Методы машинного обучения. Векторные представления текстов и графов

Воронцов Константин Вячеславович
www.MachineLearning.ru/wiki?title=User:Vokov
вопросы к лектору: voron@forecsys.ru

материалы курса:
github.com/MSU-ML-COURSE/ML-COURSE-21-22
орг.вопросы по курсу: ml.cmc@mail.ru

МФТИ • 5 апреля 2022

Содержание

Дистрибутивная гипотеза и виды семантической близости слов

«Смысл слова определяется множеством его контекстов»

- 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.

Формализация дистрибутивной гипотезы

Дано: текст $(w_1 \dots w_n)$, состоящий из слов словаря W

Найти: векторные представления слов $v_w \in \mathbb{R}^d$, так, чтобы близкие по смыслу слова имели близкие векторы

Модель CBOW (continuous bag-of-words) для вероятности слова w_i в заданном контексте $C_i = (w_{i-k} \dots w_{i-1} w_{i+1} \dots w_{i+k})$:

$$p(w_i = w | C_i) \;=\; \underset{w \in W}{\mathrm{SoftMax}} \langle u_w, v^{-i} \rangle,$$

$v^{-i} = \frac{1}{2k} \sum_{w \in C_i} v_w$ — средний вектор слов из контекста C_i,

v_w — векторы предсказывающих слов,

u_w — вектор предсказываемого слова, в общем случае $u_w \neq v_w$.

Критерий максимума log-правдоподобия, $U, V \in \mathbb{R}^{|W| \times d}$:

$$\sum_{i=1}^{n} \log p(w_i | C_i) \to \max_{U,V}$$

Ещё одна формализация дистрибутивной гипотезы

Дано: текст $(w_1 \dots w_n)$, состоящий из слов словаря W

Найти: векторные представления слов $v_w \in \mathbb{R}^d$, так, чтобы близкие по смыслу слова имели близкие векторы

Модель Skip-gram для предсказания вероятности слов контекста $C_i = (w_{i-k} \dots w_{i-1} w_{i+1} \dots w_{i+k})$ по слову w_i:

$$p(w|w_i) = \underset{w \in W}{\mathrm{SoftMax}} \langle u_w, v_{w_i} \rangle \equiv \underset{w \in W}{\mathrm{norm}} \big(\exp \langle u_w, v_{w_i} \rangle \big),$$

v_w — вектор предсказывающего слова,

u_w — вектор предсказываемого слова, в общем случае $u_w \neq v_w$.

Критерий максимума log-правдоподобия, $U, V \in \mathbb{R}^{|W| \times d}$:

$$\sum_{i=1}^{n} \sum_{w \in C_i} \log p(w|w_i) \to \max_{U,V}$$

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

Сравнение моделей CBOW и Skip-gram

- Различие — в структуре оптимизационного критерия:

$$\text{CBOW:} \quad \sum_{i=1}^{n} \log \underset{w_i \in W}{\text{SoftMax}} \left(\frac{1}{2k} \sum_{c \in C_i} \langle u_{w_i}, v_c \rangle \right) \to \max_{U,V}$$

$$\text{Skip-gram:} \quad \sum_{i=1}^{n} \sum_{c \in C_i} \log \underset{c \in W}{\text{SoftMax}} \langle u_c, v_{w_i} \rangle \to \max_{U,V}$$

- Skip-gram точнее моделирует вероятности редких слов
- Обе модели можно обучать с помощью SGD
- Обе модели реализованы в программе word2vec [Mikolov]
- Два способа обойти трудности с оптимизацией SoftMax:
  — иерархический SoftMax (Hierarchical SoftMax)
  — сведение к задаче двухклассовой классификации (SGNS)

Связь word2vec с матричными разложениями

d — размерность векторов слов v_w и u_w
$V = (v_w)_{W \times d}$ — матрица предсказывающих векторов слов
$U = (u_w)_{W \times d}$ — матрица предсказываемых векторов слов

Модель Skip-gram строит матричное разложение $P \approx UV^{\mathrm{T}}$ матрицы оценок shifted PMI (Point-wise Mutual Information):

$$P_{ab} = \ln \frac{n_{ab} n}{n_a n_b} - \ln k,$$

n_{ab} — частота пары слов a, b в окне $\pm k$ слов,
n_a, n_b — число пар с участием слова a и b соответственно,
n — число всех пар слов в коллекции.

