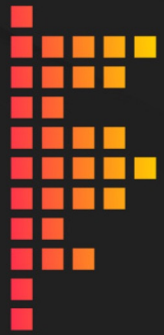


Reinforcement learning

Введение в DeepRL

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Ставим “+”, если все хорошо
“-”, если есть проблемы

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в учебной группе

Задаем вопрос
в чат

Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения

Индивидуально

Время, необходимое
на активность

Пишем в чат

Говорим голосом

Документ

Ответьте себе или
задайте вопрос

Тема вебинара

Reinforcement learning. Введение в DeepRL

Игорь Стурейко

Руководитель курсов: **Reinforcement Learning, ML Professional, ML Basic, MLOps, FinML**

Teamlead, главный инженер проекта,
Физический факультет МГУ, PhD теоретическая физика

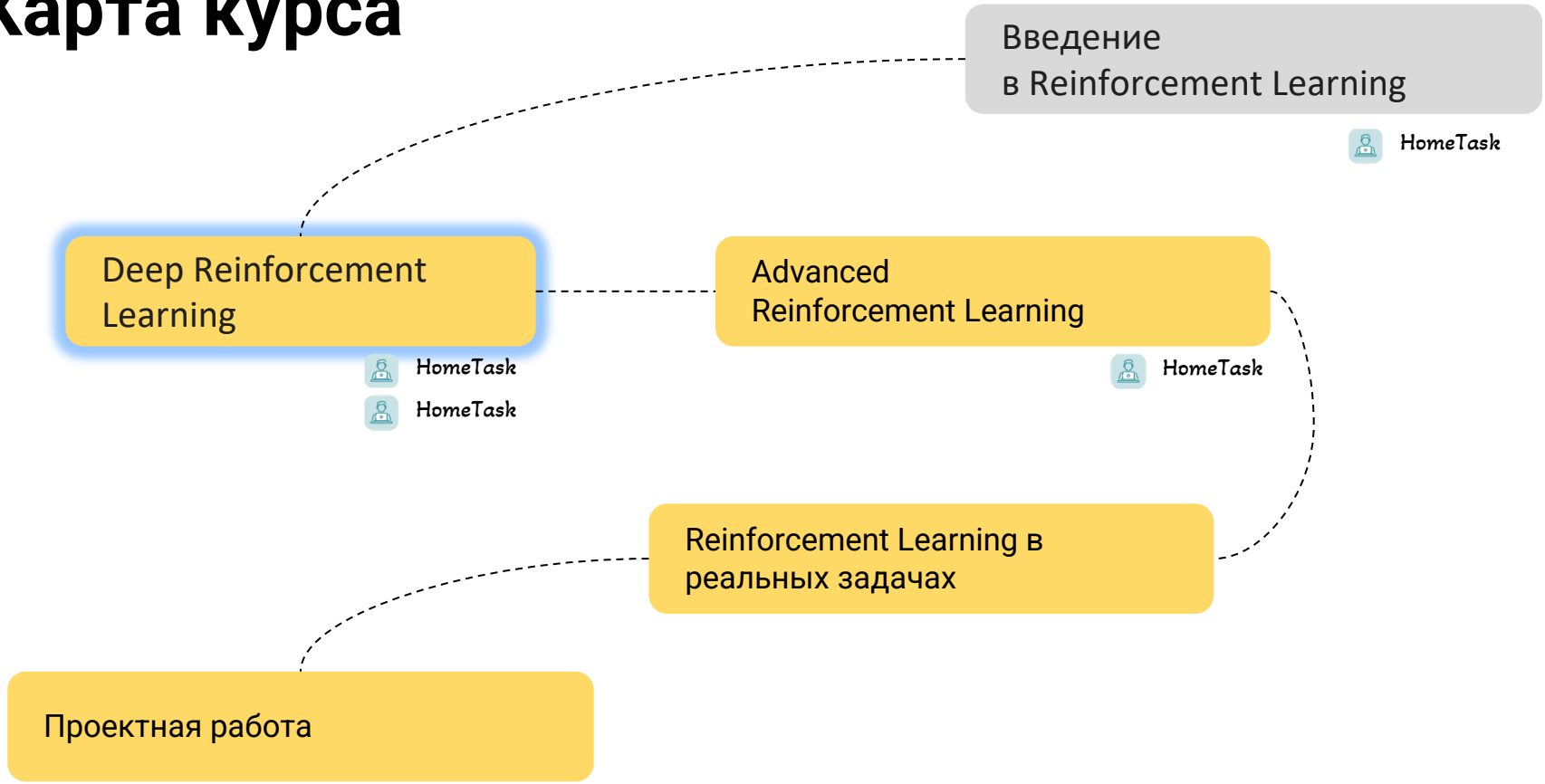
Опыт:
Более 15 лет занимался прикладной математикой и мат моделированием (Data Scientist) (Python, C++) в НИИ ПАО Газпром

