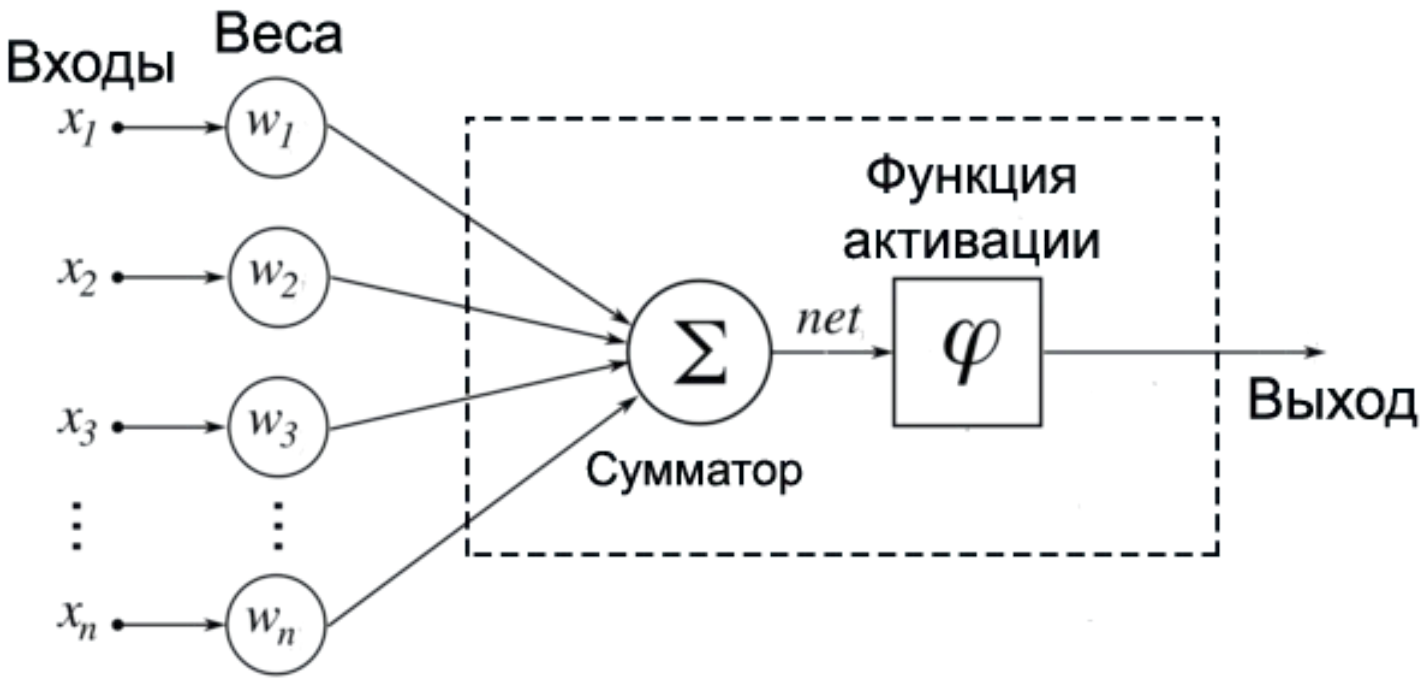
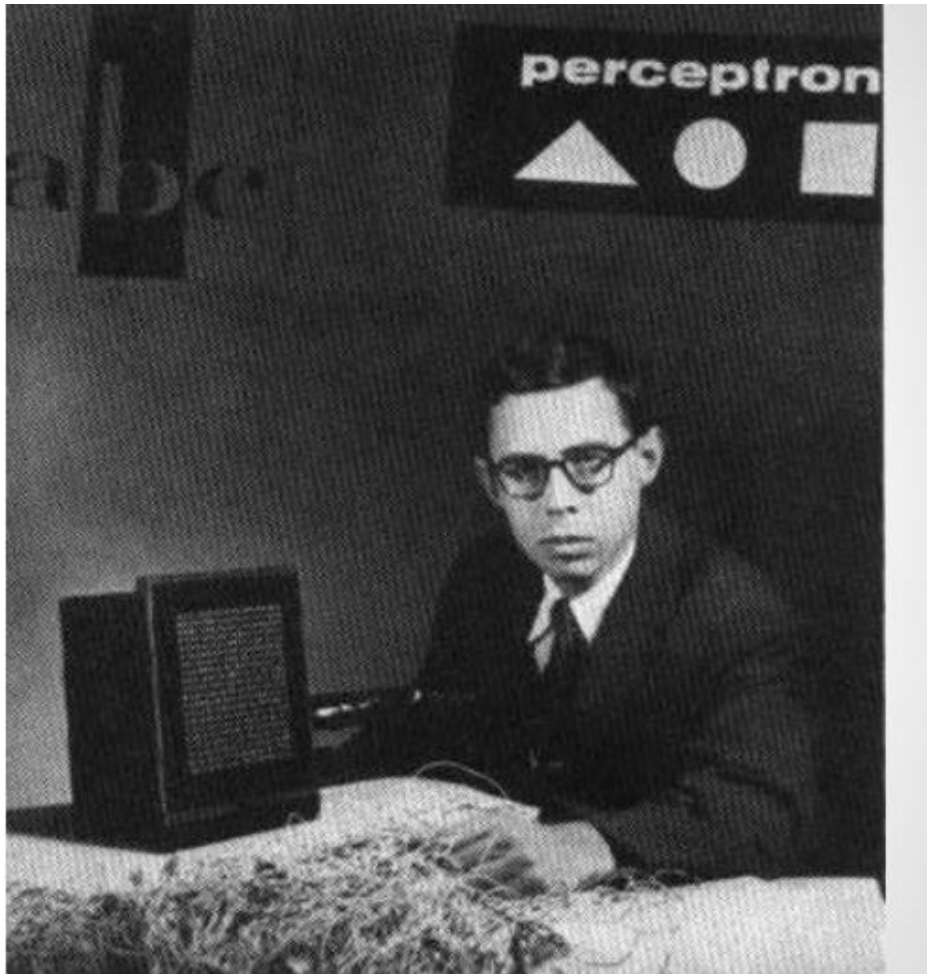
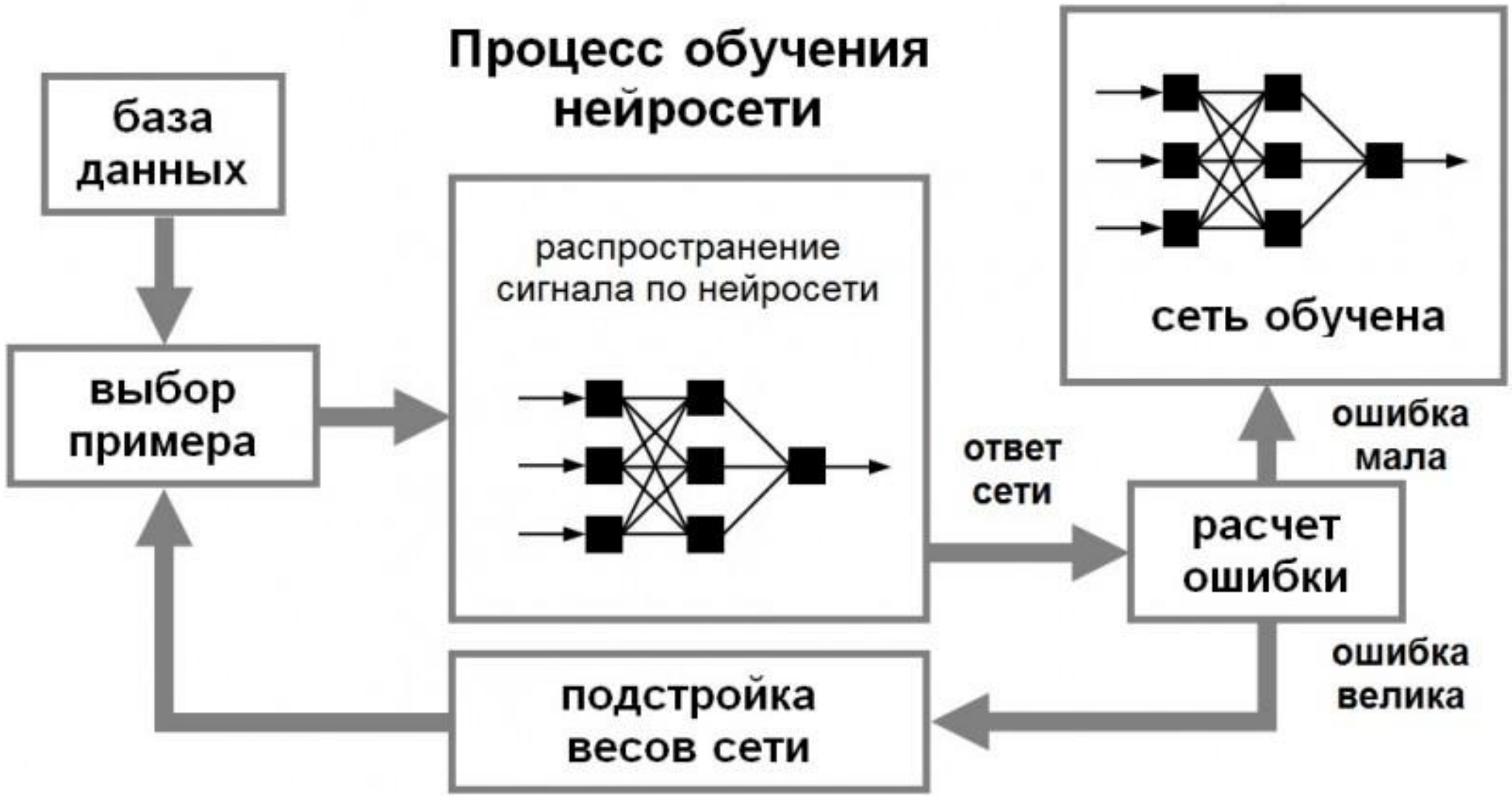
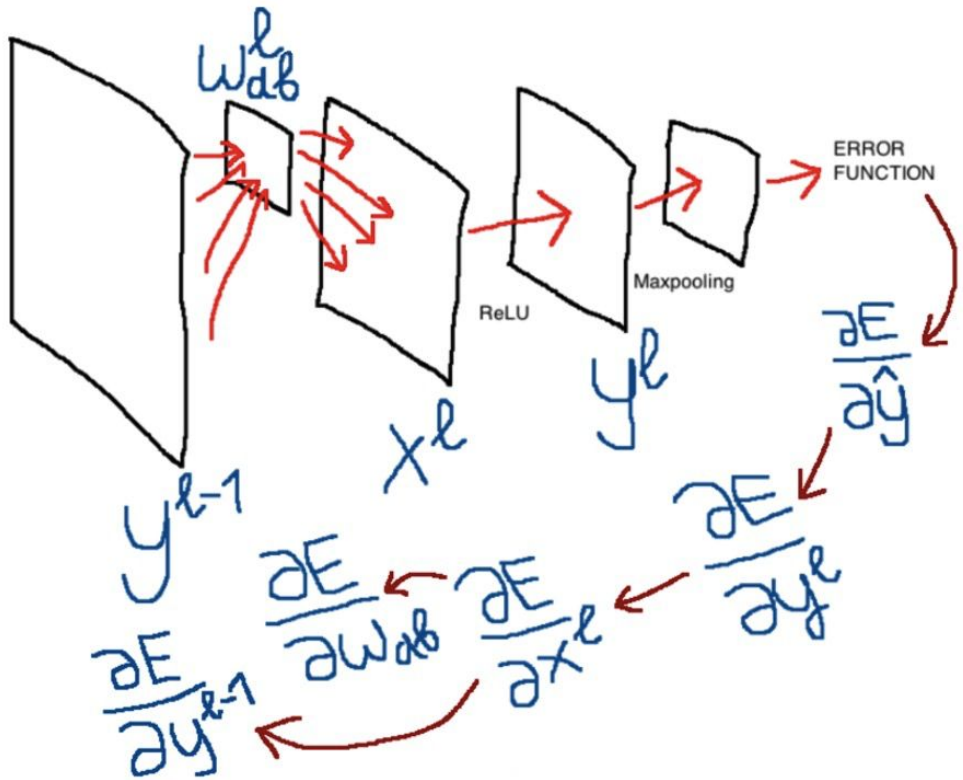

