

ML для изображений и сверточные нейросети

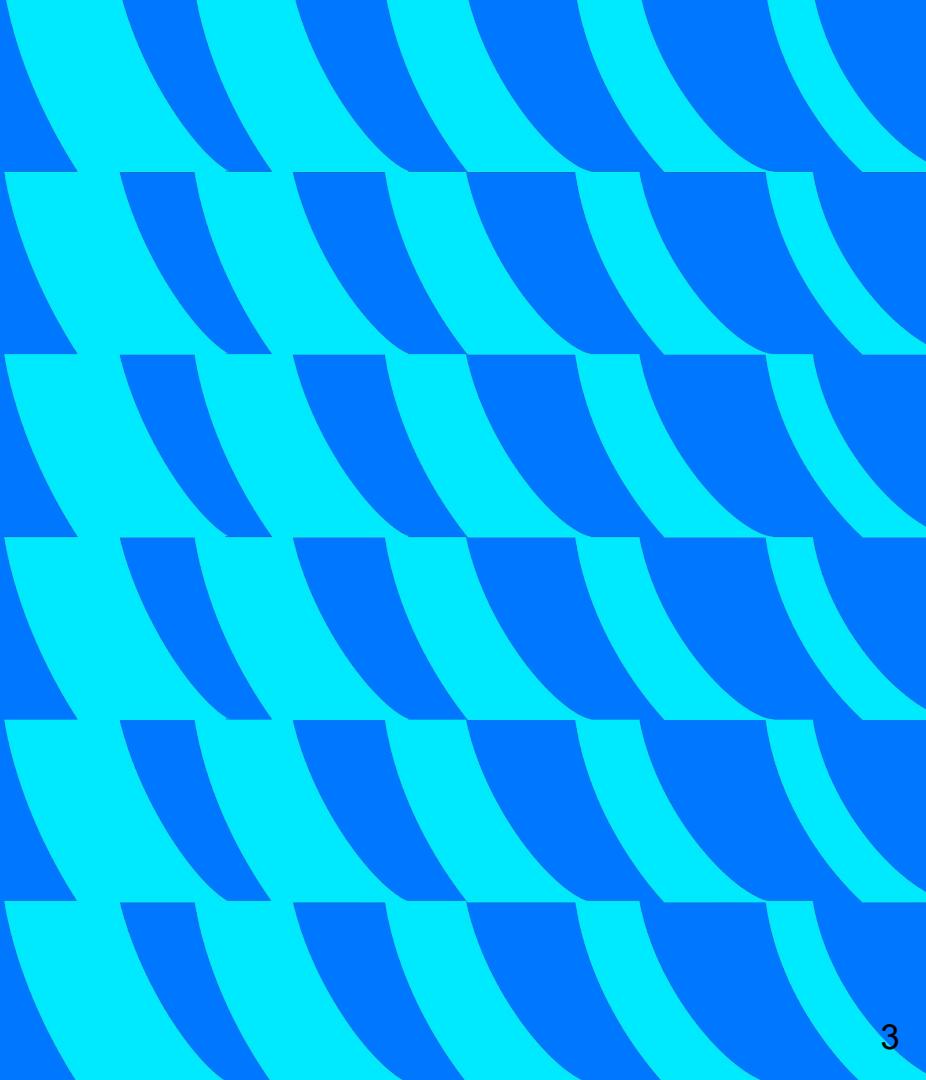
Даниил Лысухин, Ведущий разработчик в команде Perception



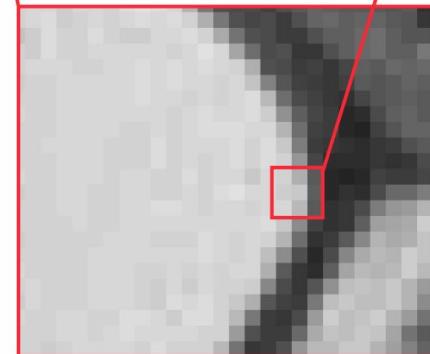
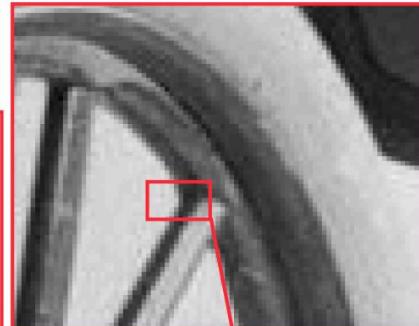
План

- Как устроены изображения
- ML на изображениях
- Свертка
- Сверточный слой
- Пулинг
- CNN
- Итоги

Как устроены изображения



Изображение = пиксели



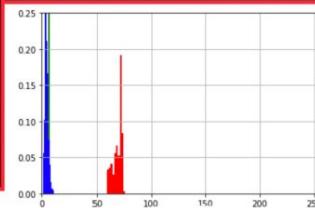
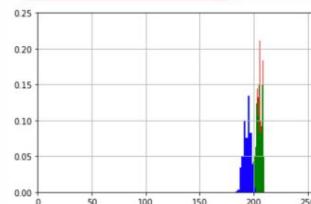
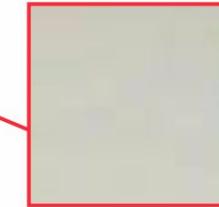
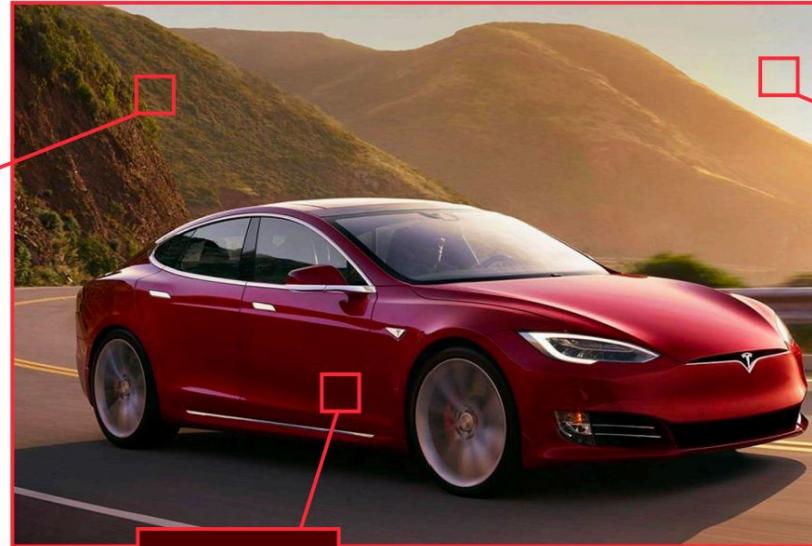
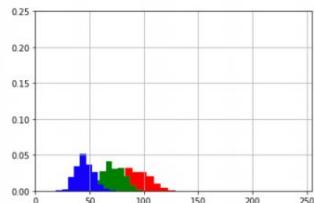
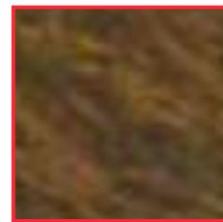
[214, 176, 90]
[207, 204, 97]
[218, 186, 93]

214	176	90
207	204	97
218	186	93

Представление цвета

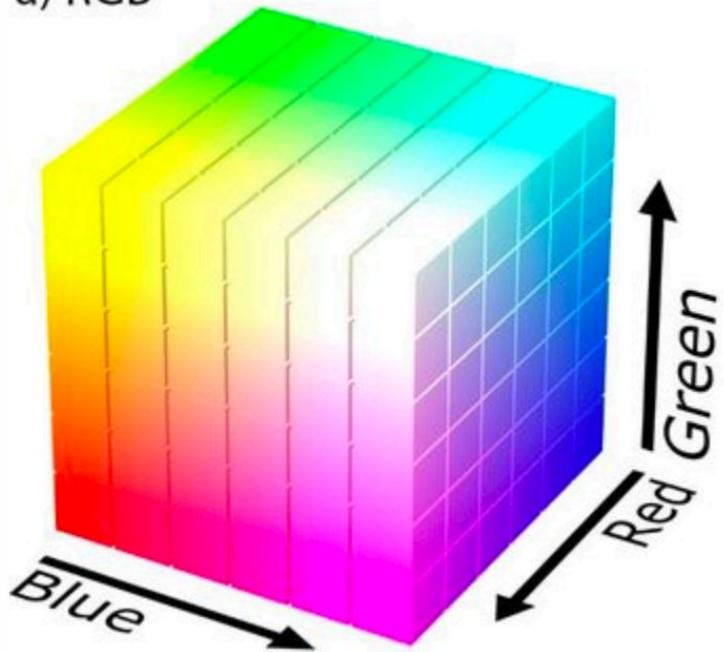


Представление цвета

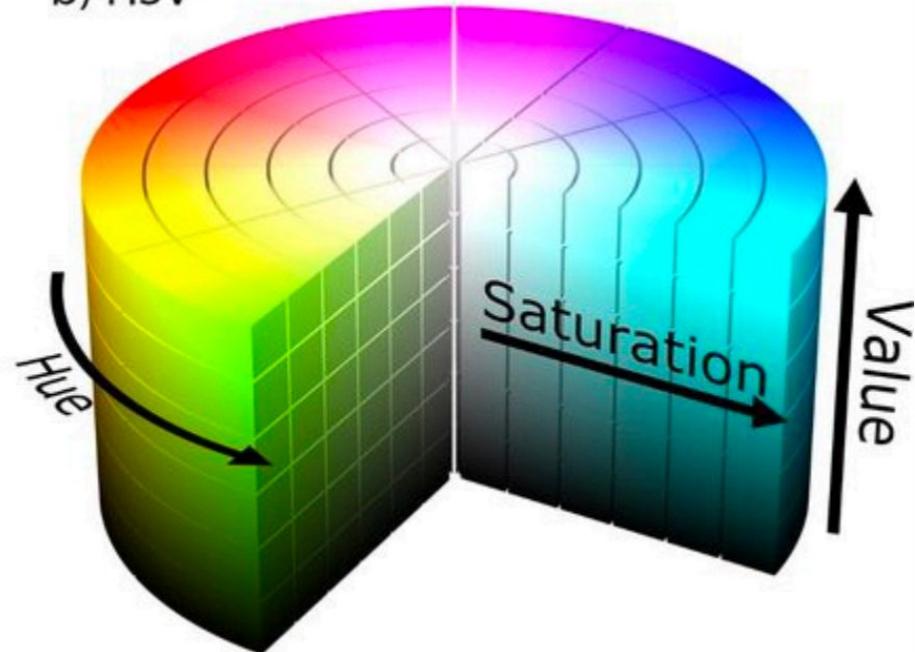


Цветовые пространства - не только RGB

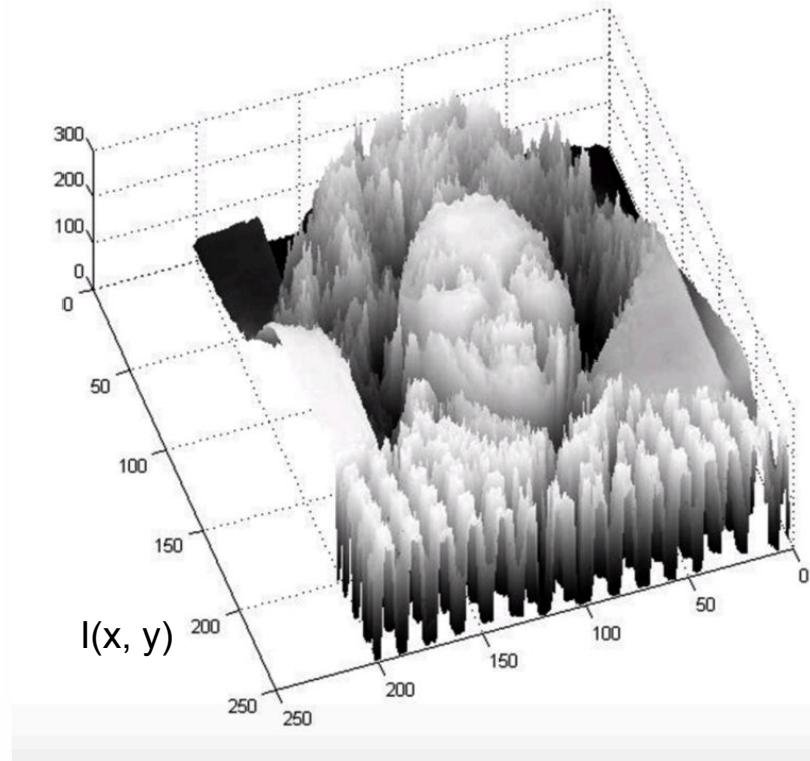
a) RGB



b) HSV



Изображение как функция



ML на изображениях



Как засунуть изображение в модель ML

- Для многих ML моделей (линейных/деревянных/...) хорошо подходят "табличные" данные
 - объект = набор численных признаков
- Как превратить изображение в набор признаков?

Как засунуть изображение в модель ML

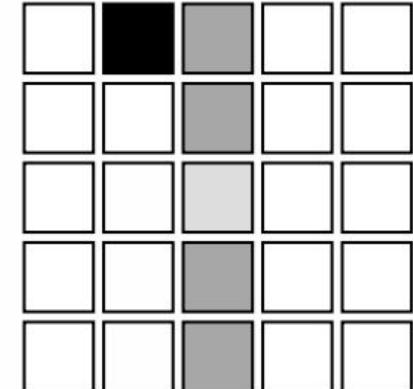
- Для многих ML моделей (линейных/деревянных/...) хорошо подходят "табличные" данные
 - объект = набор численных признаков
- Как превратить изображение в набор признаков?
 - Признаки = исходные значения яркости пикселей
 - Признаки извлекаются вручную (hand-crafted)
 - Признаки извлекаются автоматически

Как засунуть изображение в модель ML

- Для многих ML моделей (линейных/деревянных/...) хорошо подходят "табличные" данные
 - объект = набор численных признаков
- Как превратить изображение в набор признаков?
 - **Признаки = исходные значения яркости пикселей**
 - Признаки извлекаются вручную (hand-crafted)
 - Признаки извлекаются автоматически

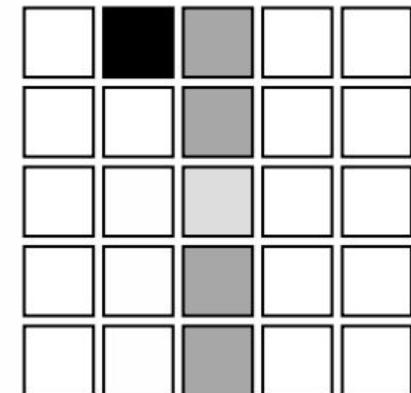
Признаки = яркости всех пикселей

- Хотим сделать классификатор grayscale-изображений цифр от 0 до 9
- Размер изображений - 5×5 пикселей
- Используем полносвязную нейросеть с 1 скрытым слоем
- Какова размерность весов этого слоя?