@stureiko (TG)

LinkedIn: igor-stureiko

@rl_fintech (Мой канал о моделях в бизнесе)

Карта курса


Введение
в Reinforcement Learning
HomeTask
Deep Reinforcement
Learning
HomeTask
HomeTask
Advanced
Reinforcement Learning
HomeTask
Reinforcement Learning в
реальных задачах
Проектная работа

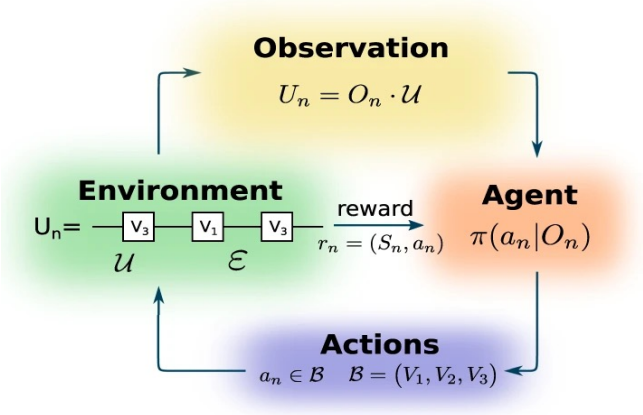
Программа курса

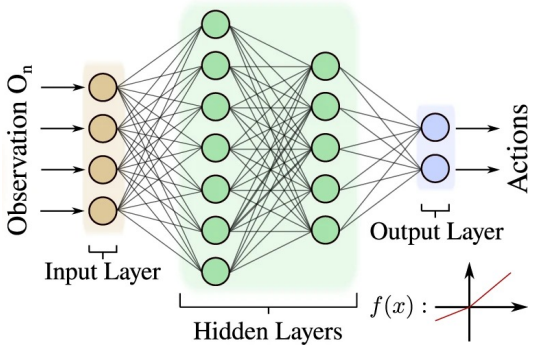
Deep Reinforcement Learning

- Введение в Deep Reinforcement Learning
- Deep Q-Network (DQN) алгоритм
- Policy Gradient (PG) алгоритм
- Actor-Critic алгоритм
- TRPO -> PPO
- DDPG -> TD3 -> LSTM-TD3

Маршрут вебинара

- Знакомство
- Применение классического RL в непрерывных средах
- Проблемы классического RL
- Идея Deep RL
- Deep Cross-Entropy
- Заключение





Цели вебинара

К концу занятия вы сможете

1. Понять сложности применения классических алгоритмов к непрерывным средам
2. Понять подход к применению нейросети для предсказания политики
3. Понять применение нейросети в алгоритме кросс-энтропии

Смысл

Зачем вам это уметь

1. Понимать границы применимости классических алгоритмов
2. Понимать переход к Deep Reinforcement Learning

Постановка задачи

Markov Property

$$\mathbb{P}[S_{t+1}|S_tA_t] = \mathbb{P}[S_{t+1}|S_1A_1, S_2A_2, \dots, S_tA_t]$$

$$\mathbb{P}[R_t|S_tA_t] = \mathbb{P}[R_t|S_1A_1, S_2A_2, \dots, S_tA_t] = 1$$

Markov Decision Process $\langle \mathcal{S}, \mathcal{S}_F, \mathcal{A}, \mathcal{P}, \mathcal{P}_0, \mathcal{R}, \gamma \rangle$

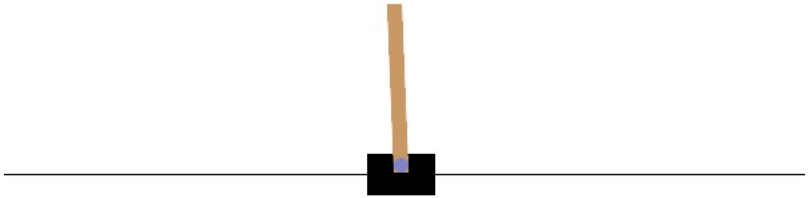
- $\mathcal{S}$ – is an infinite ($|\mathcal{S}| = n$) state space
- $\mathcal{A}$ – is a finite/infinite ($|\mathcal{A}| = m$) action space
- $\mathcal{P}$ – is a known deterministic transition probability function

  $$\mathcal{P}(s'|s,a) = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]$$
- $\mathcal{P}_0$ – is a deterministic initial state probability function
- $\mathcal{R}$ – is a known reward function

  $$\mathcal{R}(s,a) = R_t, \Leftrightarrow \mathbb{P}[R_t|S_t = s, A_t = a] = 1$$
- $\gamma \in [0, 1]$ – is a discount coefficient

