

Практический анализ данных и машинное обучение: искусственные нейронные сети

Ульянкин Филипп, Соловей Влад

25 апреля 2019 г.

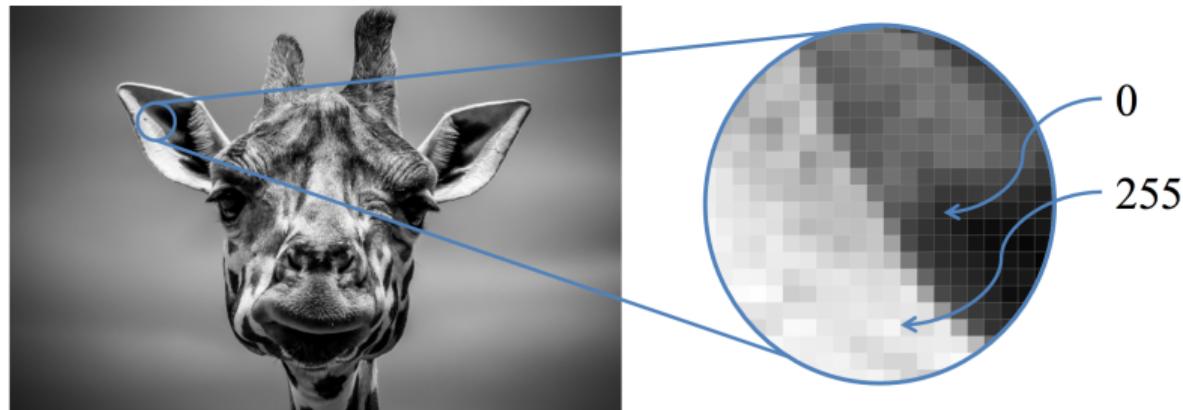
Свёрточные нейросетки

Agenda

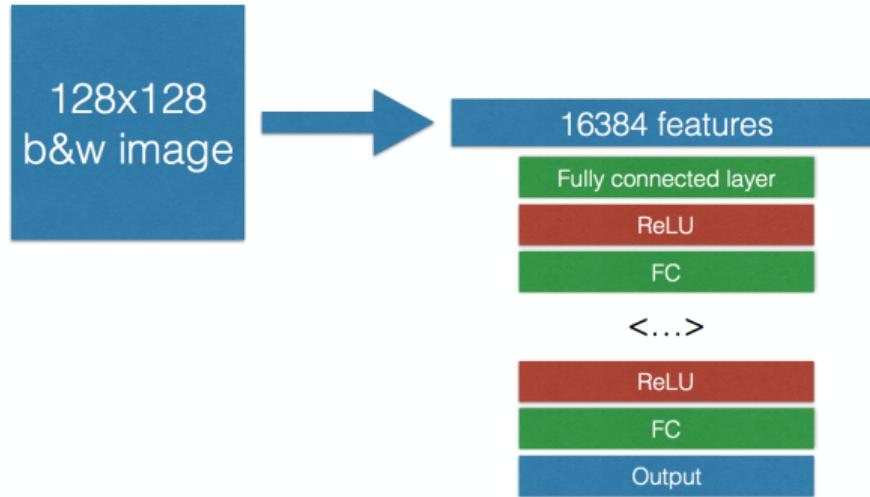
- Что такое свёртка или как видит компьютер
- Собираем свёрточную нейронную сетку
- Говорим про разные приёмы, используемые для свёрточных сеток

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 (rgb)



