

Основные архитектуры нейронных сетей

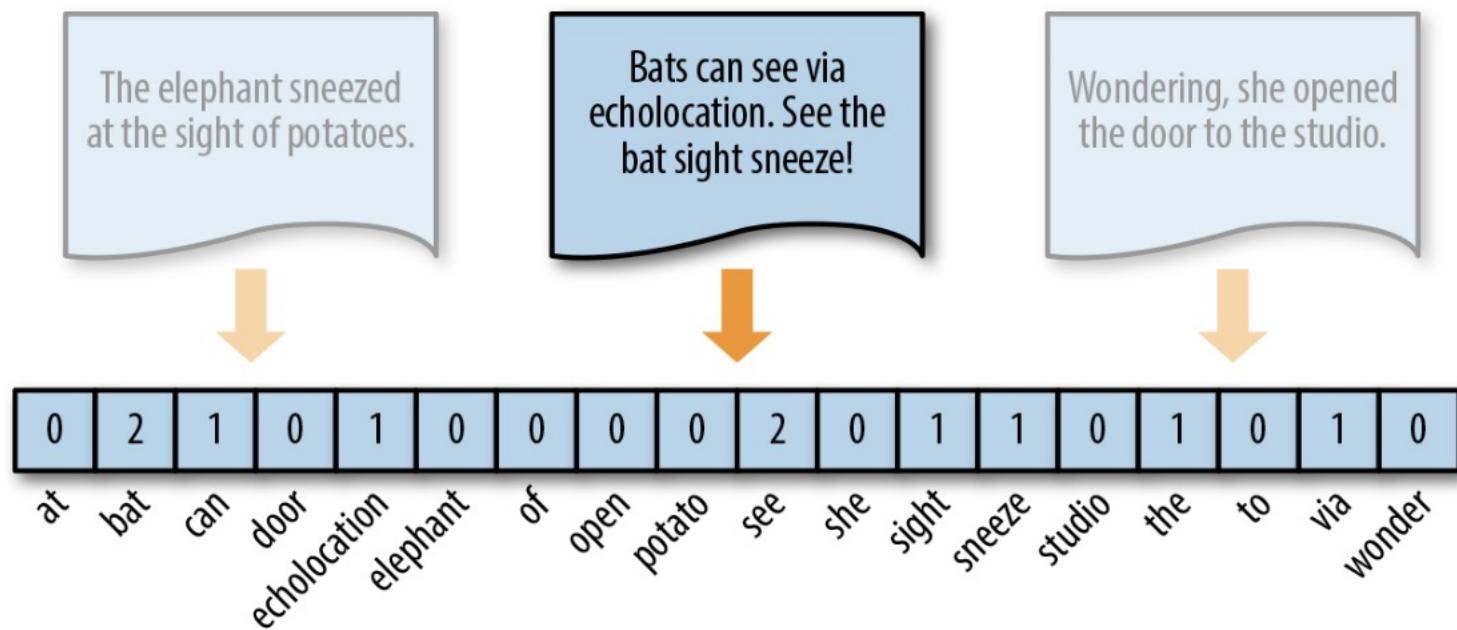
Занятие 11. Продолжение.

Представления для текстов

Bag of words

- Заводим словарь, состоящий из всех слов в выборке
- Делаем признак-индикатор для каждого слова из словаря
- Можно добавлять n-граммы

Bag of words



Bag of words

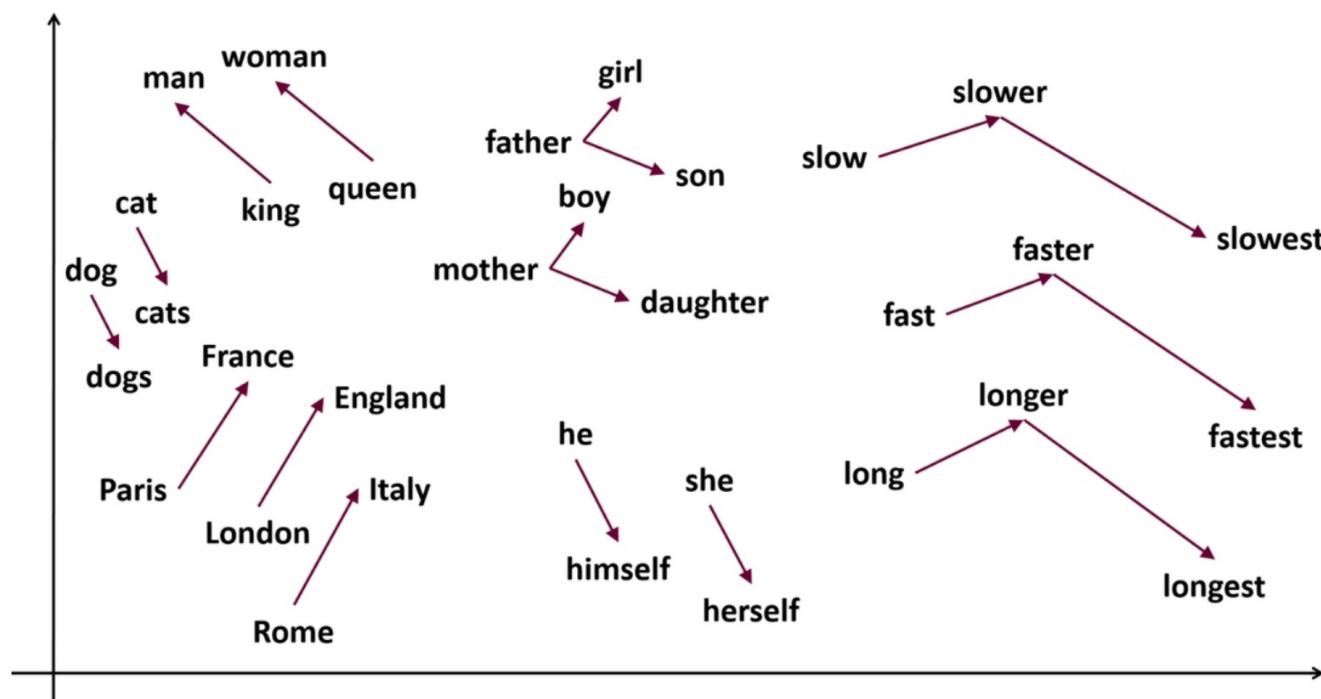
- Слишком много признаков
- Не учитываем смыслы слов
- Семантически похожие тексты могут иметь очень разные представления

word2vec

- Попробуем обучить вектор-представление для каждого слова
- Что потребовать от такого представления?

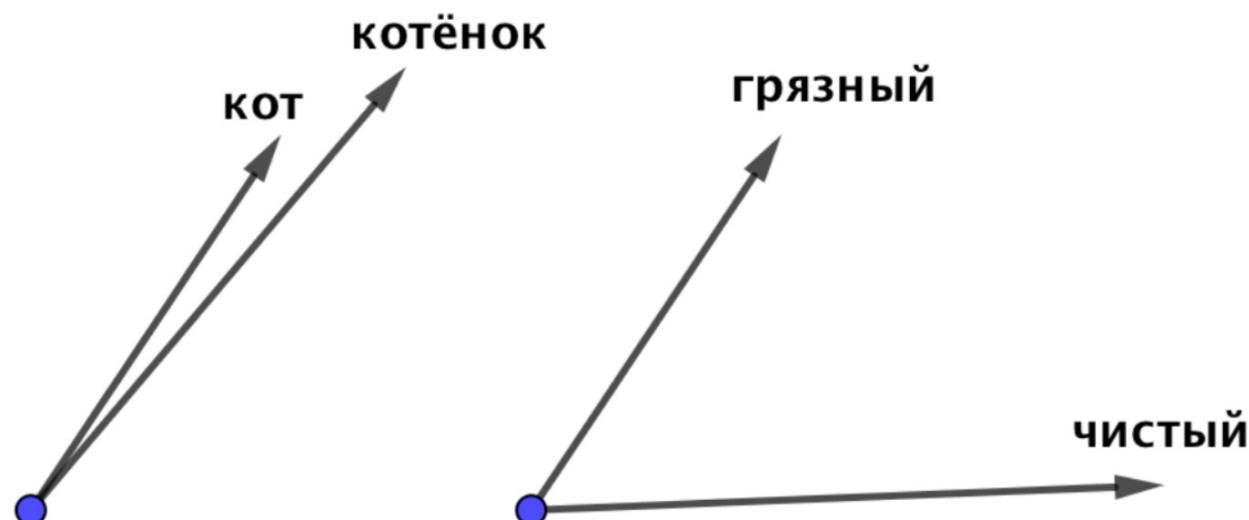
Хотелка раз

- Сохранение семантических свойств



Хотелка два

- Близость по смыслу



Хотелка три

- Арифметика

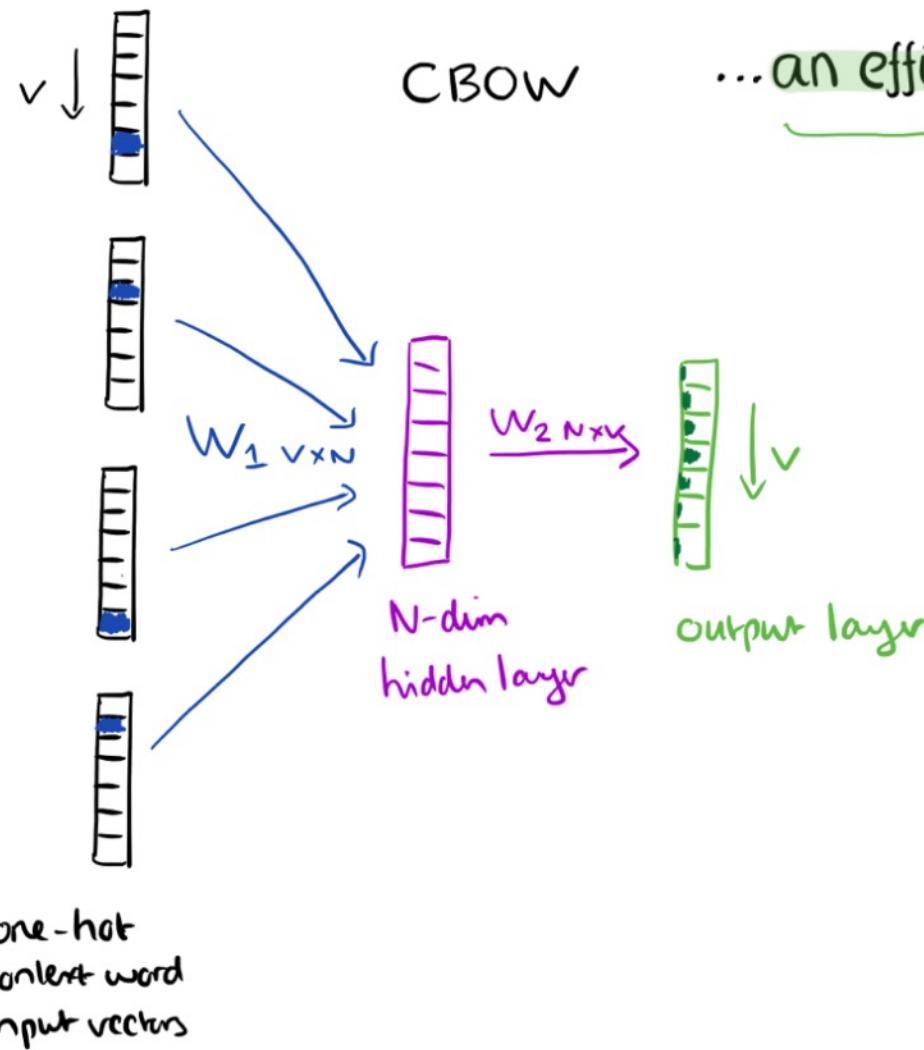


$$- \text{ } \begin{array}{c} \textcolor{pink}{\text{+}} \\ \textcolor{blue}{\text{+}} \end{array} =$$

A mathematical equation consisting of three parts: a minus sign followed by two human figures (one pink, one blue), and an equals sign followed by a third figure.



CBOW



...an efficient method for learning high quality distributed vector ...

content

↑
focus
word

content

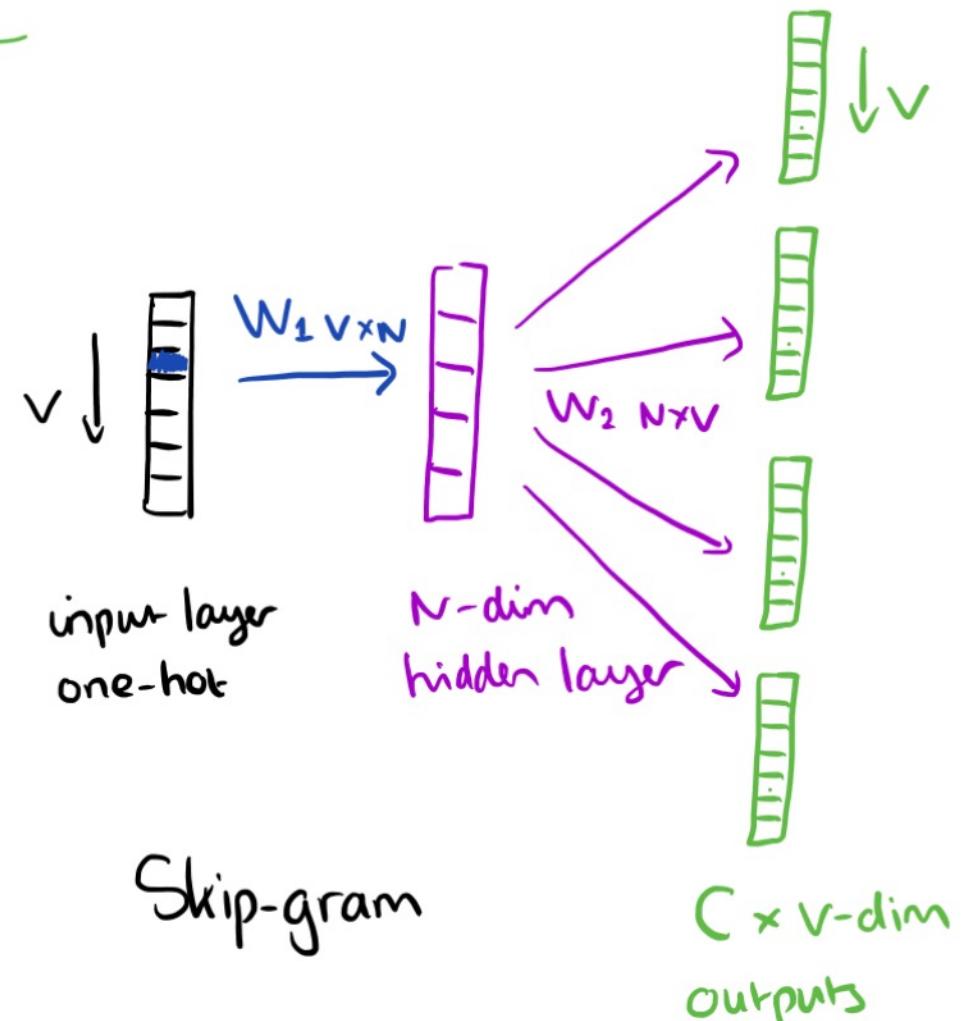
Пробуем предсказать
целевое слово по контексту

Skip-gram model

...an efficient method for learning high quality distributed vector ...



