• Вероятностные языковые модели •
Лекция 3.
Нейросетевые языковые модели

Константин Вячеславович Воронцов
k.vorontsov@iai.msu.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Вероятностные языковые модели (курс лекций, К.В.Воронцов)»

ВМК МГУ • 16 марта 2026

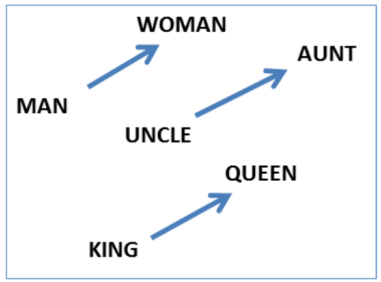
Содержание

Эволюция подходов машинного обучения в анализе текстов

Анализ текстов 15 лет назад: пирамида NLP

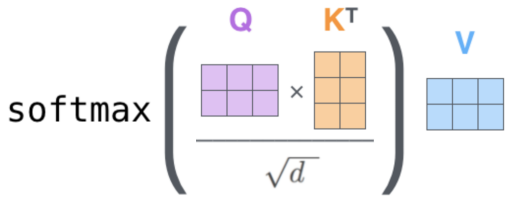
- морфологический анализ, лемматизация, опечатки
- синтаксический анализ, выделение терминов, NER
- семантический анализ, выделение фактов, тем

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, 2003], 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]
- тематические модели внимания?

Общая постановка большинства задач машинного обучения

Дано: X — пространство *объектов*
$X^\ell = \{x_1, \ldots, x_\ell\} \subset X$ — *обучающая выборка* (training sample)
$a(x, w)$, $a\colon X \times W \to Y$ — параметрическая модель, гипотеза

Найти $w \in W \subseteq \mathbb{R}^d$ — вектор параметров модели $a(x, w)$

Критерий минимум эмпирического риска (с регуляризацией)
(ERM — Empirical Risk Minimization (with regularization)):

$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(w, x_i) + \tau \mathscr{R}(w) \to \min_w$$

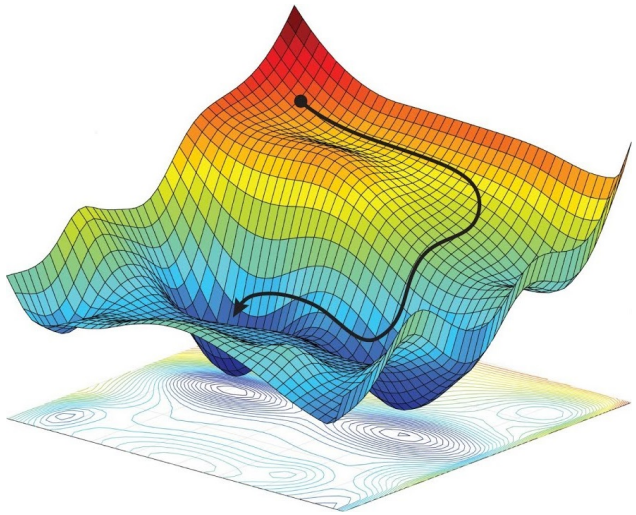
$\mathscr{L}(w, x)$ — *функция потерь* (loss function),
тем больше, чем хуже ответ модели $a(x, w)$ на объекте x:

- $\mathscr{L}(w, x) = (a(w, x) - y(x))^2$ для регрессии, $Y = \mathbb{R}$
- $\mathscr{L}(w, x) = [a(w, x) \neq y(x)]$ для классификации, $|Y| < \infty$

$\mathscr{R}(w)$ — *регуляризатор*, непрецедентные требования к модели

Градиентная минимизация эмпирического риска

$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}(w, x_i) + \tau\mathscr{R}(w) \to \min_w$$



Метод *градиентного спуска*:

$w^{(0)} :=$ начальное приближение;

$w^{(t+1)} := w^{(t)} - h\nabla Q(w^{(t)})$;

где $\nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=1}^{d}$ — *вектор градиента*,

h — *градиентный шаг*, называемый также *темпом обучения*

$$w^{(t+1)} := w^{(t)} - h\left(\frac{1}{\ell}\sum_{i=1}^{\ell}\nabla\mathscr{L}\left(w^{(t)}, x_i\right) + \tau\nabla\mathscr{R}\left(w^{(t)}\right)\right)$$

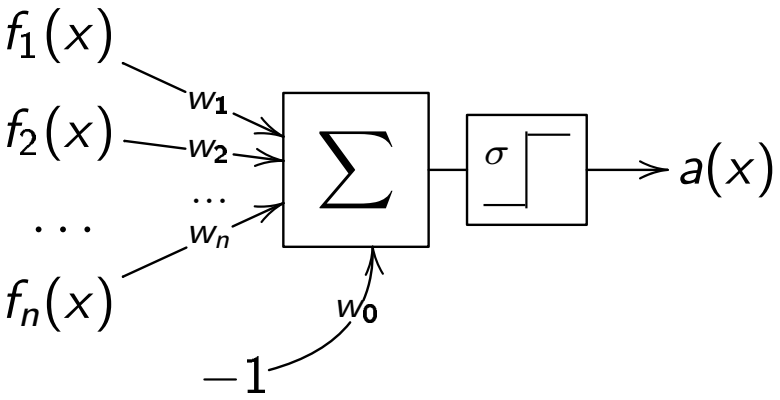
Ускорение сходимости: чтобы чаще обновлять вектор w, берём случайные подмножества или даже объекты по одному — *метод стохастического градиента* (Stochastic Gradient, SG)

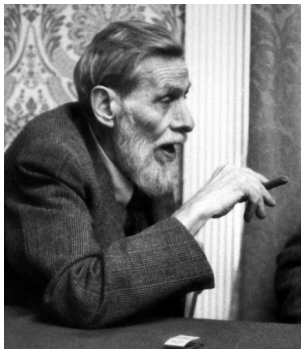
Искусственный нейрон — линейная модель классификации

Линейная модель нейрона (1943):

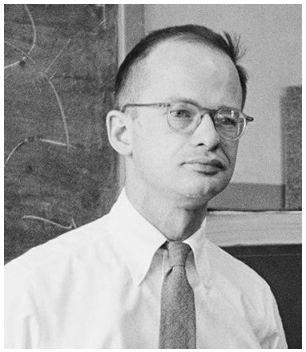
$$a(x,w) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right)$$

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

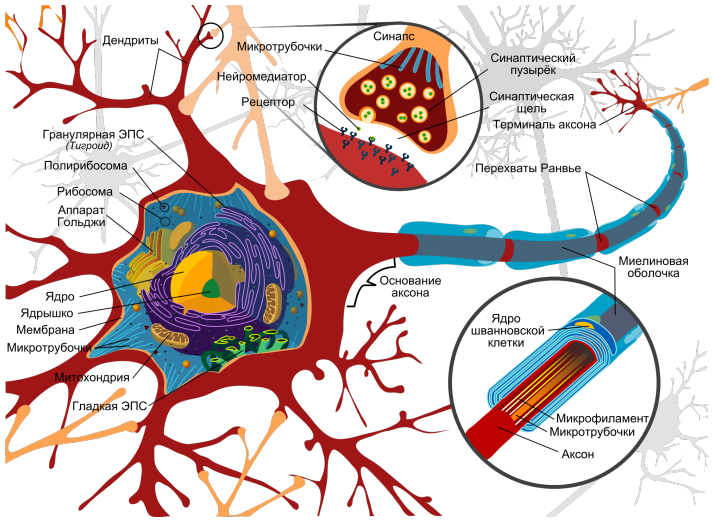




Уоррен МакКаллок (1898–1969)