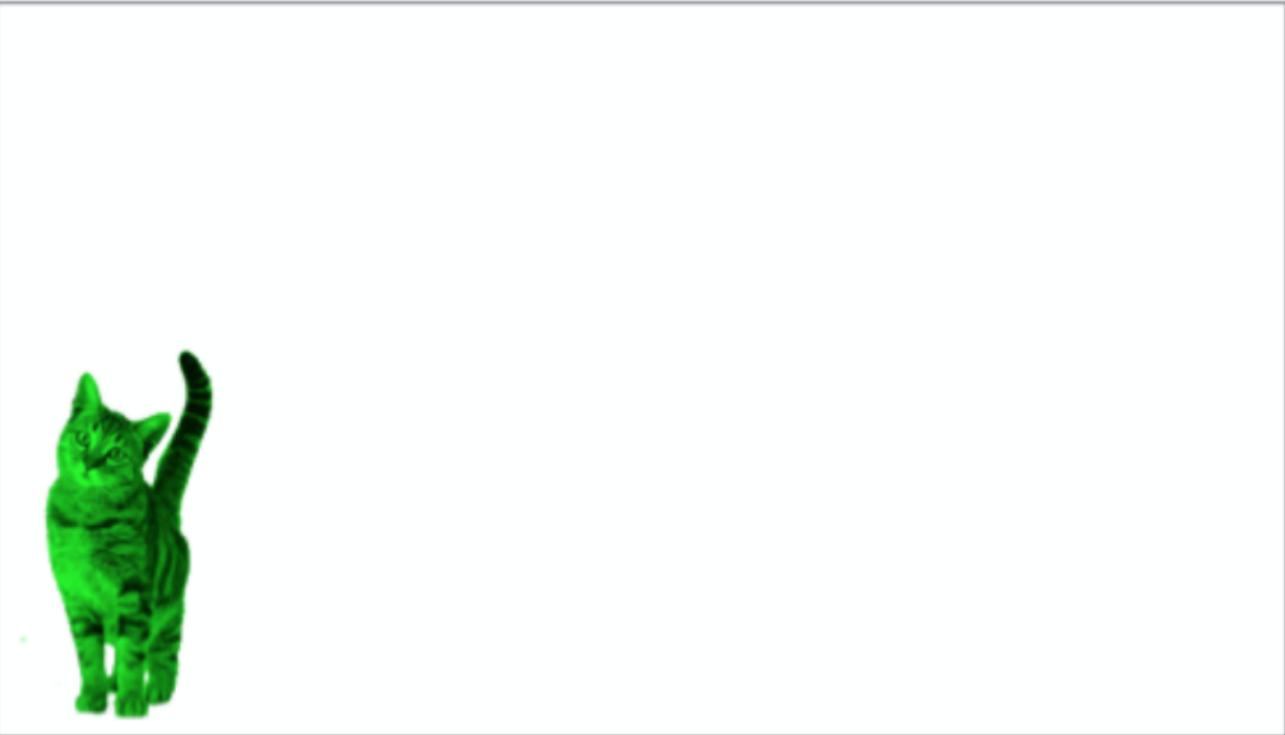
Обычная сеть



- 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



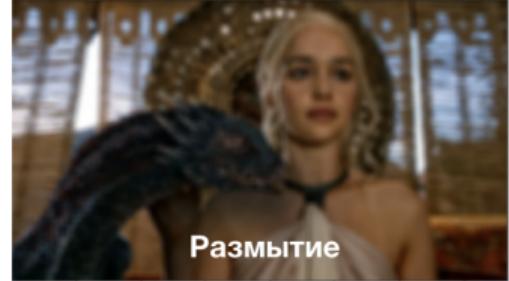
```
[[62, 36, 9], [[62, 36, 9], [[62, 36, 9],  
[62, 36, 9], [61, 35, 8], [60, 34, 7],  
[61, 35, 8], [59, 33, 6], [57, 31, 4],  
..., ..., ...,  
[58, 43, 20], [53, 41, 17], [50, 38, 14],  
[57, 45, 21], [53, 41, 17], [49, 37, 13],  
[57, 45, 21]] [52, 40, 16]] [48, 36, 12]]
```

Красный

Зеленый

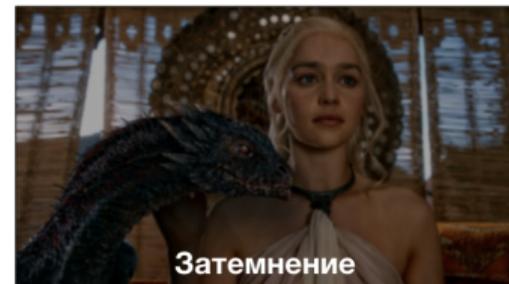
Синий

$$\frac{1}{9} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



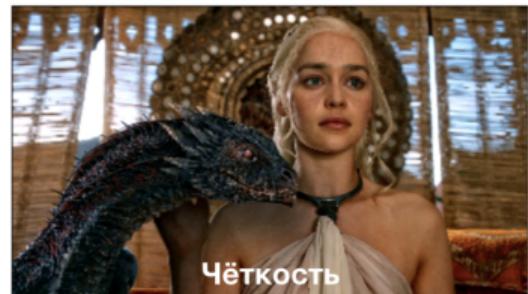
Размытие

$$\begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}$$



Затемнение

$$\begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 2 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix}$$



Чёткость

Свёртка

- Разные ядра помогают накладывать на кртинку различные эффекты
- Какие-то ядра помогают искать границы
- Возможно, с помощью некоторых ядер можно искать что-то кроме границ...

Классификатор слэшей

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

Input Kernel Output

$$\begin{array}{|c|c|c|c|} \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Input Kernel Output

Классификатор слэшей

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

Input

*

1	0
0	1

Kernel

=

0	0	0
0	1	0
0	0	2

Output

Max = 2



Simple classifier



Max = 1

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

Input

*

1	0
0	1

Kernel

=

0	0	0
0	0	1
0	1	0

Output

Свёртка инвариантна к расположению

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

Input Kernel Output

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

Input 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

Max = 2

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

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:

$$b$$

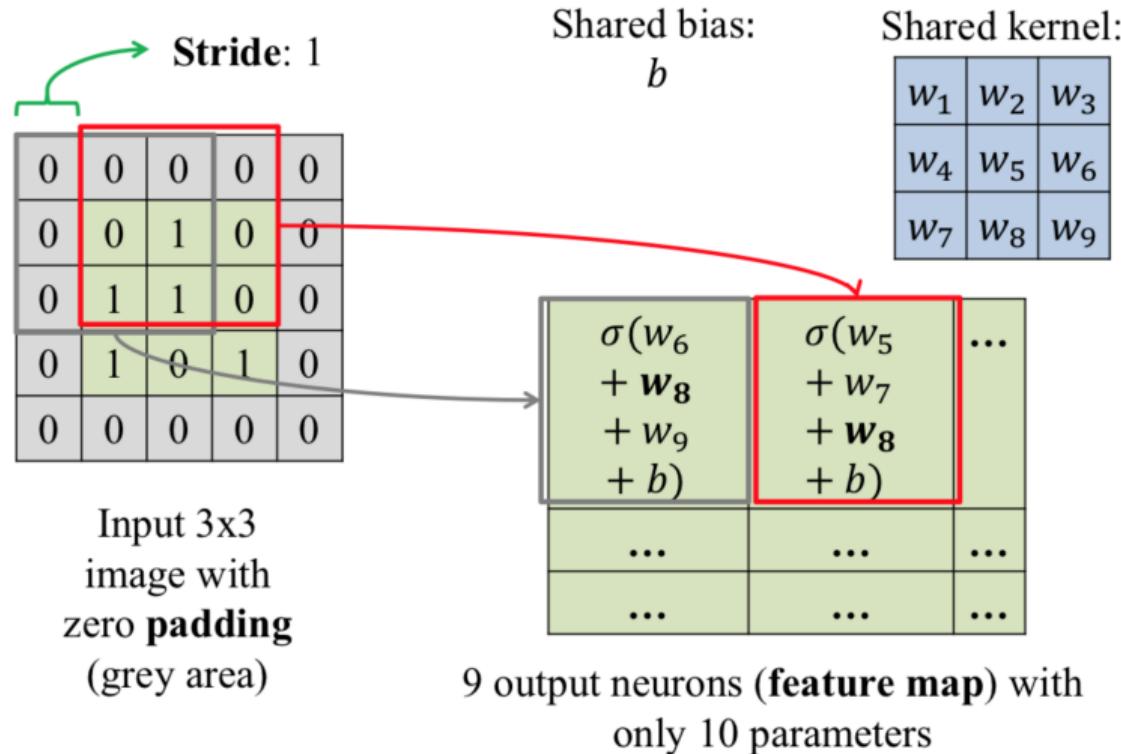
Shared kernel:

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

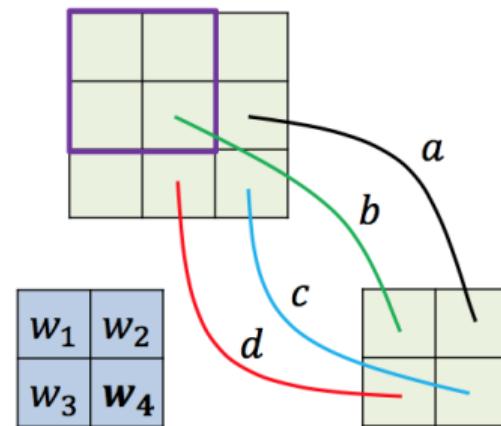
$\sigma(w_6 + w_8 + w_9 + b)$
...
...

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

Свёрточный слой



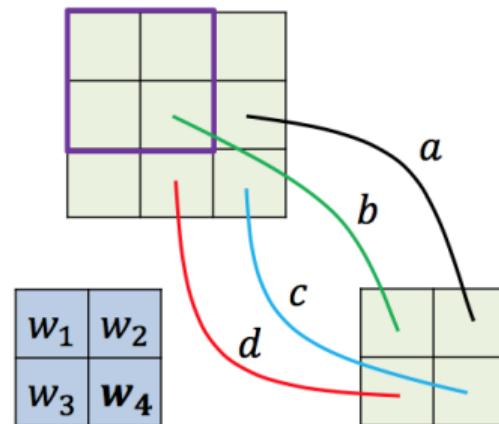
Backpropagation для свёрточного слоя



$$b = w_1 x_{11} + w_2 x_{12} + w_3 x_{21} + w_4 x_{22}$$

$$a = w_1 x_{12} + w_2 x_{13} + w_3 x_{22} + w_4 x_{23}$$

Backpropagation для свёрточного слоя

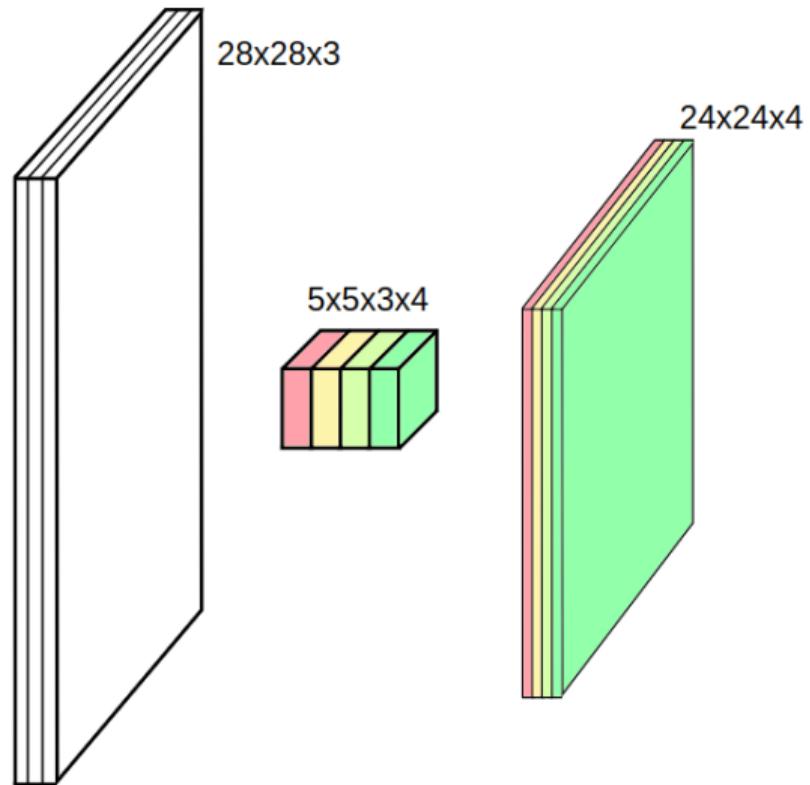


$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a} \cdot x_{11} + \frac{\partial L}{\partial b} \cdot x_{12} + \frac{\partial L}{\partial c} \cdot x_{22} + \frac{\partial L}{\partial d} \cdot x_{12}$$

