

Word2vec. Задачи NLP.

Маша Шеянова, masha.shejanova@gmail.com

Эмбеддинги слов

Дистрибутивная семантика

Что мы хотим:

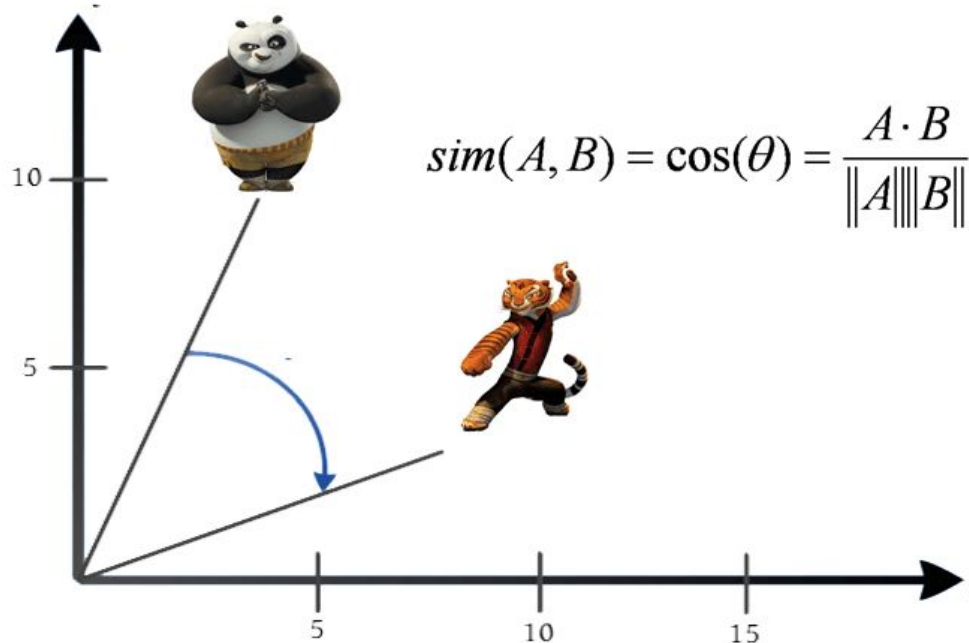
- формальный способ считать лексическую близость
- глобально: научить компьютер извлекать смыслы из текста

Как делать это автоматически?

Дистрибутивная гипотеза: значения слов полностью определяются их контекстами. Слова с похожими типичными контекстами имеют схожее значение.

Как найти, насколько близки слова?

Cosine Similarity



- надо найти способ превратить слова в вектора так, чтобы они отражали **контекст**
- найти расстояние между этими векторами одним из способов

Источник картинки.

Как сделать из слов вектора?

Итак, основная идея — **учитывать контекст**. Но как? А вот про это есть большая наука.

Самый простой-наивный метод — **счётный**. Идея: для каждого слова возьмём ближайшие в некотором окне (например, -5 +5). Сделаем такой же мешок слов, как делали для документов с *CountVectorizer*, но для контекста.

Плюсы: легко и быстро.

Минусы: для большого корпуса — очень большие вектора.

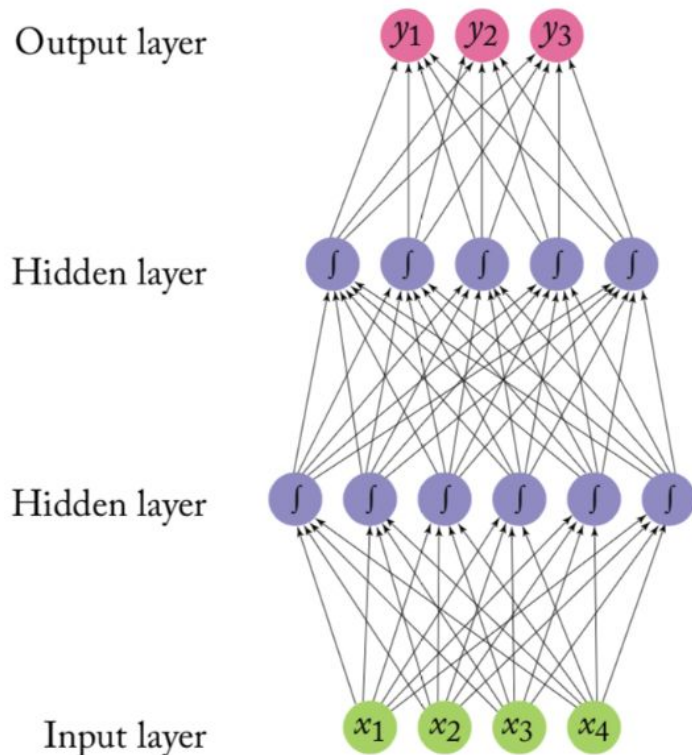
Пример таблички с контекстами

По итогу у овощей будут контексты, похожие друг на друга, а у животных — друг на друга.

