

Векторные представления (word embeddings) в задачах информационного поиска

Информационный поиск: проблемы модели bag of words

- В традиционных подходах к информационному поиску (поиск, классификация текстов) не известны никакие отношения между словами (например, семантическая близость)
 - В поиске нужно расширение запроса
 - В классификации – появление слов, которых не было в обучающей выборке – огромная проблема
 - Тематические модели – попытка сгруппировать слова в темы

Векторные представления слов?

- Дистрибутивная семантика
 - Идея с 1954 года
 - Реальные эксперименты с 1990 годов
- Использование контекстов для построения векторов слов
 - похожие по смыслу слова встречаются в похожих контекстах.
 - Можно использовать контексты, чтобы сопоставить целевому слову вектор контекстов
 - И тогда можно будет находить сходство между словами (косинусная близость) на основе их векторов
 - Большие размерности – применение латентного семантического анализа (LSA) для сокращения пространства, основанного на svd

Проблемы с SVD

- Вычислительная сложность квадратичная
 - Миллионы слов и документов
- Трудно учитывать новые слова и документы
- Нельзя ли как-то иначе сокращать размерность?

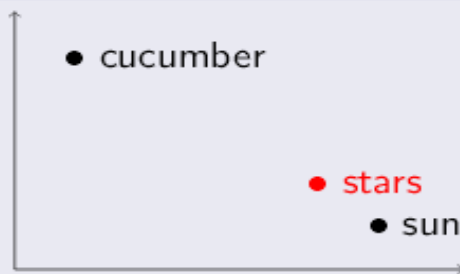
До 2013 года: дистрибутивная семантика

he curtains open and the stars shining in on the barely
ars and the cold , close stars " . And neither of the w
rough the night with the stars shining so brightly , it
made in the light of the stars . It all boils down , wr
surely under the bright stars , thrilled by ice-white
sun , the seasons of the stars ? Home , alone , Jay pla
m is dazzling snow , the stars have risen full and cold
un and the temple of the stars , driving out of the hug
in the dark and now the stars rise , full and amber a
bird on the shape of the stars over the trees in front
But I could n't see the stars or the moon , only the
they love the sun , the stars and the stars . None of
r the light of the shiny stars . The plash of flowing w
man 's first look at the stars ; various exhibits , aer
rief information on both stars and constellations, inc

Construct vector representations

	shining	bright	trees	dark	look
stars	38	45	2	27	12

Similarity in meaning as vector similarity



Но вектора большой размерности: размера словаря

Нейронные языковые модели в дистрибутивной семантике

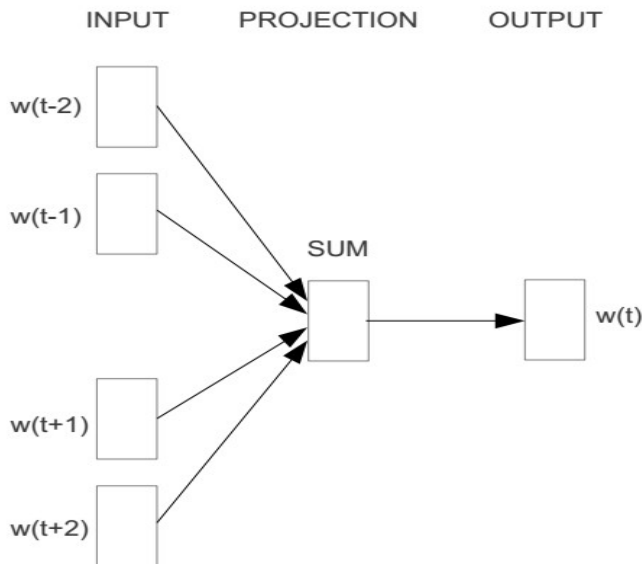
- (Baroni et al., 2014) Don't count, predict!
- Т.е. классическая дистрибутивная семантика подсчитывает количество совместных встречаемостей слов и вычисляет вектора
- А новые подходы получают векторное представление слов на основе предсказания соседних слов
 - Обучаются векторным представлениям небольшой размерности
 - Mikolov et al. 2013: пакет word2vec

Распределенные представления слов (word embeddings)

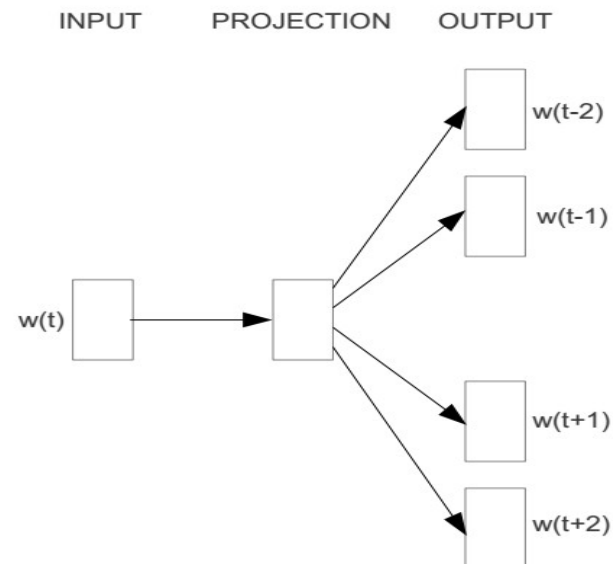
- Комбинирование векторной семантики с вероятностными языковыми моделями
- Слово представляется как вектор низкой размерности (100-1000 измерений)
 - Word embedding
- Обучение происходит при решении задачи языкового моделирования, т.е. предсказания последовательностей слов
- Пакеты Word2vec, glove, Fasttext
 - С 2013

Представление значения слова – word2vec (Mikolov et al., 2013)

- 2 базовые архитектуры нейронных сетей:
 - Continuous Bag of Word (CBOW): использует окно контекста для предсказания слова
 - Skip-gram (SG): используется слово для предсказания окружающих слов