Свёрточный слой

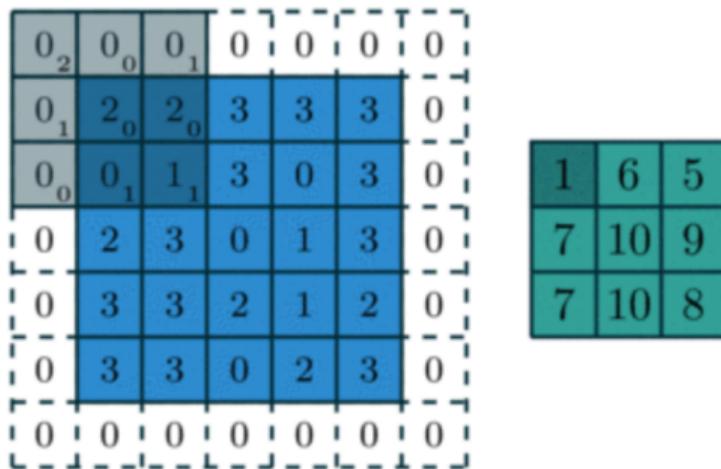
- Слой действует одинаково для каждого участка картинки, в отличие от полносвязного
- Нужно оценивать меньшее количество параметров
- Слой учитывает взаимное расположение пикселей
- Можно учить тем же самым backpropagation
- Свёрточный слой - это полносвязный слой с ограничениями

Слой из нескольких фильтров



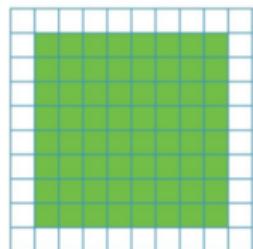
Снижение размерности

Дополнение (Padding)

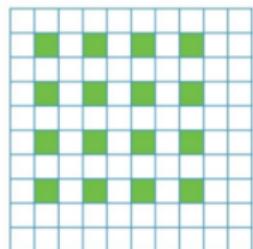


- Padding используют, чтобы пространственная размерность картинки не уменьшалась после применения свертки или уменьшалась не так быстро

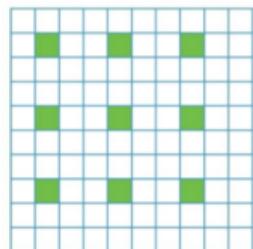
Сдвиг (Stride)



Stride = 1



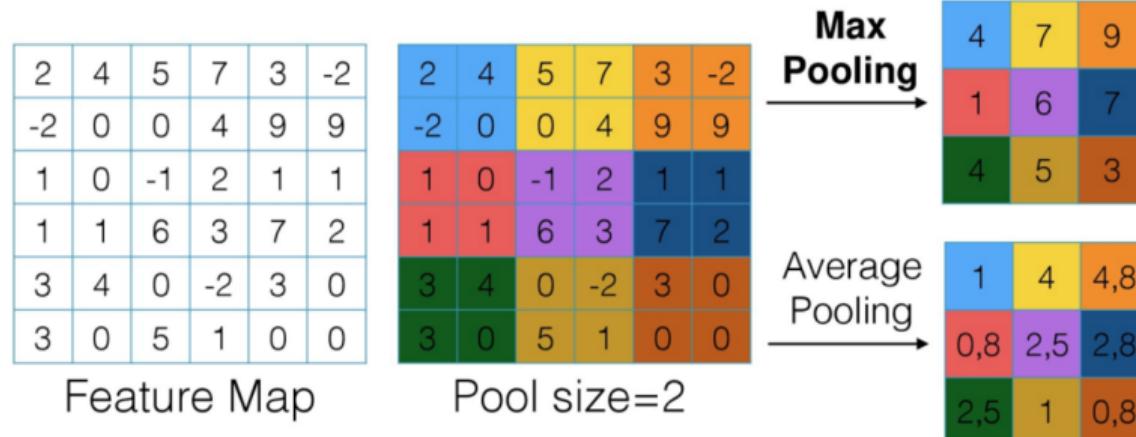
Stride = 2



Stride = 3

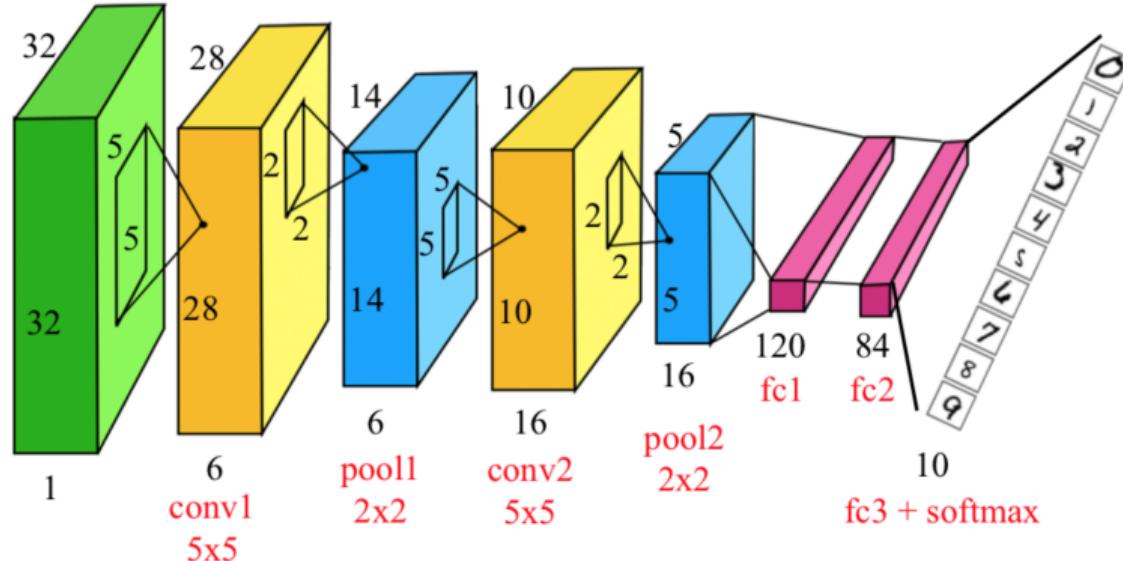
- Пиксели локально скоррелированы — соседние пиксели, как правило, не сильно отличаются друг от друга
- Если будем делать свёртка идёт с каким-то шагом, сэкономим мощности компьютера (число весов) и не потеряем в информации
- Очень агрессивная стратегия снижения размерности картинки