В качестве эвристики используют также Shifted Positive PMI:

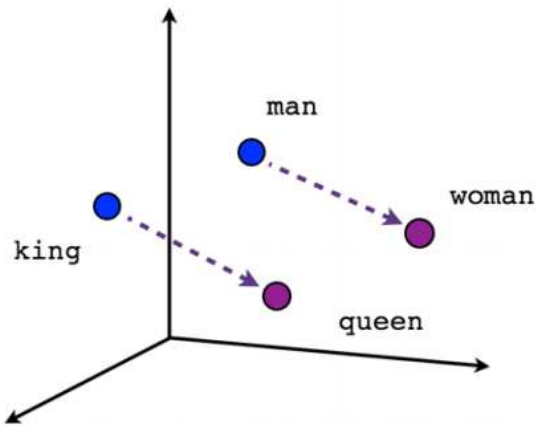
$$P^+_{ab} = \left(\ln \frac{n_{ab} n}{n_a n_b} - \ln k \right)_+ .$$

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

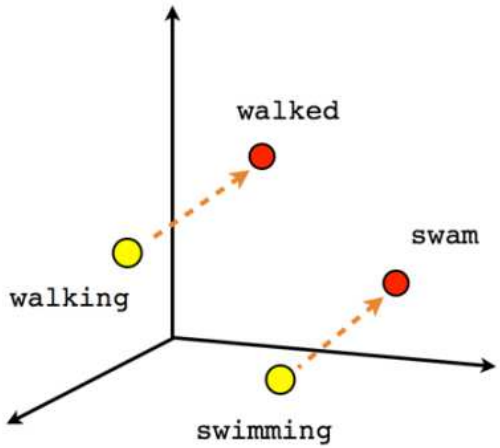
Проверка на задачах семантической близости и аналогии слов

Задача семантической близости слов:
по выборке пар слов (a, b) оценивается корреляция Спирмена между $\cos(v_a, v_b)$ и экспертными оценками близости $y(a, b)$

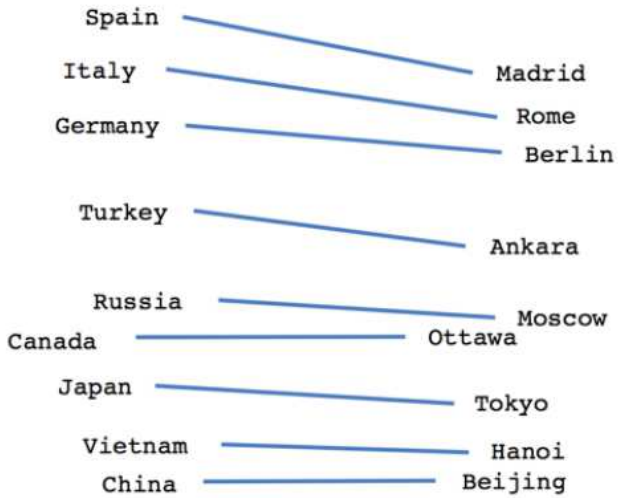
Задача семантической аналогии слов:
по трём словам угадать четвёртое



Male-Female



Verb tense



Country-Capital

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 для 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: эмбединги фраз и предложений 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.

Задачи обработки и преобразования последовательностей

Обработка, тегирование, разметка: $(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n)$

- классификация фрагментов текстовых документов
- автоматическая разметка / тегирование текстов
- анализ тональности документа / предложений / аспектов

Преобразование, генерация, seq2seq: $(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m)$

- машинный перевод (machine translation)
- ответы на вопросы (question answering)
- суммаризация текста (text summarization)
- описание изображений, аудио, видео (multimedia description)
- распознавание речи (speech recognition)
- синтез речи (speech synthesis)

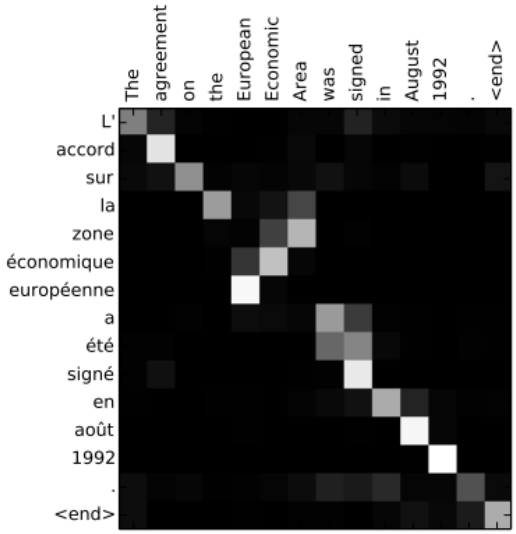
Моделирование внимания в машинном переводе

