

# Глубокие свёрточные нейронные сети

Алексей Артёмов

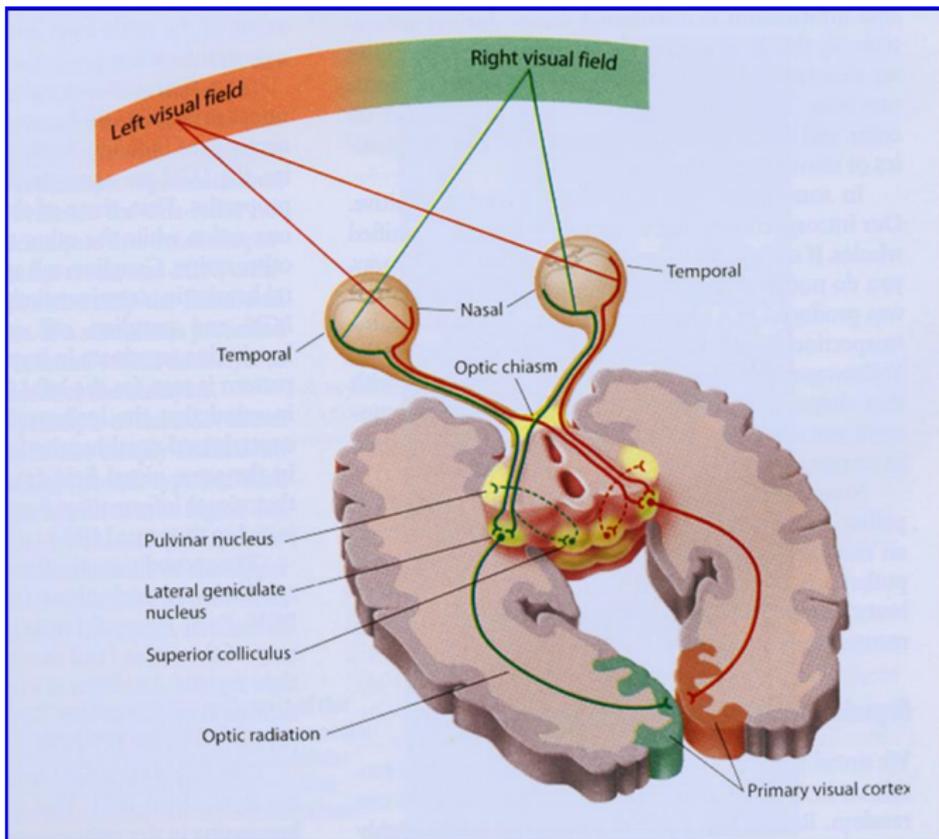
17 июля 2016 г.

JetBrains,  
Санкт-Петербург

# Содержание

- 1 Мотивация
- 2 Свертка
- 3 Сверточный слой
- 4 Сверточная сеть

# Зрительная система млекопитающих



# Зрительная система млекопитающих

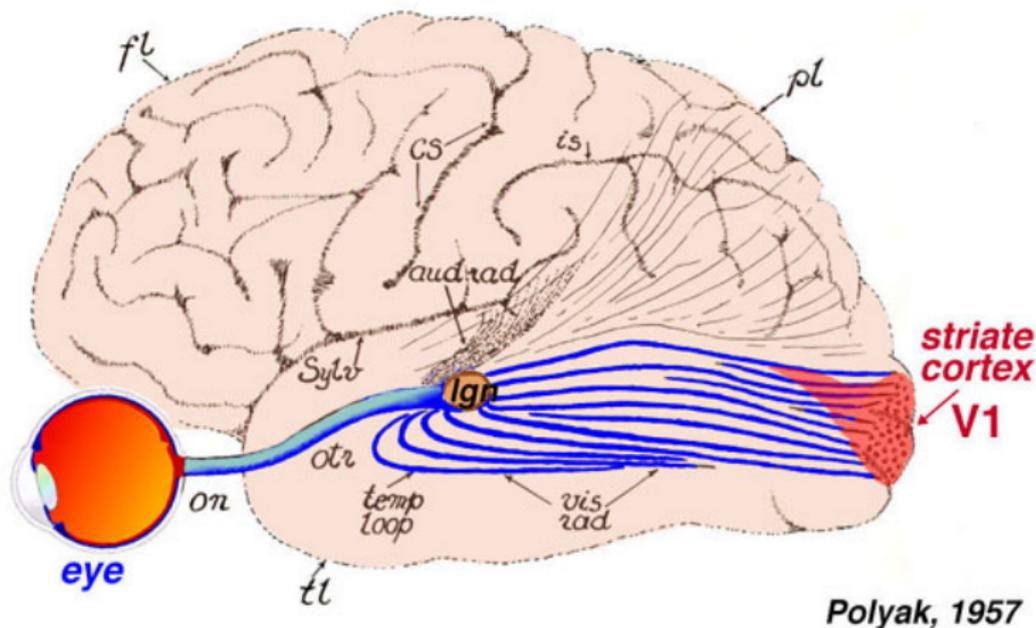
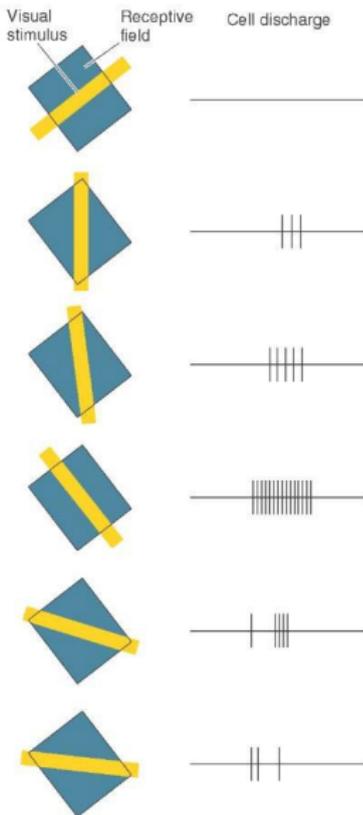
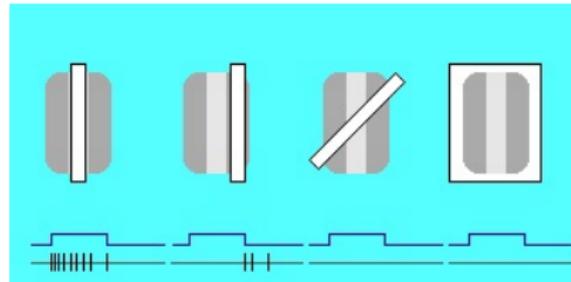