Пуллинг (Pooling)



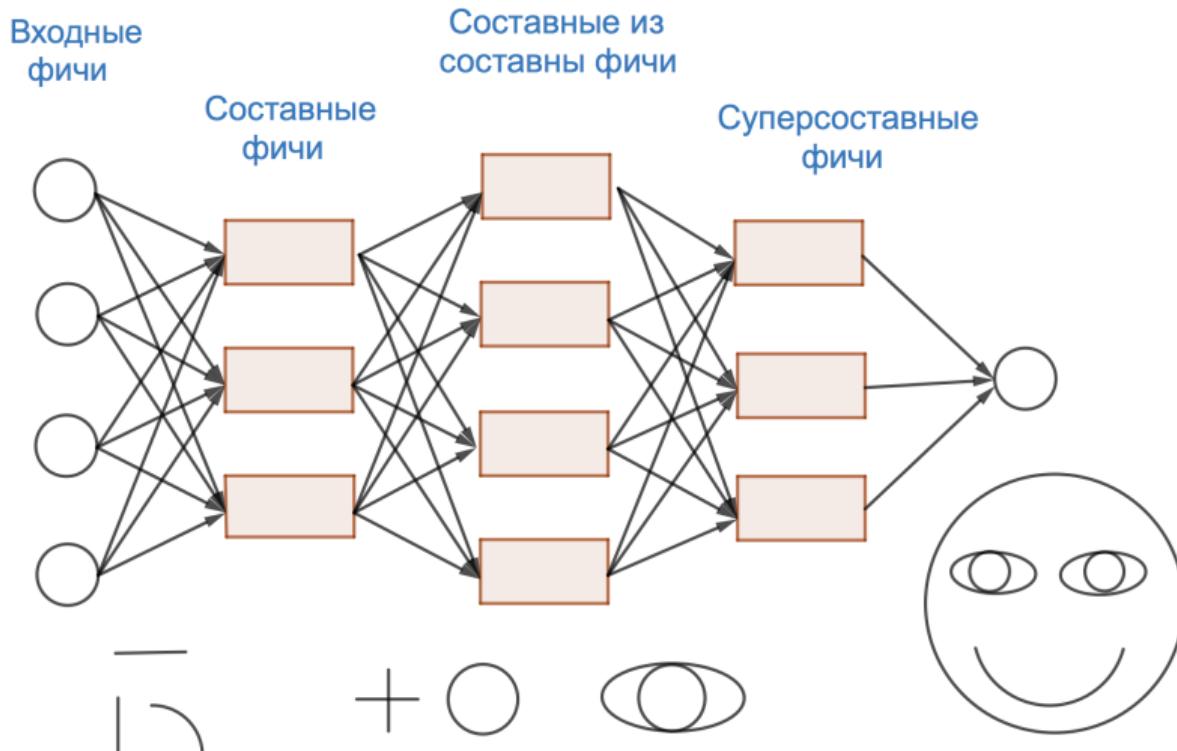
- Будем считать внутри какого-то окна максимум или среднее и сворачивать размерность, пользуясь локальной коррелированностью

Простейшая CNN



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

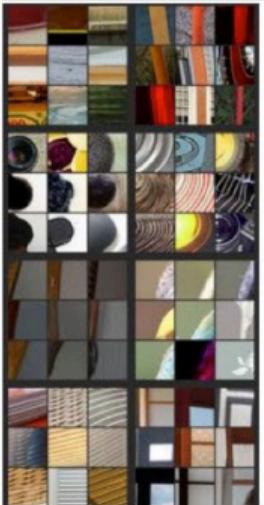
Что выучивают нейросети



Что выучивают нейросети



Layer 1



Layer 2



Layer 3



Layer 4



Layer 5

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

Что выучивают нейросети

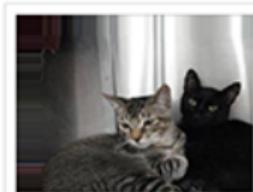
Подробнее про это в следующий раз :)

