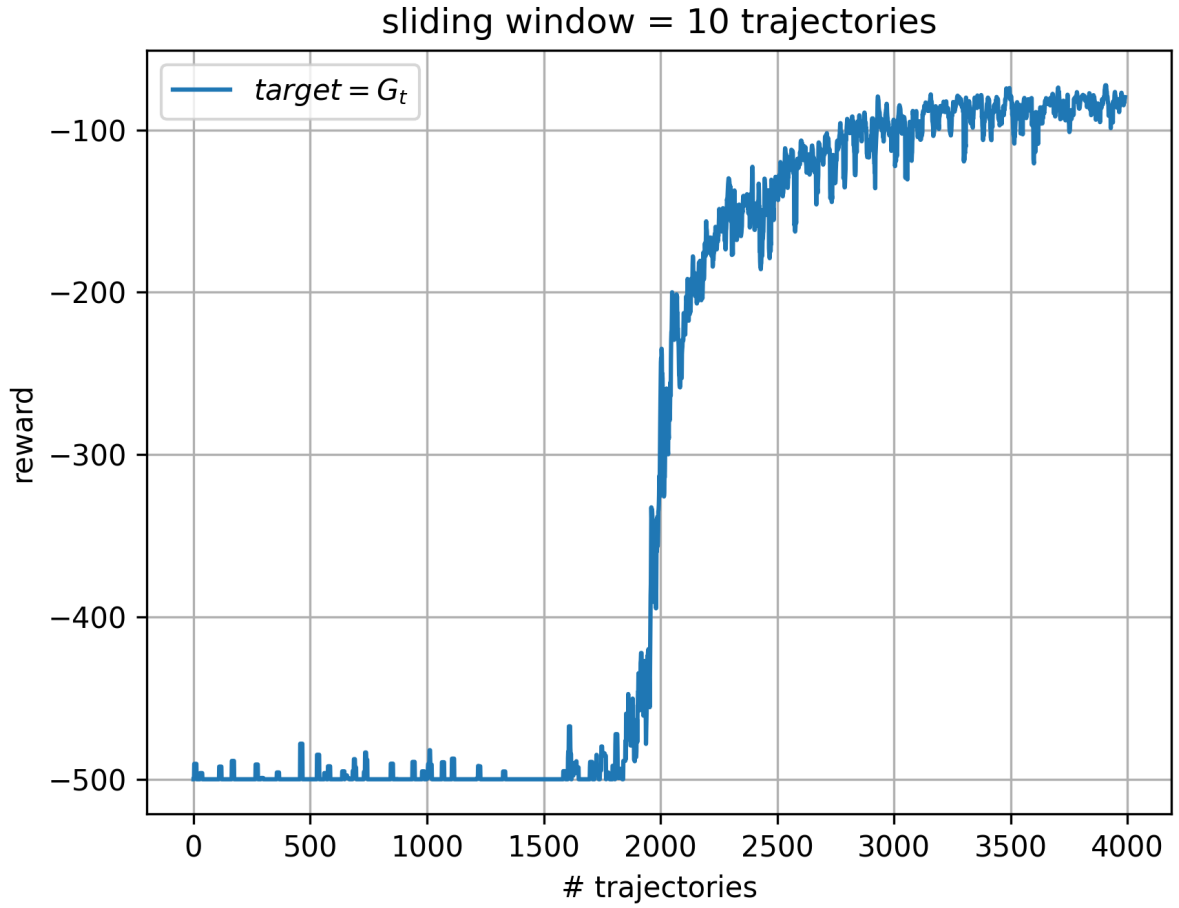
Во втором задании нужно было переписать алгоритм для возможности его использования в многомерном случае.
Я решил, что мы можем вместо одного последнего слоя с размерностью 2*action_dim создать два слоя с выходами размерностью action_dim для средних и логарифмов дисперсий соответственно.
Такой подход избавляет нас от необходимости лишний раз задумываться о размерностях (транспонировать результат работы модели и тд) и позволяет применять разные функции активации для предсказанных средних и логарифмов дисперсий. Для логарифмов дисперсий я оставил выход без какой-либо функции активации (я проверил, что из-за случайно инициализации весов в среднем логарифм дисперсии вначале равен 0, то есть на начальных этапах обучения в среднем мы принимаем решение с нормальным распределением с дисперсией равной 1, что меня устраивало).



P.S. После обучения на Lunar Lander я восхитился производительностью данного алгоритма, посчитав его лучшим среди ранее пройденных. Однако в 3 задании реализовав алгоритм для дискретного пространства действий я уже сомневаюсь, что он сильно быстрее решает задачу, чем другие подходы.

3.

В третьем задании было необходимо переписать алгоритм для сред с дискретным пространством действий. Для этого я просто заменил нормальное распределение на категориальное и убрал часть модели отвечающую за предсказание логарифмов дисперсий. И уже в этот момент понял, что возможно в данном случае из-за неудачной инициализации весов я не буду получать достаточно разнообразные траектории, что тем более важно для таких сред как Acrobot и Mountain Car где награда очень редкая и сложно достижимая. Поэтому я добавил возможность усреднять равномерное распределение и категориальное получение в результате работы модели. В итоге данный алгоритм показал хорошую и стабильную сходимость.



Епсилон отвечающая за исследование линейно менялась от 1 до 0, с 0 по 2500-ый эпизод.