



ТЕХНОСФЕРА

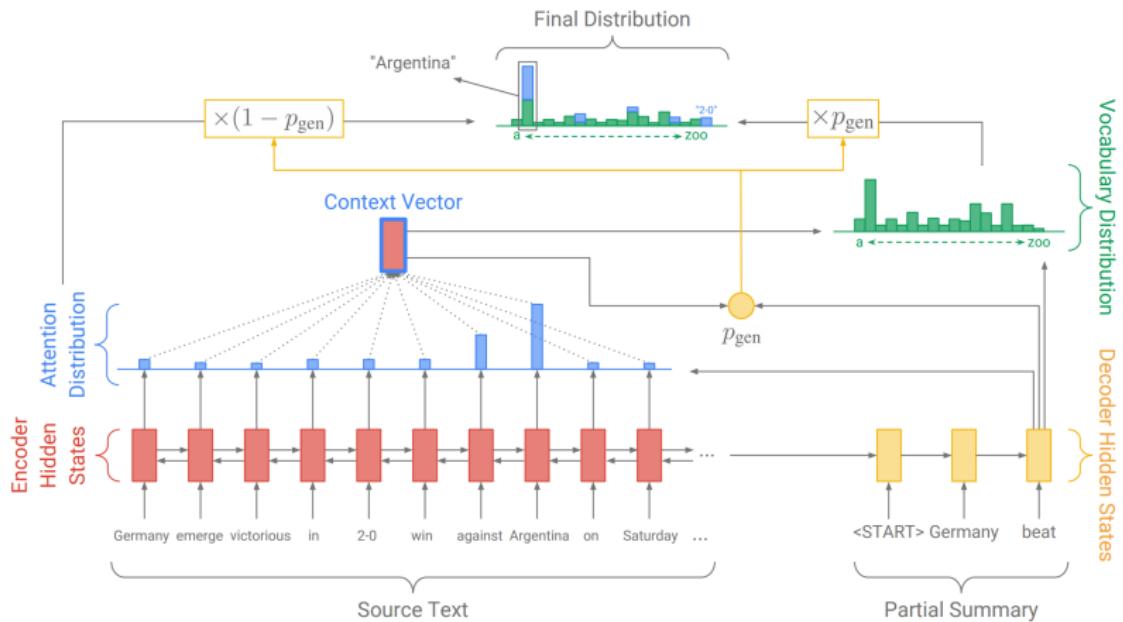
Лекция 8 Нейронные сети для снижения размерности

Байгушев Данила

9 декабря 2019 г.

Обещанная часть про обработку языка

Attention



Attention to images

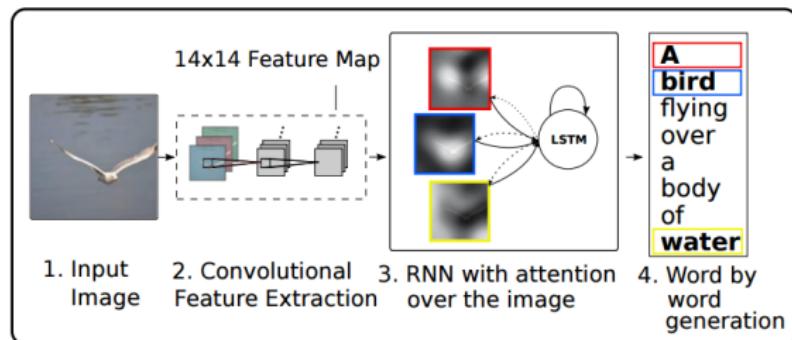
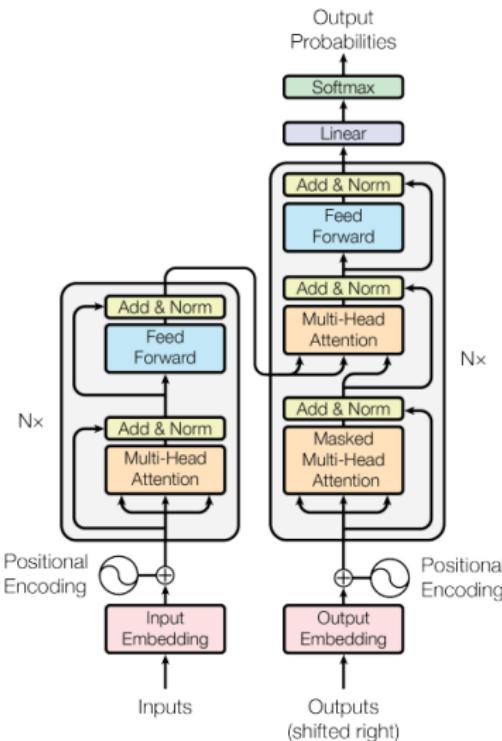


Figure 3. Examples of attending to the correct object (white indicates the attended regions, *underlines* indicated the corresponding word)