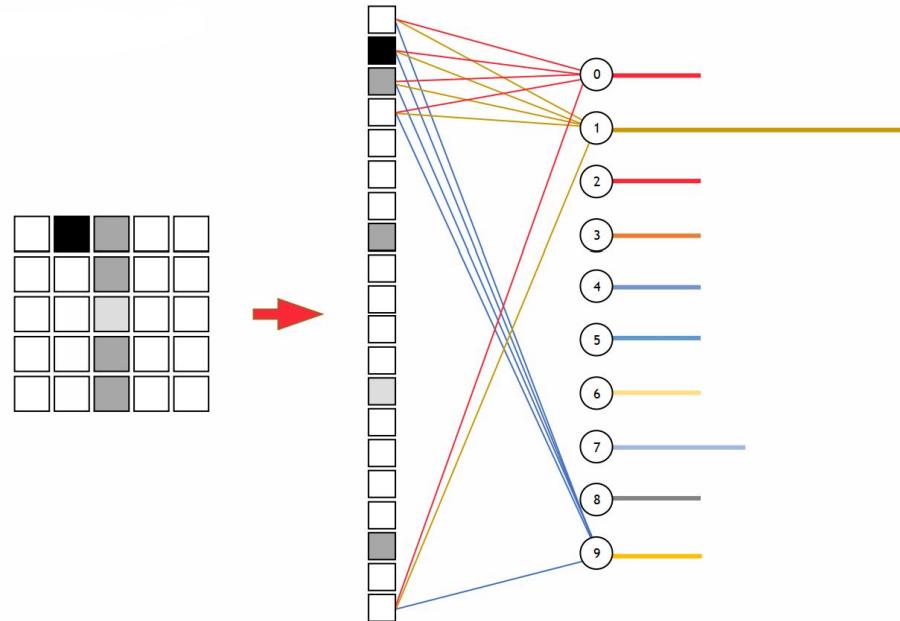


Признаки = яркости всех пикселей

- Хотим сделать классификатор grayscale-изображений цифр от 0 до 9
- Размер изображений - 5 x 5 пикселей
- Используем полносвязную нейросеть с 1 скрытым слоем
- Какова размерность весов этого слоя?
 - Число входов = 25
 - Число выходов = 10
 - Число весов ~ 250 (без bias)

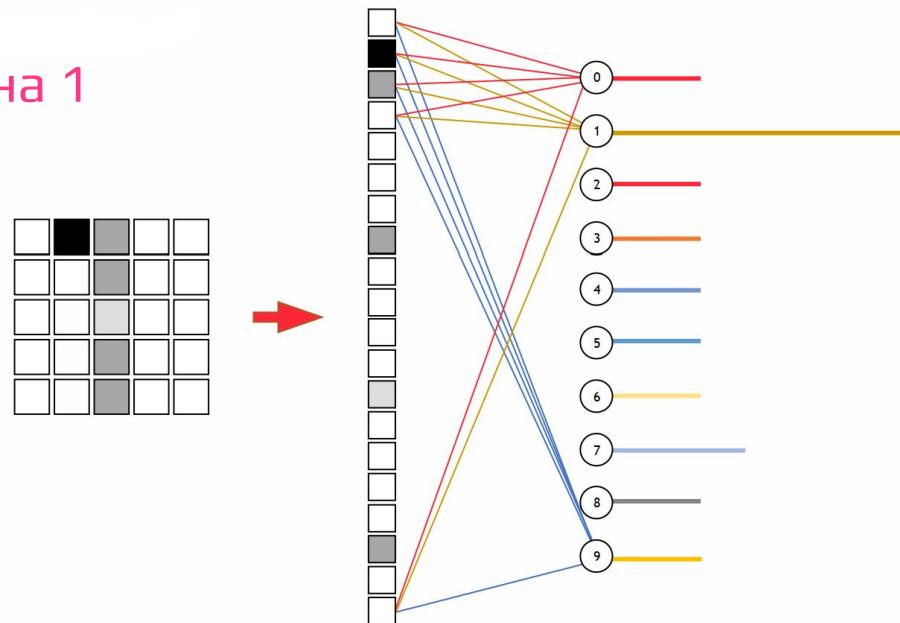


Признаки = яркости всех пикселей



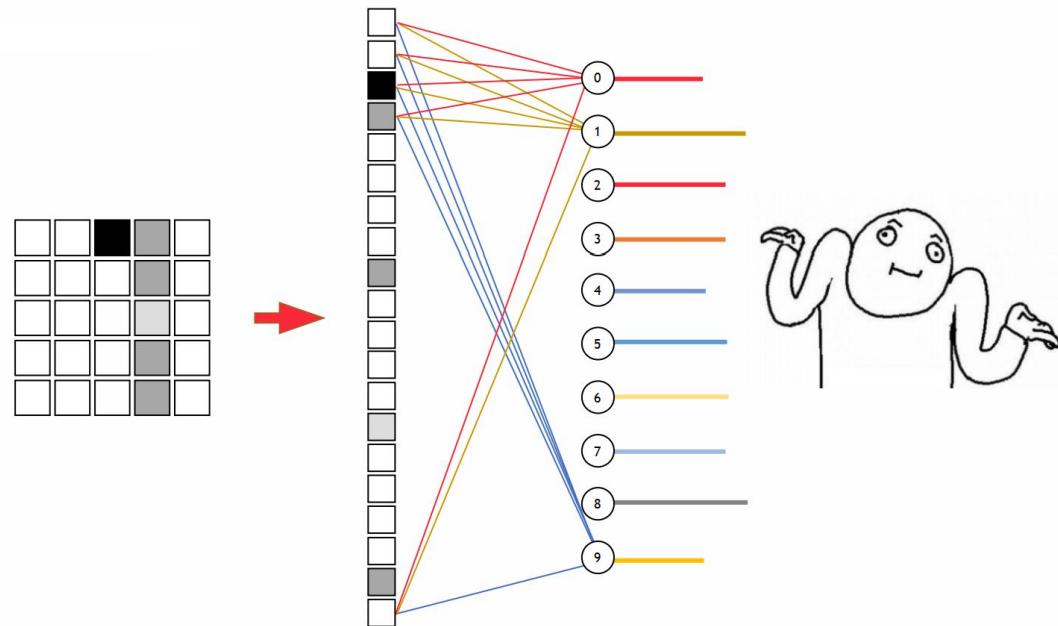
Признаки = яркости всех пикселей

К чему приведет
смещение объекта на 1
пиксель?



Признаки = яркости всех пикселей

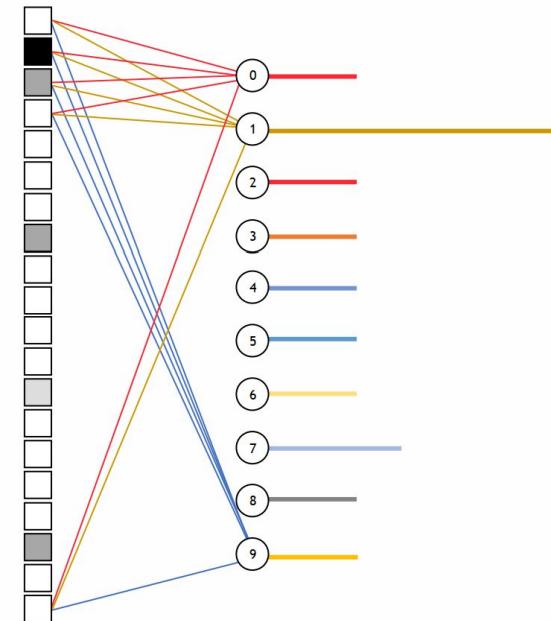
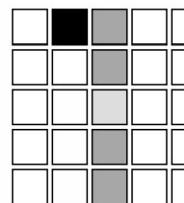
Проблема: нет
инвариантности к
смещениям



Признаки = яркости всех пикселей

К чему приведет

- увеличение размера картинки,
- добавление цвета,
- увеличение числа классов?



Признаки = яркости всех пикселей

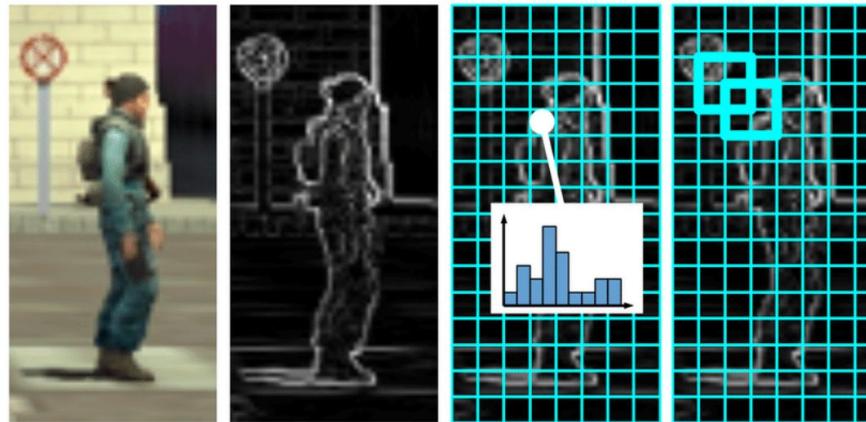
- Пример: ImageNet
- Изображения 224x224x3
- 1000 классов
- Число параметров однослойной нейросети:
 - $224 \times 224 \times 3 \times 1000 \sim 150\ 000\ 000$

Как засунуть изображение в модель ML

- Для многих ML моделей (линейных/деревянных/...) хорошо подходят "табличные" данные
 - объект = набор численных признаков
- Как превратить изображение в набор признаков?
 - Признаки = исходные значения яркости пикселей
 - **Признаки извлекаются вручную (hand-crafted)**
 - Признаки извлекаются автоматически

Как извлечь признаки вручную

- В методах feature engineering для изображений чаще всего используют статистики
 - Гистограмма направленности градиентов (HOG)



detection window
slides over an
image

at each location where
the window is applied,
gradients are
computed

window is evenly
partitioned into cells
and each pixel of the
cell contributes to cell
gradient orientation
histogram

orientation histograms
for overlapping 2x2
blocks of cells are
normalized and
collected to form the
final descriptor

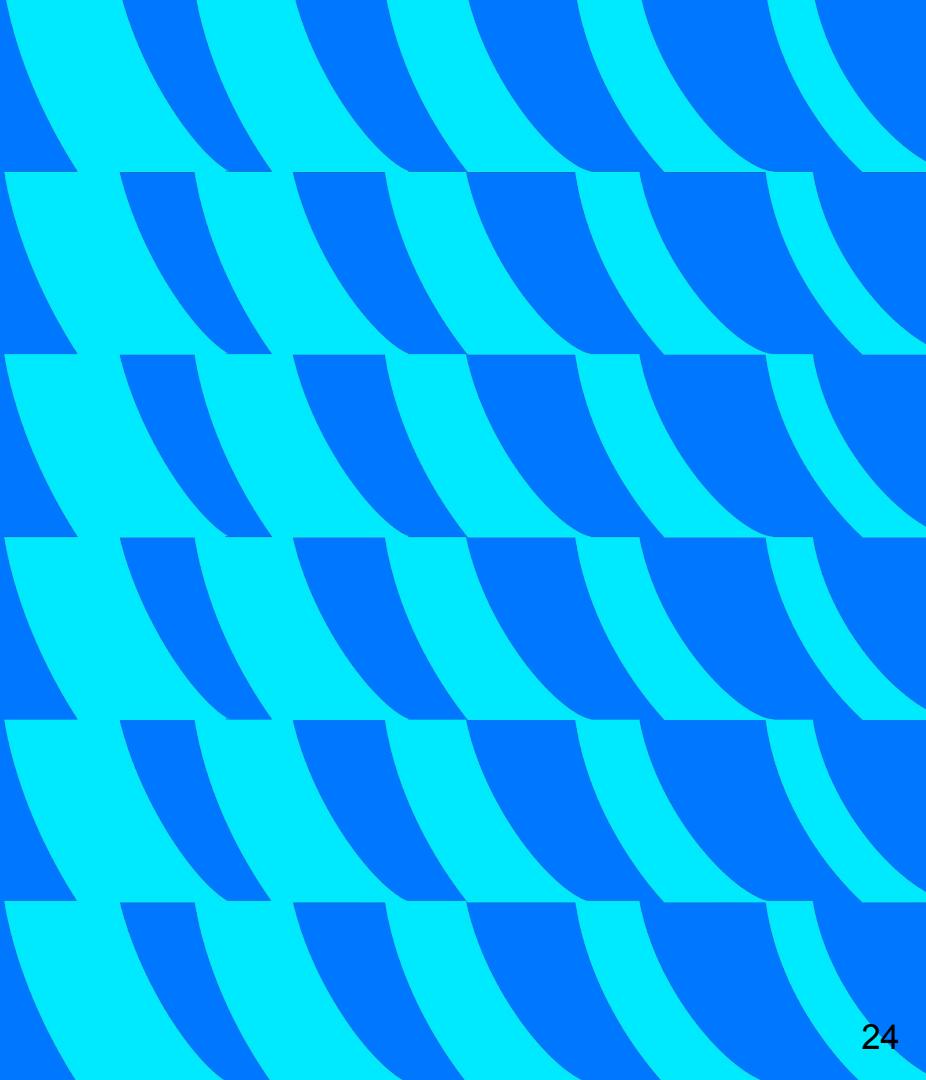
Как извлечь признаки вручную

- Основные проблемы "ручных" признаков
 - Долго считаются
 - Плохо обобщаются на новые случаи
 - Неустойчивы к изменениям яркости, контраста, цвета, ...

Как засунуть изображение в модель ML

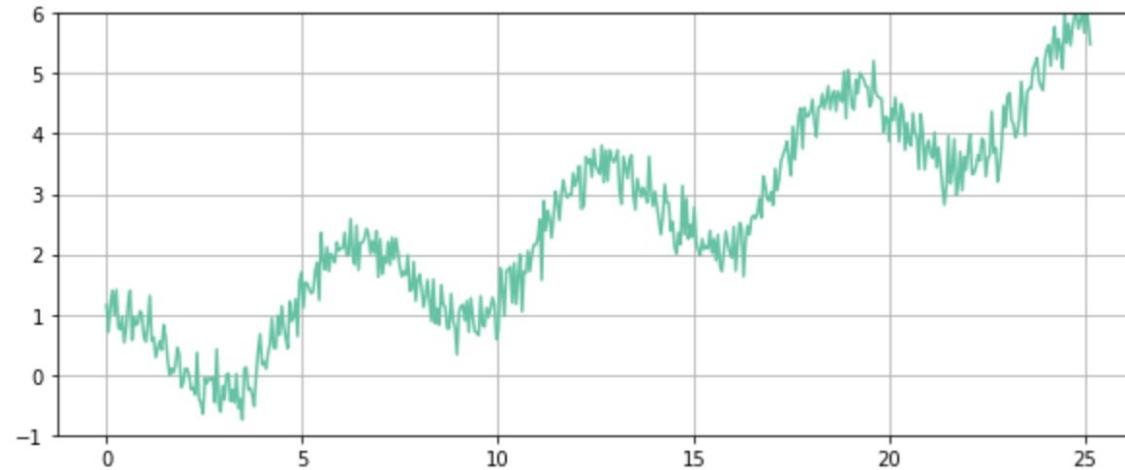
- Для многих ML моделей (линейных/деревянных/...) хорошо подходят "табличные" данные
 - объект = набор численных признаков
- Как превратить изображение в набор признаков?
 - Признаки = исходные значения яркости пикселей
 - Признаки извлекаются вручную (hand-crafted)
 - **Признаки извлекаются автоматически**

Свертка



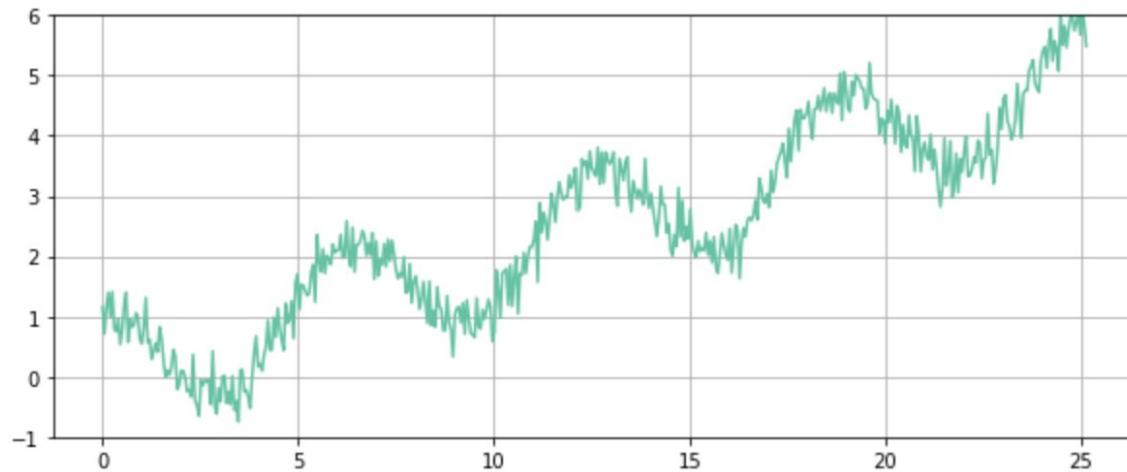
Пример

- Имеется некоторый сигнал
 - Например, показания физического датчика за промежуток времени
- Сигнал явно шумный - как его "почистить"?



