Искусственный интеллект: эволюция идей от Фрэнсиса Бэкона до фундаментальных моделей и ChatGPT

Воронцов Константин Вячеславович
д.ф.-м.н., профессор РАН • руководитель лаборатории машинного обучения и семантического анализа Института ИИ МГУ, профессор ВМК МГУ

Институт искусственного интеллекта МГУ

Научный семинар «Спектральная теория дифференциальных операторов» под руководством академика РАН, профессора В. А. Садовничего

• 19 апреля 2023 •

Содержание

Принцип эмпирической индукции

«Не следует полагаться на сформулированные аксиомы и формальные базовые понятия, какими бы привлекательными и справедливыми они не казались. Законы природы нужно «расшифровывать» из фактов опыта. Следует искать правильный метод анализа и обобщения опытных данных; здесь логика Аристотеля не подходит в силу её абстрактности, оторванности от реальных процессов и явлений.»

Фрэнсис Бэкон (1561–1626)

Таблицы открытия: множество случаев x, когда

- свойство y *присутствовало* $y(x) = 1$
- свойство y *отсутствовало* $y(x) = 0$
- наблюдалось изменение *степени* свойства $y(x)$

Фрэнсис Бэкон. Новый органон. 1620.

Метод наименьших квадратов (Гаусс, 1795)

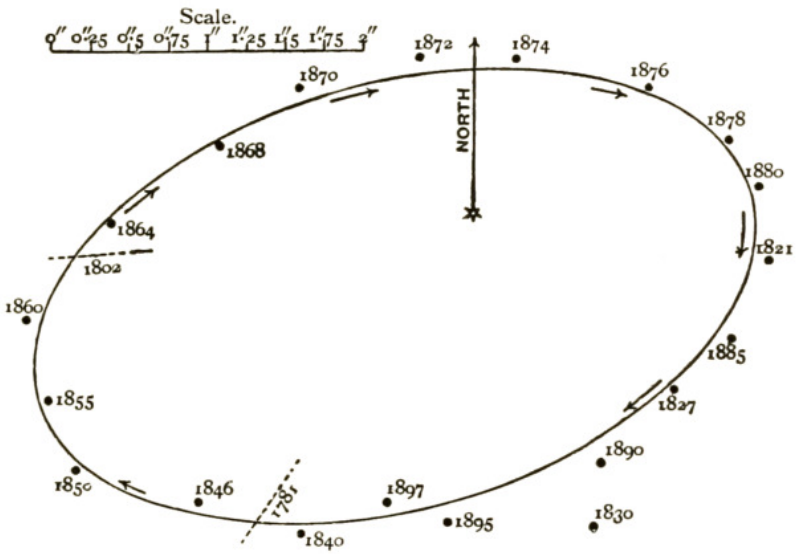
Линейная модель регрессии:

$$a(x, w) = \sum_{j=1}^{n} w_j f_j(x), \qquad w \in \mathbb{R}^n.$$

Метод наименьших квадратов:

$$Q(w) = \sum_{i=1}^{\ell} \big(a(x_i, w) - y_i\big)^2 \to \min_w.$$

Карл Фридрих Гаусс (1777–1855)



«Our principle, which we have made use of since 1795, has lately been published by Legendre...»

C.F.Gauss. Theory of the motion of the heavenly bodies moving about the Sun in conic sections. 1809.

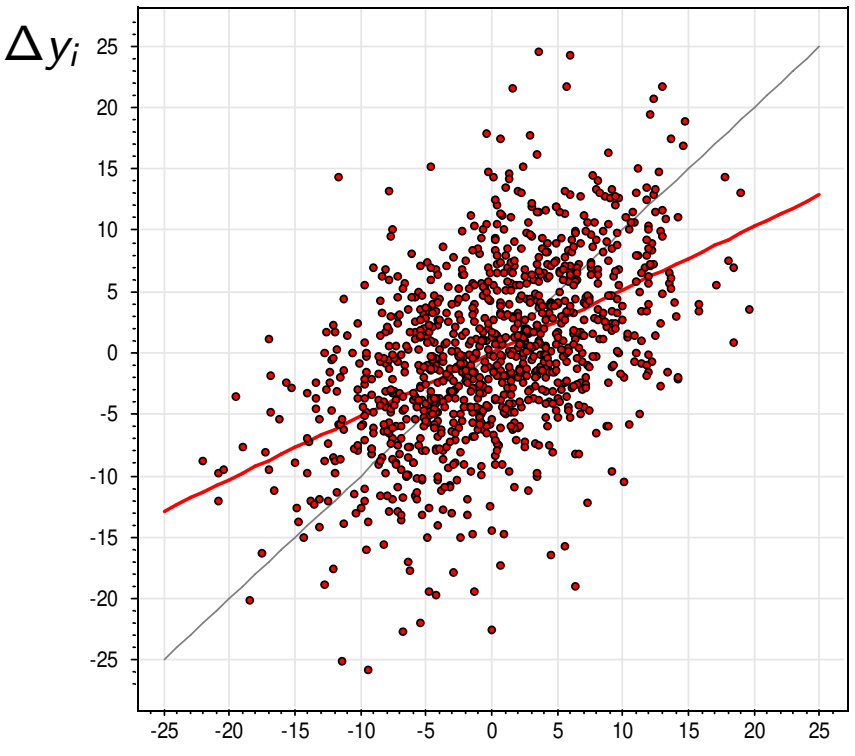
История термина «регрессия» (Гальтон, 1886)

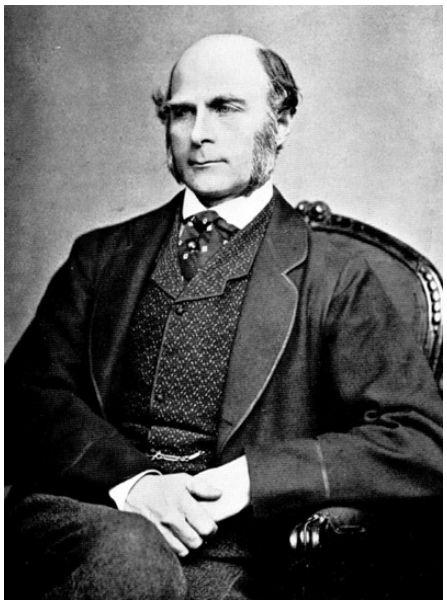
Исследование наследственности роста.

отклонение роста от среднего в популяции:

Δx_i — отклонение роста отца

Δy_i — отклонение роста взрослого сына

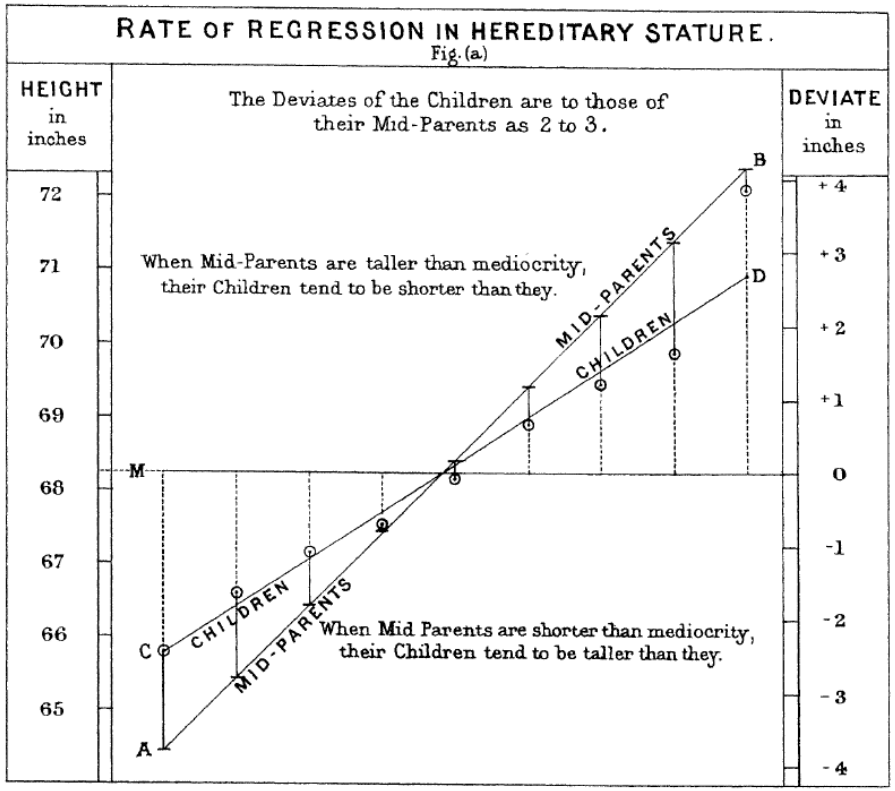


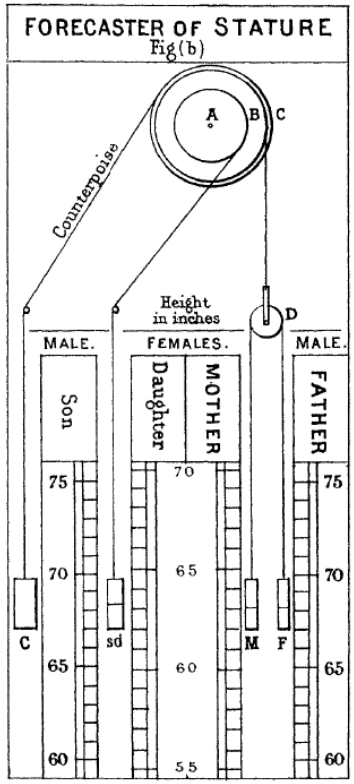


Фрэнсис Гальтон (1822–1911)

История термина «регрессия» (Гальтон, 1886)

«Регрессия к посредственности» — угол наклона меньше 1
Скрытый смысл: обратный ход исследования от данных к модели





Galton F. Regression towards mediocrity in hereditary stature. 1886.

