Обучение с подкреплением

Евгений Борисов

обучение с подкреплением

способы организации данных и типы задач ML

supervised learning
- размеченный датасет {*X,target*}, (регрессия, классификация)

unsupervised learning
- НЕразмеченный датасет *{X}* (кластеризация)

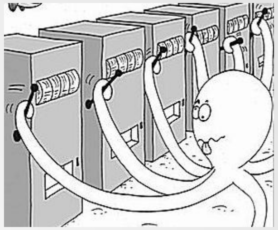
semi-supervised learning
- частично размеченный датасет {*X,target,X’*}, (трансдуктивные модели)

reinforcement learning
- нет датасета { ??? }

обучение с подкреплением

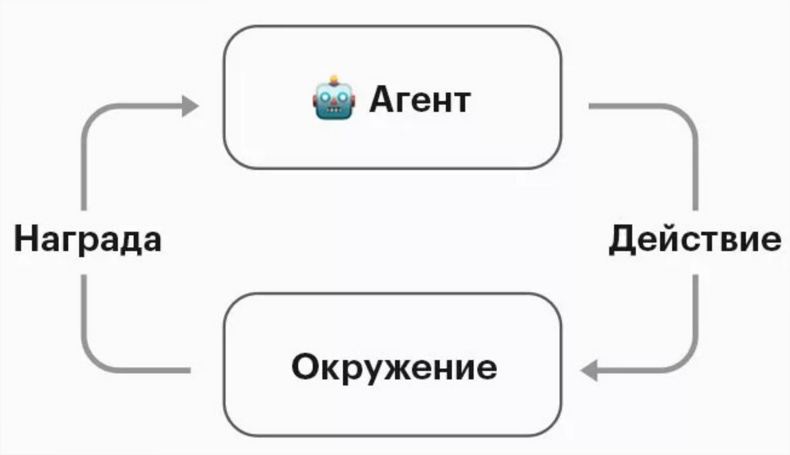
Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие,
на каждое действие получаем "**премию**", которая заранее не известна,

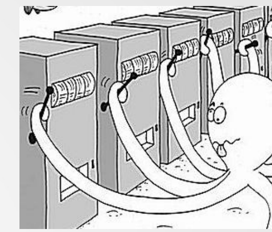
цель: максимизировать "**премию**"



обучение с подкреплением

Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие,
на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

A — множество возможных *действий*
$p(r|a)$ — неизвестное распределение *премии* $r \in \mathbb{R}$ для $a \in A$
$\pi_t(a)$ — *стратегия* (policy) агента в момент t, распределение на A

Игра агента со средой:

инициализация стратегии $\pi_1(a)$;
для всех $t = 1, \dots, T, \dots$
  агент выбирает действие $a_t \sim \pi_t(a)$;
  среда генерирует премию $r_t \sim p(r|a_t)$;
  агент корректирует стратегию $\pi_{t+1}(a)$;

$$Q_t(a) = \frac{\sum_{i=1}^{t} r_i[a_i = a]}{\sum_{i=1}^{t}[a_i = a]}$$ — средняя премия в t раундах

$$Q^*(a) = \lim_{t\to\infty} Q_t(a) \to \max_{a\in A}$$ — *ценность действия* a

Средняя премия Q — частотный вектор оценки действий A

обучение с подкреплением

Задача о многоруком бандите

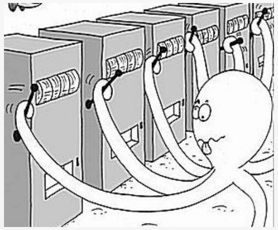
Herbert Robbins Some aspects of the sequential design of experiments. 1952

имеем фиксированного набор **действий**, выбираем и выполняем действие, на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

Возможные стратегии для достижения цели

- Жадная стратегия
- ε-Жадная стратегия
- Softmax стратегия
- Полужадная стратегия

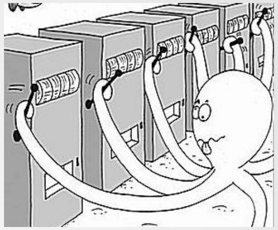


- Стратегия сравнения с подкреплением
- Стратегия преследования жадного

обучение с подкреплением

Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие,
на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

