МИСИС
УНИВЕРСИТЕТ

Свёрточные нейронные сети

Convolution Neural Network

Корнет Мария Евгеньевна
старший преподаватель кафедры
Инженерная кибернетика

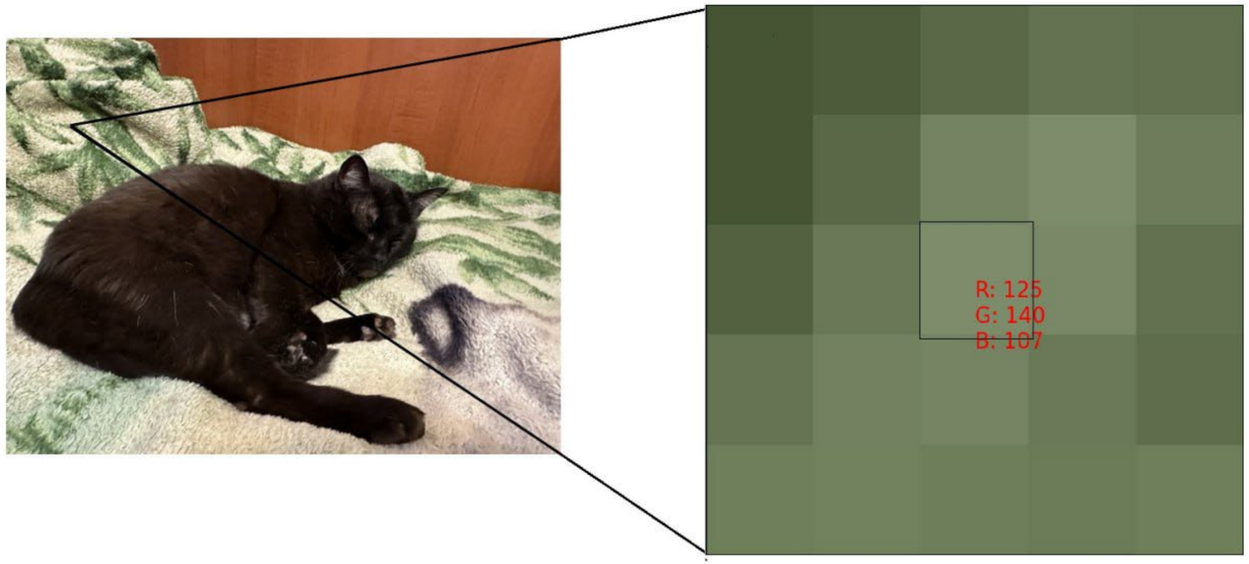
План

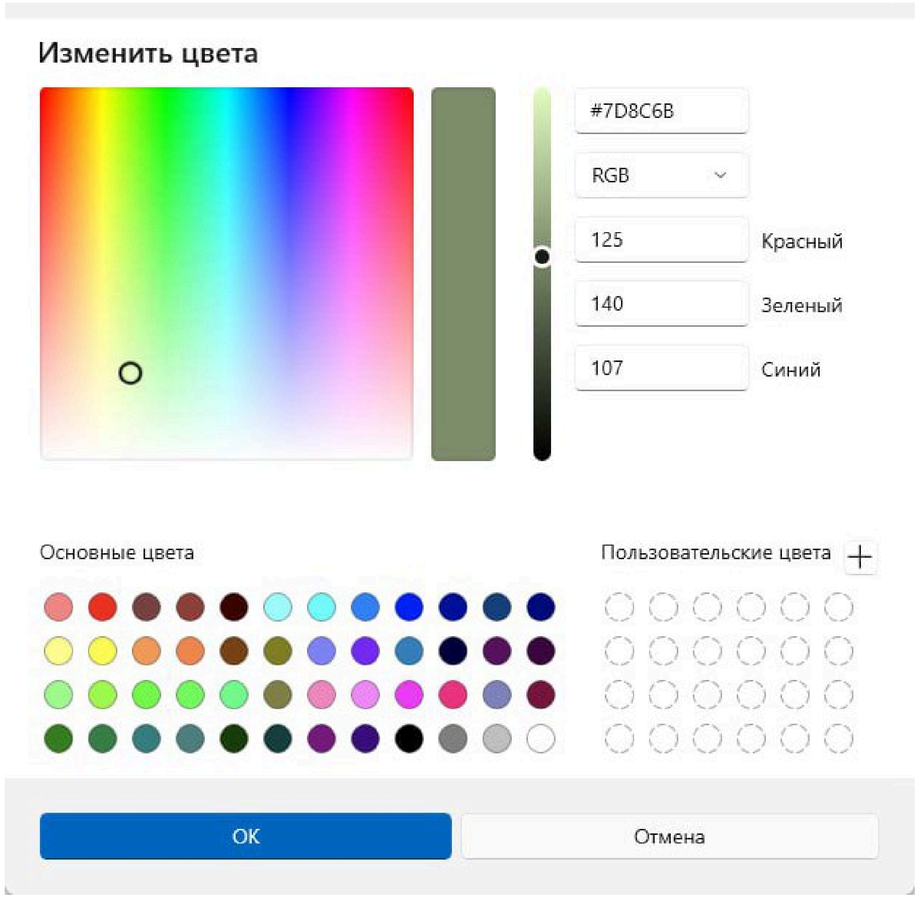
- Как представить изображение тензором
- Задача классификации изображений
- Свёртка
- Дополнительные операции (padding, stride,)
- Pooling
- Свёрточные нейронные сети

Свёрточная НС

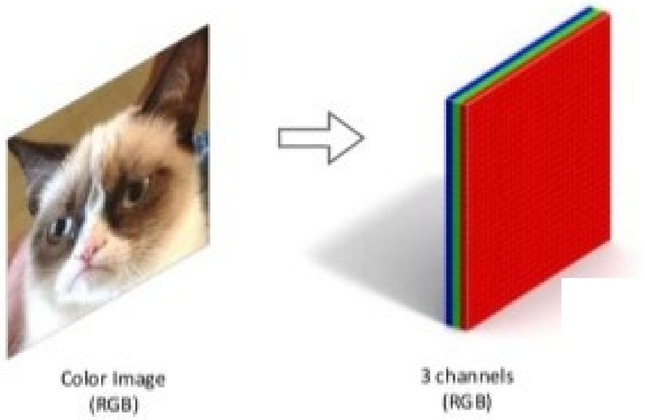
Свёрточная нейронная сеть (Convolutional Neural Network) — специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году для решения задач классификации изображений.

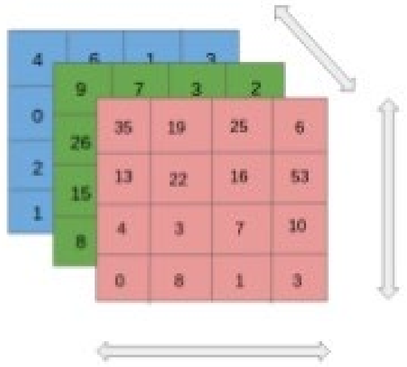
Как видит компьютер





H-W матрица





H (Height)

W (Width)

Изображение представлено в виде тензора:

$$H \times W \times 3$$

составленного из чисел – интенсивностей цвета пиксела.