Общая оптимизационная задача машинного обучения

Дано: обучающая выборка объектов $\{x_i\}_{i=1}^{\ell}$

Найти: вектор параметров w предсказательной модели $a(x, w)$

Критерий: минимум эмпирического риска

$$\sum_{i=1}^{\ell} L_i(w) \;\to\; \min_w$$

где $L_i(w)$ — функция потерь модели $a(x, w)$ на объекте x_i

Обобщение: минимум регуляризованного эмпирического риска

$$\sum_{i=1}^{\ell} L_i(w) + \sum_{j=1}^{r} \tau_j R_j(w) \;\to\; \min_w$$

где R_j — регуляризаторы, τ_j — коэффициенты регуляризации

Оптимизационная задача обучения модели регрессии

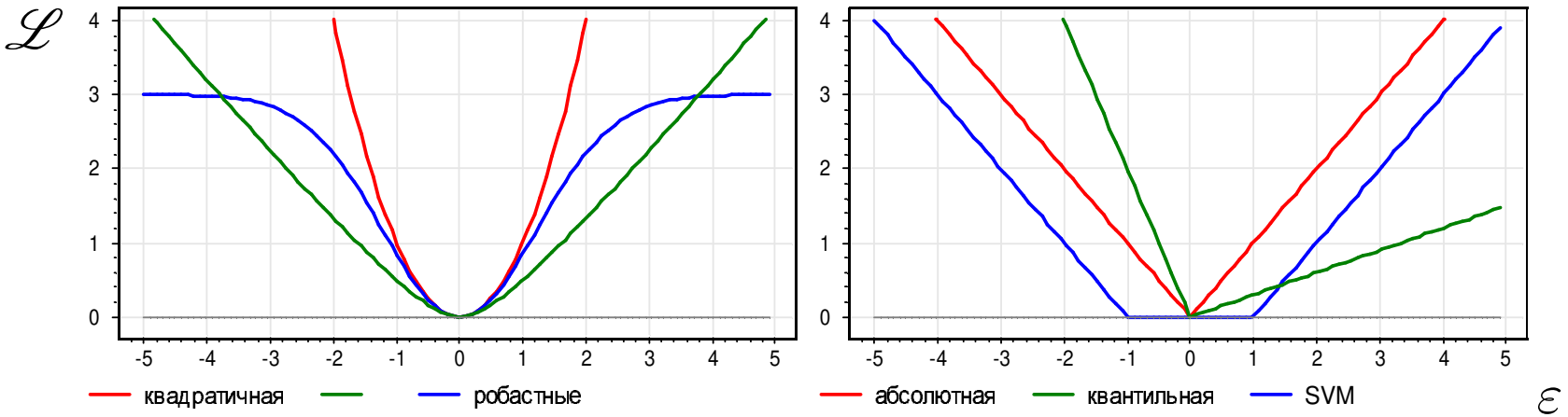
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \mathbb{R}$

Найти: вектор параметров w модели регрессии $a(x, w)$

Критерий: *минимизация эмпирического риска*

$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w) - y_i\big) \;\to\; \min_w$$

Унимодальные функции потерь $\mathscr{L}(\varepsilon)$ от невязки $\varepsilon = a(x, w) - y$:



Оптимизационная задача обучения модели классификации

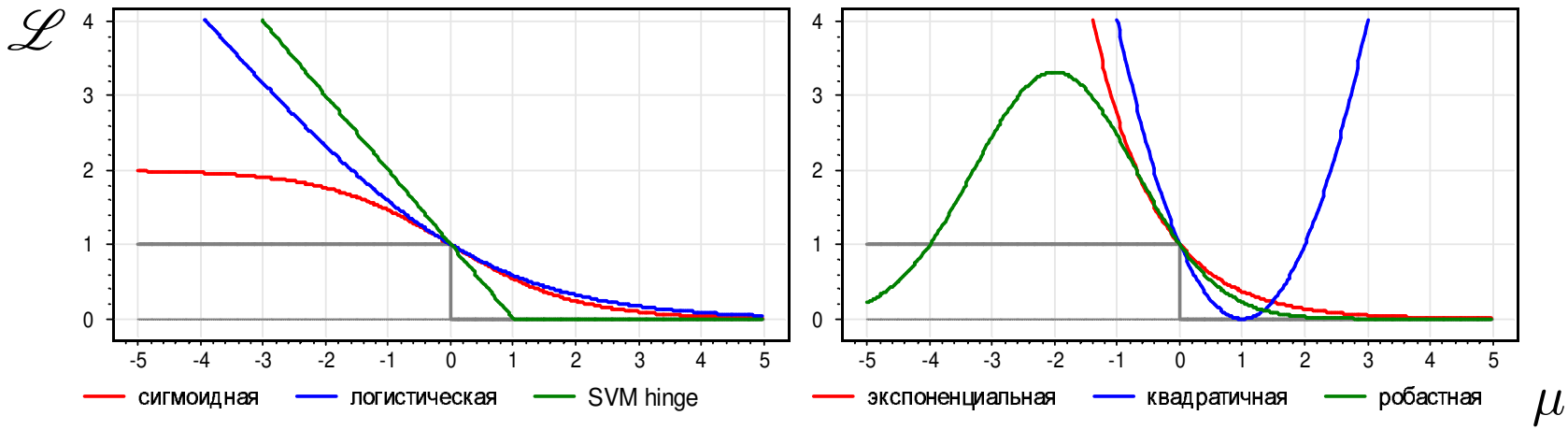
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \{-1, +1\}$

Найти: вектор w модели классификации $a(x, w) = \operatorname{sign} g(x, w)$

Критерий: *аппроксимация эмпирического риска*

$$\sum_{i=1}^{\ell} \big[g(x_i, w)y_i < 0\big] \leqslant \sum_{i=1}^{\ell} \mathscr{L}\big(g(x_i, w)y_i\big) \;\to\; \min_w$$

Убывающие функции потерь $\mathscr{L}(\mu)$ от отступа $\mu = g(x, w)y$:



Задача максимизации правдоподобия

Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in Y$, $|Y| < \infty$

Найти: модель классификации: $a(x, w) = \arg\max_{y \in Y} g(x_i, w_y)$

модель вероятности того, что объект x относится к классу y:

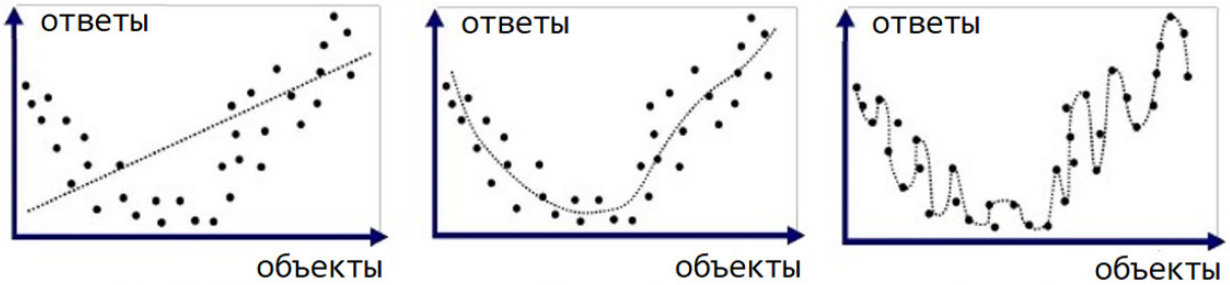
$$P(y|x, w) = \frac{\exp g(x, w_y)}{\sum_{z \in Y} \exp g(x, w_z)} = \underset{y \in Y}{\mathrm{SoftMax}}\, g(x, w_y),$$

где SoftMax: $\mathbb{R}^Y \to \mathbb{R}^Y$ — гладкое преобразование произвольного вектора в нормированный вектор дискретного распределения.

Критерий: максимум правдоподобия (log-loss):

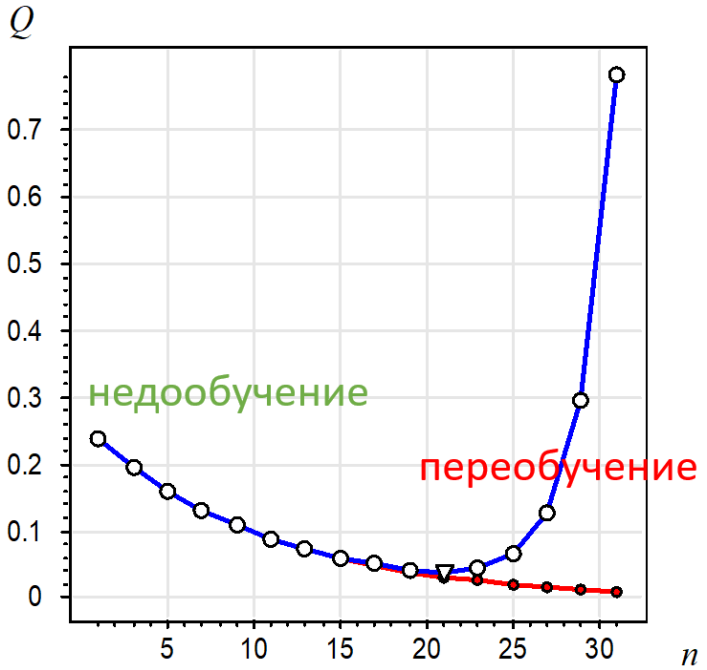
$$-\sum_{i=1}^{\ell} \ln P(y_i|x_i, w) \;\to\; \min_w$$

Проблемы недообучения и переобучения



недообучение переобучение

- Недообучение (underfitting): модель слишком проста, недостаточное число параметров n
- Переобучение (overfitting): модель слишком сложна, избыточное число параметров n



Понятие обучаемости в SLT, Statistical Learning Theory

Семейство классификаторов A *обучаемо*:

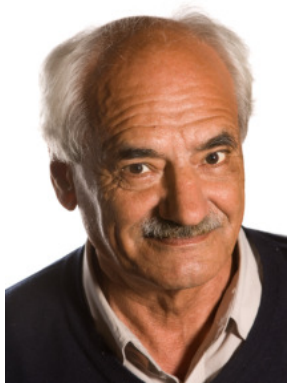
$$\mathsf{P}\left\{\sup_{a\in A}\left|P(a)-\nu(a,X^\ell)\right|>\varepsilon\right\}\leqslant\eta,$$

$P(a)$ — вероятность ошибки классификатора,
$\nu(a,X^\ell)$ — эмпирический риск (частота ошибки классификатора a на выборке).

Основные результаты VC-теории:

- Обосновано ограничение сложности A
- Понятие *ёмкости* семейства, VCdim
- Метод структурной минимизации риска

Владимир Наумович Вапник



Алексей Яковлевич Червоненкис (1938–2014)

Вапник В. Н., Червоненкис А. Я.
Теория распознавания образов. М.: Наука, 1974.

Задачи, некорректно поставленные по Адамару

Причина переобучения — потеря устойчивости модели по мере роста числа параметров (степеней свободы)

Задача корректно поставлена, если её решение:

- существует
- единственно
- устойчиво

Задачи восстановления зависимостей по эмпирическим данным — всегда некорректно поставленные.

Жак Саломон Адамар (1865–1963)

Регуляризация — это введение ограничений на модель.

Hadamard J. Sur les problèmes aux dérivées partielles et leur signification physique. 1902.
Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. 1974.

