DQN with PER on MsPacman-v0

Fyodor Karpeyev

April 2022

Содержание

Ключевые слова

обучение с подкреплением, реплей, реплей с приоритетом, DQN, Пакман, MsPacman-v0

1 Аннотация

Обучение с подкреплением сейчас являеться одной из важнейших состовляющих машинного обучения. Данная работа представляет из себя исследование обученния агента в среде MsPacman-v0 с помощью DQN.

2 Введение

Обучение с подкреплением [1] сейчас является важной составляющей машинного обучения. Оно широко используеться для решения огромного количества различных сред и может комбинироваться с широким набором дополнительных методов для улучшения показателей скорости и/или стабильности обучения. Существует множество модификаций DQN, но среди них существует большая группа моделей использующая как один из методов реплеи.

Данная работа предлагает рассмотреть процесс улучшения модели DQN с использованием реплеев в среде Atari MsPacman-v0. Исследуеться влияние разных методов и начальных значений на итоговое качество и скорость обучения модели.

3 Обучение с подкреплением

В модели мы используем обучение с подкреплением основывающиеся на Марковском процессе принятия решений [2]. Для автоматического обучения оптимальной стратегии, модель совершает действия и обновляется используя новые полученные данные о среде, в процессе максимизируется функция награды полученная от среды.

Мы формально определяем Марковский процесс принятия решений как набор из 4 элементов (S, A, P_a, R_a), где S - это набор состояний среды, A - набор действий в данной среде, которые могут быть выполнены отталкиваясь от текущего состояния, $P_a(s, s') = P(s_{t+1} = s' s_t = s, a_t = a)$ - вероятность того что действия a в состоянии s в момент времени t приведет к состоянию s' в момент времени $t+1$, $R_a(s, s')$ - вознаграждение, получаемое после перехода из состояния s в состояние s' с помощью действия a.

В процессе обучения модель формирует собственную «политику», другими словами стратегию. Выбор агента можно преставить как:

$$\pi : S \times A \to [0, 1]$$

$$\pi(a|s) = P(a_t = a|s_t = s)$$

где π - вероятностная политика выбора действий, S - состояния, A - действия.

Нейронные сети используются в обучении с подкреплением в качестве апроксиматоров, например в качестве апроксиматора Q функции, определяющей качество выполнения определённого действия в определённом состоянии:

$$Q : S \times A \to \mathbb{R}$$

DQN(Deep Q-network [3]) основывается на Q-learning [4]. Для каждой пары состояние и действие в момент времени насчитывается опередленное число $Q(s_t, a_t)$, которое изменяется в ходе обучения по формуле:

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{old\ value} + \alpha \cdot \delta$$

Где α это гиперпараметр отвечающий за то, насколько сильно изменяються значения Q-функция за 1 шаг. А δ или TD (Temporal Difference) - это

ошибка временной разницы которую и пытаться минимизировать модель, и которая вычисляеться по формуле:

$$\underbrace{\underbrace{r_t}_{reward} + \underbrace{\gamma}_{discount\ factor} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{estimate\ of\ optimal\ future\ value}}_{new\ value(temporal\ difference\ target)} - \underbrace{Q(s_t, a_t)}_{old\ value}$$

Где γ - это параметр, который отвечает за то, насколько сильно обесценивается долгосрочная награда.

Существует модификация модели DQN, которая называеться DDQN-Double Deep Q-Network. Она отличаеться наличием второй сети, а именно таргет сети, которая являеться той же сетью, но отдельно скопированной с основной, какое-то время назад. Модификация отличаеться отличаеться функцией ошибки, а именно имеет функцию:

$$\delta = r_t + \gamma Q_{target}(s_{t+1}, \text{argmax}_{a \in \mathcal{A}} Q(s_{t+1}, a)) - Q(s_t, a_t)$$

4 Реплеи

Для улучшения качества обучения используются разные методы и эвристики, одна из таких это реплеи. Они запоминают и используют старые результаты игр для стабилизации и ускорения обучения. Формально реплей - набор записей о состояниях Марковского процесса принятия решений, которые наблюдает агент в процессе обучения, а именно: состояний, действий, полученных состояний, наград и дополнительных данных. Его размер ограничен, поэтому самая бесполезная информация каким-то образом удаляеться для записи новых данных.

