Введение в CNN

Кафедра технологий проектирования сложных технических систем

Сегодня в лекции:

- История возникновения сверточных нейросетей, конкурс ImageNet
- Свертка, сверточные сети
- Pooling
- Задачи компьютерного зрения

ImageNet

IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications Updates About

Not logged in. Login | Signup

База данных изображений, поделенных на 1000 классов.

http://www.image-net.org

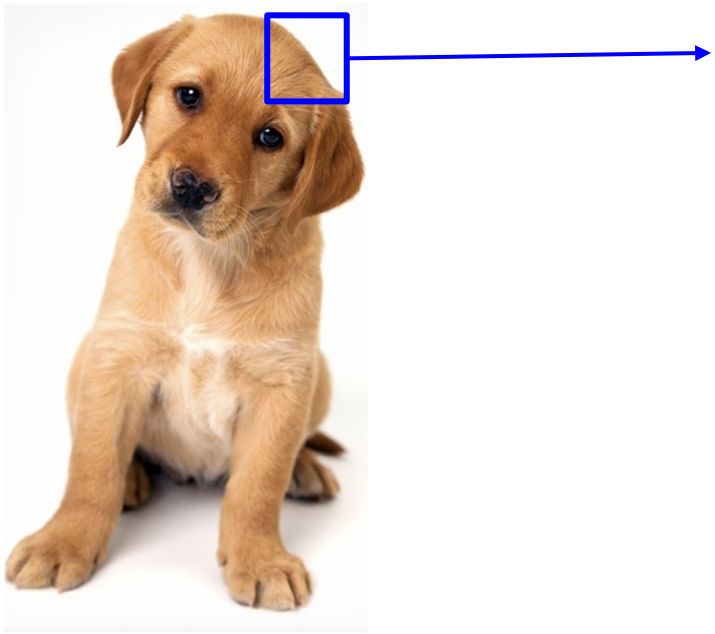
http://image-net.org/explore

ImageNet Timeline

До 2012 года в обработке картинок использовались НЕ нейронные подходы.

Подходы, основанные на нейросетях, существовали, но считались неспособными превзойти классические алгоритмы.

HOG (Histogram of Oriented Gradients)



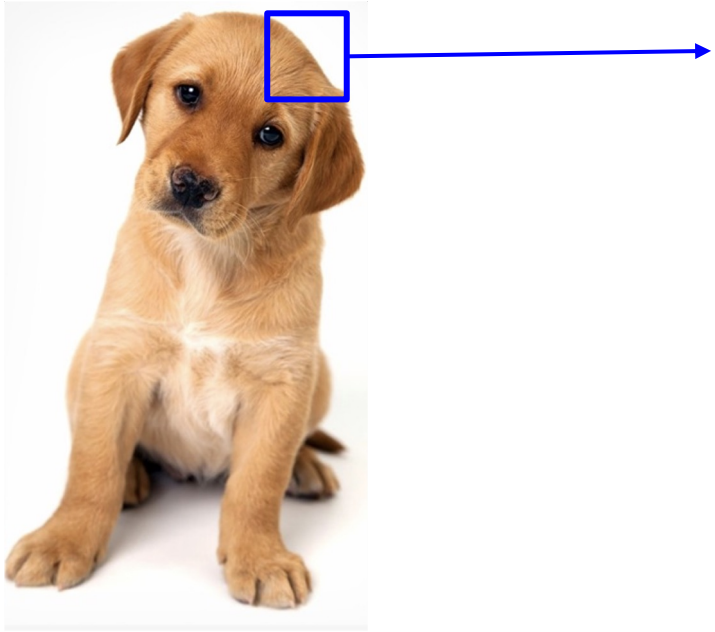
121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Градиенты для выделенного пикселя:

по OY: 68-56 = 8

по OX: 89-78 = 11

HOG (Histogram of Oriented Gradients)

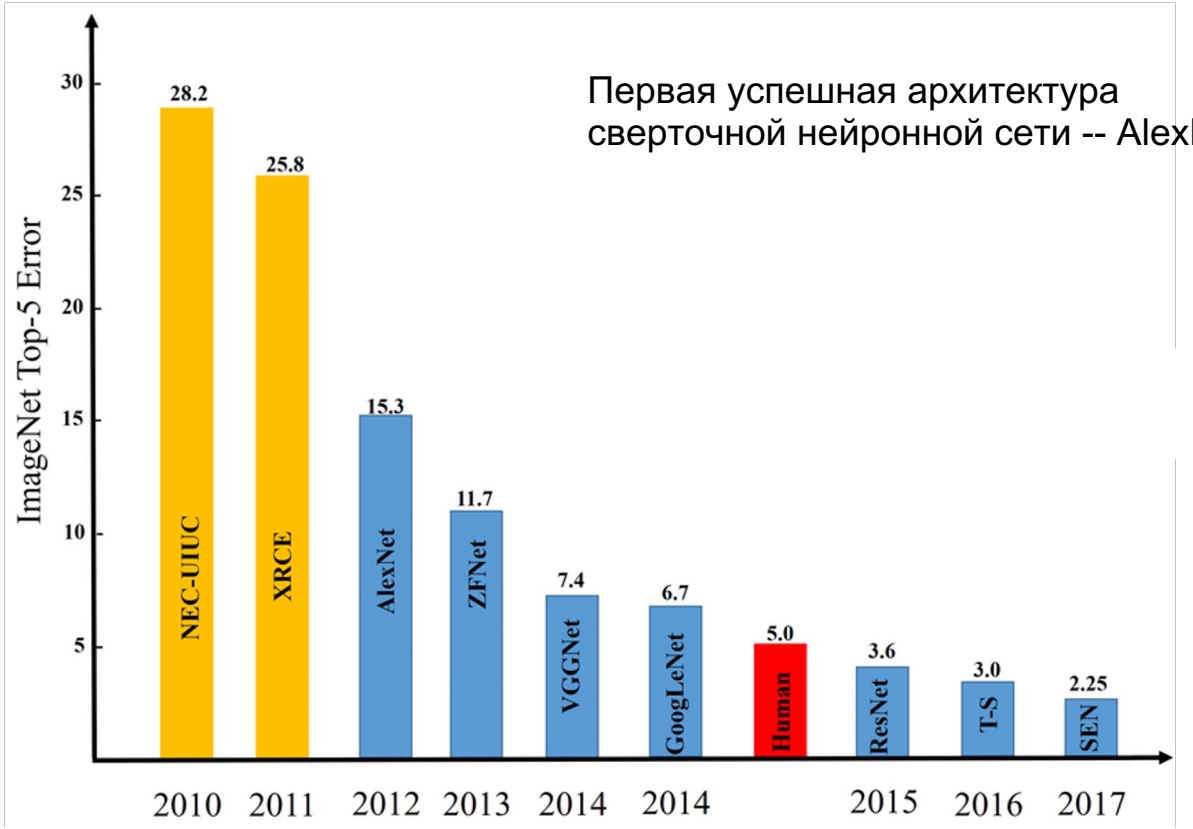


121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Градиенты будут отличаться для пикселей, находящихся на границе изображения и внутри объектов.

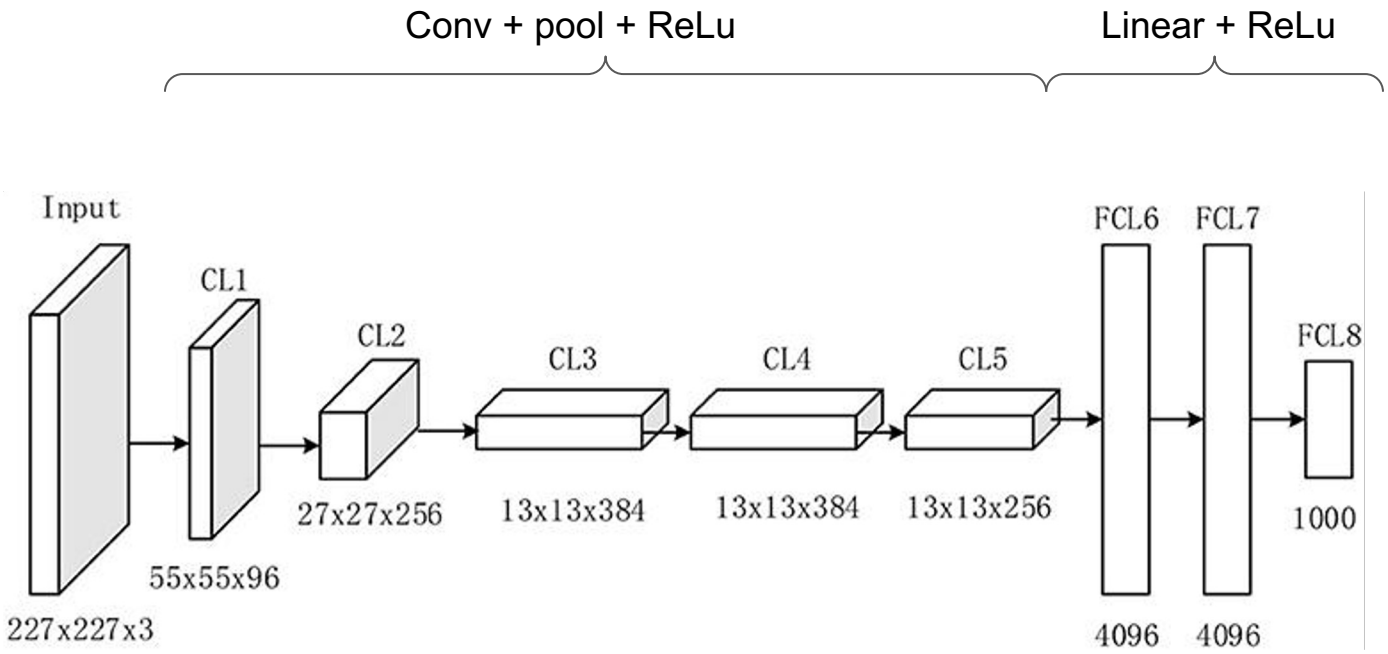
Далее на полученных фичах обучим классификатор изображений.

ImageNet Timeline



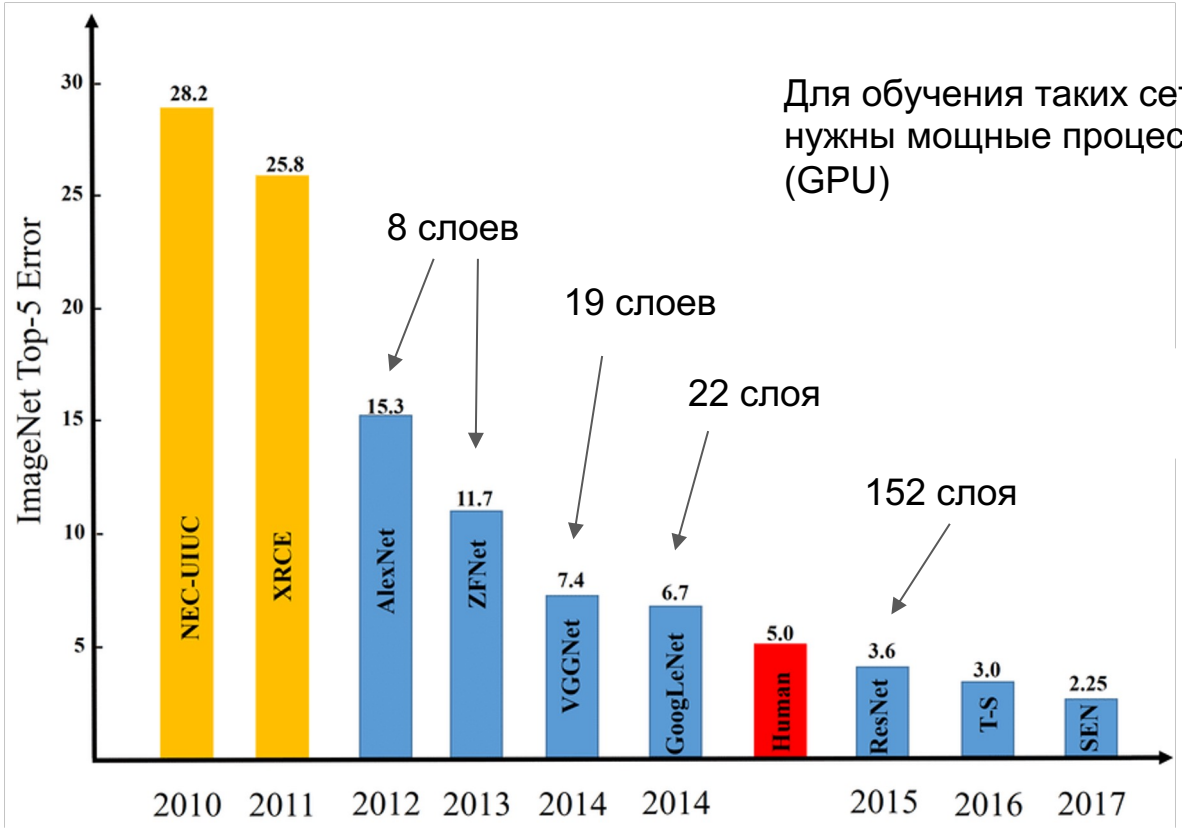
Первая успешная архитектура сверточной нейронной сети -- AlexNet

