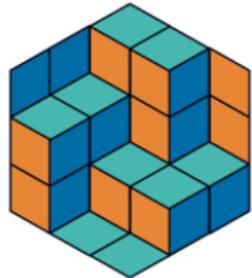


Лекция 13. Введение в обучение с подкреплением

Александр Юрьевич Авдюшенко

МКН СПбГУ

12 мая 2022



Факультет
математики
и компьютерных
наук
СПбГУ

Пятиминутка

- ▶ Опишите варианты постановки задачи сегментации изображений
- ▶ В чем основная идея архитектуры U-net?
- ▶ Выпишите пару метрик в задаче сегментации

Обучение с учителем

Дано:

Обучающая выборка в виде пар (x, y)

Нужно найти $a_\theta(x) \approx y(x)$ в смысле функции потерь
 $\mathcal{L}(y, a_\theta(x))$

Обучение с учителем

Дано:

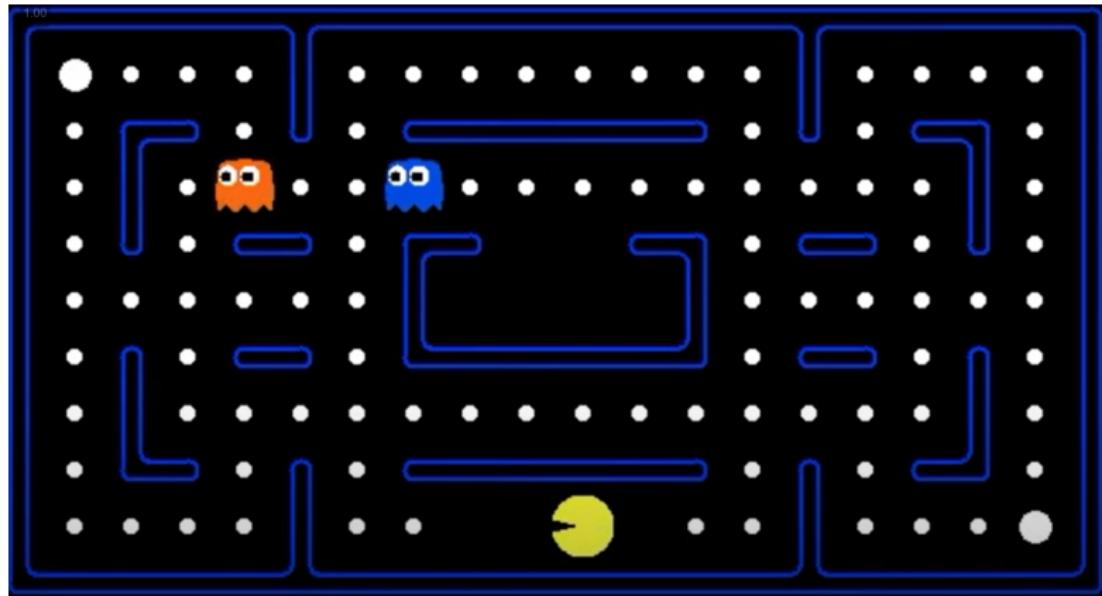
Обучающая выборка в виде пар (x, y)

Нужно найти $a_\theta(x) \approx y(x)$ в смысле функции потерь
 $\mathcal{L}(y, a_\theta(x))$

Замечание

В реальном мире такого почти никогда нет.

Реальный мир



Deep Reinforcement Learning in Pac-man

Многорукие бандиты

Напоминание

Возможные применения:

- ▶ рекламные баннеры
- ▶ рекомендации (товары, музыка, фильмы, лента)
- ▶ игровые автоматы

Подходы:

- ▶ сэмплирование Томпсона
- ▶ Upper Confidence Bound (UCB)
- ▶ ε -жадная стратегия

Главное отличие от более общего обучения с подкреплением — в многоруких бандитах нет состояния среды