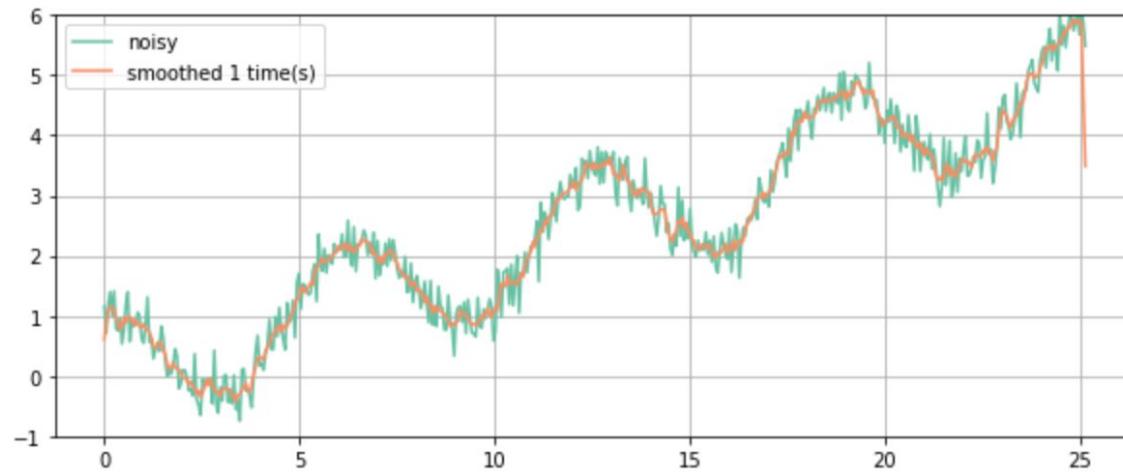
Пример

- Применим к "сглаживание" с помощью скользящего среднего
- Выберем размер окна; например, 5
- Каждое значение сигнала заменим средним по окрестности размера 5; можно повторить несколько раз



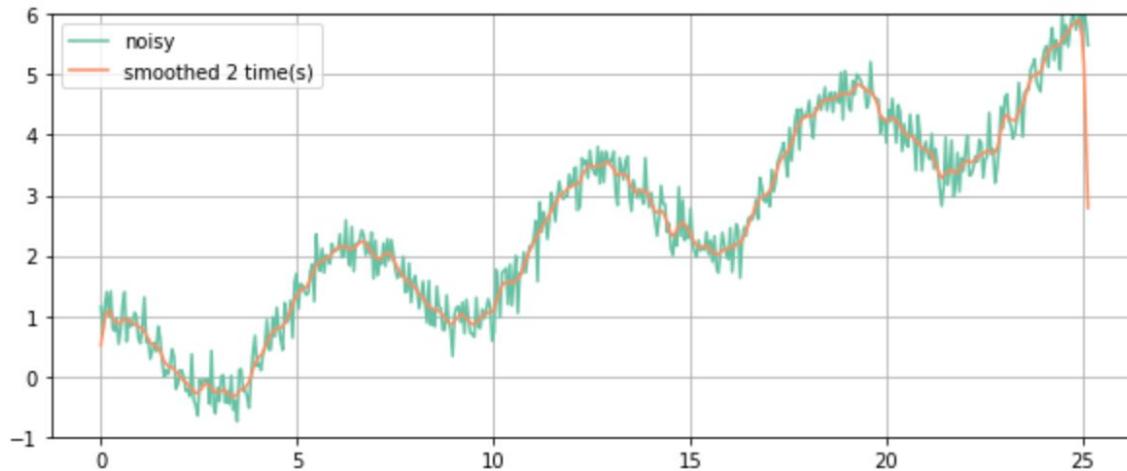
Пример

- Применим к "сглаживание" с помощью скользящего среднего
- Выберем размер окна; например, 5
- Каждое значение сигнала заменим средним по окрестности размера 5; можно повторить несколько раз



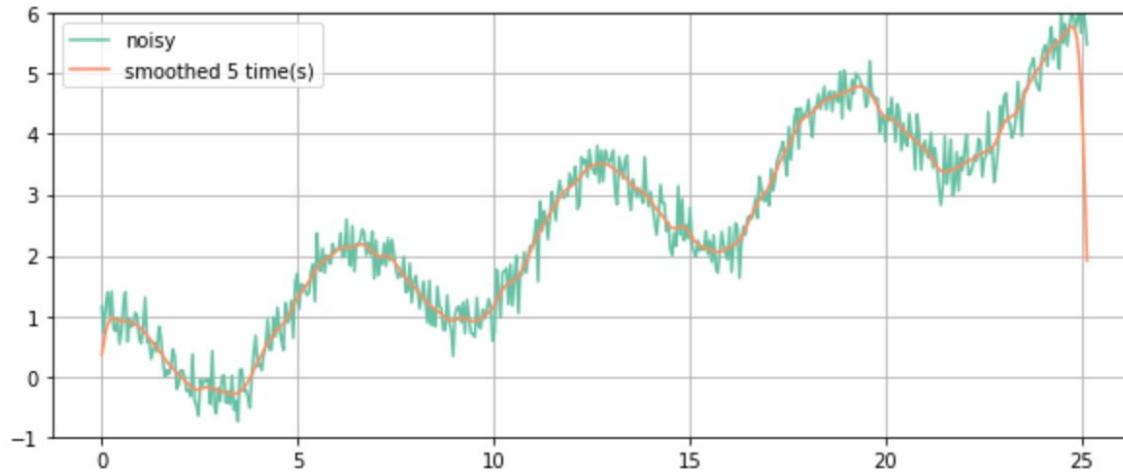
Пример

- Применим к "сглаживание" с помощью скользящего среднего
- Выберем размер окна; например, 5
- Каждое значение сигнала заменим средним по окрестности размера 5; можно повторить несколько раз



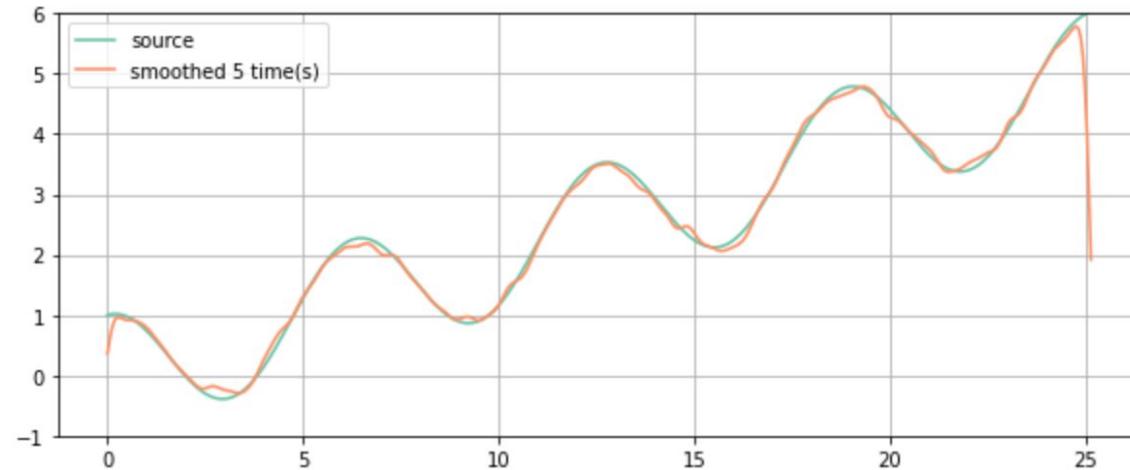
Пример

- Применим к "сглаживание" с помощью скользящего среднего
- Выберем размер окна; например, 5
- Каждое значение сигнала заменим средним по окрестности размера 5; можно повторить несколько раз



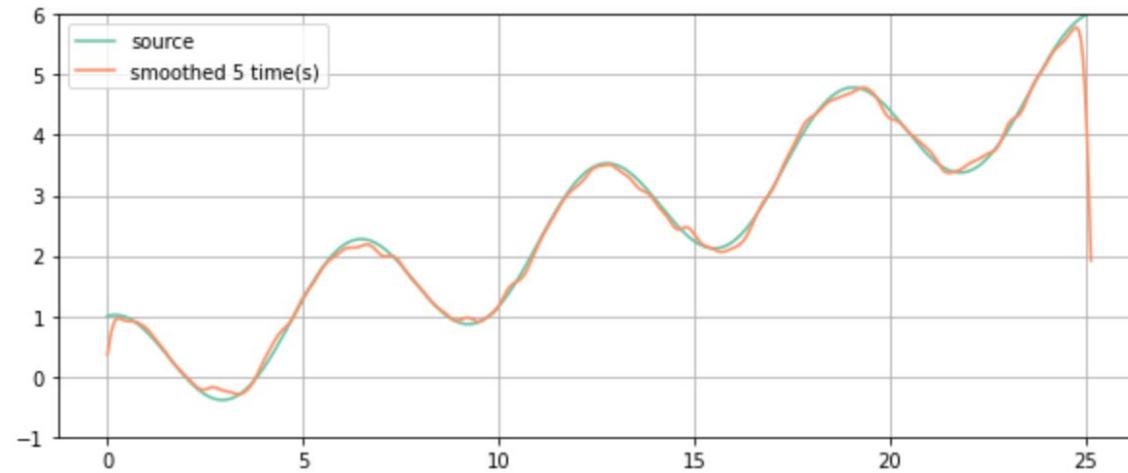
Пример

- Применим к "сглаживание" с помощью скользящего среднего
- Выберем размер окна; например, 5
- Каждое значение сигнала заменим средним по окрестности размера 5; можно повторить несколько раз



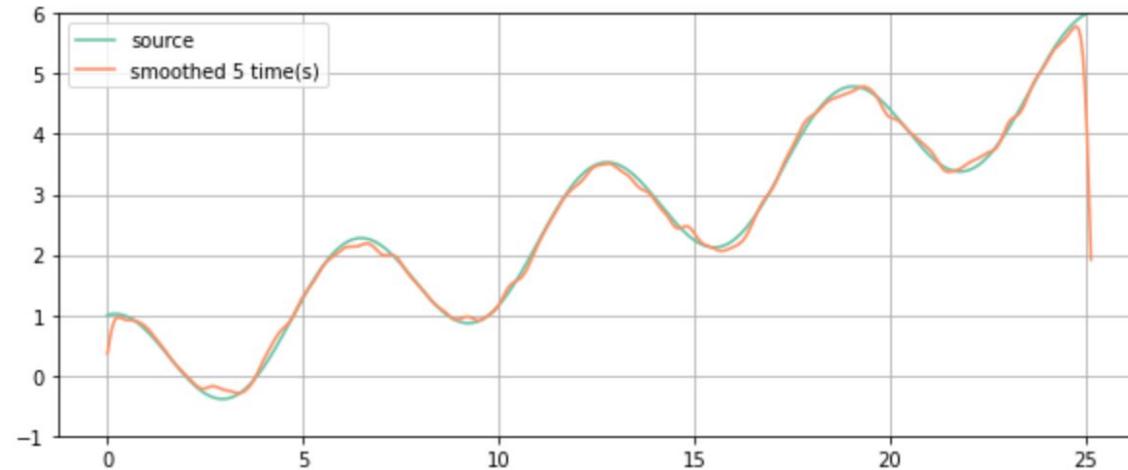
Пример

- Сигнал стал более "гладким"
- Появились краевые эффекты - почему?



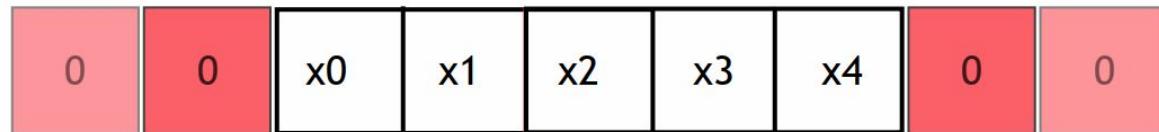
Пример

- Сигнал стал более "гладким"
- Появились краевые эффекты - почему?
 - Для первых и последних точек нет полного числа соседей, поэтому пришлось добавить нули по краям сигнала

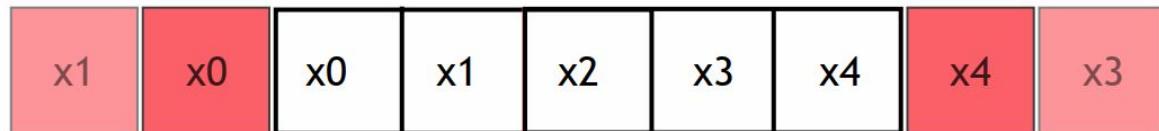


Padding

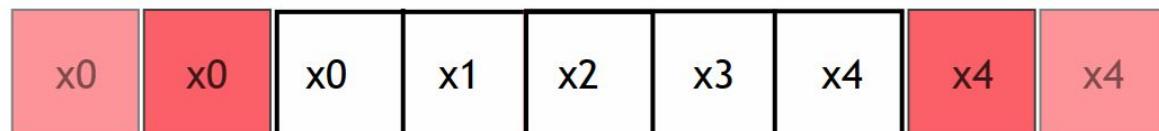
- Zero padding



- Reflect padding



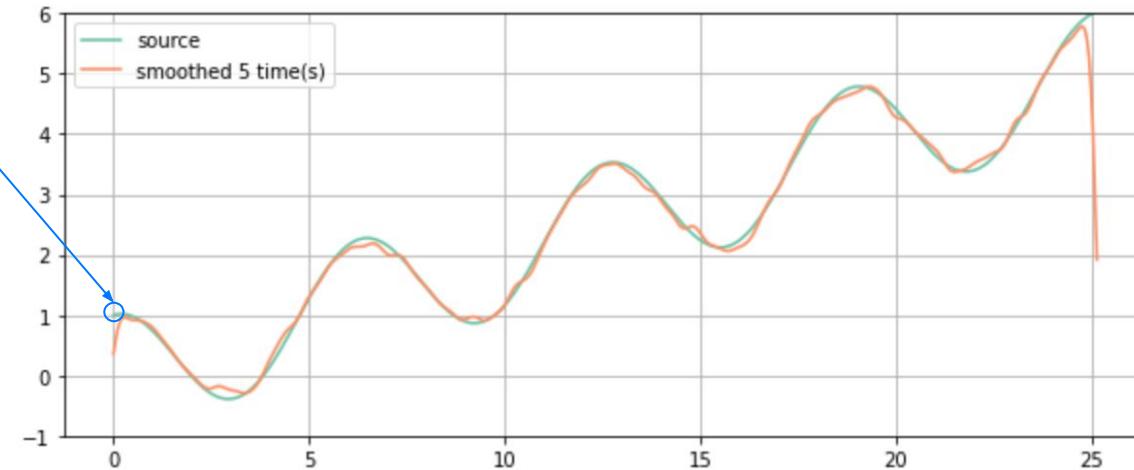
- Replicate padding



Пример

- Что именно мы сделали?

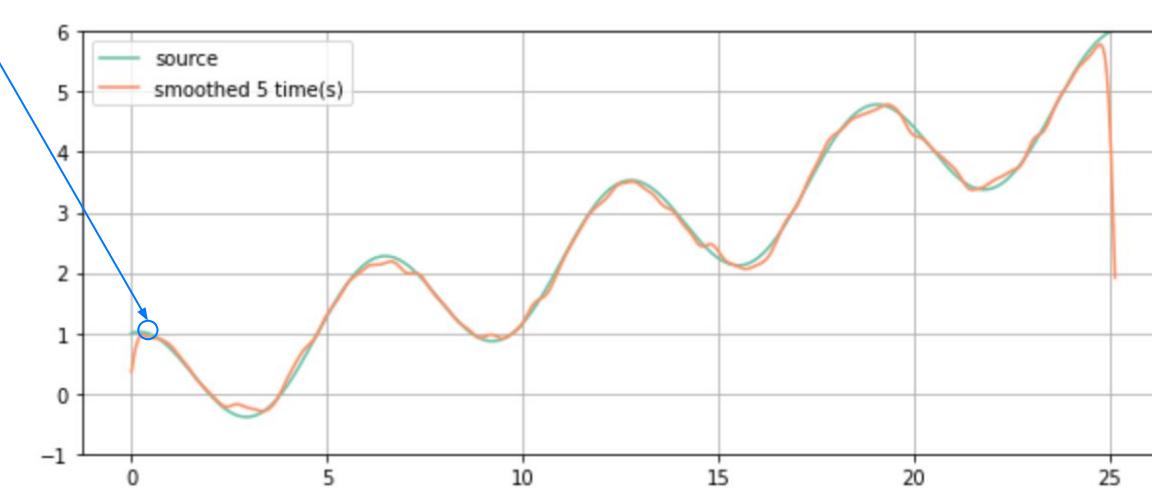
			1.15	0.71	1.17	1.40	1.00	1.41	0.79	0.75	0.97	0.54	0.76	1.19	1.40	0.58	0.95
$\frac{1}{5}$	$x \frac{1}{5}$	$x \frac{1}{5}$	$x \frac{1}{5}$	$x \frac{1}{5}$													



Пример

- Что именно мы сделали?

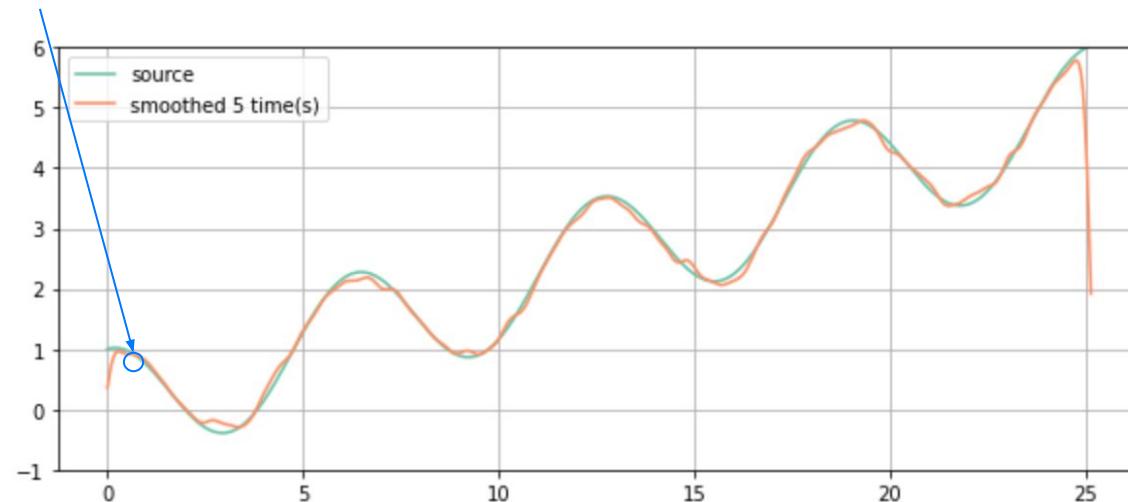
	1.15	0.71	1.17	1.40	1.00	1.41	0.79	0.75	0.97	0.54	0.76	1.19	1.40	0.58	0.95
$x^{\frac{1}{5}}$	$x^{\frac{1}{5}}$	$x^{\frac{1}{5}}$	$x^{\frac{1}{5}}$	$x^{\frac{1}{5}}$											



Пример

- Что именно мы сделали?

