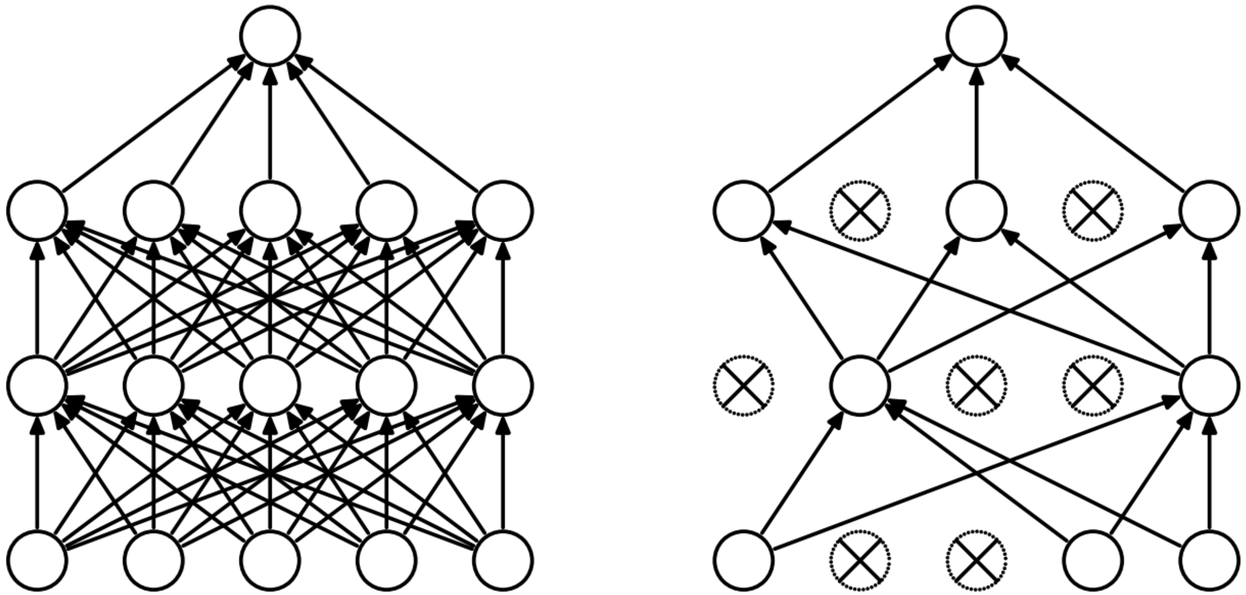
Глубинное обучение

Сверточные нейросети.

Ирина Сапарина

Dropout



(a) Standard Neural Net (b) After applying dropout.

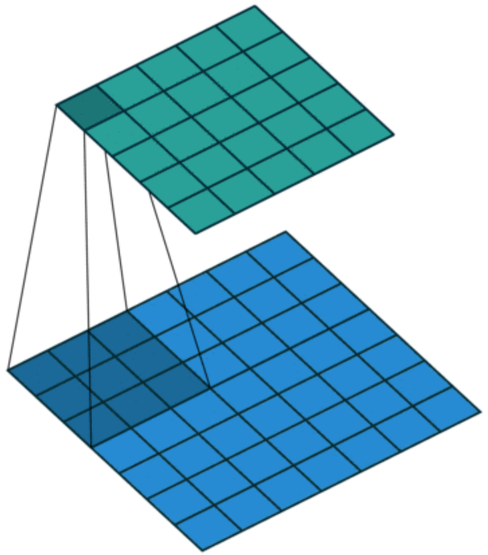
Image credit

Случайно выкидываем нейроны на forward pass с вероятностью p

Обучение: умножаем оставшиеся нейроны на $\frac{1}{1-p}$

Сверточные нейросети (CNN)

Свертка



Параметры: ядро (kernel) - w

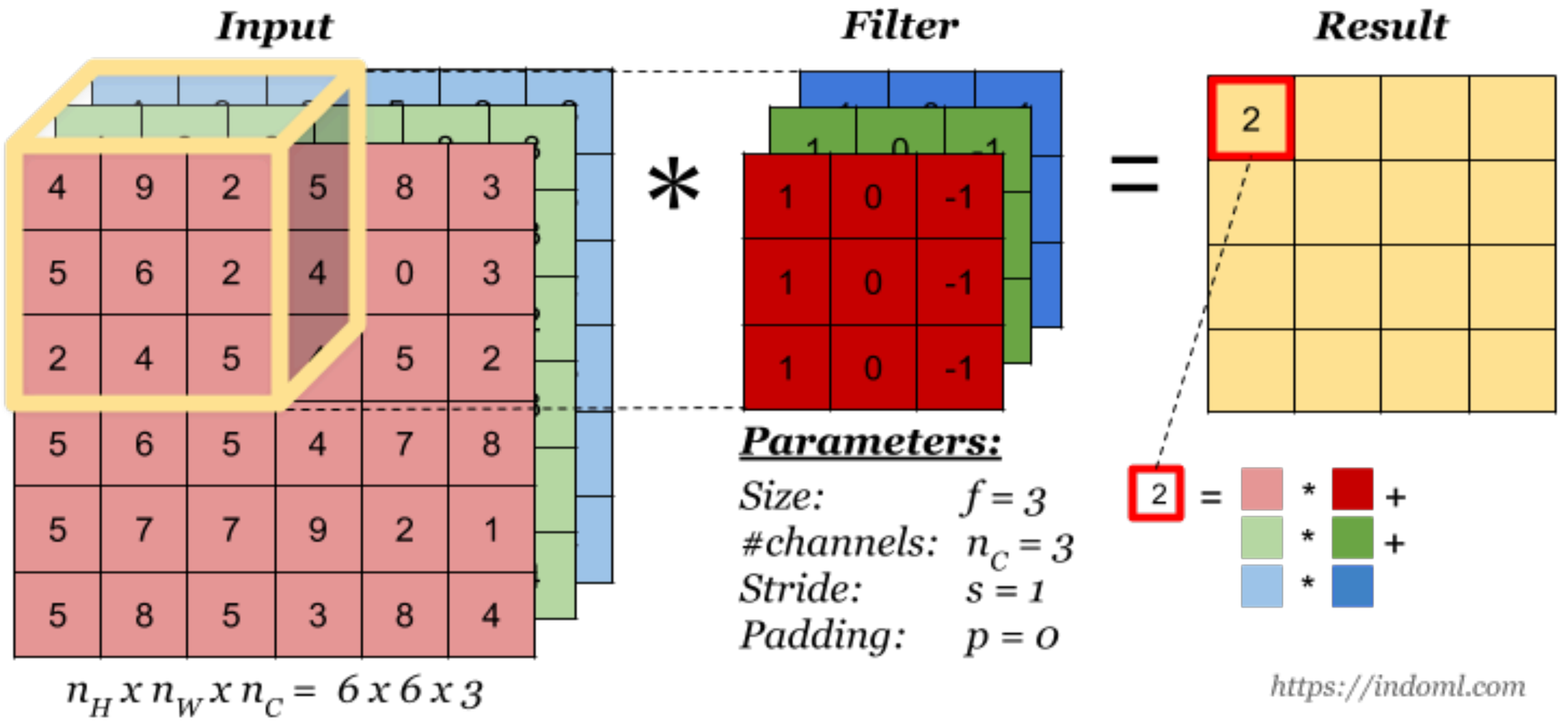
размер - $K \times K \times C_{in} \times C_{out}$

+ *bias размера* C_{out}

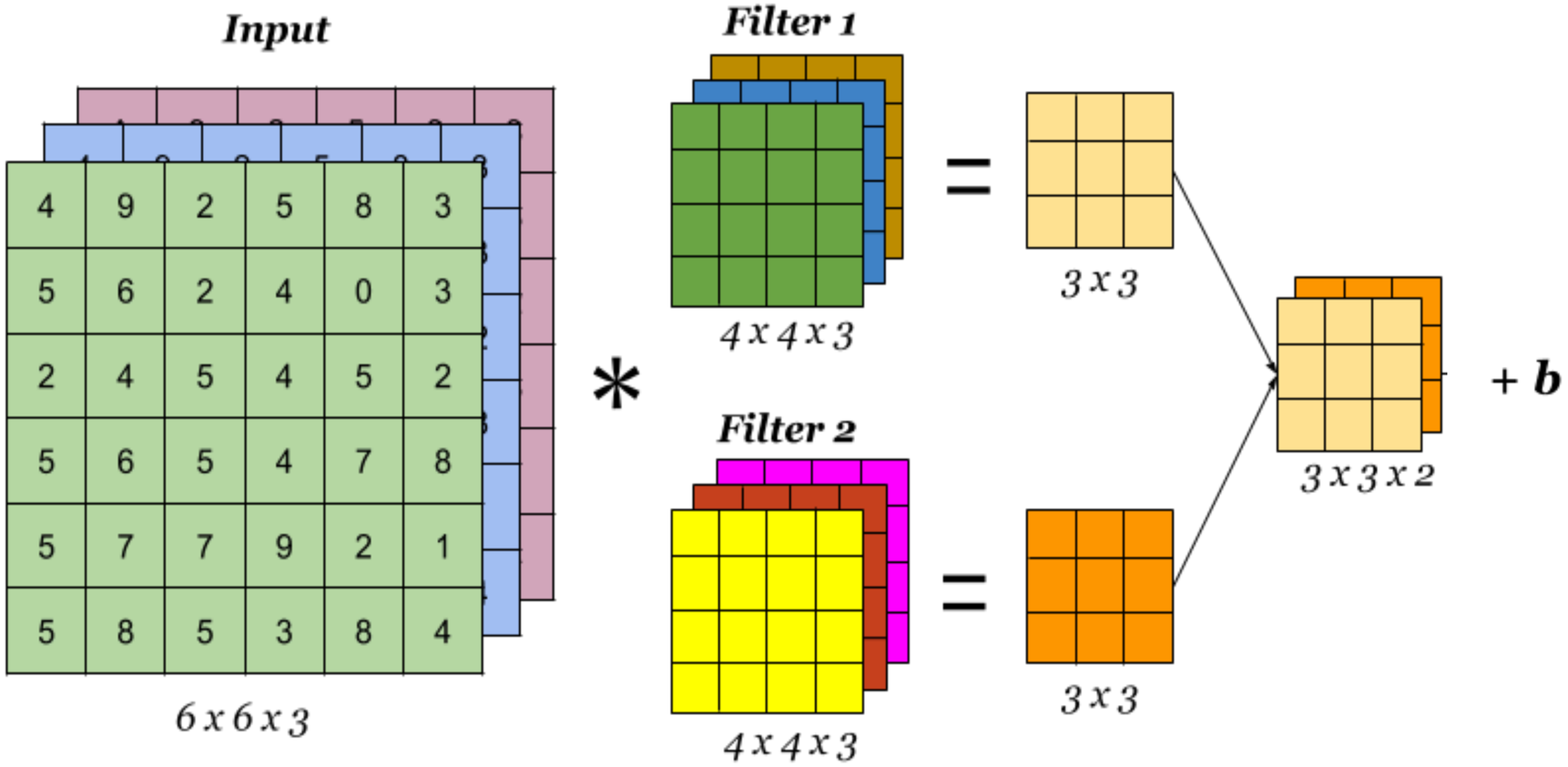
Много параметров: пропуски, шаги, …

$$H_{in} \times W_{in} \times C_{in} \rightarrow H_{output} \times W_{output} \times C_{output}$$

