Введение в NLP и обработку звука

Лектор: Ксения Мелещеня

План лекции

NLP

Кодирование текстов

Основные модели

LLM

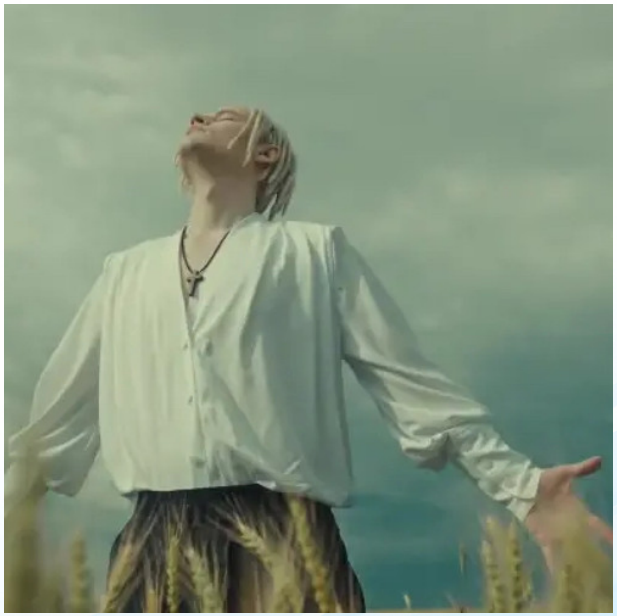
Natural Language Processing (NLP) — область машинного обучения, посвященная обучению моделей для понимания естественного языка.

- NLP позволяет применять методы машинного обучения для текста и речи.
- Задачи NLP могут включать в себя работу не только с текстом, но и с непрерывными данными — звуком и видео.

Пример решения задачи Image captioning

По картинке надо дать её текстовое описание

Шапка, мандарин, макияж.



Человек на природе

счастливый

Пример решения задачи классификации текста

По комментарию в интернете надо определить его токсичность

comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
Тебя следует уволить, ты тупица и трус, слишком ленивый, чтобы провести исследование. Меня тошнит от мысли, что такие люди, как ты, существуют в этом мире.	1	0	0	0	1	0
О, я не знал, спасибо	0	0	0	0	0	0

План лекции

NLP

Кодирование текстов

Основные модели

LLM

Кодирование текстов

Рассмотрим задачу классификации:

по некоторому набору характеристик отеля определить, сколько у него звезд (5 классов).

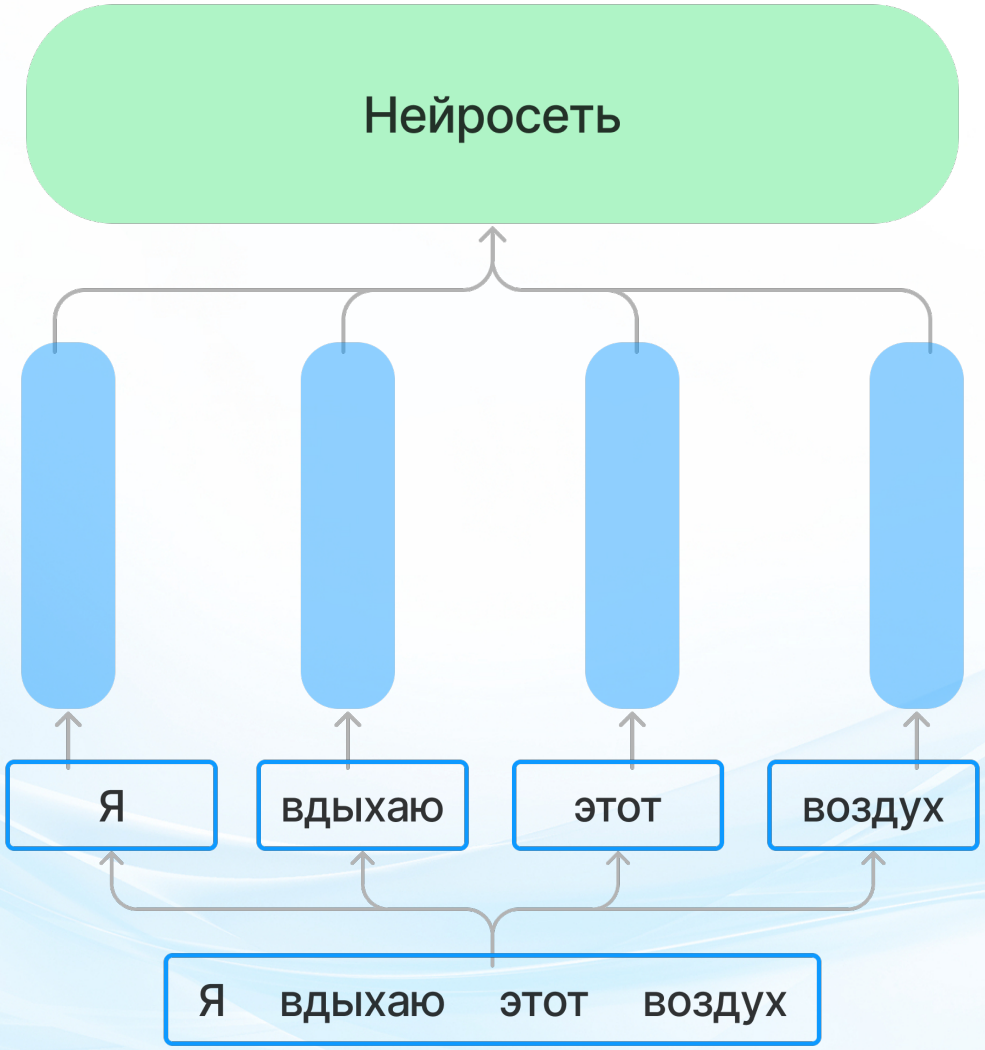
Данные

Среди признаков есть текстовое описание отеля, которое содержит много информации.