Многорукие бандиты

Недостаток

Не учитывают отложенные по времени последствия

Например, эффект от кликбайта в рекламе

Многорукие бандиты

Недостаток

Не учитывают отложенные по времени последствия

Например, эффект от кликбайта в рекламе

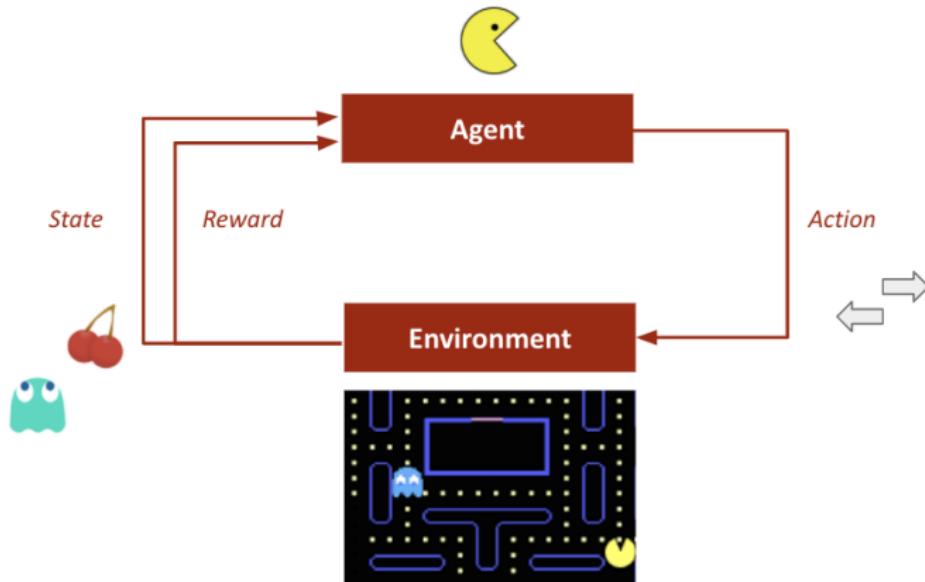


pikabu.ru

ШОК! Коты хотят поработить нас

Обучение с подкреплением

Примеры



- ▶ управление роботами
- ▶ (видео)игры
- ▶ управление цennыми бумагами

Модель агента в изменяющейся среде

Определения:

- ▶ состояния среды (state) $s \in S$
- ▶ действия агента (action) $a \in A$
- ▶ награды (reward) $r \in R$
- ▶ динамика переходов между состояниями
 $P(s_{t+1}|s_t, a_t, \dots, s_{t-i}, a_{t-i}, \dots, s_0, a_0)$
- ▶ функция выигрыша

Задача:

$$r_t = r(s_t, a_t, \dots, s_0, a_0)$$

$$\pi(a|s) : \mathbb{E}_\pi[R] \rightarrow \max$$

- ▶ абсолютный выигрыш (total reward)

$$R = \sum_t r_t$$

- ▶ стратегия агента (policy)

$$\pi(a|s)$$

Метод кросс-энтропии. Алгоритм

Траектория — $[s_0, a_0, s_1, a_1, s_2, \dots, a_{T-1}, s_T]$

Инициализируем модель стратегии $\pi(a|s)$

Повторяем:

- ▶ играем N сессий
- ▶ выбираем из них K лучших и берем их траектории
- ▶ настраиваем $\pi(a|s)$ так, чтобы в состоянии s максимизировать вероятности действий из лучших траекторий

Пример обучения

Метод кросс-энтропии. Реализация с помощью таблицы

В качестве модели стратегии берем просто матрицу π размерности $|S| \times |A|$

$$\pi(a|s) = \pi_{s,a}$$

после отбора лучших траекторий получаем набор пар

$$\text{Elite} = [(s_0, a_0), (s_1, a_1), (s_2, a_2), \dots, (s_H, a_H)]$$

и максимизируем правдоподобие

$$\pi_{s,a} = \frac{\sum_{s_t, a_t \in \text{Elite}} [s_t = s][a_t = a]}{\sum_{s_t, a_t \in \text{Elite}} [s_t = s]}$$

Проблема..

