• Введение в машинное обучение •
Обучаемая векторизация данных

Воронцов Константин Вячеславович
k.v.vorontsov@phystech.edu
http://www.MachineLearning.ru/wiki?title=User:Vokov

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Введение в машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ.ФПМИ.ИС.ИАД • 12 марта 2026

Содержание

Метод главных компонент (Principal Component Analysis, PCA)

Возможно ли описать объекты меньшим числом признаков?

Дано: выборка объектов $\{x_i\}_{i=1}^{\ell}$,
$f_1(x), \ldots, f_n(x)$ — числовые признаки объектов

Найти:
$g_1(x), \ldots, g_m(x)$ — новые числовые признаки, $m \leqslant n$, и линейную реконструкцию старых признаков $f_j(x)$ по новым:

$$\hat{f}_j(x) = \sum_{t=1}^{m} g_t(x) u_{jt}, \quad j = 1, \ldots, n, \quad \forall x \in X,$$

Критерий: точность реконструкции f_j на обучающей выборке:

$$Q = \sum_{i=1}^{\ell} \sum_{j=1}^{n} \left(\hat{f}_j(x_i) - f_j(x_i)\right)^2 \to \min_{\{g_t(x_i)\},\{u_{jt}\}}$$

Это обучение без учителя и линейный автокодировщик.

Задача низкорангового матричного разложения

Матрицы «объекты–признаки», старая и новая:

$$\underset{\ell\times n}{F} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}; \quad \underset{\ell\times m}{G} = \begin{pmatrix} g_1(x_1) & \dots & g_m(x_1) \\ \dots & \dots & \dots \\ g_1(x_\ell) & \dots & g_m(x_\ell) \end{pmatrix}.$$

Матрица линейного преобразования новых признаков в старые:

$$\underset{n\times m}{U} = \begin{pmatrix} u_{11} & \dots & u_{1m} \\ \dots & \dots & \dots \\ u_{n1} & \dots & u_{nm} \end{pmatrix}; \qquad \hat{F} = GU^{\mathsf{T}} \overset{\text{хотим}}{\approx} F.$$

Критерий в матричном виде — ищем одновременно G и U:

$$Q(G,U) = \sum_{i=1}^{\ell}\sum_{j=1}^{n}\big(\hat{f}_j(x_i) - f_j(x_i)\big)^2 = \big\|GU^{\mathsf{T}} - F\big\|^2 \to \min_{G,U},$$

Основная теорема метода главных компонент

Теорема

Если $m \leqslant \mathrm{rk}F$, *то минимум* $\left\|GU^{\mathsf{T}} - F\right\|^2$ *достигается, когда столбцы* U *— это с.в. матрицы* $F^{\mathsf{T}}F$, *соответствующие* m *максимальным с.з.* $\lambda_1, \ldots, \lambda_m$, *и* $G = FU$, *при этом:*

1. *матрица* U *ортонормирована:* $U^{\mathsf{T}}U = I_m$;
2. *матрица* G *ортогональна:* $G^{\mathsf{T}}G = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$;
3. $U\Lambda = F^{\mathsf{T}}FU$; $G\Lambda = FF^{\mathsf{T}}G$;
4. $Q = \left\|GU^{\mathsf{T}} - F\right\|^2 = \|F\|^2 - \mathrm{tr}\Lambda = \lambda_{m+1} + \cdots + \lambda_n$.

При $m = n$ разложение $F = GU^{\mathsf{T}}$ является точным $(Q = 0)$ и совпадает с сингулярным разложением $F = (G\Lambda^{-\frac{1}{2}}) \cdot \Lambda^{\frac{1}{2}} \cdot U^{\mathsf{T}}$

Вопрос: как доказывать эту теорему, какие есть идеи?

Использовано тождество: $\|F\|^2 = \mathrm{tr}(F^{\mathsf{T}}F) = \lambda_1 + \cdots + \lambda_n$

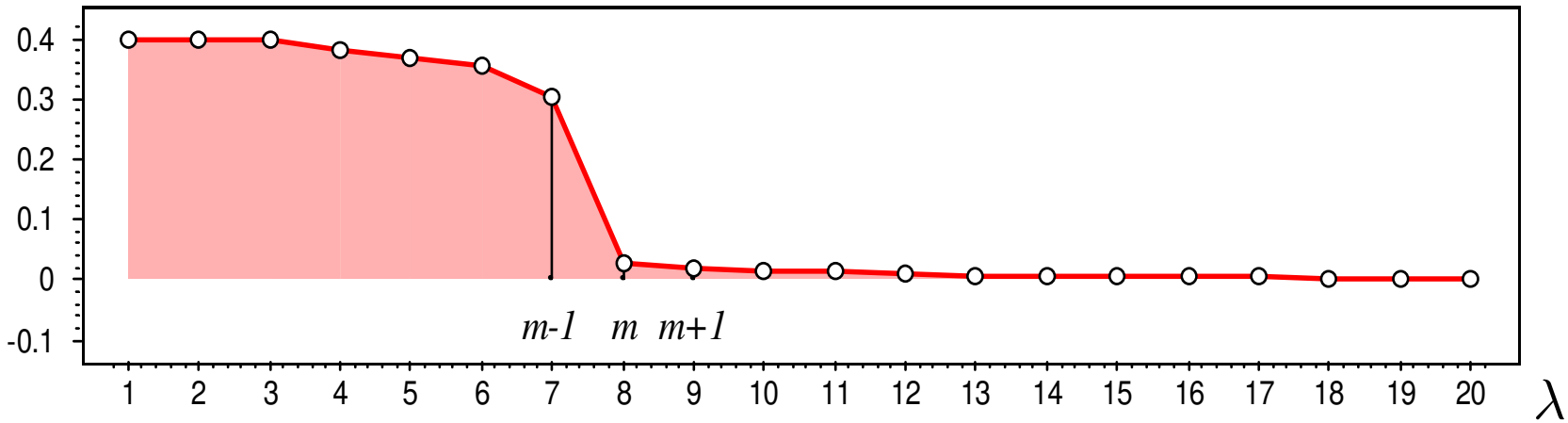
Эффективная размерность выборки

Упорядочим с.з. $F^\mathsf{T}F$ по убыванию: $\lambda_1 \geqslant \ldots \geqslant \lambda_n \geqslant 0$.

Эффективная размерность выборки — наименьшее целое m, при котором относительная погрешность достаточно мала:

$$E_m = \frac{\|GU^\mathsf{T} - F\|^2}{\|F\|^2} = \frac{\lambda_{m+1} + \cdots + \lambda_n}{\lambda_1 + \cdots + \lambda_n} \leqslant \varepsilon.$$

Критерий «крутого склона»: находим m: $E_{m-1} \gg E_m$:



Вопрос: что в таком случае можно сказать о признаках $(f_j)_{j=1}^n$?