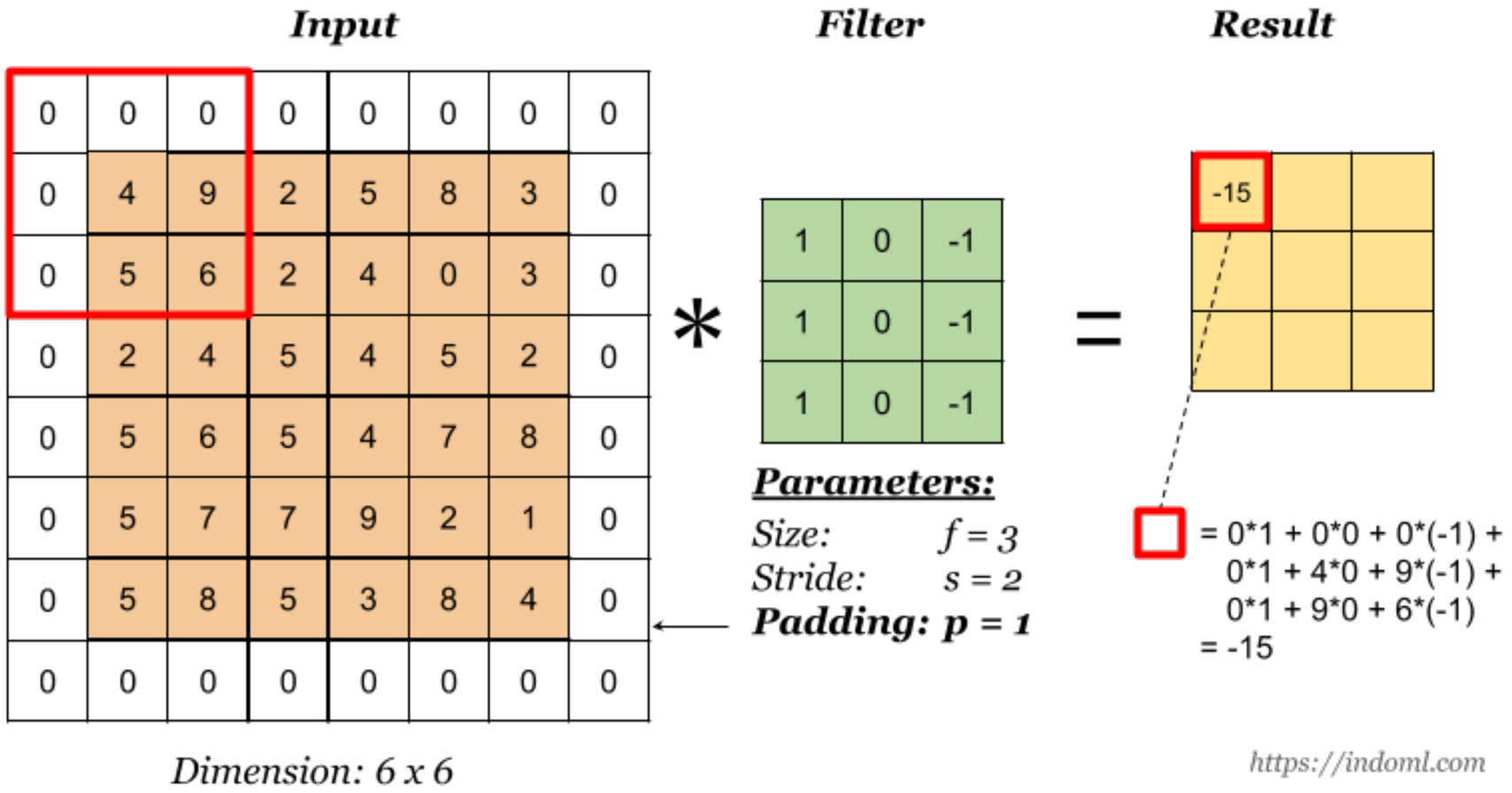
Свертка


Input
Filter
Result
4 9 2 5 8 3
5 6 2 4 0 3
2 4 5 5 2
5 6 5 4 7 8
5 7 7 9 2 1
5 8 5 3 8 4
1 0 -1
1 0 -1
1 0 -1
2
Parameters:
Size: f = 3
#channels: n_C = 3
Stride: s = 1
Padding: p = 0
n_H x n_W x n_C = 6 x 6 x 3
2 = * + * + *
https://indoml.com

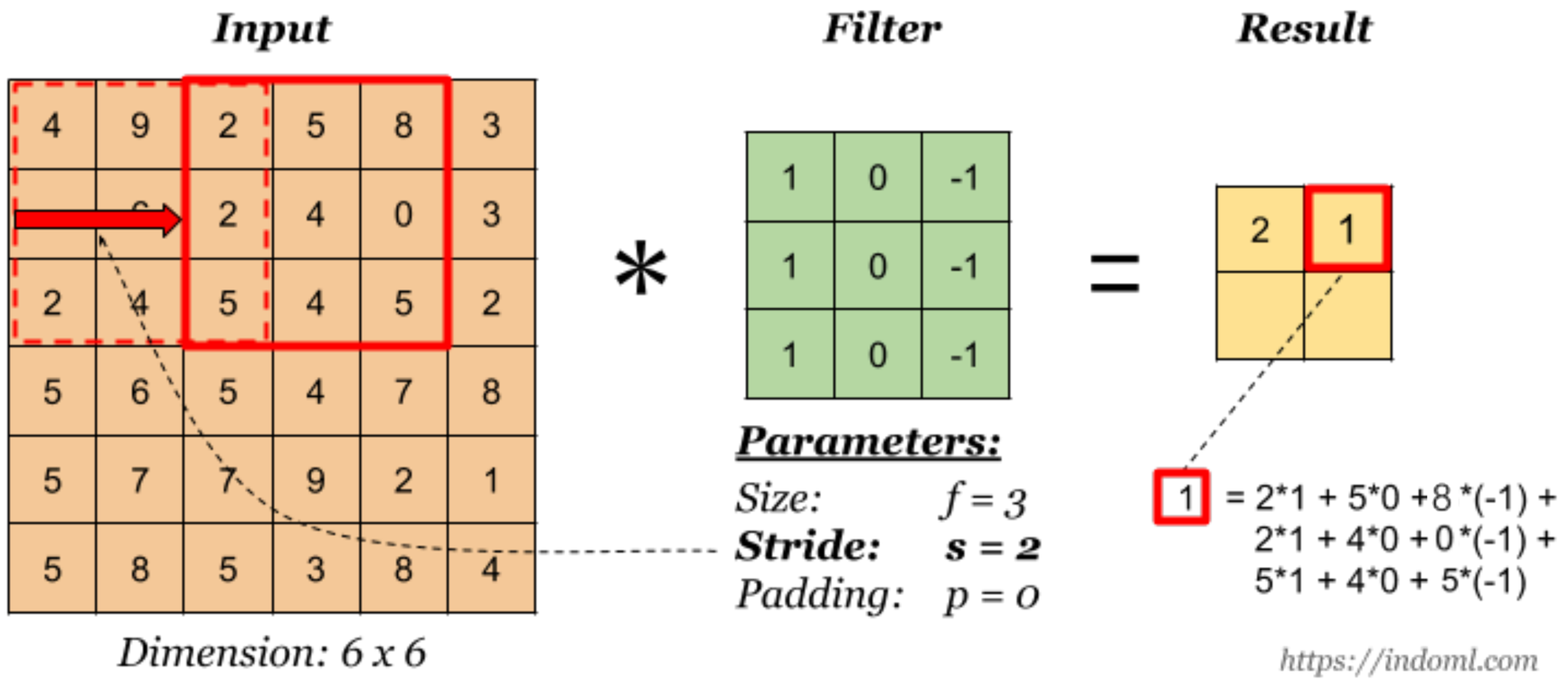
Свертка


Input
4 9 2 5 8 3
5 6 2 4 0 3
2 4 5 4 5 2
5 6 5 4 7 8
5 7 7 9 2 1
5 8 5 3 8 4
6 x 6 x 3
*
Filter 1
4 x 4 x 3
=
3 x 3
Filter 2
4 x 4 x 3
=
3 x 3
3 x 3 x 2
+ b

Свертка: padding



Свертка: stride



Свертка как перемножение матриц

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}$$ - input

$$\begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}$$ - kernel

Свертка как перемножение матриц

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}$$ - input

$$\begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}$$ - kernel

$$\begin{bmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} k_1x_1 + k_2x_2 + k_3x_4 + k_4x_5 \\ k_1x_2 + k_2x_3 + k_3x_5 + k_4x_6 \\ k_1x_4 + k_2x_5 + k_3x_7 + k_4x_8 \\ k_1x_5 + k_2x_6 + k_3x_8 + k_4x_9 \end{bmatrix}$$

- circulant matrix

Свертка как перемножение матриц

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}$$ - input

$$\begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}$$ - kernel

$$\begin{bmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} k_1x_1 + k_2x_2 + k_3x_4 + k_4x_5 \\ k_1x_2 + k_2x_3 + k_3x_5 + k_4x_6 \\ k_1x_4 + k_2x_5 + k_3x_7 + k_4x_8 \\ k_1x_5 + k_2x_6 + k_3x_8 + k_4x_9 \end{bmatrix}$$

- circulant matrix

Свертка: визуализация

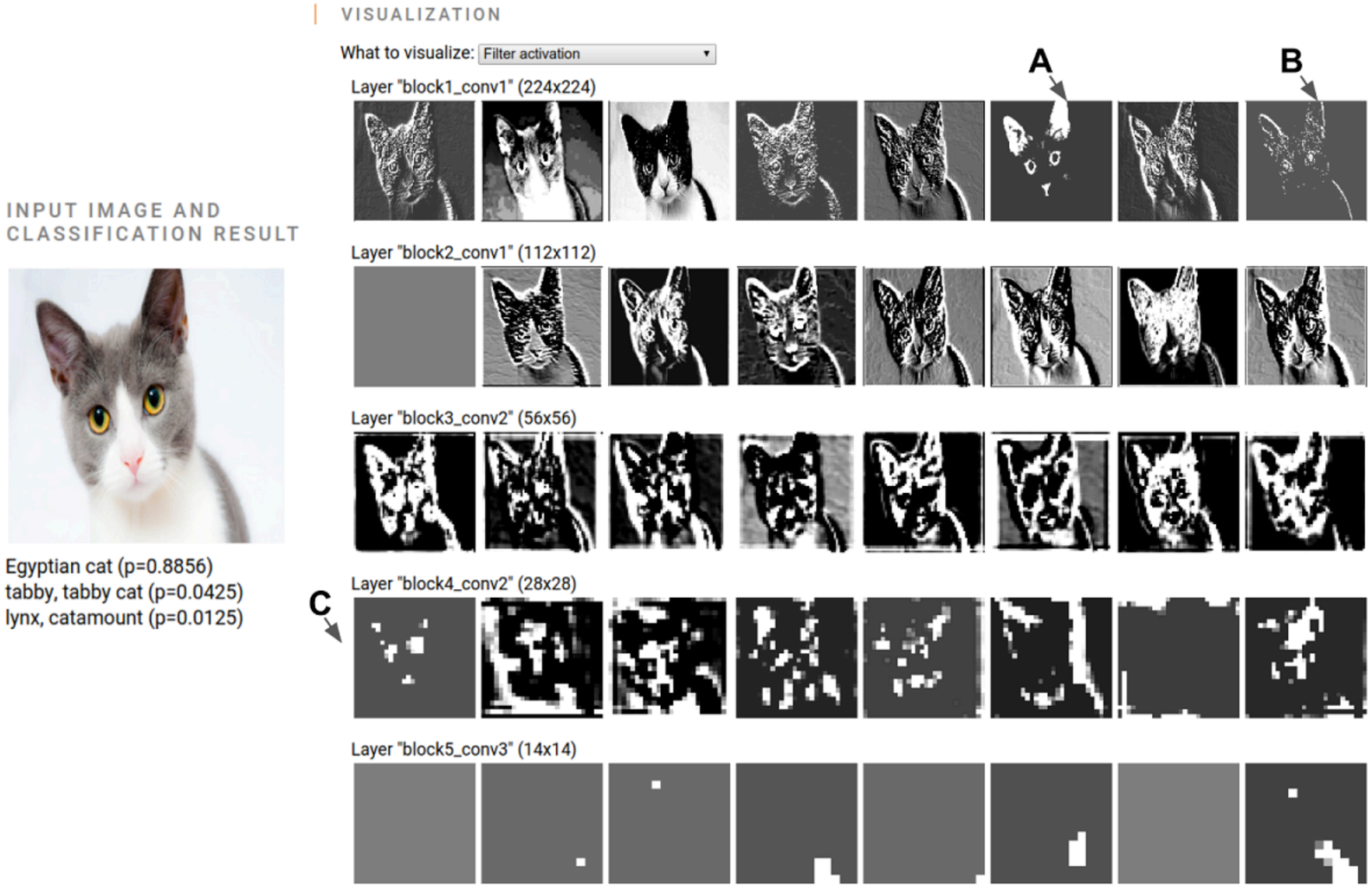
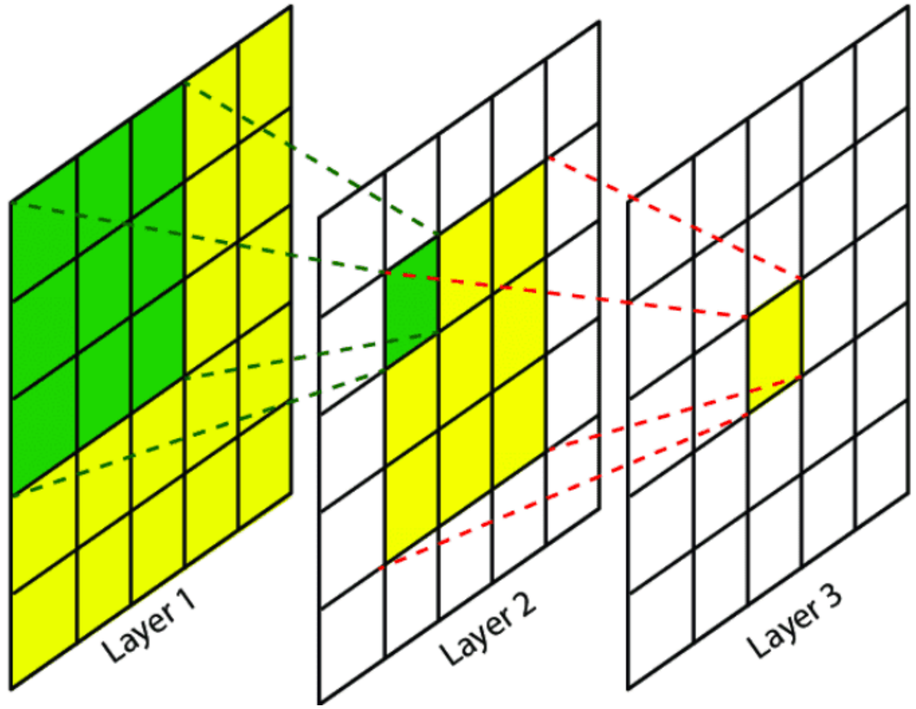


Image credit

Свертка: receptive field



Область исходного изображения, которая влияет на конкретный признак

Архитектуры CNN

LeNet-5 (LeCun et al., 1998)

Задача: распознавание рукописных цифр (MNIST)