AlexNet



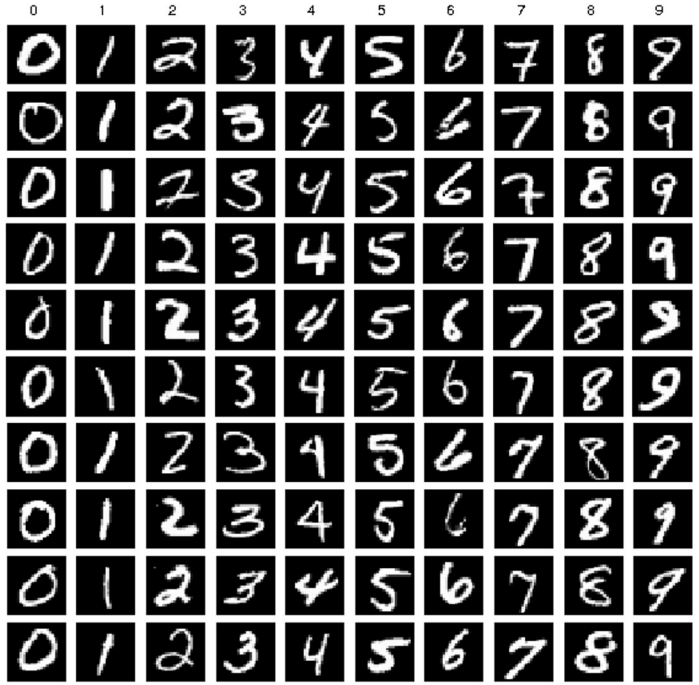
https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

ImageNet Timeline



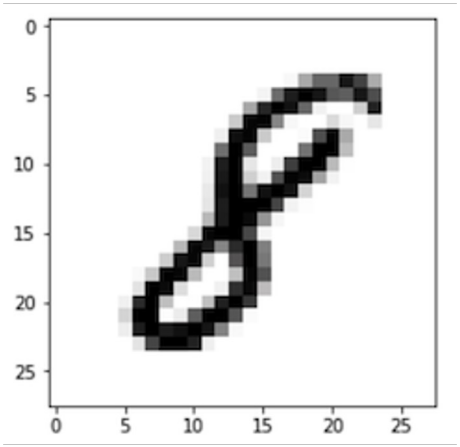
Сверточные нейронные сети

Датасет MNIST



Задача классификации на 10 классов черно-белых изображений размера 32*32

Черно-белая картинка представляется матрицей чисел из отрезка [0, 255] размера 32*32



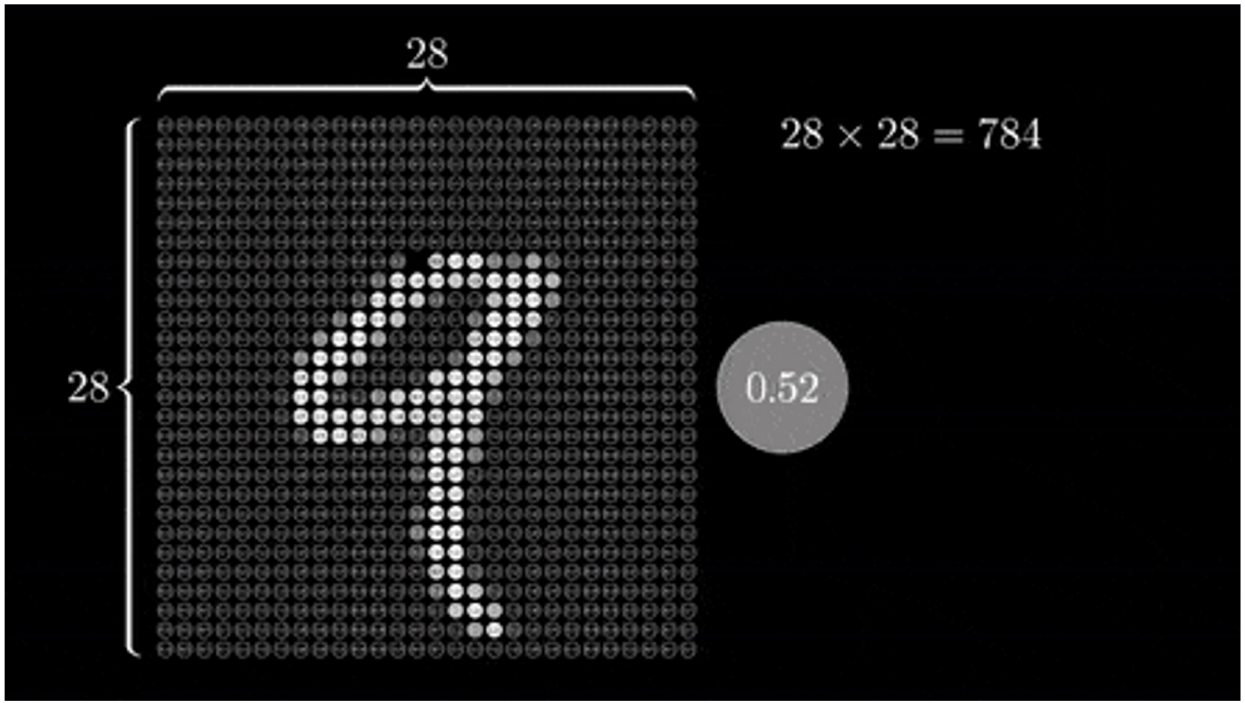
=

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	1	12	0	11	39	137	37	0	152	147	84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	41	160	250	255	235	162	255	238	206	11	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	16	9	9	150	251	45	21	184	159	154	255	233	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	145	146	3	10	0	11	124	253	255	107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	0	4	15	236	216	0	0	38	109	247	240	169	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	0	0	0	253	253	23	62	224	241	255	164	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	4	0	3	252	250	228	255	255	234	112	28	0	2	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	1	4	0	21	255	253	251	255	172	31	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	163	225	251	255	229	120	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	21	162	255	255	254	255	126	6	0	10	14	6	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	79	242	255	141	66	255	245	189	7	8	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	221	237	98	0	67	251	255	144	0	8	0	0	7	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
125	255	141	0	87	244	255	208	3	0	0	13	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
145	248	228	116	235	255	141	34	0	11	0	1	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	237	253	246	255	210	21	1	0	1	0	0	6	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	23	112	157	114	32	0	0	0	0	2	0	8	0	7	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

32

32

Растягивание картинки в вектор и подача на вход сети



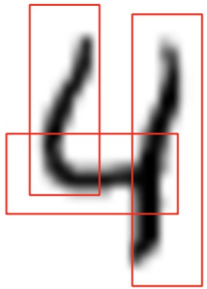
Классификация картинок полносвязной сетью:

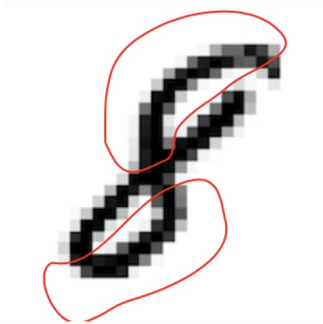
Недостатки:

- слишком много нейронов в 1 слое сети
- ломаются пространственные отношения на картинке, которые могли бы помочь сети в задаче классификации

Что отличает четверку от восьмерки?

Что отличает четверку от восьмерки?

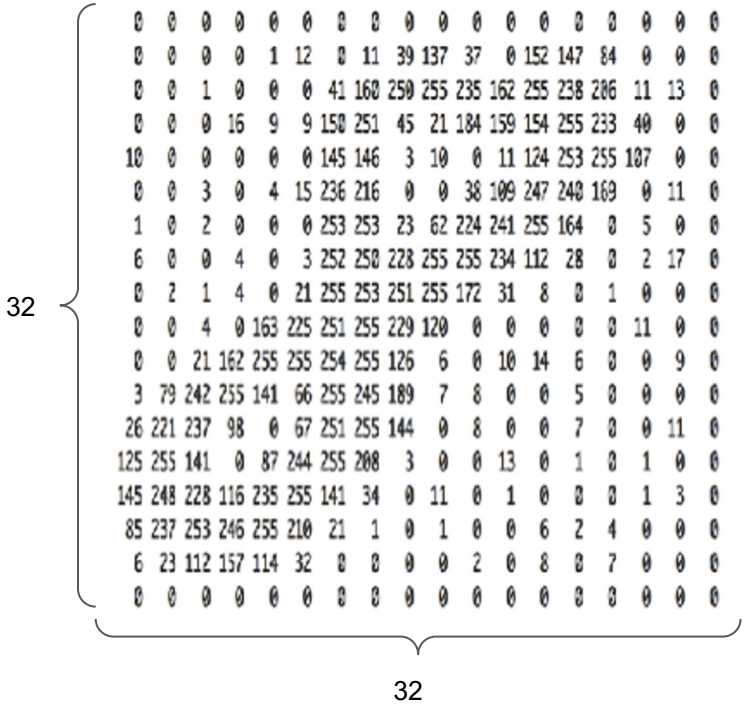




У четверки преимущественно горизонтальные и вертикальные линии,
у восьмерки линии плавные

Свертка

Изображение



Ядро (фильтр)

1	2	3
-4	7	4
2	-5	1

0	0	0
0	0	0
0	0	1

$1_{\times}0 + 2_{\times}0 + 3_{\times}0 + (-4)_{\times}0 + 7_{\times}0 + 4_{\times}0 + 2_{\times}0 + (-5)_{\times}0 + 1_{\times}1 = 1$

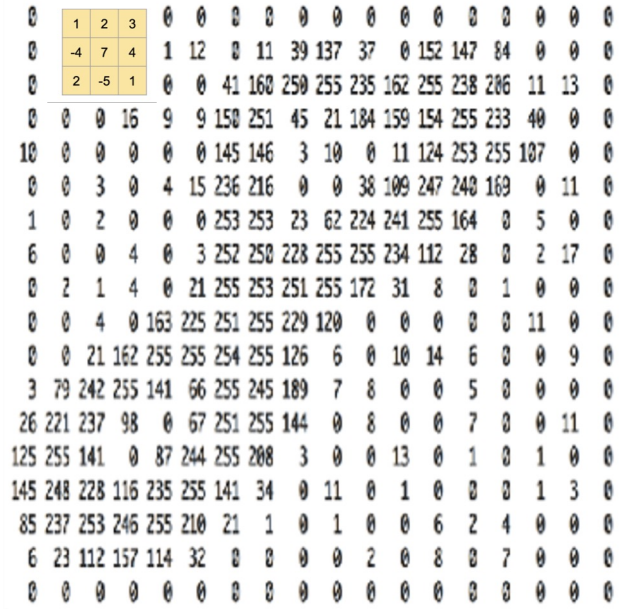
1	2	3
-4	7	4
2	-5	1

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	12	0	11	39	137	37	0	152	147	84	0	0	0		
0	0	0	41	160	250	255	235	162	255	238	206	11	13	0		
0	0	0	16	9	9	150	251	45	21	184	159	154	255	233	40	0	0
10	0	0	0	0	0	145	146	3	10	0	11	124	253	255	107	0	0
0	0	3	0	4	15	236	216	0	0	38	109	247	248	169	0	11	0
1	0	2	0	0	0	253	253	23	62	224	241	255	164	0	5	0	0
6	0	0	4	0	3	252	250	228	255	255	234	112	28	0	2	17	0
0	2	1	4	0	21	255	253	251	255	172	31	8	0	1	0	0	0
0	0	4	0	163	225	251	255	229	120	0	0	0	0	0	11	0	0
0	0	21	162	255	255	254	255	126	6	0	10	14	6	0	0	9	0
3	79	242	255	141	66	255	245	189	7	8	0	0	5	0	0	0	0
26	221	237	98	0	67	251	255	144	0	8	0	0	7	0	0	11	0
125	255	141	0	87	244	255	208	3	0	0	13	0	1	0	1	0	0
145	248	228	116	235	255	141	34	0	11	0	1	0	0	0	1	3	0
85	237	253	246	255	210	21	1	0	1	0	0	6	2	4	0	0	0
6	23	112	157	114	32	0	0	0	0	2	0	8	0	7	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1

0	0	0
0	0	0
0	1	0

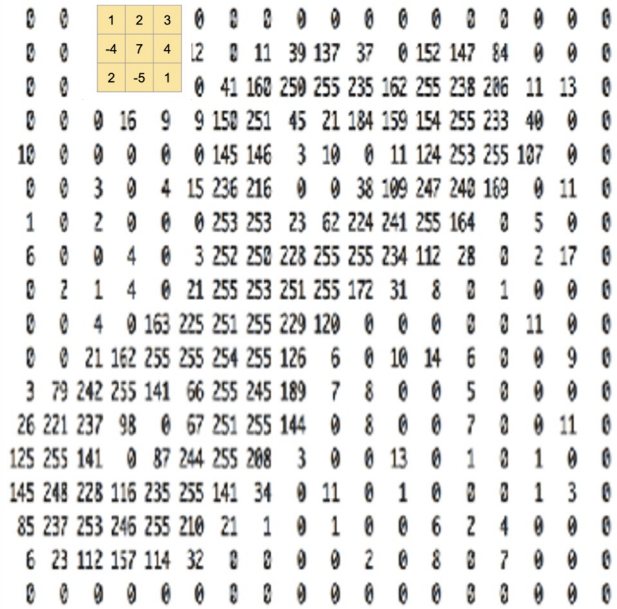
1 (stride)