Метод главных компонент и линейный автокодировщик

Линейный автокодировщик: $f(x, A) = Ax$, $g(z, B) = Bz$,

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

Метод главных компонент: $f(x, U) = U^{\mathsf{T}}x$, $g(z, U) = Uz$,
в матричных обозначениях $F = (x_1 \ldots x_\ell)^{\mathsf{T}}$, $U^{\mathsf{T}}U = I_m$, $G = FU$,

$$\|GU^{\mathsf{T}} - F\|^2 = \sum_{i=1}^{\ell} \|UU^{\mathsf{T}}x_i - x_i\|^2 \to \min_{U}$$

Автокодировщик обобщает метод главных компонент:

- не обязательно $B = A^{\mathsf{T}}$ (хотя часто именно так и делают)
- произвольные A, B вместо ортогональных
- нелинейные модели $f(x, \alpha)$, $g(z, \beta)$ вместо Ax, Bz
- произвольная функция потерь $\mathscr{L}$ вместо квадратичной
- SG оптимизация вместо сингулярного разложения SVD

Разреженное низкоранговое матричное разложение

Дано: матрица $F = (f_{ij})_{\ell \times n}$, $(i,j) \in \Omega \subseteq \{1,\dots,\ell\} \times \{1,\dots,n\}$

Найти: матрицы $G = (g_{it})_{\ell \times m}$ и $U^{\mathsf{T}} = (u_{tj})_{m \times n}$

Критерий: $\|GU^{\mathsf{T}} - F\|_{\Omega}^2 = \sum_{(i,j)\in\Omega} \left(\sum_{t=1}^{m} g_{it} u_{tj} - f_{ij} \right)^2 \to \min_{G,U}$

Классический SVD становится неприменим, когда

- данные разреженные: $|\Omega| < \ell n$, зачастую $|\Omega| \ll \ell n$
- функция потерь неквадратичная
- матричное разложение неотрицательное: $g_{it} \geqslant 0$, $u_{tj} \geqslant 0$
  или стохастическое: $\sum_t g_{it} = 1$, $\sum_t u_{tj} = 1$, $g_{it} \geqslant 0$, $u_{tj} \geqslant 0$

Ситуации применения:

- снижение размерности вектора признаков, $m \ll n$
- выявление латентной внутренней структуры данных
- восстановление пропущенных значений (missing values)

Модель латентных факторов (LFM, Latent Factor Model)

В рекомендательных системах:

f_{ij} — выбор (покупка, лайк, рейтинг) клиентом i товара j

$g_i = (g_{it})_t$ — латентный вектор интересов (embedding) клиента i

$u_j = (u_{tj})_t$ — латентный вектор интересов (embedding) товара j

Метод стохастического градиента:

выбираем $(i,j) \in \Omega$ в случайном порядке;

градиентный шаг для задачи $\varepsilon_{ij}^2 \to \min\limits_{g_i,u_j}$, где $\varepsilon_{ij} = \sum\limits_t g_{it}u_{tj} - f_{ij}$:

$$g_{it} := g_{it} - \eta\varepsilon_{ij}u_{tj}, \quad t = 1,\ldots,m$$
$$u_{tj} := u_{tj} - \eta\varepsilon_{ij}g_{it}, \quad t = 1,\ldots,m$$

B1: как повлияет регуляризация $\varepsilon_{ij}^2 + \lambda\|g_i\|^2 + \mu\|u_j\|^2 \to \min\limits_{g_i,u_j}$?

B2: как ввести ограничения $g_{it} \geqslant 0$, $u_{tj} \geqslant 0$?

Tacáks G., Pilászy I., Németh B., Tikk D. Scalable collaborative filtering approaches for large recommendation systems // JMLR, 2009, No. 10, Pp. 623–656.

NNMF (Non-Negative Matrix Factorization)

Неотрицательное матричное разложение: метод ALS — *чередующихся наименьших квадратов* (Alternating Least Squares):

$$Q = \Big\|\sum_t g_t u_t^{\mathsf{T}} - F\Big\|^2 = \big\|g_t u_t^{\mathsf{T}} - F_t\big\|^2 \to \min_{\{g_t \geqslant 0,\ u_t \geqslant 0\}}$$

Идея: искать поочерёдно то столбцы g_t, то столбцы u_t при фиксированных остальных, где $F_t = F - \sum_{s \neq t} g_s u_s^{\mathsf{T}}$

$$\frac{\partial Q}{\partial g_t} = 0 \quad \Rightarrow \quad (g_t u_t^{\mathsf{T}} - F_t)u_t = 0 \quad \Rightarrow \quad g_t = \left(\frac{F_t u_t}{u_t^{\mathsf{T}} u_t}\right)_+$$

$$\frac{\partial Q}{\partial u_t} = 0 \quad \Rightarrow \quad g_t^{\mathsf{T}}(g_t u_t^{\mathsf{T}} - F_t) = 0 \quad \Rightarrow \quad u_t = \left(\frac{F_t^{\mathsf{T}} g_t}{g_t^{\mathsf{T}} g_t}\right)_+$$

положительная срезка $(\cdot)_+$ — из условий Каруша–Куна–Таккера

A.Cichocki, R.Zdunek, S.Amari. Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. 2007

PLSA (Probabilistic Latent Semantic Analysis)

В вероятностном тематическом моделировании:

$f_{ij} = \hat{p}(j|i)$ — частота слова $j \in \{1..n\}$ в документе $i \in \{1..\ell\}$

$g_{it} = p(t|i)$ — вероятность темы $t \in \{1..m\}$ в документе i

$u_{tj} = p(j|t)$ — вероятность слова j в теме t

$\sum_t g_{it} u_{tj} = p(j|i)$ — вероятностная тематическая модель языка

Критерий — max правдоподобия (min кросс-энторпии):

$$\sum_{(i,j)\in\Omega} f_{ij} \ln \sum_t g_{it} u_{tj} \to \max_{G,U}; \quad g_{it} \geqslant 0,\ \sum_t g_{it} = 1,\ u_{tj} \geqslant 0,\ \sum_j u_{tj} = 1$$

Алгоритм EM (Expectation–Maximization) — из условий KKT:

$$p_{tij} = \text{norm}_t(g_{it} u_{tj})$$
$$g_{it} = \text{norm}_t\left(\sum_j f_{ij} p_{tij}\right)$$
$$u_{tj} = \text{norm}_j\left(\sum_i f_{ij} p_{tij}\right)$$

операция нормировки вектора:

$$\text{norm}_i(x_i) = \frac{\max(x_i, 0)}{\sum_k \max(x_k, 0)}$$

M.Shashanka et al. Probabilistic latent variable models as nonnegative factorizations. 2008.