1.15 0.71 1.17 1.40 1.00 1.41 0.79 0.75 0.97 0.54 0.76 1.19 1.40 0.58 0.95
 $x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$

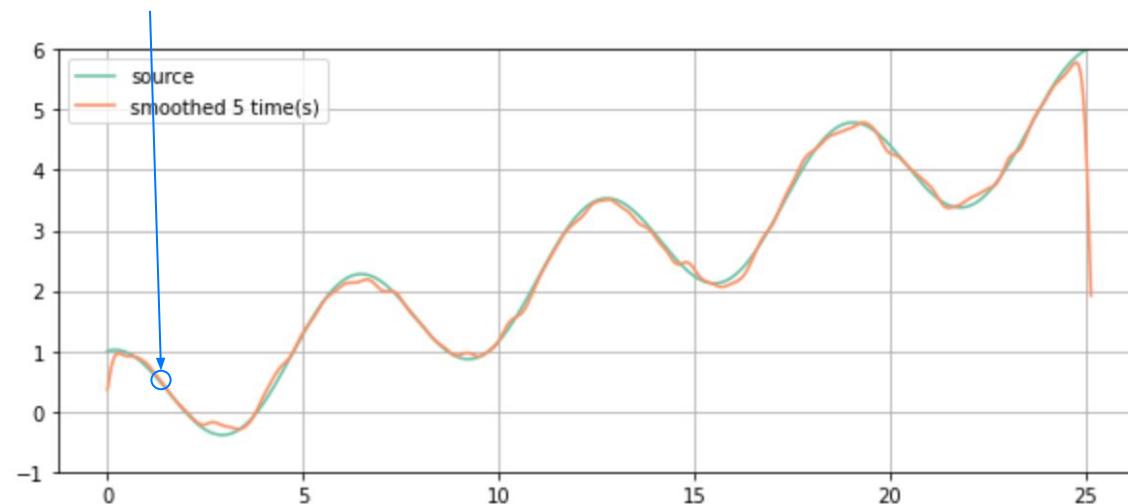


Пример

- Что именно мы сделали?

1.15 **0.71 1.17 1.40 1.00 1.41** 0.79 0.75 0.97 0.54 0.76 1.19 1.40 0.58 0.95

$x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$ $x^{\frac{1}{5}}$

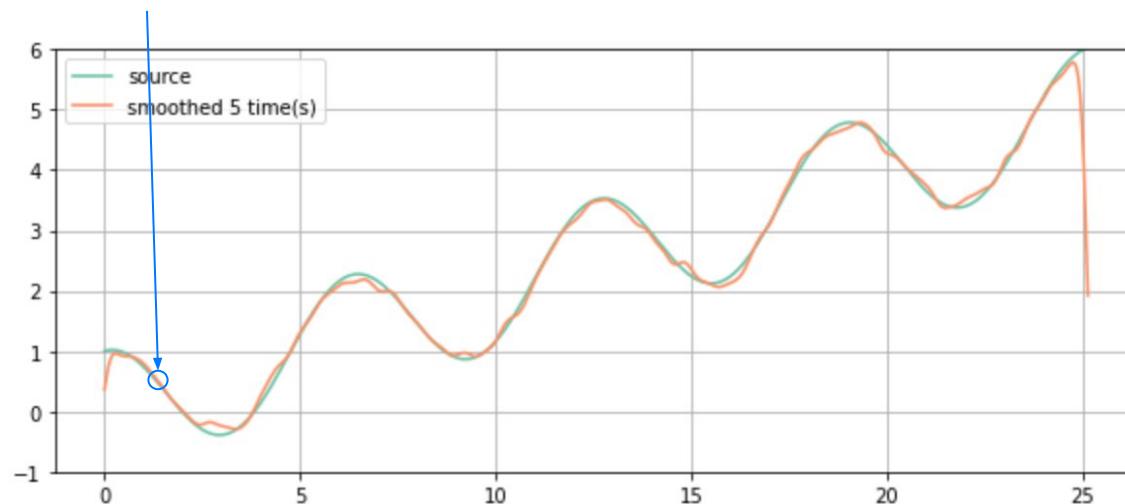


Пример

- Что именно мы сделали? Свертку с ядром $\left[\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right]$

1.15 0.71 1.17 1.40 1.00 1.41 0.79 0.75 0.97 0.54 0.76 1.19 1.40 0.58 0.95

$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$
---------------	---------------	---------------	---------------	---------------

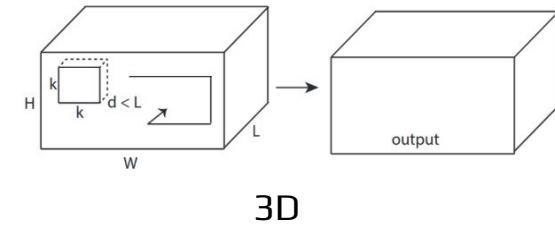
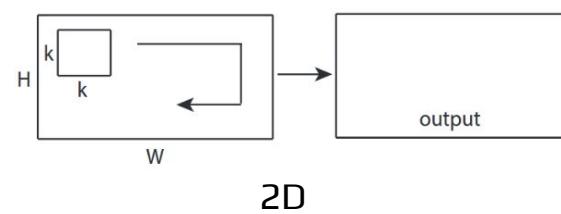
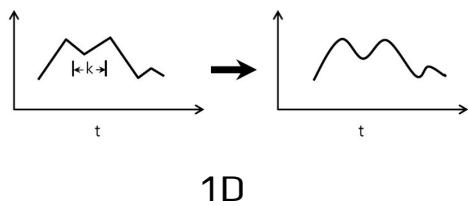


Свертка

- Свертка - операция, при которой
 - Ядро свертки "скользит" по входному сигналу
 - В каждой точке приложения ядра вычисляется скалярное произведение
 - Результат записывается в выходной сигнал
- Дискретная формула свертки (1D) сигнала y с ядром w размера K :
 - $$y_i = (x * w)_i = \sum_{k=0}^{K-1} x_{i+k} \cdot w_k$$

Свертка

- Сглаживание сигнала в прошлом примере = одномерная свертка
- Свертка может быть и более высокой размерности
- Размерность свертки = количество направлений, вдоль которых движется ядро свертки, проходя по сигналу



Свертка - двумерный случай

Входной сигнал X_{ij}

X00	X01	X02	X03
X10	X11	X12	X13
X20	X21	X22	X23
X30	X31	X32	X33

$$Y_{ij} = (X * W)_{ij} = \sum_{u=0}^{K-1} \sum_{v=0}^{K-1} X_{i+u, j+v} W_{uv}$$

Результат свертки Y_{ij}

Y00	Y01
Y10	Y11

Ядро свертки W
Размер **3x3**
Веса W_{ij}

Свертка

- 2D-свертки с разными ядрами используются в обработке изображений
 - Сглаживание
 - Выделение границ и углов
 - Удаление шумов
 - ...

Свертки для обработки изображений

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

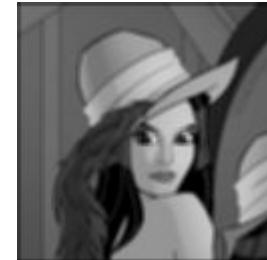


Смещение на 1
пиксель вправо



0	0	0
-1	0	1
0	0	0

???



Размытие окном
3x3

0	0	0
1	0	0
0	0	0



Выделение
вертикальных
границ

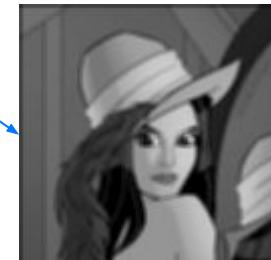
Свертки для обработки изображений

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



Смещение на 1
пиксель вправо

0	0	0
-1	0	1
0	0	0

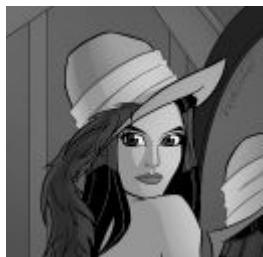


Размытие окном
3x3

0	0	0
1	0	0
0	0	0



Выделение
горизонтальных
границ



Свертки для обработки изображений

- Если во входном изображении больше 1 канала (например, RGB)?



Тензор $H \times W \times 3$

$$\begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$
$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$
$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

З **двумерных**
свертки



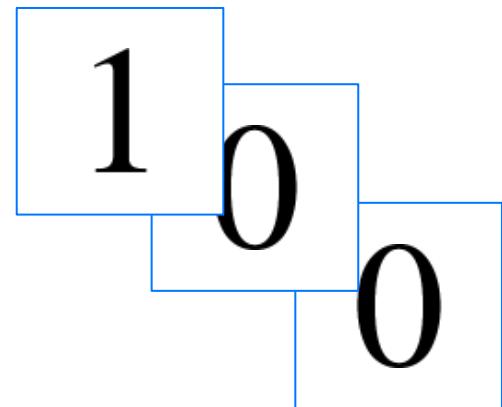
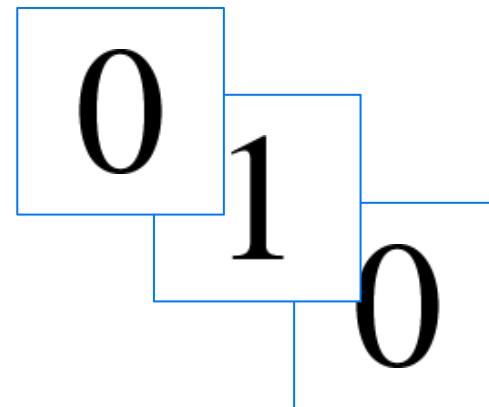
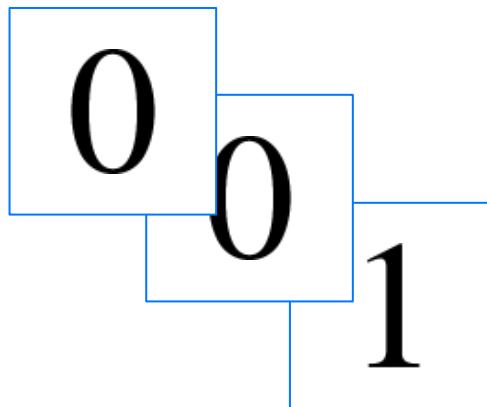
Тензор $H \times W \times 3$

Свертки для обработки изображений

- Как будет выглядеть набор ядер свертки для "обращения" порядка каналов изображения (RGB -> BGR)?

Свертки для обработки изображений

- Как будет выглядеть набор ядер свертки для "обращения" порядка каналов изображения (RGB -> BGR)?



Свертка - свойства

- Ассоциативность:
 - $a * (b * c) = (a * b) * c$
- Коммутативность:
 - $a * b = b * a$
- Линейность
 - $(a + b) * c = a * c + b * c$
 - $(ka * b) = k(a * b)$

Свертка

- На самом деле, рассмотренная операция называется не **сверткой, а кросс-корреляцией**
- Разница в том, что при свертке ядро должно транспонироваться

Свертка

- На самом деле, рассмотренная операция называется не **сверткой, а кросс-корреляцией**
- Разница в том, что при свертке ядро должно транспонироваться
- Для математиков различие играет роль
- Для обучателей нейросетей - нет

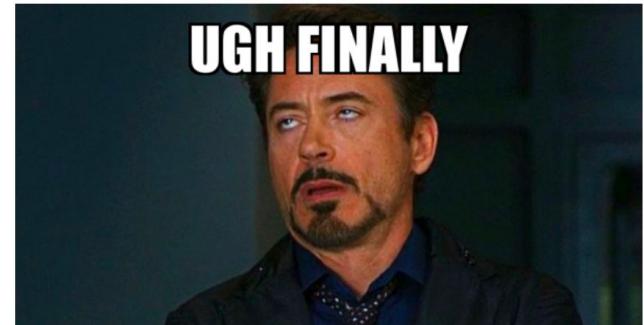


Свертка

- В области обработки изображений используются свертки с явно заданными ядрами
- Что получится, если сделать веса ядер свертки обучаемыми?

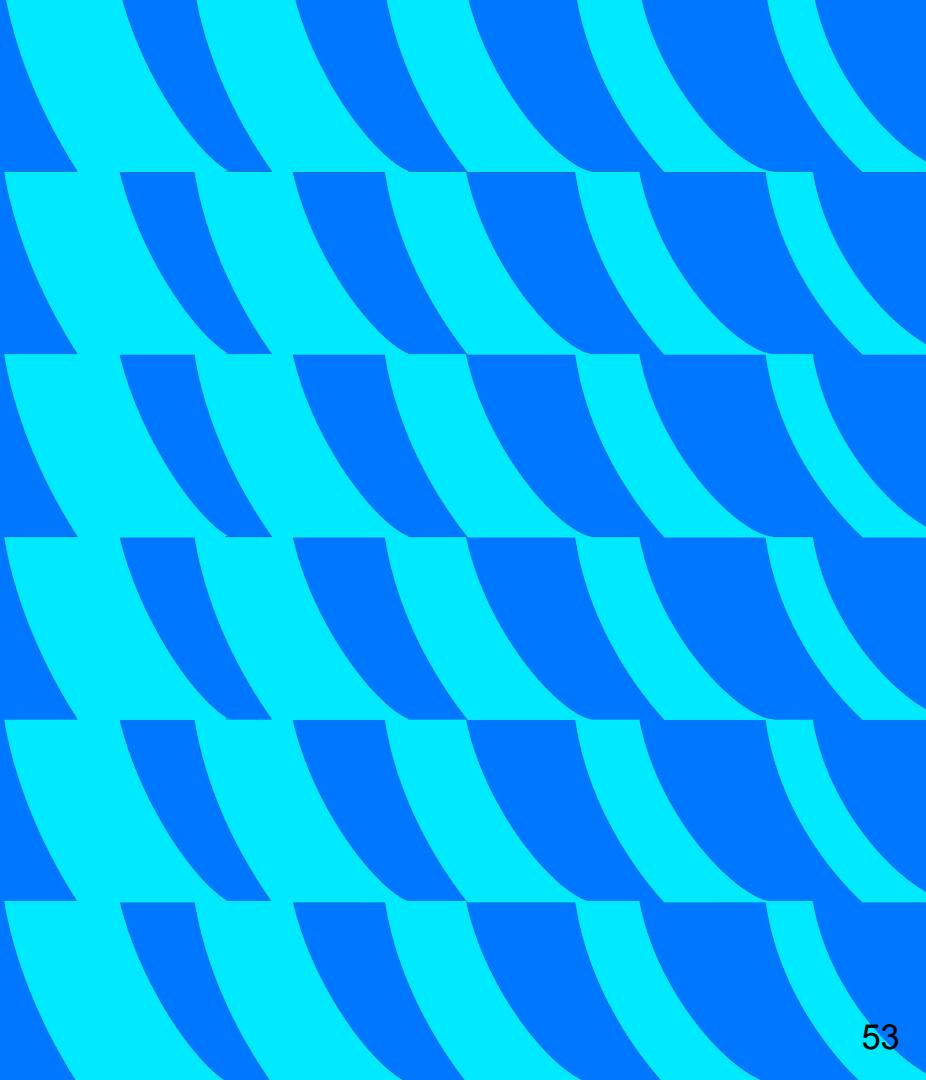
Свертка

- В области обработки изображений используются свертки с явно заданными ядрами
- Что получится, если сделать веса ядер свертки обучаемыми?
 - Сверточный слой



Сверточный слой

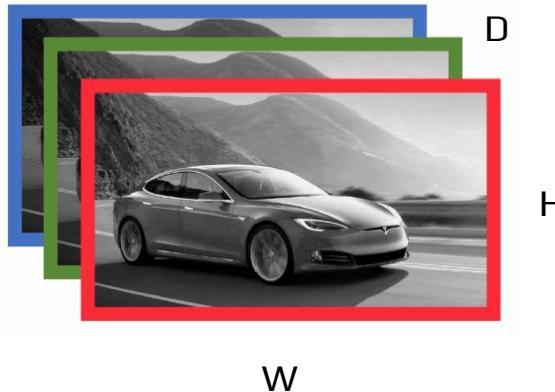
.....