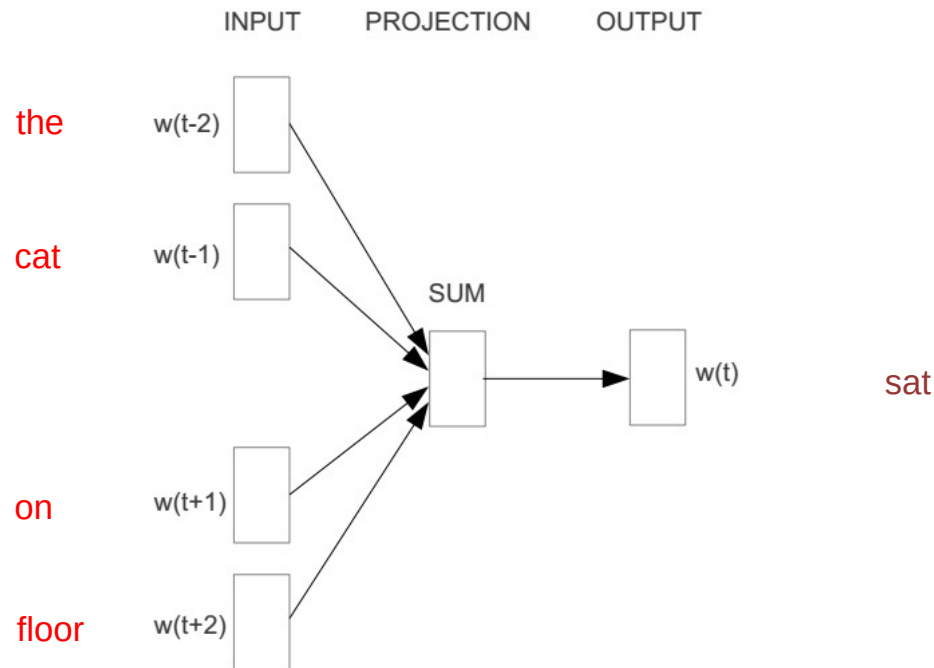
CBOW



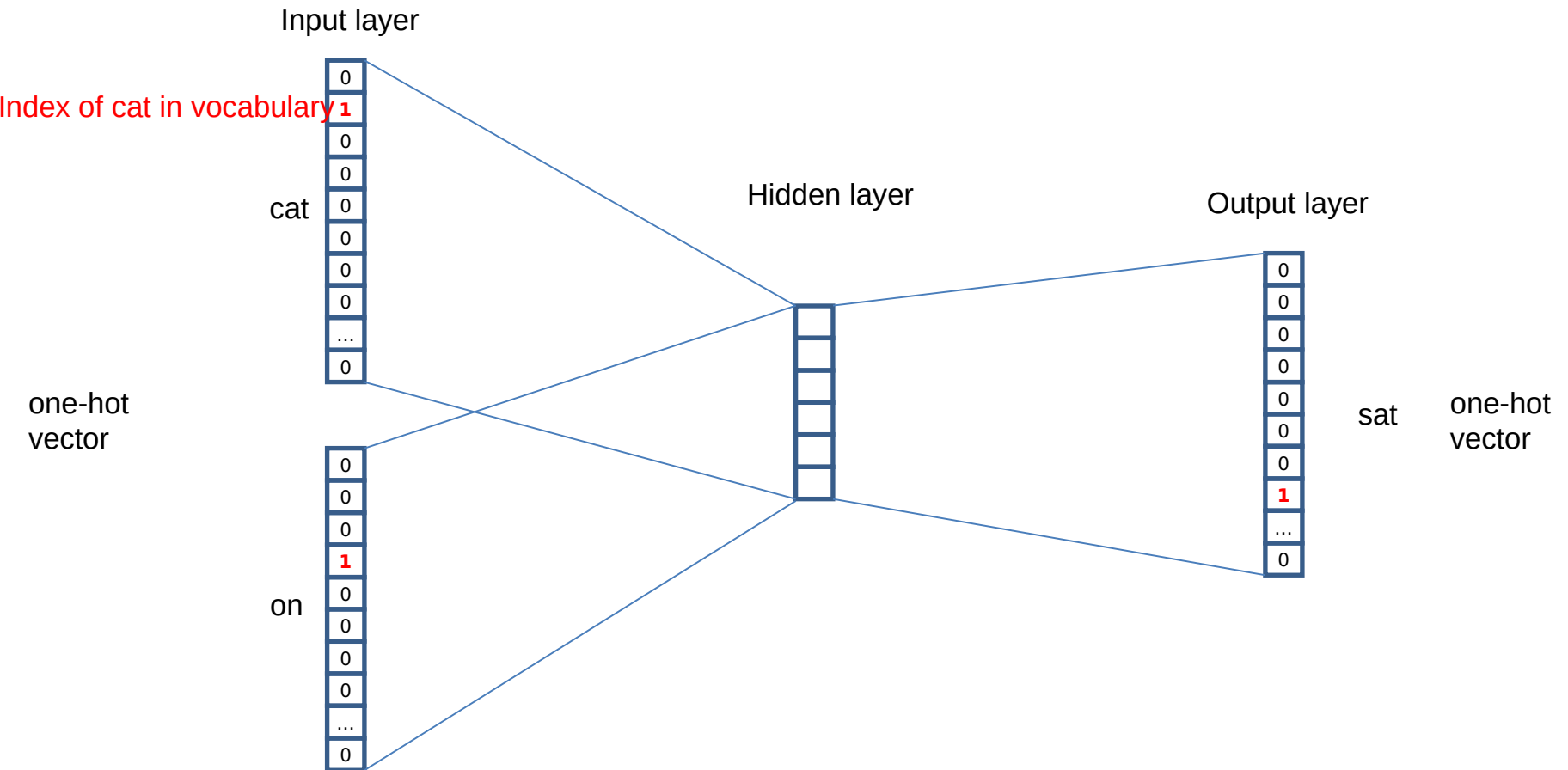
Skip-gram

Word2vec – Continuous Bag of Word

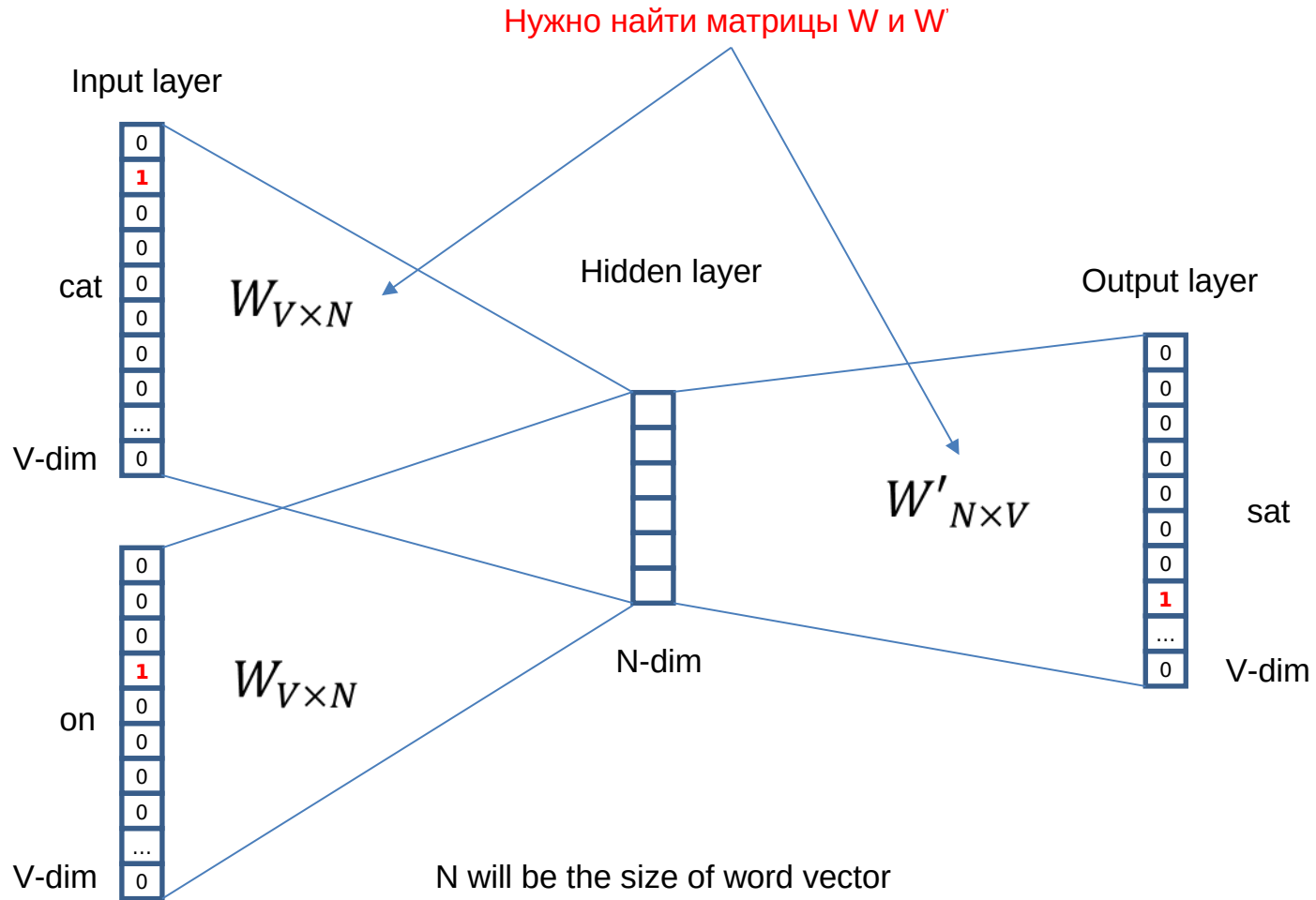
- “The cat sat on floor”
 - Window size = 2