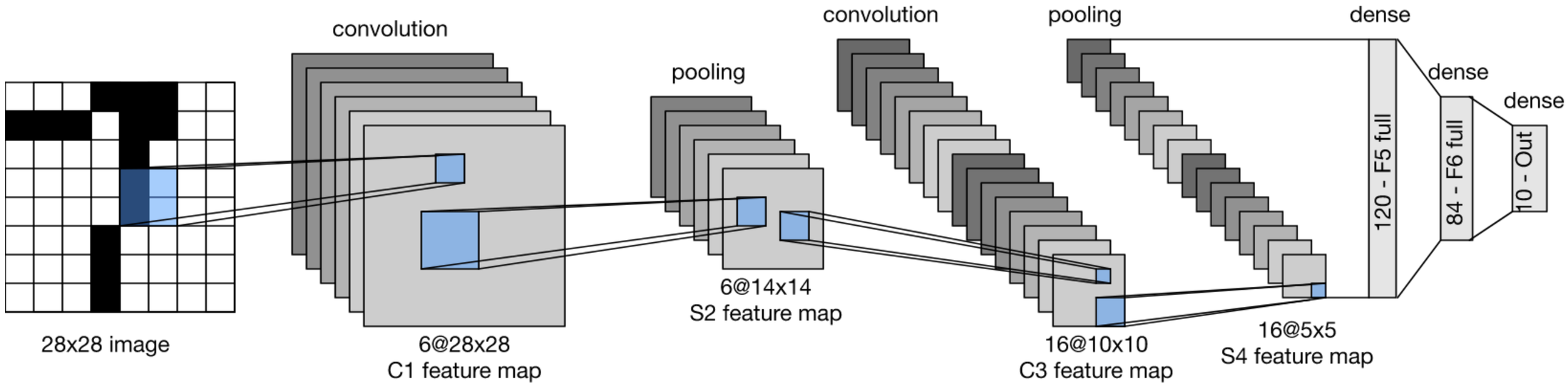


Image credit

LeNet-5 (LeCun et al., 1998)

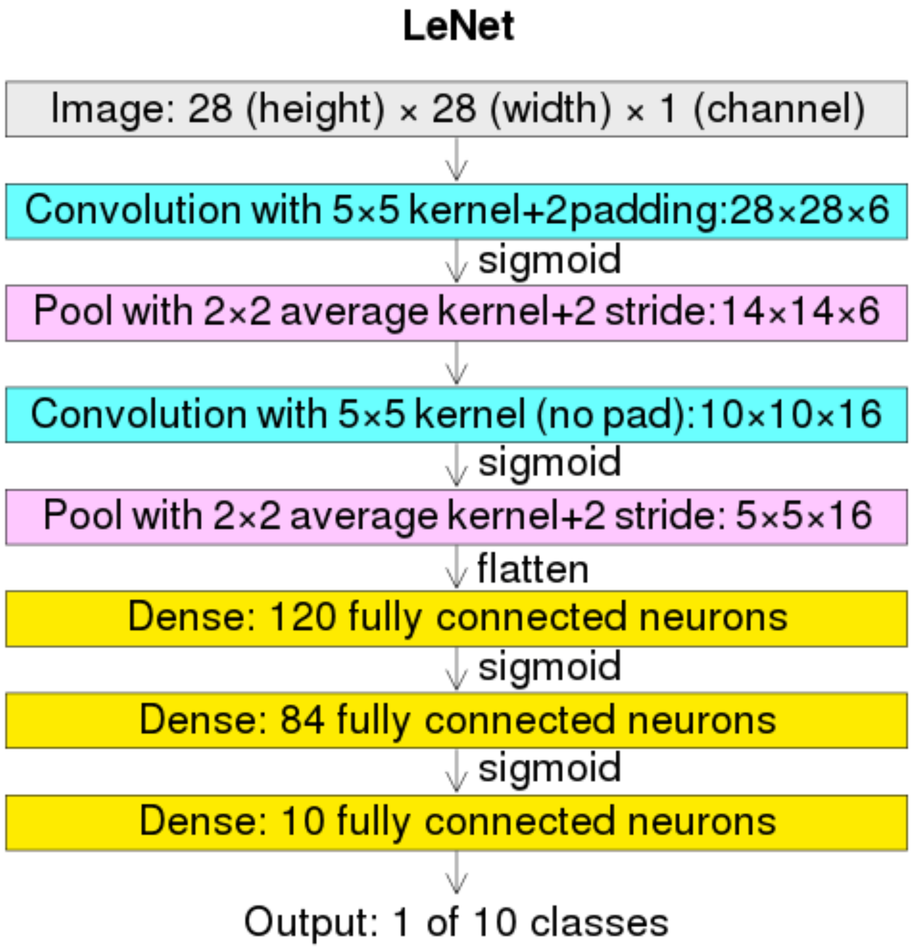


Image credit

ImageNet

ImageNet Large Scale Visual Recognition Challenge (ILSVRC):

- 1.2M цветных изображений
- 1K классов (ImageNet-1k)
- Данные из интернета (flikr), аннотированы Amazon MTurk

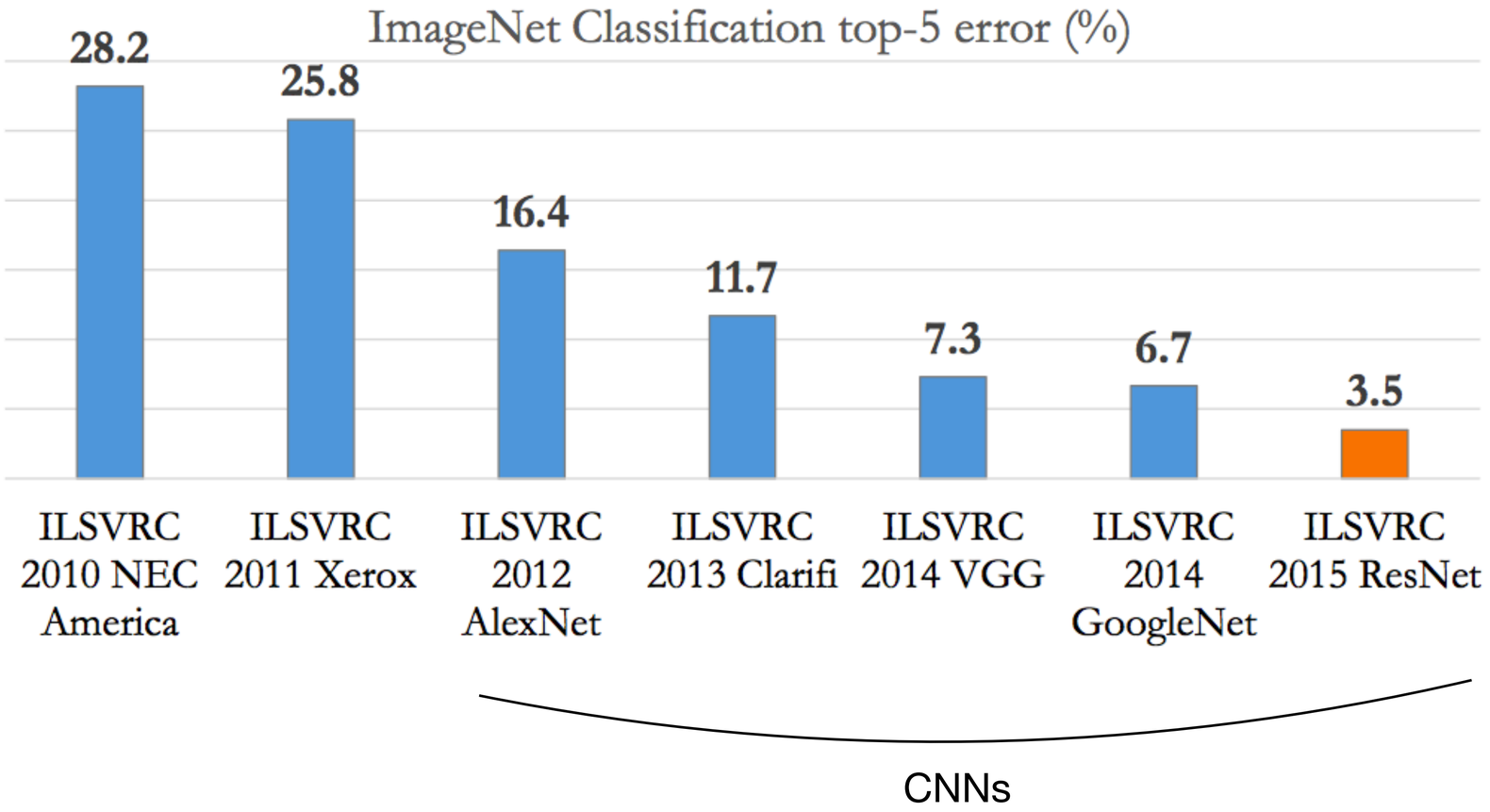
sailboat

husky

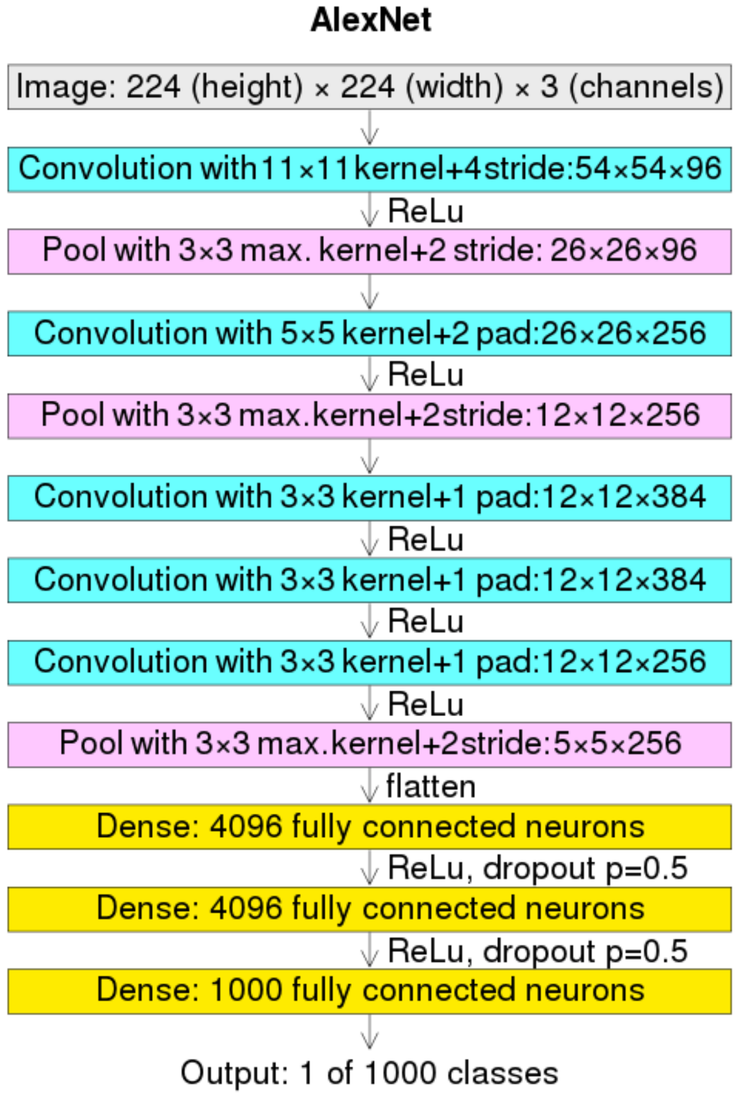
ImageNet

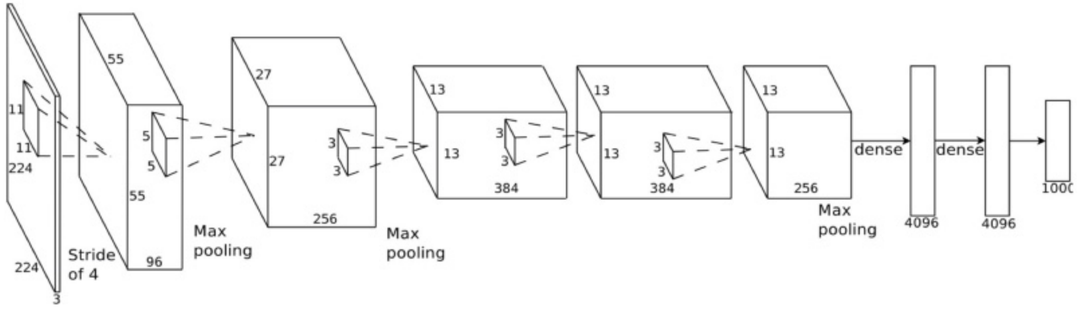
Метрики

- Top-1 accuracy
- Top-5 accuracy



AlexNet (Krizhevsky et al., 2012)

