

# Практический анализ данных и машинное обучение: искусственные нейронные сети

Ульянкин Филипп

16 октября 2018 г.

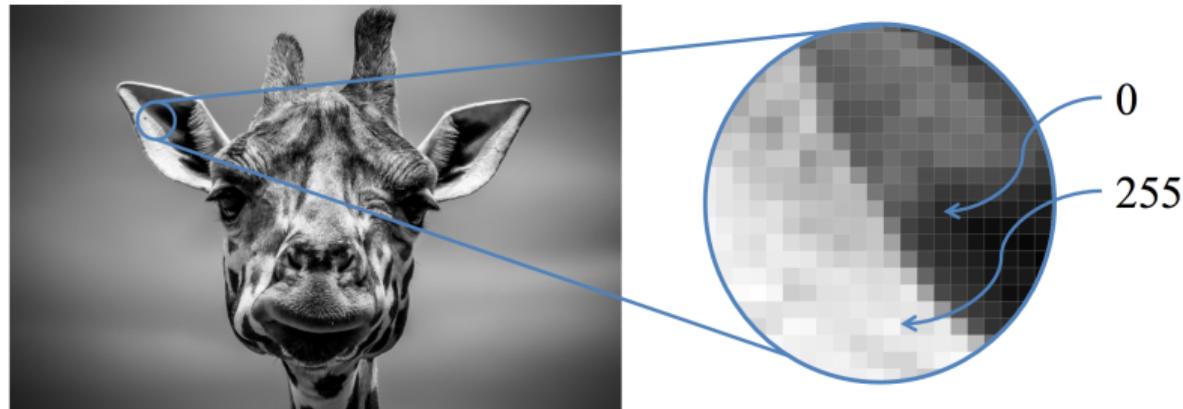
Свёрточные нейросетки

# Agenda

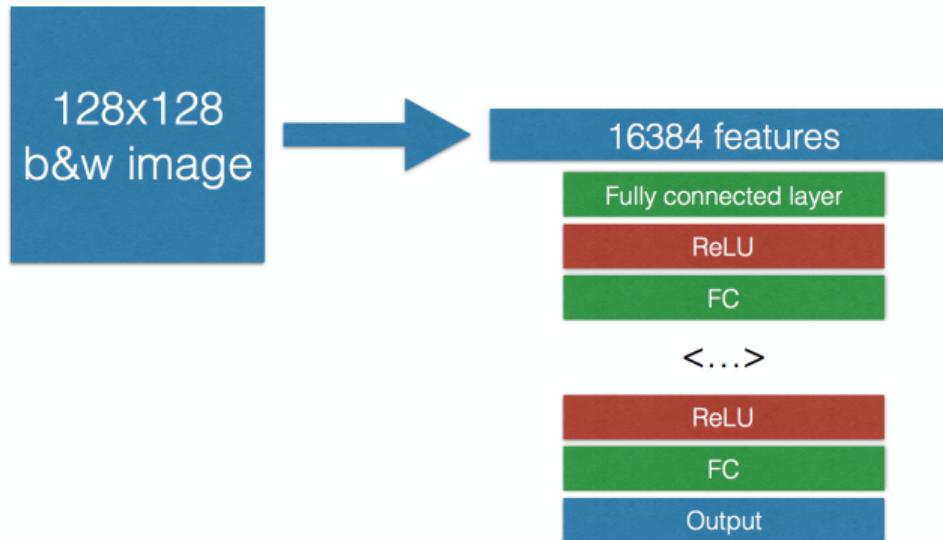
- Что такое свёртка или как видит компьютер
- Снижаем размерность картинки
- Собираем свою собственную свёрточную нейросеть

# Картина – тензор

- Каждая картинка – это матрица из пикселей
- Каждый пиксель обладает яркостью по шкале от 0 до 255
- Цветное изображение имеет три канала пикселей: красный, зелёный и синий