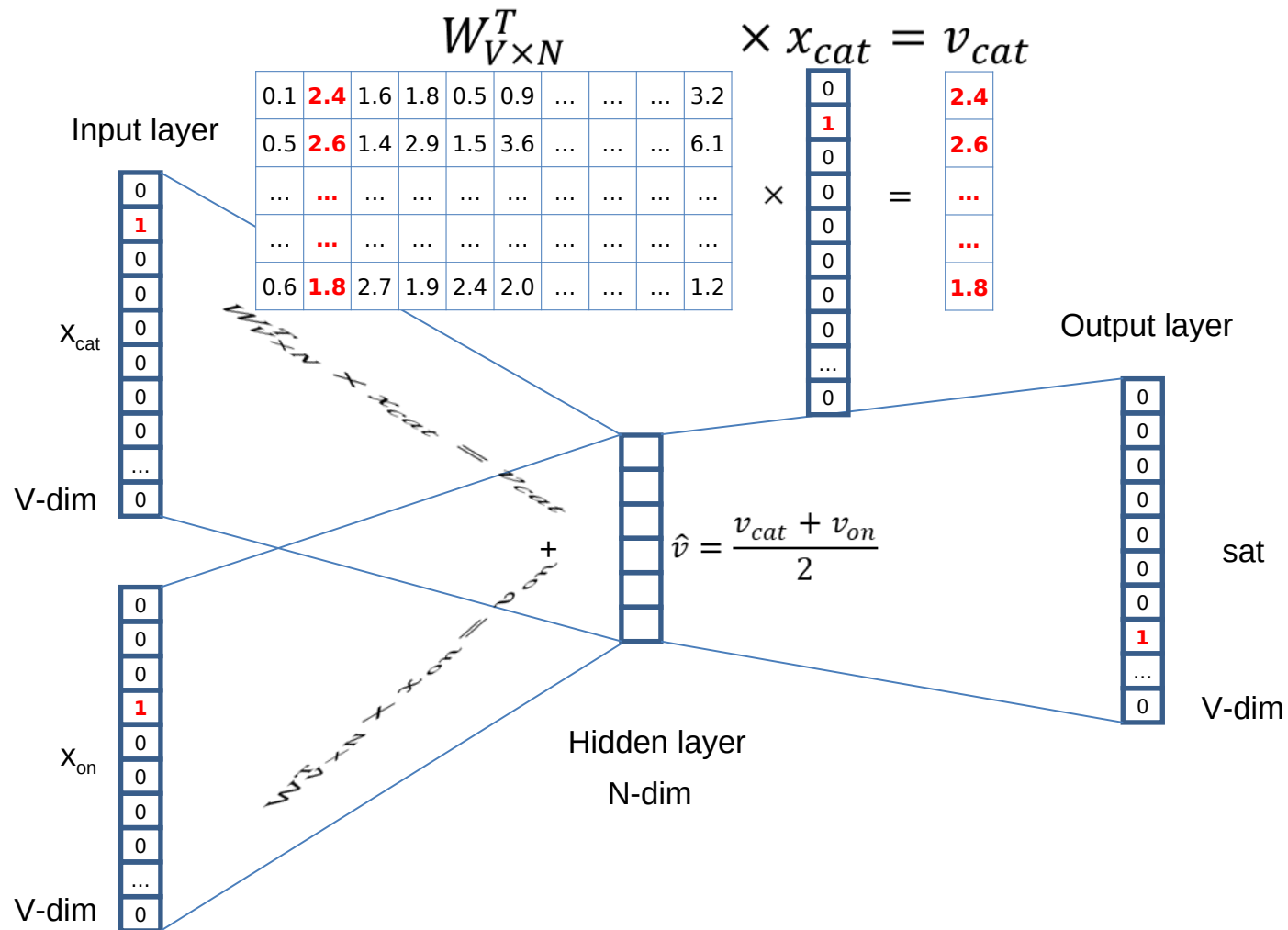
Word2Vec: Continuous bag of words



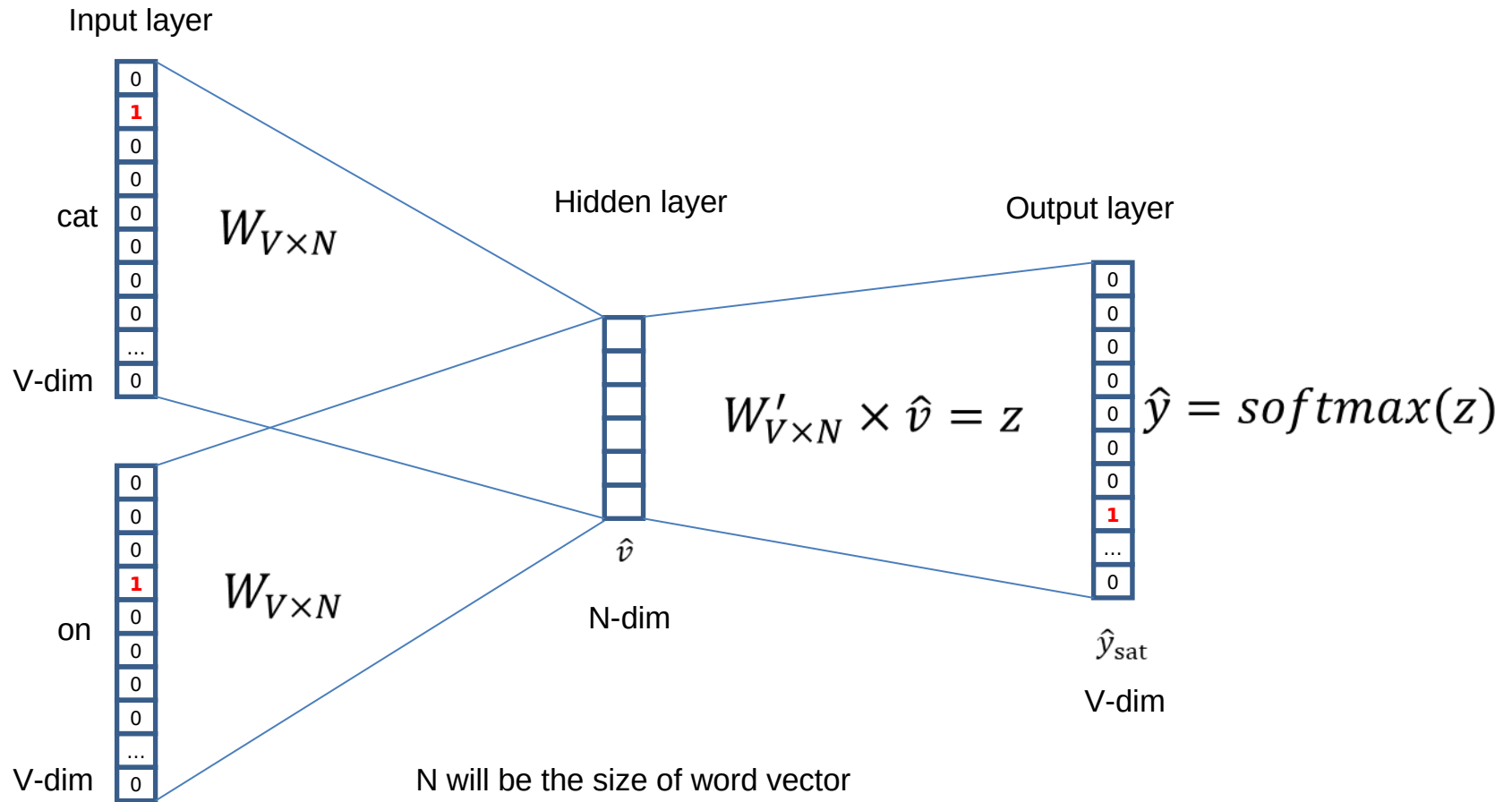
Word2Vec: Continuous bag of words



Word2Vec: Continuous bag of words



Word2Vec: Continuous bag of words



Нейронная языковая модель:

- Вход – one-hot vector – вектор всех нулей и одной 1 в позиции текущего слова
- Projection – layer – выделяет из матрицы вектор, соотв. данному слову (h)
- Выходной уровень получается линейной комбинацией:
 - $S = Wh$
- Результат выходного уровня вероятность появления слова, так называемый softmax

$$p_i = \text{softmax}(s_i, \vec{s}) = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

- Word2vec - это однослойный персептрон с логистической функцией активации (обобщение для многомерного случая)