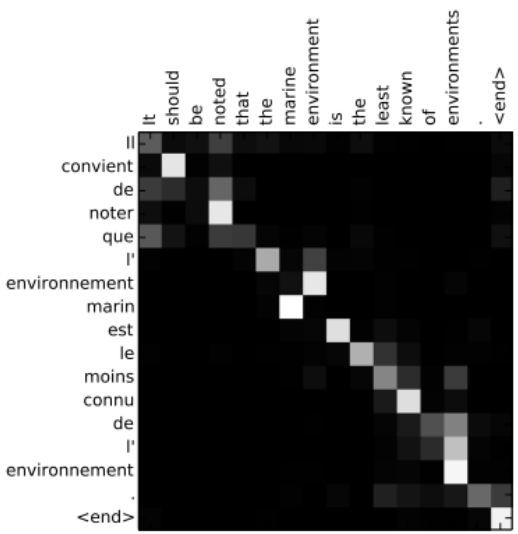
Генерация фразы перевода $(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m)$

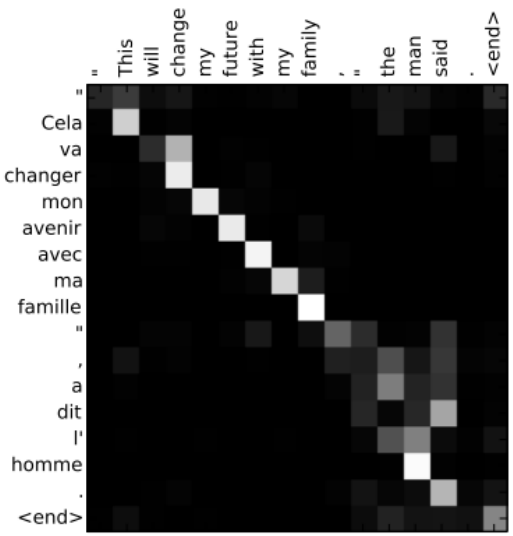
$a(x_i, y_t)$ — оценка семантической близости слов x_i и y_t

$\alpha_{ti} = \underset{i}{\text{norm}}\, a(x_i, y_t)$ — оценка важности (attention score) слова x_i на входе для генерации слова y_{t+1} на выходе, $\sum_i \alpha_{ti} = 1$

Интерпретируемость модели внимания:







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

Моделирование внимания для описания изображений

При генерации каждого слова в описании изображения модель обращает внимание на определённые области изображения:

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

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

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

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

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

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

Модель внимания запрос–ключ–значение (Query–Key–Value)

q — *вектор-запрос*, для которого хотим вычислить контекст
$K = (k_1, \ldots, k_n)$ — *векторы-ключи*, сравниваемые с запросом
$V = (v_i, \ldots, v_n)$ — *векторы-значения*, формирующие контекст
$a(k_i, q)$ — *оценка релевантности* (сходства) ключа k_i запросу q

c — искомый вектор контекста, релевантный запросу

Формула внимания — 3х-слойная нейросеть, вычисляющая выпуклую комбинацию значений v_i, релевантных запросу q:

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

$c_t = \text{Attn}(W_q h'_{t-1}, W_k H, W_v H)$ — контекст для генерации выходного h'_t по входным $H = (h_1, \ldots, h_n)$ в задачах seq2seq

Внутреннее внимание или «самовнимание» (self-attention):
$c_i = \text{Attn}(W_q h_i, W_k H, W_v H)$ — контекст для обработки $h_i \in H$

Разновидности функций сходства векторов

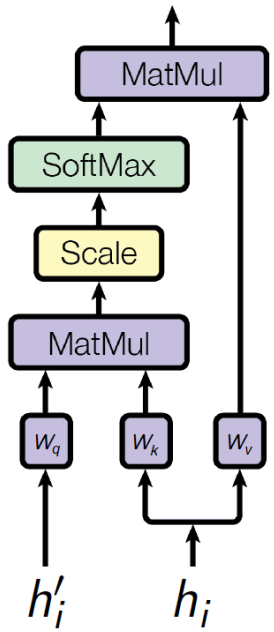
$a(k,q) = k^\mathsf{T} q$ — скалярное произведение
$a(k,q) = \exp(k^\mathsf{T} q)$ — после нормировки получится SoftMax
$a(k,q) = k^\mathsf{T} W q$ — с матрицей обучаемых параметров W
$a(k,q) = w^\mathsf{T} \operatorname{th}(Uk + Vq)$ — аддитивное внимание с w, U, V