Жадная стратегия

Собираем статистику [действие,премия] и выбираем действие,
которое до этого выигрывало чаще.
Недостаток: некоторые действия могут вообще не применяться

$$A_t = \operatorname{Arg}\max_{a \in A} Q_t(a)$$

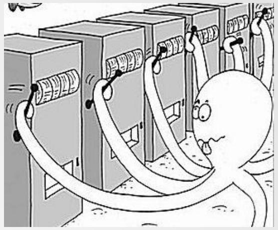
ε-Жадная стратегия

компромисс «изучение—применение»
Применяем жадную стратегию, периодически, с вероятностью ε (параметр), выбираем случайное (не максимальное) действие.

обучение с подкреплением

Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие,
на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

Softmax стратегия

считаем вероятности действий и выбираем случайно в соответствии с вероятностями,
позволяет выбирать "хорошие" действия но с оценкой ниже максимума

Мягкий вариант компромисса «изучение—применение»:
чем больше $Q_t(a)$, тем больше вероятность выбора a:

$$\pi_{t+1}(a) = \frac{\exp\big(Q_t(a)/\tau\big)}{\sum\limits_{b \in A} \exp\big(Q_t(b)/\tau\big)}$$

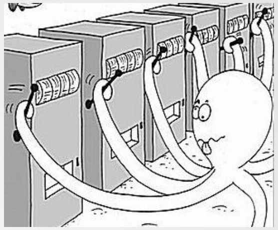
где τ — параметр *температуры*,
при $\tau \to 0$ стратегия стремится к жадной,
при $\tau \to \infty$ — к равномерной, т.е. чисто исследовательской

Эвристика: параметр τ имеет смысл уменьшать со временем.

обучение с подкреплением

Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие,
на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

Полужадная стратегия (UCB)

Определим оценку ценности действия,
чем менее исследована стратегия (редко выбирали действие)
тем выше должна быть его оценка,
и выбираем действие с максимальной верхней оценкой ценности,

$$A_t = \operatorname*{Arg\,max}_{a \in A} \left(Q_t(a) + \varepsilon \sqrt{\frac{2 \ln t}{k_t(a)}} \right)$$

где $k_t(a) = \sum_{i=1}^{t} [a_i = a]$, ε — параметр exr/ext-компромисса.

Интерпретация:
чем меньше $k_t(a)$, тем менее исследована стратегия,
тем выше должна быть вероятность выбрать a;

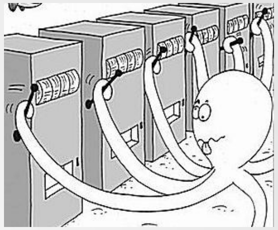
чем больше ε, тем стратегия более исследовательская.

Эвристика: параметр ε уменьшать со временем.

обучение с подкреплением

Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие, на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

$$Q_t(a) = \frac{\sum_{i=1}^{t} r_i [a_i = a]}{\sum_{i=1}^{t} [a_i = a]} \quad \text{— средняя премия в } t \text{ раундах}$$

$$Q^*(a) = \lim_{t \to \infty} Q_t(a) \to \max_{a \in A} \quad \text{— \textit{ценность действия a}}$$

заменим среднюю премию Q на сглаженную оценку

Рекуррентная формула Moving Average для усреднения Q_t:

$$Q_t(a) = \alpha r_t + (1 - \alpha) Q_{t-1}(a) = \mathrm{MA}_\alpha(r_t)$$

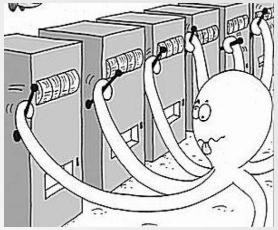
При $\alpha = \mathrm{const}$ это экспоненциальное скользящее среднее (EMA)

При $\alpha = \frac{1}{k_t(a)}$ это среднее арифметическое

обучение с подкреплением

Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие,
на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

Стратегия преследования жадного

Применяем жадную стратегию к сглаженной оценке средней премии Q

$$\pi_{t+1}(a) = \mathrm{EMA}_\alpha\left(\frac{[a \in A_t]}{|A_t|}\right), \quad a \in A$$