Сверточный слой

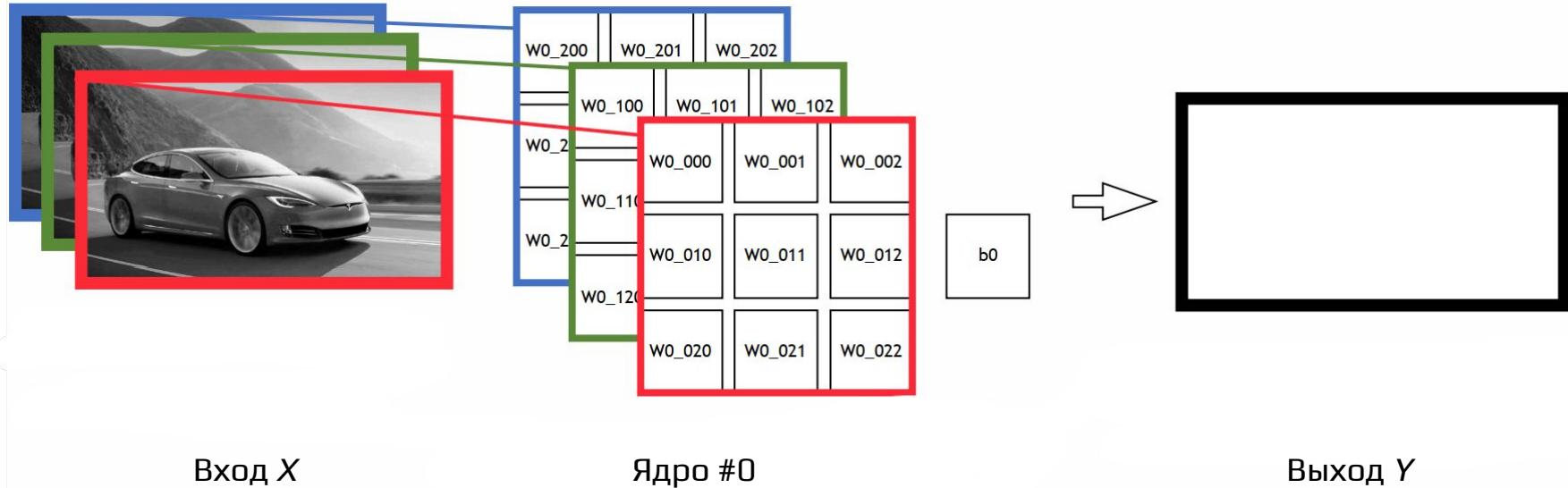
- Сверточный слой состоит из нескольких ($K \geq 1$) ядер свертки
 - К каждому результату свертки можно добавить bias
- На вход подается тензор X размера $H \times W \times D$
- Каждое из K ядер сворачивается с входным тензором
- В итоге получается новый тензор Y размера $H' \times W' \times K$
 - Применив padding, можно сделать $H = H'$, $W = W'$

Сверточный слой с *K* различных ядер

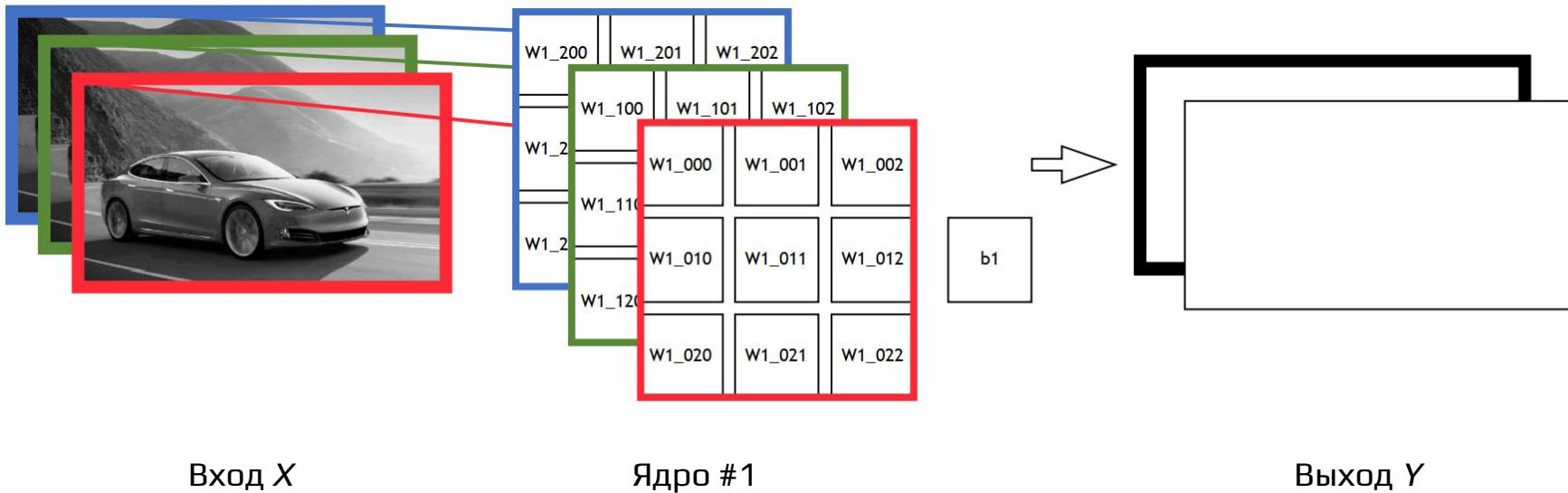


Вход X

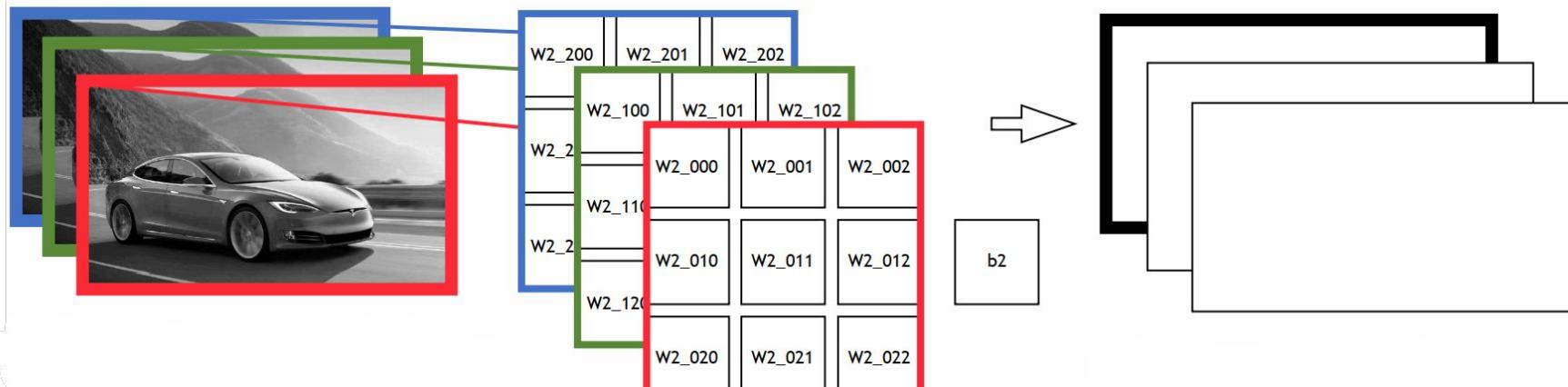
Сверточный слой с К различных ядер



Сверточный слой с К различных ядер



Сверточный слой с К различных ядер

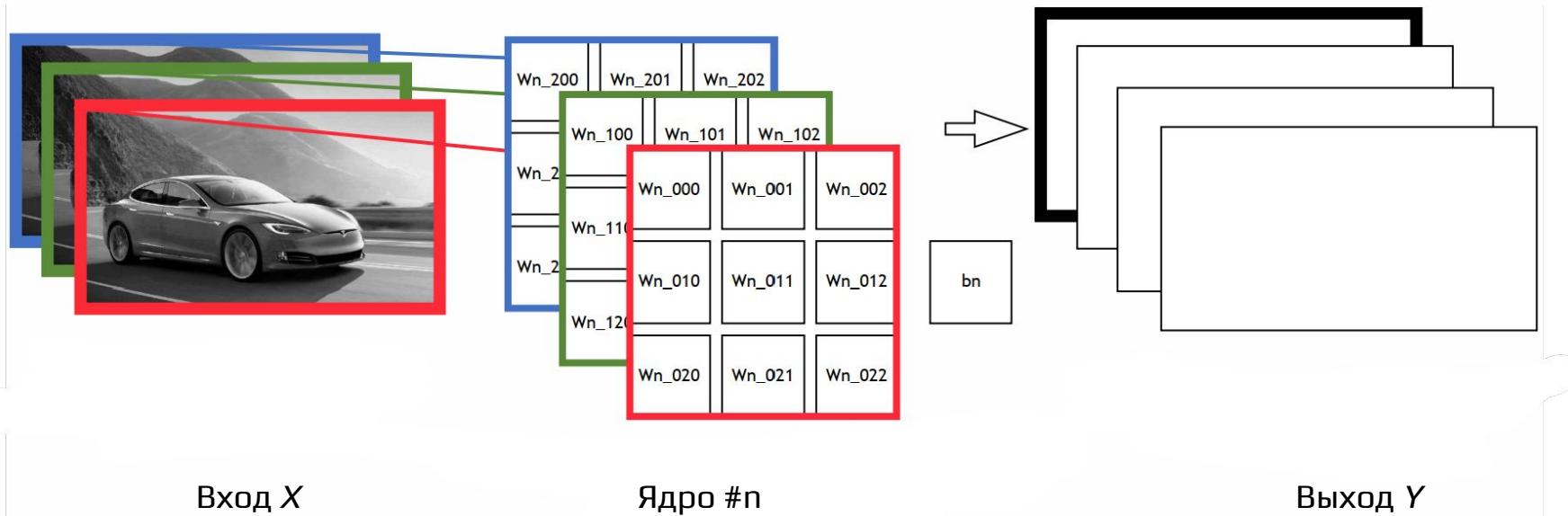


Вход X

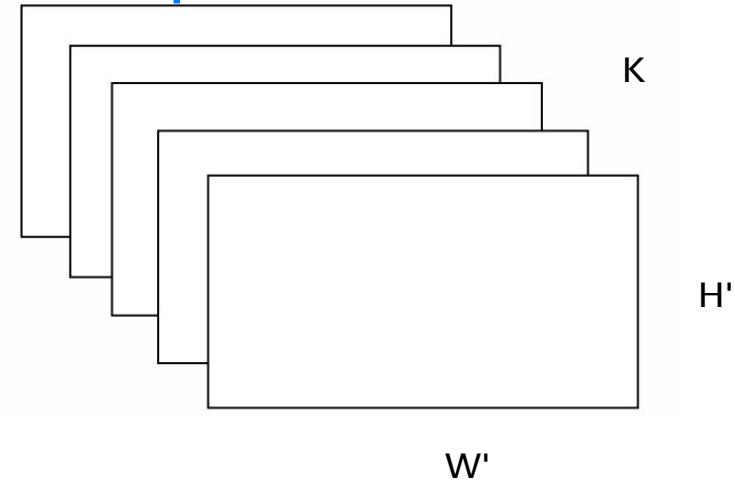
Ядро #2

Выход Y

Сверточный слой с K различных ядер



Сверточный слой с К различных ядер



Вход X

Выход Y

Сверточный слой

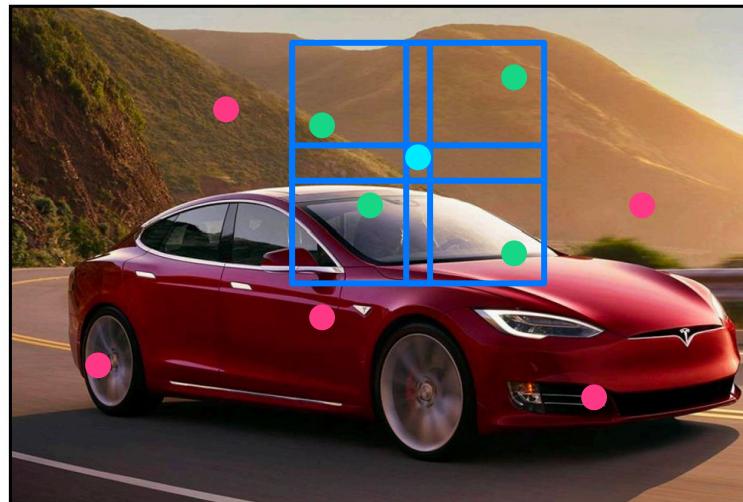
- Сколько всего обучаемых весов у сверточного слоя, если
 - размер входного тензора $H \times W \times D$
 - количество ядер K
 - размер ядра 3×3
 - используется $bias$?

Сверточный слой

- Сколько всего обучаемых весов у сверточного слоя, если
 - размер входного тензора $H \times W \times D$
 - количество ядер K
 - размер ядра 3×3
 - используется $bias$?
- Число весов на одно ядро свертки: $3 \times 3 \times D$ (weights) + 1 (bias)
- Всего ядер $K \Rightarrow$ весов $K \times (3 \times 3 \times D + 1)$

Сверточный слой - отличия от полносвязного

- Сверточный слой обрабатывает входной сигнал локально
 - "Взаимодействуют" только те ячейки, которые находятся рядом (попадают в одно окно свертки)



Сверточный слой - отличия от полносвязного

- Сверточный слой позволяет сделать сеть инвариантной к смещениям объектов
 - Ядра сверток применяются одинаковым образом ко всей "площади" входного тензора (по умолчанию)



Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

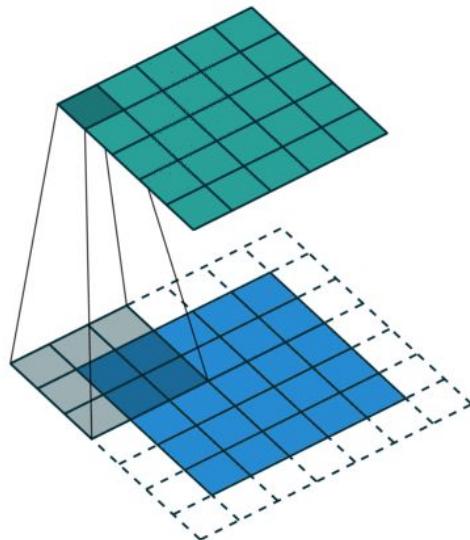
Сверточный слой - параметры

- **Число ядер (оно же: "число фильтров", "ширина слоя")**
- **Размер ядра (3×3 , 5×5 , 3×5 , ...)**
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

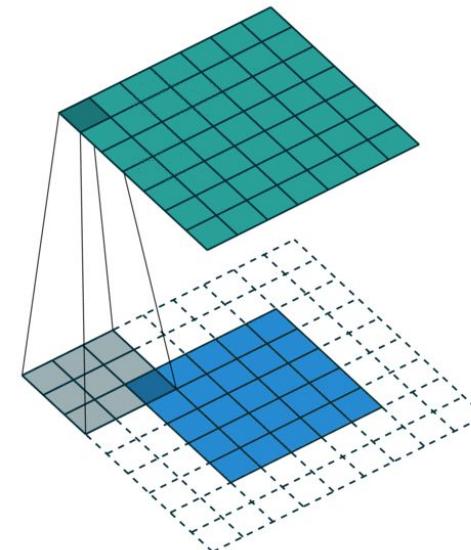
Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- **Padding (добавление значений по краям входного сигнала)**
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

Сверточный слой - Padding



Размер ядра 3x3
Padding = 1

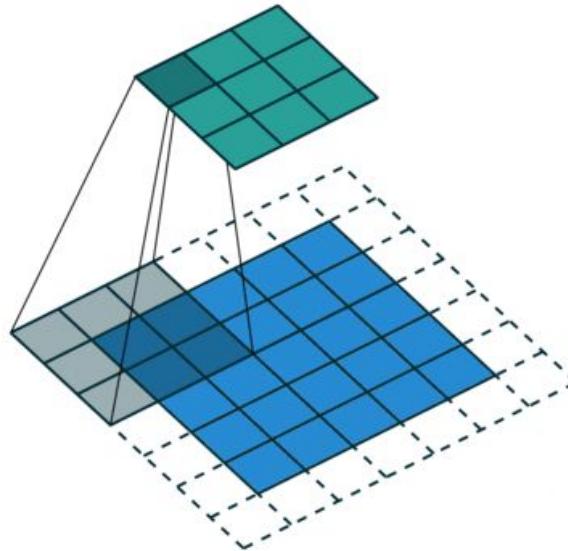


Размер ядра 3x3
Padding = 2

Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- **Stride (величина шага скользящего окна, бывает и >1)**
- Dilation (добавление "разреженности" в матрицу ядра свертки)

Сверточный слой - Stride



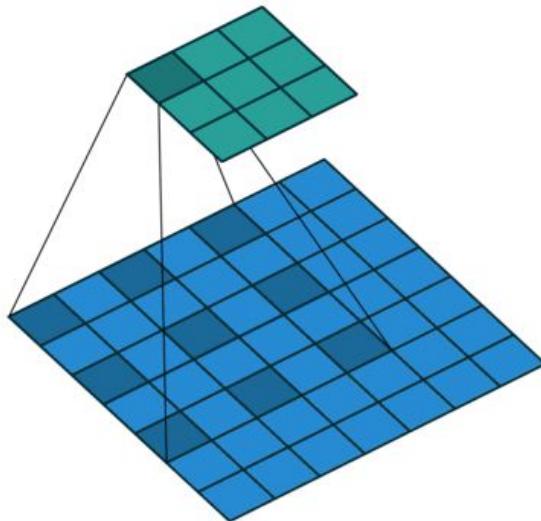
Размер ядра 3×3
Padding = 1
Stride = 2

https://github.com/vdumoulin/conv_arithmetic

Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- **Dilation (добавление "разреженности" в матрицу ядра свертки)**