Линейные преобразования векторов query, key, value:

$$a(h_i, h'_{t-1}) = (W_k h_i)^\mathsf{T} (W_q h'_{t-1}) / \sqrt{d}$$
$$\alpha_{ti} = \operatorname{SoftMax}_i a(h_i, h'_{t-1})$$
$$c_t = \sum_i \alpha_{ti} W_v h_i$$

W_q $d \times \dim(h')$, W_k $d \times \dim(h)$, W_v $d \times \dim(h)$ — матрицы весов линейных нейронов (обучаемые линейные преобразования в пространство размерности d)

Возможно упрощение модели: $W_k \equiv W_v$



Dichao Hu. An introductory survey on attention mechanisms in NLP problems. 2018.

Многомерное внимание (multi-head attention)

Идея: J разных моделей внимания совместно обучаются выделять различные аспекты входной информации (например, части речи, синтаксис, фразеологизмы):

$$c^j = \text{Attn}(W_q^j q, W_k^j H, W_v^j H), \quad j = 1, \ldots, J$$

Варианты агрегирования выходного вектора:

$c = \frac{1}{J}\sum_{j=1}^{J} c^j$ — усреднение

$c = \left[c^1 \cdots c^J\right]$ — конкатенация

$c = \left[c^1 \cdots c^J\right] W$ — чтобы вернуться к нужной размерности

Регуляризация: чтобы аспекты внимания были максимально различны, строки $J \times n$ матриц A, $\alpha_{ji} = \text{SoftMax}_i\, a(W_k^j h_i, W_q^j q)$, декоррелируются $(\alpha_s^\intercal \alpha_j \to 0)$ и разреживаются $(\alpha_j^\intercal \alpha_j \to 1)$:

$$\|AA^\intercal - I\|^2 \to \min_{\{W_k^j, W_q^j\}}$$

Zhouhan Lin, Y.Bengio et al. A structured self-attentive sentence embedding. 2017.

BERT (Bidirectional Encoder Representations from Transformers)

Трансформер — это глубокая нейросеть, суперпозиция моделей самовнимания, контекстно-зависимая векторизация слов, обучаемая для решения широкого класса задач NLP

Схема преобразования данных в задачах NLP:

- $S = (w_1, \ldots, w_n)$ — токены предложения входного текста
  $\downarrow$ обучение эмбедингов вместе с трансформером
- $X = (x_1, \ldots, x_n)$ — эмбединги токенов входного предложения
  $\downarrow$ трансформер-кодировщик
- $Z = (z_1, \ldots, z_n)$ — 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.

Архитектура трансформера-кодировщика

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

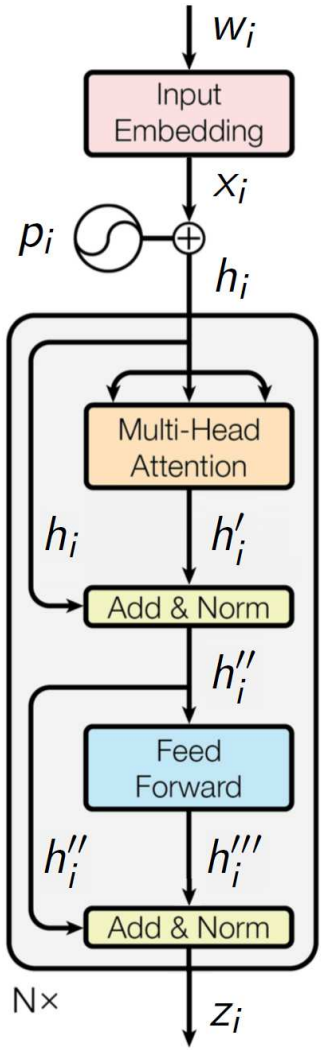
2. Многомерное самовнимание:
$h_i^j = \mathrm{Attn}(W_q^j h_i, W_k^j H, W_v^j H)$
$j = 1, \ldots, J = 8$
$\dim h_i^j = 64$
$\dim W_q^j, W_k^j, W_v^j = 64 \times 512$