Проблема

Как использовать текстовое описание в модели?

Кодирование текстов



Эмбеддинги слов

Постановка задачи

Пусть V — словарь из всех слов: $V = \{w_i\}_{i=1}^{n}$.

Необходимо построить отображение $f\colon V \longrightarrow \mathbb{R}^d$,

где d — гиперпараметр.

Вектор $f(w)$ называется **эмбеддингом** слова w.

Ограничение: $f(w_1) = f(w_2) \implies w_1 = w_2$.

Кроме этого хотим:

- $f(w_1) \neq f(w_2) \implies w_1 \neq w_2$;
- $f(w_1) \approx f(w_2)$, если w_1 и w_2 похожи по смыслу;
- $f(w_1) \not\approx f(w_2)$, если w_1 и w_2 не похожи по смыслу.

One-hot векторы

Пусть V — словарь из всех слов: $V = \{w_i\}_{i=1}^{n}$.
Каждому слову поставим в соответствие вектор из $\{0, 1\}^{|V|}$,
где 1 стоит на позиции i для слова w_i.

Примеры

$\texttt{motel} = (0, ..., 0, 1, 0, ..., 0)$
$\texttt{hotel} = (0, ..., 0, 0, 1, ..., 0)$

Минусы

- Большой размер векторов.
- Векторы не содержат информацию о значении слов.
- Все векторы ортогональны, включая векторы схожих по смыслу слов, то есть нет понятия сходства векторов.

Bag of Words

Bag of Words (BOW)

Метод построения эмбеддинга для всего предложения.

Эмбеддинг текста — сумма one-hot векторов входящих в него слов.

Пример

$V = \{\texttt{it}, \texttt{was}, \texttt{the}, \texttt{worst}, \texttt{of}, \texttt{times}, \texttt{age}, \texttt{wisdom}, \texttt{foolishness}\}$

$\texttt{It was the worst of times} = (111111000)$

$\texttt{It was the age of wisdom} = (111010110)$

$\texttt{It was the age of foolishness} = (111010101)$

Bag of Words

Плюсы

- Предложения, содержащие одинаковые слова, имеют схожие векторы.

Минусы

- Большой размер векторов.
- Схожие по смыслу слова рассматриваются как разные.
- Векторы очень разрежены (sparse).

Этим методом можно строить эмбеддинг слова, рассматривая его как набор букв, но такие векторы не очень информативны.

Дистрибутивная гипотеза

Что означает слово **пакс**?

Примеры использования

- Ведаю правду, один и тот же **пакс** едет сегодня на Лидере, завтра на 306, послезавтра на Яндекс....и так по кругу.
- Это был очень долгий и очень скучный рейс с идеальными **паксами**.
- Это все городская легенда, что после 'кола' система не сведет больше с этим **паксом**.
- Некоторые из **паксов** достали мобильники и стали снимать весь процесс.
- Компания экономит на незадачливых **паксах**?

Можно понять, что **пакс** — разговорное название пассажира.

Примеры взяты из @UznalSlovo

Дистрибутивная гипотеза

"*Скажи мне кто твой друг и скажу кто ты*" (Стэтхем, 1999)

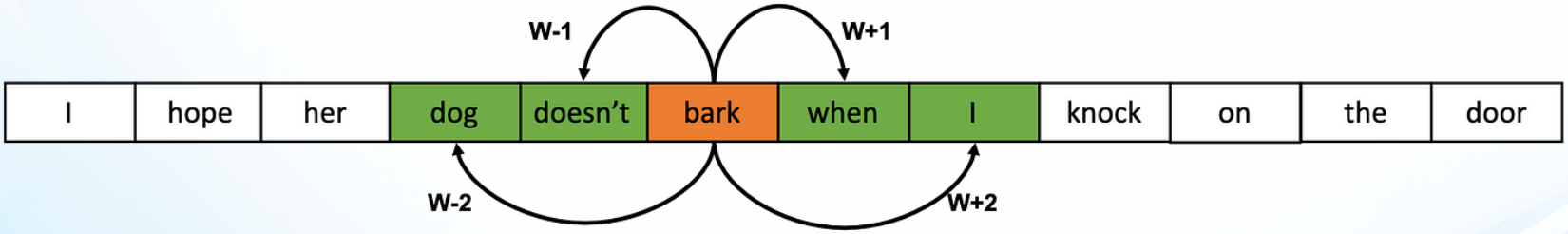
Дистрибутивная гипотеза:

"*Слова, встречающиеся в схожих контекстах, похожи по смыслу*."

Word2Vec

Идея: Давайте обучим модель из слова получать вектор!

Из самого слова извлечь информацию сложно, возьмём контекст.
Под *контекстом* понимается окно некоторого размера вокруг слова.
Порядок при этом не учитывается.