Собираем свою собственную CNN

Data augmentation

Data augmentation

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



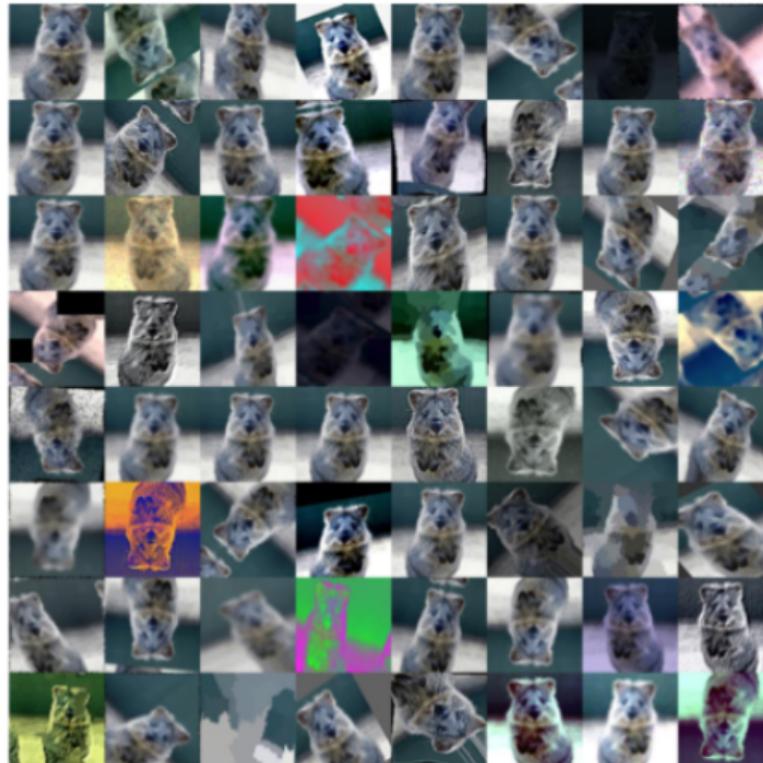
<https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>

Data augmentation

- Сдвиги
- Увеличение, уменьшение
- Повороты
- Искажение
- Затенение
- Смена стиля (красок)

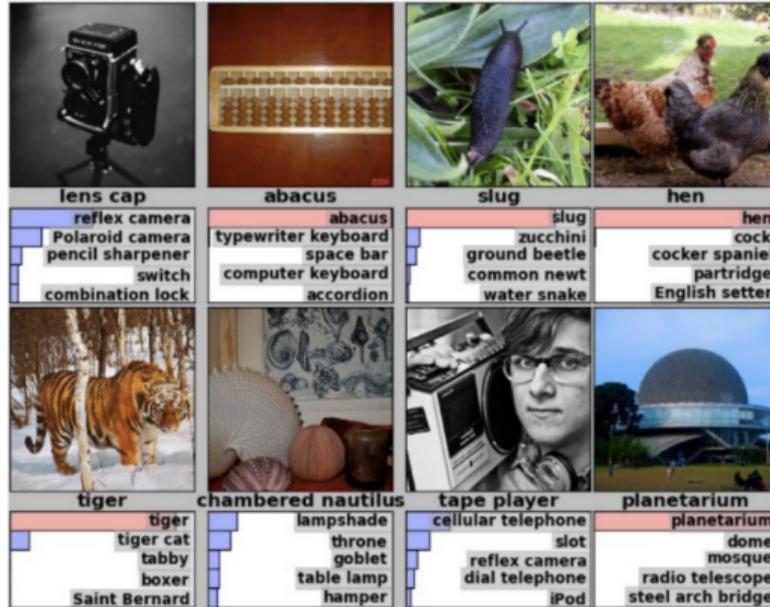
Делает модель более устойчивой, полезна при маленьких выборках. На больших датасетах также улучшает результаты.

Data augmentation



Сказ про то как люди ImageNet
выигрывали

ImageNet



- около 10 миллионов размеченных изображений из интернета

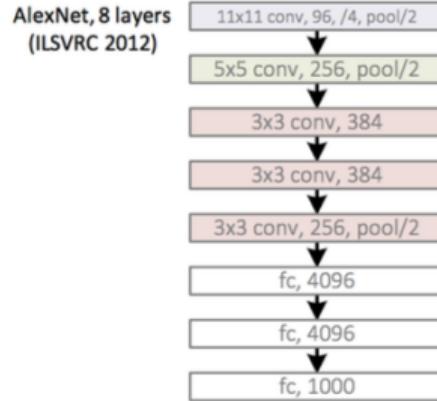


- бывают спорные изображения: тут вишня, если распознать как далматинец, будет неправильно

ImageNet

- выборка очень неоднородная, на ней проводятся соревнования по распознанию образов
- обычно изображение требуется отнести к одному из 1000 классов, можно давать несколько ответов
- если один из пяти вариантов оказался верным, то классификация считается верной
- до 2012 года лучшие алгоритмы дают ошибку в 25%
- в 2012 году на арену выходят глубокие нейронные сети

AlexNet



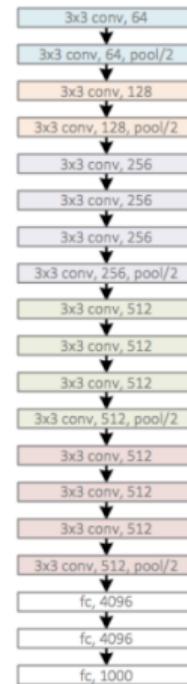
- 5 свёрточных слоёв, два полносвязных слоя, выходной слой $\Rightarrow 16.5\%$ ошибок

Ансамбль сетей

- Давайте обучим много сеток
- Каждая сетка будет классифицировать
- Итоговый результат будем выбирать методом большинства
- Ошибка упала до 11.7%