Softmax

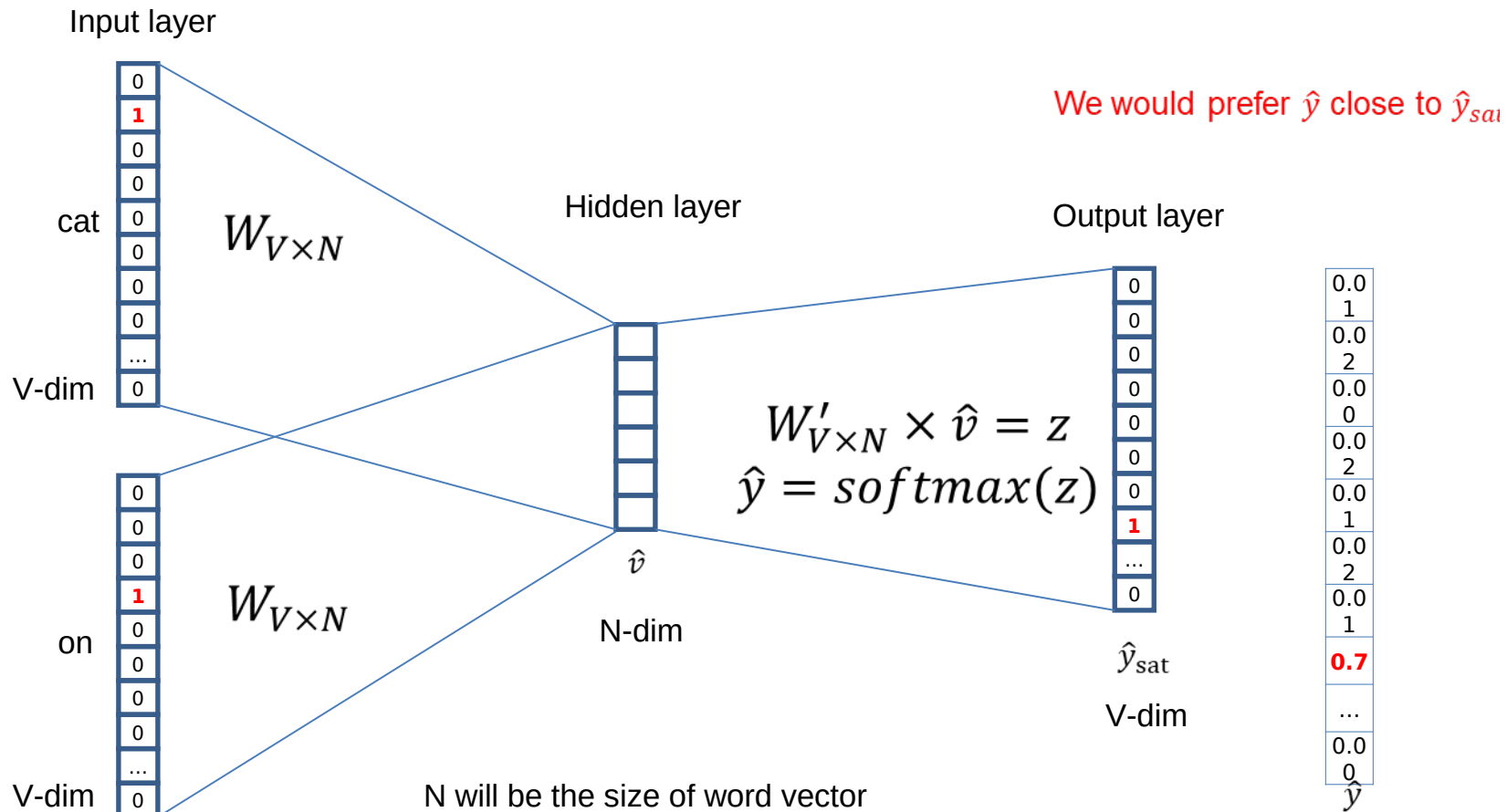
$$f(x) = \frac{1}{1 + e^{-x}}$$

- Softmax – обобщение применения логистической функции для многомерного случая

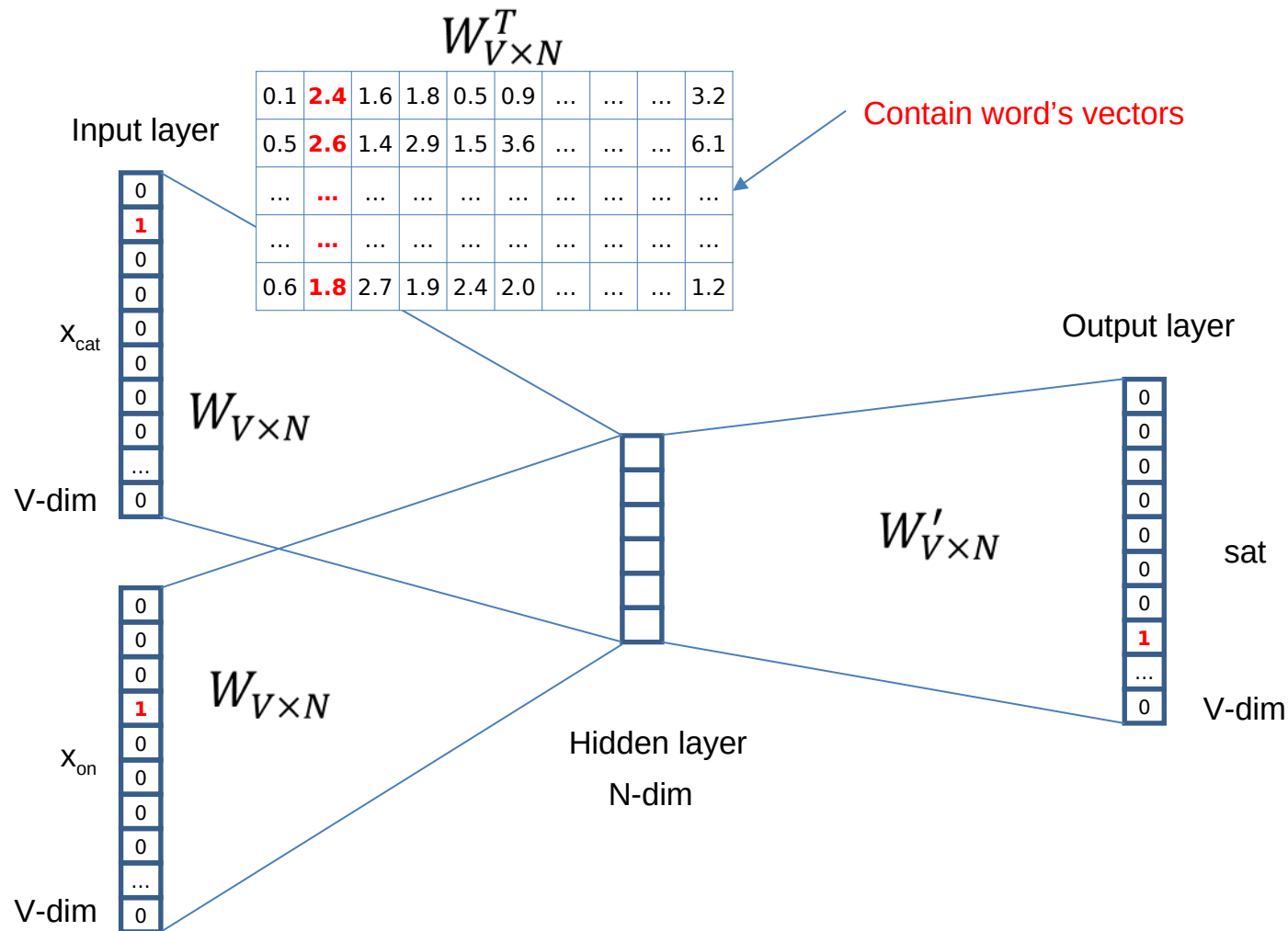
$$p_i = \text{softmax}(s_i, \vec{s}) = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

- Softmax повышает максимальную величину и «прижимает» меньшие величины
