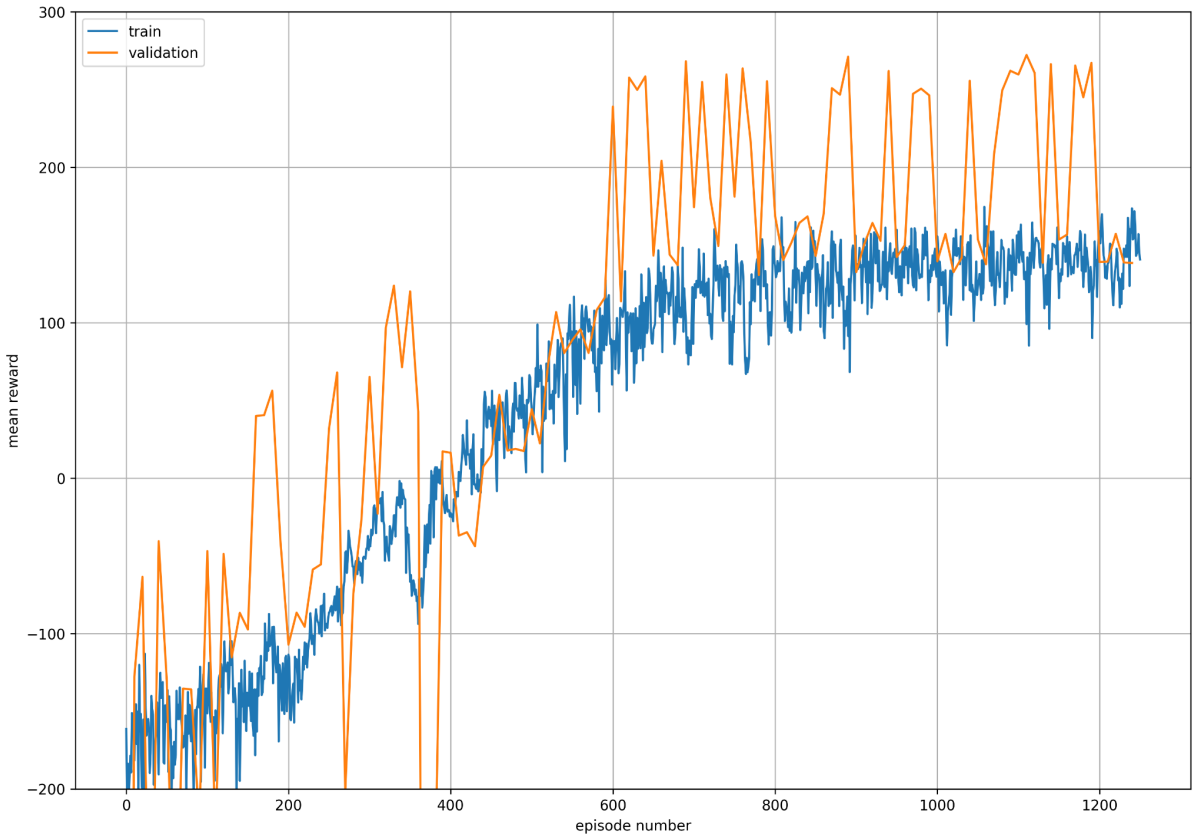
1. Для выполнения первого задания была выбрана среда LunarLander-v2. Данная среда имеет непрерывное пространство наблюдений и дискретное пространство действий. Для решения данной задачи предлагалось использовать метод cross-entropy с нейронной сетью в качестве функции определяющей политику.
   Нейросеть состояла из двух скрытых слоев по 32 нейрона каждый. В качестве функции активации я использовал ReLU.

   Исходя из описания среды, целью является достижения средней награды выше 200 очков. 140-180 из которых выдаются за успешное приземление на посадочную площадку и еще дополнительные 100, если после приземления LunaLender остается неподвижным некоторое время (одно из действий доступных в среде - ничего не делать).

