Санкт-Петербургский политехнический университет Петра Великого

Институт металлургии, материаловедения и транспорта

Высшая школа автоматизации и робототехники

**Отчет**

Дисциплина: Методы искусственного интеллекта в Мехатронике и

Робототехнике

Тема: «Глубокое обучение с подкреплением (DRL)»

|  |  |
| --- | --- |
| Выполнил студент | Г. В. Казанцев |
| Преподаватель | Т.Т. Исаков |

Санкт-Петербург

2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc199752047)

[1. Принципы работы глубокого обучения с подкреплением 4](#_Toc199752048)

[1.1 Применение в реальном мире 6](#_Toc199752049)

[1.2 Алгоритмы DQN 7](#_Toc199752050)

[2. Описание окружения Lunar-Lander-V3 10](#_Toc199752051)

[2.1 Функции награды 12](#_Toc199752052)

[2.1.1 Базовая функция награды 12](#_Toc199752053)

[2.1.2 Умеренное поощрение за выживание и точность 13](#_Toc199752054)

[2.1.4 Награда, зависящая от высоты (y) 18](#_Toc199752055)

[2.1.5 Разреженная награда 21](#_Toc199752056)

[ЗАКЛЮЧЕНИЕ 24](#_Toc199752057)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 26](#_Toc199752058)

# ****ВВЕДЕНИЕ****

**Deep Reinforcement Learning (DRL) сочетает в себе принципы Reinforcement Learning (RL) и Deep Learning (DL). Оно позволяет агентам обучаться оптимальному поведению в сложных, часто высокоразмерных средах методом проб и ошибок [1]. В отличие от традиционного RL, который может не справиться с огромными пространствами состояний, например, с данными о пикселях с камеры, DRL использует глубокие нейронные сети (NN) для аппроксимации функций, необходимых для обучения, таких как функция ценности (предсказание будущего вознаграждения) или политика (отображение состояний на действия). Это позволяет агентам DRL решать ранее неразрешимые проблемы, обучаясь непосредственно на сложных сенсорных входах, таких как изображения или показания датчиков.**

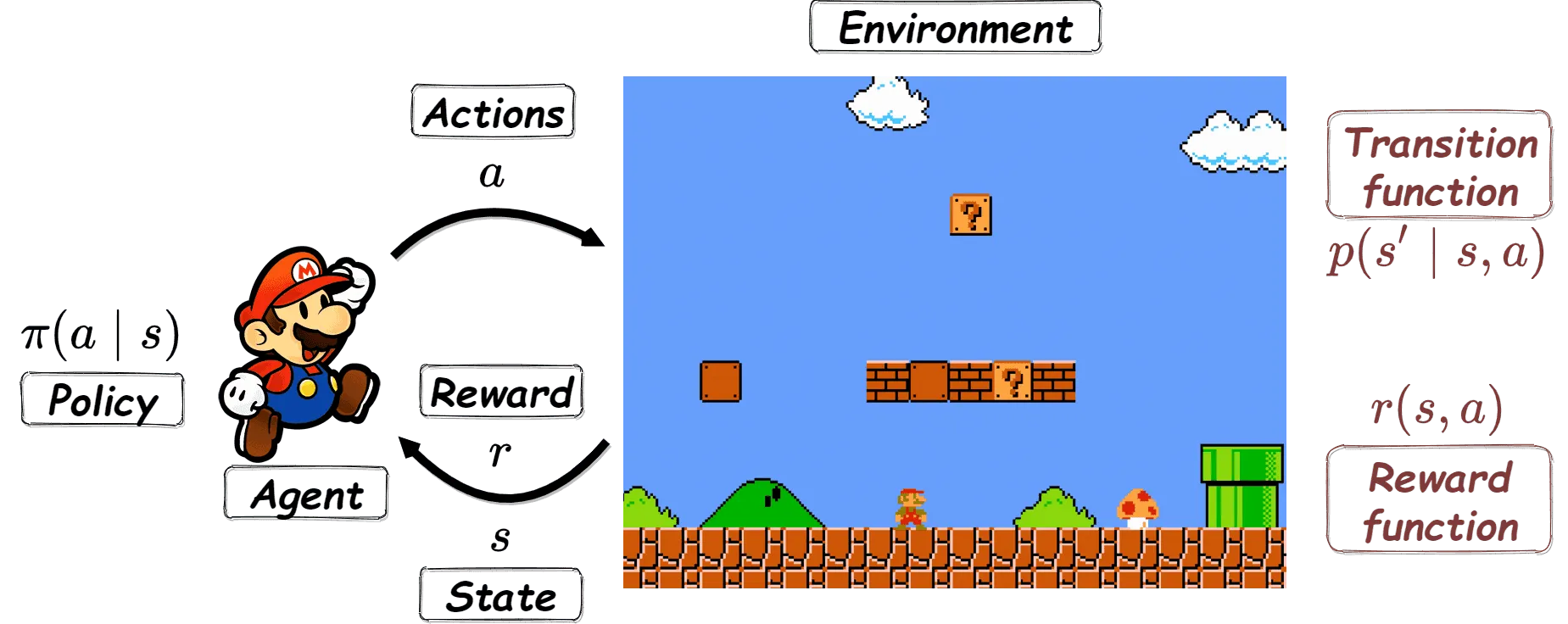
**В последние годы DRL произвела революцию во многих областях, добившись впечатляющих результатов, превосходящих возможности человека, в таких задачах, как игры (например, Atari, Go, Dota 2), робототехника (антропоморфные роботы), управление ресурсами, разработка лекарств и т.д. Это стало возможно благодаря сочетанию вычислительной мощности современных компьютеров, больших объемов данных для обучения и усовершенствованных архитектур нейронных сетей. Однако, несмотря на свой успех, DRL по-прежнему остается активно развивающейся областью исследований, сталкивающейся с рядом проблем, таких как высокая чувствительность к гиперпараметрам, нестабильность обучения, необходимость в больших объемах данных и сложность переноса знаний между разными задачами. Преодоление этих трудностей является ключевым для раскрытия полного потенциала DRL и применения его в еще более сложных и реальных задачах.**