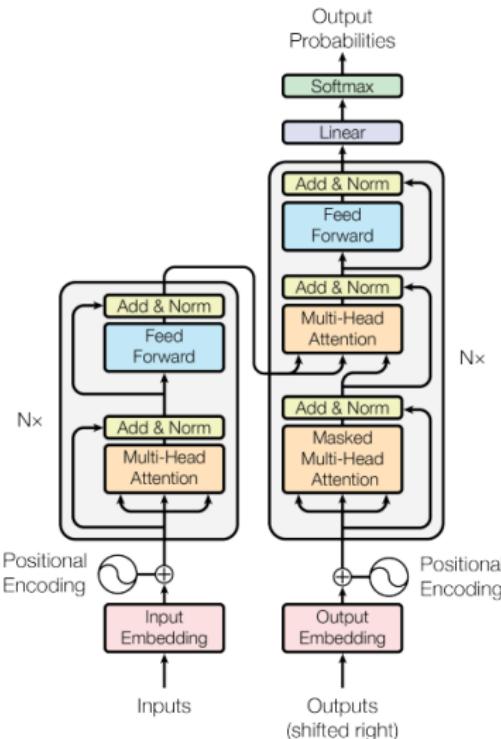


Transformer¹

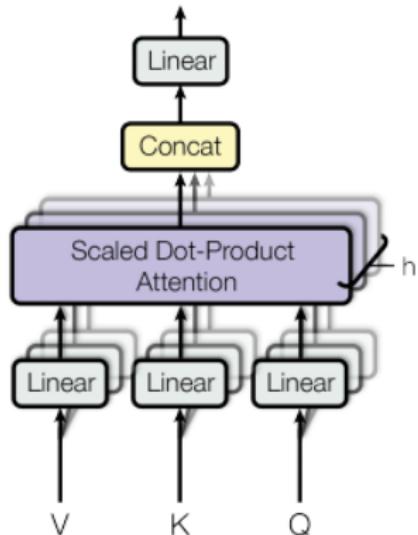


¹<https://arxiv.org/abs/1706.03762>

Transformer¹



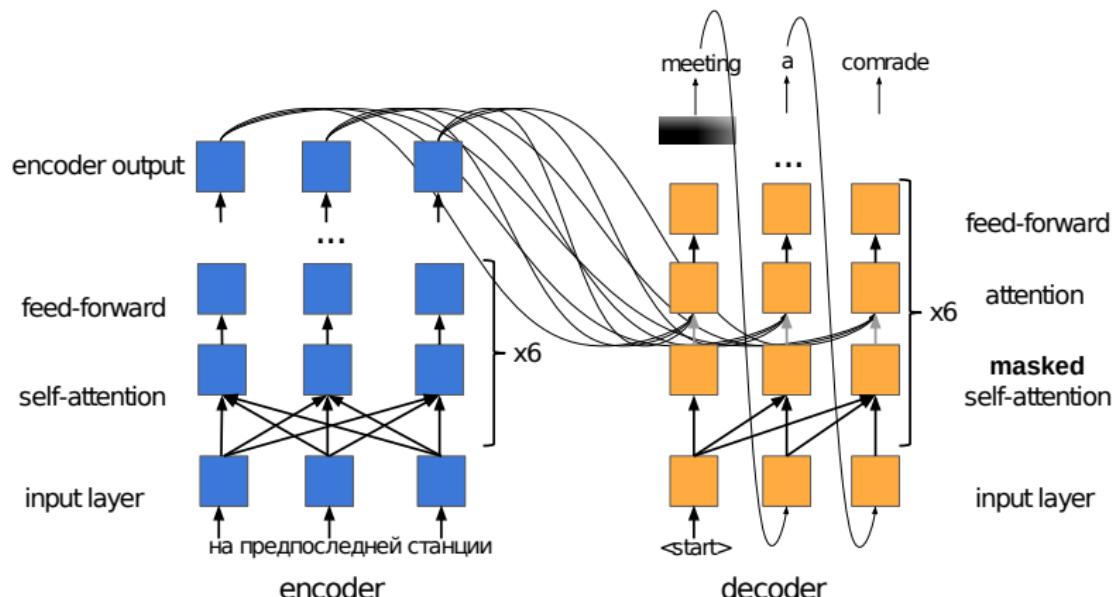
Multi-Head Attention



¹<https://arxiv.org/abs/1706.03762>

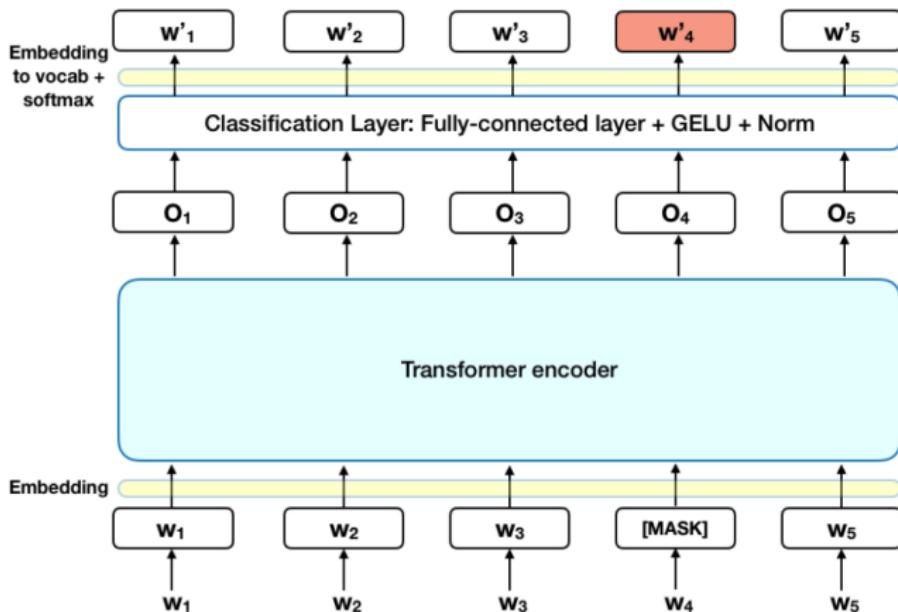
Transformer

Визуализация (gif)



BERT²

Bidirectional Encoder Representations from Transformers



²<https://arxiv.org/abs/1810.04805>

BERT

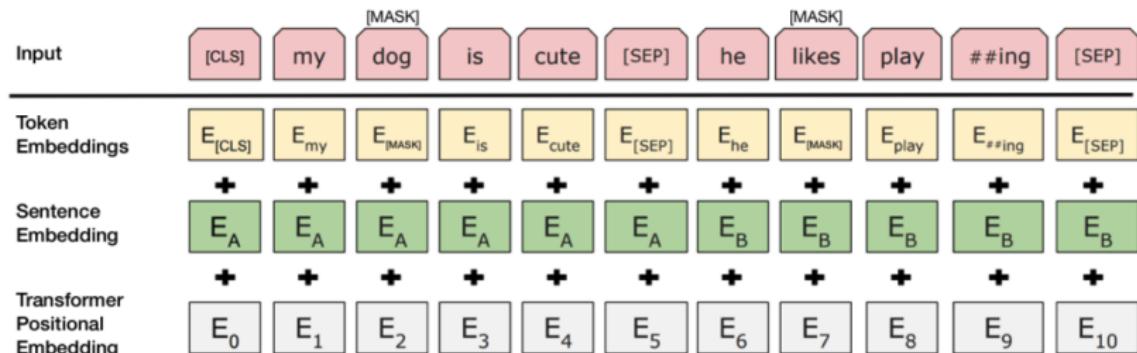


Figure: Предсказание следующего предложения

	Training Compute + Time	Usage Compute
BERT _{BASE}	4 Cloud TPUs, 4 days	1 GPU
BERT _{LARGE}	16 Cloud TPUs, 4 days	1 TPU

Обратно к задаче снижения размерности

Задача снижения размерности

Dimensionality reduction

Дано. N обучающих D -мерных объектов $\mathbf{x}_i \in \mathcal{X}$, образующих тренировочный набор данных (training data set) \mathbf{X} .

Найти. Найти преобразование $A : \mathcal{X} \rightarrow \mathcal{P}$, $\dim(\mathcal{P}) = d < D$, сохранив при этом большую часть “полезной информации” об \mathcal{X} .

Задача снижения размерности

Dimensionality reduction

Дано. N обучающих D -мерных объектов $\mathbf{x}_i \in \mathcal{X}$, образующих тренировочный набор данных (training data set) \mathbf{X} .

Найти. Найти преобразование $A : \mathcal{X} \rightarrow \mathcal{P}$, $\dim(\mathcal{P}) = d < D$, сохранив при этом большую часть “полезной информации” об \mathcal{X} .
Применение:

- ▶ Визуализация в 2D или 3D (поиск структуры и закономерностей)
- ▶ Уменьшение затрат на ресурсы (память, время)
- ▶ Снижение уровня шума в данных

Снижение размерности, примеры

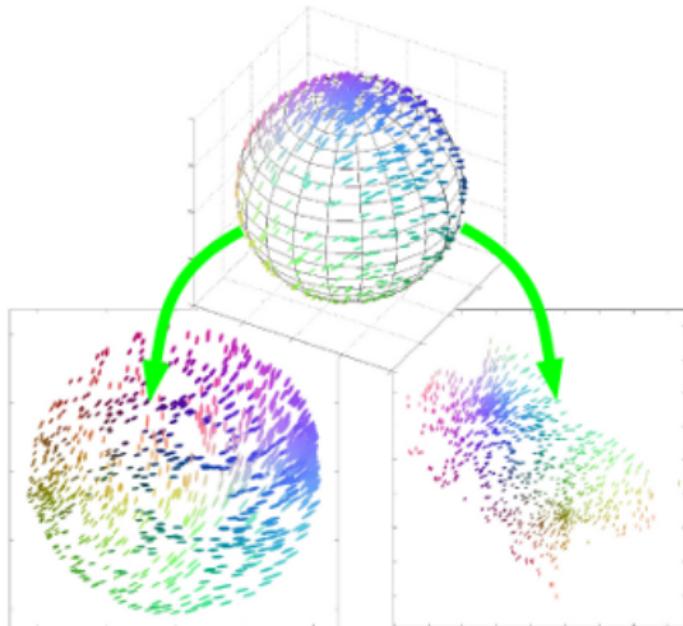


Figure: Примеры данных³

³[stats.stackexchange.com/questions/56589/
visualizing-high-dimensional-data](https://stats.stackexchange.com/questions/56589/visualizing-high-dimensional-data)