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

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

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

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



Несколько дополнений и замечаний

- $N = 6$ блоков $h_i \to \square \to z_i$ соединяются последовательно
- эмбединги x_i — пред-обученные или обучаемые
- 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

- кодирование позиций (positional encoding) векторами p_i, $i = 1, \ldots, n$; чем больше $|i - j|$, тем больше $\|p_i - p_j\|$:

$$c_j = \mathrm{Attn}(q_j, K, V) = \sum_i (v_i + p^v_{i \boxminus j}) \,\mathrm{SoftMax}_i\, a(k_i + p^k_{i \boxminus j}, q_j),$$

усечённая разность $i \boxminus j = \max(\min(i - j, \delta), -\delta)$, $\delta = 5..16$

Shaw, Uszkoreit, Vaswani. Self-attention with relative position representations. 2018.

Критерий MLM (masked language modeling) для обучения BERT

Языковая модель, предсказывающая i-й токен предложения S:

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

$z_i(S,W_T)$ — контекстный эмбединг i-го токена предложения S на выходе Трансформера с параметрами W_T,
W — все параметры Трансформера и языковой модели.

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 маскированных токенов из S.

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.

Трасформер для машинного перевода

Схема преобразований данных в машинном переводе:

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

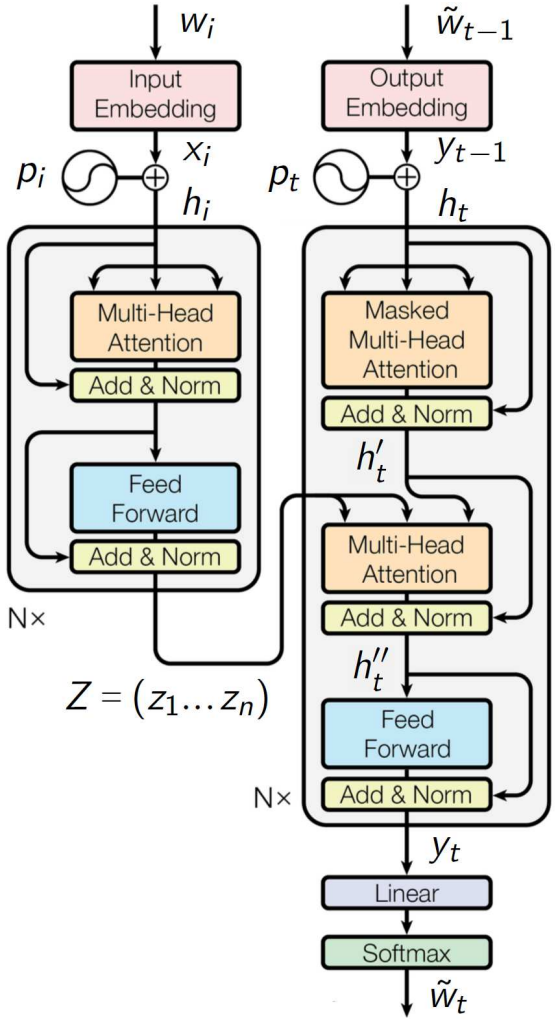
Vaswani et al. (Google) Attention is all you need. 2017.

Архитектура трансформера декодировщика

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

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

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



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

Критерии обучения и валидации для машинного перевода

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

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

Критерии 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.

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, в IsoMAP R_{ij} — длина кратчайшего пути по графу.

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

Критерий стресса (stress):