# ****Принципы глубокого обучения с подкреплением****

## ****Процесс обучения****

**DRL предполагает взаимодействие агента с окружающей средой на протяжении дискретных временных шагов [2]. Обычно этот процесс происходит следующим образом (см. рис. 1):**

* **Наблюдение: Агент воспринимает текущее состояние среды. В отличие от традиционных подходов, DRL может работать с очень сложными данными, например, с изображениями. В этом случае, для понимания увиденного, агент использует сверточные нейронные сети (CNN), которые анализируют пиксели и извлекают из них полезную информацию.**
* **Выбор действия: Основываясь на наблюдаемом состоянии, агент выбирает действие, используя свою политику, которая представлена глубокой нейронной сетью.**
* **Взаимодействие: Агент выполняет выбранное действие, что приводит к переходу окружения в новое состояние.**
* **Обратная связь: Окружающая среда предоставляет скалярный сигнал вознаграждения, указывающий на то, насколько хорошим или плохим было действие в предыдущем состоянии.**
* **Обучение: Агент использует сигнал вознаграждения и переход состояния для обновления своей нейронной сети (политики или функции ценности) с помощью таких алгоритмов, как обратное распространение и градиентный спуск. Цель - настроить веса сети так, чтобы максимизировать кумулятивное будущее вознаграждение с течением времени. Этот цикл обучения повторяется, позволяя агенту постепенно улучшать свою стратегию принятия решений.**

**Рисунок 1 – Процесс обучения на примере Mario Bros**

**Понимание DRL предполагает знакомство с несколькими основными идеями из RL, которые теперь масштабируются с помощью методов глубокого обучения:**

* **Агент: Алгоритм или модель, обучающаяся принимать решения.**
* **Окружающая среда: Мир или система, с которой взаимодействует агент (например, симуляция игры, окружение физического робота). Стандартизированные среды для исследований часто предоставляются наборами инструментов в Gymnasium.**
* **Состояние: Представление окружающей среды в определенный момент времени. DRL отлично справляется с состояниями, представленными большими объемами данных, например изображениями или массивами датчиков.**
* **Действие: Решение, принятое агентом, которое влияет на окружающую среду.**
* **Вознаграждение: Числовая обратная связь от окружающей среды, указывающая на непосредственную желательность действия, совершенного в том или ином состоянии.**
* **Политика: Стратегия агента, отображающая состояния на действия. В DRL это, как правило, глубокая нейронная сеть.**
* **Функция ценности в обучении с подкреплением, часто реализуемая с помощью глубокой нейронной сети, предсказывает общий ожидаемый "выигрыш" в будущем, начиная с конкретного состояния или комбинации состояния и действия. Другими словами, она помогает определить, насколько "хорошо" находиться в определенном положении или сделать определенный шаг.**
* **Исследование против эксплуатации: Фундаментальный компромисс, при котором агент должен балансировать между попыткой попробовать новые действия, чтобы обнаружить лучшие стратегии (разведка), и тем, чтобы придерживаться известных хороших действий (эксплуатация).**

**DRL в сравнении с другими парадигмами машинного обучения DRL существенно отличается от других первичных подходов к машинному обучению (ML):**

* **Супервизорное обучение: Обучается на наборе данных, содержащем помеченные примеры (пары вход-выход). В таких задачах, как классификация изображений или обнаружение объектов, используются такие модели, как Ultralytics YOLO относятся к этой категории. DRL, напротив, учится на сигналах вознаграждения без явных правильных ответов для каждого состояния.**
* **Неподконтрольное обучение: Изучает паттерны и структуры на основе немаркированных данных (например, кластеризация). DRL фокусируется на обучении целеустремленному поведению через взаимодействие и обратную связь.**
* **Обучение с подкреплением (RL): DRL - это особый тип RL, в котором используются глубокие нейронные сети. В традиционном RL часто используются более простые представления, такие как таблицы (Q-таблицы), которые невыполнимы для задач с очень большими или непрерывными пространствами состояний, где DRL блистает.**

## 

## ****1.2 Алгоритмы DQN****

Q-learning, хотя и мощный алгоритм, имеет существенный недостаток: он не умеет обобщать. Если агент Q-learning сталкивается с состоянием, которого раньше не видел, он не знает, что делать. Чтобы решить эту проблему, был разработан DQN (Deep Q-Network)[5], который заменяет двумерную таблицу Q-значений нейронной сетью.

