



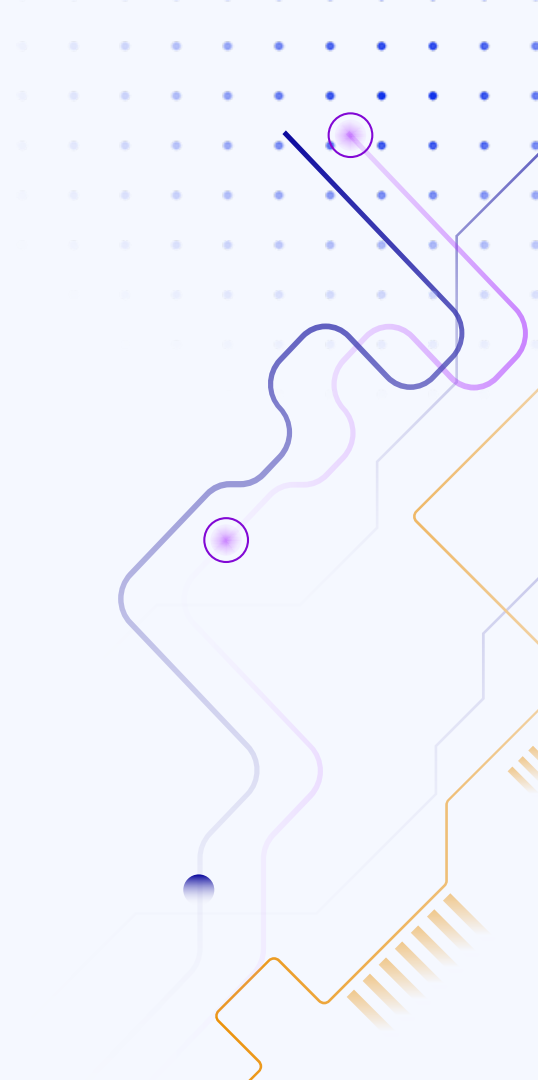
1D CNN. Transfer Learning

Свертки для работы с текстами



01

Эмбеддинги



Идея эмбедингов в нейронных сетях

- Хотим рекомендовать фильмы
- Нужно фильмы как-то сгруппировать по признакам: детям нравятся детские, любителям арт хауса нравится арт хаус
- Сделаем шкалу:



Shrek



Incredibles



The Triplets
of Belleville



Harry Potter



Star Wars



Bleu



The Dark
Knight Rises

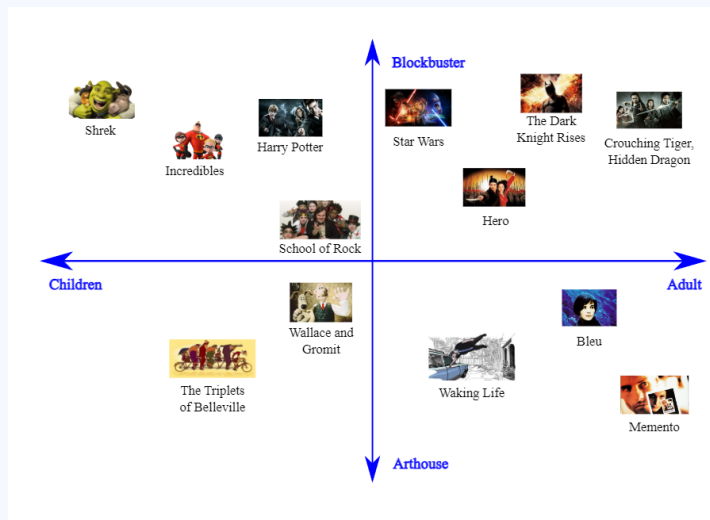


Memento



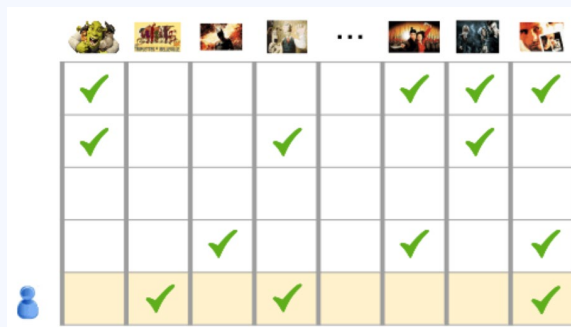
2D-похожесть









- Одномерная шкала мало что передает
- Попробуем 2D
- Каждый фильм можно представить как его координаты:



Sparse representation

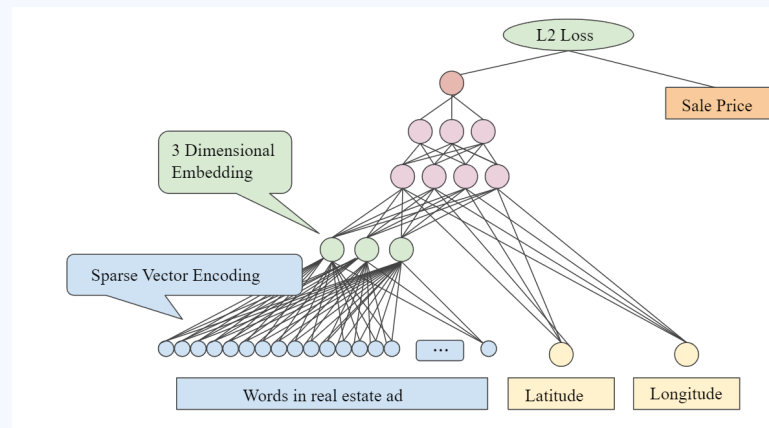
- Хотим, конечно, гораздо больше двух признаков
- Как закодировать каждый объект в числовом виде?
- Самый простой способ — One Hot Encoding
- Но он же самый неэффективный...



					...			
	✓					✓	✓	✓
	✓			✓			✓	
			✓			✓		✓
		✓		✓				✓

Слой эмбединга

- ДАВАЙТЕ ЗАВЕДЕМ ПРОСТО СЛОЙ С ВЕСАМИ, КОТОРЫЙ БУДЕТ УЧИТЬСЯ ИЗ НАШЕГО SPARSE REPRESENTATION
 - Для слов: каждое слово первоначально ONE
 - Потом мы берем слой с рандомно установленным числом весов
 - Учим
 - Profit!
-
- Получается, модель пытается извлечь из нашего ONE такие признаки, которые будут наиболее релевантными для той конкретной задачи, которую она теперь пытается решить





Слой эмбединга

- Итого, нужно:
 1. Словарь word2id (и id2word для раскодирования) – чтобы делать ONE
 2. Слой эмбедингов (обучаемый!)
- Можно использовать предобученные чужие эмбединги
- Эмбединги могут быть для чего угодно: все, что можно закодировать ONE. Символы, токены, ВРЕ
токены, предложения, фильмы...

Лирическое отступление: BPE

- За токен можно считать разные вещи. Например, New York – это один токен или больше?
- Вместо того, чтобы нам придумывать определение токена и самим писать правила, заставим текст сообщить нам, что в нем – токены
- Byte Pair Encoding – алгоритм автоматической токенизации по **подсловам** (subwords)
- Современные нейронные сети используют BPE
- Понадобятся **обучающие данные** – набор сырых текстов, на которых будем учить, какие бывают токены
- «Собираем» токены из символов

Лирическое отступление: ВРЕ

Слова в наших обучающих данных: ("hug", 10), ("puq", 5), ("pun", 12), ("bun", 4), ("hugs", 5)

Алфавит: ["b", "q", "h", "n", "p", "s", "u"], значит, можем представить слова так:

("h" "u" "q", 10), ("p" "u" "q", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "q" "s", 5)

Устанавливаем, какого объема словарь хотим.

Вычисляем самые частотные сочетания: самая частотная пара – "uq", она встретится 20 раз

Соединим эти два символа и получим новый словарь: ["b", "q", "h", "n", "p", "s", "u", "uq"]

Тогда наш корпус будет выглядеть так:

("h" "uq", 10), ("p" "uq", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "uq" "s", 5)

Какие два "слова" тогда будут чаще всего вместе? Найдите.