Примеры прикладных задач с графовыми данными

Дано: попарные расстояния между объектами $R(x_i, x_j)$
Найти: представления объектов в $\mathbb{R}^d$, сохраняющие расстояния

Дано: молекулярные графы (вершины-атомы, рёбра-связи)
Найти: (предсказать) свойства химических соединений

Дано: граф связей между пользователями соцсети
Найти: сообщества, социальные роли, центры влияния

Дано: граф транзакций между клиентами и товарами
Найти: рекомендации, аномалии, прогнозы спроса

Дано: граф транспортной сети
Найти: прогноз трафика, оптимальные маршруты

Основные типы задач:

- классификация вершин графа
- рекомендации, предсказание связей (Link Prediction)
- классификация графа или компонент связности целиком

Многомерное шкалирование (multidimensional scaling, MDS)

Дано: $(i,j) \in E$ — выборка рёбер графа $\langle V, E \rangle$
R_{ij} — расстояния между вершинами ребра (i,j)
Например, R_{ij} — длина кратчайшего пути по графу (IsoMAP)

Найти: векторные представления вершин $z_i \in \mathbb{R}^d$ так, чтобы близкие вершины (в смысле малого R_{ij}) имели близкие z_i и z_j

Критерий стресса (stress):

$$\sum_{(i,j)\in E} R_{ij}^{\gamma} \big(\rho(z_i, z_j) - R_{ij}\big)^2 \to \min_Z, \quad Z \in \mathbb{R}^{V \times d},$$

где $\rho(z_i, z_j) = \|z_i - z_j\|$ — обычно евклидово расстояние,

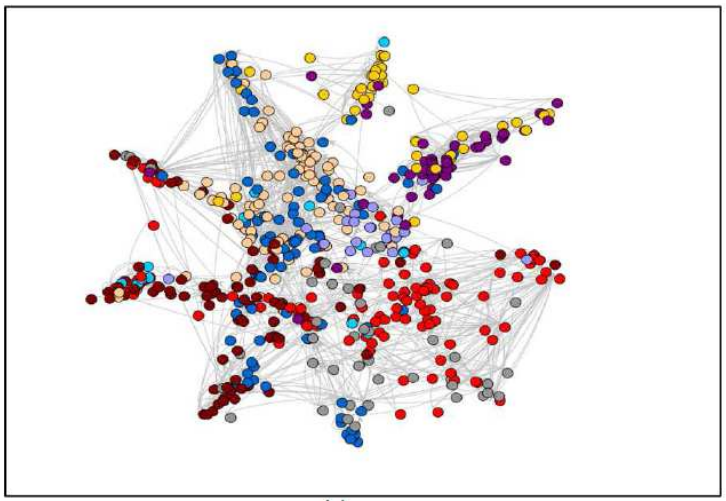
Решение — обычно методом стохастического градиента (SG)

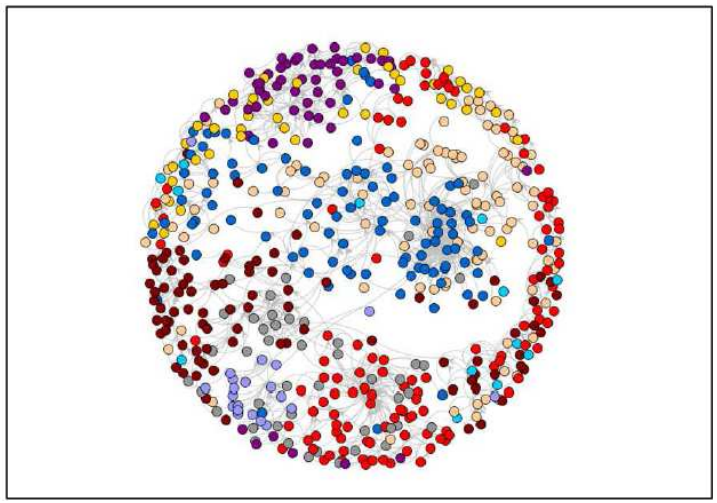
Вопрос: как лучше задавать веса: $\gamma > 0$ или $\gamma < 0$?

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

Многомерное шкалирование для визуализации данных

При $d = 2$ осуществляется проекция выборки на плоскость





- используется для визуализации кластерных структур
- форму облака точек можно настраивать весами и метрикой
- наиболее популярные методы — t-SNE, UMAP (быстрый)
- недостаток всех методов — искажения неизбежны

Laurens van der Maaten, Geoffrey Hinton. Visualizing data using t-SNE. 2008
Leland McInnes, John Healy, James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. 2020

Графовые матричные разложения (graph factorization)

Дано: граф $\langle V, E \rangle$
S_{ij} — близость между вершинами $(i,j) \in E$
Например, $S_{ij} = [(i,j) \in E]$ — матрица смежности вершин

Найти: векторные представления вершин, так, чтобы близкие (по графу) вершины имели близкие векторы.

Критерий для неориентированного графа (S симметрична):

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

Критерий для ориентированного графа (S несимметрична):

$$\left\| S - \Phi\Theta^{\mathsf{T}} \right\|_E = \sum_{(i,j)\in E} \left(\langle \varphi_i, \theta_j \rangle - S_{ij} \right)^2 \to \min_{\Phi,\Theta}, \quad \Phi, \Theta \in \mathbb{R}^{V \times d}$$

Решение — обычно методом стохастического градиента (SG)

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

Векторные представления графов как автокодировщики

Дано: граф $\langle V, E \rangle$

$X = (x_i \in \mathbb{R}^n : i \in V)$ — векторы признаков в вершинах графа

$W = (w_{ij} \in \mathbb{R} : (i,j) \in E)$ — числовые данные о рёбрах графа

$Y = (y_i \in Y : i \in V')$ — целевой отклик на вершинах из $V' \subseteq V$

Найти:

$Z = (z_i \in \mathbb{R}^d : i \in V)$ — векторные представления вершин графа

параметры α кодировщика $z_i(\alpha) = f_i(W, X, \alpha)$

параметры β декодировщика $\hat{w}_{ij} = g(z_i, z_j, \beta)$

параметры γ предсказательной модели $y_i = \hat{y}(z_i, \gamma)$

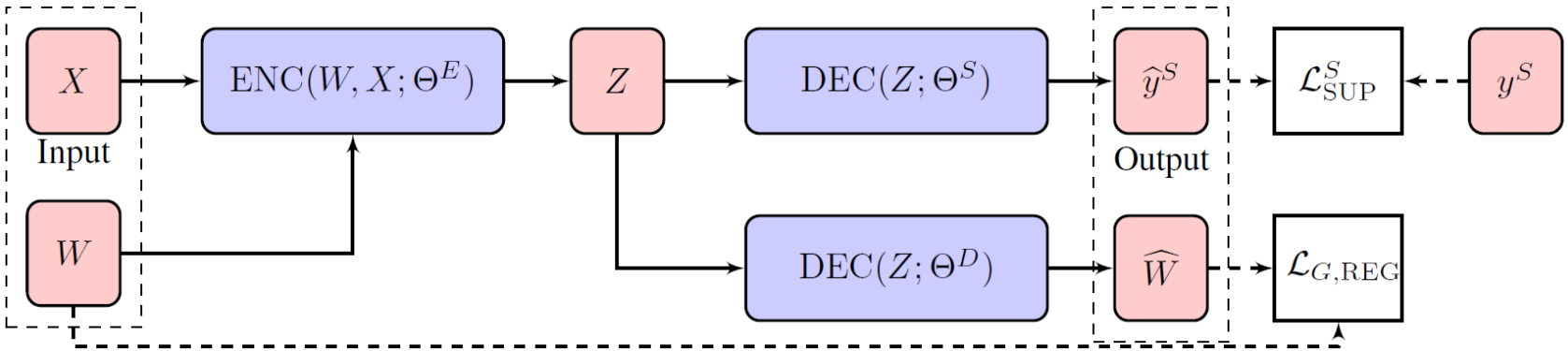
Критерий: потери реконструкции рёбер $\hat{w}_{ij}$ по вершинам (z_i, z_j) и потери предсказательного моделирования y_i по z_i:

$$\sum_{(i,j)\in E} \mathscr{L}\big(g(z_i(\alpha), z_j(\alpha), \beta), w_{ij}\big) + \lambda \sum_{i \in V'} \tilde{\mathscr{L}}\big(\hat{y}(z_i(\alpha), \gamma), y_i\big) \to \min_{\alpha,\beta,\gamma}$$