	редис	картошка	кот	...	собака
редис	-	5	1		0
картошка	5	-	0		1
кот	1	0	-		6
...					
собака	0	1	6		-

Как устроена нейросеть

нейросеть как направленный граф



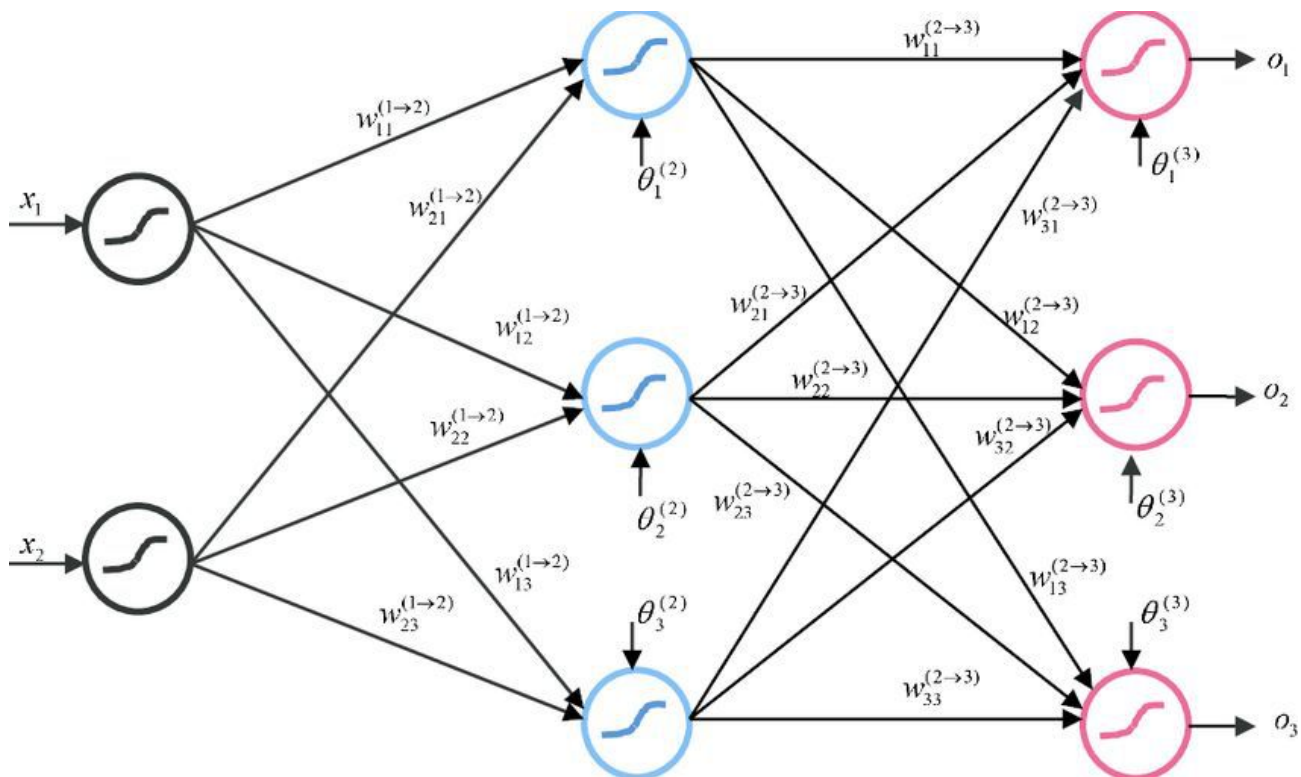
На входе — вектор признаков.

На каждой стрелочке — какие-то коэффициенты.

На выходе — вектор вероятностей того или иного класса.

“Нейрон” == один кружочек == функция от выдачи предыдущего слоя.

