

Эмбеддинги. Word2vec, Doc2Vec, fasttext

Название курса

Что будет на уроке

1. Word2Vec
2. Второе
3. Третье

Представления слов

Слова представляются в виде one-hot вектора

умный	$[0, \dots, 1, \dots, 0, \dots, 0, \dots, 0]$
сообразительный	$[0, \dots, 0, \dots, 1, \dots, 0, \dots, 0]$
собака	$[0, \dots, 0, \dots, 0, \dots, 1, \dots, 0]$

Недостатки:

Большая размерность векторов

Все вектора одинаково похожи

Сложно кодировать дополнительную информацию

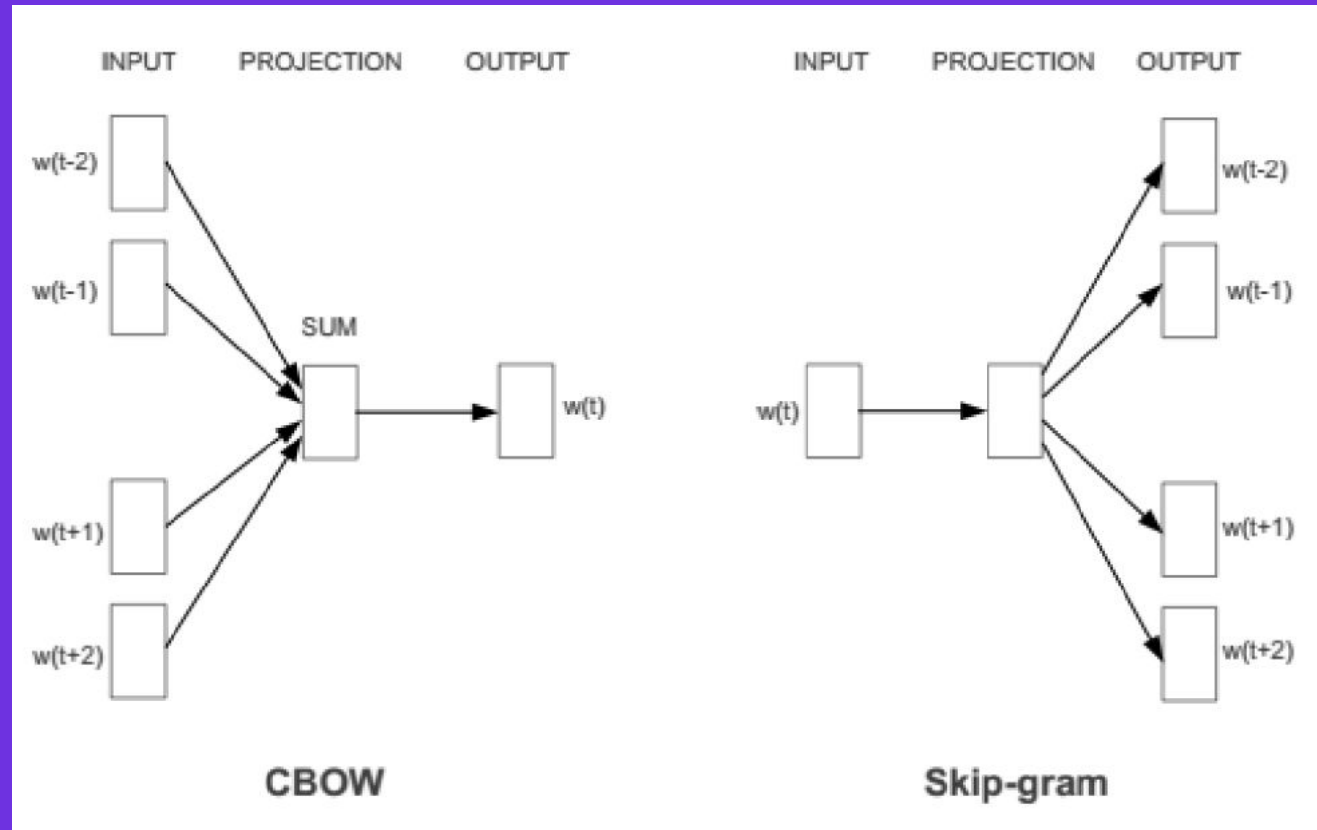
Гипотеза компактности

Близкие (похожие) слова встречаются в похожем контексте

Цель:

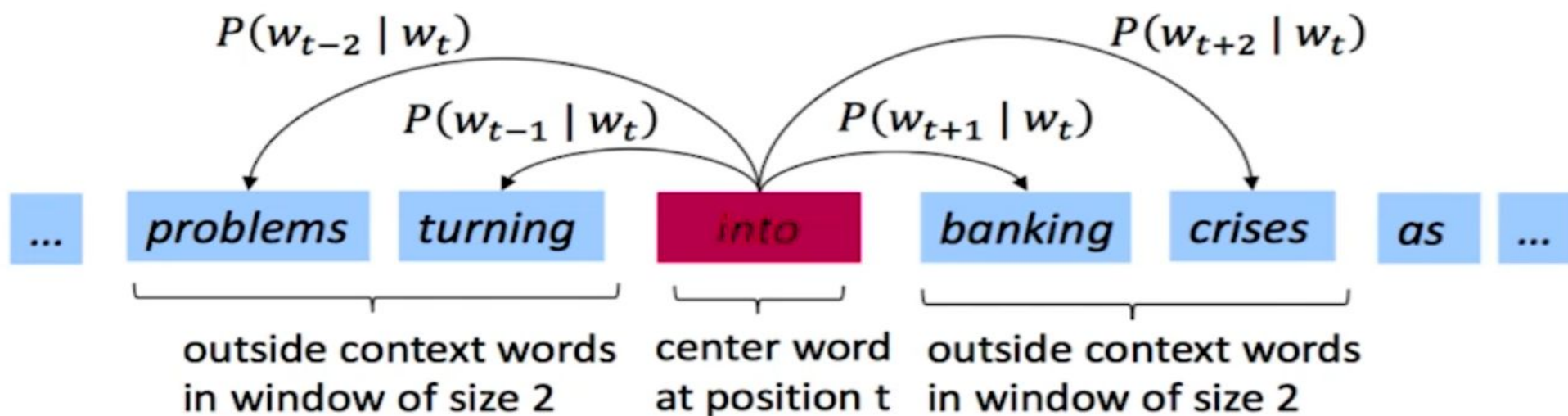
У близких слов (синонимов, родственных слов) и представления получают близкие

Word2Vec



Skip-gram

Зная контекст мы хотим максимизировать вероятность центрального слова. Перемещаемся скользящим окном



Skip-gram

- Вероятность встретить слово w_o рядом со словом w_I :

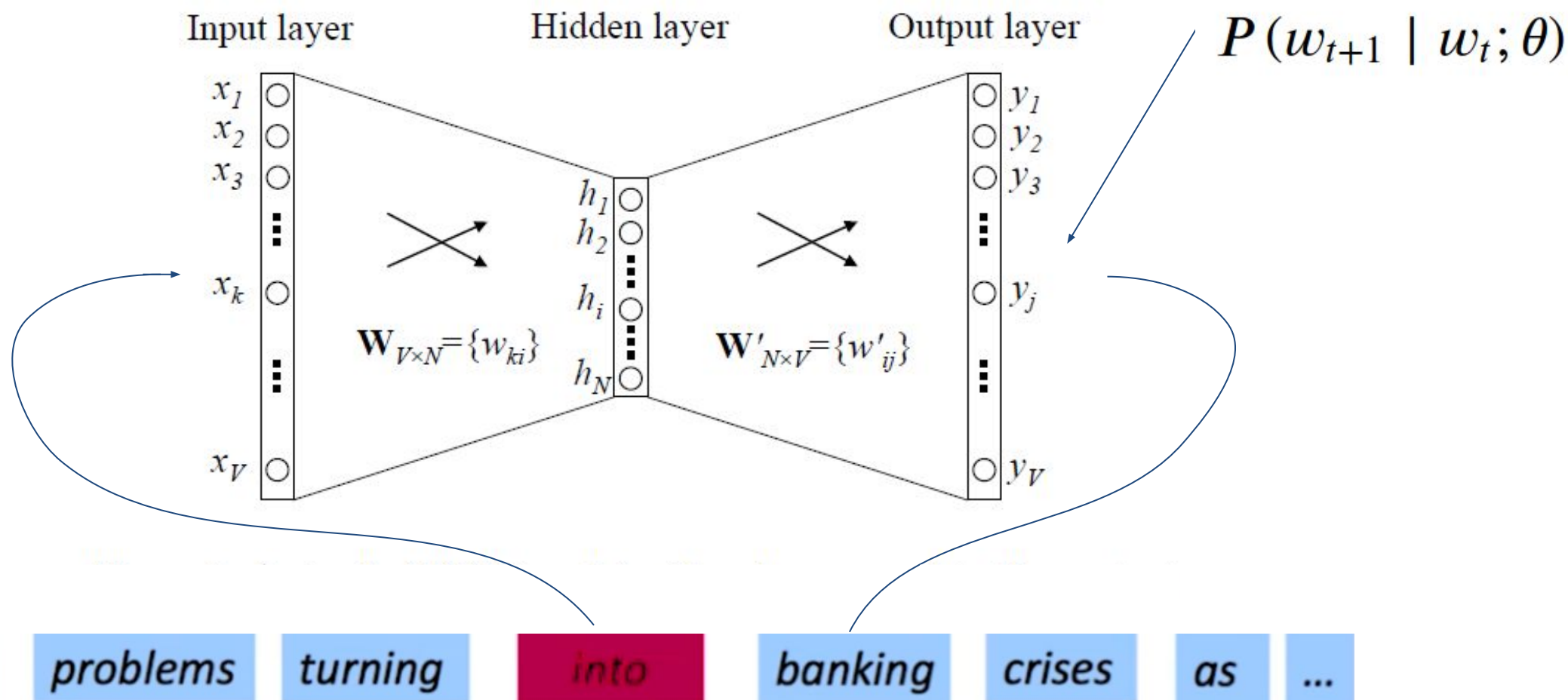
$$p(w_o|w_I) = \frac{\exp(\langle v'_{w_o}, v_{w_I} \rangle)}{\sum_{w \in W} \exp(\langle v'_w, v_{w_I} \rangle)}$$