Одним из улучшений базового реплея, основанном на том, что удаляються самые старые записи, является реплей с приоритетом [5]. В нем объекты на удаление выбираються согласно приоритетам $P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$, где α - гиперпараметр приоритезации. Таким образом он сохраняет только самые полезные записи.

Одним из вариантов реплея с приоритетом являеться реплей в котором $p_i = |\delta_i| + \epsilon$, где ϵ это маленькая константа, гарантирующая, то что все элементы реплея имеют не нулевой шансы быть выбранными.

Для стабилизации реплея с приоритетом используеться выборка с важность. Для каждого объекта в реплее мы добавим вес:

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^{\beta}$$

Где N - это количество записей в реплее в данный момент, а β - параметр, отвечающий за то, насколько сильно влияет шанс на взятие объекта из реплея на его вес.

5 Эксперемент

Все эксперементы поставлены с использованием DDQN в среде MsPacman-v0.

5.1 Реплей с приоритетом

Рассмотрим влияние реплея с приоритетом(и выборки с важностью) на качество обучения.

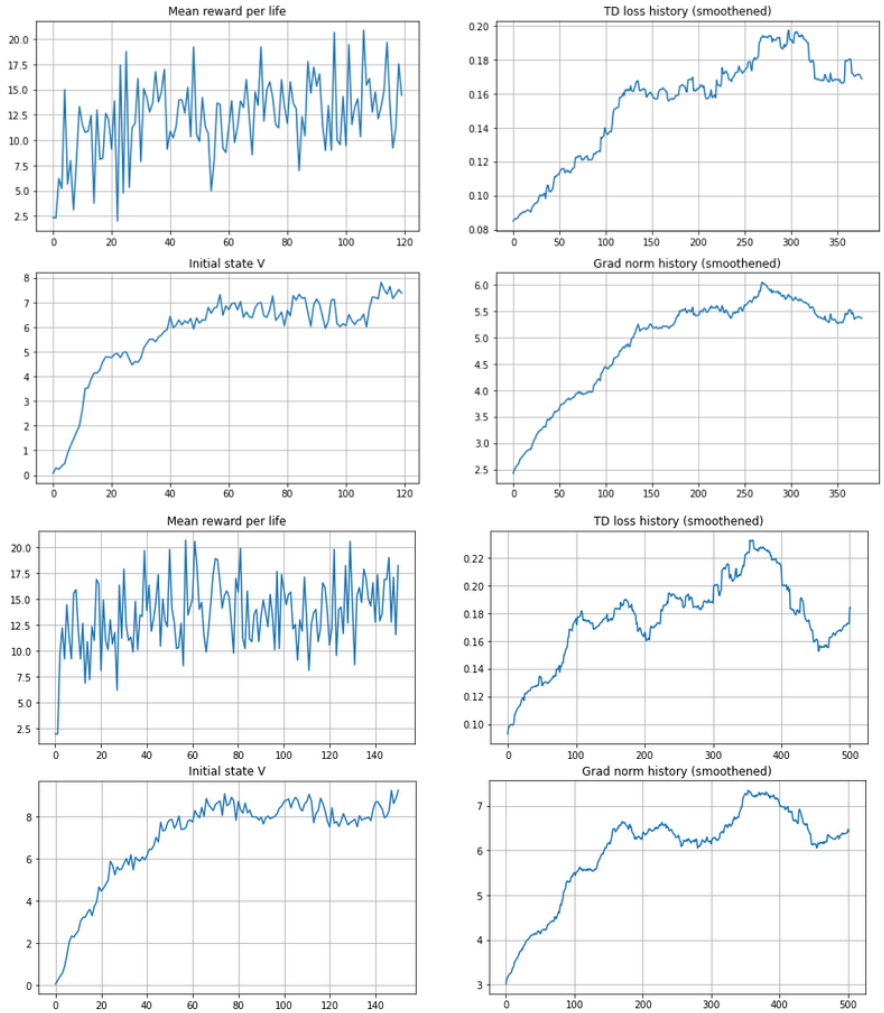


Рис. 1: Сравнение графиков обучения. Сверху с обычным реплеем. Снизу с репелеем с приоритетом.

Итоговые результаты по этим графикам это соответсвенно 450 и 700. И хотя такое большое различие в результатах скорее всего являеться следствием случайности, на нижнем графике средней награды видно, что реплей с приоритетом позволяет быстрее достигать хороших значений и стабильнее их сохранять.

5.2 Использование горячего старта

С помощью горячего старта от одного из лучших агентов обученных за все время эксперементов с результатом 750.