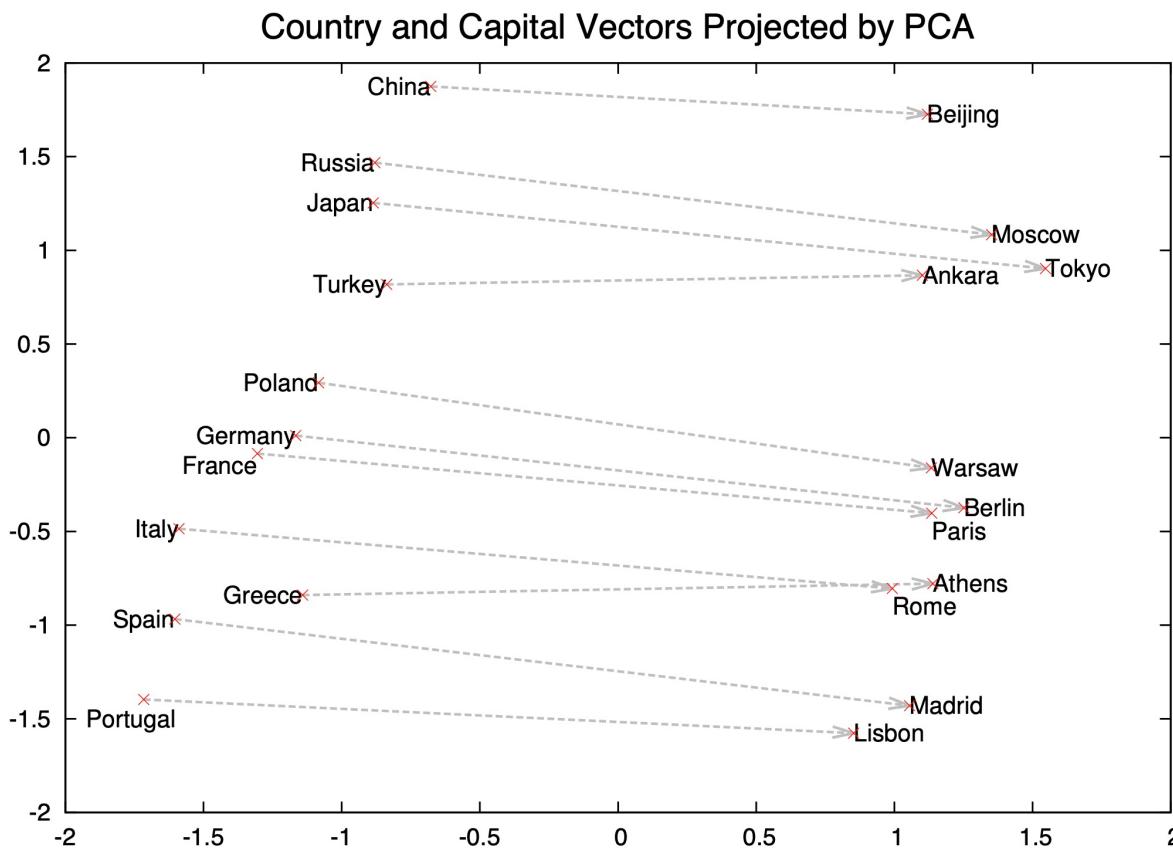
Пробуем предсказать
Контекст по целевому слову



word2vec

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

word2vec



Проблемы word2vec

- Не учитываем структуру слов
- Не закладываем никакой априорной информации о разных формах одного слова
- Не умеем обрабатывать опечатки

Что еще

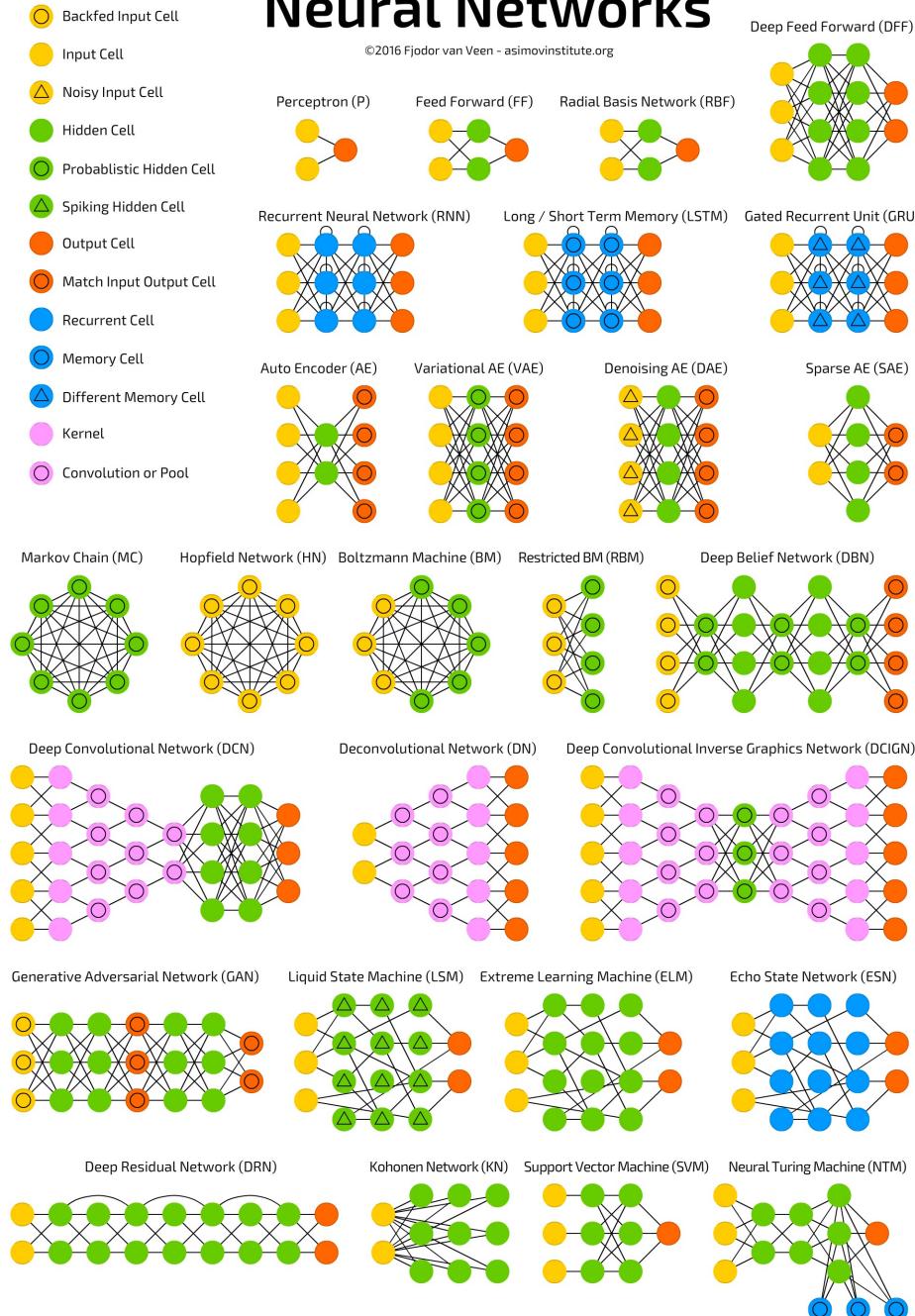
- FastText
- GLoVo
- ELMo/BERT

A mostly complete chart of

Neural Networks

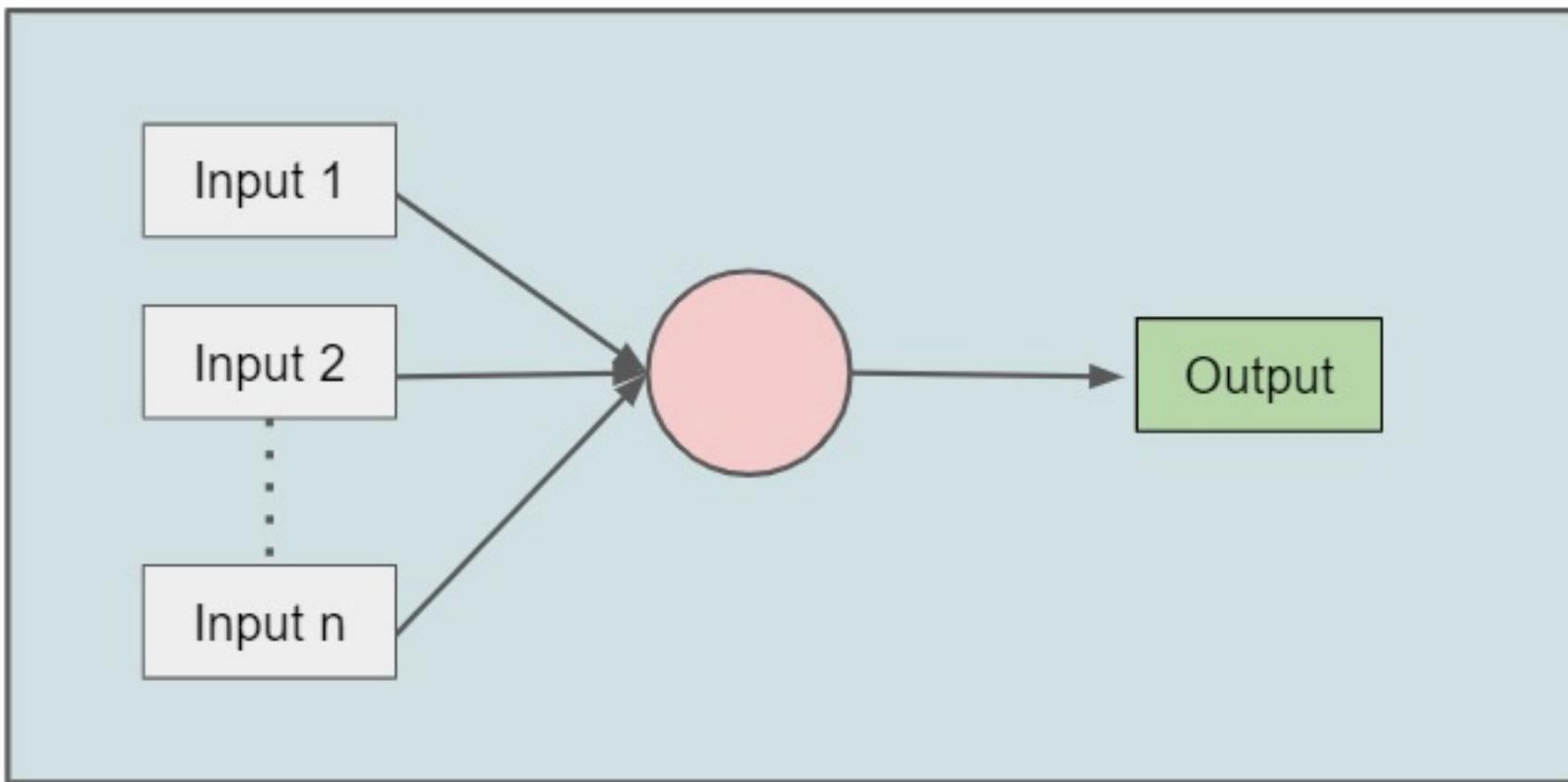
©2016 Fjodor van Veen - asimovinstitute.org

