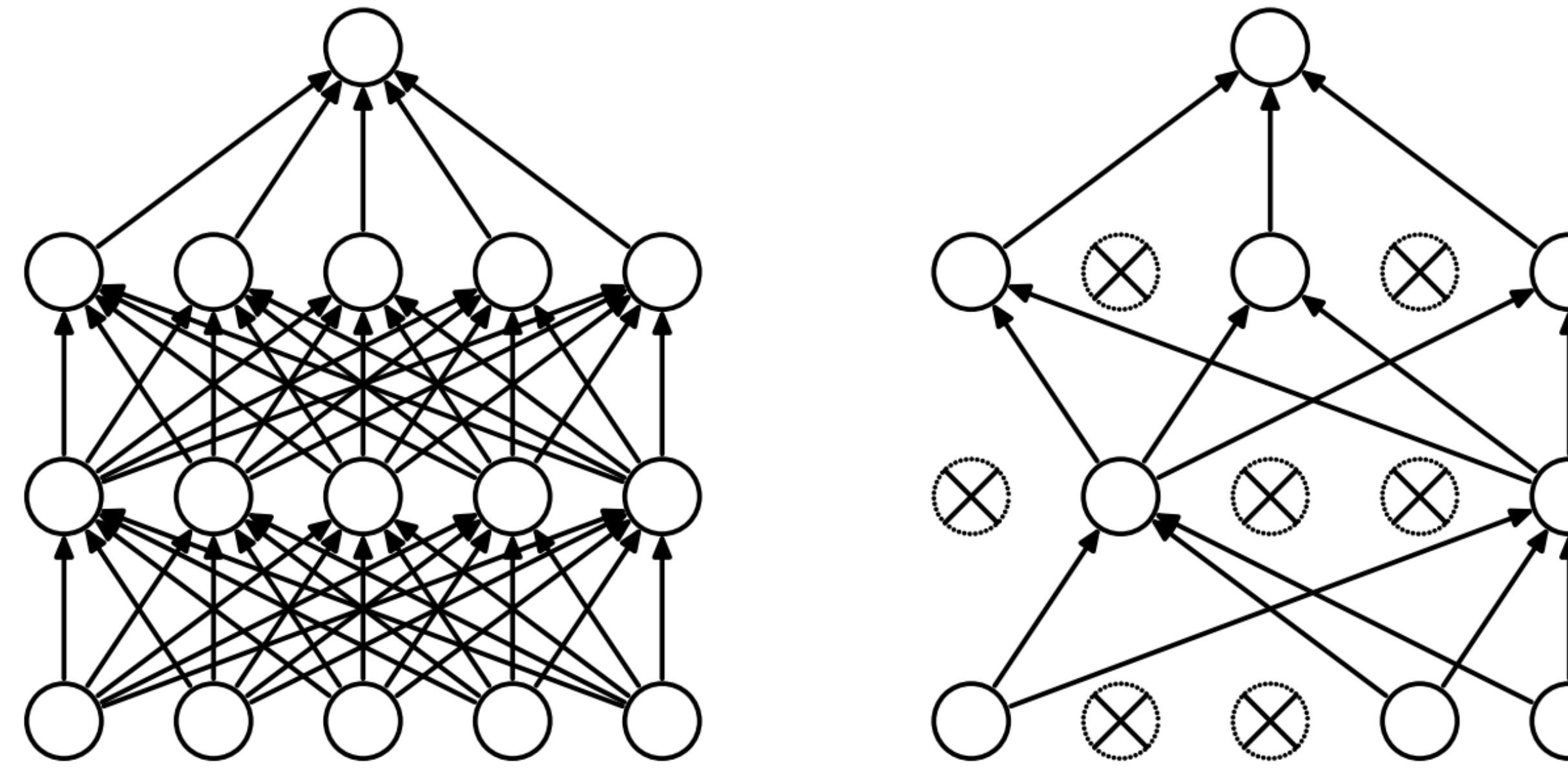


Глубинное обучение

Оптимизация. Обучение нейросетей.

Ирина Сапарина

Dropout

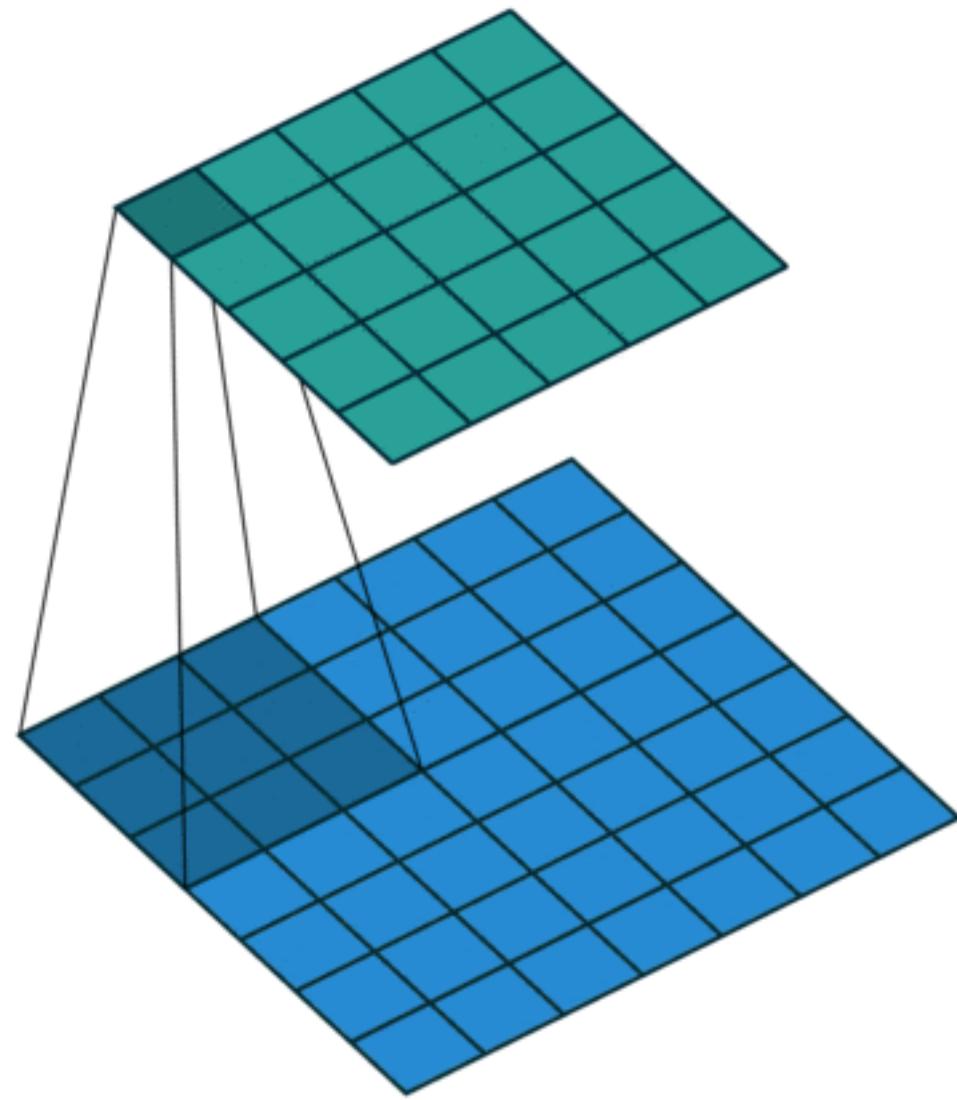


[Image credit](#)

Случайно выкидываем нейроны на forward pass с **вероятностью** p
Обучение: умножаем оставшиеся нейроны на $\frac{1}{1-p}$

Сверточные нейросети (CNN)

Свертка

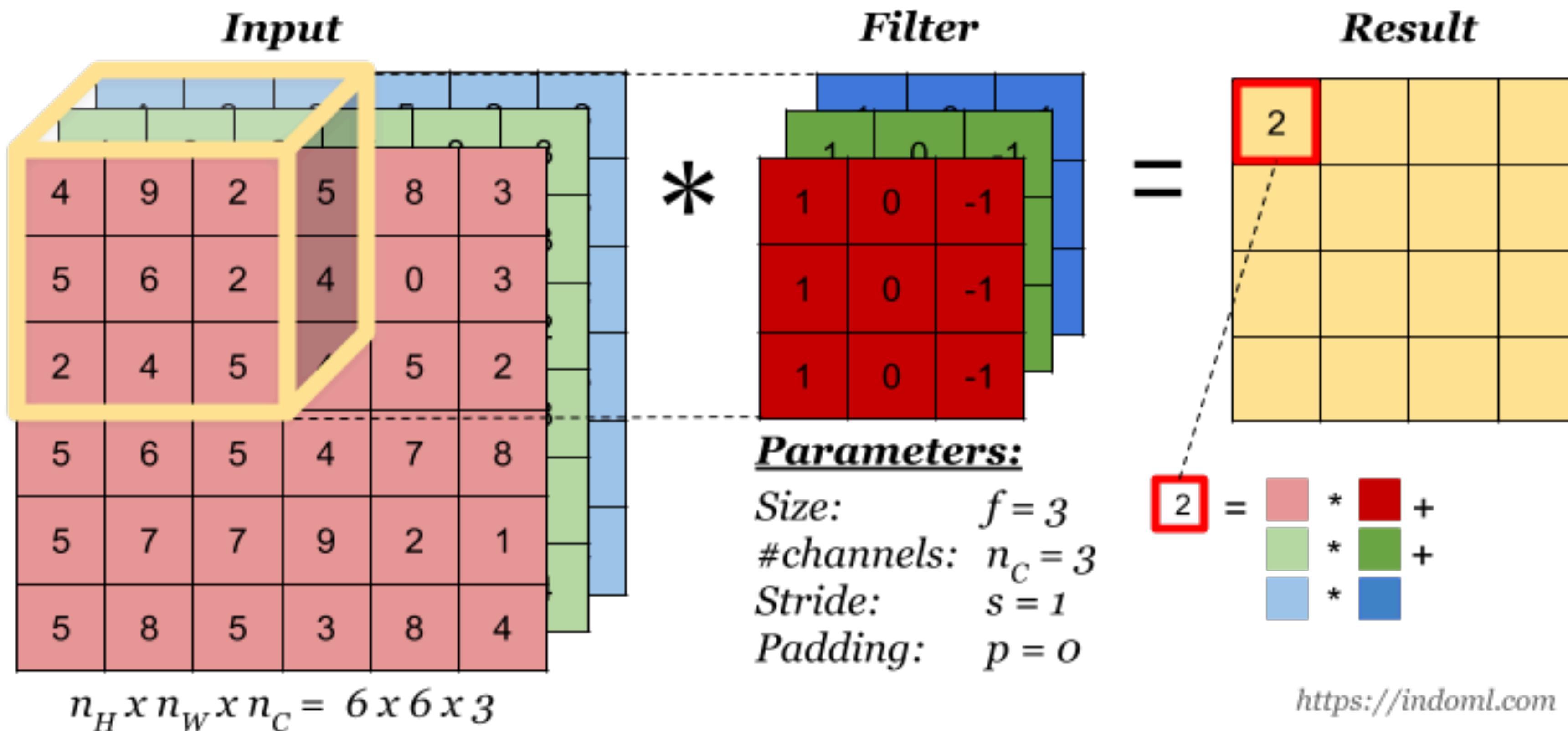


Параметры: ядро (kernel) - w
размер - $K \times K \times C_{in} \times C_{out}$
+ bias размера C_{out}

Много параметров: пропуски, шаги, ...

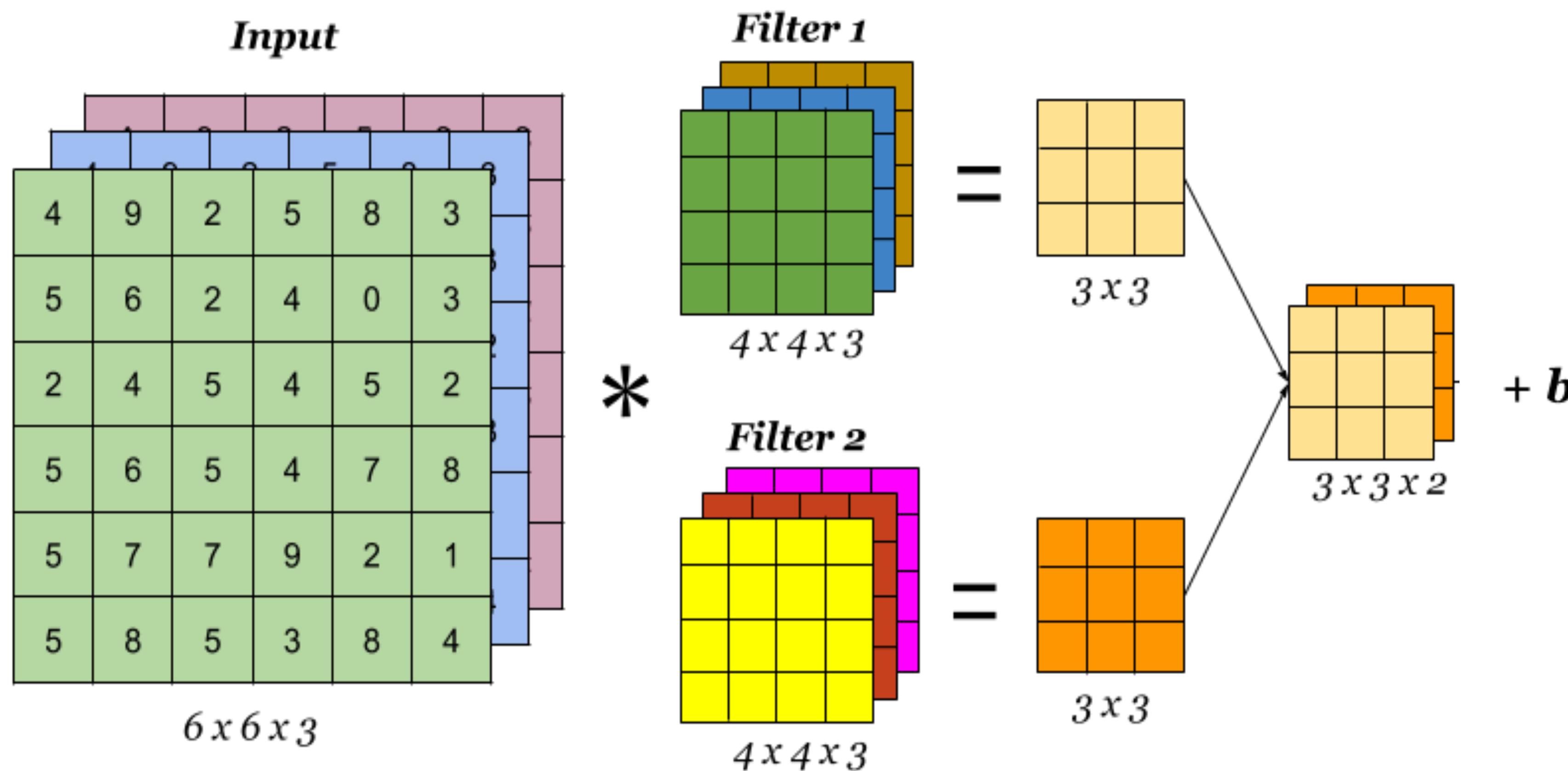
$$H_{in} \times W_{in} \times C_{in} \rightarrow H_{output} \times W_{output} \times C_{output}$$