- Функционал для текста $T = (w_1 w_2 \dots w_n)$:

$$\sum_{i=1}^n \sum_{\substack{-c \leq j \leq c \\ j \neq 0}} \log p(w_{i+j}|w_i) \rightarrow \max$$

Skip-gram

SoftMax models



Skip-gram

Простейшая модель классификации — линейный классификатор, применяемый к вектору контекста

$$\text{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} = p_i$$

Для обучения минимизируем функцию потерь

Skip-gram

- Вероятность встретить слово w_O рядом со словом w_I :

$$p(w_O | w_I) = \frac{\exp(\langle v'_{w_O}, v_{w_I} \rangle)}{\sum_{w \in W} \exp(\langle v'_w, v_{w_I} \rangle)}$$

- Считать знаменатель ОЧЕНЬ затратно
- Значит, и производные считать тоже долго

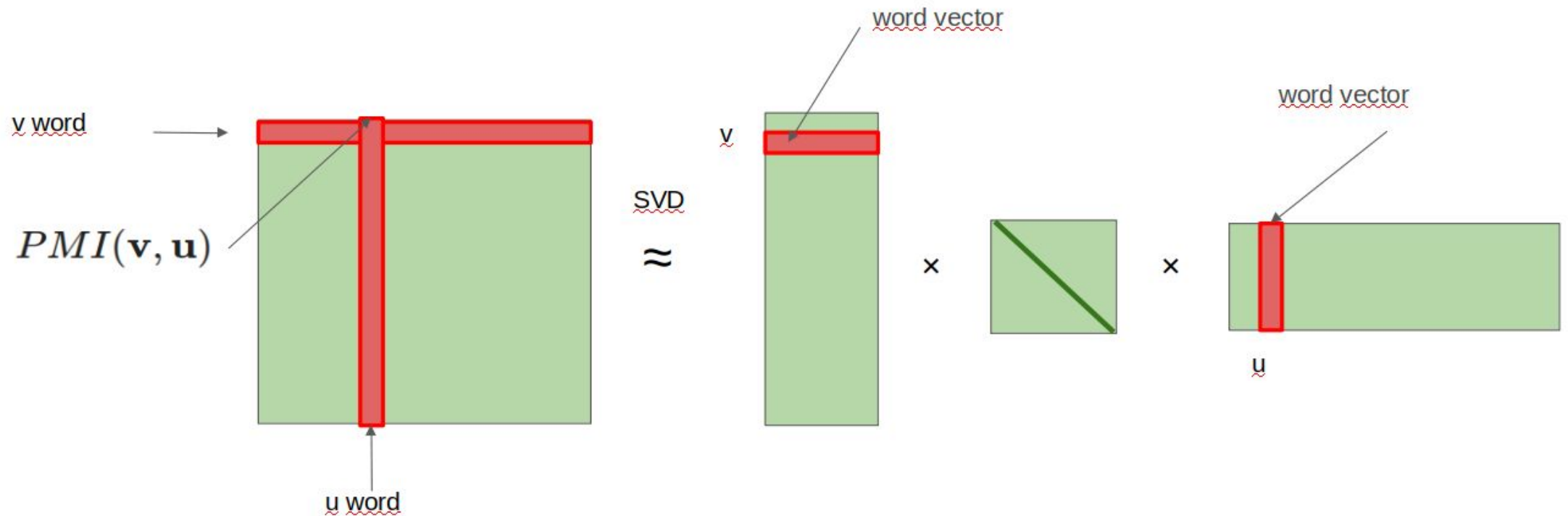
Negative sampling

$$p(w_O | w_I) = \log \sigma(\langle v'_{w_O}, v_{w_I} \rangle) + \sum_{i=1}^k \log \sigma(-\langle v'_{w_i}, v_{w_I} \rangle)$$