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool



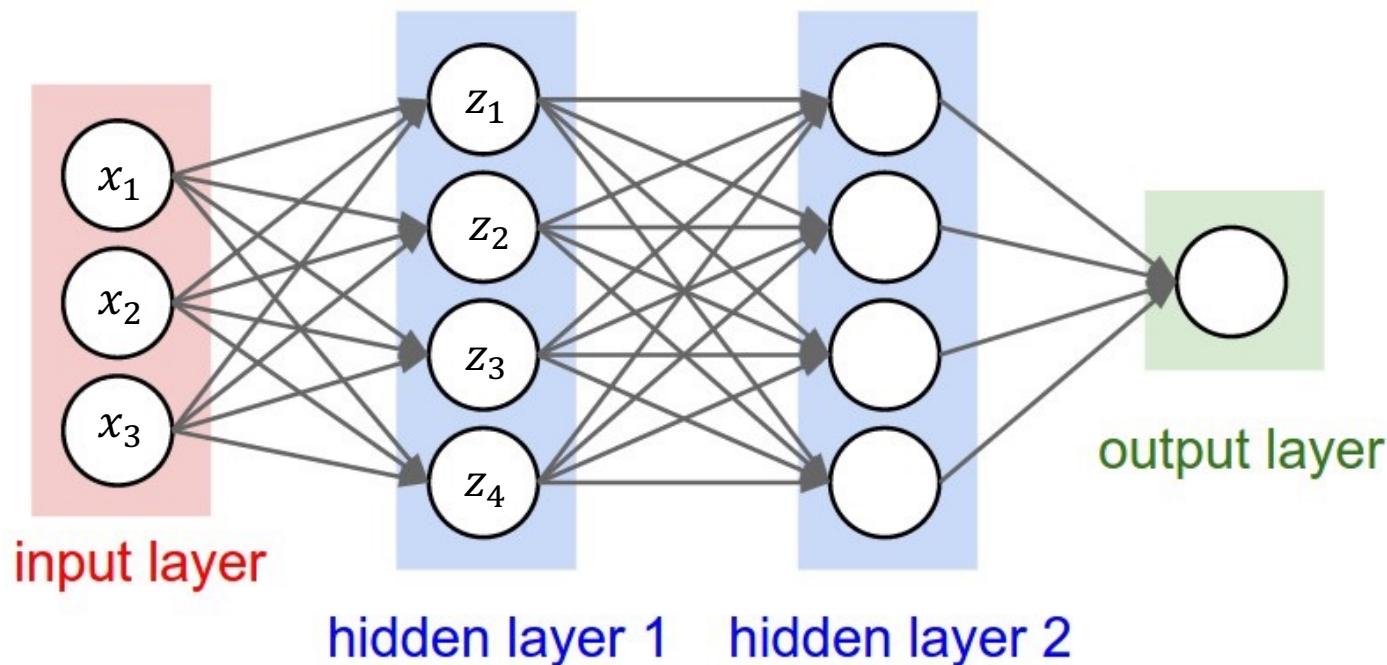
<https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>

1. Perceptron



2. Feed Forward Neural Network

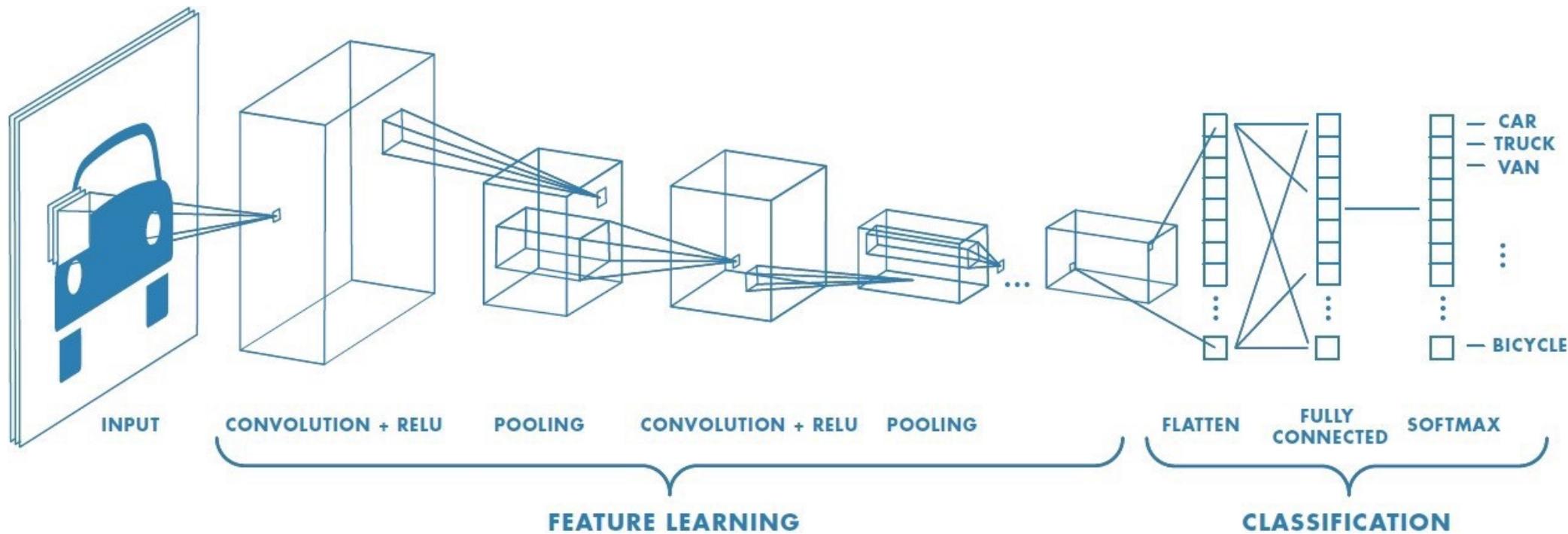
$$z_j = \sum_{i=1}^n w_{ij} x_i$$



Сфера применения

- Предиктивные задачи

3. Convolutional Neural Networks



MNIST

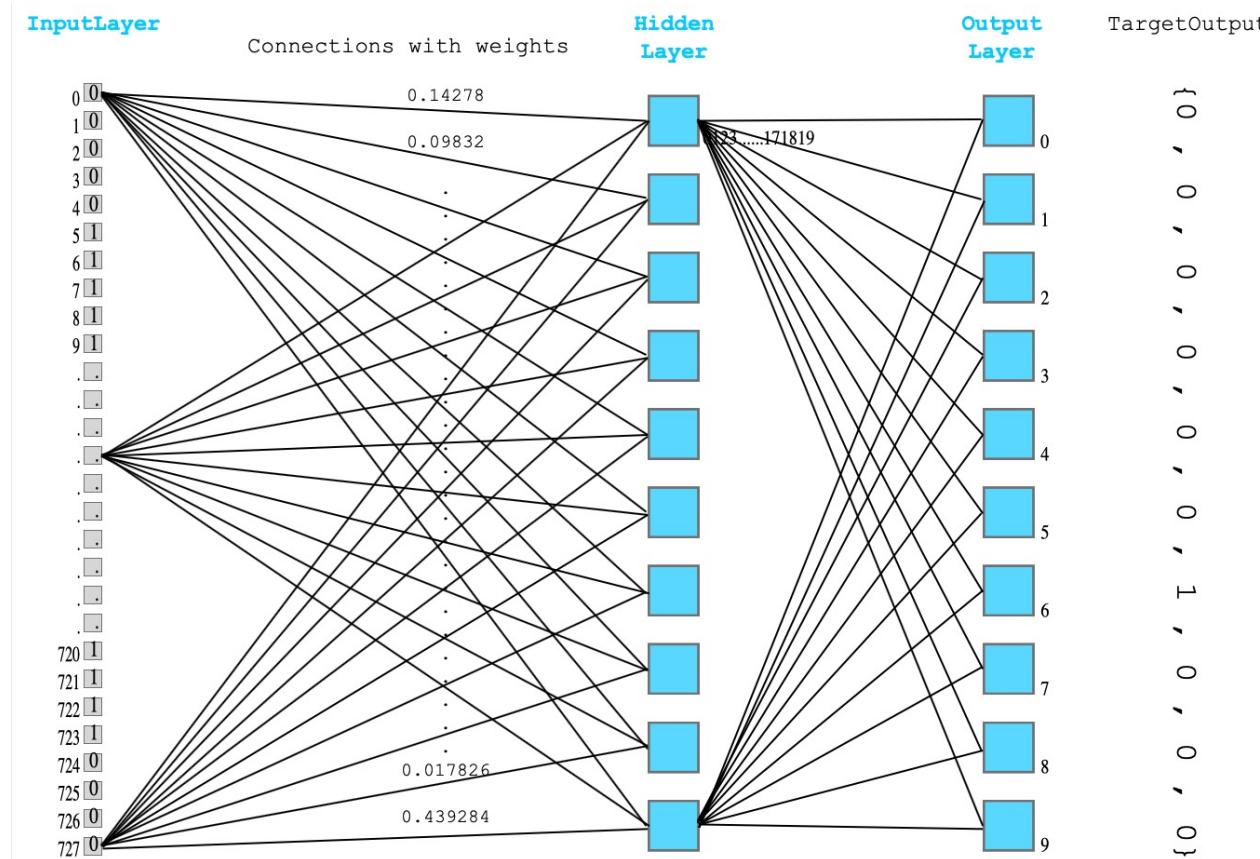


MNIST

- Изображения 28 x 28
- Изображения центрированы
- 60.000 объектов в обучающей выборке

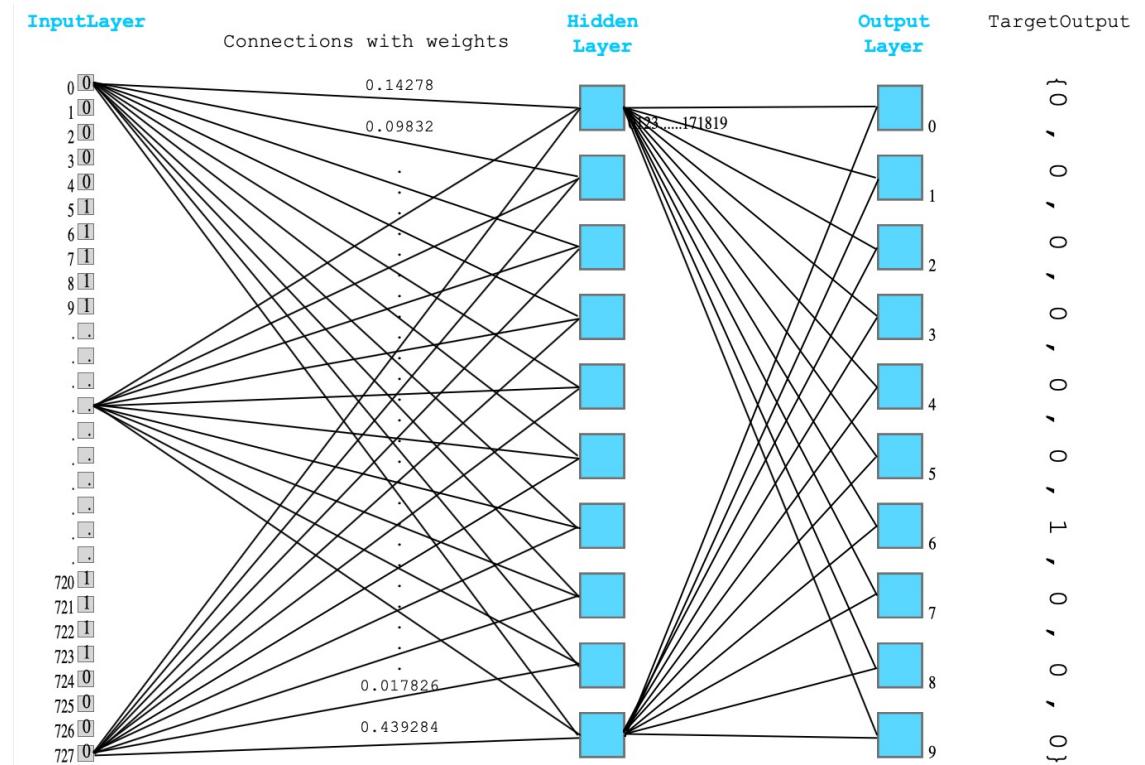
MNIST

- Что может выучить полносвязная сеть?

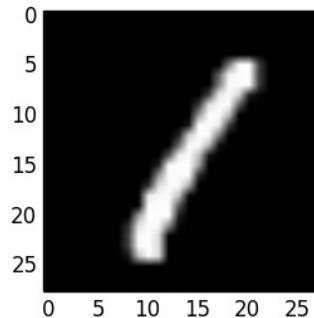
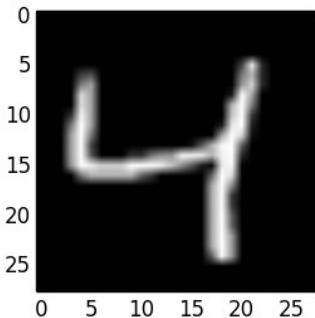
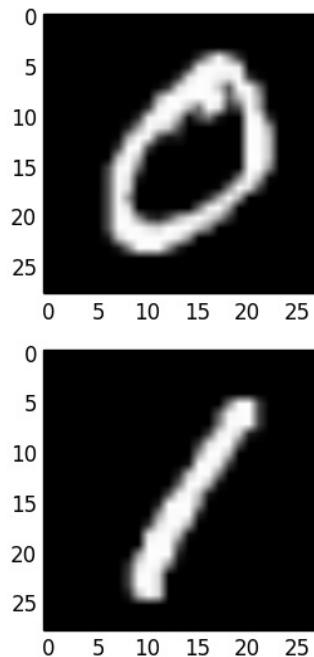
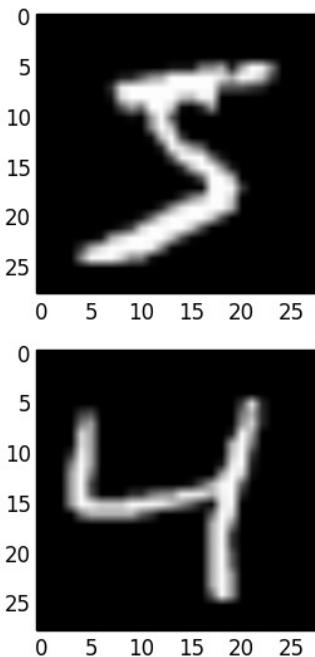


MNIST

- Каждый нейрон может детектировать заполненность конкретного набора пикселей

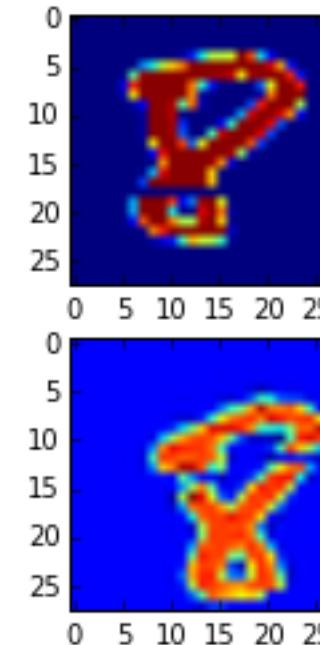
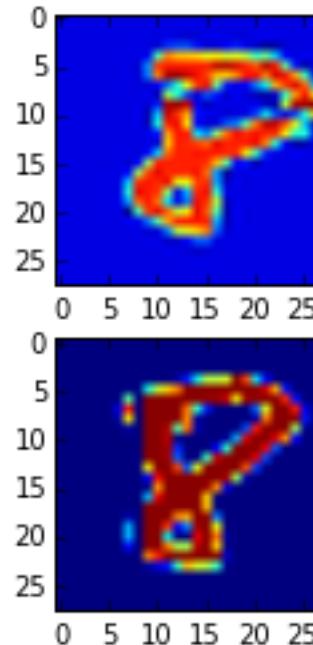