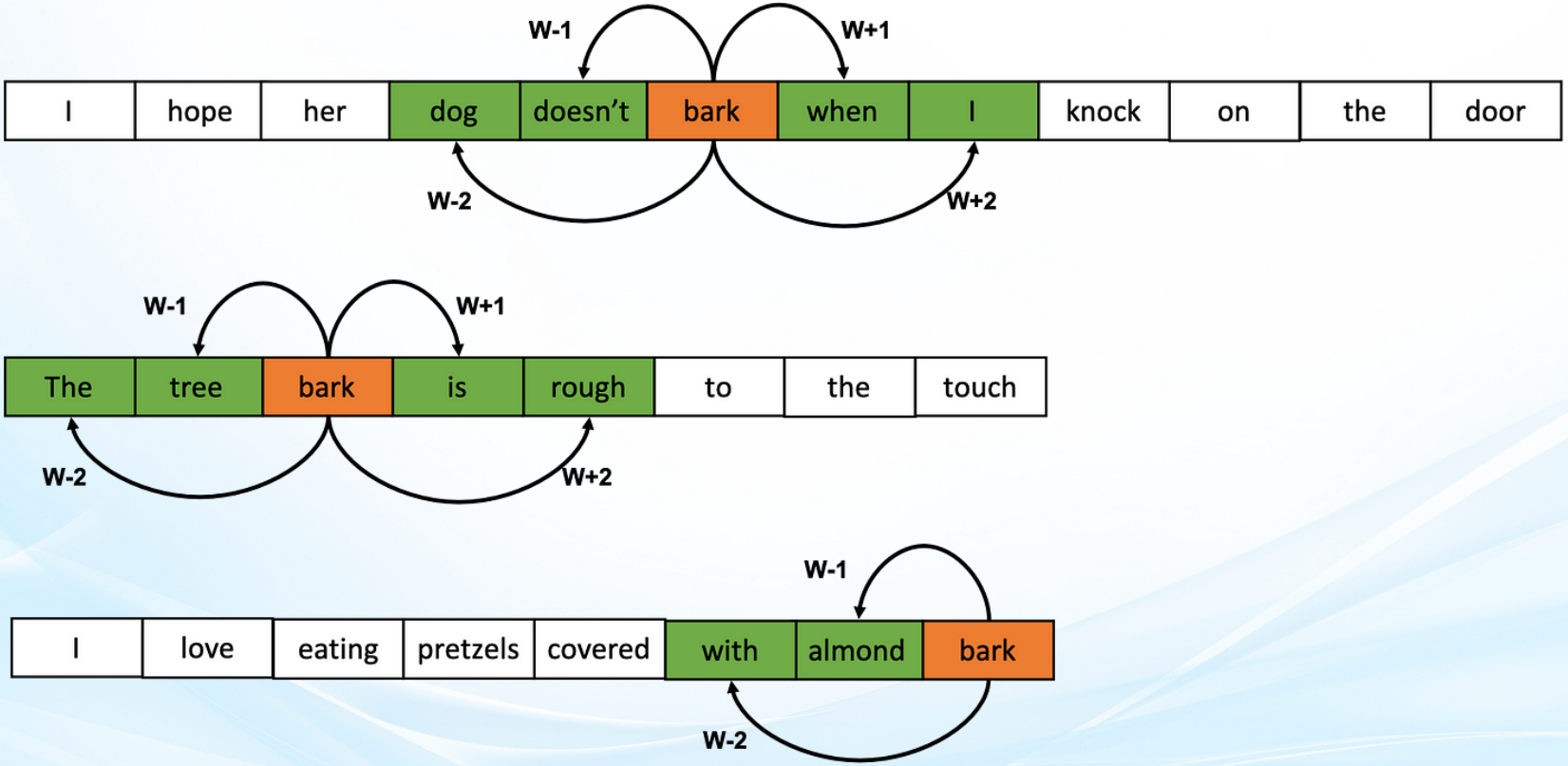


Пусть $P_c(w)$ — вероятность встретить слово w
в контексте центрального слова c.

Вероятность $P_c(w)$ будем предсказывать с помощью
некоторой нейронной сети.

Word2Vec

Различные варианты



Word2Vec

Закодируем каждое слово w one-hot вектором.

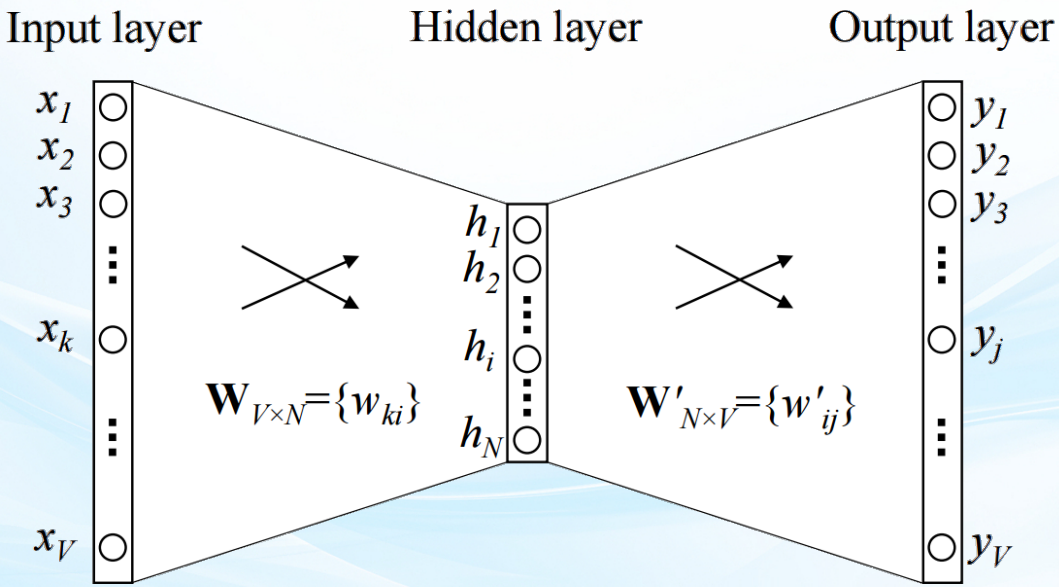
Модель:

двухслойная нейросеть без нелинейностей, с softmax на конце.

Вход модели: one-hot вектор для слова w.

Выход: для каждого слова из словаря v вер-ть быть в контексте w.

Таргет: one-hot вектор для слова v из контекста w.