Вальтер Питтс (1923–1969)



Многослойный персептрон (MultiLayer Perceptron, MLP)

Архитектура сети: H_l — число нейронов в l-м слое, $l = 1, \ldots, L$

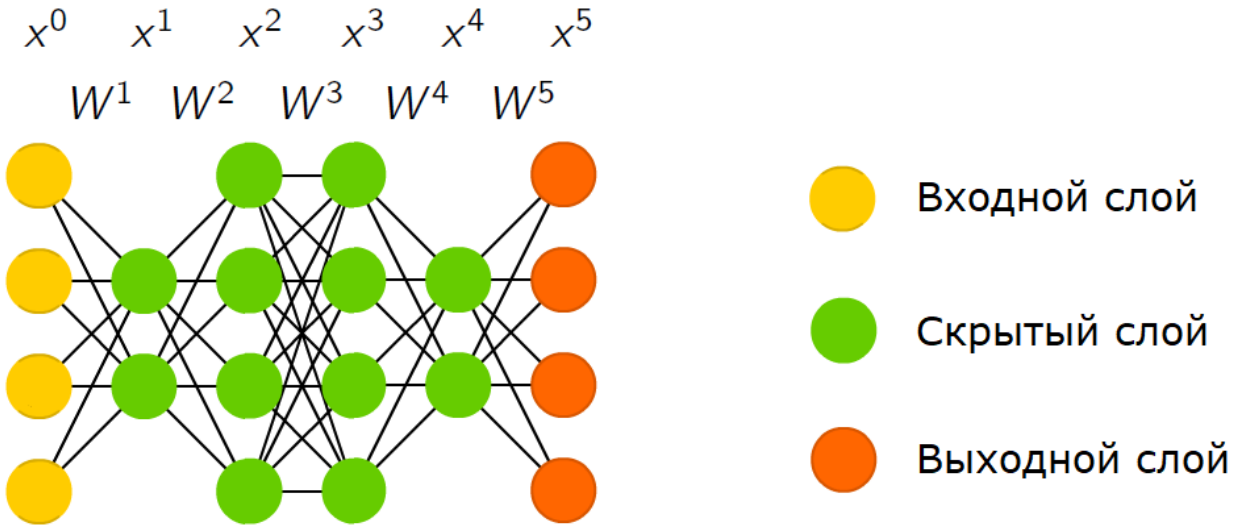
$x^0 \in \mathbb{R}^{n+1}$ — вектор признаков на входе сети, $x_0^0 = -1$

$x^l \in \mathbb{R}^{H_l}$ — вектор «признаков» на выходе l-го слоя, $x_0^l = -1$

$x^L \in \mathbb{R}^{H_L}$ — вектор на выходе сети

W^l — матрица весов l-го слоя, размера $(H_{l-1}+1) \times H_l$

$x^l = \sigma^l\left(W^l x^{l-1}\right)$ — вычисление сети по слоям $l = 1, \ldots, L$