Снижение размерности, примеры

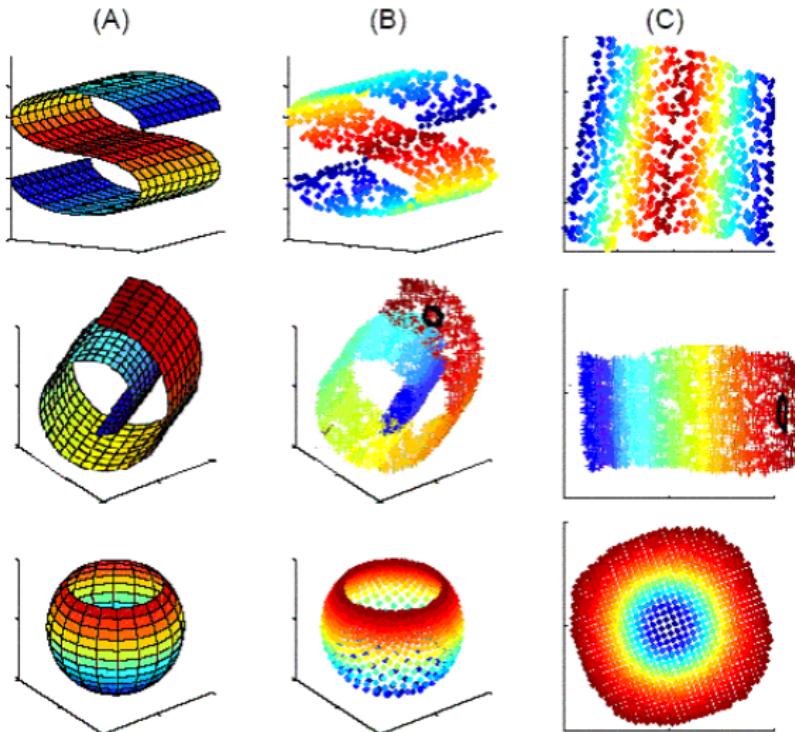


Figure: Примеры данных⁴

⁴<http://jntsai.blogspot.ru/2015/04/ammai-nonlinear-dimensionality.html>

Подходы к снижению размерности

Сохранение расстояний

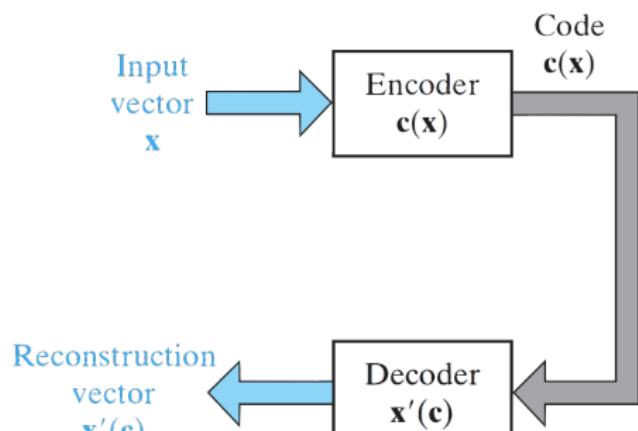
$$\sum_{i,j=1}^N (\|X_i - X_j\|_X - \|y_i - y_j\|_P)^2 \rightarrow \min$$

На этом принципе основаны MDS (евклидово расстояние) и Isomap (топологическое расстояние)

Сохранение содержания (точность восстановления)

$$\sum_{i=1}^N \|X_i - X'_i\|^2 \rightarrow \min$$

На этом принципе построены Автокодировщики, SOM, PCA



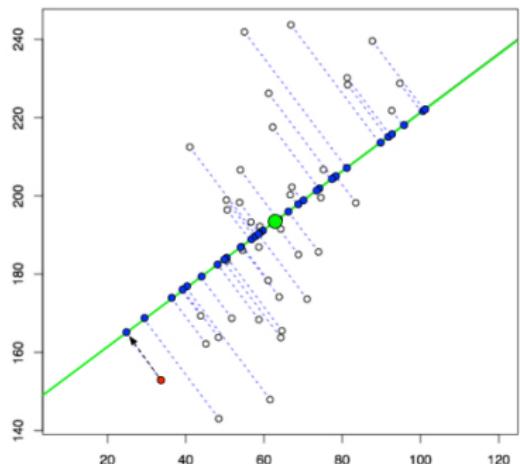
Метод главных компонент

PCA

Постановка задачи

Требуется найти гиперплоскость, задаваемую векторами v_1, v_2, \dots, v_d , минимизирующую суммарное расстояние объектов до плоскости:

$$\sum_{n=1}^N \|h_n\|^2 \rightarrow \min_{v_1, v_2, \dots, v_d}$$



Решение задачи

План доказательства

1. Переход к задаче максимизации

$$\begin{aligned} \|x\|^2 &= \|h\|^2 + \|p\|^2 \Rightarrow \|h\|^2 \rightarrow \min \Leftrightarrow \|p\|^2 \rightarrow \max \\ \sum_{n=1}^N \|h_n\|^2 &\rightarrow \min_{v_1, v_2, \dots, v_d} \Leftrightarrow \sum_{n=1}^N \|p_n\|^2 \rightarrow \max_{v_1, v_2, \dots, v_d} \end{aligned}$$

Решение задачи

План доказательства

1. Переход к задаче максимизации

$$\begin{aligned} \|x\|^2 &= \|h\|^2 + \|p\|^2 \Rightarrow \|h\|^2 \rightarrow \min \Leftrightarrow \|p\|^2 \rightarrow \max \\ \sum_{n=1}^N \|h_n\|^2 &\rightarrow \min_{v_1, v_2, \dots, v_d} \Leftrightarrow \sum_{n=1}^N \|p_n\|^2 \rightarrow \max_{v_1, v_2, \dots, v_d} \end{aligned}$$

2. Жадное построение векторов

$$\begin{cases} \|Xv_k\|^2 \rightarrow \max_{v_k} \\ \|v_k\|^2 = 1 \\ \langle v_i, v_k \rangle = 0 \quad i = \overline{1, k-1} \end{cases}$$

Решение задачи

План доказательства

1. Переход к задаче максимизации

$$\|x\|^2 = \|h\|^2 + \|p\|^2 \Rightarrow \|h\|^2 \rightarrow \min \Leftrightarrow \|p\|^2 \rightarrow \max$$
$$\sum_{n=1}^N \|h_n\|^2 \rightarrow \min_{v_1, v_2, \dots, v_d} \Leftrightarrow \sum_{n=1}^N \|p_n\|^2 \rightarrow \max_{v_1, v_2, \dots, v_d}$$

2. Жадное построение векторов

$$\begin{cases} \|Xv_k\|^2 \rightarrow \max_{v_k} \\ \|v_k\|^2 = 1 \\ \langle v_i, v_k \rangle = 0 \quad i = \overline{1, k-1} \end{cases}$$

3. Доказательство оптимальности жадного набора векторов

Сравним $L[v_1, v_2, \dots, v_{k-1}]$ с другим набором $L[b_1, b_2, \dots, b_{k-1}]$

$$\sum_{i=1}^{k-1} \|Xv_i\|^2 \geq \sum_{i=1}^{k-1} \|Xb_i\|^2 \text{ (индуктивное предположение)}$$

$\|Xv_k\|^2 \geq \|Xb_k\|^2$, т.к. v_k — решение оптимизационной задачи

PCA & SVD

Любая матрица может быть представлена в виде

$$X = U\Sigma V^T,$$

где

$U(N \times N)$ - ортогональная матрица левого сингулярного базиса
(собственные векторы матрицы XX^T)

$V(D \times D)$ - ортогональная матрица правого сингулярного базиса
(собственные векторы матрицы X^TX)

$\Sigma(N \times D)$ - диагональная матрица с сингулярными числами на
главной диагонали (собственные значения X^TX)

Матрица главных компонент может быть вычислена:

$$XV = U\Sigma$$

Свойства SVD

$$X = U \begin{array}{|c|} \hline S \\ \hline \end{array} V^T$$

- ▶ Число ненулевых сингулярных чисел σ_i^2 совпадает рангом X
- ▶ $\|X\|_F = \sqrt{\sum_{i,j=1}^{N,D} X_{i,j}^2} = \sqrt{\sum_{i=1}^r \sigma_i^2}$
- ▶ Принято упорядочивать сингулярные числа: $\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_r^2$
- ▶ Оптимальное по норме Фробениуса приближение X матрицой \tilde{X}_r ранга r получается занулением всех кроме r наибольших сингулярных чисел (оставляем первые r)
- ▶ Низкоранговое приближение соответствует выбору главных компонент

Выбор размерности нового пространства

Критерий

Ошибка восстановления:

$$L(r) = \|X - \tilde{X}_r\|_F^2 = \sum_{i=r}^D \sigma_i^2$$

Относительная ошибка
восстановления:

$$\hat{L}(r) = \frac{\|X - \tilde{X}_r\|_F^2}{\|X\|_F^2} = \frac{\sum_{i=r}^D \sigma_i^2}{\sum_{i=1}^D \sigma_i^2}$$