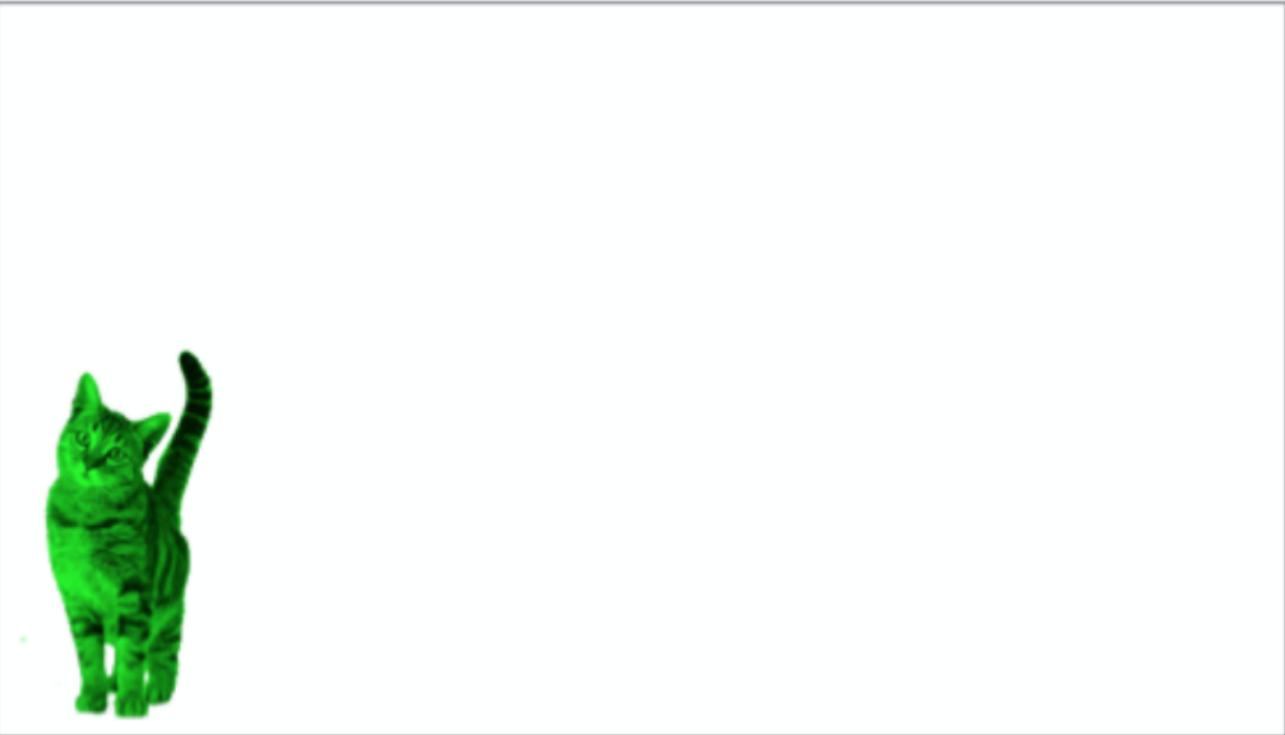
# Обычная сеть



- Очень много весов
- Теряется информация о взаимном расположении пикселей
- Изображение в разных местах картинки даёт разные веса







# Обычная сетка

- Хотим, чтобы информация не терялась
- Хотим одинаковые веса  $\Rightarrow$  свёртка

# Свёртка в питоновской тетрадке

# Свёртка

0	0	0	0
0	0	0	0
0	0	1	0
0	0	0	1

Input

\*

1	0
0	1

=

0	0	0
0	1	0
0	0	2

Output

0	0	0	0
0	0	0	0
0	0	0	1
0	0	1	0

Input

\*

1	0
0	1

=

0	0	0
0	0	1
0	1	0

Output

# Свёртка

$$\begin{array}{c} \text{Input} \\ \begin{array}{|c|c|c|c|} \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{Kernel} \\ \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \end{array} = \begin{array}{c} \text{Output} \\ \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 2 \\ \hline \end{array} \end{array}$$

Max = 2

↑

Simple classifier

↓

$$\begin{array}{c} \text{Input} \\ \begin{array}{|c|c|c|c|} \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{Kernel} \\ \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \end{array} = \begin{array}{c} \text{Output} \\ \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \end{array}$$

Max = 1

# Свёртка инвариантна к расположению

0	0	0	0
0	0	0	0
0	0	1	0
0	0	0	1

Input

\*

1	0
0	1

=

0	0	0
0	1	0
0	0	2

Kernel

Output

1	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0

Input

\*

1	0
0	1

=

2	0	0
0	1	0
0	0	0

Kernel

Output

# Свёртка инвариантна к расположению

0	0	0	0
0	0	0	0
0	0	1	0
0	0	0	1

Input

\*

1	0
0	1

=

0	0	0
0	1	0
0	0	2

Output

Max = 2

↑  
Didn't  
change  
↓

1	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0

Input

\*

1	0
0	1

=

2	0	0
0	1	0
0	0	0

Output

Max = 2

# Свёрточный слой

0	0	0	0	0
0	0	1	0	0
0	1	1	0	0
0	1	0	1	0
0	0	0	0	0

Input 3x3  
image with  
**zero padding**  
(grey area)