Применение классического RL в непрерывной среде

CartPole



Action Space	Discrete(2)
Observation Space	Box([-4.8000002e+00 -3.4028235e+38 -4.1887903e-01 -3.4028235e+38], [4.8000002e+00 3.4028235e+38 4.1887903e-01 3.4028235e+38], (4,), float32)

SARSA and Q-learning

- Устанавливаем $Q(s,a)$=0,K>0,ε=1.
- Для каждого $k \in 1,K$:

  двигаясь по текущей траектории из состояния S_t действуя A_t в силу политики $\pi(\cdot|S_t)$, и обновляя политику $\pi=\varepsilon-greedy(Q)$ получаем R_t и переходим в состояние S_{t+1} с действием $A_{t+1} \sim \pi(\cdot|S_{t+1})$
- Обновляем Q:

Q-learning	SARSA
$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha\left(R_t \gamma \max_{a'} Q(S_{t+1},a') - Q(S_t,A_t)\right)$	$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha\left(R_t + \gamma Q(S_{t+1},A_{t+1}) - Q(S_t,A_t)\right)$

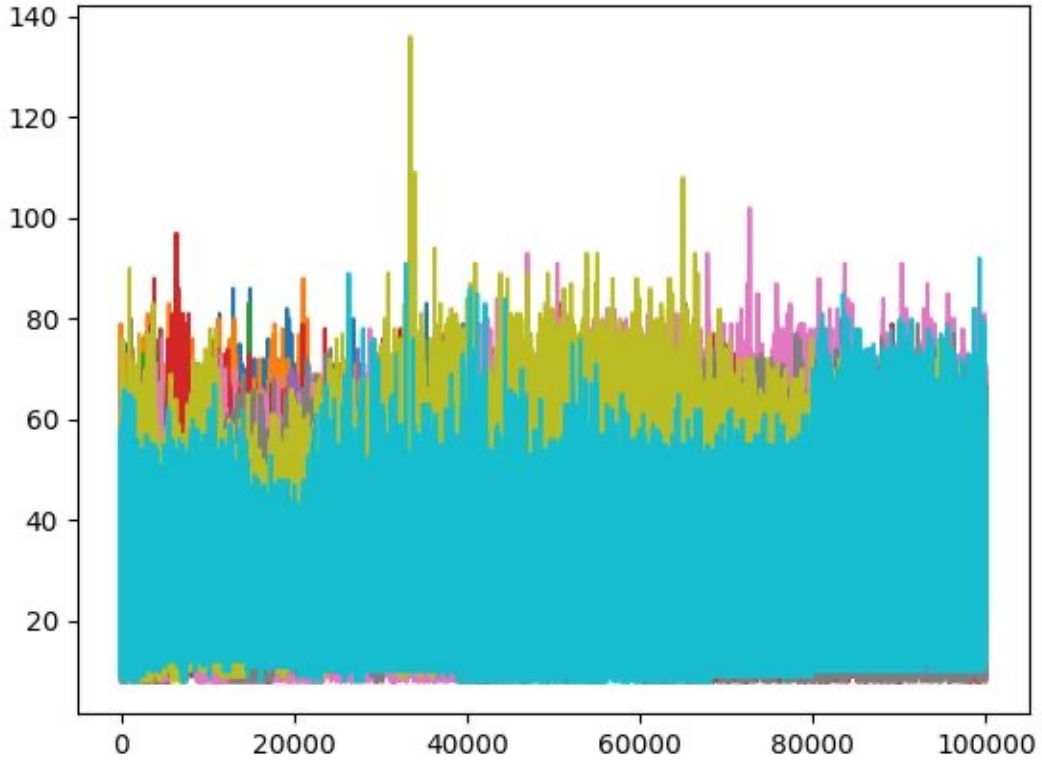
- Обновляем ε=1/k

CartPole дискретизация среды

- Разделим пространство состояний на дискретные интервалы. Тогда размер q-таблицы в памяти будет следующий:
- 10 интервалов = $\frac{10^4 \cdot 2 \cdot 32\text{bit}}{1024^2} = 0.6\ \text{Mb}$, 20.000 состояний q-table
- 20 интервалов = $\frac{20^4 \cdot 2 \cdot 32\text{bit}}{1024^2} = 9.77\ \text{Mb}$, 320.000 состояний q-table
- 50 интервалов = $\frac{50^4 \cdot 2 \cdot 32\text{bit}}{1024^2} = 381{,}74\ \text{Mb}$, 12.500.000 состояний q-table
- 100 интервалов = $\frac{100^4 \cdot 2 \cdot 32\text{bit}}{1024^2} = 6.1\ \text{Gb}$, 200.000.000 состояний q-table

Когда мы рассматривали Taxi там было 500 состояний.