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 признаковые описания вершин
$Z \in \mathbb{R}^{V \times d}$ — 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 — данные о вершинах в supervised-задаче
$\mathcal{L}$ — функции потерь

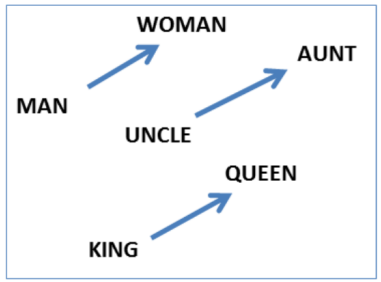
I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

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

Контекстно независимые эмбединги слов в вероятностных моделях языка на основе матричных разложений



- модели дистрибутивной семантики
  word2vec [Mikolov, 2013], FastText [Bojanowski, 2016]
- тематические модели LDA [Blei, 2001], ARTM [2014]

Контекстно зависимые нейросетевые эмбединги

- рекуррентные нейронные сети LSTM [1997]
- модели внимания и трансформеры NMT [2015]
  BERT [2018], GPT-3 [2020], GPT-4 [2023]

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

Дистрибутивная гипотеза и виды семантической близости слов

Смысл слова есть множество всех контекстов его употребления

- Words that occur in the same contexts tend to have similar meanings [Harris, 1954].
- You shall know a word by the company it keeps [Firth, 1957].

Синтагматическая близость слов:
сочетаемость слов в одном контексте.

здание–строитель, кран–вода, функция–точка

Парадигматическая близость слов:
взаимозаменяемость слов в одном контексте.

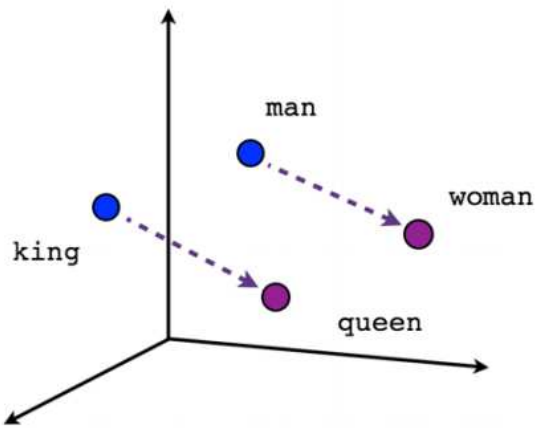
здание–дом, кран–смеситель, функция–отображение

Z.Harris. Distributional structure. 1954.
J.R.Firth. A synopsis of linguistic theory 1930-1955. Oxford, 1957.
P.Turney, P.Pantel. From frequency to meaning: vector space models of semantics. 2010.

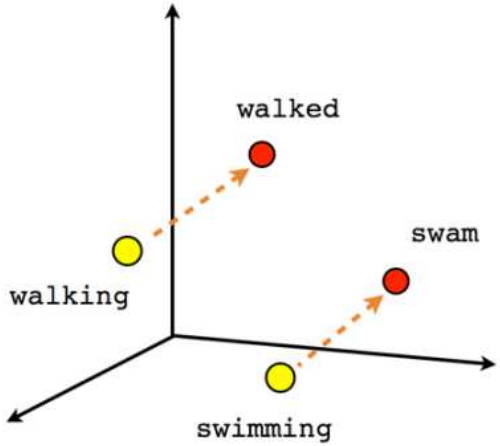
Задача семантического векторного представления слов

Задача: по наблюдаемой синтагматической близости слов построить *векторные представления слов* (word embedding, WE) $x_w \in \mathbb{R}^d$, $w \in W$, отражающие их парадигматическую близость, т.е. близкие по смыслу слова должны иметь близкие векторы.

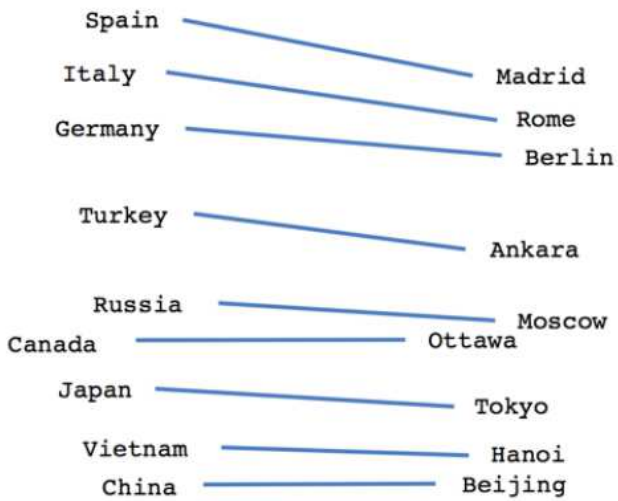
Способ проверки — задача семантической аналогии слов: по трём словам угадать четвёртое.



Male-Female



Verb tense



Country-Capital

Формализация дистрибутивной гипотезы в программе word2vec

Дано: частоты n_{wu} пар слов w, u в контекстном окне $\pm k$ слов

Найти: векторные представления слов x_w и предсказывающих слов-из-контекста y_u в вероятностной языковой модели

$$p(w|u) = \underset{w \in W}{\mathrm{SoftMax}} \langle x_w, y_u \rangle = \underset{w \in W}{\mathrm{norm}} \big(\exp \langle x_w, y_u \rangle \big)$$

Критерий: максимум правдоподобия для предсказания слов

$$\sum_{w,u \in W} n_{wu} \ln p(w|u) \;\to\; \max_{x_w, y_u}$$

или для классификации пар слов на 2 класса близки/далеки:

$$\sum_{w,u \in W} \left(n_{wu} \ln p(w,u) + \sum_{i=1}^{n_{wu}} \ln \big(1 - p(w,u_i)\big) \right) \;\to\; \max_{x_w, y_u}$$

$p(w,u) = \sigma \langle x_w, y_u \rangle$ — вероятность встретить w, u рядом;
u_i сэмплируется из $p(w)^{3/4}$ (skip-gram negative sampling, SGNS)

T.Mikolov et al. Efficient estimation of word representations in vector space, 2013.

Связь word2vec с матричными разложениями

d — размерность векторов слов x_w и слов-из-контекста y_u
$X = (x_w)_{W \times d}$ — матрица векторов предсказываемых слов
$Y = (y_u)_{W \times d}$ — матрица векторов слов-из-контекста

SGNS строит матричное разложение $P \approx XY^{\mathrm{T}}$ матрицы $W \times W$
Shifted PMI (Point-wise Mutual Information):

$$P_{wu} = \ln \frac{n_{wu} n}{n_w n_u} - \ln k,$$

n_{wu} — частота пары слов w, u в контекстном окне $\pm k$ слов,
n_w, n_u — число пар с участием слова w и u соответственно,
n — число всех пар слов в коллекции.