Входы
Веса
x_1
x_2
x_3
x_n
w_1
w_2
w_3
w_n
Функция активации
Σ
net
φ
Сумматор
Выход


perceptron


Процесс обучения нейросети
база данных
выбор примера
распространение сигнала по нейросети
ответ сети
расчет ошибки
ошибка мала
сеть обучена
ошибка велика
подстройка весов сети

Сверточные нейронные сети



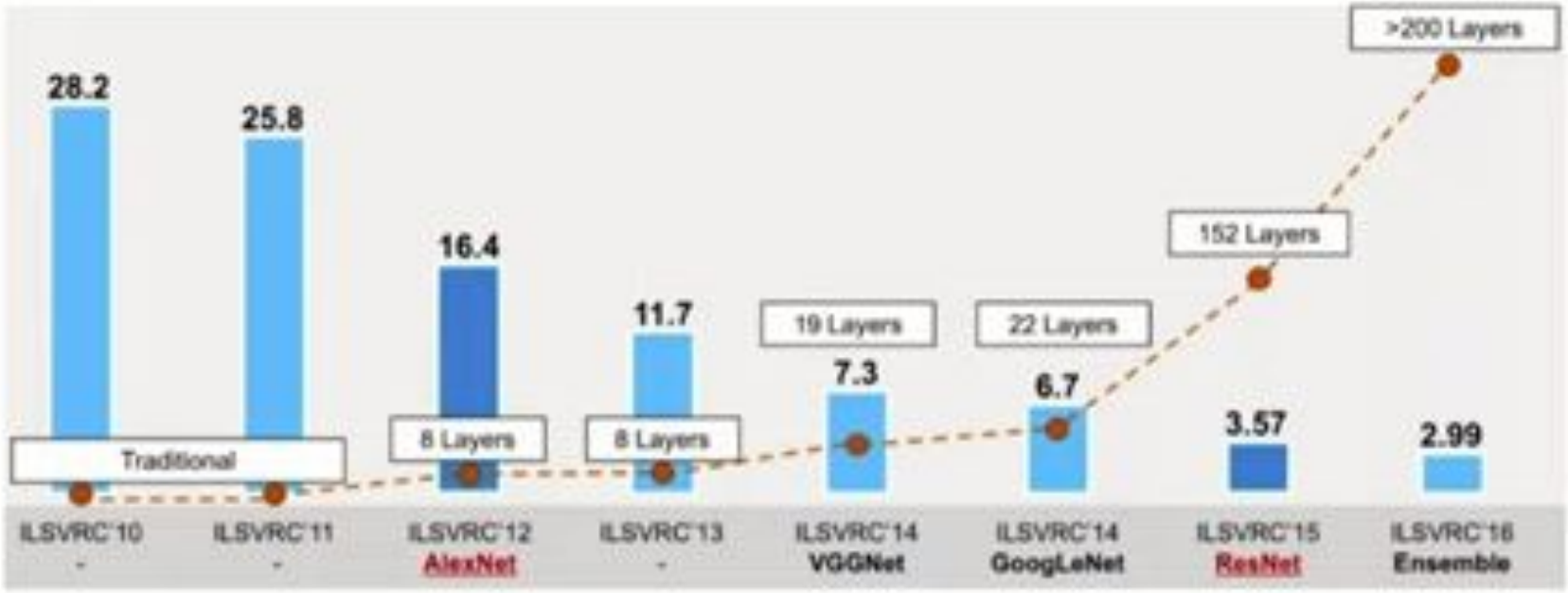
Поиск характерных элементов

Наложение шаблона

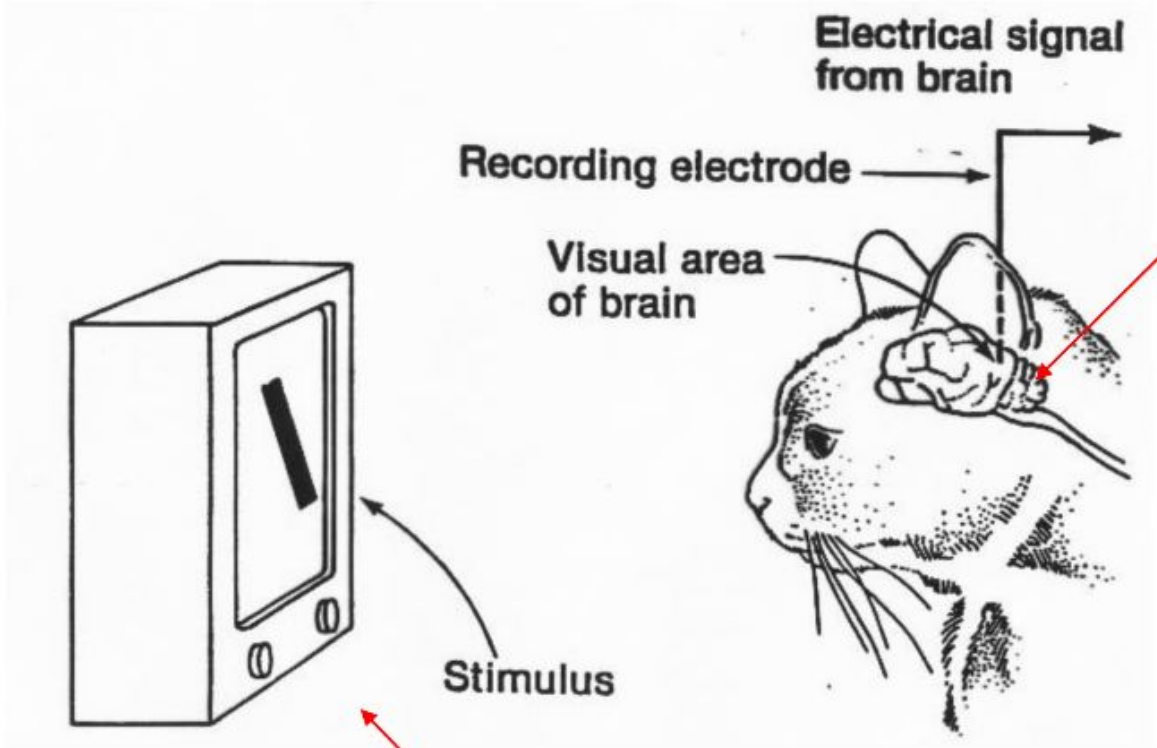
Ян Лекун

Свёрточная нейронная сеть (англ. *convolutional neural network*, *CNN*) — специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году

История развития CNN



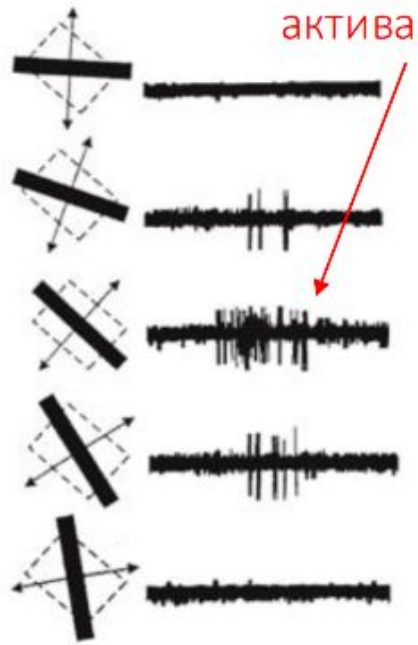
- 1998 год — Ян ЛеКун создал первую CNN
- 2012 год — свёрточная нейронная сеть AlexNet победила в конкурсе ImageNet
- 2015 год — свёрточная нейронная сеть ResNet обогнала обученного человека



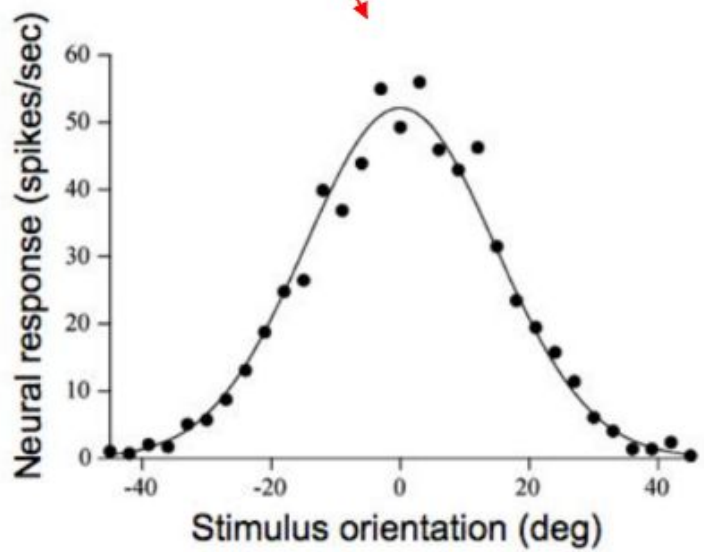
V1

стимул

активация



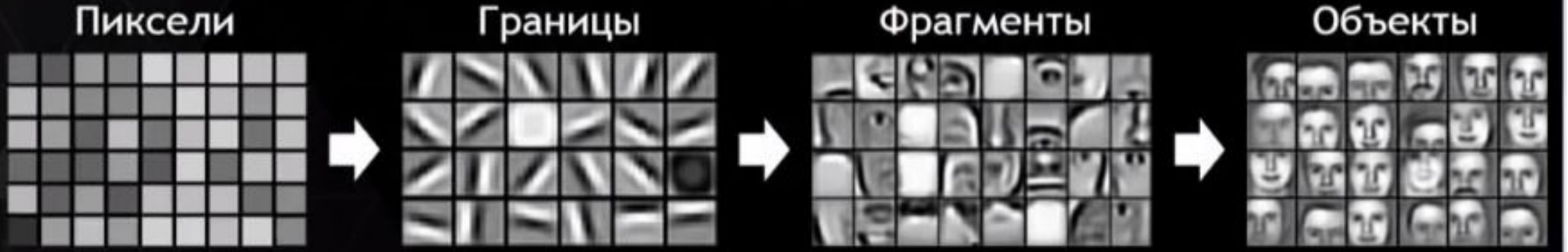
отклик клетки при изменении ориентации стимула



