Введение в обучение с подкреплением

# Тема 1: Базовые понятия RL

Лектор: Кривошеин А.В.

# Парадигмы машинного обучения

**Машинное обучение (МО)** — это инструмент для поэтапной автоматизации решения интеллектуальных задач с помощью компьютера.

Есть три основных парадигмы МО.

1. **Обучение с учителем (англ. Supervised Learning)**: обобщение известного опыта (например, задачи классификации, регрессии, ранжирования).

2. **Обучение без учителя (англ. Unsupervised Learning)**: автоматическое извлечение закономерности из данных (например, задачи кластеризации, снижение размерности пространства признаков).

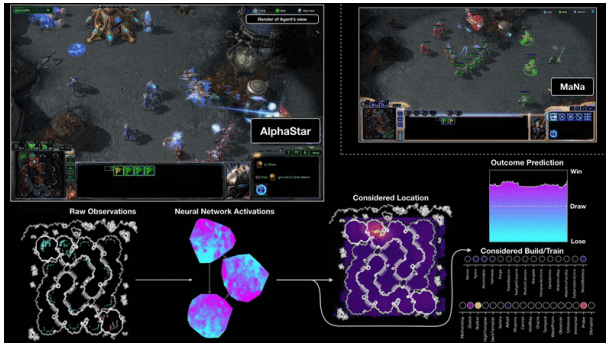
3. **Обучение с подкреплением (англ. Reinforcement Learning, RL)**: моделирование человекоподобного поведения (например, обучение игре в шахматы, го, Starcraft и пр., управление роботами, в частности, беспилотным транспортом, роботом-манипулятором и др.)

# Приложения обучения с подкреплением

**AlphaGo:** программа для игры в го от компании Google DeepMind, которая в 2016 году обыграла со счётом 4:1 одного из сильнейших игроков в го Ли Седоля.

AlphaGo обучалась на записанных человеческих партиях.
Улучшенная версия AlphaGo Zero обучается игре в го в ходе игр с собой. После порядка 5 млн партий AlphaGo Zero всухую обыгрывает AlphaGo.

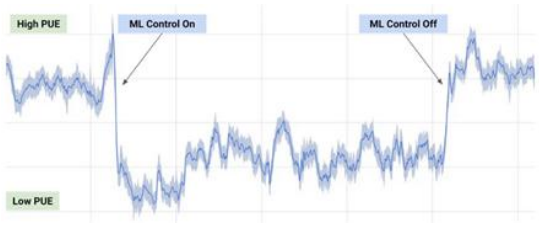
Фильм о AlphaGo: https://www.youtube.com/watch?v=WXuK6gekU1Y

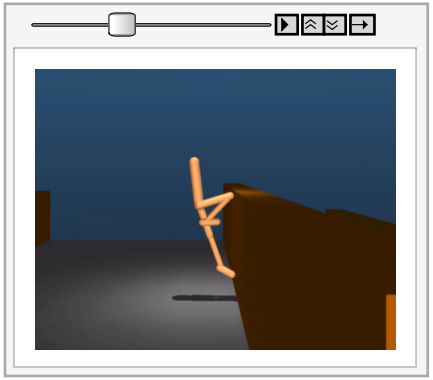




Боты для видео-игр: 2D игры от Atari, Quake III, Dota 2 (OpenAI Five, 2017), Starcraft 2 (AlphaStar, 2019)

# Приложения обучения с подкреплением

**Управление микро-климатом.** Алгоритмы охлаждения дата-центров компании Google на основе RL, оценивающие систему охлаждения серверов и предсказывающие наиболее эффективные параметры для поддержания стабильной температуры и максимального энергосбережения (снижение энергопотребления до 40%).



**Обучение роботов** различного назначения: роботы-манипуляторы со сложным поведением, движущиеся роботы, автопилоты.

**Рекомендательные системы** на основе RL. Такого типа системы, например, используются для рекомендации товаров на сайтах Alibaba Group или для рекомендаций видео на YouTube.