- Примеры: сеть предсказала значения
- [1,2,3,**4**,1,2,3] -> softmax [0.024, 0.064, 0.175, **0.475**, 0.024, 0.064, 0.175]

Word2Vec: Continuous bag of words



Word2Vec: Continuous bag of words



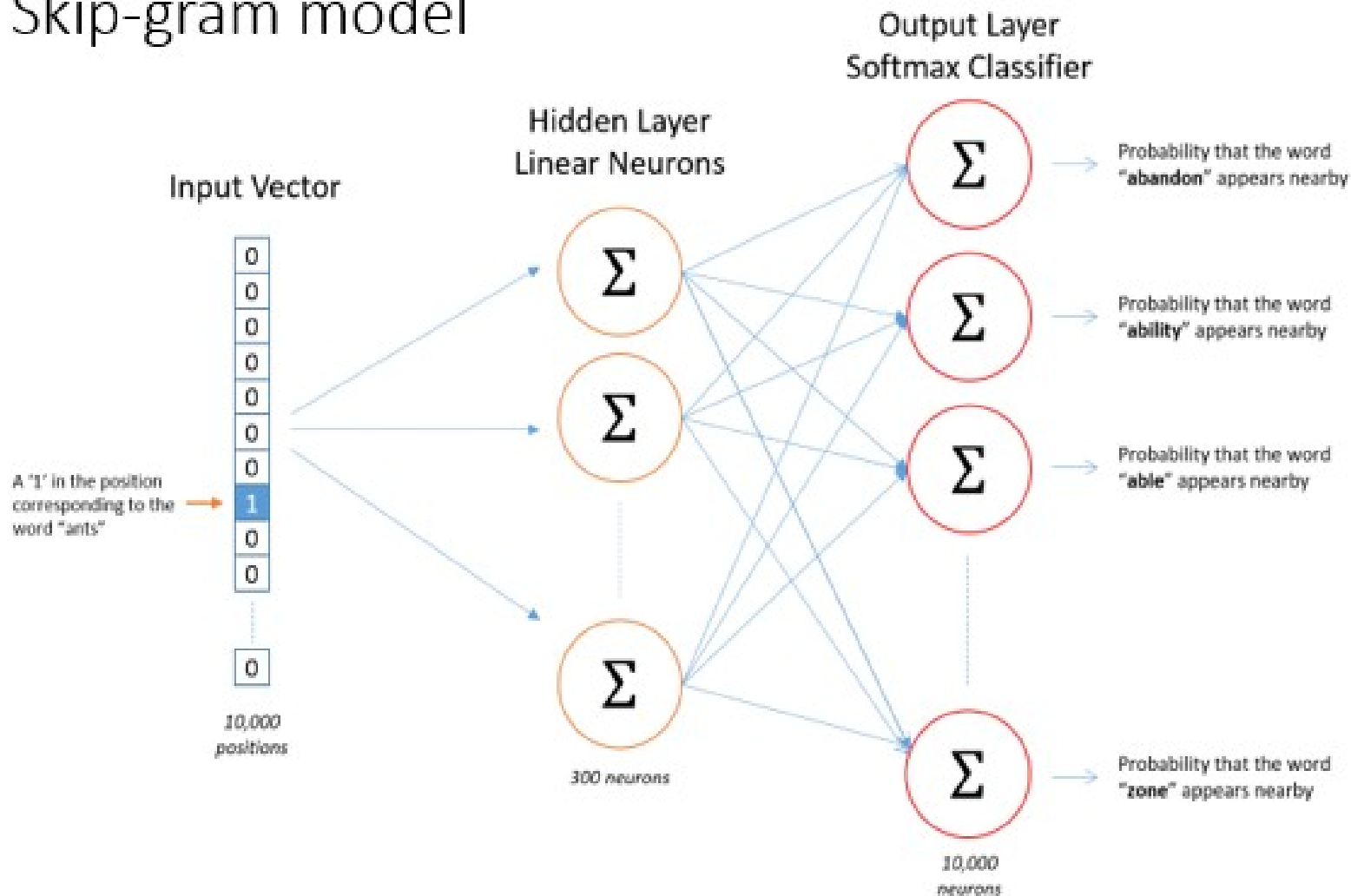
И W , и W' (представление слов контекста) можно рассматривать как представления слов. Но word2vec - W

