Computer Vision

Nikita Kisialeu

fork by Tatiana Gaintseva
(DLS)

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.

Класс	Вероятность
Самолет	0.40
Танк	0.25
Автомобиль	0.15
Велосипед	0.10
Корабль	0.05
Мотоцикл	0.04
Поезд	0.01

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 = 12

по OX: 89-78 = 11

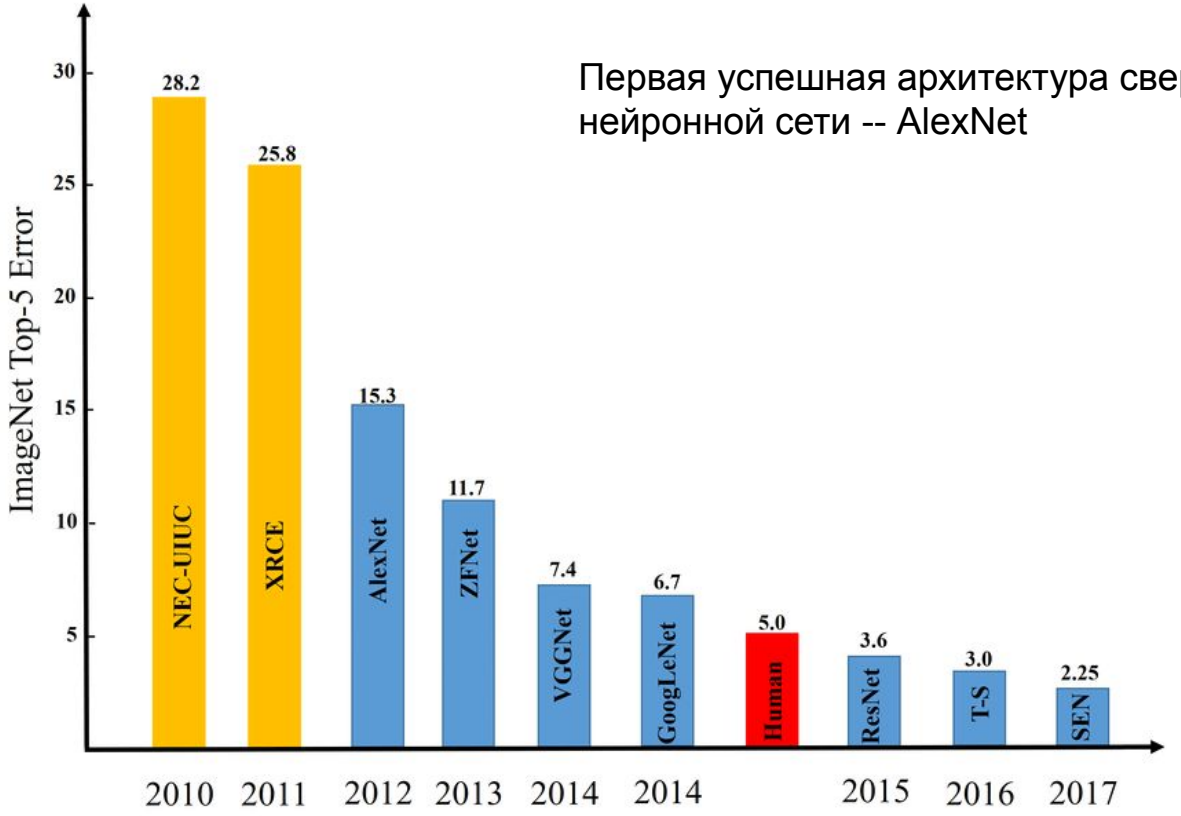
HOG (Histogram of Oriented Gradients)

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.

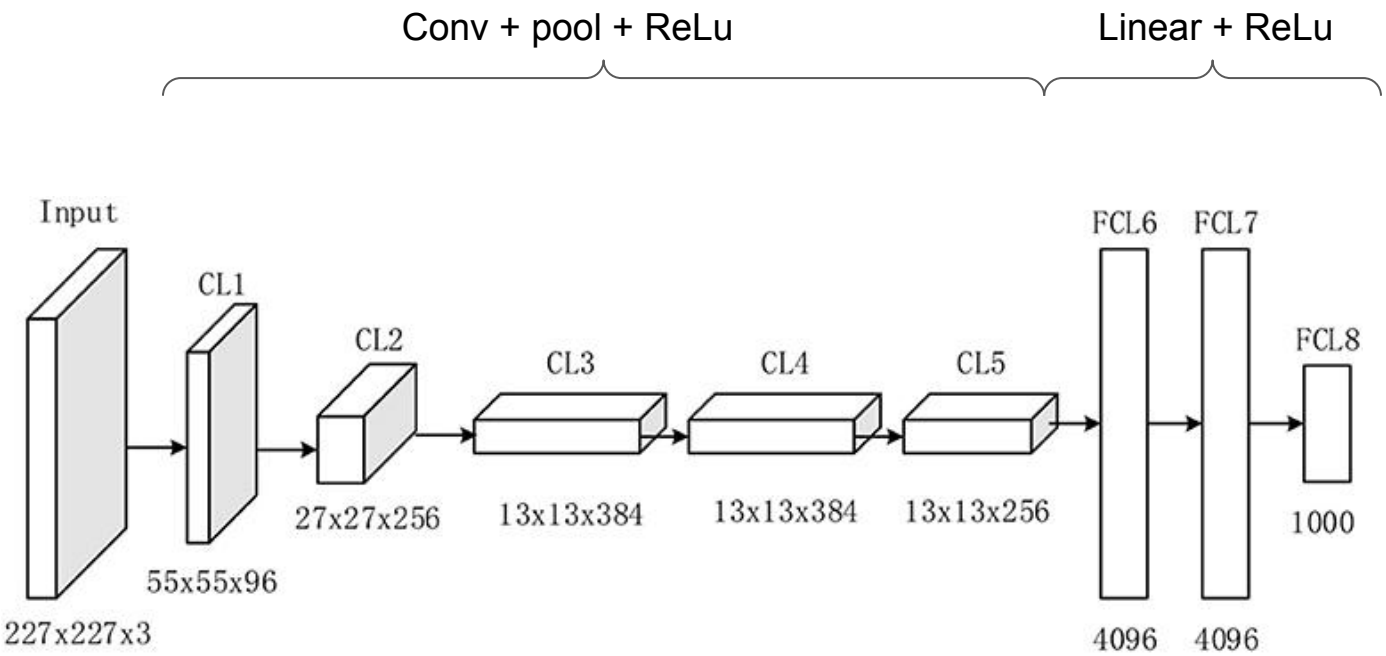
Далее на полученных фичах обучим классификатор изображений.

ImageNet Timeline



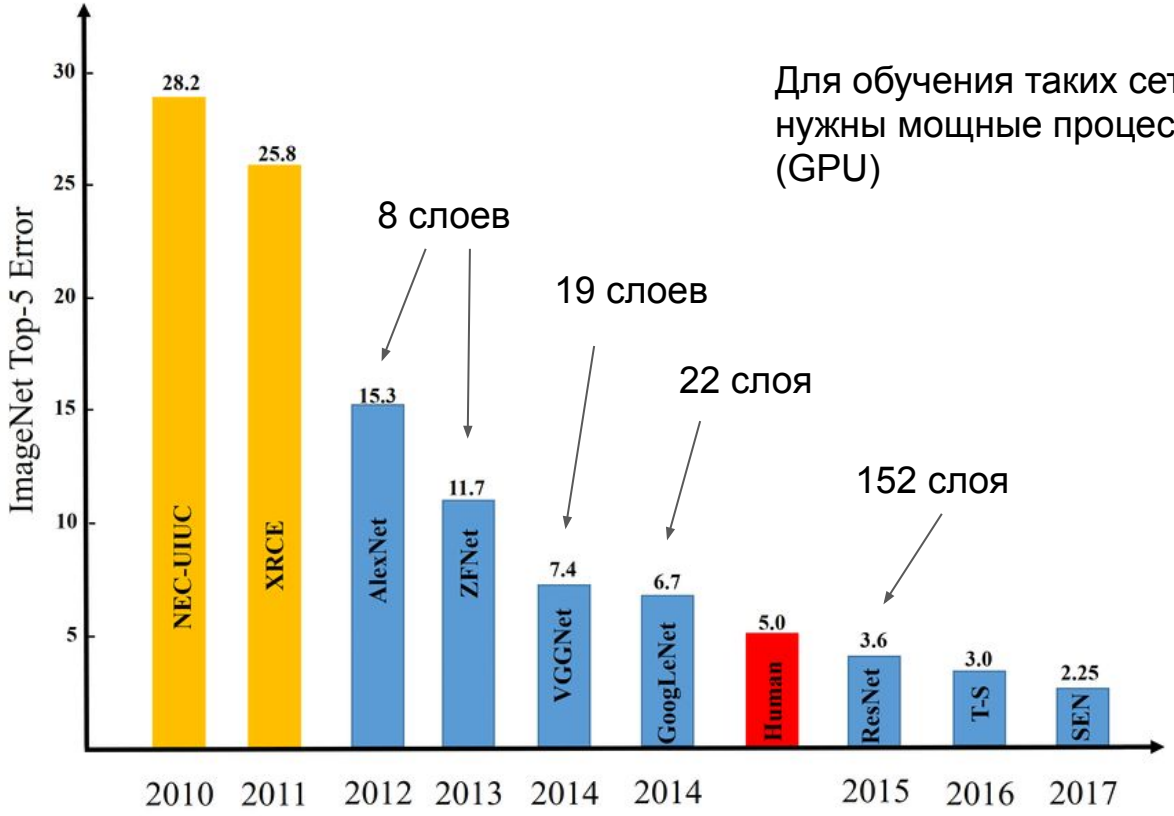
Первая успешная архитектура сверточной нейронной сети -- AlexNet

AlexNet



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

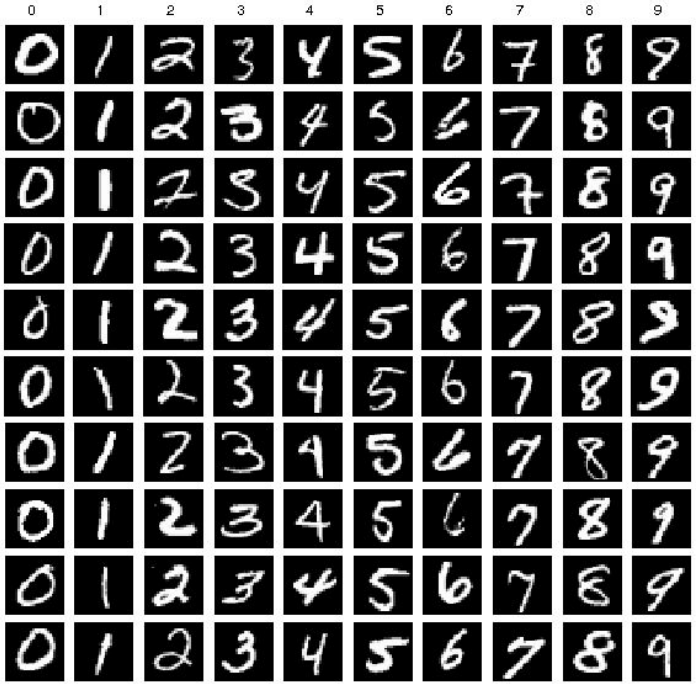
ImageNet Timeline



Для обучения таких сетей нужны мощные процессоры! (GPU)