Word2Vec

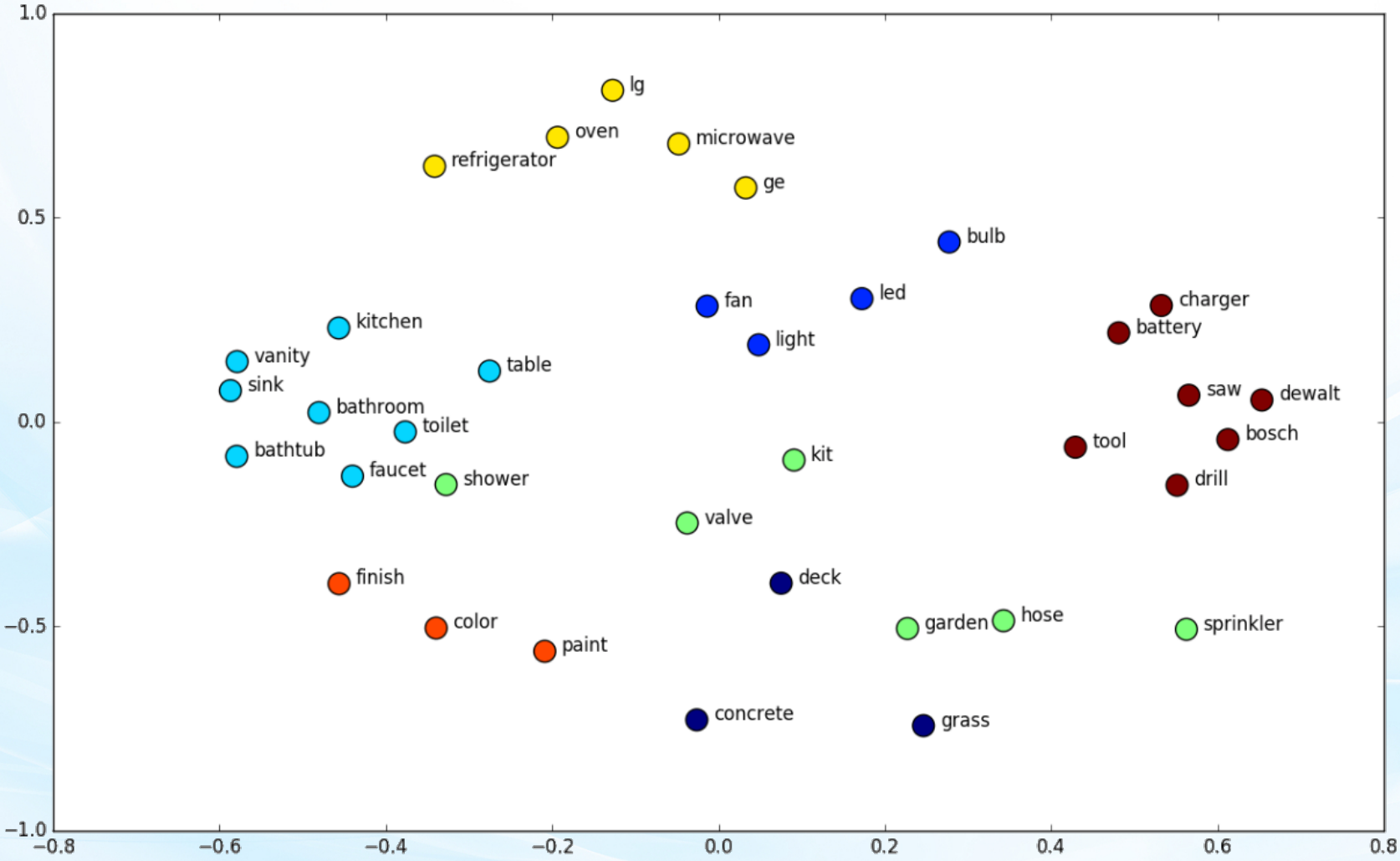
В чистом виде описанная архитектура редко используется.

Она принимает для обучения объекты вида
<*центральное слово*, *слово из его контекста*>,
но слов в контексте центрального слова много.

Про другие архитектуры Word2Vec вы узнаете на DS-потоке!

Word2Vec

Пример работы с переводом слов в двумерное пространство



План лекции

NLP

Кодирование текстов

Основные модели

LLM

Основные модели

Что мы уже умеем?

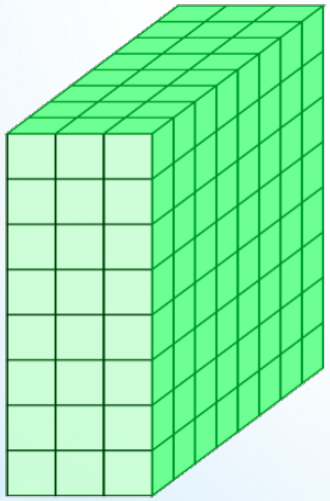
Строить векторные представления слов и текстов —
из слова получать его числовое представление,
которое можно использовать в существующих моделях ML.

Теперь можем приступать к самому интересному.
Обучим нейросеть решать задачи!

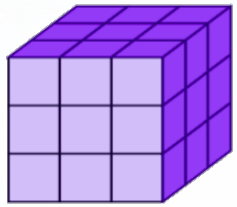
Проблема:

полносвязная нейронная сеть не учитывает природу текста —
работает с ним как с набором векторов,
а не как с последовательностью.

Напоминание: 2D-свертка

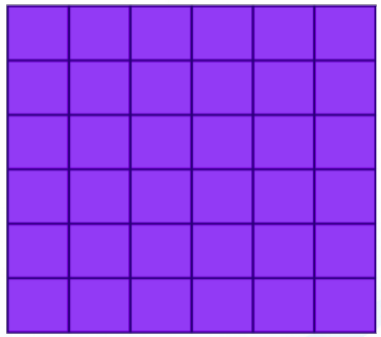


изображение 8 x 8 x 3



фильтр 3 x 3 x 3

смещение



результат — карта 6 x 6

1D-свертка

На вход приходит некоторая посл-ть из объектов $\{A_1, ..., A_n\}$.
Каждый объект представляется вектором его признаков.