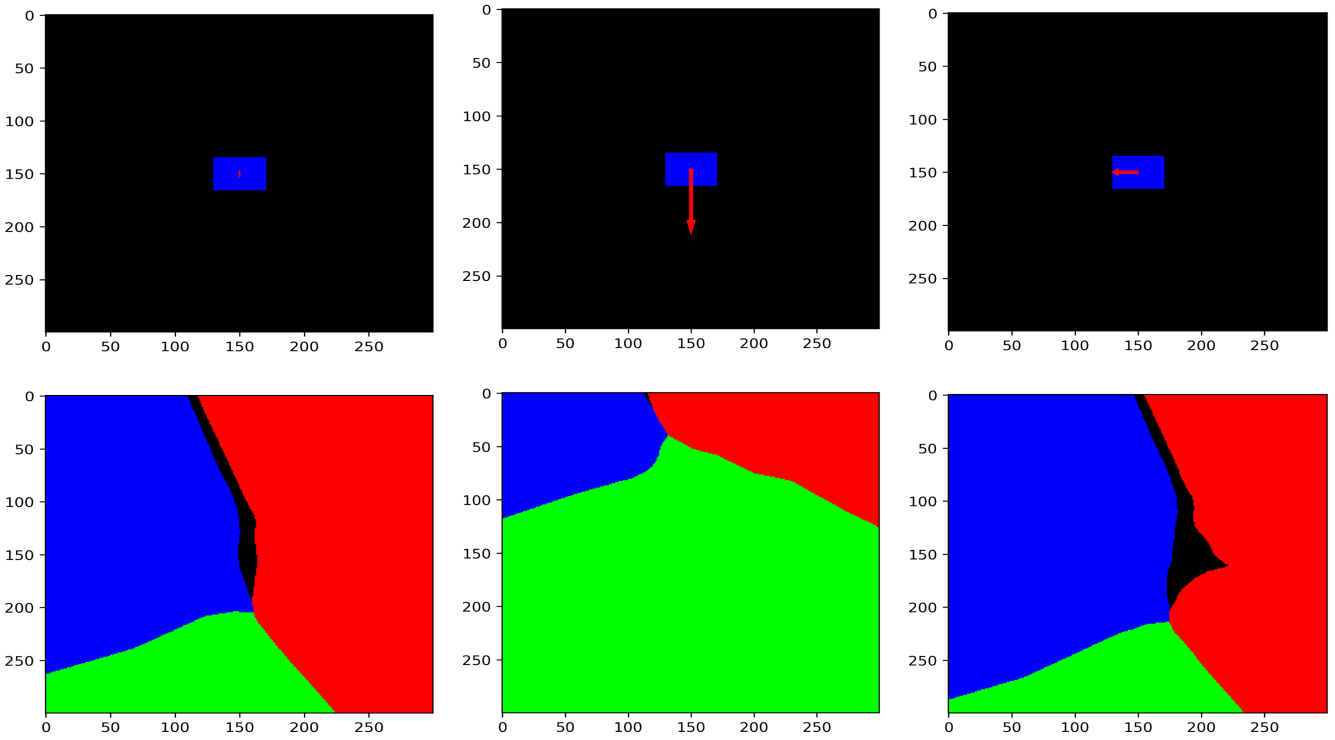
   Как и в первой домашней работе, основная задача состояла в подборе хороших гиперпараметров. Среди которых:
   a. Шум (параметр определяющий степень исследования) и функция по которой этот шум убывает с ростом количества эпох обучения.
   b. Параметр q
   c. Количество траекторий в одном эпизоде.
   В итоге q было взято равным 0.8 и количество траекторий в эпизоде 20. Обучение в таком случае происходило на 4 лучших траекториях на начальных этапах. При это я сохранял в отдельный буфер траектории с наградой свыше 200 (они были довольно редкими, по причине которую опишу позже). И из этого буфера каждый эпизод в список лучших 4х траекторий добавлялась одна с наградой свыше 200 очков. Но даже такое обучение не позволяло получать среднюю награду выше 200 очков.

   Я считаю, что это было вызвано тем, что из-за случайного выбора действий (хоть и распределение этого выбора не было равномерным) было довольно мало траекторий, где агент после приземления выбирал несколько раз подряд действие отвечающее за бездействие. Такие траектории при обучении попадались (из них и состоял дополнительный буфер). Но в этой траектории, помимо последней успешной последовательности бездействия на площадке, содержались и другие действия в том же состоянии, которые портили тем самым обучающую выборку.
   Поскольку отдельно от целых траекторий отбор действий этим методом обучения не предусмотрен, мной было решено отказаться от случайного выбора действий на этапе валидации, что дало ожидаемый результат. Теперь модель определенно знала, что ей надо делать после приземления, и если это действие было бездействием, то оно повторялось до тех пор пока эпизод не будет считаться оконченным (эпизод заканчивается, если агент получил 100 очков за бездействие). Однако это не решает эту проблему на этапе обучения и лишь позволяет выхватывать удачные модели на этапе валидации.



(Возможно более долгая выборка и ограничение длины траектории, могли бы решить эту проблему. Но достигнутый результат вполне отвечает поставленным целям).