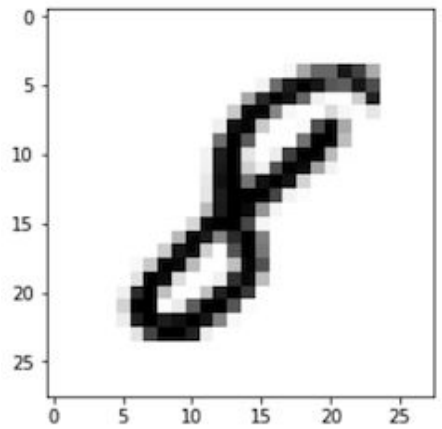
Сверточные нейронные сети

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



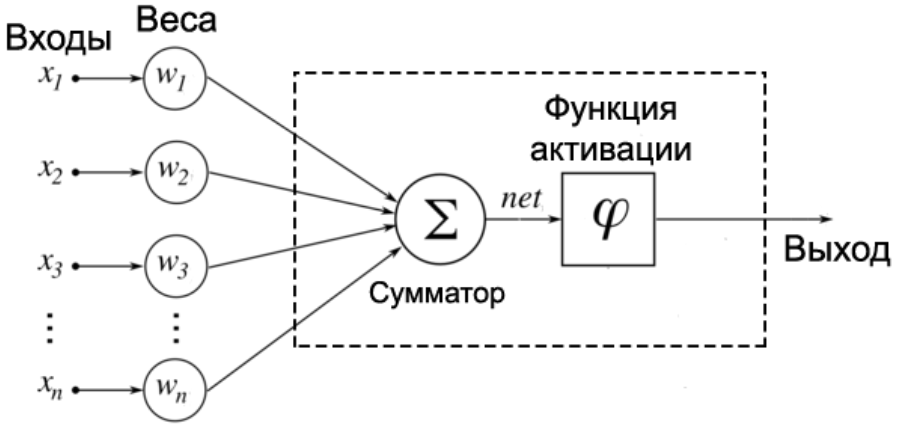
=

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	1	12	0	11	39	137	37	0	152	147	84	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	16	9	9	158	251	45	21	184	159	154	255	233	40	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	3	0	4	15	236	216	0	0	38	109	247	240	169	0	11	0	0	0
1	0	2	0	0	0	253	253	23	62	224	241	255	164	0	5	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	2	1	4	0	21	255	253	251	255	172	31	8	0	1	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	21	162	255	255	254	255	126	6	0	10	14	6	0	0	9	0	0	0
3	79	242	255	141	66	255	245	189	7	8	0	0	5	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
125	255	141	0	87	244	255	208	3	0	0	13	0	1	0	1	0	0	0	0
145	248	228	116	235	255	141	34	0	11	0	1	0	0	0	1	3	0	0	0
85	237	253	246	255	210	21	1	0	1	0	0	6	2	4	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

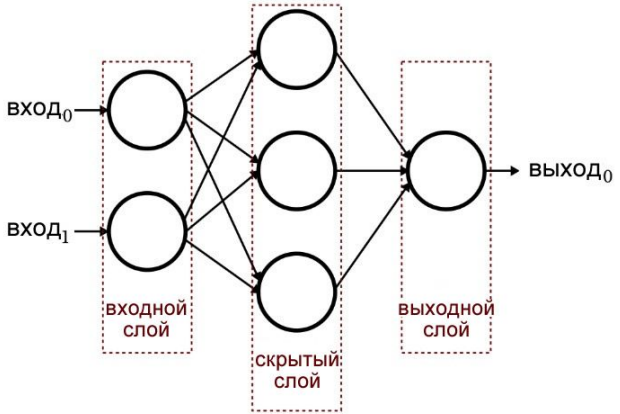
32

32

Перцептрон



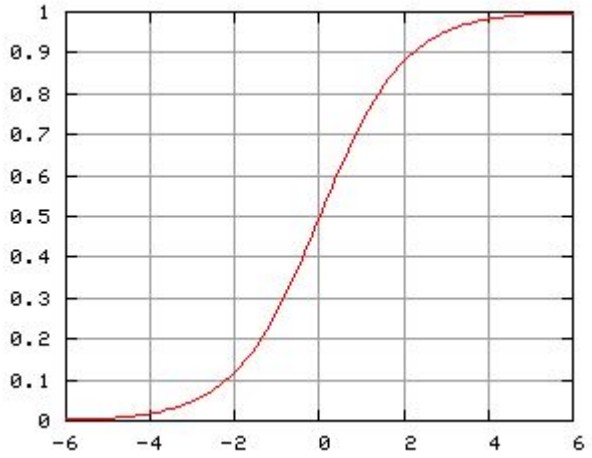
Полносвязная нейронная сеть



$$w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w}$$

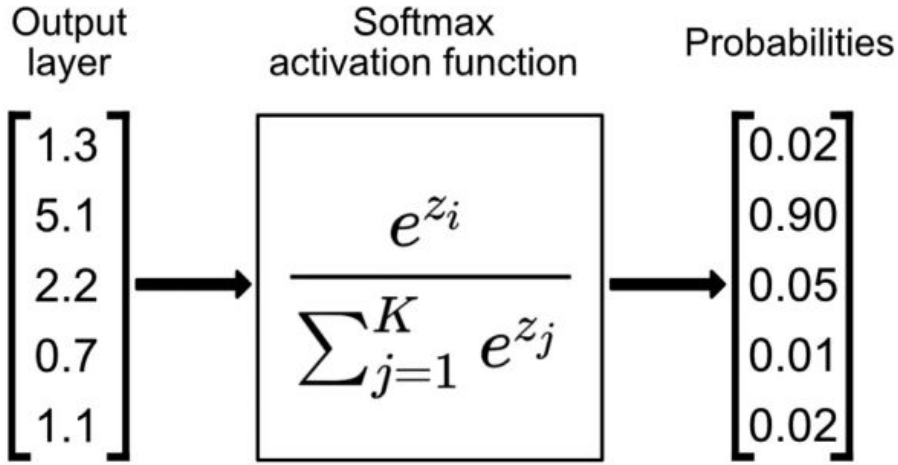
Updated weight — Current weight — Learning Rate — Gradient

Sigmoid

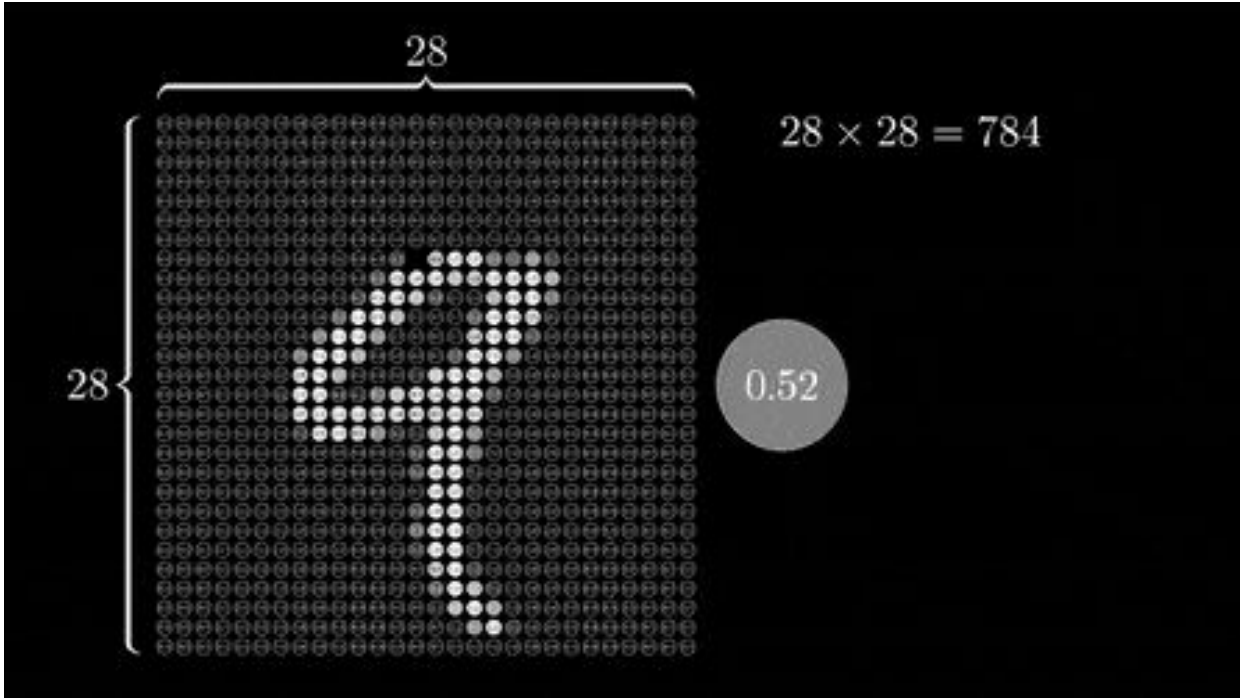


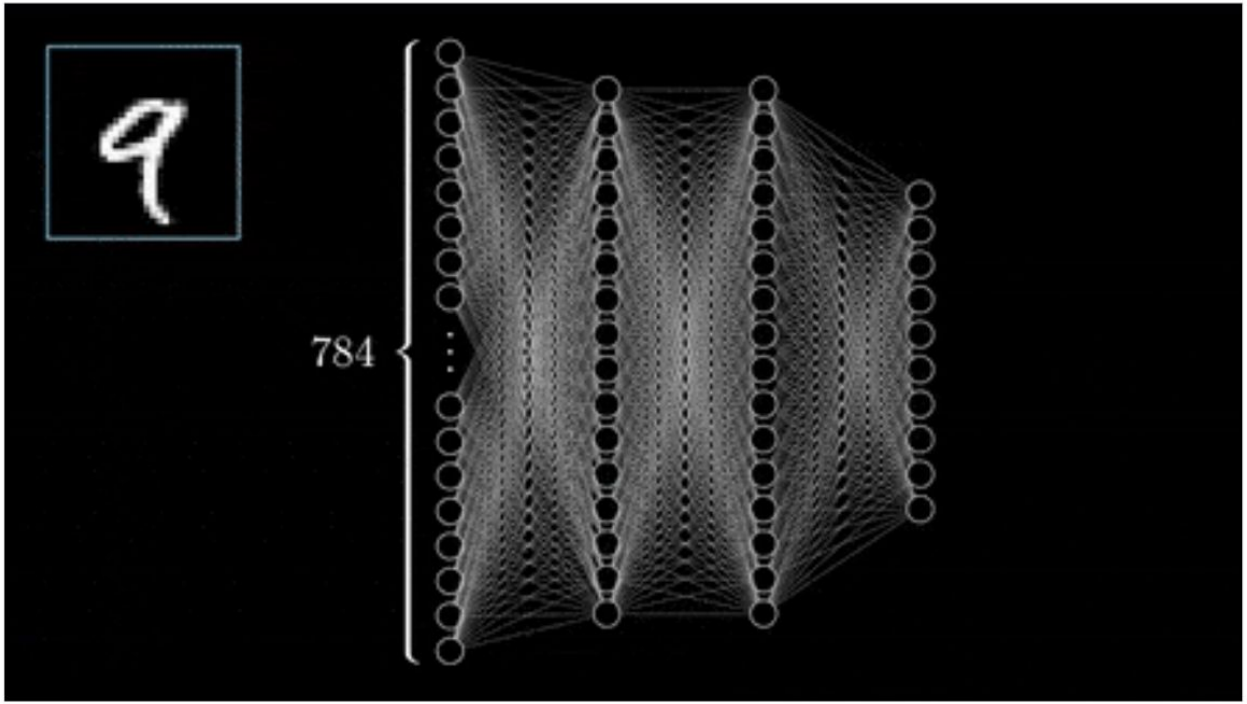
$$S(x) = \frac{1}{1 + e^{-x}}$$

SoftMax



Растягивание картинки в вектор и подача на вход сети




784

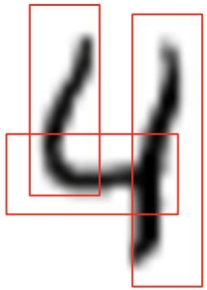
Классификация картинок полносвязной сетью:

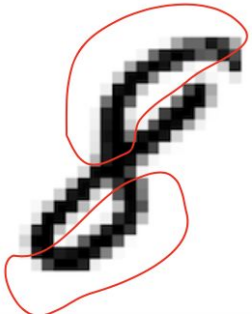
Недостатки:

- 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?