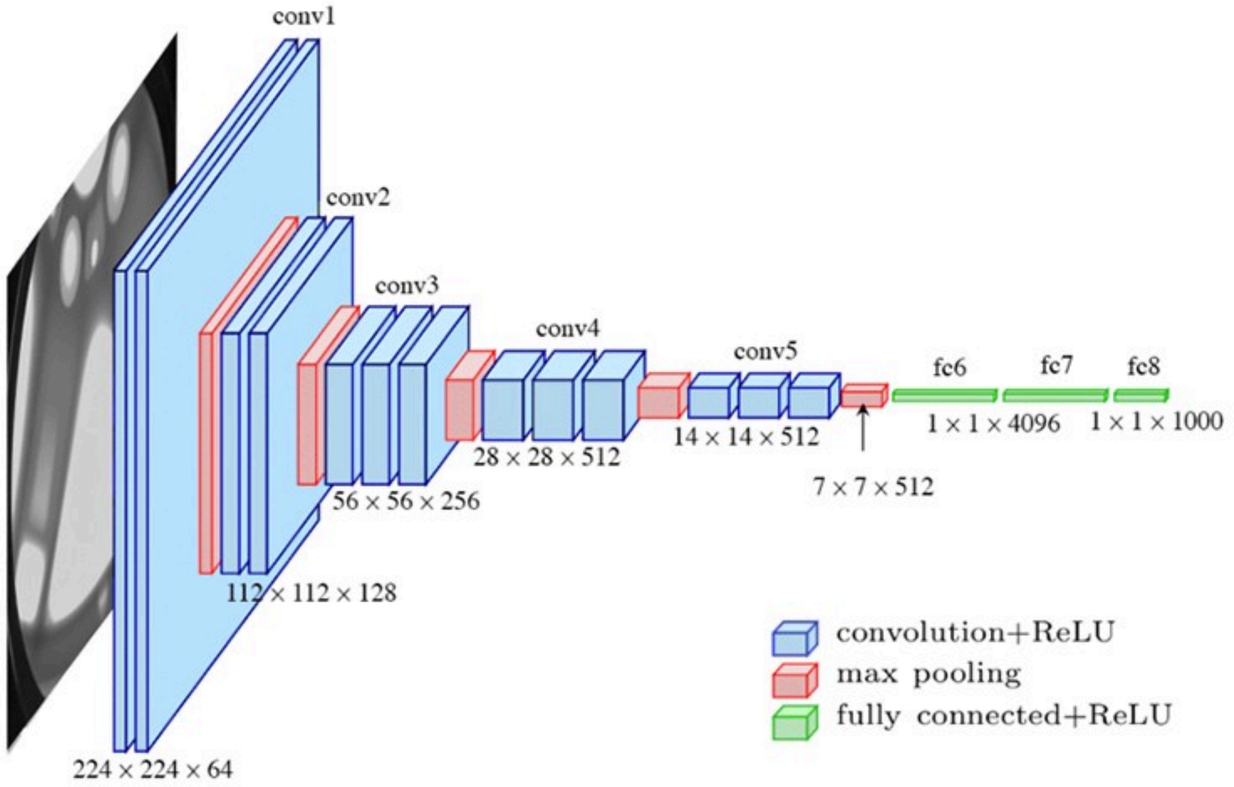




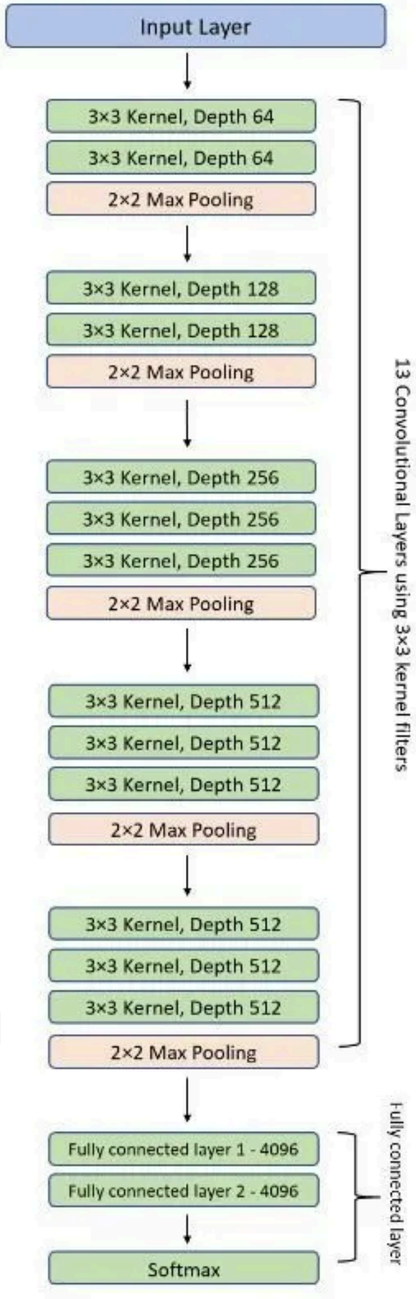
- ReLu, MaxPool
- 60M parameters
- Augmentations + Dropout
- GPU (50x speedup)
- 1 week of training on 2 GPUs

Image credit

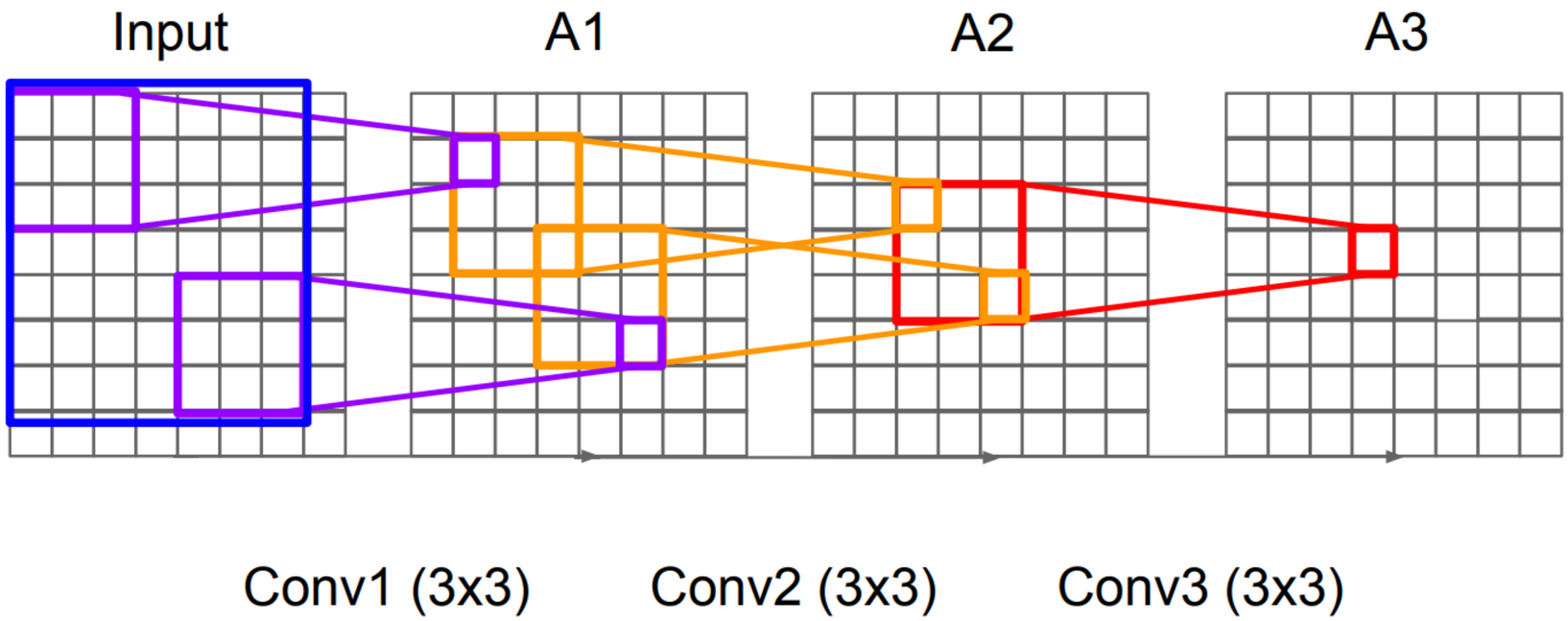
VGGNet (Simonyan and Zisserman, 2014)

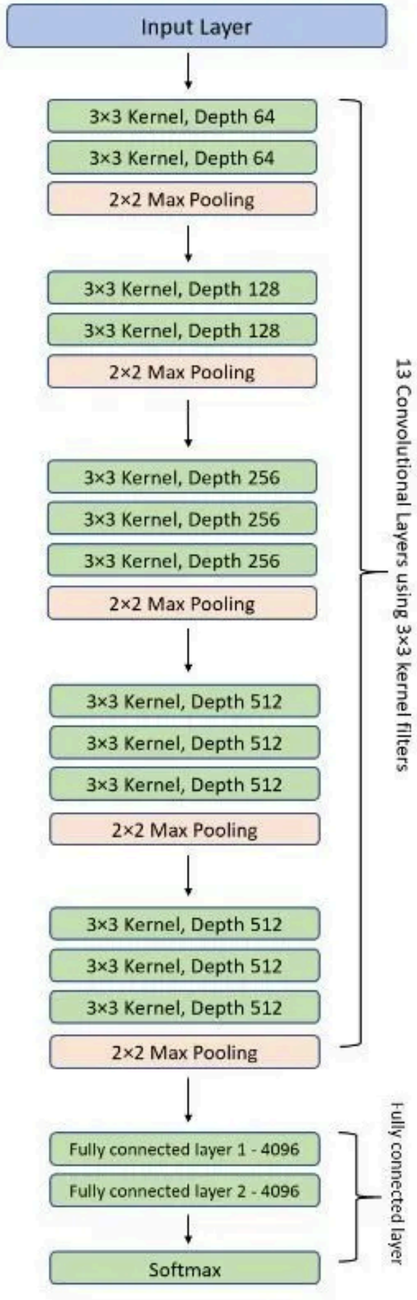


- Свертки 3x3
- Больше слоев
- 140M parameters
- Обучается стадиями



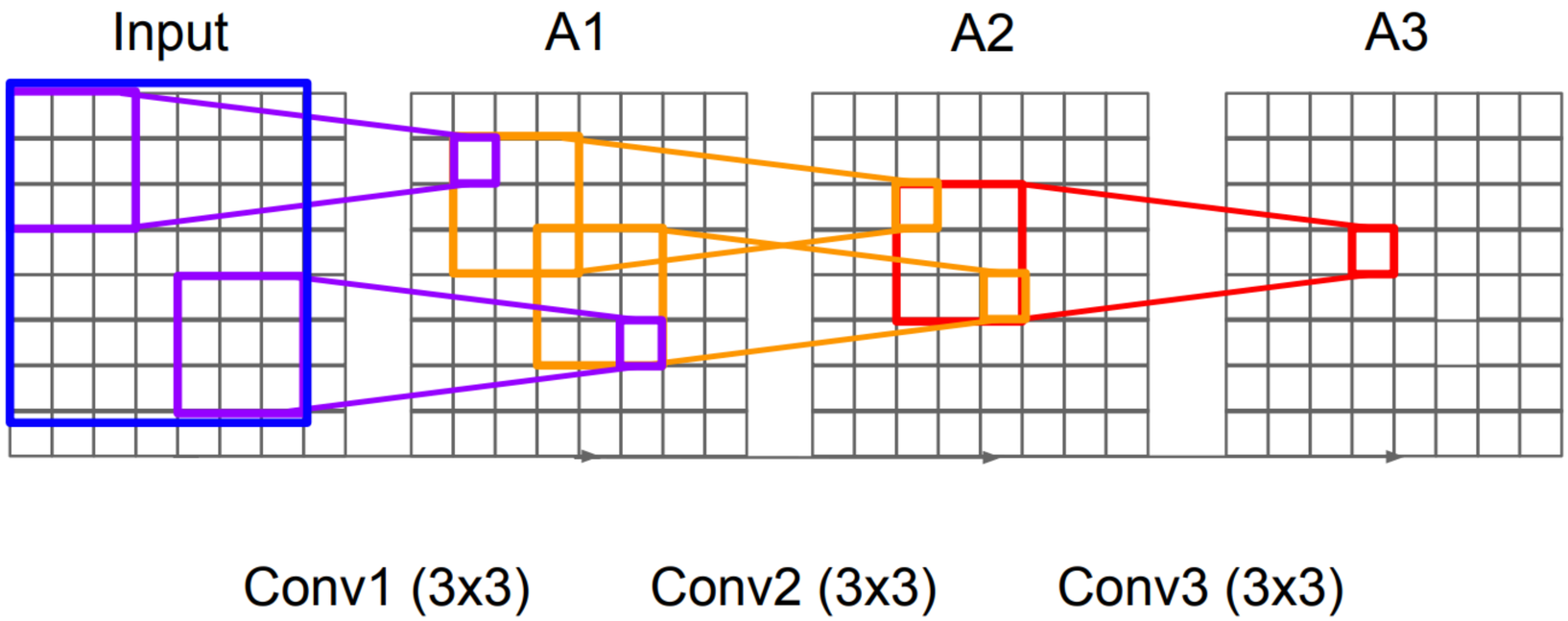
VGGNet (Simonyan and Zisserman, 2014)

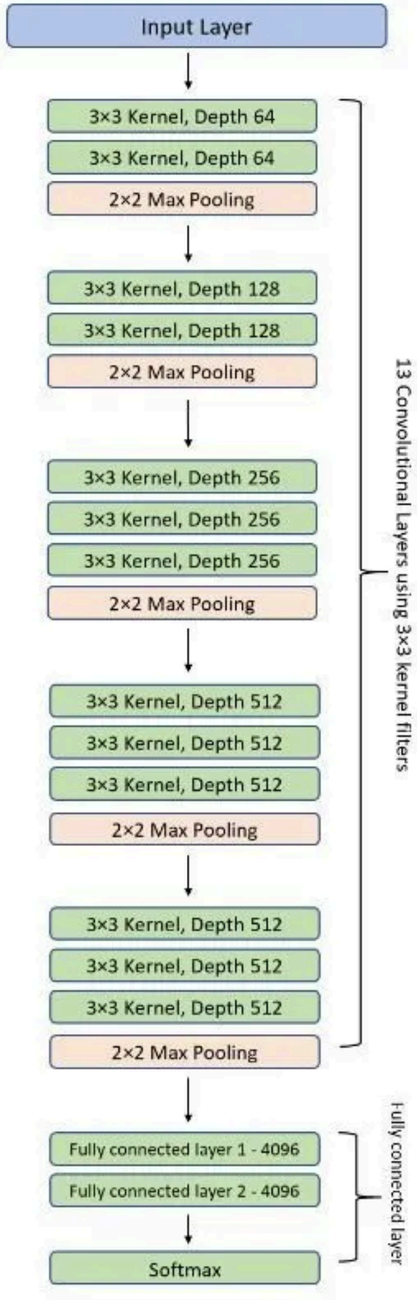




3 свертки 3x3 имеют receptive field как одна 7x7

VGGNet (Simonyan and Zisserman, 2014)