Также для общего понимания мною были построены карты поведения агента в зависимости от его координат и скоростей. Ниже представлены некоторые из них.

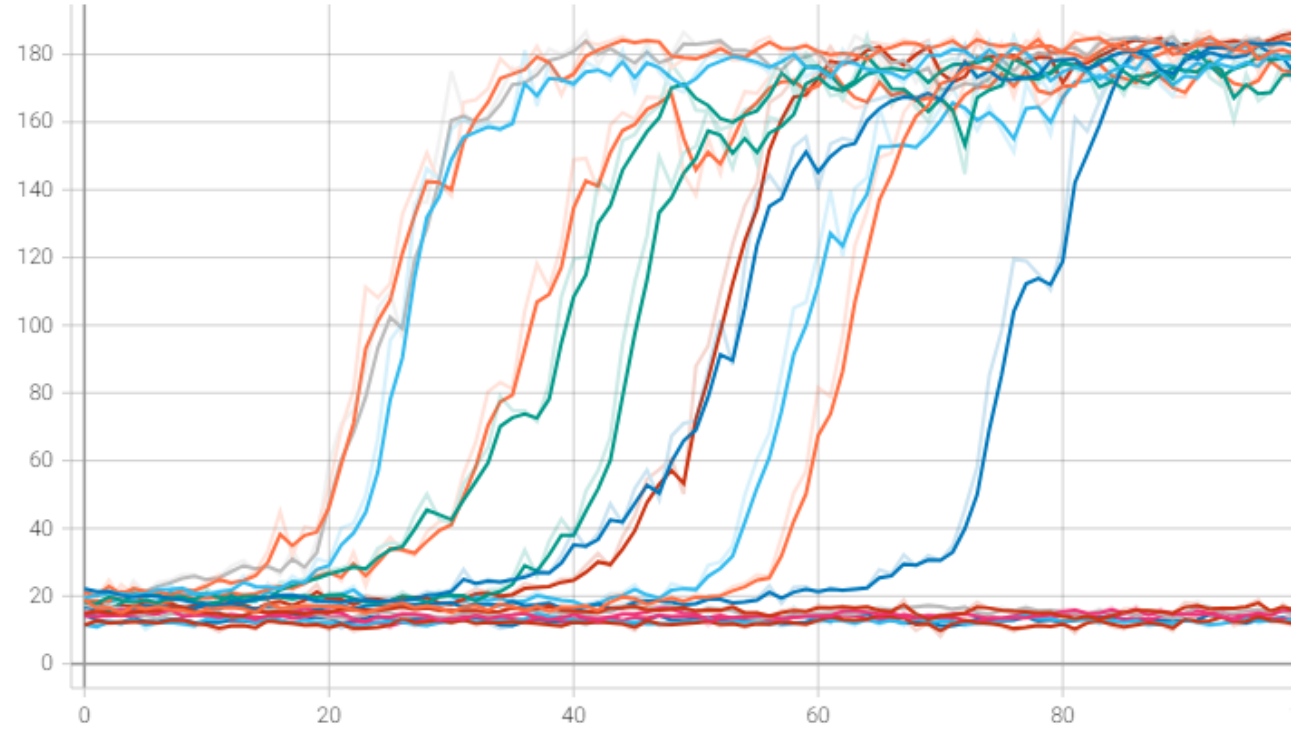


2. Во второй задаче была выбрана среда MountainCarContinuous-v0. В данном случае пространство действий тоже непрерывно, но это не было главной сложностью данного задания.

   Награда в данной среде устроена таким образом, что мы начинаем из локального оптимума (бездействие), поскольку получаем штраф за любые действия и награду лишь за решение задачи, которая требует большого количества правильно сделанных последовательных действий. Метод cross-entropy в данном случае требует большого количества траекторий в эпизоде, и высокого значения параметра q.

   Чтобы оценить частоту появления удачных траекторий, и тем самым подобрать гиперпараметры я смоделировал полностью случайного агента (вне среды, а используя уравнения представленные в документации gym, для ускорения получения результата).
   В итоге я получил, что бездействующий агент (нет детерминированного слагаемого) с нормальным центрированным шумом при дисперсии равной 1 достигает цели в 1035 из 60000 запусков, что примерно равно 1.725%. Затем, я решил посмотреть, как изменится вероятность, если инициализация начальных весов привела к появлению на выходе отрицательного множителя -1 от позиции агента (f = -1*(pos+0.5)). В таком случае количество удачных траекторий с 1.7 процента падает до 0.04%, в таком случае нам нужно иметь выборку из 2500 траекторий и q = 0.9996, что меня не устраивало. Тогда было решено, что нужно просто надеяться на удачную инициализацию весов.

