Обзор оптимизационных задач машинного обучения: от персептрона до векторных представлений сложно структурированных данных

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

Институт искусственного интеллекта МГУ

Научный симпозиум
Искусственный интеллект и его применения
28–29 ноября 2022

Содержание

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 восстановления регрессии

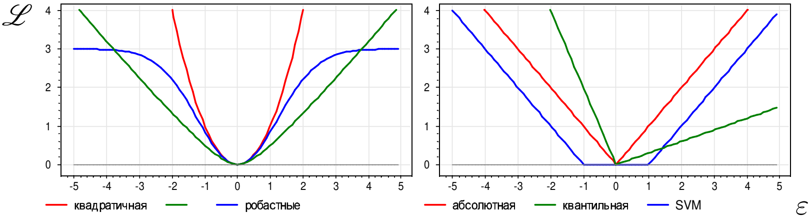
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \mathbb{R}$

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

Критерий: *минимизация эмпирического риска*

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

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



Оптимизационная задача обучения классификации

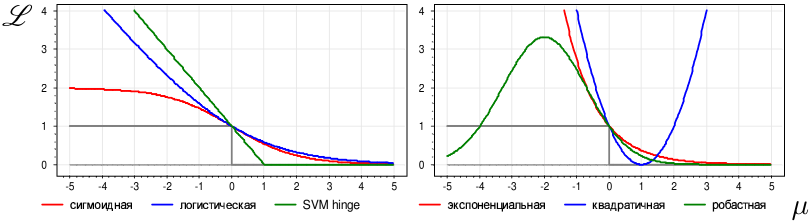
Дано: обучающая выборка $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \{-1, +1\}$

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

Критерий: *аппроксимация эмпирического риска*

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

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: $a(x, w) = \arg\max\limits_{y \in Y} g(x_i, w_y)$

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

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

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

Критерий: максимум правдоподобия (log-loss):

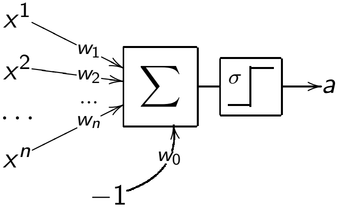
$$-\sum_{i=1}^{\ell} \ln P(y_i|x_i, w) \;\to\; \min_w$$

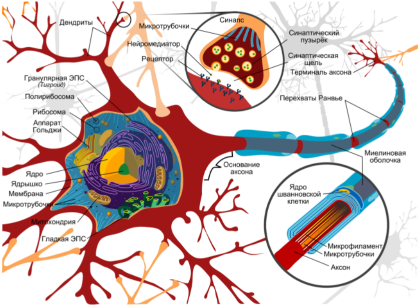
Линейная модель нейрона [МакКаллок, Питтс, 1943]

$f_j(x)$, $j = 1, \ldots, n$ — числовые признаки объекта x;

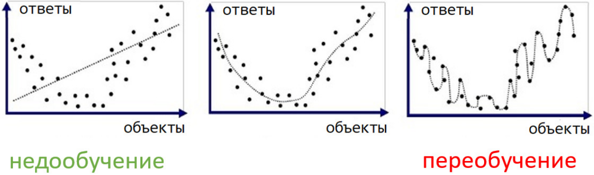
$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0 \right),$$

w_j — веса признаков, $\sigma(z)$ — функция активации





Проблемы недообучения и переобучения



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

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

Жак Саломон Адамар (1865–1963)

Задачи восстановления зависимостей по эмпирическим данным — всегда некорректно поставленные.

