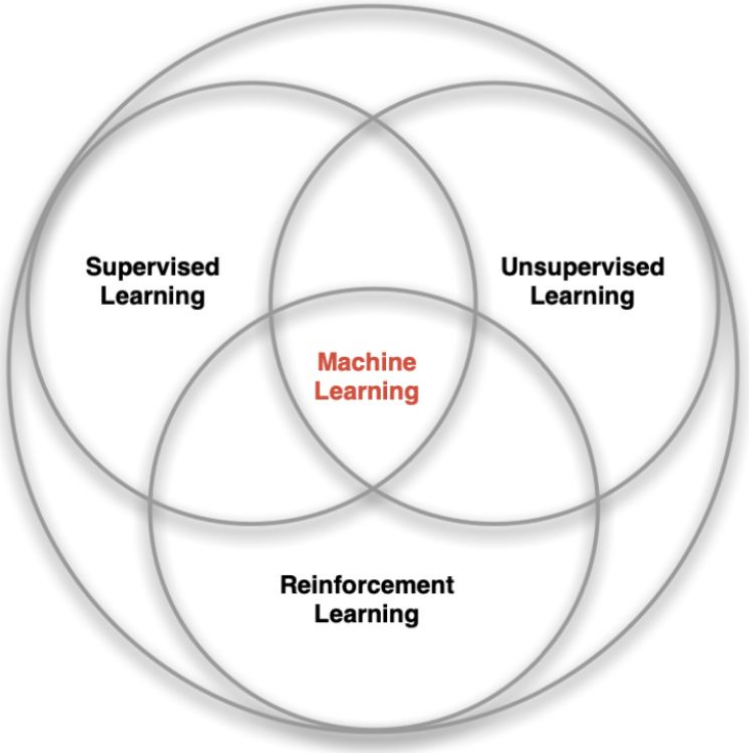
Разработка интеллектуальных агентов компьютерных игр

Веселко Никита Игоревич

- CEO и Lead Product Owner студии Винторог (ex IThub games) и студии Contrast Games
- CPO и Co-Founder в Current Agency Development
- CEO в HashCats
- Аспирант ФКН НИУ ВШЭ
- Преподаватель НИУ ВШЭ и школы Нарраторика
- Соавтор книги “Программирование в Unreal Engine 5”

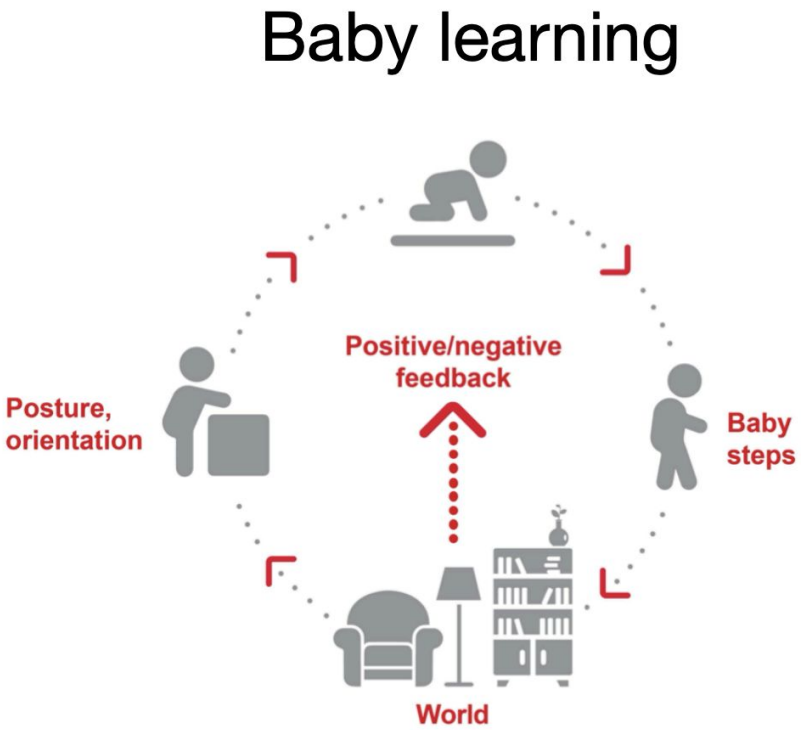
Reinforcement Learning in Machine Learning