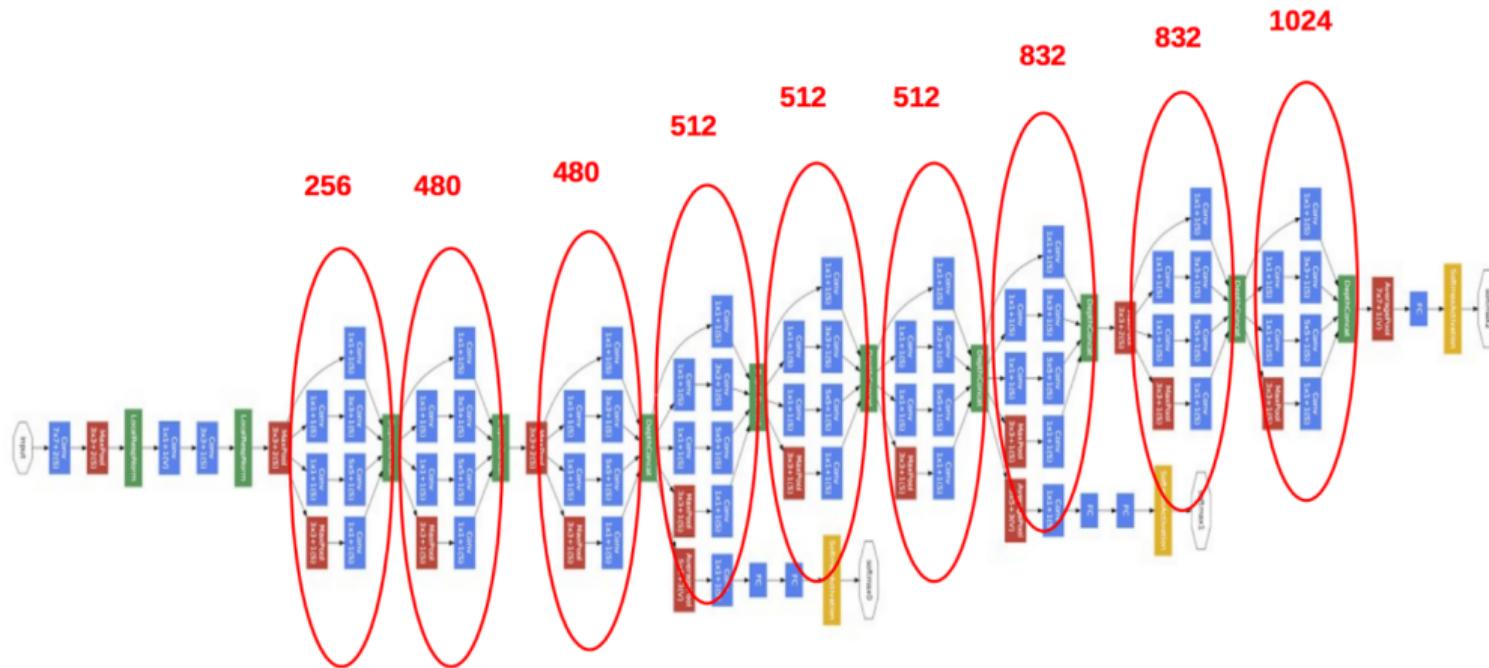
VGG

VGG, 19 layers
(ILSVRC 2014)

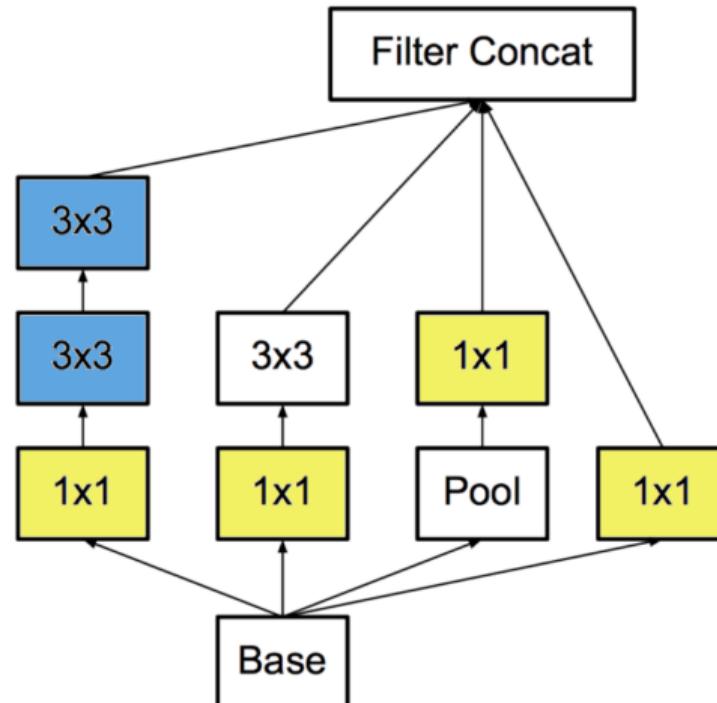


- 19 слоёв, большая часть свёрточные, все свёртки $3 \times 3 \Rightarrow$ ошибка 7.3%.