Figure 8. Visual input to the brain goes from eye to LGN and then to primary visual cortex, or area V1, which is located in the posterior of the occipital lobe.  
 Adapted from Polyak (1957).

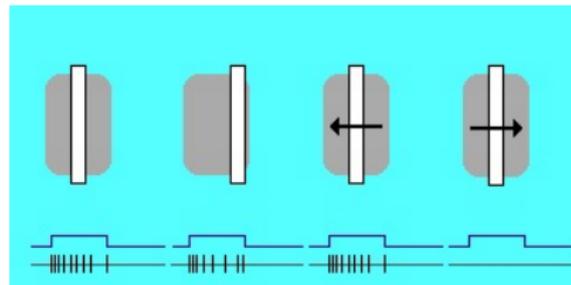
# Клетки зрительной коры — детекторы признаков!



## Простые клетки:



## Сложные клетки:



## Иерархия представлений

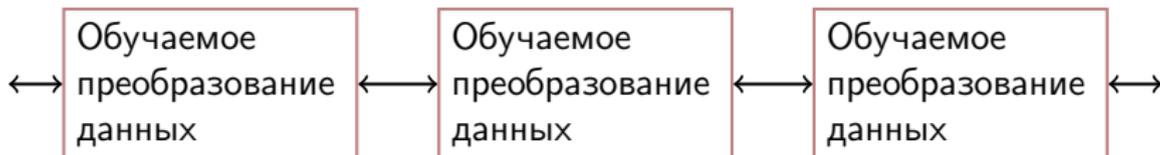
- Сетчатка - LGN - V1 - V2 - V4 - IT - ...