Сравнение с подкреплением (reinforcement comparison):
$\bar{r}_t = \mathrm{EMA}_\alpha(r_t)$ — *средняя премия* по всем действиям,
$p_t(a_t) = \mathrm{EMA}_\beta(r_t - \bar{r}_t)$ — *преимущество* (advantage) действия,

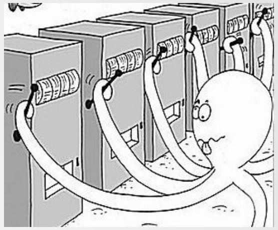
$$\pi_{t+1}(a) = \frac{\exp\left(\frac{1}{\tau} p_t(a)\right)}{\sum_{a'} \exp\left(\frac{1}{\tau} p_t(a')\right)},$$

при $\tau \to 0$ стратегия стремится к жадной,
при $\tau \to \infty$ — к равномерной, т.е. чисто исследовательской.

обучение с подкреплением

Задача о многоруком бандите

Herbert Robbins Some aspects of the sequential design of experiments. 1952



имеем фиксированного набор **действий**, выбираем и выполняем действие,
на каждое действие получаем "**премию**", которая заранее не известна,

цель: максимизировать "**премию**"

Стратегия сравнения с подкреплением

используем не сами значения премий, а их разности со средней премией.

$\bar{r}_{t+1} = \bar{r}_t + \alpha(r_t - \bar{r}_t)$ — *средняя премия*

$p_{t+1}(a_t) = p_t(a_t) + \beta(r_t - \bar{r}_t - p_t(a_t))$ — *предпочтения* действий

$$\pi_{t+1}(a) = \frac{\exp\left(p_{t+1}(a)/\tau\right)}{\sum_{b \in A} \exp\left(p_{t+1}(b)/\tau\right)}$$ — softmax-стратегия агента

Начальное приближение r_0: оптимистично завышенное стимулирует изучающие действия в начале