MNIST



MNIST

- Если немного сдвинуть цифру, то нейрон уже не будет на неё реагировать



Число параметров

- 784 входа
- Полносвязный слой: 1000 нейронов
- Выходной слой: 10 нейронов (по одному на каждый класс)
- Весов между входным и полносвязным слоями:

$$(784 + 1) * 1000 = 785.000$$

- Весов между полносвязным и выходным слоями:

$$(1000 + 1) * 10 = 10.010$$

Число параметров

- Можно добиться хорошего качества полносвязными сетями (с аугментацией)
- <https://arxiv.org/abs/1003.0358>

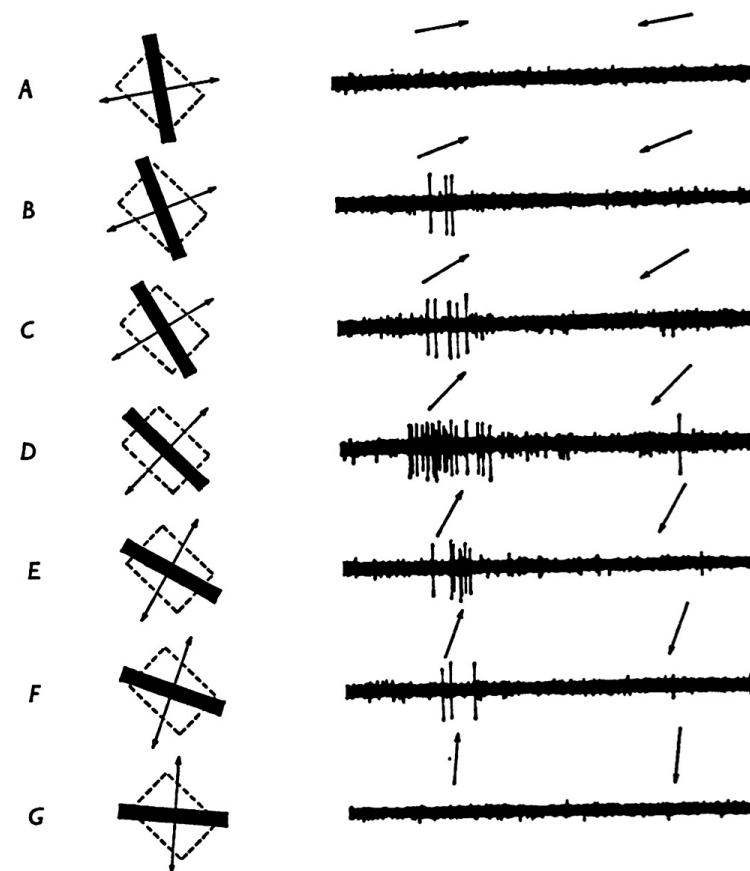
Table 1: Error rates on MNIST test set.

ID	architecture (number of neurons in each layer)	test error for best validation [%]	best test	simulation	weights
			error [%]	time [h]	[milions]
1	1000, 500, 10	0.49	0.44	23.4	1.34
2	1500, 1000, 500, 10	0.46	0.40	44.2	3.26
3	2000, 1500, 1000, 500, 10	0.41	0.39	66.7	6.69
4	2500, 2000, 1500, 1000, 500, 10	0.35	0.32	114.5	12.11
5	$9 \times 1000, 10$	0.44	0.43	107.7	8.86

Полносвязные слои для изображений

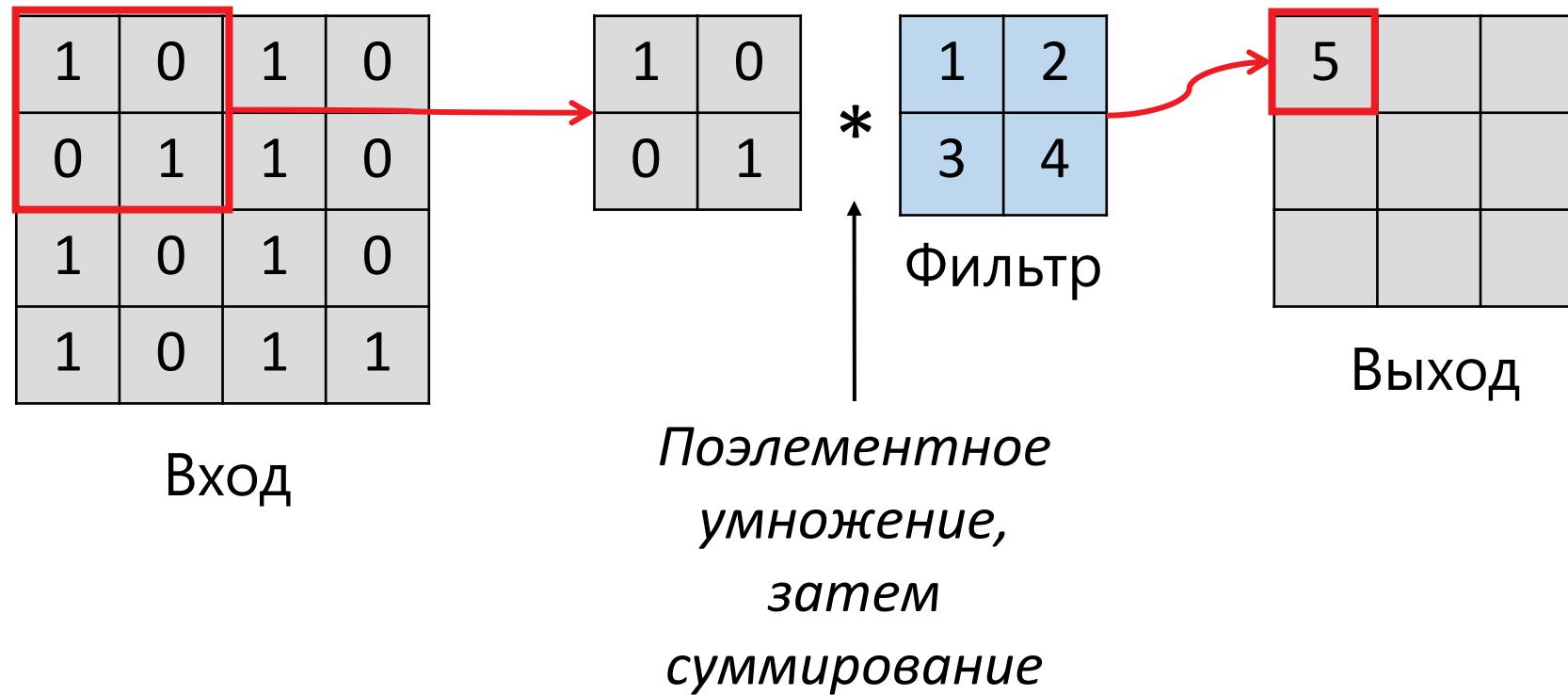
- Очень много параметров
- Легко могут переобучиться
- Не учитывают специфику изображений (сдвиги, небольшие изменения формы и т.д.)
- Один из лучших способов борьбы с переобучением — снижение числа параметров

Эксперименты со зрительной корой



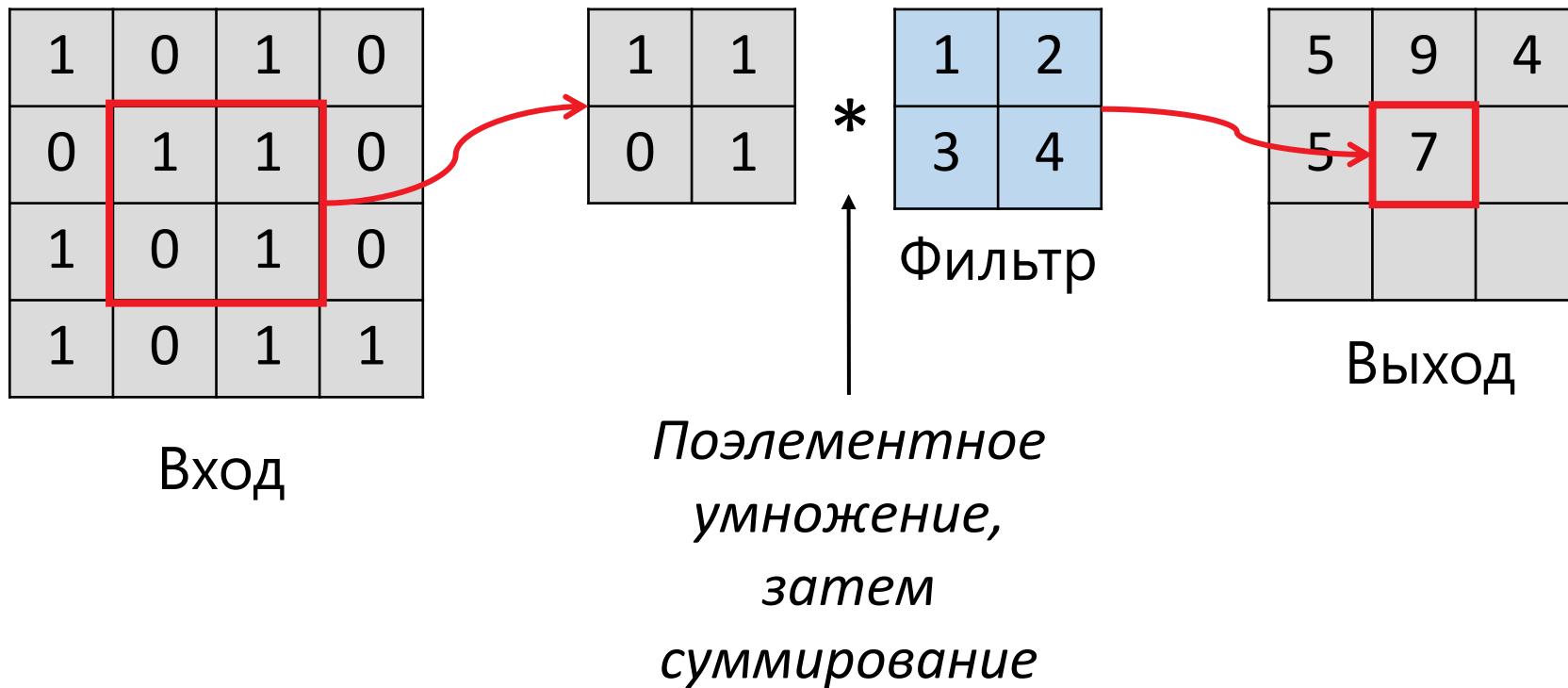
Свёртка

Поле восприятия
(receptive field)



Свёртка

Поле восприятия
(receptive field)



Свёртка

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 2$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 2$$

$$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 6$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 1$$

$$\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 10$$

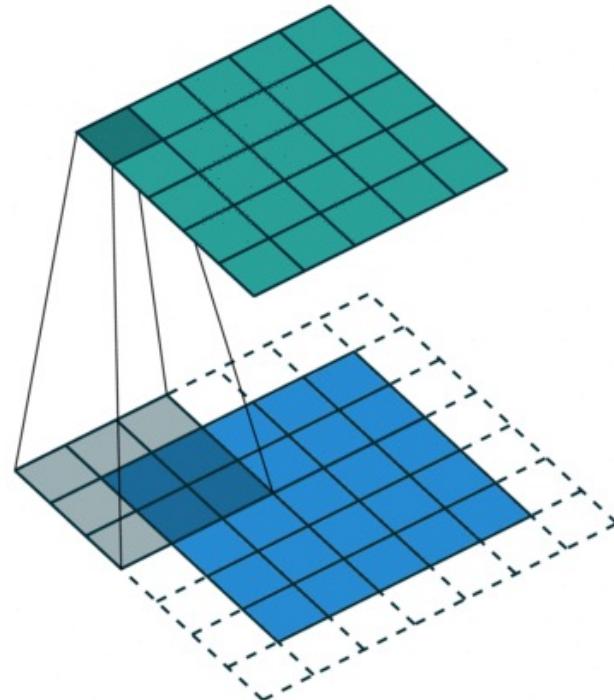
$$\begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

Свёртка

- Операция свёртки выявляет наличие на изображении паттерна, который задаётся фильтром
- Чем сильнее на участке изображения представлен паттерн, тем больше будет значение свёртки

Свёртка

- Результат свёртки изображения с фильтром — новое изображение



Свёртка

3 ₀	3 ₁	2 ₂	1	0
0 ₂	0 ₂	1 ₀	3	1
3 ₀	1 ₁	2 ₂	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3 ₀	2 ₁	1 ₂	0
0	0 ₂	1 ₂	3 ₀	1
3	1 ₀	2 ₁	2 ₂	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2 ₀	1 ₁	0 ₂
0	0	1 ₂	3 ₂	1 ₀
3	1	2 ₀	2 ₁	3 ₂
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0 ₀	0 ₁	1 ₂	3	1
3 ₂	1 ₂	2 ₀	2	3
2 ₀	0 ₁	0 ₂	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0 ₀	1 ₁	3 ₂	1
3	1 ₂	2 ₂	2 ₀	3
2	0 ₀	0 ₁	2 ₂	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0	1 ₀	3 ₁	1 ₂
3	1	2 ₂	2 ₂	3 ₀
2	0	0 ₀	2 ₁	2 ₂
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0	1	3	1
3 ₀	1 ₁	2 ₂	2	3
2 ₂	0 ₂	0 ₀	2	2
2 ₀	0 ₁	0 ₂	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0	1	3	1
3	1 ₀	2 ₁	2 ₂	3
2	0 ₂	0 ₂	2 ₀	2
2	0 ₀	0 ₁	0 ₂	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0	1	3	1
3	1	2 ₀	2 ₁	3 ₂
2	0	0 ₂	2 ₂	2 ₀
2	0	0 ₀	0 ₁	1 ₂