Регуляризация линейных моделей

Регуляризатор — аддитивная добавка к основному критерию:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle, y_i\big) + \tau\, \text{штраф}(w) \;\to\; \min_w$$

где $\mathscr{L}(a, y)$ — функция потерь, τ — коэффициент регуляризации

Регуляризаторы для линейных моделей:

L_2-регуляризация (Ridge, SVM): $\quad \text{штраф}(w) = \sum_{j=1}^{n} w_j^2$

L_1-регуляризация (LASSO): $\quad \text{штраф}(w) = \sum_{j=1}^{n} |w_j|$

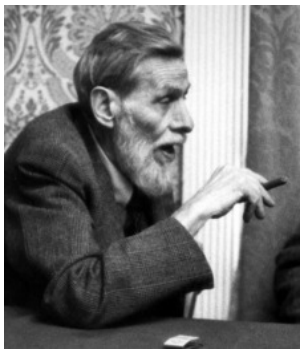
L_0-регуляризация (AIC, BIC): $\quad \text{штраф}(w) = \sum_{j=1}^{n} \big[w_j \neq 0\big]$

Искусственный нейрон — линейная модель классификации

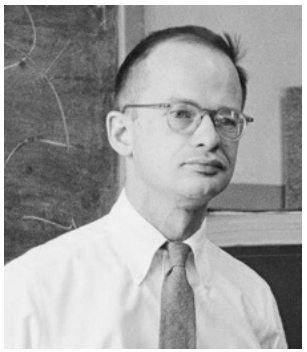
Линейная модель нейрона (1943):

$$a(x,w) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right)$$

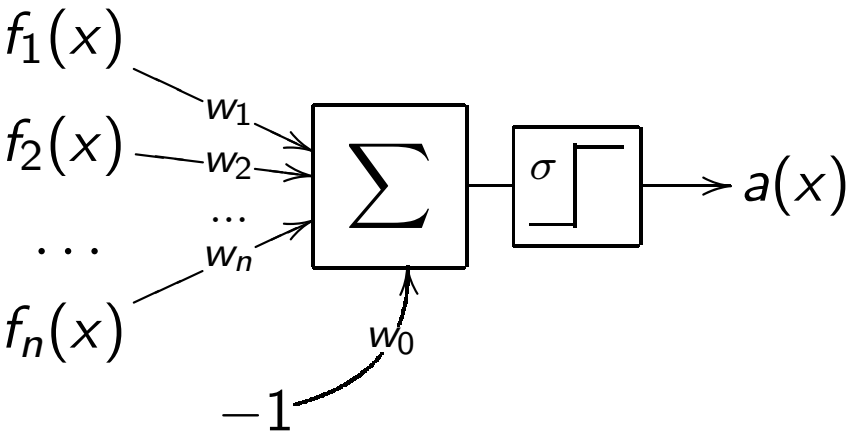
$f_j(x)$ — признаки объекта x
w_j — веса признаков
w_0 — порог активации
$\sigma(z)$ — функция активации

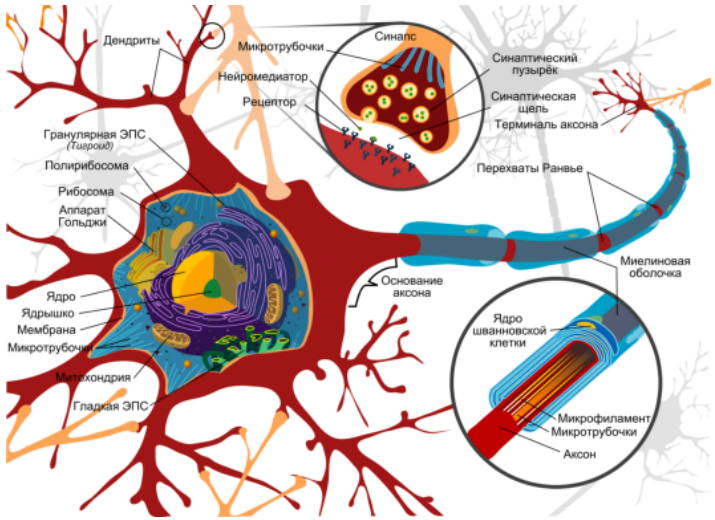


Уоррен МакКаллок (1898–1969)



Вальтер Питтс (1923–1969)





Двухслойные сети — аппроксиматоры непрерывных функций

Функция $\sigma(z)$ — *сигмоида*, если $\lim\limits_{z\to-\infty}\sigma(z)=0$ и $\lim\limits_{z\to+\infty}\sigma(z)=1$.

Теорема Цыбенко (1989)

Если $\sigma(z)$ — непрерывная сигмоида, то для любой непрерывной на $[0,1]^n$ функции $f(x)$ существуют такие значения параметров H, $\alpha_h \in \mathbb{R}$, $w_h \in \mathbb{R}^n$, $w_0 \in \mathbb{R}$, что двухслойная сеть

$$a(x) = \sum_{h=1}^{H} \alpha_h \sigma\big(\langle x, w_h\rangle - w_0\big)$$

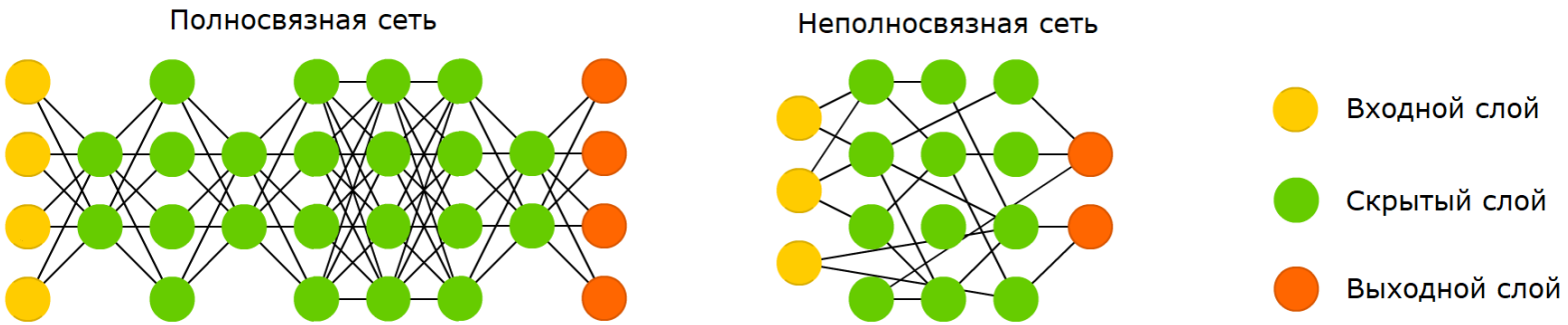
равномерно приближает $f(x)$ с любой точностью ε:

$$\big|a(x) - f(x)\big| < \varepsilon, \text{ для всех } x \in [0,1]^n.$$

George Cybenko. Approximation by Superpositions of a Sigmoidal function. Mathematics of Control, Signals, and Systems. 1989.

Глубокие нейронные сети (Deep Neural Network, DNN)

1965: первые глубокие нейронные сети
1997: рекуррентная сеть LSTM для анализа последовательностей
2012: свёрточная сеть для классификации изображений AlexNet



- *Архитектура сети* — структура слоёв и связей между ними, позволяющая наделять DNN нужными свойствами
- DNN позволяют принимать на входе и генерировать на выходе *сложно структурированные данные*

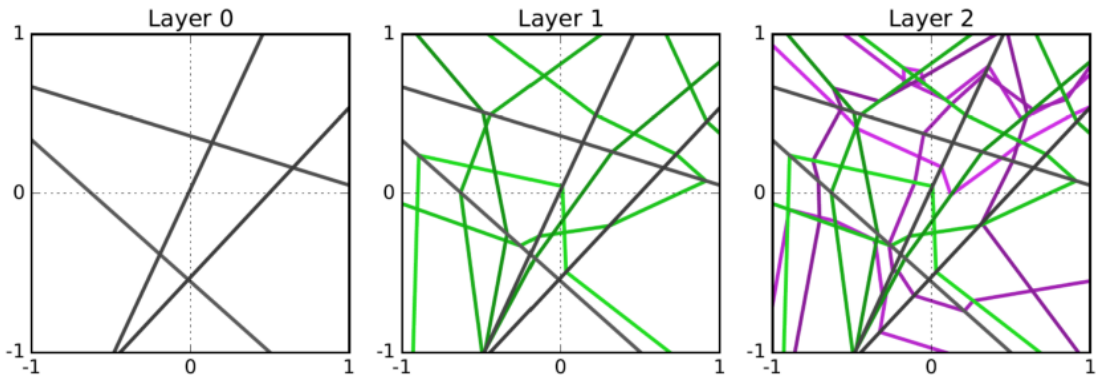
Ива́хненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965
Hochreiter S., Schmidhuber J. Neural Computation, 9(8), 1997
Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012

Глубина важнее ширины

A_{LH}^n — семейство полносвязных многослойных сетей $a(x, w)$:
n признаков, L слоёв, H нейронов в каждом слое, $x \in \mathbb{R}^n$,
функции активации кусочно-линейные: ReLU, hard-tanh и т.п.

Мера разнообразия семейства A_{LH}^n — максимальное число участков линейности $a(x, w)$ — выпуклых многогранников в $\mathbb{R}^n$.

Пример. Участки линейности, $n = 2$, $L = 3$, $H = 4$:



Теорема. Разнообразие семейства A_{LH}^n растёт как $O(H^{nL})$.

M.Raghu et al. On the Expressive Power of Deep Neural Networks, 2016.

Генерация признаков для распознавания изображений

Классический подход к распознаванию изображений:

Конструируемые признаки изображения → **Любой обучаемый классификатор**

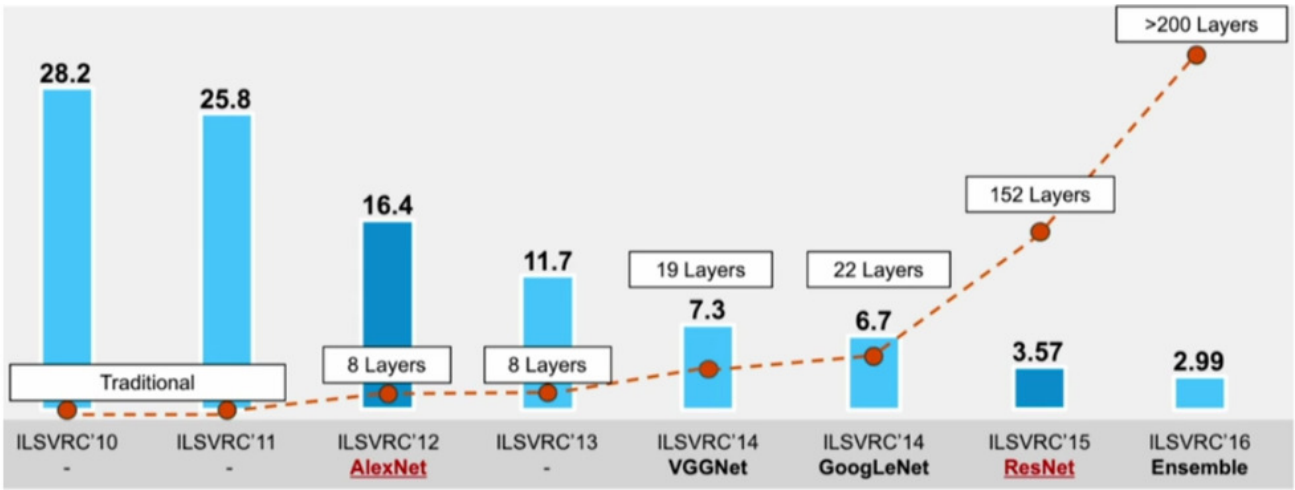
Современный подход — end-to-end Deep Learning:

Обучаемые ANN признаки изображения → **Обучаемый ANN классификатор**

Совместное обучение

Sanjeev Arora. Toward theoretical understanding of deep learning. ICML-2018 Tutorial
https://unsupervised.cs.princeton.edu/deeplearningtutorial.html