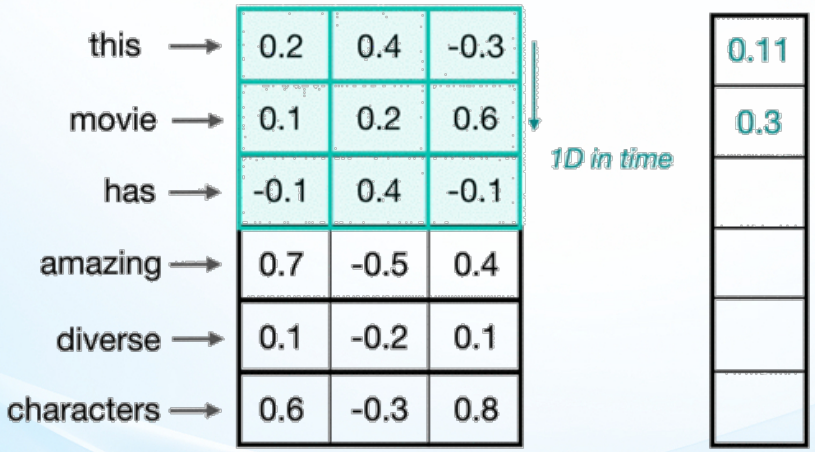
Запишем объекты в матрицу A размера $n \times m$.
В отличие от картинок в матрице A соседние по горизонтали числа не упорядочены, однако соседние по вертикали — упорядочены.

1D-свертка

Поэтому для A не совсем правильно делать 2D-свертку, поэтому применяется 1D-свертка.

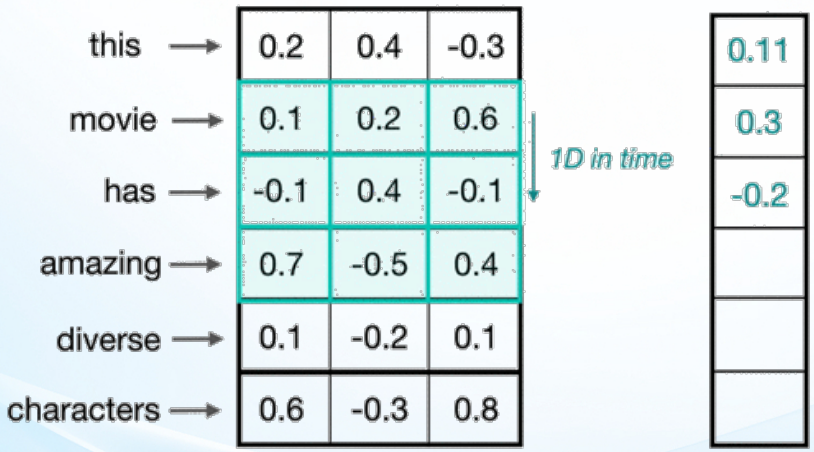
1D-свертка

Поэтому для A не совсем правильно делать 2D-свертку, поэтому применяется 1D-свертка.



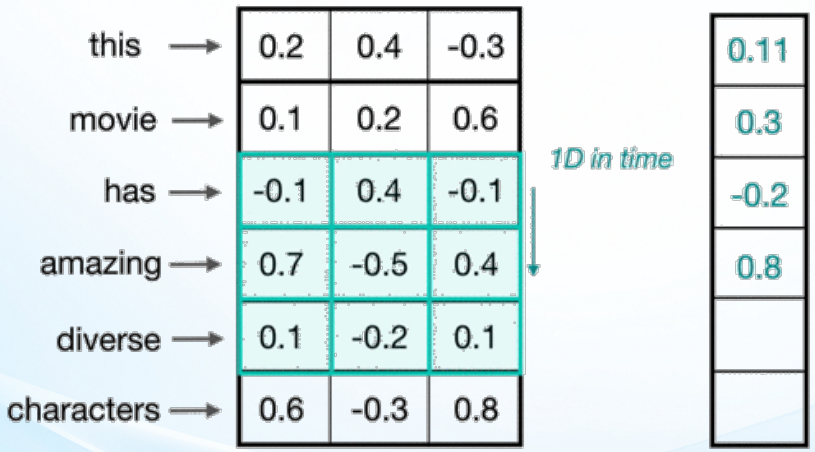
1D-свертка

Поэтому для A не совсем правильно делать 2D-свертку, поэтому применяется 1D-свертка.



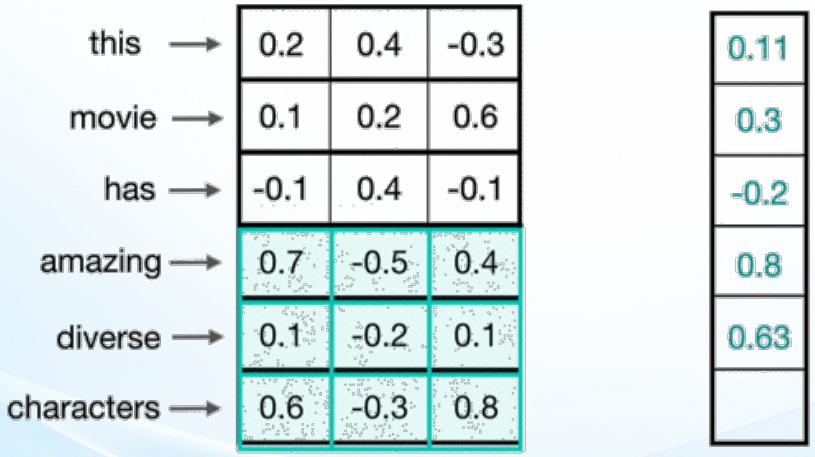
1D-свертка

Поэтому для A не совсем правильно делать 2D-свертку, поэтому применяется 1D-свертка.