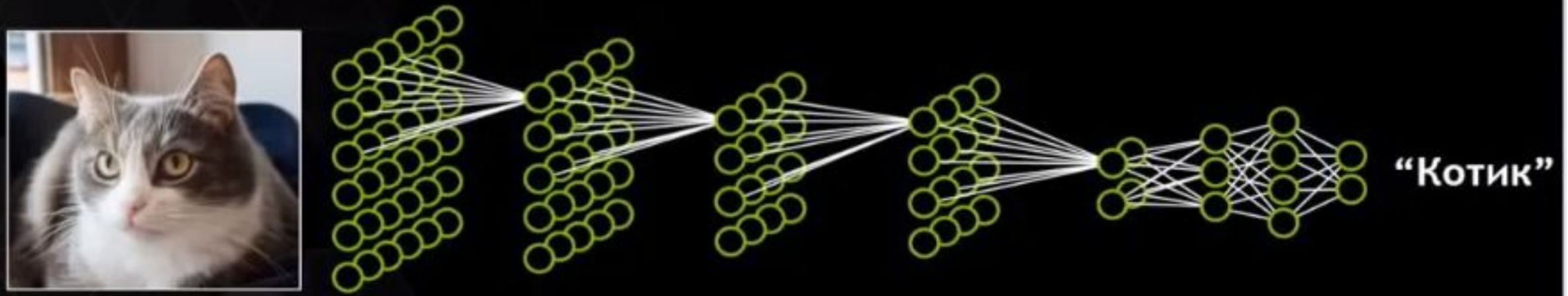
Hubel and Wiesel (1968)

- История вопроса – методы до ИНС
- Пример про цифры
- Эксперимент с кошкой
- Идея свертки
- В каком виде подают значения (тензоры и поделить на разные каналы монохромно)
- Приемы свертки
  - Страйт, паддинг, пулинг
- Примеры свертки
- Пример классификации
- Предобученные нейронные сети
- Выделение абстрактных признаков и перенос знаний

ИЕРАРХИЯ ПРИЗНАКОВ



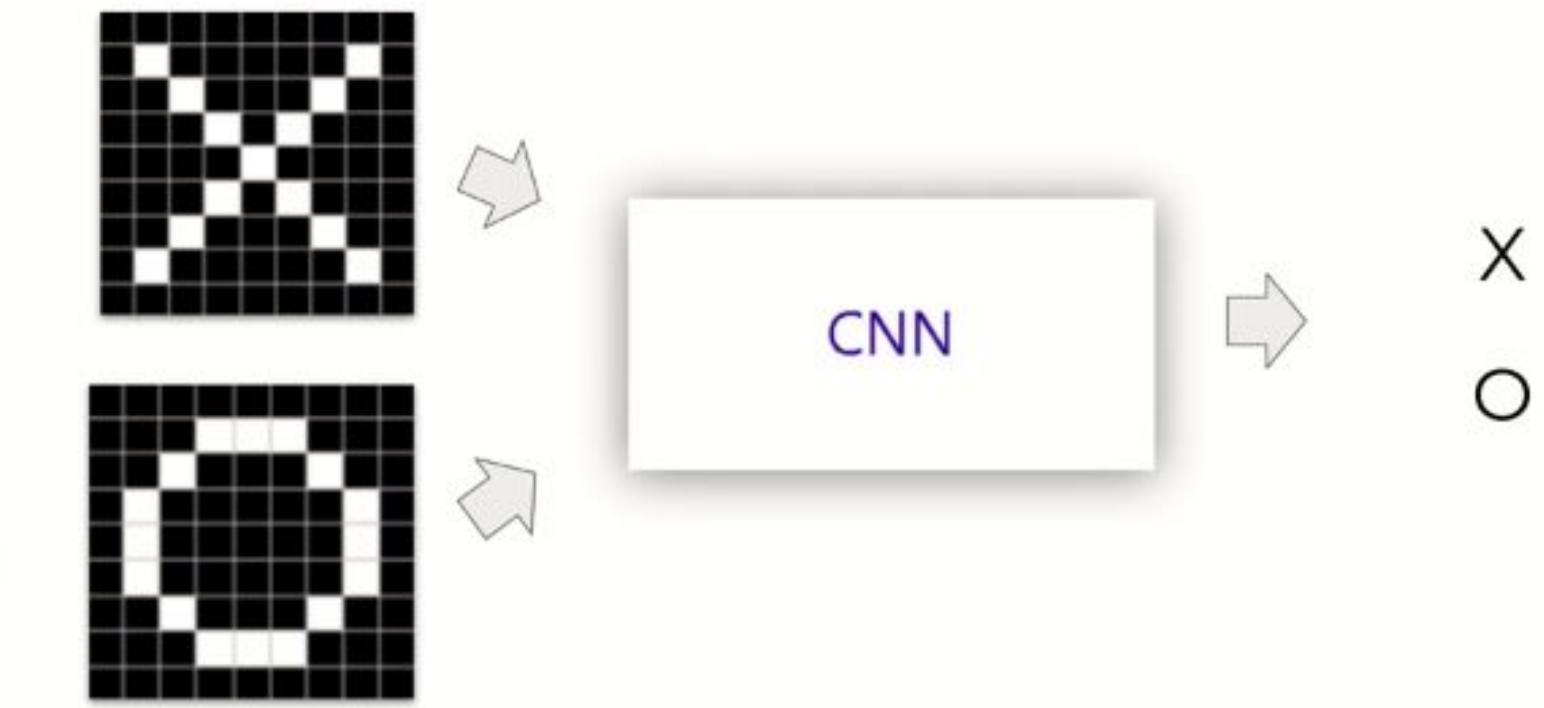
ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ



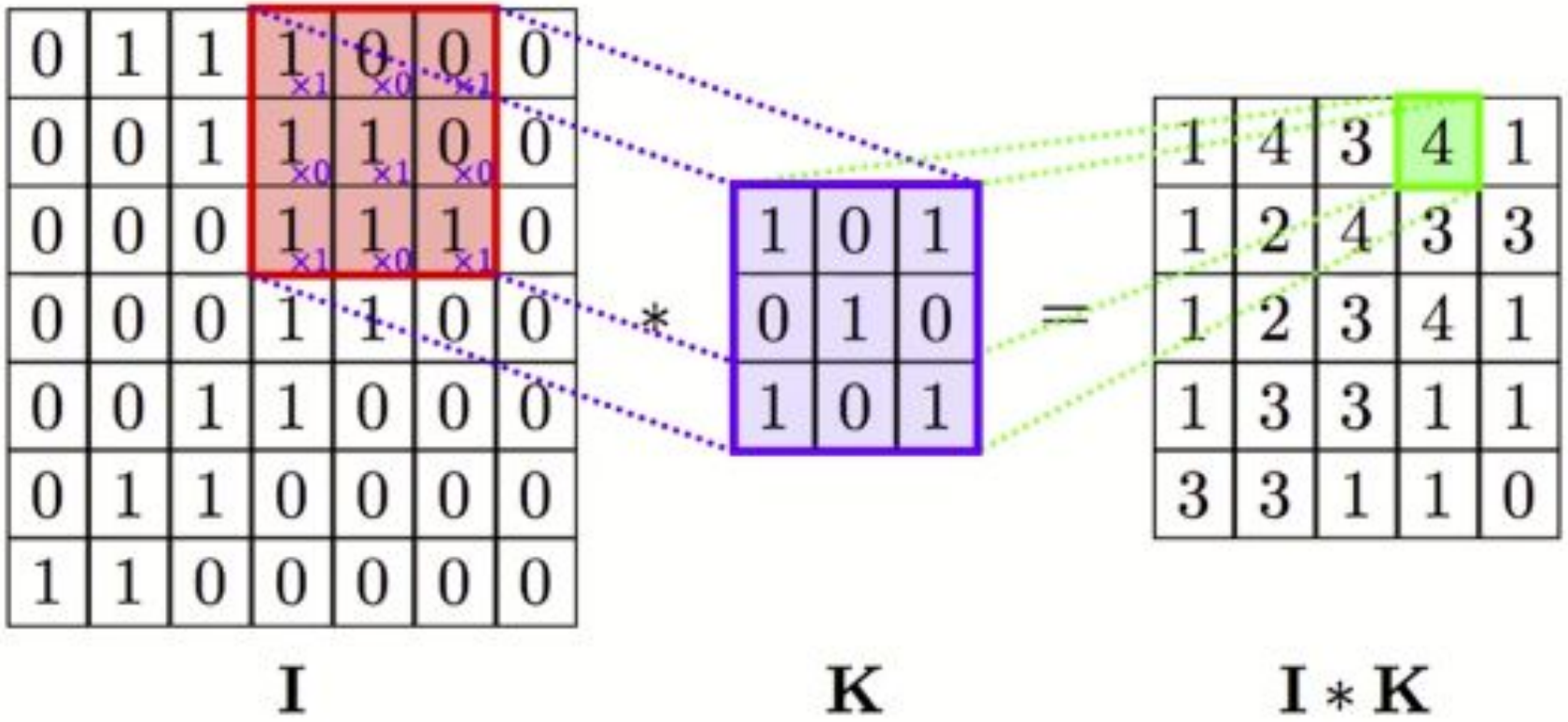
Типы слоев

- ✔ Сверточный слой
- ✔ Слой активации
- ✔ Слой пуллинга
- ✔ Полносвязный слой

Простая задача классификации

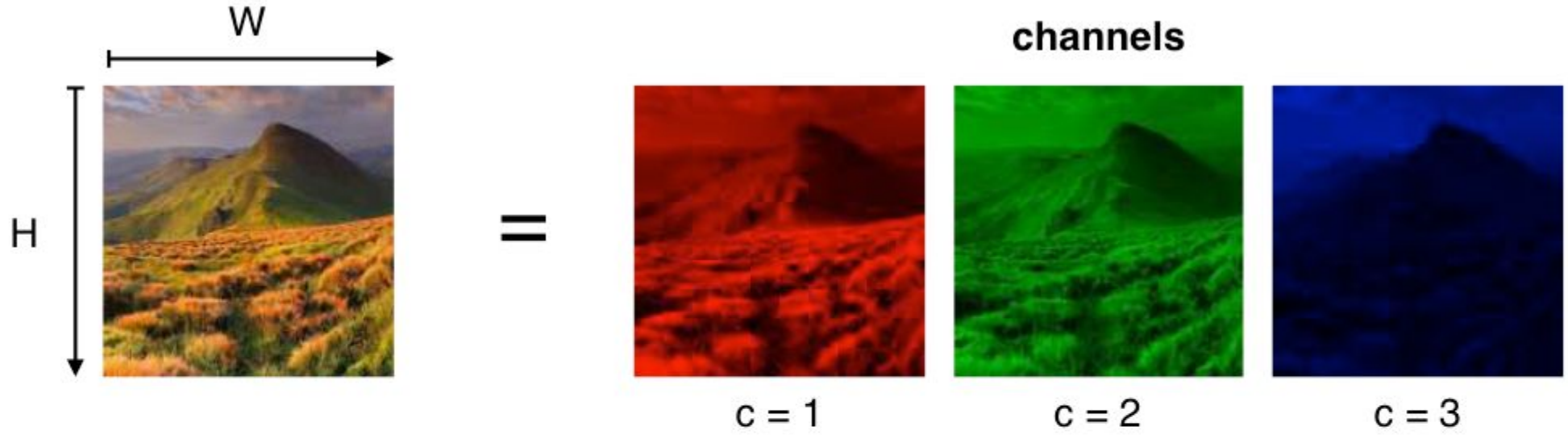


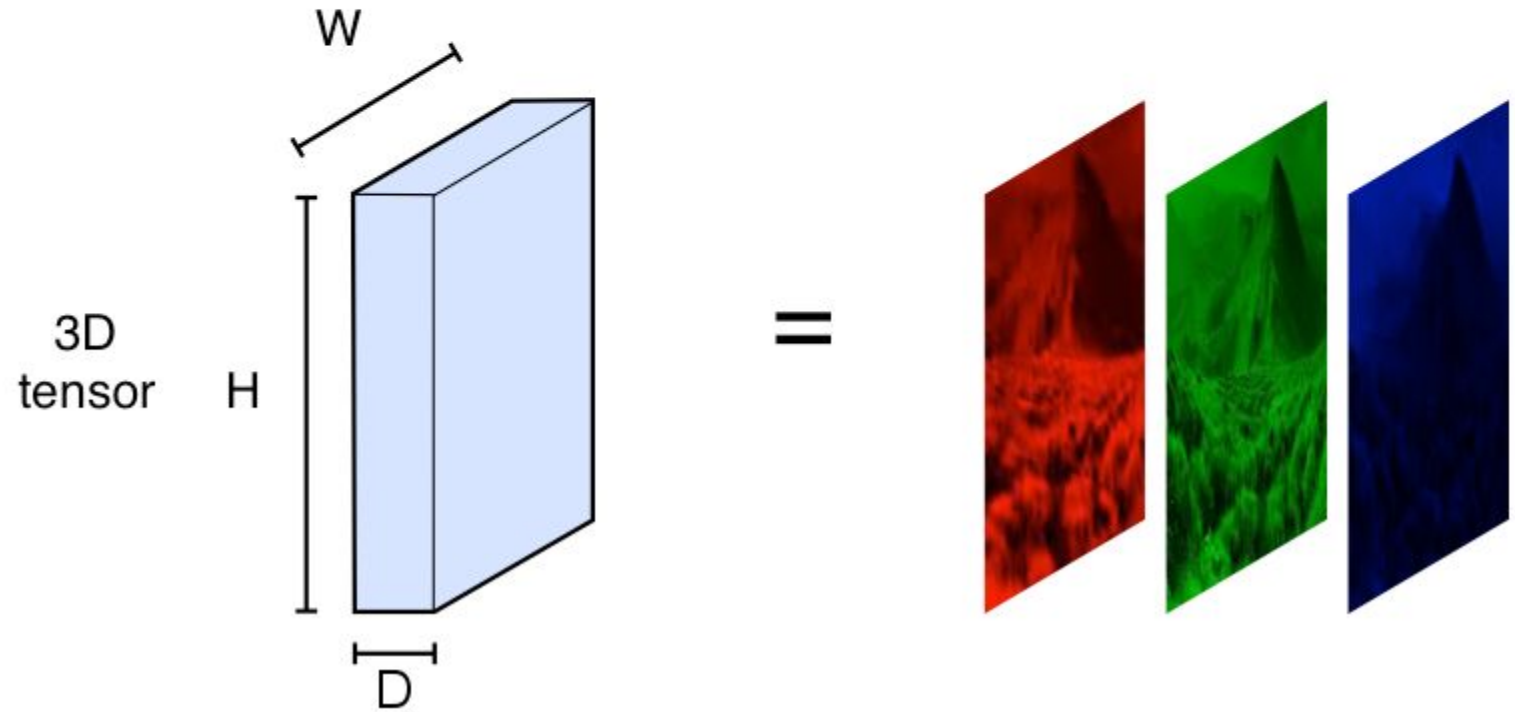
Операция свертки



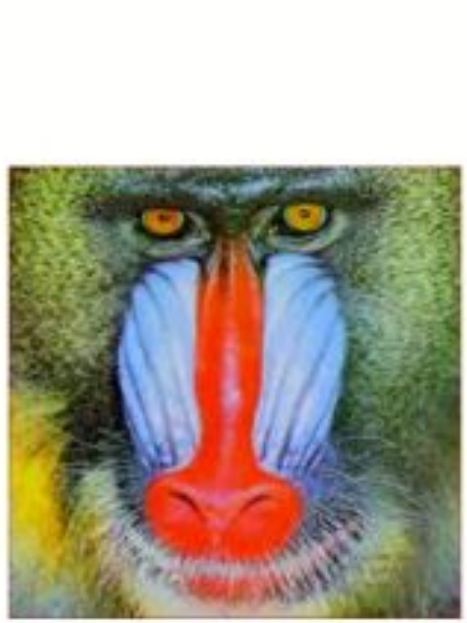
Фильтр — матрица 3x3

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	203	229	32	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	47	47	30	95	254	215	13	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	45	154	185	185	223	253	253	133	175	255	188	19	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	110	253	253	253	246	161	228	253	253	254	92	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	128	245	253	158	137	21	0	48	233	253	233	8	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	139	254	223	25	0	0	36	170	254	244	106	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	55	212	253	161	11	26	178	253	236	113	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	7	155	253	228	80	223	253	253	109	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	141	253	253	253	254	253	154	29	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	110	253	253	253	254	179	38	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3	171	254	254	254	179	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	171	253	253	253	253	178	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	26	123	254	253	203	156	253	200	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	93	253	254	121	13	93	253	158	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	64	239	253	76	8	32	219	253	126	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	133	254	191	0	5	108	234	254	106	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	132	253	190	5	85	253	236	154	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	153	253	169	192	253	253	77	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	112	253	253	254	236	129	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	17	118	243	191	113	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0


W
H
=
channels
c = 1
c = 2
c = 3


W
3D
tensor
H
D
=

Карты признаков



(11)

SCALAR

5	3	7

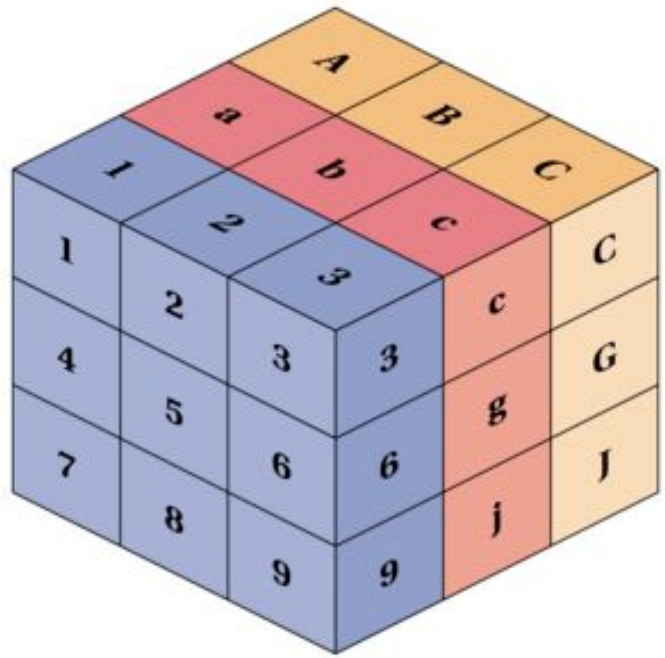
Row Vector
(shape 1x3)

5
1.5
2

Column Vector
(shape 3x1)