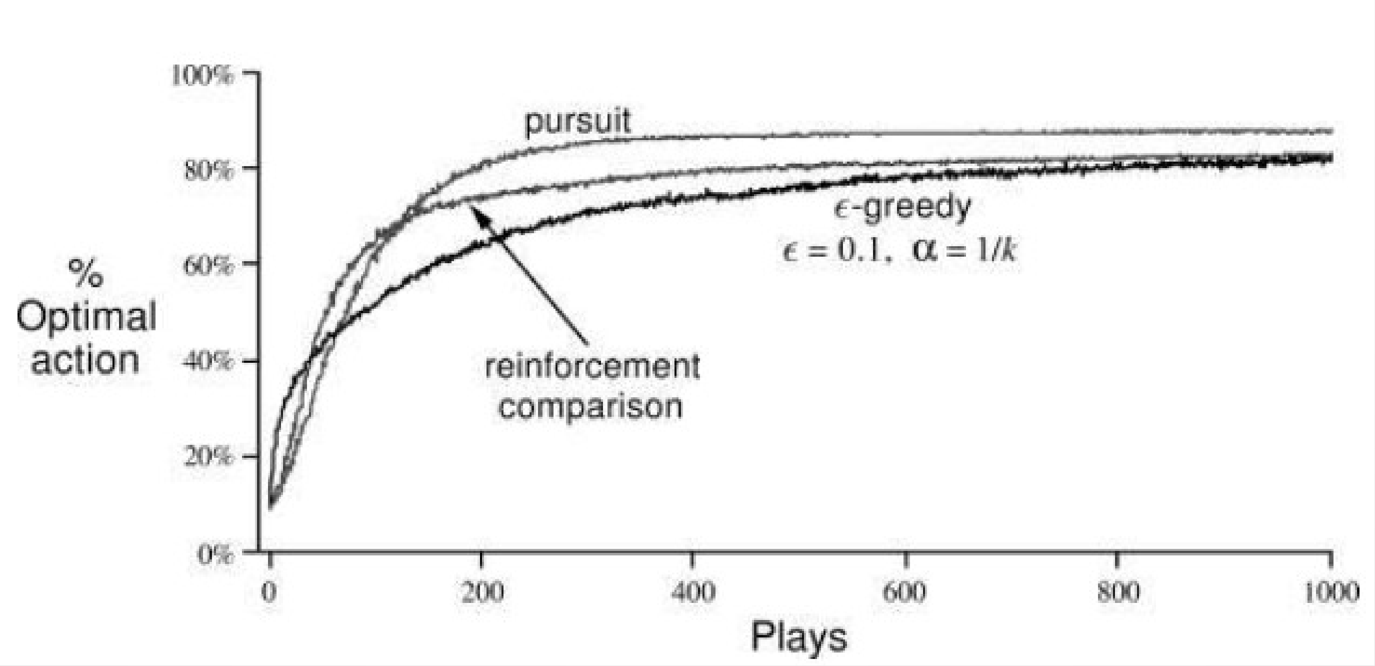
обучение с подкреплением

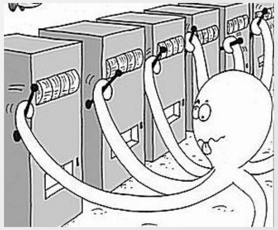
Задача о многоруком бандите
Модельные данные для тестов и оценки

«10-рукая испытательная среда»:
Генерируется 2000 задач, в каждой задаче
$|A| = 10$,
$p_a(r) = \mathcal{N}(r; Q^*(a), 1)$,
$Q^*(a) \sim \mathcal{N}(0, 1)$.

Строятся графики зависимости
— среднего вознаграждения (average reward),
— доли оптимальных действий (% optimal action),
от числа действий (сыгранных игр),
усреднённые по 2000 задачам.

Сравнение с подкреплением лучше ε-жадных стратегий
Стратегия преследования ещё лучше





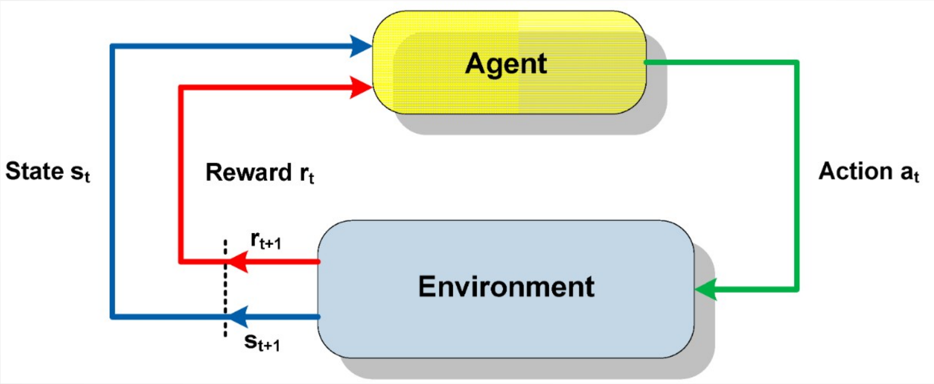
обучение с подкреплением

Модель среды

введём понятие состояния **среды**, которое **агент** может наблюдать

после совершения действия агентом
кроме выдачи премии **среда** ещё меняет своё **состояние**

пример: дом с несколькими лифтами,
состояние среды это положение лифтов,
Действие - выбор лифта,
Цель: минимизация времени ожидания



- учебного набора в явном виде нет
- собираем историю действий и последствий
- наблюдаем состояние среды
- предсказываем реакцию среды на действие
- выбираем оптимальное действие

обучение с подкреплением

Формальное описание задачи

A — конечное множество возможных *действий* (action)
S — конечное множество состояний среды (state)

Игра агента со средой:

инициализация стратегии $\pi_1(a \mid s)$ и состояния среды s_1;
для всех $t = 1, \ldots, T, \ldots$
- агент выбирает действие $a_t \sim \pi_t(a \mid s_t)$;
- среда генерирует премию $r_t \sim p(r \mid a_t, s_t)$
- и новое состояние $s_{t+1} \sim p(s \mid a_t, s_t)$;
- агент корректирует стратегию $\pi_{t+1}(a \mid s)$;

Марковский процесс принятия решений (МППР, MDP):

$$\mathsf{P}(s_{t+1}, r_t \mid s_t, a_t, r_{t-1}, s_{t-1}, a_{t-1}, r_{t-2}, \ldots, s_1, a_1) = \\ = \mathsf{P}(s_{t+1}, r_t \mid s_t, a_t)$$