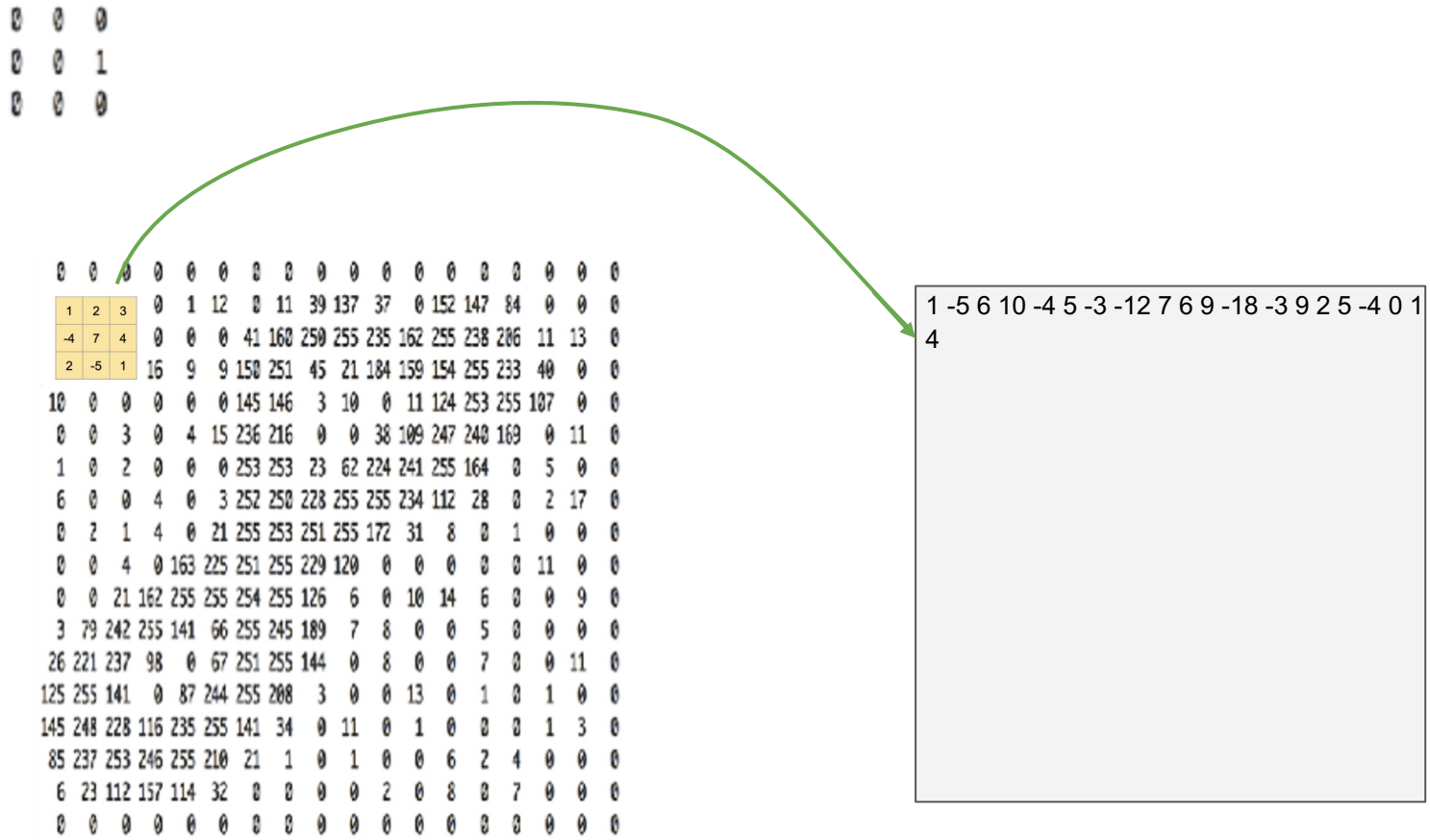
1 -5

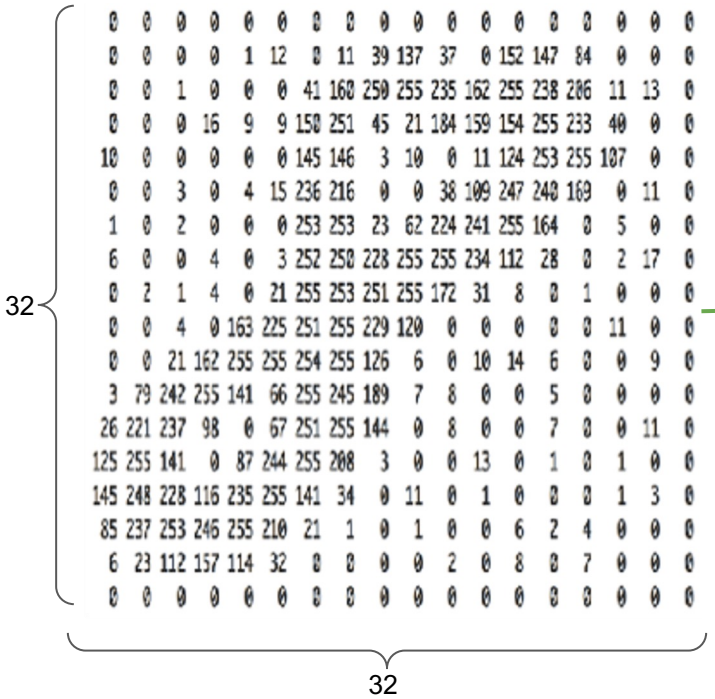
0	0	0
0	0	1
1	0	0

1 (stride)



1 -5 6


0 0 0
0 0 1
0 0 0
1 (stride)
1 2 3
-4 7 4
2 -5 1
1 -5 6 10 -4 5 -3 -12 7 6 9 -18 -3 9 2 5 -4 0 1
4



Карта активации

1 -5 6 10 -4 5 -3 -12 7 6 9 -18 -3 9 2 5 -4 0 1
4 6 -4 23 5 6 1 -7 8 -4 -3 -2 6 7 9 -4 -8 9 10 0

…

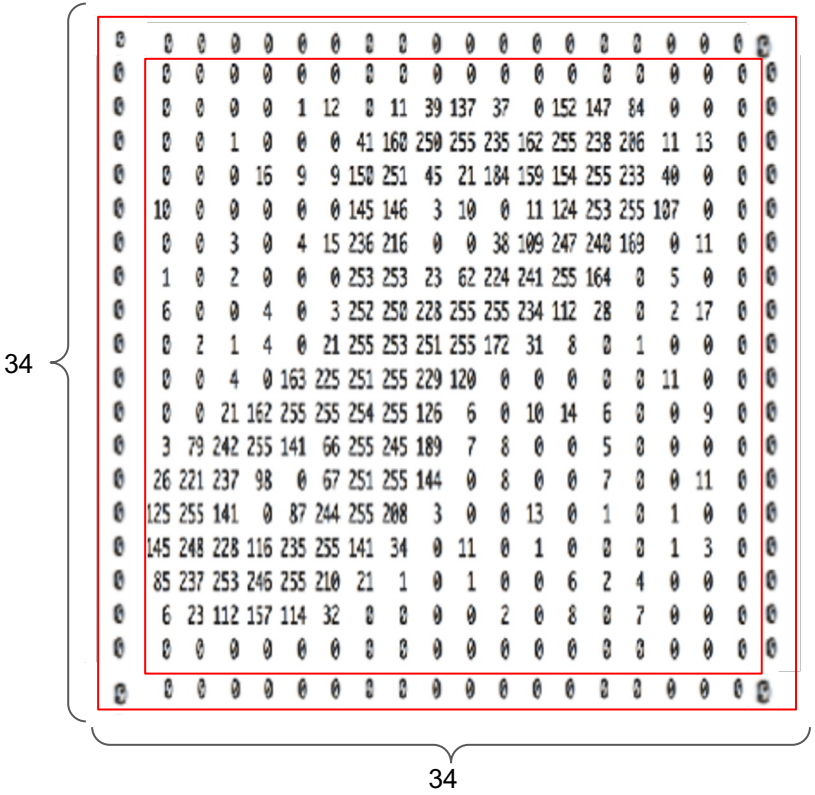
2 6 5 -3 7 -11 1 6 9 10 3 4 2 -2 4 5 9 10 -1

30

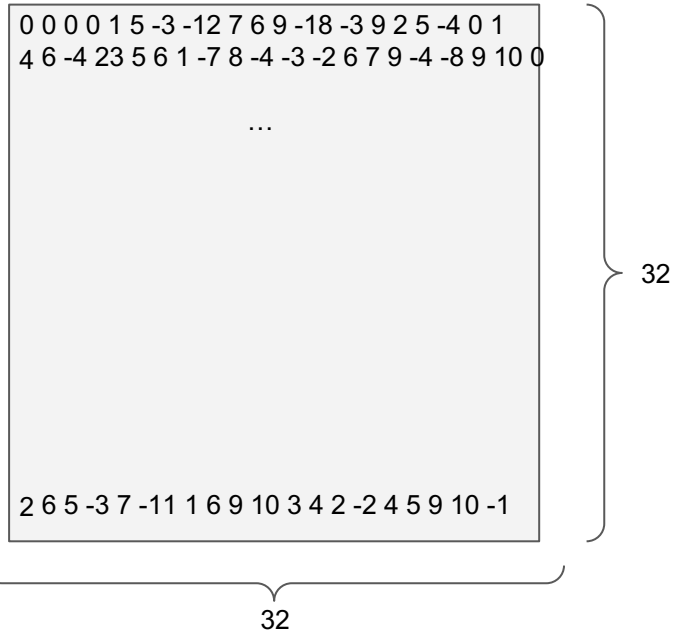
30

Padding

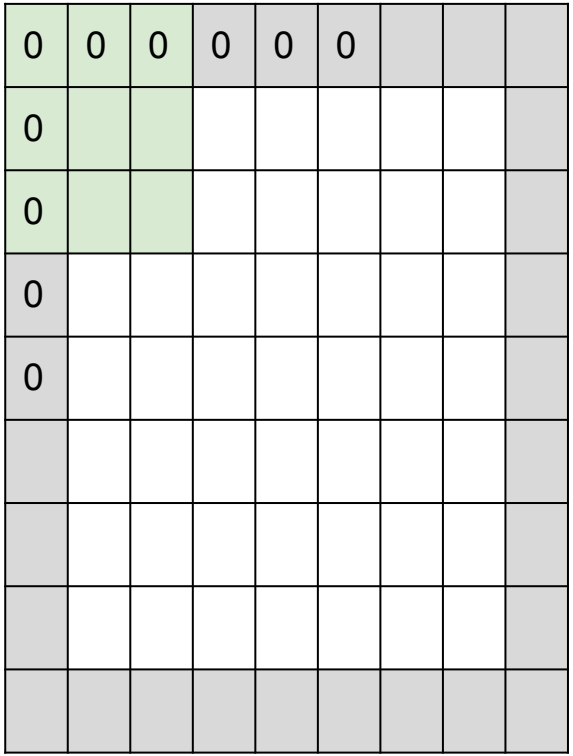
Используется для манипуляции размерами карт активаций



Карта активации



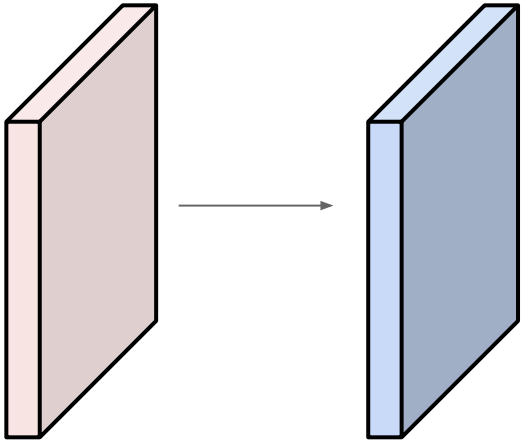
In practice: Common to zero pad the border



e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

7x7 output!
in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with (F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

Examples time:



Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Output volume size:
(32+2*2-5)/1+1 = 32 spatially, so
32x32x10

Convolution layer: summary

Let's assume input is W_1 x H_1 x C
Conv layer needs 4 hyperparameters:

- Number of filters **K**
- The filter size **F**
- The stride **S**
- The zero padding **P**

This will produce an output of W_2 x H_2 x K
where:

- $W_2 = (W_1 - F + 2P)/S + 1$
- $H_2 = (H_1 - F + 2P)/S + 1$

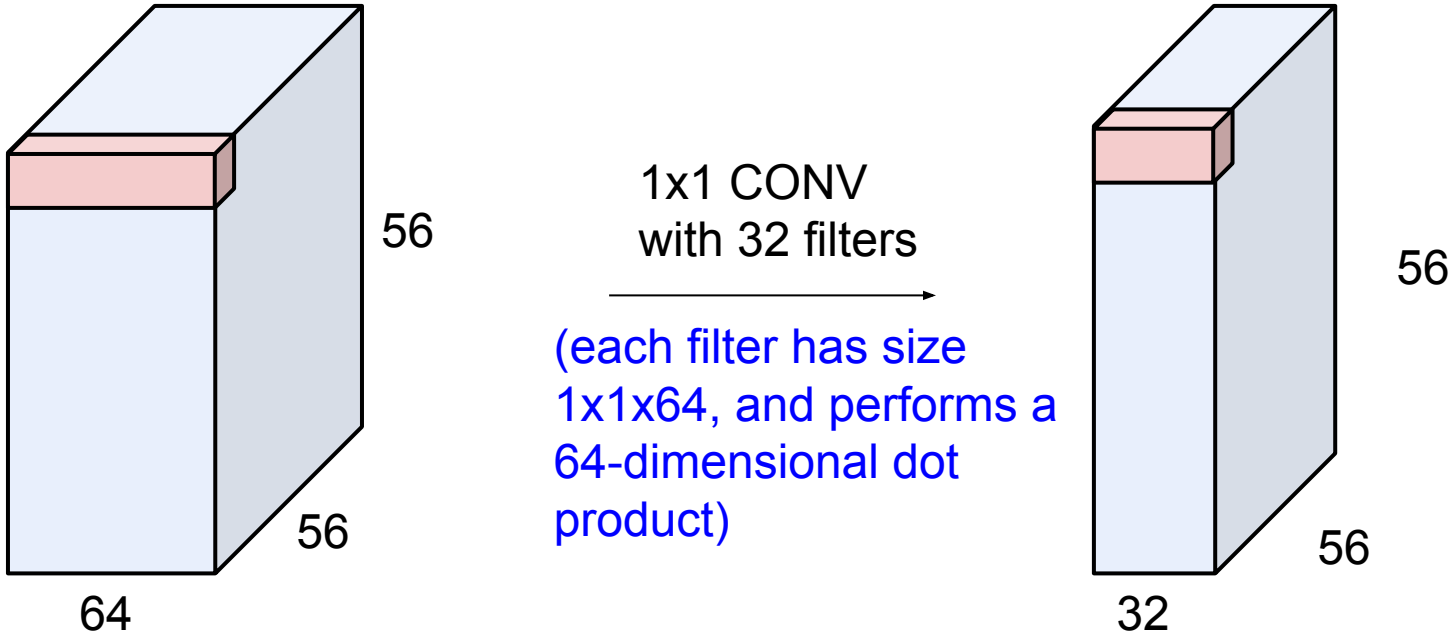
Number of parameters: F^2CK and K biases