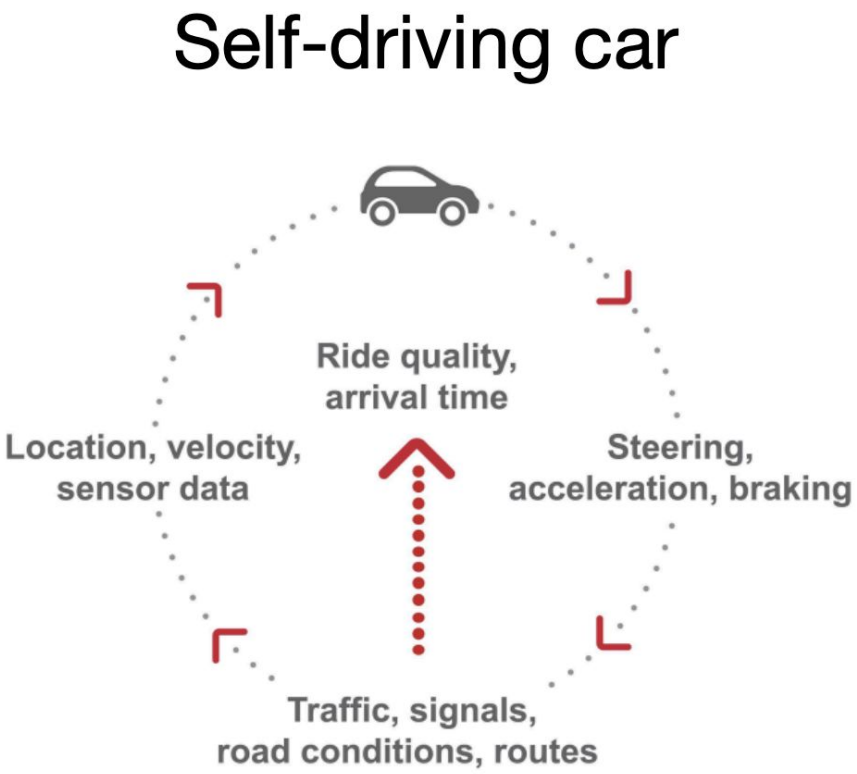
Supervised Learning

- Train sample: X, Y
- Approximation $Y \approx \hat{y} = f(X)$, f - семейство алгоритмов
- Loss minimization: $L(Y, \hat{y}) \to \min$ по f

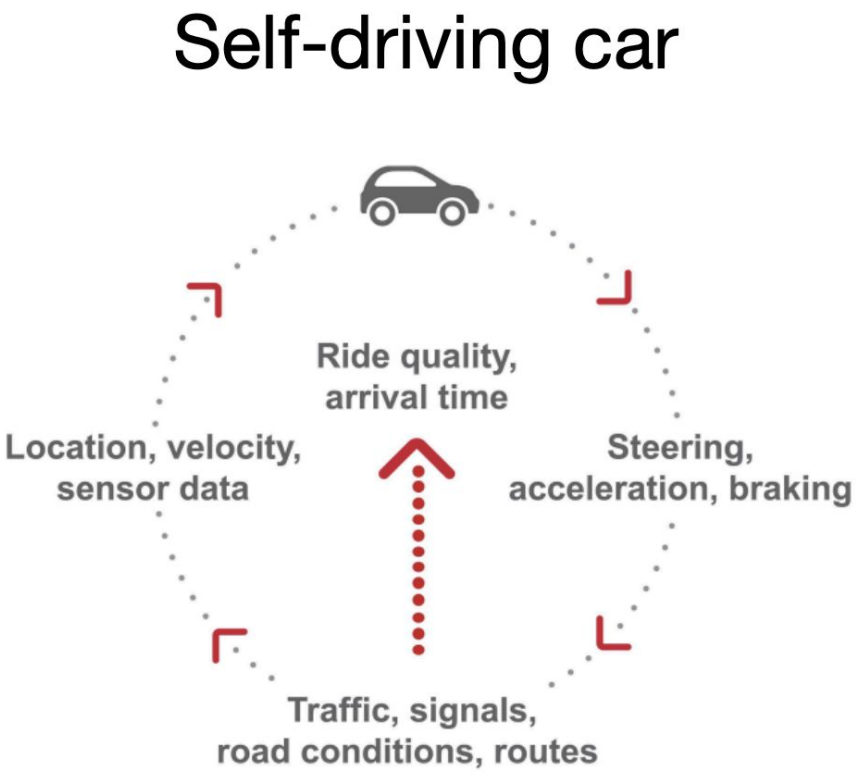
Sequential Decision Making



Sequential Decision Making



Sequential Decision Making



RL

- Отображений ситуации на действия, максимизируя численный сигнал - вознаграждения
- Агент сам должен сам понять, какое действие в какой ситуации приносит максимальную награду
  - В сложных моделях понять, как действие повлияет и на награды за следующие действия
- Восприятие, действие и цель