Для непрерывных состояний естественно аппроксимировать функцию полезности состояний нейронной сетью  с параметрами . При этом решаются обе упомянутые выше проблемы (размерность и зашумлённость). Обучать сеть следует таким образом, чтобы Q-функция удовлетворяла уравнению Беллмана. В алгоритме DQN агент, как и в Q-обучении, совершает "оптимальное" действие  , возможно, с не жадной ϵϵ-стратегией, получает от среды новое состояние и награду . В конечной памяти сохраняются четвёрки соседних значений  по достаточно большому числу эпизодов. Фактически это *не устареваемые* данные характеризующее модель среды. Тем не менее память должна быть конечной, чтобы "забывать" области пространства состояний и действий вдали от оптимальной стратегии. При обучении, после каждого временного шага, из памяти случайным образом извлекается батч (наборы четвёрок). Случайная выборка решает проблему корреляций последовательных во времени состояний, которые обычно не сильно отличаются друг от друга. Именно эта корреляция мешала глубокому обучению до работы DeepMind.

При помощи батча формируется функция ошибки равная квадратам отклонений примеров от уравнения Беллмана с жадной стратегией выбора действия (B - размер батча):

В слагаемом стоят фиксированные параметры **целевой сети**, полученные на предыдущих итерациях обучения. После нескольких циклов обучения заменяются на текущие значения. Таким образом, основная сеть оценивает ценность текущего состояния и действия, а целевая сеть оценивает ценность следующего состояния и действия. Иногда целевая сеть "замораживается" (не меняется) на протяжении нескольких десятков тысяч итераций обучения.Хотя в некоторых задачах эффективнее и жёсткое (частое) обновление Обе сети имеют одинаковую архитектуру и вначале их веса инициализируются различными случайными значениями. Использование целевой (старой) сети для действий и вычисления max делают обучения более устойчивым ("не стоит оптимизировать на ходу быстро едущую машину").

DQN обучается методом обучения с подкреплением: агент взаимодействует с окружающей средой, совершает действия, получает награды или штрафы, и на основе этого опыта корректирует свою стратегию. Ключевыми особенностями DQN являются:

* Аппроксимация Q-функции с помощью DNN: Это позволяет DQN работать со сложными пространствами состояний, такими как изображения.
* Experience Replay: DQN хранит опыт (переходы "состояние-действие-награда-следующее состояние") в буфере и обучает нейронную сеть на случайных мини-пакетах из этого буфера, снижая корреляцию между последовательными переходами и стабилизируя обучение.
* Target Network: DQN использует отдельную, "замороженную" нейронную сеть для оценки Q-значений в целевой функции, что также стабилизирует обучение, так как целевые значения меняются не так быстро.

Таким образом, DQN — один из самых важных алгоритмов глубокого обучения с подкреплением, демонстрирующий сочетание глубокого обучения и обучения с подкреплением для решения сложных задач. Он стал основой для многих других, более продвинутых алгоритмов.

# ****Описание окружения Lunar-Lander-V3****

**Среда представляет собой классическую задачу оптимизации траектории ракеты [3]. Согласно принципу максимума Понтрягина [4], оптимальным является включение двигателя на полную мощность или его выключение. Именно по этой причине в этой среде есть дискретные действия: включение или выключение двигателя.**

**Существует два варианта среды: дискретная или непрерывная. Посадочная площадка всегда находится в координатах (0,0). Координаты - это первые два числа в векторе состояния. Возможна посадка за пределами посадочной площадки. Запас топлива бесконечен, поэтому агент может научиться летать, а затем приземлиться с первой попытки.**

**Доступно четыре отдельных действия:**

* **0: ничего не делать**
* **1: запустить левый механизм ориентации**
* **2: запустить основной двигатель**
* **3: запустить правый механизм ориентации**

**Состояние среды представляет собой 8-мерный вектор: координаты посадочного модуля в координатах x и y, его линейные скорости в координатах x и y, его угол, его угловая скорость и два логических значения, которые представляют, соприкасается ли каждая опора с землей или нет.** **После каждого шага выдается награда. Общая награда за эпизод - это сумма наград за все шаги в этом эпизоде.**

**За каждый шаг в изначальной среде награда:**

* **увеличивается/уменьшается по мере приближения/удаления посадочного модуля к посадочной площадке.**
* **увеличивается/уменьшается, чем медленнее/быстрее движется посадочный модуль.**
* **уменьшается, чем больше наклонен посадочный модуль (угол не горизонтальный).**
* **увеличивается на 10 пунктов за каждую опору, соприкасающуюся с землей.**
* **уменьшается на 0,03 балла за каждый кадр, который запускает боковой движок.**
* **уменьшается на 0,3 балла за каждый кадр, на котором запускается основной двигатель.**

**За эпизод, который потерпел аварию или благополучно приземлился, вы получаете дополнительную награду в размере -100 или +100 баллов соответственно.** **Эпизод считается решенным, если он набрал не менее 200 баллов.** **В начальном состоянии посадочный модуль стартует в верхней части экрана, к центру масс которого приложена произвольная начальная сила.**