Shared bias:

$$b$$

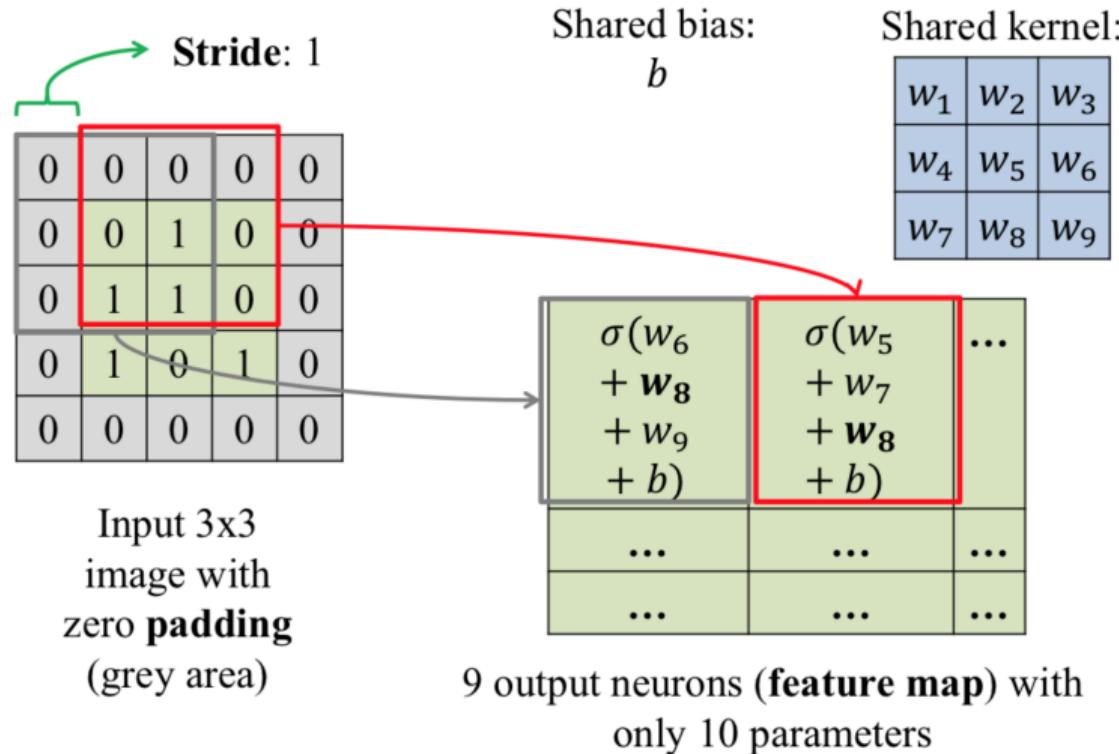
Shared kernel:

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

$\sigma(w_6 + w_8 + w_9 + b)$	...	...
...	...	...
...	...	...