Свертка

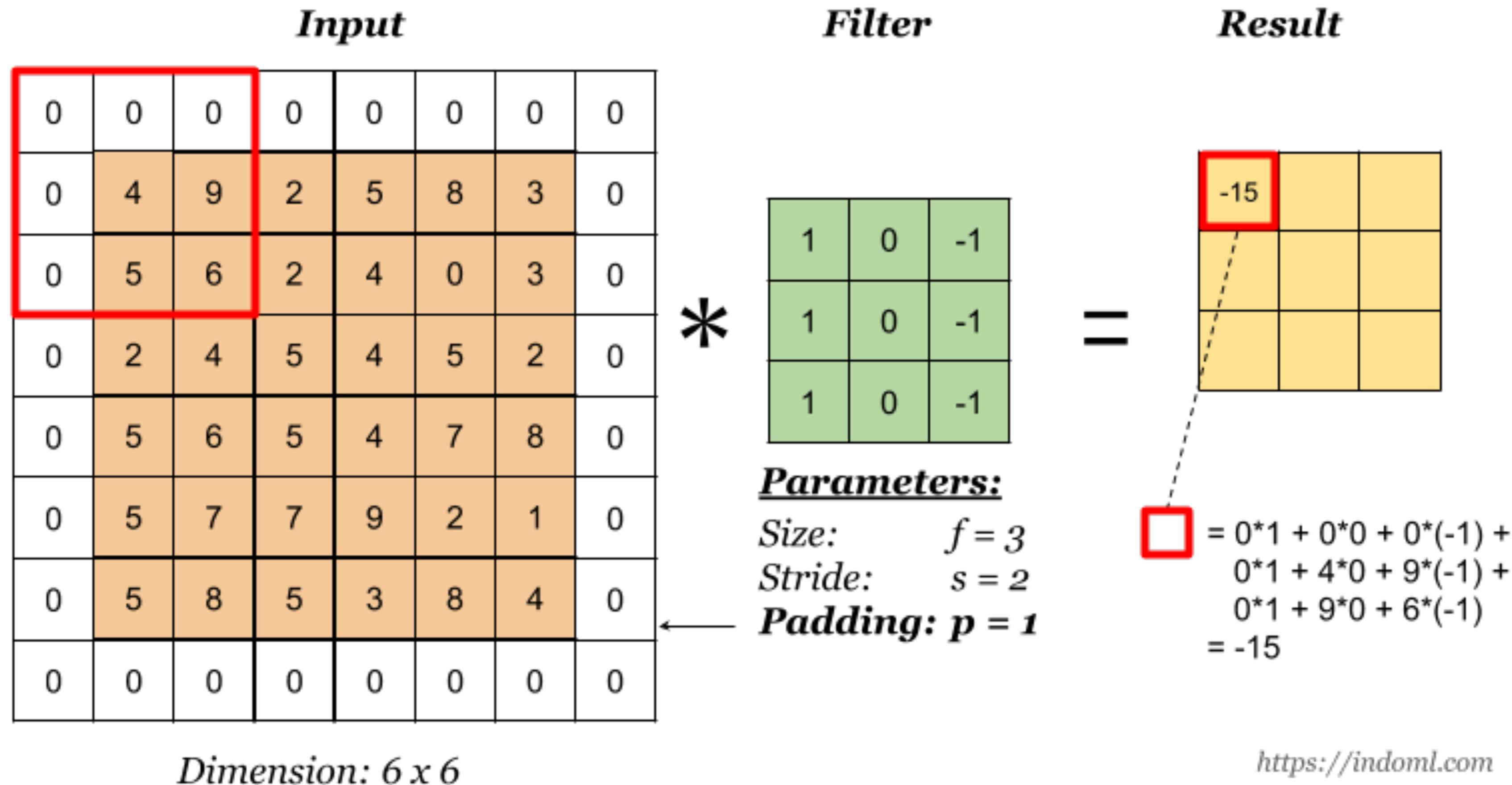


<https://indoml.com>

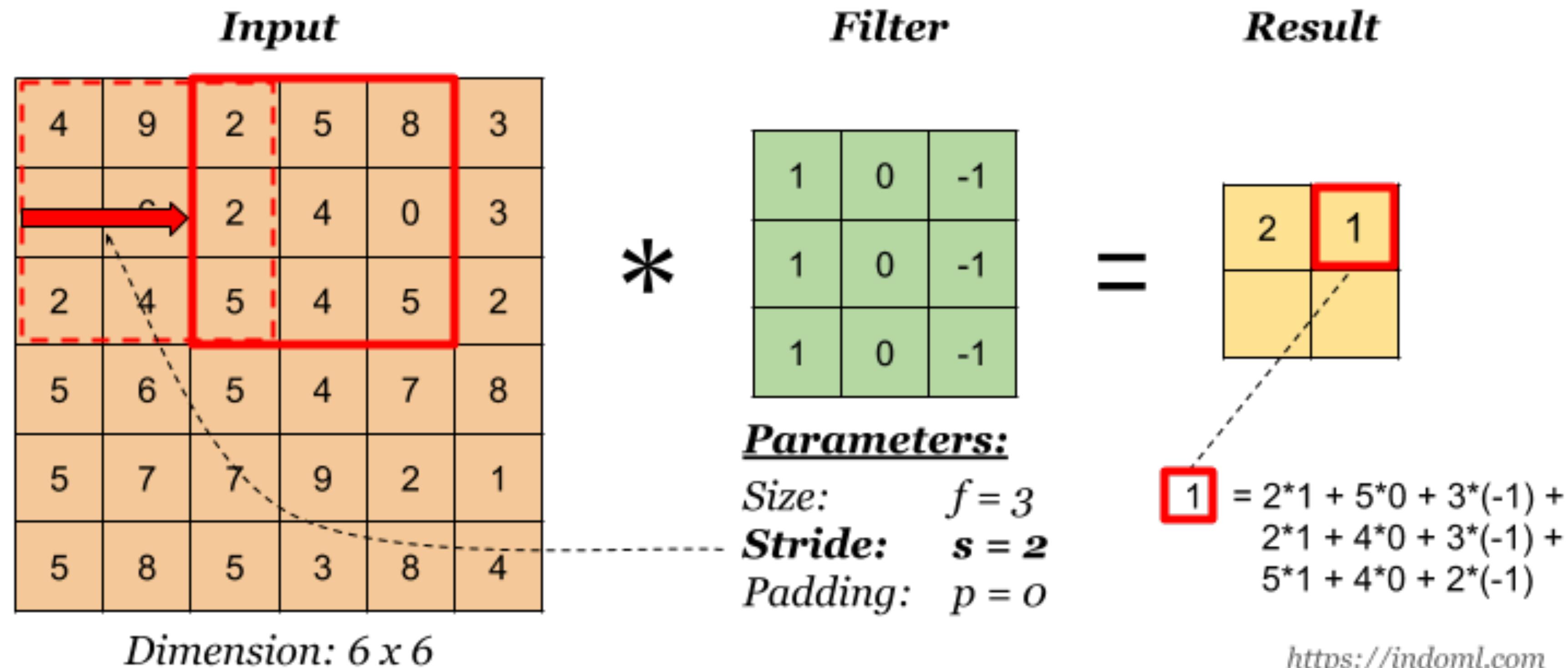
Свертка



Свертка: padding



Свертка: stride



<https://indoml.com>

Свертка как перемножение матриц

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

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

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

$$\begin{bmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{bmatrix}$$

- circulant matrix

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

$$= \begin{bmatrix} k_1x_1 + k_2x_2 + k_3x_4 + k_4x_5 \\ k_1x_2 + k_2x_3 + k_3x_5 + k_4x_6 \\ k_1x_4 + k_2x_5 + k_3x_7 + k_4x_8 \\ k_1x_5 + k_2x_6 + k_3x_8 + k_4x_9 \end{bmatrix}$$

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

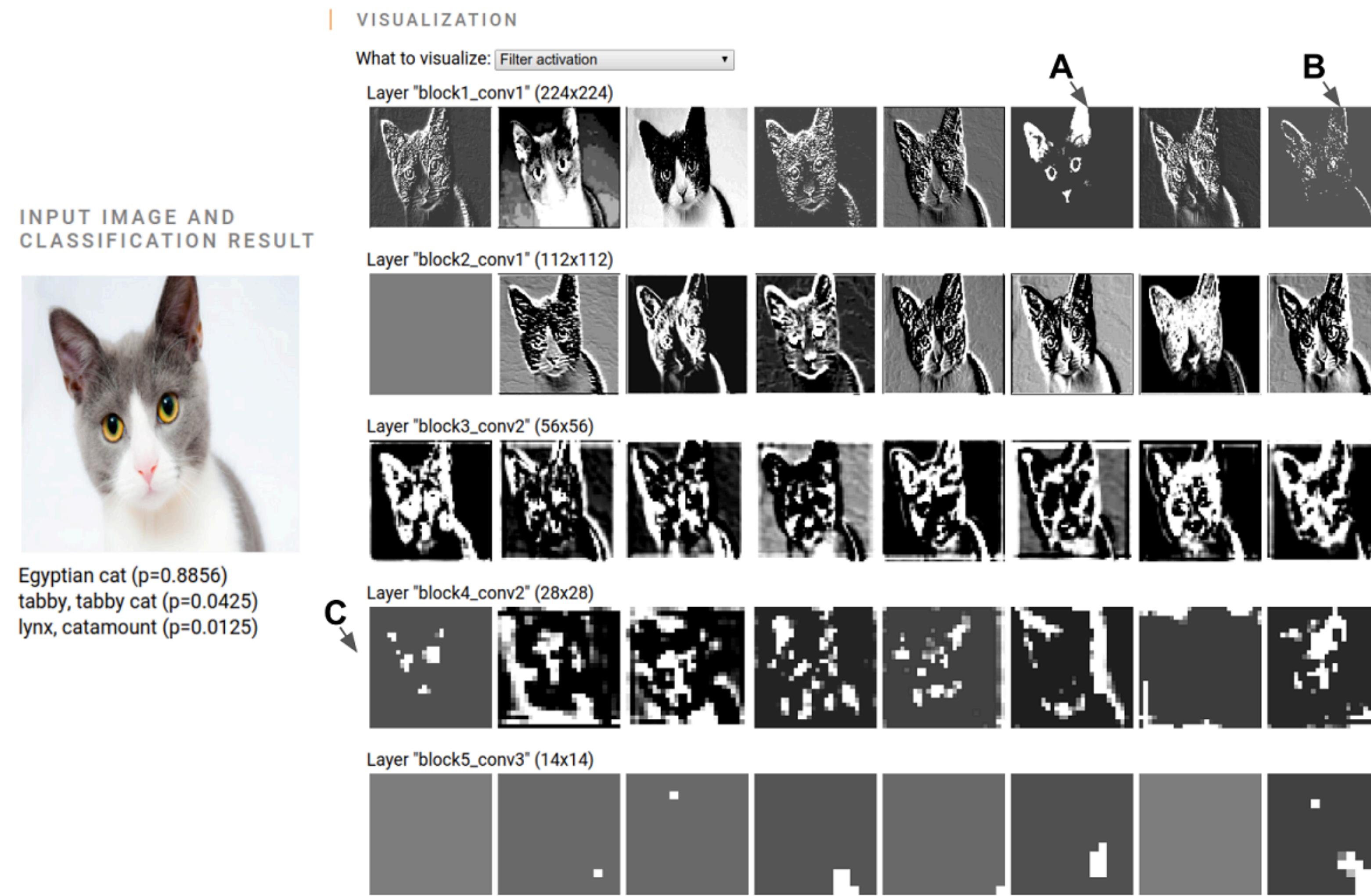
$$\begin{bmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{bmatrix}$$

- circulant matrix

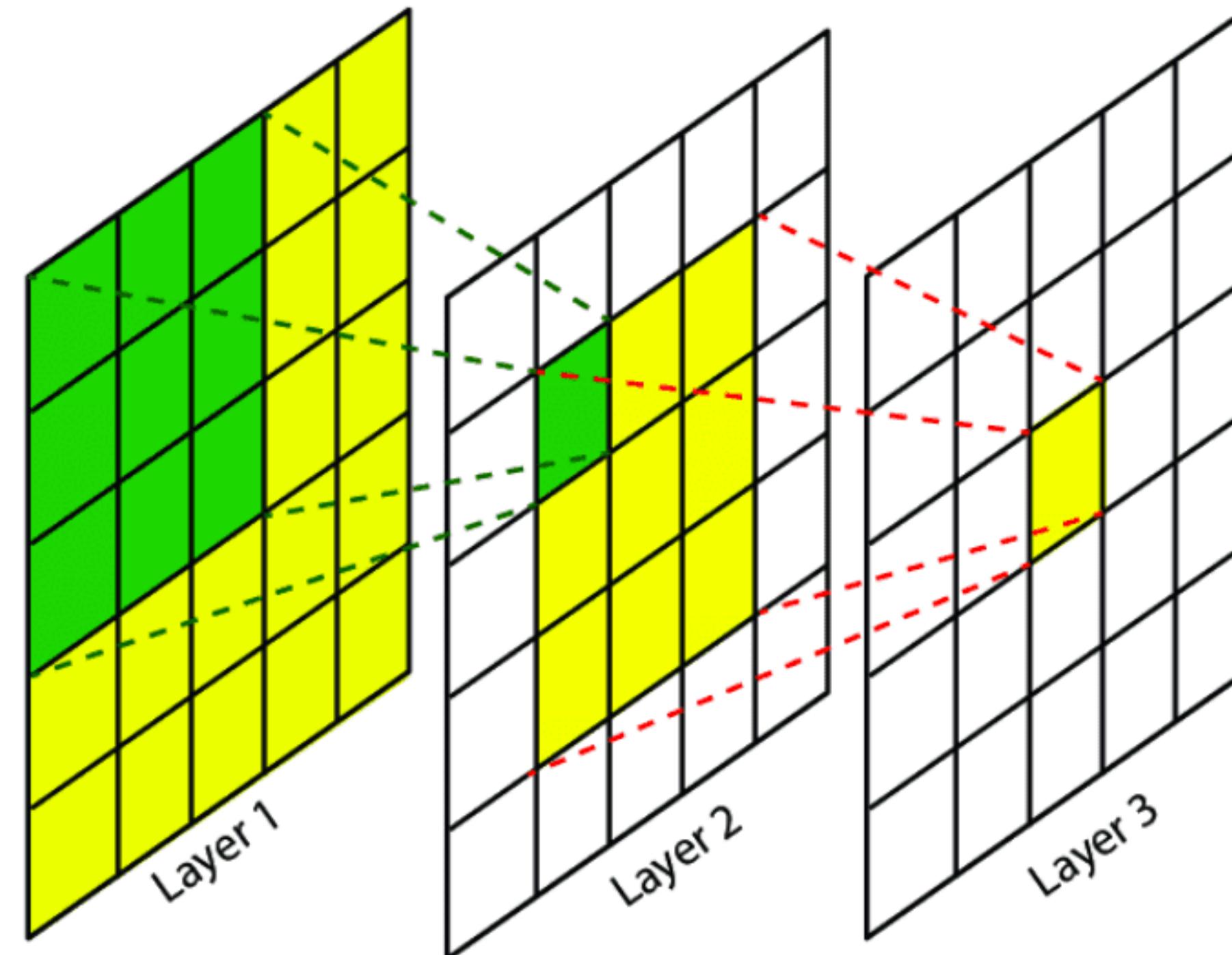
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}$$

$$= \begin{bmatrix} k_1x_1 + k_2x_2 + k_3x_4 + k_4x_5 \\ k_1x_2 + k_2x_3 + k_3x_5 + k_4x_6 \\ k_1x_4 + k_2x_5 + k_3x_7 + k_4x_8 \\ k_1x_5 + k_2x_6 + k_3x_8 + k_4x_9 \end{bmatrix}$$