Критерий: $\hat{L}(r) \leq \eta$, где $\eta \sim 0.05$

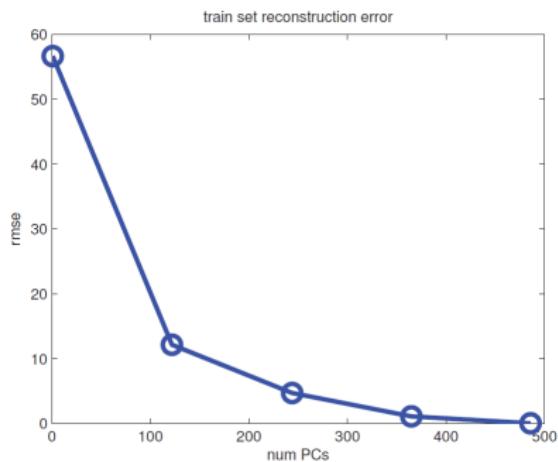
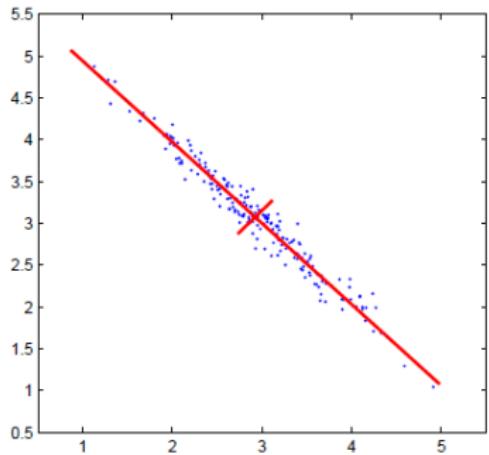
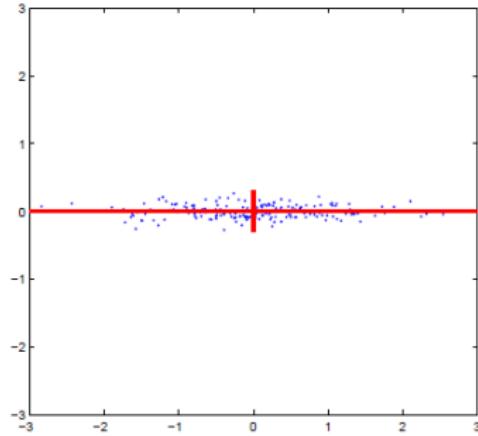


Иллюстрация PCA



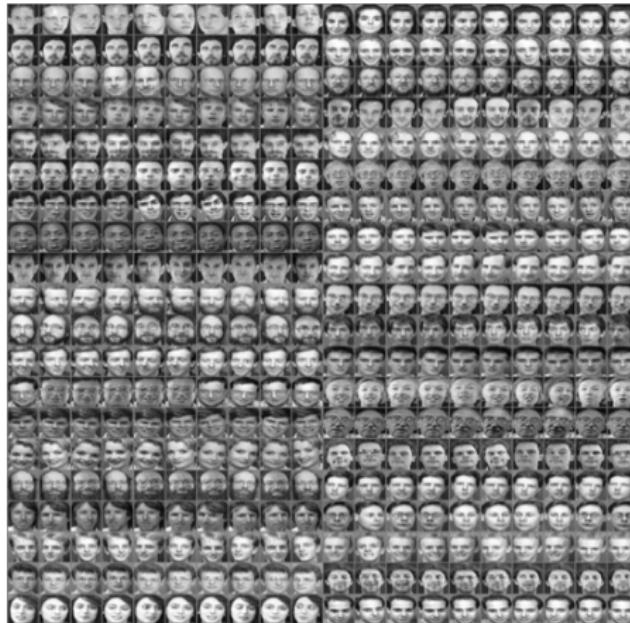
(a) Исходное пространство



(b) Итоговое пространство

- ▶ Сдвигаем начало координат в центр выборки
- ▶ Поворачиваем оси, чтобы признаки не коррелировали
- ▶ Избавляемся от координат с малой дисперсией

PCA для лиц⁵



PCA применяется для идентификации лиц людей. Для этого каждое лицо можно представить вектором координат в пространстве главных компонент.

⁵<https://en.wikipedia.org/wiki/Eigenface>

Eigenfaces

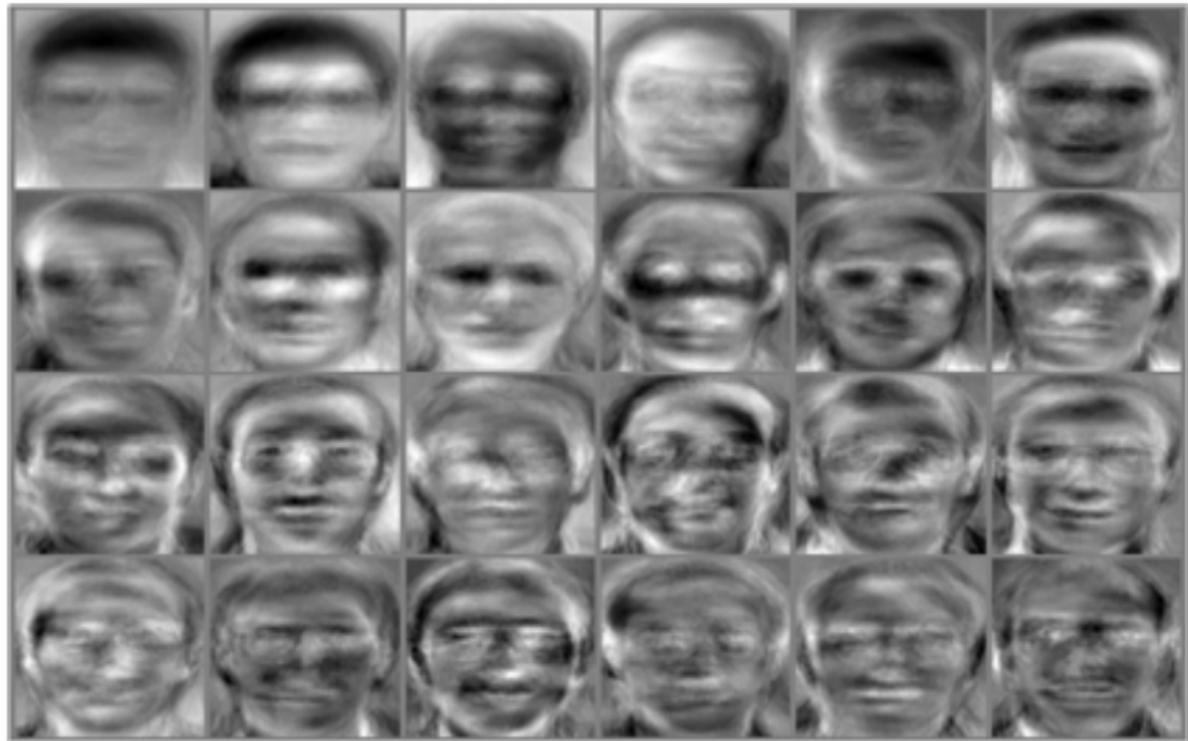
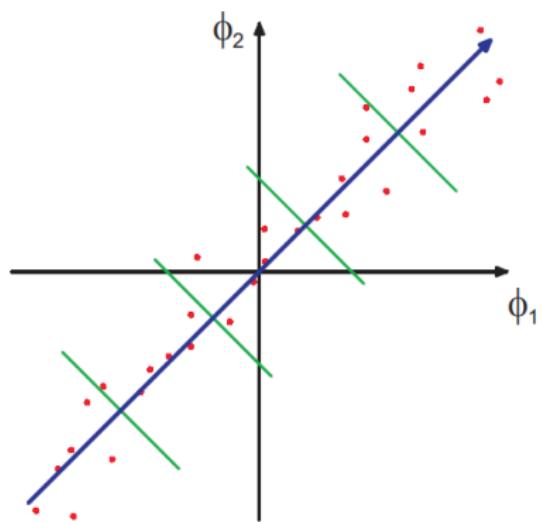
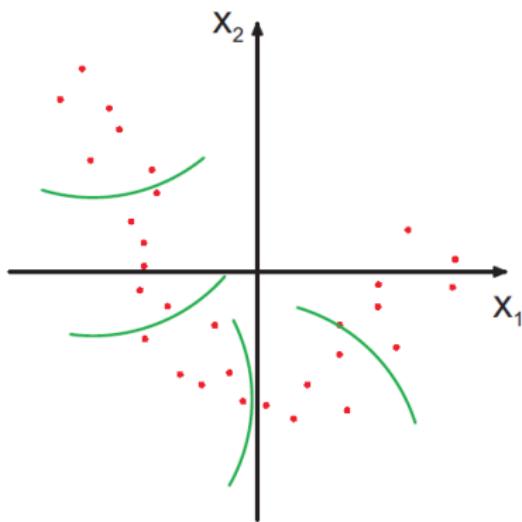


Figure: “Собственные” лица

Kernel PCA

Выберем некоторое нелинейное преобразование $\phi : \mathbb{R}^D \rightarrow H$, при котором в новом пространстве нелинейное многообразие выборки переходит в гиперплоскость.