Интенсивности RGB – кортеж (R, G, B);
(0, 0, 0) – чёрный цвет;
(255, 255, 255) – белый цвет;

Из примера: (125, 140, 170)

https://raw.githubusercontent.com/mlatsjsu/workshop-chihuahua-vs-muffin/1f3b63926d7031c4340154e4f56ae7acfa01fd64/resources/image_to_tensor.jpg

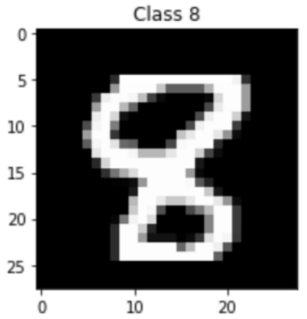
Распознавание рукописных изображений

MNIST – Mixed National Institute of Standards and Technology

База данных MNIST включает 60 000 тренировочных экземпляров изображений цифр и 10 000 проверочных изображений

На 2024г. средняя ошибка распознавания нейронной сети составляет менее 1%

Распознавание рукописных изображений

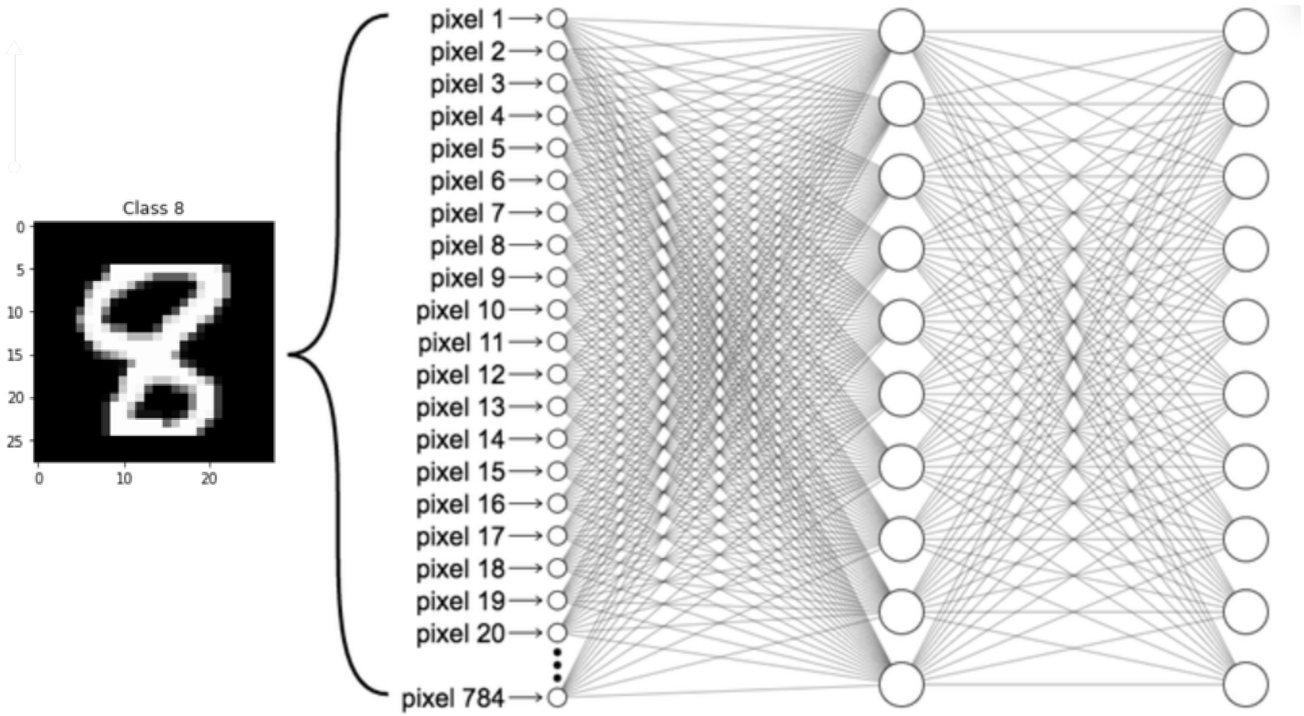



```
0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   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  45 254 254 254 254 254 254 254 254 254 254 255 242  49   0   0   0   0   0   0
0  0  0  0  0   0   0   0 153 253 253 253 253 198  95  95  95  95 177 253 253 253 156   0   0   0   0   0
0  0  0  0  0   0  58 237 251 253 246 191  64  11   0   0   0   0   9 213 253 253 156   0   0   0   0   0
0  0  0  0  0   0 159 253 253 184 101   0   0   0   0   0   0   0  74 234 253 253 156   0   0   0   0   0
0  0  0  0  0   0 237 253 215  18   0   0   0   0   0   0   0  77 237 253 253 207  21   0   0   0   0   0
0  0  0  0  0  77 245 253 124   0   0   0   0   0   0   0  73 240 253 253 253  60   0   0   0   0   0   0
0  0  0  0  0 158 253 253  52   0   0   0   0   0   0 196 232 253 253 231 147  13   0   0   0   0   0   0
0  0  0  0  0  93 246 253 227 106  95   0   0   0  70 249 253 253 253  43   0   0   0   0   0   0   0   0
0  0  0  0  0   0 231 253 253 253 246 184 184 184 229 253 253 208  69  12   0   0   0   0   0   0   0   0
0  0  0  0  0   0  41 214 253 253 253 253 253 253 253 253 253  25   0   0   0   0   0   0   0   0   0   0
0  0  0  0  0   0   0  48 116 172 219 253 253 253 253 253 154   9   0   0   0   0   0   0   0   0   0   0
0  0  0  0  0   0   0   0   2   4  13 150 253 253 253 253 249  88  34   0   0   0   0   0   0   0   0   0
0  0  0  0  0   0   0   0   0   0  27 253 253 253 253 253 253 253 199  25   0   0   0   0   0   0   0   0
0  0  0  0  0   0   0   0   0   0 185 253 253 234 200 200 231 253 253 232 149  13   0   0   0   0   0   0
0  0  0  0  0   0   0   0   0   0 202 253 248  80   0   0  72 122 190 253 253  60   0   0   0   0   0   0
0  0  0  0  0   0   0   0   7 121 244 253 220   0   0   0   0   0 131 253 253  60   0   0   0   0   0   0
0  0  0  0  0   0   0   0  44 253 253 205  38   0   0   0   0   0 131 253 253  60   0   0   0   0   0   0
0  0  0  0  0   0   0   0  44 253 253 155  18  18  18   1  11  55 224 253 245  56   0   0   0   0   0   0
0  0  0  0  0   0   0   0  44 253 253 253 253 253 253 102 189 253 253 239  86   0   0   0   0   0   0   0
0  0  0  0  0   0   0   0  44 253 253 253 253 253 253 253 253 253 253  96   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
```


hhttps://sok.susu.ru/courses/MachineLearnig/lectures/02%20Model.pdf

Flatten

Flatten (англ. выпрямление) – преобразование многомерного массива данных в одномерный массив



$28 \times 28 \Rightarrow 724$

Для примера: входной слой с изображением 224×224×3 содержит 150 528 нейронов

Даже один скрытый слой с 1000 нейронами потребует 150 млн параметров (весов)