Свертка: визуализация



Свертка: receptive field

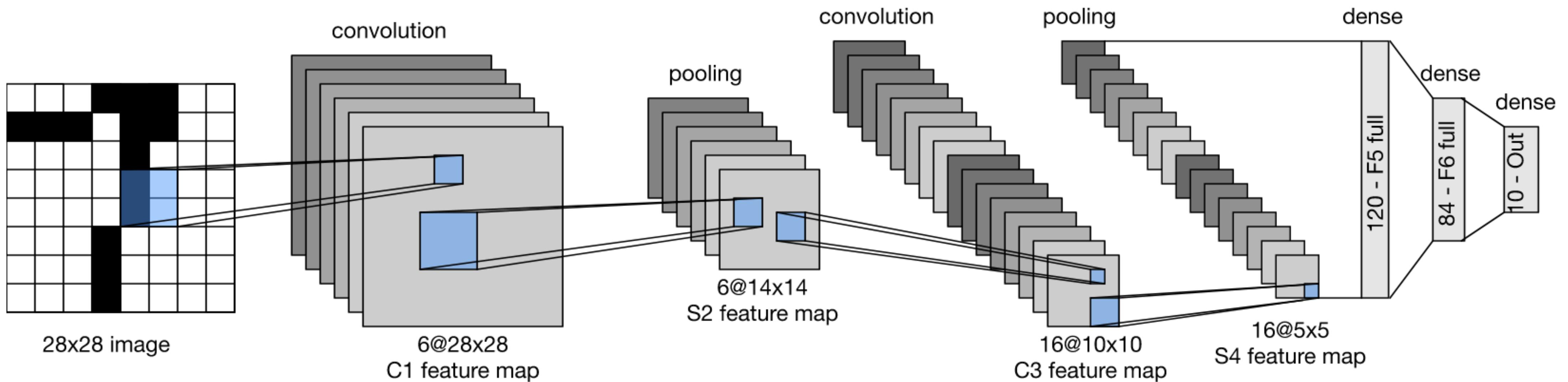


Область исходного изображения, которая влияет на конкретный признак

Архитектуры CNN

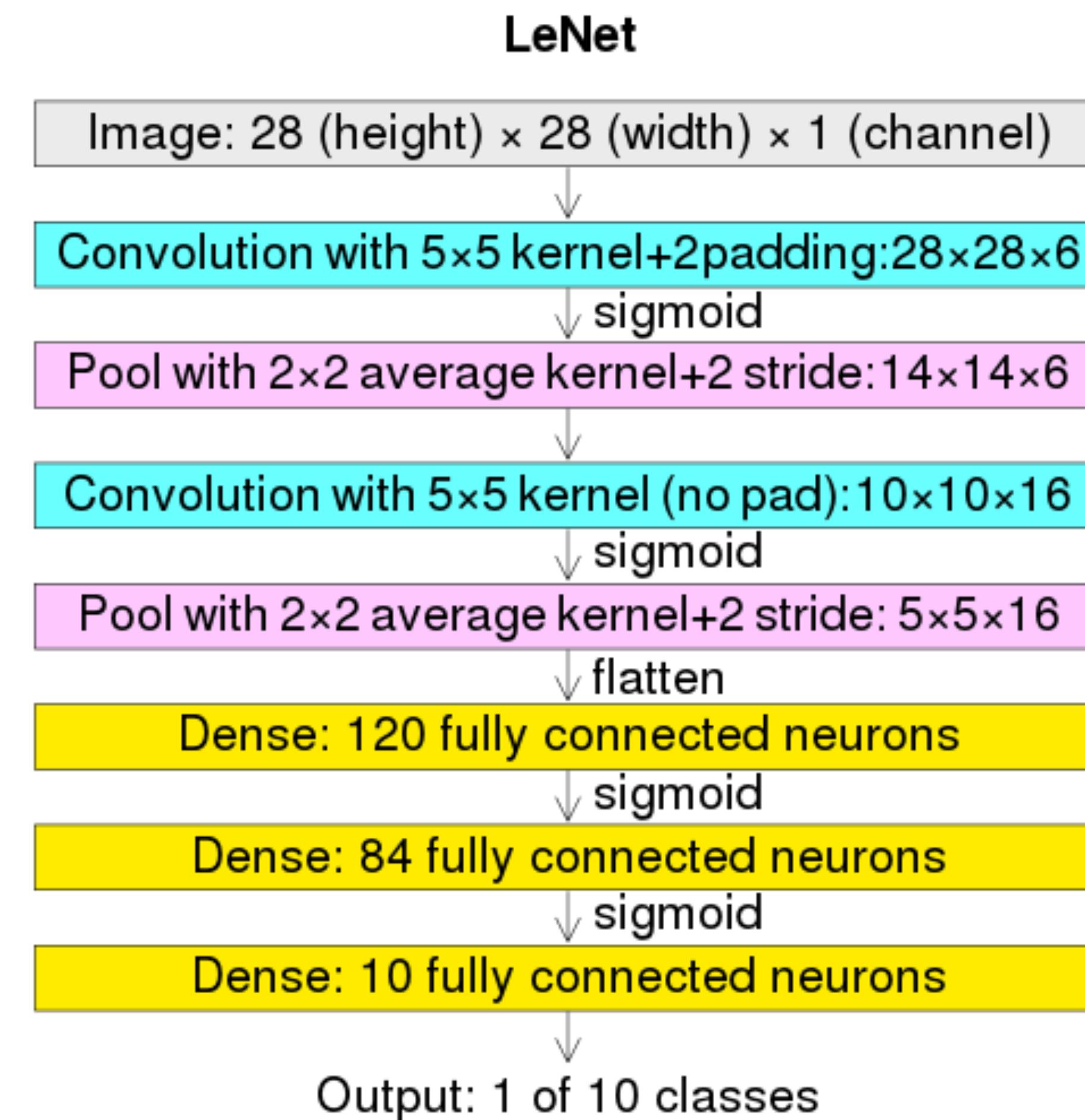
LeNet-5 (LeCun et al., 1998)

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 (flickr), аннотированы Amazon MTurk



sailboat

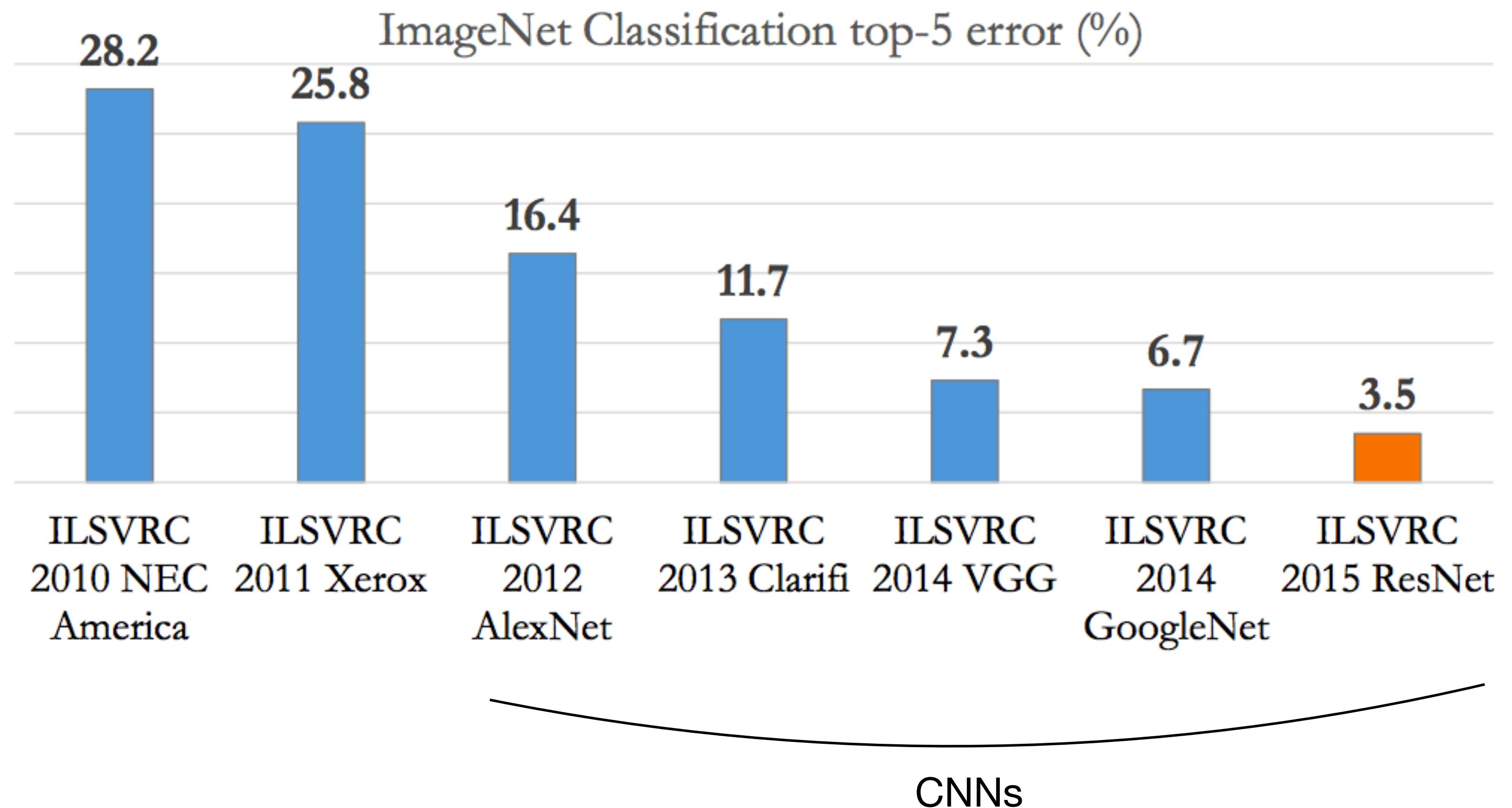


husky

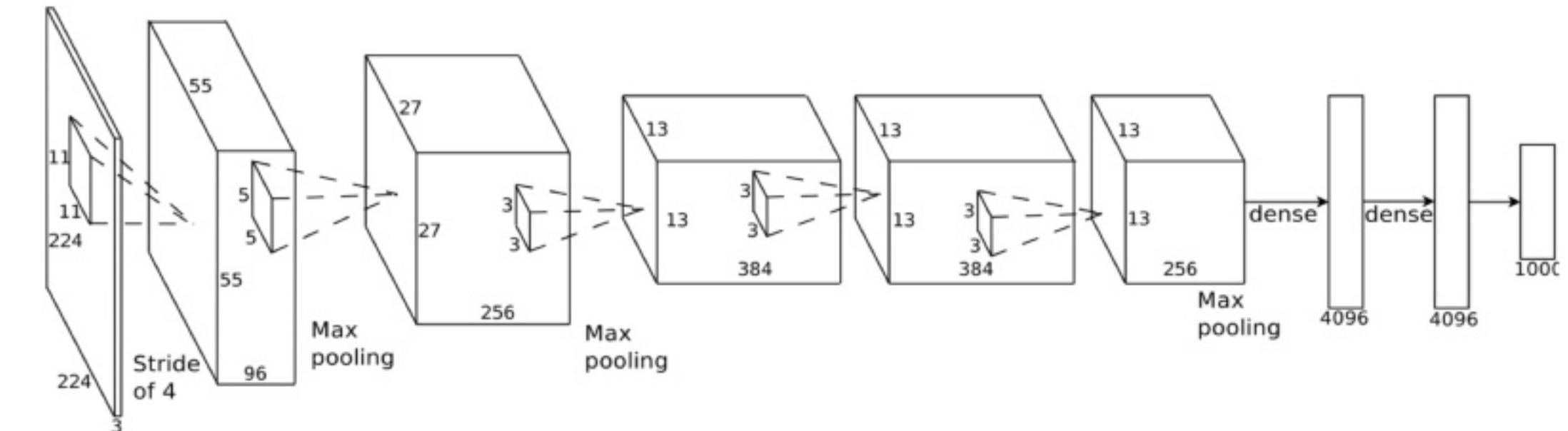
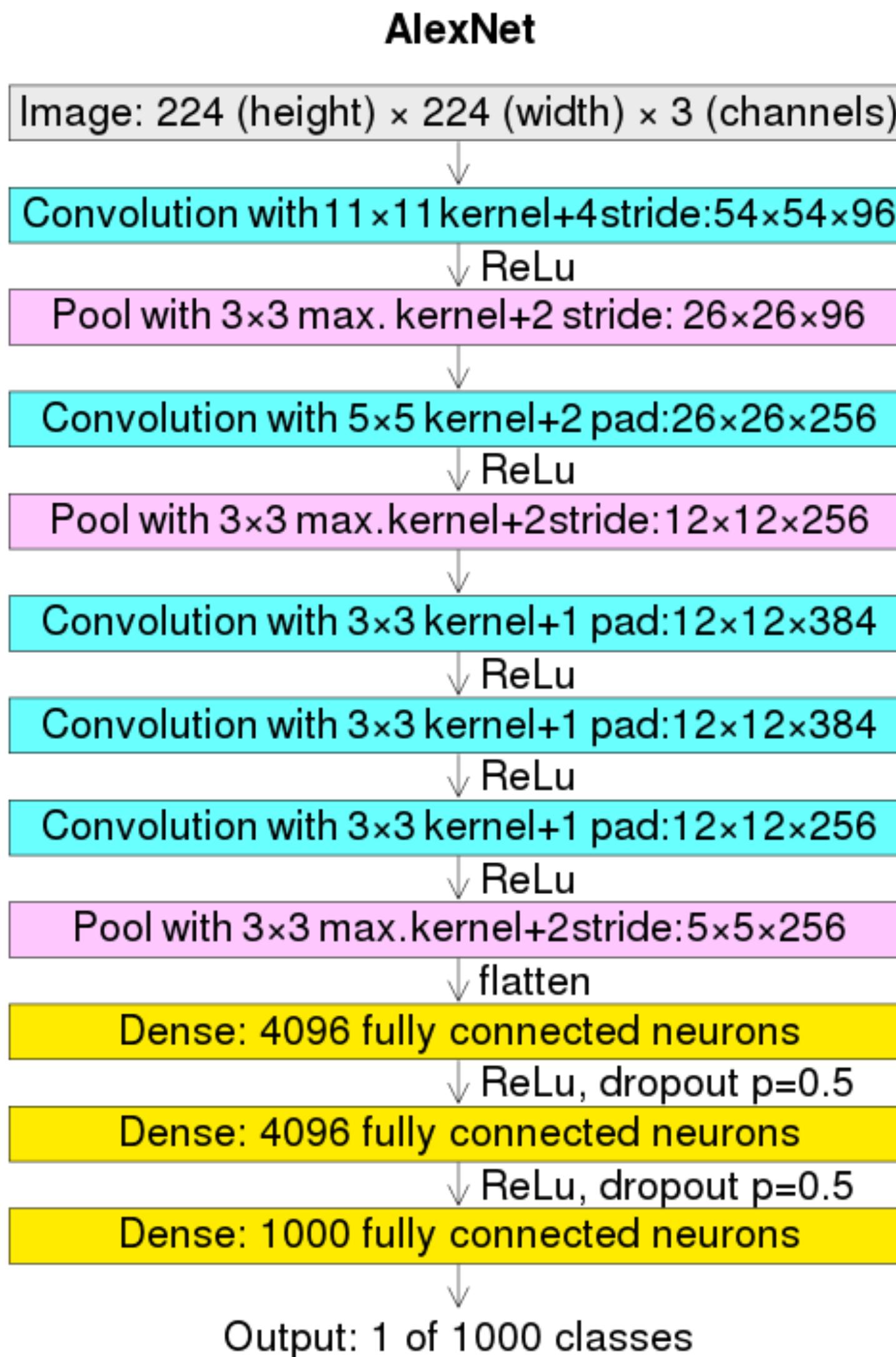
ImageNet