Ядра

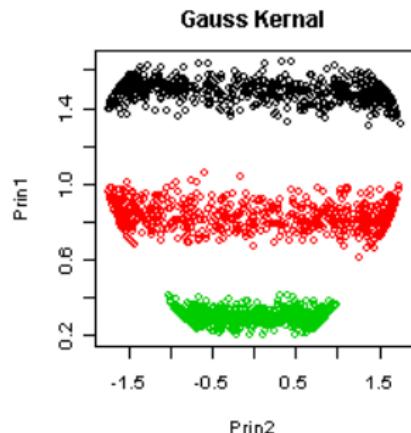
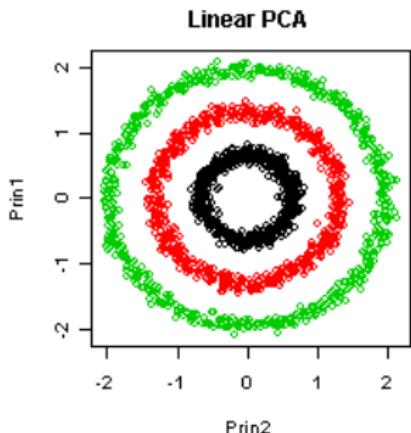
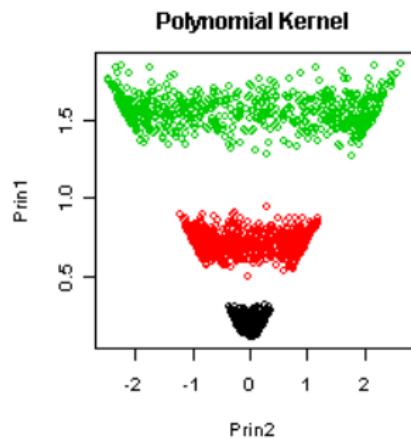
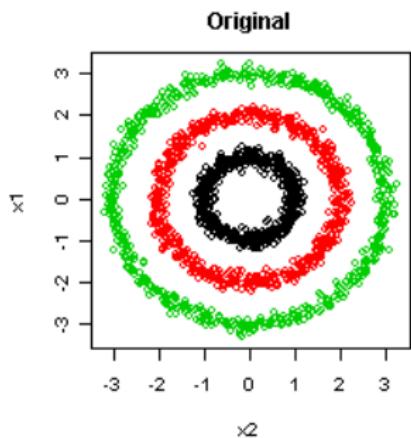
Ядро — скалярное произведение в новом пространстве
 $K(x, y) = \langle \phi(x), \phi(y) \rangle$. Его использование не требует перехода в пространство H .

PCA: требуется знать только $X^T X = K(X, X)$

Примеры ядер

- ▶ Линейное: $x^T y$
- ▶ Полиномиальное: $(1 + x^T y)^d$
- ▶ Гауссово: $\exp(-\|x - y\|^2 / \sigma^2)$

Пример Kernel PCA



t-SNE

t-SNE

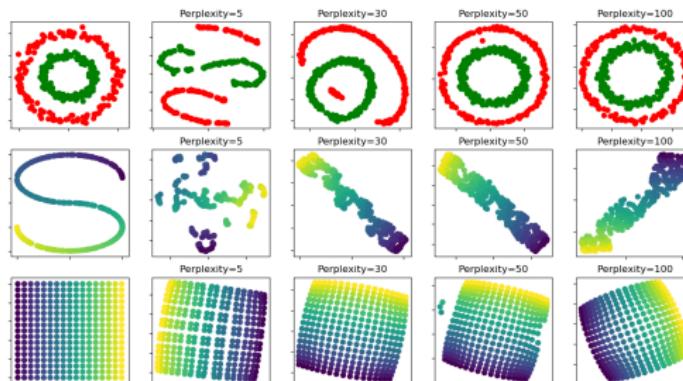
Считаем вероятность получить точку j при выборе из нормального распределения с центром в точке i :

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}, \quad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

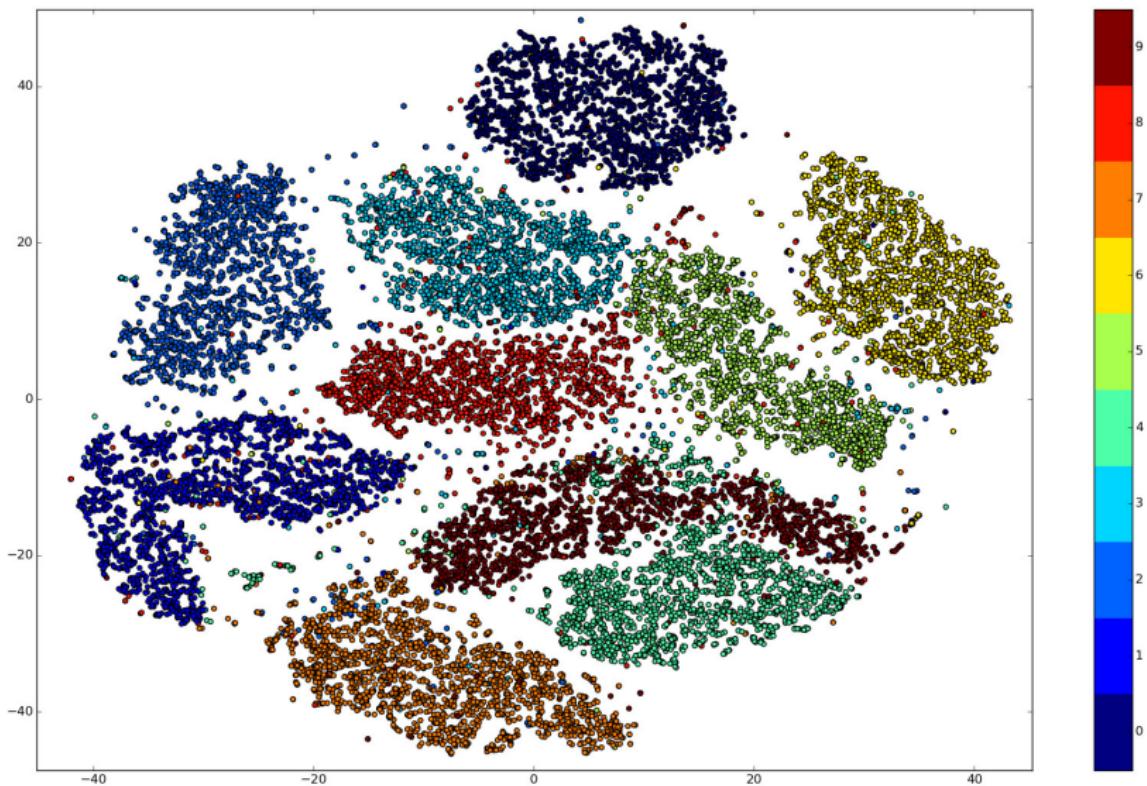
Проклятие размерности — в целевом пространстве используем исправленное распределение.

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

Дальше просто минимизируем KL -дивергенцию градиентным спуском.

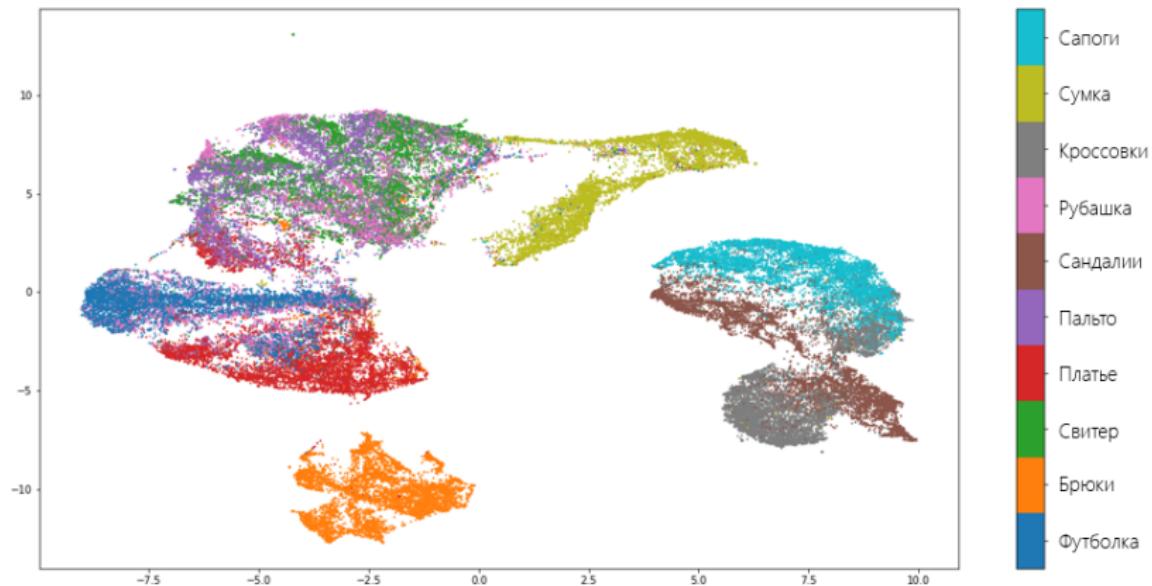


t-SNE



t-SNE UMAP⁶

Преимущества: намного быстрее, не только визуализация ($d \gg 3$), сохраняет «глобальную структуру», любые расстояния, вложение известных точек, теоретическое обоснование.