В ТОМ, ЧТО СОСТОЯНИЙ ОЧЕНЬ МНОГО:



Аппроксимированный метод кросс-энтропии

Возможные решения:

- разбить пространство состояний на участки и считать их состояниями

Аппроксимированный метод кросс-энтропии

Возможные решения:

- ▶ разбить пространство состояний на участки и считать их состояниями
- ▶ получать вероятности из модели машинного обучения
 $\pi_\theta(a|s)$: линейная модель, нейронная сеть, случайный лес
- ▶ часто эти вероятности потом ещё нужно дорабатывать

Аппроксимированный метод кросс-энтропии

Пример

- ▶ В качестве модели стратегии берем просто нейронную сеть π_θ
- ▶ Инициализируем случайными весами
- ▶ На каждой итерации после отбора лучших траекторий получаем набор пар

$$\text{Elite} = [(s_0, a_0), (s_1, a_1), (s_2, a_2), \dots, (s_H, a_H)]$$

- ▶ и выполняем оптимизацию

$$\pi = \arg \max_{\theta} \sum_{s_i, a_i \in \text{Elite}} \log \pi(a_i | s_i) = \arg \max_{\theta} \mathcal{L}(\theta)$$

- ▶ то есть

$$\theta_{t+1} = \theta_t + \alpha \nabla \mathcal{L}(\theta)$$

Почему метод кросс-энтропии?

$$\begin{aligned} KL(p_1(x) \| p_2(x)) &= E_{x \sim p_1(x)} \log \frac{p_1(x)}{p_2(x)} = \\ &= \underbrace{E_{x \sim p_1(x)} \log p_1(x)}_{\text{энтропия}} - \underbrace{E_{x \sim p_1(x)} \log p_2(x)}_{\text{кросс-энтропия}} \end{aligned}$$

Недостатки метода кросс-энтропии

- ▶ нестабилен при малых выборках
- ▶ в случае не детерминистской среды выбираем удачные случаи (случайность играла в пользу агента)
- ▶ концентрируемся на поведении в простых состояниях
- ▶ игнорируем большое количество информации
- ▶ есть задачи, в которых конец не настаёт никогда (игра на бирже)

Q-Learning

- ▶ порой для оценки эффективности стратегии не обязательно доигрывать
- ▶ эффект от действия может проявиться позднее

Markov Decision Process

Определения:

- ▶ состояния среды (state) $s \in S$
- ▶ действия агента (action) $a \in A$
- ▶ награды (reward) $r \in R$
- ▶ динамика переходов

$P(s_{t+1}|s_t, a_t, \dots, s_{t-i}, a_{t-i}, \dots, s_0, a_0) = P(s_{t+1}|s_t, a_t)$ (допущение Маркова)

- ▶ функция выигрыша

▶ стратегия агента
(policy)

$$\pi(a|s)$$

$$r_t = r(s_t, a_t)$$

(допущение Маркова)

- ▶ абсолютный выигрыш (total reward)

Задача:

$$\pi(a|s) : \mathbb{E}_\pi[R] \rightarrow \max$$

$$R = \sum_t r_t$$

Важные функции

Средний абсолютный выигрыш:

$$\mathbb{E}_{s_0 \sim p(s_0)}, \mathbb{E}_{a_0 \sim \pi(a|s_0)}, \mathbb{E}_{s_1, r_0 \sim P(s', r|s, a)} \dots [r_0 + r_1 + \dots + r_T]$$

Функция ценности состояния (value function):

$$V^\pi(s) = \mathbb{E}_\pi[R_t | s_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right],$$

Функция ценности действия в состоянии:

$$Q^\pi(s, a) = \mathbb{E}_\pi[R_t | s_t = s, a_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right],$$