# Иерархия представлений

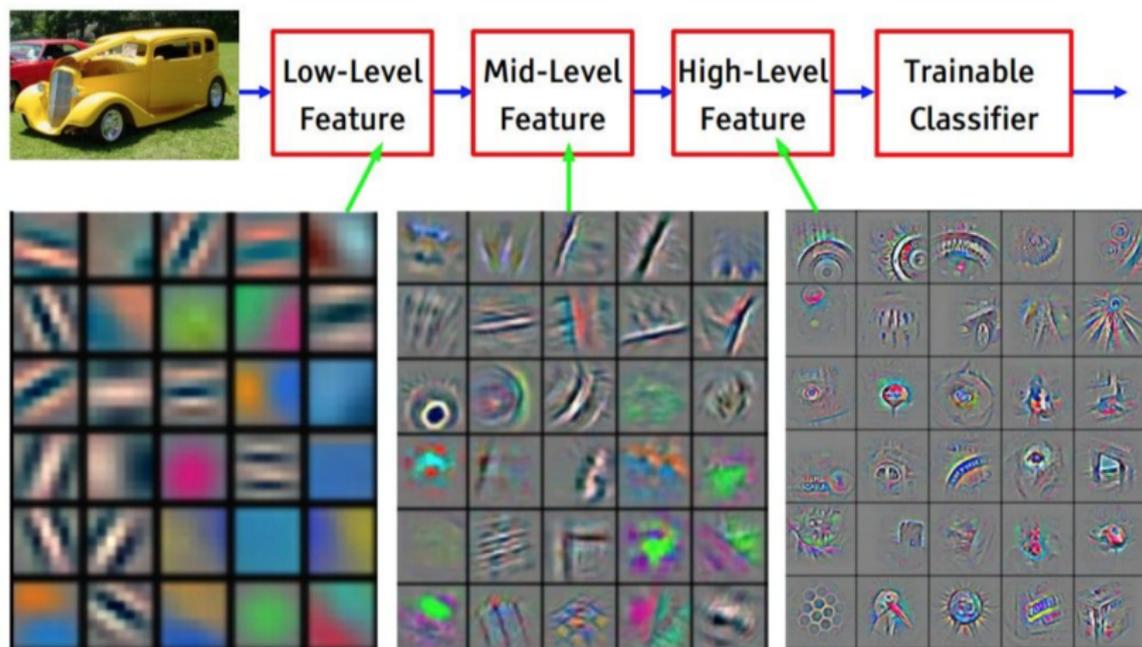
- Сетчатка - LGN - V1 - V2 - V4 - IT - ...
- Иерархия представлений с увеличением уровня абстракции
- Каждый этап — это обучаемое преобразование признаков
- Распознавание изображений
  - Пиксель → ребро → банк фильтров → шаблон → часть → объект
- Текст
  - Символ → слово → группа слов → фраза → предложение → текст
- Речь
  - Отсчет → полоса спектра → звук → фонема → слово

# Иерархия представлений

- Сетчатка - LGN - V1 - V2 - V4 - IT - ...
- Иерархия представлений с увеличением уровня абстракции
- Каждый этап — это обучаемое преобразование признаков
- Распознавание изображений
  - Пиксель → ребро → банк фильтров → шаблон → часть → объект
- Текст
  - Символ → слово → группа слов → фраза → предложение → текст
- Речь
  - Отсчет → полоса спектра → звук → фонема → слово



# Иерархия представлений



(LeCun Slide)

# Операция свертки

- $x(t)$  — некоторая функция  $t \in \mathbb{R}$
- Свертка  $x(t)$  с ядром  $k(t)$  — это функция

$$s(t) = (x * k)(t) \equiv \int_{-\infty}^{\infty} x(\tau)k(t - \tau)d\tau$$

- Если  $t \in \mathbb{Z}$  (дискретна)

$$s(t) = (x * k)(t) \equiv \sum_{\tau=-\infty}^{\infty} x(\tau)k(t - \tau)$$

- Образ свертки  $s(t)$  — карта откликов (feature map)

# Свертка изображения с ядром

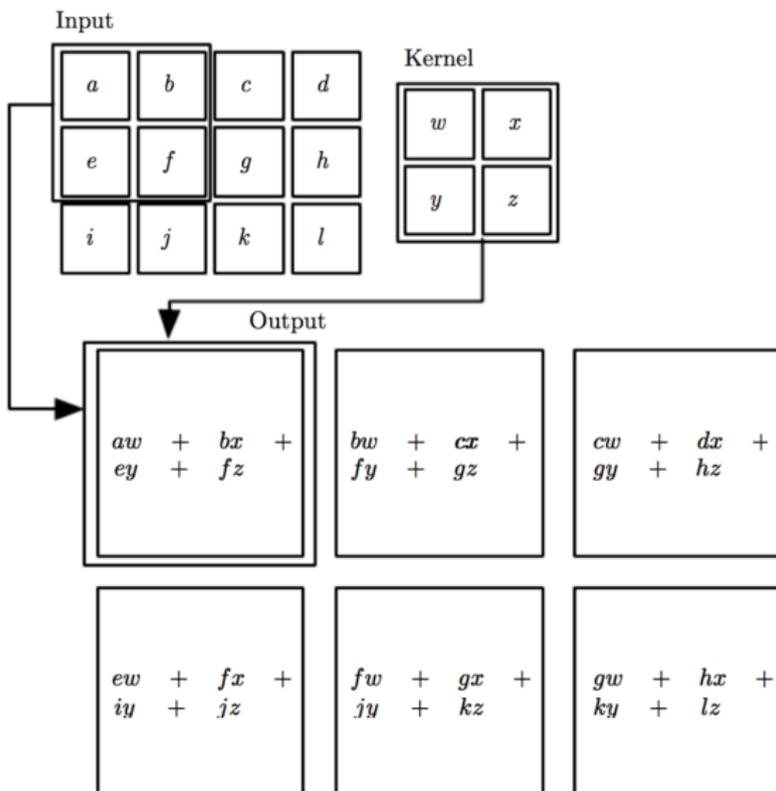
- $I(i, j)$  — изображение
- Свертка изображения  $I(i, j)$  с ядром  $K(t, s)$