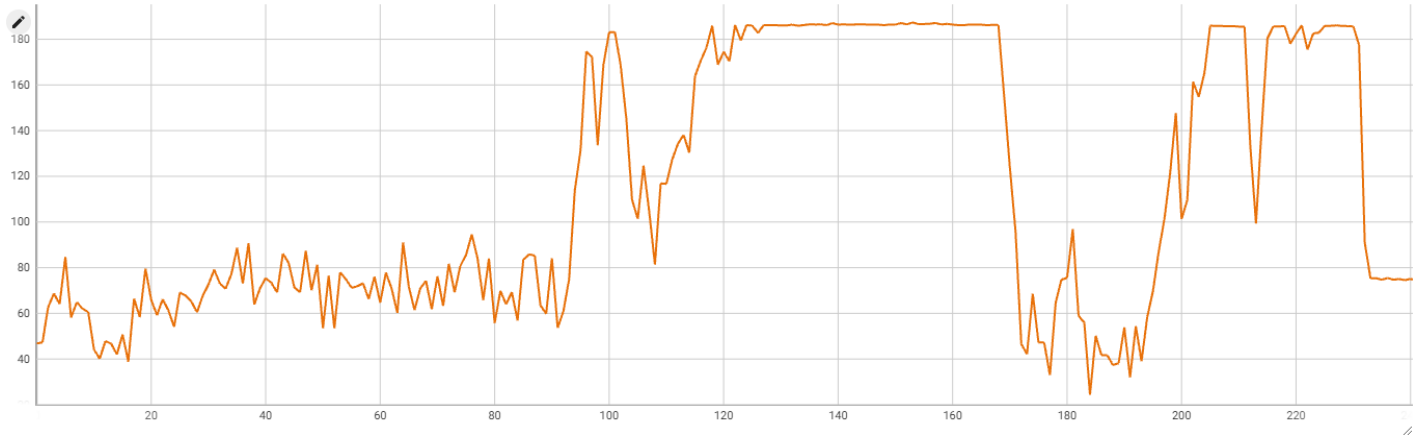
   Чтобы сократить время ожидания хороших начальных весов сети было принято решение использовать всего один скрытый слой с 10 нейронами. Также я производил отбор элитных траекторий не только по параметру q, но и проверял, чтобы их реварда превышал 50 очков.
   Данные подход дал свои результаты. Агентов с “хорошим” начальным набором весов удавалось обучать в среднем меньше чем за 5 минут. Для оценки количества “хороших” инициализаций я провел 18 запусков обучения, из них 10 сошлись к оптимуму меньше чем за 5 минут, остальные не проявляли никаких признаков к способности обучаться.



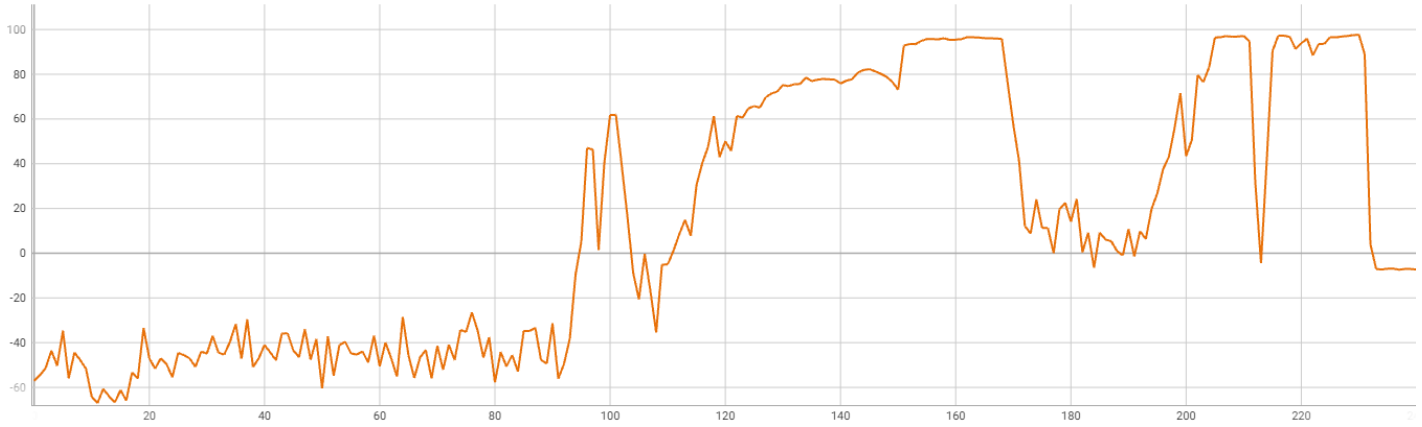
   В итоге q взял равным 0.92. Количество траекторий в эпизоде равнялось 80 и длину траекторий ограничил 400 шагами (чтобы в хороших траекториях содержалось как можно меньше неверных

