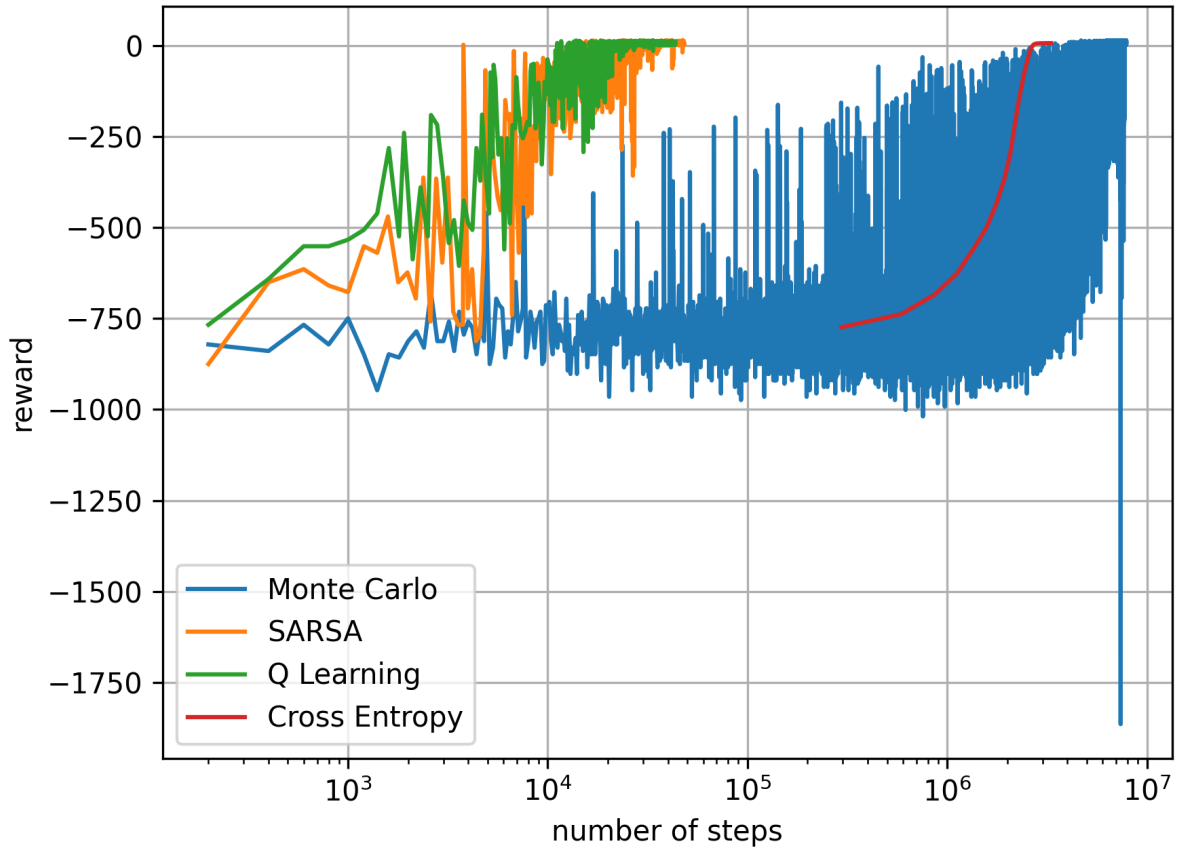
1.
После реализации Q learning алгоритма, основной сложностью для сравнения работы алгоритмов было настройка параметров метода Monte Carlo. Основным моментом, который помог в большей степени было обнуление счетчика и таблицы qfunction. Мне показалось это разумным шагом, чтобы избавиться от изначально неоптимальных шагов от полученных на ранних этапах обучения. Поэтому на половине обучения, я создал копию qfunction, которая обнуляется и заполняется в соответствии с последней сохраненной версией данной таблицы (агент действует в соответствии с последней сохраненной версией на протяжении некоторого времени). После нескольких эпох агент начинает действовать в соответствии с обновленной таблицей qfunction, тем самым забывая начальные “плохие” этапы обучения.

После таких изменений данный метод начал показывать сопоставимую производительность с методом Cross Entropy. Но все равно сильно уступал методам SARSA и Q Learning.