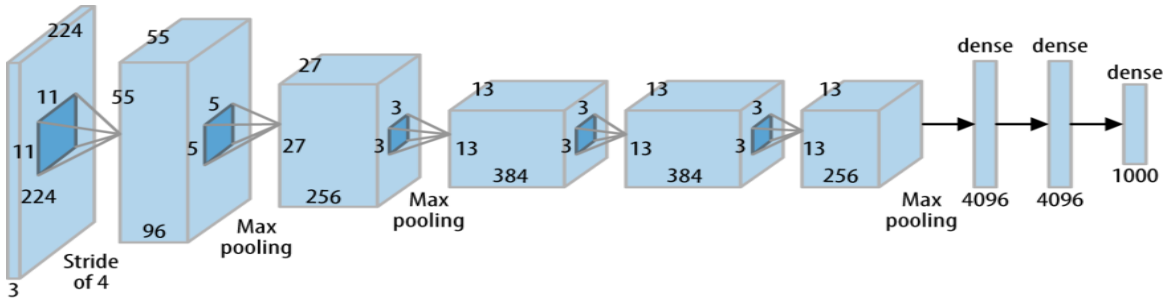
Глубокие свёрточные сети для классификации изображений



Старт в 2009

Человеческий уровень ошибок 5% пройден в 2015

Свёрточные нейронные сети
AlexNet (2012)
ResNet (2015)



Li Fei-Fei et al. ImageNet: A large-scale hierarchical image database. 2009
Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012
Kaiming He et al. Deep residual learning for image recognition. 2015

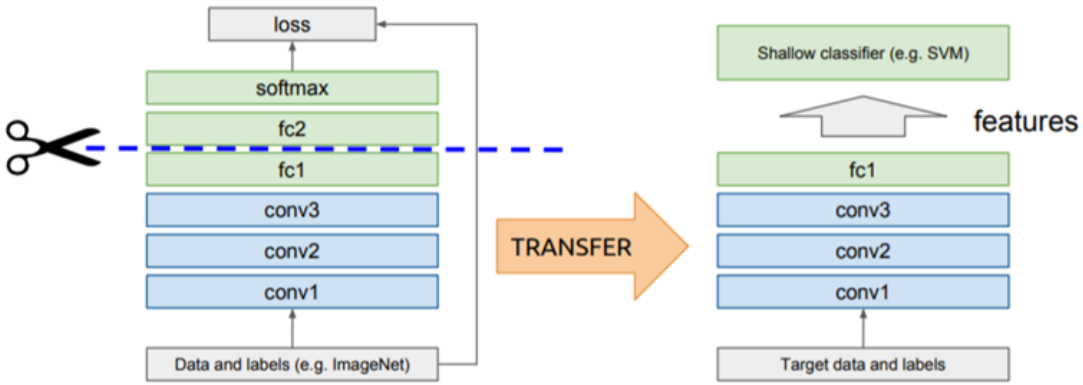
Предобучение (pre-training), перенос обучения (transfer learning)

Обучение модели векторизации $z = f(x, \alpha)$ на выборке $\{x_i\}_{i=1}^{\ell}$:

$$\sum_{i=1}^{\ell} \mathscr{L}_i\big(g(f(x_i, \alpha), \beta)\big) \;\to\; \min_{\alpha,\beta}$$

Обучение целевой модели $y = g(z, \beta)$ на малых данных:

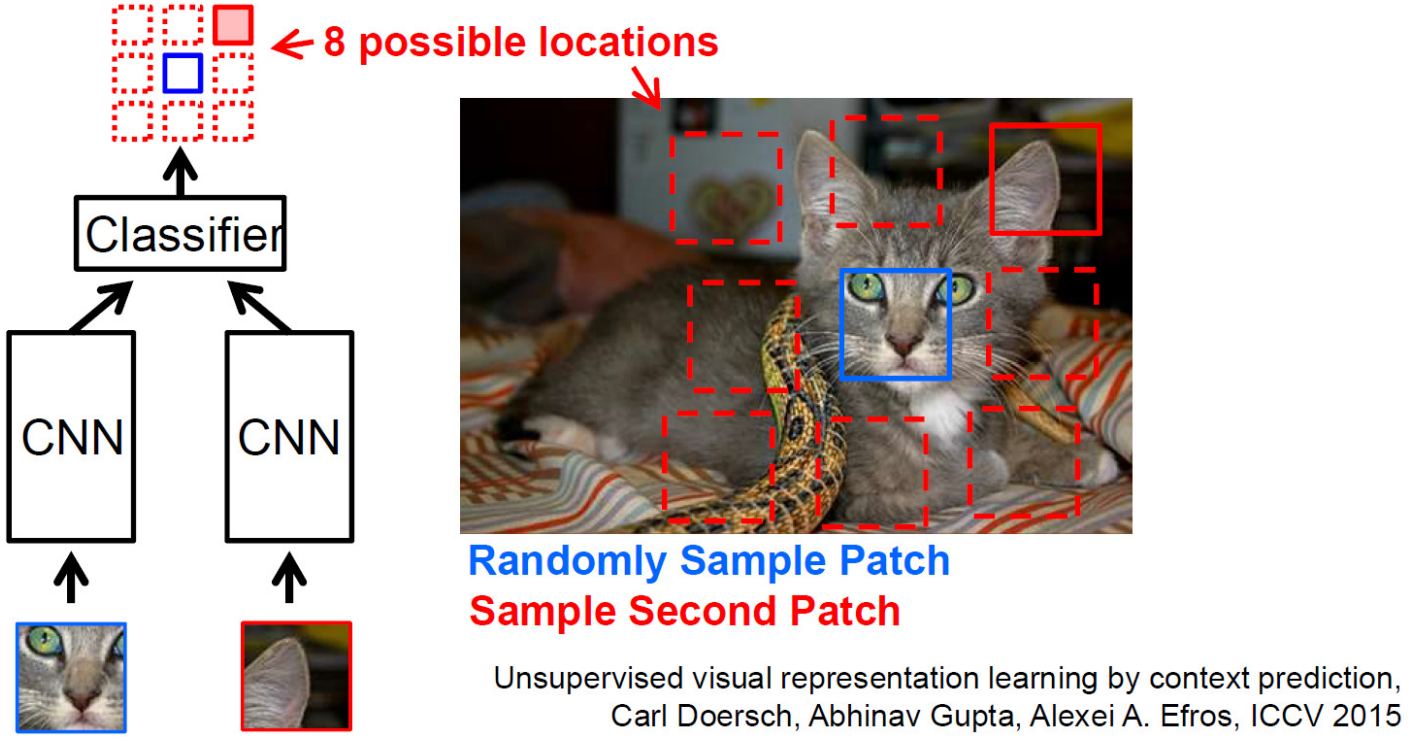
$$\sum_{i=1}^{m} \mathscr{L}'_i\big(g'(f(x'_i, \alpha), \beta')\big) \;\to\; \min_{\beta'}$$



Sinno Jialin Pan, Qiang Yang. A Survey on Transfer Learning. 2009
J.Yosinski et al. How transferable are features in deep neural networks? 2014.

Самостоятельное обучение (self-supervised learning)

Модель векторизации $z = f(x, \alpha)$ обучается предсказывать взаимное расположение пар фрагментов одного изображения



Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

Преимущество: сеть выучивает векторные представления объектов без размеченной обучающей выборки (без ImageNet).

Многозадачное обучение (multi-task learning)

$z = f(x, \alpha)$ — векторизация, универсальная для всех моделей
$g_t(z, \beta)$ — специфичная часть модели для задачи $t \in T$

Одновременное обучение модели f по задачам X_t, $t \in T$:

$$\sum_{t \in T} \sum_{i \in X_t} \mathscr{L}_{ti}\big(g_t(f(x_{ti}, \alpha), \beta_t)\big) \;\to\; \min_{\alpha, \{\beta_t\}}$$

Обучаемость (learnability): качество решения отдельной задачи $\langle X_t, \mathscr{L}_t, g_t \rangle$ улучшается с ростом объёма выборки $\ell_t = |X_t|$.

Learning to learn: качество решения каждой из задач $t \in T$ улучшается с ростом как ℓ_t, так и общего числа задач $|T|$.

Few-shot learning: для решения новой задачи t достаточно небольшого числа примеров, иногда даже одного.

M.Crawshaw. Multi-task learning with deep neural networks: a survey. 2020
Y.Wang et al. Generalizing from a few examples: a survey on few-shot learning. 2020

Обучаемая векторизация данных: задача автокодировщика

Дано: обучающая выборка объектов $\{x_i\}_{i=1}^{\ell}$

Найти: $z = f(x, \alpha)$ — модель кодировщика (encoder)
$\hat{x} = g(z, \beta)$ — модель декодировщика (decoder)

Критерий: качество реконструкции исходных объектов

$$\sum_{i=1}^{\ell} \mathscr{L}\big(g(f(x_i, \alpha), \beta), x_i\big) \;\to\; \min_{\alpha,\beta}$$

Квадратичная функция потерь: $\mathscr{L}(\hat{x}, x) = \|\hat{x} - x\|^2$

Для *линейного автокодировщика* $f(x, A) = Ax$, $g(z, B) = Bz$,
задача сводится к (низкоранговому) матричному разложению:

$$\sum_{i=1}^{\ell} \|BAx_i - x_i\|^2 \to \min_{A,B}$$

Автокодировщики для векторизации и обучения с учителем

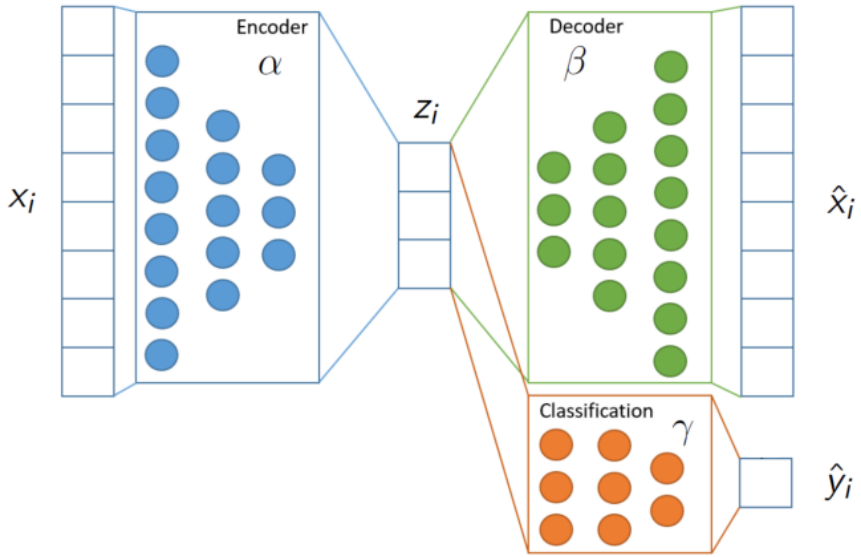
Данные: размеченные $(x_i, y_i)_{i=1}^{k}$, неразмеченные $(x_i)_{i=k+1}^{\ell}$

Найти:

$z_i = f(x_i, \alpha)$ — кодировщик

$\hat{x}_i = g(z_i, \beta)$ — декодировщик

$\hat{y}_i = \hat{y}(z_i, \gamma)$ — предиктор



Функции потерь:

$\mathscr{L}(\hat{x}_i, x_i)$ — реконструкция

$\tilde{\mathscr{L}}(\hat{y}_i, y_i)$ — предсказание

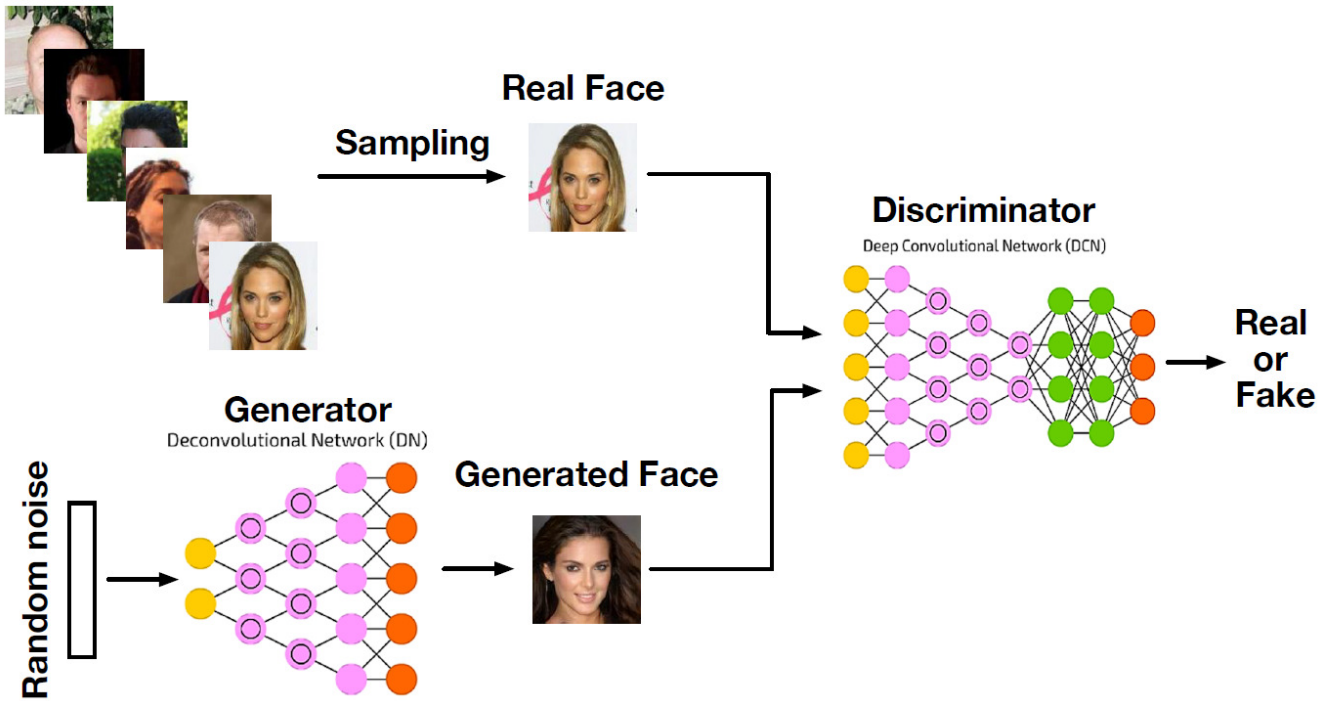
Критерий: совместное обучение автокодировщика и предсказательной модели (классификации, регрессии или др.):

$$\sum_{i=1}^{\ell} \mathscr{L}\big(g(f(x_i, \alpha), \beta), x_i\big) + \lambda \sum_{i=1}^{k} \tilde{\mathscr{L}}(\hat{y}(f(x_i, \alpha), \gamma), y_i) \;\to\; \min_{\alpha,\beta,\gamma}$$

Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders. 2020

Генеративная состязательная сеть (Generative Adversarial Net)

Генератор $G(z)$ учится порождать объекты x из шума z
Дискриминатор $D(x)$ учится отличать их от реальных объектов



Antonia Creswell et al. Generative Adversarial Networks: an overview. 2017.
Zhengwei Wang et al. Generative Adversarial Networks: a survey and taxonomy. 2019.
Chris Nicholson. A Beginner's Guide to Generative Adversarial Networks. https://pathmind.com/wiki/generative-adversarial-network-gan. 2019.

Постановка задачи GAN

Дано: выборка объектов $\{x_i\}_{i=1}^{\ell}$

Найти две вероятностные модели:

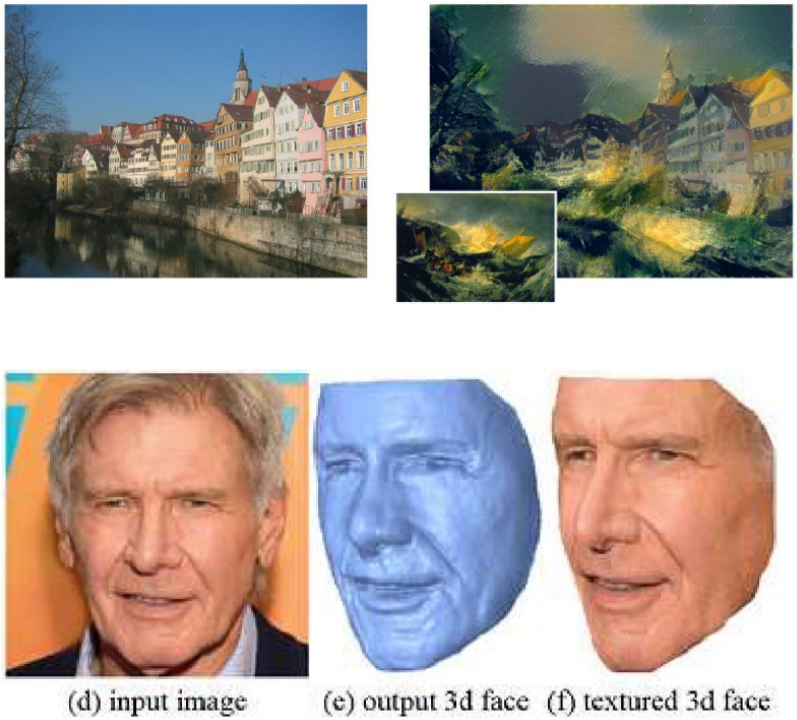
- модель $x = G(z, \alpha)$ генерации $x \sim p(x|z, \alpha)$ из шума z
- дискриминативная модель $D(x, \beta) = p(1|x, \beta)$

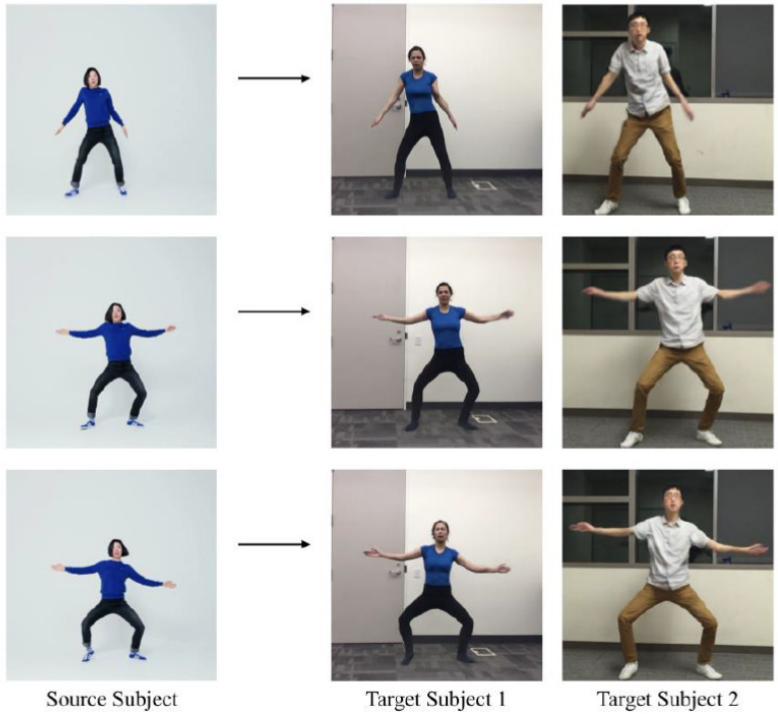
Критерий: log правдоподобия дискриминативной модели;
генератор $G(z)$ учится порождать объекты x из шума z,
дискриминатор $D(x)$ учится отличать их от реальных объектов,
в антагонистической игре генератора против дискриминатора:

$$\sum_{i=1}^{\ell} \ln D(x_i, \beta) + \ln\big(1 - D\big(G(z_i, \alpha), \beta\big)\big) \;\to\; \max_{\beta} \min_{\alpha}$$

Ian Goodfellow et al. Generative Adversarial Nets. 2014

Примеры GAN для синтеза изображений и видео





Chuan Li, Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. 2016.

Xiaoxing Zeng, Xiaojiang Peng, Yu Qiao. DF2Net: A Dense Fine Finer Network for Detailed 3D Face Reconstruction. ICCV-2019.

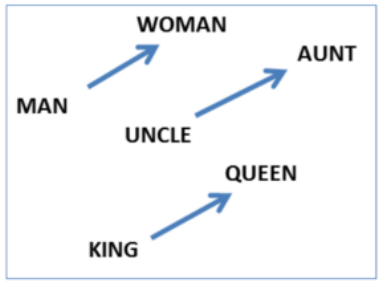
Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros. Everybody Dance Now. ICCV-2019.

Эволюция подходов машинного обучения в анализе текстов

Декомпозиция задач по уровням пирамиды NLP

- морфологический анализ, лемматизация, опечатки
- синтаксический анализ, выделение терминов, NER
- семантический анализ, выделение фактов, тем

Модели векторных представлений (эмбедингов) слов на основе матричных разложений



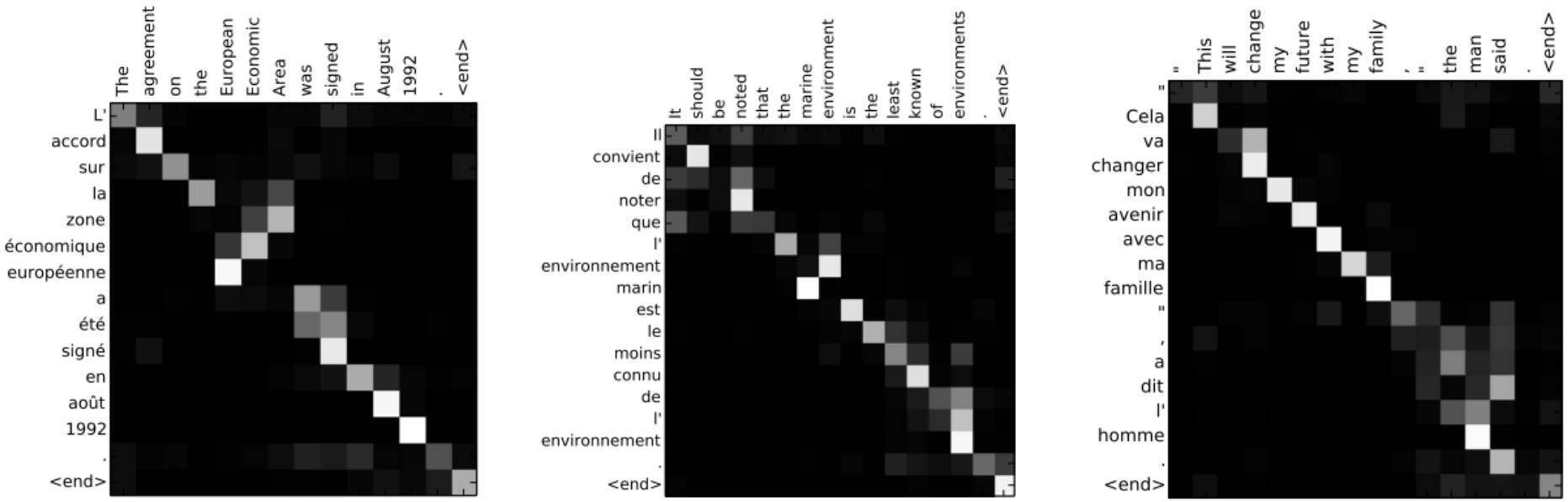
- модели дистрибутивной семантики:
  word2vec [Mikolov, 2013], FastText [Bojanowski, 2016]
- тематические модели LDA [Blei, 2003], ARTM [2014]

Нейросетевые модели локальных контекстов

- рекуррентные нейронные сети
- модели внимания и трансформеры:
  BERT [2018], GPT-3 [2020], GPT-4 [2023]

$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d}}\right) V$$

Модели внимания для машинного перевода



Вход: $\{x_i\}$ — последовательность слов входного языка

Выход: $\{y_t\}$ — последовательность слов выходного языка

Интерпретация: матрица a_{it} показывает, на какие слова x_i модель обращает внимание, генерируя слово перевода y_t

---

Bahdanau et al. Neural machine translation by jointly learning to align and translate. 2015.