**Эпизод заканчивается, если:**

* **посадочный модуль падает (корпус модуля соприкасается с Луной).;**
* **посадочный модуль выходит за пределы видового экрана (координата x больше 1).;**
* **посадочный модуль не активен. Согласно документации Box2D, тело, которое не активировано, - это тело, которое не движется и не сталкивается ни с каким другим телом:**

**Есть несколько неожиданных ошибок, связанных с реализацией среды.**

**1. Положение боковых подруливающих устройств на корпусе посадочного модуля изменяется в зависимости от ориентации посадочного модуля. Это, в свою очередь, приводит к тому, что к посадочному модулю прикладывается крутящий момент, зависящий от ориентации.**

**2. Единицы измерения состояния не согласованы. Т.е.**

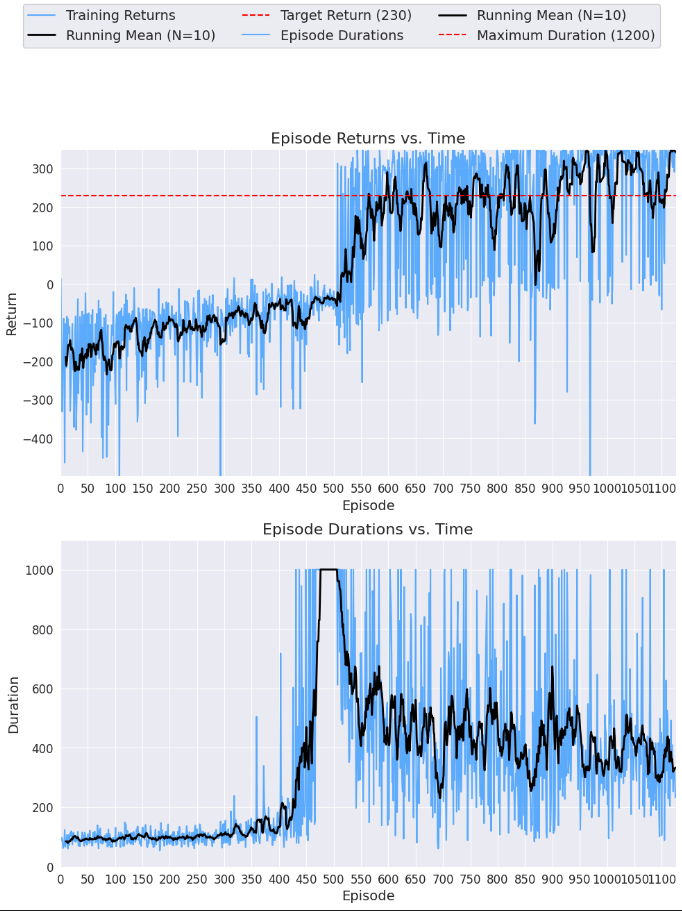
**• Угловая скорость указана в единицах измерения, равных 0,4 радиан в секунду. Для преобразования в радианы в секунду это значение необходимо умножить на коэффициент 2,5.**

**После внесения корректировки единицы измерения состояния будут следующими: "x": (единицы измерения), "y": (единицы измерения), "vx": (единицы измерения в секунду), "vy": (единицы измерения в секунду), "угол": (радианы), ‘угловая скорость": (в радианах в секунду)**

## ****2.1 Функции награды****

### ****2.1.1 Базовая функция награды****

**Функция** базовой награды **формирует поведение агента в задаче посадки, используя комбинацию штрафов и бонусов. Она непрерывно оценивает состояние агента, измеряя его удаленность от цели (горизонтального центра), угол наклона и скорость. За отклонения от идеальных параметров (нулевое расстояние, нулевой угол, низкая скорость) начисляются штрафы, которые подталкивают агента к более аккуратной посадке. В случае успешного завершения эпизода (то есть, посадки без крушения), агент получает бонус, но только если к моменту посадки он уже сумел заработать положительную награду, что исключает возможность "обучения" катастрофически быстрым, но неконтролируемым приземлениям. Результирующая награда является суммой всех штрафов и, при выполнении условий, бонуса за успешную посадку. Результаты обучения показаны на рисунке 2. Для обучения потребовалось 1100 эпизодов.**



**Рисунок 2 – Результаты обучения по базовой функции награды**

### ****2.1.2 Умеренное поощрение за выживание и точность****

Внесем изменения в функцию награды, чтобы она более полно учитывала другие аспекты среды.