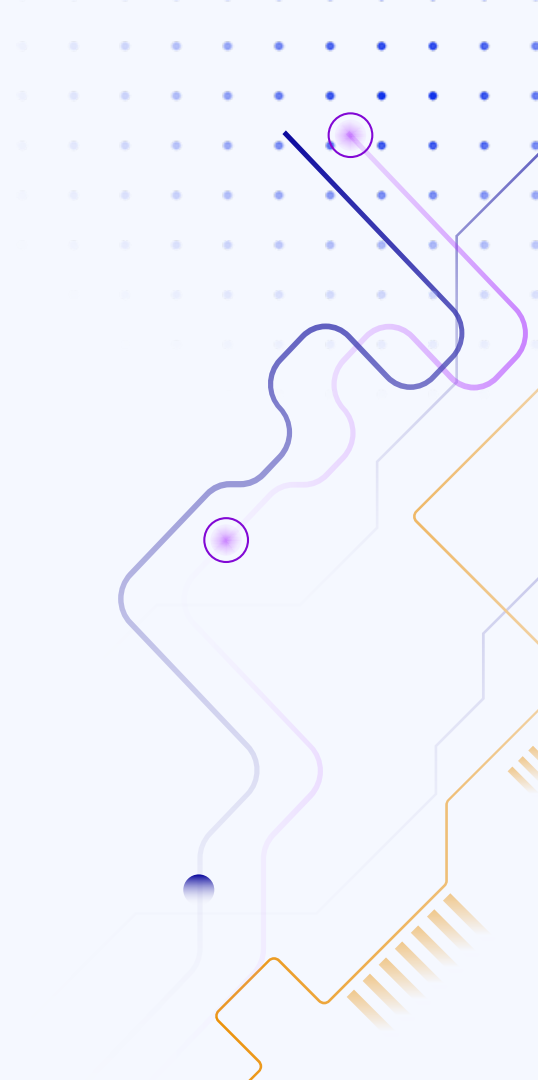
Склеим их в одно новое слово и добавим в наш словарь.

И так пока не доведем наш словарь до желаемого объема.



02

CNN и тексты



Текстовые данные

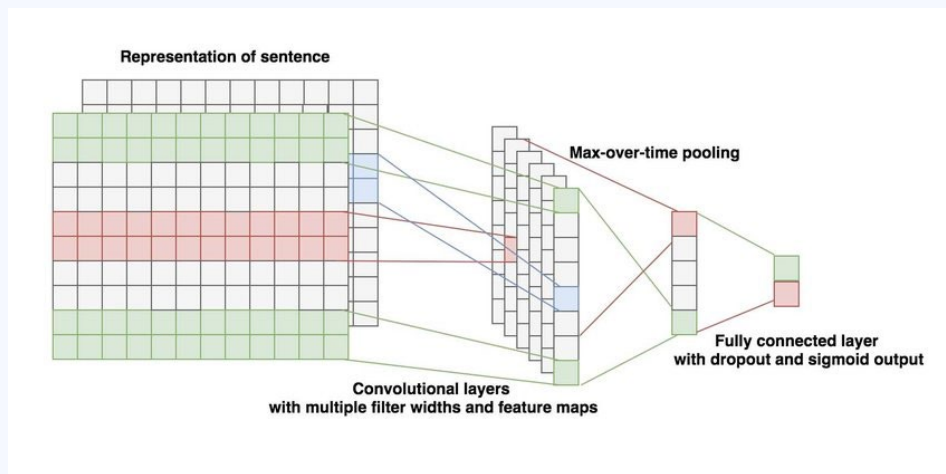
- ЗАДАЕМ ДЛИНУ ЭМБЕДДИНГА, НАПРИМЕР, 5
- ЗОЛОТОЕ ПРАВИЛО, КАКУЮ ДЕЛАТЬ ДЛИНУ:

$$length = \sqrt[4]{vocabulary}$$

[illegible]

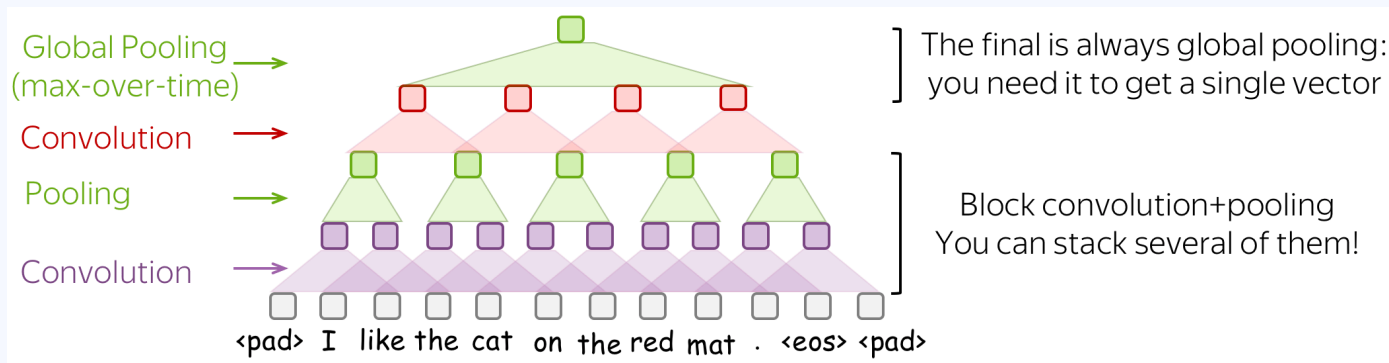
Как CNN сворачивает текст?