обучение с подкреплением

Формальное описание задачи

$\mathcal{A}$ — конечное множество возможных *действий* (action)
$\mathcal{S}$ — конечное множество состояний среды (state)

Игра агента со средой:

инициализация стратегии $\pi_1(a \mid s)$ и состояния среды s_1;
для всех $t = 1, \dots, T, \dots$
- агент выбирает действие $a_t \sim \pi_t(a \mid s_t)$;
- среда генерирует премию $r_t \sim p(r \mid a_t, s_t)$
- и новое состояние $s_{t+1} \sim p(s \mid a_t, s_t)$;
- агент корректирует стратегию $\pi_{t+1}(a \mid s)$;

Марковский процесс принятия решений (МППР, MDP):

$$\mathsf{P}(s_{t+1}, r_t \mid s_t, a_t, r_{t-1}, s_{t-1}, a_{t-1}, r_{t-2}, \dots, s_1, a_1) = \\ = \mathsf{P}(s_{t+1}, r_t \mid s_t, a_t)$$

- выборка (s_t, a_t, r_t) не является независимой
- распределение $p(s_t, a_t, r_t)$ может меняться во времени и зависеть от стратегии агента π

- премии могут
  — оценивать действия с большой задержкой
  — быть разреженными (почти всё время $r_t = 0$)
  — быть зашумлёнными (не ясно, за что именно премия)

обучение с подкреплением

Формальное описание задачи

A — конечное множество возможных *действий* (action)
S — конечное множество состояний среды (state)

Игра агента со средой:

инициализация стратегии $\pi_1(a \mid s)$ и состояния среды s_1;
для всех $t = 1, \dots, T, \dots$
- агент выбирает действие $a_t \sim \pi_t(a \mid s_t)$;
- среда генерирует премию $r_t \sim p(r \mid a_t, s_t)$
- и новое состояние $s_{t+1} \sim p(s \mid a_t, s_t)$;
- агент корректирует стратегию $\pi_{t+1}(a \mid s)$;

Марковский процесс принятия решений (МППР, MDP):

$$\mathsf{P}(s_{t+1}, r_t \mid s_t, a_t, r_{t-1}, s_{t-1}, a_{t-1}, r_{t-2}, \dots, s_1, a_1) = \\ = \mathsf{P}(s_{t+1}, r_t \mid s_t, a_t)$$

- выборка (s_t, a_t, r_t) не является независимой
- распределение $p(s_t, a_t, r_t)$ может меняться во времени и зависеть от стратегии агента π

- премии могут
  - — оценивать действия с большой задержкой
  - — быть разреженными (почти всё время $r_t = 0$)
  - — быть зашумлёнными (не ясно, за что именно премия)

Какие параметрические модели можно обучать:
- стратегию $\pi_{t+1}(a|s; \theta)$
- функцию ценности состояния $V(s; \theta)$
- функцию ценности действия в состоянии $Q(s, a; \theta)$
- модель среды $(r_t, s_{t+1}) = \mu(s_t, a_t; \theta)$

обучение с подкреплением

Формальное описание задачи

A — конечное множество возможных *действий* (action)
S — конечное множество состояний среды (state)

Игра агента со средой:

инициализация стратегии $\pi_1(a \mid s)$ и состояния среды s_1;
для всех $t = 1, \dots, T, \dots$
агент выбирает действие $a_t \sim \pi_t(a \mid s_t)$;
среда генерирует премию $r_t \sim p(r \mid a_t, s_t)$
и новое состояние $s_{t+1} \sim p(s \mid a_t, s_t)$;
агент корректирует стратегию $\pi_{t+1}(a \mid s)$;

Дисконтированная выгода (discounted return):

$$R_t = r_t + \gamma r_{t+1} + \dots + \gamma^k r_{t+k} + \dots$$

где $\gamma \in [0, 1]$ — коэффициент дисконтирования,
$1 + \gamma + \gamma^2 + \dots = \frac{1}{1-\gamma}$ — горизонт дальновидности агента.