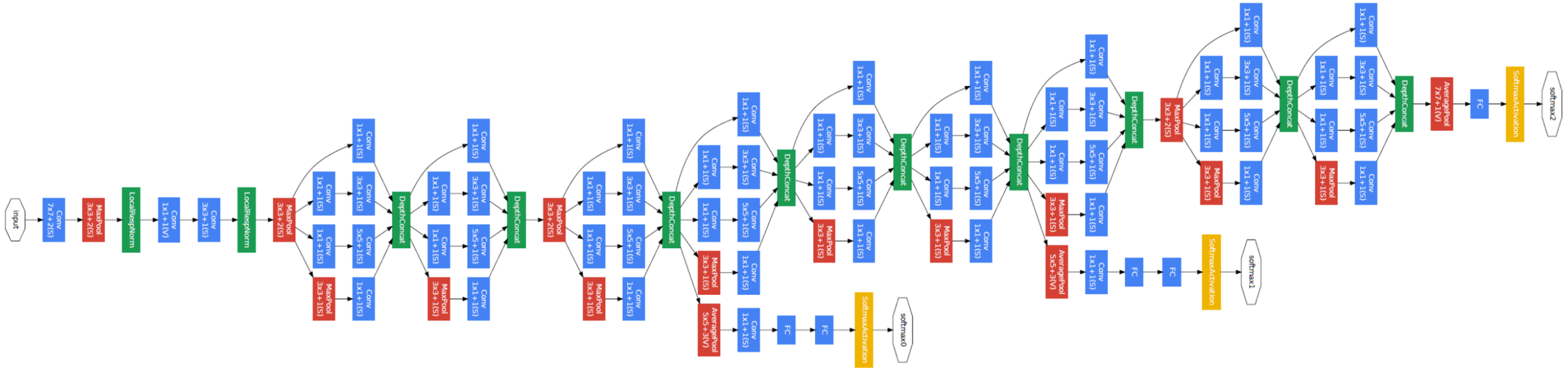




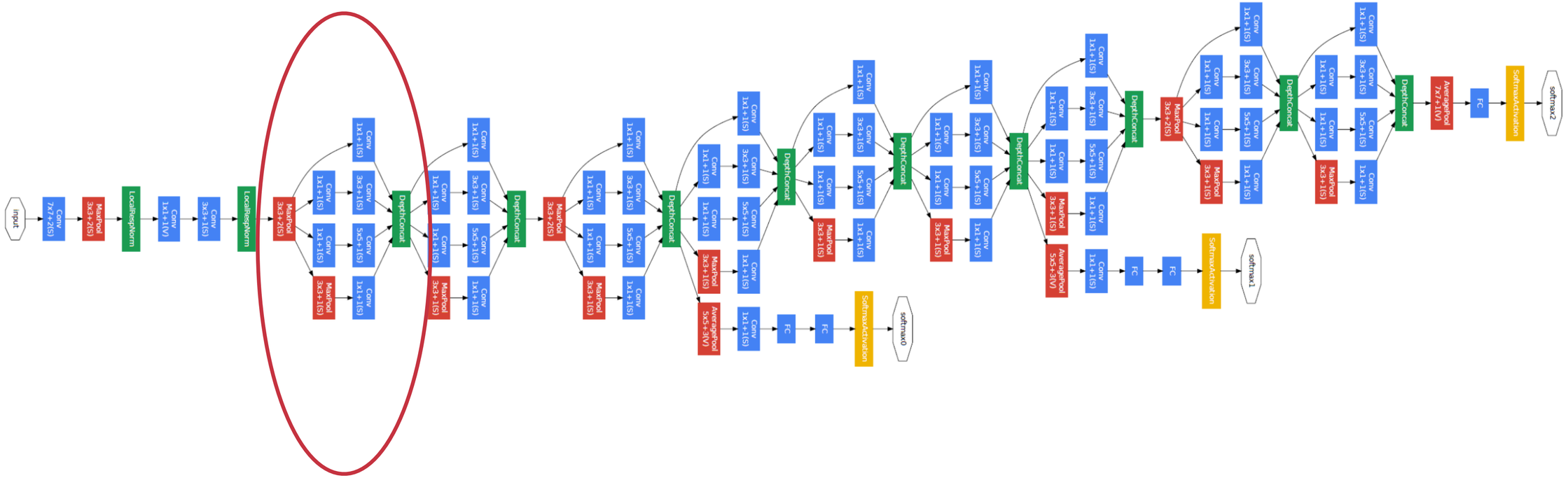
3 свертки 3x3 имеют receptive field как одна 7x7

Меньше параметров, можно сделать глубже сеть

GoogLeNet (Szegedy et al., 2014)

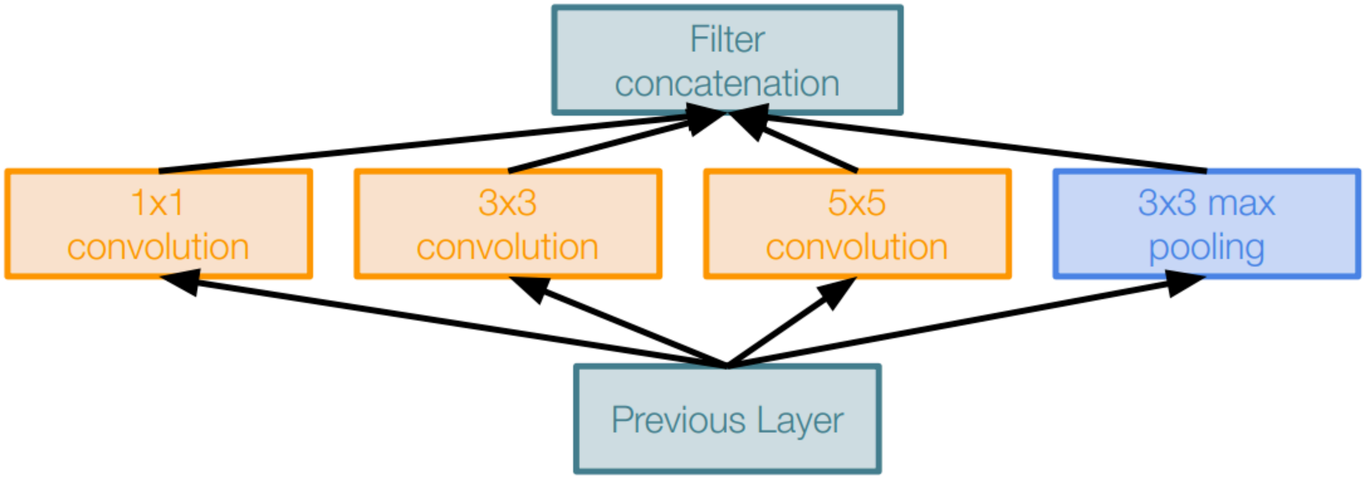


GoogLeNet (Szegedy et al., 2014)



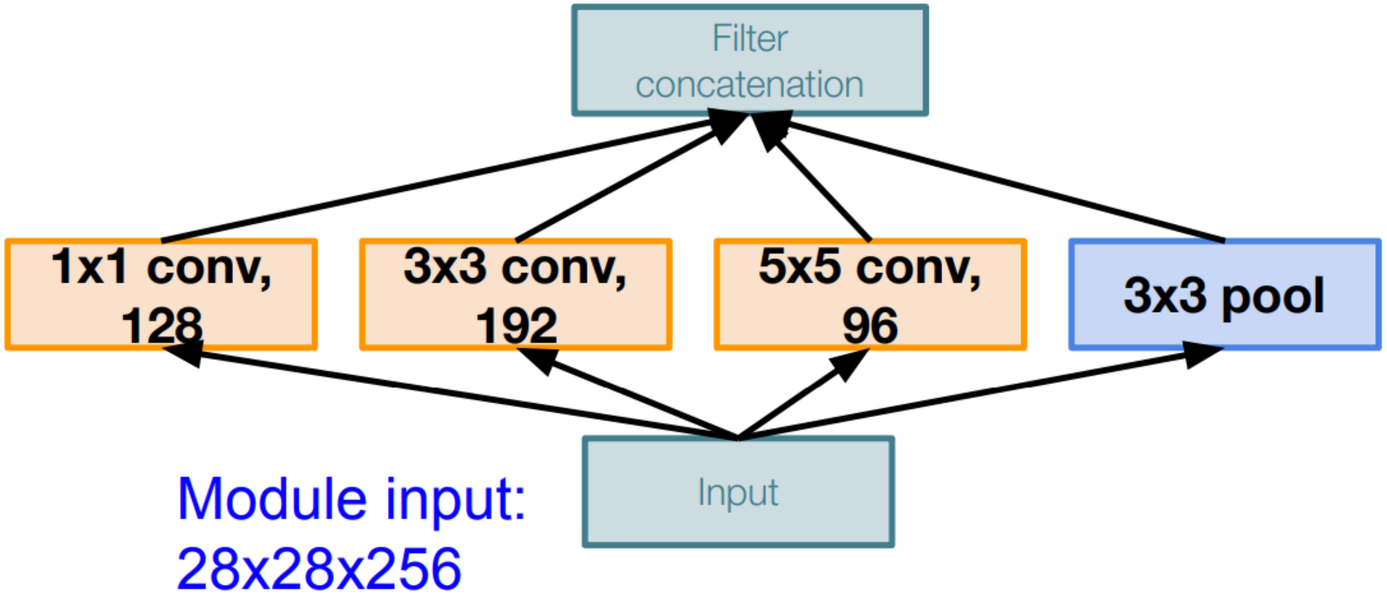
inception module

GoogLeNet (Szegedy et al., 2014)



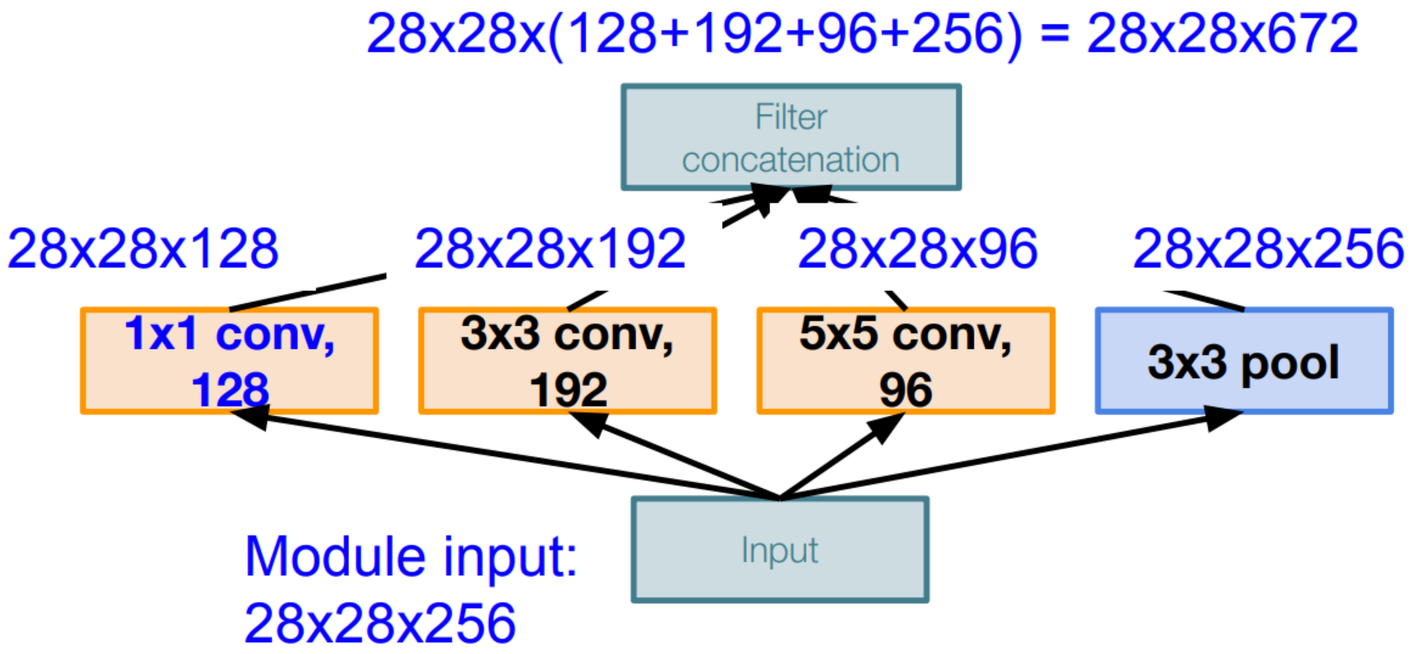
Naive Inception module

GoogLeNet (Szegedy et al., 2014)