Common settings:

K = (powers of 2, e.g. 32, 64, 128, 512)

- F = 3, S = 1, P = 1
- F = 5, S = 1, P = 2
- F = 5, S = 2, P = ? (whatever fits)
- F = 1, S = 1, P = 0

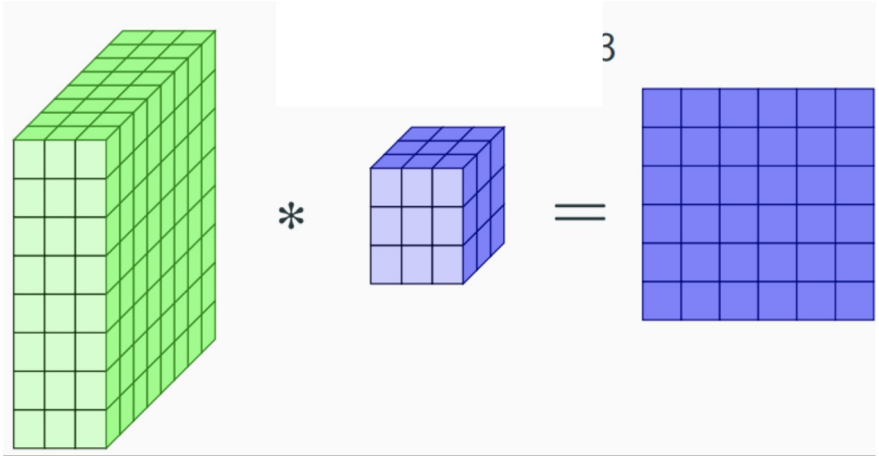
(btw, 1x1 convolution layers make perfect sense)



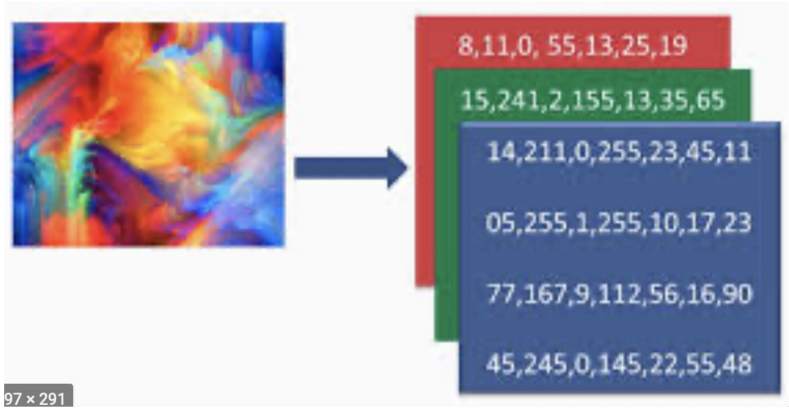
Свертка цветных изображений

Цветное (RGB) изображение трехмерное (h*w*3)

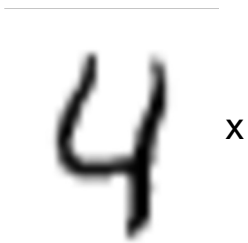
3d свертка



2d свертка отдельно по каждому цветовому каналу



Фильтры “реагируют” на паттерны на изображении. Если паттерн присутствует на изображении, то карта активации после соотв. фильтра будет содержать большие числа



x

Фильтр, который реагирует на вертикальные линии

-1	2	-1
-1	2	-1
-1	2	-1

Активация сильнее, на карте активации большие числа

34	55	64	73	23
13	15	23	-86	-96
12	-3	0.4	71	19
11	14	17	-35	19

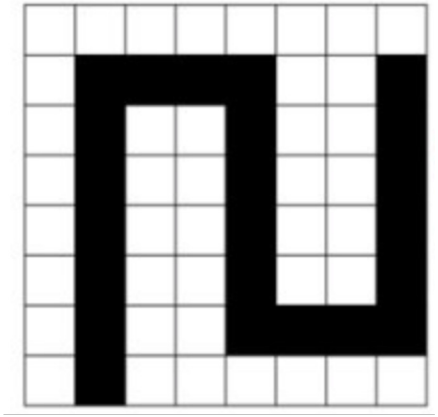
-1	-1	2
-1	2	-1
2	-1	-1

Фильтр, который реагирует на изогнутые линии

0.01	-0.2	1.8	2	-1.5
3	-0.5	6	-7	0.4
4	5	-0.8	-5	-1
1.2	0.5	3	-3	0.4

Активация слабее, на карте активации маленькие числа

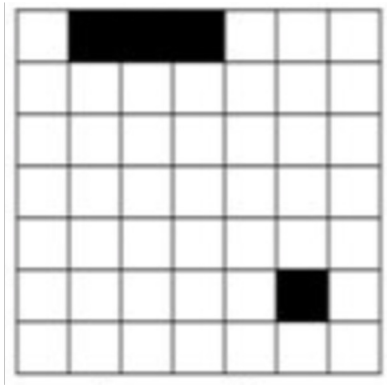
Как мог бы выглядеть фильтр, реагирующий на горизонтальные линии



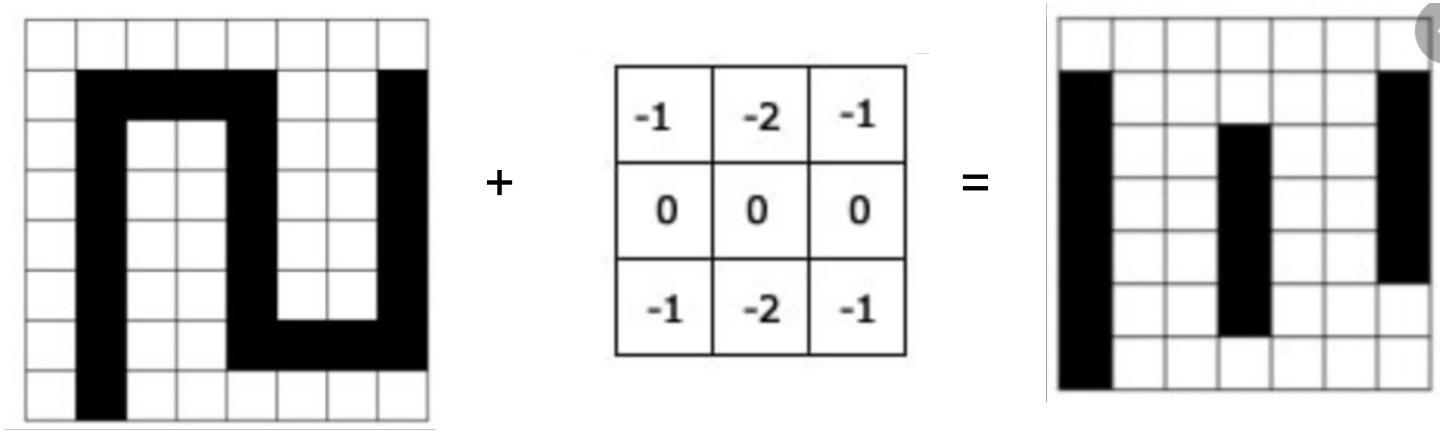
+

-1	0	1
-1	0	1
-1	0	1

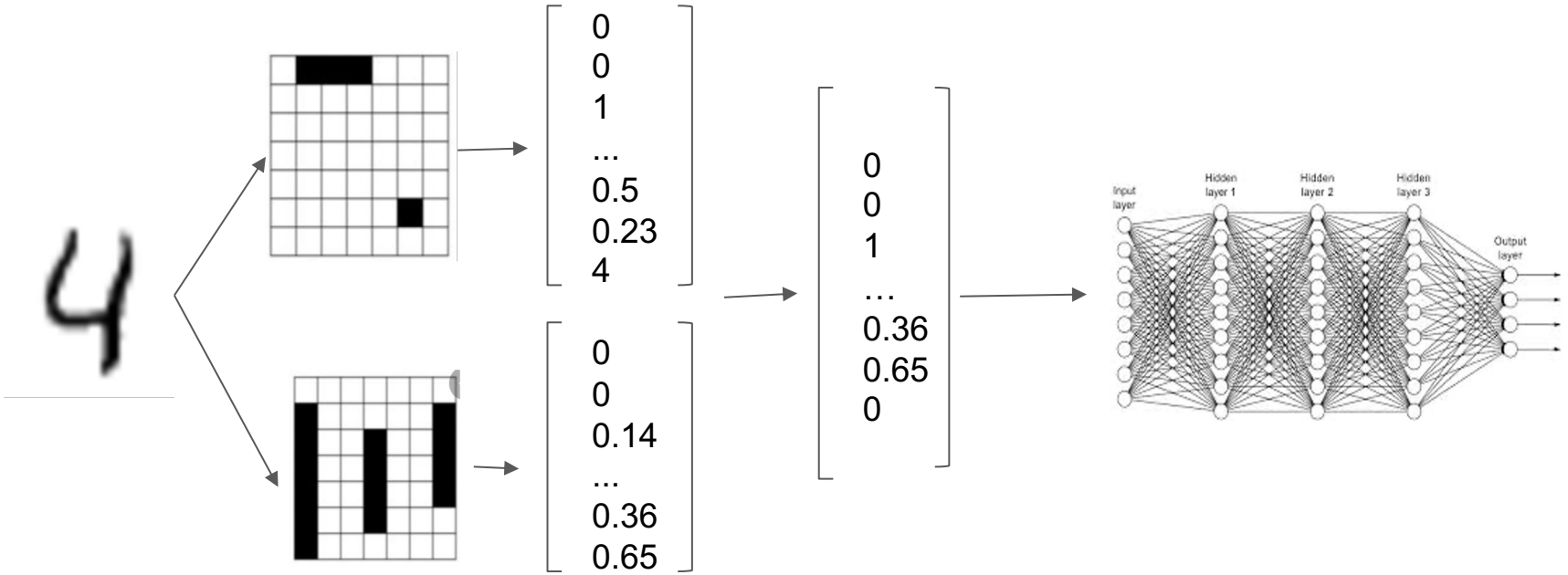
=



Как мог бы выглядеть фильтр, реагирующий на вертикальные линии

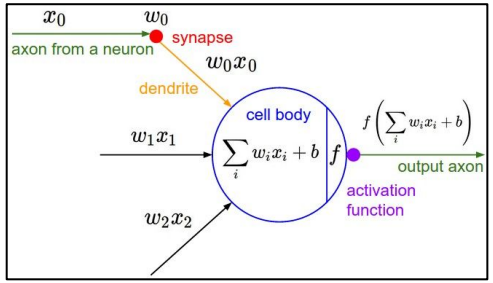


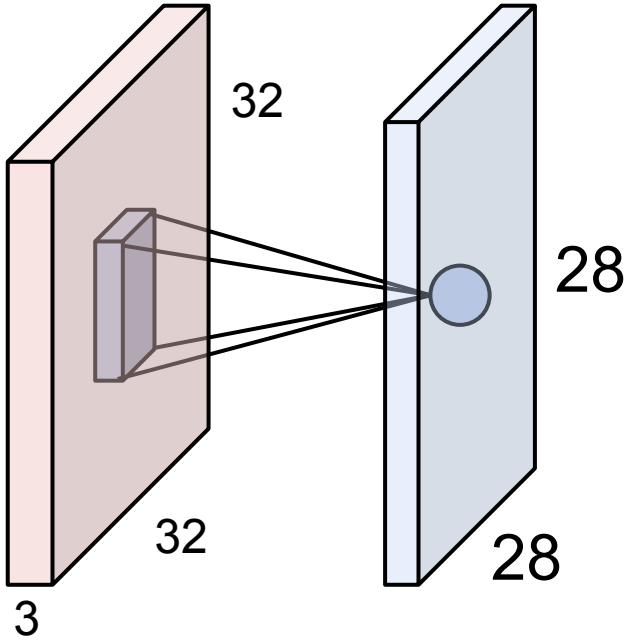
После получения карт активаций, мы **развернем все карты в векторы, сконкатенируем** и подадим на вход полносвязной сети