$$S(i, j) = (I * K)(i, j) \equiv \sum_m \sum_n I(m, n) K(i-m, j-n)$$

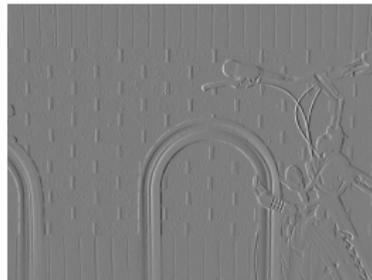
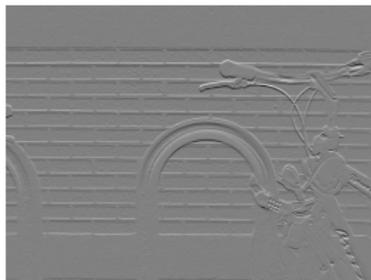
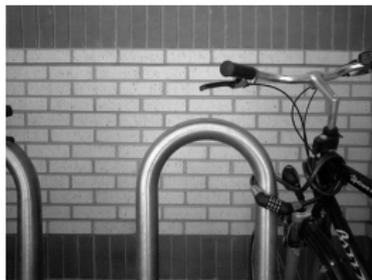
- Эквивалентно (коммутативность):

$$S(i, j) = (I * K)(i, j) \equiv \sum_m \sum_n I(i-m, j-n) K(m, n)$$

# Свертка изображения с ядром



# Свертка с ядром Собеля 3x3

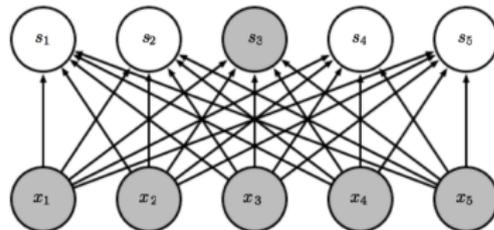
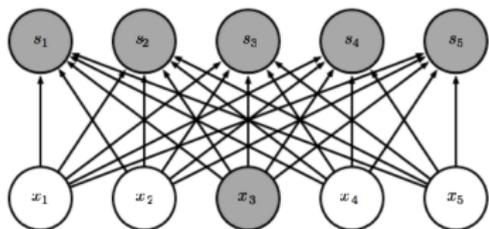
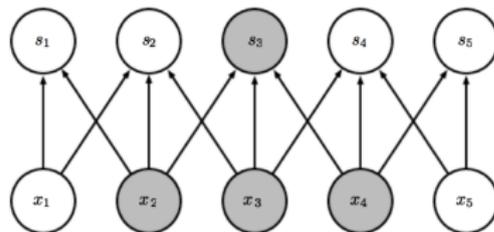
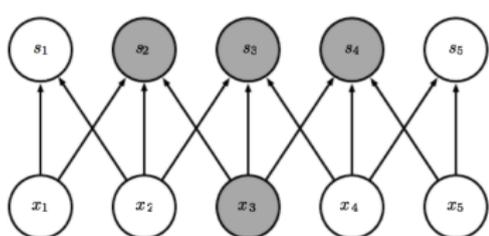


## Зачем нужны свертки?

- Разреженность взаимодействия нейронов
- Разделяемые (общие) параметры
- Эquivариантность представления
- + Работа со входом различного размера

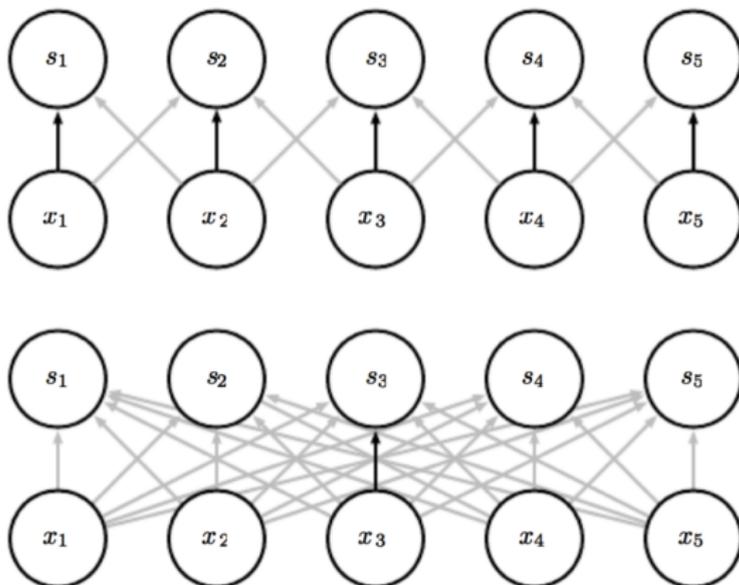
# Разреженные веса между нейронами

- Ядро имеет меньший размер, чем изображение
- Извлечение локальных признаков
- Храним меньше параметров, статистическое поведение модели лучше, вычисляем меньше