- Предложение – наstackанные эмбединги слов
- Длина эмбединга – как бы глубина
- Поэтому свертка всегда идет на всю длину эмбединга
- Размер ядра – по сути n в граммах (2 – биграмма, 3 – триграмма)

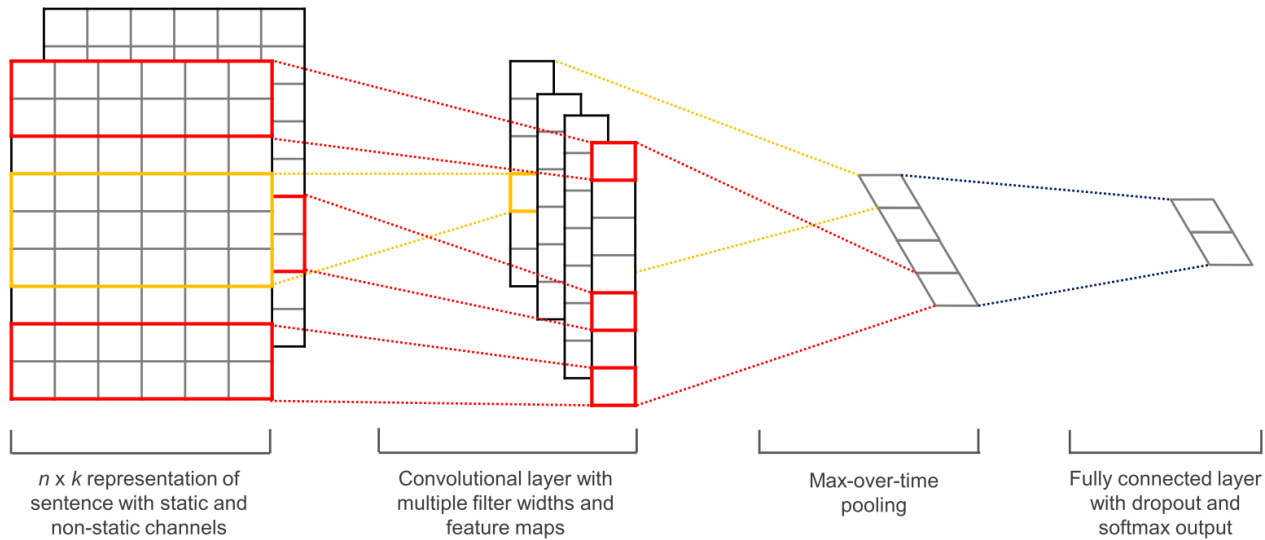


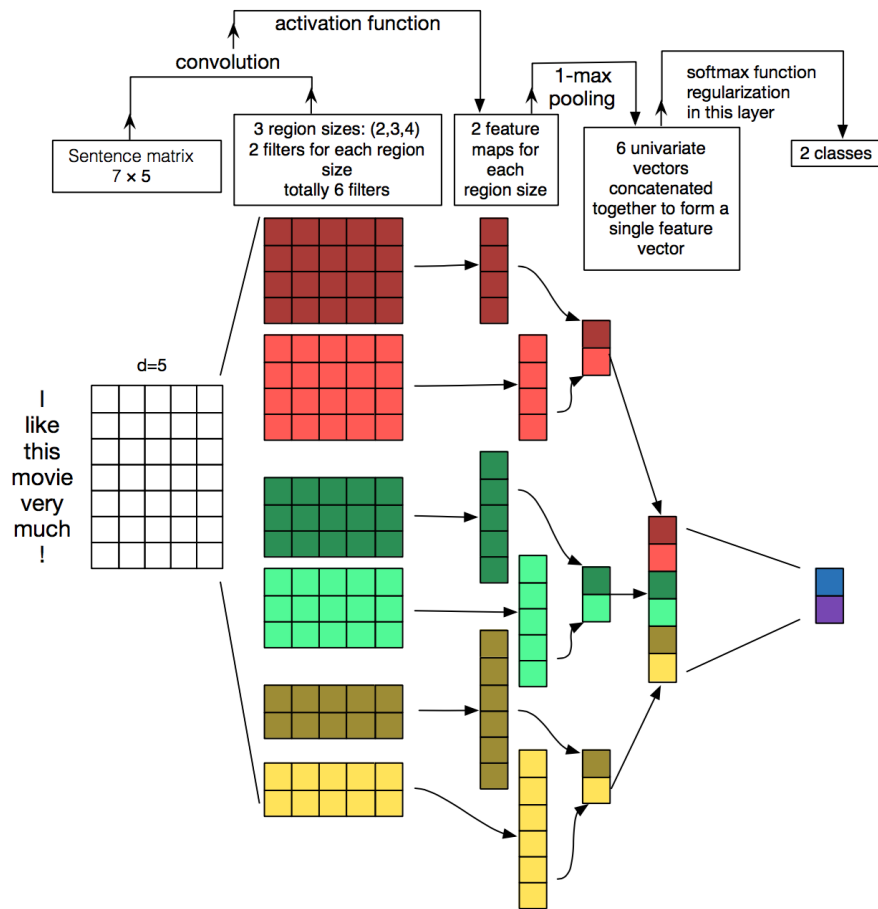
Max Over Time Pooling

- Сконкатенируем получившиеся результаты сверток (это называется карта активации)
- По каждому столбцу выберем максимальное значение