- w_i — случайно выбранные слова
- Слово w генерируется с вероятностью $P(w)$ — шумовое распределение
- $P(w) = \frac{U(w)^{\frac{3}{4}}}{\sum_{v \in W} U(v)^{\frac{3}{4}}}$, $U(v)$ — частота слова v в корпусе текстов

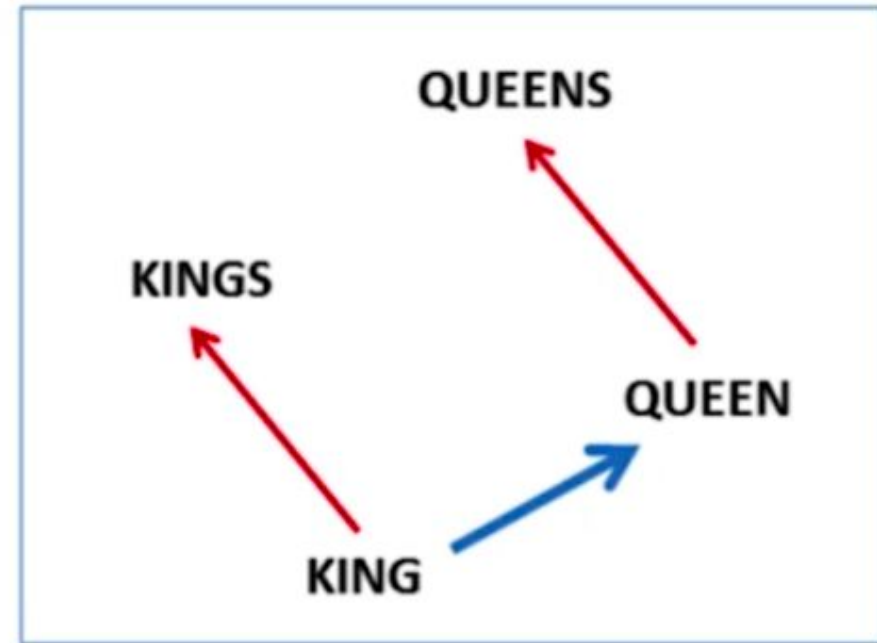
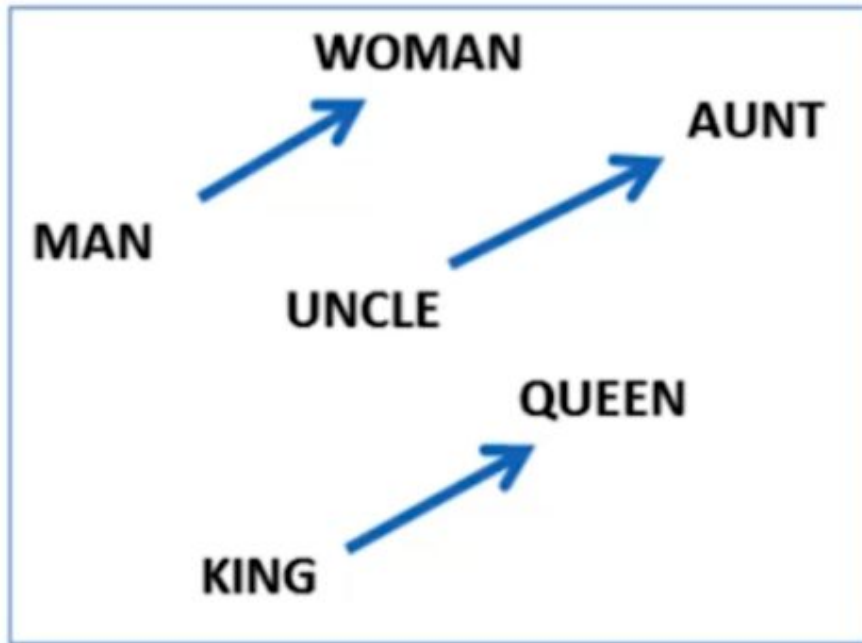
Word2vec vs SVD

Word2Vec with negative sampling \approx matrix factorization



Word2vec

$$\underline{v(\text{king})} - \underline{v(\text{man})} + \underline{v(\text{woman})} \approx \underline{v(\text{queen})}$$



Fasttext

Разделите слово на мешок из n-граммов: apple = <ap, ppl, ple, le> (BPE)

Вычислить вектор для каждого n-грамма

Вектор для слова = сумма <вектора слова, векторов для n-граммов слов>

Спасибо!

**Каждый день
вы становитесь
лучше :)**

