***Алгоритм обучения с подкреплением***

Обучение с подкреплением — это тип машинного обучения, при котором агент учится принимать решения, взаимодействуя со средой. Агент предпринимает действия в среде для достижения цели, и он получает обратную связь в виде поощрений или наказаний.

# [**Введение в обучение с подкреплением**](https://habr.com/ru/companies/otus/articles/429090/)

Почему нам это подходит?

Обучение с подкреплением обеспечивает адаптивность и гибкость роботов. Они могут обучаться и адаптироваться к изменениям в среде, новым задачам или неожиданным ситуациям. В задаче прыжка однозначного “ответа” нет, поэтому обучение с подкреплением в этом кейсе играет важную роль.

**Подход к обучению. Curriculum**

Curriculum learning — это методология обучения, в которой данные представляются, начиная с более простых экземпляров и постепенно переходя к более сложным.

[What is Curriculum Learning (activeloop.ai)](https://www.activeloop.ai/resources/glossary/curriculum-learning/)

Будем использовать её, тк задача легко разбивается на этапы, упорядоченные по сложности:

1. Научить собаку прыгать на месте
2. Научить собаку прыгать вперед
3. Научить прыгать выше, чем на предыдущем этапе.

**Симуляция среды. Физические движки:**

***Mujoco (Multi-Joint dynamics with Contact)***

[MuJoCo Overview (roboti.us)](https://www.roboti.us/book/index.html)

Плюсы:

- Высокая физическая точность симуляций

- Эффективные вычисления и высокая производительность

- Гибкость в создании собственных моделей

Минусы:

- Ограниченная визуализация и графика (из-за большей физической точности)

***Pybullet***

[Обучение с подкреплением в среде PyBullet / Хабр (habr.com)](https://habr.com/ru/articles/420897/)

Плюсы:

- Бесплатная открытая лицензия

- Простая интеграция с Python

- Возможность переноса на реальную робототехнику

Минусы:

- Менее точная физика по сравнению с MuJoCo

- Ограниченная визуализация и графика

***Isaac Gym***

[Isaac Gym - Preview Release | NVIDIA Developer](https://developer.nvidia.com/isaac-gym)

Плюсы

* Использование GPU: Isaac Gym оптимизирован для работы на графических процессорах (GPU), что позволяет добиться высокой производительности и масштабируемости при симуляции сложных физических сред.
* Масштабируемость: Благодаря использованию GPU, Isaac Gym может симулировать большое количество объектов и агентов одновременно, что особенно полезно для задач, требующих параллелизма, таких как обучение с подкреплением.
* Интеграция с PyTorch: Isaac Gym тесно интегрирован с библиотекой PyTorch, что упрощает совместное использование с моделями машинного обучения.

Минусы: не такая высокая физическая точность как у mujoco.

**Итог:** из движков можем использовать MuJoCo, поскольку благодаря нему обеспечивается высокая точность физических симуляций, также этот продукт является бесплатным с 2014 года, но **также** нам подходит Isaac Gym, поскольку он требует наличия графического процессора NVIDIA и может показывать лучшую производительность.

**Среды симуляции:**

***Unity ML-Agents***

[Машинное обучение агентов в Unity / Хабр (habr.com)](https://habr.com/ru/articles/454612/)

Плюсы:

- Богатые возможности визуализации благодаря Unity

- Простота создания реалистичных 3D сред

- Интеграция с Unity Asset Store (множество готовых ресурсов)

Минусы:

- Менее точная физика по сравнению с MuJoCo

- Высокие системные требования для сложных симуляций

***Microsoft AirSim***

[Home - AirSim (microsoft.github.io)](https://microsoft.github.io/AirSim/)

Плюсы:

- Богатые возможности визуализации благодаря Unity

- Простота создания реалистичных 3D сред

- Интеграция с Unity Asset Store (множество готовых ресурсов)

Минусы:

- Менее точная физика по сравнению с MuJoCo

- Высокие системные требования для сложных симуляций

***Legged Gym***

[Isaac Gym Environments for Legged Robots (pythonrepo.com)](https://pythonrepo.com/repo/leggedrobotics-legged_gym-python-programming-with-hardware)

Плюсы:

* Специализация: Legged Gym сфокусирован на симуляции ходьбы и передвижения робототехнических систем с ногами, что делает его отличным инструментом для разработки и обучения алгоритмов управления такими системами. Это плюс для нашего кейса.
* Разнообразие сред: Legged Gym предоставляет широкий спектр сред, включая различные типы поверхностей, препятствия и внешние воздействия, что позволяет обучать и тестировать алгоритмы в разнообразных условиях.
* Интеграция с OpenAI Gym: Legged Gym построен на базе популярной среды OpenAI Gym, что облегчает его использование и интеграцию с существующими алгоритмами обучения с подкреплением.
* Открытый исходный код: Legged Gym распространяется с открытым исходным кодом, что позволяет пользователям вносить изменения, расширять его функциональность и адаптировать под свои нужды.