**Компоненты Награды:**

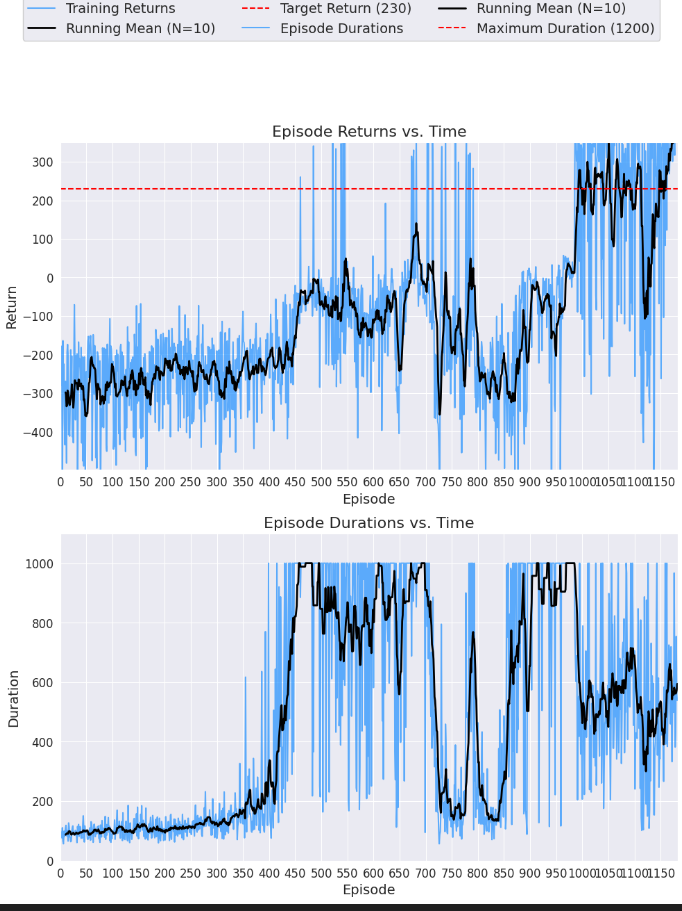
* **Штрафует за горизонтальное расстояние от центра.** **Заставляет агента располагаться рядом с местом посадки.**
* **Штрафует за большие углы наклона, стимулируя агента оставаться** **в вертикальном положении.**
* **Штрафует за высокие скорости, стимулируя мягкую посадку.**
* **Бонус за успешную посадку, если агент накопил положительную награду.**
* **Штраф за неудачную посадку, если агент не накопил** **положительную награду.**
* **Штрафует за преждевременное завершение эпизода из-за** **прерывания (например, истечение времени).**
* **Предоставляет небольшую положительную награду за каждый шаг, когда агент остается в живых и в полете.**

**Эта функция балансирует между точной посадки и предотвращением** **преждевременного завершения. Она использует стратегию формирующего награждения,** **основанную на штрафах, чтобы направить агента к желаемому поведению при посадке.** **Бонус/штраф за посадку зависит от накопленной агентом награды, а штраф за** **прерывание стимулирует агента избегать преждевременного завершения из-за** **истечения времени. Небольшая награда за выживание предоставляет постоянный** **стимул оставаться в живых и продолжать обучение.**

**Ключевые отличия от базовой функции награды:**

**1. Штраф за Неудачную Посадку: Предыдущая функция давала только бонусную награду за посадку и не штрафовала за крушения явно (кроме кумулятивных штрафов за плохое положение/угол/скорость). Эта версия штрафует завершенный эпизод (например, крушение), если накопленная награда не положительная. Это делает агента более нетерпимым к крушениям.**

**2. Штраф за Прерывание: Эта версия добавляет штраф за прерванный эпизод (например, истечение времени). Это препятствует агенту просто ждать до истечения времени и стимулирует его активно пытаться приземлиться.**

**3. Награда за Выживание: Эта версия добавляет небольшую постоянную награду (+0.1) за каждый шаг, который агент выживает. Это обеспечивает небольшой постоянный стимул оставаться в полете и исследовать, особенно на ранних этапах обучения. Это помогает предотвратить быстрое нахождение плохого решения (например, немедленное крушение) и прекращение исследования.** **Результаты обучения показаны на рисунке 3. Для обучения потребовалось 1150 эпизодов.**

**Рисунок 3 – Результаты обучения по умеренному поощрению функции награды**

**В заключение, эта функция награды более сложная. Она пытается непосредственно устранить потенциальные проблемы в предыдущей версии путем.**

**2.1.3 Агрессивное поощрение за скорость и точность (риск)**

**Эта функция награды разработана для обучения агента выполнять быструю, точную и стабильную посадку. Она использует многокомпонентный подход, включающий штрафы, бонусы и стимулы, чтобы сформировать желаемое поведение. В отличие от предыдущих версий, эта функция явно наказывает за неудачи и поощряет активное стремление к посадке, а также выживание в процессе обучения.**