CartPole дискретизация среды



n = 20

discrete_state_n = 320.000

episode_n = 100.000 x 10

Deep cross-entropy

Метод кросс-энтропии

1. Оценка политики $\mathbb{E}_{\pi}[G]$
   - Устанавливаем начальное состояние S_0 и политику π_0
   - Действуем A_0=$\pi(S_0)$, получаем награду R_0=(S_0,A_0) и переходим в состояние S_1
   - ...
   - Получили траекторию τ=$\{S_0,A_0,S_1,A_1,...,S_F\}$, и награду по траектории $G(\tau) = \sum_{t=0}^{T-1} \gamma R_t$

2. Улучшение политики $\pi \rightarrow \pi'$ $(\mathbb{E}_{\pi'}[G] \geq \mathbb{E}_{\pi}[G])$
   - Выбираем $k\%$ лучших траекторий
   - "Улучшаем" политику ← собираем средние действия по каждому состоянию
   - Назначаем новой политикой среднее действие по лучшим траекториям в каждом состоянии

Deep Cross-entropy

Пусть у нас $\mathcal{S} \in \mathbb{R}^n$ и $\mathcal{A} \in \mathbb{R}^m$, т.е. пространство действий и пространство состояний конечномерно и непрерывно.

Будем использовать нейросеть F^θ для аппроксимации политики, т.е $F^\theta: \mathbb{R}^n \mapsto \mathbb{R}^m$, θ – веса нейросети.
Будем действовать в среде N раз (эпизодов обучения) и соберем K траекторий для каждого эпизода.
Тогда на каждом эпизоде обучения $n \in [1, N]$:

$$\pi_n(s) = \left[F^{\theta_n}(s) + noise(\varepsilon)\right]_{\mathcal{A}}$$

Оценка политики $\mathbb{E}_{\boldsymbol{\pi}}[\boldsymbol{G}]$:

Подавая на вход сети состояние s получаем оценку действия a и действуя получаем следующее состояние.
Получаем траекторию в среде.

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

Нам необходимо найти «наилучшие» действия по конечному набору точек – задача регрессии.

Deep CrossEntropy

Улучшение политики для каждого эпизода обучения $n \in [1, N]$:

Выбираем «элитные» траектории и усредняем действие на них для каждого состояния.

Затем считаем лосс и обновляем веса модели:

$$Loss(\theta_n) = \frac{1}{|\tau_n|} \sum_{(a|s) \in \tau_n} \left\| F^{\theta_n}(s) - a \right\|^2$$

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta Loss(\theta_n)$$

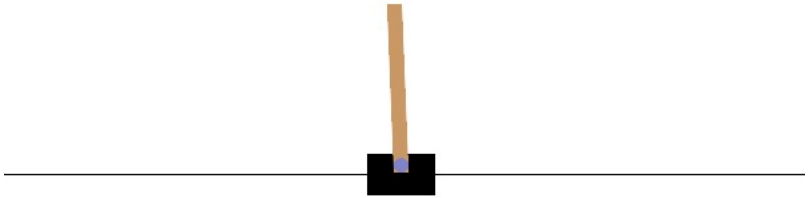
Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