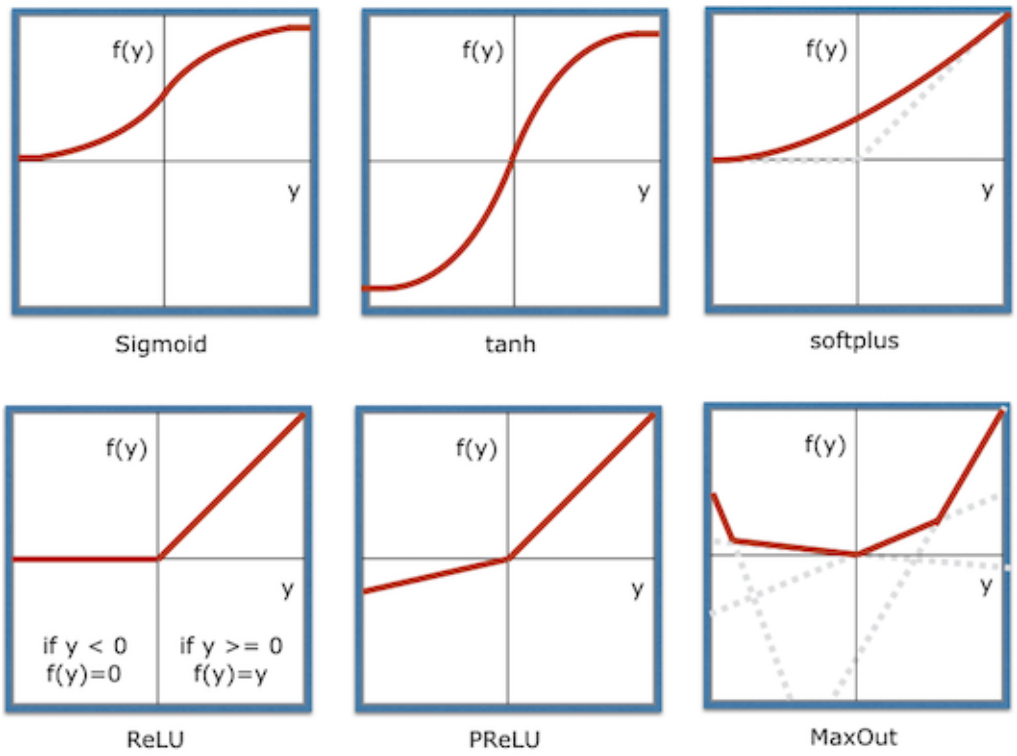


Функции активации

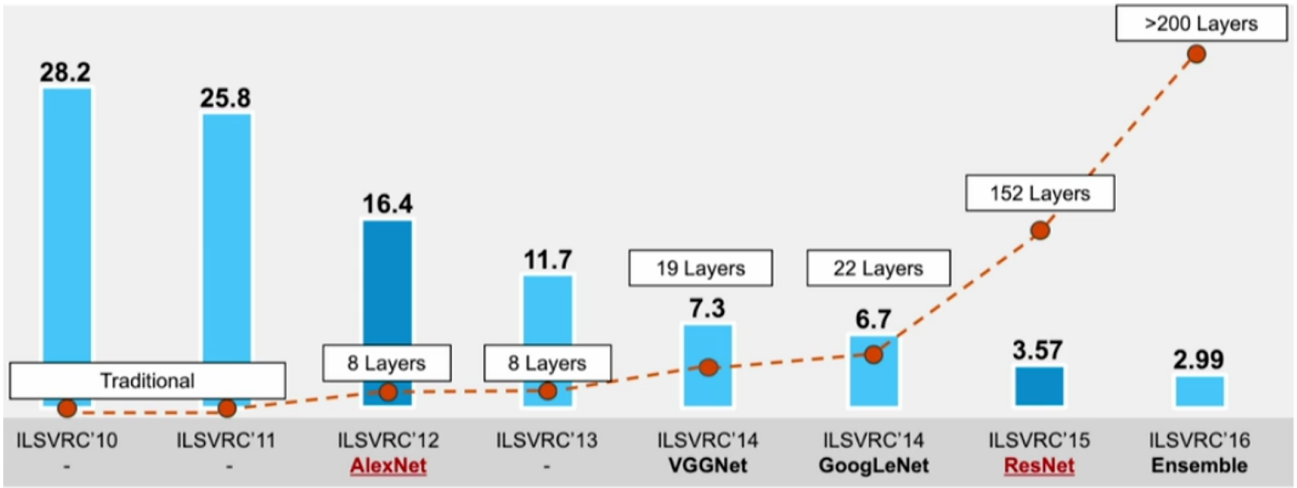
Функции $\sigma(y) = \frac{1}{1+e^{-y}}$ и $\text{th}(y) = \frac{e^{y}-e^{-y}}{e^{y}+e^{-y}}$ могут приводить к затуханию градиентов или «параличу сети»

Функция положительной срезки (Rectified Linear Unit, ReLU)

$$\text{ReLU}(y) = \max\{0, y\}; \qquad \text{PReLU}(y) = \max\{0, y\} + \alpha \min\{0, y\}$$



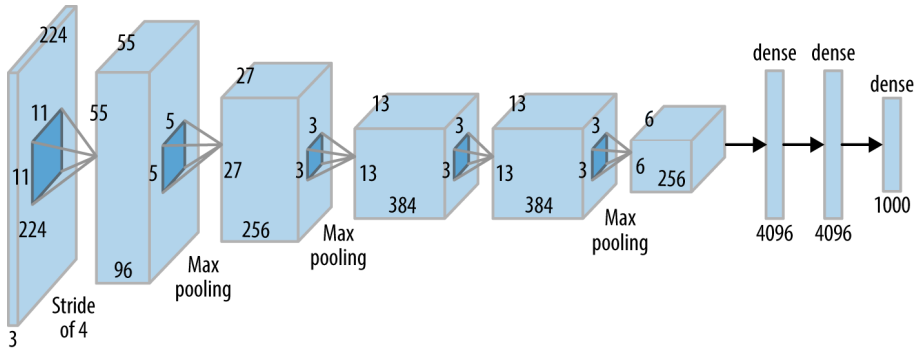
Глубокие свёрточные сети для классификации изображений



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

Свёрточная сеть **AlexNet**:

+ ReLU + Dropout
+ data augmentation
+ обучение на GPU
+ подбор размеров слоёв
+ в итоге 62M параметров

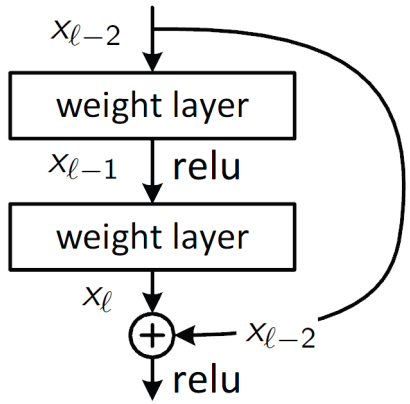


Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012.

ResNet: остаточная нейронная сеть (Residual NN)

Сквозная связь (skip connection) слоя l с предшествующим слоем $l-d$:

$$x_l = \sigma(Wx_{l-1}) + x_{l-d}$$



Слой l выучивает не новое векторное представление x_l, а его приращение $x_l - x_{l-d}$

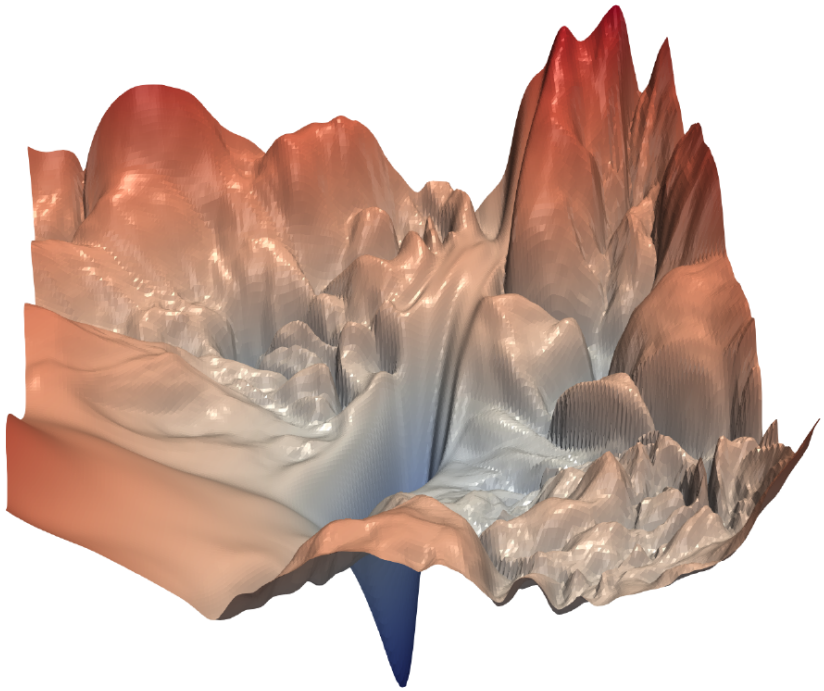
- Приращения более устойчивы $\Rightarrow$ улучшается сходимость
- Появляется возможность увеличивать число слоёв
- Обобщение — Highway Networks:

$$x_l = \sigma(Wx_{l-1})\underbrace{\tau(W'x_{l-1})}_{\text{transform gate}} + x_{l-d}\underbrace{\big(1-\tau(W'x_{l-1})\big)}_{\text{carry gate}}$$

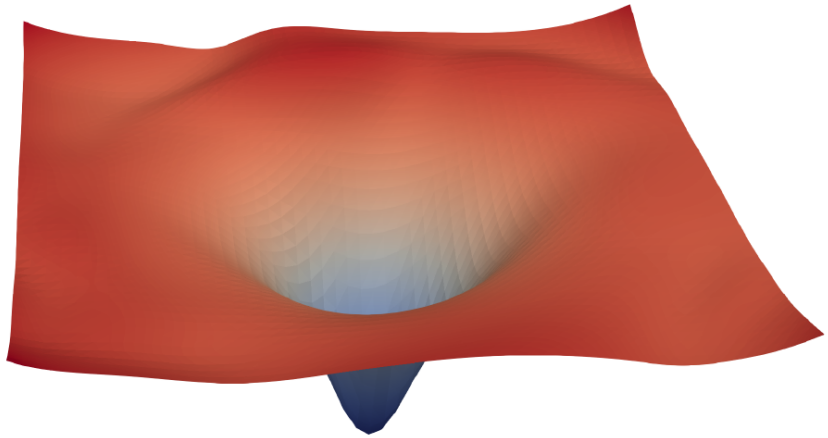
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. 2015
R.K.Srivastava, K.Greff, J.Schmidhuber. Highway Networks. 2015