Сверточный слой - Dilation



Размер ядра 3x3
Padding = 0
Dilation = 2

https://github.com/vdumoulin/conv_arithmetic

Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

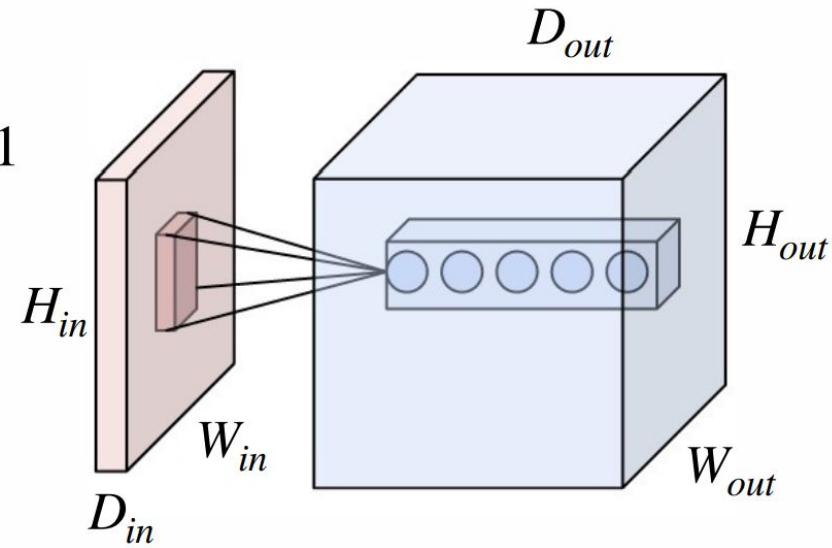
Влияют на H/W выходного тензора

Сверточный слой - размер выхода

Размер выходного тензора: $D_{out} \times H_{out} \times W_{out}$:

$$H_{out} = \frac{(H_{in} - size + 2 \times padding)}{stride} + 1$$

$$W_{out} = \frac{(W_{in} - size + 2 \times padding)}{stride} + 1$$



Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

Влияют на H/W выходного тензора

Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

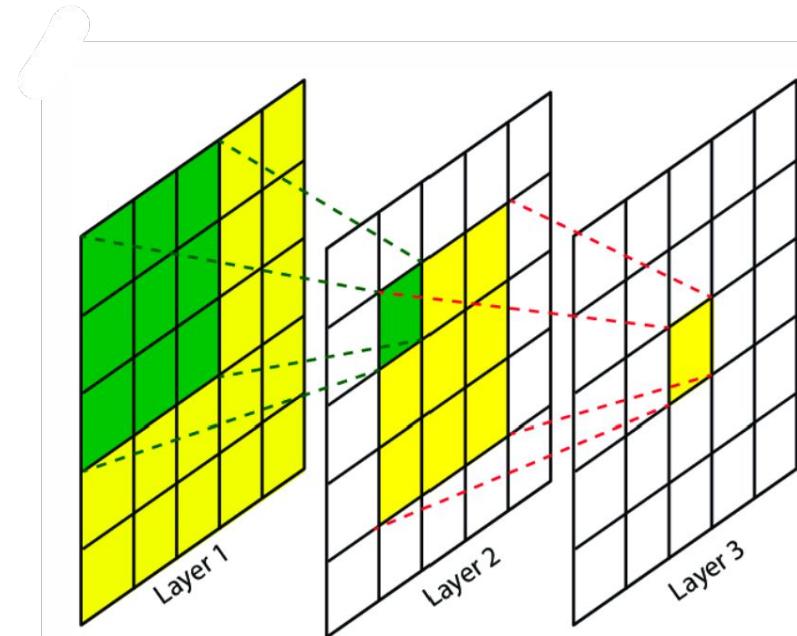
Влияют на H/W выходного тензора

Определяют рецептивное поле ячеек
выходного тензора (нейронов)

Сверточный слой - рецептивное поле

Рецептивное поле (receptive field)

нейрона - размер области исходного сигнала (изображения), которая может вносить вклад в активацию данного нейрона



Рецептивное поле нейрона на Layer 2: 3×3

Рецептивное поле нейрона на Layer 3: 5×5

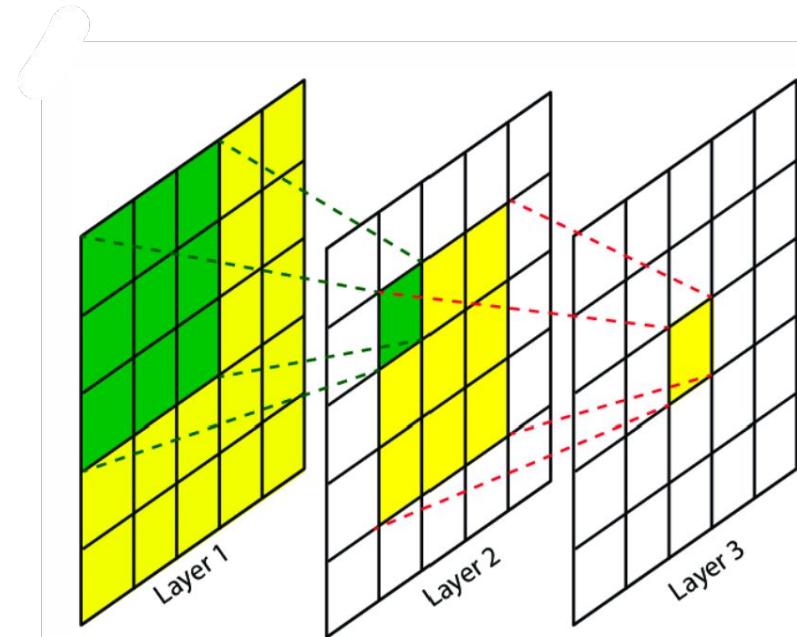
Сверточный слой - рецептивное поле

Рецептивное поле (receptive field)

нейрона - размер области исходного сигнала (изображения), которая может вносить вклад в активацию данного нейрона

Как меняется РП выхода слоя при:

- Увеличении ядер сверток?



Рецептивное поле нейрона на Layer 2: 3×3

Рецептивное поле нейрона на Layer 3: 5×5

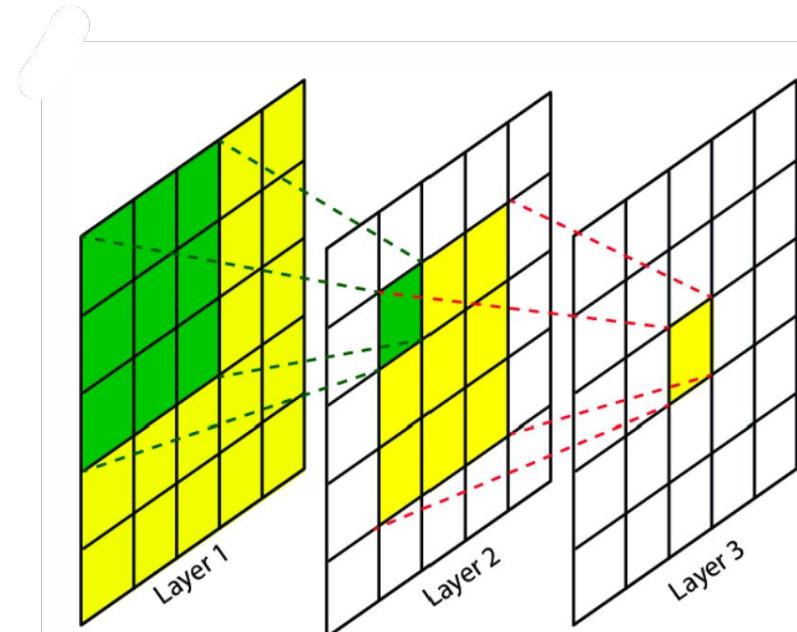
Сверточный слой - рецептивное поле

Рецептивное поле (receptive field)

нейрона - размер области исходного сигнала (изображения), которая может вносить вклад в активацию данного нейрона

Как меняется РП выхода слоя при:

- Увеличении ядер сверток? Растет
- Увеличении dilation?



Рецептивное поле нейрона на Layer 2: 3×3

Рецептивное поле нейрона на Layer 3: 5×5

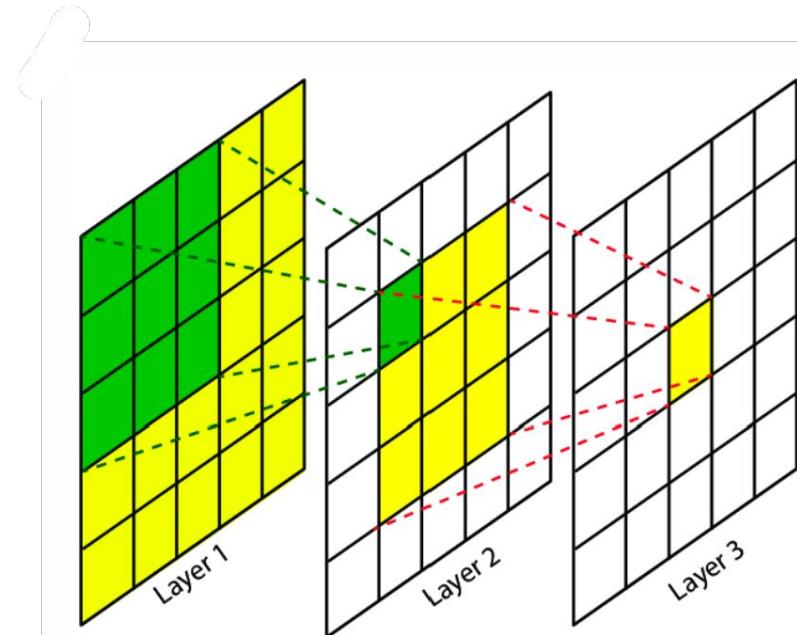
Сверточный слой - рецептивное поле

Рецептивное поле (receptive field)

нейрона - размер области исходного сигнала (изображения), которая может вносить вклад в активацию данного нейрона

Как меняется РП выхода слоя при:

- Увеличении ядер сверток? Растет
- Увеличении dilation? Растет
- Добавлении padding?



Рецептивное поле нейрона на Layer 2: 3x3

Рецептивное поле нейрона на Layer 3: 5x5

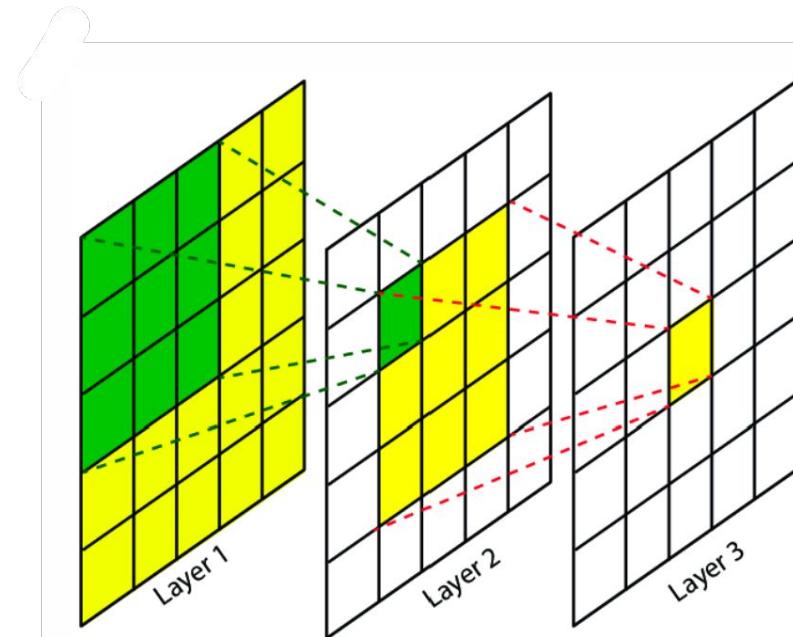
Сверточный слой - рецептивное поле

Рецептивное поле (receptive field)

нейрона - размер области исходного сигнала (изображения), которая может вносить вклад в активацию данного нейрона

Как меняется РП выхода слоя при:

- Увеличении ядер сверток? Растет
- Увеличении dilation? Растет
- Добавлении padding? Никак



Рецептивное поле нейрона на Layer 2: 3x3

Рецептивное поле нейрона на Layer 3: 5x5

Сверточный слой - параметры

- Число ядер (оно же: "число фильтров", "ширина слоя")
- Размер ядра (3×3 , 5×5 , 3×5 , ...)
- Padding (добавление значений по краям входного сигнала)
- Stride (величина шага скользящего окна, бывает и >1)
- Dilation (добавление "разреженности" в матрицу ядра свертки)

Влияют на H/W выходного тензора

Определяют рецептивное поле ячеек
выходного тензора (нейронов)

Сверточный слой

```
import torch
```

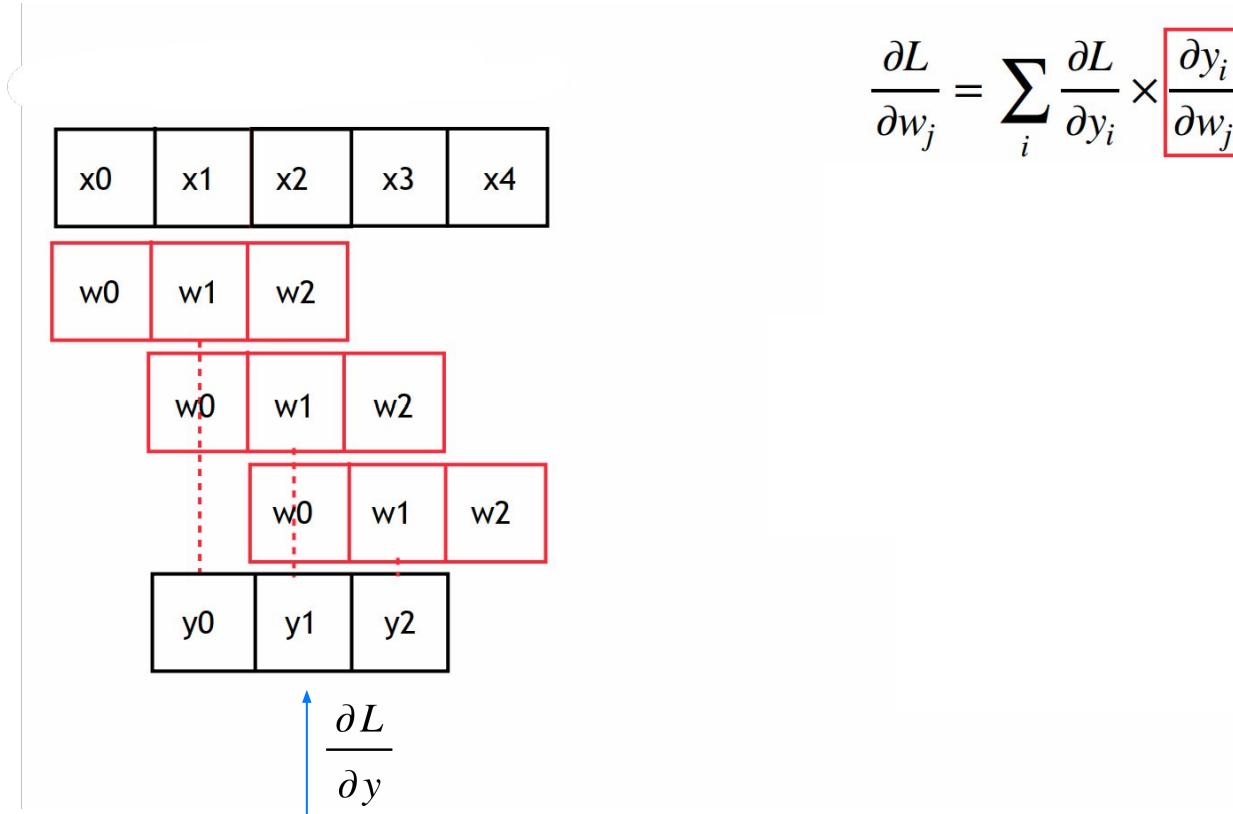
```
conv_layer = torch.nn.Conv2d()
```

Init signature:
torch.nn.Conv2d(
 in_channels,
 out_channels,
 kernel_size,
 stride=1,
 padding=0,
 dilation=1,
 groups=1,
 bias=True,
 padding_mode='zeros',
)

Сверточный слой

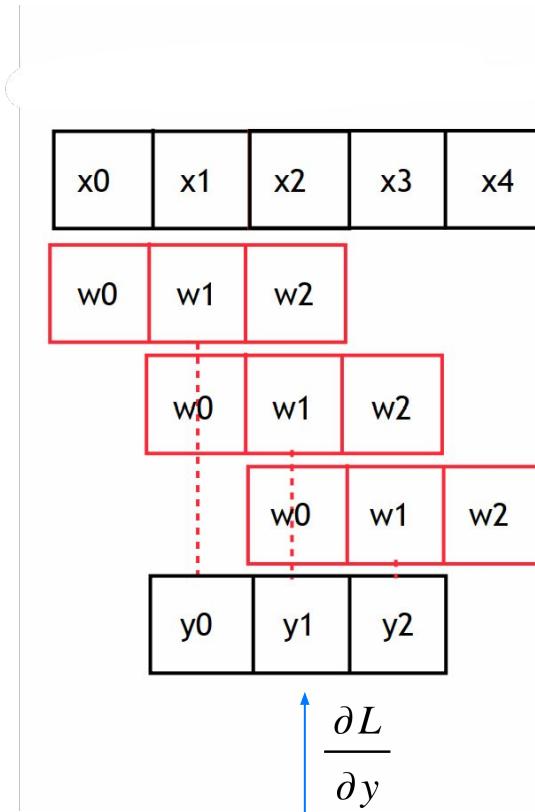
- Forward pass разобрали
- Разберем теперь backward pass

Сверточный слой - backprop



$$\frac{\partial L}{\partial w_j} = \sum_i \frac{\partial L}{\partial y_i} \times \boxed{\frac{\partial y_i}{\partial w_j}}$$

Сверточный слой - backprop



$$\frac{\partial L}{\partial w_j} = \sum_i \frac{\partial L}{\partial y_i} \times \boxed{\frac{\partial y_i}{\partial w_j}}$$

$$y_i = (x * w)_i = \sum_{k=0}^{K-1} x_{i+k} w_k$$

Сверточный слой - backprop

x0	x1	x2	x3	x4
----	----	----	----	----

w0	w1	w2
----	----	----

w0	w1	w2
----	----	----

w0	w1	w2
----	----	----

y0	y1	y2
----	----	----

$$\frac{\partial L}{\partial y}$$

$$\frac{\partial L}{\partial w_j} = \sum_i \frac{\partial L}{\partial y_i} \times \boxed{\frac{\partial y_i}{\partial w_j}}$$