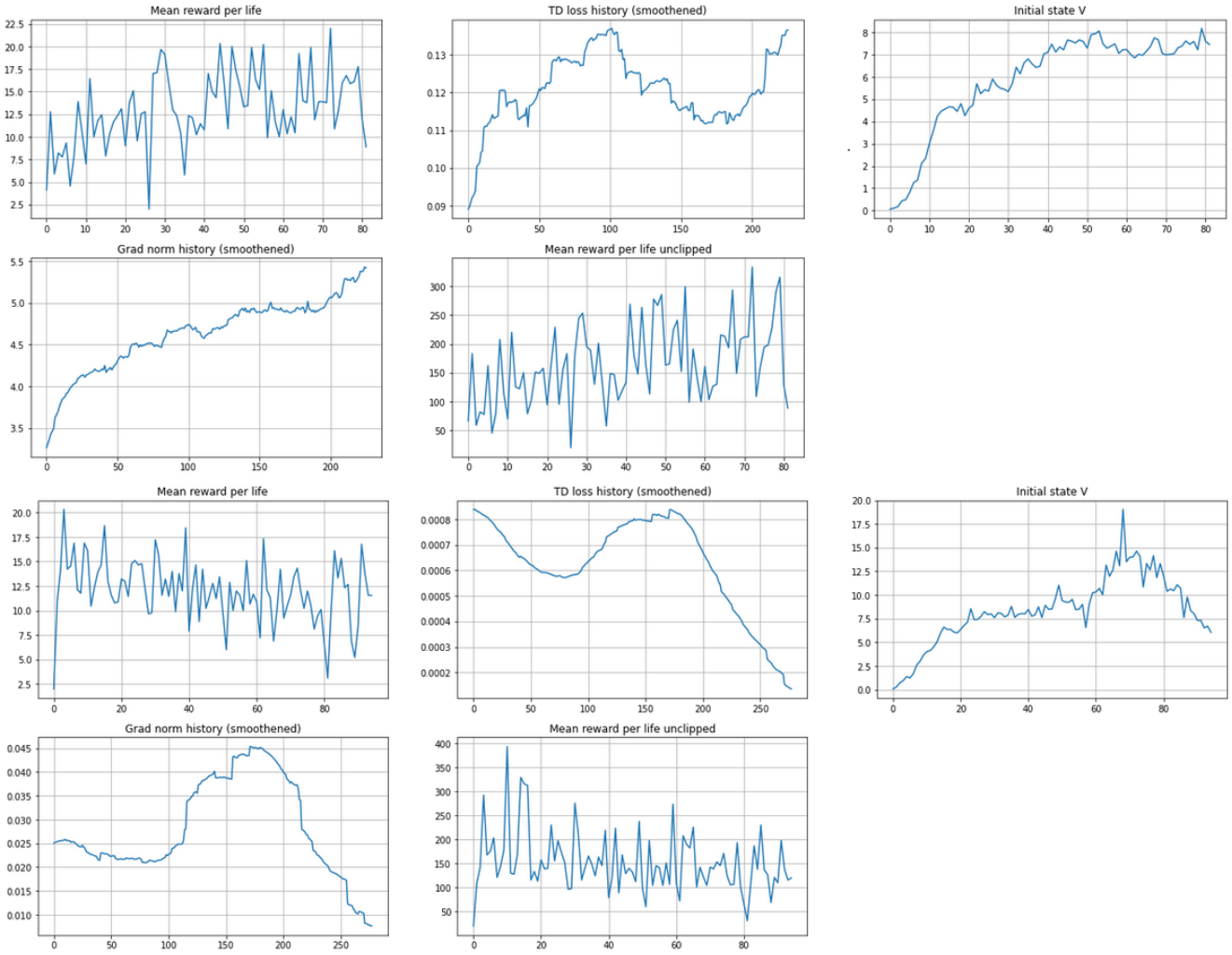


Рис. 2: Сравнение графиков обучения. Сверху без. Снизу с горячим стартом.

Как видно по графикам горячий старт помогает быстро достичь хорошей средней награды, но после какого-то времени начинает падать вниз, а графики не награды показывают просто ужастные результаты в отличие от отсутсутствия горячего старта.