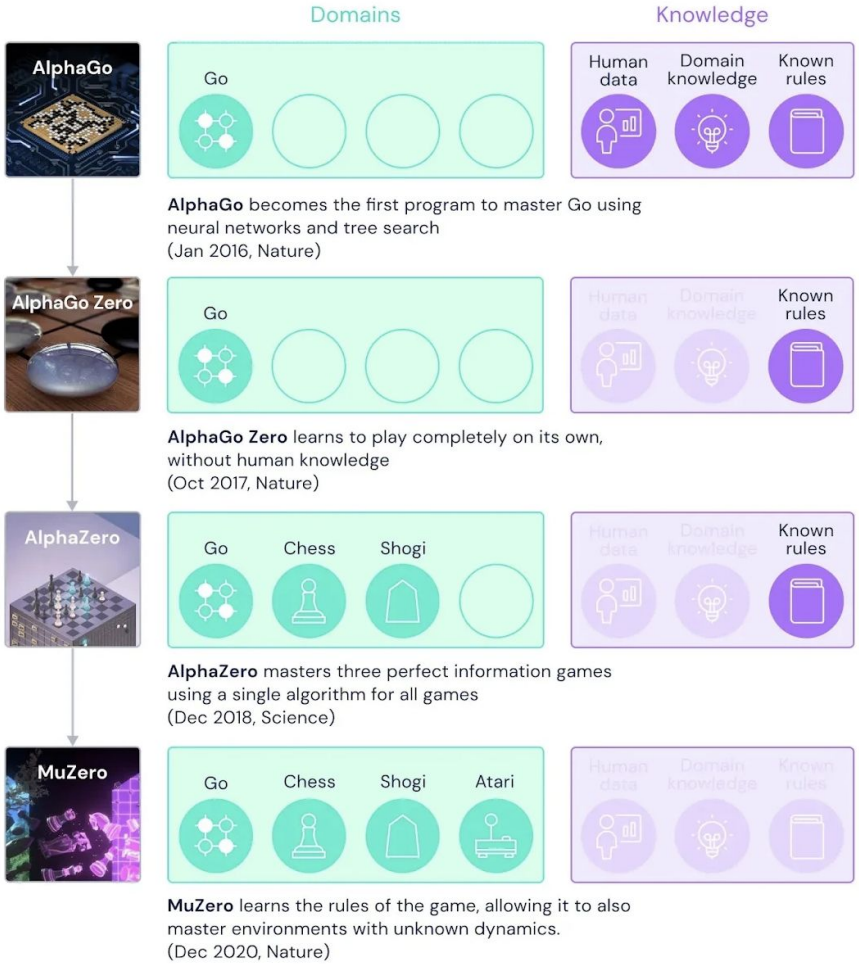
RL vs (Un)Supervised Learning

- Supervised Learning
  - Есть обучающая размеченная выборка с метками правильного действия под каждую ситуацию
  - Модель экстраполирует или обобщает свою реакцию на ситуации, которые не предъявлены в обучающей выборке
  - Не подходит для обучения с помощью взаимодействия, так как интерактивных задачах нельзя получить примеры желаемого поведения
- Upsupervised Learning
  - Цель: обнаружить структуру, скрытую в наборе данных, а не максимизировать награду
- RL
  - Компромисс между исследованием и использованием
  - Рассмотрение целостной проблемы
    - В Sup не ставится вопрос о конечной пользе приобретенных способностей обобщения
  -