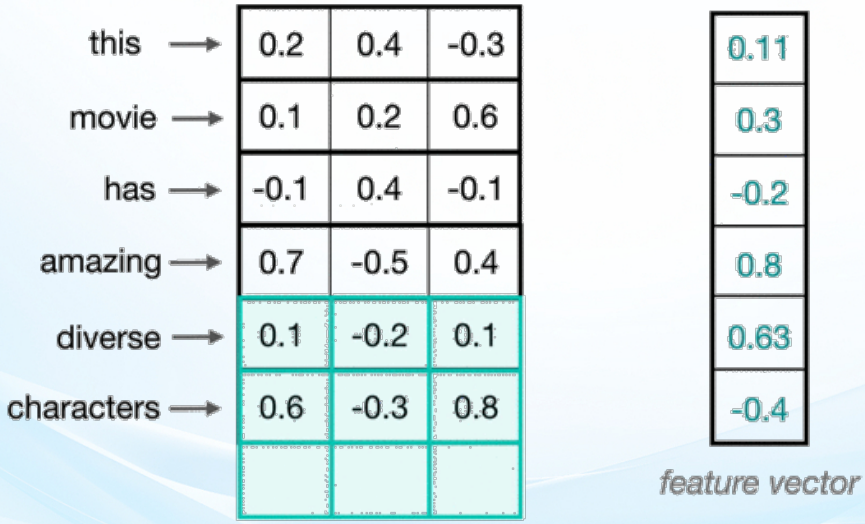
1D-свертка

Поэтому для A не совсем правильно делать 2D-свертку, поэтому применяется 1D-свертка.



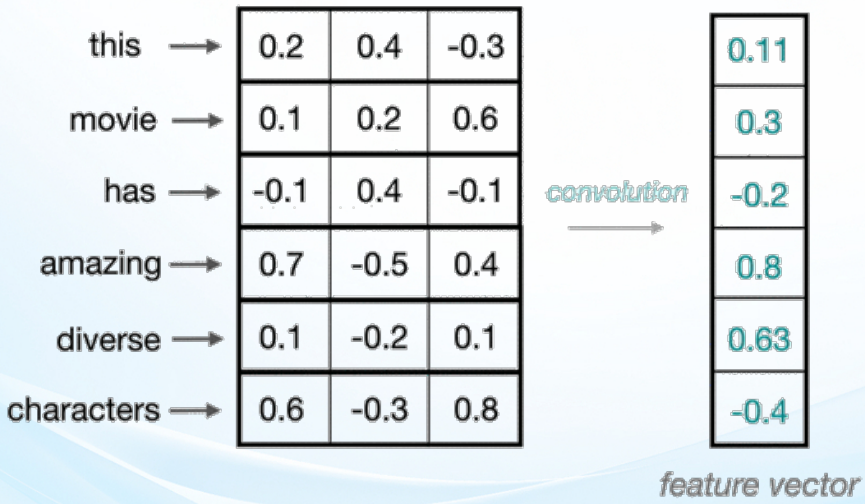
1D-свертка

Поэтому для A не совсем правильно делать 2D-свертку, поэтому применяется 1D-свертка.

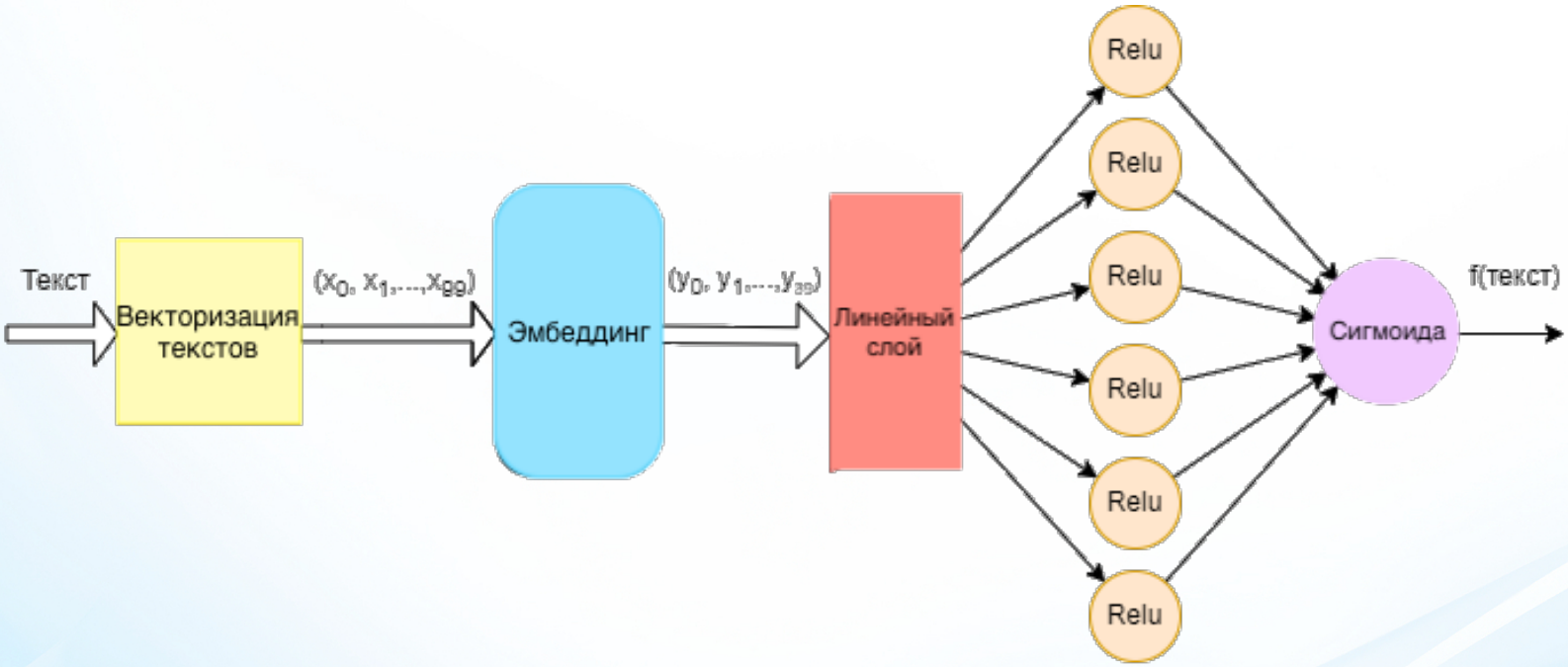


1D-свертка

Поэтому для A не совсем правильно делать 2D-свертку, поэтому применяется 1D-свертка.