Метрики

- Top-1 accuracy
- Top-5 accuracy



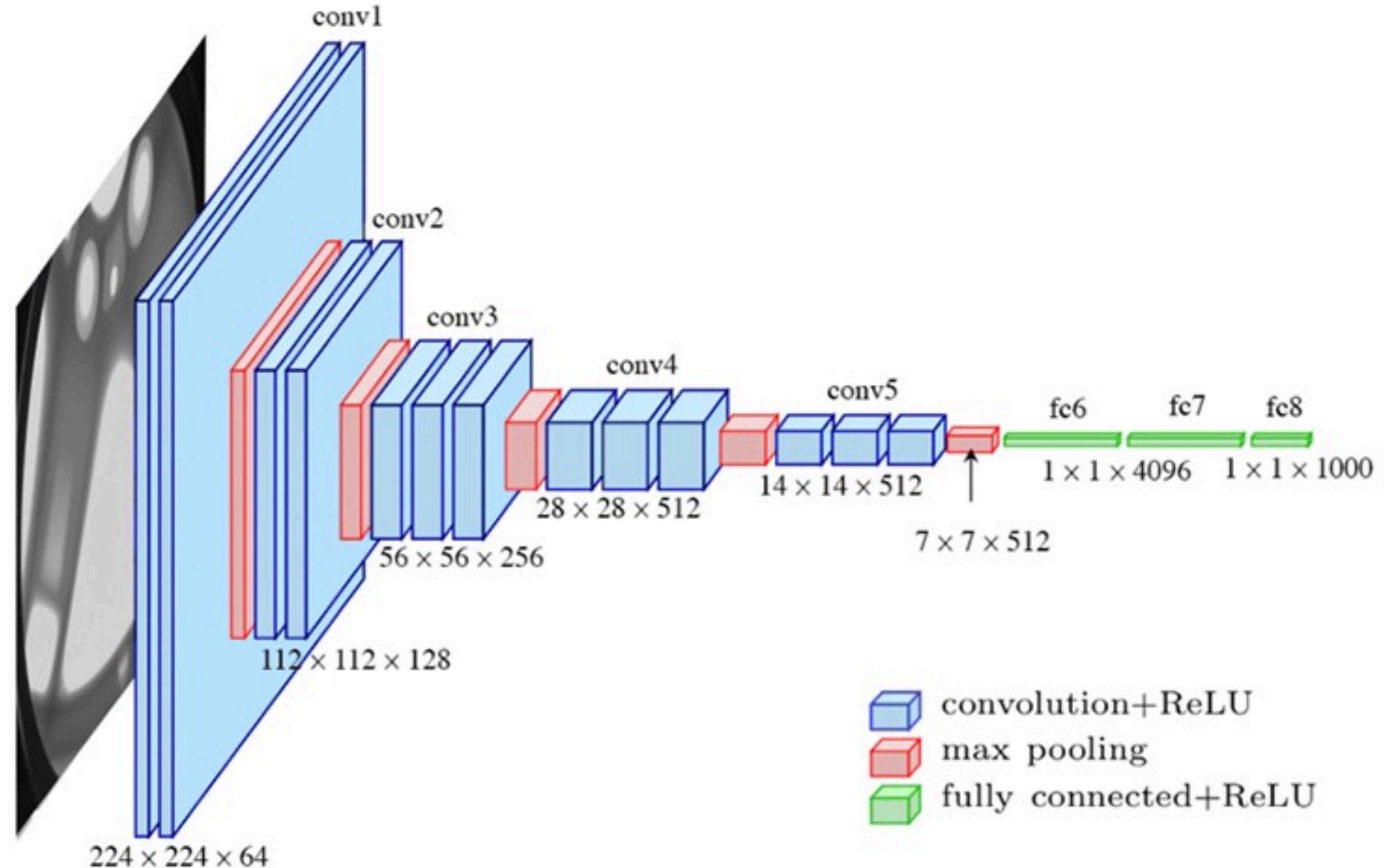
AlexNet (Krizhevsky et al., 2012)



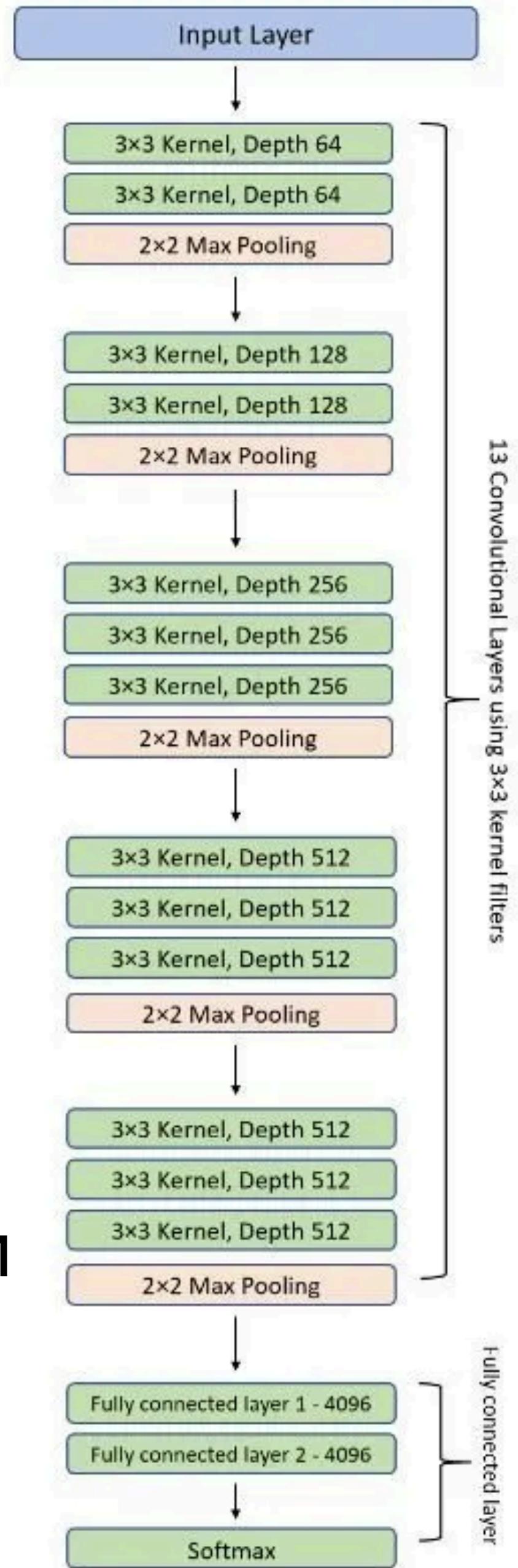
- ReLu, MaxPool
- 60M parameters
- Augmentations + Dropout
- GPU (50x speedup)
- 1 week of training on 2 GPUs

Image credit

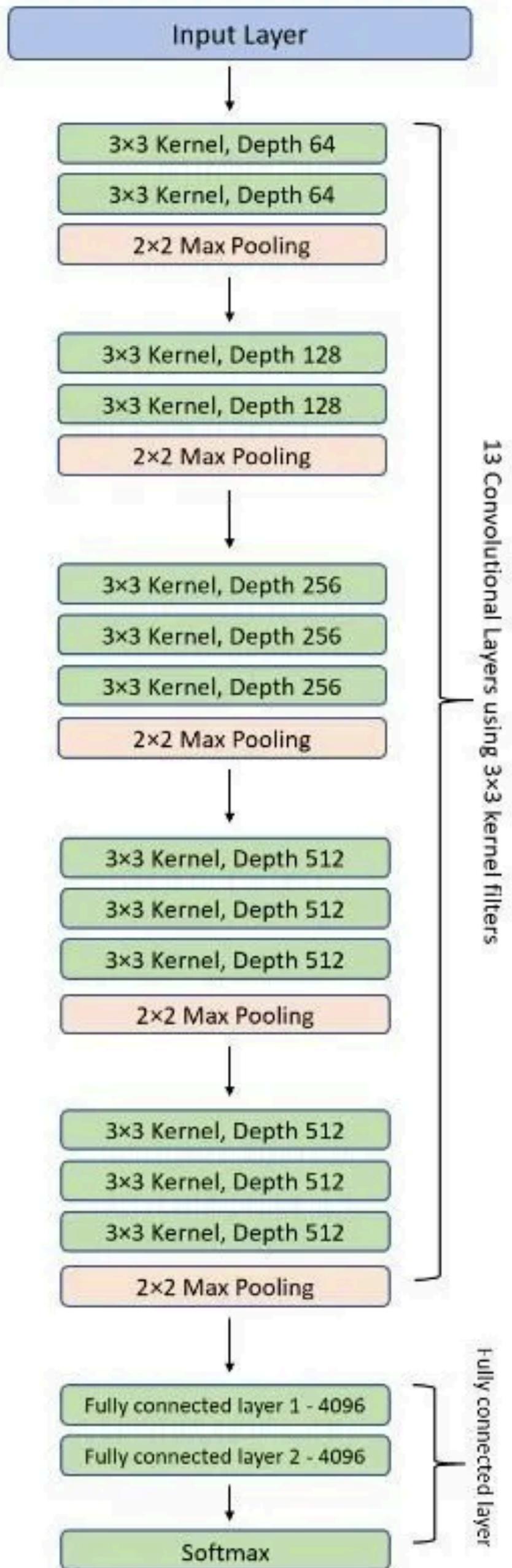
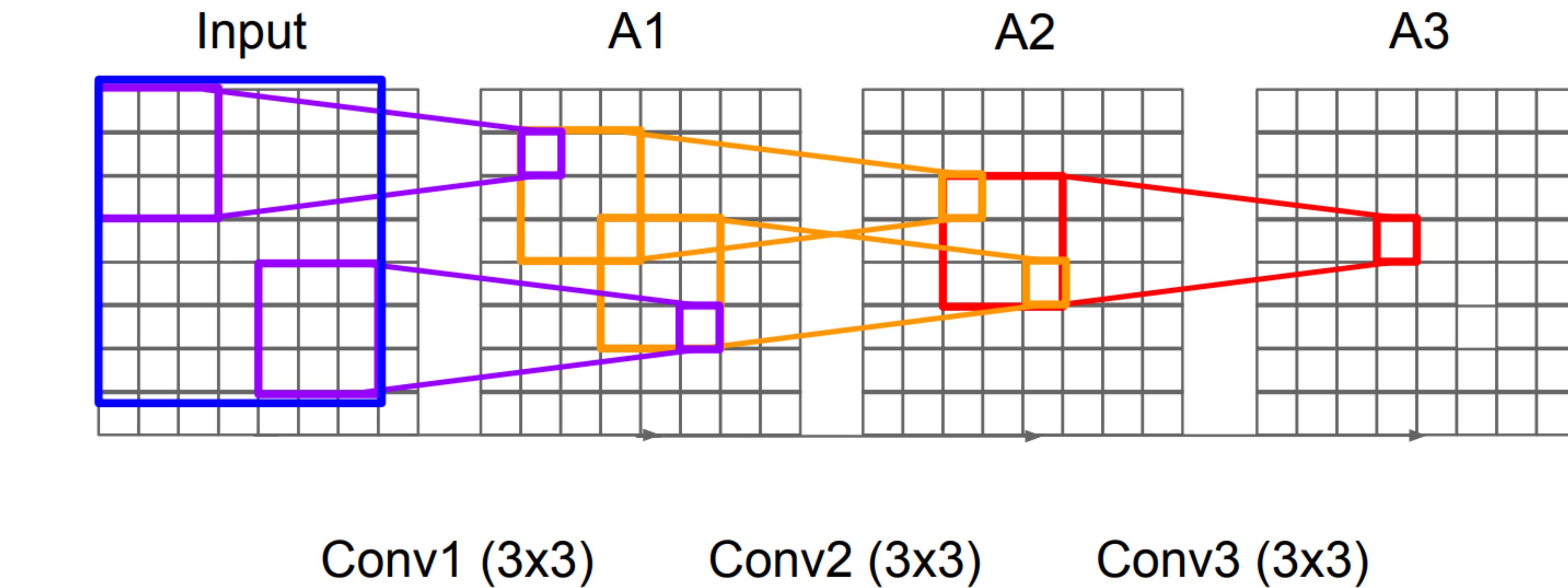
VGGNet (Simonyan and Zisserman, 2014)



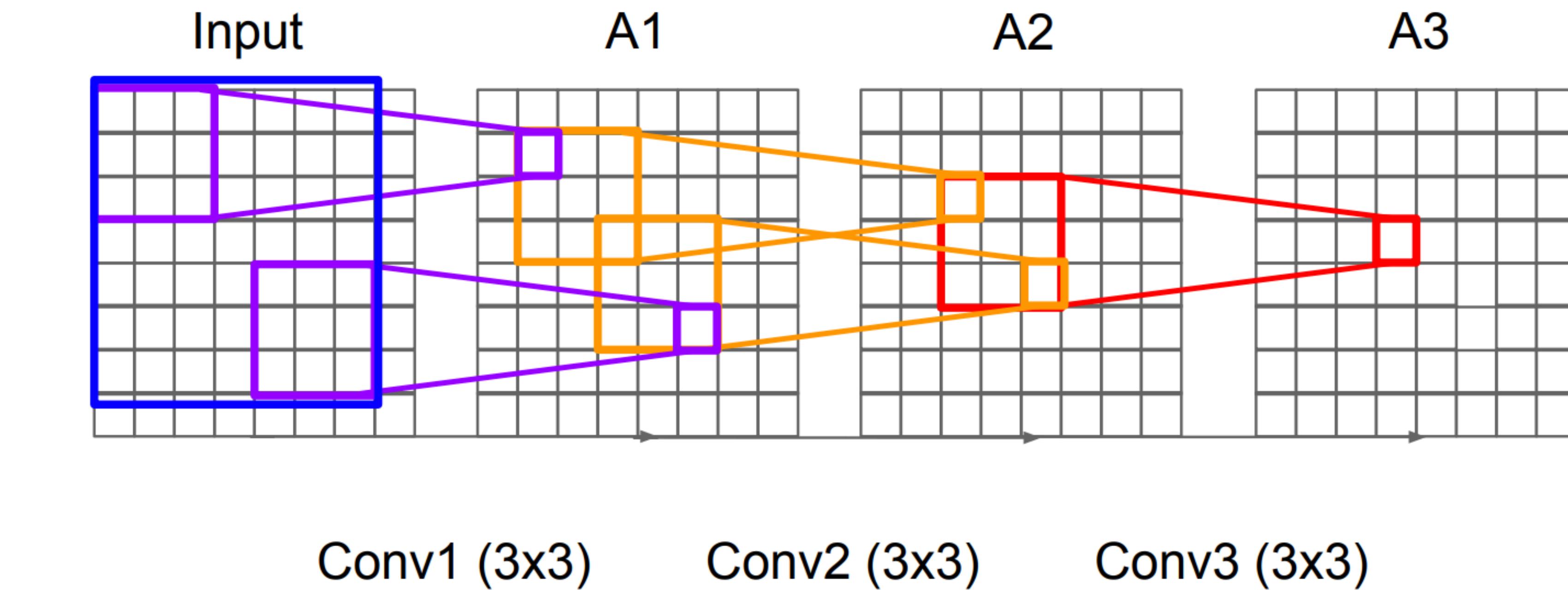
- Свертки 3x3
- Больше слоев
- 140M parameters
- Обучается стадиями