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:

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

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

Регуляризаторы для линейных моделей:

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

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

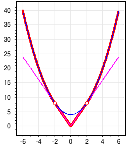
L_0-регуляризация (AIC, BIC): $\text{штраф}(w) = \sum_{j=1}^{n} [w_j \neq 0]$

Негладкие регуляризаторы для отбора признаков

С параметром селективности μ и группировкой признаков:

$$\sum_{i=1}^{\ell} \mathscr{L}\bigl(\langle x_i, w\rangle, y_i\bigr) + \tau \sum_{j=1}^{n} R_\mu(w_j) \;\to\; \min_{w}.$$

Elastic Net: $R_\mu(w) = \mu|w| + w^2$

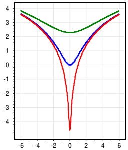


Support Features Machine (SFM):

$$R_\mu(w) = \begin{cases} 2\mu|w|, & |w| \leqslant \mu; \\ \mu^2 + w^2, & |w| \geqslant \mu; \end{cases}$$

Relevance Features Machine (RFM):

$$R_\mu(w) = \ln\bigl(\mu w^2 + 1\bigr)$$



Tatarchuk A., Urlov E., Mottl V., Windridge D. A support kernel machine for supervised selective combining of diverse pattern-recognition modalities. 2010.

Задачи обучения ранжированию (Learning to Rank)

Дано: обучающая выборка объектов $\{x_i\}_{i=1}^{\ell}$
$i \prec j$ — отношение частичного порядка на парах (x_i, x_j)

Найти: модель ранжирования $a\colon X \to \mathbb{R}$ такую, что

$$i \prec j \quad\Rightarrow\quad a(x_i, w) < a(x_j, w)$$

Критерий: число неверно упорядоченных пар (x_i, x_j)
или аппроксимированный попарный эмпирический риск:

$$\sum_{i \prec j} \bigl[a(x_j, w) < a(x_i, w)\bigr] \leqslant \sum_{i \prec j} \mathscr{L}\bigl(\underbrace{a(x_j, w) - a(x_i, w)}_{\mu_{ij}(w)}\bigr) \to \min_w$$

где $\mathscr{L}(\mu)$ — убывающая функция *попарного отступа* $\mu_{ij}(w)$

Задача восстановления плотности распределения

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

Найти: вектор параметров θ в модели $p(x|\theta)$

Критерий: максимум правдоподобия

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta) \;\to\; \max_{\theta}$$

или максимум апостериорной вероятности

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta) + \ln p(\theta|\gamma) \;\to\; \max_{\theta}$$

где γ — вектор гиперпараметров априорного распределения
$R(\theta, \gamma) = \ln p(\theta|\gamma)$ — вероятностный регуляризатор

Задача восстановления смеси плотностей распределения

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

Найти: параметры w_k, θ_k в модели $p(x|\theta, w) = \sum\limits_{k=1}^{K} w_k p(x|\theta_k)$

Критерий: максимум правдоподобия

$$\sum_{i=1}^{\ell} \ln \sum_{k=1}^{K} w_k p(x_i|\theta_k) \to \max_{\theta, w}$$

или максимум апостериорной вероятности

$$\sum_{i=1}^{\ell} \ln \sum_{k=1}^{K} w_k p(x_i|\theta_k) + \ln p(\theta, w|\gamma) \to \max_{\theta, w}$$

где γ — вектор гиперпараметров априорного распределения
$R(\theta, w, \gamma) = \ln p(\theta, w|\gamma)$ — вероятностный регуляризатор

Задача кластеризации (clustering)

Дано: обучающая выборка $\{x_i \in \mathbb{R}^n \colon i = 1, \dots, \ell\}$

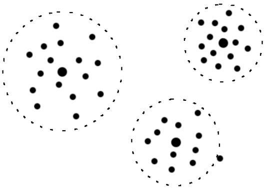
Найти:

— центры кластеров $\mu_k \in \mathbb{R}^n$, $k = 1, \dots, K$

— какому кластеру принадлежит каждый объект $a_i \in \{1, \dots, K\}$

Критерий: минимум суммы внутрикластерных расстояний

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 \to \min_{\{a_i\}, \{\mu_k\}}$$



Метрика, как правило, евклидова (но может быть и другая):

$$\|x_i - \mu_k\|^2 = \sum_{d=1}^{n} \bigl(x_{id} - \mu_{kd}\bigr)^2$$

Задача частичного обучения (semi-supervised learning, SSL)

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

Найти: классификации $(a_i)_{i=k+1}^{\ell}$ неразмеченных объектов

Критерий и кластеризации, и классификации:

- без модели классификации (transductive learning):