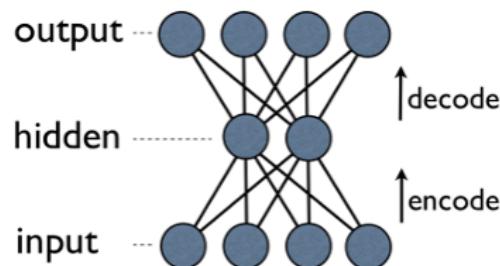


⁶<https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668>

Автокодировщики

Структура

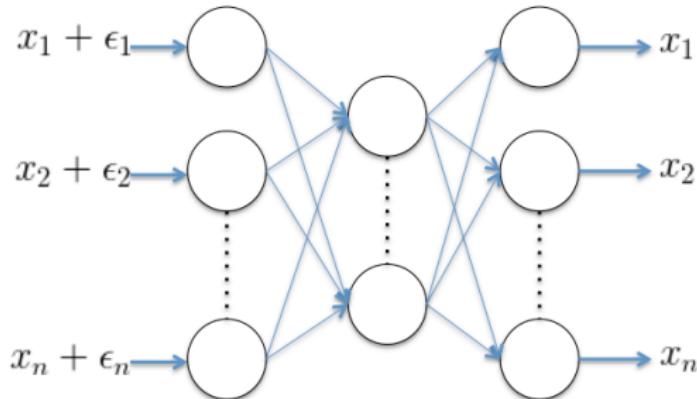
- ▶ Рассматривается сеть, обучаемая на отображении $f(x) = x$
- ▶ Внутри сети есть bottleneck слой, активации которого — представление объектов в низкоразмерном пространстве
- ▶ В сверточных сетях: pooling/stride и deconvolution / unpooling



Применение

- ▶ Выделение признаков для других алгоритмов
- ▶ Снижение размерности
- ▶ Предобучение на неразмеченных данных

Denoising autoencoder



Примеры шума

- ▶ Нормальный шум: $\mathcal{N}(\mu, \sigma^2 I)$
- ▶ Маскирующий шум: часть элементов обнуляется
- ▶ Соль и перец: часть элементов принимают максимальное/минимальное допустимое значение

Denoising autoencoder

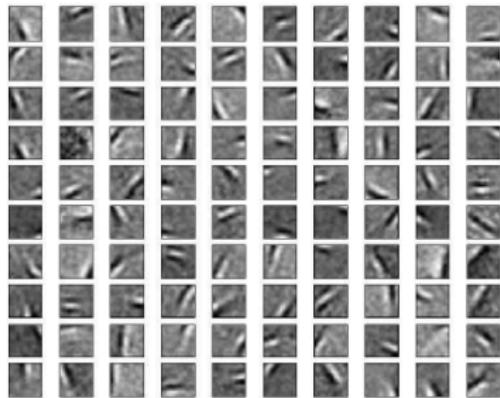
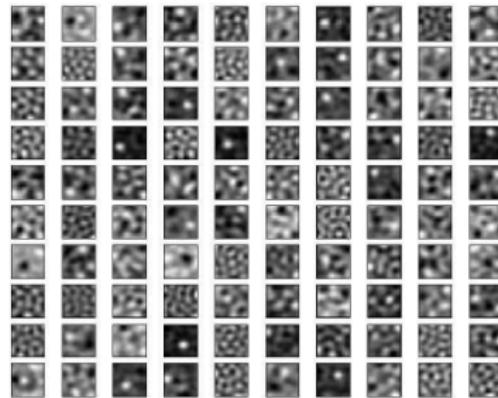
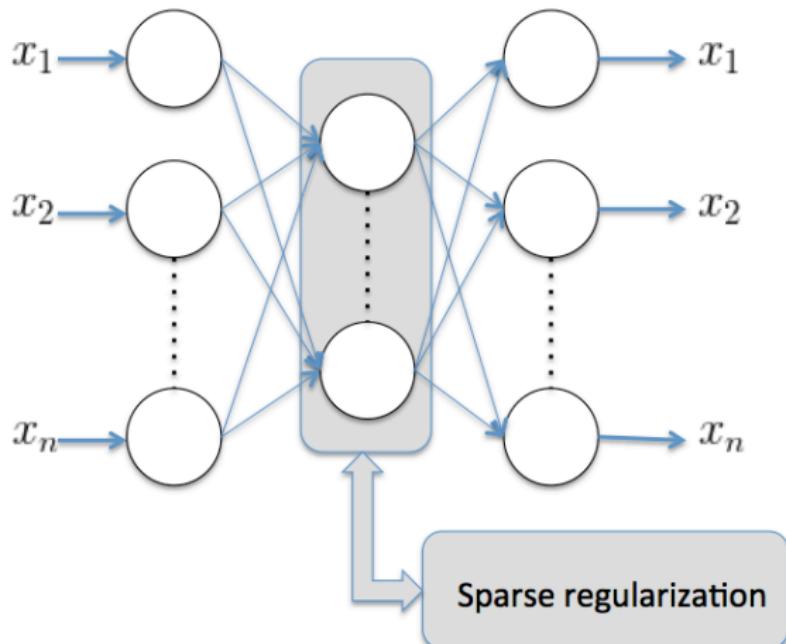


Figure: Слева автоэнкодер, справа автоэнкодер с гауссовым шумом⁷

⁷Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, 2010, P. Vincent, Y. Bengio, and others

Разреженный автокодировщик



Идея: можем использовать больший скрытый слой, если введем регуляризацию

Sparse autoencoder

Регуляризатор разреженности

- ▶ Хотим, чтобы каждый нейрон в среднем активировался в ρ случаях ($\rho = 0.05$)
- ▶ Пусть средняя активация нейрона $\hat{\rho}$
- ▶ Регуляризатор: $KL(\rho \parallel \hat{\rho}) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}$

KL дивергенция

$$KL(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

- ▶ $KL(p \parallel q) \geq 0$
- ▶ $KL(p \parallel q) = 0 \Leftrightarrow p(x) = q(x)$ п.в