Минусы:

* Потенциальные проблемы с производительностью: В зависимости от используемого физического движка и сложности симулируемых сцен, Legged Gym может столкнуться с проблемами производительности при симуляции большого количества агентов или сложных сред.

**Итог:** для обучения робота-собаки наиболее подходящим инструментом будет Legged Gym, поскольку он разработан для симуляции и обучения роботов с конечностями. Также Legged Gym может использовать физические движки, такие как PyBullet или MuJoCo, что обеспечит высокую физическую точность.

**Tensorboard**

TensorBoard - это инструмент визуализации данных и анализа производительности, который тесно интегрирован с TensorFlow - популярной библиотекой для машинного обучения и глубокого обучения. TensorBoard позволяет легко визуализировать и анализировать различные метрики, графики вычислений, графики гиперпараметров.

[**«Cделать красиво». Визуализация обучения с Tensorboard от Google / Хабр (habr.com)**](https://habr.com/ru/articles/349338/)

Этот инструмент можно использовать для отслеживания метрик обучения, построения графиков. Так мы сможем понять, когда модель успешно обучилась.

**PPO**

[Архитектуры RL: DDPG иPPO / Хабр (habr.com)](https://habr.com/ru/companies/otus/articles/771412/)

PPO является алгоритмом усиления обучения с политикой (policy-based), который используется для обучения агентов в задачах с непрерывным пространством действий и состояний. Основная идея PPO заключается в том, чтобы обновлять политику агента (нейронную сеть, которая выбирает действия) с помощью градиентного спуска, при этом ограничивая изменения политики относительно предыдущей версии.

*Алгоритм PPO работает следующим образом:*

1. Агент взаимодействует со средой, собирая траектории (последовательности состояний, действий и вознаграждений).

2. Эти траектории используются для вычисления функции потерь (loss function), которая представляет собой суррогатную функцию для максимизации ожидаемого вознаграждения.

3. Функция потерь состоит из двух компонентов: потери политики (policy loss) и потери ценности (value loss).

- Потеря политики стремится максимизировать ожидаемое вознаграждение, но при этом ограничивает изменения политики относительно предыдущей версии с помощью коэффициента отсечения (clipping).

- Потеря ценности минимизирует ошибку между предсказанной функцией ценности (оценкой качества состояния) и фактическими вознаграждениями, полученными агентом.

4. Нейронная сеть агента обновляется с помощью градиентного спуска, минимизируя функцию потерь.

5. Процесс повторяется для нескольких эпох обучения, пока агент не достигнет удовлетворительной производительности.

*Основными преимуществами PPO являются:*

- Стабильность и надежность обучения по сравнению с другими алгоритмами усиления обучения.

- Возможность обучения в средах с непрерывным пространством действий и состояний.

- Параллельное взаимодействие агента с множеством копий среды, что ускоряет сбор данных.

**Характеристики нашего робота**

[Собака-робот за 1 миллион рублей (Unitree Robotics A1, обзор) / Хабр (habr.com)](https://habr.com/ru/companies/parallels/articles/558786/)



*Статья от IT-центра МАИ:*

[Сказ про робота Unitree A1 — Часть 1: Базовые возможности / Хабр (habr.com)](https://habr.com/ru/companies/itmai/articles/788936/)

В этой статье описано, что может делать собака и как с ней работать.

Видео с нашей собакой :3

[doggo4 - YouTube](https://www.youtube.com/watch?v=vEtpFmvaF24)

Инструкция по использованию робота-собаки:

https://www.mybotshop.de/Datasheet/Unitree\_A1\_User\_Manual\_v1.0.pdf

*Рассматриваемые программные зависимости*

Среды симуляции: Unity ML-Agents, Microsoft AirSim, Legged Gym.

Физические движки: Mujoco (Multi-Joint dynamics with Contact), Pybullet, Isaac Gym.

Библиотеки машинного обучения: TensorFlow, PyTorch;

Инструмент для визуализации и отладки: TensorBoard.

Подход к обучению: Curriculum (обучение по учебной программе).

Алгоритм обучения с подкреплением: PPO (Proximal Policy Optimization).

Необходимые файлы с примерами обучения прыжкам можно найти на гитхабе:

<https://github.com/inspirai/MetalHead>

Видео со средой симуляции и прыжками Доры:

<https://www.youtube.com/watch?v=IdzfE9rXoqY>

Файлы обучения робота-собаки nitree go1:

<https://github.com/dexfrost89/legged_gym/tree/master>