действий). Поскольку я не уменьшал шаг обучения, на поздних этапах это приводило к падению средней награды (вплоть до отрицательных значений), но это вызвано тем, что из-за штрафа за действия в точке близкой к финишу модели выгодно двигаться по инерции, что приводило из-за большого шага градиентного спуска к отрицательным действиям в этой области и агент после одной эпохе переставал учиться (из-за порога в 50 очков для попадания в элитные траектории и низкой степени исследования под конец обучения).

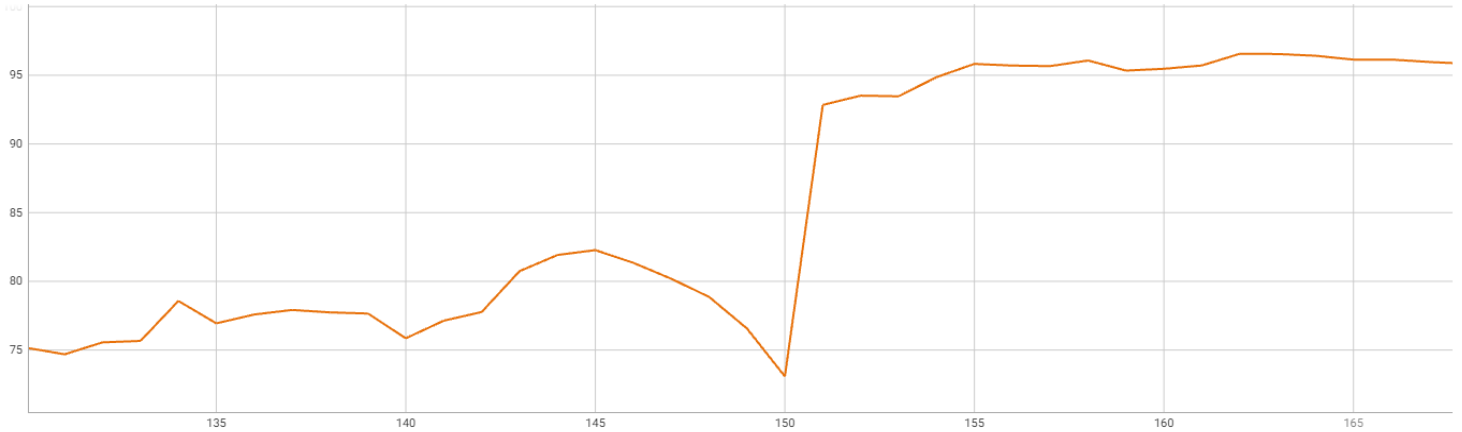
Чтобы не зависеть от начальной инициализации весов, я так же пробовал на начальных этапах переопределять награду в зависимости от состояния агента на каждом шагу, поощряя агента за достижения как можно большей высоты (удаление от координаты -0.5). Тем самым, мы получали агента способного к обучения за разумное время на стандартной награде, без зависимости от начальных весов.
На графике ниже показана средняя награда определенная мной



На 150 шаге было произведено переключение на стандартную награду. Ниже график стандартной награды для тех же эпизодов, которая не использовалась для обучения до 150 эпизода.



Как можно видеть, после переключения сеть ухудшила свои показания, но после смогла восстановиться (в районе 210 эпизода опять ухудшилась), но эти ухудшения не смотря на реварда близкий к 0 не имеют ничего общего с локальным минимумом. Эти падения были мной описаны выше и связаны с большим шагом обучения и приводят к тому, что машинка катается из стороны в сторону вплотную приближаясь к финишной точке, но не достигая ее.
Это отчетливо видно по верхнему графику (его награда упала ровно на 100 очков в районе 210 эпизода, но при это все равно оставалась положительной), машинка продолжала получать награду за удаления от точки равновесия, но уже не получала награду за преодоление финишной прямой. Видео с такой странной машинкой можно найти в папке. При этом обучение после переключения на основную награду давало прирост, так как теперь агент уже пытался оптимизировать функционал, который был изначально задан как критерий оценивания.



Отчетливо видно улучшение на 150 эпизоде из-за того, что теперь оптимизируется другой функционал.