Нейросеть как функция



Один нейрон —
функция от **вектора**
параметров (w_{11} , w_{21}),
умноженного на
вектор объекта x (+ b).
 $f(\mathbf{w}x + b)$

Слой — функция от
матрицы параметров
* x + **вектор** b .
 $f(\mathbf{W}x + \mathbf{b})$.

Нейросеть как функция

x — входные данные (признаки); W — веса

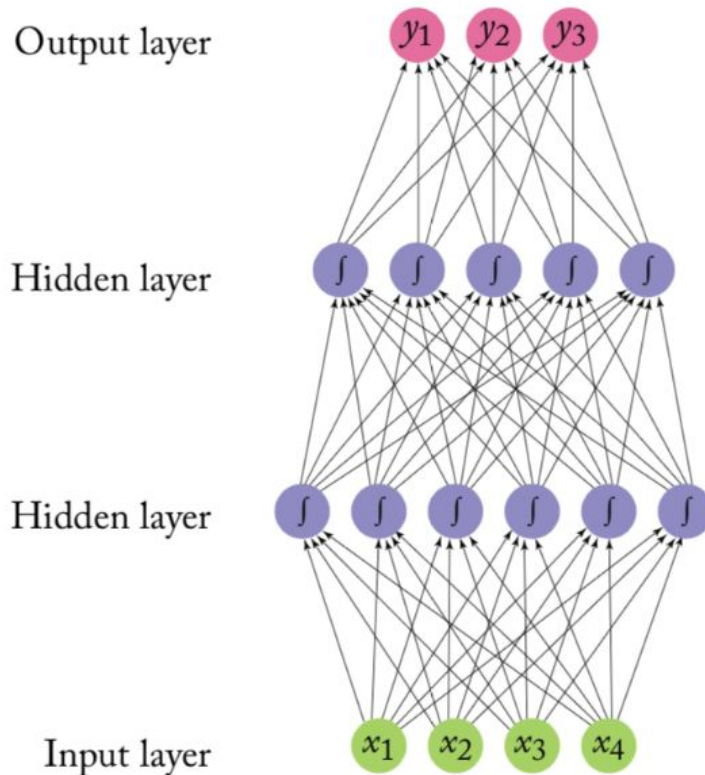
$$h_1 = f_1(W_1 * x + b_1)$$

$$h_2 = f_2(W_2 * h_1 + b_2)$$

$$y_{\text{pred}} = f_3(W_3 * h_2 + b_3)$$

$$y_{\text{pred}} = f_3(W_3 * f_2(W_2 * h_1 + b_2) + b_3)$$

$$\text{loss} = ((y_{\text{pred}} - y_{\text{true}})^2).mean()$$



Word2vec

Word2vec

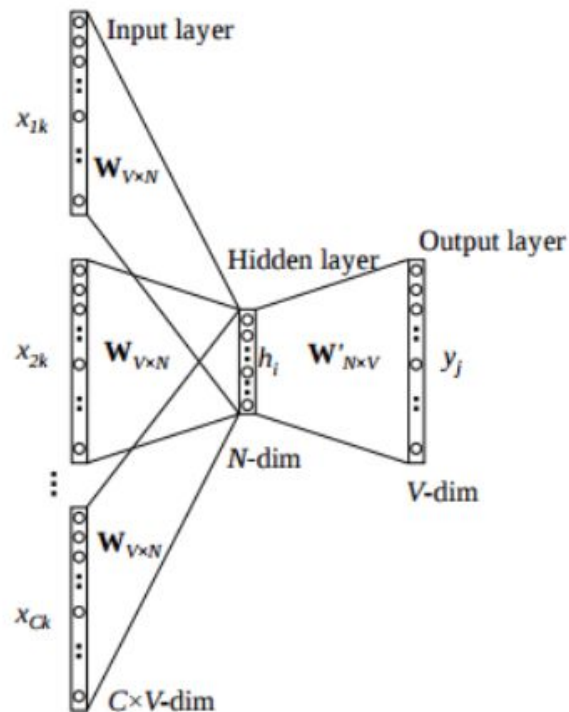
В двух словах, Word2Vec — это метод строить гораздо более компактные эмбединги с помощью нейросетей.