9 output neurons (**feature map**) with  
only 10 parameters

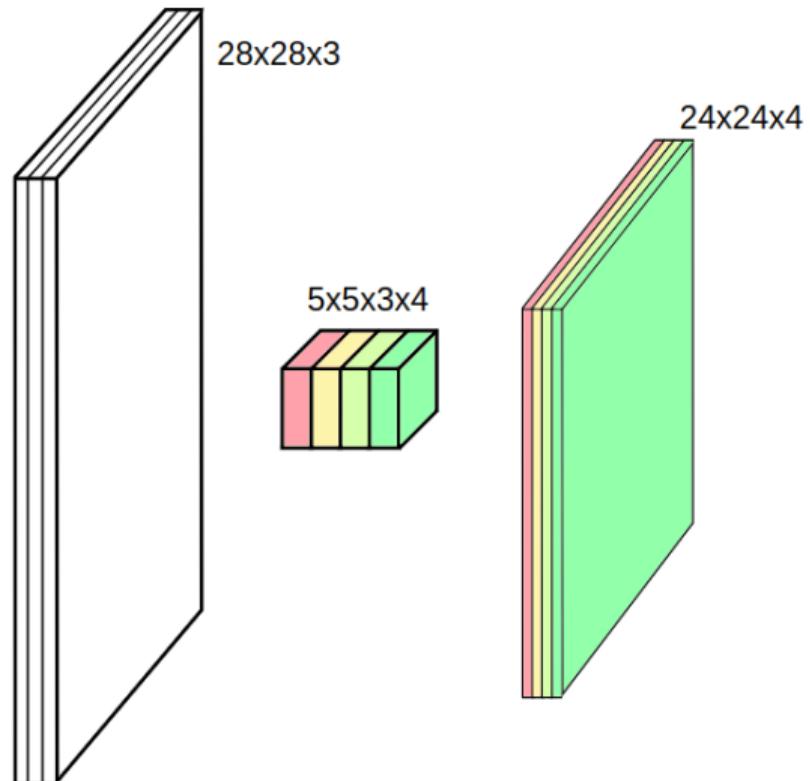
# Свёрточный слой



# Свёрточный слой

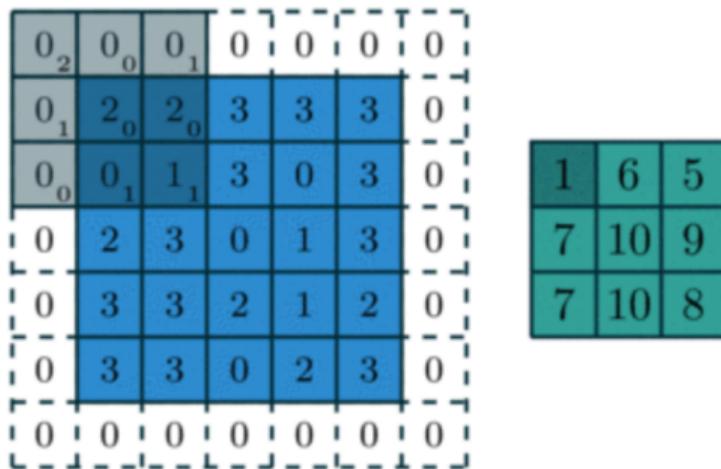
- Слой действует одинаково для каждого участка картинки, в отличие от полносвязного
- Нужно оценивать меньшее количество параметров
- Слой учитывает взаимное расположение пикселей

## Слой из нескольких фильтров



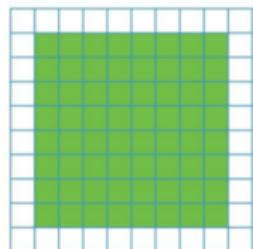
# Снижение размерности

# Дополнение (Padding)

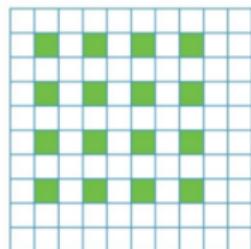


- Padding используют, чтобы пространственная размерность картинки не уменьшалась после применения свертки или уменьшалась не так быстро

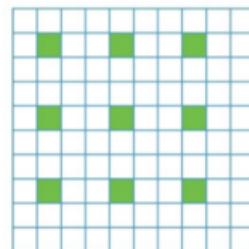
# Сдвиг (Stride)



