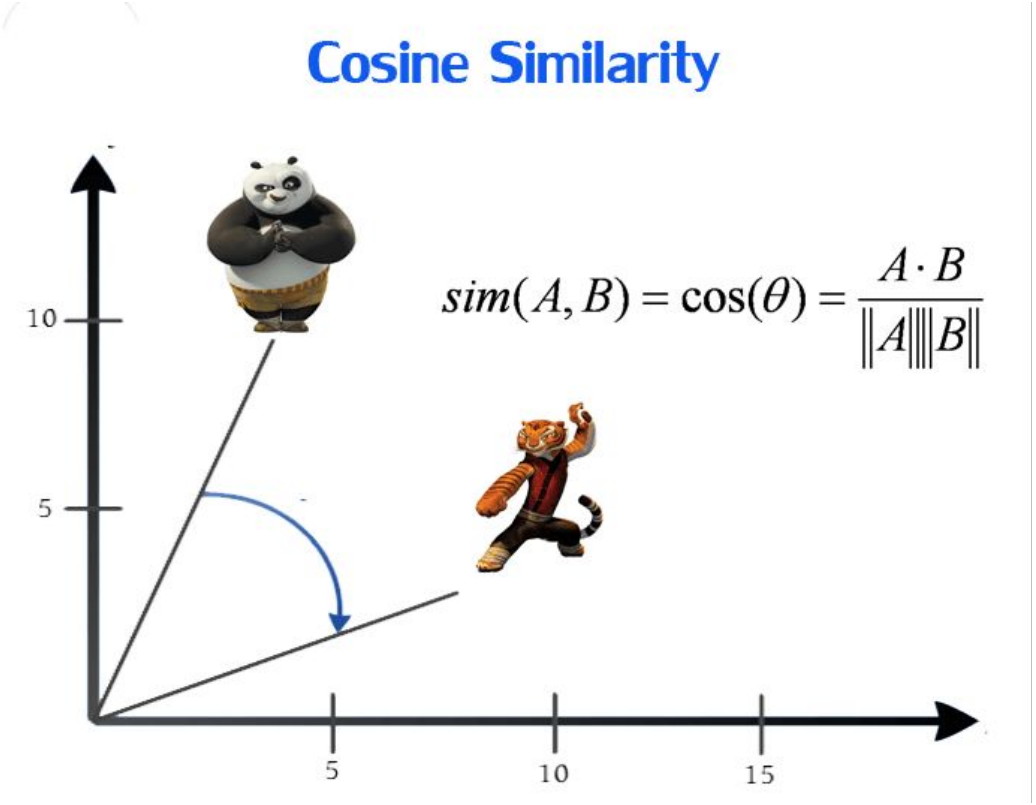
Эмбеддинги и Fasttext

Маша Шеянова, masha.shejanova@gmail.com

Как найти, насколько близки слова?



- 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?

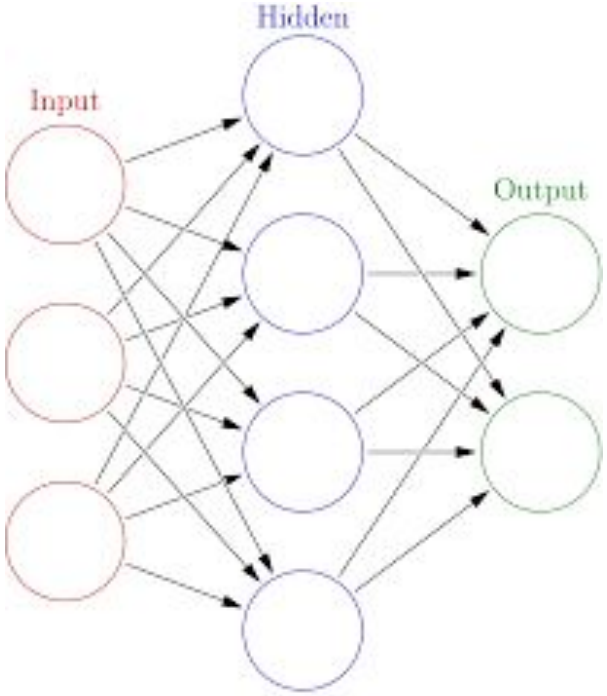
Итак, основная идея — **учитывать контекст**. Но как? А вот про это есть большая наука.

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 (CountVectorizer, TfidfVectorizer). Можно делать “скользящее окно”.

Плюсы: легко и быстро.

Минусы: для большого корпуса — очень большие вектора.

нейросеть in a nutshell



На входе — вектор признаков.

На каждой стрелочке — какие-то коэффициенты.

На выходе — вектор вероятностей того или иного класса.

“Нейрон” == один кружочек.