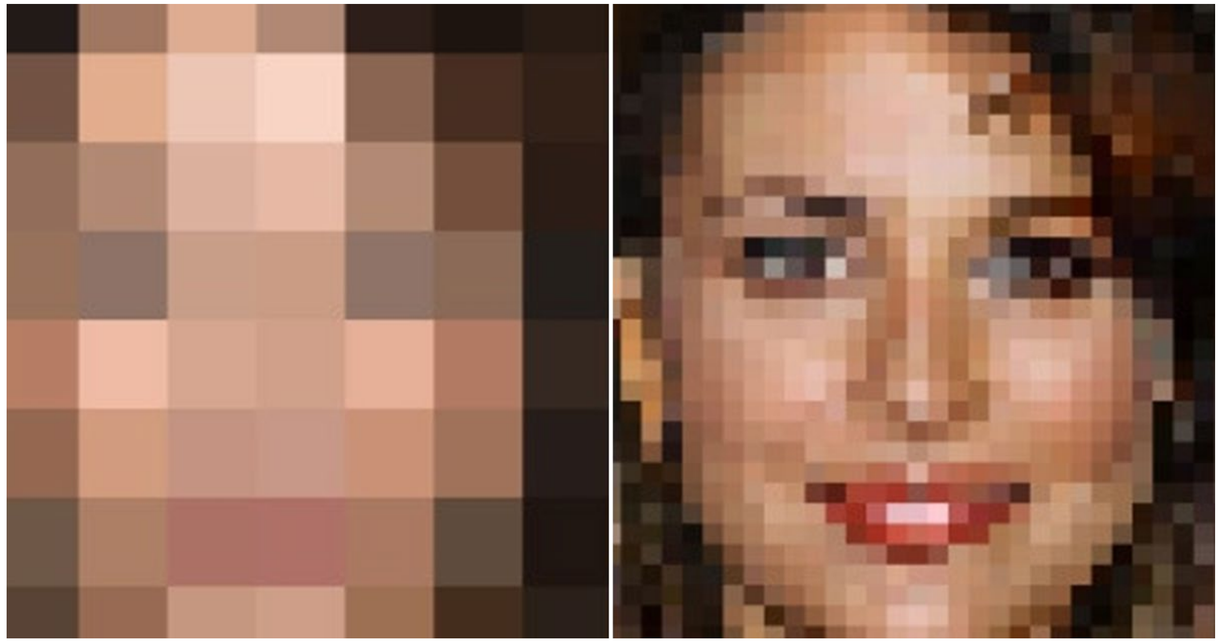
1. Слишком много параметров!

Почему не FCN?

2. Игнорирование пространственной структуры

Изображения обладают локальными корреляциями (пиксели рядом связаны) и пространственной иерархией признаков (края → текстуры → объекты)

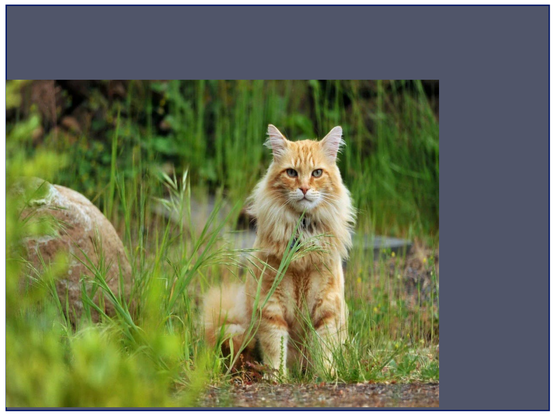
Принцип локальности:
чаще всего объект расположен на изображении в определённой области, а не размазан по всему изображению.



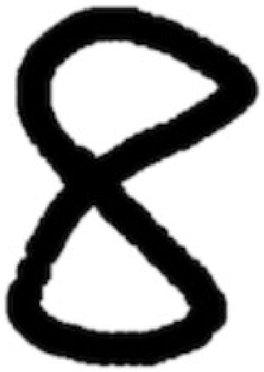
Чтобы найти нужный объект, нужно сконцентрироваться на локальных частях картинки, а не смотреть сразу на всё изображение целиком.

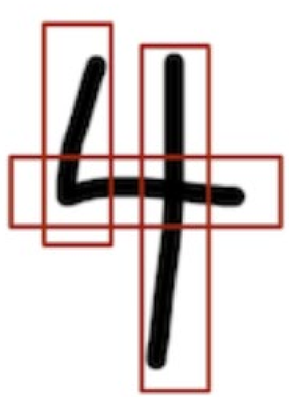
Почему не FCN?

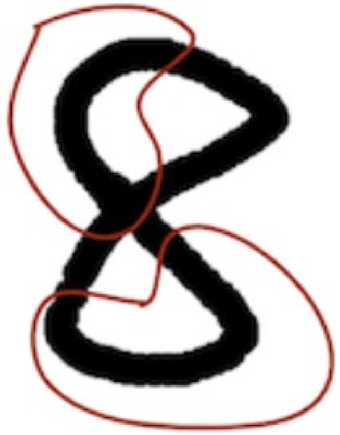
3. FCN не инвариантны к трансформациям изображения: сдвигам, поворотам, масштабированию и т.д.



Идея фильтра

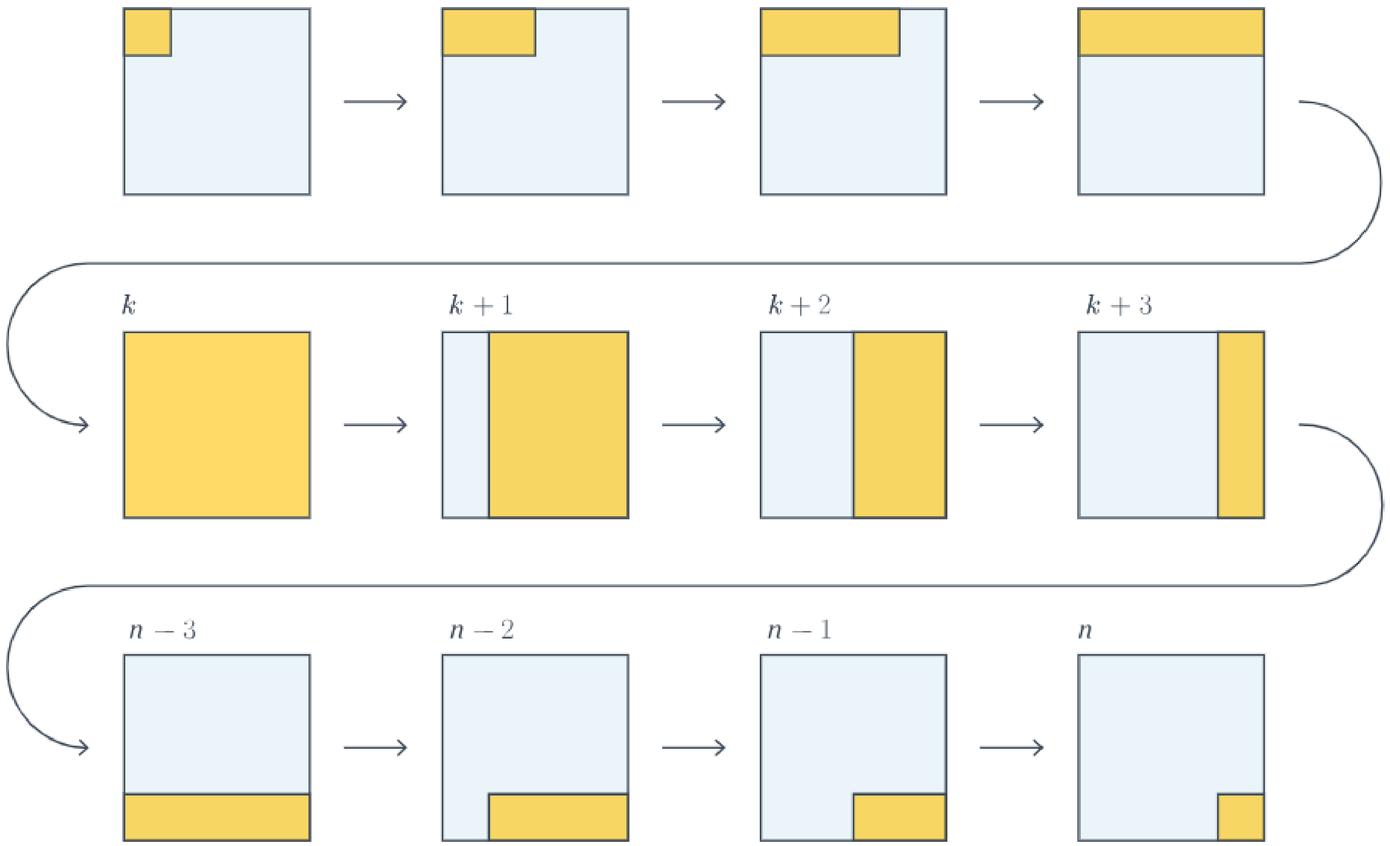






Искать на изображении паттерны (вертикальные/изогнутые линии и т.д.) и использовать эту информацию