## Разделяемые (совместные) параметры

- Один набор параметров (ядро) используется в различных частях модели
- Каждый параметр используется многократно

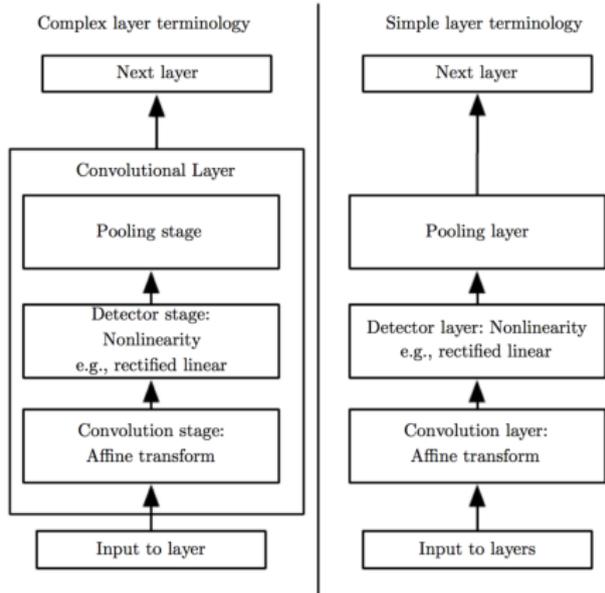


## Эквивариантность к переносам

- Эквивариантность: если изменяется вход, то выход изменяется так же
- Функция  $f(x)$  эквивариантна к функции  $g(y)$ , если  $f(g(y)) = g(f(x))$
- Свертка  $f(I)$  эквивариантна к параллельному переносу:
  - $I' = g(I) \implies I'(x, y) = I(x - 1, y)$
  - Тогда результат свертки  $f(I') = g(f(I))$

# Архитектура сверточного слоя

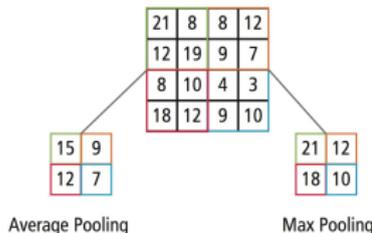
1. Свертка входа с множеством ядер
2. Пропускание откликов через нелинейную активацию (детектор)
3. Объединение соседних активаций (pooling)



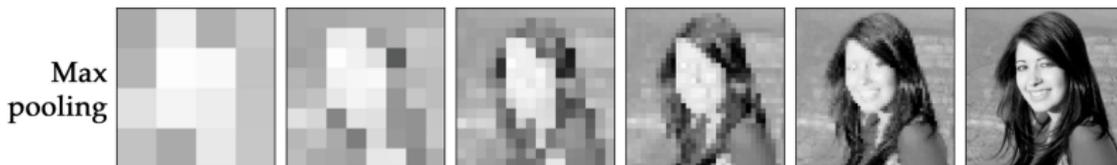
# Pooling-слой

- Замена выхода нейрона статистикой, подсчитанной по его соседям

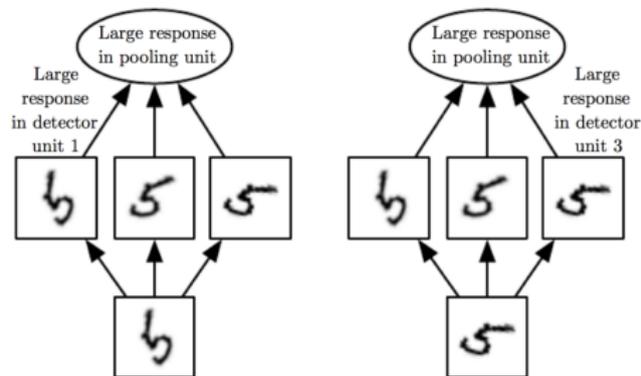
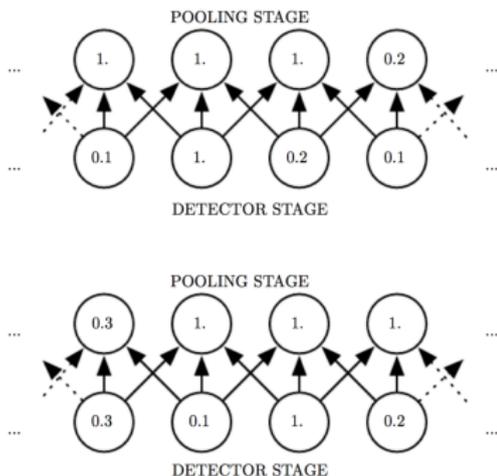
- max pooling
- (weighted) average pooling



- Обеспечивает приблизительную инвариантность выхода к малому переносу входа
- Наличие признака важнее, чем его точная позиция!