Теперь можем решать задачу классификации:



Как и в задачах компьютерного зрения, мы можем превратить выходы сверточных слоев в логиты для предсказания класса с помощью полносвязного слоя в конце.

А превратить логиты в вероятности можно с помощью softmax.

Задача продолжения текста

Задача:
хотим с помощью нейросети продолжать текст,
то есть решать задачу генерации.

В этом нам поможет **языковая модель**.

Простой способ:
по последовательности слов этого текста пытаемся
предсказать наиболее вероятное.

Тогда полученный текст будет выглядеть похожим на настоящий.

Пусть на вход приходит посл-ть объектов $x_1, \ldots, x_n$.

Цель языковой модели

По имеющемуся датасету оценить вероятность появления этой последовательности.

Вероятность последовательности —

мера того, насколько вероятна эта последовательность в реальном мире (имеющимся датасете).

Пример:

$\mathrm{P}(\texttt{я учу машинное обучение}) > \mathrm{P}(\texttt{учу машинное я обучение})$

Языковая модель может быть использована для:

- оценки качества сгенерированного текста;
- предсказания наиболее вероятного следующего токена;
- исправления опечаток;
- генерации последовательности;

Какими способами можно решить данную задачу?

Нейросети?

Свёрточные нейросети?

Не возникнет ли проблем которые встречались нам раньше?

Проблема

На практике оказывается, что построить языковую модель на основе сверточной нейросети не получается.

Причина

Сверточная нейронная сеть помнит только ограниченное число токенов.

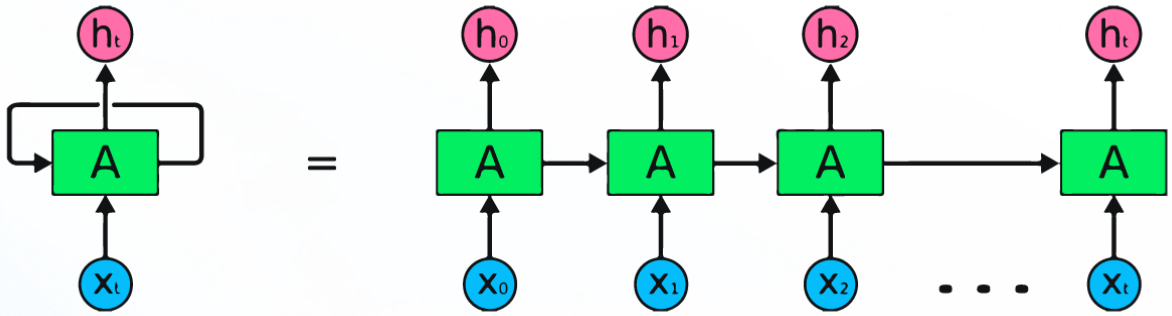
Причина

Построить нейросеть, которая запоминала бы все предыдущие токены и делала на их основе предсказание.

RNN

Recurrent Neural Network:

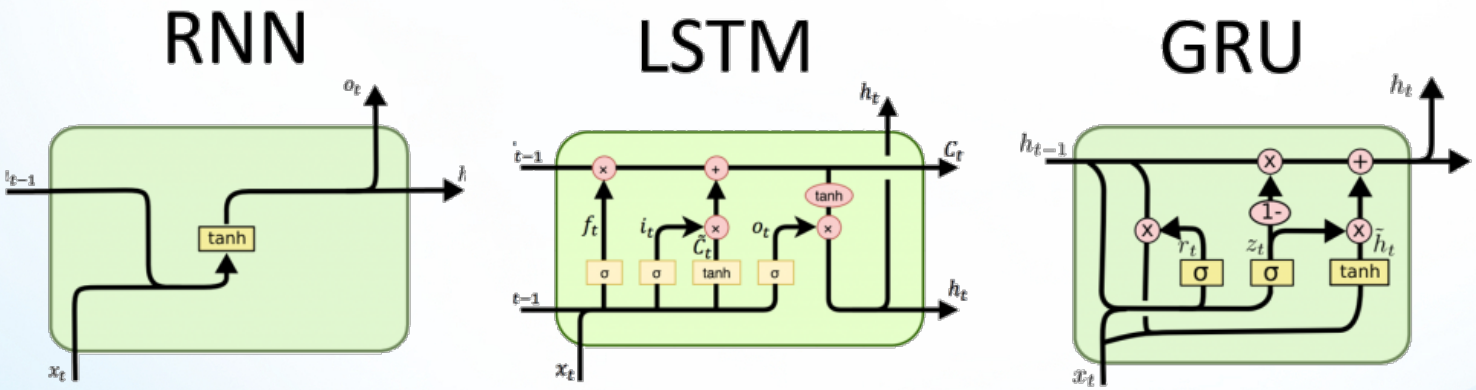
слой, умеющий обрабатывать последовательности.



Обозначения

- $x_t \in \mathbb{R}^D$ — входной элемент во время t;
- $h_t \in \mathbb{R}^d$ — скрытое состояние во время t,
  которое обычно вычисляется следующим образом:
  - x_t — входной элемент во время t;
  - $h_t = \tanh(h_{t-1} W_h + x_t W_x)$;
  - $W_h \in \mathbb{R}^{d \times d}$, $W_x \in \mathbb{R}^{D \times d}$ — обучаемые матрицы.

RNN



Подробнее о рекуррентных архитектурах вы узнаете на DS-потоке!

Transformer

Проблема:
RNN ”забывает” токены с предыдущих итераций, и качество перевода падает с увеличением длины последовательности.