Stride = 1



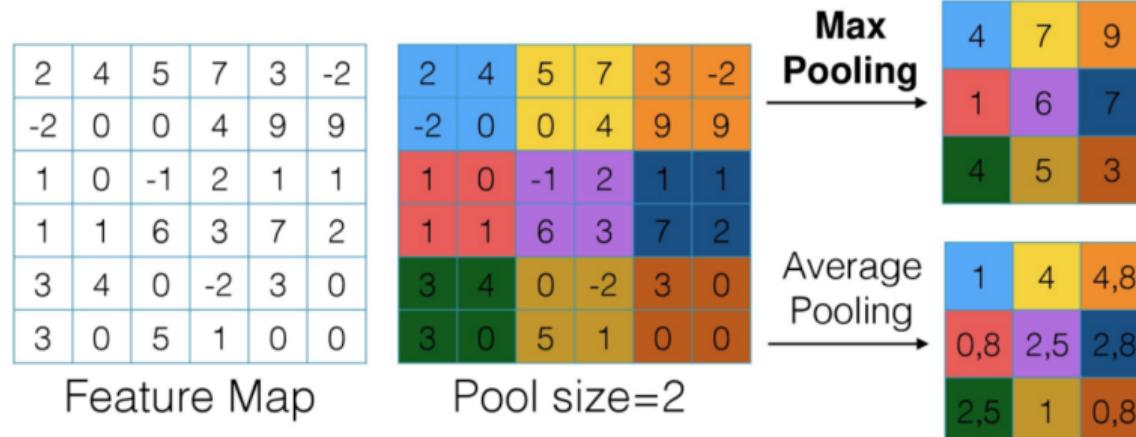
Stride = 2



Stride = 3

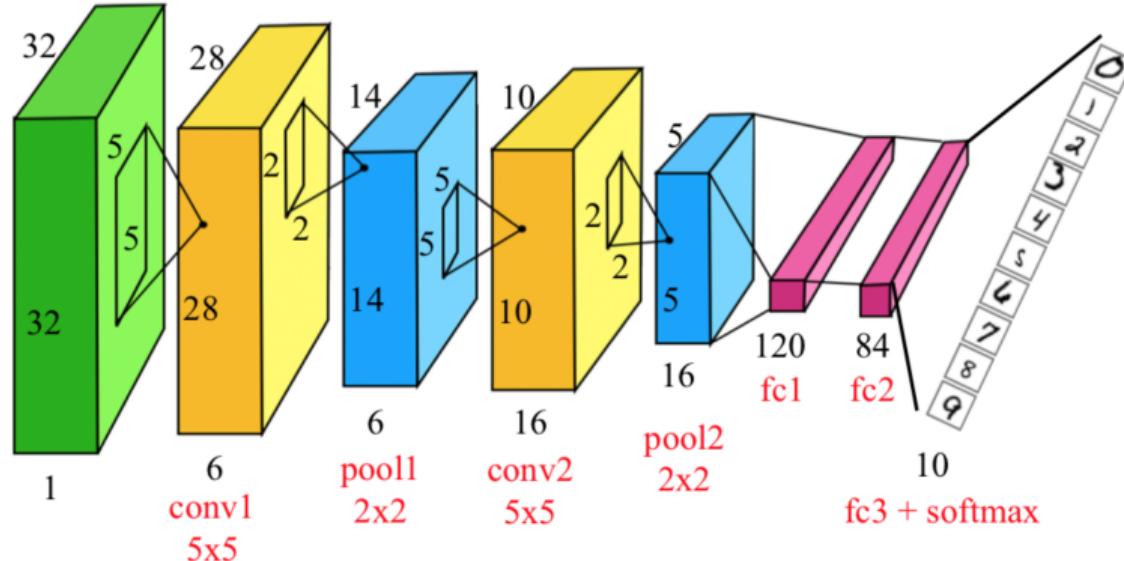
- Пиксели локально скоррелированы — соседние пиксели, как правило, не сильно отличаются друг от друга
- Если будем делать свёртка идёт с каким-то шагом, сэкономим мощности компьютера (число весов) и не потеряем в информации
- Очень агрессивная стратегия снижения размерности картинки

# Пуллинг (Pooling)



- Будем считать внутри какого-то окна максимум или среднее и сворачивать размерность, пользуясь локальной коррелированностью

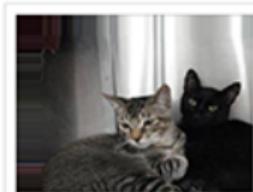
# Простейшая CNN



- Нейросетька LeNet-5 (1998) для распознавания рукописных цифр из датасета MNIST

# Data augmentation

- В сетке может быть миллионы параметров!
- Естественная регуляризация, дополнительная регуляризация, генерация новых данных (data augmentation)
- Генерируем новые цвета, сдвигаем, искажаем и тп

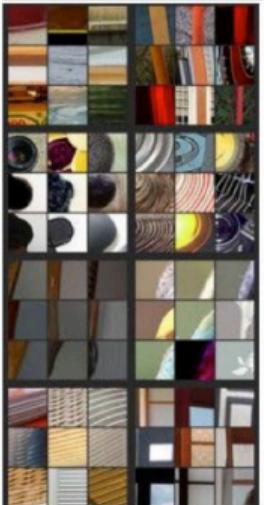


<https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>

# Что выучивают нейросети



Layer 1



Layer 2



Layer 3



Layer 4



Layer 5

<https://arxiv.org/pdf/1311.2901.pdf>

# Собираем свою собственную CNN