**AlphaFold:** программа от Google DeepMind, которая выполняет предсказания пространственной структуры белка.
**AlphaGeometry:** программа от Google DeepMind, которая может решать олимпиадные задачи по геометрии.
**DeepSeek-R1:** это языковая модель с элементами логических рассуждений

# Базовые понятия и суть RL

Ключевые понятия:

окружающая **среда** (англ. **environment**)

действующий в среде **агент** (обучаемый субъект, англ. **agent**)

Агент может предпринимать **действия** (англ. **action**), влияющие на среду, и принимать отклик от среды о своих действиях.

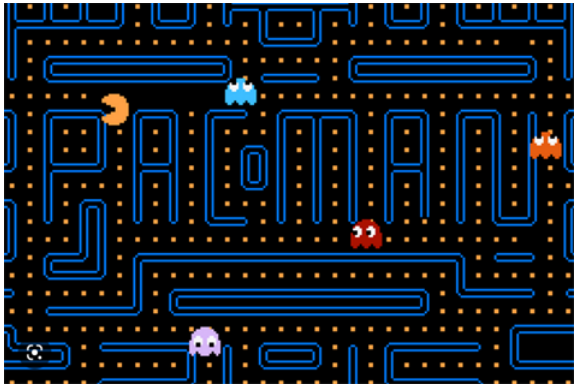
Перед агентом ставится **цель,** которая заключается в решении той или иной прикладной задачи.

Действия, которые способствуют достижению цели будем называть “хорошими” действиями,

действия, которые препятствуют достижению цели, — “плохие” действия.

**Суть RL:**

*научить обучаемого агента предпринимать "хорошие" последовательности действий в ходе взаимодействия со средой для эффективного достижения цели.*



# Базовые понятия и суть RL

**Агент = Алгоритм + Устройства** (которые совершают действия и принимают отклик от среды).

Агент ничего не знает о том, что мы с его “помощью” решаем какую-то внешнюю для него задачу.

Подход RL заключается в том, чтобы сформировать систему **вознаграждений** (англ. **reward**) за действия агента.

Агент получает

положительные вознаграждения за “хорошие” действия,

отрицательные вознаграждения за “плохие” действия.

Суммарное полученное агнетом вознаграждение в ходе взаимодействия со средой называют **доходом** (англ. **gain, return**).

Вознаграждения формируются так, что

задача **достижения цели** агентом = задача **максимизации** дохода

Как агенту понять какие действия “хорошие”? Основной подход заключается в следующих шагах:

1. исследование **методом проб и ошибок** (англ. **exploration**) действующим агентом окружающей среды;
2. анализ полученного опыта и формированием оптимальной **стратегии** (англ. **policy**) поведения;
3. **использование** (англ. **exploitation**) стратегии для получения максимального дохода.

Стратегия поведения агента — это набор правил о том, какие действия агенту выбирать в различных ситуациях.

Цель агента: *поиск стратегии поведения, дающей максимальный доход.*

# Отличие от других парадигм МО

При обучении с\без учителя обучаемый алгоритм будем также называть агентом.

**При обучении с учителем:**

Обучающая информация — это **инструкции** о том, какие действия правильные. Обучающий датасет должен быть сформирован до обучения и не зависит от действий агента.

**При обучении с подкреплением:**

Обучающая информация — это **оценка агентом** сделанных действий.

Действия влияют

на ближайшее вознаграждение,

на следующее состояние среды и вознаграждение в нём,

и так далее на все следующие вознаграждения.

Этот факт называют **отложенным вознаграждением**. Правильность тех или иных действий определяется при анализе цепочки вознаграждений.

До обучения может **не быть никаких инструкций** о том, какие действия правильные.

У обучения без учителя и RL различны конечные цели обучения.