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Свёртка инвариантна к сдвигам

0	0	0	0
0	0	0	0
0	0	1	0
0	0	0	1

Вход

*

1	0
0	1

Фильтр

=

0	0	0
0	1	0
0	0	2

Выход

Max = 2

1	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0

Вход

*

1	0
0	1

Фильтр

=

2	0	0
0	1	0
0	0	0

Выход

Не меняется

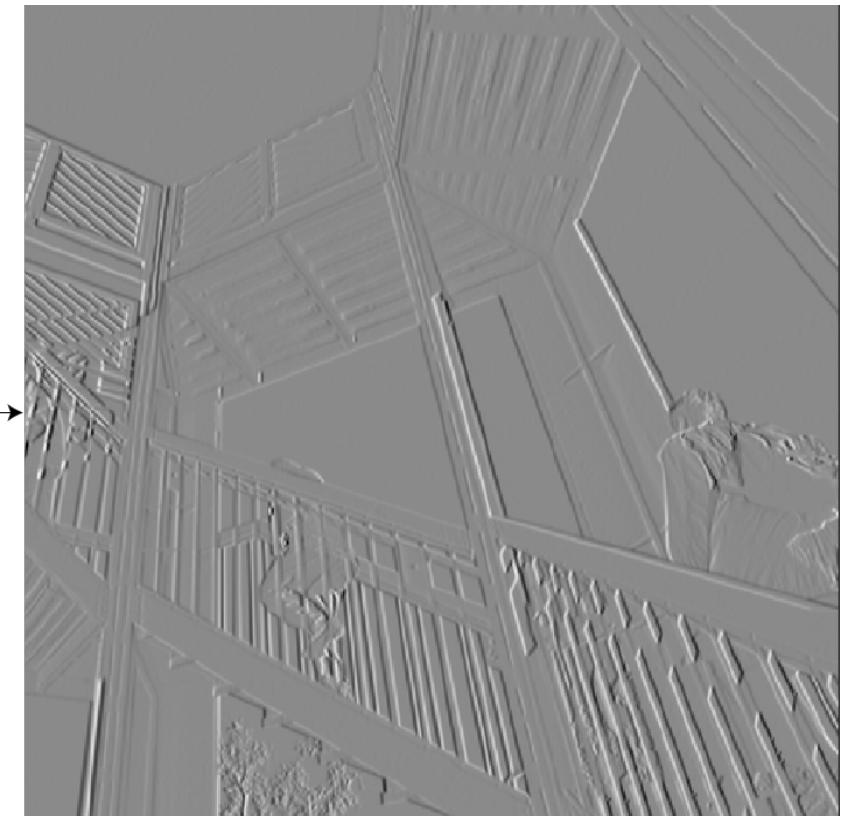
Max = 2

Свёртки в компьютерном зрении



$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

Horizontal Sobel kernel

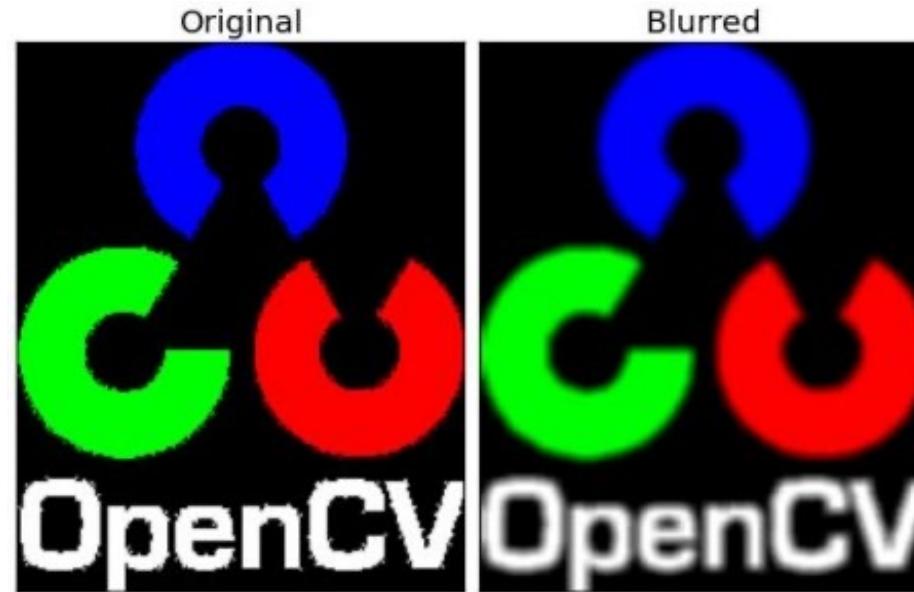


Свёртки в компьютерном зрении



$$\begin{array}{|c|c|c|} \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \bullet 0 & \bullet 1 & \bullet 0 \\ \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \bullet 0 & \bullet 1 & \bullet 0 \\ \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|} \hline \bullet 1 & \bullet 1 & \bullet 1 \\ \hline \bullet 1 & \bullet 1 & \bullet 1 \\ \hline \bullet 1 & \bullet 1 & \bullet 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \bullet 0 & \bullet 2 & \bullet 0 \\ \hline \bullet 0 & \bullet 0 & \bullet 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|} \hline \bullet 1 & \bullet 1 & \bullet 1 \\ \hline \bullet 1 & \bullet 1 & \bullet 1 \\ \hline \bullet 1 & \bullet 1 & \bullet 1 \\ \hline \end{array}$$

Свёртки в компьютерном зрении



$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Pooling

1	0	2	1	0	0
0	1	3	2	1	2



1	3	2

Max-pooling с фильтром 2x2

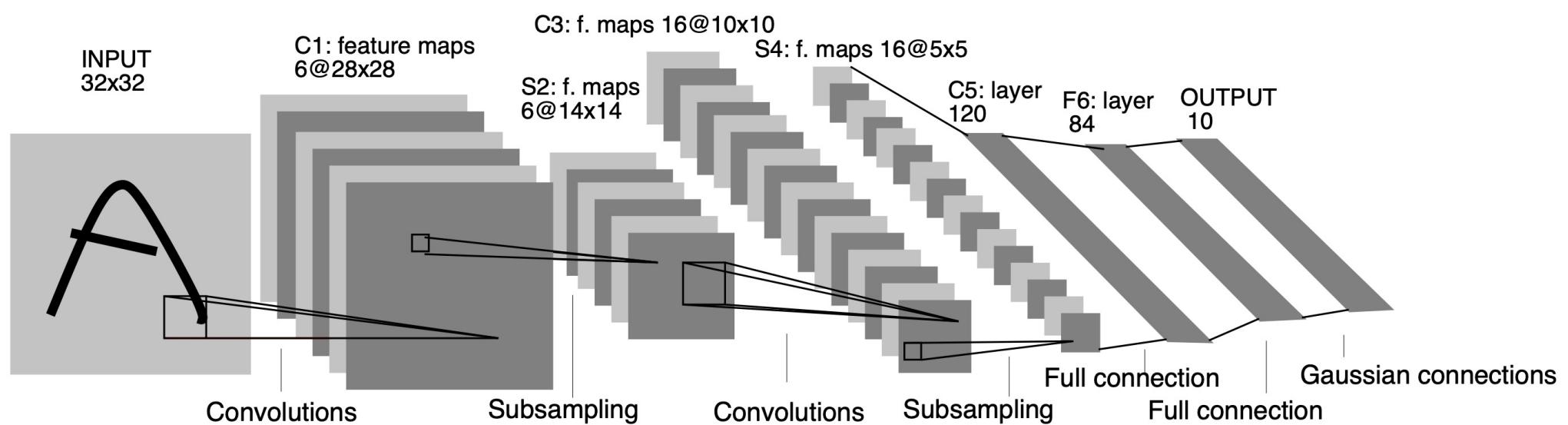
Pooling

- Разбивает изображение на участки $n \times m$ и считает некоторую статистику в каждом участке (обычно максимум)
- Существенно сокращает размер изображения (значит, увеличивает поле восприятия следующих слоёв)
- Не имеет параметров

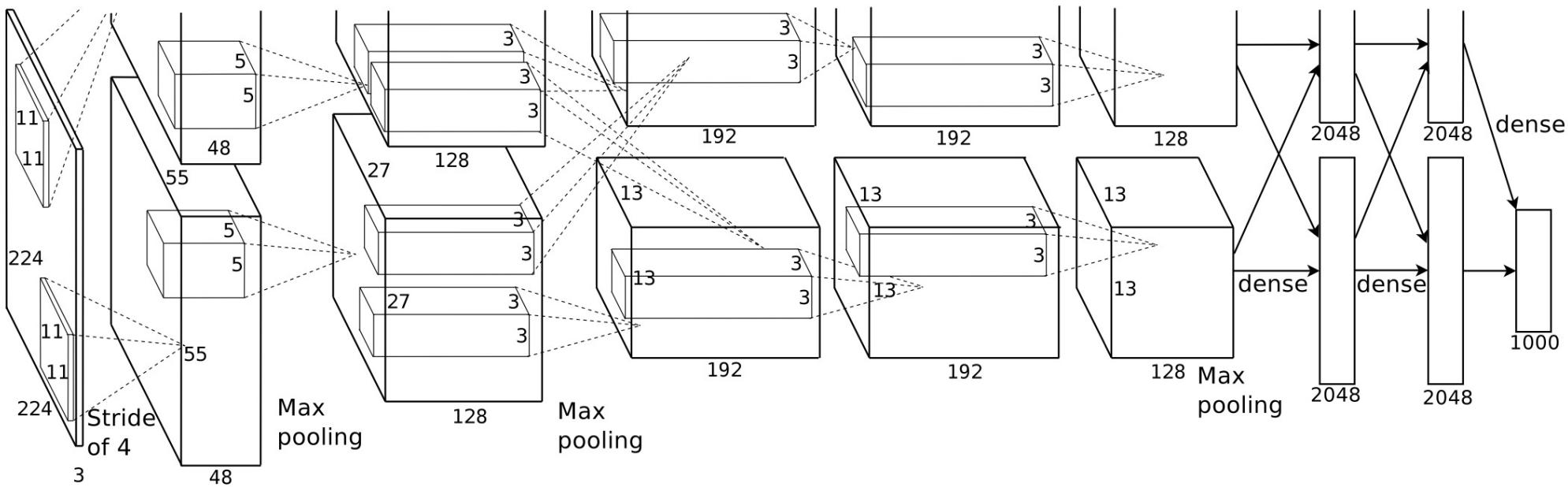
Типичная архитектура

- Последовательное применение комбинаций вида «свёрточный слой -> нелинейность -> pooling» или «свёрточный слой -> нелинейность»
- Выпрямление (flattening) выхода очередного слоя
- Серия полносвязных слоёв

LeNet

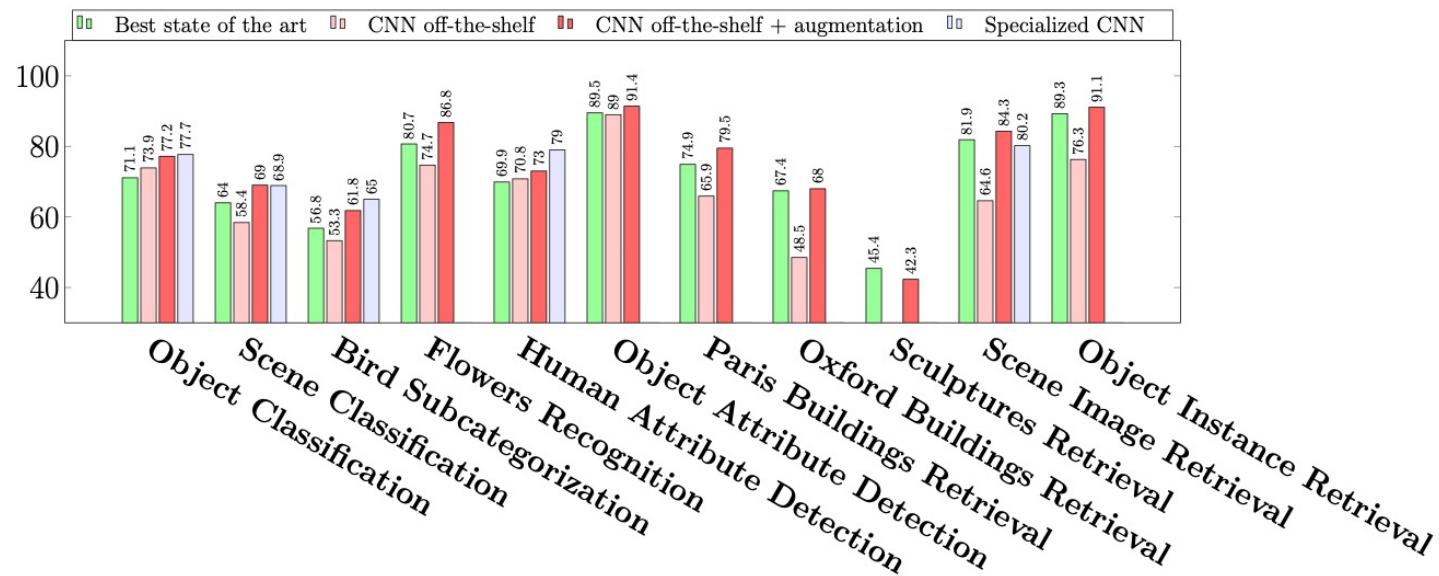
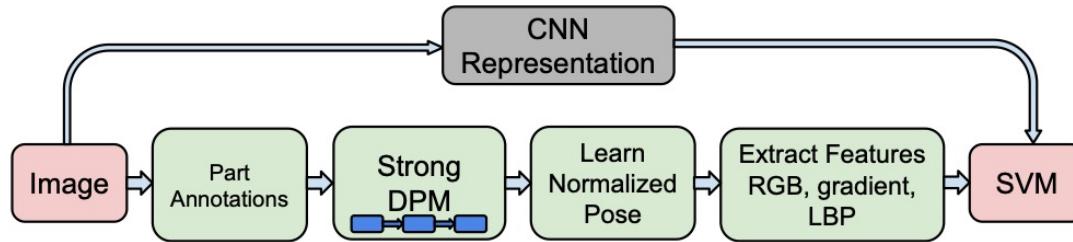