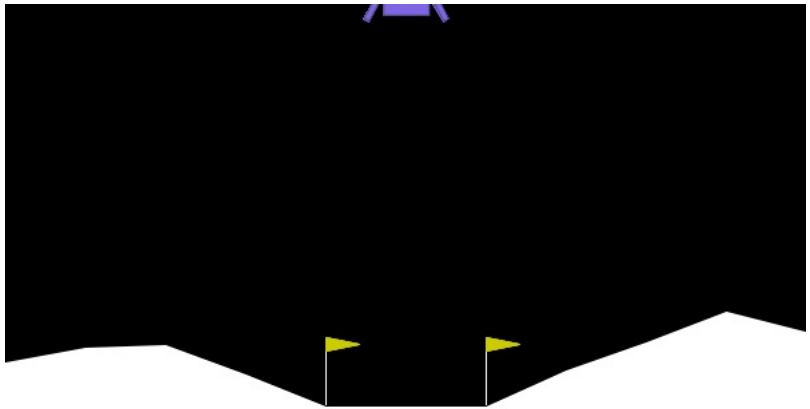
Окружения

CartPole



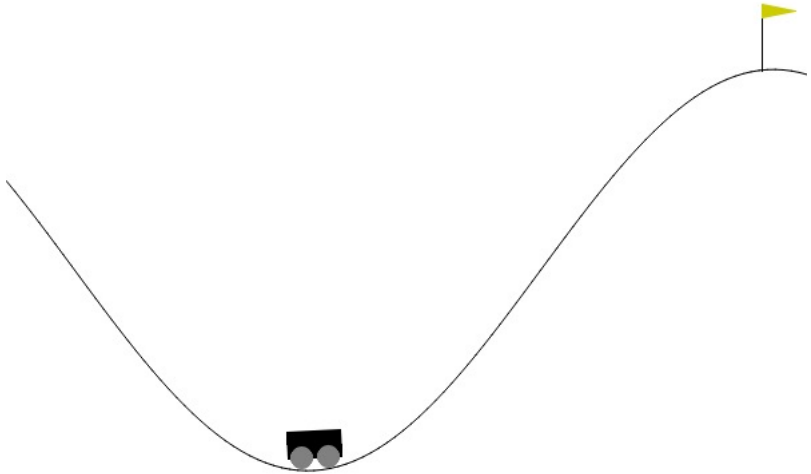
Action Space	Discrete(2)
Observation Space	Box([-4.8000002e+00 -3.4028235e+38 -4.1887903e-01 -3.4028235e+38], [4.8000002e+00 3.4028235e+38 4.1887903e-01 3.4028235e+38], (4,), float32)

LunaLander



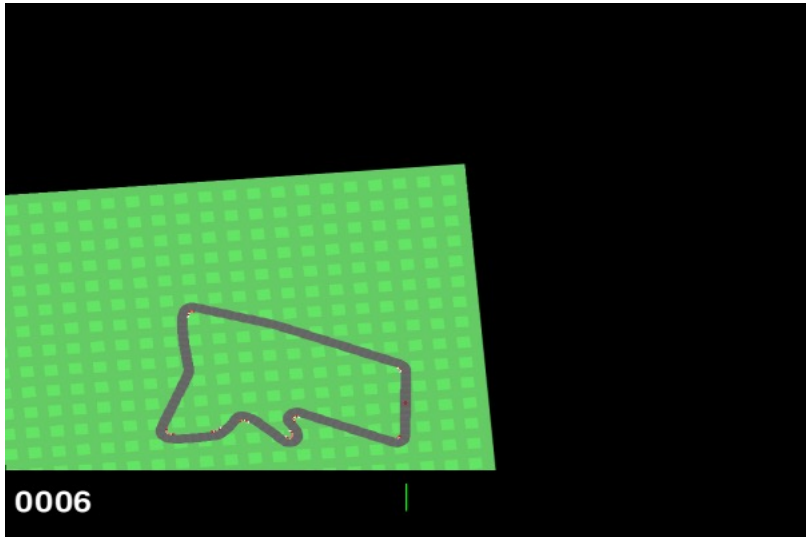
Action Space	Discrete(4)
Observation Space	Box([-1.5 -1.5 -5. -5. -3.1415927 -5. -0. 0.], [1.5 1.5 5. 5. 3.1415927 5. 1. 1.], (8,), float32)

MountainCar



Action Space	Discrete(3)
Observation Space	Box([-1.2 -0.07], [0.6 0.07], (2,), float32)

CarRacing



Action Space	Box([-1. 0. 0.], 1.0, (3,), float32)
Observation Space	Box(0, 255, (96, 96, 3), uint8)

Atari battleZone

Action Space	Discrete(18)
Observation Space	Box(0, 255, (210, 160, 3), uint8)

Практика

Практика

1. Дискретизация CartPole
2. Deep Cross-Entropy метод

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

Заполните, пожалуйста, опрос о занятии по ссылке в чате

Спасибо за внимание!

Следующие вебинары

Deep q-learning (DQN)

Игорь Стурейко

Руководитель курсов: **Reinforcement Learning, ML Professional, ML Basic, MLOps, FinML**

Teamlead, главный инженер проекта,
Физический факультет МГУ, PhD теоретическая физика

Опыт:
Более 15 лет занимался прикладной математикой и мат моделированием (Data Scientist) (Python, C++) в НИИ ПАО Газпром

@stureiko (TG)

LinkedIn: igor-stureiko

@rl_fintech (Мой канал о моделях в бизнесе)