wait
for
the
video
and
do
n't
rent
it





Inception CNN



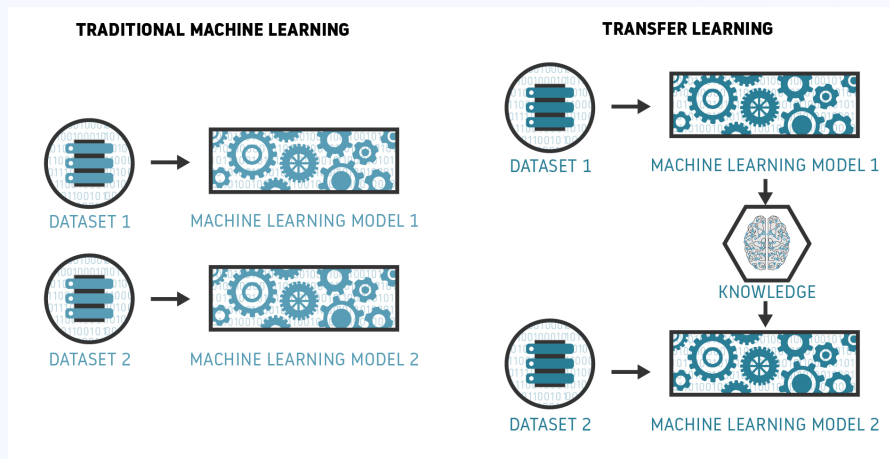
03

Transfer Learning



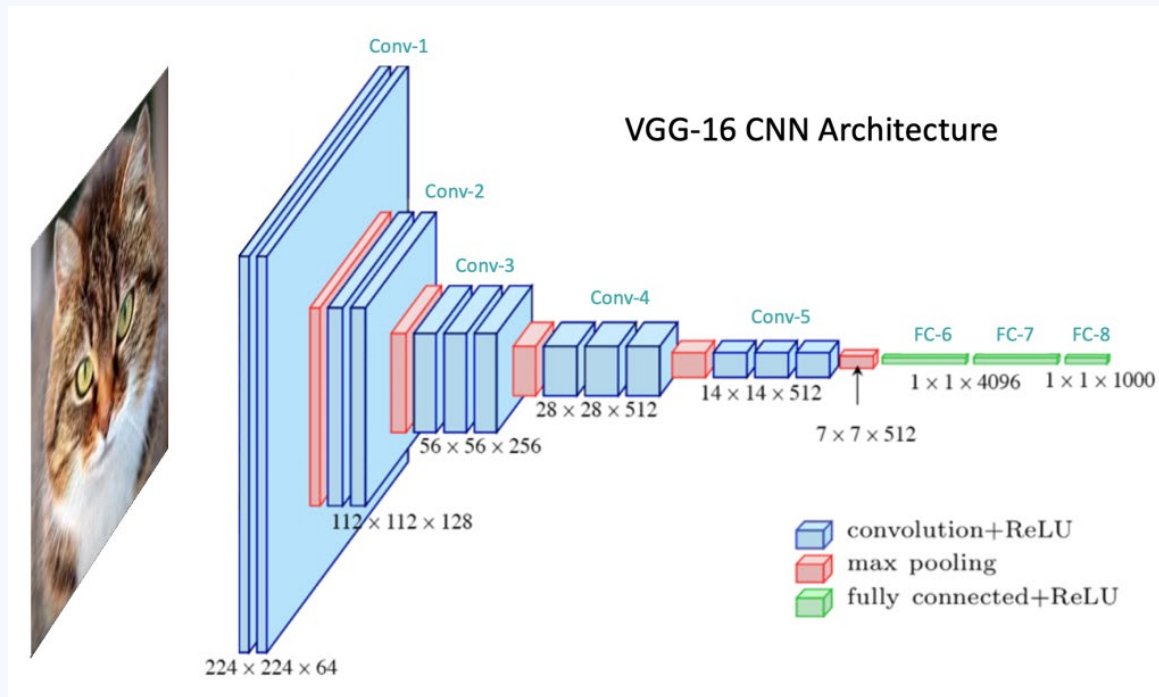
Transfer Learning

- На практике свёрточные сети с нуля обучают только большие технологические компании
- Это происходит из-за ограниченности ресурсов