В качестве эвристики используют также Shifted Positive PMI:

$$P^+_{wu} = \left(\ln \frac{n_{wu} n}{n_w n_u} - \ln k \right)_+.$$

O.Levy, Y.Goldberg. Neural word embedding as implicit matrix factorization, 2014.

Модель векторных представлений FastText

Идея: векторное представление слова w определяется как сумма векторов всех его буквенных n-грамм $G(w)$:

$$u_w = \sum_{g \in G(w)} u_g$$

В Skip-gram вместо векторов слов u_w обучаются векторы u_g

Пример: $G(\text{дармолюб}) = \{<\text{да}, \text{арм}, \text{рмо}, \text{мол}, \text{олю}, \text{люб}, \text{юб}>\}$

Преимущества:

- Это решает проблемы новых слов и слов с опечатками
- Подходит для обработки текстов социальных медиа
- Словарь 2- и 3-грамм обычно меньше словаря W
- Существует много предобученных моделей

Bojanowski et al. Enriching word vectors with subword information. 2016.

Модели векторных представлений текстов и графов

word2vec: *эмбединги* (векторные представления) слов
T.Mikolov et al. Efficient estimation of word representations in vector space. 2013.

paragraph2vec: эмбединги фрагментов или документов
Q.Le, T.Mikolov. Distributed representations of sentences and documents. 2014.

sent2vec: эмбединги предложений
M.Pagliardini et al. Unsupervised learning of sentence embeddings using compositional n-gram features. 2017.

FastText: эмбединги символьных *n*-грамм
`https://github.com/facebookresearch/fastText`

node2vec: эмбединги вершин графа
A.Grover, J.Leskovec. Node2vec: scalable feature learning for networks. 2016.

graph2vec: более общие эмбединги на графах
A.Narayanan et al. Graph2vec: learning distributed representations of graphs. 2017.

StarSpace: эмбединги чего угодно от Facebook AI Research
L.Wu, A.Fisch, S.Chopra, K.Adams, A.B.J.Weston. StarSpace: embed all the things! 2018.

BERT: контекстно-зависимые эмбединги от Google AI Language
J.Devlin et al. BERT: pre-training of deep bidirectional transformers for language understanding. 2018.

GPT-3: эмбединги, предобученные по 570Gb текстов от OpenAI
T.B.Brown et al. Language Models are Few-Shot Learners. 2020.

Трасформер для машинного перевода

Трасформер (transformer) — это нейросетевая архитектура для трансформации векторов слов в контекстно-зависимые

Схема преобразований данных в машинном переводе:

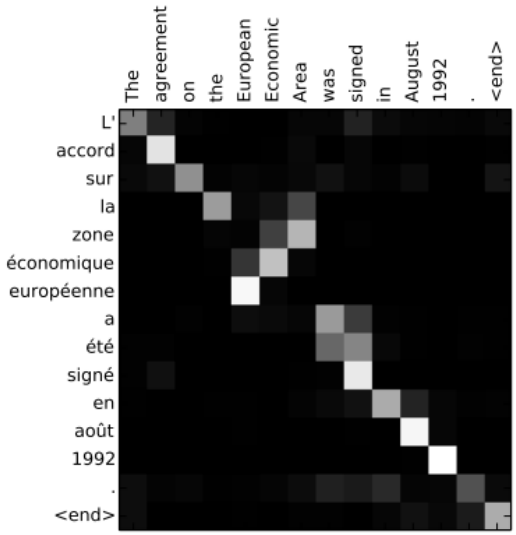
- $S = (w_1, \ldots, w_n)$ — слова предложения на входном языке
  $\downarrow$ обучаемая или пред-обученная векторизация слов
- $X = (x_1, \ldots, x_n)$ — векторы слов входного предложения
  $\downarrow$ трансформер-кодировщик
- $Z = (z_1, \ldots, z_n)$ — контекстно-зависимые векторы слов
  $\downarrow$ 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

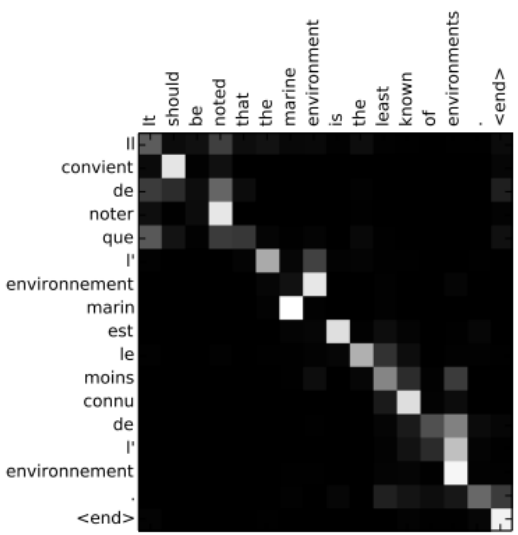
Vaswani et al. (Google) Attention is all you need. 2017.

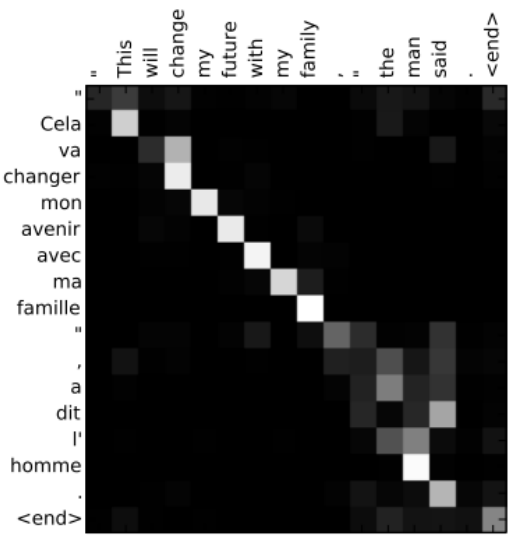
Модели внимания для машинного перевода

$X = (x_1, \ldots, x_n)$ — векторы слов входного предложения
$Y = (y_1, \ldots, y_m)$ — векторы слов выходного предложения

Модель внимания оценивает матрицу семантического сходства $A_{ti} = a(x_i, y_t)$ — насколько входное слово x_i важно (требует внимания) для обработки выходного слова y_t







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

Модель внимания Query–Key–Value

q — *вектор-запрос* для трансформации в вектор-контекст z
$K = (k_1, \ldots, k_n)$ — *векторы-ключи*, сравниваемые с запросом
$X = (x_1, \ldots, x_n)$ — *векторы-значения*, образующие контекст

Модель внимания — трёхслойная сеть, вычисляющая z как выпуклую комбинацию векторов x_i, релевантных запросу q:

$$z = \text{Attn}(q, K, X) = \sum_i x_i \,\text{SoftMax}_i\, a(k_i, q),$$

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):

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

трансформирует входную последовательность $X = (x_1, \ldots, x_n)$
в выходную последовательность векторов контекста $(z_1, \ldots, z_n)$