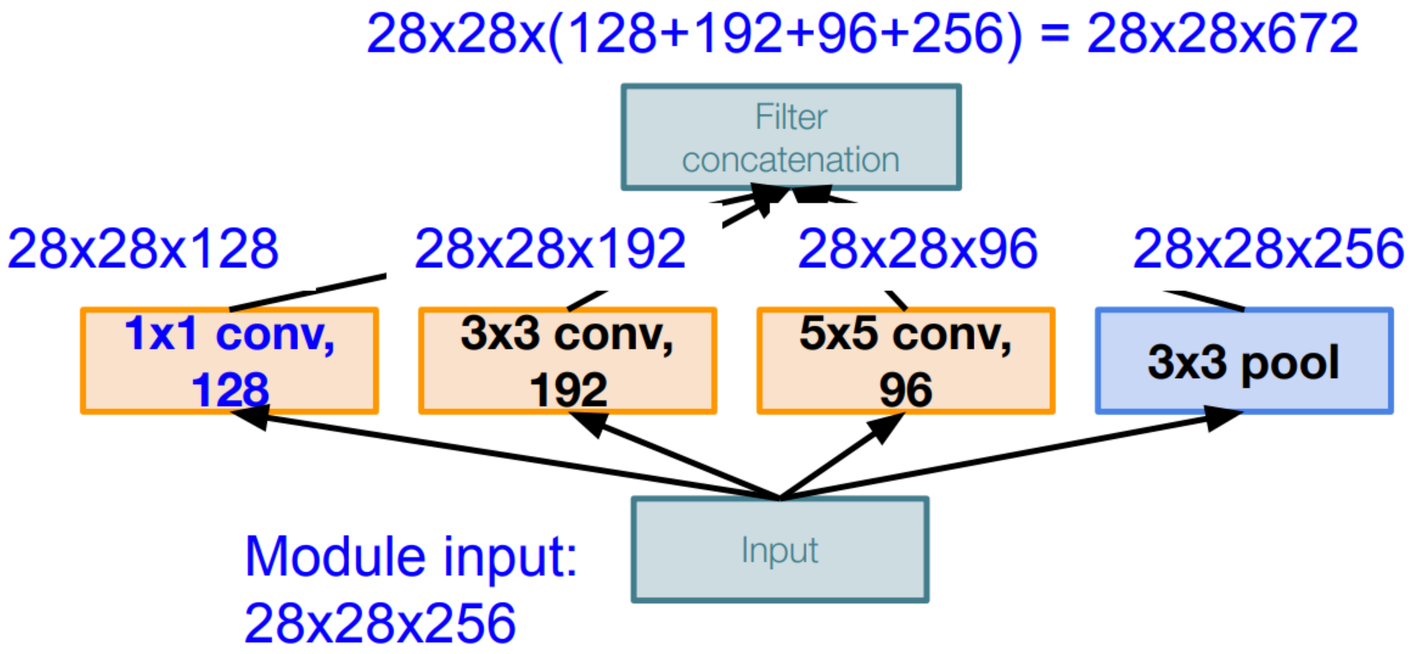
Вычислительная сложность?

GoogLeNet (Szegedy et al., 2014)



Вычислительная сложность?

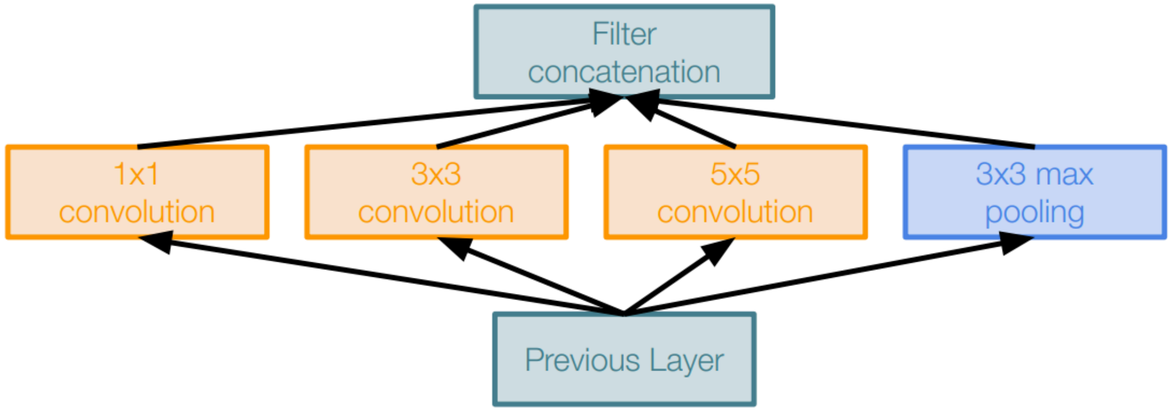
GoogLeNet (Szegedy et al., 2014)



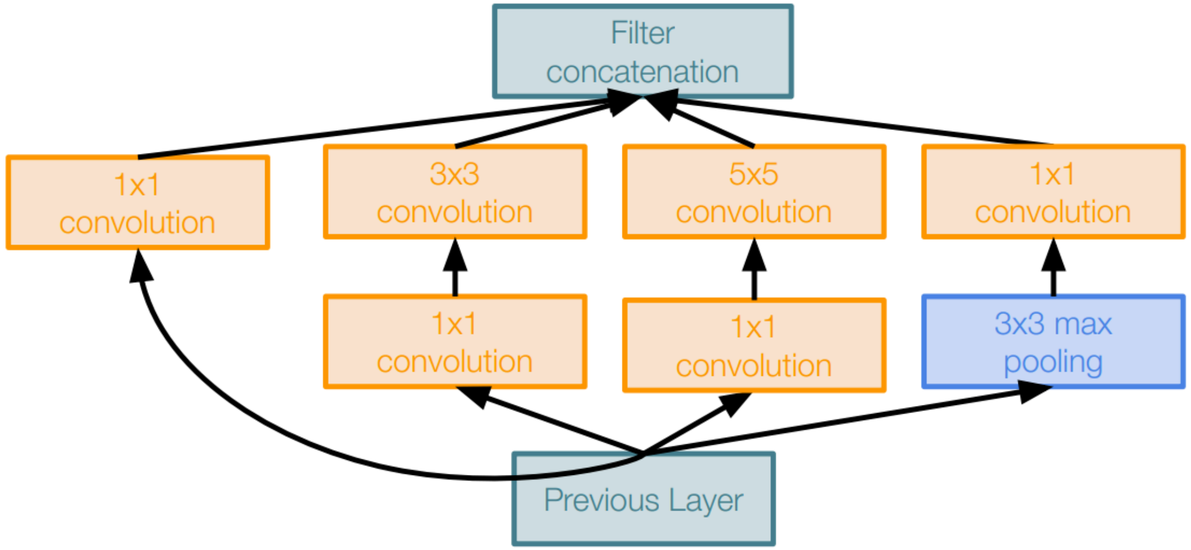
Conv Ops:
[1x1 conv, 128] 28x28x128x1x1x256
[3x3 conv, 192] 28x28x**192x3x3x256**
[5x5 conv, 96] 28x28x**96x5x5x256**
Total: 854M ops

Вычислительная сложность?

GoogLeNet (Szegedy et al., 2014)

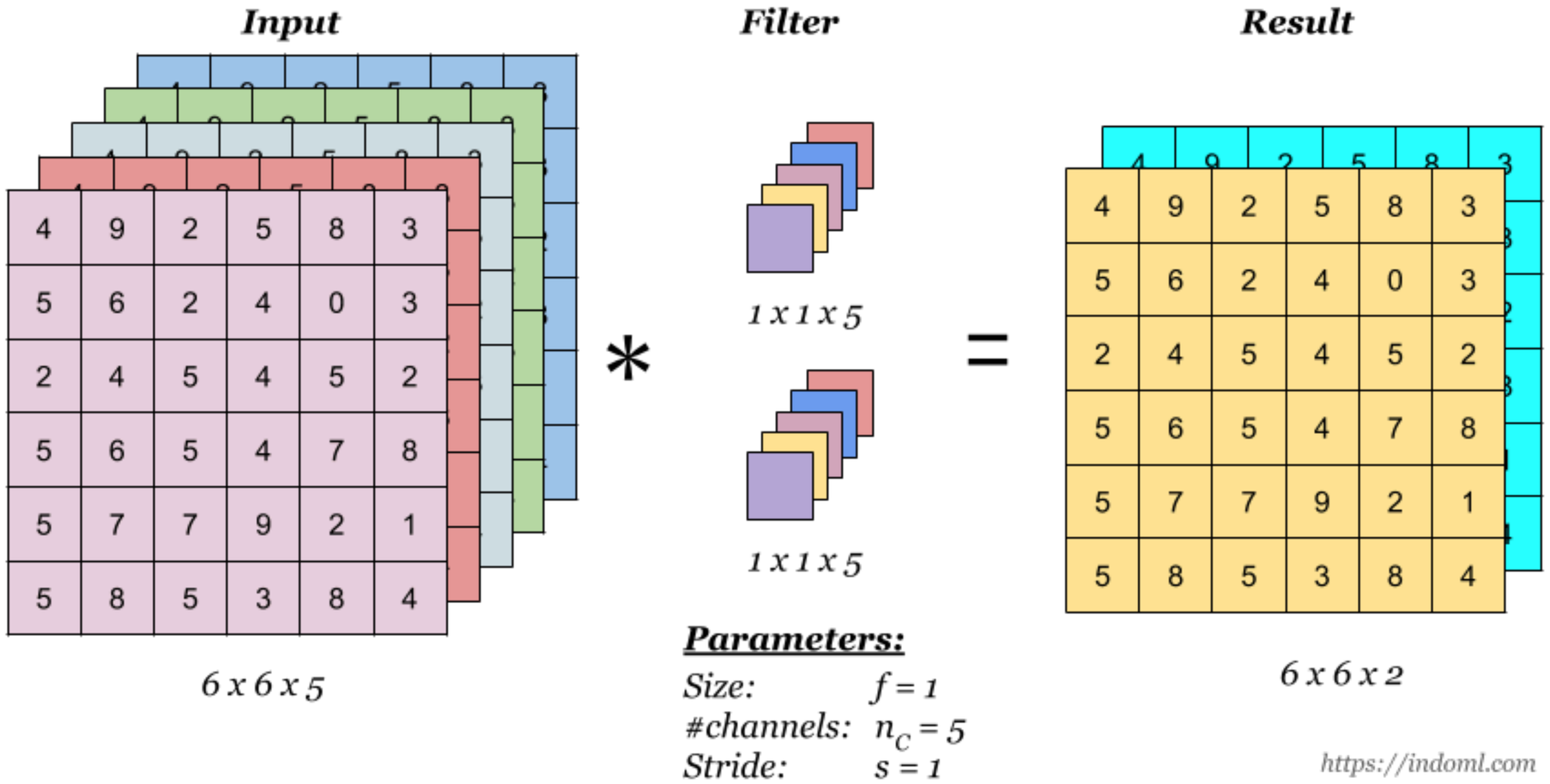


Naive Inception module

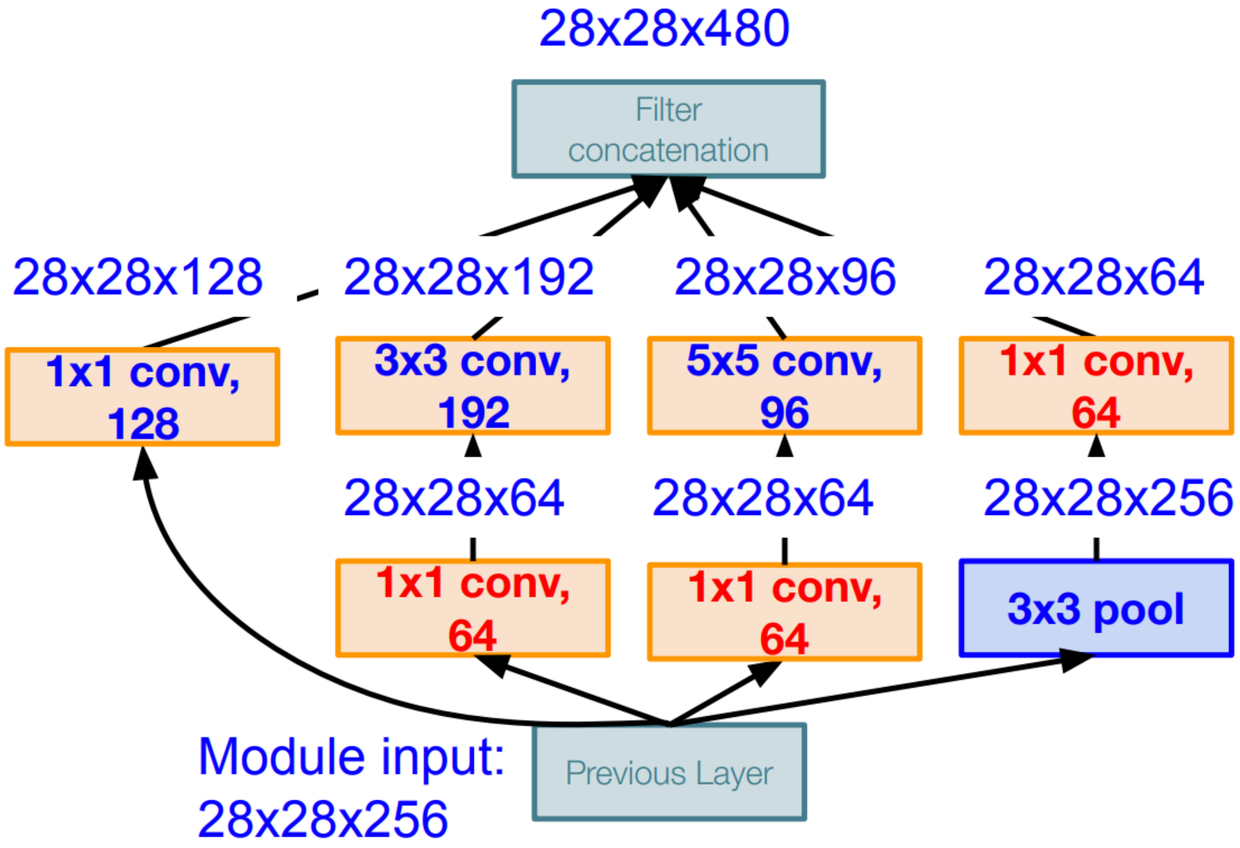


Inception module with dimension reduction

GoogLeNet (Szegedy et al., 2014)