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^{k} [a_i \neq y_i] \;\to\; \min_{\{a_i\},\,\{\mu_j\}}$$

- при построении модели классификации, $a_i = a(x_i, w)$:

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^{k} \mathscr{L}(a(x_i, w), y_i) \;\to\; \min_{\{a_i\},\,\{\mu_j\},\,w}$$

Задачи низкорангового матричного разложения

Дано: матрица $X = \|x_{ij}\|_{\ell\times n}$, $(i,j) \in \Omega \subseteq \{1..\ell\} \times \{1..n\}$

Найти: матрицы $Z = \|z_{it}\|_{\ell\times m}$ и $B = \|b_{tj}\|_{m\times n}$, $m \ll \ell, n$

- z_i — сжатые векторные представления объектов x_i
- $\hat{x}_{ij}$ — пропущенные значения в матрице, $(i,j) \notin \Omega$

Критерий: точность восстановления X произведением ZB:

$$\|X - ZB\|_\Omega = \sum_{(i,j)\in\Omega} \mathscr{L}\Bigl(x_{ij} - \sum_t z_{it} b_{tj}\Bigr) \to \min_{Z,B}$$

Отличия от классического SVD:

- разреженные данные: $|\Omega| \ll \ell n$
- неквадратичная функция потерь $\mathscr{L}$
- неотрицательное матричное разложение: $z_{it} \geqslant 0$, $b_{tj} \geqslant 0$
- ортогональность z_i не нужна или не интерпретируема

Задача построения автокодировщика (обучение без учителя)

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

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

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

Пример 1. Линейный автокодировщик: $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$

$$f(x, A) = \underset{m \times n}{A}\, x, \qquad g(z, B) = \underset{n \times m}{B}\, z$$

Пример 2. Двухслойная сеть с функциями активации σ_f, σ_g:

$$f(x, A) = \sigma_f(Ax + a), \qquad g(z, B) = \sigma_g(Bz + b)$$

Автокодировщики для векторизации и обучения с учителем

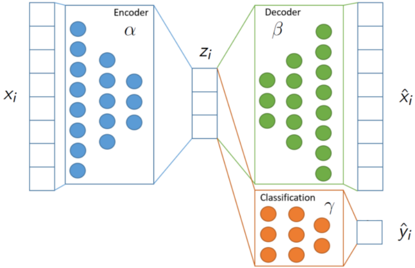
Данные: размеченные $(x_i, y_i)_{i=1}^{k}$, неразмеченные $(x_i)_{i=k+1}^{\ell}$

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.):

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

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

Графовые (матричные) разложения (graph factorization)

Дано: $(i,j) \in E$ — выборка рёбер графа $\langle V, E\rangle$,
x_{ij} — сходство (similarity) между вершинами ребра (i,j)
Например, $x_{ij} = [(i,j) \in E]$ — матрица смежности вершин

Найти: векторные представления вершин $z_i \in \mathbb{R}^m$, так, чтобы близкие (по графу) вершины имели близкие векторы

Критерий:

- для неориентированного графа (X симметрична):

$$\left\|X - ZZ^{\mathsf{T}}\right\|_E = \sum_{(i,j)\in E} \bigl(x_{ij} - \langle z_i, z_j\rangle\bigr)^2 \to \min_Z, \quad Z \in \mathbb{R}^{V\times m}$$

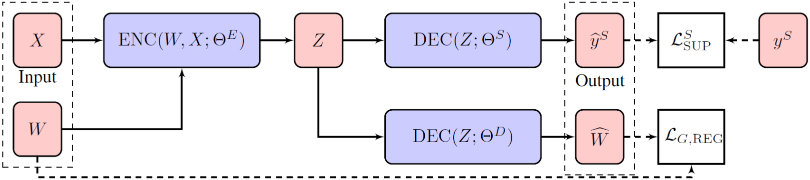
- для ориентированного графа (X несимметрична):

$$\left\|X - ZB^{\mathsf{T}}\right\|_E = \sum_{(i,j)\in E} \bigl(x_{ij} - \langle z_i, b_j\rangle\bigr)^2 \to \min_{Z,B}, \quad Z, B \in \mathbb{R}^{V\times m}$$