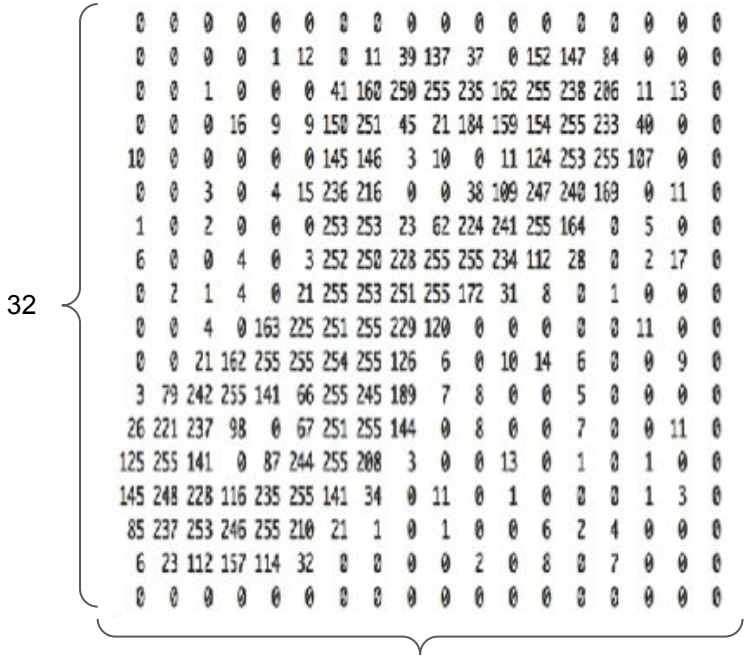




У четверки преимущественно горизонтальные и вертикальные линии,
у восьмерки линии плавные

Свертка

Изображение



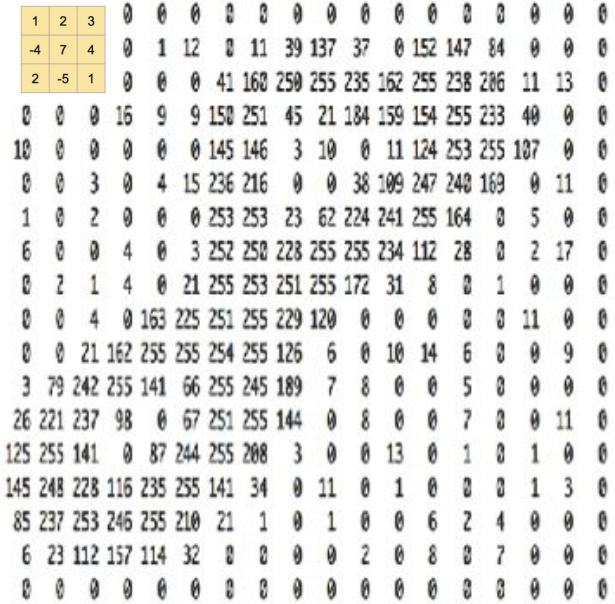
Ядро (фильтр)

1	2	3
-4	7	4
2	-5	1

0 0 0
0 0 0
0 0 1

1□0 + 2□0 + 3□0 + (-4)□0 + 7□0 + 4□0 + 2□0 + (-5)□0 +
1□1 = 1

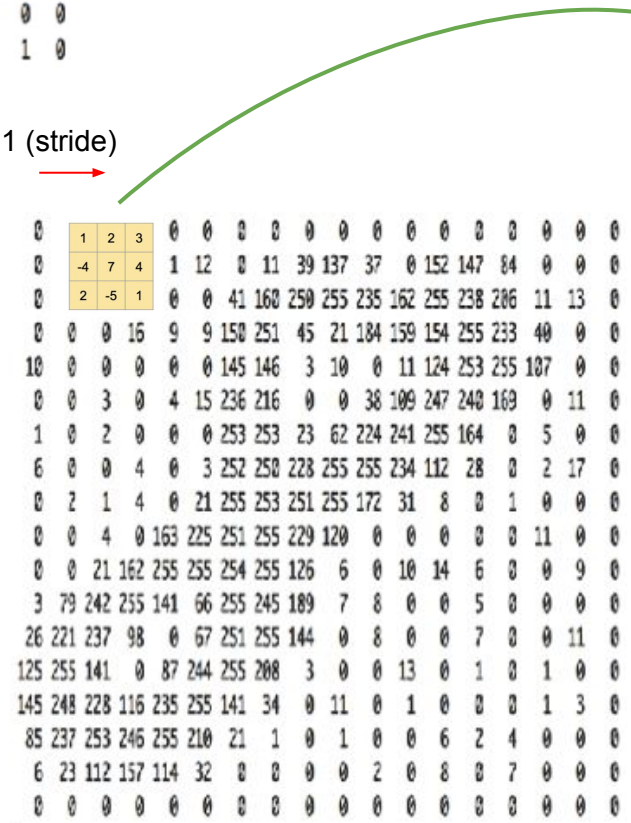
1	2	3
-4	7	4
2	-5	1



1

0 0 0
0 0 0
0 1 0

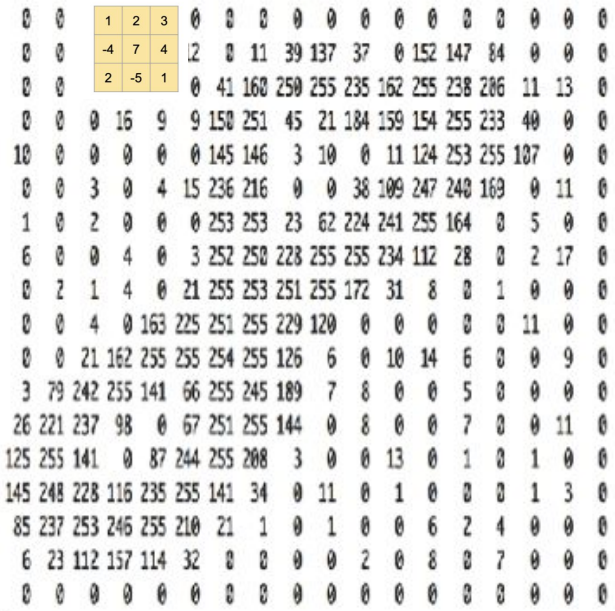
1 (stride)