Архитектура трансформера-кодировщика

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

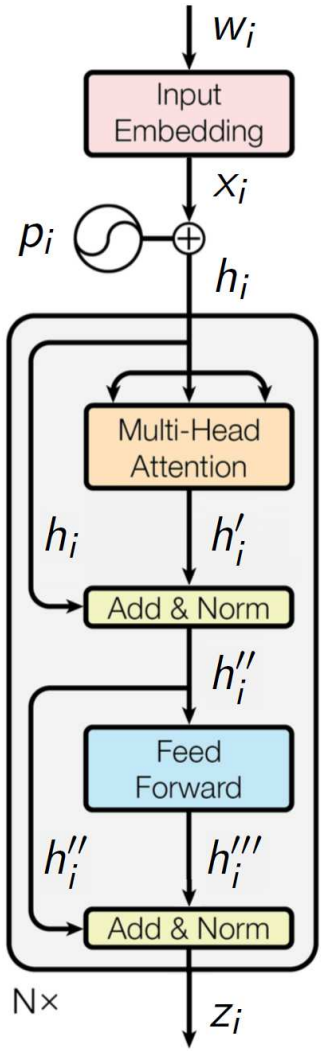
2. Многомерное самовнимание:
$h_i^j = \text{Attn}(W_q^j h_i, W_k^j H, W_v^j H)$
$j = 1, \ldots, J = 8$
$\dim h_i^j = 64$
$\dim W_q^j, W_k^j, W_v^j = 64 \times 512$

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

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:
$z_i = \text{LN}(h_i''' + h_i''; \mu_2, \sigma_2)$
$\dim z_i, \mu_2, \sigma_2 = 512$



Несколько дополнений и замечаний

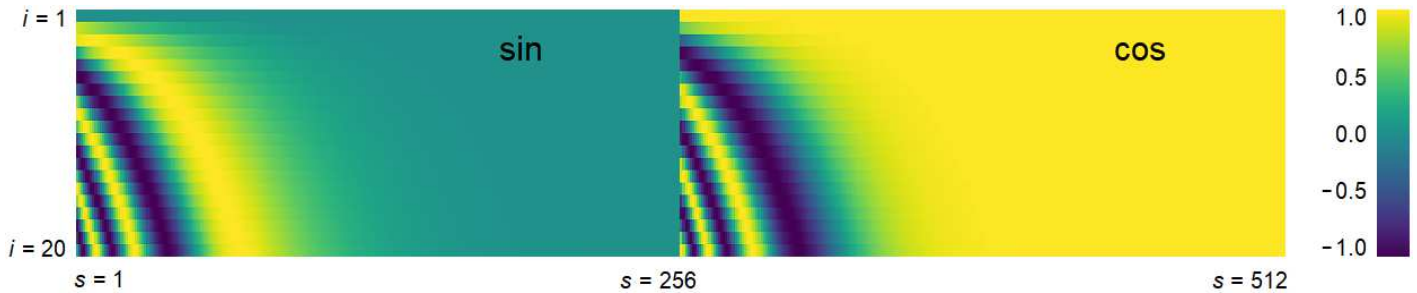
- $N = 6$ блоков $h_i \to \square \to z_i$ соединяются последовательно
- эмбединги слов $x_i \in \mathbb{R}^d$ — обучаемые или пред-обученные
- нормировка уровня (Layer Normalization), $x, \mu, \sigma \in \mathbb{R}^d$:

$$\mathrm{LN}_s(x; \mu, \sigma) = \sigma_s \frac{x_s - \bar{x}}{\sigma_x} + \mu_s, \quad s = 1, \ldots, d,$$

$\bar{x} = \frac{1}{d}\sum_s x_s$ и $\sigma_x^2 = \frac{1}{d}\sum_s (x_s - \bar{x})^2$ — среднее и дисперсия x

- Позиции слов i кодируются векторами p_i, $i = 1, \ldots, n$;
чем больше $|i - j|$, тем больше $\|p_i - p_j\|$, n не ограничено:

$$p_{is} = \sin(i\, 10^{-8\frac{s}{d}}), \quad p_{i,s+\frac{d}{2}} = \cos(i\, 10^{-8\frac{s}{d}}), \quad s = 1, \ldots, \frac{d}{2}$$

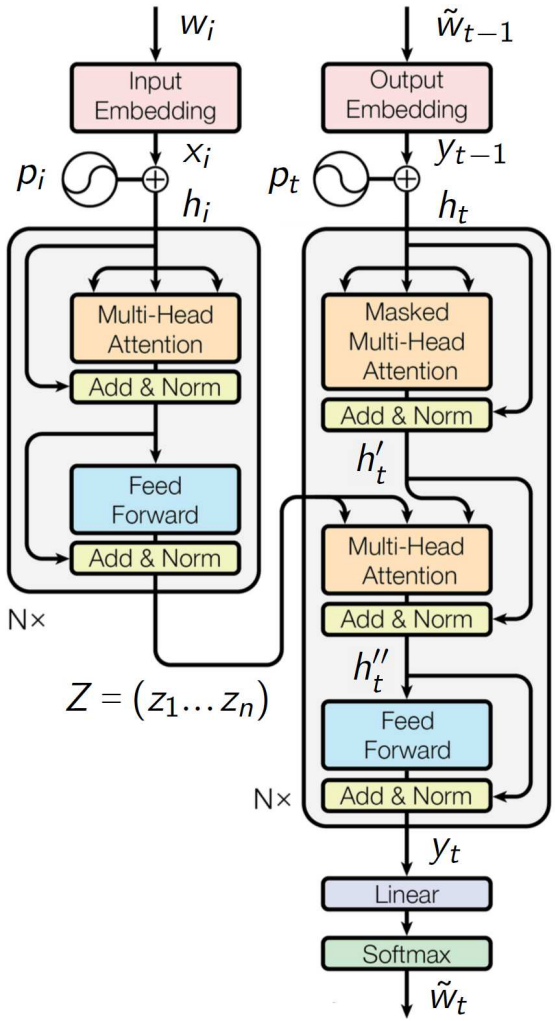


Архитектура трансформера декодировщика

Авторегрессионный синтез последовательности:
$y_0 = \langle \mathtt{BOS} \rangle$ — вектор 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:
   $h'_t = \mathrm{LN} \circ \mathrm{MH}_j \circ \mathrm{Attn}(W^j_q h_t, W^j_k H_t, W^j_v H_t)$
3. Многомерное внимание на кодировку Z:
   $h''_t = \mathrm{LN} \circ \mathrm{MH}_j \circ \mathrm{Attn}(\tilde{W}^j_q h'_t, \tilde{W}^j_k Z, \tilde{W}^j_v Z)$
4. Двухслойная полносвязная сеть:
   $y_t = \mathrm{LN} \circ \mathrm{FFN}(h''_t)$
5. Линейный предсказывающий слой:
   $p(\tilde{w}|t) = \mathrm{SoftMax}_{\tilde{w}}(W_y y_t + b_y)$

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



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

Критерии обучения и валидации для машинного перевода

Критерий для обучения параметров нейронной сети W по обучающей выборке предложений S с переводом $\tilde{S}$:

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

Критерий оценивания моделей (недифференцируемые) по выборке пар предложений «перевод S, эталон S_0»:

BiLingual Evaluation Understudy:

$$\text{BLEU} = \min\left(1, \frac{\Sigma \text{len}(S)}{\Sigma \text{len}(S_0)}\right) \underset{(S_0,S)}{\text{mean}} \left(\prod_{n=1}^{4} \frac{\#n\text{-грамм из } S, \text{ входящих в } S_0}{\#n\text{-грамм в } S} \right)^{\frac{1}{4}}$$

Word Error Rate:

$$\text{WER} = \underset{(S_0,S)}{\text{mean}} \left(\frac{\#\text{вставок} + \#\text{удалений} + \#\text{замен}}{\text{len}(S)} \right)$$

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 автоматической обработки текста

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.