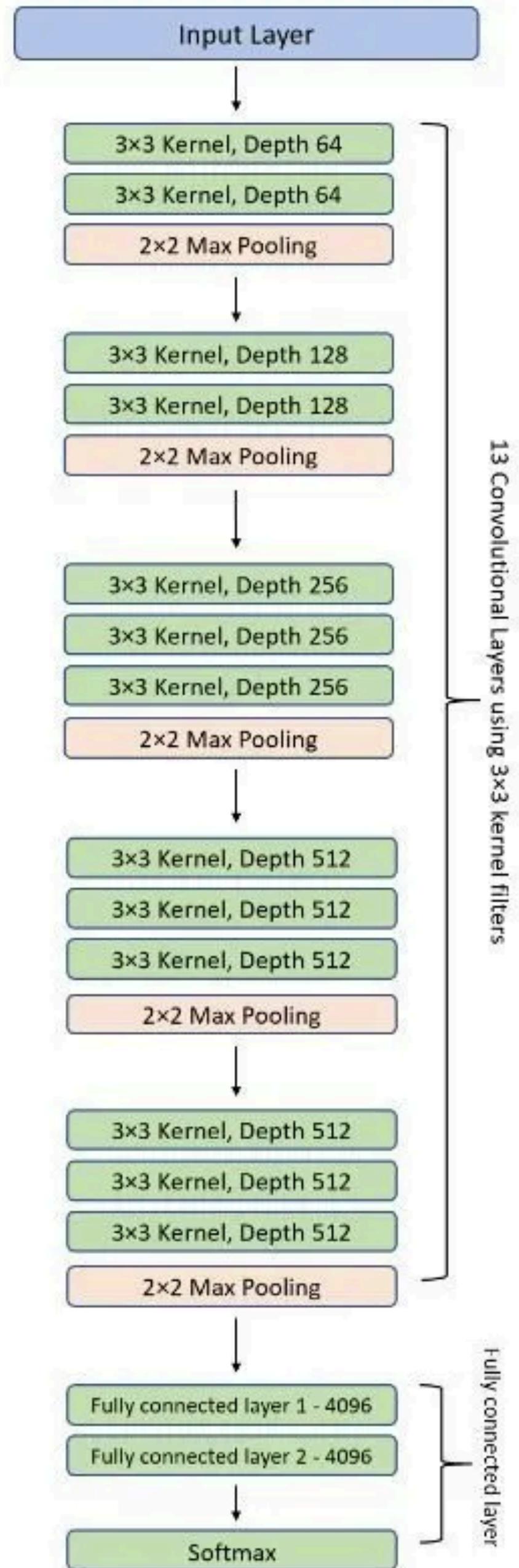
VGGNet (Simonyan and Zisserman, 2014)



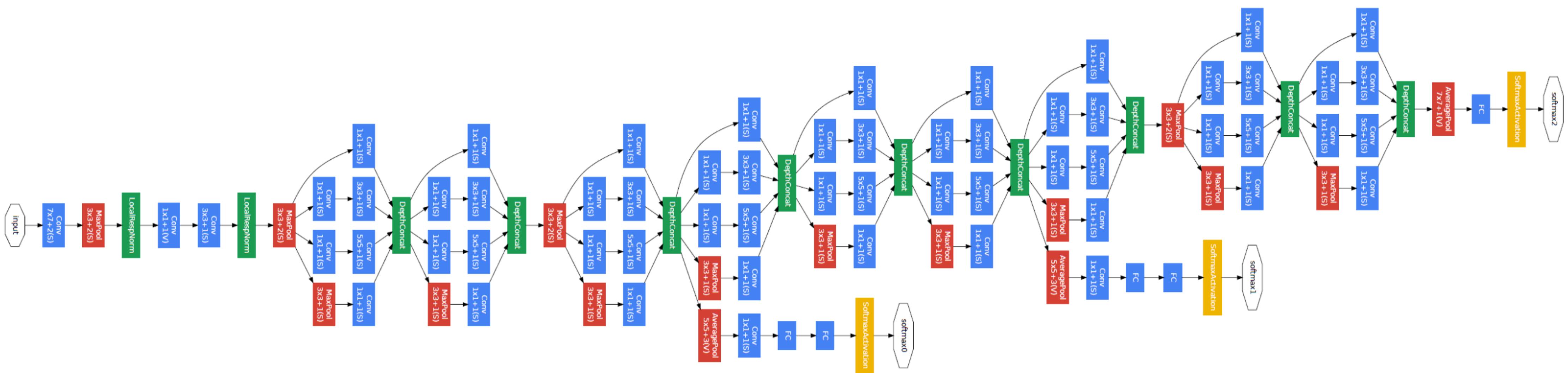
VGGNet (Simonyan and Zisserman, 2014)



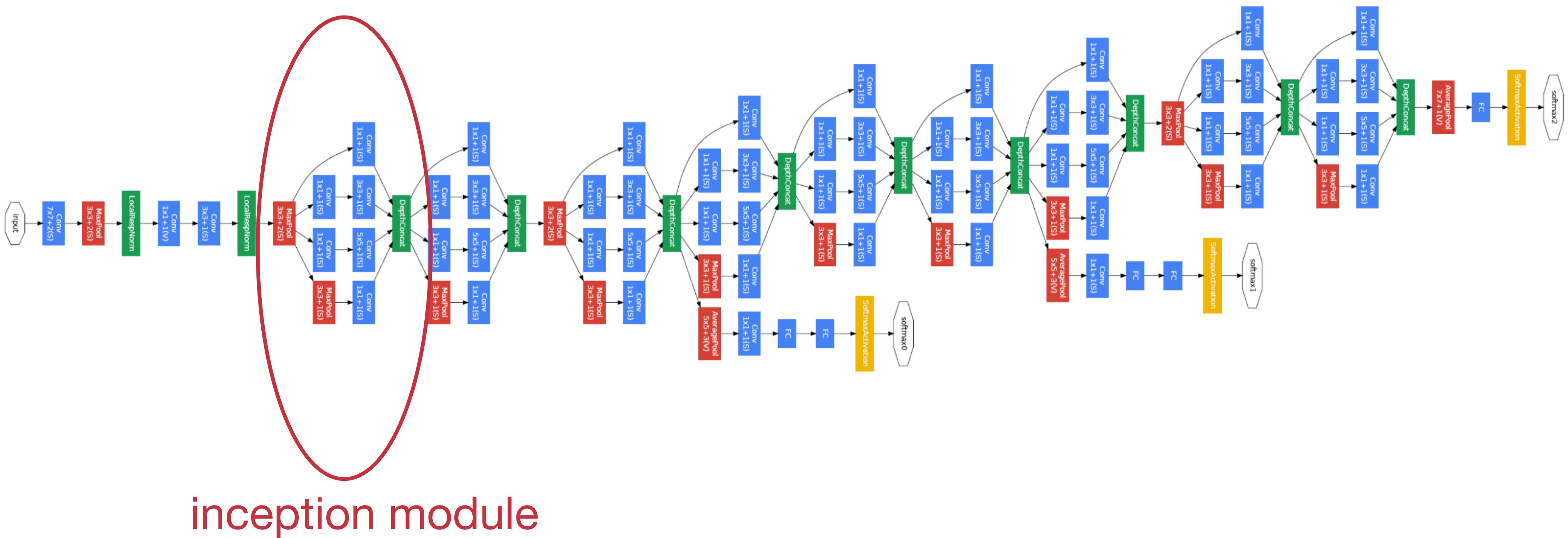
3 свертки 3x3 имеют receptive field как одна 7x7
Меньше параметров, можно сделать глубже сеть



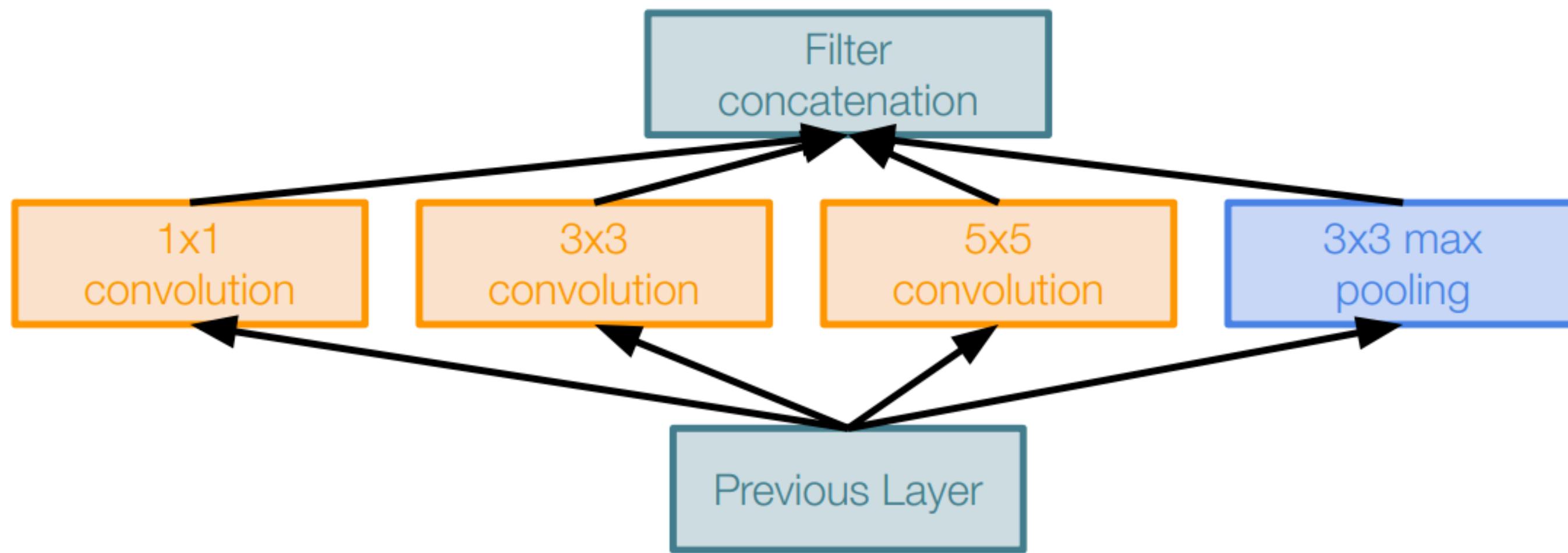
GoogLeNet (Szegedy et al., 2014)



GoogLeNet (Szegedy et al., 2014)

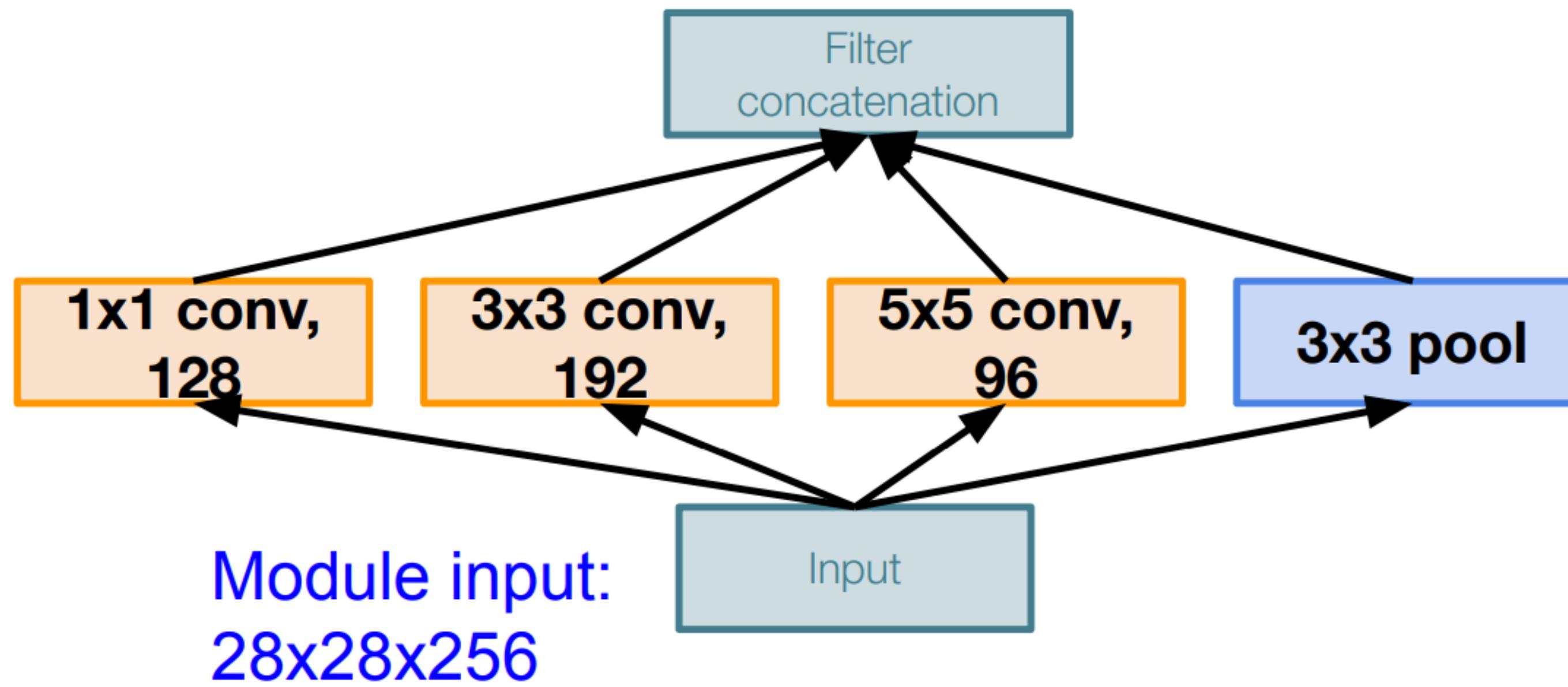


GoogLeNet (Szegedy et al., 2014)



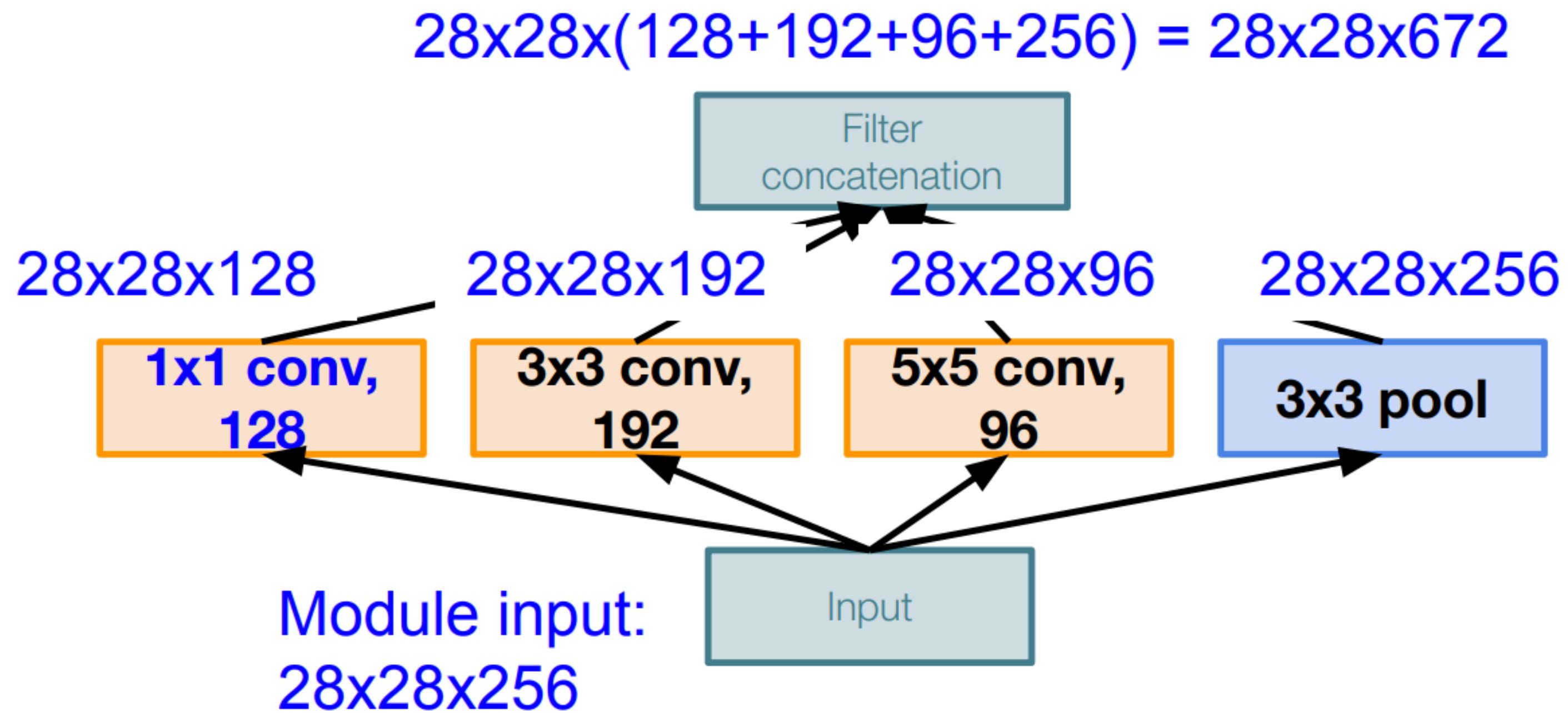
Naive Inception module

GoogLeNet (Szegedy et al., 2014)