AlexNet



<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

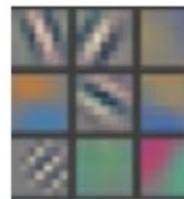
Представления с последних слоёв



Представления с последних слоёв

- Не интерпретируется (в отличие от классического компьютерного зрения)
- По смыслу — «индикаторы» наличия каких-то паттернов

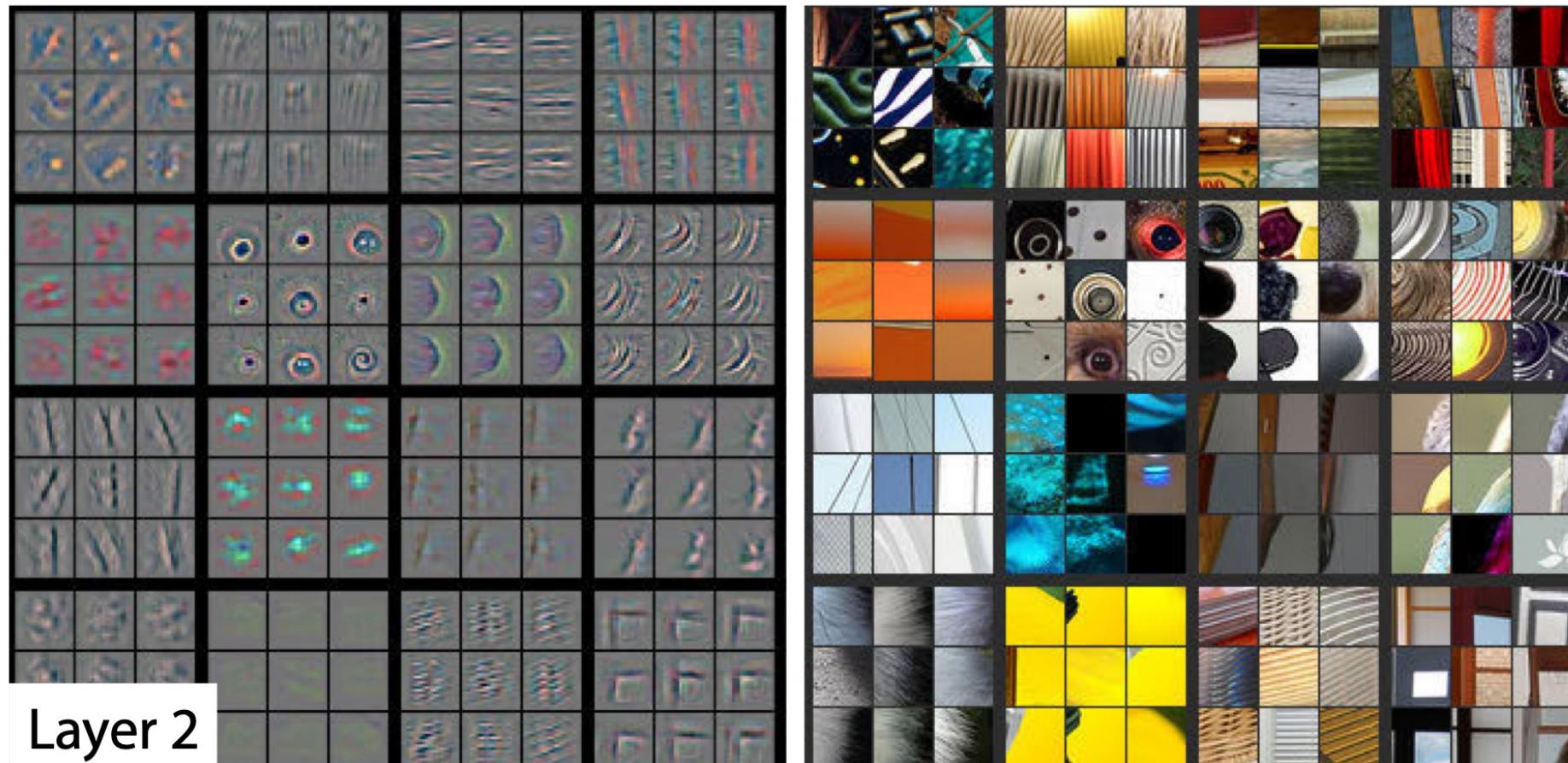
Представления с последних слоёв



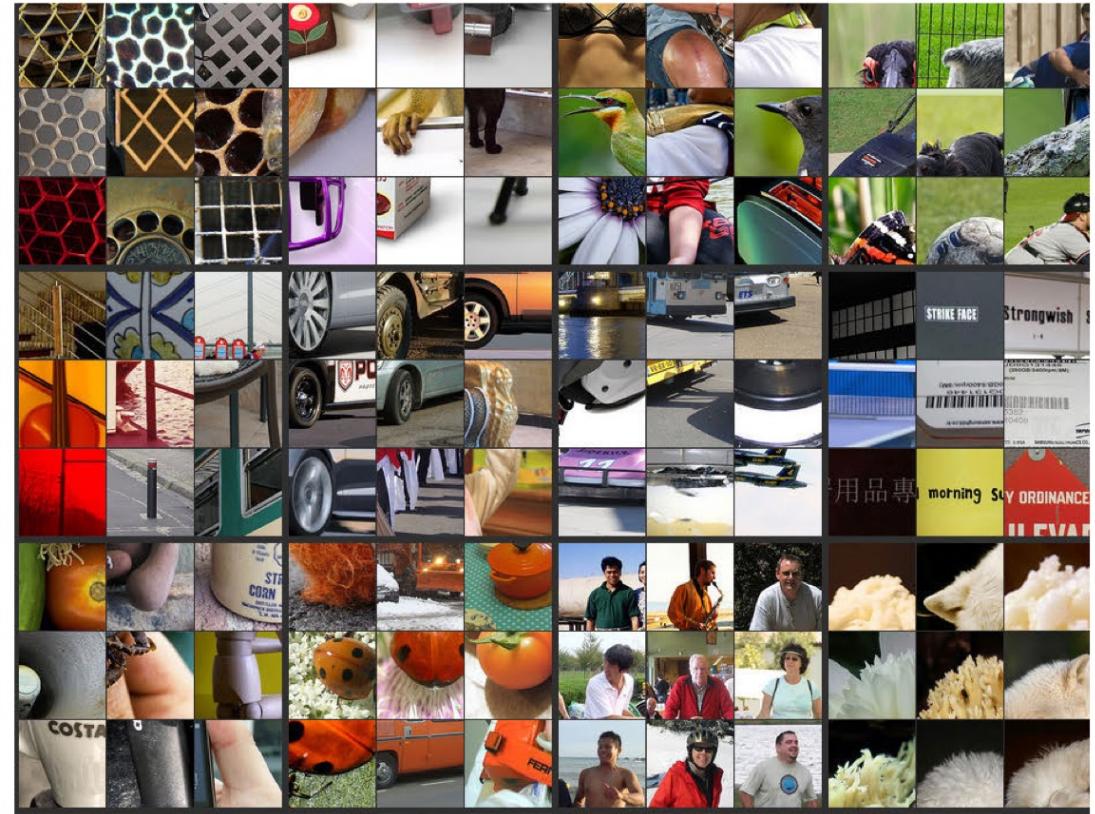
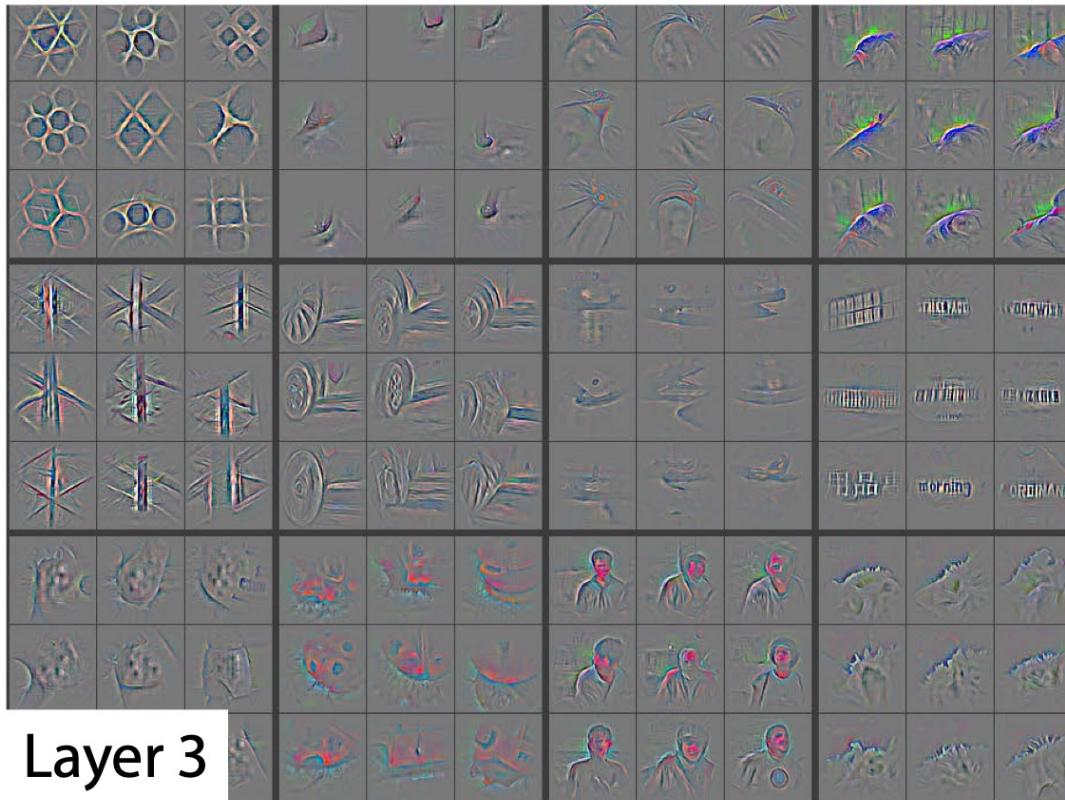
Layer 1