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

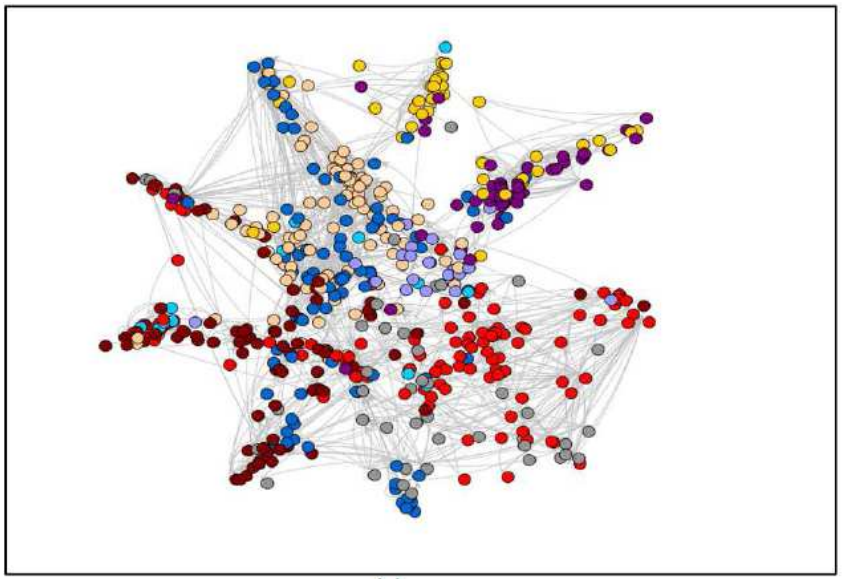
где $\rho(z_i,z_j) = \|z_i - z_j\|$ — обычно евклидово расстояние,
$w(R_{ij})$ — веса (какие расстояния важнее, большие или малые).

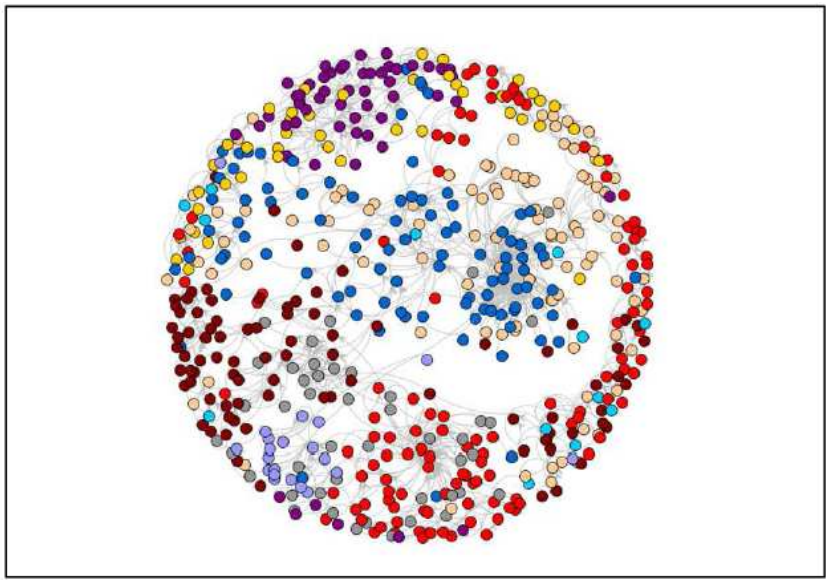
Обычно решается методом стохастического градиента (SG).

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
- Недостаток — искажения неизбежны
- Наиболее популярная разновидность метода — t-SNE

Laurens van der Maaten, Geoffrey Hinton. Visualizing data using t-SNE. 2008

Метод векторного представления соседства (Stochastic Neighbor Embedding, SNE)

Дано: исходные точки $x_i \in \mathbb{R}^n$, $i = 1, \ldots, \ell$

Найти: точки на карте-проекции $z_i \in \mathbb{R}^d$, $i = 1, \ldots, \ell$, $d \ll n$

Критерий: расстояния $\|z_i - z_j\|$ близки к исходным $\|x_i - x_j\|$

Вероятностная модель события «j является соседом i»
на основе перенормированных гауссовских распределений:

$p(j|i) = \underset{j \neq i}{\operatorname{norm}} \exp\left(-\frac{1}{2\sigma_i^2}\|x_i - x_j\|^2\right)$ — в исходном пространстве;

$q(j|i) = \underset{j \neq i}{\operatorname{norm}} \exp\left(-\|z_i - z_j\|^2\right)$ — в пространстве проекции;