Модели внимания для аннотирования изображений

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Подсвечены области, на которые модель обращает внимание, когда генерирует подчёркнутое слово в аннотации изображения

Kelvin Xu et al. Show, attend and tell: neural image caption generation with visual attention. 2016

Трасформер для машинного перевода

Трасформер (transformer) — это нейросетевая архитектура на основе моделей внимания и полносвязных слоёв

Схема преобразований данных в машинном переводе:

- $S = (w_1, \ldots, w_n)$ — слова предложения на входном языке
  $\downarrow$ обучаемая или пред-обученная векторизация слов
- $X = (x_1, \ldots, x_n)$ — векторы слов входного предложения
  $\downarrow$ трансформер-кодировщик
- $Z = (z_1, \ldots, z_n)$ — контекстные векторы слов
  $\downarrow$ трансформер-декодировщик, похож на кодировщика
- $Y = (y_1, \ldots, y_m)$ — векторы слов выходного предложения
  $\downarrow$ генерация слов из построенной языковой модели
- $\tilde{S} = (\tilde{w}_1, \ldots, \tilde{w}_m)$ — слова предложения на выходном языке

Vaswani et al. (Google) Attention is all you need. 2017.

Модель внимания Query–Key–Value

q — *вектор-запрос* для трансформации в вектор-контекст c
$K = (k_1, \ldots, k_n)$ — *векторы-ключи*, сравниваемые с запросом
$V = (v_i, \ldots, v_n)$ — *векторы-значения*, образующие контекст

Модель внимания — трёхслойная сеть, вычисляющая c как выпуклую комбинацию векторов v_i, релевантных запросу q:

$$c = \text{Attn}(q, K, V) = \sum_i v_i \,\text{SoftMax}_i\, a(k_i, q),$$

где $a(k, q)$ — *оценка релевантности* ключа k запросу q,
например $a(k, q) = k^\intercal q$ или $k^\intercal W q$ с матрицей параметров W

Модель внутреннего внимания (самовнимания, self-attention):

$$c_i = \text{Attn}(W_q x_i, W_k X, W_v X)$$

трансформирует входную последовательность $X = (x_1, \ldots, x_n)$
в выходную последовательность векторов контекста $(c_1, \ldots, c_n)$

Архитектура трансформера-кодировщика

1. Добавляются позиционные векторы p_i:
$h_i = x_i + p_i, \quad H = (h_1, \ldots, h_n)$
$d = \dim x_i, p_i, h_i = 512$
$\dim H = 512 \times n$

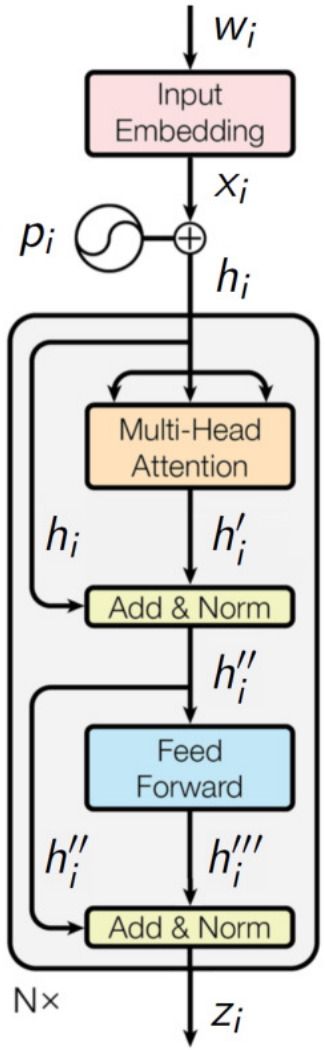
2. Многомерное самовнимание:
$h_i^j = \text{Attn}(W_q^j h_i, W_k^j H, W_v^j H)$
$j = 1, \ldots, J = 8$
$\dim h_i^j = 64$
$\dim W_q^j, W_k^j, W_v^j = 64 \times 512$

3. Конкатенация:
$h'_i = \text{MH}_j(h_i^j) \equiv \left[h_i^1 \cdots h_i^J\right]$
$\dim h'_i = 512$

4. Сквозная связь + нормировка уровня:
$h''_i = \text{LN}(h'_i + h_i; \mu_1, \sigma_1)$
$\dim h''_i, \mu_1, \sigma_1 = 512$

5. Полносвязная 2х-слойная сеть FFN:
$h'''_i = W_2\,\text{ReLU}(W_1 h''_i + b_1) + b_2$
$\dim W_1 = 2048 \times 512$
$\dim W_2 = 512 \times 2048$

6. Сквозная связь + нормировка уровня:
$z_i = \text{LN}(h'''_i + h''_i; \mu_2, \sigma_2)$
$\dim z_i, \mu_2, \sigma_2 = 512$

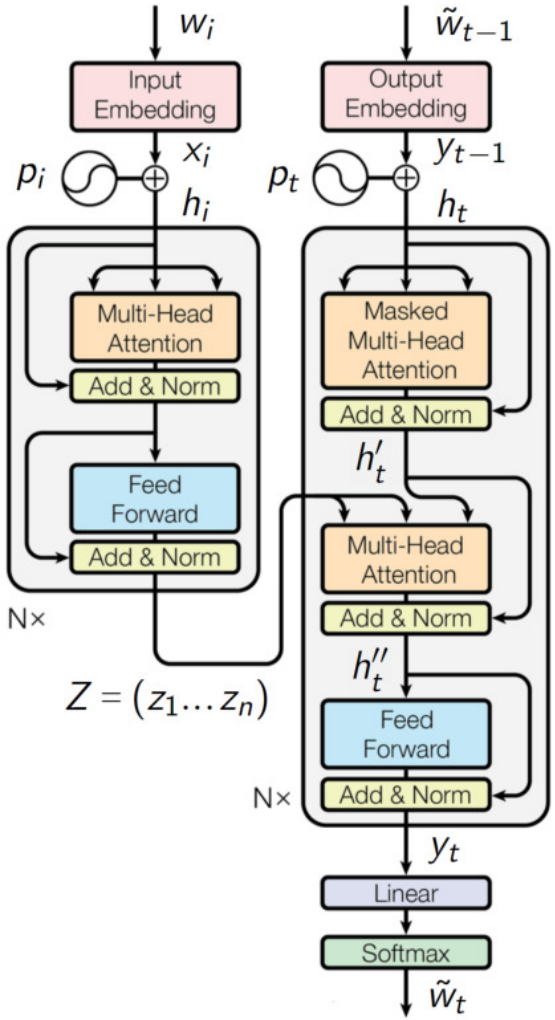


Архитектура трансформера декодировщика

Авторегрессионный синтез последовательности:
$y_0 = \langle \text{BOS} \rangle$ — эмбеддинг символа начала;
для всех $t = 1, 2, \ldots$:

1. Маскирование «данных из будущего»:
   $h_t = y_{t-1} + p_t; \quad H_t = (h_1, \ldots, h_t)$
2. Многомерное самовнимание:
   $h'_t = \text{LN} \circ \text{MH}_j \circ \text{Attn}(W^j_q h_t, W^j_k H_t, W^j_v H_t)$
3. Многомерное внимание на кодировку Z:
   $h''_t = \text{LN} \circ \text{MH}_j \circ \text{Attn}(\tilde{W}^j_q h'_t, \tilde{W}^j_k Z, \tilde{W}^j_v Z)$
4. Двухслойная полносвязная сеть:
   $y_t = \text{LN} \circ \text{FFN}(h''_t)$
5. Линейный предсказывающий слой:
   $p(\tilde{w}|t) = \text{SoftMax}_{\tilde{w}}(W_y y_t + b_y)$

генерация $\tilde{w}_t = \arg\max_{\tilde{w}} p(\tilde{w}|t)$ **пока** $\tilde{w}_t \neq \langle \text{EOS} \rangle$



Vaswani et al. (Google) Attention is all you need. 2017.

BERT (Bidirectional Encoder Representations from Transformers)

Трансформер BERT — это кодировщик без декодировщика, предобучаемый на большой текстовой коллекции для решения широкого класса задач NLP

Схема преобразования данных в задачах NLP:

- $S = (w_1, \ldots, w_n)$ — токены предложения входного текста
  $\downarrow$ обучение эмбедингов вместе с трансформером
- $X = (x_1, \ldots, x_n)$ — эмбединги токенов входного предложения
  $\downarrow$ трансформер кодировщика
- $Z = (z_1, \ldots, z_n)$ — трансформированные эмбединги
  $\downarrow$ дообучение на конкретную задачу
- Y — выходной текст / разметка / классификация и т.п.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language) BERT: pre-training of deep bidirectional transformers for language understanding. 2019.

