

# Методы машинного обучения. Обучение без учителя: векторизация графовых и текстовых данных

Воронцов Константин Вячеславович

[www.MachineLearning.ru/wiki?title=User:Vokov](http://www.MachineLearning.ru/wiki?title=User:Vokov)

вопросы к лектору: [k.vorontsov@iai.msu.ru](mailto:k.vorontsov@iai.msu.ru)

материалы курса:

[github.com/MSU-ML-COURSE/ML-COURSE-24-25](https://github.com/MSU-ML-COURSE/ML-COURSE-24-25)

орг.вопросы по курсу: [ml.cmc@mail.ru](mailto:ml.cmc@mail.ru)

## Выявление структуры данных на основе сходства:

- кластеризация (clustering) и квантизация (quantization)
- оценивание плотности распределения (density estimation)
- одноклассовая классификация (anomaly detection)

## Преобразование признакового пространства:

- автокодировщики (autoencoders)
- метод главных компонент (principal components analysis)
- многомерное шкалирование (multidimensional scaling)
- разложения графов (graph factorization)
- матричные разложения (matrix factorization)

## Поиск взаимосвязей в данных или синтез учителя:

- частичное обучение (semi-supervised learning)
- поиск ассоциативных правил (association rule learning)
- самостоятельное обучение (self-supervised learning)

## 1 Многомерное шкалирование

- Постановка задачи многомерного шкалирования
- Визуализация данных
- Векторизация отношения соседства

## 2 Векторные представления графов

- Модели матричных разложений
- Модель случайных блужданий
- Графовые автокодировщики

## 3 Векторные представления текста

- Гипотеза дистрибутивной семантики
- Модели word2vec
- Модель FastText

## 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):

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

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

$R_{ij}^{\gamma}$  — веса, важнее расстояния большие ( $\gamma > 0$ ) или малые ( $\gamma < 0$ )

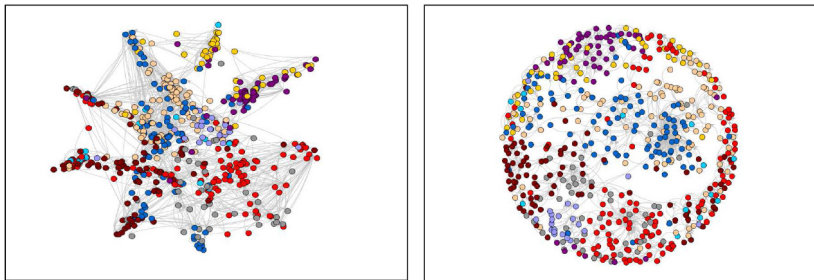
Обычно решается методом стохастического градиента (SG)

---

*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 модификация MDS — 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  $x_i \in \mathbb{R}^n$ ,  $i = 1, \dots, \ell$

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

**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:

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

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

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

Максимизация правдоподобия (стохастическим градиентом):

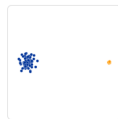
$$\sum_i \sum_{j \neq i} p(j|i) \ln q(j|i) \rightarrow \max_{\{z_i\}}$$

## Преимущества метода SNE

- Преобразование расстояний в вероятности устраняет дисбалансы между большими и малыми расстояниями
- Дисбаланс между точками с большой и малой плотностью соседей выравнивается настройкой  $\sigma_i$  по перплексии

$H(i) = -\sum_j p(j|i) \log_2 p(j|i)$  — энтропия распределения  $p(j|i)$ ;  
 $2^{H(i)}$  это перплексия = «эффективное число соседей у  $x_i$ »  
 (если  $p(j|i) = \frac{1}{k}$ , то  $2^{H(i)} = k$ ); обычно перплексия = 5..50.