Функции ценности состояния $V^\pi(s)$ *и ценности действия в состоянии* $Q^\pi(s, a)$ при условии, что агент следует стратегии π:

$$V^\pi(s) = \mathsf{E}_\pi(R_t \mid s_t = s) \qquad = \mathsf{E}_\pi\Big(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \Bigm| s_t = s\Big)$$

$$Q^\pi(s, a) = \mathsf{E}_\pi(R_t \mid s_t = s,\ a_t = a) = \mathsf{E}_\pi\Big(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \Bigm| s_t = s,\ a_t = a\Big)$$

Рекуррентная формула для функции ценности $Q^\pi(s, a)$:

$$\begin{aligned} Q^\pi(s, a) &= \mathsf{E}_\pi\big(\textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k} \bigm| s_t = s,\ a_t = a\big) \\ &= \mathsf{E}_\pi\big(r_t + \gamma \textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \bigm| s_t = s,\ a_t = a\big) \\ &= \mathsf{E}_\pi\big(r_t + \gamma Q^\pi(s_{t+1}, a_{t+1}) \bigm| s_t = s,\ a_t = a\big) \end{aligned}$$

Уравнение Беллмана для оптимальной функции ценности Q^*:

$$Q^*(s, a) = \mathsf{E}_\pi\big(r_t + \gamma \max_{a' \in A} Q^*(s_{t+1}, a') \bigm| s_t = s,\ a_t = a\big)$$

Утв. *Жадная стратегия* π относительно $Q^*(s, a)$ «выбирать то действие, на котором достигается максимум в уравнениях Беллмана», является оптимальной:

$$A_t = \operatorname*{Arg\,max}_{a \in A} Q^*(s_t, a)$$

обучение с подкреплением

Метод SARSA (state-action-reward-state-action)

Апроксимируем оценнку действия в состоянии Q(s,a) экспоненциальным скользящим средним

$$Q(s_t, a_t) = \mathrm{EMA}_\alpha \big(r_t + \gamma Q(s_{t+1}, a') \big)$$

инициализация стратегии $\pi_1(a \mid s)$ и состояния среды s_1;
для всех $t = 1, \dots T, \dots$
- агент выбирает действие $a_t \sim \pi_t(a \mid s_t)$, например, $a_t = \arg\max_a Q(s_t, a)$ — жадная стратегия;
- среда генерирует $r_t \sim p(r \mid a_t, s_t)$ и $s_{t+1} \sim p(s \mid a_t, s_t)$;
- агент разыгрывает ещё один шаг: $a' \sim \pi_t(a \mid s_{t+1})$;
- $Q(s_t, a_t) := Q(s_t, a_t) + \alpha \big(r_t + \gamma Q(s_{t+1}, a') - Q(s_t, a_t) \big)$;

обучение с подкреплением

Метод Q-learning

Апроксимируем оптимальную оценнку действия в состоянии Q*(s,a) экспоненциальным скользящим средним

$$Q(s_t, a_t) = \mathrm{EMA}_\alpha\left(r_t + \gamma \max_{a'} Q(s_{t+1}, a')\right)$$

инициализация стратегии $\pi_1(a \mid s)$ и состояния среды s_1;