**Обучение без учителя:** нахождение структуры в неразмеченных данных;

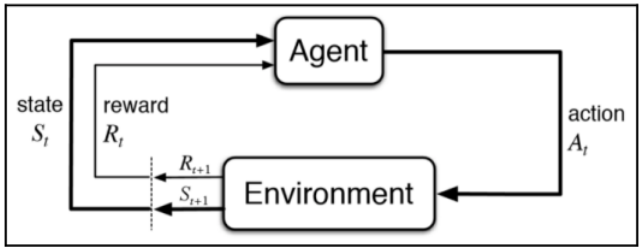
**RL:** формирование оптимальной стратегии поведения.

Для описанных выше примеров задач практически невозможно составить классический алгоритм решения и только подход на основе RL позволил получить решение этих задач.

# Взаимодействие агента со средой

**Схема взаимодействия агента со средой:**

обучаемый агент действует в окружающей его среде, среда реагирует на действия агента.




Предполагаем, что взаимодействие агента со средой происходит в дискретные моменты времени.

1. В момент времени $t$, находясь в текущем состоянии среды $S_t$, агент производит действие $A_t$.

2. В момент времени $t + 1$ агент получает вознаграждение $R_{t+1}$, среда переходит в новое состояние $S_{t+1}$. В этом новом состоянии агент делает новое действие $A_{t+1}$ и т.д.

Под **состоянием среды** (англ. **state**) мы будем понимать доступную для агента информацию об этой среде, которая важна для достижения целей агента.

# Взаимодействие агента со средой

Общие обозначения:

$\mathcal{S}$ — множество состояний,

$\mathcal{A}$ — множество действий,

$\mathcal{R}$ — множество вознаграждений.

Взаимодействие агента со средой удобно записывать в виде последовательности случайных величин, то есть с помощью **дискретного случайного процесса**:

$S_0, A_0, R_1, S_1, A_1, \ldots$

Конкретную реализацию этого процесса называют **траекторией** взаимодействия агента со средой.

Для траектории можно использовать те же самые обозначения $S_0, A_0, R_1, S_1, A_1, \ldots$, понимая под этими символами конкретные состояния, действия, вознаграждения в текущем взаимодействии агента со средой.

Взаимодействие агента со средой по длительности является

либо не ограниченным по времени: $S_0, A_0, R_1, S_1, A_1, \ldots$

либо конечным по длительности: $S_0, A_0, R_1, S_1, A_1, \ldots, R_{T-1}, S_{T-1}, A_{T-1}, R_T, S_T$.

В последнем случае

либо мы искусственно ограничиваем сверху время взаимодействия агента со средой,

либо же среда имеет терминальные состояния, в которых взаимодействие агента со средой прекращается.

Конечный временной период, в течении которого агент действует в среде до завершения взаимодействия, называется **эпизодом**, а сам процесс, описывающий взаимодействие агента со средой называют **эпизодическим**.

# Пример

Out[ ]//DisplayForm=


Проиллюстрируем описанные выше понятия
на примере игры в шахматы.

Состояние среды $S_t$ — это текущее положение фигур на доске.

Действие агента $A_t$ — это ход той или иной фигурой.

Новое состояние $S_{t+1}$ — это положение после хода противника.

$\mathcal{S}$ — множество всевозможных положений фигур на доске,

$\mathcal{A}$ — множество всевозможных ходов, причём
для конкретного состояния $S_t$ есть свой набор допустимых ходов $\mathcal{A}(S_t)$.
Взаимодействие агента со средой эпизодическое.

Вознаграждения можно формировать различными способами.
1. Простейший способ заключается в том, чтобы дать большое вознаграждение за победу и штраф за проигрыш. Однако, обычно при начальном взаимодействии со средой агент совершает действия случайным образом, поскольку у него нет каких либо априорных знаний о среде. Добиться положительного вознаграждения, то есть победы, при случайных ходах практически невозможно. А значит обучение будет чрезвычайно долгим.

2. Более разумным подходом к формированию системы вознаграждений будет следующий:

за ход, ведущий к взятию фигуры противника, можно давать положительное вознаграждение,

за ход, ведущий к потере своей фигуры, — отрицательное вознаграждение.