GoogleNet aka Inception V1



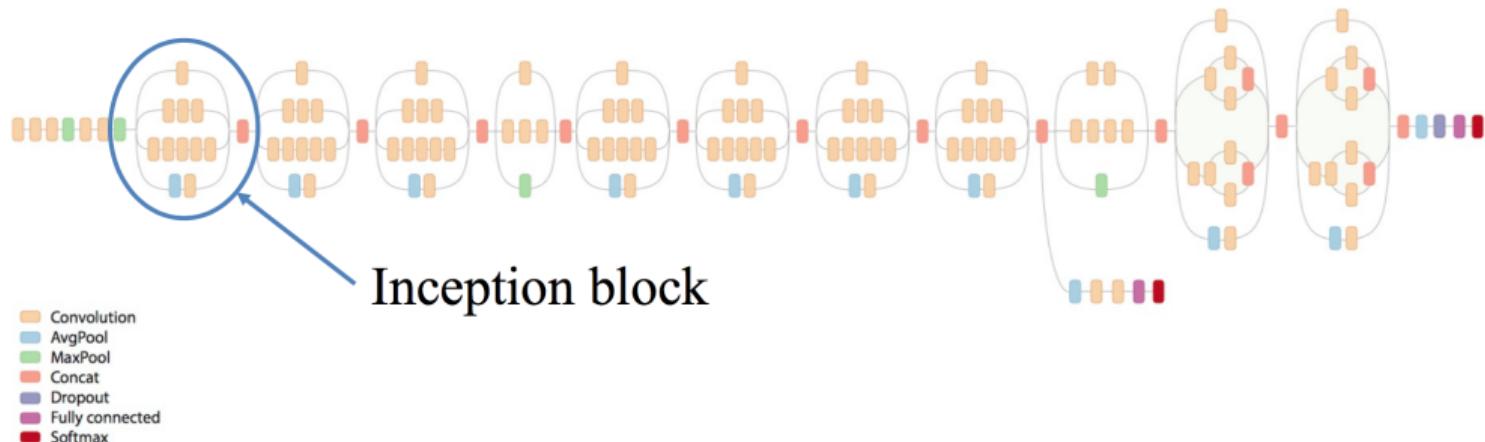
GoogleNet aka Inception V1



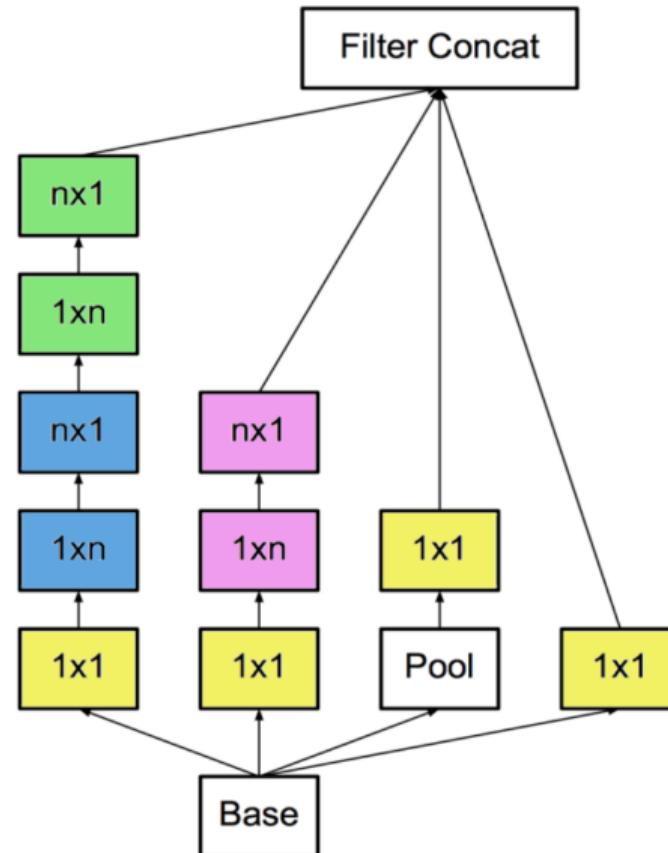
GoogleNet aka Inception V1

- Сетка собирается из компонент
- На каждом слое используется не одна свёртка, а несколько разных, что помогает реагировать на сигналы разного масштаба и улучшает работу
- Несколько дополнительных классификаторов на разных уровнях
- идея в том, что такие классификаторы позволяют «протолкнуть» градиенты к ранним слоям и тем самым уменьшить эффект затухания градиента
- Итоговое качество 6.7% ошибок

Inception V3



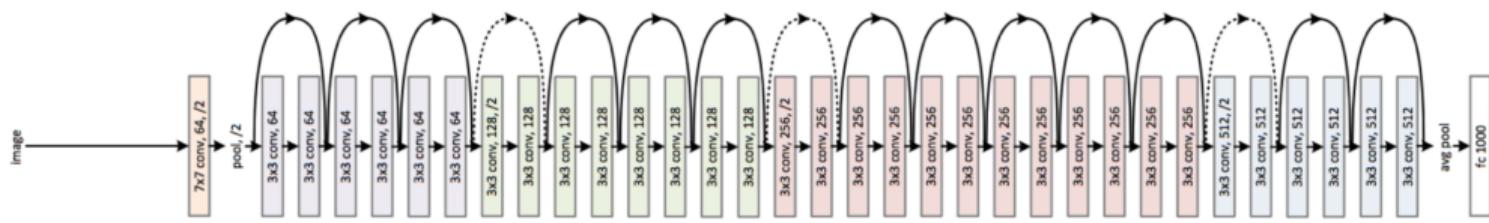
Inception V3



Inception V3

- Сетка собирается из компонент
- На каждом слое используется ни одна свёртка, а несколько разных, что помогает реагировать на сигналы разного масштаба и улучшает работу
- Несколько дополнительных классификаторов на разных уровнях
- идея в том, что такие классификаторы позволяют «протолкнуть» градиенты к ранним слоям и тем самым уменьшить эффект затухания градиента (позже от идеи отказались)
- Итоговое качество 5.6% ошибок

ResNet (Microsoft)



- 152 слоя, ошибка составила 3.75%
 - ключевым элементом архитектуры является связь, которая пропускает несколько слоёв, передавая результат предыдущего слоя
 - такое изменение позволило полностью отказаться от таких техник регуляризации, как DropOut

Итоги