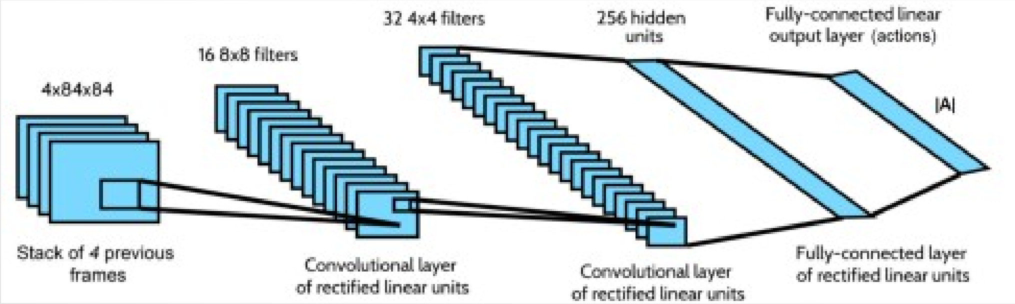
для всех $t = 1, \dots, T, \dots$

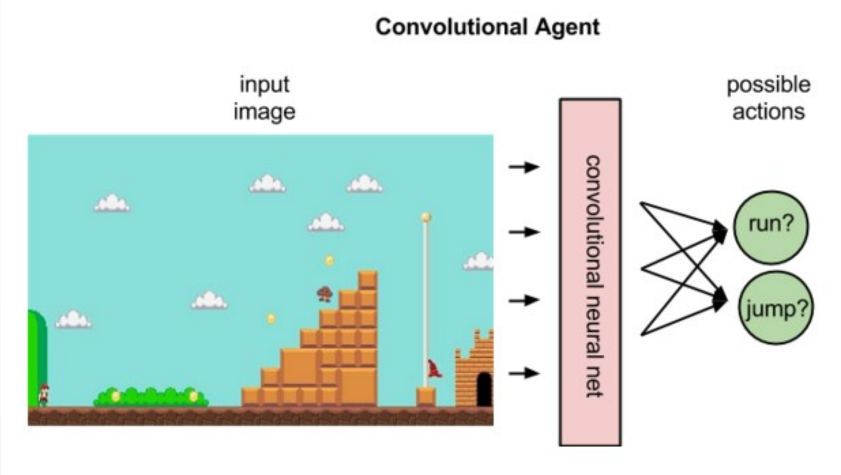
- агент выбирает действие $a_t \sim \pi_t(a \mid s_t)$;
- среда генерирует $r_t \sim p(r \mid a_t, s_t)$ и $s_{t+1} \sim p(s \mid a_t, s_t)$;
- $Q(s_t, a_t) := Q(s_t, a_t) + \alpha\left(r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right)$;

обучение с подкреплением

Метод DQN (Deep Q-learning Network)

Среда — эмулятор игр Atari
Состояние - 4 последовательных кадра
Действия - зависят от игры
Премия — текущий score игры
Функция ценности действия в состоянии — свёрточная сеть





V.Mnih et al. (DeepMind). Playing Atari with deep reinforcement learning. 2013

обучение с подкреплением

Метод DQN (Deep Q-learning Network)

Сохранение траекторий $(s_t, a_t, r_t)_{t=1}^{T}$ в памяти (reply memory) для многократного *воспроизведения опыта* (experience replay)

Аппроксимация оптимальной функции ценности $Q(s_t, a_t)$ при фиксированных текущих параметрах сети w_t:

$$y_t = \begin{cases} r_t, & \text{если состояние } s_{t+1} \text{ терминальное} \\ r_t + \gamma \max_a Q(s_{t+1}, a; w_t), & \text{иначе} \end{cases}$$

Функция потерь для обучения нейросевой модели $Q(s, a; w)$:

$$\mathscr{L}_t(w) = \left(Q(s_t, a_t; w) - y_t\right)^2$$

Стохастический градиент SGD (по мини-батчам длины 32):

$$w_{t+1} = w_t - \eta\left(Q(s_t, a_t; w_t) - y_t\right)\nabla_w Q(s_t, a_t; w_t)$$