- Уже обученные архитектуры пытаются адаптировать под новые задачи, это называется transfer learning (перенос знаний)



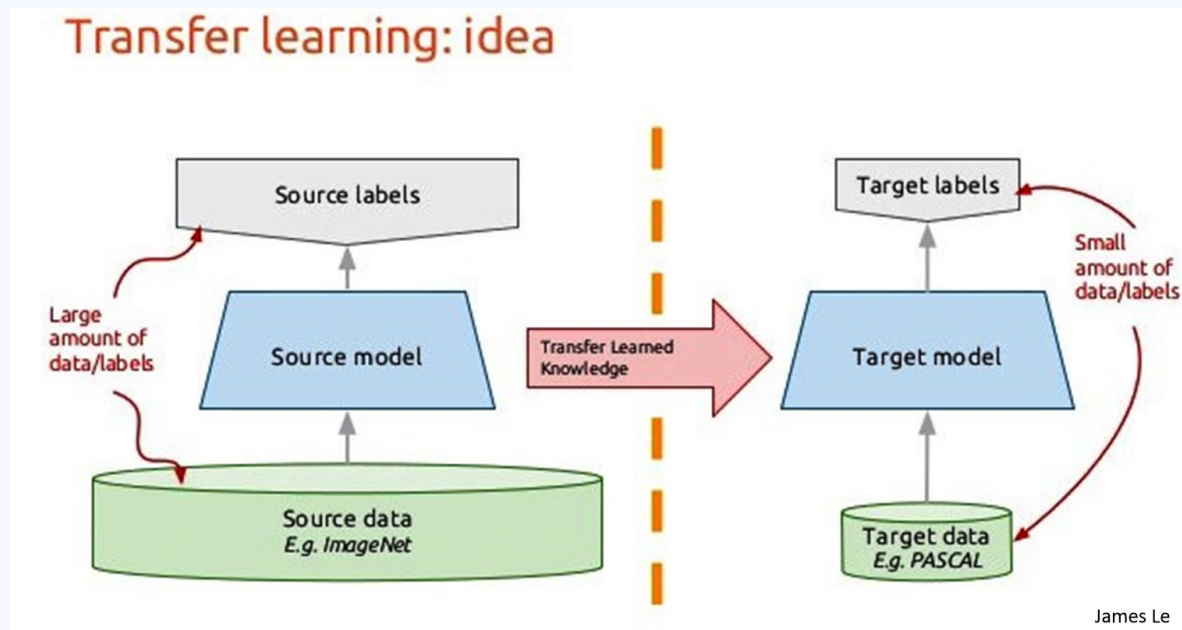
Transfer Learning

Глубокие сети извлекают из изображений сложные фичи, но для их обучения нужно много данных...



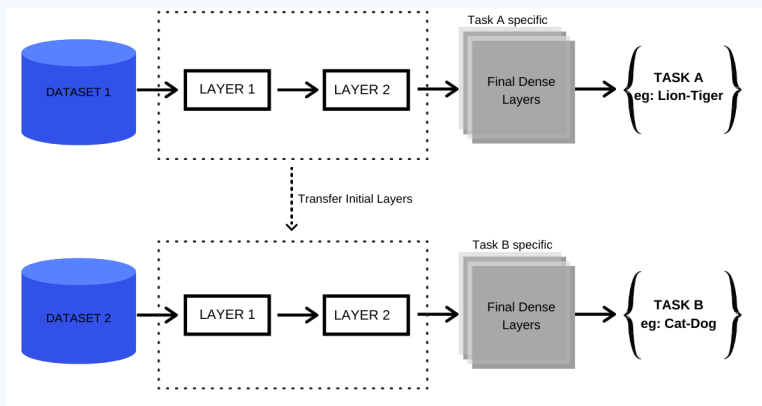
Transfer Learning

ДАВАЙТЕ ПОВТОРНО ИСПОЛЬЗОВАТЬ УЖЕ ПРЕДОБУЧЕННУЮ СЕТЬ!