Критерий MLM (masked language modeling) для обучения BERT

Критерий маскированного языкового моделирования MLM, строится автоматически по текстам (self-supervised learning):

$$\sum_S \sum_{i \in M(S)} \ln p(w_i | i, S, W) \to \max_W,$$

где $M(S)$ — подмножество (15%) маскированных токенов из S,

$$p(w | i, S, W) = \underset{w \in V}{\text{SoftMax}}(W_z z_i(S, W_T) + b_z)$$

— языковая модель, предсказывающая i-й токен предложения S;

$z_i(S, W_T)$ — контекстный эмбединг i-го токена предложения S на выходе трансформера-кодировщика с параметрами W_T;

$W = (W_T, W_z, b_z)$ — все параметры языковой модели

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

Критерий NSP (next sentence prediction) для обучения BERT

Критерий предсказания связи между предложениями NSP, строится автоматически по текстам (self-supervised learning):

$$\sum_{(S,S')} \ln p\big(y_{SS'}|S,S',W\big) \to \max_W,$$

где $y_{SS'} = [\text{за } S \text{ следует } S']$ — классификация пары предложений,

$$p(y|S,S',W) = \underset{y\in\{0,1\}}{\text{SoftMax}}\big(W_y\,\text{th}(W_s z_0(S,S',W_T) + b_s) + b_y\big)$$

— вероятностная модель бинарной классификации пар (S,S'), $z_0(S,S',W_T)$ — контекстный эмбединг токена $\langle\texttt{CLS}\rangle$ для пары предложений, записанной в виде $\langle\texttt{CLS}\rangle\ S\ \langle\texttt{SEP}\rangle\ S'\ \langle\texttt{SEP}\rangle$

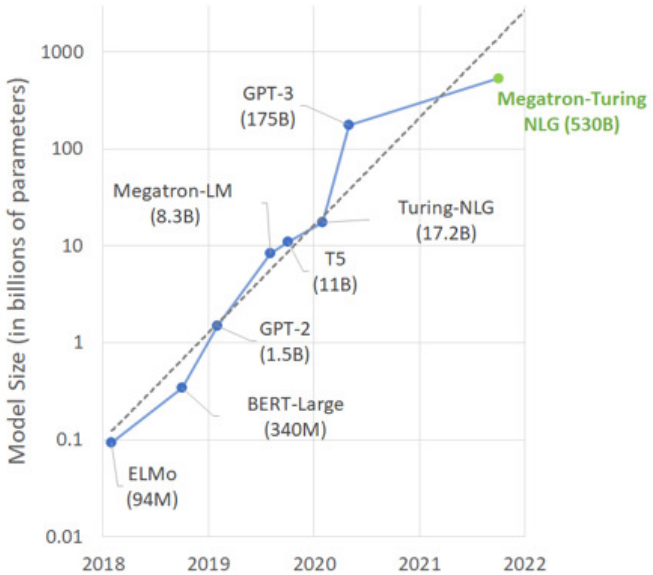
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language) BERT: pre-training of deep bidirectional transformers for language understanding. 2019.

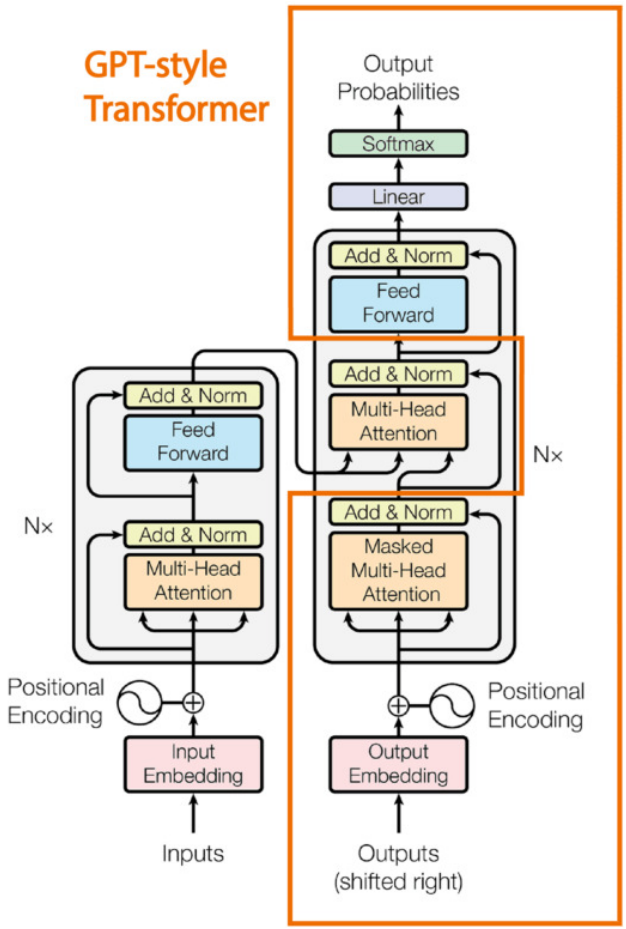
Ещё несколько замечаний про трансформеры

- **Fine-tuning:** для дообучения на задаче задаётся модель $f(Z(S, W_T), W_f)$, выборка $\{S\}$ и критерий $\mathscr{L}(S, f) \to \max$
- **Multi-task learning:** для дообучения на наборе задач $\{t\}$ задаются модели $f_t(Z(S, W_T), W_t)$, выборки $\{S\}_t$ и сумма критериев $\sum_t \lambda_t \sum_S \mathscr{L}_t(S, f_t) \to \max$
- *GLUE*, *SuperGLUE*, *Russian SuperGLUE*, *MERA*, *SLAVA* — наборы тестовых задач на понимание и генерацию языка
- Трансформеры обычно строятся не на словах, а на токенах, получаемых BPE (Byte-Pair Encoding) или WordPiece
- Первый трансформер: $N = 6$, $d = 512$, $J = 8$, весов 65M
- $\text{BERT}_{\text{BASE}}$, GPT1: $N = 12$, $d = 768$, $J = 12$, весов 110M
- $\text{BERT}_{\text{LARGE}}$: $N = 24$, $d = 1024$, $J = 16$, весов 340M

Генеративный предобученный трансформер (GPT, Open AI)

Generative pre-trained transformer:
— архитектура декодировщика остаётся (отличия непринципиальны)
— размер моделей имеет значение:



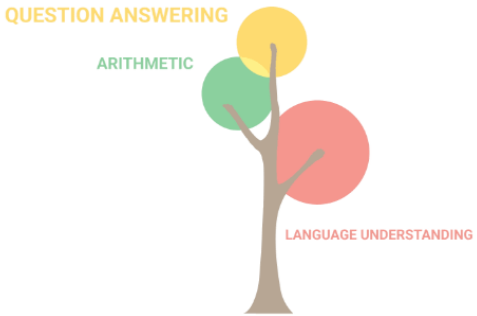


A.Radford et al. Improving language understanding by generative pre-training. 2018
A.Radford et al. Language models are unsupervised multitask learners. 2019 (GPT-2)
T.B.Brown et al. Language models are few-shot learners. 2020 (GPT-3)

Эмерджентность — появление качественно новых способностей

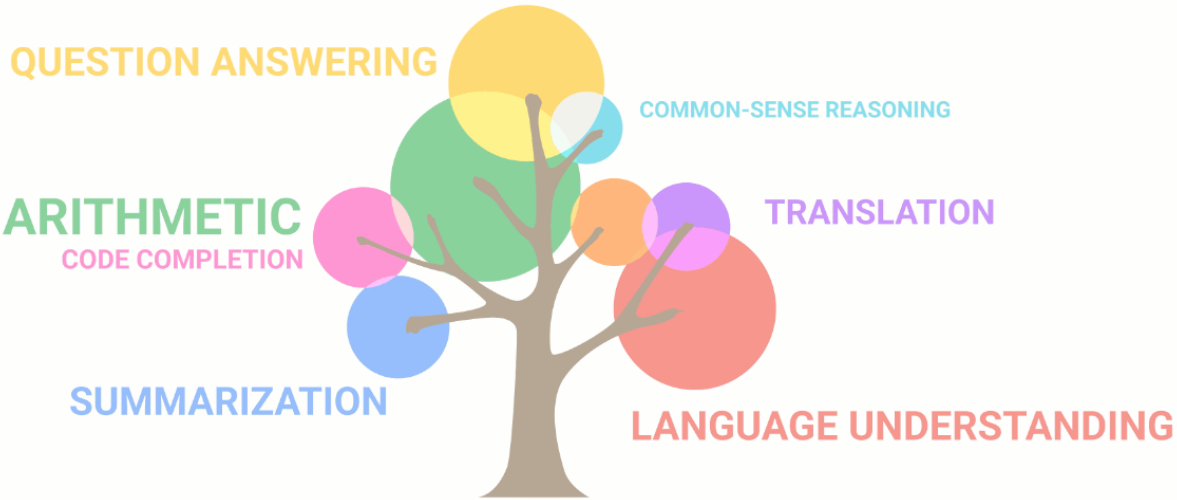
Впервые за 70 лет истории AI/ML модель выучила намного больше того, чему её, казалось бы, учили (MLM, NSP)

Почему? Человеческий язык является коммуникативным инструментом для решения любых задач в реальном мире



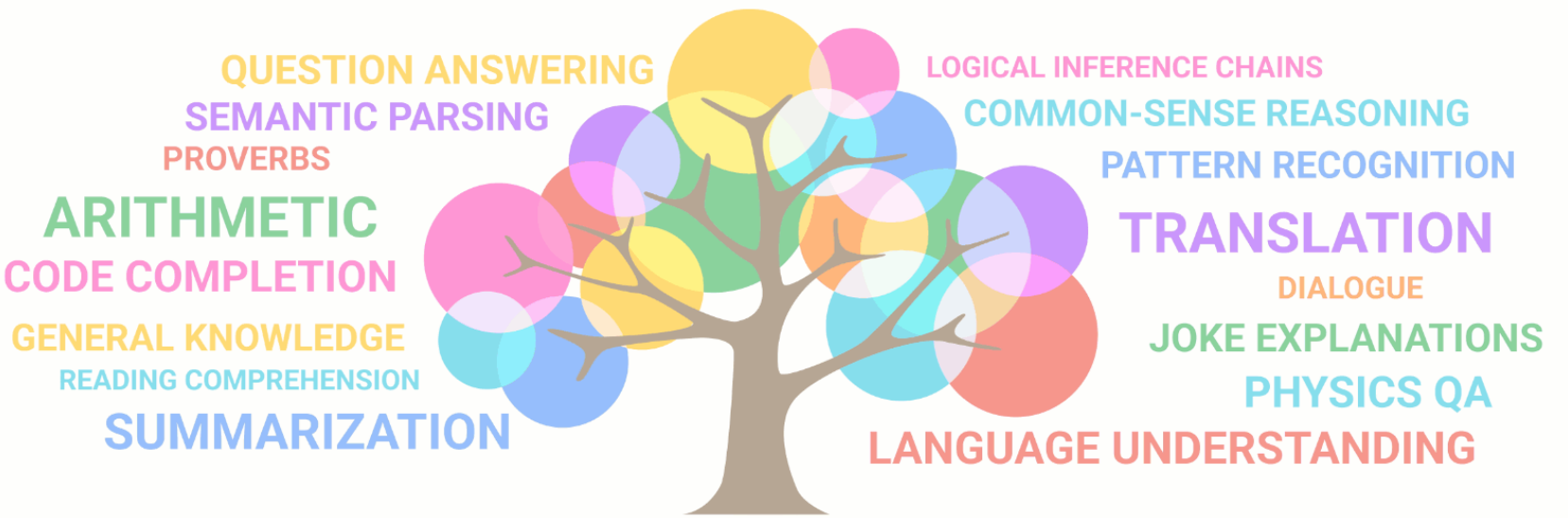
- GPT-2: 14/Feb/2019, контекст 768 слов (1,5 страницы)
- 1,5 млрд. параметров, корпус 10 млрд. токенов (40Gb)
- генерация литературного эссе, которое конкурсное жюри не смогло отличить от написанного человеком

Эмерджентность — появление качественно новых способностей



- GPT-3: 11/Jun/2020, контекст 1536 слов (3 страницы)
- 175 млрд. параметров, корпус 500 млрд. токенов
- переводы на другие языки
- решение простых логических и математических задач
- генерация программного кода по описанию требований

Эмерджентность — появление качественно новых способностей



- GPT-4: 14/Mar/2023, контекст 24 000 слов (48 страниц)
- >1 трл. параметров, корпус >1Tb
- исправление ошибки по подсказке «let's think step by step»
- описание и анализ изображений
- решение качественных физических задач по картинке

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 логические и математические задачи
- искать и исправлять собственные ошибки по подсказке

Резюме

- Обучаемая векторизация сложно структурированных данных — одна из важнейших концепций ML/DL
- Представление текста в виде графа (системы локальных контекстов слов) содержит информацию о смыслах слов
- Эмбединги графов обобщают многие задачи векторизации текстов, дискретных сигналов, изображений и др.
- Доказано, что модель внимания multi-head self-attention (MHSA) эквивалентна свёрточной сети [Cordonnier, 2020]

Открытые проблемы

- Возможно ли упростить архитектуру Трансформера, сильно сократив число параметров без потери качества?
- Возможно ли извлечь из Трансформера понятные людям хорошо структурированные знания о языке и о мире?

Xipeng Qiu et al. Pre-trained models for natural language processing: A survey. 2020.
Cordonnier et al. On the relationship between self-attention and convolutional layers. 2020