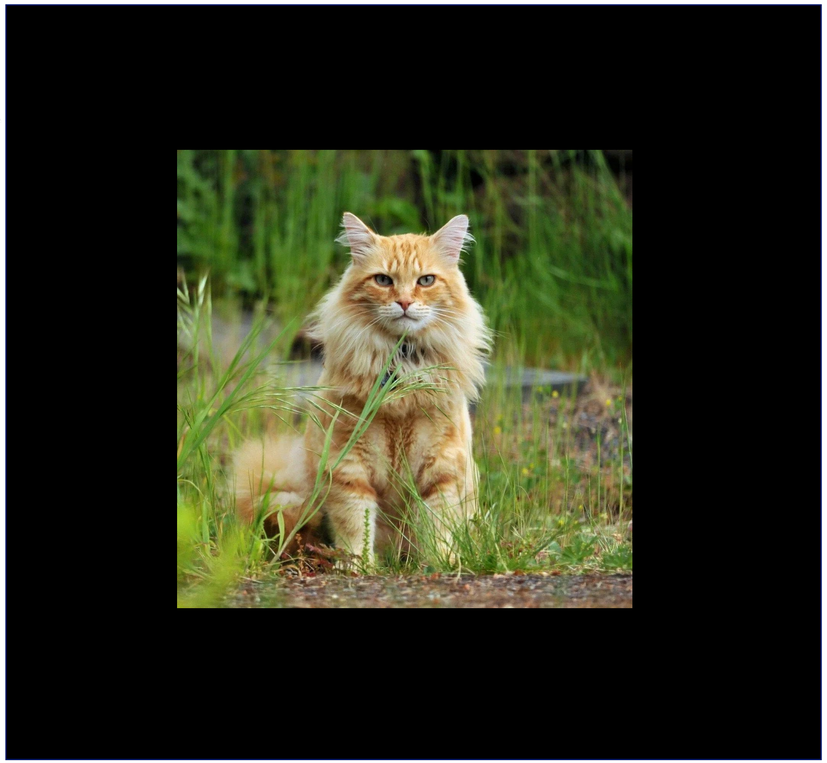
Идея фильтра

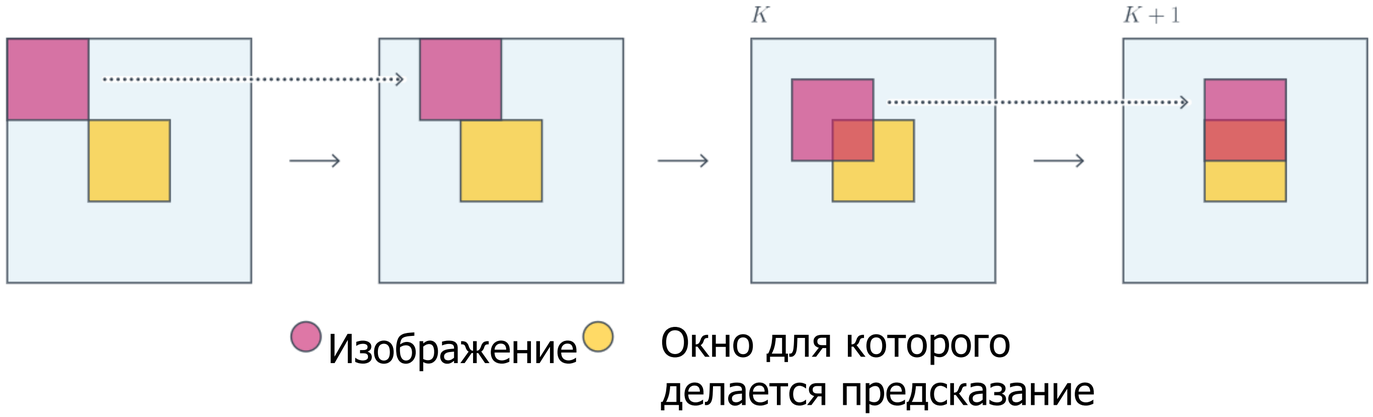


- Область изображения
- Область изображения с кошкой

https://education.yandex.ru/handbook/ml/article/svyortochnye-nejroseti

Идея фильтра





Свертка

Свертка - это операция "скользящего окна" (kernel/filter), которое проходит по всему входному изображению и вычисляет взвешенную сумму пикселей в области своего действия.

Фильтр (или, другое название, ядро; англ. kernel) — это матрица чисел, которая применяется к региону входных данных для выполнения операции свертки

0	50	0	29
0	80	31	2
33	90	0	75
0	9	0	95

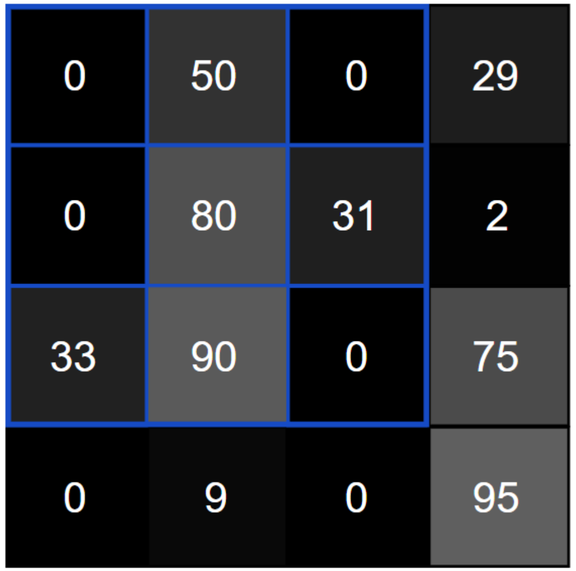
Черно-белое изображение 4×4

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

Фильтр 3×3

https://victorzhou.com/blog/intro-to-cnns-part-1/#reason-1-images-are-big

Свертка



*

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

=

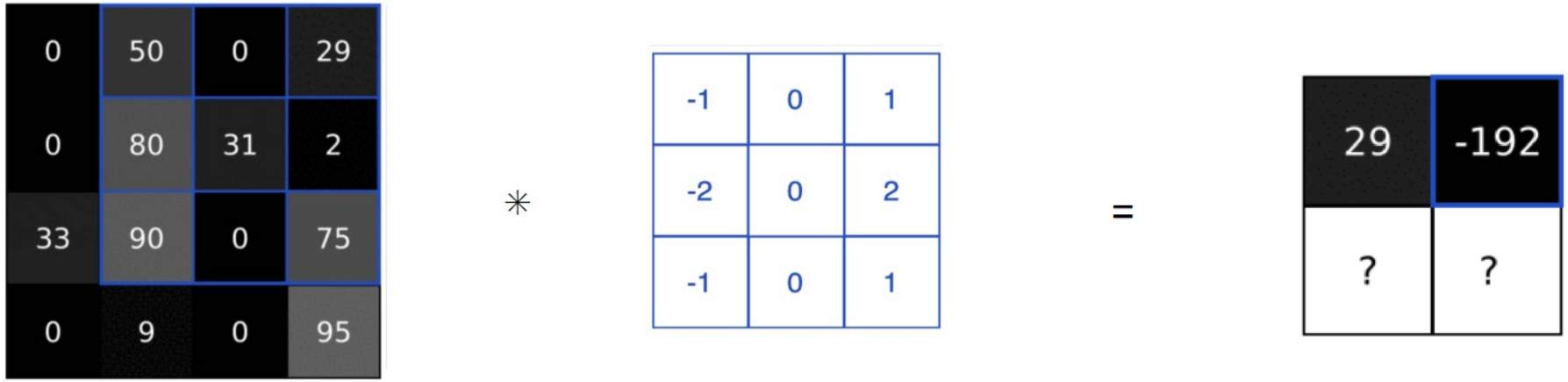
29	?
?	?