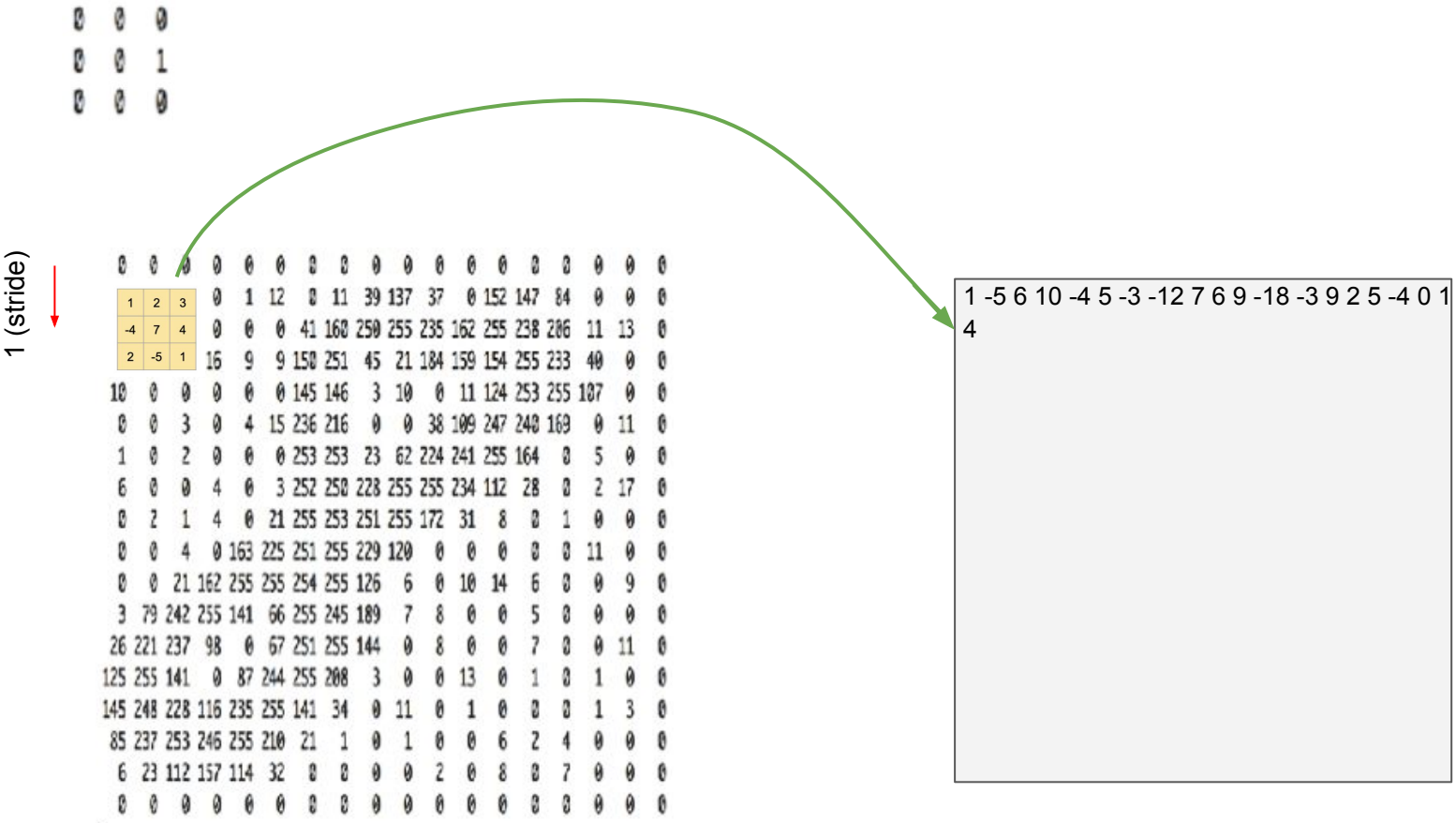
1 -5

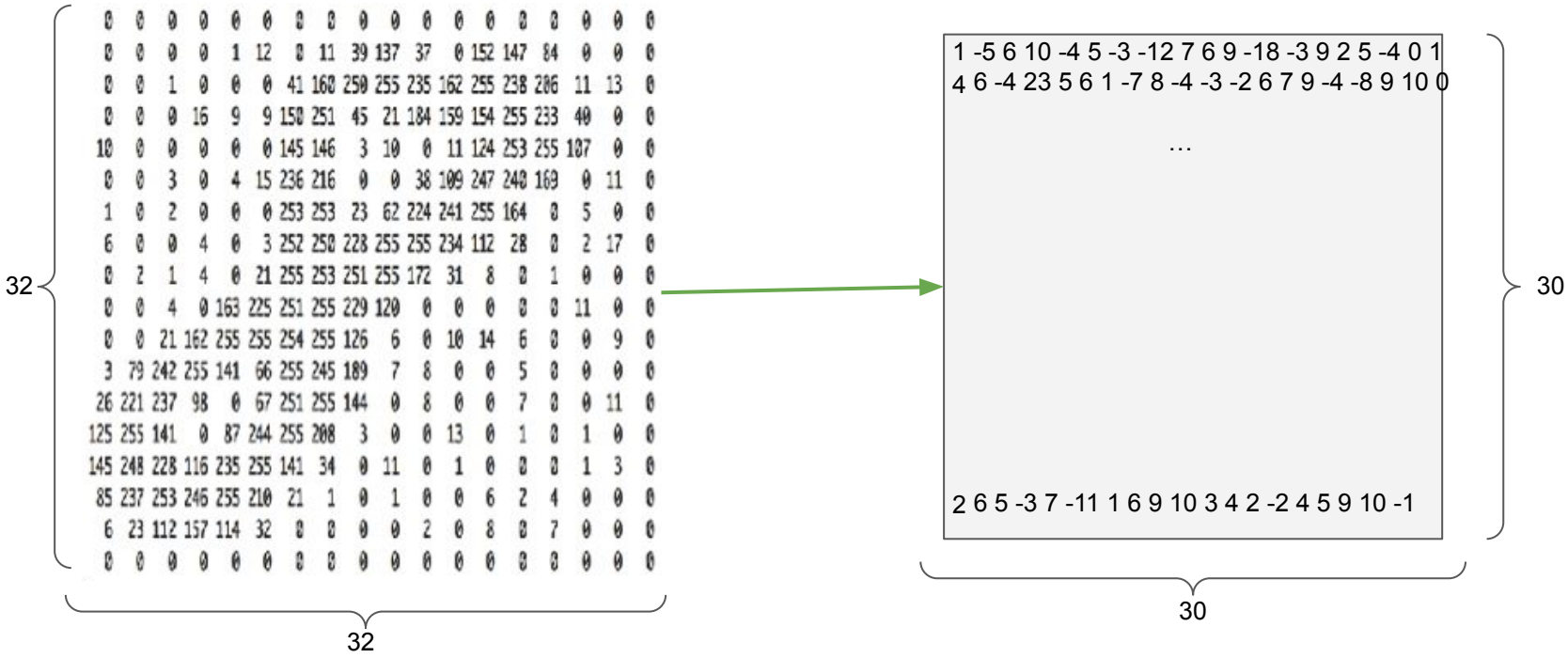
0 0 0
0 0 1
1 0 0

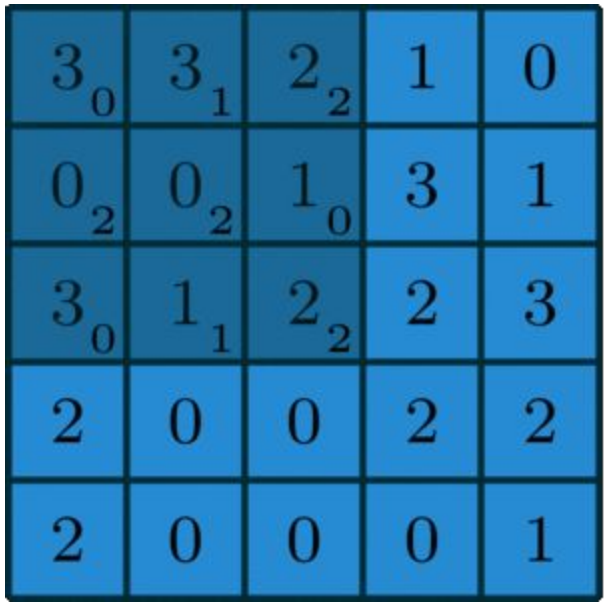
1 (stride)



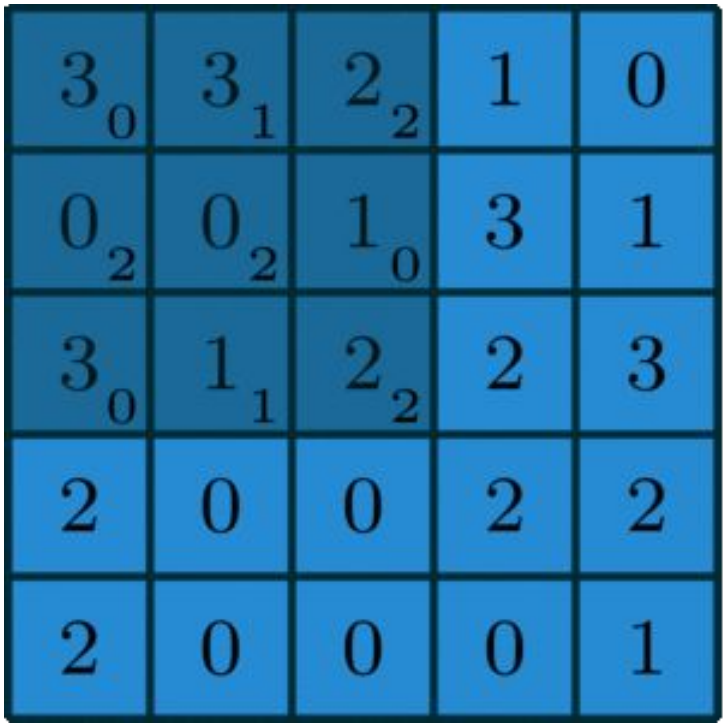
1 -5 6


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
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 1 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
32
32
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


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

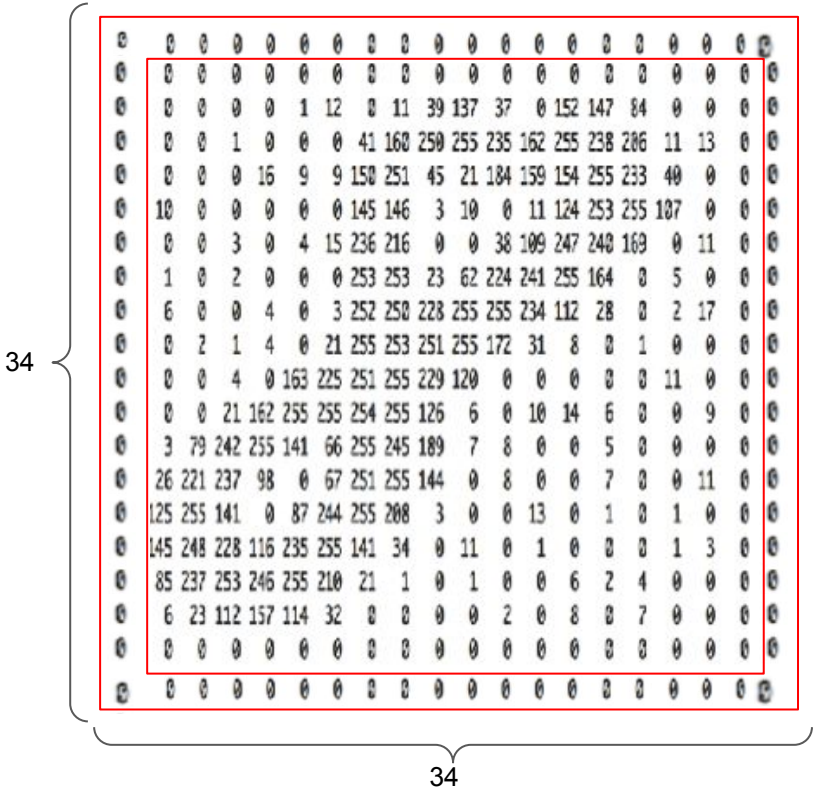

12.0 12.0 17.0
10.0 17.0 19.0
9.0 6.0 14.0


3_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

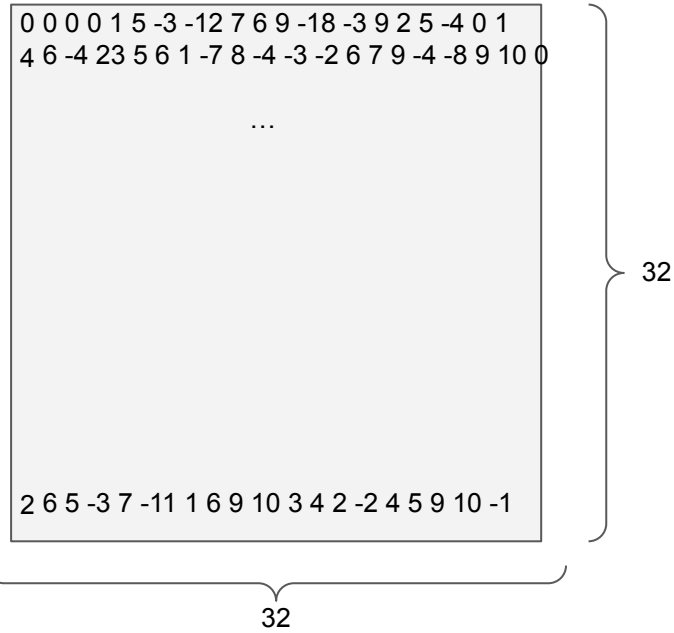

12.0 12.0 17.0
10.0 17.0 19.0
9.0 6.0 14.0

Padding

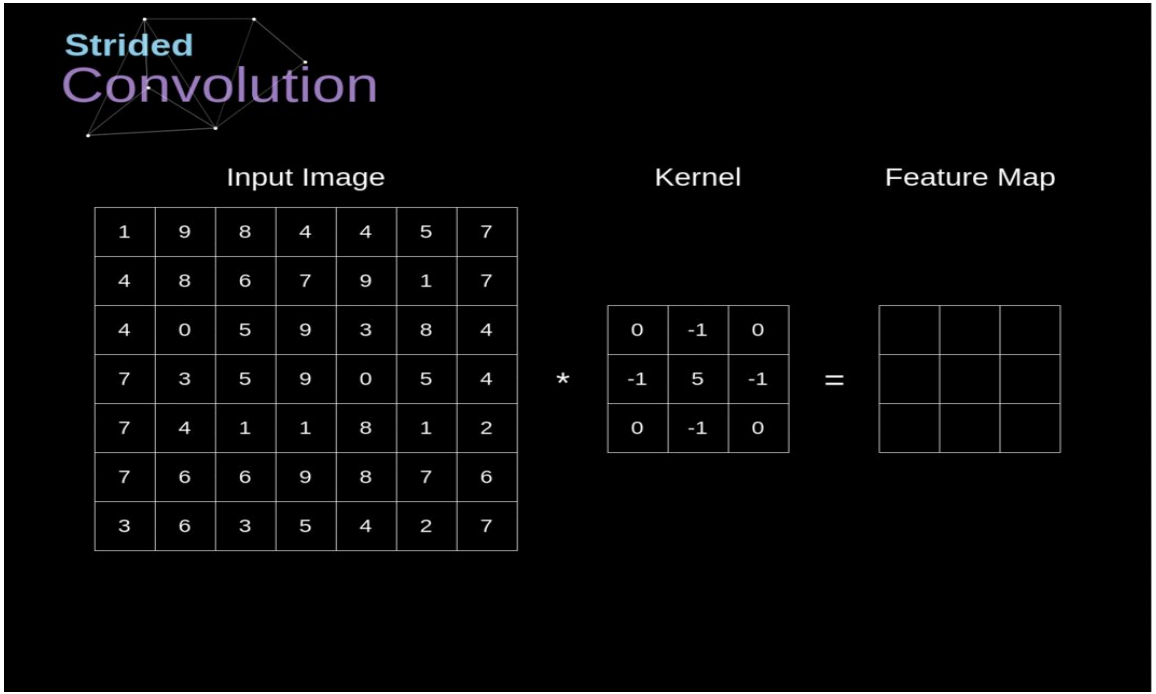
Используется для манипуляции размерами карт активаций



Карта активации



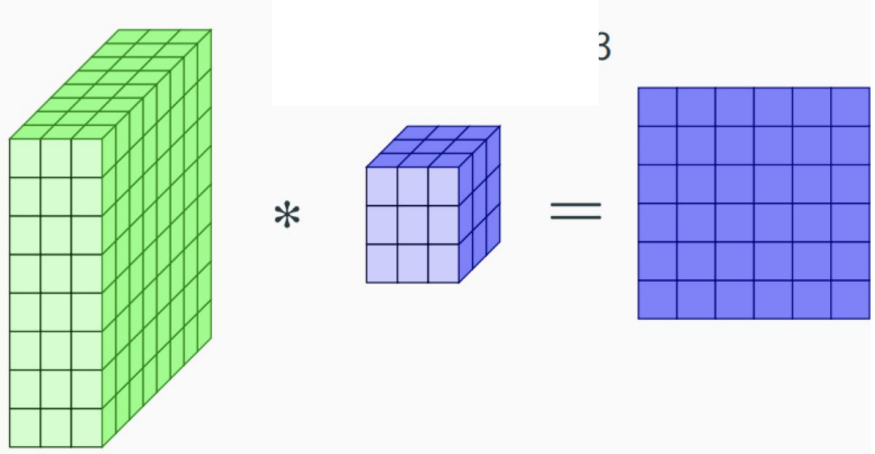
Исходное изображение Ядро Карта активации



