

# Лекция 1: Введение в обучение с подкреплением. Метод Cross-Entropy.

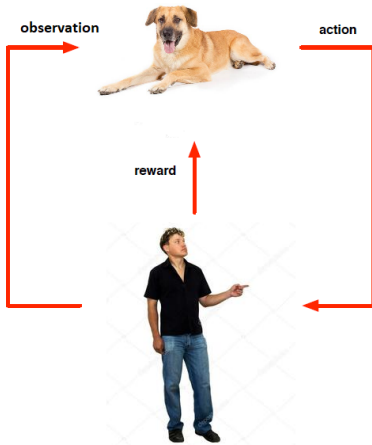
Антон Романович Плаксин

# Организационные вопросы

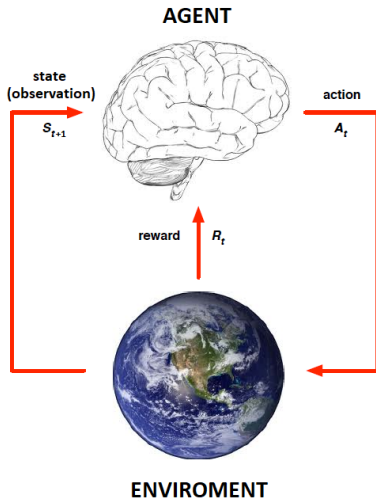
- Пятница, 17:50, аудитория 622
- Лекции и практики
- Отчетность: домашние работы
- Слайды: [https://github.com/imm-rl-lab/UrFU\\_course](https://github.com/imm-rl-lab/UrFU_course)
- E-mail для связи: [a.r.plaksin@gmail.com](mailto:a.r.plaksin@gmail.com)
- Вопросу по ходу можно и нужно!

# Что такое Reinforcement Learning?

# Что такое Reinforcement Learning?

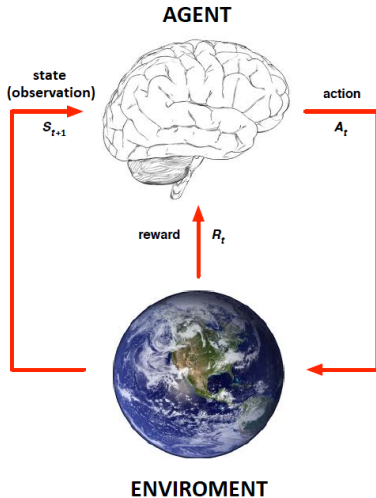


# Что такое Reinforcement Learning?



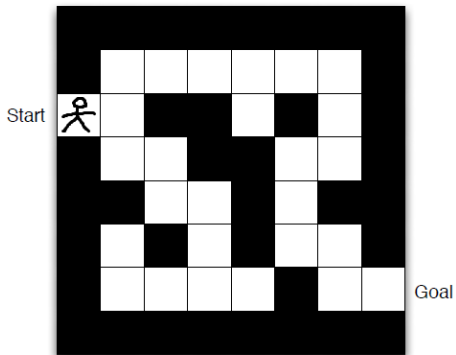
Цель агента - ???

# Что такое Reinforcement Learning?

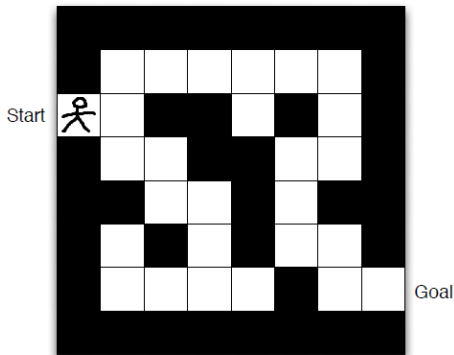


Цель агента - максимизировать  $G = \sum_{t=0}^T \gamma^t R_t$ ,  $\gamma \in [0, 1]$ .

# Пример: лабиринт



# Пример: лабиринт



- Состояния: белые клетки
- Действия:  $\uparrow$ ,  $\rightarrow$ ,  $\downarrow$ ,  $\leftarrow$
- Награда: -1 на каждом шаге



## Frozen Lake World (OpenAI GYM)



Agent

(1) Action (right, left, up down)



(2) state, reward



S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

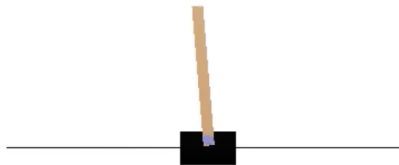
Environment

# Пример: Atari Games

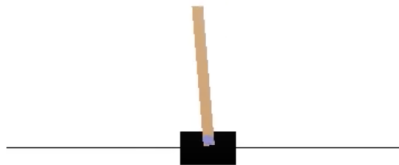


- Состояния: пиксели с экрана
- Действия:  $\rightarrow$ ,  $\leftarrow$ , «0»
- Награда: очки в игре