Методы:

- CBOW (Common Bag Of Words)
- skipgram

CBOW (common bag of words)

Источник картинки



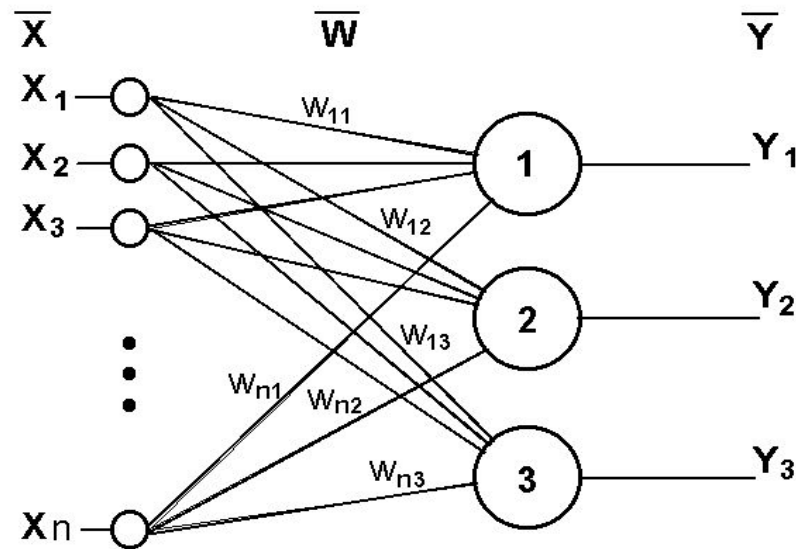
Метод CBOW пытается **предсказать слово по его контексту**. Он берёт каждое слово из контекста слова u и пытается по нему предсказать слово u .

Здесь, каждый x и каждый y — one-hot вектора, где нули везде, кроме позиции, отвечающей за слово.

А h — внутренний слой такой ширины, как мы захотим.

Откуда берутся эмбединги

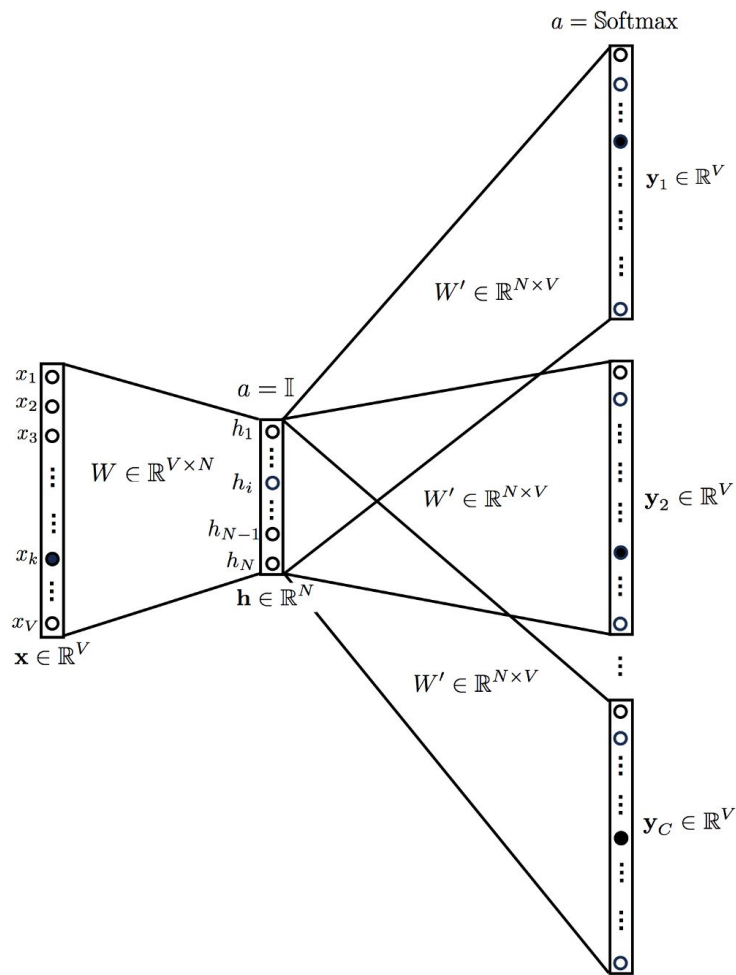
- на картинке — однослойная нейросеть, как word2vec
- веса на стрелочках обучаются (градиентным спуском) так, чтобы слова из контекста хорошо предсказывались
- каждое слово — one-hot вектор (везде нули, кроме позиции слова)
- а значит, информацию о контексте слова номер k хранят все веса на стрелочках, выходящих из узла x_k



skipgram

skipgram, в отличие от CBOW, пытается предсказывать контекст по слову.