$$\begin{bmatrix} 4 & 19 & 8 \\ 16 & 3 & 5 \end{bmatrix}$$

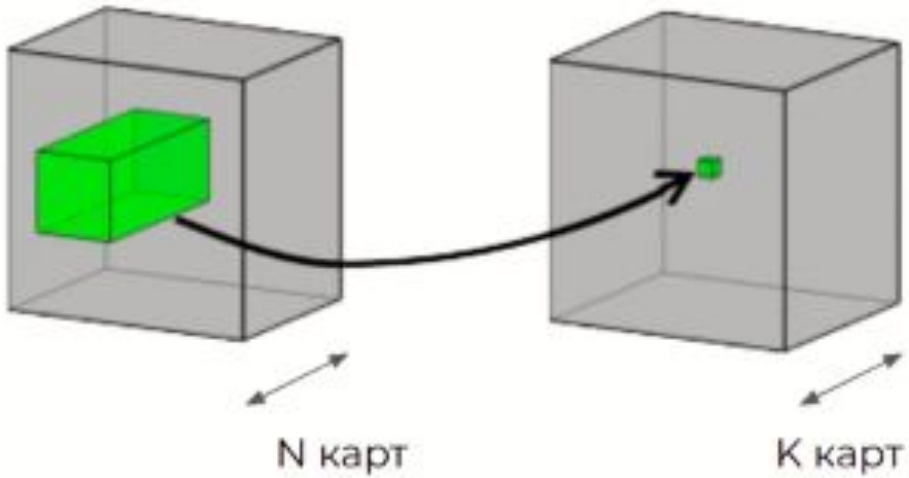
MATRIX



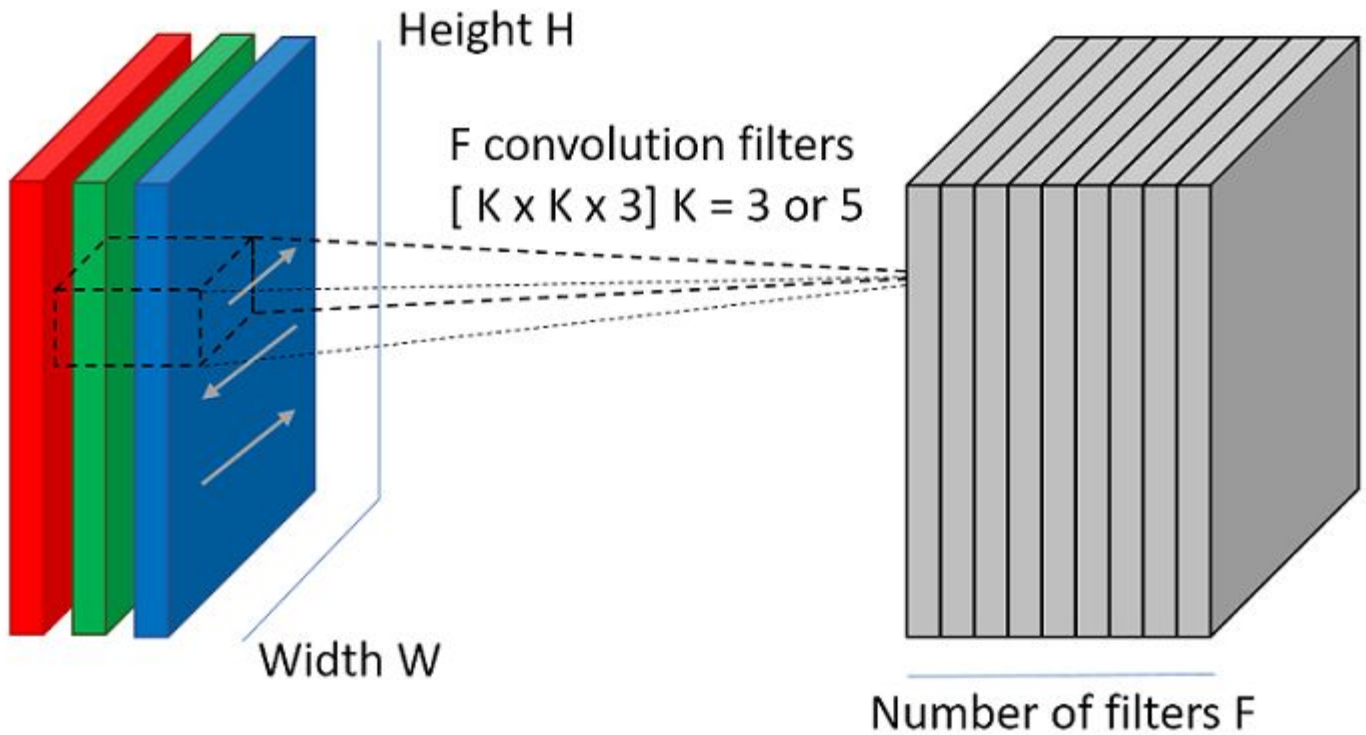
TENSOR

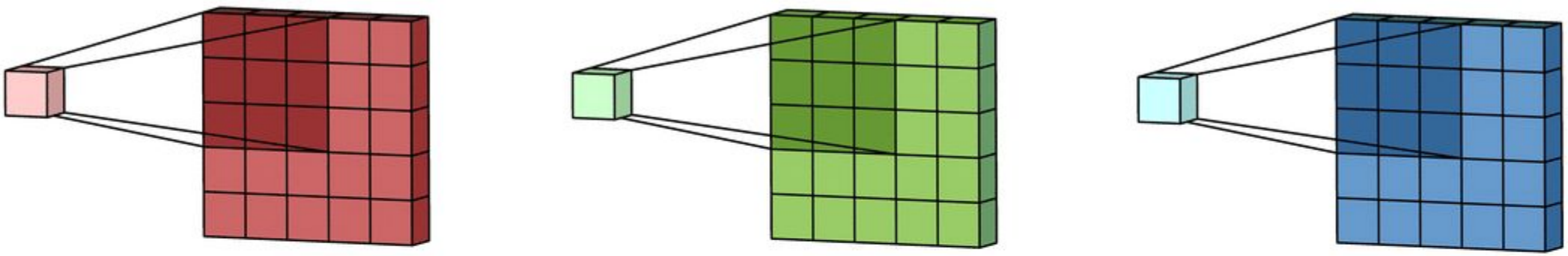

Scalar
Vector
Matrix
Tensor
1
1
2
1 2
3 4
1 2
3 2
1 7
5 4

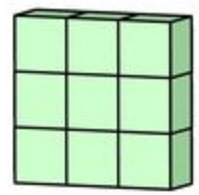
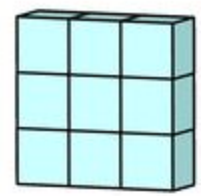
Параметры сверточного слоя

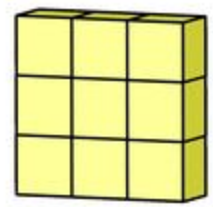


- Фильтр — матрица 3x3xN
- Всего K фильтров
- Параметры свёрточного слоя нейронной сети — тензор размерности 3x3xNxK

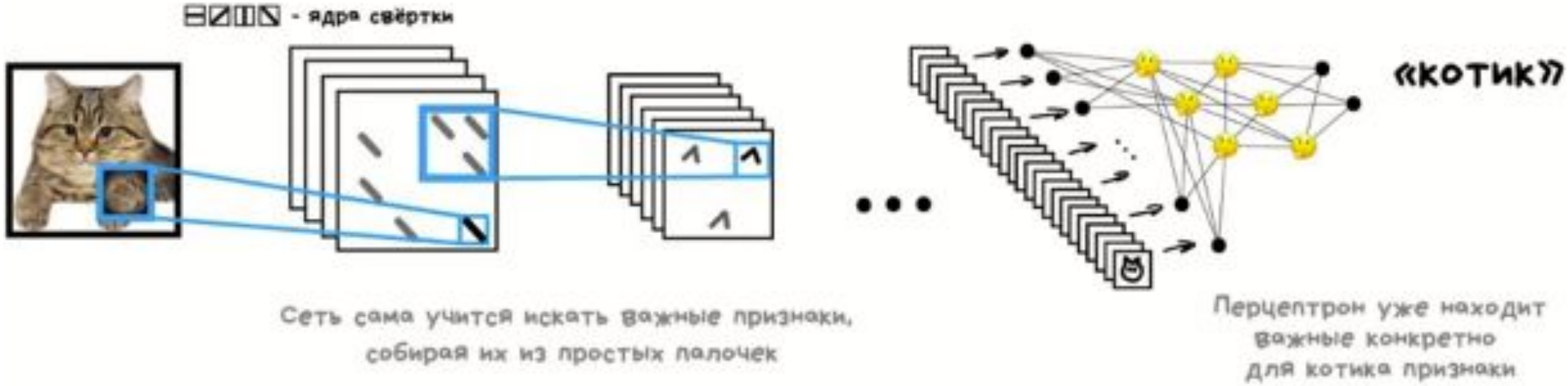

Height H
F convolution filters
[K x K x 3] K = 3 or 5
Width W
Number of filters F





Работа сверточной нейронной сети

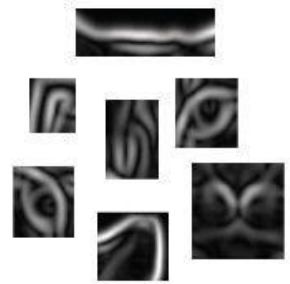


Свёрточная Нейросеть (CNN)

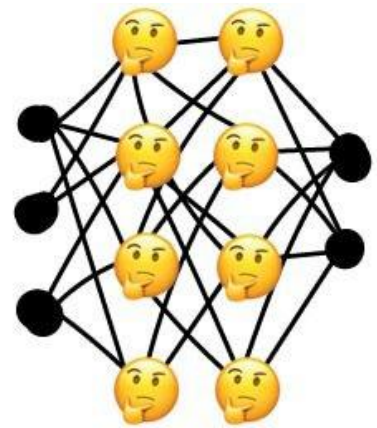
Каждый фильтр свёртки формирует собственную карту

Исходное изображение

Предварительная обработка



Отобранные вручную признаки

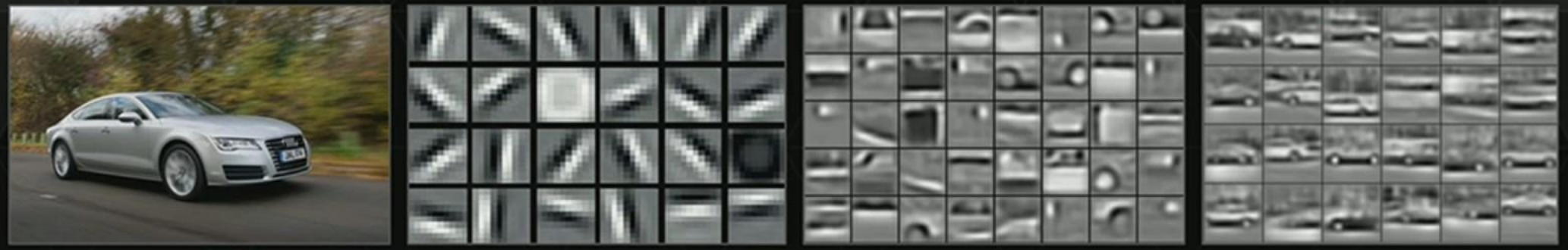


Нейросеть

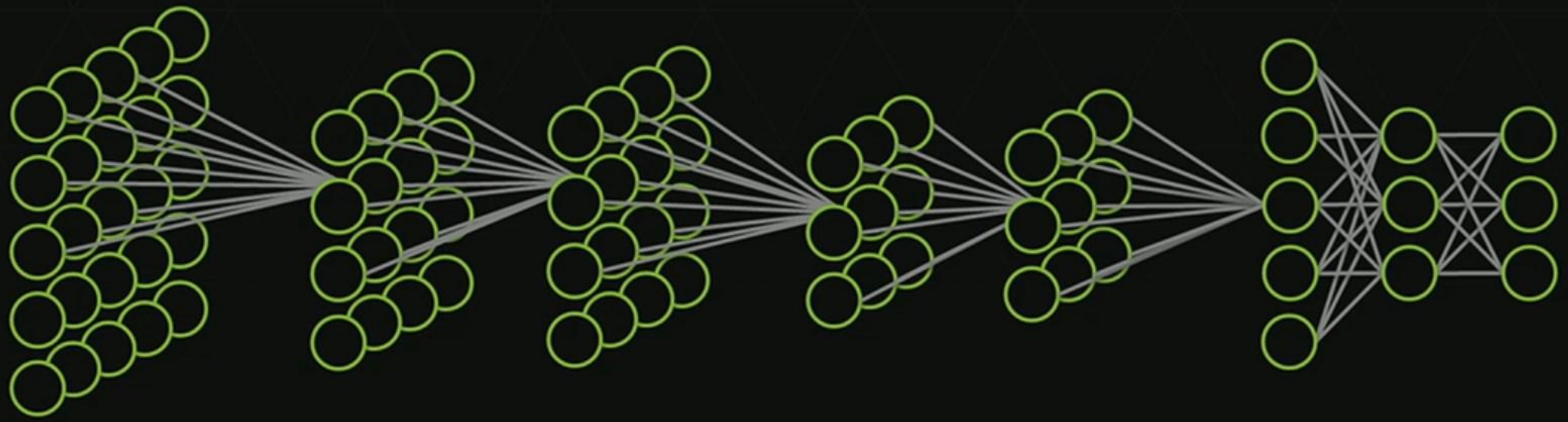
«котик»

Результат

HOW A DEEP NEURAL NETWORK SEES



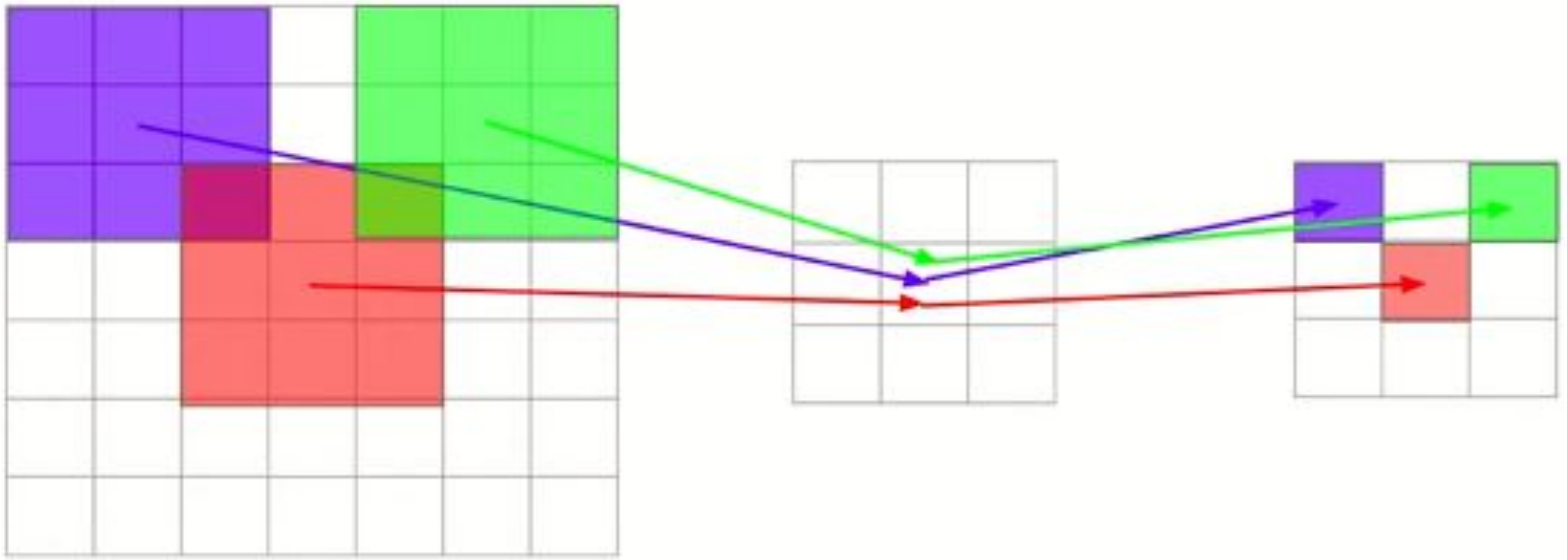
Image



"Audi A7"

Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.