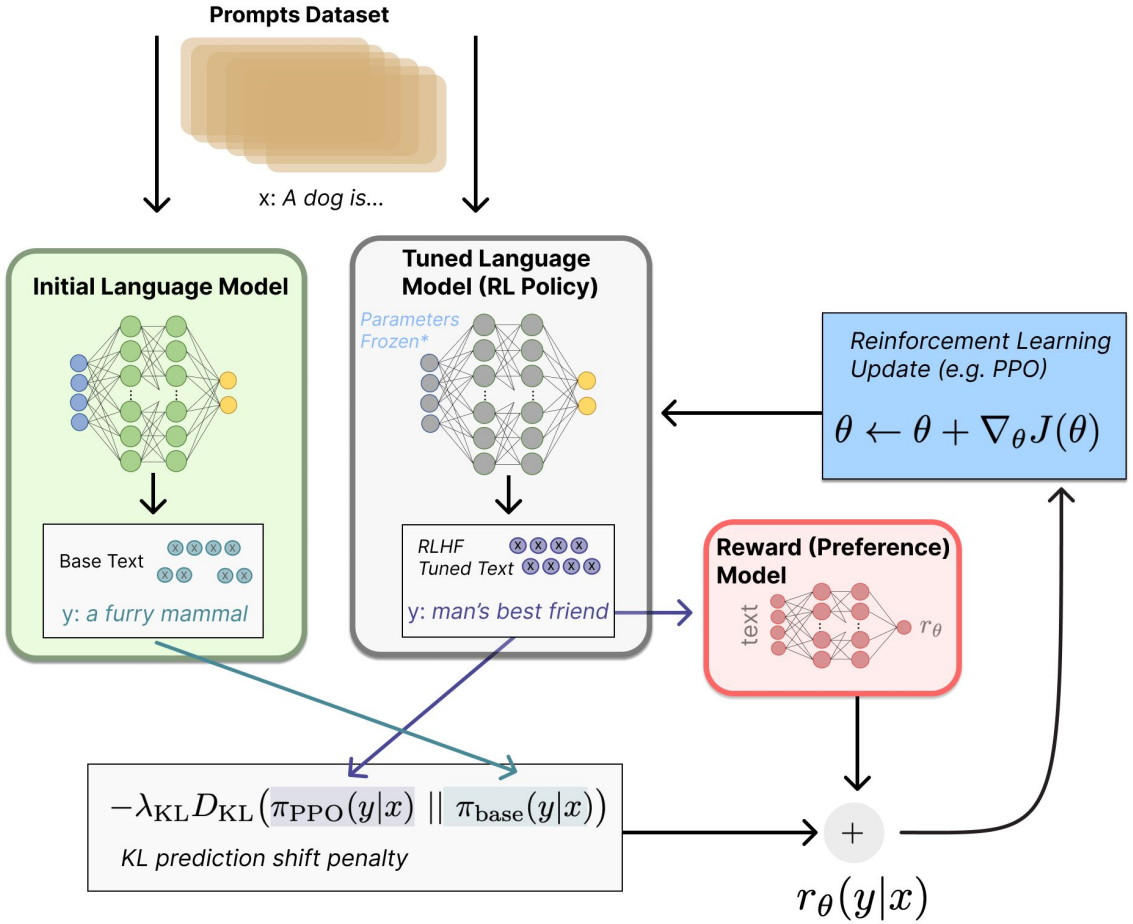
Характеристики RL

- Нет учителя, только награды
- Фидбек отложенный, а не мгновенный
- Время имеет значение (последовательность)
- Действия агента влияют на последующие данные

Пример


Domains
Knowledge
AlphaGo
Go
Human data
Domain knowledge
Known rules
AlphaGo becomes the first program to master Go using neural networks and tree search
(Jan 2016, Nature)
AlphaGo Zero
Go
Known rules
AlphaGo Zero learns to play completely on its own, without human knowledge
(Oct 2017, Nature)
AlphaZero
Go
Chess
Shogi
Known rules
AlphaZero masters three perfect information games using a single algorithm for all games
(Dec 2018, Science)
MuZero
Go
Chess
Shogi
Atari
MuZero learns the rules of the game, allowing it to also master environments with unknown dynamics.
(Dec 2020, Nature)