Критерии обучения трансформеров

- **Машинный перевод:** максимизация правдоподобия слов перевода $\tilde{w}_t$ по выборке пар предложений «S, перевод $\tilde{S}$»:

$$\sum_{(S,\tilde{S})} \sum_{\tilde{w}_t \in \tilde{S}} \ln p(\tilde{w}_t | t, S, W) \to \max_W$$

- **BERT MLM (masked language modeling):** предсказание пропущенных слов по локальному контексту
- **BERT NSP (next sentence prediction):** предсказание, следуют ли два предложения друг за другом
- **Fine-tuning:** дообучение трансформера $Z(S, W)$ на задаче с моделью $f(Z(S, W), W_f)$, выборкой $\{S\}$ и $\mathscr{L}(S, f) \to \max$
- **Multi-task learning:** дообучение на наборе задач $\{t\}$ с моделями $f_t(Z(S, W), W_t)$, выборками $\{S\}_t$, по сумме критериев $\sum_t \lambda_t \sum_S \mathscr{L}_t(S, f_t) \to \max$

ChatGPT и GPT-4: проблески общего искусственного интеллекта

Sparks of Artificial General Intelligence: Early experiments with GPT-4

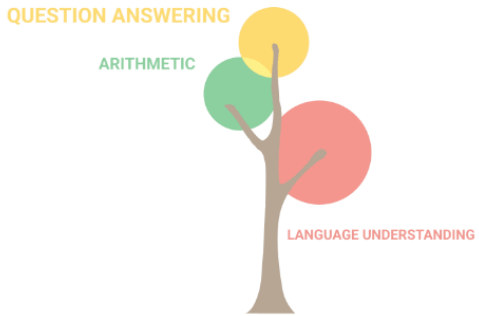
Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

Microsoft Research *(27 March 2023)*

Новые способности модели, не закладывавшиеся при обучении:

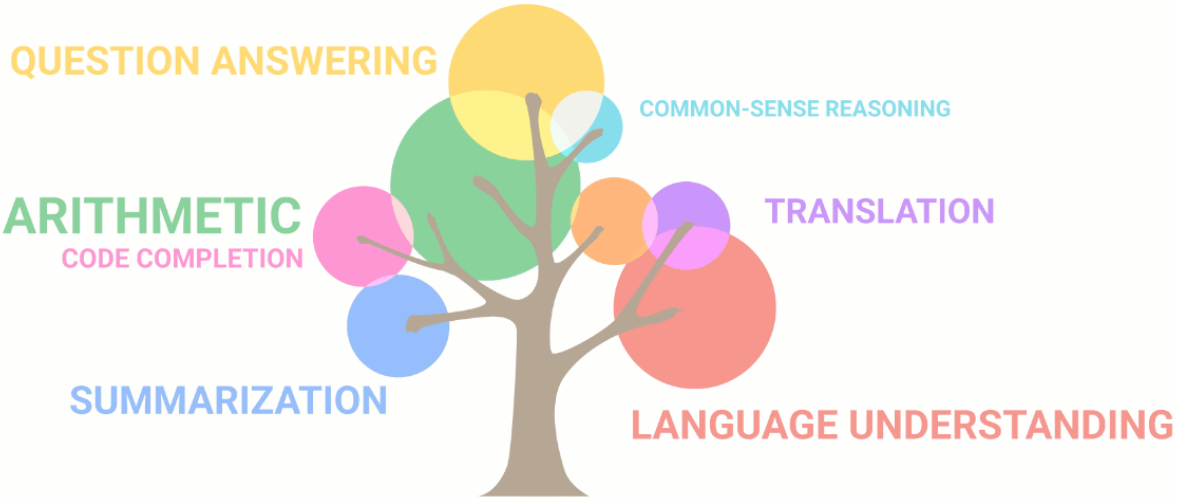
- объяснять свои ответы, перефразировать
- реферировать, генерировать планы, сценарии, шаблоны
- переводить на другие языки, строить аналогии, менять тональность, стиль, глубину изложения
- генерировать программный код на различных языках
- решать некоторые математические задачи
- искать и исправлять собственные ошибки по подсказке

Появление у модели качественно новых способностей



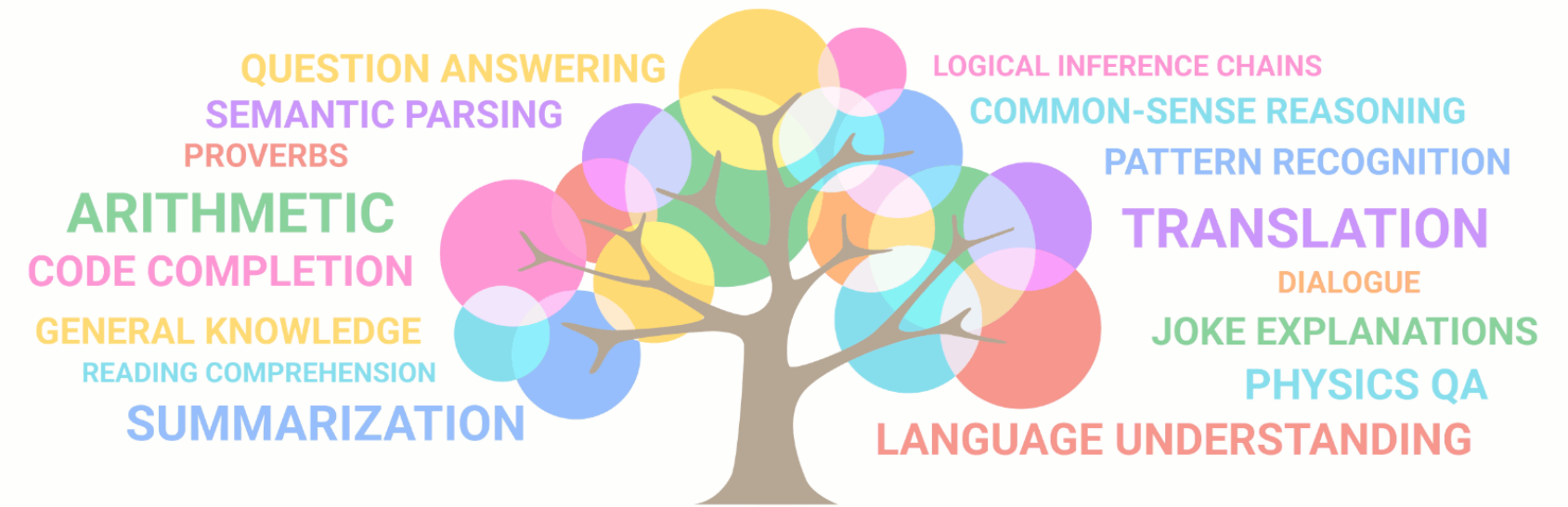
- GPT-2: 14/Feb/2019, контекст 768 слов (1,5 страницы)
- 1,5 млрд. параметров, корпус 10 млрд. токенов (40Gb)
- способность написать эссе, которое конкурсное жюри не смогло отличить от написанного человеком

Появление у модели качественно новых способностей



- GPT-3: 11/Jun/2020, контекст 1536 слов (3 страницы)
- 175 млрд. параметров, корпус 500 млрд. токенов
- способность делать перевод на другие языки,
- решать логические и математические задачи,
- генерировать программный код по описанию

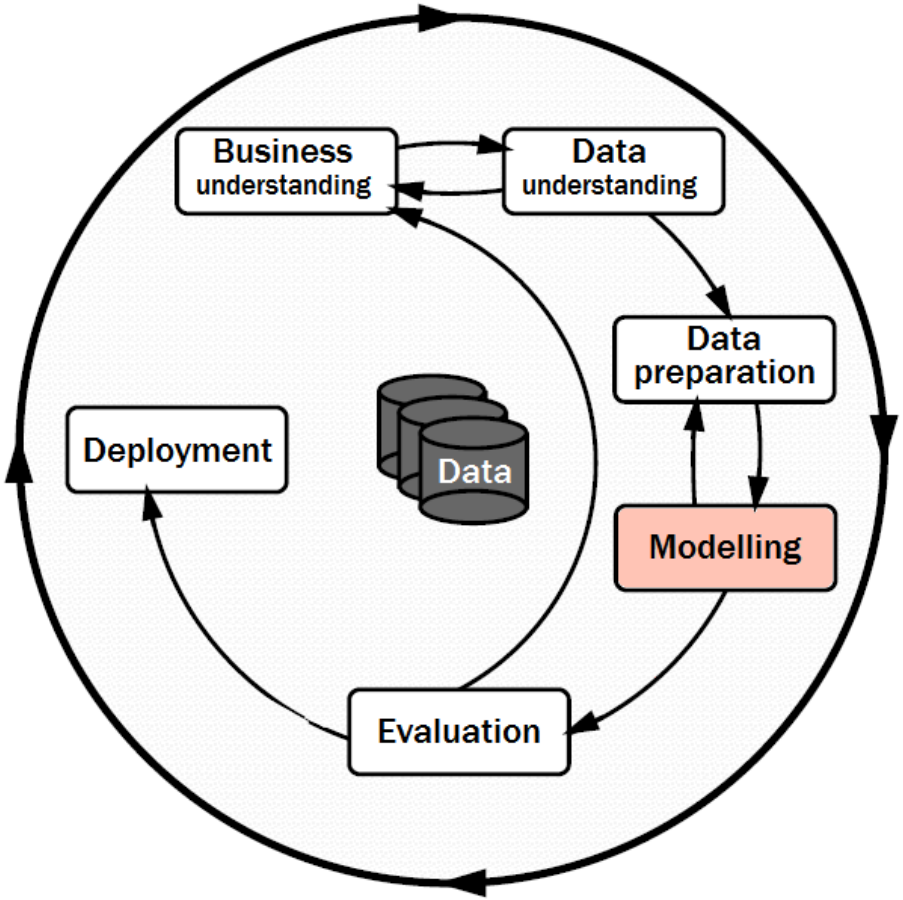
Появление у модели качественно новых способностей



- GPT-4: 14/Mar/2023, контекст 24 000 слов (48 страниц)
- >1 трл. параметров, корпус >1Tb
- способность описывать и анализировать изображения,
- реагировать на подсказки вроде «Let's think step by step»,
- решать качественные физические задачи по картинке

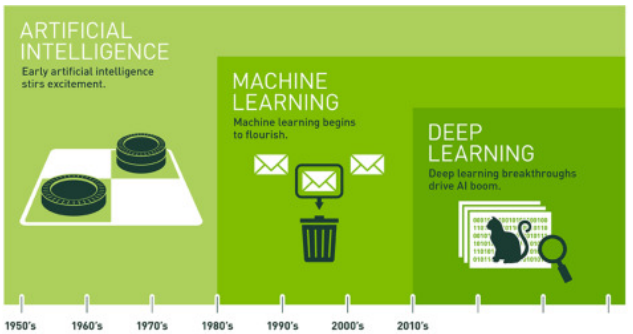
Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)



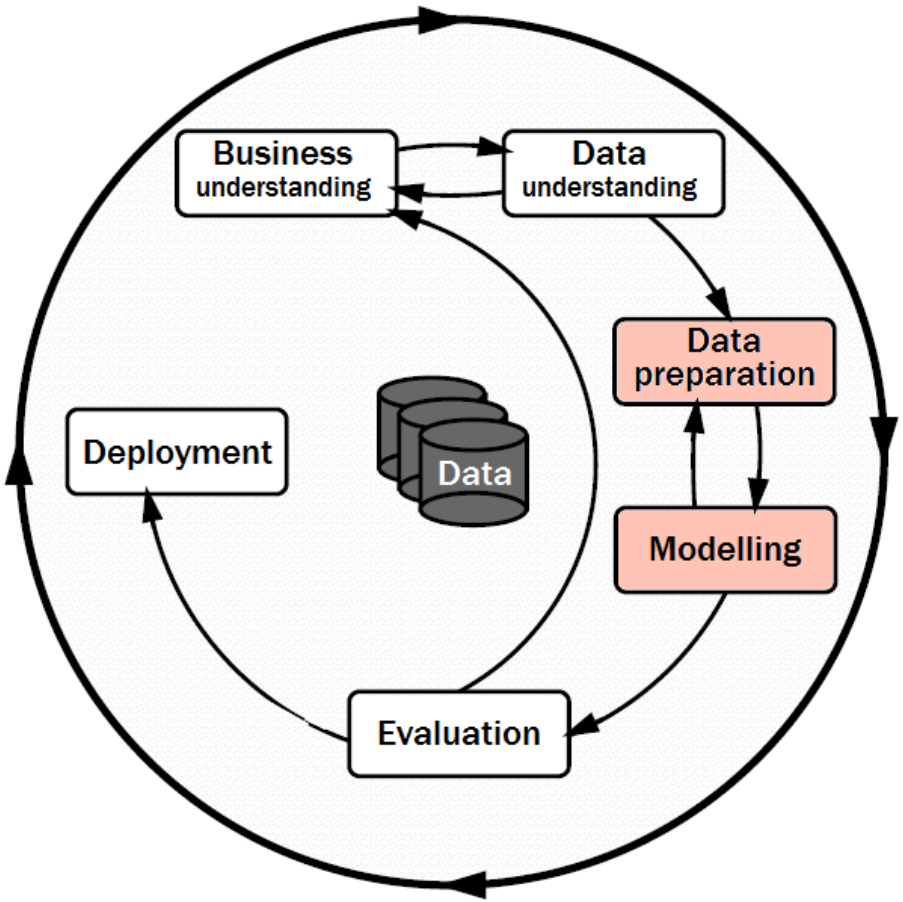
Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным



Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)

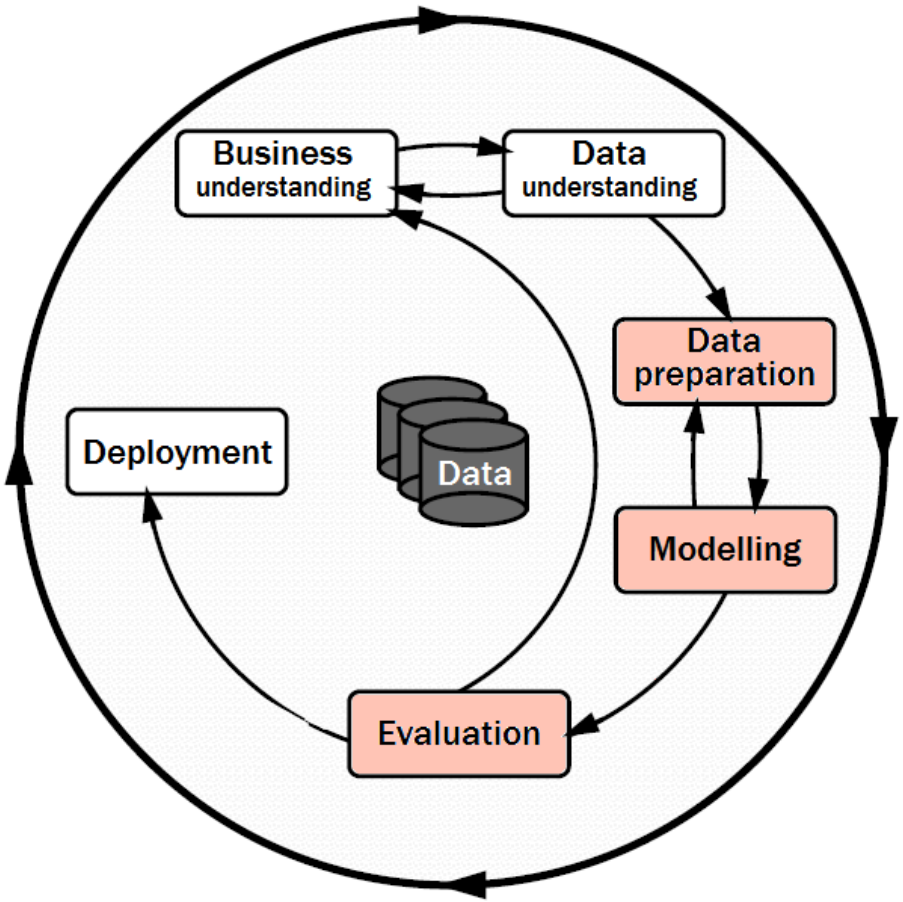


Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным
- *Deep Learning:* модели с обучаемой векторизацией данных

Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)

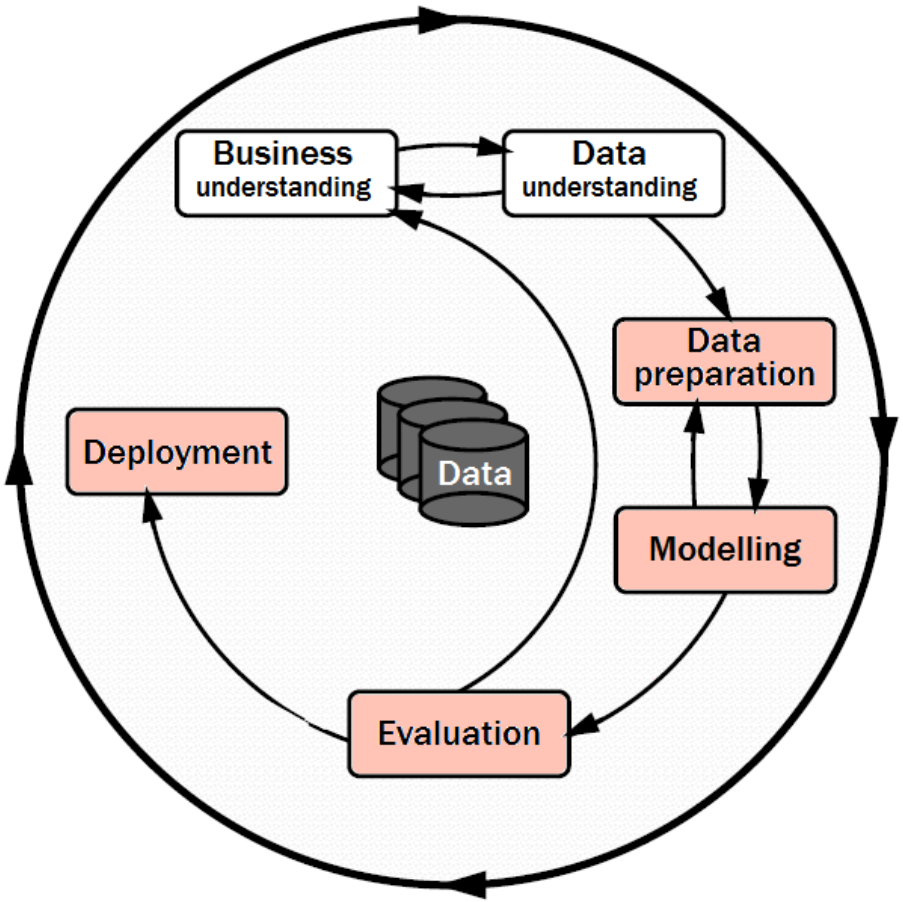


Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным
- *Deep Learning:* модели с обучаемой векторизацией данных
- *AutoML:* автоматический выбор моделей и архитектур

Понимание эволюции ИИ как автоматизации шагов CRISP-DM

CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)

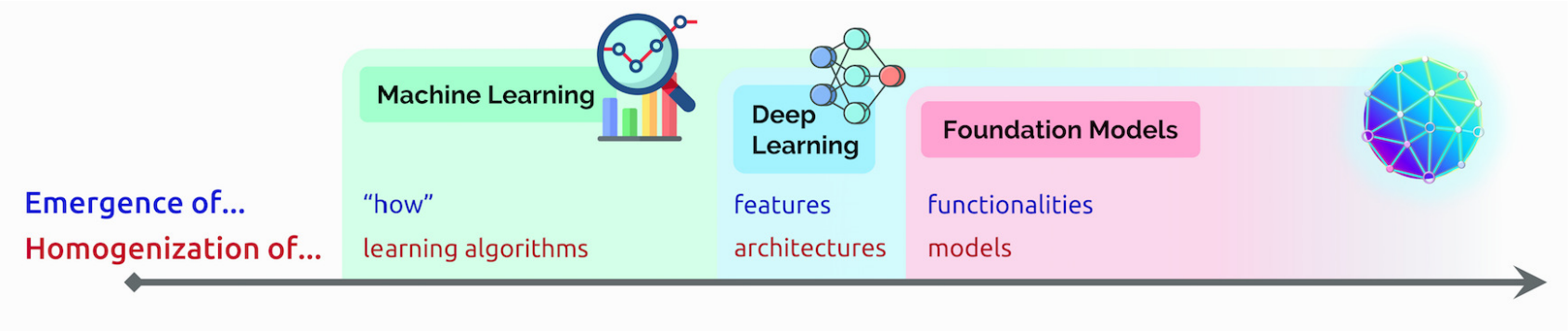


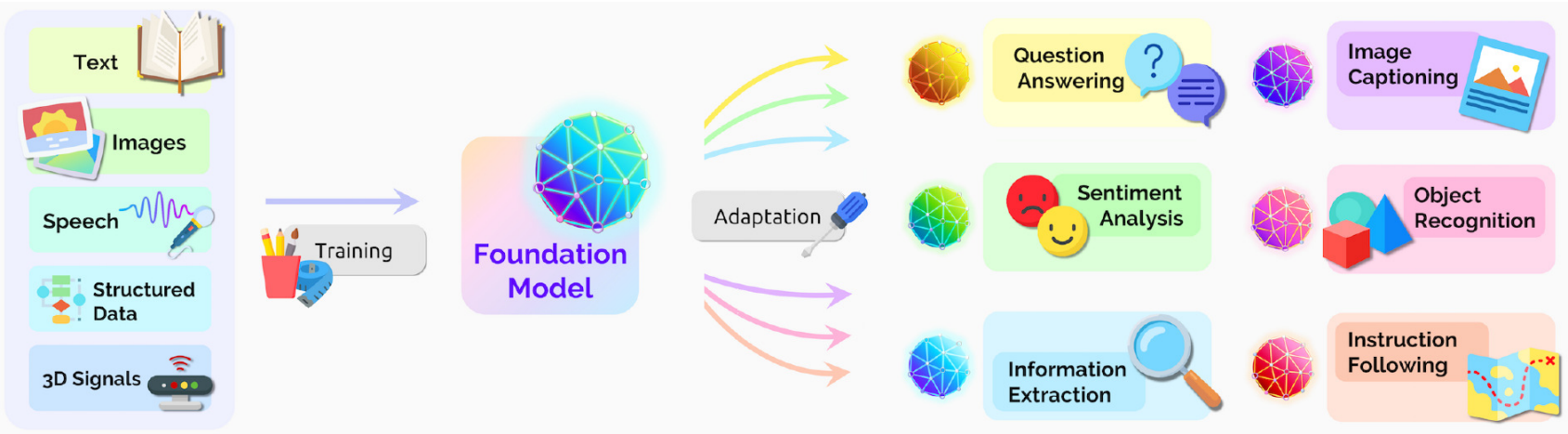
Эволюция ИИ:

- *Expert Systems:* жёсткие модели, основанные на правилах
- *Machine Learning:* параметрические модели, обучаемые по данным
- *Deep Learning:* модели с обучаемой векторизацией данных
- *AutoML:* автоматический выбор моделей и архитектур
- *Lifelong Learning:* бесшовная интеграция обучения и выбора моделей в бизнес-процесс

Гомогенизация векторных моделей (Foundation Models)

Обучаемая векторизация данных — глобальный тренд AI/ML





R.Bommasani et al. (Center for Research on Foundation Models, Stanford University) On the opportunities and risks of foundation models // CoRR, 20 August 2021.