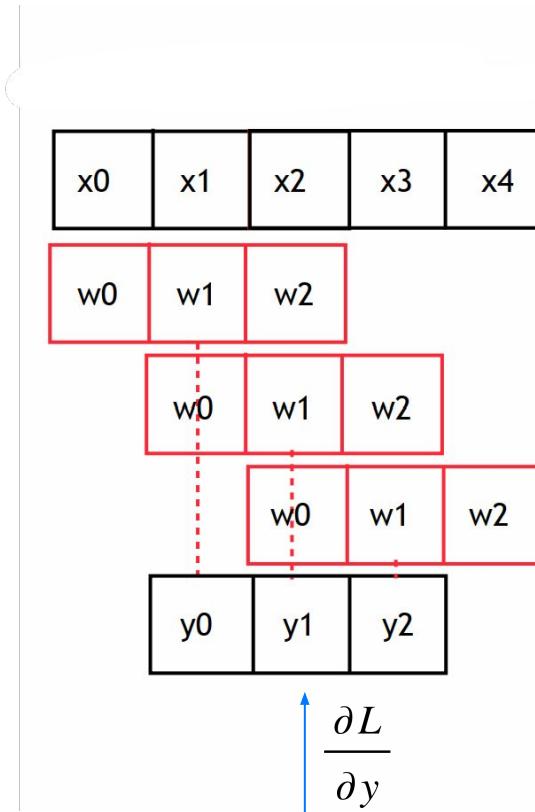
$$y_i = (x * w)_i = \sum_{k=0}^{K-1} x_{i+k} w_k$$

$$y_0 = x_0 \times w_0 + x_1 \times w_1 + x_2 \times w_2$$

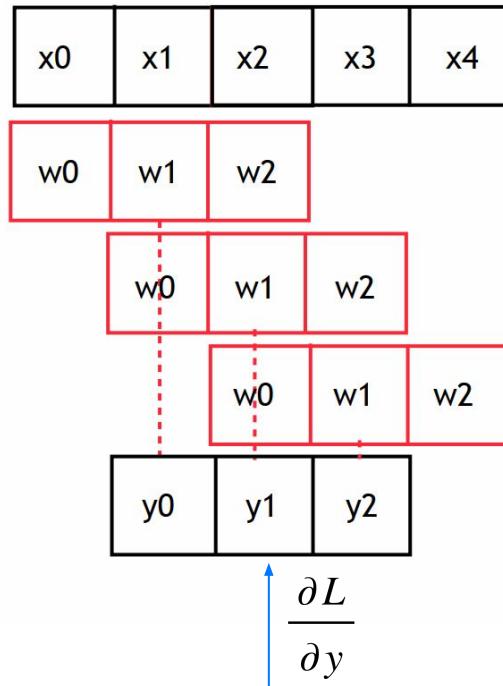
$$y_1 = x_1 \times w_0 + x_2 \times w_1 + x_3 \times w_2$$

$$y_2 = x_2 \times w_0 + x_3 \times w_1 + x_4 \times w_2$$

Сверточный слой - backprop



Сверточный слой - backprop



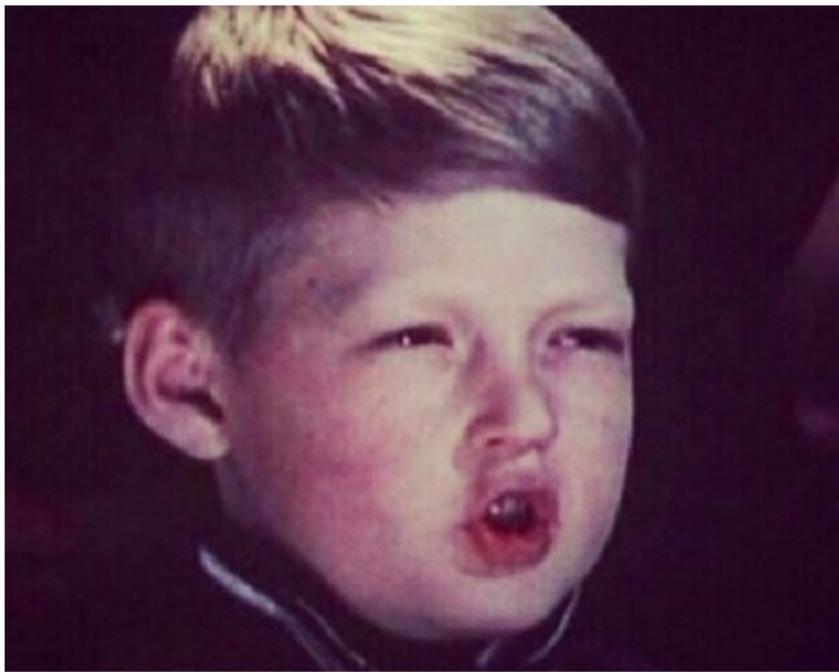
$$\frac{\partial L}{\partial w_j} = \sum_i \frac{\partial L}{\partial y_i} \times \boxed{\frac{\partial y_i}{\partial w_j}}$$

$$y_i = (x * w)_i = \sum_{k=0}^{K-1} x_{i+k} w_k$$

$$\frac{\partial y_i}{\partial w_j} = x_{i+j}$$

$$\frac{\partial L}{\partial w_j} = \sum_i \frac{\partial L}{\partial y_i} \times x_{i+j}$$

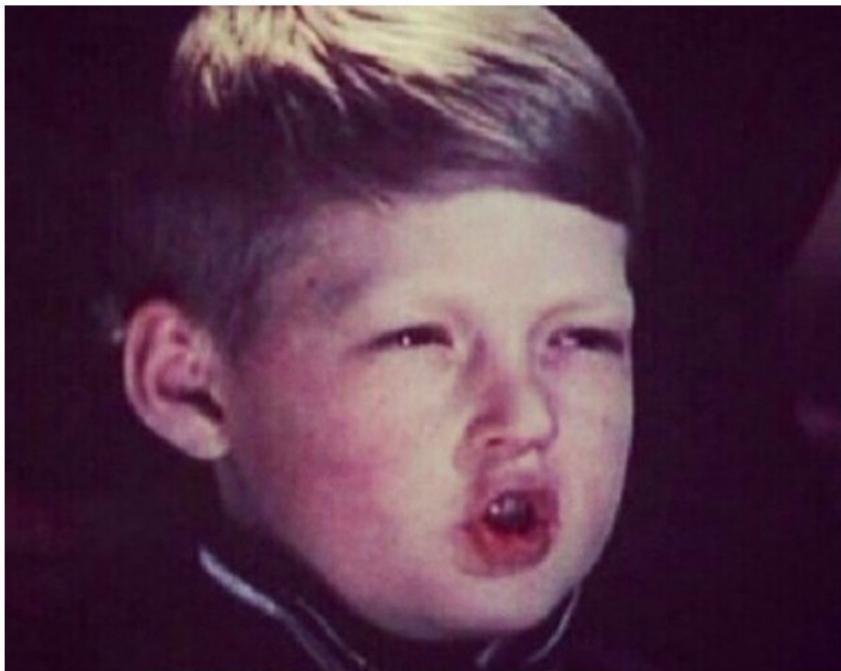
Сверточный слой - backprop



$$y_i = (x * w)_i = \sum_{k=0}^{K-1} x_{i+k} w_k$$

$$\frac{\partial L}{\partial w_j} = \sum_i \frac{\partial L}{\partial y_i} \times x_{i+j}$$

Сверточный слой - backprop



$$\frac{\partial L}{\partial w_j} = (x * \frac{\partial L}{\partial y})_j$$

Сверточный слой - backprop

- Градиент по весам сверточного слоя = свертка входного сигнала с градиентом по выходу слоя (в одномерном случае)
- Еще можно посмотреть, например, [тут](#)

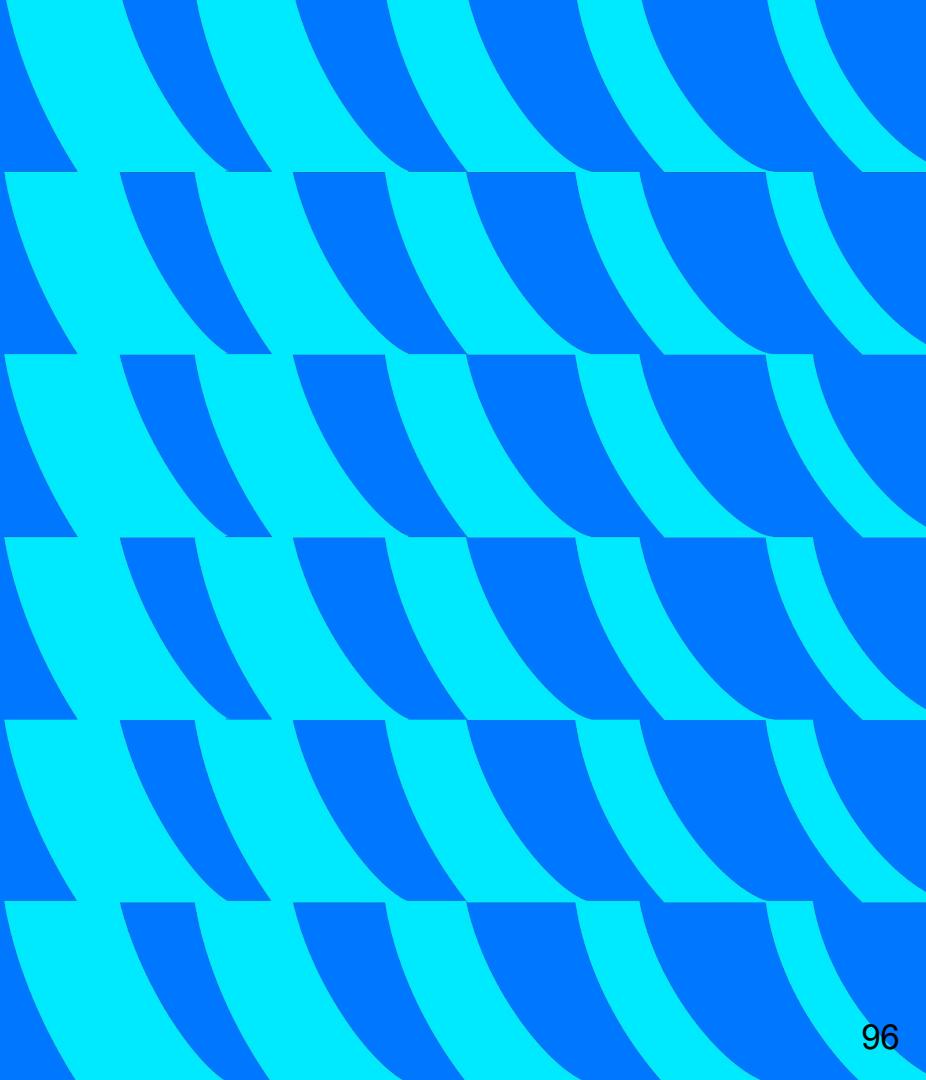
Сверточный слой

- Чтобы получить сверточную нейросеть, помимо сверточного слоя понадобятся:
 - Слои активации (Tanh , ReLU , ...)
 - Слои пулинга
 - (опционально) Полносвязные выходные слои

Сверточный слой

- Чтобы получить сверточную нейросеть, помимо сверточного слоя понадобятся:
 - Слои активации (Tanh , ReLU , ...)
 - **Слои пулинга**
 - (опционально) Полносвязные выходные слои

Пулинг



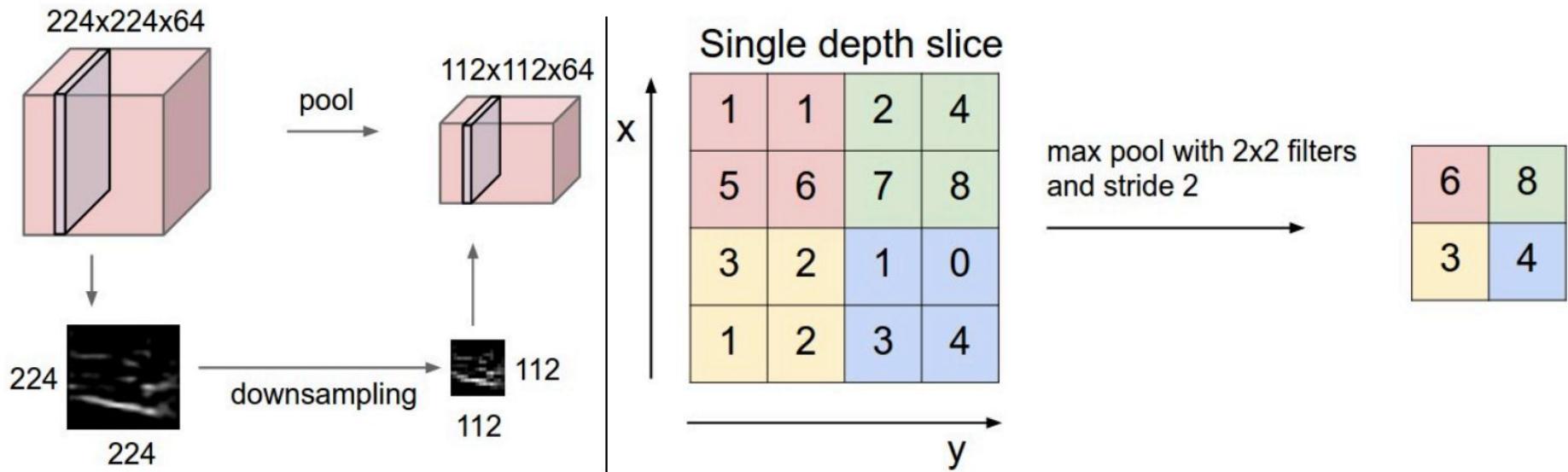
Проблема памяти

- Обычно ширина сверточных слоев растет по мере увеличения глубины сети
 - Например, в сети ResNet18 количество сверток в одном слое растет так: 64 - 128 - 256 - 512
- Чем больше размер тензоров, тем больше потребление памяти
- Слой Pooling позволяет уменьшать H/W тензоров

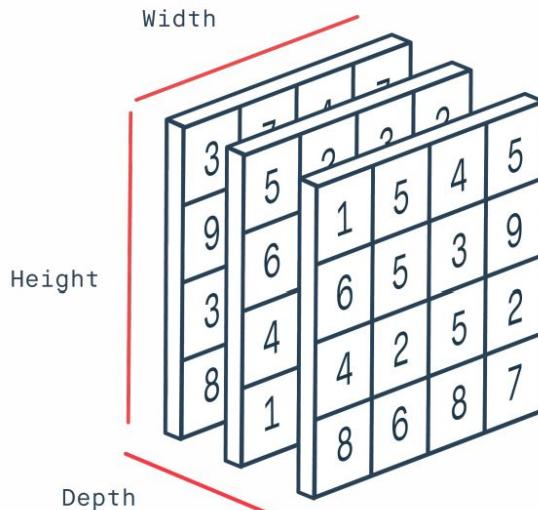
Pooling

- Разобьем тензор на части по ширине и высоте
- В каждой части независимо от других посчитаем 1 статистику (среднее - AveragePooling, максимум - MaxPooling)
- Склейм полученные статистики обратно в тензор
- Получится тензор прежней глубины, но меньшего размера по ширине и высоте

Pooling

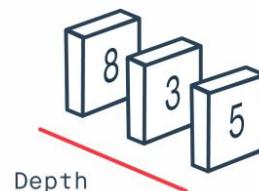


Global Pooling



Height x Width x Depth

- Можно посчитать статистики сразу по всему каналу, а не по регионам
- В результате получится тензор размера $1 \times 1 \times D$, который можно отправить в полносвязный слой
- Это будет **Global Pooling**



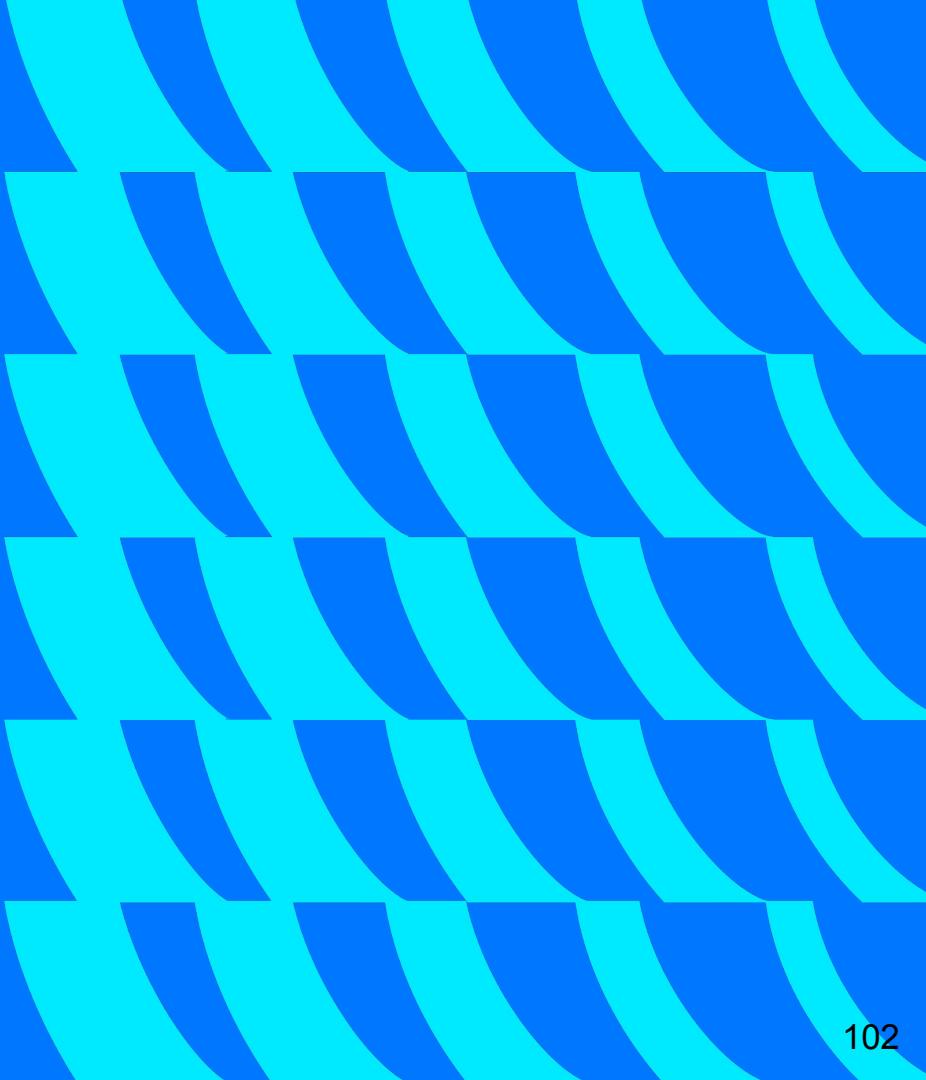
1 x 1 x Depth

Сверточный слой

- Чтобы получить сверточную нейросеть, помимо сверточного слоя понадобятся:
 - Слои активации (Tanh , ReLU , ...)
 - Слои пулинга
 - (опционально) Полносвязные выходные слои

CNN

.....