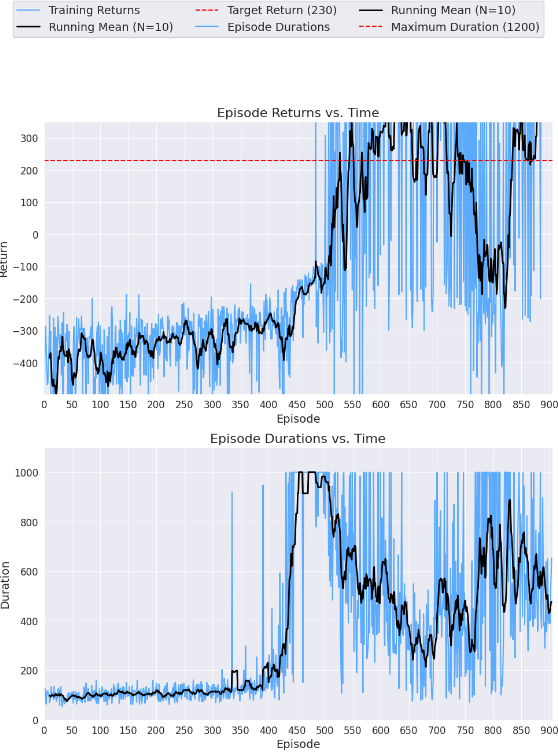
**Компоненты Награды:**

* **Штрафует агента за горизонтальное удаление от целевой точки посадки. Этот штраф стимулирует агента занимать позицию ближе к месту посадки, готовясь к безопасному приземлению.**
* **Штрафует агента за большие углы наклона. Это мотивирует агента сохранять вертикальное положение, что важно для контроля и стабильности, особенно при подходе к посадке.**
* **Штрафует агента за высокую скорость. Стимулирует более мягкий и контролируемый спуск, уменьшая риск крушения при контакте с землей.**
* **Бонус за успешную посадку: Предоставляется агенту, успешно завершившему эпизод (например, благополучно приземлился) и при этом накопившему положительную награду за предыдущие шаги. Этот бонус усиливает желаемое поведение и поощряет к повторению успешных стратегий.**
* **Штраф за неудачную посадку: Назначается агенту при завершении эпизода с отрицательным общим вознаграждением (например, в результате крушения). Этот штраф делает агента более чувствительным к ошибкам и стимулирует избегать ситуаций, приводящих к крушениям. Ключевое отличие от предыдущих версий: явное наказание за неудачи.**
* **Штраф за Прерывание: Штрафует агента за эпизоды, которые заканчиваются преждевременно из-за прерывания (например, истечение времени). Это препятствует пассивному поведению и стимулирует агента активно пытаться приземлиться до истечения времени. Ключевое отличие от предыдущих версий: стимулирует активное стремление к посадке.**
* **Награда за Выживание: Предоставляет небольшую положительную награду за каждый шаг, когда агент остается в живых и находится в полете (эпизод не завершен и не прерван). Этот небольшой, но постоянный стимул поощряет агента оставаться в воздухе и исследовать окружающую среду, особенно на ранних этапах обучения, предотвращая преждевременное принятие плохих стратегий (например, быстрое крушение).**

**Ключевое отличие от предыдущих версий: обеспечивает постоянный стимул для исследования.**

**Эта функция награды использует стратегию формирующего награждения, чтобы направить агента к желаемому поведению при посадке. Штрафы за расстояние, угол и скорость формируют траекторию полета и помогают агенту подготовиться к мягкой посадке. Бонус за успешную посадку закрепляет удачные попытки, а штраф за неудачную посадку мотивирует агента избегать крушений. Штраф за прерывание стимулирует активные попытки приземлиться, а награда за выживание обеспечивает постоянный стимул для обучения и исследования. Зависимость бонуса/штрафа за посадку от накопленной награды позволяет дифференцировать успешные и неудачные завершения эпизодов.**

**Ключевые отличия от предыдущей функции награды:**

1. **Штраф за Неудачную Посадку: Предыдущая функция давала только \*бонус\* за успешную посадку и \*не штрафовала\* явно за крушения (кроме кумулятивного эффекта штрафов за плохое положение, угол и скорость). Эта версия \*штрафует\* завершенный эпизод (например, крушение) \*отрицательным\* landing\_bonus, если накопленная награда не положительная. Это делает агента более нетерпимым к крушениям и более склонным к поиску безопасных посадок.**
2. **Штраф за Прерывание: Эта версия добавляет штраф за \*прерванный\* эпизод (например, из-за истечения времени). Это препятствует агенту просто ждать до истечения времени и стимулирует его активно пытаться приземлиться, даже если времени осталось мало.**
3. **Награда за Выживание: Эта версия добавляет небольшую постоянную награду за каждый шаг, который агент выживает. Это обеспечивает небольшой постоянный стимул оставаться в полете и исследовать, особенно на ранних этапах обучения. Это помогает предотвратить быстрое нахождение плохого решения (например, немедленное крушение) и прекращение исследования.  
   В заключение, эта функция награды представляет собой более сложную и сбалансированную систему, направленную на обучение агента выполнению точной, стабильной и быстрой посадки. Она явно наказывает за ошибки, поощряет активное обучение и исследование, а также формирует желаемое поведение с помощью многокомпонентного подхода. Она непосредственно устраняет потенциальные проблемы, замеченные в предыдущих версиях, делая обучение более эффективным и стабильным. Результат обучения показан на рисунке 4. Для обучения потребовалось 900 эпизодов.**

**Рисунок 4 – Результаты обучения по рисковой функции награды**

2.1.4 Награда, зависящая от высоты

**Эта функция награды направлена на обучение агента мягкой и точной посадке с учетом высоты над землей. Она использует сочетание штрафов за отклонение от целевой позиции и угла, а также штрафов за скорость, которые усиливаются по мере приближения к земле, стимулируя замедление перед посадкой.**