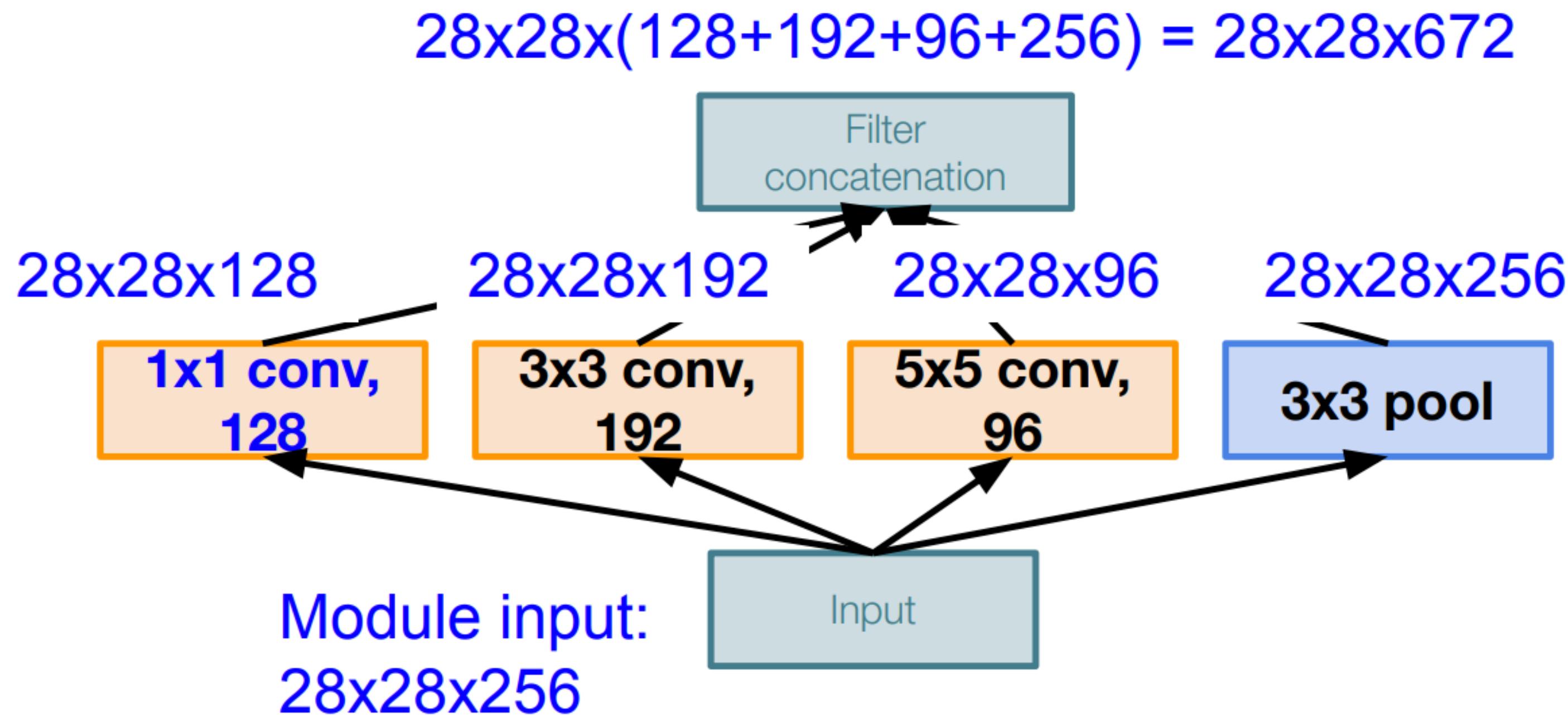
Вычислительная сложность?

GoogLeNet (Szegedy et al., 2014)



Вычислительная сложность?

GoogLeNet (Szegedy et al., 2014)



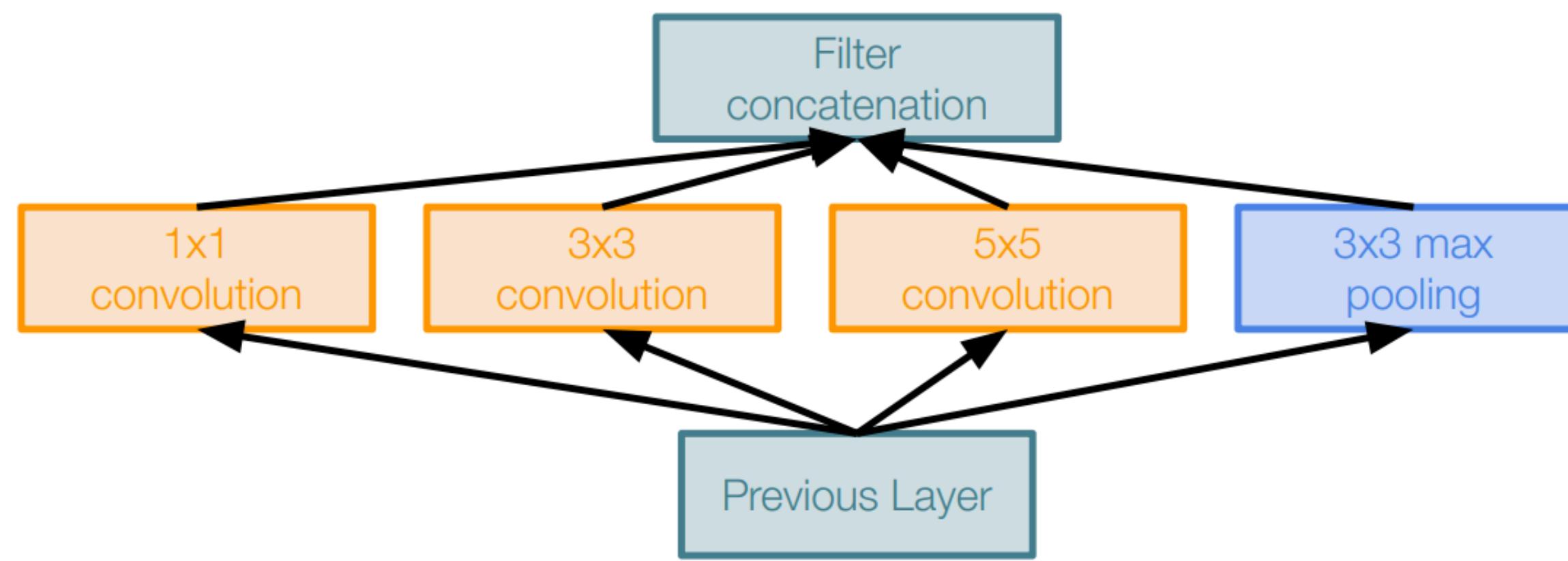
Conv Ops:

[1x1 conv, 128] 28x28x128x1x1x256
[3x3 conv, 192] 28x28x192x3x3x256
[5x5 conv, 96] 28x28x96x5x5x256

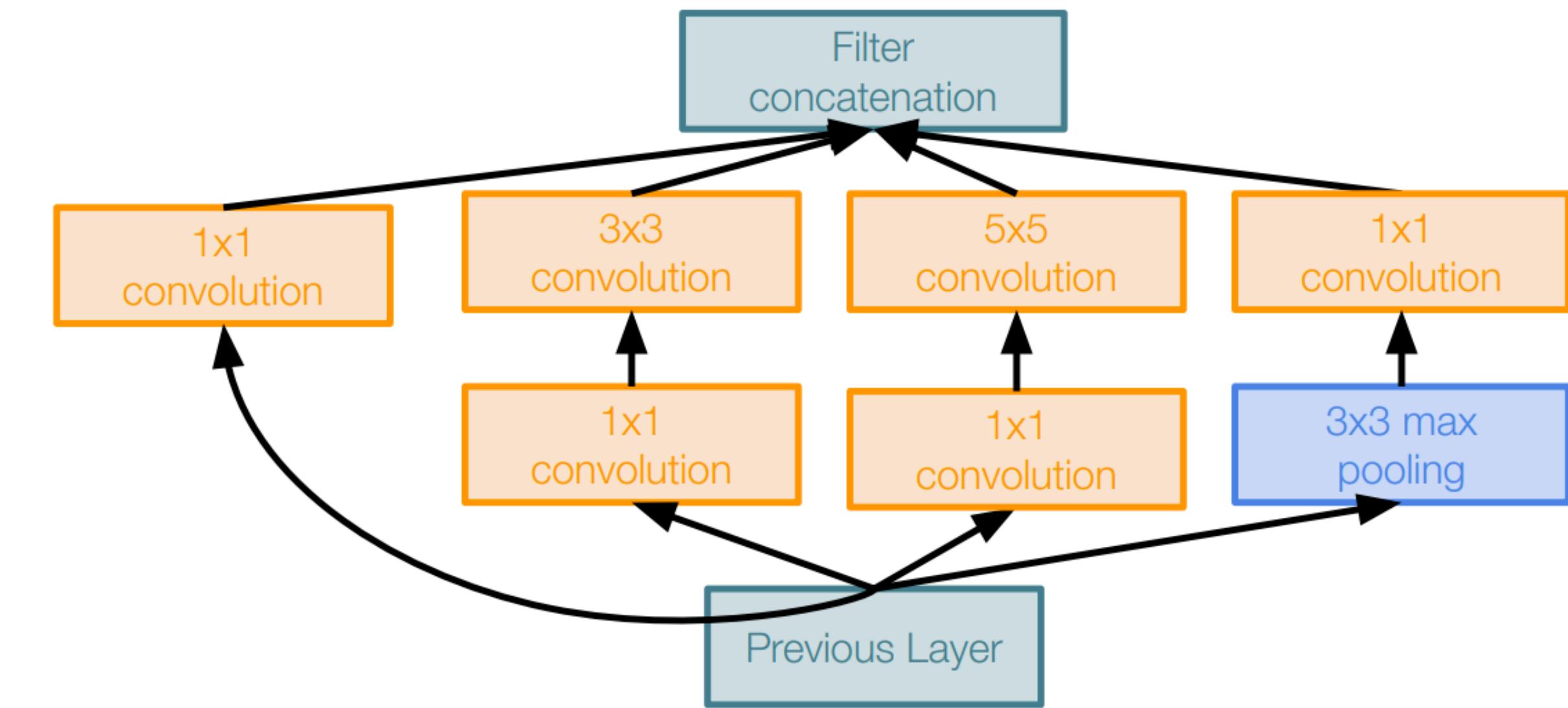
Total: 854M ops

Вычислительная сложность?

GoogLeNet (Szegedy et al., 2014)