Выбор перплексии может существенно влиять на вид проекции:



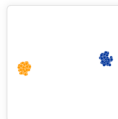
Original



Perplexity: 2



Perplexity: 5



Perplexity: 30

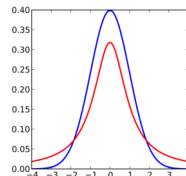
G.E.Hinton, S.T.Roweis. Stochastic Neighbor Embedding. 2002.

## Вероятностная модель t-SNE: два усовершенствования SNE

**Проблема скученности в SNE:** окрестность вмещает гораздо больше точек в  $n$ -мерном пространстве, чем в  $d$ -мерном

- Использование  $t$ -распределения Стьюдента с более тяжёлым хвостом и симметричного совместного распределения  $q(i, j)$ :

$$q(i, j) = \text{norm}_{(i, j): i \neq j} (1 + \|z_i - z_j\|^2)^{-1}$$



- Использование совместного распределения  $p(i, j)$ :

$$p(i, j) = \frac{1}{2\ell} (p(j|i) + p(i|j))$$

Максимизация правдоподобия (стохастическим градиентом):

$$\sum_{(i, j): i \neq j} p(i, j) \ln q(i, j) \rightarrow \max_{\{z_i\}}$$

*L.J.P. van der Maaten, G.Hinton. Visualizing data using t-SNE. 2008*

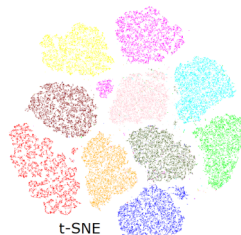
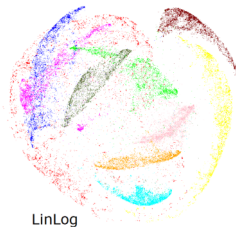


## Преимущества и недостатки t-SNE

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

0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4

5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9



Ложные кластерные структуры при низкой перплексии  
Размеры кластеров и расстояния между ними неинформативны  
Трудно отличить реальные структуры от артефактов метода  
Нет ясного критерия качества для подбора перплексии

---

*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):

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

**Критерий** для ориентированного графа ( $S$  несимметрична):

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

Обычно решается методом стохастического градиента (SG).

## Модель случайных блужданий для больших графов

Дано:  $(i, j) \in E$  — выборка рёбер графа  $\langle V, E \rangle$

$S_{ij} = [j \text{ достижима из } i \text{ коротким путём}]$

$$\sum_{i \in V} \left( \sum_{j \in C_i} \log \sigma(\langle \varphi_i, \theta_j \rangle) + \sum_{j \in \bar{C}_i} \log \sigma(-\langle \varphi_i, \theta_j \rangle) \right) \rightarrow \max_{\Phi, \Theta}$$

$C_i$  — окрестность («контекст») вершины  $i$ , сэмплируемая случайным блужданием длины  $k$  (DeepWalk, node2vec),

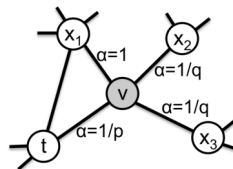
$\bar{C}_i$  — вершины, далёкие от  $i$ , сэмплируемые  $j \sim p(j)^{3/4}$

**Параметризация случайных блужданий:**

вероятность  $p(v \rightarrow w)$  после перехода  $t \rightarrow v$

$p \downarrow \quad q \uparrow$  — ближе к поиску в ширину (BFS)

$p \uparrow \quad q \downarrow$  — ближе к поиску в глубину (DFS)



*B. Perozzi et al.* DeepWalk: online learning of social representations. SIGKDD-2014.

*A. Grover, J. Leskovec.* Node2vec: scalable feature learning for networks. SIGKDD-2016.