Вектора word2vec

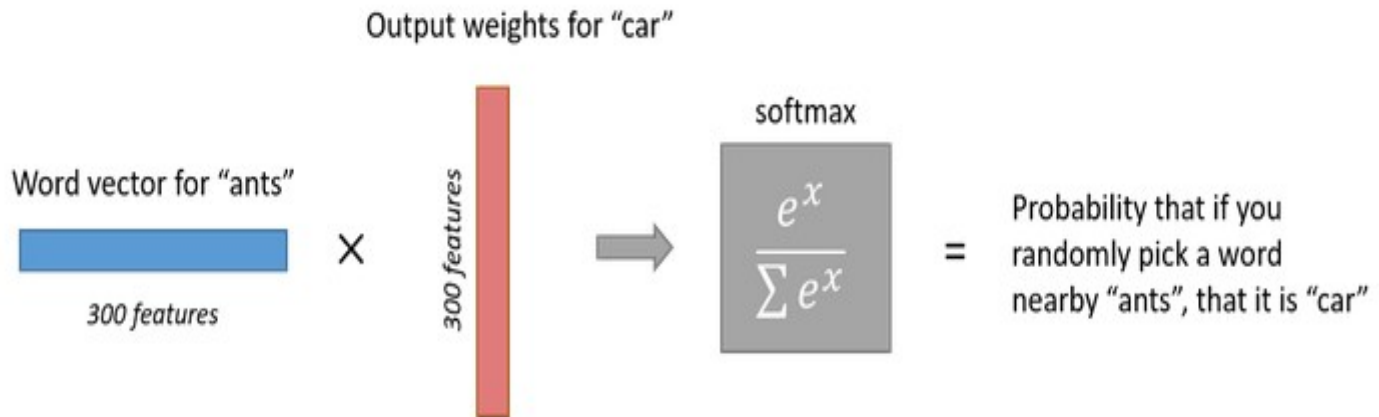
- Для каждого слова порождаются два вектора
 - Вектор слова как целевого
 - Вектор слова как контекст
 - Это разные вектора
 - Фактически мы хотим, чтобы вектор целевого слова был похож на вектора слов в контексте.

Word2vec

Skip-gram model



Предсказание слова car около слова ants



Большие матрицы, долгая обработка

- Подходы
 - Subsampling frequent words
 - Для the – слишком много контекстов,
 - The – мало что говорит о соседних словах
 - Решение
 - Слова выкидываются из текста с вероятностью пропорциональной их частоте
 - Negative sampling (негативное сэмплирование)

Негативное сэмплирование

- Обучение по каждому примеру требует пересчета весов для всех слов в выходном слове
- Идея: выбрать некоторое количество (например, 5) негативных слов (т.е. тех которых нет в контексте) и только для них перестроить веса (и также веса пересчитываются для положительных слов)
- Негативные слова выбираются с вероятностью, связанной с их частотой. В реализации word2vec – это выглядит так:

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^n \left(f(w_j)^{3/4} \right)}$$

Реализации word2vec

- Исходный код:
 - <https://github.com/tmikolov/word2vec>
- Gensim
 - <https://radimrehurek.com/gensim/models/word2vec.html>
- Есть реализации в пакетах нейронных сетей (Torch, TensorFlow, Theano)
- Есть уже насчитанные модели
 - Для русского языка (rusvectors.org)

МОДЕЛИ

В настоящий момент вы можете скачать следующие модели (жирным выделены модели, доступные для выбора в веб-интерфейсе):

Идентификатор ▲ ▼	Скачать ▲ ▼	Корпус ▲ ▼	Размер корпуса ▲ ▼	Объём словаря ▲ ▼	Частотный порог ▲ ▼	Target ▲ ▼	Al ▲
ruscorpora_upos_skipgram_300_5_2018	191 Мбайт	НКРЯ	250 миллионов слов	195 071	20	Universal Tags	Сс Sk
ruwikiruscorpora_upos_skipgram_300_2_2018	376 Мбайт	НКРЯ и Википедия за декабрь 2017	600 миллионов слов	384 764	40	Universal Tags	Сс Sk
news_upos_cbow_600_2_2018	547 Мбайт	Русскоязычные новости, с сентября 2013 до ноября 2016	почти 5 миллиардов слов	289 191	200	Universal Tags	Сс Ве Wi
araneum_upos_skipgram_300_2_2018	192 Мбайта	Araneum Russicum Maximum	около 10 миллиардов слов	196 620	400	Universal Tags	Сс Sk
araneum_none_fasttextcbow_300_5_2018	1 Гбайт	Araneum Russicum Maximum	около 10 миллиардов слов	195 782	400	Нет	fas CE (3-г
araneum_none_fasttextskipgram_300_5_2018	675 Мбайт	Araneum Russicum Maximum	около 10 миллиардов слов	195 782	400	Нет	fas Sk (3-г
ruwikiruscorpora-nobigrams_upos_skipgram_300_5_2018	385 Мбайт	НКРЯ и Википедия за декабрь 2017 (без склеивания биграмм)	600 миллионов слов	394 332	40	Universal Tags	Сс Sk

Сайт rusvectors.ru:

СХОДСТВО ВЕКТОРОВ СЛОВ

Semantic associates for *стол* (ALL)

Ruscorpora and Russian Wikipedia

1. столик 0.679
2. табурет 0.526
3. табуретка
0.515
4. подоконник
0.501
5. диван 0.491
6. стул 0.484
7. кровать 0.476
8. тумбочка 0.447
9. парта 0.439
10. кушетка 0.428

Ruscorpora

1. столик 0.794
2. подоконник
0.642
3. табуретка
0.637
4. табурет 0.623
5. диван 0.582
6. кровать 0.573
7. стул 0.570
8. кушетка 0.561
9. тумбочка 0.561
10. кресло 0.552

Web corpus

1. столик 0.637
2. стул 0.570
3. табурет 0.554
4. поднос 0.525
5. тумбочка 0.517
6. табуретка
0.497
7. обеденный
0.490
8. кушетка 0.482
9. кресло 0.479
10. сервировочный
0.470

Word Analogies

Test for linear relationships, examined by Mikolov et al. (2014)

a:b :: c:?



$$d = \arg \max_x \frac{(w_b - w_a + w_c)^T w_x}{\|w_b - w_a + w_c\|}$$

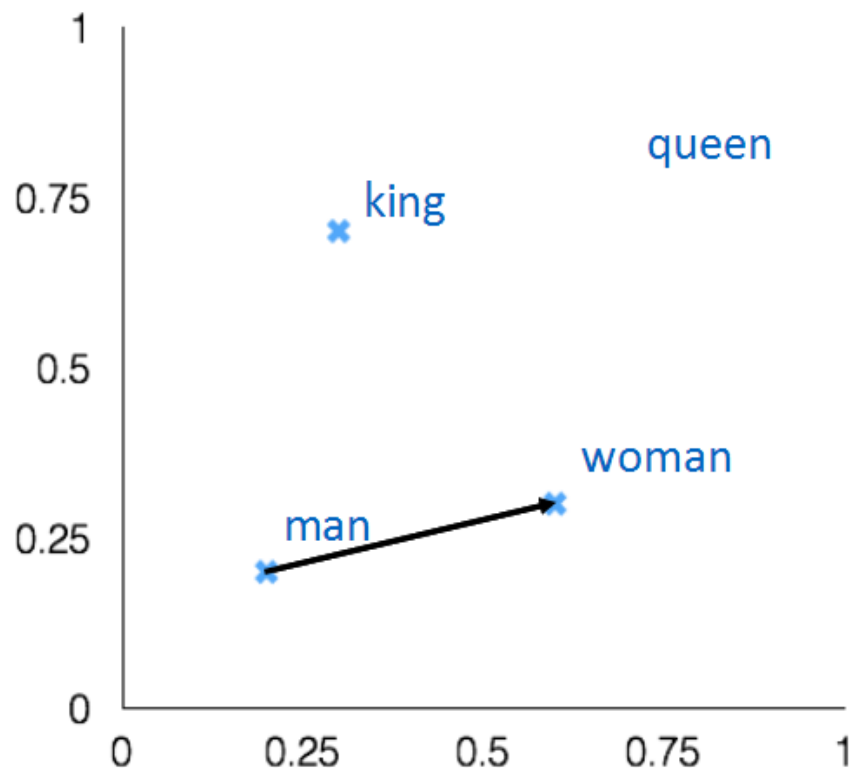
man:woman :: king:?