ResNet: визуализация оптимизационного критерия

Сквозные связи (skip connection) упрощают оптимизируемый критерий, устраняя локальные экстремумы и седловые точки:



without skip connections



with skip connections

Hao Li et al. Visualizing the Loss Landscape of Neural Nets. 2018

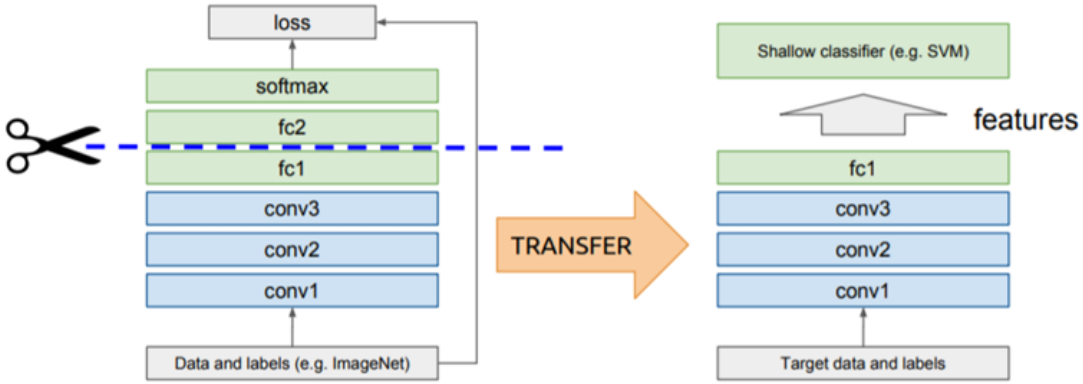
Предобучение (pre-training), перенос обучения (transfer learning)

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 $\{x'_i\}_{i=1}^{m}$:

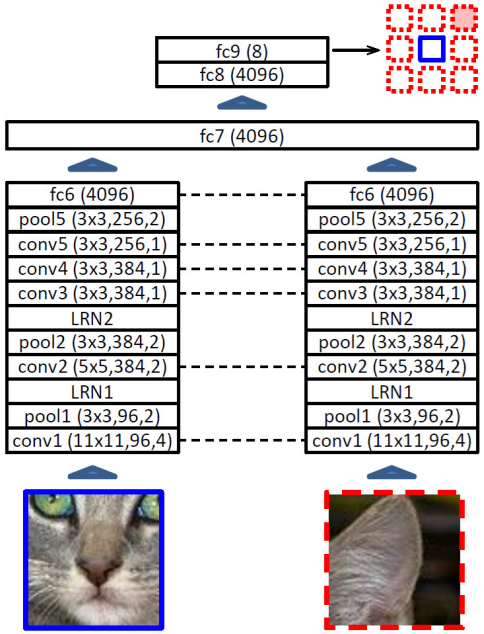
$$\sum_{i=1}^{m} \mathscr{L}'_i\big(g'(f(x'_i, \alpha), \beta')\big) \;\to\; \min_{\beta'}$$

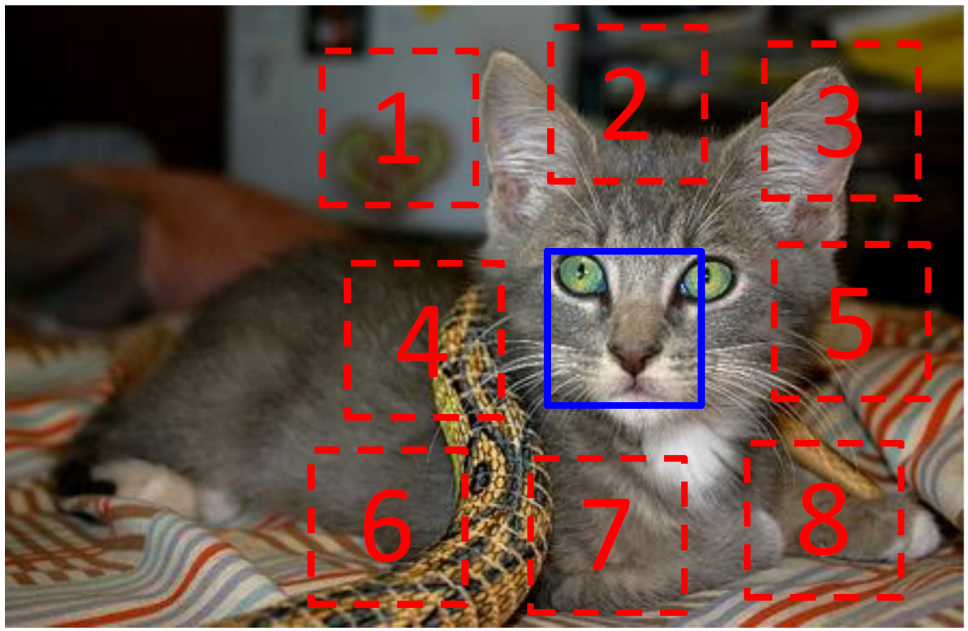


Sinno Jialin Pan, Qiang Yang. A Survey on Transfer Learning. 2009
J.Yosinski et al. How transferable are features in deep neural networks? 2014.

Самостоятельное обучение (self-supervised learning)

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





Векторные представления, обученные по искусственной задаче, оказываются не хуже обученных по данным ImageNet

---

C.Doersch, A.Gupta, A.Efros. Unsupervised visual representation learning by context prediction. ICCV 2015.

Глубокие сети — не мозг, а обучаемая векторизация данных

Предсказательное моделирование:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}\big(a(f(x_i,\alpha),w),y_i\big)+\tau\mathscr{R}(\alpha,w)\to\min_{\alpha,w},$$

$f_\alpha\colon X\to\mathbb{R}^d$ преобразует данные x в вектор признаков $z=f(x,\alpha)$
$a_w\colon\mathbb{R}^d\to Y$ по вектору z предсказывает $\hat{y}=a(z,w)\approx y(x)$
$\mathscr{L}(\hat{y},y)$ оценивает ошибку предсказания $\hat{y}$ при целевом y

Автокодировщики:

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

$f_\alpha\colon X\to\mathbb{R}^d$ кодирует x в кодовый вектор $z=f(x,\alpha)$
$g_\beta\colon\mathbb{R}^d\to X$ декодирует z в реконструкцию $\hat{x}=g(z,\beta)\approx x$
$\mathscr{L}(\hat{x},x)$ оценивает отличие реконструкции $\hat{x}$ от целевого x

Трасформер для машинного перевода

Трасформер (transformer) — это нейросетевая архитектура для трансформации векторов слов с учётом их контекста

Схема преобразований данных в машинном переводе:

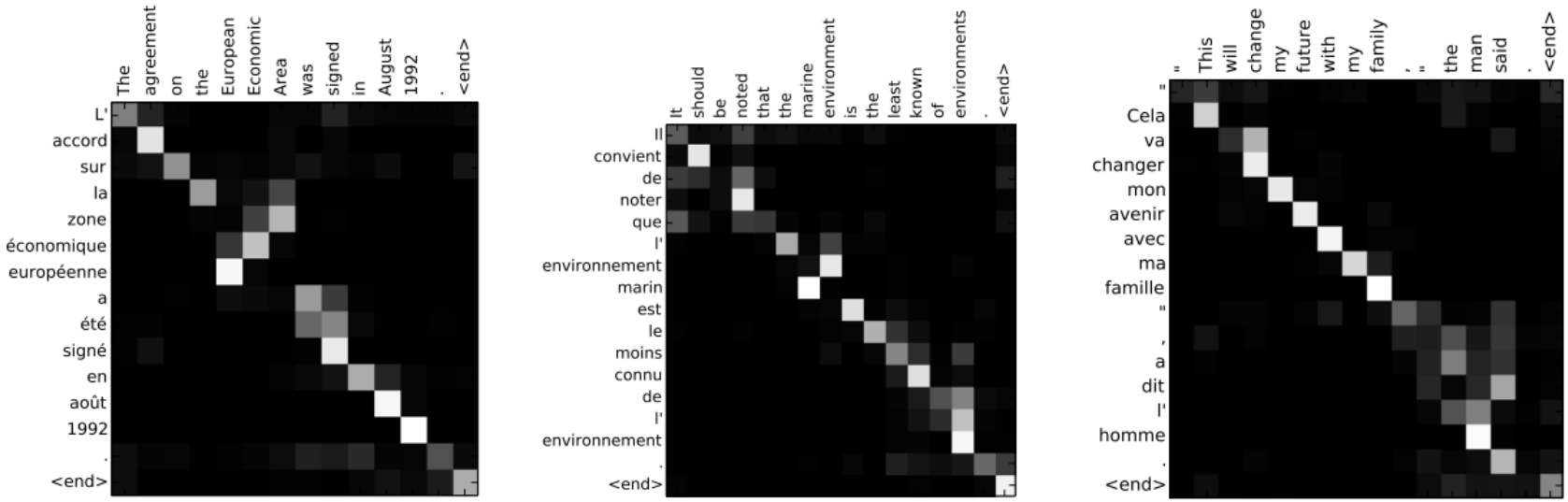
- $S = (w_1, \ldots, w_n)$ — слова предложения на входном языке
  $\downarrow$ обучаемая или пред-обученная векторизация слов
- $X = (x_1, \ldots, x_n)$ — векторы слов входного предложения
  $\downarrow$ трансформер-кодировщик
- $Z = (z_1, \ldots, z_n)$ — контекстно-зависимые векторы слов
  $\downarrow$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

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

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

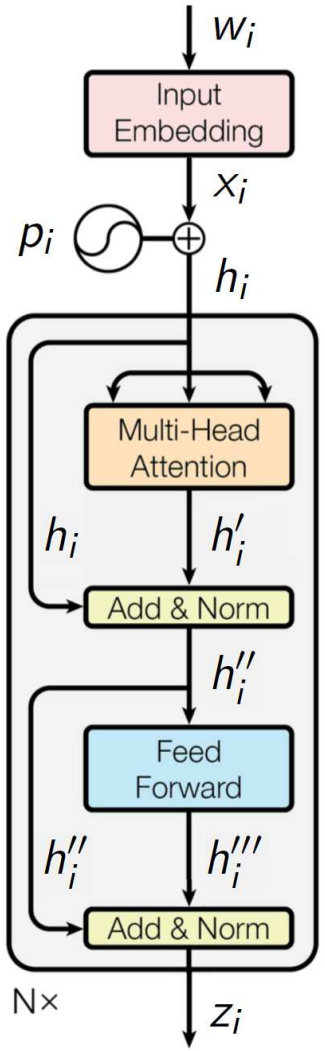
2. J голов самовнимания:
$h_i^j = \mathrm{Attn}(W_q^j h_i, W_k^j H, W_v^j H)$ $\quad j = 1, \ldots, J = 8$, $\dim h_i^j = 64$, $\dim W_q^j, W_k^j, W_v^j = 64 \times 512$

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

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

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

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



Несколько дополнений и замечаний

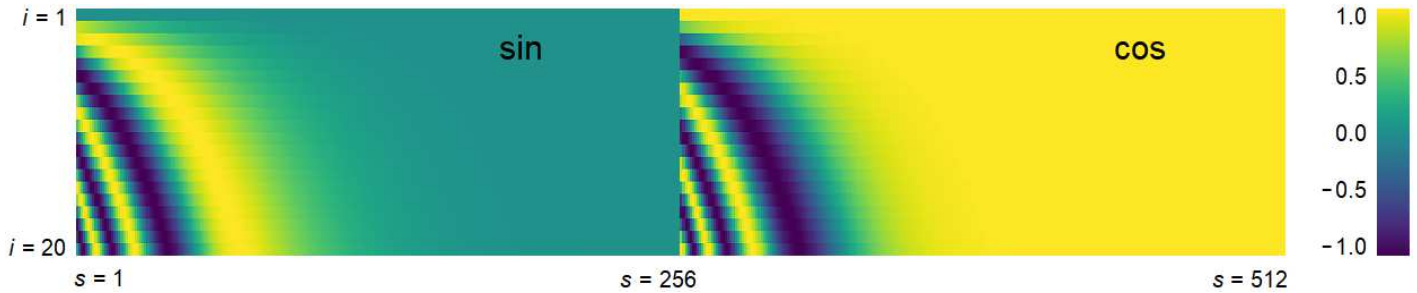
- $N = 6$ блоков $h_i \to \square \to z_i$ соединяются последовательно
- эмбединги слов $x_i \in \mathbb{R}^d$ — обучаемые или пред-обученные
- нормировка уровня (Layer Normalization), $x, \mu, \sigma \in \mathbb{R}^d$:

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

$\bar{x} = \frac{1}{d}\sum_s x_s$ 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

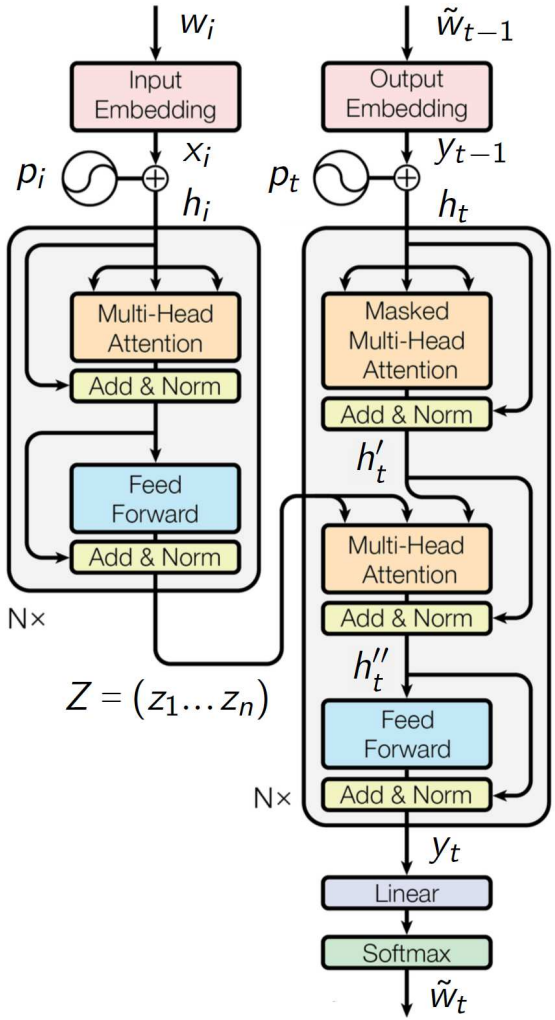
Авторегрессионный синтез последовательности:

$y_0 = \langle \mathtt{BOS} \rangle$ — 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:
   $p(\tilde{w}|t) = \underset{\tilde{w}}{\mathrm{SoftMax}}(W_y y_t + b_y)$

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 оценивания 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,

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

— 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,

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

— 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

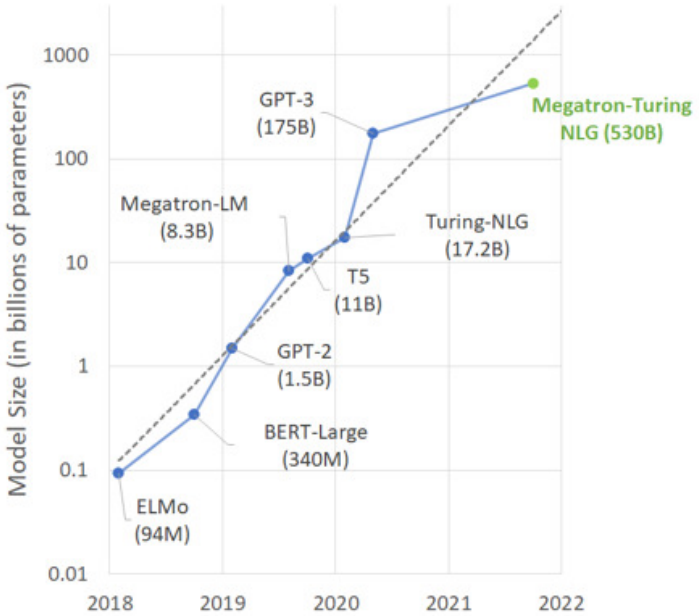
- **Fine-tuning:** для дообучения на задаче задаётся модель $f(Z(S, W_T), W_f)$, выборка $\{S\}$ и критерий $\mathscr{L}(S, f) \to \max$
- **Multi-task learning:** для дообучения на наборе задач $\{t\}$ задаются модели $f_t(Z(S, W_T), W_t)$, выборки $\{S\}_t$ и сумма критериев $\sum_t \lambda_t \sum_S \mathscr{L}_t(S, f_t) \to \max$
- 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
- *GLUE*, *SuperGLUE*, *Russian SuperGLUE*, *MERA*, *SLAVA* — наборы тестовых задач на понимание и генерацию языка

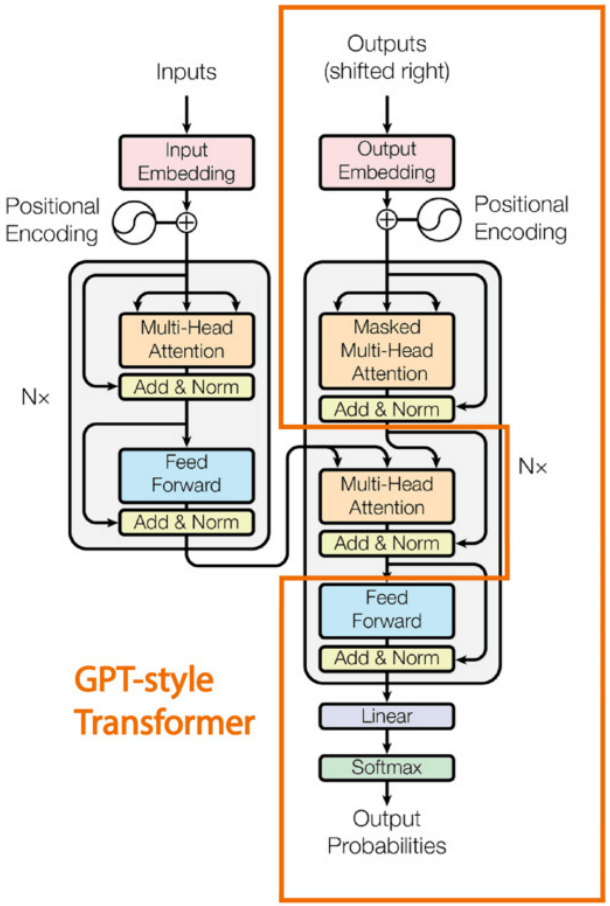
Генеративный предобученный трансформер (GPT, Open AI)

Generative Pre-trained Transformer:

— архитектура декодировщика остаётся (отличия не принципиальные)

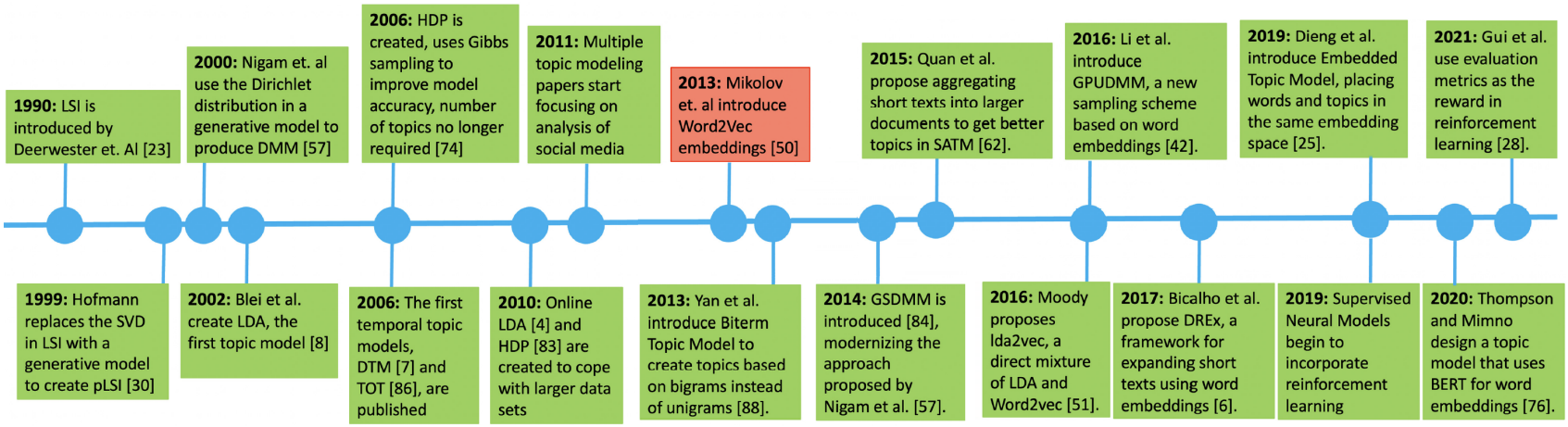
— размер моделей имеет значение:





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)

Эволюция тематического моделирования



1999 PLSA — Probabilistic Latent Semantic Analysis

2001 LDA — Latent Dirichlet Allocation

200x мультимодальные, темпоральные, иерархические модели

2013 модели битермов и WNTM — аналоги word2vec

2016 тематические модели на основе предобученных word2vec

2020 BERTopic — TM на основе предобученного BERT

202x огромное разнообразие NTMs — Neural Topic Models...