# Pooling-слой и инвариантные представления



- Изменение входных данных может достигать 100%, но изменение выхода остается небольшим
- Инвариантность к более сложным преобразованиям входных данных (поворот, ...)

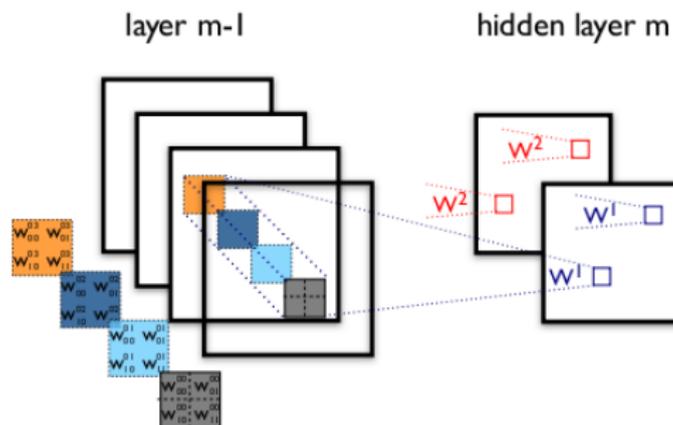
## Реализация сверточного слоя

- Входное цветное изображение — тензор глубины 3 (RGB)
- Входное изображение второго сверточного слоя — тензор, глубина которого равна числу фильтров первого слоя
- Пакетный режим добавляет четвертую размерность (пока опустим)

# Стандартная реализация сверточного слоя

- $V_{i,j,k}$  — пиксель входа (канал  $i$ , строка  $j$ , столбец  $k$ )
- $K_{i,j,k,l}$  — элемент ядра (входной канал  $i$ , выходной канал  $j$ , строка  $k$ , столбец  $l$ )
- Базовое соотношение для свертки

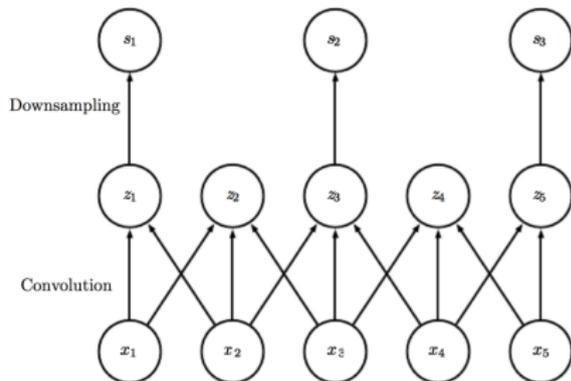
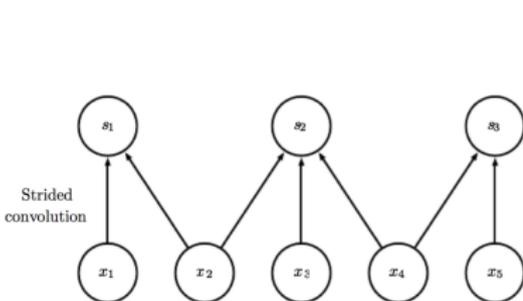
$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1,k+n-1} K_{i,l,m,n}$$



# Сверточный слой с шагом $s$

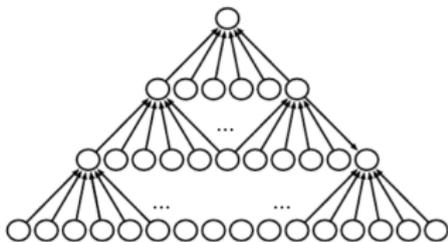
- Уменьшение вычислительных затрат в  $s$  раз
- Шагающая (strided) свертка

$$z_{i,j,k} = \sum_{l,m,n} V_{l,(j-1) \times s + m, (k-1) \times s + n} K_{i,l,m,n}$$