Sparse autoencoder⁸

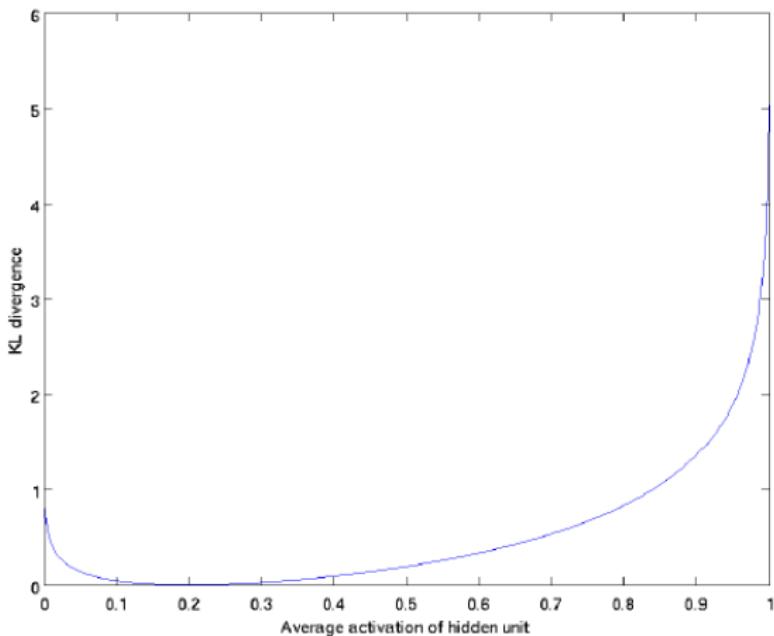


Figure: KL достигает минимального значение в точке $\hat{\rho}_j = \rho$

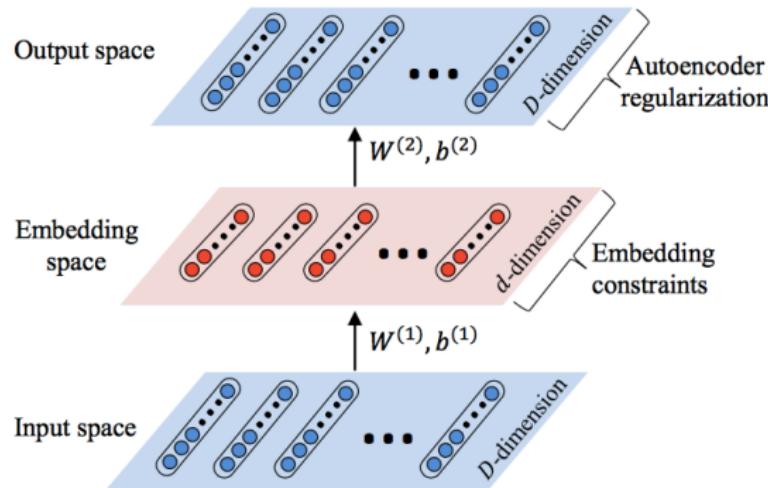
⁸Sparse autoencoder, CS294A Lecture notes, Andrew Ng

Embedding regularized AutoEncoder⁹

Совмещаем идею MDS и AE:

$$\begin{cases} \sum_i \|X_i - \hat{X}_i\|^2 \rightarrow \min \\ \sum_{i,j} (\|X_i - X_j\| - \|E_i - E_j\|)^2 \rightarrow \min \end{cases}$$

В результате получаем более качественный embedding.



⁹<http://www.ecmlpkdd2013.org/wp-content/uploads/2013/07/196.pdf>

Самоорганизующиеся карты Кохонена

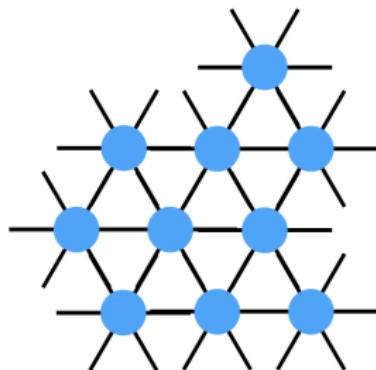
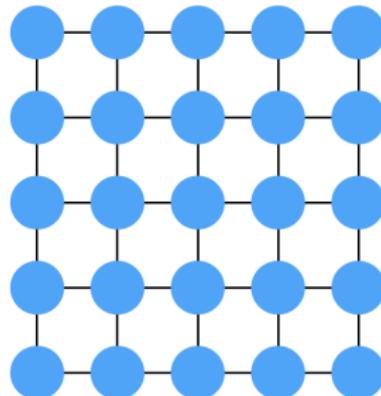
Описание структуры

Построим отображение узлов решетки (прямоугольной или шестиугольной) в пространство данных.

$$\phi : \mathcal{A} \rightarrow \mathbb{R}^D$$

$$\phi(i) = w_i, i \in \mathcal{A}$$

- ▶ Соседние узлы решетки должны быть близки после отображения
- ▶ Образ узлов должен “хорошо” приближать данные



Визуализация

- ▶ 2D визуализация
- ▶ 3D визуализация

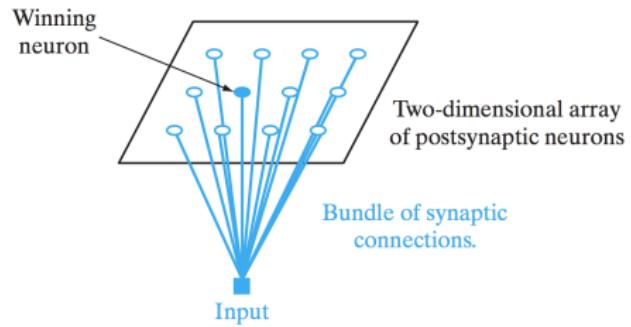
Шаги обучения SOM

Выбираем случайную точку из данных. Затем проводим обучение в 3 этапа:

- ▶ Соревнование — нейроны борются за право быть активированными (только один победитель)
- ▶ Кооперация — соседние с активным нейроном также активируются
- ▶ Адаптация — изменение положений образов

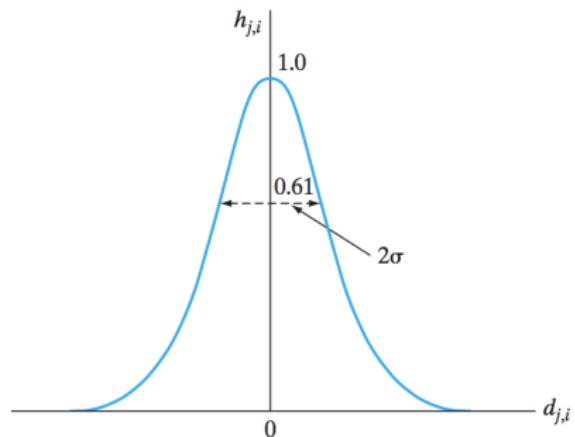
Соревнование

- ▶ $i(x) = \arg \min_{i \in \mathcal{A}} \|x - w_i\|$
- ▶ $i(x)$ может быть
рассмотрен как механизм
внимания
- ▶ Эта часть алгоритма —
кодирование, т.е. $\mathbb{R}^D \rightarrow \mathcal{A}$



Кооперация

- ▶ Зададим значение активаций нейронов
- ▶ Более далекие от победителя нейроны получают меньшую активацию
- ▶ Расстояние на решетке: $d_{j,i}$
- ▶ Активация:
$$h_{j,i}(x) = \exp(-d_{j,i}^2/2\sigma^2)$$
- ▶ $\sigma(n) = \sigma_0 \exp(-n/\tau_1)$



Адаптация

- ▶ $w_j = (1 - \eta h_{j,i(x)})w_j + \eta h_{j,i(x)}x$
- ▶ $\eta(n) = \eta_0 \exp(-n/\tau_2)$

Пример:

$$\|x - w_0\| = 0.1$$

$$\|w_0 - w_1\| = \|w_0 - w_2\| = \|w_0 - w_3\| =$$

$$\|w_0 - w_4\| = 1$$

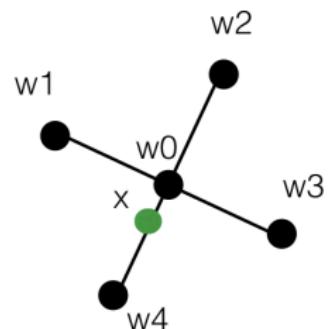
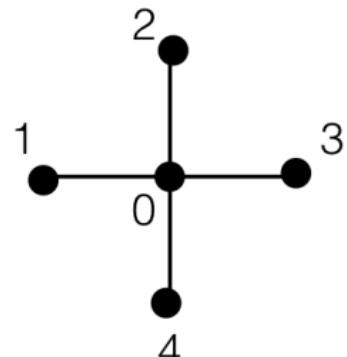
$$\sigma = \eta = 1$$

$$\exp(-1) \simeq 0.4$$

$$\exp(-0.5) \simeq 0.6$$

Найти вектор обновлений образов

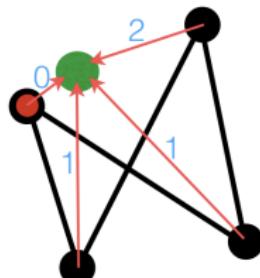
$$w_0, w_1, w_2, w_3, w_4$$



Фазы обучения

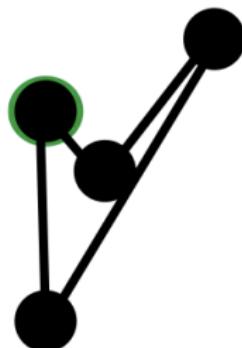
Этап сортировки

Карта производит топологическую сортировку своих узлов



Этап сходимости

Карта ищет оптимальное квантование исходного пространства



Количество итераций: $\sim 500 \cdot |\mathcal{A}|$
 $0 < \eta \leq 0.01$
 $h_{j,i(x)}$ содержит малую окрестность