- **Skip Gram** хорошо работает с маленьким объёмом данных и **лучше представляет редкие слова**
- **CBOW** работает быстрее и **лучше представляет наиболее частые слова**



Fasttext

Fasttext — почти то же самое, что и word2vec, но работает на уровне меньше, чем слово.

Идея такая: разбиваем каждое слово на *символьные нграммы*. Например, так:
apple → **app, ppl, ple**

Обучаем нейросетку так, чтобы получить эмбединги этих кусочков.
Финальный эмбединг слова — сумма эмбедингов его кусочков.

В чём профит? Умеем представлять даже слова, которых не было в корпусе!

Главное достоинство word2vec

Так как мы можем регулировать ширину внутреннего слоя:

- мы можем “ужать” информацию о контексте слова до его размера, эффективно используя память
- мы можем выбирать силу сжатия
- модели, обученные на больших корпусах не будут весить сильно больше

Проблемы word2vec

Невозможно установить тип семантических отношений между словами: синонимы, антонимы и т.д. будут одинаково близки, потому что обычно употребляются в схожих контекстах (например, слова **хороший** и **плохой**).

Поэтому близкие в векторном пространстве слова называют семантическими **ассоциатами**. Это значит, что они семантически связаны, но как именно — непонятно.

Rusvectors, word2vec для русского

На rusvectors можно найти слова, наиболее близкие к данному, построить семантическую пропорцию и многое другое.

The screenshot shows the Rusvectors web interface. At the top, there are two input boxes: 'человек_S' and 'кошка_S'. Below 'человек_S' is a blue arrow pointing to 'нога_S'. Below 'кошка_S' is a blue arrow pointing to '???'. To the right of these inputs is a search bar and a 'Calculate!' button. Below the inputs, there are two columns of results: 'News corpus' and 'Ruscorpора'. Each column lists five words with their similarity scores. At the bottom, there are checkboxes for 'Choose the model:' and 'Show only results which belong to:'.

человек_S

нога_S

News corpus

1. ступня 0.430
2. котенок 0.424
3. кошачий 0.409
4. пес 0.403
5. ножка 0.388

Ruscorpора

1. лапка 0.499
2. ножка 0.485
3. лапа 0.482
4. ножища 0.482
5. ножонка 0.479

Web corpus

1. лапа 0.534
2. ступня 0.519
3. колено 0.508
4. спина 0.484
5. туловище 0.472

Choose the model:

☒ Ruscorpора and Russian Wikipedia

Show only results which belong to:

☐ Nouns ☐ Verbs ☐ Adverbs

Calculate!

Choose the model:

☒ Ruscorpора and Russian Wikipedia ☒ News corpus ☒ Ruscorpора ☒ Web corpus

Где взять готовые эмбединги

Я рассказала, как обучить свои эмбединги. Но это долго, заморочно и не всегда нужно. Есть ли уже обученные эмбединги? Конечно!

[Rusvectors](#)! (для русских слов)

Задачи NLP в МО

О чём уже говорили

Класификация текста:

- spam detection
- жанры
- sentiment analysis (тональность)
- предсказание темы

Кластеризация текстов:

- новости
- topic modelling

Что ещё?

- на уровне текстов
- на уровне предложений
- на уровне слов
- speech, OCR, image captioning

Предложения

- Paraphrase
- Textual entailment
- QA systems
- machine translation

Слова

- POS-tagging
- named entity recognition

О ДЗ 2

Домашнее задание: классификация отзывов

Данные: отзывы на фильмы.

Что сделать:

- 3 балла — считать датасет, обучить на нём любой из описанных сегодня классификаторов, измерить качество
- 3 балла — перебрать как минимум 3 классификатора, найти лучший
- 2 балла (*) — попробовать разные гиперпараметры, найти лучший
- бонусный 1 балл за понятный и чистый код :)

Дедлайн: следующий четверг

Ресурсы

Почитать

- [про word2vec по-русски](#)
- [Introduction to Word Embedding and Word2Vec](#)
- [Word2Vec and FastText Word Embedding with Gensim](#)
- [word2vec tutorial на kaggle](#)