Strided

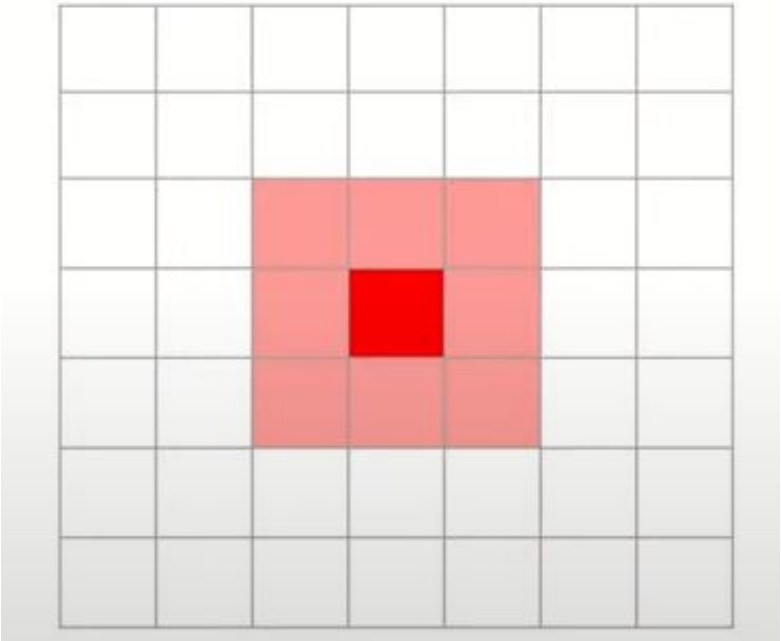


- Прогоняем фильтр не по всей карте, а через фиксированный шаг
- Резко уменьшаем размер изображения
- Повышаем receptive field

Receptive field

Дано: многослойная свёрточная нейронная сеть с фильтрами свёртки 3x3

Найти: сколько пикселей видит нейрон каждого слоя



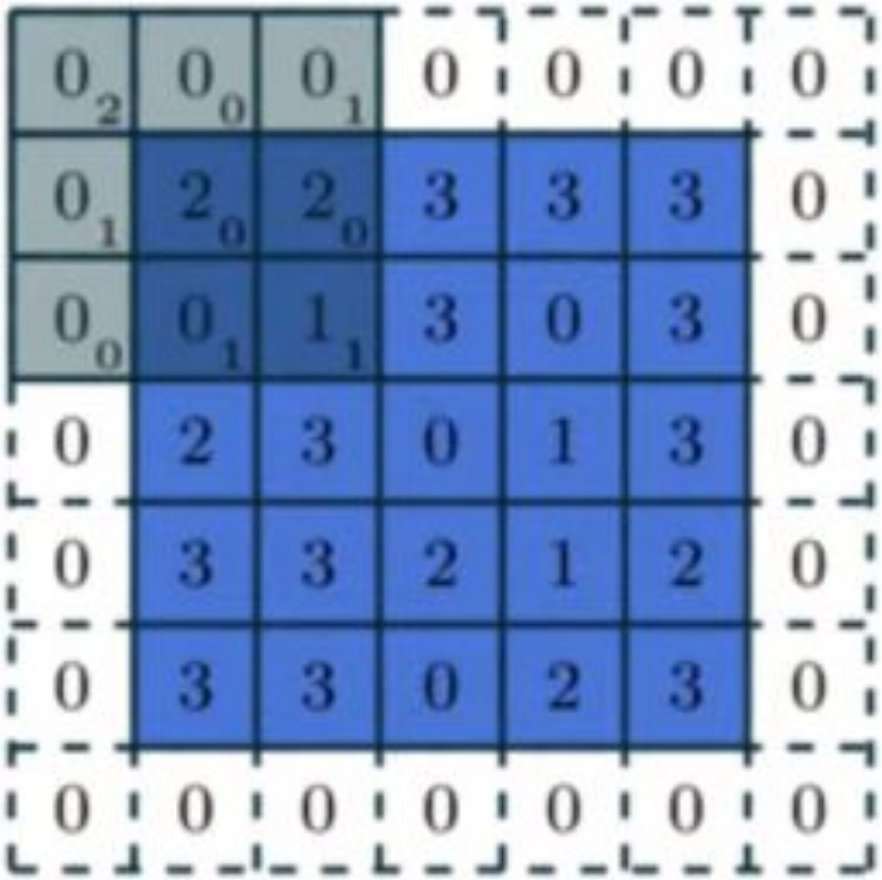
Padding

“Набивка” — искусственно увеличиваем размер изображения перед свёрткой.

- Same — сохраняет размер изображения
- Valid — не использует паддинг

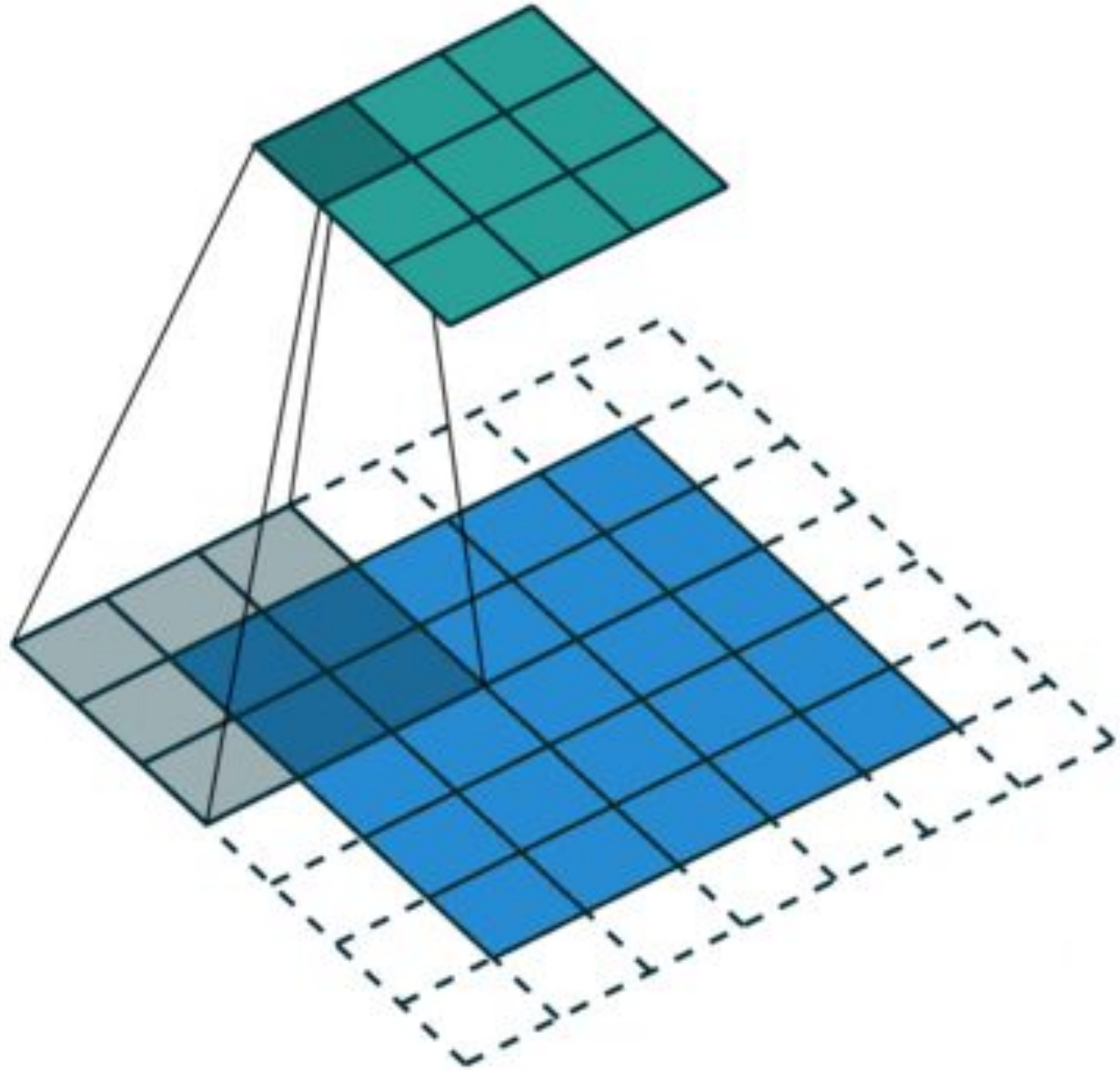
Можно выполнять паддинг нулями или как-то иначе

Padding



1	6	5
7	10	9
7	10	8

padding = 1, stride = 2



Параметры свёрточного слоя

Число признаков (filters count, fc) – это количество фильтров, которые есть в слое.

Размер фильтров (filter size, fs) – это высота и ширина тензора фильтров. Обычно является нечётным числом, наиболее часто используются фильтры размером 3 или 5.

Шаг свёртки (stride, S) – это количество пикселей, на которое перемещается матрица фильтра по входному изображению. Когда шаг равен 1, фильтры перемещаются по одному пикселю за раз. Когда шаг равен 2, тогда фильтры перескакивают на 2 пикселя за раз. Чем больше шаг, тем меньшего размера карты признаков получаются на выходе.

Дополнения нулями (padding, P) – количество пикселей, которые добавляются с каждого края изображения. Это позволяет избежать уменьшения изображения на размер фильтра, поскольку фильтр может накладываться лишь в тех местах, в которых под каждым значением фильтра будет значение входного изображения.

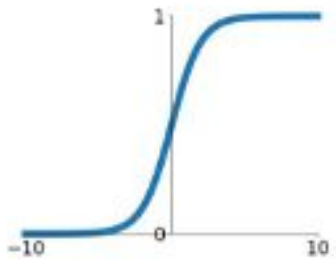
Таким образом, входными параметрами свёрточного слоя являются:

- тензор размером $W_1 x H_1 x D_1$;
- 4 гиперпараметра: fc, fs, S, P;
- А выходным параметром слоя является тензор размером $W_2 x H_2 x D_2$, где $W_2 = (W_1 - fs + 2P) / S + 1$, $H_2 = (H_1 – fs + 2P) / S + 1$, $D_2 = fc$.