Свертка:
поэлементно умножаем пиксели картинки на соответствующие им числа фильтра:

$$(0 \times -1) + (50 \times 0) + (0 \times 1) + (0 \times -2) + (80 \times 0) + (31 \times 2) + (33 \times -1) + (90 \times 0) + (0 \times 1) = 29$$

https://victorzhou.com/blog/intro-to-cnns-part-1/#reason-1-images-are-big

Свертка

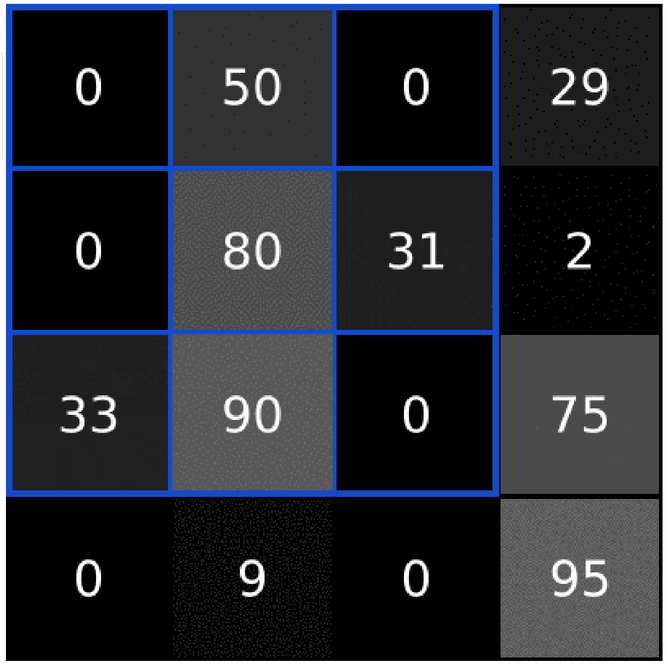


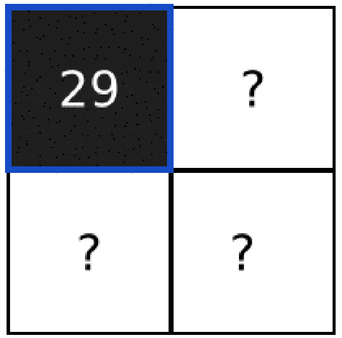
$$(50 \times -1) + (0 \times 0) + (29 \times 1) + (80 \times -2) + (31 \times 0) + (2 \times 2) + (90 \times -1) + (0 \times 0) + (75 \times 1) = -192$$

https://victorzhou.com/blog/intro-to-cnns-part-1/#reason-1-images-are-big

Карта признаков

Карта признаков – результат наложения фильтра на исходное изображение



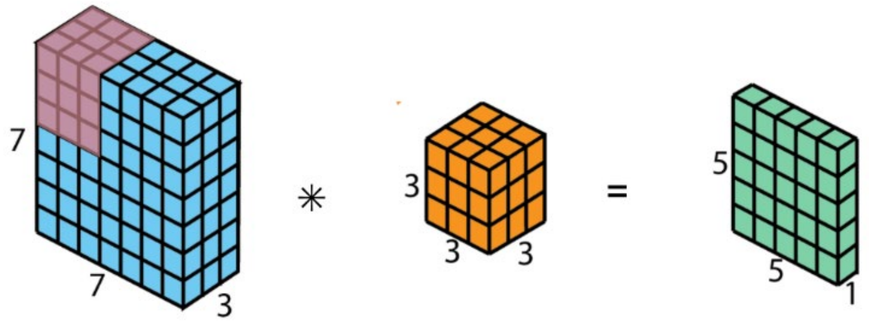


$$m = i - f + 1$$

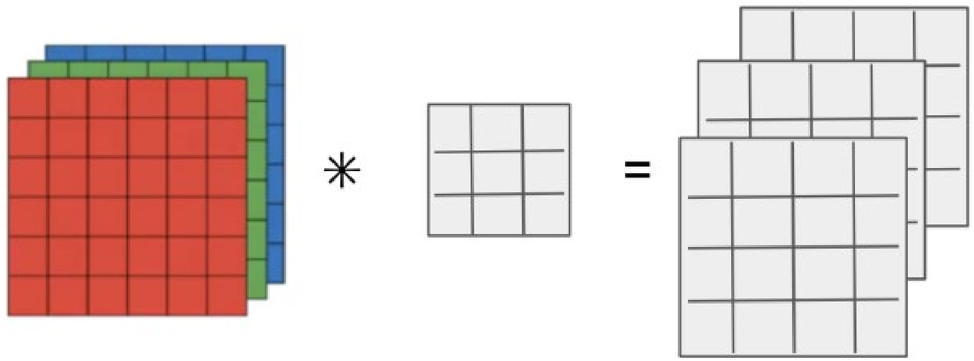
где:
m — размер карты активации
i — размер изображения
f — размер фильтра