Вот еще 1 график с использованием горячего старта уже для другой конфиругации модели и другого агента для заполнения:

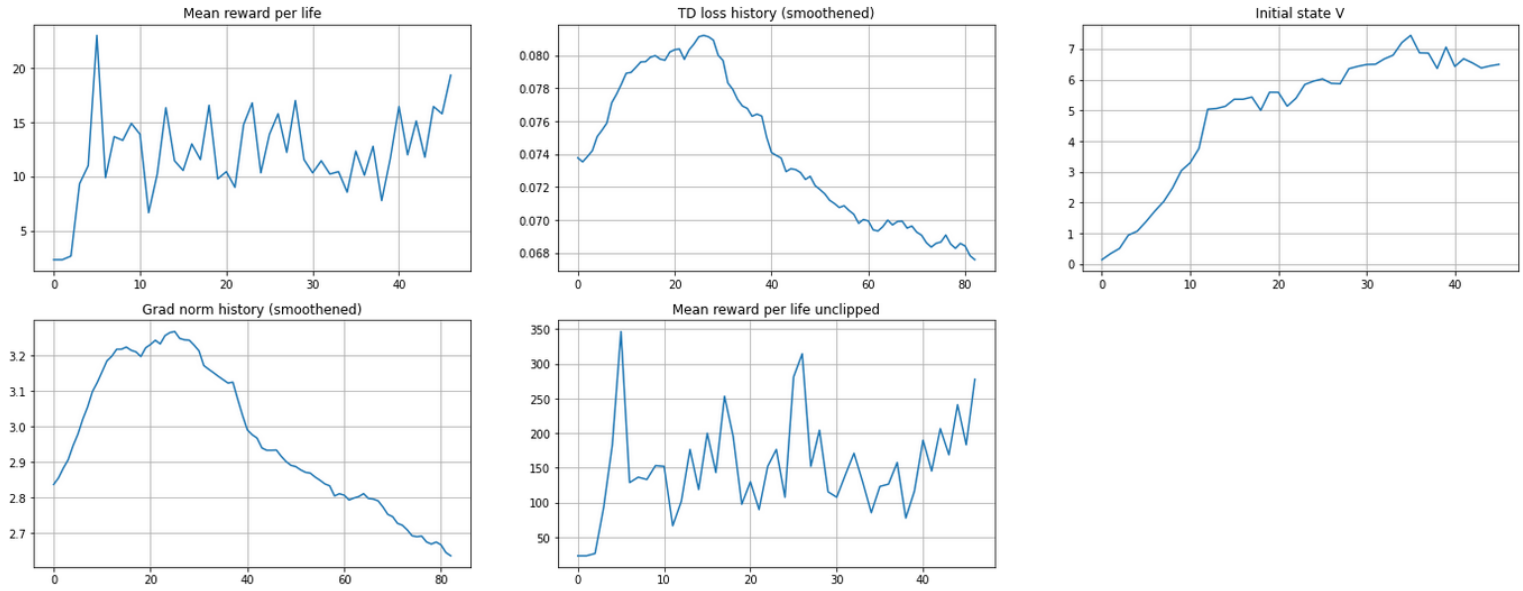
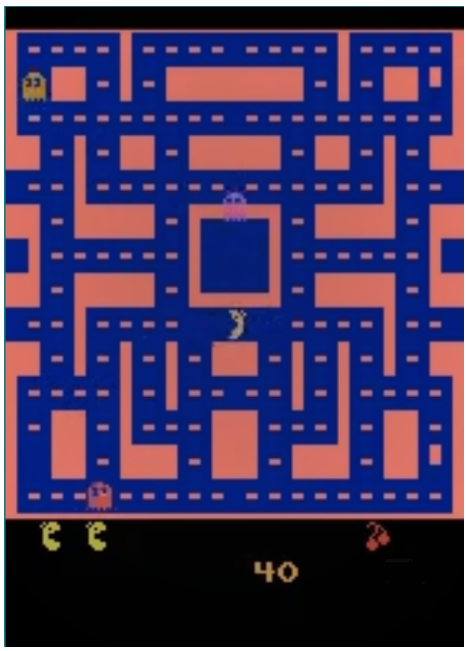


Рис. 3: График обучения с горячим стартом.

Видны те же самые проблемы, и хотя тут график Initial State V еще не упал, виден его отрицательный тренд.