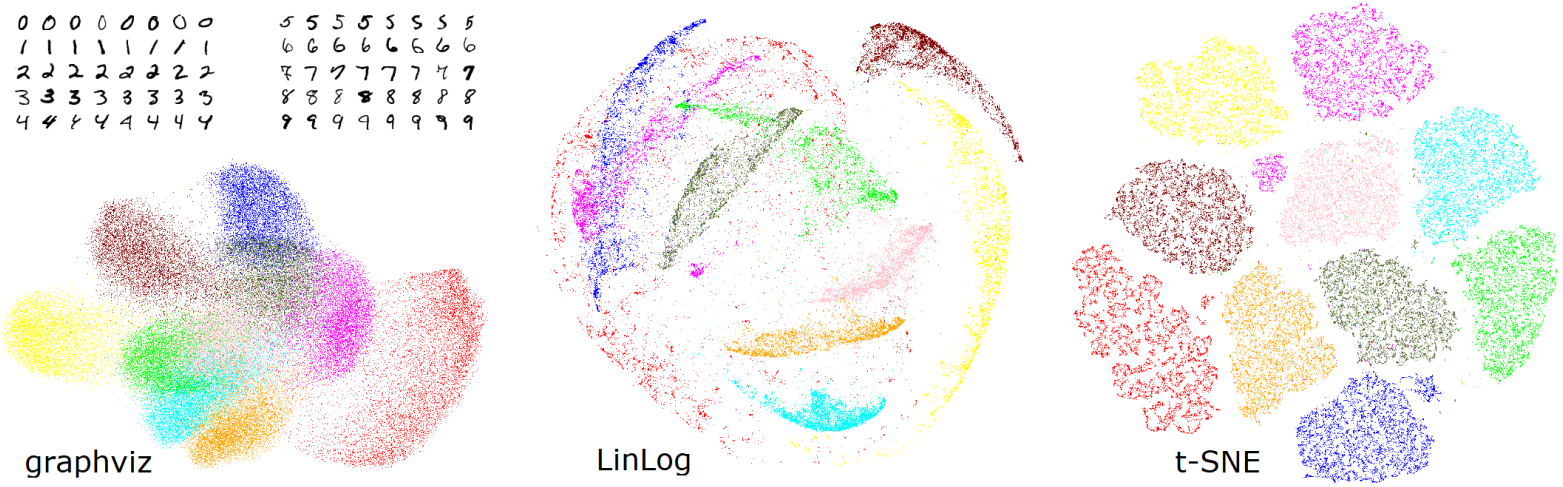
где $p(j) = \underset{j}{\operatorname{norm}}(z_j) = \frac{z_j}{\sum_k z_k}$ — операция нормировки вектора.

Максимизация правдоподобия (стохастическим градиентом):

$$\sum_i \sum_{j \neq i} p(j|i) \ln q(j|i) \;\to\; \max_{\{z_i\}}$$

Преимущества и недостатки t-SNE

Лучшее представление структур сходства по сравнению с другими методами многомерного шкалирования (mnist)



Ложные кластерные структуры при низкой перплексии

Размеры кластеров и расстояния между ними неинформативны

Трудно отличить реальные структуры от артефактов метода

M.Wattenberg, F.Viegas, I.Johnson (Google). How to use t-SNE effectively. 2016. https://distill.pub/2016/misread-tsne

Матричные разложения (graph factorization)

Дано: $(i,j) \in E$ — выборка рёбер графа $\langle V, E \rangle$,
S_{ij} — близость между вершинами ребра (i,j).
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):

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

Критерий для ориентированного графа (S несимметрична):

$$\sum_{(i,j)\in E} \big(\langle \varphi_i, \theta_j \rangle - S_{ij}\big)^2 \to \min_{\Phi,\Theta}, \quad \Phi, \Theta \in \mathbb{R}^{V\times d}$$

Обычно решается методом стохастического градиента (SG).

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

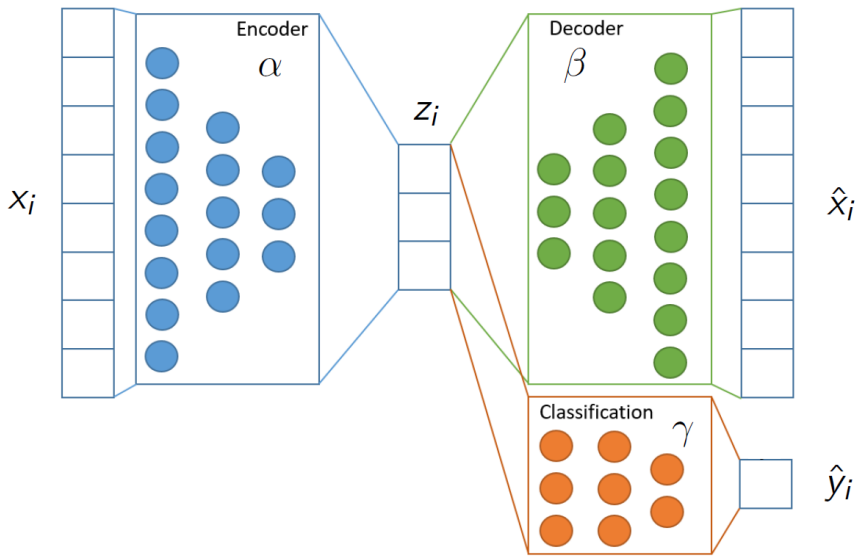
Напоминание. Автокодировщики для обучения с учителем