https://victorzhou.com/blog/intro-to-cnns-part-1/#reason-1-images-are-big

Свертка RGB изображения

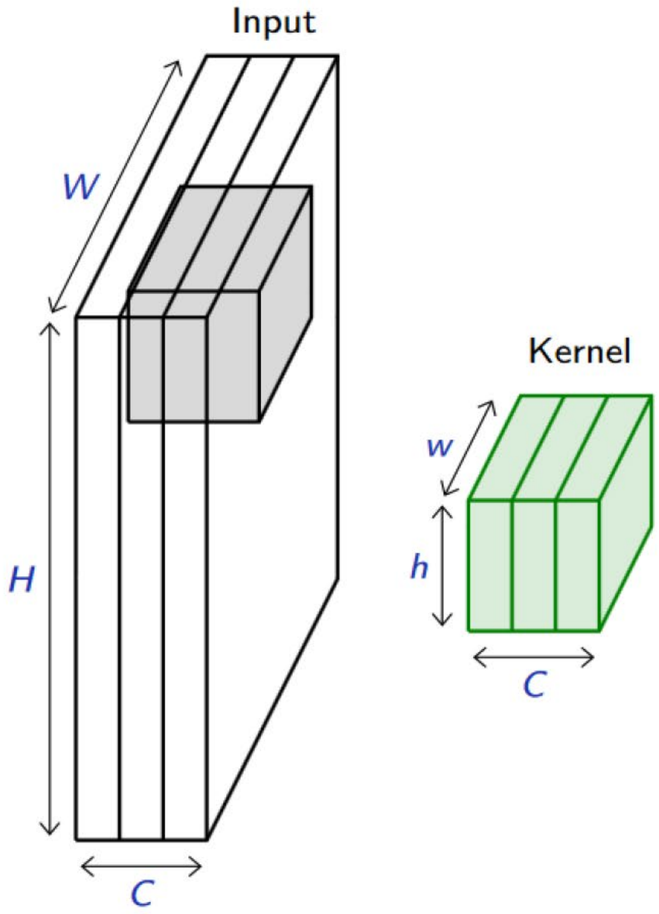


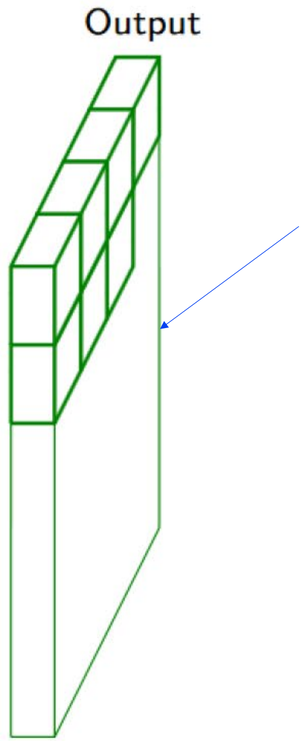
1 вариант: свертка трехмерным фильтром



2 вариант: свертка двумерным фильтром по каждому цветовому каналу

Свертка

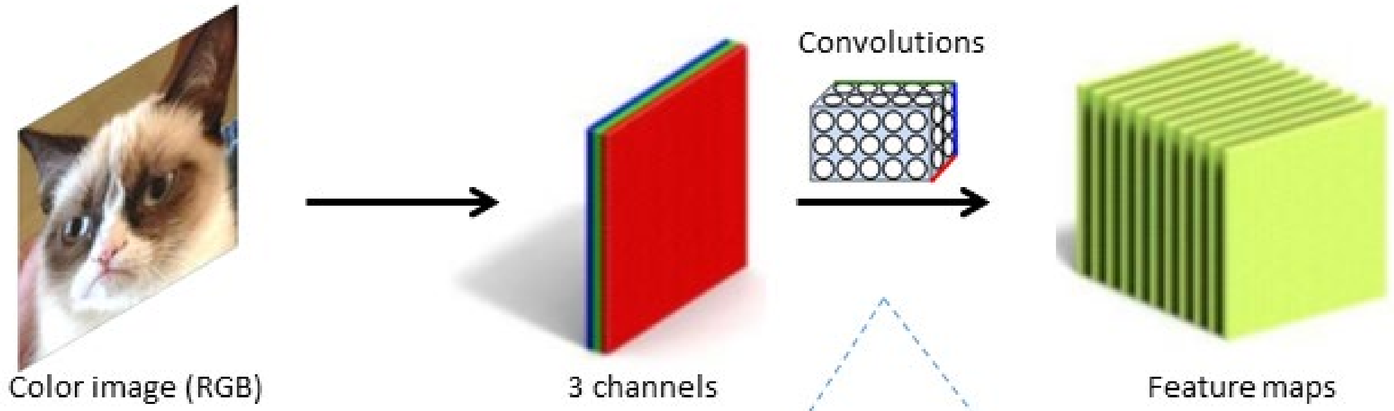


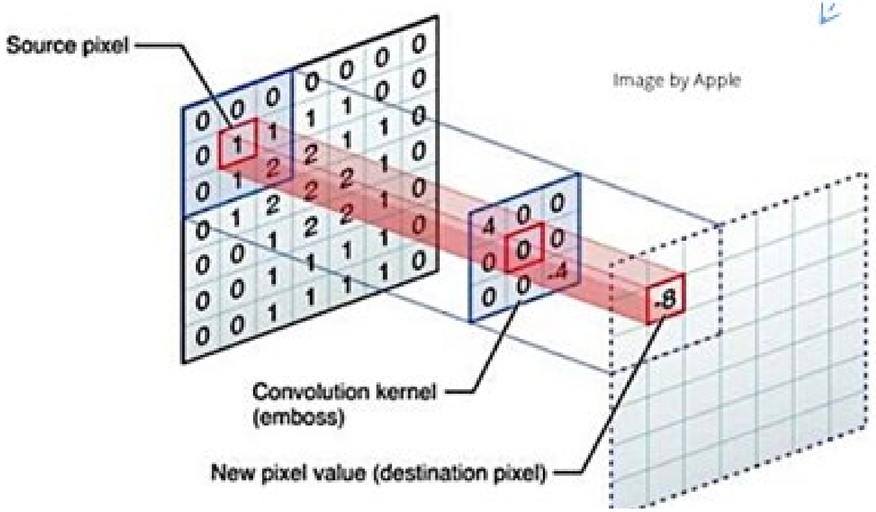


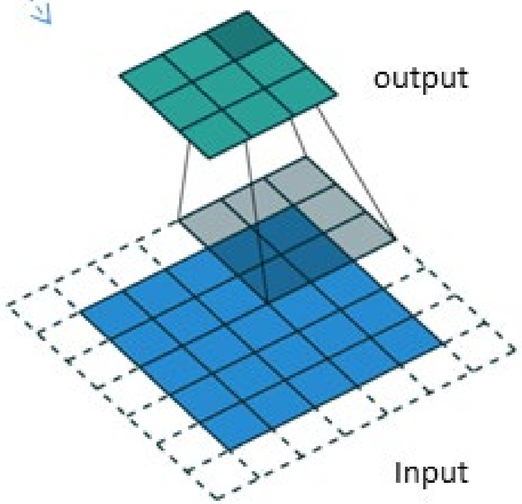
Глубина свертки = Числу каналов изображения

Карта признаков представляет низкоуровневые и среднеуровневые паттерны
Ее характеризуют:
Размер: пространственные размеры (высота × ширина)
Значения: активации определенных паттернов

Свертка


Convolutions
Color image (RGB)
3 channels
Feature maps


Source pixel
Image by Apple
Convolution kernel
(emboss)
New pixel value (destination pixel)


output
Input

Свертка

Для входного тензора I размера $H \times W$ и ядра K размера $k_h \times k_w$:

$$(I * K)[i,j] = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} I[i+m, j+n] \cdot K[m,n]$$

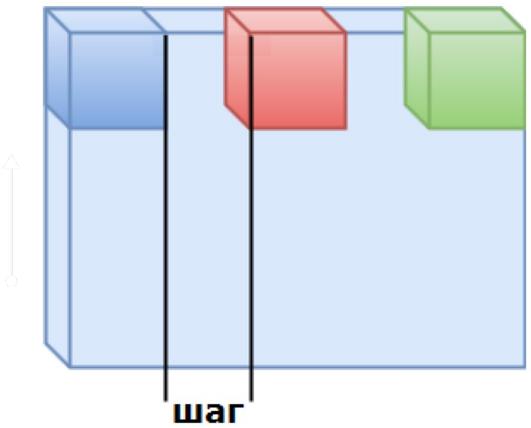
I - входное изображение

K - ядро свертки (фильтр)

i, j - координаты на выходной карте признаков

m, n - смещения внутри ядра

Шаг свёртки



Stride – шаг свертки

Насколько смещается ядро при вычислении свёрток (чем больше, тем меньше размер итогового изображения)

Вертикальный фильтр Собеля

Картинка

$*$

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

Вертикальный фильтр Собеля

$=$

Карта активации

https://victorzhou.com/blog/intro-to-cnns-part-1/#reason-1-images-are-big

Горизонтальный фильтр Собеля

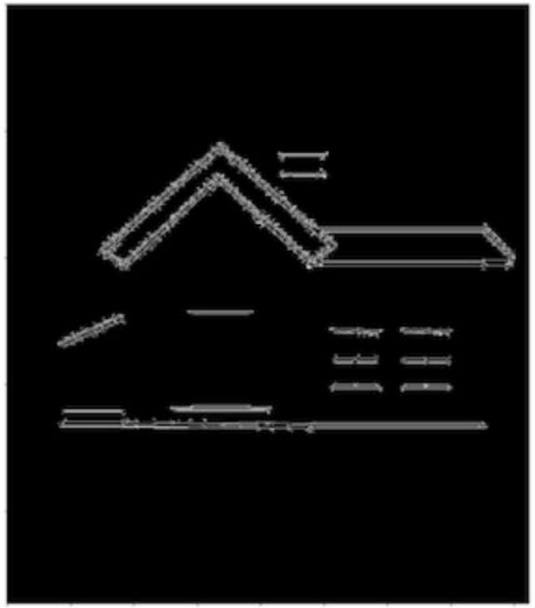
Картинка

*

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

Горизонтальный фильтр Собеля

=



Карта активации

https://victorzhou.com/blog/intro-to-cnns-part-1/#reason-1-images-are-big

Фильтр Собеля

R: 125
G: 140
B: 107

формула яркости по NTSC-стандарту:

$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$

1. Переводим в градацию серого:

Подставляем значения:
$Y = 0.299 \times 125 + 0.587 \times 140 + 0.114 \times 107$Y=0.299×125+0.587×140+0.1
14×107
$Y \approx 37.375 + 82.18 + 12.198 = 131.75$
Округлим: Y ≈ 132
Теперь работаем с этим значением

Фильтр Собеля

Применяем две матрицы (ядра свёртки):

По горизонтали:

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

По вертикали:

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

Вычисляем новый пиксель
Допустим, этот пиксель находится в окрестности 3×3, наш пиксель в самом центре:

$$\begin{bmatrix} 110 & 120 & 130 \\ 125 & 132 & 135 \\ 140 & 145 & 150 \end{bmatrix}$$

Считаем горизонтальную компоненту

$$Gx = (-1 \cdot 110) + (0 \cdot 120) + (1 \cdot 130) + (-2 \cdot 125) + (0 \cdot 132) + (2 \cdot 135) + (-1 \cdot 140) + (0 \cdot 145) + (1 \cdot 150) = -110 + 0 + 130 - 250 + 0 + 270 - 140 + 0 + 150 = -110 + 0 + 130 - 250 + 0 + 270 - 140 + 0 + 150 = 50$$

Фильтр Собеля

$$\begin{bmatrix} 110 & 120 & 130 \\ 125 & 132 & 135 \\ 140 & 145 & 150 \end{bmatrix}$$

Считаем вертикальную компоненту

$$Gy = (-1\cdot110)+(-2\cdot120)+(-1\cdot130)+(0\cdot125)+(0\cdot132)+(0\cdot135)+(1\cdot140)+(2\cdot145)+(1\cdot150)$$
$$=-110-240-130+0+0+0+140+290+150=-110-240-130+0+0+0+140+290+150=100=100$$

Находим итоговое значение

$G = \sqrt{(50^2 + 100^2)} = \sqrt{(2500 + 10000)} = \sqrt{12500} \approx 111.8$

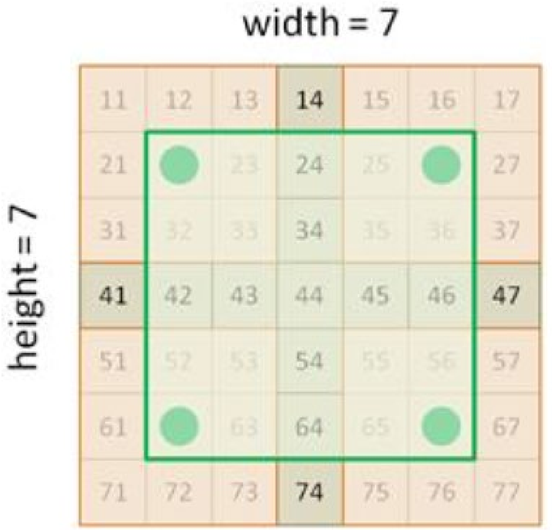
Результат: новый уровень яркости ≈ 112.

Паддинг

Входное изображение 7×7

Фильтр 3×3

Карта признаков 5×5



Количество "вхождений" каждого пикселя в свертку:

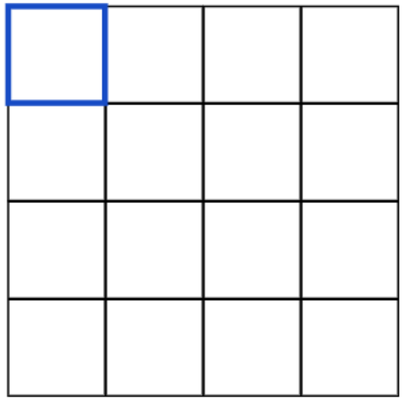
- Угловые пиксели (некоторые): 1 раз
- Центральные пиксели: 9 раз

Паддинг выравнивает это неравенство!

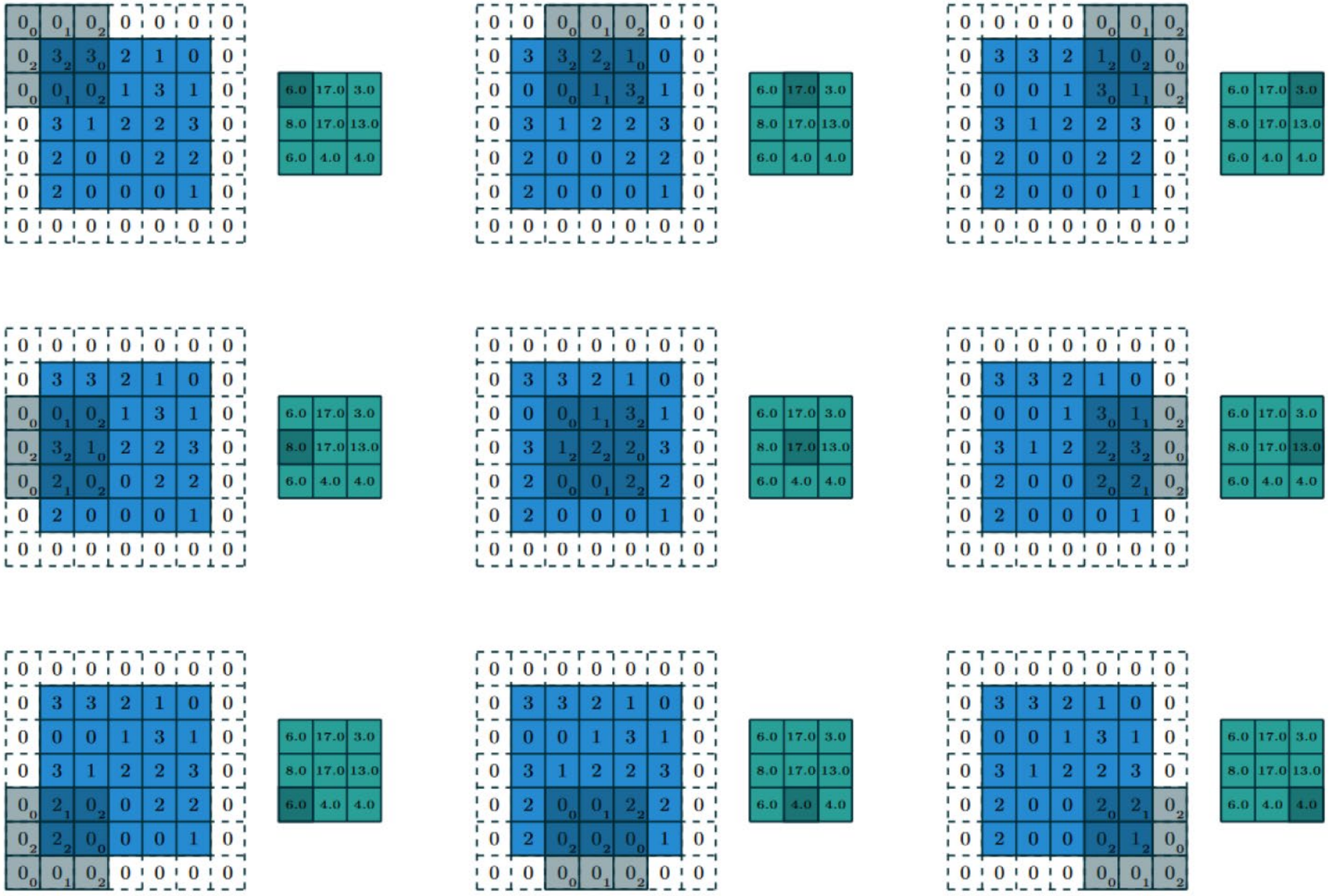
Паддинг

Паддинг (padding) – техника для расширения входных данных путем добавления «рамки» из пикселей перед операцией свертки этого изображения фильтром