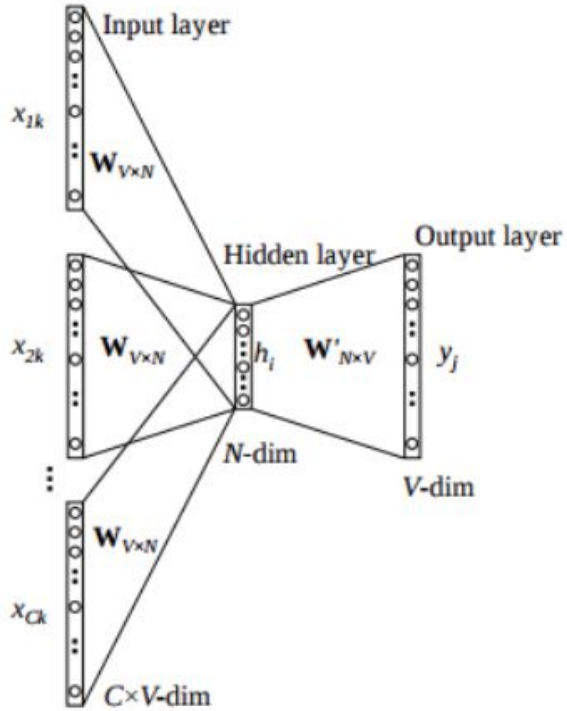
Word2vec

In general, Word2Vec — это метод строить гораздо более компактные эмбеддинги с помощью нейросетей.

Методы:

- CBOW (Common Bag Of Words)
- skipgram

CBOW (common bag of words)



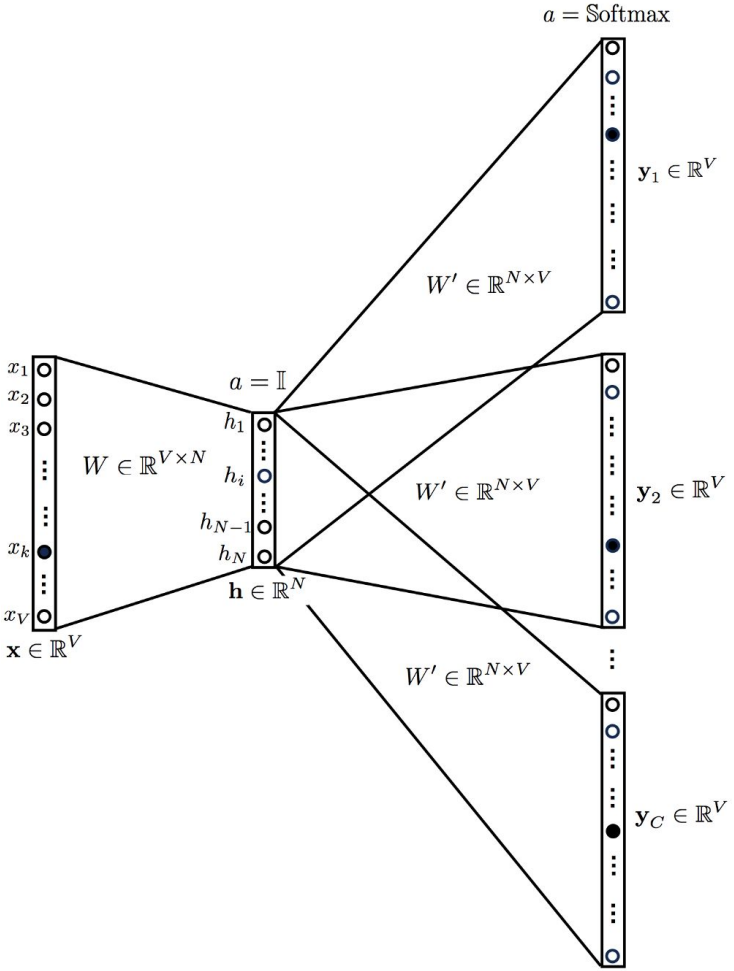
Источник картинки

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

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**



Веб-интерфейсы и ресурсы про word2vec

rusvectores — для русского

tutorial по word2vec — для английского

хорошее объяснение про word2vec и fasttext (англ)

word2vec tutorial на kaggle

Fasttext

Fasttext — почтиии 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 эмбеддинги этих кусочков. Финальный эмбеддинг слова — сумма эмбеддингов его кусочков.

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

О проекте

Что обязательно должно быть в проекте

Проект — это jupyter-notebook со спеллчекером, у которого:

- с хорошо оформленным кодом
  - всё, что можно вынести в отдельную функцию, — в отдельной функции
  - про самые главные или нетривиальные шаги описано, что происходит
  - переменные названы понятно :))))
- если в самом первом слове опечатка, его тоже надо исправить
- работает на русском языке
- анализ ошибок: придумайте пример, где ваш спеллчекер ошибётся; предположите, как это можно было бы улучшить

Будет плюсом, например

- придумать умный способ сократить количество кандидатов на исправление
- придумать, как сделать языковую модель (поиск вероятностей) получше
- добавить ещё какой-то интересный функционал
- любой глубокий анализ
- любой полезный креатив :)