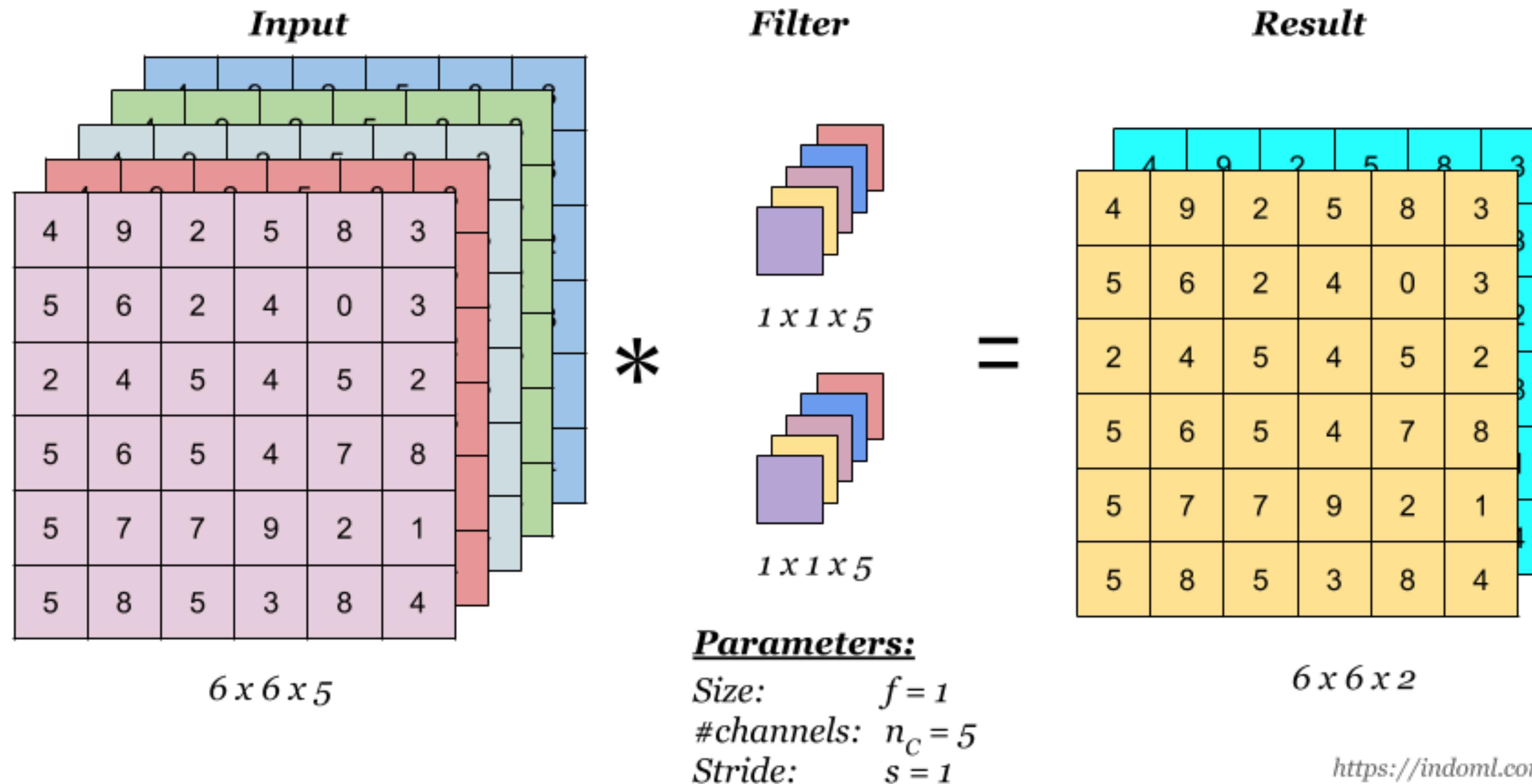


Naive Inception module

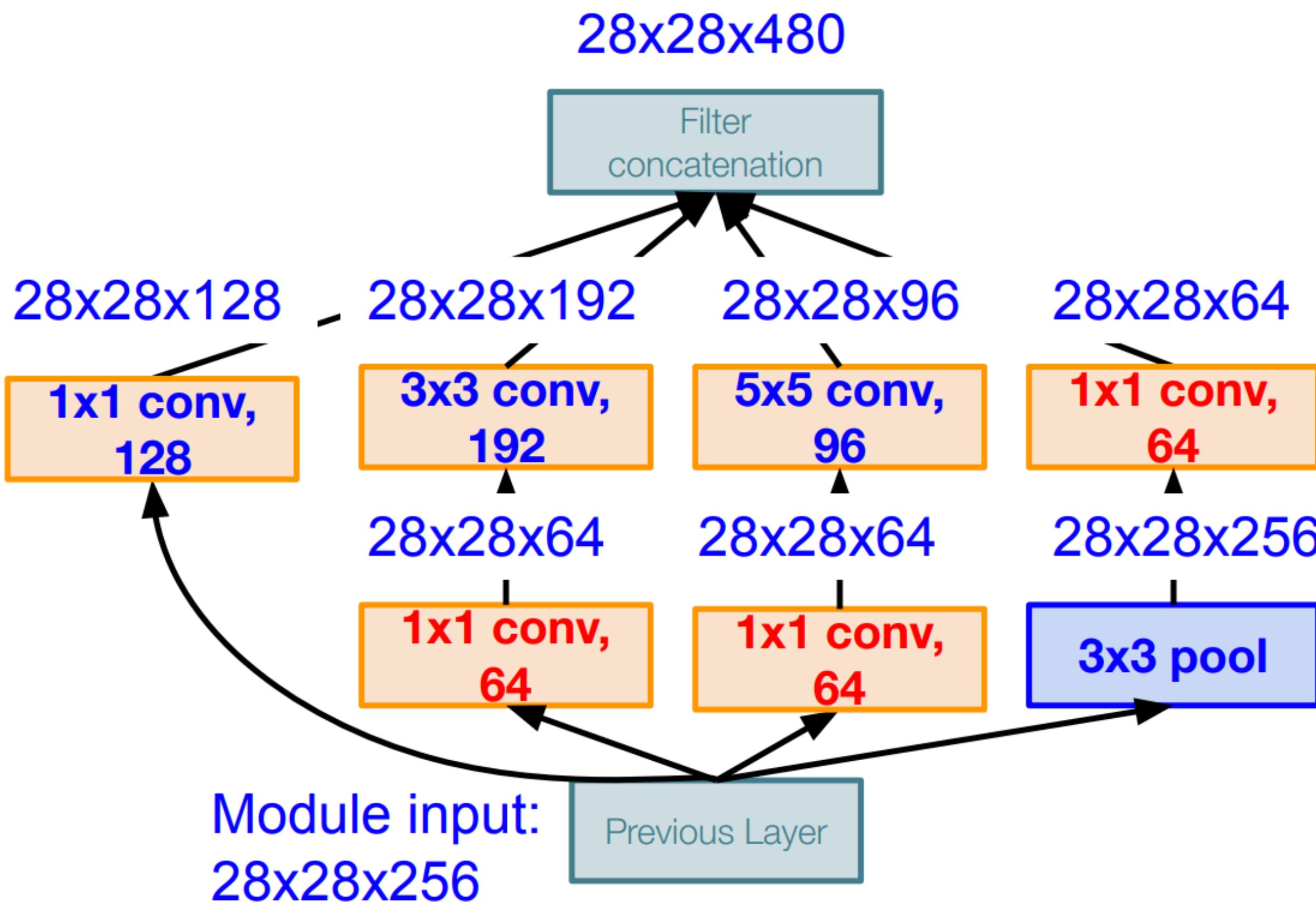


Inception module with dimension reduction

GoogLeNet (Szegedy et al., 2014)



GoogLeNet (Szegedy et al., 2014)

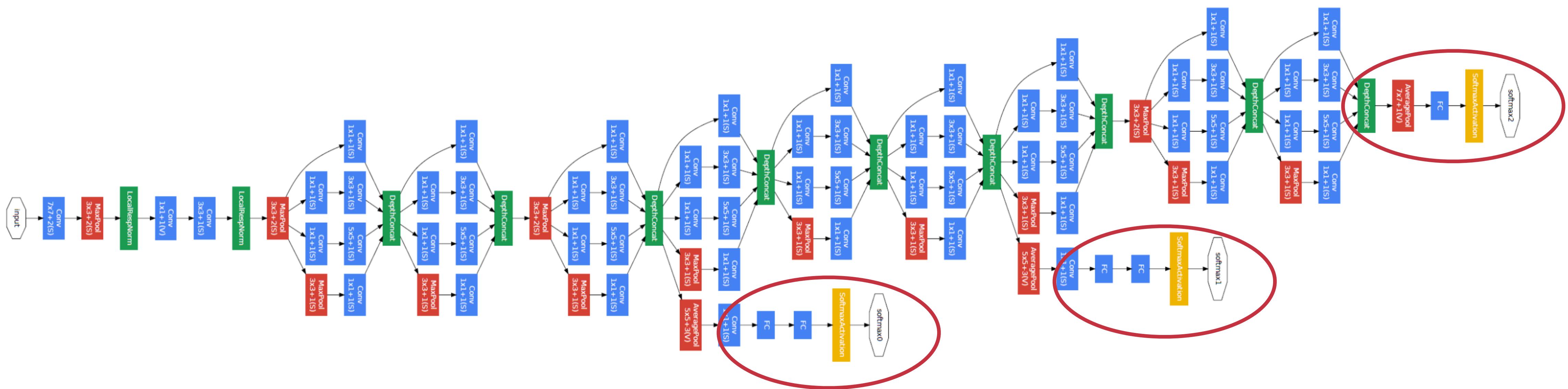


Conv Ops:

- [**1x1 conv, 64**] $28 \times 28 \times 64 \times 1 \times 1 \times 256$
- [**1x1 conv, 64**] $28 \times 28 \times 64 \times 1 \times 1 \times 256$
- [**1x1 conv, 128**] $28 \times 28 \times 128 \times 1 \times 1 \times 256$
- [**3x3 conv, 192**] $28 \times 28 \times 192 \times 3 \times 3 \times 64$
- [**5x5 conv, 96**] $28 \times 28 \times 96 \times 5 \times 5 \times 64$
- [**1x1 conv, 64**] $28 \times 28 \times 64 \times 1 \times 1 \times 256$

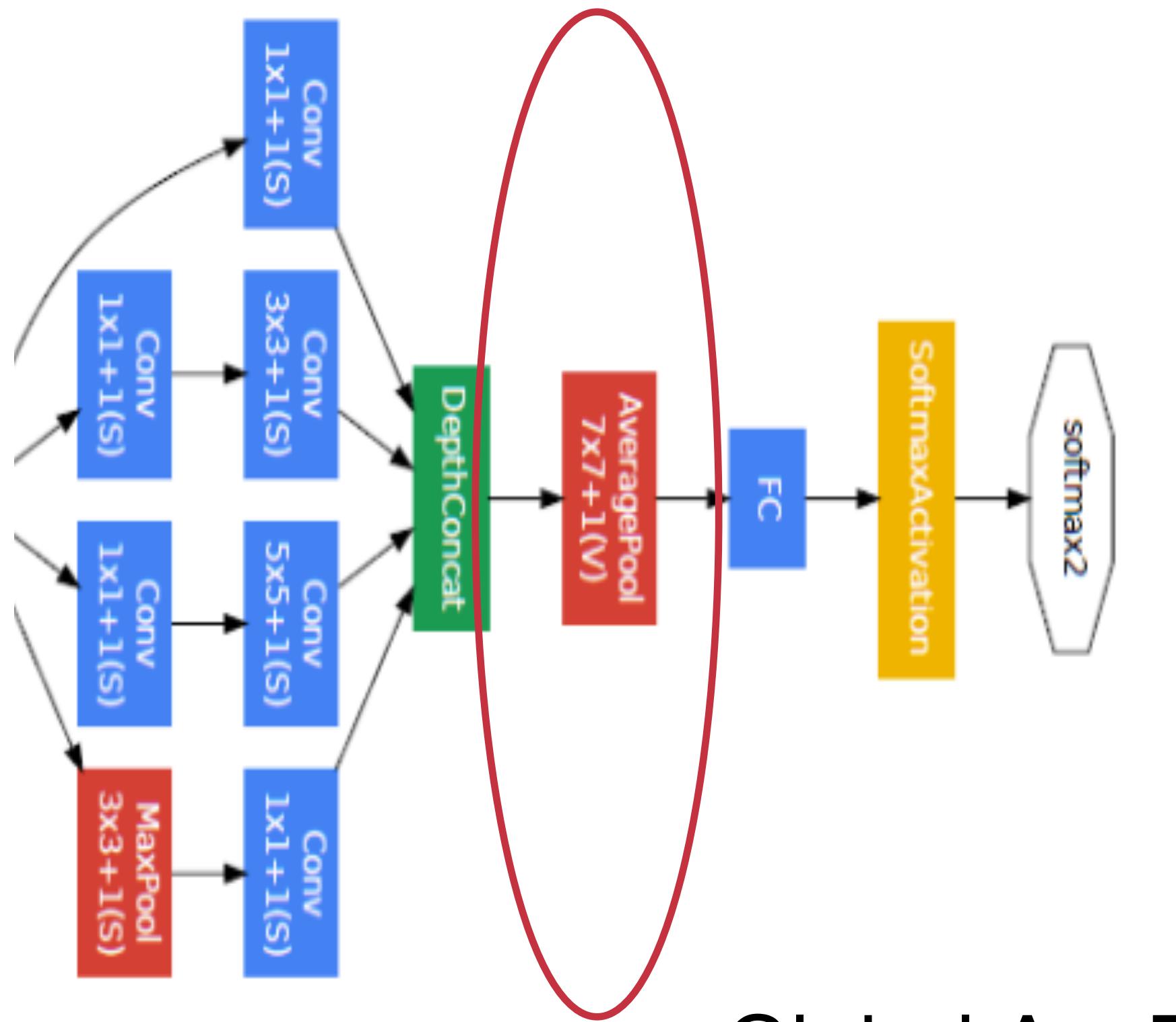
Total: 358M ops

GoogLeNet (Szegedy et al., 2014)

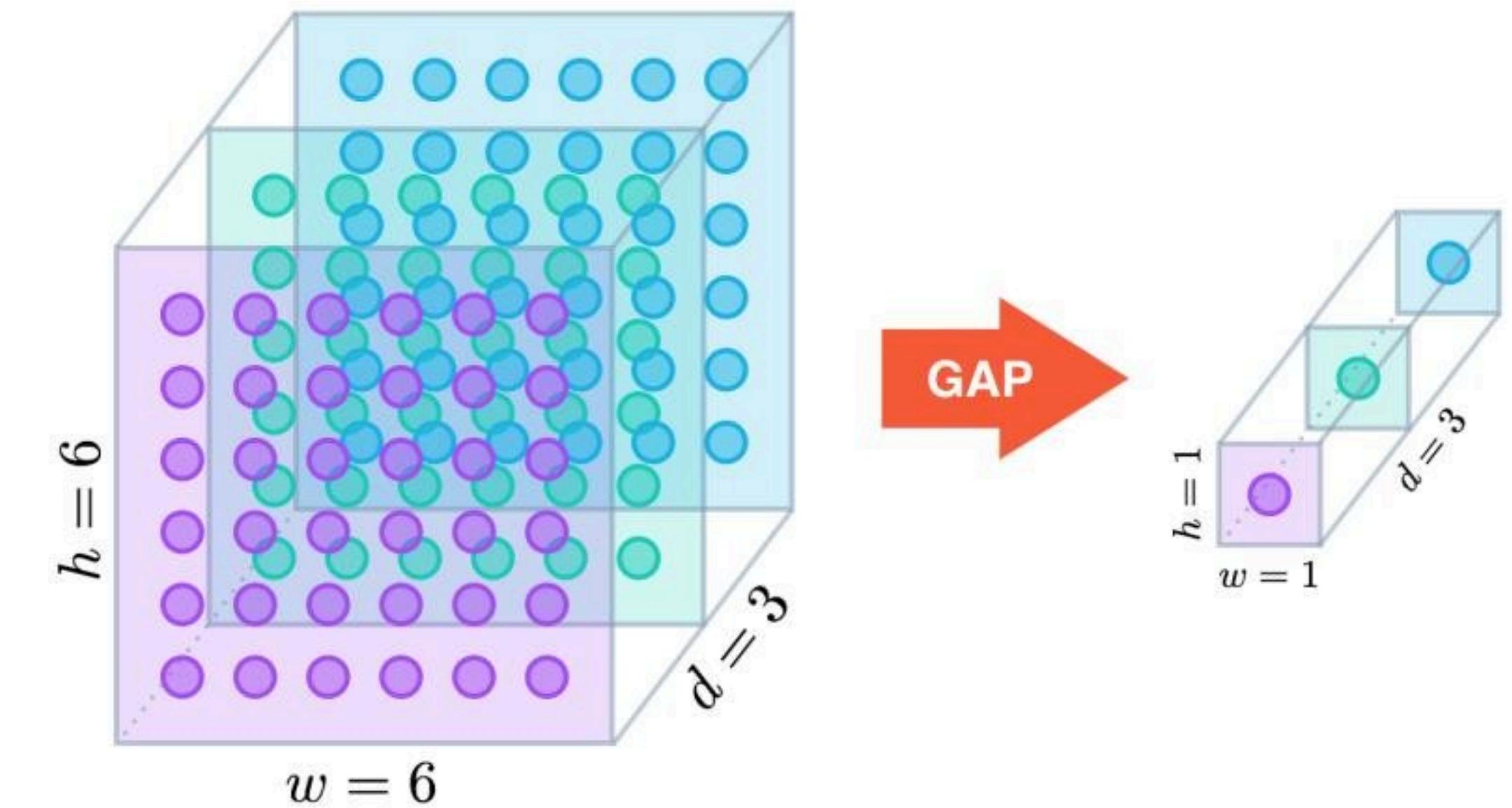


вспомогательные выходы для обучения

GoogLeNet (Szegedy et al., 2014)



Global AvgPool

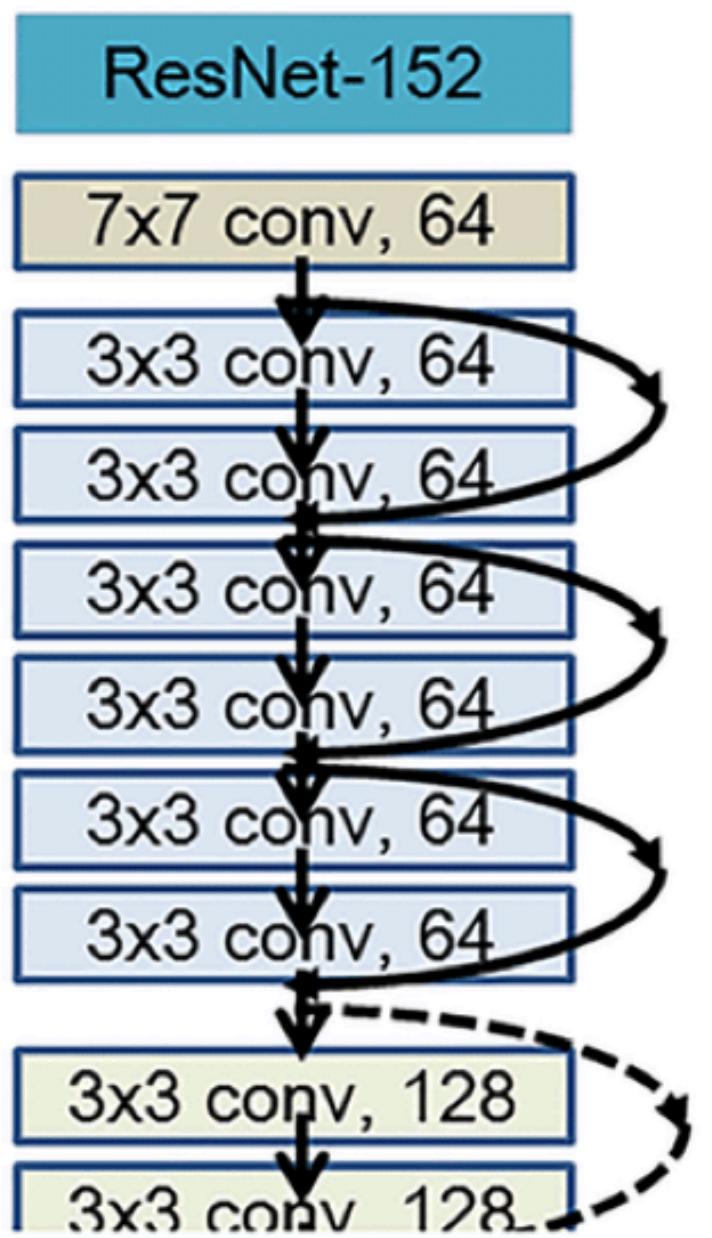


GoogLeNet (Szegedy et al., 2014)