обучение с подкреплением

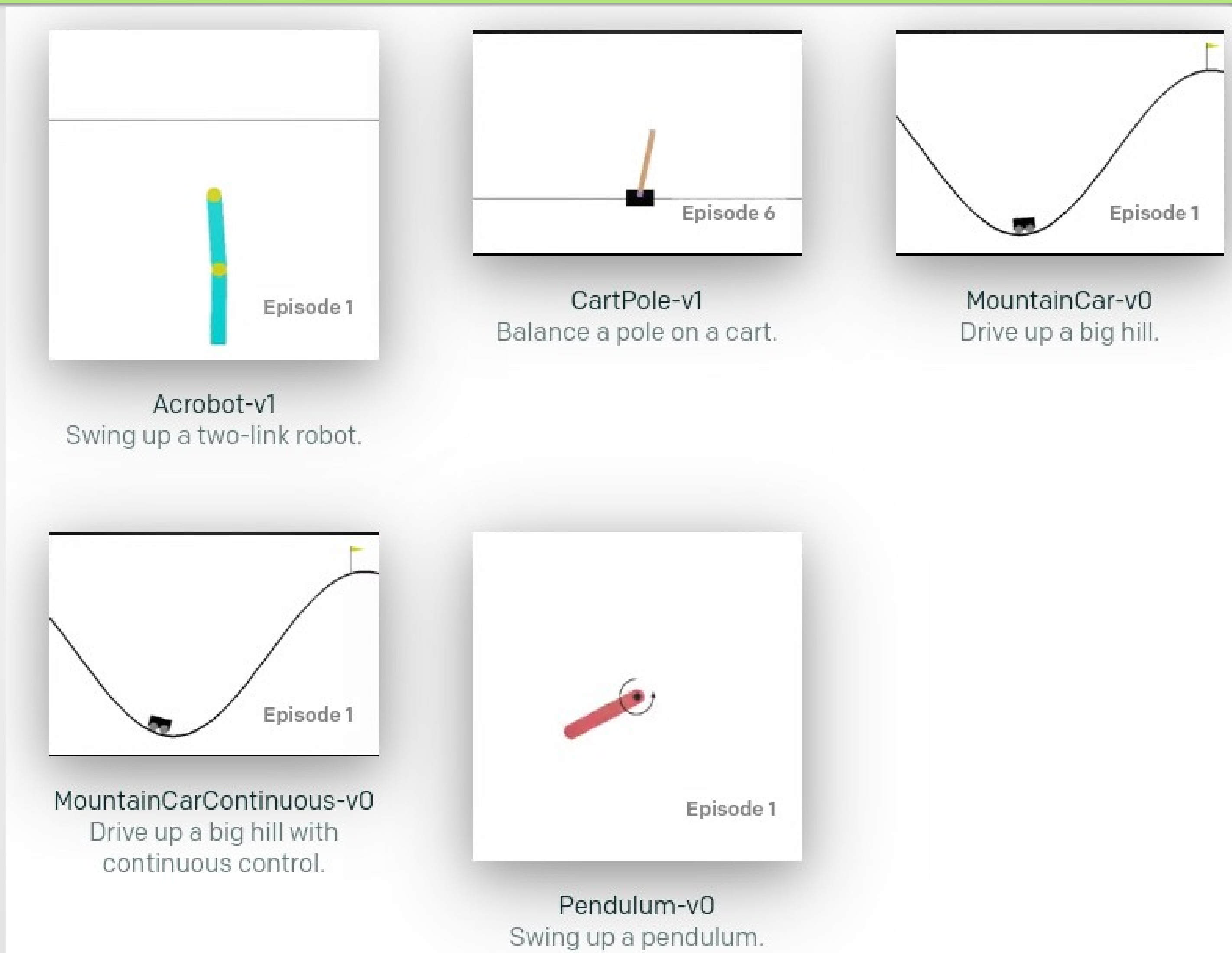
Метод DQN (Deep Q-learning Network)

инициализация reply-памяти и параметров сети w;
для всех эпизодов $m = 1, \ldots, M$
    инициализация состояния среды s_1;
    для всех $t = 1, \ldots, T_m$ (длина m-го эпизода)

$$a_t = \begin{cases} \text{случайное действие}, & \text{с вероятностью } \varepsilon; \\ \arg\max\limits_a Q(s_t, a, w), & \text{с вероятностью } 1 - \varepsilon; \end{cases}$$

        среда генерирует $r_t \sim p(r \mid a_t, s_t)$ и $s_{t+1} \sim p(s \mid a_t, s_t)$;
        запомнить (s_t, a_t, r_t) в reply-памяти;
        выбрать случайный фрагмент траектории из памяти;
        для всех $j = 1, \ldots, J$ (длина мини-батчей)
            оценить y_j;
            сделать градиентный шаг, обновить w;

обучение с подкреплением



https://gym.openai.com

обучение с подкреплением : литература

git clone *https://github.com/mechanoid5/ml_lectorium.git*

К.В. Воронцов Обучение с подкреплением.
https://www.youtube.com/watch?v=ZkZQwKizgLM

Радослав Нейчев Intro to Reinforcement Learning.
https://www.youtube.com/watch?v=BwLIPEUkjxQ

Саттон Р.С., Барто Э. Г. Обучение с подкреплением. - Москва:Бином, 2014г.

Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. - "Питер", 2018 г.