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

GraphEDM: обобщённый автокодировщик на графах

Graph Encoder Decoder Model — обобщает более 30 моделей:

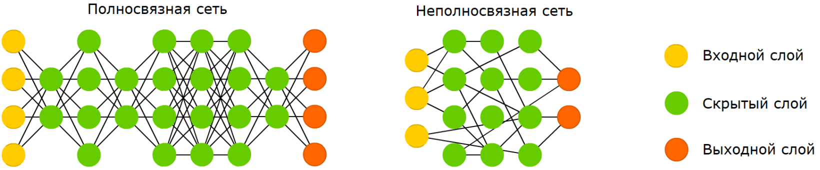


$W \in \mathbb{R}^{V \times V}$ — входные данные о рёбрах
$X \in \mathbb{R}^{V \times n}$ — входные данные о вершинах, признаковые описания
$Z \in \mathbb{R}^{V \times m}$ — векторные представления вершин графа
$\mathrm{DEC}(Z; \Theta^D)$ — декодер, реконструирующий данные о рёбрах
$\mathrm{DEC}(Z; \Theta^S)$ — декодер, решающий supervised-задачу
y^S — (semi-)supervised данные о вершинах или рёбрах
$\mathcal{L}$ — функции потерь

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

Глубокие нейронные сети (Deep Neural Network, DNN)

1965: первые глубокие нейронные сети
2012: свёрточная сеть для классификации изображений AlexNet



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

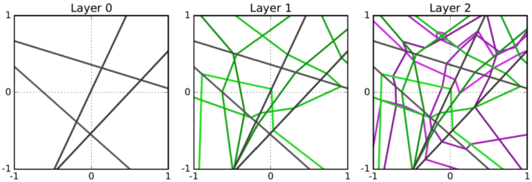
Ива́хненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965.
Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012.

Глубина важнее ширины

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

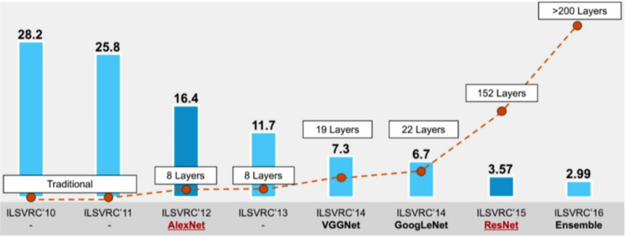
Пример. Участки линейности, $n = 2$, $L = 3$, $H = 4$:



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

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

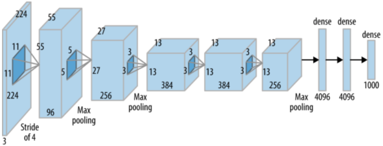
Глубокие свёрточные сети для классификации изображений



Старт в 2009. Человеческий уровень ошибок 5% пройден в 2015

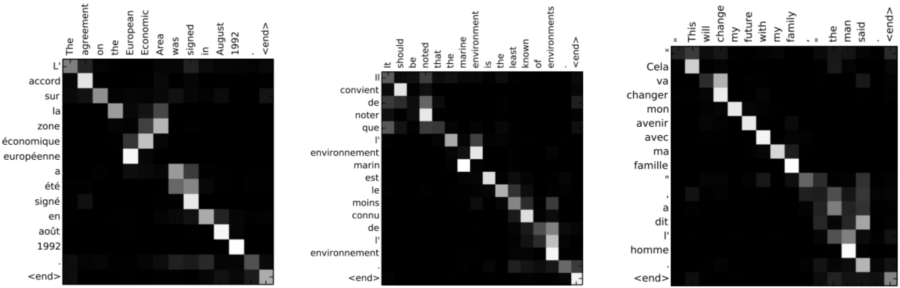
Свёрточная сеть **AlexNet**

Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks. 2012.



Li Fei-Fei et al. ImageNet: A large-scale hierarchical image database. 2009.
Li Fei-Fei et al. Construction and analysis of a large scale image ontology. 2009.