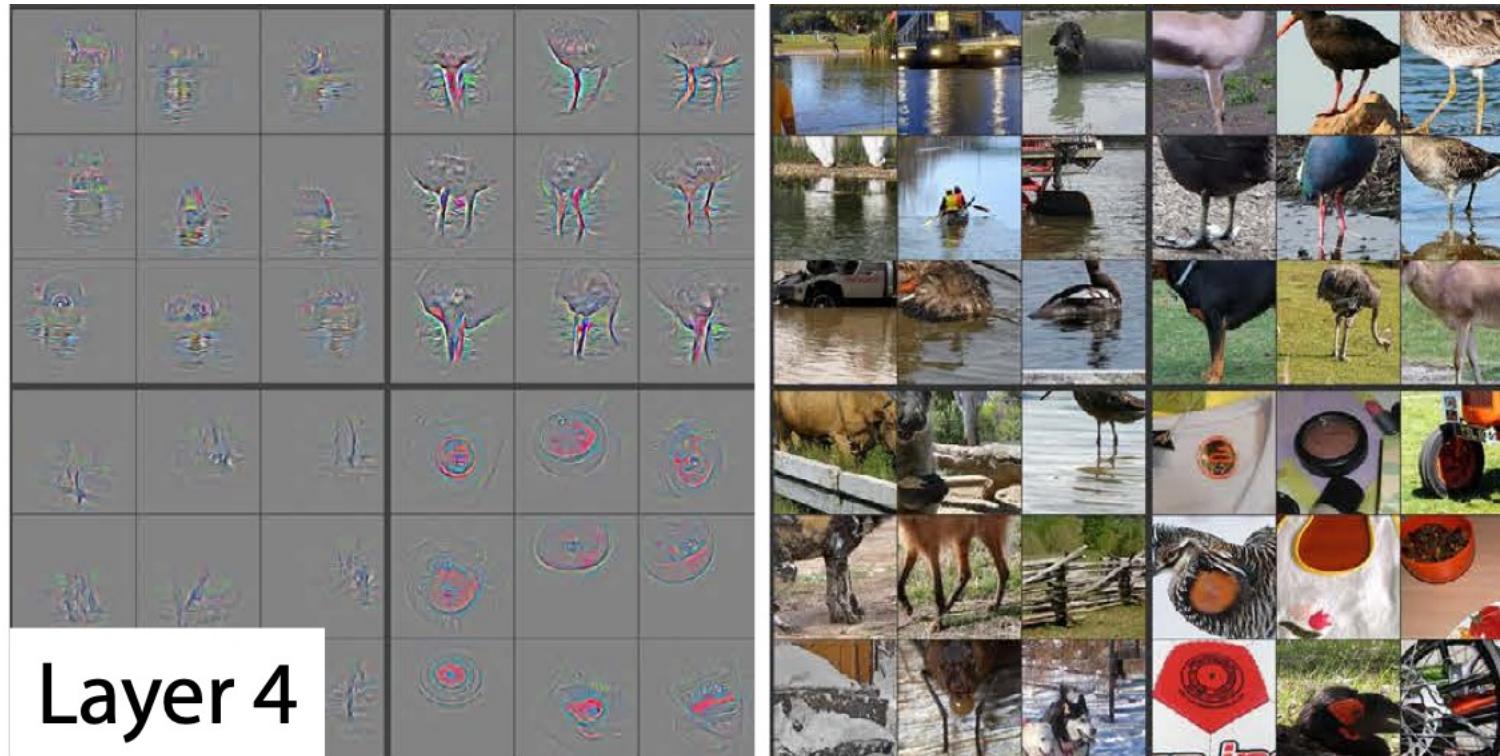
Представления с последних слоёв



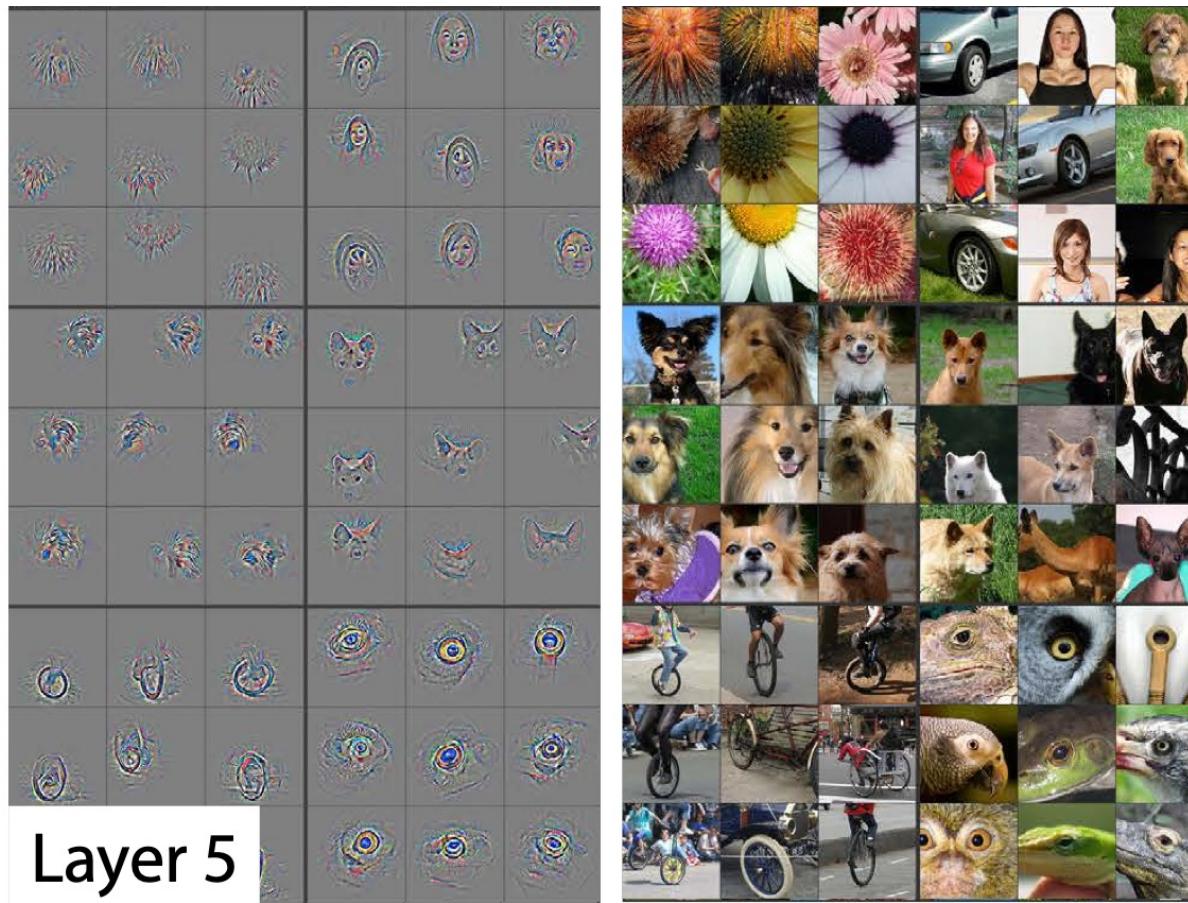
Представления с последних слоёв



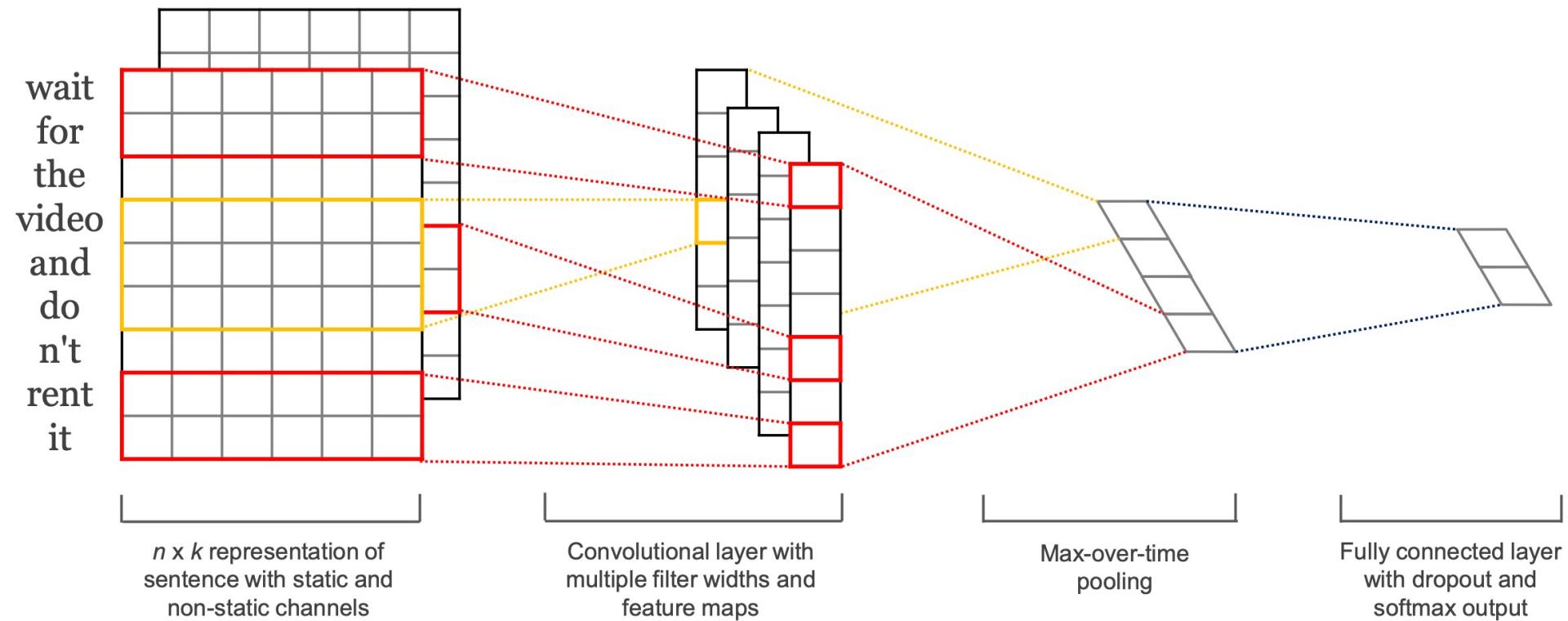
Представления с последних слоёв



Представления с последних слоёв



CNN для последовательностей



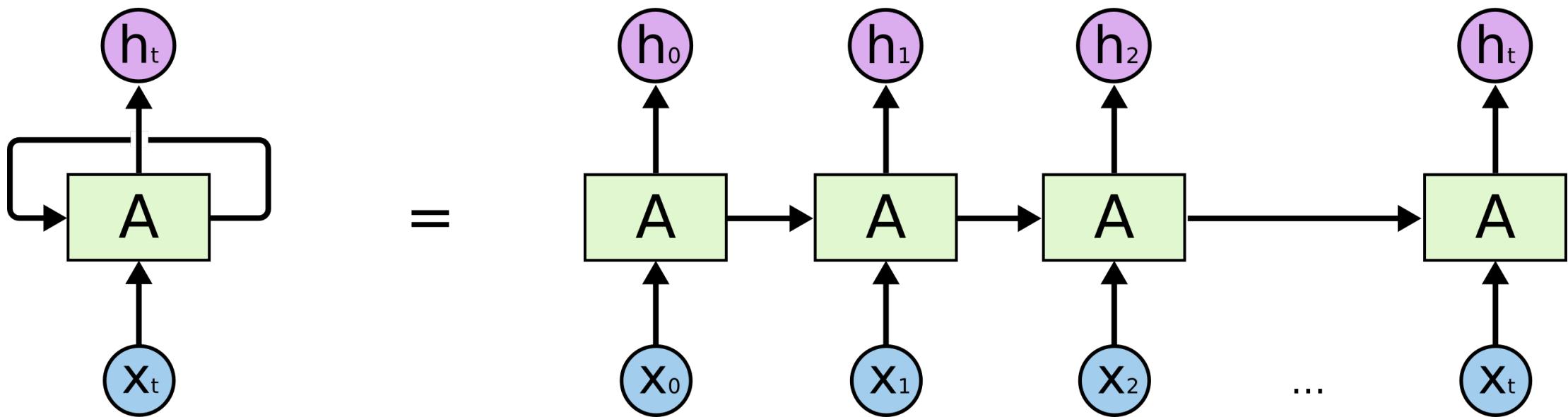
CNN для последовательностей

- Можно обучать представления слов с нуля
- А можно инициализировать с помощью w2v — это сильно лучше!

Сфера использования

- Компьютерное зрение:
 - распознавание лица
 - детекция лица
 - детекция объектов
 - распознавание символов
 - диагностика в медицине

4. Recurrent Neural Networks



Рекуррентные сети

- Мы читаем текст последовательно
- И постепенно всё лучше понимаем, о чём он

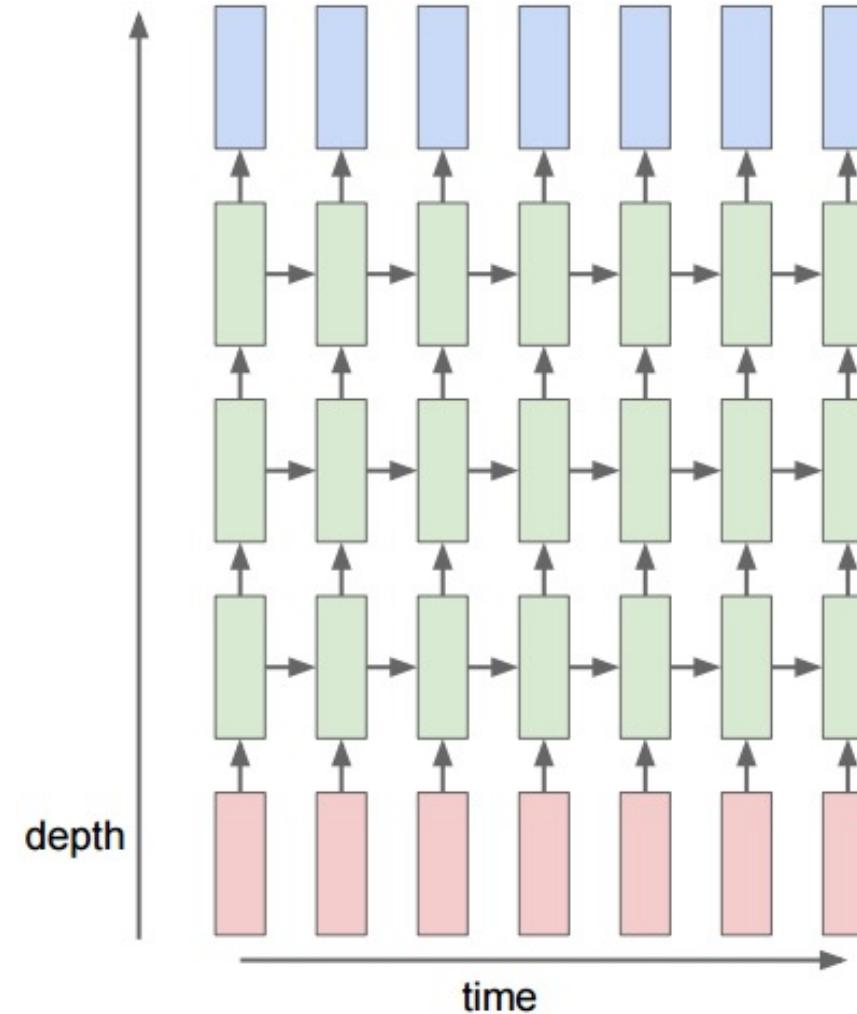
Рекуррентные сети

- Последовательность: $x_1, x_2, \dots, x_n, \dots$
- Читаем слева направо
- h_t — накопленная информация после чтения t элементов (вектор)

Рекуррентные сети (RNN)

- Последовательность: $x_1, x_2, \dots, x_n, \dots$
- Читаем слева направо
- h_t — накопленная информация после чтения t элементов (вектор)
- $h_t = f(W_{xh}x_t + W_{hh}h_{t-1})$
- Если хотим что-то выдавать на каждом шаге: $o_t = f_o(W_{ho}h_t)$