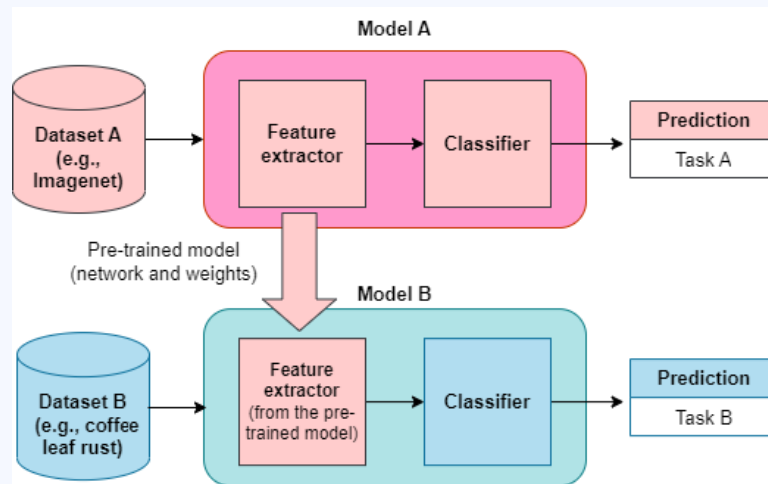
Transfer Learning

- Нужно меньше данных для обучения, так как нас интересуют лишь последние слои
- Как правило, на первых слоях фильтры похожие для всех задач
- Чем сильнее новая задача отличается от исходной, тем больше слоёв нужно переучивать
- Например, если мы хотим распознавать эмоции, в датасете для нашей сетки должны были быть ЧЕЛОВЕЧЕСКИЕ ЛИЦА



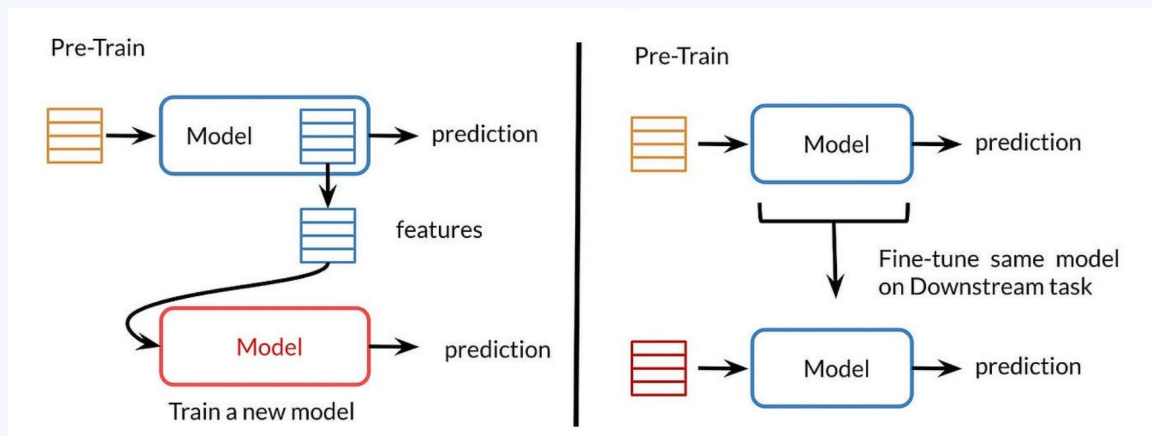
Transfer Learning

- Обычно наша сетка состоит из двух частей: первая извлекает признаки (Embedding layer, CNN, RNN, LSTM...), а вторая уже решает конкретную задачу (FC): эту задачу называют downstream task
- Иногда про решающий слой говорят «голова»
- Его обычно и снимают, чтобы заменить на новый под другую задачу