Слой активации

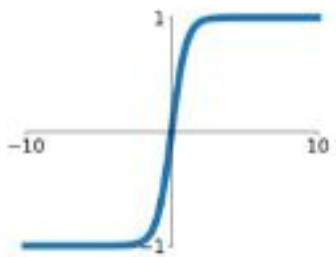
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



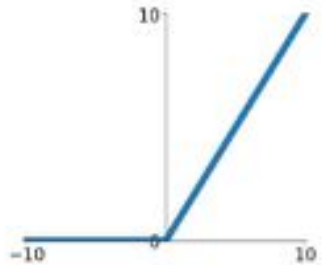
tanh

$$\tanh(x)$$



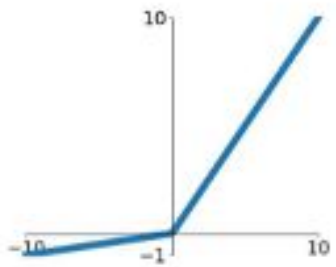
ReLU

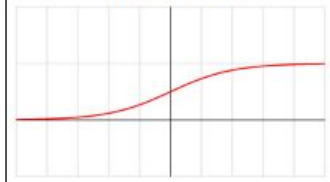
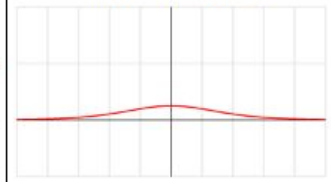
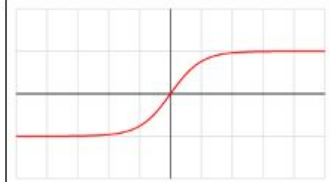
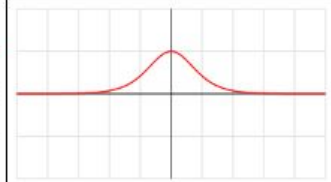
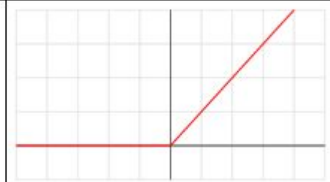
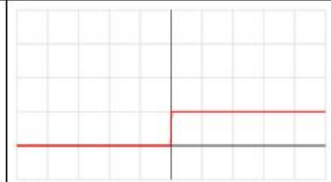
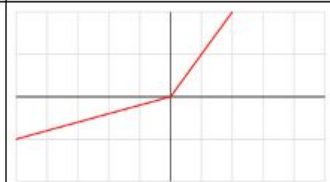
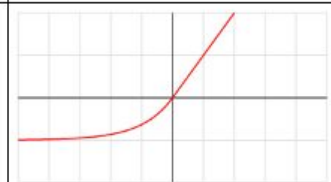
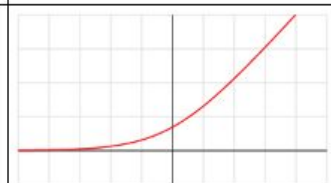
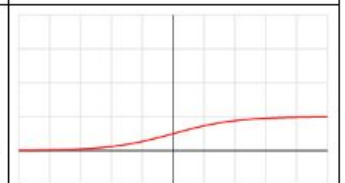
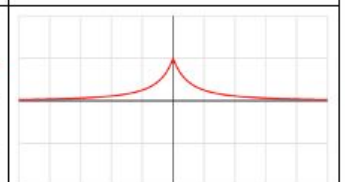
$$\max(0, x)$$



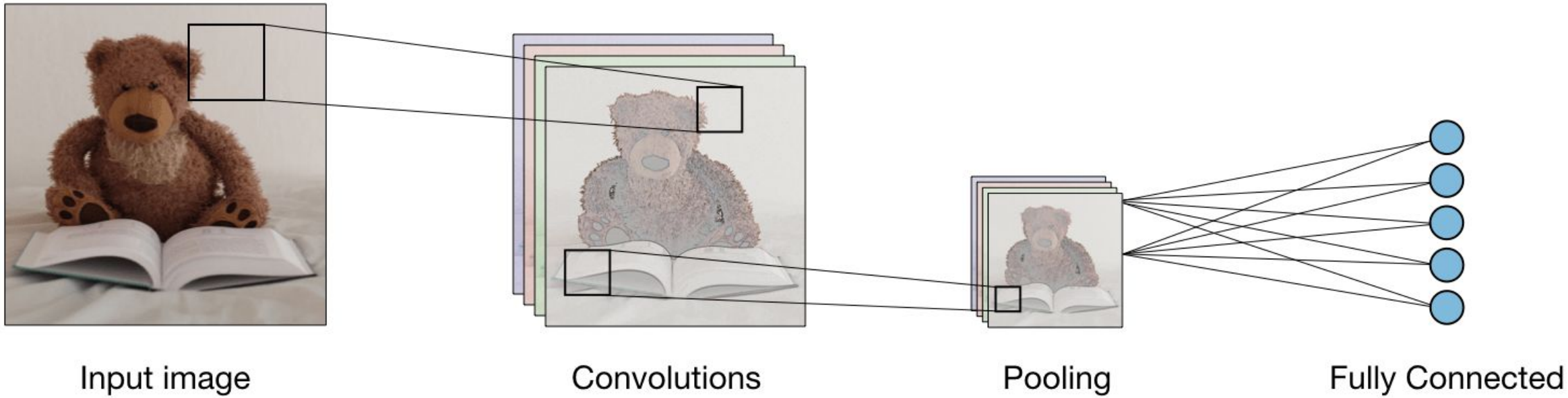
Leaky ReLU

$$\max(0.1x, x)$$

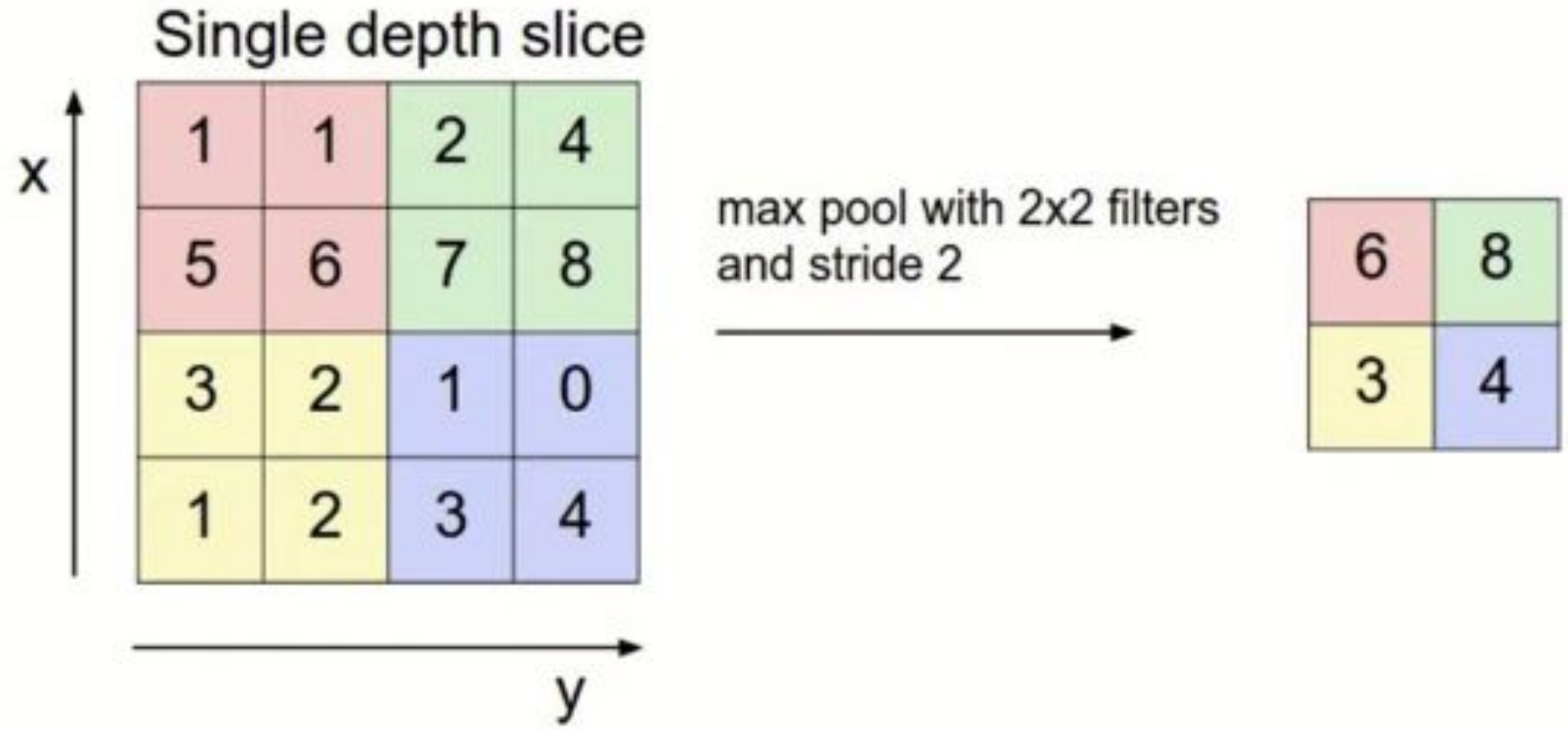


Название	Функция	Производная	Область значений	График функции	График производной
Sigmoid	$\frac{1}{1+e^{-x}}$	f(x)(1 - f(x))	(0, 1)		
Tanh	$\frac{e^{2x}-1}{e^{2x}+1}$	$1 - f^2(x)$	(-1, 1)		
ReLU	x, x > 0 0, x ≤ 0	1, x > 0 0, x ≤ 0	[0, +∞)		
Leaky ReLU	x, x > 0 αx, x ≤ 0	1, x > 0 α, x ≤ 0	(-∞, +∞)		
ELU	x, x > 0 $\alpha(e^x - 1)$, x ≤ 0	1, x > 0 αe^x, x ≤ 0	[-α, +∞)		
Softplus	$\ln(1 + e^x)$	$\frac{1}{1+e^{-x}}$	(0, +∞)		
Softsign	$\frac{x}{1+\lvert x\rvert}$	$\frac{1}{(1+\lvert x\rvert)^2}$	(-1, 1)		