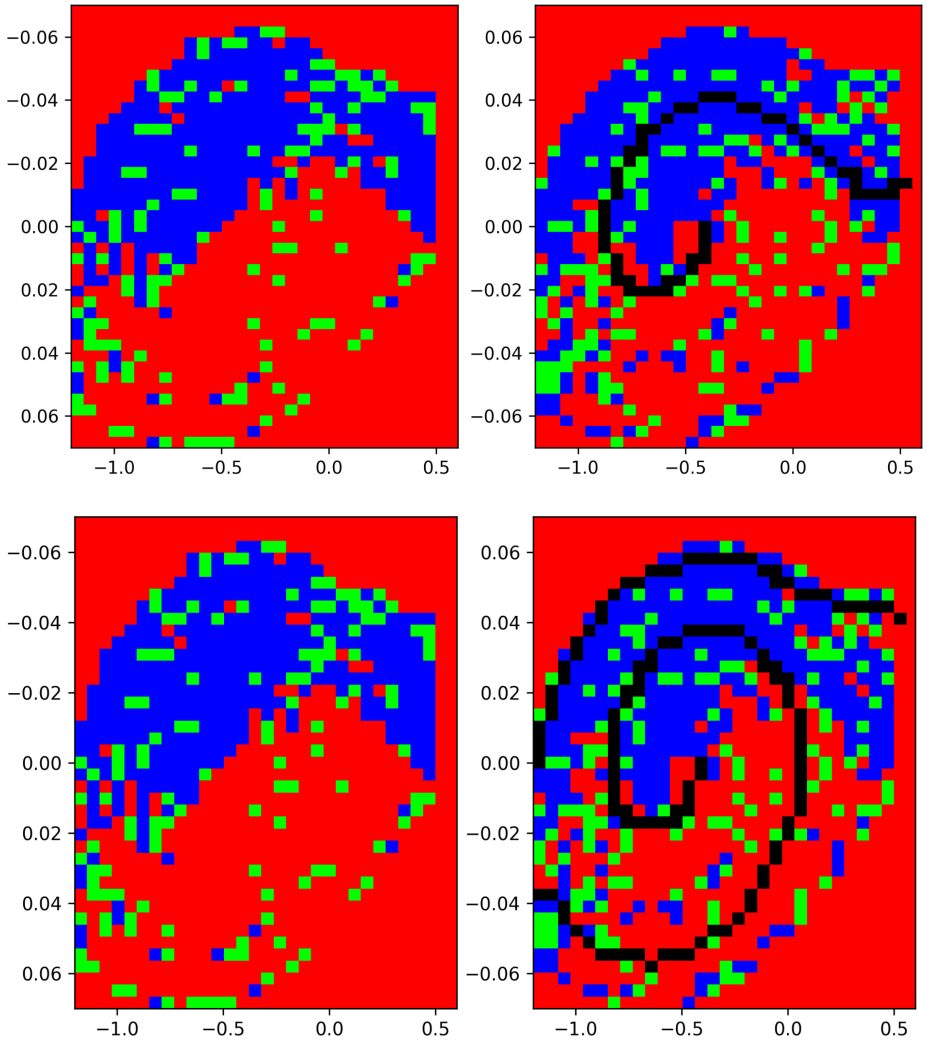
На графике ниже представлен процесс обучения в зависимости от количества произведенных действий (количество вызова функции step()).



По итогу все методы показали схожий средний реварда после обучения.

2.
Для дискретизации я выбрал среду MountainCarV0, однако сразу вяз среду с дискретным пространством действий, для упрощения процедуры дискретизации. Обучение с помощью методов SARSA и Q Learning мне показалось более стабильным чем обучение с помощью метода Cross Entropy. (В домашнем задании номер два относительно стабильного обучения мне удалось добиться только для маленькой модели надеясь на удачную начальную инициализацию весов). В случае SARSA и Q Learning повторяемость результатов гораздо выше. Иногда обучение сваливалось в локальные минимумы, что удалось решить волнообразным изменением параметра отвечающего за исследование (На первой половине обучения он менялся от 1 до нуля линейно, затем на третьей четверти от 0.5 до 0 линейно и 0.01 далее). Так же в конце уменьшался параметр alpha.

Данный подход позволяет стабильно получать модель обучающуюся решать поставленную задачу близким к оптимальному способу. Однако добиться того, чтобы qfunction была полностью заполнена оптимально у меня не получилось. Всегда в таблице оставался шум в виде неоптимальных действий в некоторых состояниях (Ниже на графике зеленые пятна отвечающие за бездействие агента).
Ниже на графике показаны две траектория полученные вследствие использования qfunction полученной методом SARSA. Первая из них удачная, вторая нет.



Результат qfunction очень похож на оптимальный. Если применить к нему какое ни будь сглаживание, то это еще улучшит результат, но это уже ближе к нейронным сетям следующего занятия.