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

Совместное обучение кодировщика, декодировщика и предсказательной модели (классификации, регрессии или др.):

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

$z_i = f(x_i,\alpha)$ — кодировщик
$\hat{x}_i = g(z_i,\beta)$ — декодировщик
$\hat{y}_i = \hat{y}(z_i,\gamma)$ — классификатор



Функции потерь:
$\mathscr{L}(\hat{x}_i, x_i)$ — реконструкция
$\tilde{\mathscr{L}}(\hat{y}_i, y_i)$ — предсказание

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

Векторные представления графов как автокодировщики

Все рассмотренные выше методы векторных представлений графов суть автокодировщики данных о рёбрах:

- многомерное шкалирование: $R_{ij} \to \|z_i - z_j\|$
- SNE и t-SNE: $p(i,j) \to q(i,j) \propto K(\|z_i - z_j\|)$
- матричные разложения: $S_{ij} \to \langle \varphi_i, \theta_j \rangle$

Вход кодировщика:

- W_{ij} — данные о ребре графа (i,j)

Выход кодировщика:

- векторные представления вершин z_i

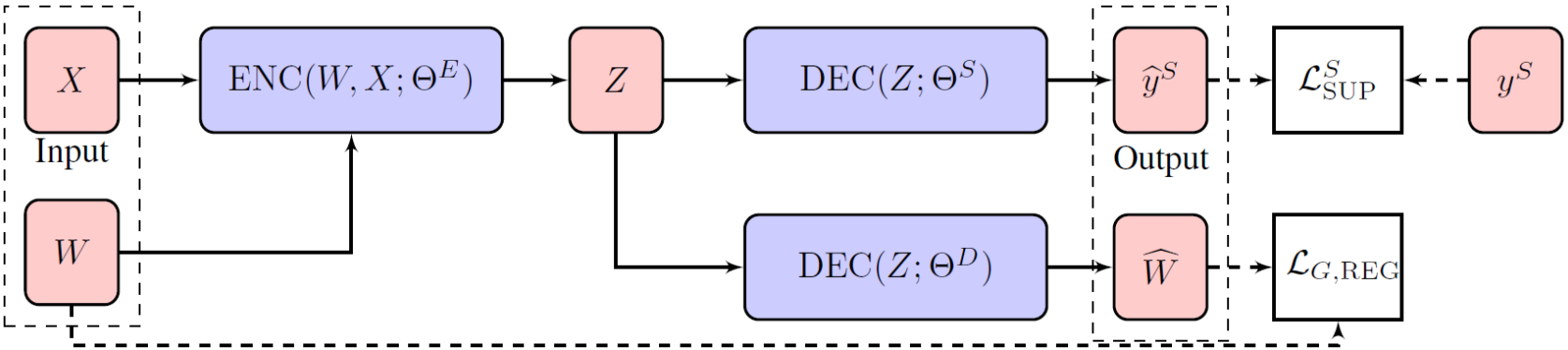
Выход декодировщика:

- аппроксимация $\hat{W}_{ij}$, вычисляемая по (z_i, z_j)

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
$Z \in \mathbb{R}^{V \times d}$ — 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 — (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.

Резюме

- Синтез векторных представлений (эмбедингов) — это
  — *обучение представлений* (Representation Learning)
  — *генерация признаков* (Feature Generation)
  — векторизация сложно структурированных данных
  — построение латентных векторных представлений вершин графа по эмпирическим данным о его рёбрах
- Многокритериальная оптимизация эмбедингов — обучение на нескольких задачах одновременно (multi-task learning)
  — качество реконструкции объекта по эмбедингу
  — качество предсказательной модели
- Эмбединги графов обобщают многие задачи векторизации текстов, дискретных сигналов, изображений и др.
- Модель внимания и трансформеры — наиболее прогрессивные модели для обработки естественного языка (BERT, GPT-2/3, XLNet, ELECTRA и др.)