Fine Tuning

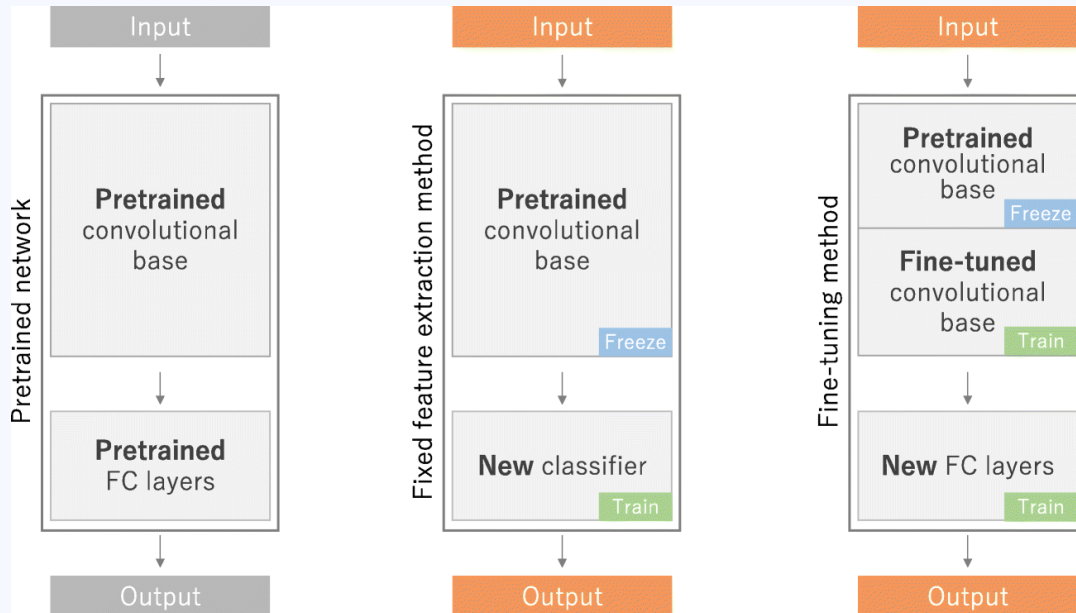
- Можно инициализировать веса переучиваемых слоёв весами с предобученной сети
- Это называется fine tuning, так как инициализация неслучайная



Transfer Learning vs Fine Tuning

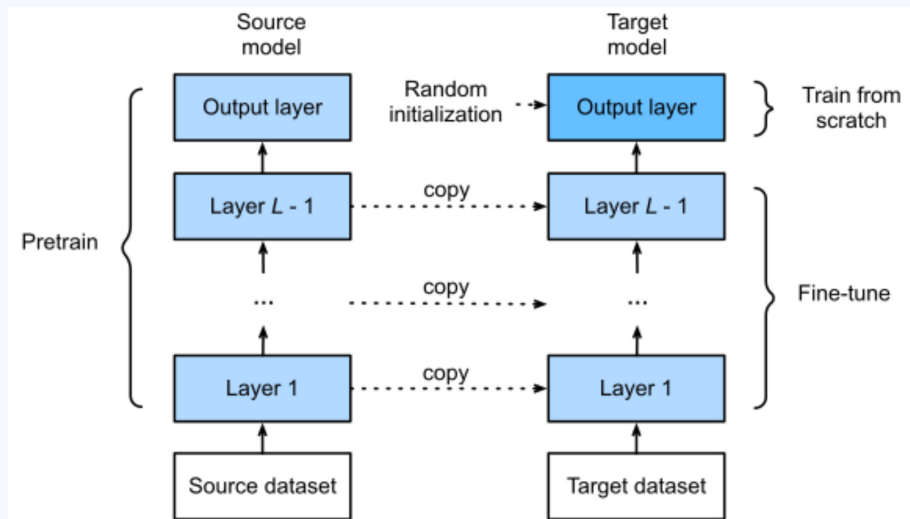
- При Transfer Learning не обучаем взятые от исходной модели слои: мы **замораживаем** их веса
- При Fine Tuning мы берем слои от исходной модели с их весами, но обучаем их с медленным learning rate

Соответственно:



Transfer Learning

- Чем больше датасет, тем больше слоёв можно доучивать
- При finetuning выставляйте более низкую скорость обучения
- Для начала попробуйте 0.1 от оригинальной



Виды Transfer Learning

FINE TUNING: берем за базу предобученную модель, инициализируем свою новую ее весами, часть слоев дообучаем (можно все, можно только некоторые) с более маленьким learning rate

FEATURE EXTRACTION: берем от предобученной модели ту ее часть, которая извлекает признаки (CNN, RNN...), обучаем собственную голову-классификатор на них

DOMAIN ADAPTATION: адаптируем предобученную на данных из другого домена

MULTI TASK LEARNING: обучаем одну модель решать сразу несколько задач, чтобы она одновременно улучшала свои предсказания во всех

ZERO SHOT LEARNING: берем предобученную модель и в лоб применяем ее на новых данных без предобучения

Зоопарки моделей

В интернете есть зоопарки с моделями (и, например, есть huggingface.co, до которого уже скоро доберемся...)

- Один большой зоопарк
- Зоопарк для любителей pytorch