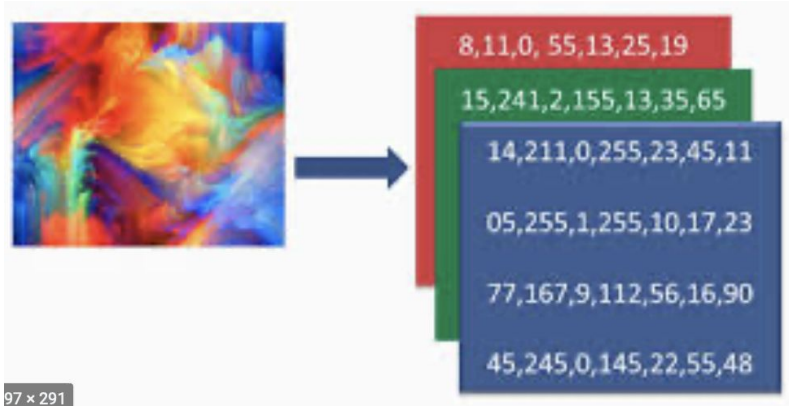
Свертка цветных изображений

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

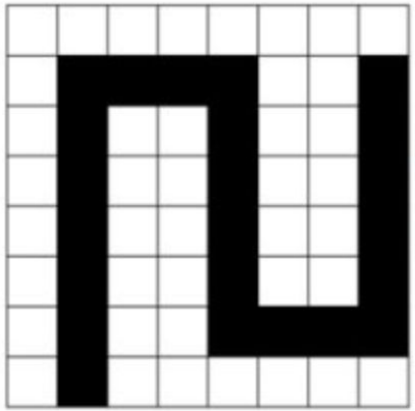
-1	-1	2
-1	2	-1
2	-1	-1

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

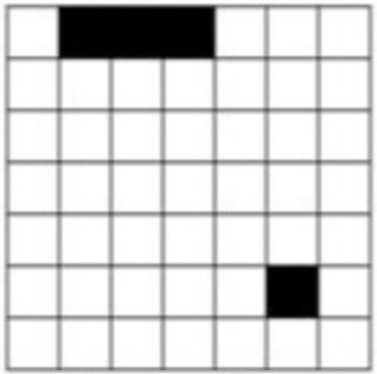
Как мог бы выглядеть фильтр, реагирующий на горизонтальные линии



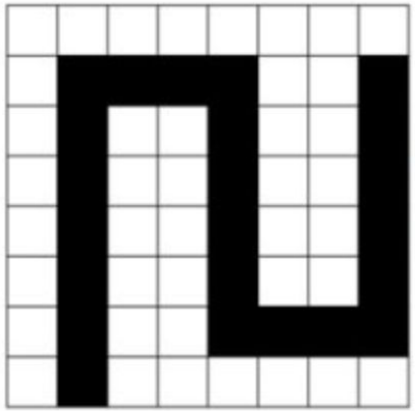
+

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

=



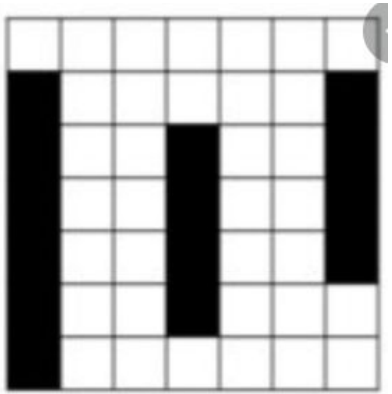
Как мог бы выглядеть фильтр, реагирующий на вертикальные линии



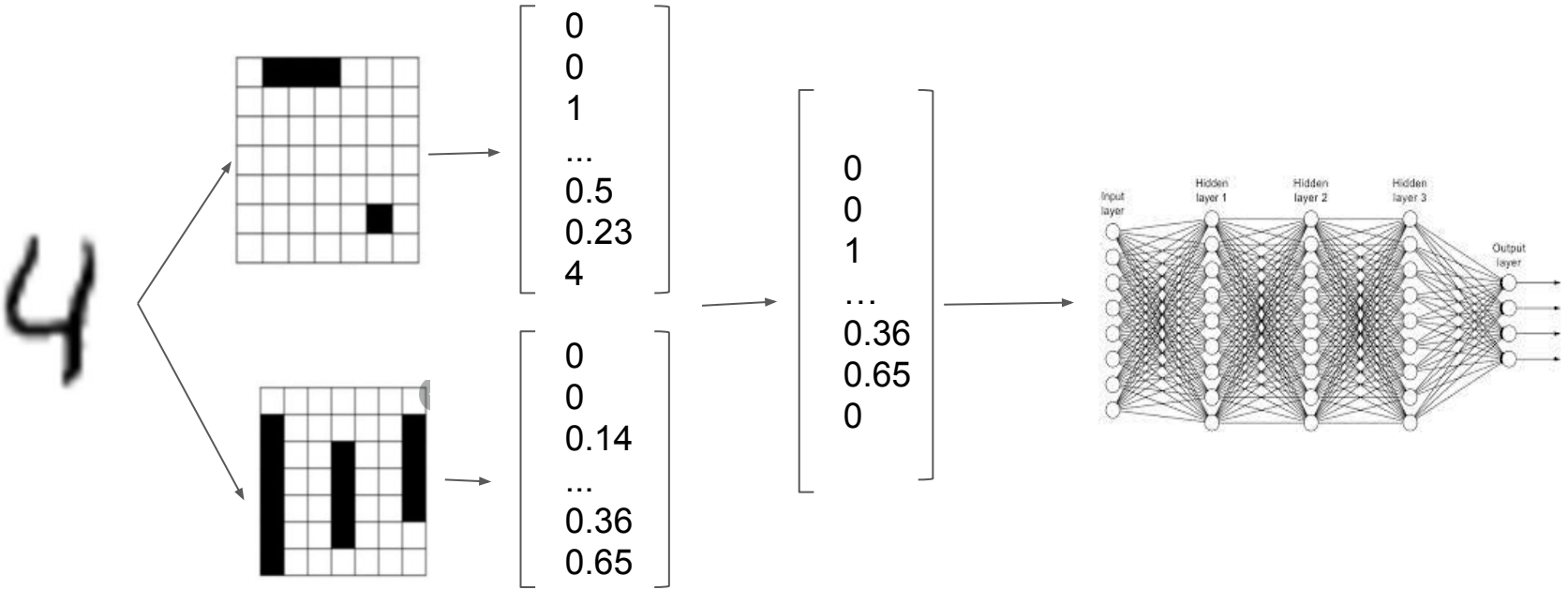
+

-1	-2	-1
0	0	0
-1	-2	-1

=

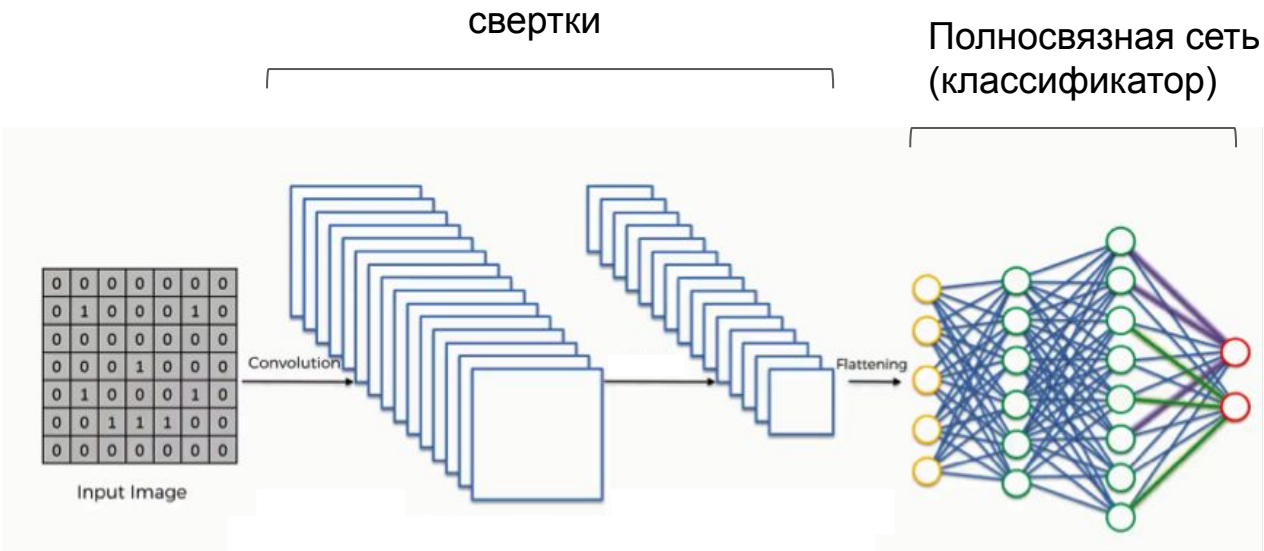


После получения карт активаций, мы **развернем все карты в векторы, сконкатенируем** и подадим на вход полносвязной сети



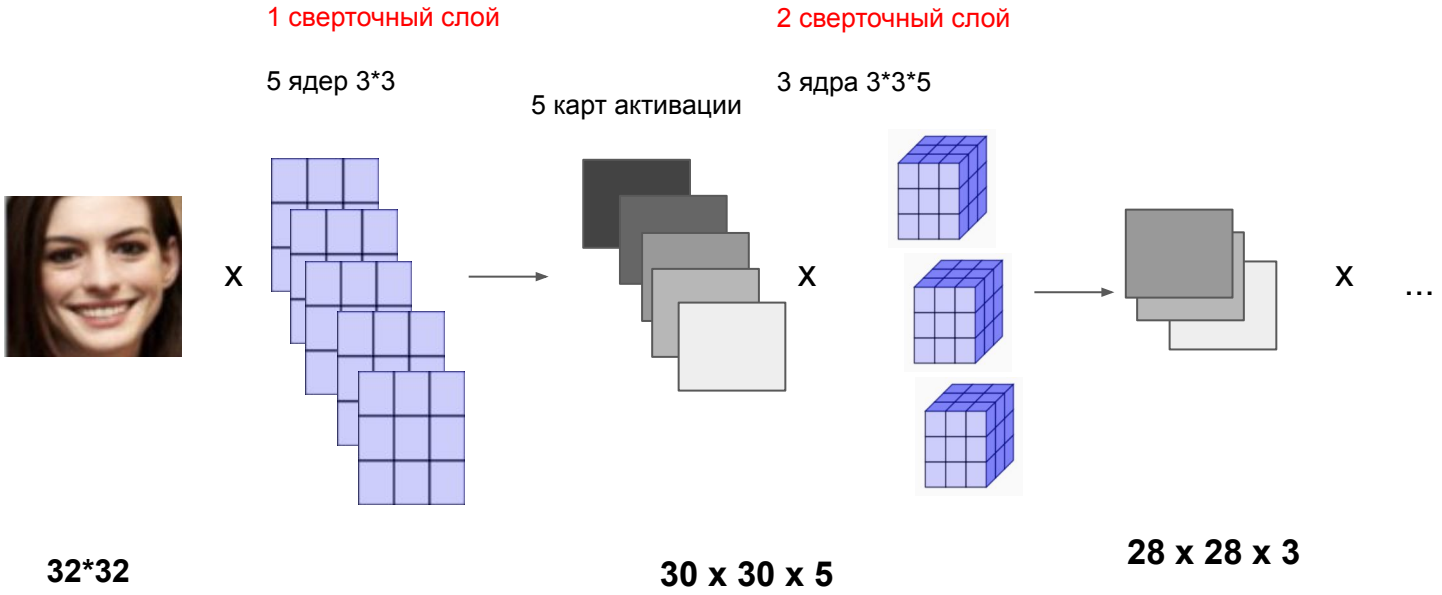
свертки flatten конкатенация подача в полносвязную сеть