Пример

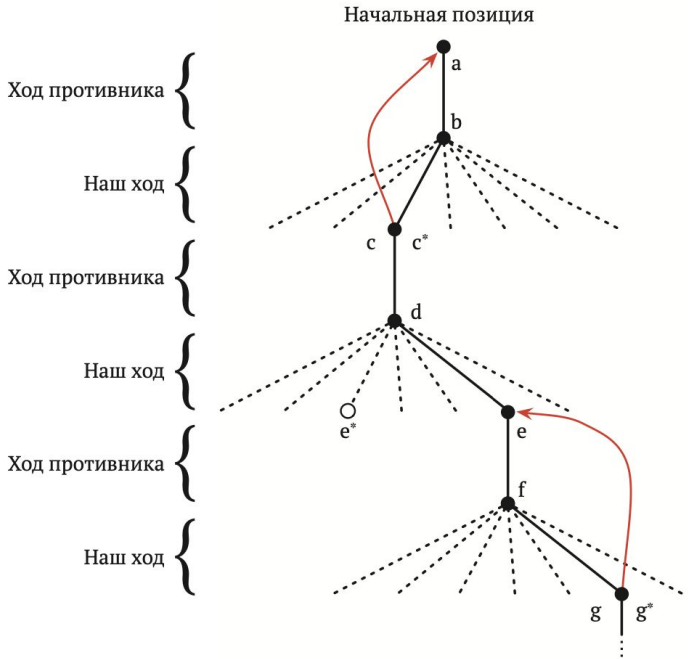

Prompts Dataset
x: A dog is...
Initial Language Model
Base Text
y: a furry mammal
Tuned Language Model (RL Policy)
Parameters Frozen*
RLHF Tuned Text
y: man's best friend
Reinforcement Learning Update (e.g. PPO)
$\theta \leftarrow \theta + \nabla_\theta J(\theta)$
Reward (Preference) Model
text
r_θ
$-\lambda_{\mathrm{KL}} D_{\mathrm{KL}}\left(\pi_{\mathrm{PPO}}(y|x) \;||\; \pi_{\mathrm{base}}(y|x)\right)$
KL prediction shift penalty
+
$r_\theta(y|x)$

Элементы RL

- Агент
- Окружающая среда
- Стратегия – отображение множества воспринимаемых состояний среды на действия, предпринимаемые в этих состояниях
- Сигнал вознаграждения – определяет цель в задаче обучения с подкреплением
- Функция ценности – это полное вознаграждение, которого агент может ожидать в будущем, если начнет работу в этом состоянии
- *Модель окружающей среды – используются для планирования

Пример “Крестики-Нолики”

- Таблица чисел для каждого состояния игры - последняя оценка вероятности выиграть, начав с этого состояния - функция ценности



На подумать

- Предположим, что вместо игры со случайным противником описанный выше алгоритм обучения с подкреплением играет против себя самого, и обе стороны обучаются. Как вы думаете, что произойдет в таком случае? Обучится ли игрок другой стратегии выбора ходов?