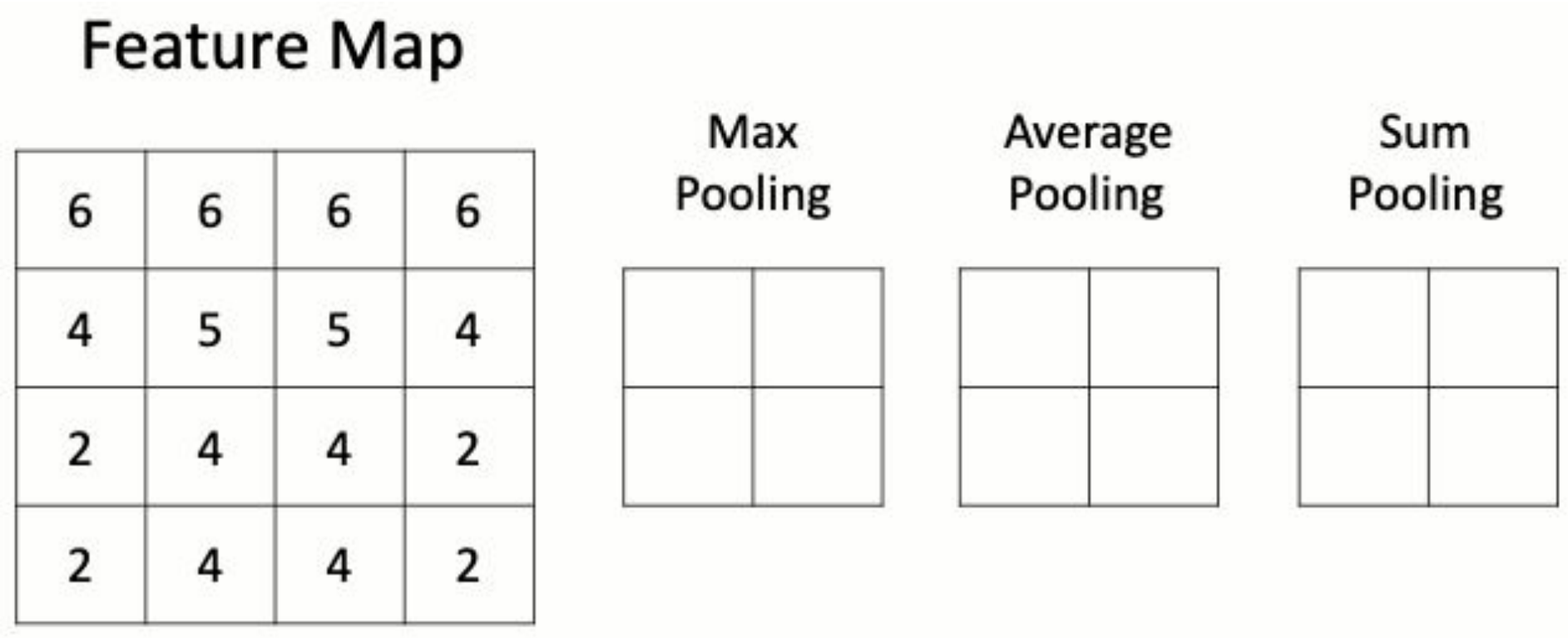
Пуллинг



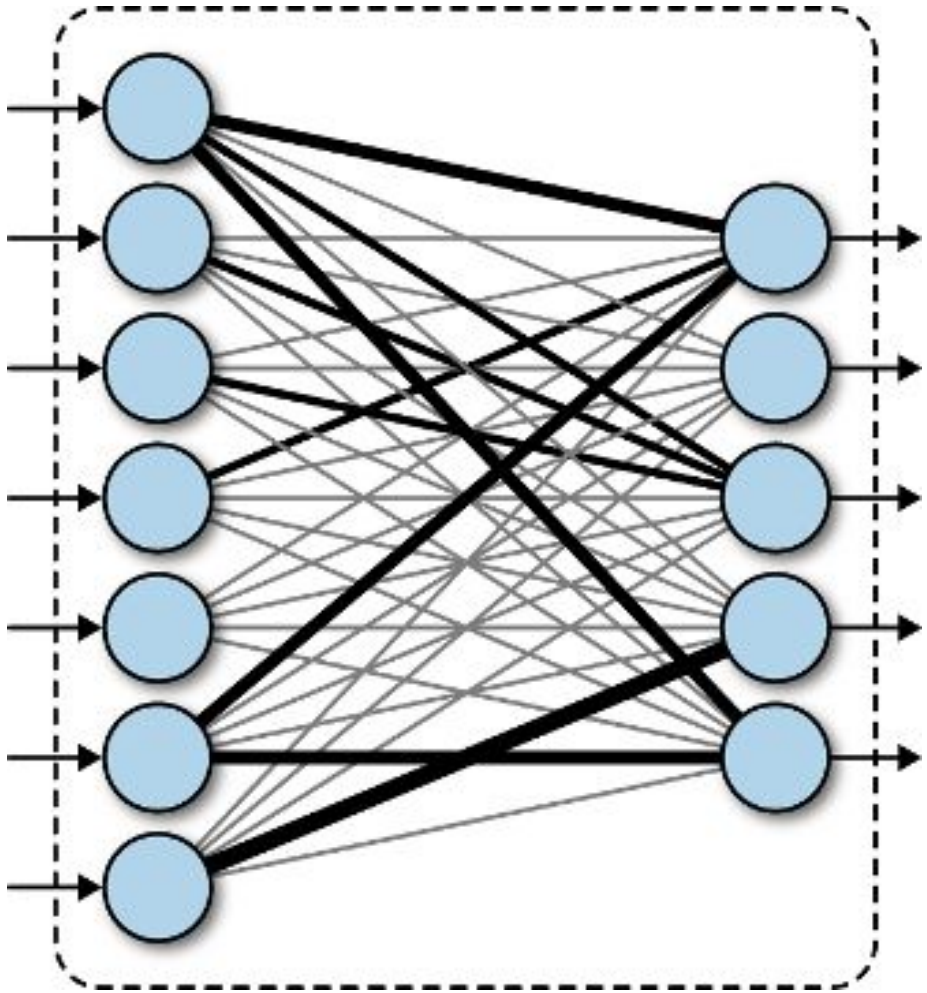
Пуллинг


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

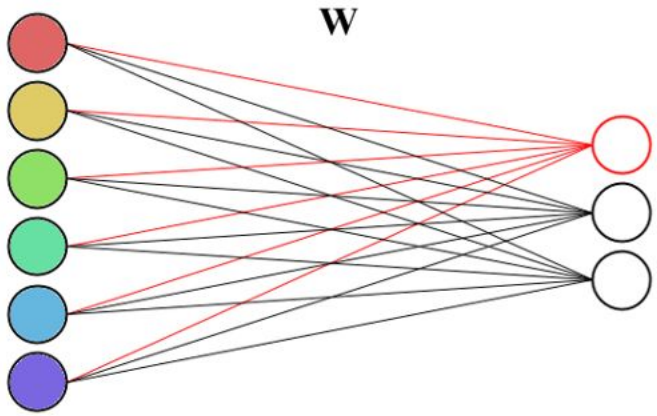
Пуллинг


Feature Map
6 6 6 6
4 5 5 4
2 4 4 2
2 4 4 2
Max Pooling
Average Pooling
Sum Pooling

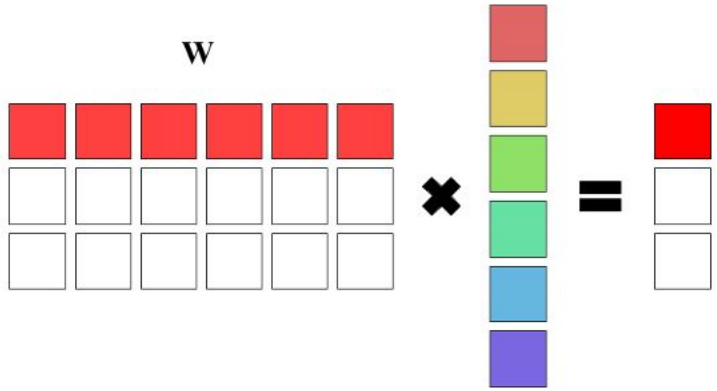
Полносвязный слой



Полносвязный слой

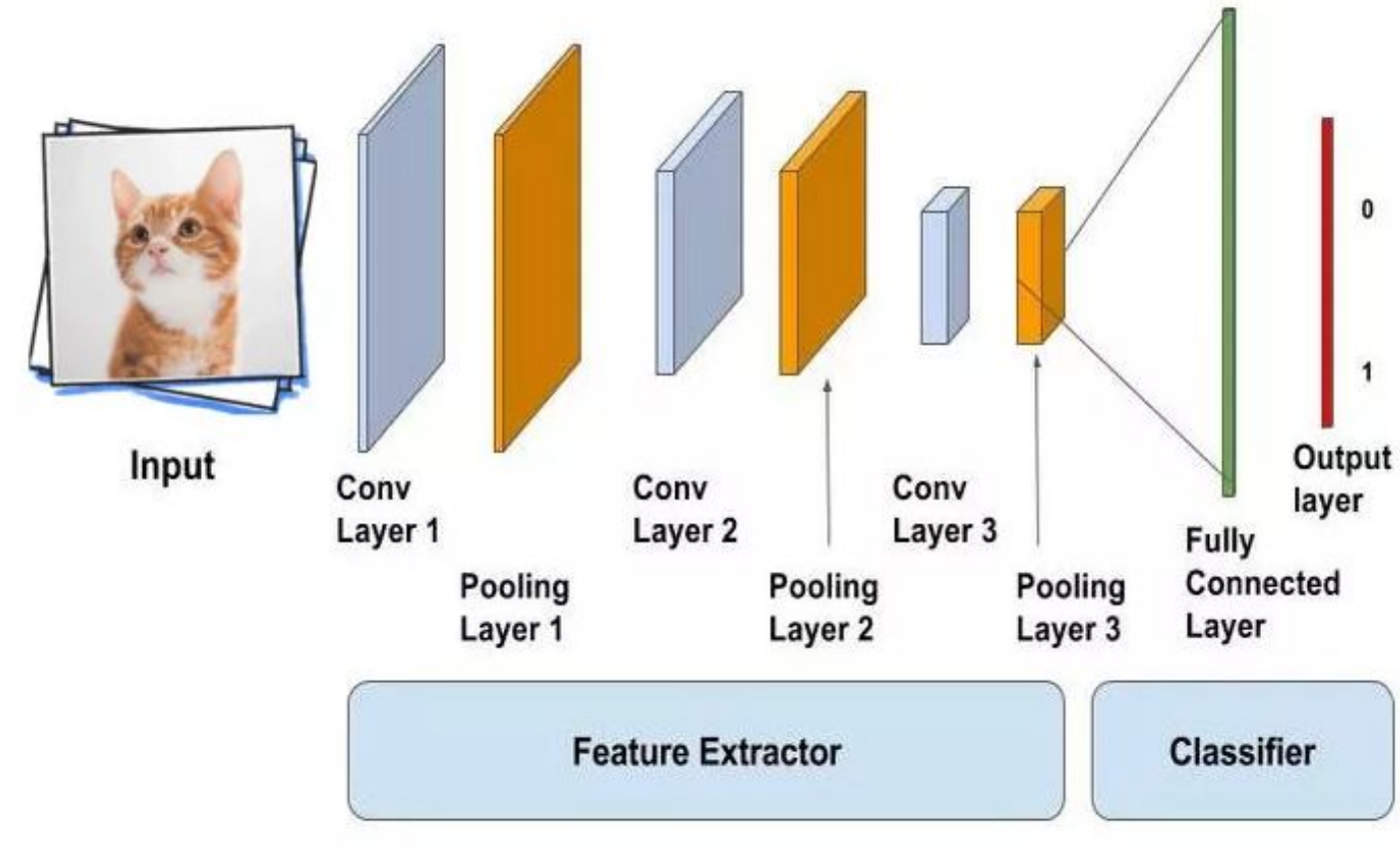


Полносвязный слой



Полносвязный слой в виде матрицы

Работа сверочной нейронной сети



Использование полносвязного слоя

```
model = keras.Sequential([
    Conv2D(32, (3,3), padding='same', activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D((2, 2), strides=2),
    Conv2D(64, (3,3), padding='same', activation='relu'),
    MaxPooling2D((2, 2), strides=2),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10,  activation='softmax')
])
```

Давайте выведем структуру этой сети и посмотрим на число весовых коэффициентов в каждом слое:

```
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 32)        320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 14, 14, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 3136)              0
_________________________________________________________________
dense (Dense)                (None, 128)               401536
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 421,642
Trainable params: 421,642
Non-trainable params: 0
```

```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import mnist          # библиотека базы выборок Mnist
from tensorflow import keras
from tensorflow.keras.layers import Dense, Flatten, Dropout, Conv2D, MaxPooling2D

(x_train, y_train), (x_test, y_test) = mnist.load_data()

# стандартизация входных данных
x_train = x_train / 255
x_test = x_test / 255

y_train_cat = keras.utils.to_categorical(y_train, 10)
y_test_cat = keras.utils.to_categorical(y_test, 10)
```

```
x_train = np.expand_dims(x_train, axis=3)
x_test = np.expand_dims(x_test, axis=3)
print( x_train.shape )
```

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

his = model.fit(x_train, y_train_cat, batch_size=32, epochs=5, validation_split=0.2)
```

```
model.evaluate(x_test, y_test_cat)
```