0	0	0	0	0	0
0	0	50	0	29	0
0	0	80	31	2	0
0	33	90	0	75	0
0	0	9	0	95	0
0	0	0	0	0	0

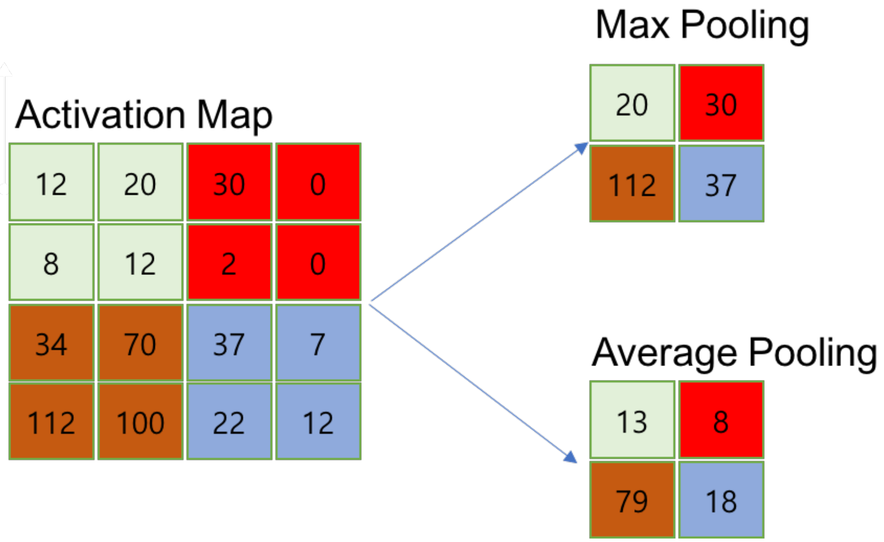


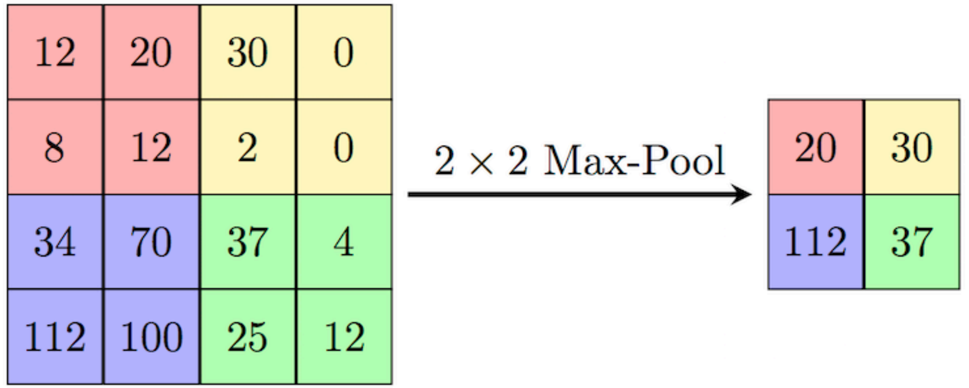
Паддинг


0 0 0 0 0 0 0
0 3 3 2 1 0 0
0 0 0 1 3 1 0
0 3 1 2 2 3 0
0 2 0 0 2 2 0
0 2 0 0 0 1 0
0 0 0 0 0 0 0
6.0 17.0 3.0
8.0 17.0 13.0
6.0 4.0 4.0

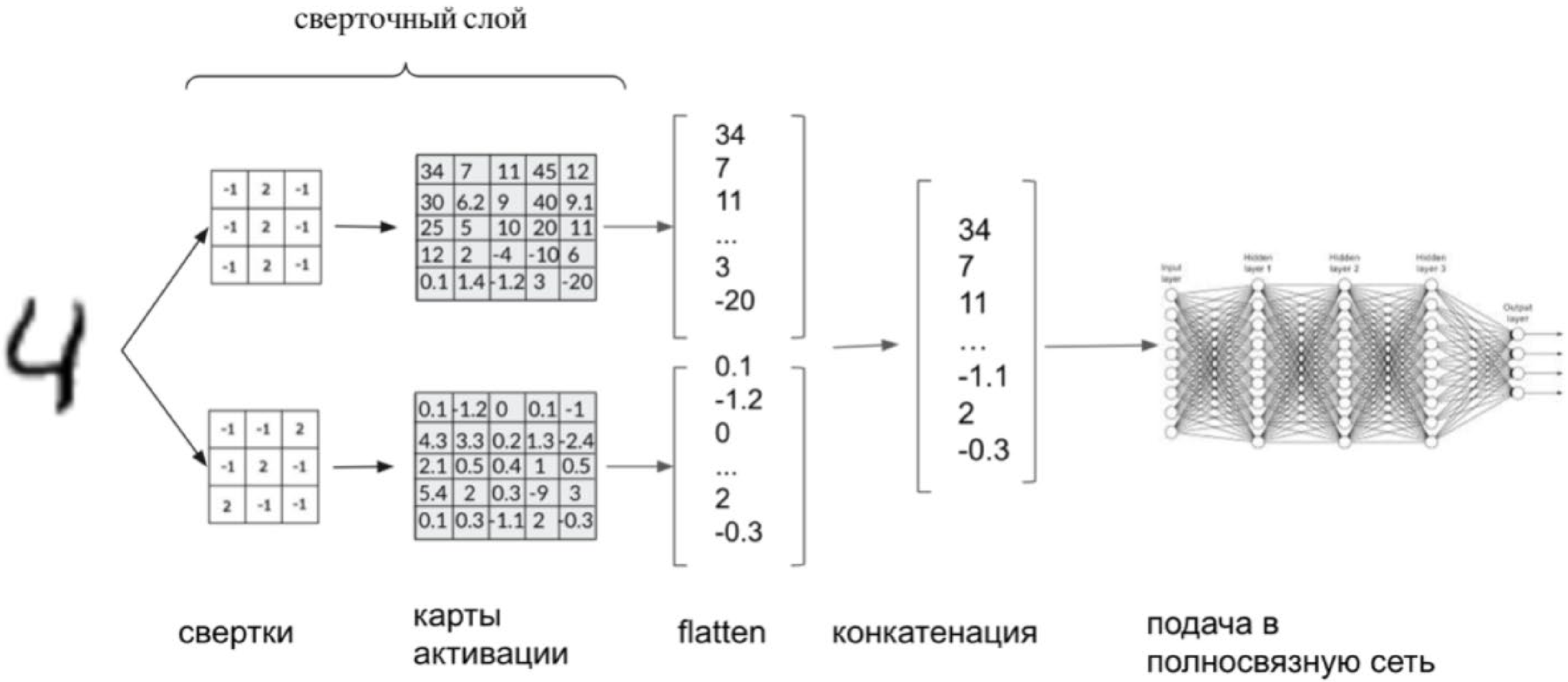
Пуллинг

Пуллинг (pulling) – операция уменьшения карт активации





CNN для MNIST



https://victorzhou.com/blog/intro-to-cnns-part-1/#reason-1-images-are-big

МИСИС
УНИВЕРСИТЕТ

Спасибо
за внимание!