где π — это стратегия, которой следует агент, $\gamma \in [0, 1]$

TD-обучение

temporal difference

Произвольно инициализируем функцию $V(s)$ и стратегию π
Далее повторяем

- ▶ инициализируем s
- ▶ для каждого шага агента
 - ▶ выбрать a по стратегии π
 - ▶ сделать действие a , получить результат r и следующее состояние s'
 - ▶ обновить функцию $V(s)$ по формуле

$$V(s) = V(s) + \alpha (r + \gamma V(s') - V(s))$$

- ▶ перейти к следующему шагу, присвоив $s := s'$

Уравнение Беллмана

для оптимальной функции ценности Q^*

$$Q^*(s, a) = \mathbb{E}_\pi \left[r_t + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a \right]$$

Замечание

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

Пересчет полезности состояний

$$V(s) = \max_a [r(s, a) + \gamma V(s'(s, a))]$$

то есть при вероятностных переходах

$$V(s) = \max_a [r(s, a) + \gamma \mathbb{E}_{s' \sim P(s'|s, a)} V(s')]$$

Итеративная формула пересчета полезности состояния

$$\forall s \quad V_0(s) = 0$$

$$V_{i+1}(s) = \max_a [r(s, a) + \gamma \mathbb{E}_{s' \sim P(s'|s, a)} V_i(s')]$$

Пересчет полезности состояний

$$V(s) = \max_a [r(s, a) + \gamma V(s'(s, a))]$$

то есть при вероятностных переходах

$$V(s) = \max_a [r(s, a) + \gamma \mathbb{E}_{s' \sim P(s'|s, a)} V(s')]$$

Итеративная формула пересчета полезности состояния

$$\forall s \quad V_0(s) = 0$$

$$V_{i+1}(s) = \max_a [r(s, a) + \gamma \mathbb{E}_{s' \sim P(s'|s, a)} V_i(s')]$$

Замечание

Чтобы пользоваться этой формулой на практике, нужно знать вероятности переходов $P(s'|s, a)$

Полезность действия

$$Q(s, a) = r(s, a) + \gamma V(s')$$

Стратегия игры определяется следующим образом

$$\pi(s) : \arg \max_a Q(s, a)$$

Полезность действия

$$Q(s, a) = r(s, a) + \gamma V(s')$$

Стратегия игры определяется следующим образом

$$\pi(s) : \arg \max_a Q(s, a)$$

Снова из-за стохастичности

$$Q(s, a) = \mathbb{E}_{s'} [r(s, a) + \gamma V(s')]$$

Можно оценить матожидание без явного распределения методом Монте-Карло и усреднением по исходам:

$$Q(s_t, a_t) \leftarrow \alpha \left(r_t + \gamma \max_a Q(s_{t+1}, a) \right) + (1 - \alpha) Q(s_t, a_t)$$

DQN (Deep Q-Learning Network)

Среда — эмулятор игр Atari, каждый кадр $210 \times 160\text{pix}$ 128col



Pong

Breakout

Space Invaders

Seaquest

Beam Rider

Состояния s : 4 последовательных кадра, сжатых до 84×84

Действия a : от 4 до 18, в зависимости от игры

Награды r : текущий SCORE в игре

Функция ценности $Q(s, a; w)$: ConvNN со входом s и $|A|$ выходами

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

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

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

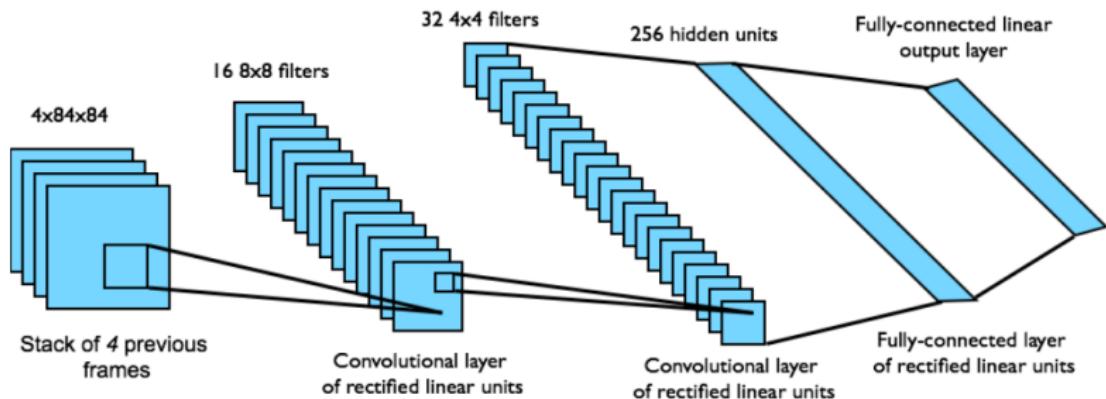
Функция потерь для обучения нейросетевой модели $Q(s, a; w)$:

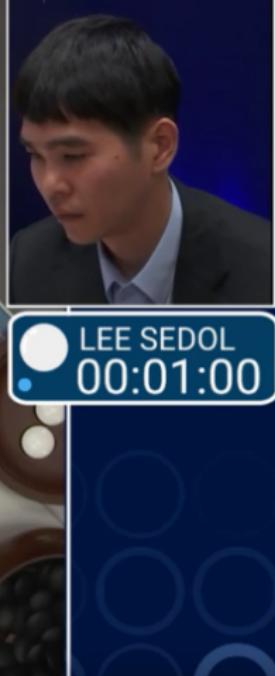
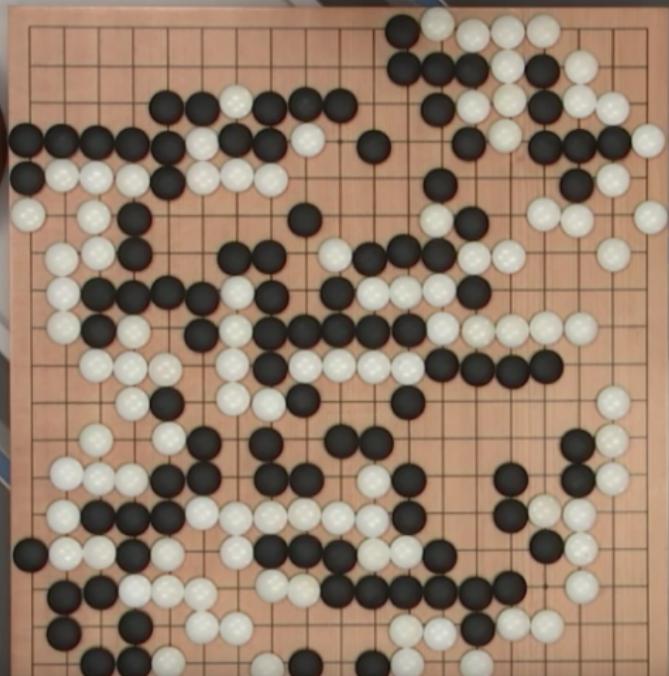
$$\mathcal{L}(w) = (Q(s_t, a_t; w) - y_t)^2$$

Стochastic gradient SGD (по мини-батчам длины 32):

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

Архитектура сети для функции ценности





PRODUCTION

DeepMind
STAR CRAFT
DEMONSTRATION



Резюме

- ▶ познакомились с постановкой задачи обучения с подкреплением
- ▶ вспомнили многоруких бандитов
- ▶ познакомились с методами кросс-энтропии, TD-обучения и Q-обучения
- ▶ обсудили Deep Q-learning Network
- ▶ НЕ обсудили центральный метод в роботике — градиентный спуск по стратегиям (policy gradient)

Резюме

- ▶ познакомились с постановкой задачи обучения с подкреплением
- ▶ вспомнили многоруких бандитов
- ▶ познакомились с методами кросс-энтропии, TD-обучения и Q-обучения
- ▶ обсудили Deep Q-learning Network
- ▶ НЕ обсудили центральный метод в роботике — градиентный спуск по стратегиям (policy gradient)

Что ещё можно посмотреть?

- ▶ главная книга по теме: [Reinforcement Learning: An Introduction](#), Richard S. Sutton and Andrew G. Barto
- ▶ <https://spinningup.openai.com/en/latest>
- ▶ Yuxi Li. Resources for Deep Reinforcement Learning. 2018