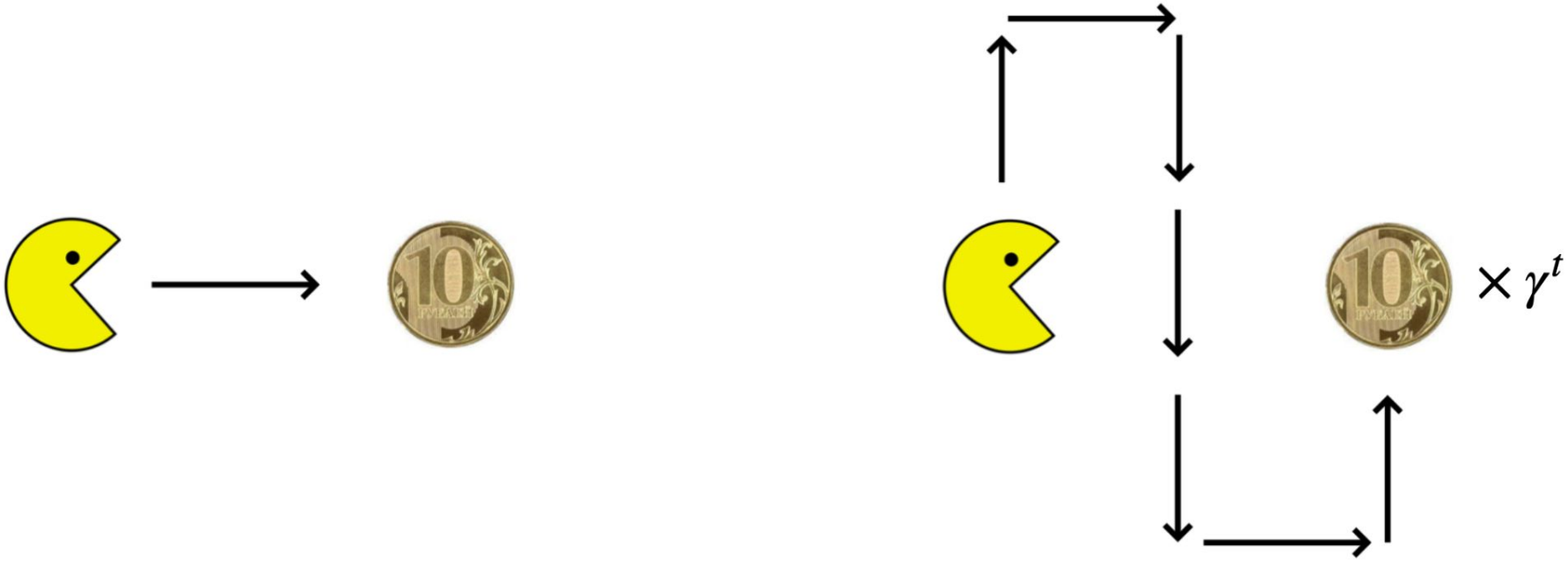
Награды

- Награда R_t - скалярный сигнал вознаграждения
- Показывает, на сколько хорошее действие принял агент на шаге t
- Задача агента - максимизировать кумулятивную сумму наград (функцию ценности)
- The reward hypothesis
    - All goals can be described by the maximisation of expected cumulative reward

Дизайн наград

Дисконтирование

$\gamma \in [0,1]$ is a discount factor

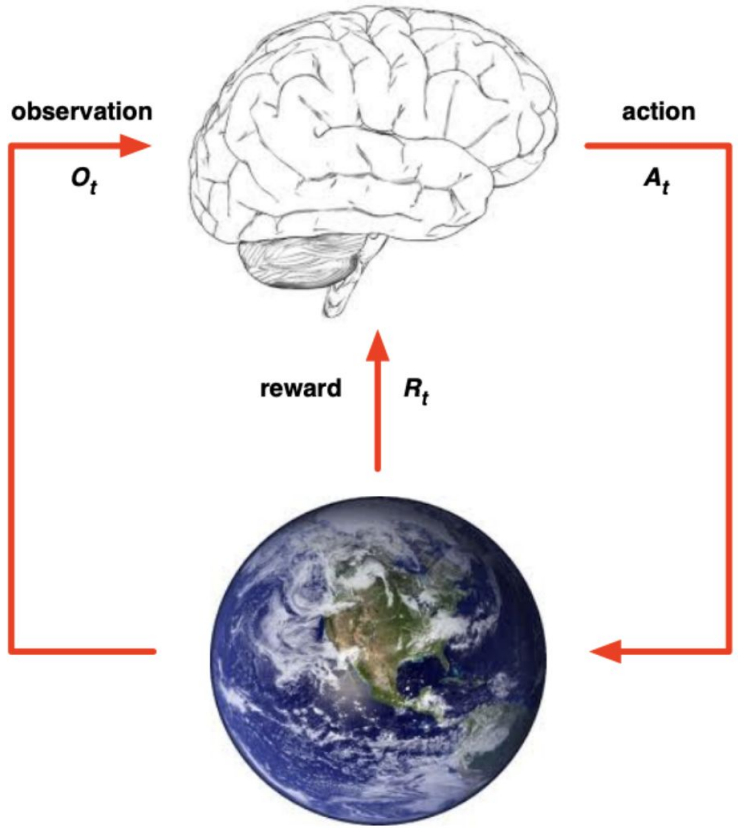


Зачем дисконтировать

- Избегать бесконечные награды
- Неопределенность в отношении будущего
- Поведение людей демонстрирует предпочтение немедленным наградам