свертки flatten конкатенация подача в полносвязную сеть

Receptive field



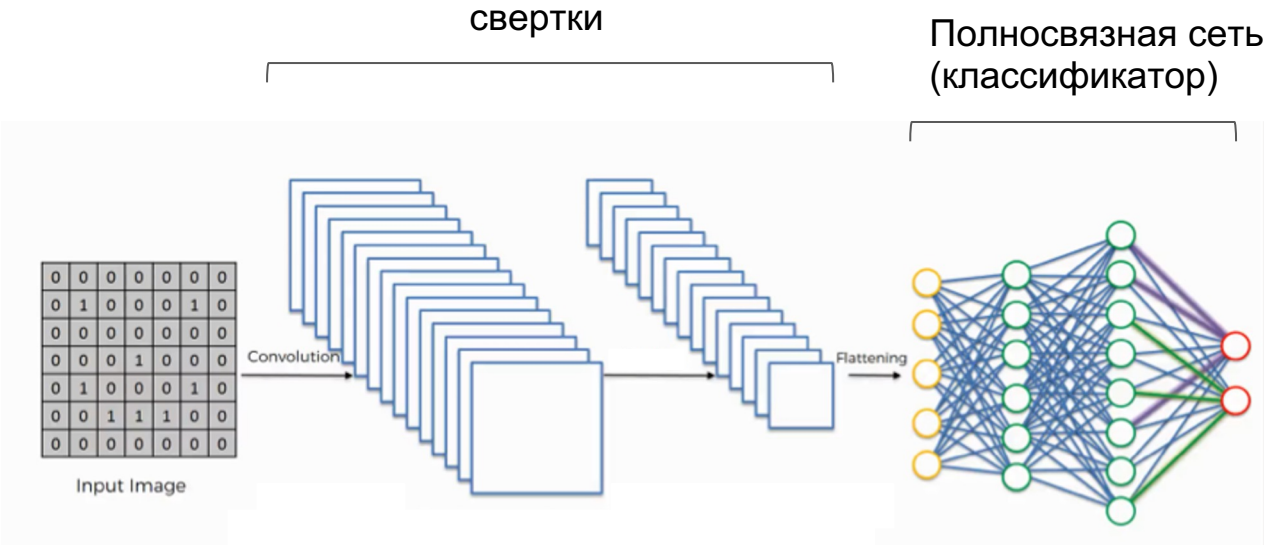


An activation map is a 28x28 sheet of neuron outputs:

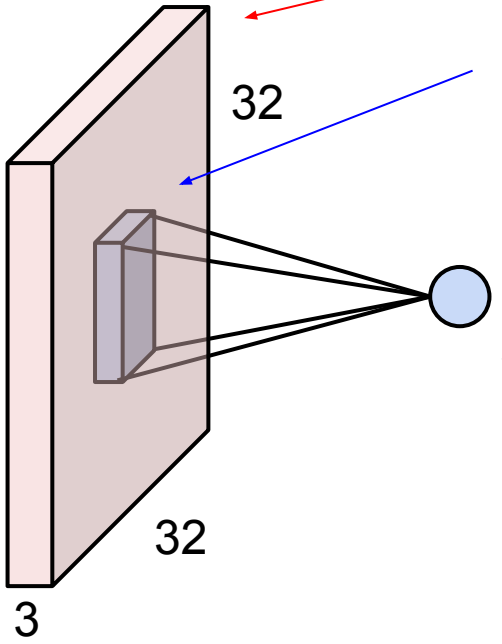
1. Each is connected to a small region in the input
2. All of them share parameters

"5x5 filter" -> "5x5 receptive field for each neuron"

Сверточная нейросеть



Convolution Layer



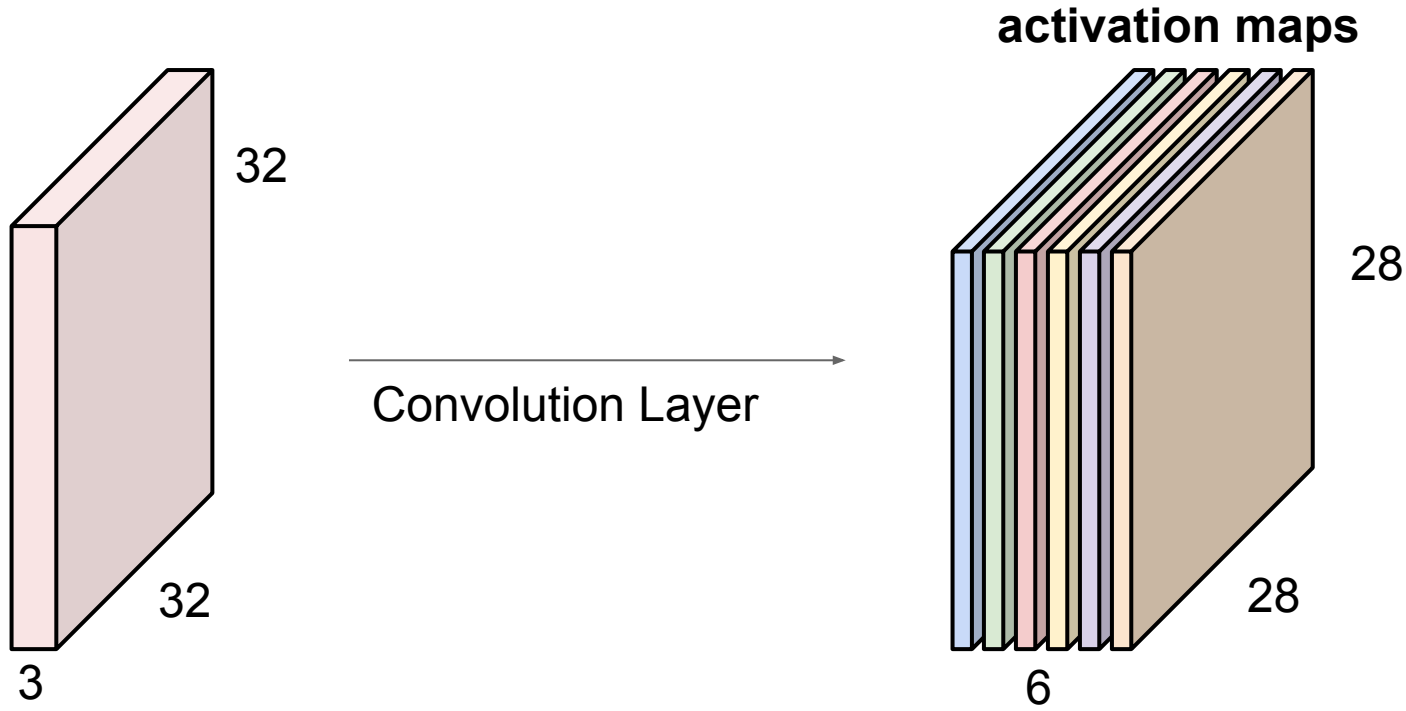
32x32x3 image

5x5x3 filter w

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

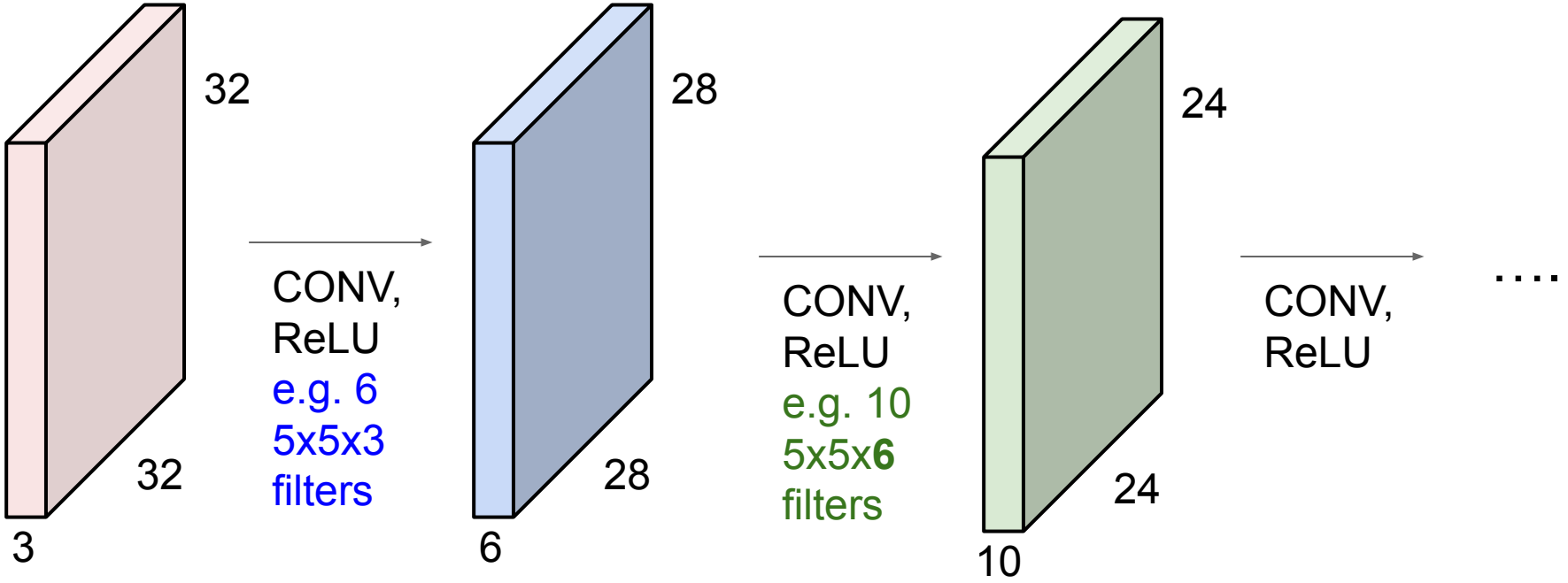
$$w^T x + b$$

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

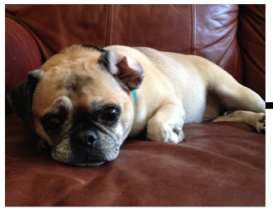
Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



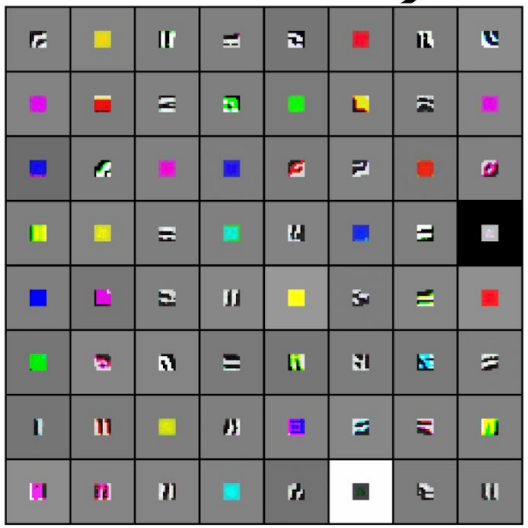
Preview

[Zeiler and Fergus 2013]

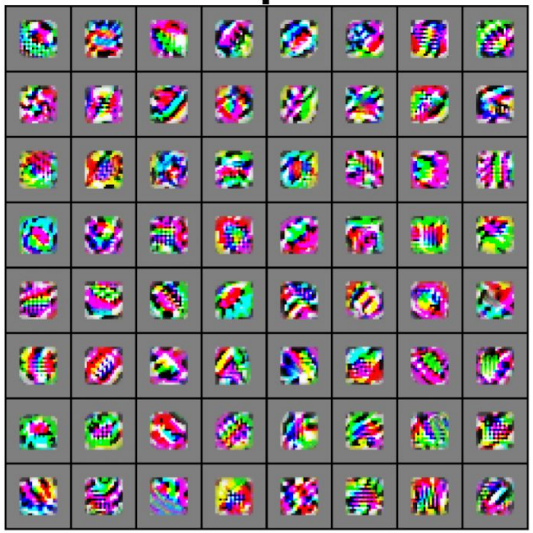
Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].



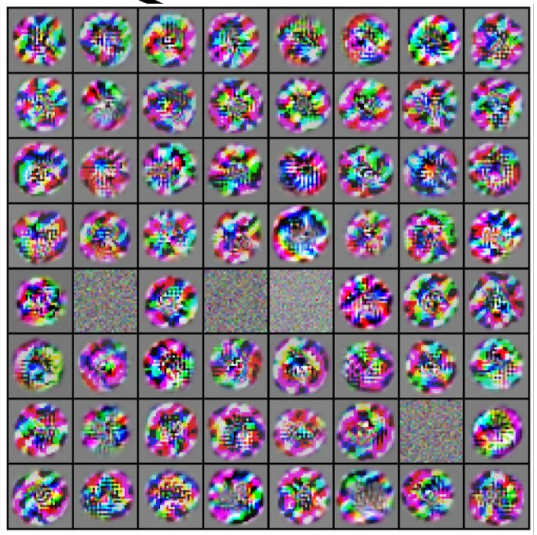
Low-level features → Mid-level features → High-level features → Linearly separable classifier