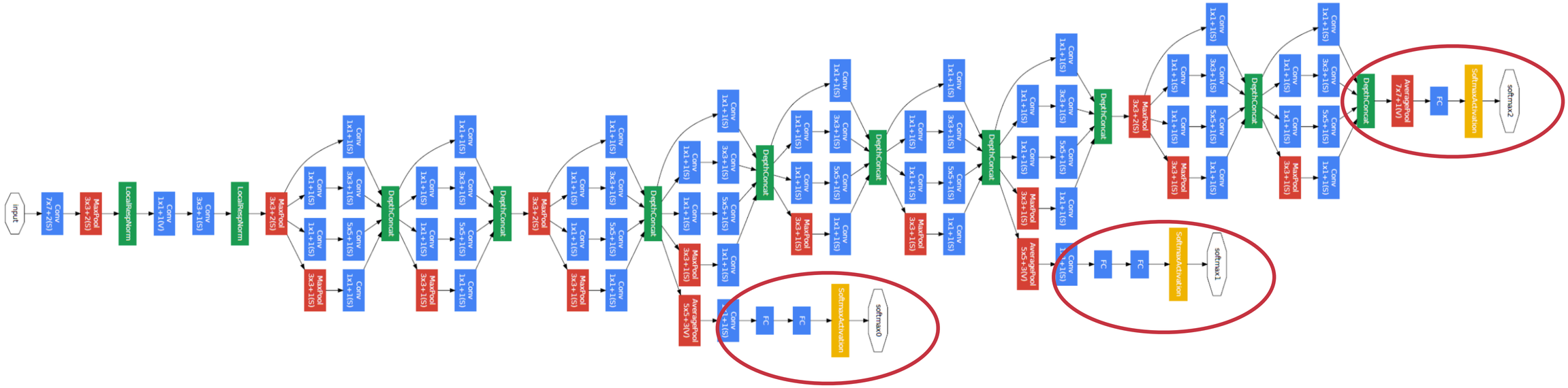


GoogLeNet (Szegedy et al., 2014)



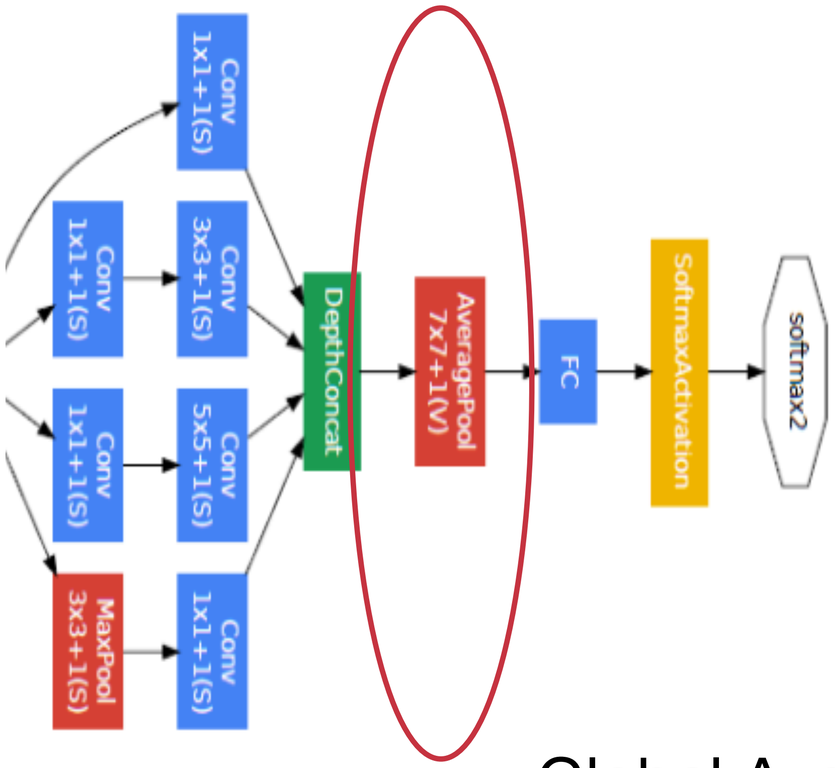
Conv Ops:
[1x1 conv, 64] 28x28x64x1x1x256
[1x1 conv, 64] 28x28x64x1x1x256
[1x1 conv, 128] 28x28x128x1x1x256
[3x3 conv, 192] 28x28x192x3x3x64
[5x5 conv, 96] 28x28x96x5x5x64
[1x1 conv, 64] 28x28x64x1x1x256
Total: 358M ops

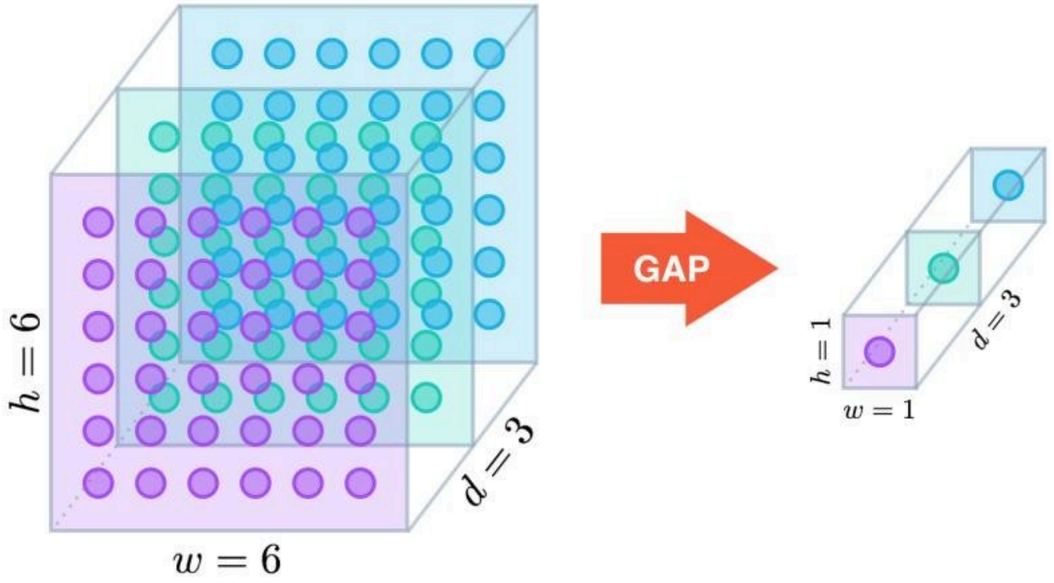
GoogLeNet (Szegedy et al., 2014)



вспомогательные выходы для обучения

GoogLeNet (Szegedy et al., 2014)





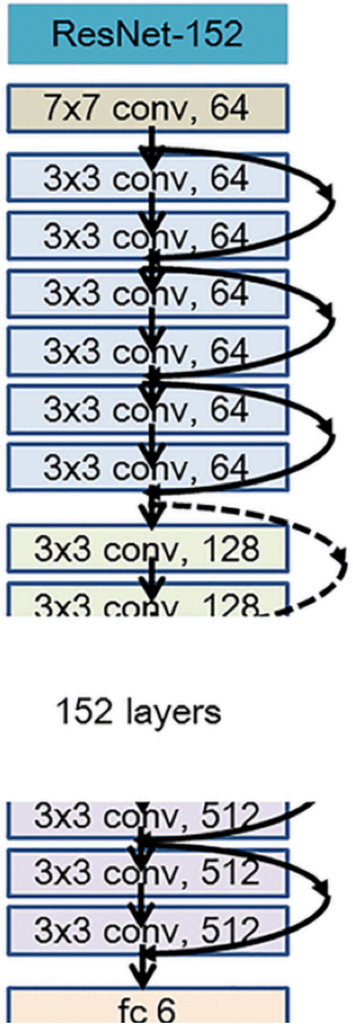
Global AvgPool

GoogLeNet (Szegedy et al., 2014)

type	patch size/ stride	output size	depth
convolution	7×7/2	112×112×64	1
max pool	3×3/2	56×56×64	0
convolution	3×3/1	56×56×192	2
max pool	3×3/2	28×28×192	0
inception (3a)		28×28×256	2
inception (3b)		28×28×480	2
max pool	3×3/2	14×14×480	0
inception (4a)		14×14×512	2
inception (4b)		14×14×512	2
inception (4c)		14×14×512	2
inception (4d)		14×14×528	2
inception (4e)		14×14×832	2
max pool	3×3/2	7×7×832	0
inception (5a)		7×7×832	2
inception (5b)		7×7×1024	2
avg pool	7×7/1	1×1×1024	0
dropout (40%)		1×1×1024	0
linear		1×1×1000	1
softmax		1×1×1000	0

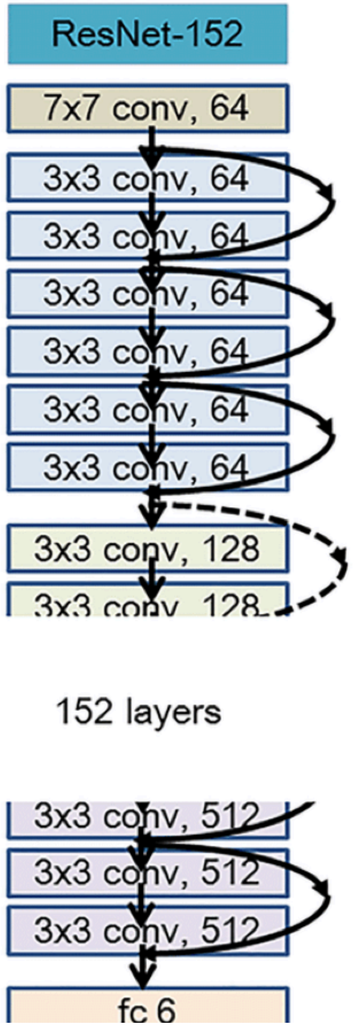
- Нелинейная архитектура
- 22 слоя
- ~5M parameters

ResNet (He et al., 2015)

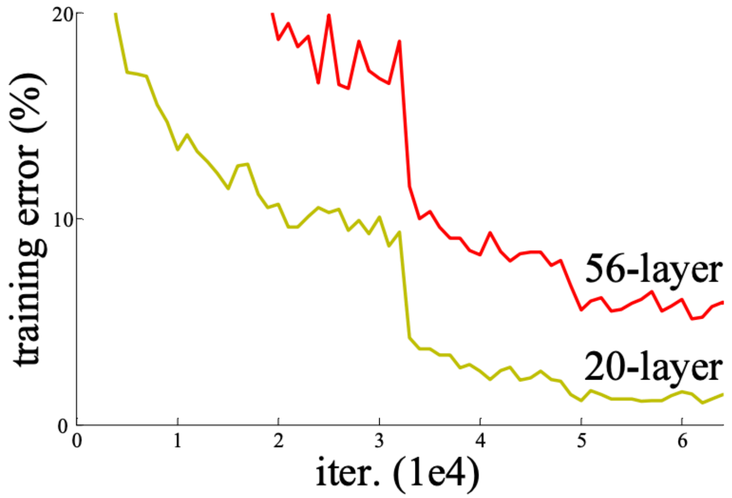


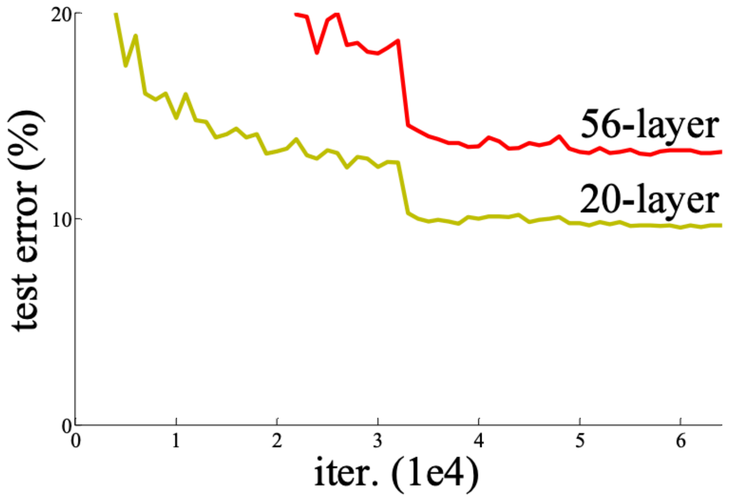
152 слоя!

ResNet (He et al., 2015)



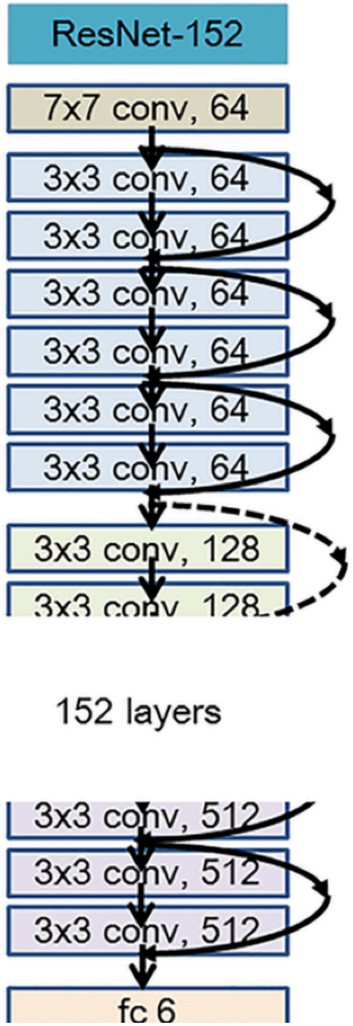
152 слоя!