**Компоненты Награды:**

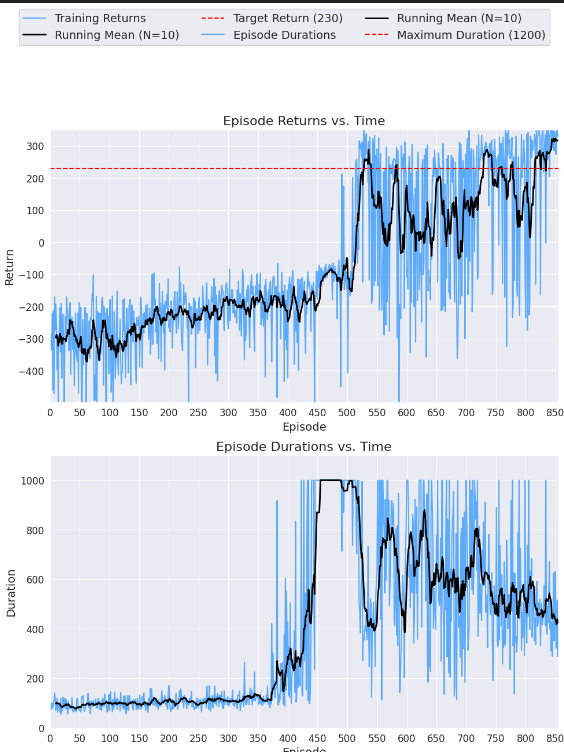
* **Линейный штраф за отклонение по горизонтали от целевой точки посадки.**
* **Линейный штраф за отклонение от вертикального положения.**
* **Линейный штраф за скорость, учитывающий абсолютные значения скоростей по обеим осям.**
* **Штраф за скорость в зависимости от высоты:  
  Если высота агента (**y**) меньше 50 (метров, единиц измерения), применяется дополнительный штраф за скорость.  
  Штраф за скорость удваивается, если агент находится достаточно близко к земле. Это мотивирует агента замедлиться перед посадкой. Ключевая особенность: контекстная награда, зависящая от высоты.**
* **Бонусы и штрафы за завершение эпизода:  
  Если эпизод завершился успешно (и накоплена положительная награда), добавляется бонус за посадку.  
  Если эпизод завершился неудачно (и/или накоплена отрицательная награда), применяется штраф за неудачу  
  Если эпизод был прерван, назначается небольшой штраф**

**Функция использует стратегию формирующего награждения, которая учитывает высоту агента. Линейные штрафы за расстояние и угол направляют агента к целевой точке и ориентации. Увеличение штрафа за скорость при приближении к земле способствует мягкой посадке. Бонусы и штрафы за завершение эпизода усиливают желаемое поведение и наказывают за неудачи. Использование линейных штрафов вместо квадратичных может привести к более стабильному и предсказуемому обучению, особенно на начальных этапах.**

**Ключевые особенности и отличия от предыдущих версий:**

* **Основное отличие заключается в том, что функция \*усиливает\* штраф за скорость, когда агент находится на определенной высоте. Это побуждает агента замедлиться перед посадкой, что критически важно для безопасного и успешного приземления.**
* **Функция использует линейные штрафы за расстояние, угол и скорость, в отличие от квадратичных штрафов в некоторых предыдущих версиях. Линейные штрафы могут быть более стабильными и менее чувствительными к выбросам в данных, что может способствовать более плавному обучению.**
* **Функция имеет более простую и прямолинейную структуру по сравнению с более сложными стратегиями, такими как явное наказание за крушения или награда за выживание. Это может сделать ее легче для понимания и отладки.**

**В заключение, эта функция награды направлена на обучение агента безопасной и точной посадке, акцентируя внимание на замедлении скорости при приближении к земле. Она представляет собой более простой и, возможно, более стабильный подход к формированию желаемого поведения по сравнению с более сложными стратегиями награждения. Результат обучения показан на рисунке 5. Для обучения потребовалось 850 эпизодов.**



**Рисунок 5 – Результаты обучения по зависящей от высоты функции награды**

### ****2.1.5 Разреженная награда****

**Эта функция награды реализует стратегию редкого награждения (sparse reward), при которой агент получает значительную награду только при выполнении целевой задачи (успешная посадка) или наказание при неудаче. В отличие от функций с формирующим награждением, здесь отсутствует постепенное направление агента к цели через промежуточные награды. Основная цель - заставить агента исследовать среду и самостоятельно обнаружить способ успешной посадки, полагаясь в основном на случайные действия. Функция реализует стратегию редкого награждения, которая характеризуется:**

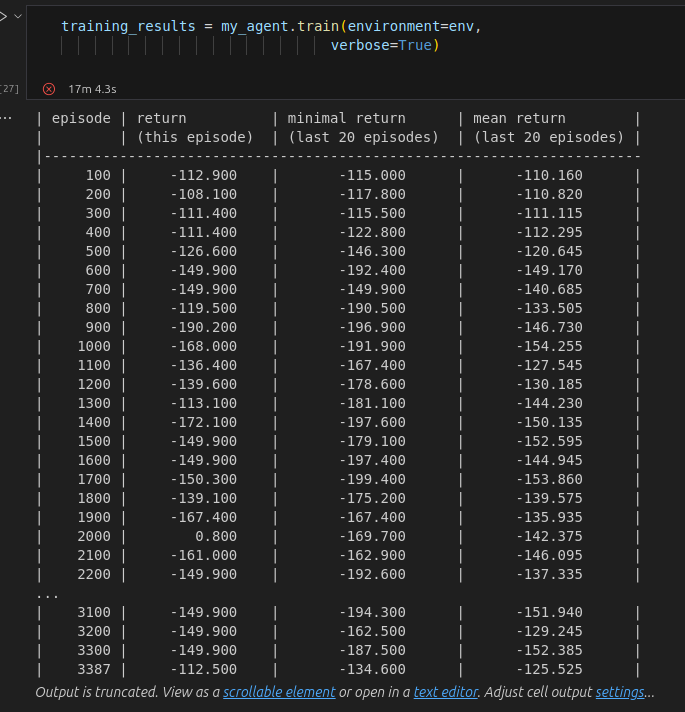
**Минимальным формирующим награждением: Агенту не предоставляется никакой информации о том, как продвигаться к цели, кроме как через итоговый результат (успешная посадка или неудача).**