Количество итераций: 1000

Отображение модели

- U-matrix: каждому нейрону приписывается среднее расстояние до его топологических соседей

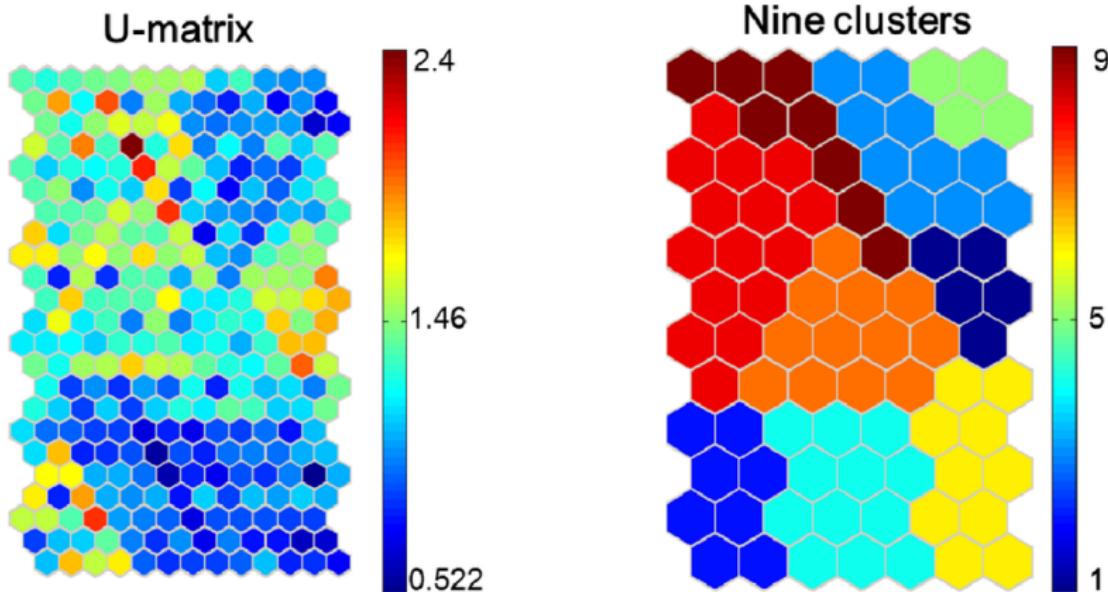


Figure: U-matrix и k-means кластеризация в новом пространстве¹⁰

¹⁰<http://www.mdpi.com/1660-4601/11/4/3618/htm>

Уровень бедности

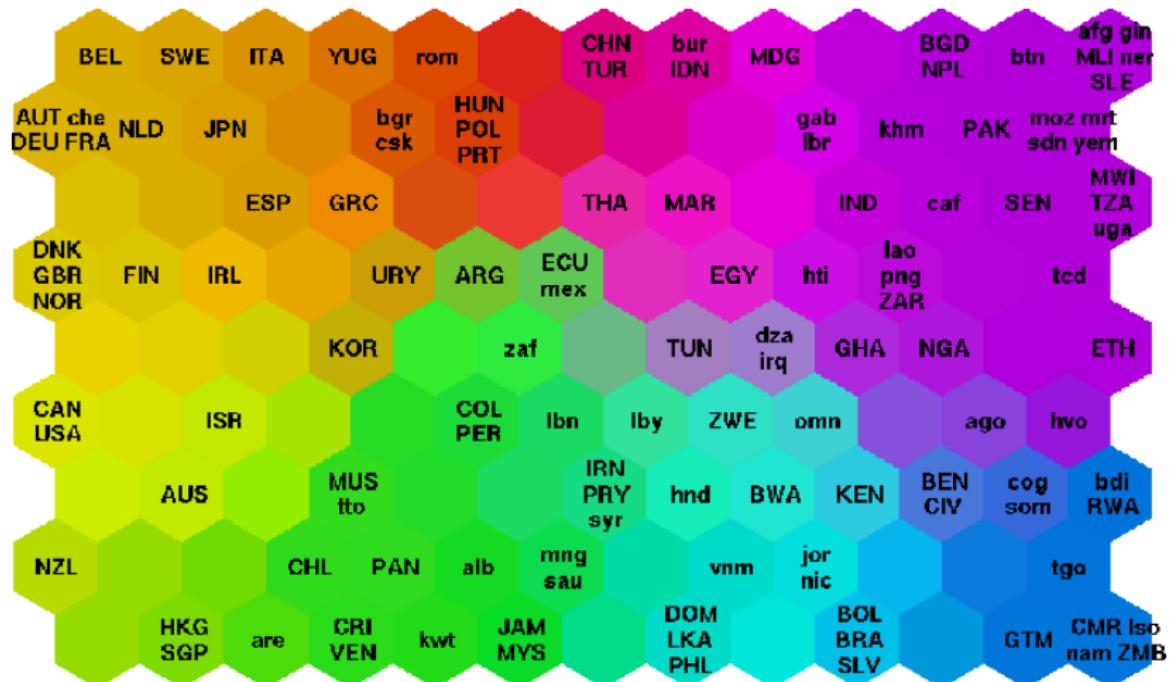


Figure: SOM стран по 39 показателям World Bank¹¹

¹¹<http://www.cis.hut.fi/research/som-research/worldmap.html>

Уровень бедности

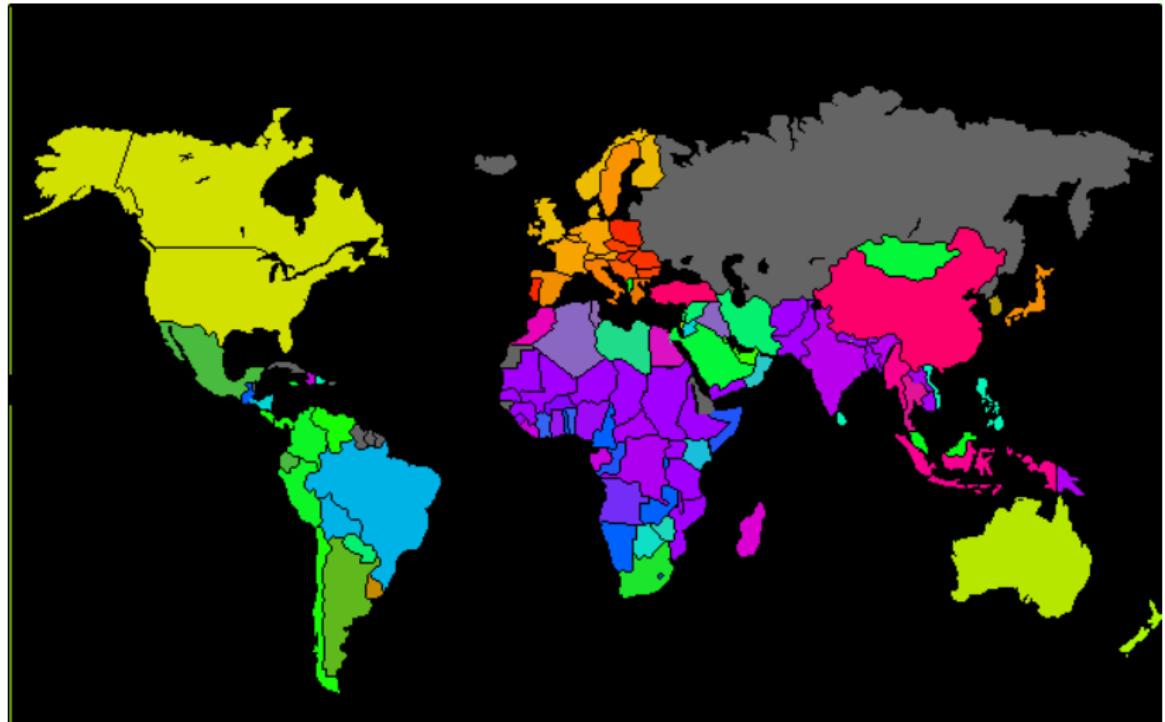


Figure: Цвета кластеров перенесены на реальную карту¹²

¹²<http://www.cis.hut.fi/research/som-research/worldmap.html>

Вопросы