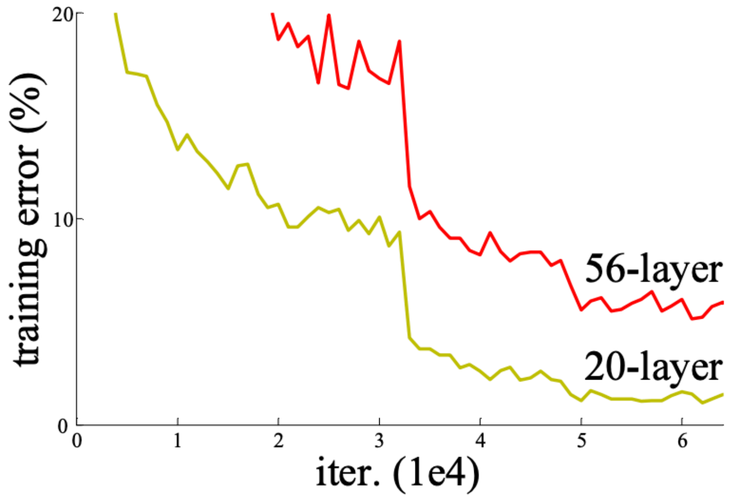


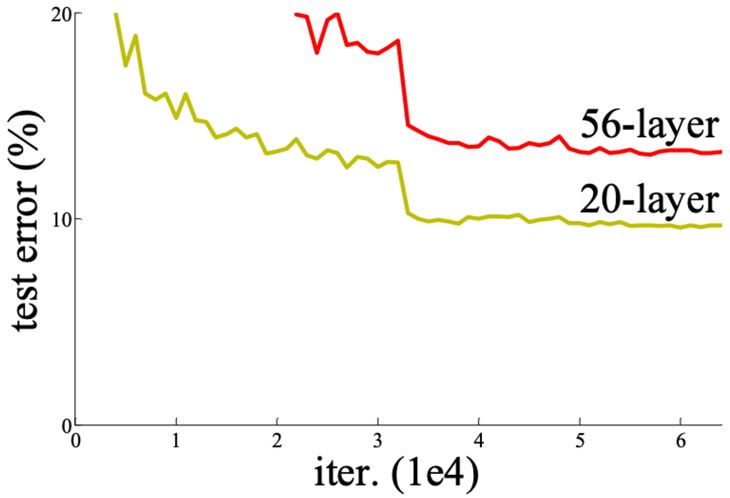
Просто увеличить число слоев не поможет

ResNet (He et al., 2015)



152 слоя!

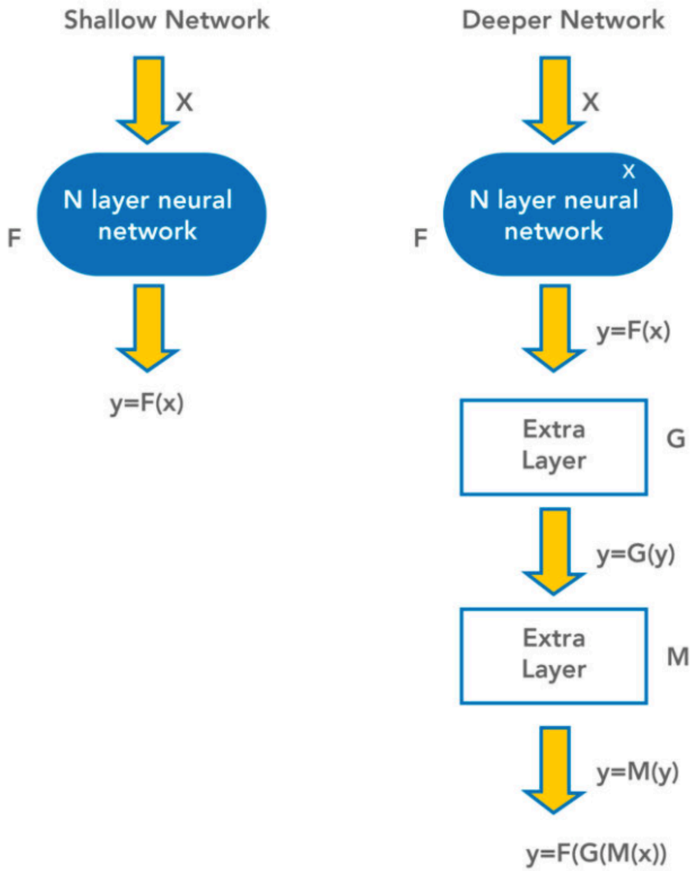




Просто увеличить число слоев не поможет
- слишком сложно обучать (оптимизировать)

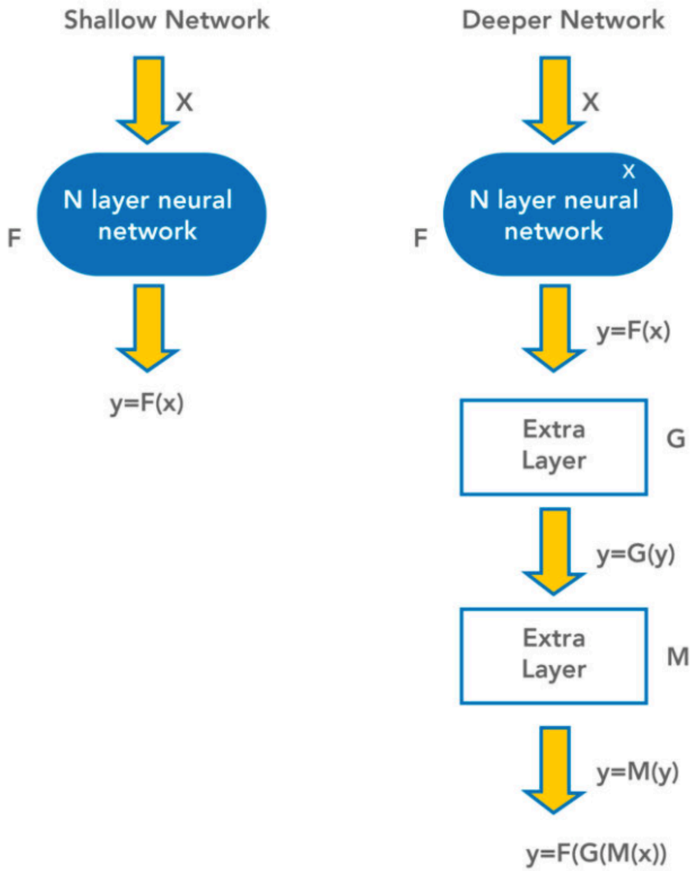
ResNet (He et al., 2015)

Как не испортить неглубокую нейросеть, добавляя слои?



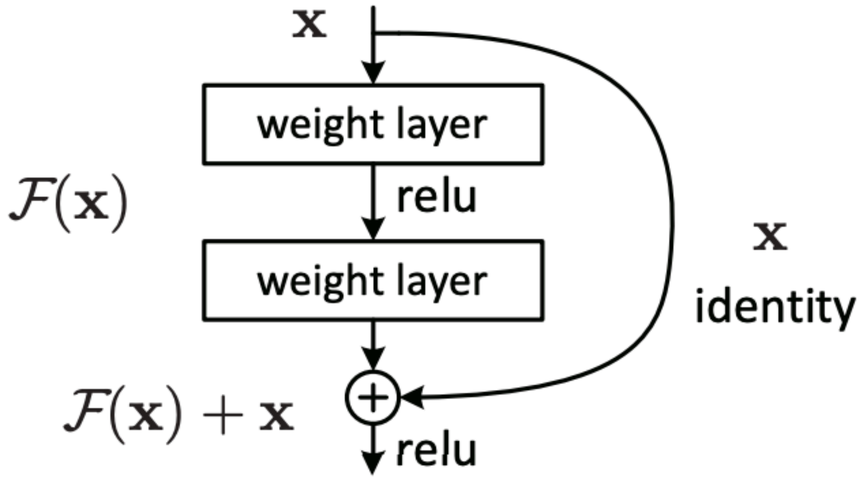
ResNet (He et al., 2015)

Как не испортить неглубокую нейросеть, добавляя слои? Identity blocks



ResNet (He et al., 2015)

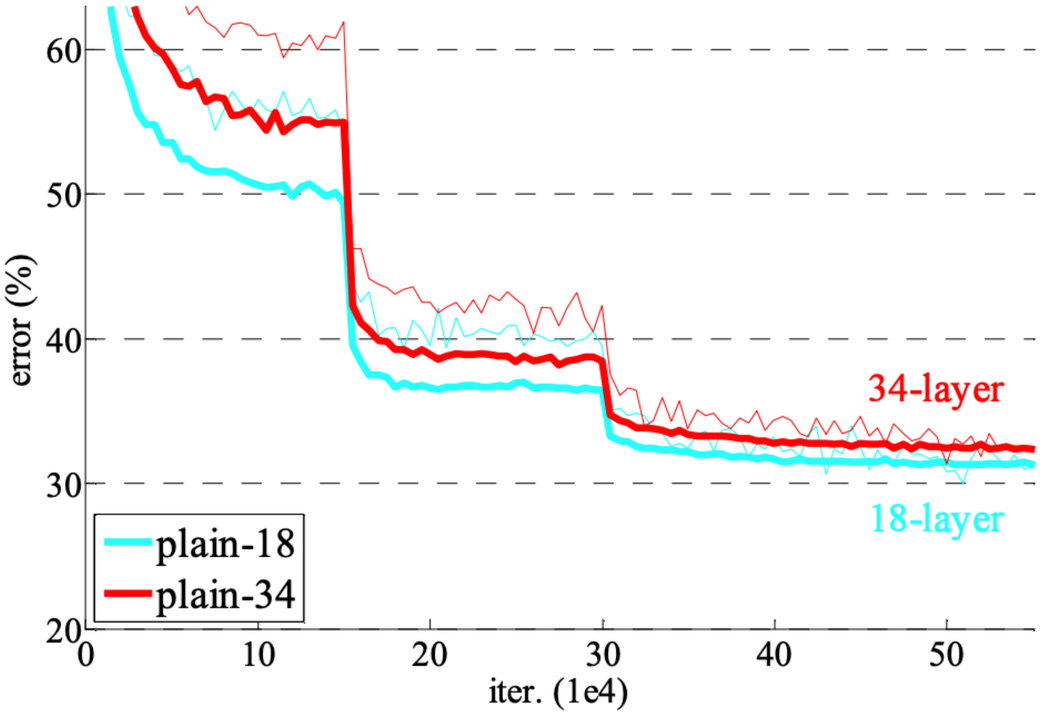
Residual block

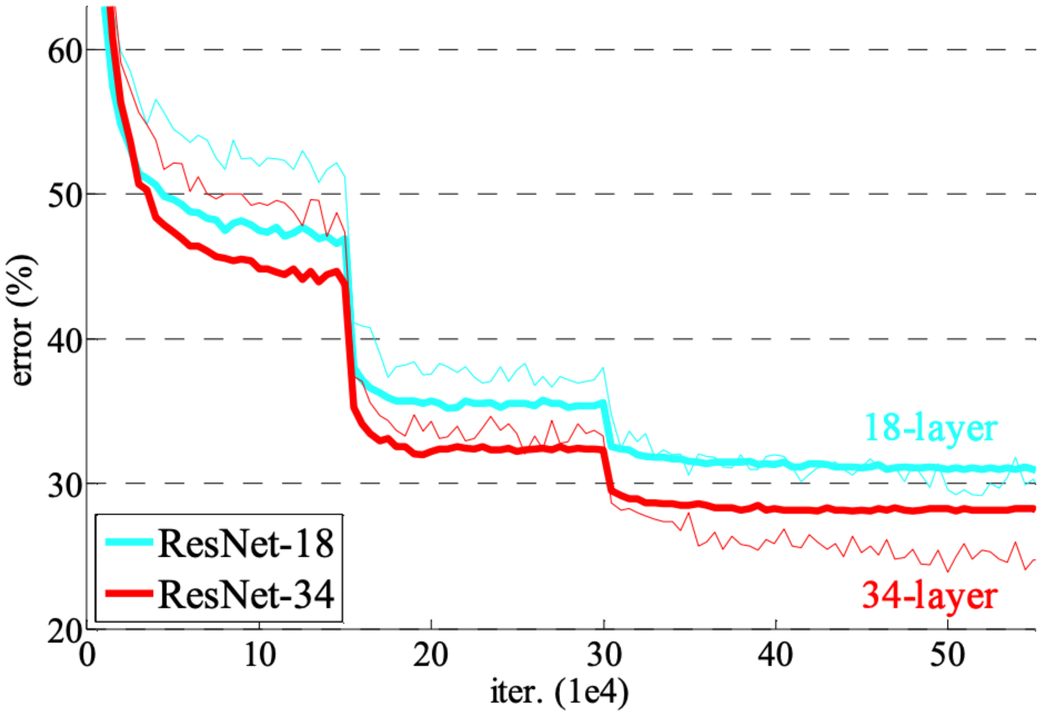


$H(x) = F(x) + x$

$H(x) = x$ if $F(x) = 0$

ResNet (He et al., 2015)





ResNet (He et al., 2015)

- Skip-connections
- Разные варианты (от 34 до 152 слоев)
- BatchNorm, Xavier init
- Dropout не используется