5.3 Методы сжатия входных данных

Изначально картинка в Пакамане имеет вид: особа сжатия цветов картинки в оттенки серого, и в либо черный либо белый цвет:



И помимо обрезания и сжатия, которые банально ускоряют процесс, засчет меньшего числа данных, и использования бафера из 4 последовательных кадров, для понимания направления и скорости движения. Мы рассмотрим 2 способа сжатия цветов картинки в оттенки серого, и в либо черный либо белый цвет:

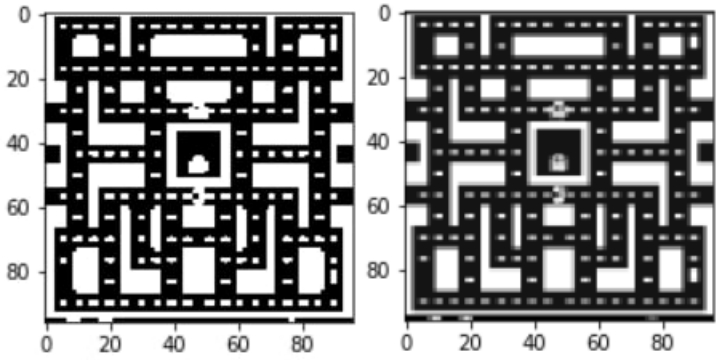


Рис. 4: Версия просле обработки слева только черный или белый, справа оттенки серого.

На левой картинке не хватает некоторых деталей, и это подтверждаеться обучением. Не понятным мне образом приведение картинки в бинарный вид пикселей вызывает взрыв коэфицентов даже при рабочем нормализаторе:

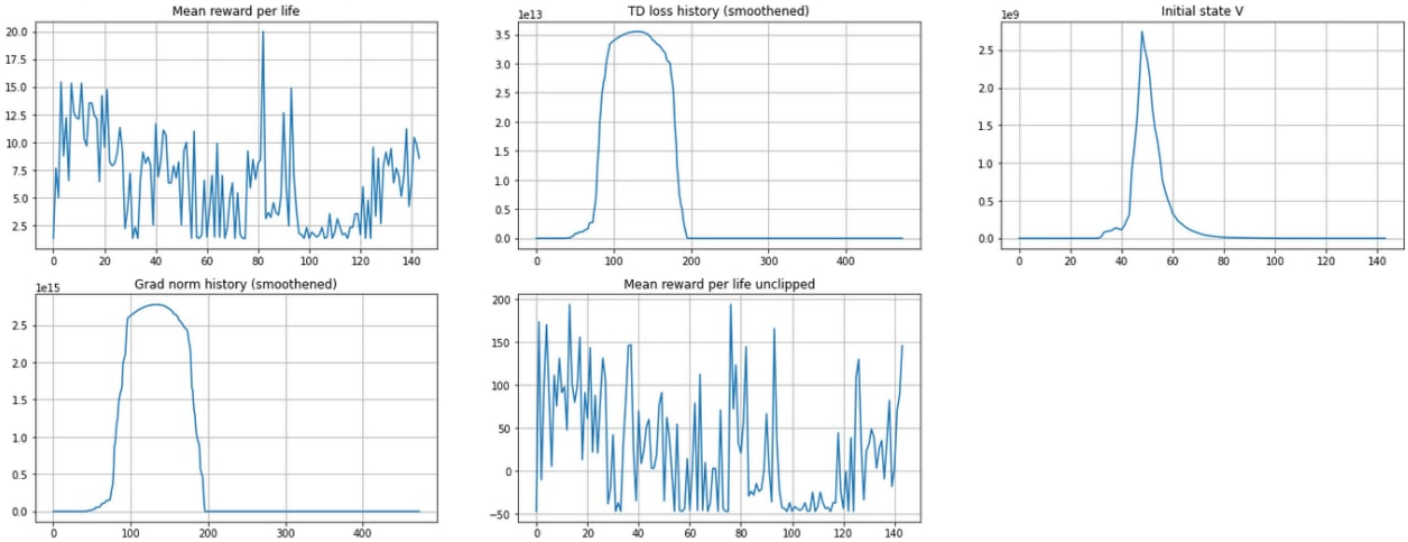


Рис. 5: График обучения со взрывом коэфицентов.

Несмотря на взрыв коэфицентов, агент обученный таким образом показывает не лучший, но неплохой результат.

Можно посмотреть на еще один подобный пример, с немного другой внутренней архитектурой и другими параметрами сети.