VGG-16 Conv1_1



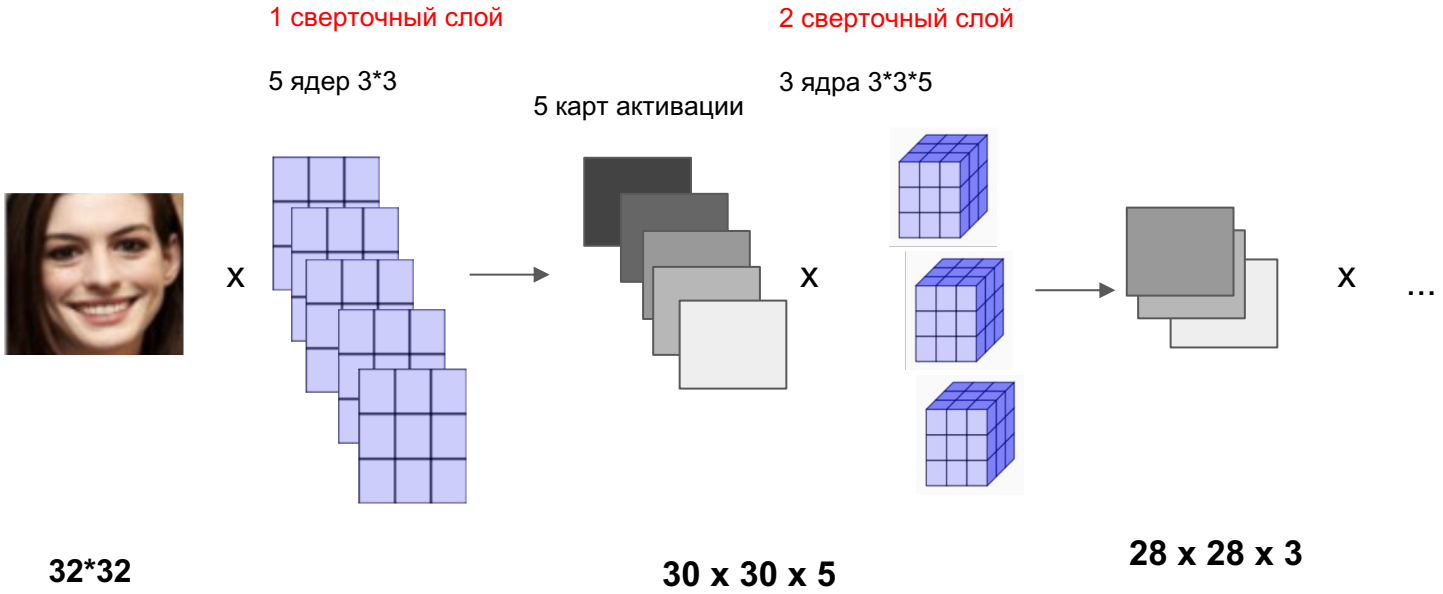
VGG-16 Conv3_2



VGG-16 Conv5_3

Для real-world изображений одной операции свертки не хватит, чтобы выделить всю нужную информацию из изображения

Потребуется несколько слоев сверток

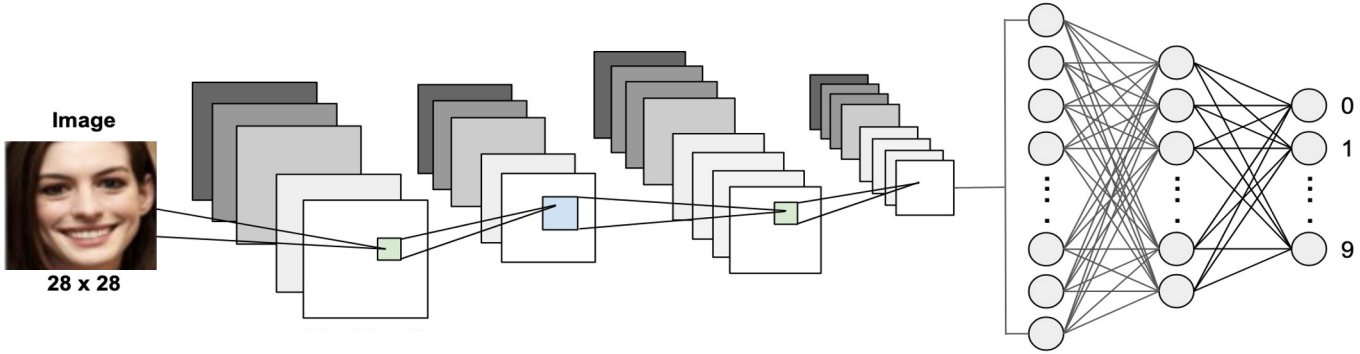


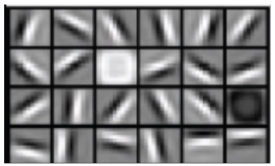
1 слой сверток

2 слой сверток

3 слой сверток

4 слой сверток





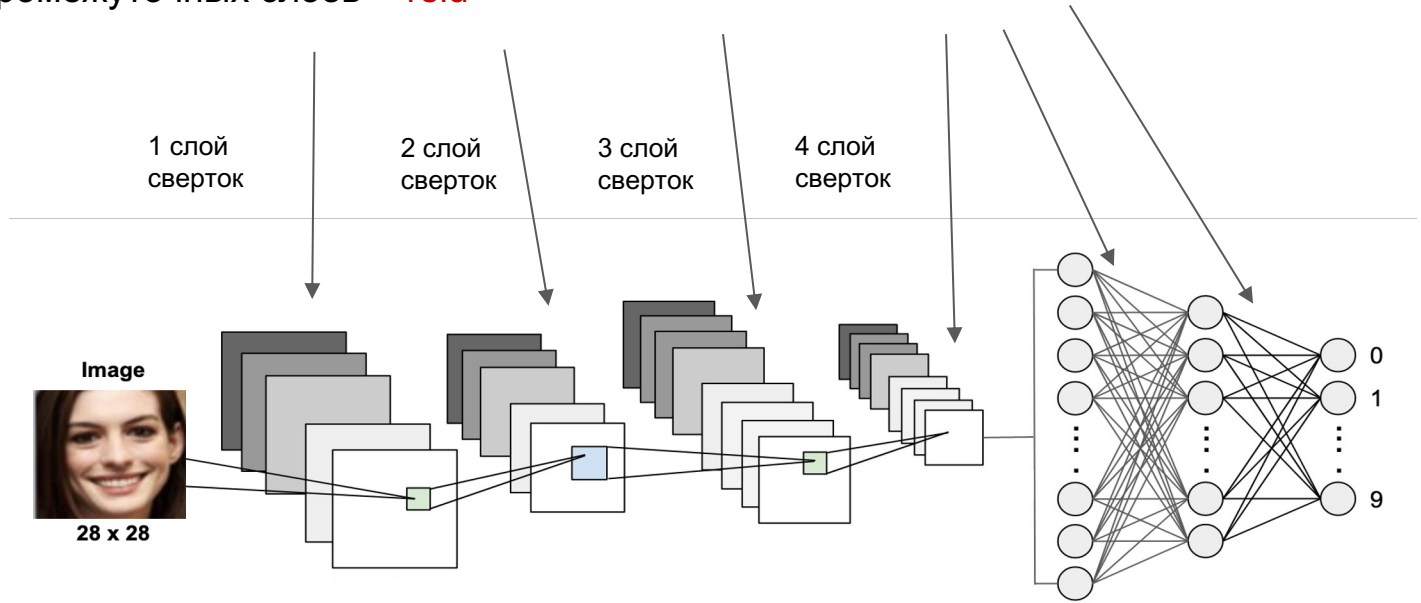
Низкоуровневые паттерны

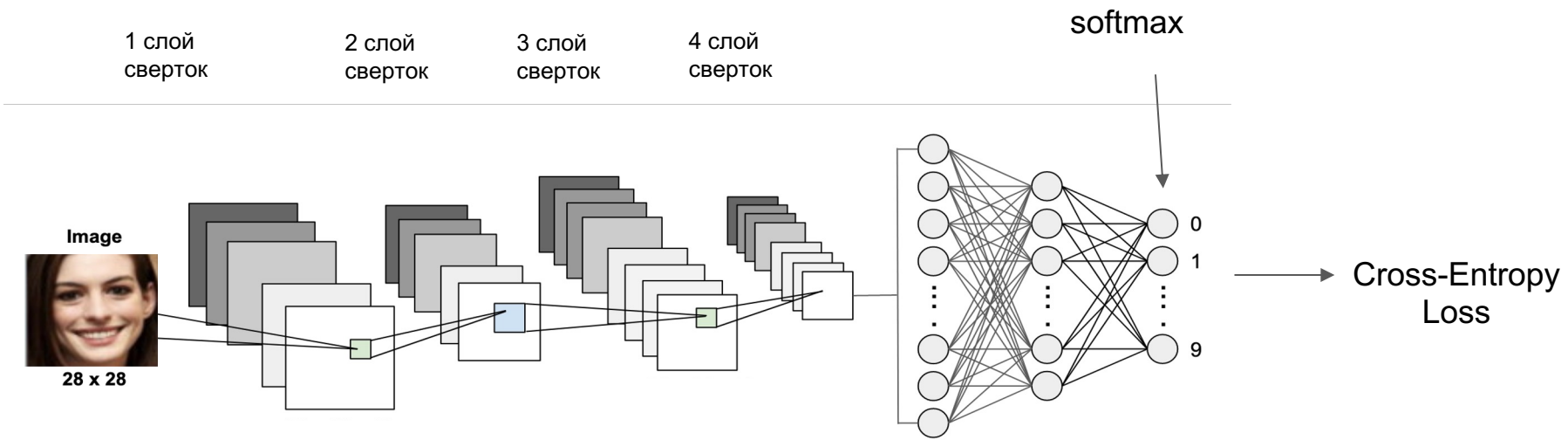
Вырисовываются отдельные части

Выделены признаки изображения, важные для задачи

После сверточных слоев, как и после полносвязных, используется функция активации.

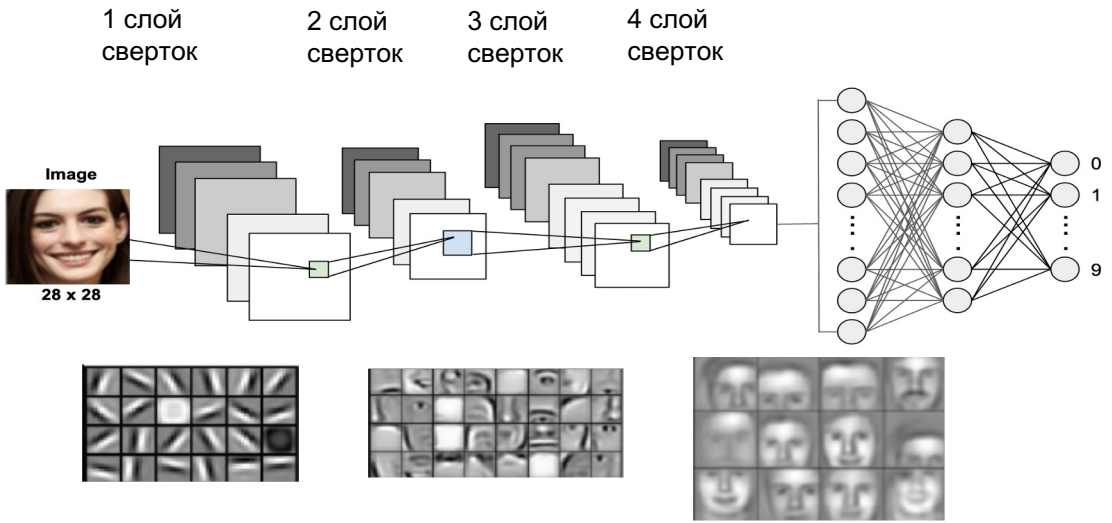
Самая популярная и хорошо работающая функция активации промежуточных слоев -- relu




1 слой сверток
2 слой сверток
3 слой сверток
4 слой сверток
softmax
Image
28 x 28
0
1
9
Cross-Entropy Loss

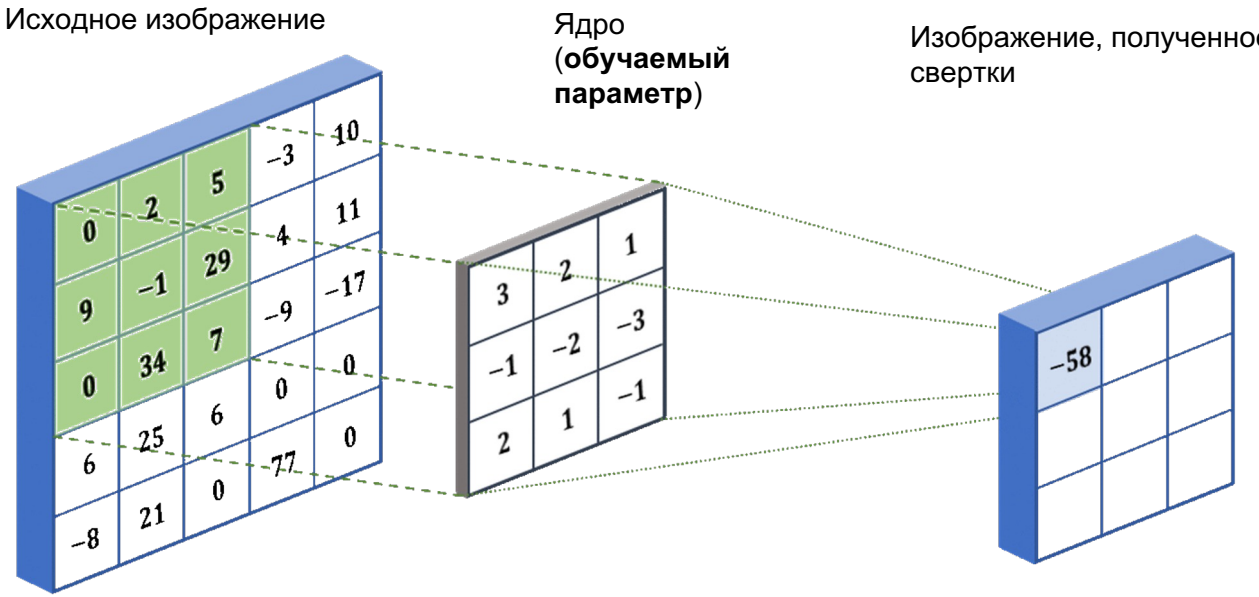
Свертки -- как лампочки, которые “загораются” сильнее, если на изображении есть определенный паттерн

На картах активации вы НЕ найдете никакого “понятного” рисунка: они есть индикаторы наличия некоего паттерна на картинке



Обучение нейросети

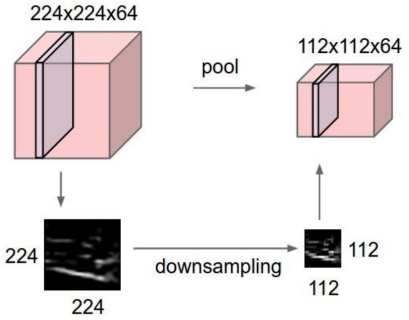
Нейросеть сама учится понимать, какие паттерны на изображении ей важно уметь находить.