* **Большими, но редкими наградами: Бонус за успешную посадку и штраф за падение имеют значительную величину, чтобы привлечь внимание агента к этим исходам.**
* **Штрафом за бездействие: Небольшой штраф за каждый шаг, когда эпизод не завершается, мотивирует агента к активным действиям.**

**Такая стратегия может быть полезна в задачах, где трудно определить промежуточные цели или где экспертные знания ограничены. Однако, она может быть очень сложной для обучения, так как агенту необходимо случайно наткнуться на решение (успешную посадку), прежде чем начать улучшать свою стратегию.  
Ключевые особенности и отличия от предыдущих версий:**

* **Отсутствие формирующего награждения. Нет штрафов за расстояние, угол или скорость. Агент получает награду только в конце эпизода.**
* **Перезапись, а не добавление: Функция \*перезаписывает\* значение награды, а не добавляет к нему штрафы или бонусы.**
* **Сильный акцент на исследовании: Такая стратегия подразумевает, что агент должен в основном полагаться на случайное исследование, чтобы найти успешные стратегии.**
* **Потенциальная сложность обучения: Обучение с редким награждением может быть гораздо сложнее и требовать больше вычислительных ресурсов и времени, чем с формирующим награждением.**

**Результаты обучения показаны на рисунке 6. За 3300 эпизодов агент не смог найти оптимальную по награде стратегию.**



**Рисунок 6 – Результаты обучения по разреженной функции награды**

**В заключение, эта функция награды использует стратегию редкого награждения, что может быть эффективным подходом в задачах, где трудно или нежелательно предоставлять агенту подробную информацию о том, как достичь цели. Однако, этот подход может быть сложным для обучения и требует от агента значительного количества исследований. Она подходит для ситуаций, когда желаемый результат четко определен, но путь к нему неизвестен или не может быть легко сформулирован в виде промежуточных целей.**

# ****ЗАКЛЮЧЕНИЕ****

**Функция награды DRL оказывает значительное влияние на процесс обучения в глубоком обучении с подкреплением (DRL). Она является единственным механизмом обратной связи, который агент получает от окружающей среды, и определяет, какое поведение считается "хорошим", а какое - "плохим". В сущности, функция награды определяет цель обучения агента. Функция награды напрямую формирует поведение агента. Правильно разработанная функция награды направляет агента к желаемому поведению, поощряя полезные действия и наказывая нежелательные. Качество функции награды сильно влияет на скорость и стабильность обучения. Четко определенная и информативная функция награды позволяет агенту быстрее находить оптимальную стратегию и избежать проблем с расходимостью обучения. Неправильно сформулированная функция награды, в свою очередь, может привести к тому, что агент будет оптимизировать поведение, отличное от задуманного (несоответствие цели). Это может привести к опасным решениям. Например, агент может найти ошибку в системе награждения и использовать её для получения высокой награды, не решая при этом реальную задачу. Принципы, лежащие в основе проектирования эффективной функции награды, могут существенно отличаться в зависимости от конкретной задачи. То, что хорошо работает в одной среде, может оказаться контрпродуктивным в другой.**

**Таким образом, функция награды является краеугольным камнем DRL. Её разработка требует тщательного обдумывания и понимания как самой задачи, так и потенциальных последствий различных схем награждения. Недостаточно продуманная функция награды может не только замедлить обучение, но и привести к нежелательному или даже вредному поведению агента. Разработка эффективных функций награды является одной из ключевых областей исследований в DRL.**

# ****СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ****

1.Ultralytics. (n.d.). \*Deep Reinforcement Learning\*. Ultralytics Glossary. Режим доступа: свободный <https://www.ultralytics.com/ru/glossary/deep-reinforcement-learning> , дата обращения 02.06.2025.

2.Li, Y. (2018). \*Deep reinforcement learning: An overview\*. arXiv preprint arXiv:1701.07274. Режим доступа: свободный <https://arxiv.org/abs/1701.07274> дата обращения 02.06.2025.

3.Farama Foundation. (n.d.). \*LunarLander-v2\*. Gymnasium Documentation. Режим доступа: свободный <https://gymnasium.farama.org/environments/box2d/lunar_lander/> , - дата обращения 02.06.2025.

4.Mangasarian, O. L. (1966). Sufficient conditions for the optimal control of nonlinear systems. \*SIAM Journal on Control, 4\*(1), 139-152. https://doi.org/10.1137/0304013

5.Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. \*Nature, 518\*(7540), 529-533.