Сверточная нейросеть



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

Потребуется несколько слоев сверток

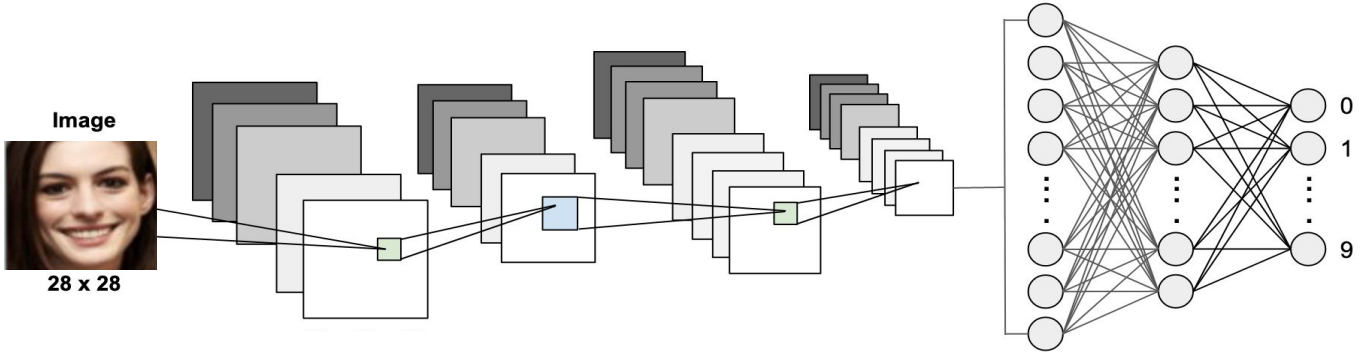


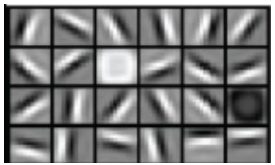
1 слой сверток

2 слой сверток

3 слой сверток

4 слой сверток





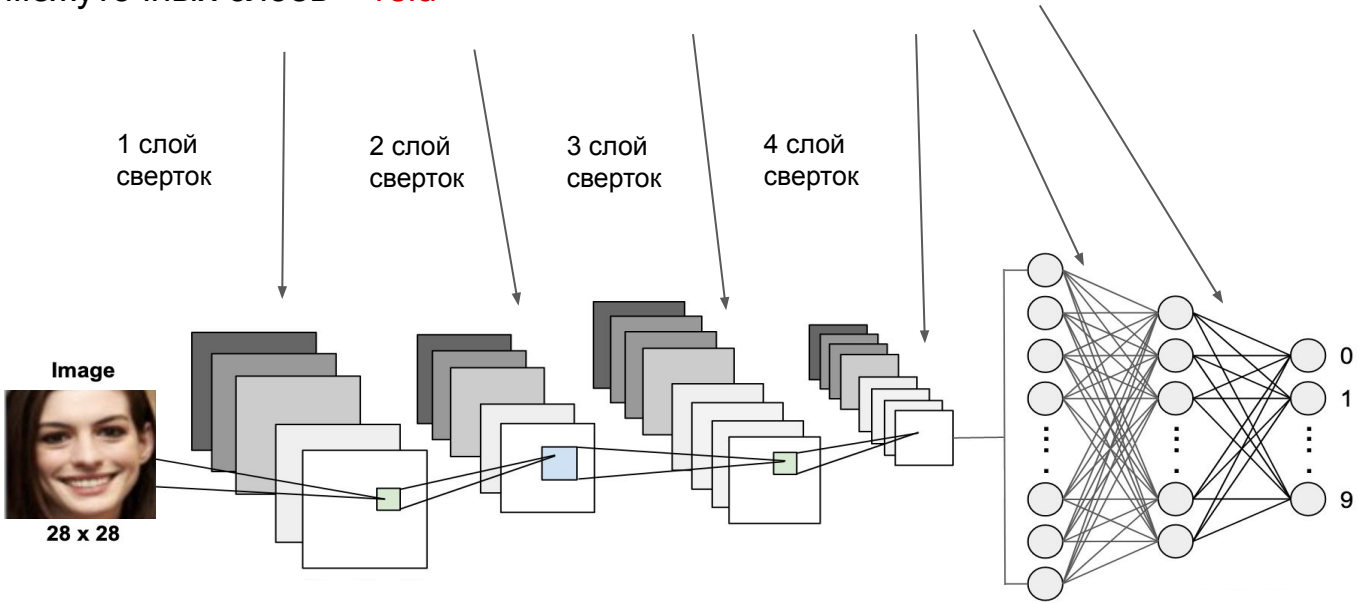
Низкоуровневые паттерны

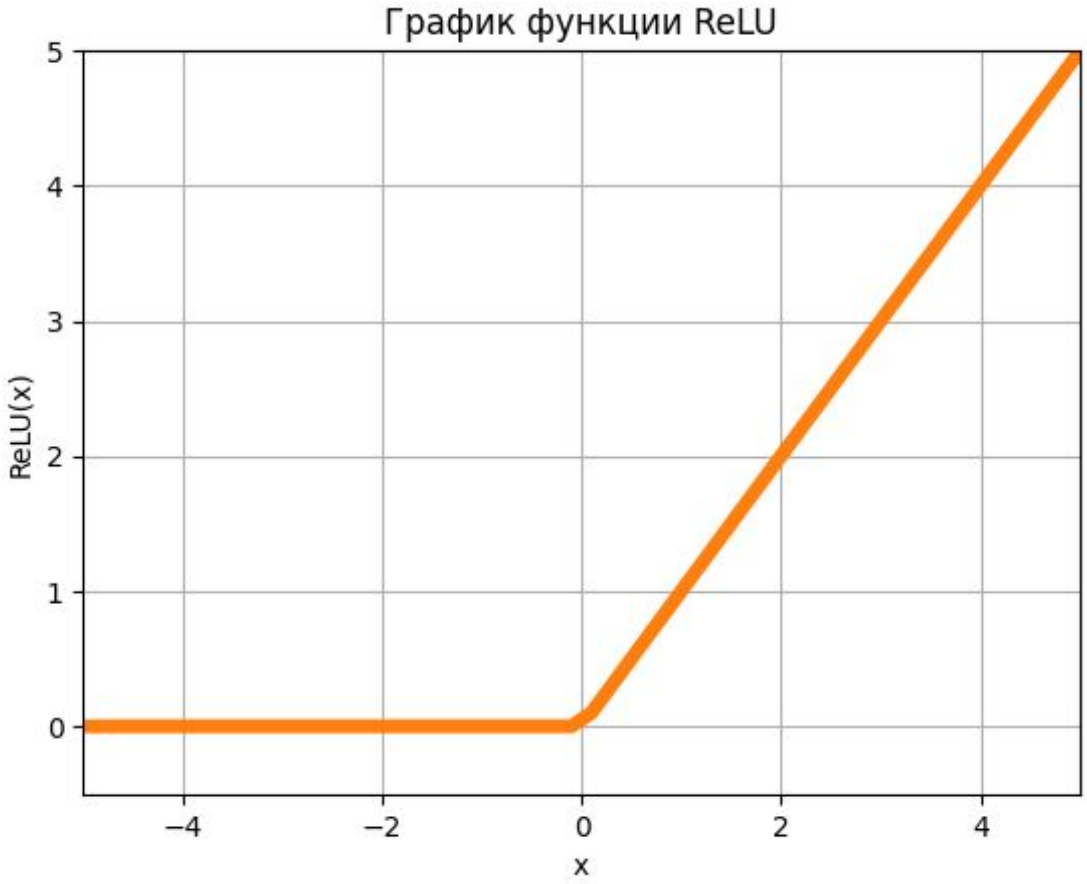
Вырисовываются отдельные части

Выделены признаки изображения, важные для задачи

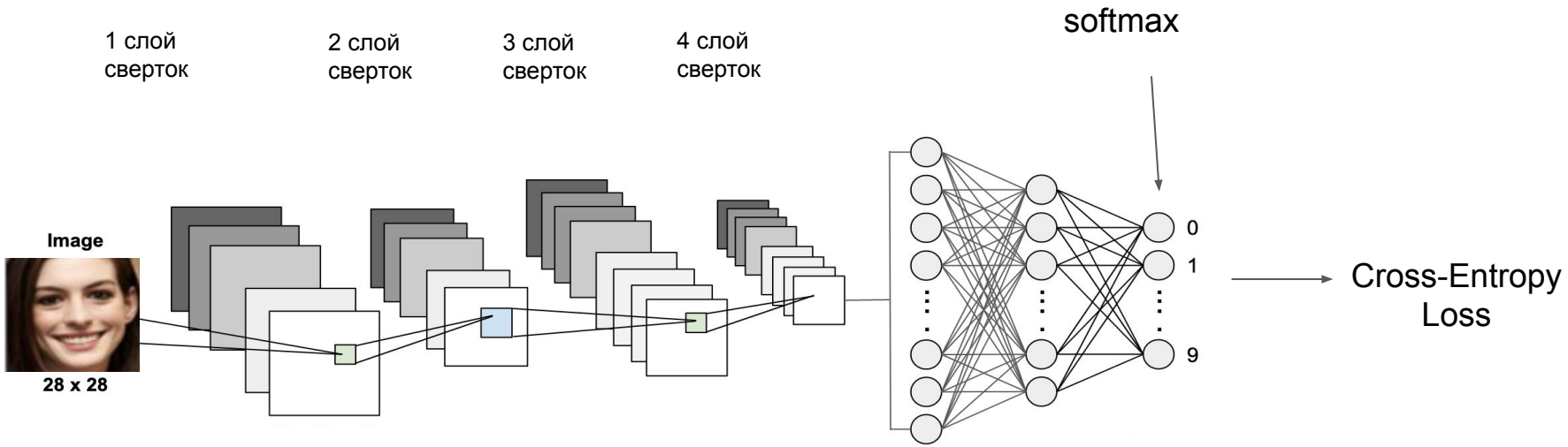
После сверточных слоев, как и после полносвязных, используется функция активации.

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



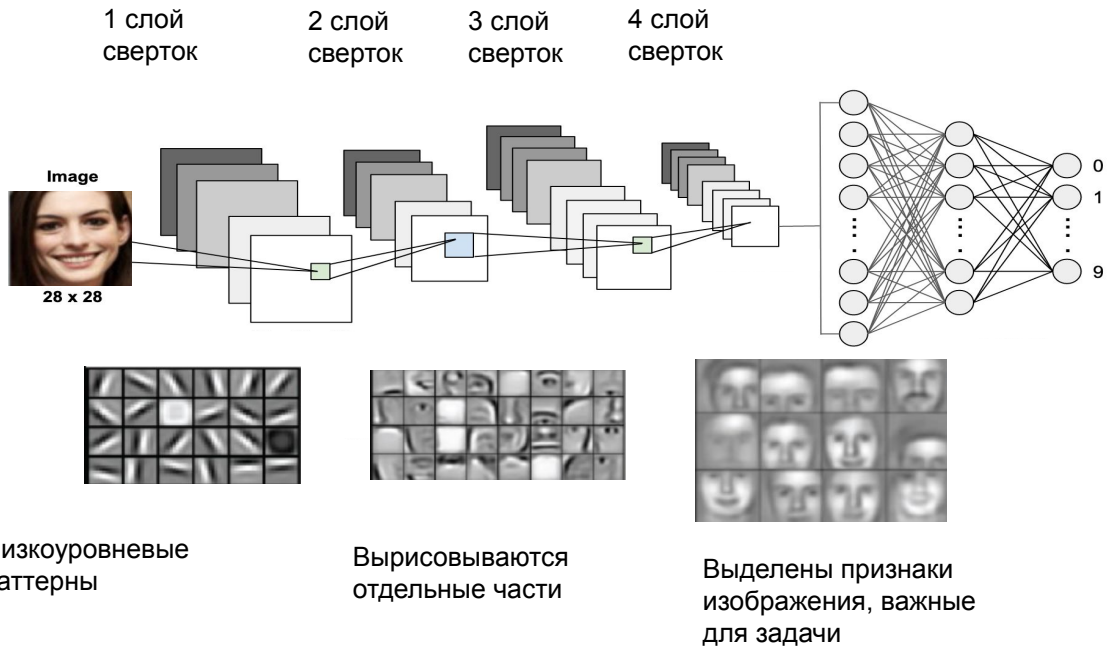


max(0,x)