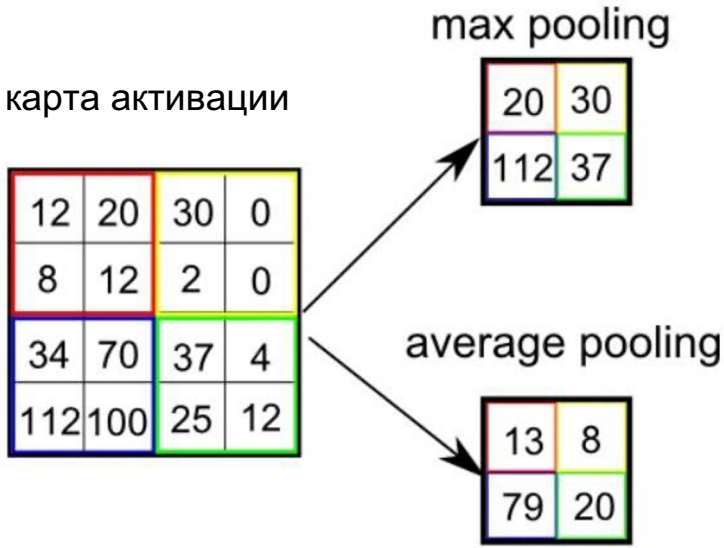
Pooling

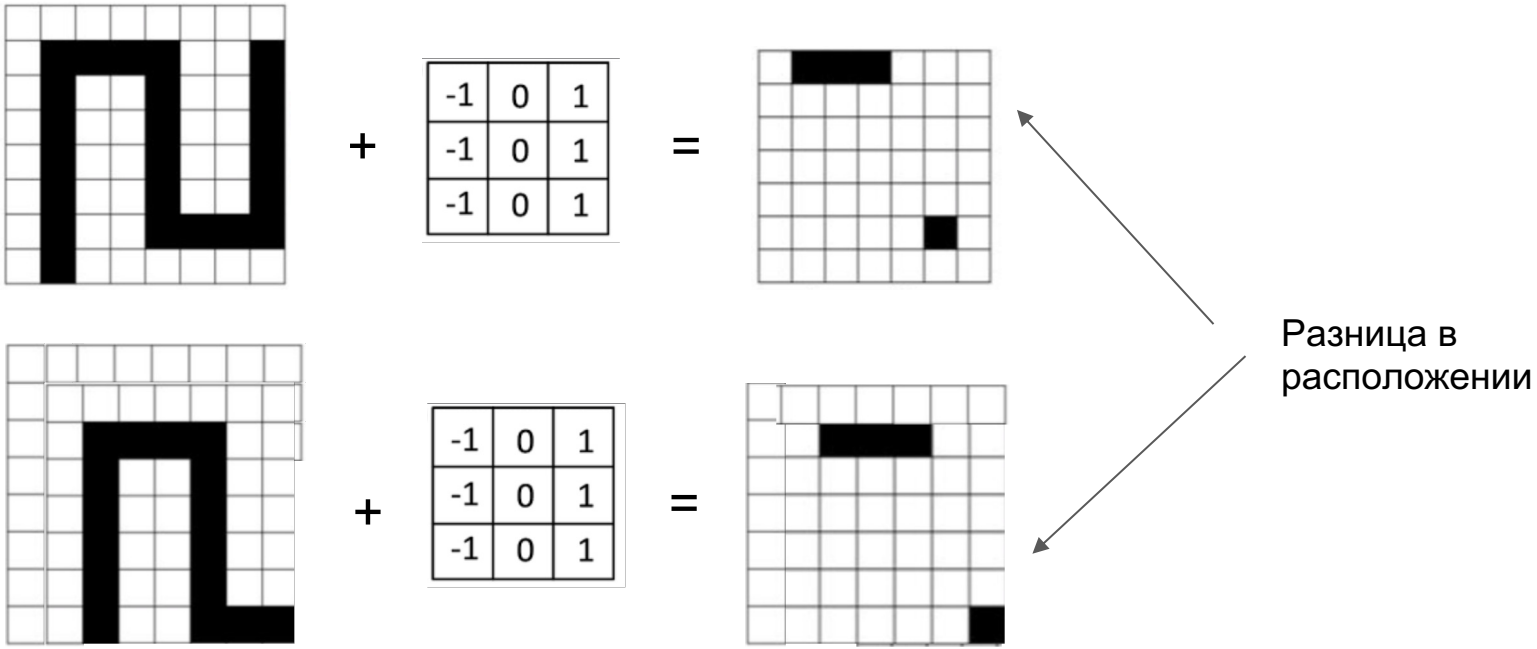
Pooling

Техника уменьшения размерности (downsampling’a) карт активаций

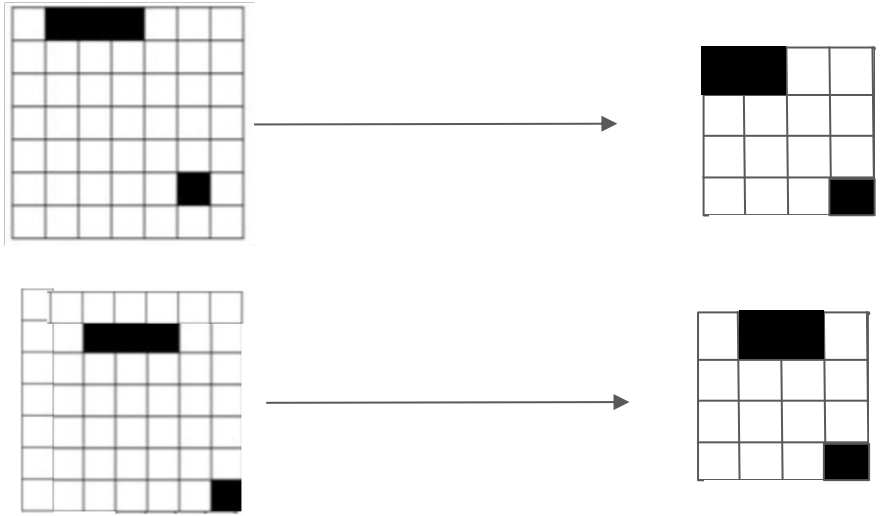


Используется для:
— уменьшения размерности очень больших изображений
—уменьшения чувствительности сверток к положению объектов на картинке




-1 0 1
-1 0 1
-1 0 1
+
=
-1 0 1
-1 0 1
-1 0 1
+
=
Разница в расположении

Результат применения 2x2 MaxPooling’а к картам активаций:



Pooling layer

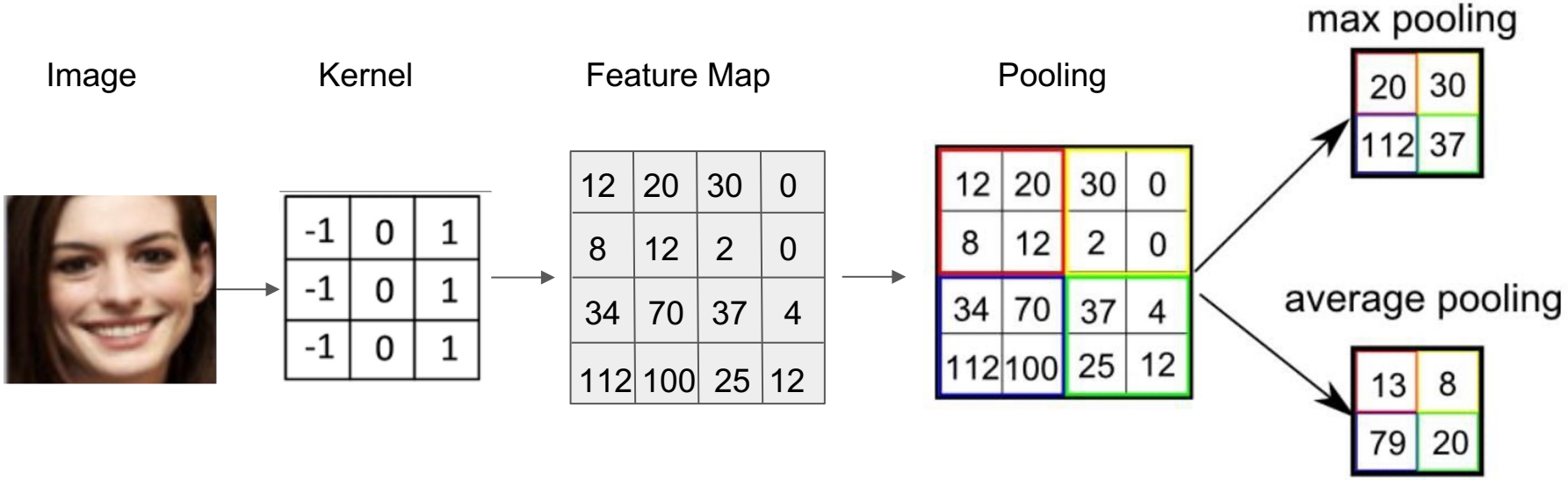


Image -> conv -> act -> pool -> conv ...

Pooling layer: summary

Let's assume input is W_1 x H_1 x C
Conv layer needs 2 hyperparameters:

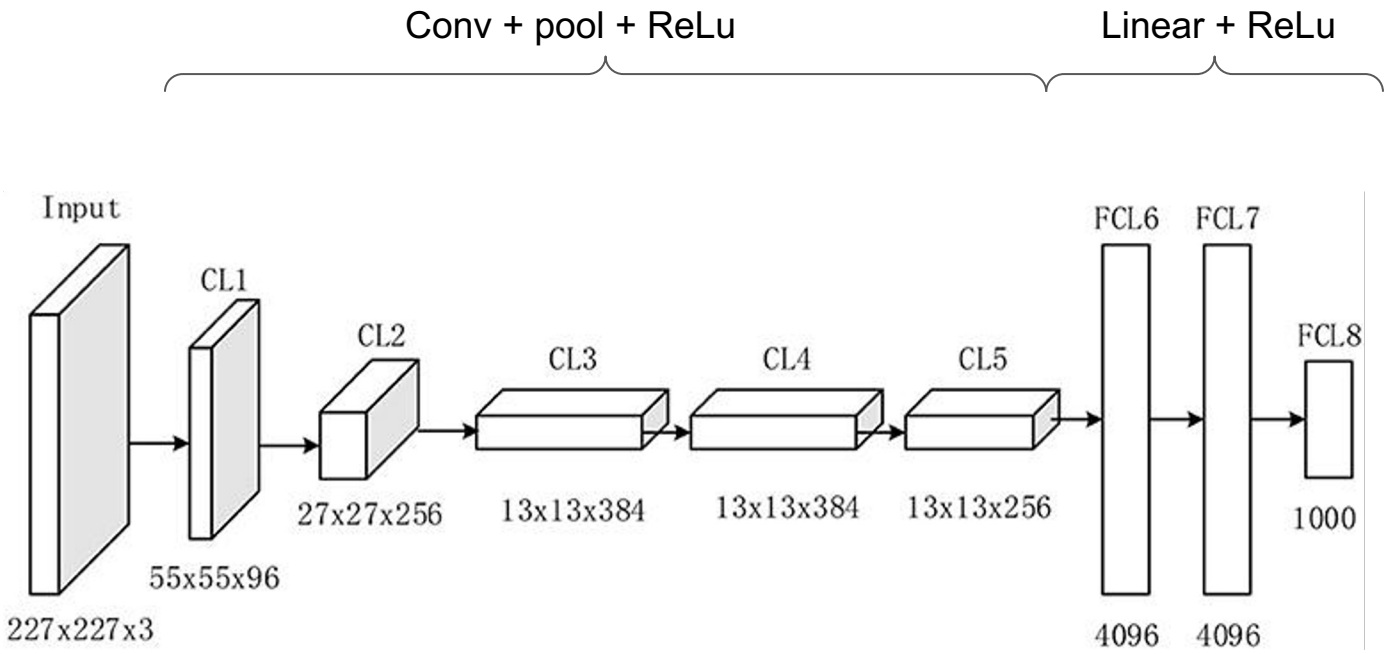
- The spatial extent **F**
- The stride **S**

This will produce an output of W_2 x H_2 x C where:

- $W_2 = (W_1 - F)/S + 1$
- $H_2 = (H_1 - F)/S + 1$

Number of parameters: 0

AlexNet

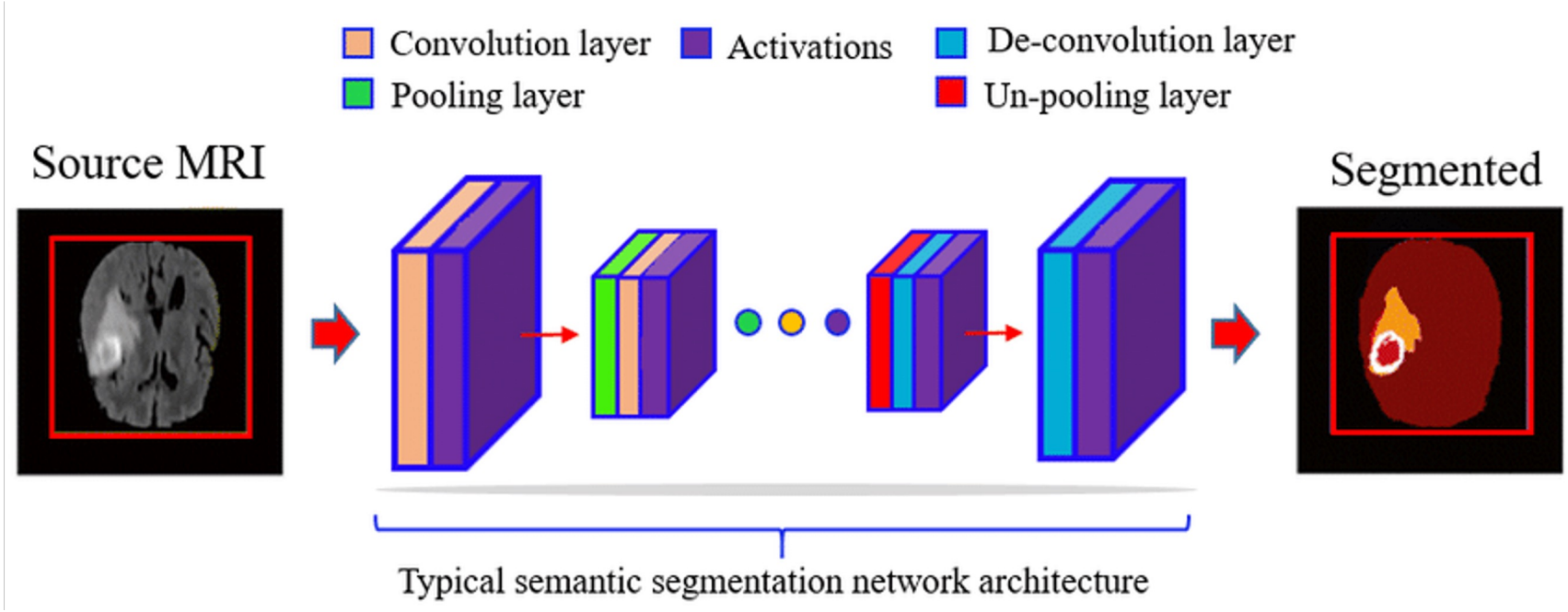


https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Задачи компьютерного зрения (CV)

Классификация, детекция, сегментация

Сегментация — важная задача для медицины

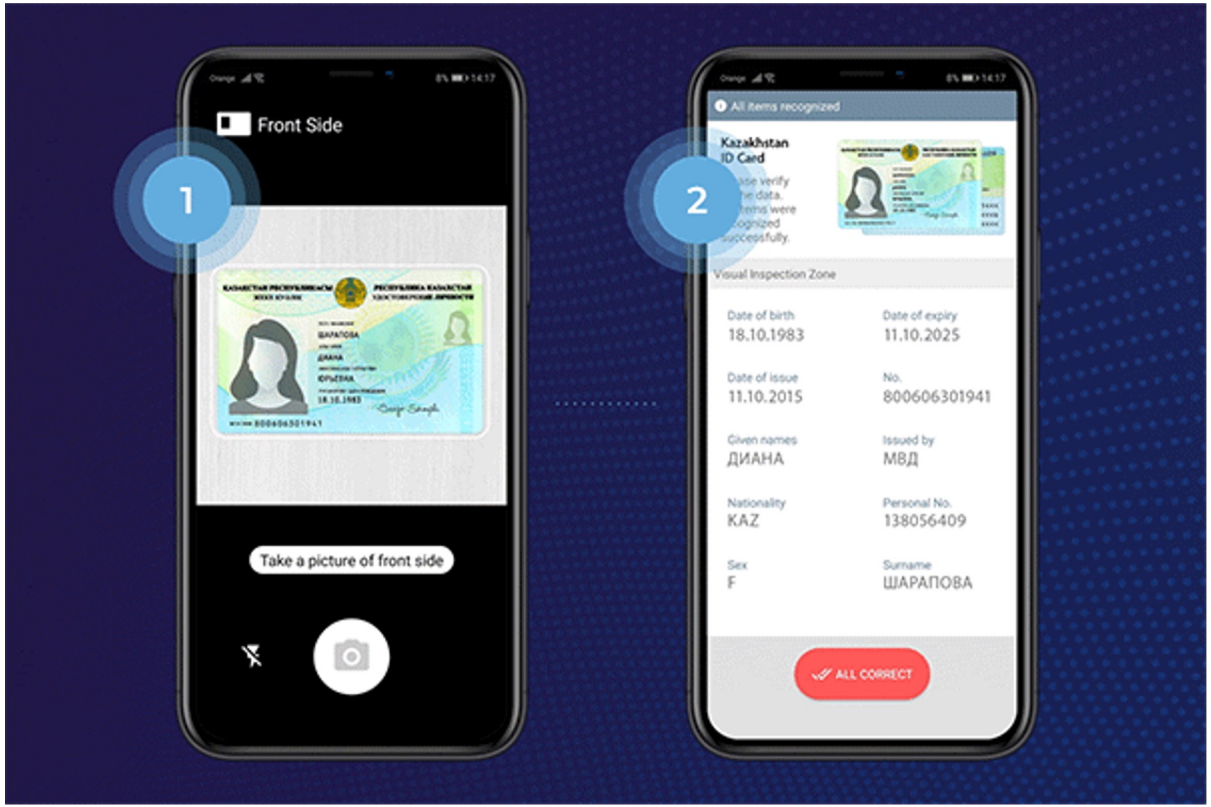


...и все эти задачи очень важны для беспилотных автомобилей.

- детекция
- классификация
- сегментация
- поиск по изображениям
- оценка положения

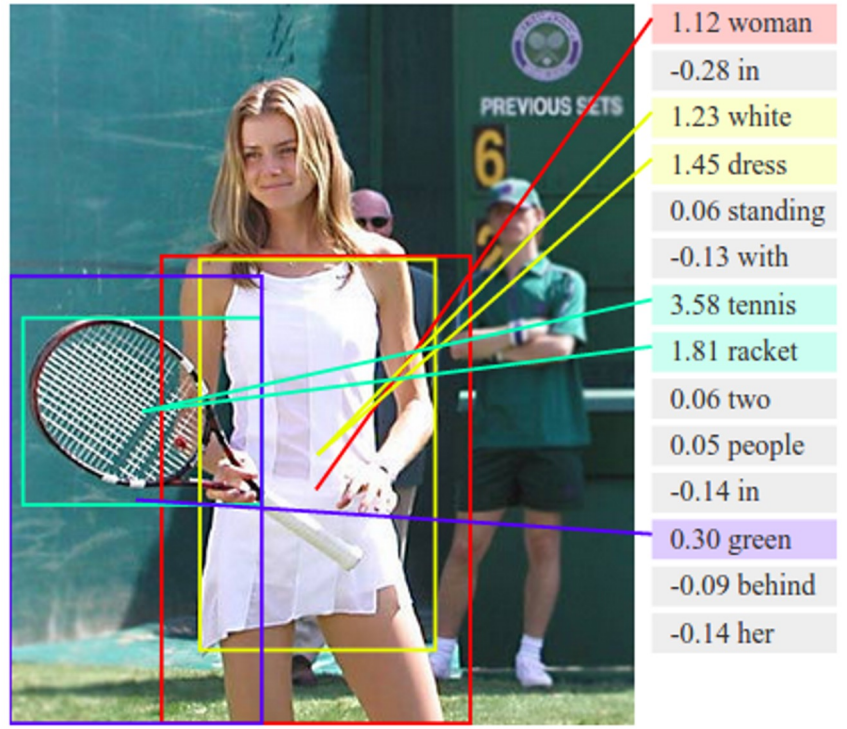
Optical Character Recognition

- автоматический перевод
- улучшение качества фотографий документов
- скан чеков/визиток/etc



Video Understanding

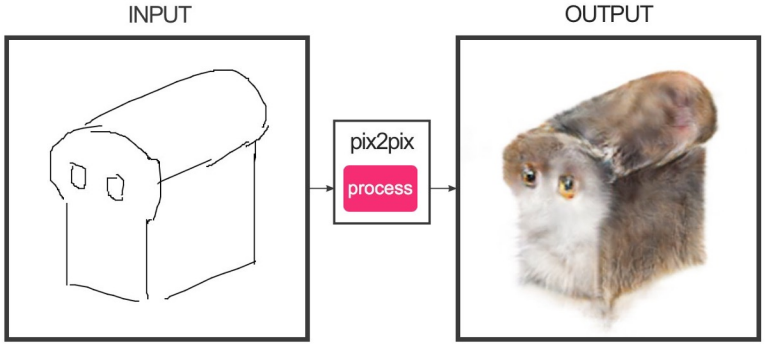
- image/video description
- subtitles to video
- action description
- pose estimation



GANs (о них на последнем занятии)

Style transfer

Image generation/completion



https://www.youtube.com/watch?v=PCBTZh41Ris

https://www.youtube.com/watch?v=PCBTZh41Ris

This image by Christin Khan is in the public domain and originally came from the U.S. NOAA.

Whale recognition, Kaggle Challenge

Photo and figure by Lane McIntosh; not actual example from Mnih and Hinton, 2010 paper.

Mnih and Hinton, 2010

Image Captioning

[Vinyals et al., 2015]
[Karpathy and Fei-Fei, 2015]

No errors

A white teddy bear sitting in the grass

A man riding a wave on top of a surfboard

Minor errors

A man in a baseball uniform throwing a ball

A cat sitting on a suitcase on the floor

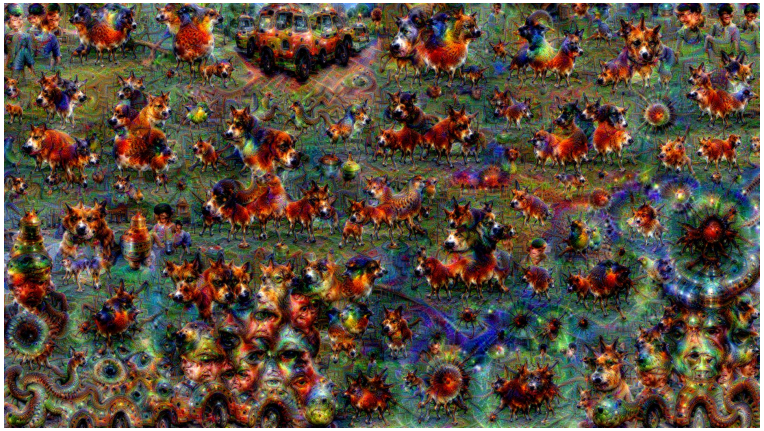
Somewhat related

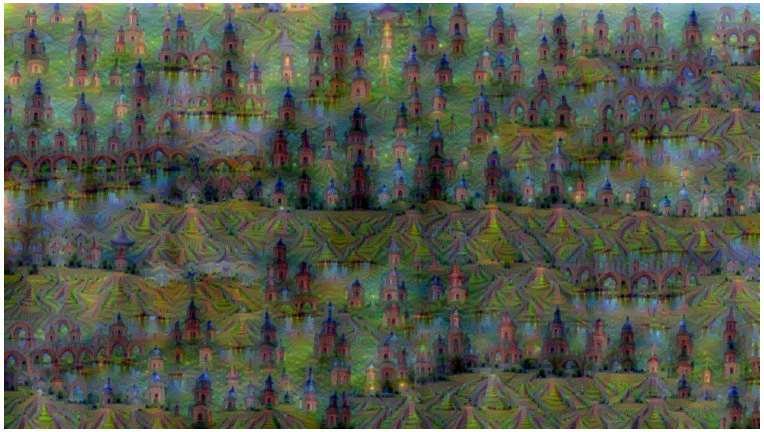
A woman is holding a cat in her hand

A woman standing on a beach holding a surfboard

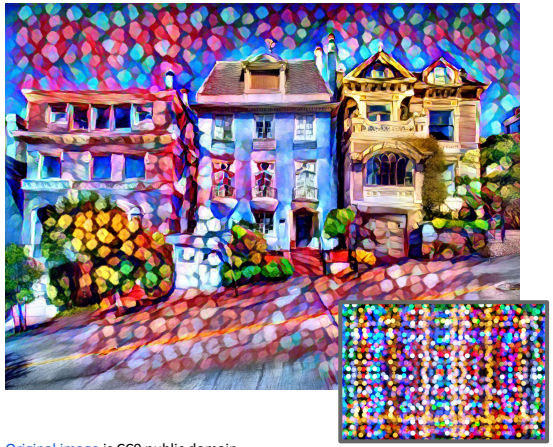
All images are CC0 Public domain:
https://pixabay.com/en/luggage-antique-cat-1643010/
https://pixabay.com/en/teddy-plush-bears-cute-teddy-bear-1623436/
https://pixabay.com/en/surf-wave-summer-sport-litoral-1668716/
https://pixabay.com/en/woman-female-model-portrait-adult-983967/
https://pixabay.com/en/handstand-lake-meditation-496008/
https://pixabay.com/en/baseball-player-shortstop-infield-1045263/

Captions generated by Justin Johnson using Neuraltalk2





Figures copyright Justin Johnson, 2015. Reproduced with permission. Generated using the Inceptionism approach from a blog post by Google Research.



Original image is CC0 public domain
Starry Night and Tree Roots by Van Gogh are in the public domain
Bokeh image is in the public domain
Stylized images copyright Justin Johnson, 2017;
reproduced with permission

Gatys et al, "Image Style Transfer using Convolutional Neural Networks", CVPR 2016
Gatys et al, "Controlling Perceptual Factors in Neural Style Transfer", CVPR 2017