Причём в зависимости от значимости фигуры можно варьировать модуль вознаграждения. Также можно ввести вознаграждения за шах и мат.

# Основные концепции RL: стратегия

Рассмотрим подробно основные концепции обучения с подкреплением, относящиеся к агенту:

1) стратегия,
2) вознаграждения и доход,
3) функция ценности состояний,
4) модель среды.

**Стратегия** (англ. policy) — набор правил, определяющих поведение агента в любом состоянии.

Детерминированная стратегия — это отображение $\pi$ из множества состояний $\mathcal{S}$ в множество действий $\mathcal{A}$, то есть $\pi : \mathcal{S} \to \mathcal{A}$.

Стохастическая стратегия – это отображение из множества состояний $\mathcal{S}$ в набор вероятностей выбора действий из множества $\mathcal{A}$. В этом случае будем использовать обозначение $\pi(a|s)$ для вероятности выбора действия $a \in \mathcal{A}$ в состоянии $s \in \mathcal{S}$, причём

$\pi(a|s) \in [0, 1]$ и $\sum_{a \in \mathcal{A}} \pi(a|s) = 1$.

# Основные концепции RL: вознаграждение и доход

**Вознаграждение** —это то число, которое получает агент после совершения действия в среде.

**Детерминированное** вознаграждение — это функция $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

То есть вознаграждение $r(s, a)$ зависит от текущего состояния среды $s$ и от выбранного действия $a$.

Для конкретной траектории взаимодействия: $R_t = r(S_{t-1}, A_{t-1})$.

**Стохастическое** вознаграждение — это случайная величина, распределение которой зависит от текущего состояния среды $s$ и от выбранного действия $a$.

**Доход**, полученный после совершения действия в момент времени $t$, обозначается $G_t$ и его обычно определяют по формуле

$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots,$

где $\gamma \in [0, 1]$ **коэффициент обесценивания** (англ. discount factor). Коэффициент $\gamma$ отражает баланс между влиянием на общий доход ближайшего вознаграждения и будущих вознаграждений за совершённые действия.

Если $\gamma = 0$, то $G_t = R_{t+1}$ и максимизация дохода — это максимизация ближайшего вознаграждения.

Если $\gamma = 1$, то будущие вознаграждения одинаково важны для общего дохода.

При эпизодическом взаимодействии выбор $\gamma < 1$ стимулирует агента быстрее завершить эпизод.

Например, если успешное завершение эпизода даёт большое вознаграждение, то

завершение эпизода через 10 ходов прибавит к доходу вознаграждение с коэффициентом $\gamma^{10}$,

а завершение через 1000 ходов — с коэффициентом $\gamma^{1000}$.

# Основные концепции RL: стратегия, максимизирующая доход

Доход агента с начала взаимодействия — это

$$G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + \ldots = \sum_{t=0}^{\infty} \gamma^t R_{t+1}$$

Цель агента —это максимизация дохода при взаимодействии со средой. Формализуем эту цель.

Агент действует по некоторой стратегии $\pi$. Введём следующую функцию, позволяющую оценить стратегию $\pi$ с точки зрения получаемого дохода:

$J(\pi) = \mathbb{E}_\pi[G_0]$.

Эта величина означает ожидаемый доход при взаимодействии со средой агента, действующего стратегии $\pi$.

Если сгенерировать множество траекторий взаимодействия со средой для агента, действующего стратегии $\pi$, и по каждой траектории найти полученный доход, то усреднение этих доходов является приближением величины $J(\pi)$.

**Цель агента** — это поиск **оптимальной стратегии** $\pi_*$, **максимизирующей** величину $J(\pi)$, то есть

$\pi_* = \underset{\pi}{\operatorname{argmax}} J(\pi)$.