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

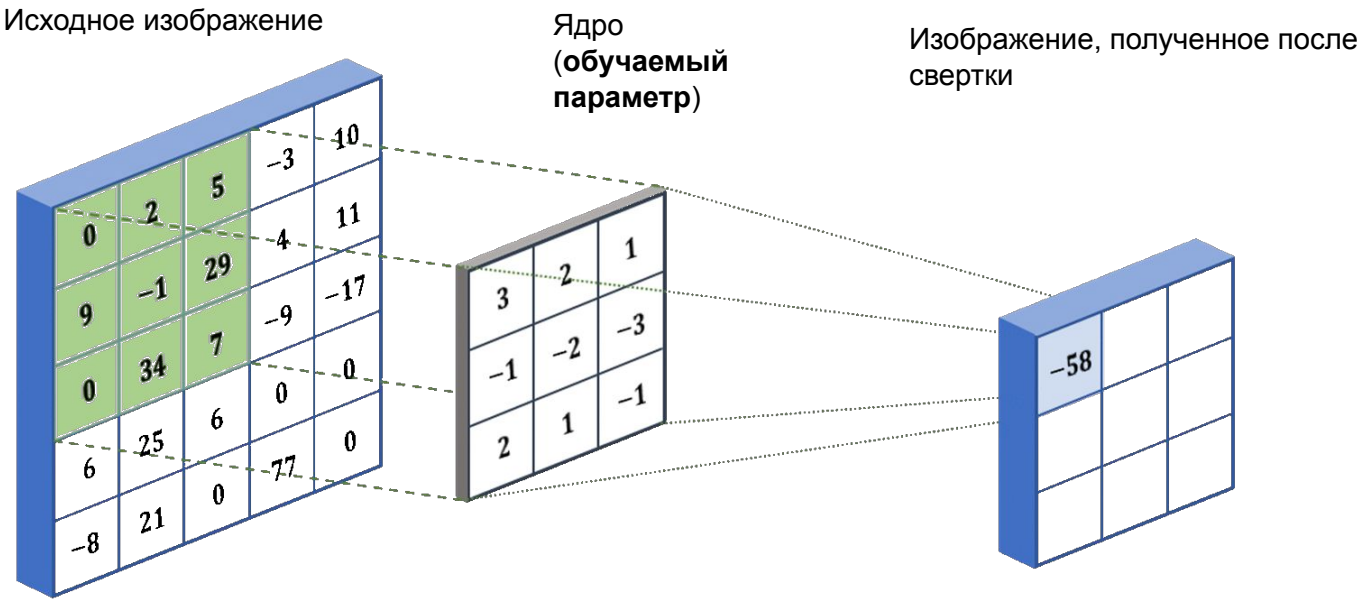
Свертки -- как лампочки, которые “загораются” сильнее, если на изображении есть определенный паттерн

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.

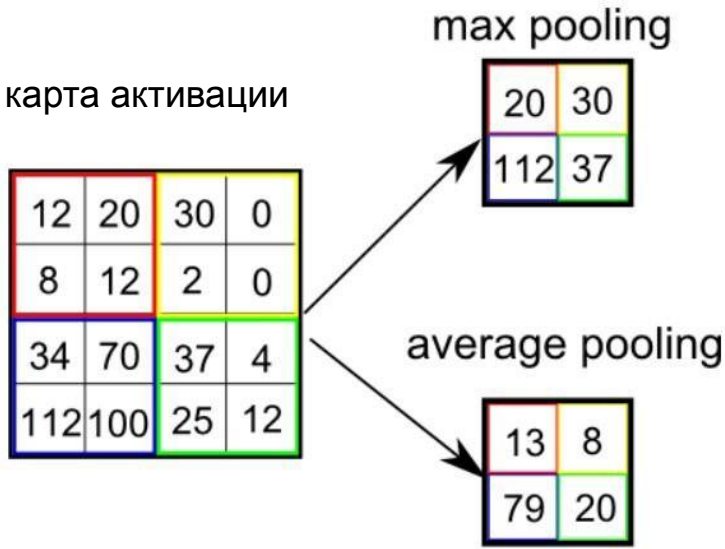


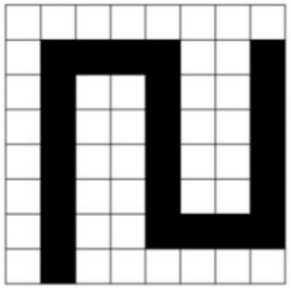
Pooling

Pooling

Техника уменьшения размерности (downsampling’a) карт активаций

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

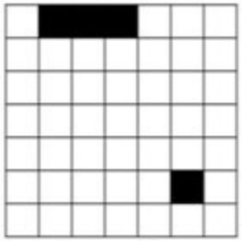


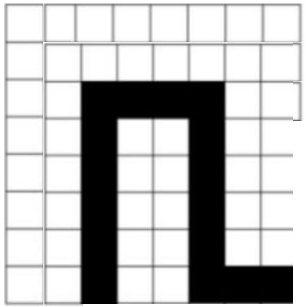


+

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

=

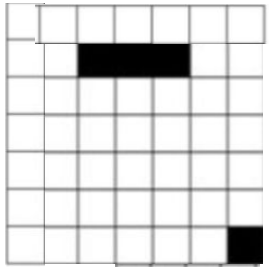




+

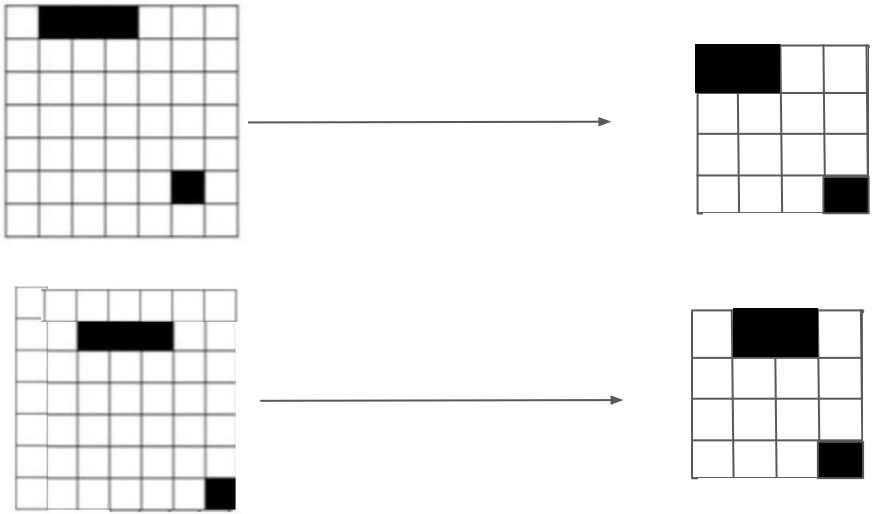
-1	0	1
-1	0	1
-1	0	1

=



Разница в расположении

Результат применения 2x2 MaxPooling’a к картам активаций:



Pooling layer

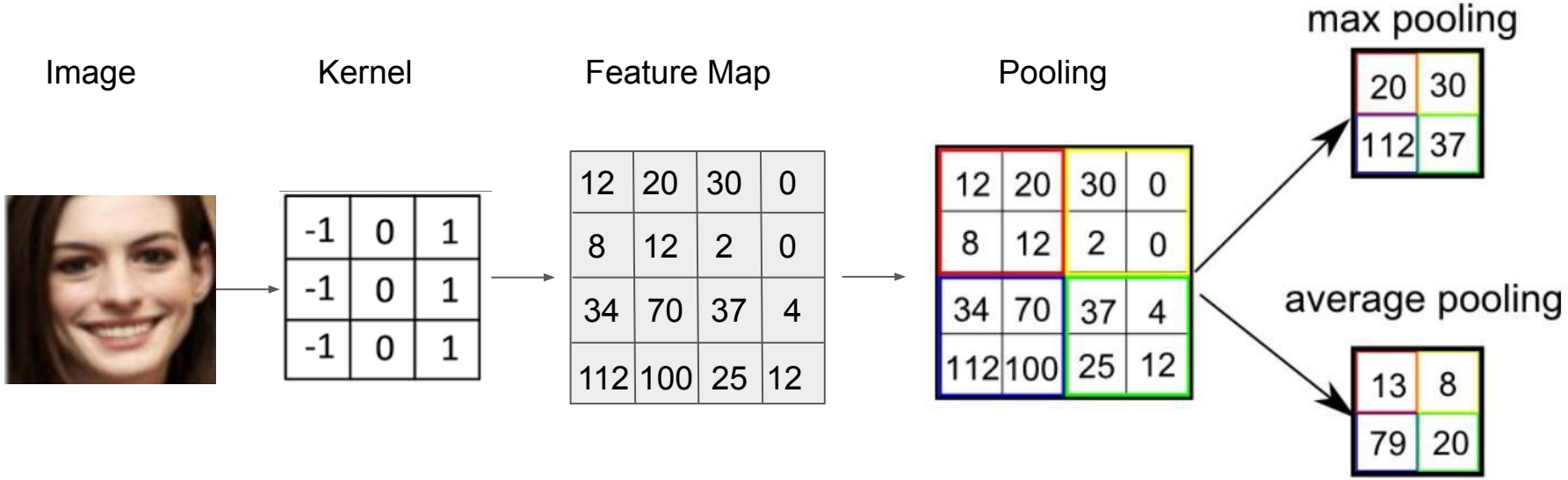
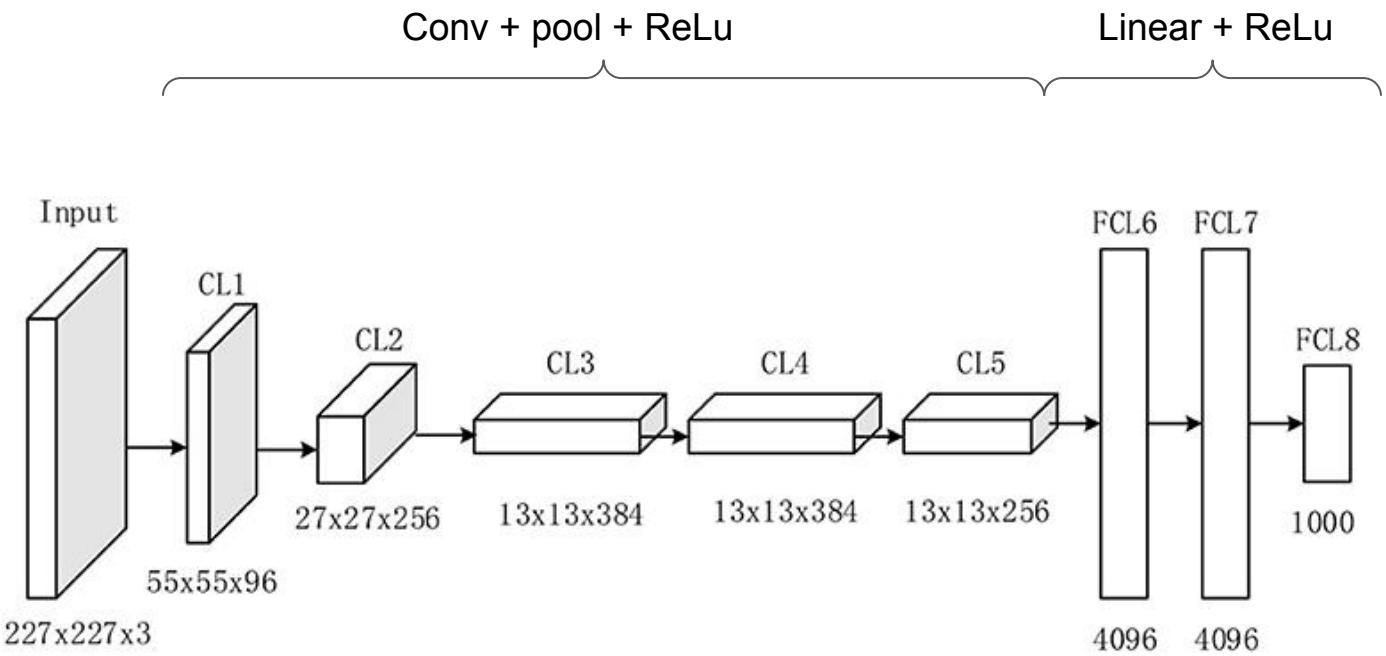
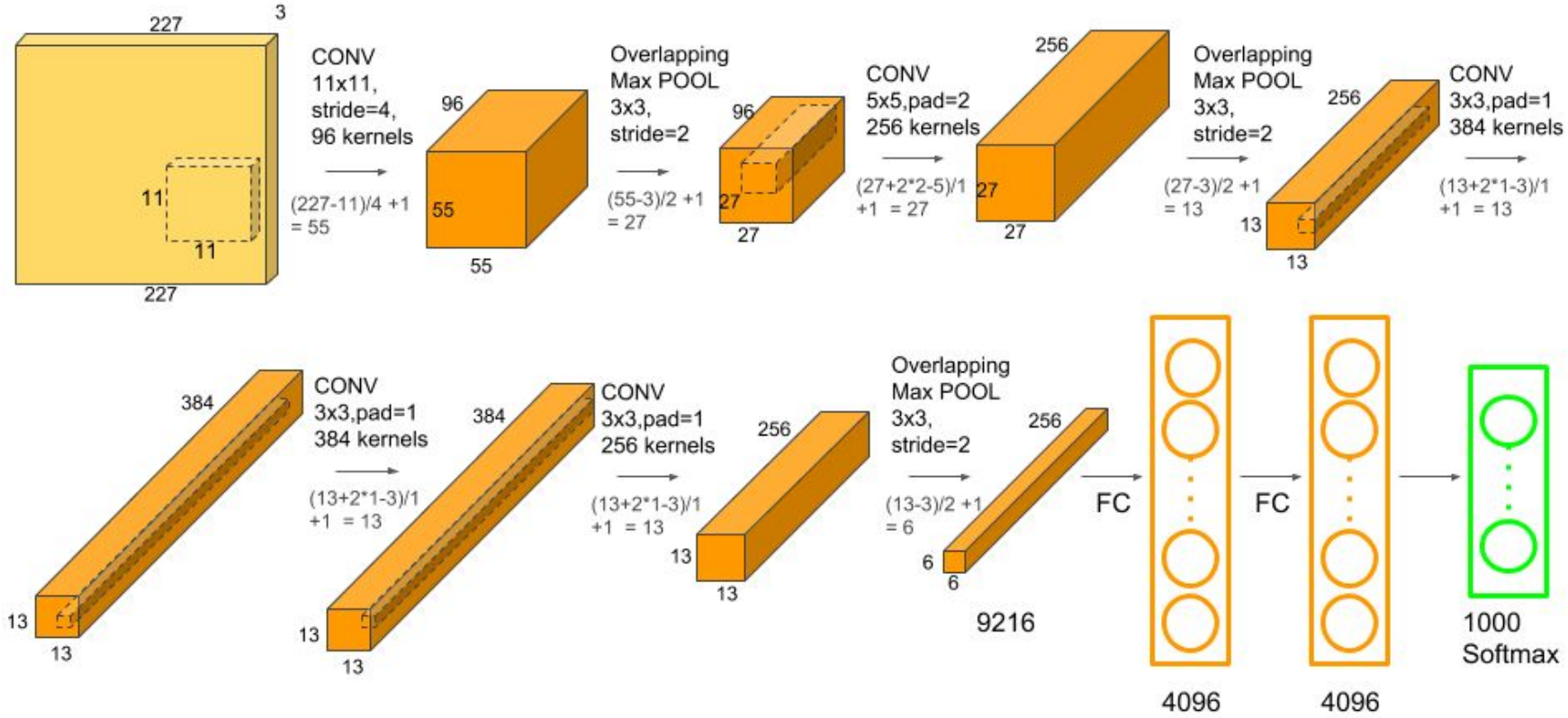