# Пример: Cartpole

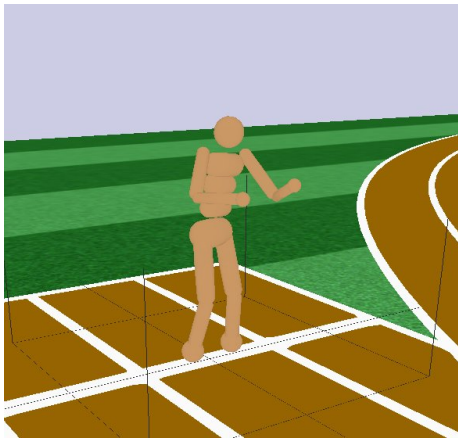


# Пример: Cartpole



- Состояния:  $\mathbb{R}^4$   
или пиксели с экрана
- Действия:  $\rightarrow$ ,  $\leftarrow$ , «0»
- Награда: +1 на каждом шаге

# Пример



- Состояния:  $\mathbb{R}^{26}$
- Действия:  $\mathbb{R}^6$
- Награда: +1 в каждый момент времени

# Markov Decision Process

## Markov Property

$$\mathbb{P}[R_t|S_t, A_t] = \mathbb{P}[R_t|S_1, A_1, S_2, A_2 \dots, S_t, A_t] = 1$$

$$\mathbb{P}[S_{t+1}|S_t, A_t] = \mathbb{P}[S_{t+1}|S_1, A_1, S_2, A_2 \dots, S_t, A_t]$$

# Markov Decision Process

## Markov Property

$$\mathbb{P}[R_t|S_t, A_t] = \mathbb{P}[R_t|S_1, A_1, S_2, A_2 \dots, S_t, A_t] = 1$$

$$\mathbb{P}[S_{t+1}|S_t, A_t] = \mathbb{P}[S_{t+1}|S_1, A_1, S_2, A_2 \dots, S_t, A_t]$$

## Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$  — пространство состояний
- $\mathcal{A}$  — пространство действий
- $\mathcal{P}$  — функция (матрица) вероятностей переходов между состояниями

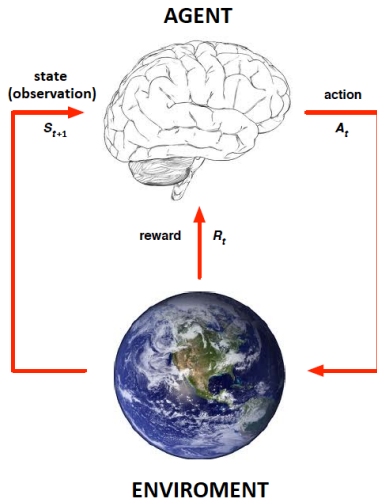
$$\mathcal{P}(s'|s, a) = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

- $\mathcal{R}$  — функция (вектор) вознаграждений

$$\mathcal{R}(s, a) = R_t \quad \Leftrightarrow \quad \mathbb{P}[R_t | S_t = s, A_t = a] = 1$$

- $\gamma \in [0, 1]$  — коэффициент дисконтирования

# Наша задача. Что мы хотим?





$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$

$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$

$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$
- совершает действие  $A_0 = \pi(S_0)$

$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$
- совершает действие  $A_0 = \pi(S_0)$
- получает награду  $R_0 = \mathcal{R}(S_0, A_0)$  и переходит в следующее состояние  $S_1 \sim \mathcal{P}(\cdot | S_0, A_0)$

$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$
- совершает действие  $A_0 = \pi(S_0)$
- получает награду  $R_0 = \mathcal{R}(S_0, A_0)$  и переходит в следующее состояние  $S_1 \sim \mathcal{P}(\cdot | S_0, A_0)$
- совершает действие  $A_1 = \pi(S_1)$

$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$
- совершает действие  $A_0 = \pi(S_0)$
- получает награду  $R_0 = \mathcal{R}(S_0, A_0)$  и переходит в следующее состояние  $S_1 \sim \mathcal{P}(\cdot | S_0, A_0)$
- совершает действие  $A_1 = \pi(S_1)$
- получает награду  $R_1 = \mathcal{R}(S_1, A_1)$  и переходит в следующее состояние  $S_2 \sim \mathcal{P}(\cdot | S_1, A_1)$