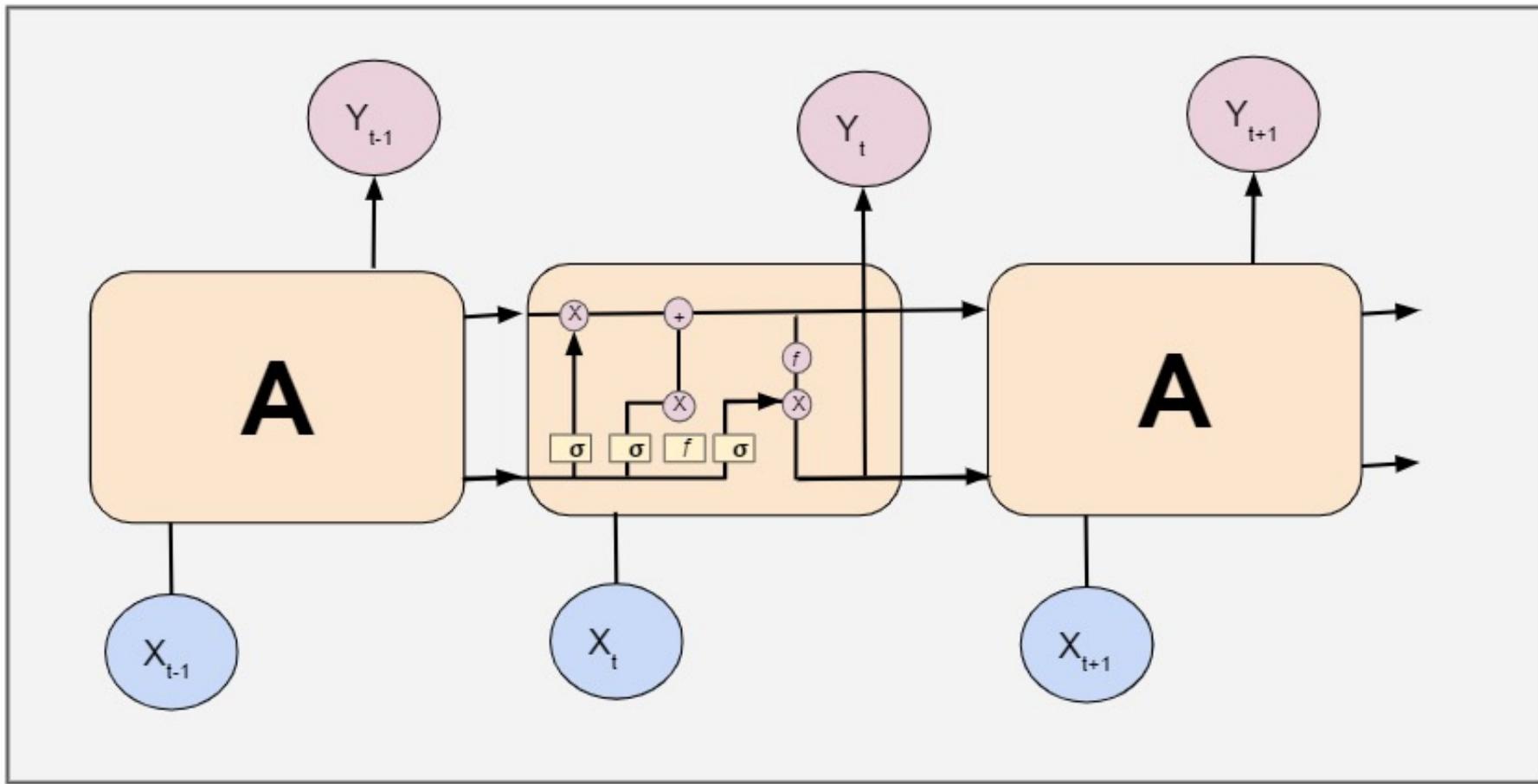
Можно делать многослойные RNN



Сфера использования

- Для последовательностей
- Прогноз временных рядов
- Генерация текстов
- Генерация музыки

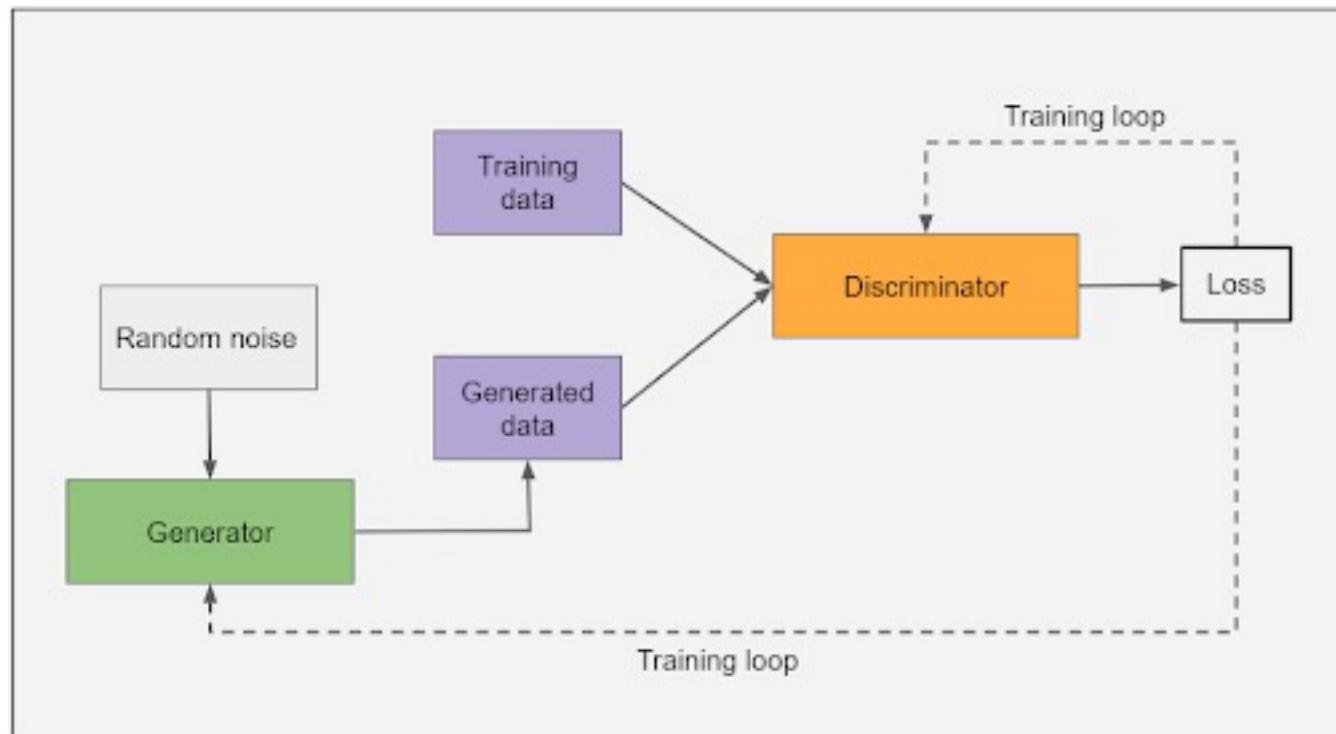
5. Long Short-Term Memory Networks



Сфера использования

- Переводы
- Детекция аномалий
- Распознавание речи
- Классификация видео

6. Generative Adversarial Networks

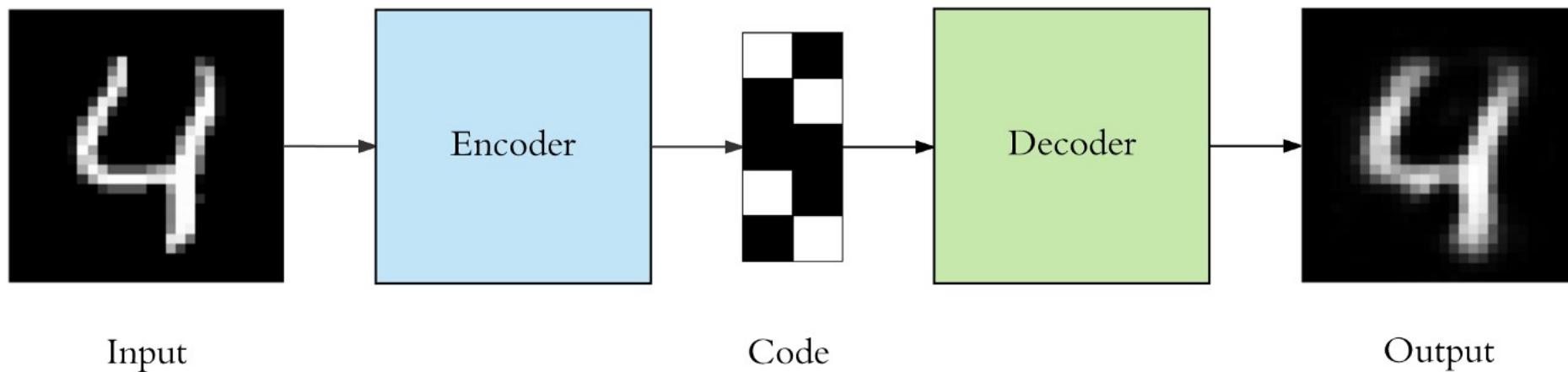




Сфера использования

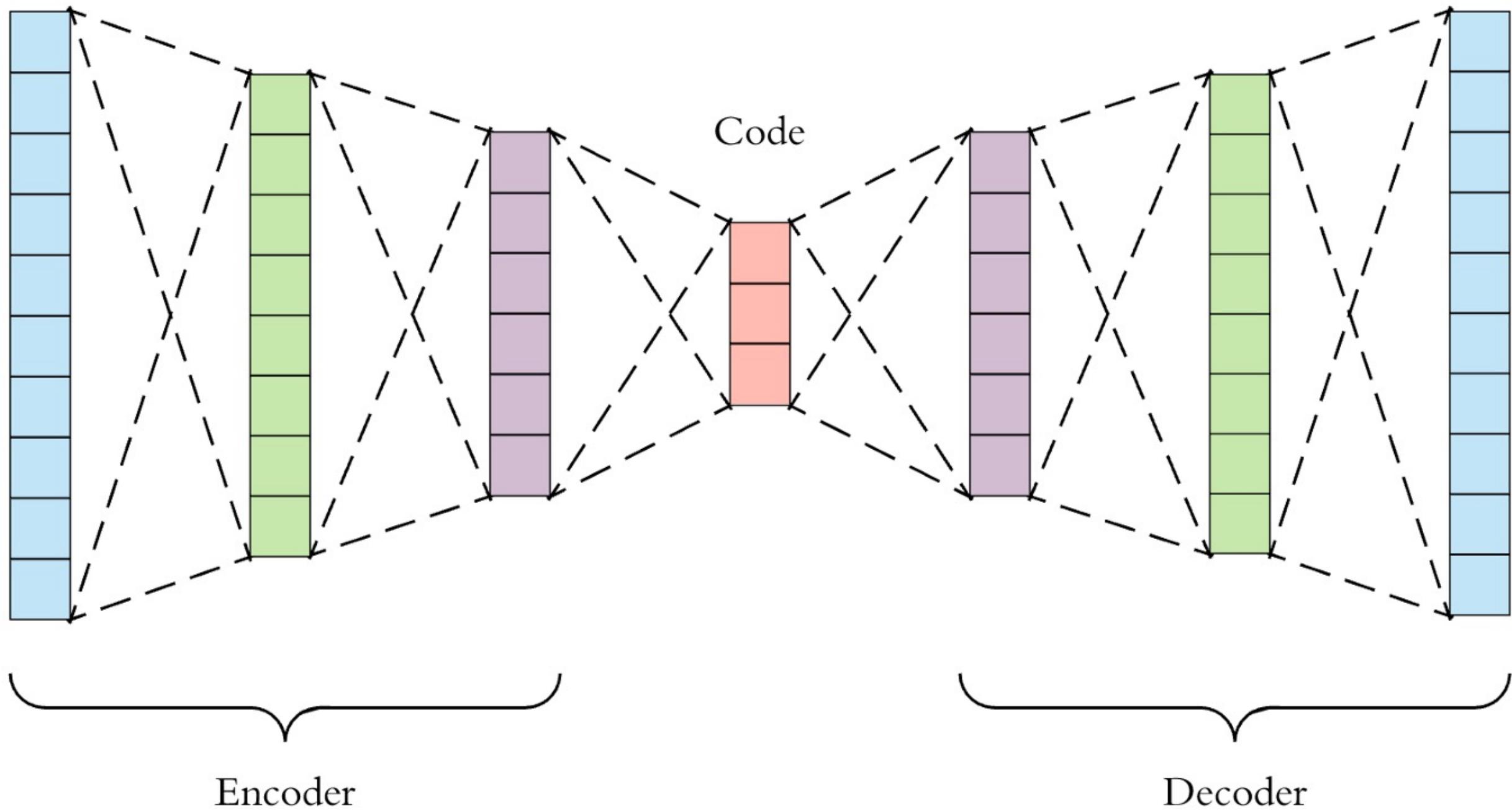
- Генерация изображений
- Генерация видео

7. Autoencoders



Input

Output



Сфера применения

- Понижение размерности
- Сжатие данных
- Кластеризация
- Генерация новых объектов

8. Transformers

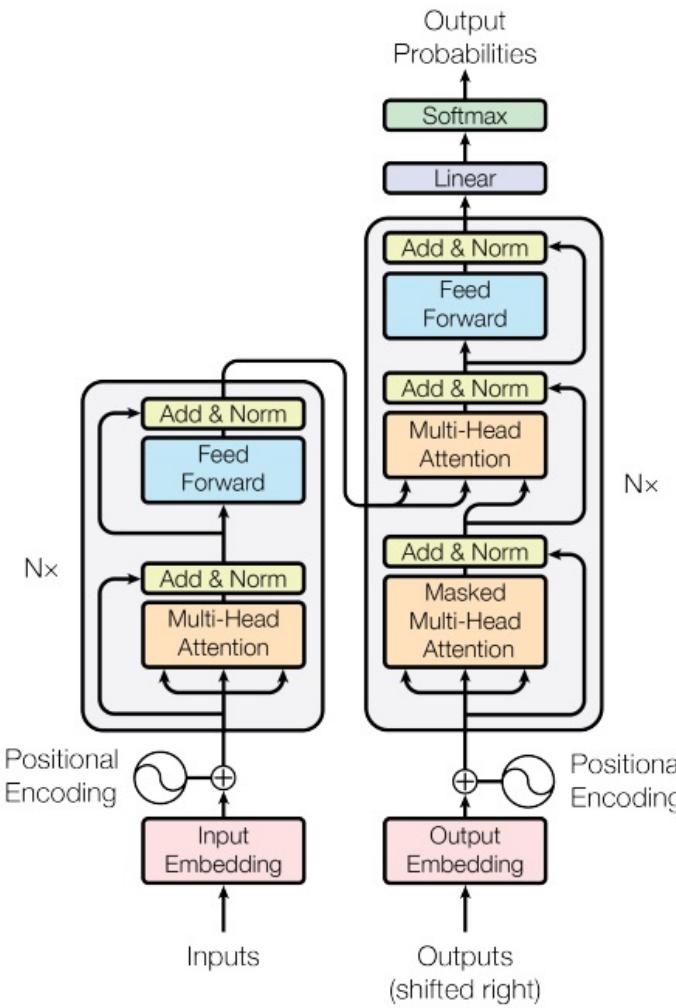


Figure 1: The Transformer - model architecture.

Сфера применения

- языковые представления
- переводы

Резюме

- Типов нейронных сетей много
- В зависимости от природы данных используются разные нейронки
- Иногда применение известной нейронной сети к новой задаче – win-win-стратегия