Rob Churchill, Lisa Singh. The Evolution of Topic Modeling. 2022.

Нейросетевые и тематические языковые модели

Преимущества глубоких нейросетевых моделей

- *генеративность:* способны порождать связный текст
- *универсальность:* решают широкий класс задач NLP/NLU
- *предобученность:* «знают всё о языке» (и о мире)

Преимущества вероятностных тематических моделей:

- *интерпретируемость* тематических эмбедингов
- *эффективность* для узкого класса задач NLP/NLU
- *полнота* тематической кластерной структуры коллекции

Как «объединить лучшее от двух миров»?

Что объединяет PTM и LLM, и что их разобщает:

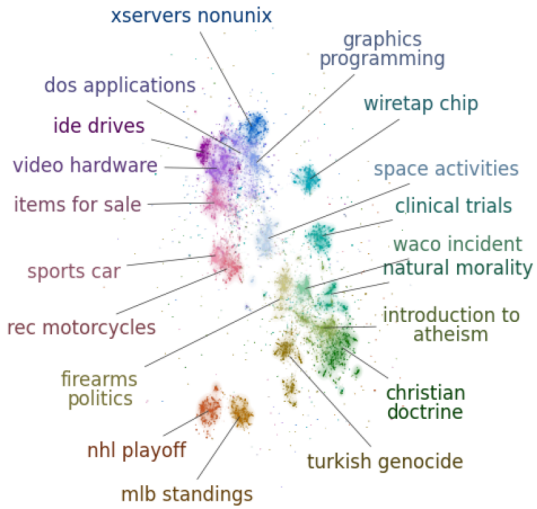
⊕ обе — вероятностные языковые модели,

⊕ обе — автокодировщики, векторные представления текста

⊖ PTM: ~~мешок слов~~, архитектура матричного разложения, ~~байесовское обучение~~, трудности предобучения и др.

Нейросетевая тематическая модель Contextual-Top2Vec

Вместо PTM — конвейер 8 технологий:



1. векторизация токенов (Sentence-BERT)
2. векторизация предложений скользящим окном в 50 токенов (mean pooling)
3. понижение размерности векторов (UMAP)
4. иерархическая кластеризация (hDbscan), автоматическое определение числа тем
5. иерархичекое укрупнение тем слиянием мелких кластеров с ближайшими соседями (Top2Vec)
6. разбиение документа на монотематические сегменты
7. $p(t|d)$ = доля векторов данной темы в документе
8. именование тем: поиск фраз, ближайших к центроиду темы

Dimo Angelov. Top2vec: Distributed representations of topics. 2020.
D.Angelov, D.Inkpen. Topic modeling: contextual token embeddings are all you need. 2024.

Нейросетевая тематическая модель Contextual-Top2Vec

Недостатки:

- это не единая модель, а конвейер эвристических моделей
- долго-дорого, особенно на больших коллекциях
- инкрементное добавление документов не предполагается

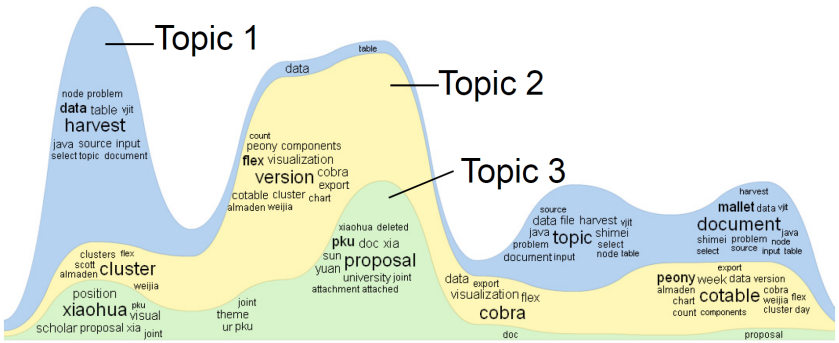
Достоинства — что хотелось бы перенять и встроить в ARTM:

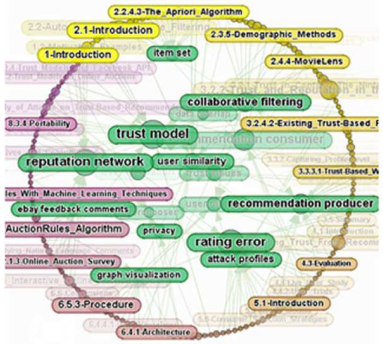
- модель внимания, локальные контексты вместо документов
- отбор релевантных фраз и n-грамм по каждой теме
- именование и суммаризация тем на основе этих фраз
- инициализация тем по предобученным эмбедингам BERT, чтобы обеспечить качество тем даже на малых коллекциях
- автоматическое определение числа тем
- разбиение документа на монотематичные сегменты

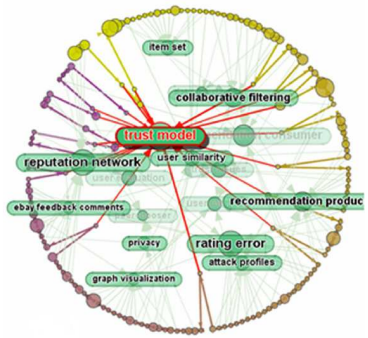
Dimo Angelov. Top2vec: Distributed representations of topics. 2020.
D.Angelov, D.Inkpen. Topic modeling: contextual token embeddings are all you need. 2024.

Мотивации: что хотим от PTMs глядя на BERT, LLM, NTM

- последовательность $w_1, \ldots, w_n$ вместо «мешка слов»
- локальные контексты, как в BERT, вместо документов
- быстрая тематизация любого текста или фрагмента
- быстрый поиск фрагментов, относящихся к данной теме,
- в том числе фраз для суммаризации документа или темы
- сегментация документа на тематически однородные части
- визуализация тематической структуры документа:







Анонс. Контекстная тематическая модель Attentive ARTM

Дано: коллекция текстовых документов, $w_1, \ldots, w_n$
$C_i \subset \{1, \ldots, n\}$ — локальный контекст (окружение) терма w_i
α_{ci} — коэффициент внимания, вес терма w_c из C_i для w_i

Найти: $\phi_{tw} = p(t|w)$ — параметры тематической модели

$$p(w|C_i) = \sum_{t \in T} p(w|t) p(t|C_i) = \sum_{t \in T} p(t|w) \frac{p(w)}{p(t)} p(t|C_i)$$

$$p(t|C_i) \equiv \theta_{ti} = \sum_{c \in C_i} \alpha_{ci} p(t|w_c), \quad \sum_{c \in C_i} \alpha_{ci} = 1, \quad \alpha_{ci} \geqslant 0$$

Критерий: максимум log правдоподобия с регуляризатором R:

$$\sum_{i=1}^{n} \ln \sum_{t \in T} \phi_{tw_i} \frac{p(w_i)}{p(t)} \sum_{c \in C_i} \alpha_{ci} \phi_{tw_c} + R(\Phi) \;\to\; \max_{\Phi}$$

И.А.Ирхин, В.Г.Булатов, К.В.Воронцов. Аддитивная регуляризация тематических моделей с быстрой векторизацией текста, 2020.