## Векторные представления графов как автокодировщики

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

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

**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  $(z_i, z_j)$

## Напоминание. Автокодировщики для частичного обучения

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

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

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

$z_i = f(x_i, \alpha)$  — кодировщик

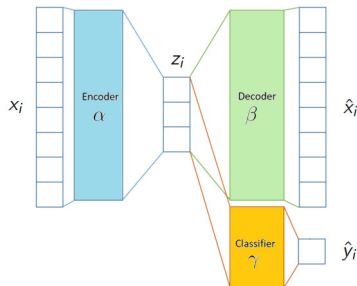
$\hat{x}_i = g(z_i, \beta)$  — декодировщик

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

**Функции потерь:**

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

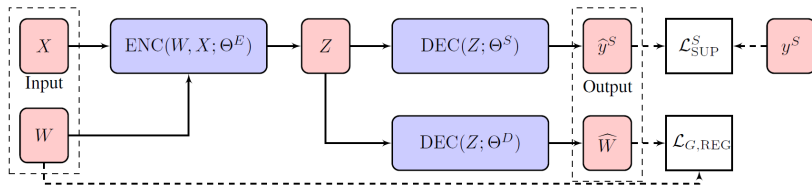
$\tilde{\mathcal{L}}(\hat{y}_i, y_i)$  — предсказание



Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders. 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

$DEC(Z; \Theta^D)$  — декодер, реконструирующий данные о рёбрах

$DEC(Z; \Theta^S)$  — 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.*

## 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  $(w_1 \dots w_n)$ , последовательность 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  $w_i$ :

$$p(w|w_i) = \underset{w \in W}{\text{SoftMax}} \langle u_w, v_{w_i} \rangle$$

$v_w$  — вектор предсказывающего слова,

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

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

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

---

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



## Ещё одна формализация дистрибутивной гипотезы

**Дано:** текст  $(w_1 \dots w_n)$ , последовательность 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  $C_i = (w_{i-k} \dots w_{i-1} w_{i+1} \dots w_{i+k})$ :

$$p(w_i = w | C_i) = \text{SoftMax}_{w \in W} \langle u_w, v[C_i] \rangle,$$

$v[C_i] = \frac{1}{|C_i|} \sum_{w \in C_i} v_w$  — 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,  $U, V \in \mathbb{R}^{|W| \times d}$ :

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

## Сравнение моделей CBOW и Skip-gram

- Различие — в структуре оптимизационного критерия:

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

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

- Skip-gram точнее моделирует вероятности редких слов
- Обе модели можно обучать с помощью SGD
- Обе модели реализованы в программе word2vec [Mikolov]
- Оба критерия трудно оптимизировать из-за SoftMax
- Что делать? Заменять либо SoftMax, либо критерий

---

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

## Иерархический SoftMax

**Идея:** заменить SoftMax на другую функцию потерь, сложность вычисления которой  $O(\log |W|)$  вместо  $O(|W|)$ .

**Предварительный этап:**

- По словарю частот строится *бинарное дерево Хаффмана* — каждому слову  $w \in W$  соответствует ровно один лист — путь от корня до  $w$  тем короче, чем выше частота  $w$
- Каждая внутренняя вершина  $n$  хранит вектор  $u_n \in \mathbb{R}^d$
- Каждый лист хранит вектор  $v_w \in \mathbb{R}^d$  для слова  $w$
- Модель перехода из внутренней вершины  $n$  в дочернюю:

$$\text{направо: } p(+1|n, w) = \sigma(\langle u_n, v_w \rangle)$$

$$\text{налево: } p(-1|n, w) = \sigma(-\langle u_n, v_w \rangle) = 1 - p(+1|n, w)$$

**Обучаются** векторы  $v_w$  в листьях и  $u_n$  во внутренних вершинах

---

*F.Morin, Y.Bengio.* Hierarchical probabilistic neural network language model. 2005.  
*A.Mnih, G.E.Hinton.* A scalable hierarchical distributed language model. 2009.