# Основные концепции RL: функция ценности

**Функция ценности состояния** (англ. state value function) или $V$-функция — это функция $v_\pi : \mathcal{S} \to \mathbb{R}$, выражающая насколько хорошо для агента нахождение в данном состоянии с точки зрения длительной перспективы при заданной стратегии поведения $\pi$.

Формально, ценность состояния $s$ определяется как математическое ожидание дохода $G_t$ при старте из состояния $S_t = s$ и следовании стратегии $\pi$:

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s].$$

Аналогичным образом определяется **функция ценности состояния-действия** или $Q$-функция. Это функция $q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, значения которой определяются в виде математического ожидания дохода $G_t$ при старте из состояния $S_t = s$, осуществления действия $A_t = a$ и затем следовании стратегии $\pi$:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s,\ A_t = a].$$

Ряд методов поиска оптимальной стратегии основан именно на функциях ценности. Далее будет ясно, что оптимальная стратегия — это выбор тех действий, которые приводят в состояния с наибольшей ценностью.

# Основные концепции RL: модель среды

**Модель среды** — это набор правил, имитирующий поведение среды в ответ на действия агента.

В реальных задача модель среды, как правило, не доступна.

В простейшем случае модель среды выдаёт состояние $S_{t+1}$, если сейчас состояние $S_t$ и агент совершил действие $A_t$.

Вообще говоря, каждое новое состояние среды может зависеть от всех прошлых состояний и действий агента.

Пусть $h = (S_0, A_0, \ldots, A_{t-1}, S_t)$ траектория взаимодействия агента со средой до момента времени $t$.

Вероятность перехода в новое состояние $S_{t+1} = s'$ при совершении действия $A_t = a$ обозначим за

$$\mathbb{P}(S_{t+1} = s' | H_t = h, A_t = a).$$

Работа с моделью в такой постановке сложна, при формировании стратегии надо учитывать все прошлые шаги.

На практике полагают, что среда **марковская**, то есть

**вероятность нового состояния** среды $S_{t+1} = s'$ при текущем состоянии среды $S_t = s$ и действии агента $A_t = a$ **не зависит от предыдущих состояний и действий:**

$$\mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a) = \mathbb{P}(S_{t+1} = s' | H_t = h, A_t = a).$$

В марковском случае задать модель среды — это значит задать вероятности

$$\mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a), \text{ для всех возможных } s', s \in \mathcal{S}, a \in \mathcal{A}.$$

Как правило, марковость среды всегда можно обеспечить, если определять состояние среды $s$ как информацию обо всех аспектах прошлых взаимодействий агента со средой, которые важны для будущего.

# Основные концепции RL: модель среды

Если **модель среды есть**, то агент может заранее **спланировать оптимальную стратегию** действий.

Надо решить некоторую задачу оптимизации.

При этом агенту не нужно даже взаимодействовать со средой.

Если же **модели среды нет**, то агент **обучается в ходе взаимодействия со средой**.

Базовый подход — это метод проб и ошибок с анализом полученного опыта, в ходе которого формируются оценки для функции ценности состояний и постепенно улучшается стратегия поведения. При необходимости модель среды может быть построена по ходу обучения.

**Основные методы решения задач RL на основе функций ценности:**

1. методы динамического программирования (применяются при наличии полной модели среды),

2. методы проб и ошибок, методы Монте-Карло (могут применяться, когда модели среды нет, а задача разбивается на эпизоды, тогда агент может обучаться после каждого эпизода),

3. методы на основе временных различий или TD-методы (англ. Time Difference, могут также применяться, когда модели среды нет, обучение может происходить после каждого действия).

Также, ряд методов основан на непосредственной максимизации целевой функции $J(\pi)$.

# Общая структура курса

1. Многорукие бандиты
2. Марковский процесс вознаграждения
3. Марковский процесс принятия решений
4. Методы Монте-Карло
5. TD методы
6. Методы, основанные на приближении функций
7. Deep Q Network и модификации
8. Метод REINFORCE и его модификации, метод Актор-Критик A2C
9. Метод PPO, DDPG