Image -> conv -> act -> pool -> conv ...

AlexNet



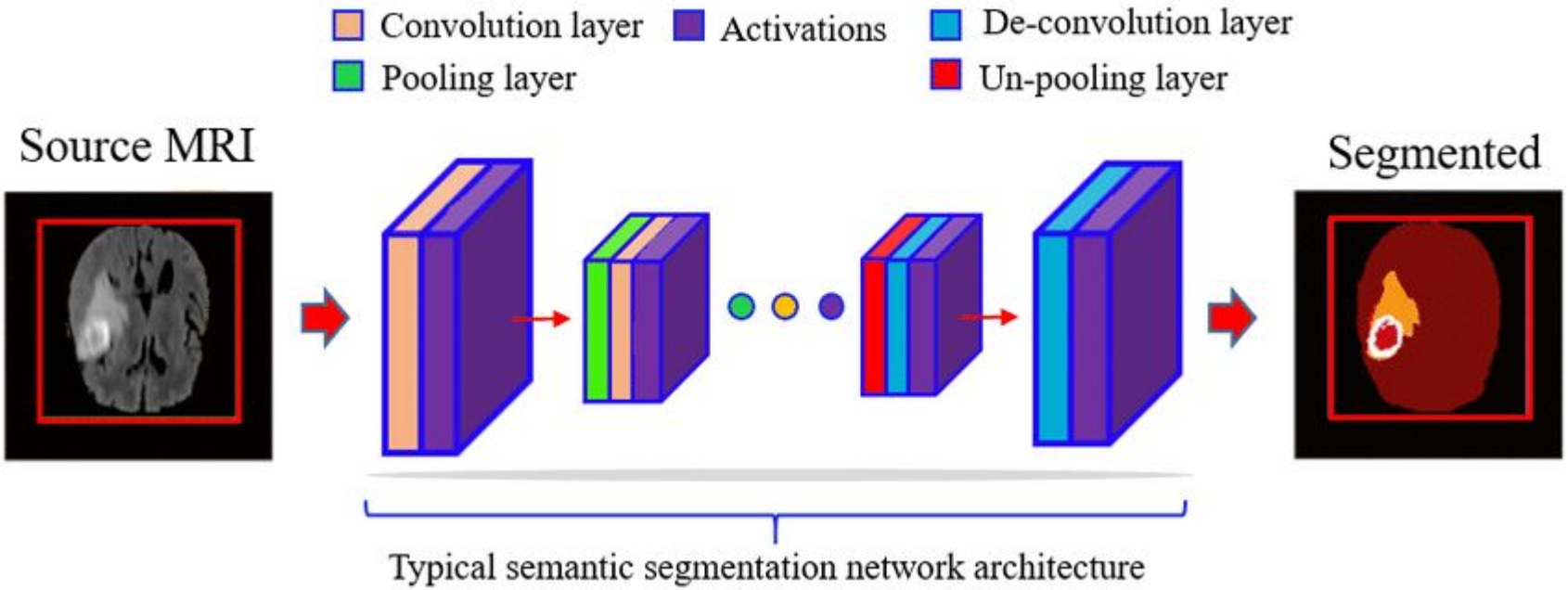
https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf


227
3
227
11
11
CONV
11x11,
stride=4,
96 kernels
(227-11)/4 +1
= 55
96
55
55
Overlapping
Max POOL
3x3,
stride=2
(55-3)/2 +1
= 27
96
27
27
CONV
5x5,pad=2
256 kernels
(27+2*2-5)/1
+1 = 27
256
27
27
Overlapping
Max POOL
3x3,
stride=2
(27-3)/2 +1
= 13
256
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
256 kernels
(13+2*1-3)/1
+1 = 13
256
13
13
Overlapping
Max POOL
3x3,
stride=2
(13-3)/2 +1
= 6
256
6
6
9216
FC
4096
FC
4096
1000
Softmax

Задачи компьютерного зрения (CV)

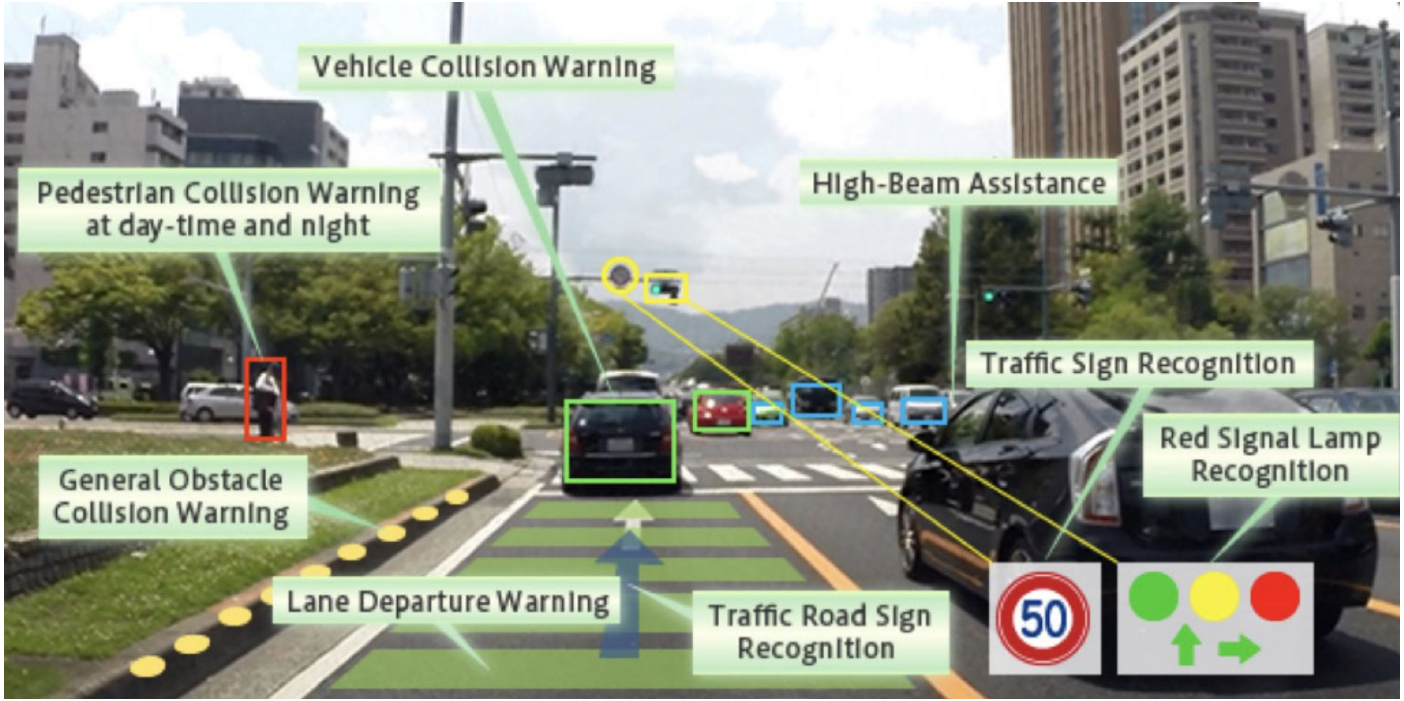
Классификация, детекция, сегментация

Сегментация — важная задача для медицины



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

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



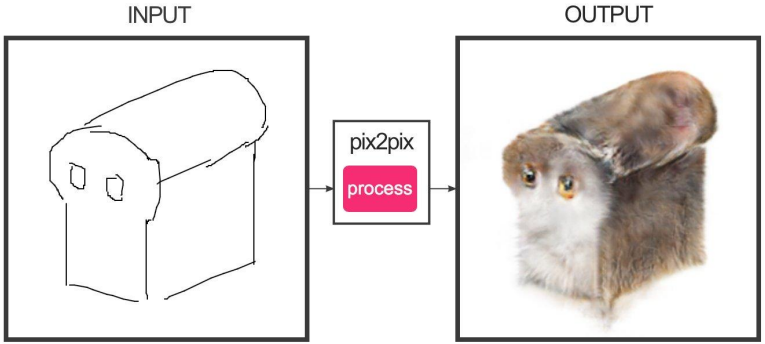
Optical Character Recognition

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

GANs

Style transfer

Image generation/completion



Приятного продолжения изучения!