$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$
- совершает действие  $A_0 = \pi(S_0)$
- получает награду  $R_0 = \mathcal{R}(S_0, A_0)$  и переходит в следующее состояние  $S_1 \sim \mathcal{P}(\cdot | S_0, A_0)$
- совершает действие  $A_1 = \pi(S_1)$
- получает награду  $R_1 = \mathcal{R}(S_1, A_1)$  и переходит в следующее состояние  $S_2 \sim \mathcal{P}(\cdot | S_1, A_1)$
- ...
- $\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}, \quad G(\tau) = \sum_{t=0}^T \gamma^t \mathcal{R}(S_t, A_t)$



$$\pi: \mathcal{S} \mapsto \mathcal{A}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$
- совершает действие  $A_0 = \pi(S_0)$
- получает награду  $R_0 = \mathcal{R}(S_0, A_0)$  и переходит в следующее состояние  $S_1 \sim \mathcal{P}(\cdot | S_0, A_0)$
- совершает действие  $A_1 = \pi(S_1)$
- получает награду  $R_1 = \mathcal{R}(S_1, A_1)$  и переходит в следующее состояние  $S_2 \sim \mathcal{P}(\cdot | S_1, A_1)$
- ...
- $\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}, \quad G(\tau) = \sum_{t=0}^T \gamma^t \mathcal{R}(S_t, A_t)$

Наша задача

$$\mathbb{E}_{\pi}[G] \longrightarrow \max_{\pi}$$

# Stochastic policy

$$\pi(a|s) \in [0, 1], \quad a \in \mathcal{A}, \quad s \in \mathcal{S}$$

- Мы задаем  $\pi$
- АГЕНТ находится в начальном состоянии  $S_0 \in \mathcal{S}$
- совершает действие  $A_0 \sim \pi(\cdot|S_0)$
- получает награду  $R_0 = \mathcal{R}(S_0, A_0)$  и переходит в следующее состояние  $S_1 \sim \mathcal{P}(\cdot|S_0, A_0)$
- совершает действие  $A_1 \sim \pi(\cdot|S_1)$
- получает награду  $R_1 = \mathcal{R}(S_1, A_1)$  и переходит в следующее состояние  $S_2 \sim \mathcal{P}(\cdot|S_1, A_1)$
- ...
- $\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}, \quad G(\tau) = \sum_{t=0}^T \gamma^t \mathcal{R}(S_t, A_t)$

Наша задача

$$\mathbb{E}_{\pi}[G] \longrightarrow \max_{\pi}$$

# Что такое $\mathbb{E}_\pi[G]$

$$\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}$$

Что такое  $\mathbb{E}_\pi[G]$

$$\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}$$

$$\mathbb{P}(\tau|\pi) = ?$$

# Что такое $\mathbb{E}_\pi[G]$

$$\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}$$

$$\begin{aligned}\mathbb{P}(\tau) &= \mathbb{P}(A_0|S_0)\mathbb{P}(S_1|S_0, A_0) \\ &\times \mathbb{P}(A_1|S_1)\mathbb{P}(S_2|S_1, A_1) \\ &\times \dots \\ &\times \mathbb{P}(A_T|S_T)\end{aligned}$$

## Что такое $\mathbb{E}_\pi[G]$

$$\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}$$

$$\begin{aligned}\mathbb{P}(\tau) &= \mathbb{P}(A_0|S_0)\mathbb{P}(S_1|S_0, A_0) \\ &\times \mathbb{P}(A_1|S_1)\mathbb{P}(S_2|S_1, A_1) \\ &\times \dots \\ &\times \mathbb{P}(A_T|S_T)\end{aligned}$$

$$\mathbb{P}(\tau) = \prod_{t=0}^{T-1} \pi(A_t|S_t) \mathcal{P}(S_{t+1}|S_t, A_t) \cdot \pi(A_T|S_T)$$

Что такое  $\mathbb{E}_\pi[G]$

$$\tau = \{S_0, A_0, S_1, A_1, \dots, S_T, A_T\}$$

$$\begin{aligned}\mathbb{P}(\tau) &= \mathbb{P}(A_0|S_0)\mathbb{P}(S_1|S_0, A_0) \\ &\times \mathbb{P}(A_1|S_1)\mathbb{P}(S_2|S_1, A_1) \\ &\times \dots \\ &\times \mathbb{P}(A_T|S_T)\end{aligned}$$

$$\mathbb{P}(\tau) = \prod_{t=0}^{T-1} \pi(A_t|S_t) \mathcal{P}(S_{t+1}|S_t, A_t) \cdot \pi(A_T|S_T)$$

$$\mathbb{E}_\pi[G] = \int_{\tau \sim \pi} G(\tau) \mathbb{P}(d\tau)$$

# Общая схема алгоритмов в RL

На каждой итерации:

- Policy evaluation
- Policy improvement



# Policy evaluation

## Закон больших чисел

Если  $X_k$ ,  $k \in \overline{1, K}$  — независимые случайные величины с одним распределением. Тогда

$$\frac{1}{K} \sum_{k=1}^K X_k \rightarrow E[X] \text{ по вероятности}$$

# Policy evaluation

## Закон больших чисел

Если  $X_k$ ,  $k \in \overline{1, K}$  — независимые случайные величины с одним распределением. Тогда

$$\frac{1}{K} \sum_{k=1}^K X_k \rightarrow E[X] \text{ по вероятности}$$

## Подход Монте-Карло

$$E_{\pi}[G] \approx \frac{1}{K} \sum_{k=1}^K G(\tau_k)$$

# Cross-Entropy Method

Пусть  $\pi_0$  — начальная (равномерная) policy,  $N$  — количество итераций алгоритма,  $p \in [0, 1]$  — порог для элитных траекторий. Для каждого  $n \in \overline{0, N}$  делаем

# Cross-Entropy Method

Пусть  $\pi_0$  — начальная (равномерная) policy,  $N$  — количество итераций алгоритма,  $p \in [0, 1]$  — порог для элитных траекторий. Для каждого  $n \in \overline{0, N}$  делаем

- (Policy evaluation) Действуя в согласии с текущей policy  $\pi_n$  реализуем  $K$  сессий, получаем траектории  $\tau_k$ ,  $k \in \overline{1, K}$  и награды  $G(\tau_k)$  для каждой из них. Оцениваем policy  $\pi_n$ :

$$\mathbb{E}_{\pi_n}[G] \approx V_{\pi_n} := \frac{1}{K} \sum_{k=1}^K G(\tau_k)$$

Если  $V_{\pi_n} \ll V_{\pi_{n-1}}$ , то break с ответом  $\pi_{n-1}$

# Cross-Entropy Method

Пусть  $\pi_0$  — начальная (равномерная) policy,  $N$  — количество итераций алгоритма,  $p \in [0, 1]$  — порог для элитных траекторий. Для каждого  $n \in \overline{0, N}$  делаем

- **(Policy evaluation)** Действуя в согласии с текущей policy  $\pi_n$  реализуем  $K$  сессий, получаем траектории  $\tau_k$ ,  $k \in \overline{1, K}$  и награды  $G(\tau_k)$  для каждой из них. Оцениваем policy  $\pi_n$ :

$$\mathbb{E}_{\pi_n}[G] \approx V_{\pi_n} := \frac{1}{K} \sum_{k=1}^K G(\tau_k)$$

Если  $V_{\pi_n} \ll V_{\pi_{n-1}}$ , то break с ответом  $\pi_{n-1}$

- **(Policy improvement)** По значениям  $G(\tau_k)$  выбираем  $L = (1 - p)K$  элитных траекторий  $\mathcal{T}_n = \{\tau_1, \tau_2, \dots, \tau_L\}$  и по ним определяем следующую policy

$$\pi_{n+1}(a|s) = \frac{|\{(a|s) \in \mathcal{T}_n\}|}{|\{s \in \mathcal{T}_n\}|}$$

# В чем проблема алгоритма?

# В чем проблема алгоритма?

- Необходима большое количество сессий
- Выбор policy сильно зависит от случайности
- Проблемы со стохастической средой
- Работает только с конечными  $\mathcal{S}$  и  $\mathcal{A}$



# Проблема: выбор policy сильно зависит от случайности

## РЕШЕНИЕ:

- Сглаживание по Лапласу

$$\pi_{n+1}(a|s) = \frac{|(a|s) \in \mathcal{T}_n| + \lambda}{|s \in \mathcal{T}_n| + \lambda|\mathcal{A}|}, \quad \lambda > 0$$

- Сглаживание по policy

$$\pi_{n+1}(a|s) \leftarrow \lambda \pi_{n+1}(a|s) + (1 - \lambda) \pi_n(a|s), \quad \lambda \in (0, 1]$$

## РЕШЕНИЕ:

По стохастической policy  $\pi_n$  насэмплировать детерминированные policy  $\pi_{n,m}$ ,  $m \in \overline{1, M}$ . В согласии с каждой из них реализовать  $K$  сессий и получить траектории  $\tau_{m,k}$ ,  $m \in \overline{1, M}$ ,  $k \in \overline{1, K}$ .

Определить величины

$$V_{\pi_{n,m}} = \frac{1}{K} \sum_{k=1}^K G(\tau_{m,k})$$

По этим значениям выбираем  $L = (1 - p)K$  элитных траекторий

$$\mathcal{T}_n = \{\tau_{m,k}\}_{m=1, k=1}^{m=M, k=L}$$

ВОПРОСЫ?