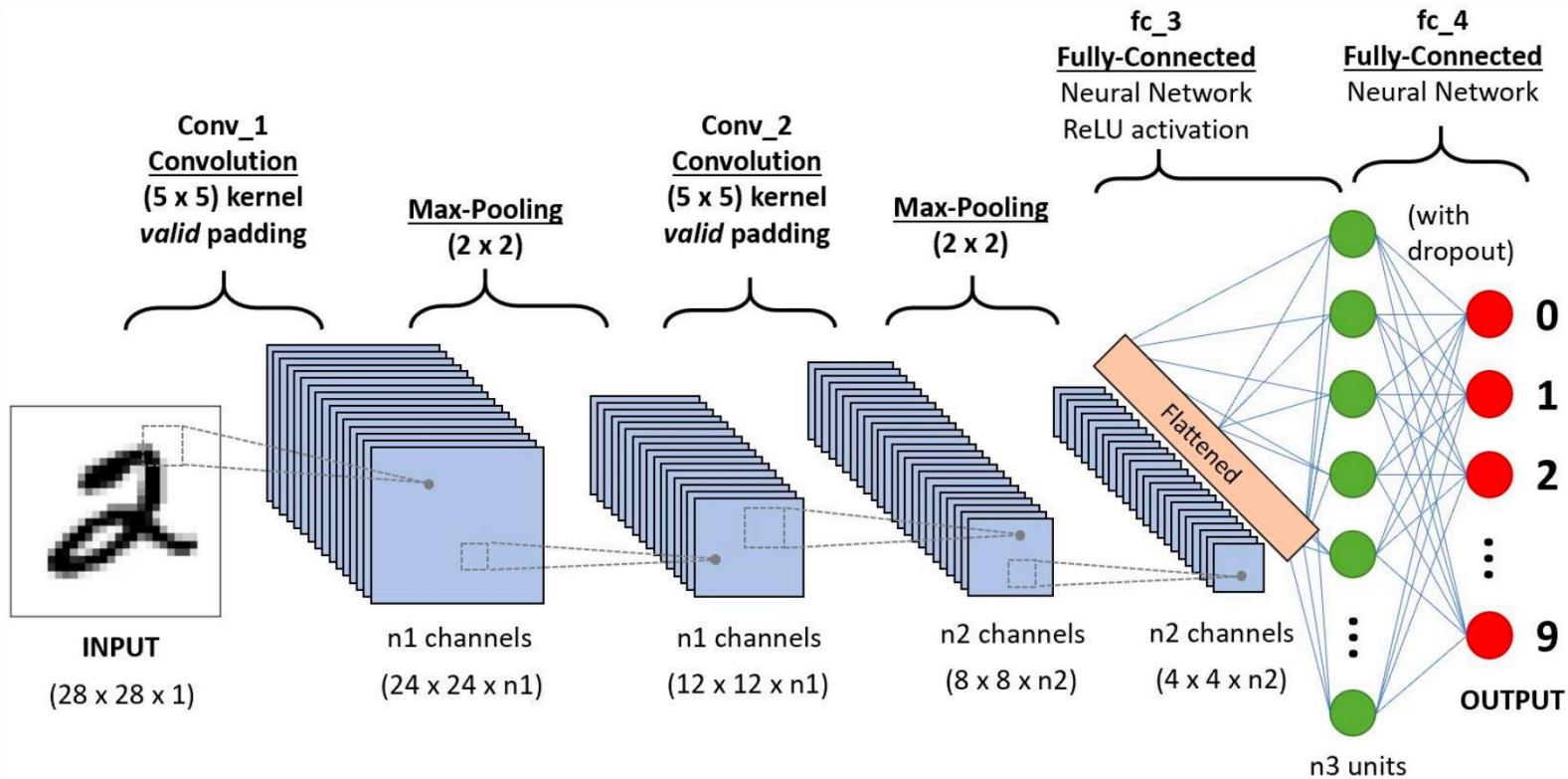
CNN

- CNN = convolutional neural network
- Типичная структура CNN:

((сверточный слой \Rightarrow активация) * k \Rightarrow пулинг) * m \Rightarrow линейный слой

- В современных архитектурах как правило используются свертки 3x3

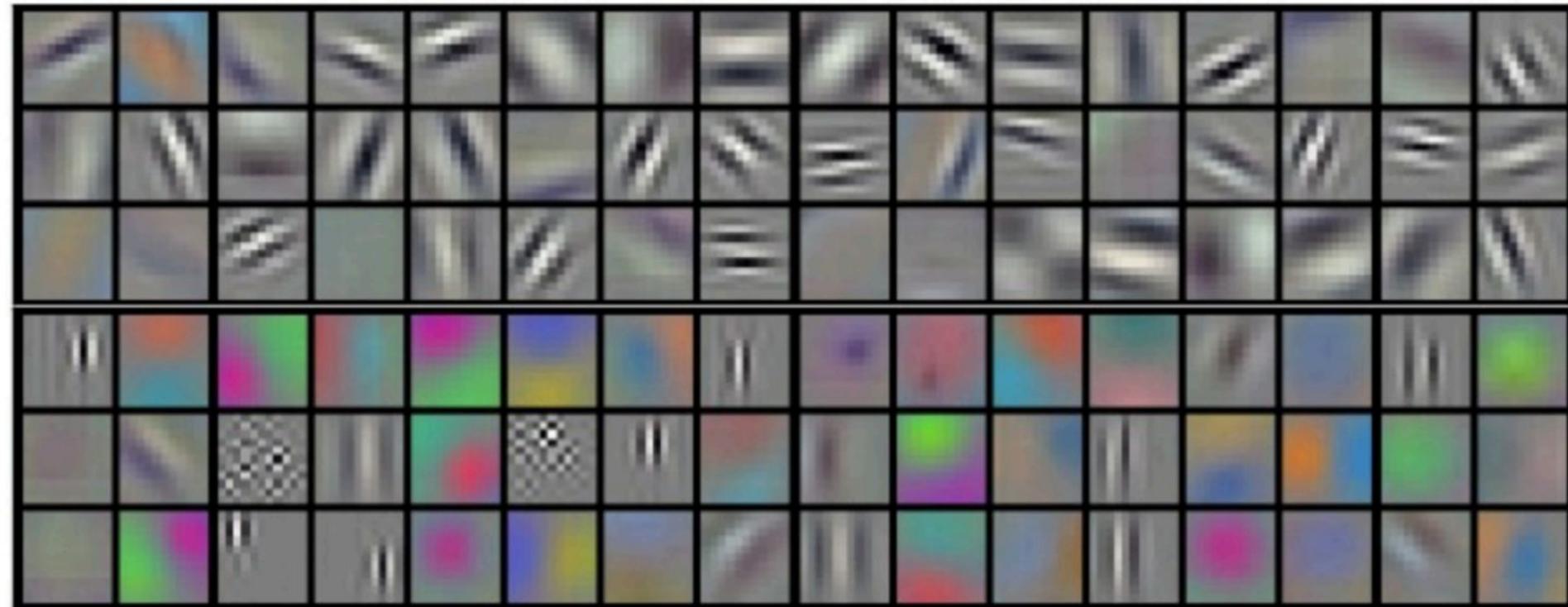
CNN - LeNet5



CNN

- Первый сверточный слой видит "сырые" данные
 - Его веса худо-бедно "понятны"
- Чем глубже в сеть, тем более абстрактным становится представление исходного объекта

Визуализация ядер первого сверточного слоя



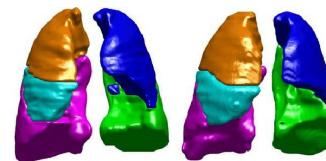
<https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

CNN

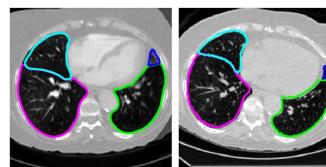
- Сверточные сети хорошо работают на данных с "пространственными размерностями"
 - Звук (1d)
 - Изображения (2d)
 - Сканы КТ (3d)
 - ...



Cat



Dog



CNN

- Применительно к изображениям, CNN хорошо работают в задачах:
 - Классификации / Регрессии
 - Детектирования
 - Сегментации
 - Генерации

CNN - Object Detection



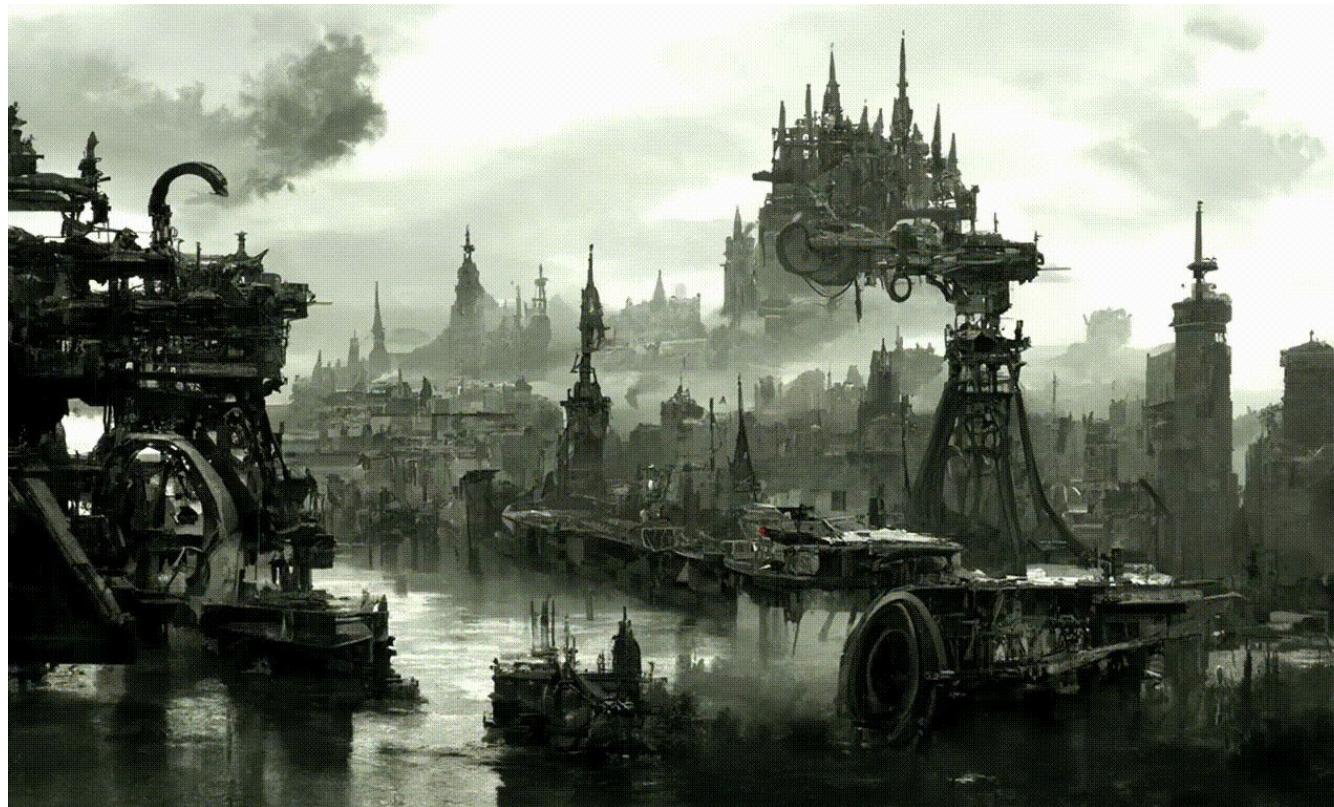
CNN - Instance Segmentation



CNN - Generative models



CNN - Generative models

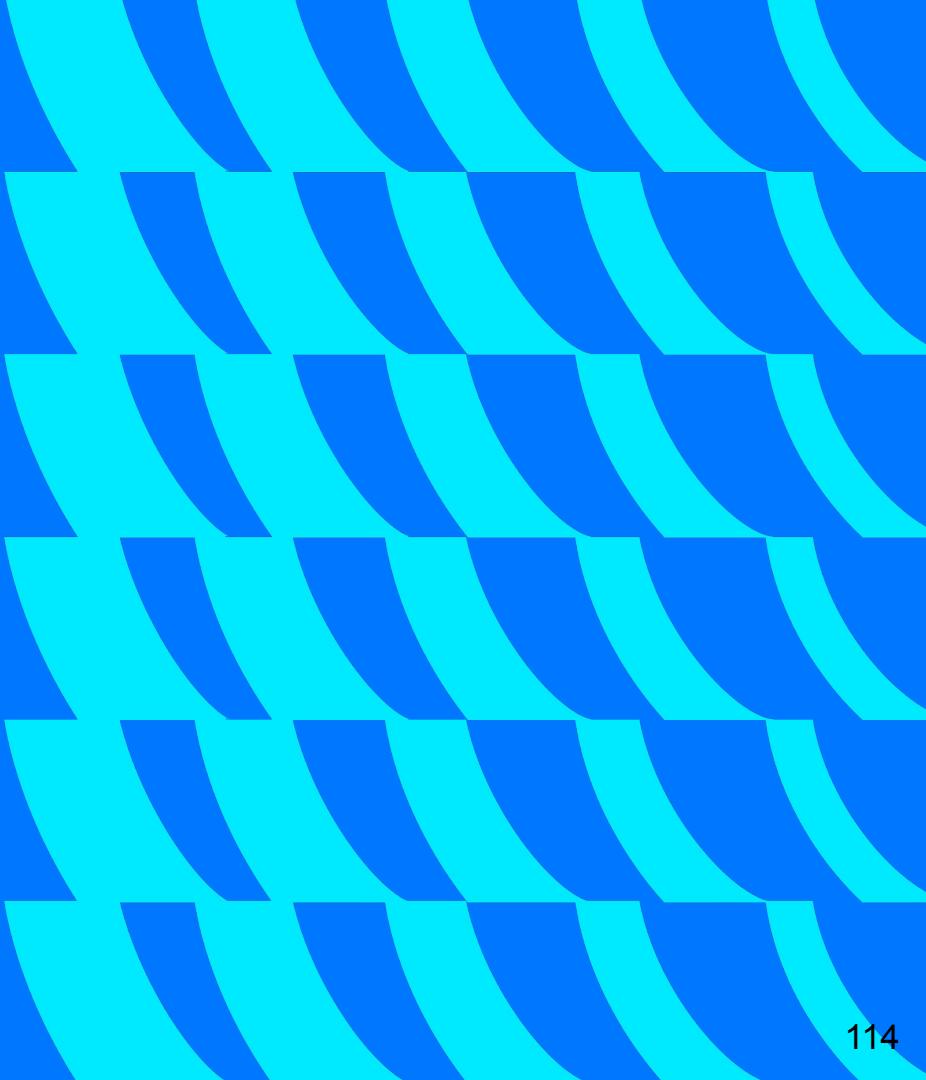


Курс по компьютерному зрению

- Эти и другие темы обсудим в рамках курса в MADE
- Записывайтесь!

ИТОГИ

.....



Итоги

- Сверточный слой
 - Основан на операции свертки
 - Сам "извлекает признаки"
 - Хорошо работает для данных с локальной связностью
 - Эффективнее полносвязного слоя по числу весов
- Сверточная нейросеть
 - Базовые слои - сверточный, активация, пулинг, полносвязный(е) в конце
 - Чем глубже слой в сети, тем с более абстрактными признаками он работает

Почитать еще

- <https://cs231n.github.io/convolutional-networks/>

Контрольные вопросы

- Пусть имеется БД с записями о пациентах, содержащая признаки вида "возраст", "пол", "вес" и т.д, а также поставленный диагноз. Почему использовать сверточную нейросеть для предсказания диагноза - плохая идея?
- Сколько операций (FLOPS) потребуют вычисления в сверточном слое с 32 свертками размера 3x3, padding = 1, stride = 1, на входе цветное (RGB) изображение размера 64x64?
- Каким будет размер тензора на выходе слоя MaxPooling(2, 2), если на вход пришел тензор с $H \times W \times D = 32 \times 16 \times 256$?
- Составьте сверточный слой, который обратит порядок каналов входного изображения с 3 каналами (напр. RGB → BGR)