king [0.30 0.70]

man [0.20 0.20]

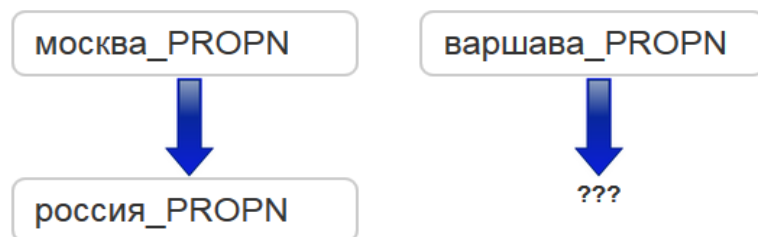
woman [0.60 0.30]

queen [0.70 0.80]








Семантический калькулятор

Вы можете вычислять отношения. Например, «**найти слово D, связанное со словом C таким же образом, как слово A связано со словом B**». Таким образом можно определять семантические связи между понятиями. В форме ввода приведен пример: какое слово относится к слову «**лондон**», так же, как «**россия**» относится к «**москве**»? Ответ — «**великобритания**»: Лондон столица Великобритании, а Москва — столица России. [Подробнее...](#)



Новостной корпус

1. **польша** 0.41 
2. **белоруссия** 0.39 
3. **страна** 0.39 
4. **германия** 0.38 
5. **европа** 0.38 

Выберите модель:

☒ Новостной корпус ☐ Araneum fastText ☐ НКРЯ и Wikipedia ☐ НКРЯ

Показывать только:

☐ Прилагательные ☐ Имена собственные ☐ Глаголы ☐ Существительные ☐ Наречия ☒ Все части речи

Вычислить!

Вы также можете попробовать более сложные операции над векторами, чем простое решение пропорции.

Введите в «**положительную**» и «**отрицательную**» формы не более 10 слов через пробел. *RusVectōrēs* сложит вектора положительных слов и вычитет из них отрицательные. Затем он выдаст слова, наиболее близкие к получившемуся вектору. Если вы оставите отрицательное поле пустым, *RusVectōrēs* просто найдет центр лексического кластера, образованного положительными словами.

+

телефон маленький

-

Выберите модель:

☒ Новостной корпус ☐ Araneum fastText ☐ НКРЯ и Wikipedia ☐ НКРЯ

Показывать только:

☐ Прилагательные ☐ Имена собственные ☐ Глаголы ☐ Существительные ☐ Наречия ☒ Все части речи

Вычислить!

Сексизм нейронных языковых моделей

RusVectōrēs

Похожие слова

Визуализации

Калькулятор

Различные операции

Модели

О проекте

Контакты

RU/EN

Вы можете вычислять отношения. Например, «**найти слово D, связанное со словом С таким же образом, как слово А связано со словом В**». Таким образом можно определять семантические связи между понятиями. В форме ввода приведен пример: какое слово относится к слову «**лондон**», так же, как «**россия**» относится к «**москве**»? Ответ — «**великобритания**»: Лондон столица Великобритании, а Москва — столица России. [Подробнее...](#)

мужчина_NOUN



программист_NOUN

женщина_NOUN



???

Новостной корпус

1. **швея** 0.43
2. **логист** 0.41
3. **компьютерщик** 0.40
4. **тестировщик** 0.39
5. **продажник** 0.39

Выберите модель:

☒ Новостной корпус ☒ Araneum fastText ☐ НКРЯ и Wikipedia ☐ НКРЯ

Показывать только:

☐ Прилагательные ☐ Имена собственные ☐ Глаголы ☐ Существительные ☐ Наречия
☒ Все части речи

Вычислить!

Araneum fastText

1. **программистка** 0.83
2. **програмист** 0.82
3. **программер** 0.74
4. **программинг** 0.72
5. **программистский** 0.67

Вы также можете попробовать более сложные операции над векторами, чем простое решение пропорции.

Введите в «**положительную**» и «**отрицательную**» формы не более 10 слов через пробел. *RusVectōrēs* сложит вектора положительных слов и вычитет из них отрицательные. Затем он выдаст слова, наиболее близкие к получившемуся вектору. Если вы оставите отрицательное поле пустым, *RusVectōrēs* просто найдет центр лексического кластера, образованного положительными словами.

+

телефон маленький

-

Выберите модель:

☒ Новостной корпус ☒ Araneum fastText ☐ НКРЯ и Wikipedia
☐ НКРЯ

Показывать только:

☐ Прилагательные ☐ Имена собственные ☐ Глаголы
☐ Существительные ☐ Наречия ☒ Все части речи

Вычислить!

Слова, выделенные **зеленым**, являются высокочастотными (доля слова в корпусе выше 0.00001); слова, выделенные **красным**, являются низкочастотными (доля слова в корпусе ниже 0.0000005).