Модели внимания для машинного перевода

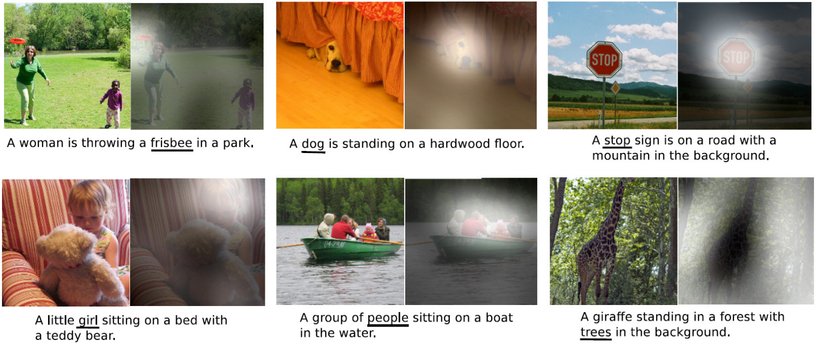


Вход: $\{x_i\}$ — последовательность слов входного языка
Выход: $\{y_t\}$ — последовательность слов выходного языка

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



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:

- $S = (w_1, \dots, w_n)$ — слова предложения на входном языке
  $\downarrow$ обучаемая или пред-обученная векторизация слов
- $X = (x_1, \dots, x_n)$ — векторы слов входного предложения
  $\downarrow$ трансформер-кодировщик
- $Z = (z_1, \dots, z_n)$ — контекстные векторы слов
  $\downarrow$ трансформер-декодировщик, похож на кодировщика
- $Y = (y_1, \dots, y_m)$ — векторы слов выходного предложения
  $\downarrow$ генерация слов из построенной языковой модели
- $\tilde{S} = (\tilde{w}_1, \dots, \tilde{w}_m)$ — 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
$K = (k_1, \dots, k_n)$ — *векторы-ключи*, сравниваемые с запросом
$V = (v_i, \dots, v_n)$ — *векторы-значения*, образующие контекст

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

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

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

Модель внутреннего внимания (самовнимания, self-attention):

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

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

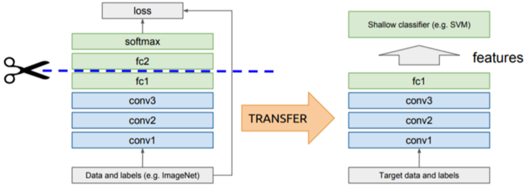
Предобучение (pre-training), перенос обучения (transfer learning)

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

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

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

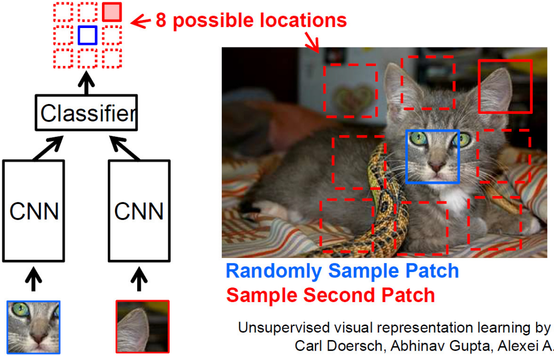
$$\sum_{i=1}^{m} \mathscr{L}'_i\bigl(g'(f(x'_i, \alpha), \beta')\bigr) \to \min_{\beta'}$$



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, $t \in T$:

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

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

Дистилляция моделей или суррогатное моделирование

Обучение сложной модели $a(x, w)$ «долго, дорого»:

$$\sum_{i=1}^{\ell} \mathscr{L}\bigl(a(x_i, w), y_i\bigr) \to \min_{w}$$

Обучение простой модели $b(x, w')$, возможно, на других данных:

$$\sum_{i=1}^{k} \mathscr{L}\bigl(b(x'_i, w'), a(x'_i, w)\bigr) \to \min_{w'}$$

Примеры задач:

- замена сложной модели (климат, аэродинамика и др.), которая вычисляется на суперкомпьютере месяцами, «лёгкой» аппроксимирующей суррогатной моделью
- замена сложной нейросети, которая обучается неделями на больших данных, «лёгкой» аппроксимирующей нейросетью с минимизацией числа нейронов и связей