Архитектурные изменения RNN не решают эту проблему

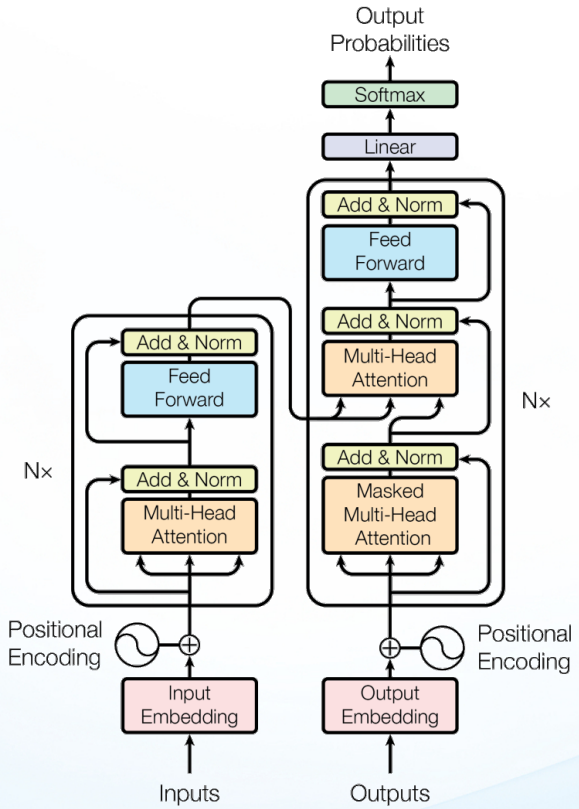
К тому же, RNN очень неэффективны.

Transformer

BERT

Encoder

Преобразует последовательность слов в вектор в вектор



GPT

Decoder

По вектору предсказывает для каждого слова из словаря быть следующим

Подробнее про трансформеры вы тоже сможете узнать на DS-потоке!

План лекции

NLP

Кодирование текстов

Основные модели

LLM

До этого мы пытались найти подходящую архитектуру отдельно для каждой задачи.

А если попробовать построить одну модель для всего сразу?

Получим **LLM**!

Large Language Models (LLM) решают ту же задачу языкового моделирования, которую мы сформулировали ранее.

В большинстве случаев LLM — это часть архитектуры Transformer, а именно Decoder.

Некоторые хитрости и техники позволяют адаптировать эти модели к решению огромного класса задач.

Предобучение LLM

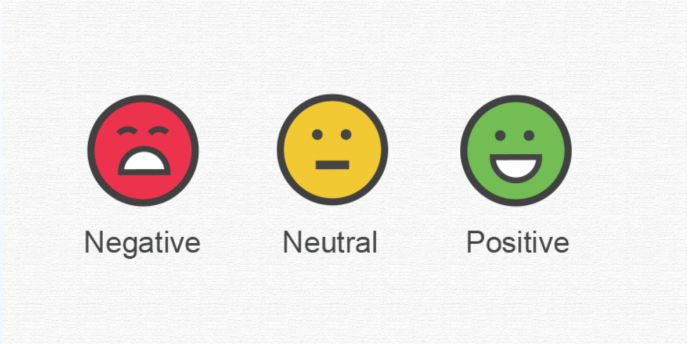
Обучение большой языковой модели обычно состоит из двух этапов.

1. **Pretraining** на большом корпусе текстов.
   В зависимости от архитектуры это могут быть разные задачи:
   LM, MLM, Denoising.

2. **Finetuning** под целевую задачу.
   Используя небольшое число примеров дообучаем модель.

Со всеми этими страшными словами
вы познакомитесь на DS-потоке!

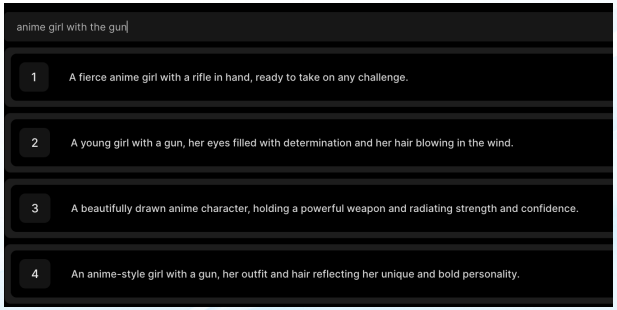
Примеры задач NLP



Анализ тональности

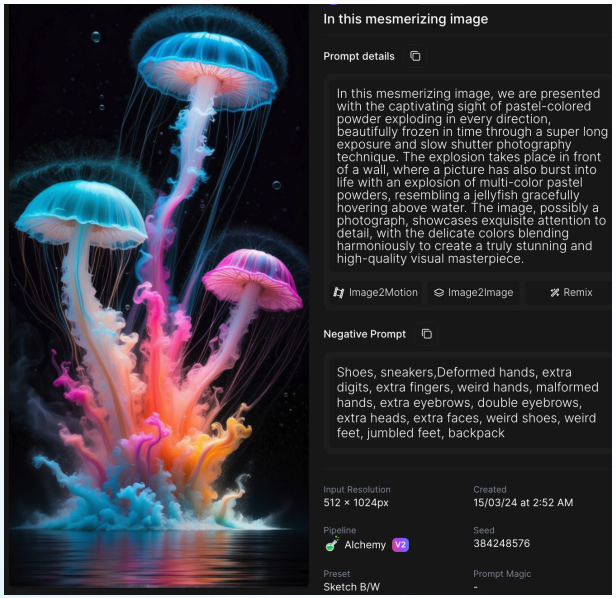
Диалоговые системы

Диалоговые системы



Дополнение запросов пользователей

Примеры задач на стыке NLP и CV



Генерация картинки по тексту

Генерация подписей к изображениям

И еще много чего...

... будет на DS-потоке! :)

ВСЁ!