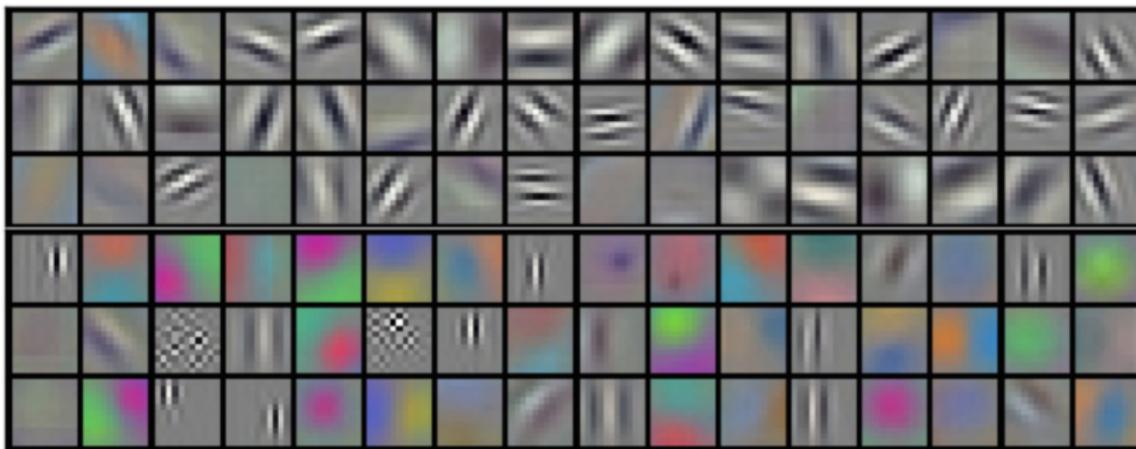


# Неявное заполнение нулями

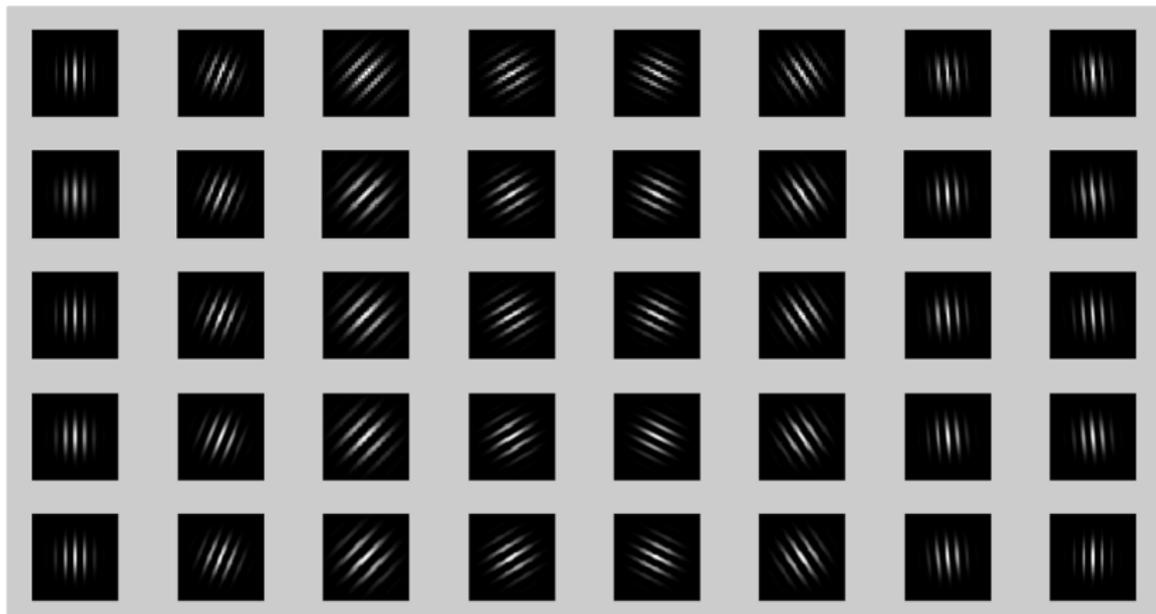
- Выход каждого слоя уменьшается на  $k - 1$  пиксель после каждой свертки ( $k$  - размер ядра)
- Неявное заполнение нулями (zero-padding) — способ управлять шириной ядра и размером выхода независимо:
  - нет заполнения
  - заполнение с сохранением размера выхода
  - заполнение с равным посещением пикселей