Задача обучения с привилегированной информацией

x_i^* — информация об объекте x_i, доступная только на обучении

Раздельное обучение модели-ученика и модели-учителя:

$$\sum_{i=1}^{\ell} \mathscr{L}\bigl(a(x_i, w), y_i\bigr) \to \min_w \qquad \sum_{i=1}^{\ell} \mathscr{L}\bigl(a(x_i^*, w^*), y_i\bigr) \to \min_w$$

Модель-ученик обучается по модели-учителю:

$$\sum_{i=1}^{\ell} \mathscr{L}\bigl(a(x_i, w), y_i\bigr) + \mu \mathscr{L}\bigl(a(x_i, w), a(x_i^*, w^*)\bigr) \;\to\; \min_w$$

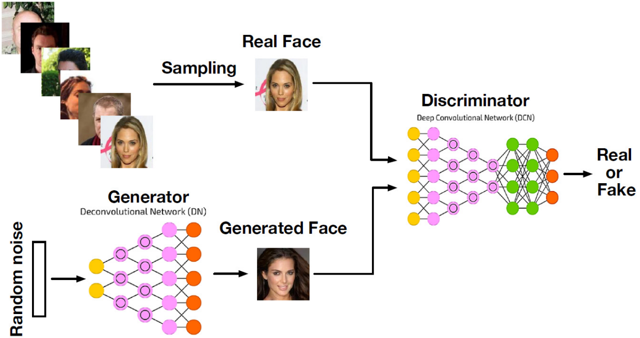
Совместное обучение модели-ученика и модели-учителя:

$$\sum_{i=1}^{\ell} \mathscr{L}\bigl(a(x_i, w), y_i\bigr) + \lambda \mathscr{L}\bigl(a(x_i^*, w^*), y_i\bigr) + \\ + \mu \mathscr{L}\bigl(a(x_i, w), a(x_i^*, w^*)\bigr) \;\to\; \min_{w, w*}$$

D.Lopez-Paz, L.Bottou, B.Scholkopf, V.Vapnik. Unifying distillation and privileged information. 2016.

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 $D(x, \beta) = p(1|x, \beta)$

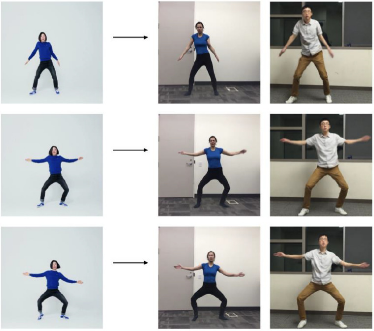
Критерий: log правдоподобия дискриминативной модели; *генератор* $G(z)$ учится порождать объекты x из шума z, *дискриминатор* $D(x)$ учится отличать их от реальных объектов, в антагонистической игре генератора против дискриминатора:

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

Ian Goodfellow et al. Generative Adversarial Nets. 2014

Примеры GAN для синтеза изображений и видео

(d) input image (e) output 3d face (f) textured 3d face



Source Subject Target Subject 1 Target Subject 2

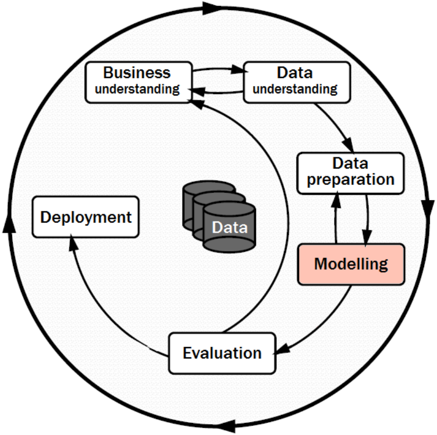
Chuan Li, Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. 2016.

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.