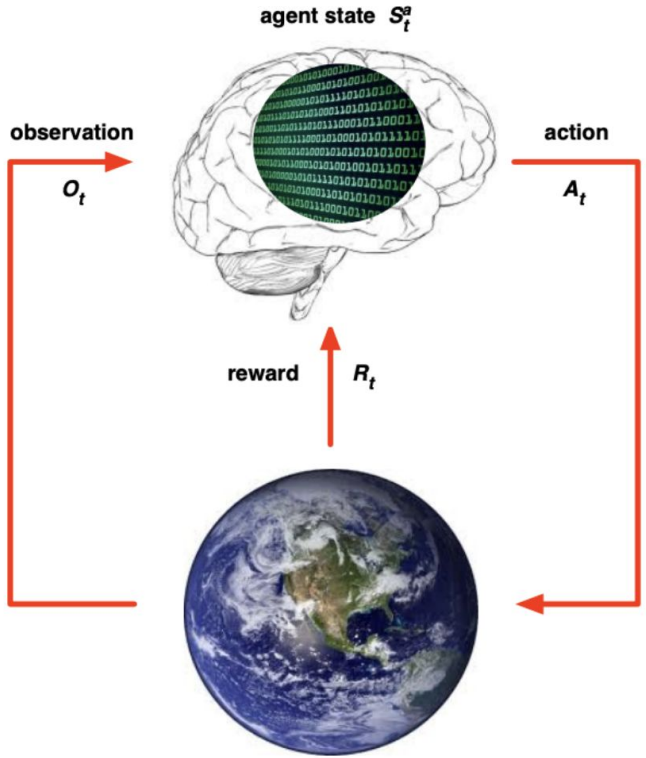
Агент и среда

- В каждом шаге t агент начинает в Ot
  - Выполняет действие At
  - Получает награду Rt
  - Получает следующее наблюдение Ot+1
- На каждом шаге t среда
  - Получает действие At
  - Передает награду Rt
  - Передает наблюдение Ot



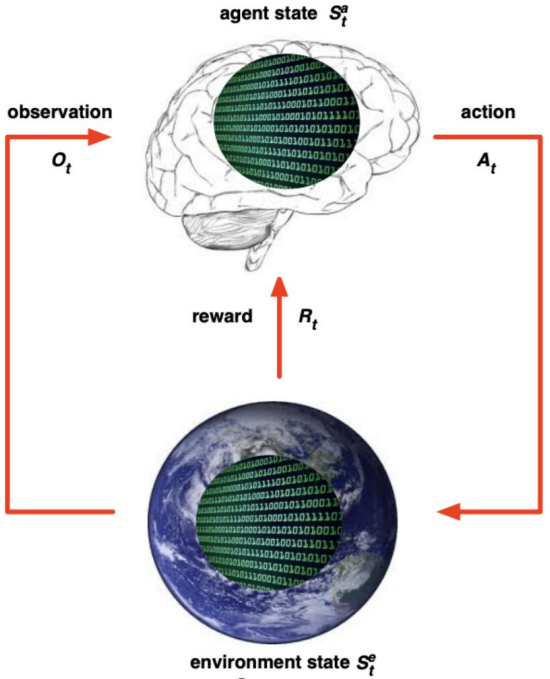
Состояние (state)

- Состояние среды St - закрытое представление среды
- Состояние агента - внутреннее представление агента

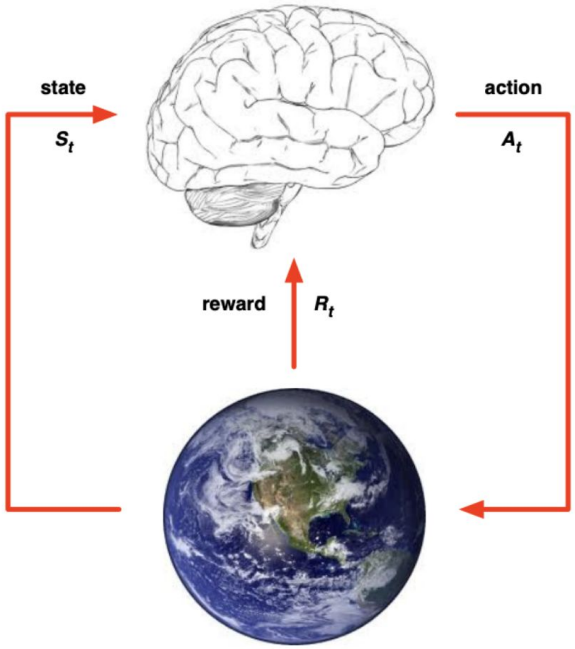


Observability (наблюдаемость)

- Частично наблюдаемы: Состояние среды и агента не равны



- Полностью наблюдаемы: Состояние среды и агента равны



Политика (Policy)

- Политика полностью определяет поведение агента (стратегия)
- Отображение из состояний в действия
- Детерминированная политика $a = \pi(s)$
- Стохастическая политика $\pi(a \mid s) = P[A_t = a \mid S_t = s]$

В итоге цель

$$\mathbb{E}_{\pi}[\sum_{t=0}^{T} \gamma^t R_t] \rightarrow \max_{\pi}$$