# Выученные фильтры [Krizhevsky et al., 2012]



# Фильтры Габора



# Общая архитектура глубокой сверточной сети

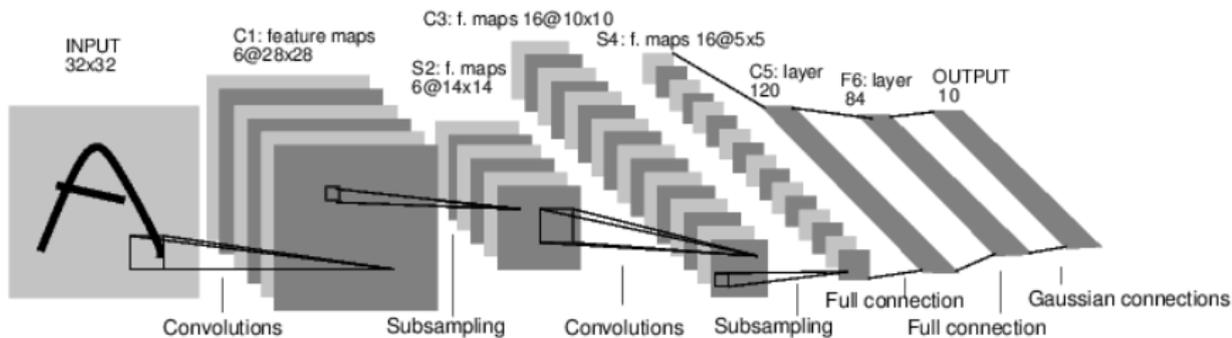
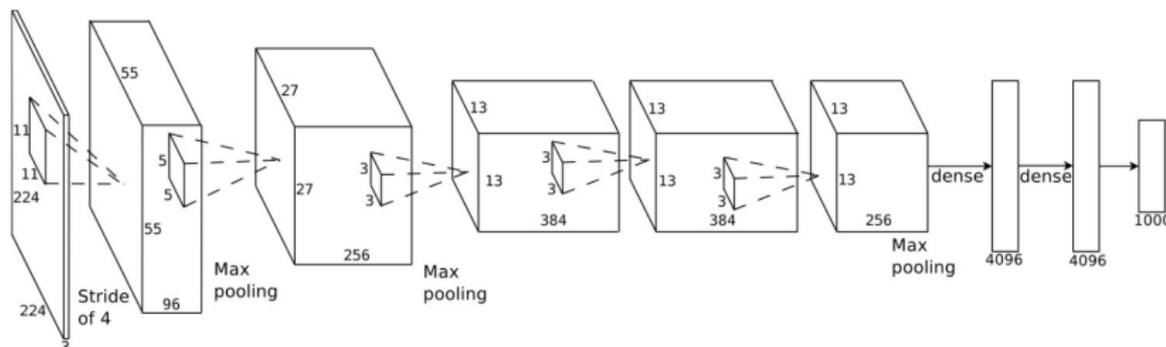


Рис.: LeNet [LeCun et al., 1998]

# Архитектура сети AlexNet [Krizhevsky et al., 2012]



- 60 млн параметров
- 27 Гб данных на диске
- Обучается 1 неделю
- 2 Гб памяти на каждом GPU
- 5 Гб системной памяти

## Архитектура Inception [Szegedy et al., 2014]

- Два способа увеличения качества сети — увеличение глубины и увеличение ширины

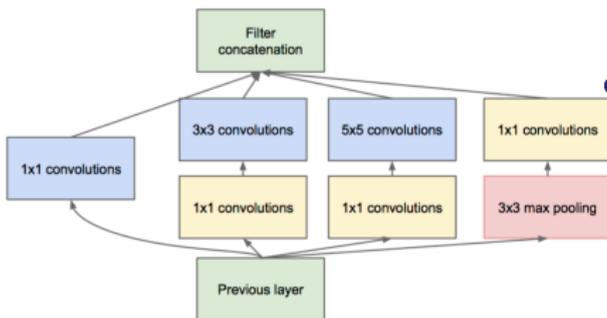
## Архитектура Inception [Szegedy et al., 2014]

- Два способа увеличения качества сети — увеличение глубины и увеличение ширины
- Опасность №1: повышение вероятности переобучения
- Опасность №2: неэффективное использование вычислительных ресурсов

## Архитектура Inception [Szegedy et al., 2014]

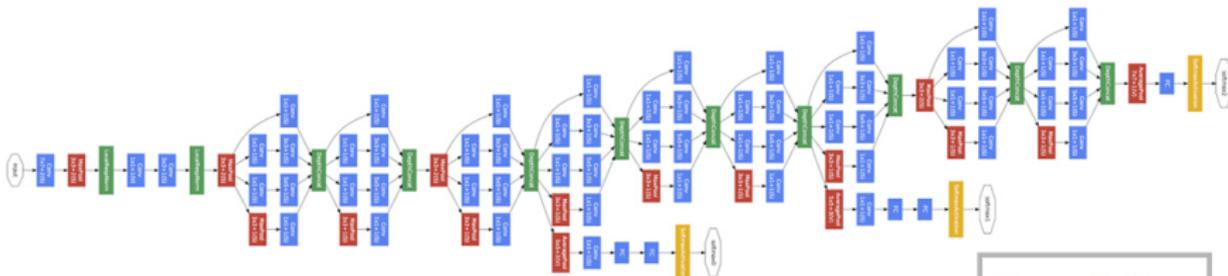
- Два способа увеличения качества сети — увеличение глубины и увеличение ширины
- Опасность №1: повышение вероятности переобучения
- Опасность №2: неэффективное использование вычислительных ресурсов
- Разреженная структура сверток неэффективна в вычислительном смысле
- Идея: использование корреляционной структуры активаций предыдущих слоев

# Архитектура Inception [Szegedy et al., 2014]



- Свертки меньшего размера сжимают пространство перед свертками большего размера

(b) Inception module with dimensionality reduction



**Convolution**  
**Pooling**  
**Softmax**  
**Other**

# Архитектура ResNet [He et al., 2015]

[Оригинальные слайды]