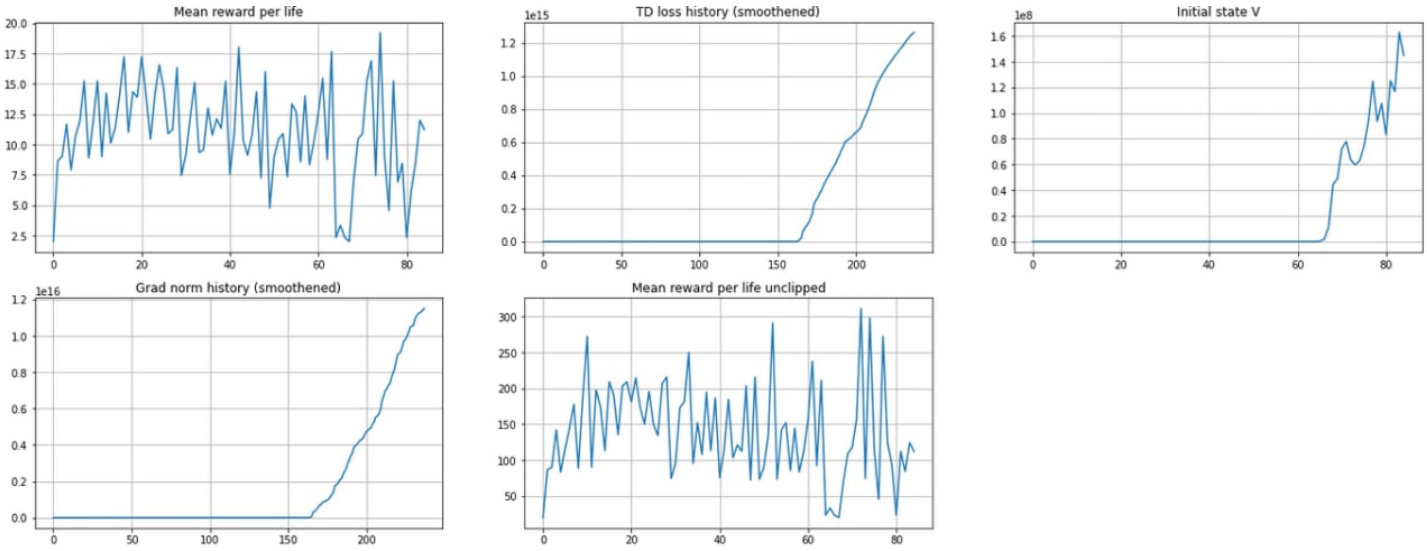


Рис. 6: Еще один график обучения со взрывом коэфицентов.

6 Выводы и дальнейшая работа

Лучщий результат у агента, которого мне удалось достичь это 750 очков, что являеться достаточно неплохим результатом для достаточно простой модели без использования разных трюков. Делая выводы из проделанной работы реплей с приоритетом улучшает качество обучения, но с другими методами надо быть осторожнее. Любой метод, который в теории должен улучшать качество, может наоборот ухудшить его или вызвать непредвиденные последствия. С точки зрения DQN расширения не в глубь, а в ширь чаще всего не дает сильно хороших результатов, так например увеличение реплея, или размера выборки из реплея, или учащение обновления таргет сети, или расширение сети на несколько слоев дает почти незаметное улучшение качесва, но при этом модель обучаеться намного дольше. Исходя из этого, можно сказать что для простой модели типо DDQN достоином способом улучшения являеться добавление дополнительных методов и эвристик.

Дальнейшая работа состоит в том, чтобы изуть разнообразные методы и эвристики и рассмотреть их влияние на качество обучения этой модели и других, а так же провсти подобные эксперементы в других средах.

7 Заключение

В данной работе рассматриваеться исследование поведения модели DDQN на среде MsPacman-v0. Было достигнуто хорошее относительно возможно-

стей итоговое качество, а полученные знания и методы можно применить для других моделей и сред.

Список литературы

[1] Vincent François-Lavet и др. «An Introduction to Deep Reinforcement Learning». Англ. В: *CoRR* abs/1811.12560 (2018). arXiv: 1811.12560. URL: http://arxiv.org/abs/1811.12560.

[2] «Markov decision process». В: (). URL: https://en.wikipedia.org/wiki/Markov_decision_process.

[3] Volodymyr Mnih и др. «Playing Atari with Deep Reinforcement Learning». Англ. В: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[4] «Q-learning». В: (). URL: https://en.wikipedia.org/wiki/Q-learning.

[5] Tom Schaul и др. «Prioritized Experience Replay». Англ. В: *ICLR*. 2016.