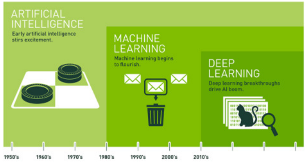
Понимание эволюции ИИ как автоматизации шагов CRISP-DM

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



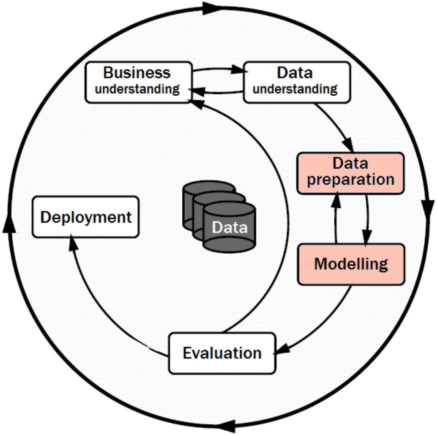
Эволюция ИИ:

- *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)

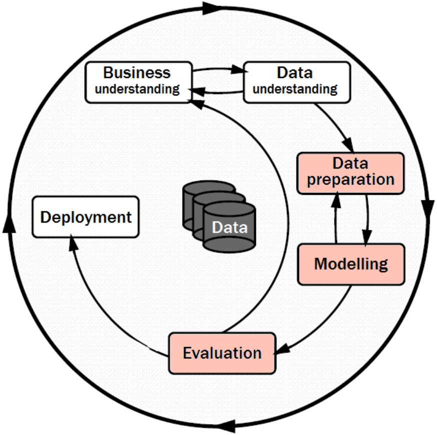


Эволюция ИИ:

- *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)

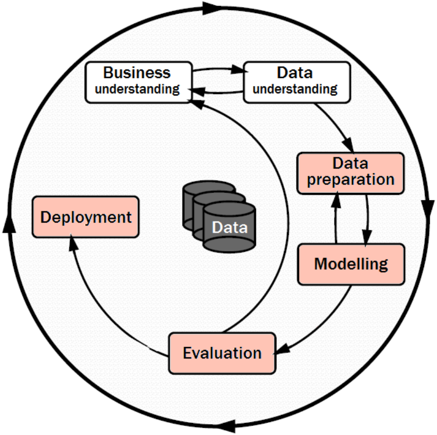


Эволюция ИИ:

- *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)

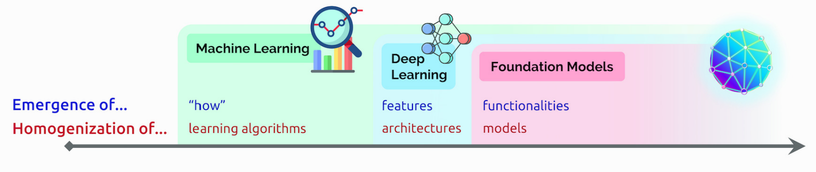


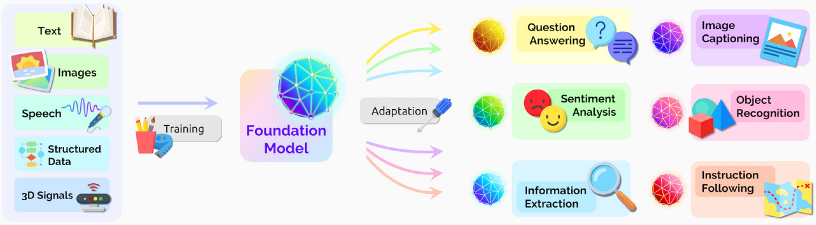
Эволюция ИИ:

- *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.

Резюме

- **Глубокие нейронные сети** — это не интеллект, а векторизация сложно структурированных данных, обучаемая совместно с предсказательной моделью
- **Проблемы переобучения** решаются приёмами регуляризации: L2, DropOut, SkipConnect, multitask и др.
- **Проблемы оптимизации** в пространствах высокой размерности решаются ускоренными методами стохастического градиента (методами первого порядка)
- **Проблемы скорости вычислений** преодолеваются распараллеливанием и приёмами быстрого (символьного) дифференцирования суперпозиций функций
- **Тенденции:** инженерия вытесняет математику, трансформеры вытесняют всё, гомогенизация моделей
- **Открытые проблемы:** этичность, интерпретируемость, доверенность, неатакуемость, распределённость. Возможно ли упрощение архитектур без потери точности?