## Иерархический SoftMax: обучение модели

Модель  $p(w|w_i)$ , гарантирующая нормировку  $\sum_w p(w|w_i) = 1$ :

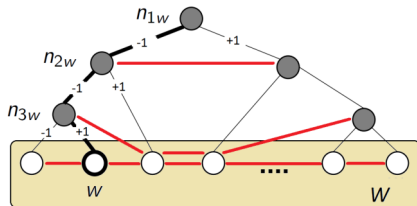
$$p(w|w_i) = \prod_{j=1}^{\ell(w)} p(\beta_{jw}|n_{jw}, w_i) = \prod_{j=1}^{\ell(w)} \sigma(\beta_{jw} \langle u_{n_{jw}}, v_{w_i} \rangle)$$

где  $\ell(w)$  — длина пути от корня дерева к листу  $w$

$n_{jw}$  —  $j$ -я внутренняя вершина на пути к листу  $w$

$\beta_{jw} \in \{-1, +1\}$  — переход из  $n_{jw}$  в дочернюю на пути к  $w$

Пример:  $p(w|w_i) = p(-1|n_{1w}, w_i) p(-1|n_{2w}, w_i) p(+1|n_{3w}, w_i)$



На каждом уровне дерева  
сохраняется нормировка  
 $\sum_w p(w|w_i) = 1$

## Третья формализация: классификация пар слов на два класса

Критерий log-loss для SGNS (Skip-gram Negative Sampling):

$$\sum_{i=1}^n \sum_{w \in C_i} \left( \log p(+1|w, w_i) + \log p(-1|\bar{w}, w_i) \right) \rightarrow \max_{U, V}$$

где  $p(y|w, w_i) = \sigma(y \langle u_w, v_{w_i} \rangle)$  — модель классификации,  $y = \pm 1$ ;  
 $y = +1$ , если пара слов  $(w, w_i)$  находится в общем контексте;  
 $y = -1$ , если пара слов  $(w, w_i)$  не находится в общем контексте;  
 $\bar{w} \sim p(w)^{3/4}$  сэмплируется из  $W \setminus C_i$  в методе SG.

**Эвристики и прочие замечания:**

- Dynamic window: случайный выбор  $k \sim [3..10]$
- Итоговые векторы слов:  $\alpha v_w + (1 - \alpha) u_w$
- Приём NS применяют, когда не хватает второго класса
- Что делать со словами, которые встречаются впервые?

## 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

SGNS строит матричное разложение  $P \approx UV^T$  матрицы

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.

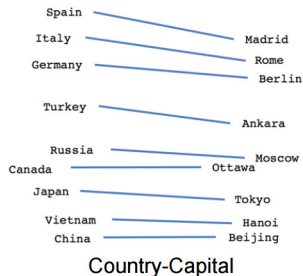
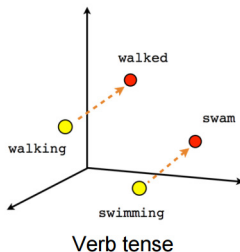
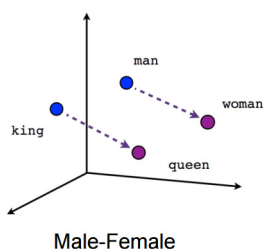
## Проверка на задачах семантической близости и аналогии слов

### 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



## 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:**  $G(\text{дармолюб}) = \{\langle \text{да, арм, рмо, мол, олю, люб, юб} \rangle\}$

**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.Leskevce.* 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.

## Разновидности векторизации данных:

- *Автокодировщики* — синтез векторных представлений (эмбедингов) объектов, обычно с понижением размерности
- *Матричные разложения графовых данных* ( $GF$ ,  $graph2vec$ ) и *многомерное шкалирование* ( $MDS$ ,  $SNE$ ) — синтез эмбедингов объектов по данным об их взаимодействии
- *Обобщённая модель GraphEDM* — синтез эмбедингов объектов по данным об объектах и их взаимодействии

## Методы обучения — на основе SG

### Текстовые данные — частный случай графовых данных:

- слова — вершины графа, сочетаемость пары слов — ребро
- слова и документы — вершины двух разных долей графа, вхождение слова в документ — ребро двудольного графа