Резюме. Открытые проблемы и семейство моделей A*RTM

Цель: «Make Topic Modeling Great Again», а именно,
создать новый стандарт тематического моделирования,
отказавшись от «мешка слов», объединив всё лучшее от:

1. ARTM: регуляризация, модальности, иерархии, транзакции
2. BigARTM: батчи, параллельность, скорость, лёгкость
3. LLM: контекстно-зависимые эмбединги и внимание
4. NTM: каждая тема должна уметь «рассказать о себе»
5. LLM: параметризация модели внимания
6. NTM: согласование тем с предобученными эмбедингами
7. AutoML: настройка гиперпараметров в потоке данных
8. статистические тесты: однородность и согласованность тем

A*RTM означает: Attentive, Apprehensive, Aware, Adaptive,
Automated, Available, etc... Additively Regularized TM

Задания по курсу

Задача-минимум: научиться решать задачи анализа текстов с использованием тематического моделирования

Задача-максимум: получить новый научный результат

виды деятельности	оценка
теоретическая задача	X
теоретическая задача*	$2X$
теоретическая задача**	$3X$
решение прикладной задачи	$10X$
обзор по последним PTM/NTM	$10X$
участие в проекте	$20X$
работа над открытой проблемой	$25X$

где X — оценка за вид деятельности по 5-балльной шкале.
score — суммарная оценка по всем видам деятельности.

Итоговая оценка: $\min\left(5, \lfloor \text{score}/20 \rfloor\right)$ по 5-балльной шкале.

Задания к лекции 1

Упражнения на принцип максимума правдоподобия:

1. Биграммная модель коллекции: $p(w|v) = \xi_{wv}$,
где v — слово, идущее в тексте перед w.
Найти параметры модели ξ_{wv}.

2. Биграммная модель документов: $p(w|v,d) = \xi_{dwv}$.
Найти параметры модели ξ_{dwv}.

Подсказка: применить условия ККТ или основную лемму.

3*. Творческое задание (возможны разные решения).
Предложите модель, разделяющую роли слов в текстах:
— тематические слова
— специфичные слова документа (шум)
— слова общей лексики (фон)

Подсказка 1: искать распределение ролей слов $p(r|w)$, $r \in \{\text{т}, \text{ш}, \text{ф}\}$.
Подсказка 2: можно разреживать $p(r|w)$ для жёсткого определения ролей.
Подсказка 3: можно использовать документную частоту слов.

Задания к лекции 2

4. Пользуясь основной леммой, докажите, что регуляризатор битермов эквивалентен добавлению псевдодокументов d_u в исходную коллекцию (см. слайд 13)

Прикладная исследовательская задача:
автоматическое выделение научных терминов (ATE)

- Дано:
  коллекция размеченных текстов конкурса ruTermEval;
  неразмеченная коллекция текстов той же тематики
- Найти:
  метод ATE на основе комбинирования ARTM и TopMine;
  обоснование, что синтаксический анализ не нужен;
  зависимость качества ATE от объёма коллекции
- Критерий:
  качество ATE (Prec, Rec, F1) на размеченных данных

Теоретическое задание к лекции 3

Выведете EM-алгоритм для тематической языковой модели:

5. $p(w|d) = \sum_t \phi_{wt}\theta_{td}$, используя в качестве исходных данных последовательность $(d_i, w_i)_{i=1}^n$ вместо счётчиков n_{dw}.
Докажите эквивалентность обычному EM-алгоритму ARTM.

6. $p(w|d) = \sum_t \phi_{tw}\frac{p(w)}{p(t)}\theta_{td}$,
где $\phi_{tw} = p(t|w)$, $\theta_{td} = p(t|d)$ — параметры модели.

7. $p(w|d) = \sum_t \phi_{tw}\frac{p(w)}{p(t)}\theta_{td}$,
где $\phi_{tw} = p(t|w)$ — параметры модели, $\theta_{td} = \sum_w \frac{n_{dw}}{n_d}\phi_{tw}$.

8*. Введение $p(t)$ как внешнего параметра упрощает выкладки, но может нарушать условия целостности модели:

$$p(t) = \sum_t \phi_{tw}p(w), \qquad p(t) = \sum_t \theta_{td}p(d).$$

Как обеспечить выполнение этих условий в EM-алгоритме?

Примеры датасетов для практических заданий по курсу

1. Открытые датасеты (английский): 20NG, NIPS, KOS
2. Ранжированные результаты поиска научных статей (по данным eLibrary, arXiv, PubMed)
3. Научно-популярные статьи: ПостНаука, Элементы, Хабр,...
4. Техноблоги: Хабр (русский), TechCrunch (английский)
5. Данные социальных сетей: VK, Twitter, Telegram,...
6. Статьи по Complexity Sciences (для хронокарты науки)

- Википедия
- Новостной поток (20 источников на русском языке)
- Данные кадровых агентств: резюме + вакансии
- Транзакции клиентов Sberbank DSD 2016
- Акты арбитражных судов РФ

---

http://bigartm.org
http://drive.google.com/drive/folders/1PPnw6aZOJAJoLRYuwdGm437RssV-XQx0

Проекты

- «Тематизатор» для социо-гуманитарных исследований:
  — пользователь задаёт грубый фильтр текстового потока;
  — задача: «классифицировать иголки в стоге сена»,
  — разделив темы на информативные и мусорные,
  — выделив аспекты и тональности в каждой теме;
  — конечная цель: кол./кач. анализ предметной области,
  — реализация данного сценария как модуля в среде Orange

- «Мастерская знаний» для научного поиска:
  — пользователь строит тематические подборки статей,
  — поисковая выдача формируется моделью SciRus;
  — задача: показать пользователю тематику подборки;
  — понадобится: автоматическое выделение терминов,
  — выделение тематических фраз из документов,
  — автоматическое именование и суммаризация тем;
  — конечная цель: помочь в понимании предметной области

Открытые проблемы тематического моделирования

1. Тематические модели внимания последовательного текста
2. Проблема несбалансированности тем в коллекции
3. Измерение интерпретируемости тем (когерентность)
4. Обеспечение 100%-й интерпретируемости тем
5. Автоматическое именование и суммаризация тем
6. Калибровка моделей тематической фильтрации
7. Согласование тем с предобученными эбмедингами LLM
8. Статистические оценки состоятельности тем
9. Обнаружение новых тем или трендов в потоке текстов
10. Обеспечение устойчивости и полноты множества тем
11. Автоматический подбор гиперпараметров, AutoML
12. Гиперграфовые тематические модели для RecSys