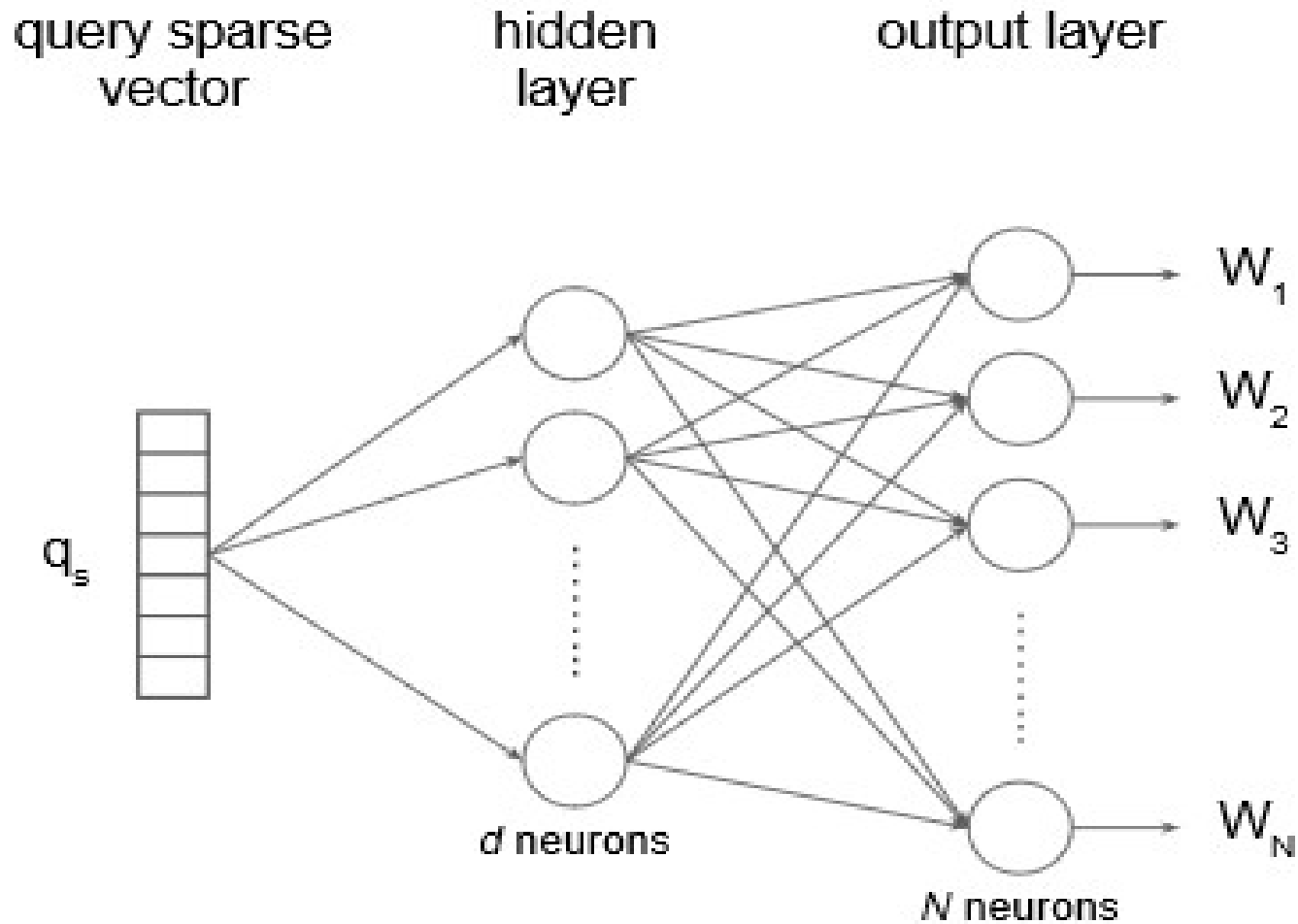
Датасеты для тестирования по аналогии

- Аналогии “a is to a * as b is to b *”
 - MSR’s analogy dataset
 - 8000 морфосинтаксических аналогий
 - good is to best as smart is to smartest
 - Google’s analogy dataset
 - 19544 морфосинтаксических и семантических аналогий
 - Paris is to France as Tokyo is to Japan

Расширение запроса: Relevance-based Word Embeddings (Zamani, Croft, 2017)

- Появление векторных представлений слов (word2vec) привело к использованию их для расширения запроса в информационном поиске
- Однако цель обучения таких представлений слов направлена на сходство слов в небольшом контексте, что не соответствует целям информационного поиска
- Нужно создать специализированные модели для обучения векторных представлений слов для целей расширения запроса

Архитектура сети



Архитектура сети

- Вход – запрос в виде обычного (sparse) вектора длины N , где N – размер словаря, т.е. стоят 1 в местах соотв. слов
- Скрытый слой преобразует исходный вектор в dense вектор, т.е. вектор низкой размер|

$$\vec{q} = \vec{q}_s \times \mathcal{W}_Q$$

- Выходной слой сети должен предсказывать слова релевантные запросу $\sigma(\vec{q} \times \mathcal{W}_w + b_w)$

Обучение

- Миллион разных запросов из AOL лога запросов (2006)
- Рассматривалось 10 первых документов выдачи
- Т.е задача нейронной сети была предсказывать распределение слов в этих первых документах

Table 1: Collections statistics.

ID	collection	queries (title only)	#docs	avg doc length	#qrels
AP	Associated Press 88-89	TREC 1-3 Ad-Hoc Track, topics 51-200	165k	287	15,838
Robust	TREC Disks 4 & 5 minus Congressional Record	TREC 2004 Robust Track, topics 301-450 & 601-700	528k	254	17,412
GOV2	2004 crawl of .gov domains	TREC 2004-2006 Terabyte Track, topics 701-850	25m	648	26,917
ClueWeb	ClueWeb 09 - Category B	TREC 2009-2012 Web Track topics 1-200	50m	1506	18,771

Результаты

Collection	Metric	MLE	word2vec		GloVe		Rel.-based Embedding	
			external	target	external	target	RLM	RPE
AP	MAP	0.2197	0.2399	0.2420	0.2319	0.2389	0.2580 ⁰¹²³⁴	0.2543 ⁰¹²³⁴
	P@20	0.3503	0.3688	0.3738	0.3581	0.3631	0.3886 ⁰¹²³⁴	0.3812 ⁰³⁴
	NDCG@20	0.3924	0.4030	0.4181	0.4025	0.4098	0.4242 ⁰¹²³⁴	0.4226 ⁰¹²³⁴
Robust	MAP	0.2149	0.2218	0.2215	0.2209	0.2172	0.2450 ⁰¹²³⁴	0.2372 ⁰¹²³⁴
	P@20	0.3319	0.3357	0.3337	0.3345	0.3281	0.3476 ⁰¹²³⁴	0.3409 ⁰²⁴
	NDCG@20	0.3863	0.3918	0.3881	0.3918	0.3844	0.3982 ⁰¹²³⁴	0.3955 ⁰
GOV2	MAP	0.2702	0.2740	0.2723	0.2718	0.2709	0.2867 ⁰¹²³⁴	0.2855 ⁰¹²³⁴
	P@20	0.5132	0.5257	0.5172	0.5186	0.5128	0.5367 ⁰¹²³⁴	0.5358 ⁰¹²³⁴
	NDCG@20	0.4482	0.4571	0.4509	0.4539	0.4485	0.4576 ⁰²³⁴	0.4557 ⁰²⁴
ClueWeb	MAP	0.1028	0.1033	0.1033	0.1029	0.1026	0.1066 ⁰¹²³⁴	0.1031
	P@20	0.3025	0.3040	0.3053	0.3033	0.3048	0.3073	0.3030
	NDCG@20	0.2237	0.2235	0.2252	0.2244	0.2244	0.2273 ⁰¹	0.2241

Подход FastText

- В качестве векторного представления для слова берется среднее из представлений входящих в него n -грамм
 - то есть слово «ребенок» - это некоторая усредненная сумма векторов «ре», «еб», «бе», «ен» и т.д. (пример для биграмм)
- Плюсы подхода:
 - Символьные n -граммы встречаются чаще, чем слова целиком.
 - Учет похожесть контекстов слов с одинаковыми аффиксами.
 - n -граммы из аффиксов «выловят» семантику и синтаксис,
 - n -граммы из корней – лексику.

Предобученные вектора FastText

<https://fasttext.cc/docs/en/crawl-vectors.html>

Нейронные сети в задачах информационного поиска

- Векторные представления слов (embeddings), обученные на основе нейронных сетей дают возможность снижения проблемы различия между запросом и документом
 - Нейронная сеть DSSM в глобальных поисковых системах
 - От векторной модели документов к векторным моделям слов и других сущностей (графов)
- Задачи классификации сейчас – лучшие результаты получаются нейронными сетями
- Вопросно-ответный поиск – используется преобразование вопроса и ответа в единое векторное пространство в процессе обучения нейронной сети

Задание-10

- Пусть есть коллекция текстов из 4 слов – a, b, c, d
- Рассмотрим последовательность слов " a b c"
- Т.е. нужно предсказать слово b
 - сжимаем в вектор длины 2

Задание-10

(продолжение)

- Случайная инициализация матрицы W задана
 - 0 1 2 2
 - 1 2 2 0
- Случайная инициализация матрицы W' такая
 - 0 1 1 2
 - 1 1 2 0
- Какие вероятности предсказания слов получатся после применения softmax

Задание 11. Синонимайзинг

- Насколько близости по векторным представлениям слов могут быть использованы для синонимической замены.
- Выбрать какой-нибудь корпус в интерфейсе
 - <https://rusvectors.org/ru/associates/>
- Синонимайзинг
- Пройдите по словам (существительное, прилагательное, глагол) из ваших трех фактов и замените на первый вариант близкого по модели слова. В какой доле случаев замены действительно будут синонимичными и смысл фактов не изменится?