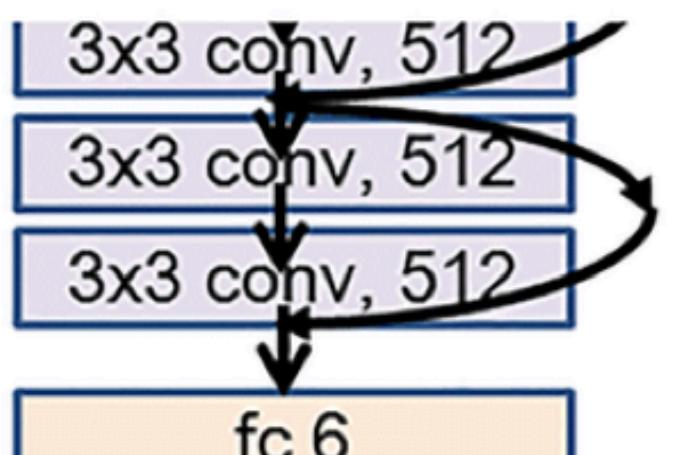
type	patch size/ stride	output size	depth
convolution	$7 \times 7 / 2$	$112 \times 112 \times 64$	1
max pool	$3 \times 3 / 2$	$56 \times 56 \times 64$	0
convolution	$3 \times 3 / 1$	$56 \times 56 \times 192$	2
max pool	$3 \times 3 / 2$	$28 \times 28 \times 192$	0
inception (3a)		$28 \times 28 \times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

- Нелинейная архитектура
- 22 слоя
- ~5M parameters

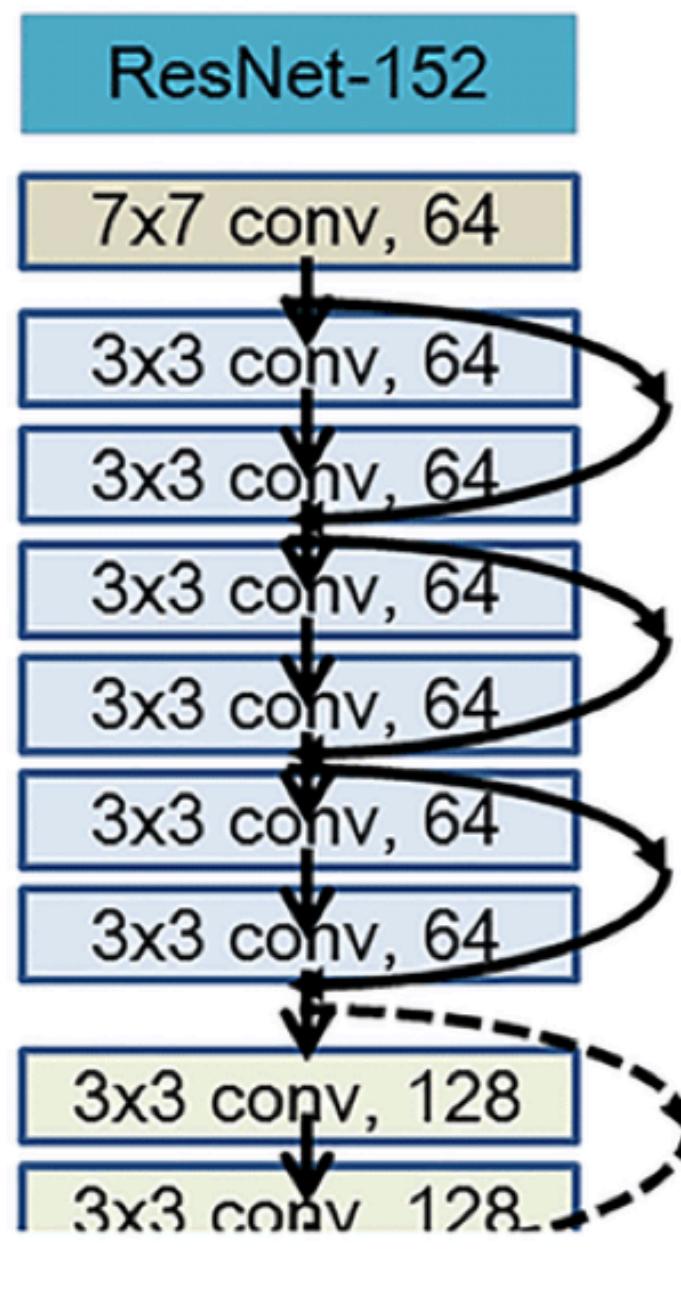
ResNet (He et al., 2015)



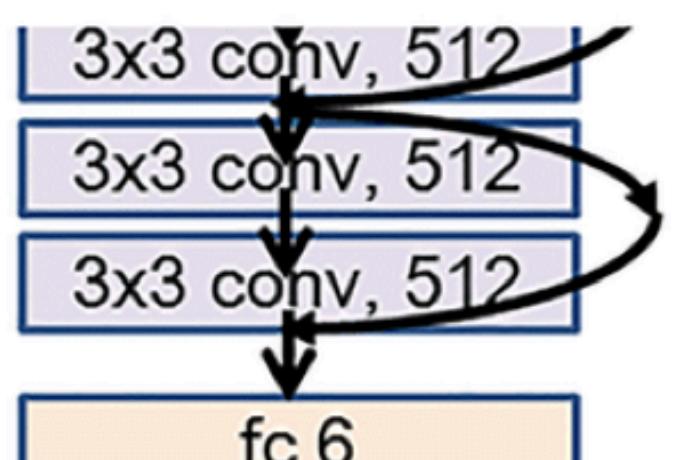
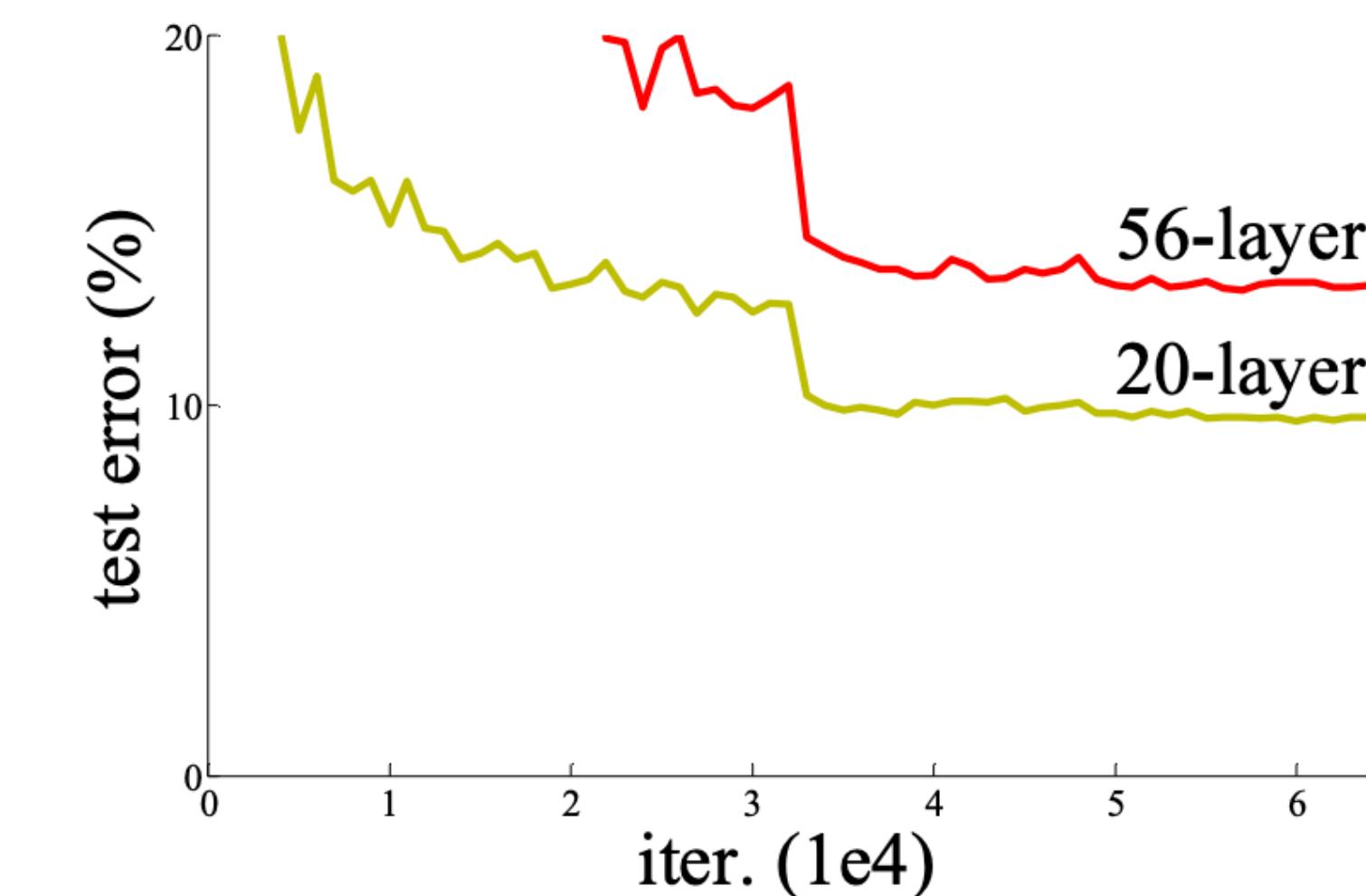
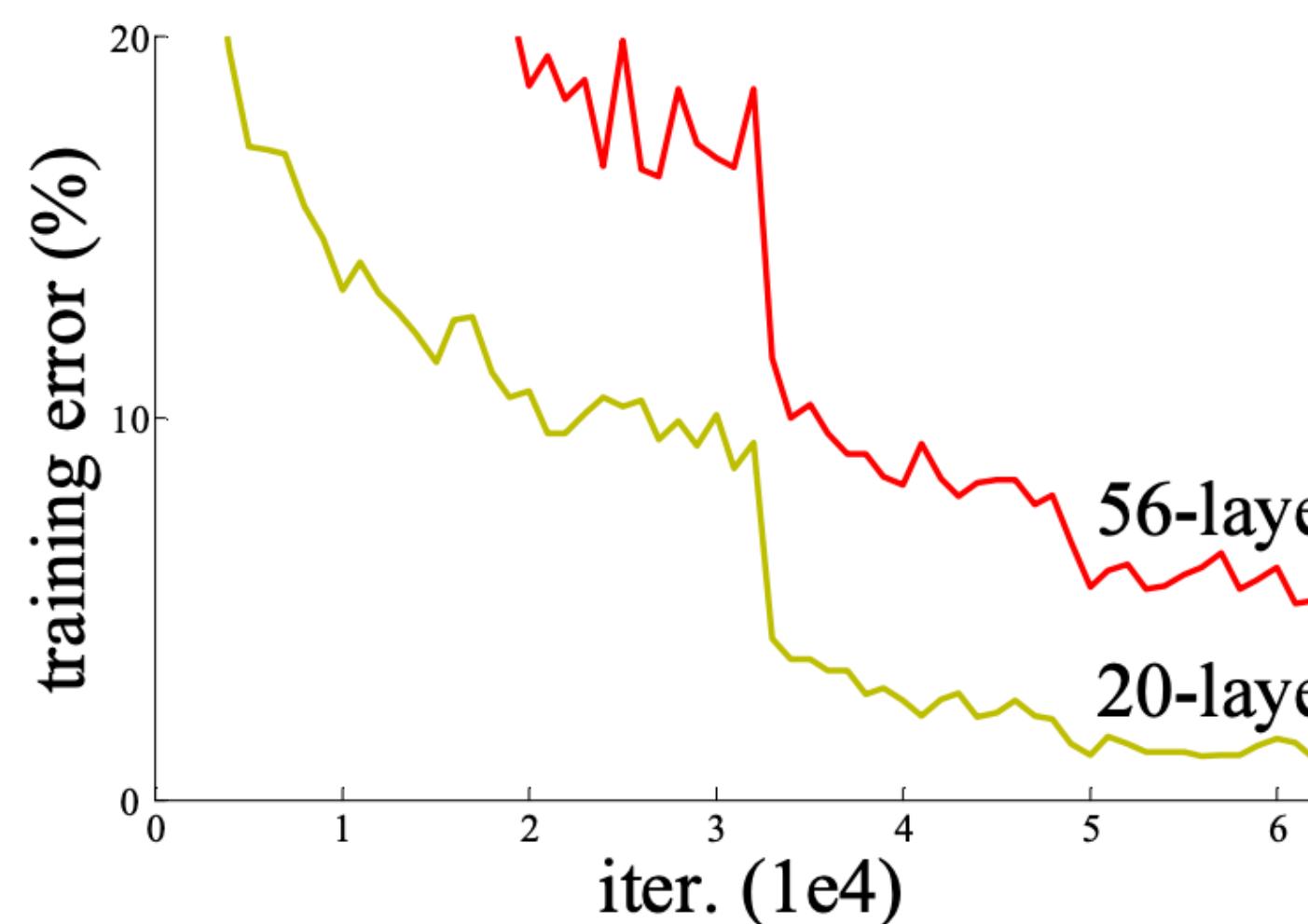
152 слоя!



ResNet (He et al., 2015)

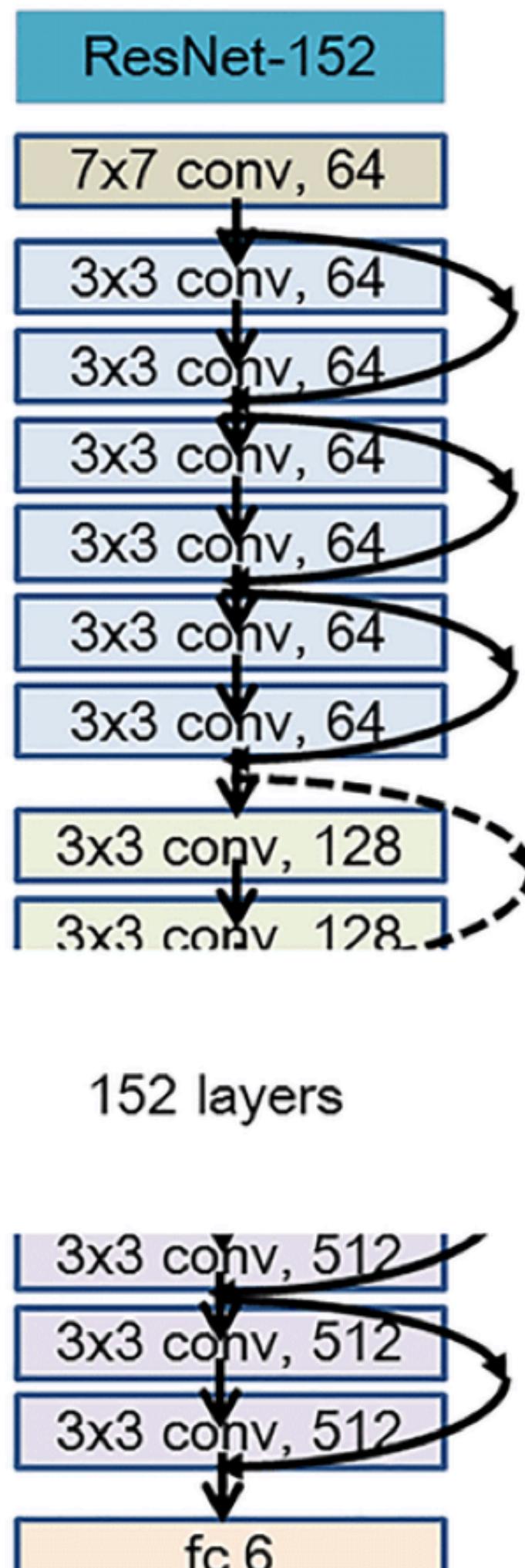


152 слоя!

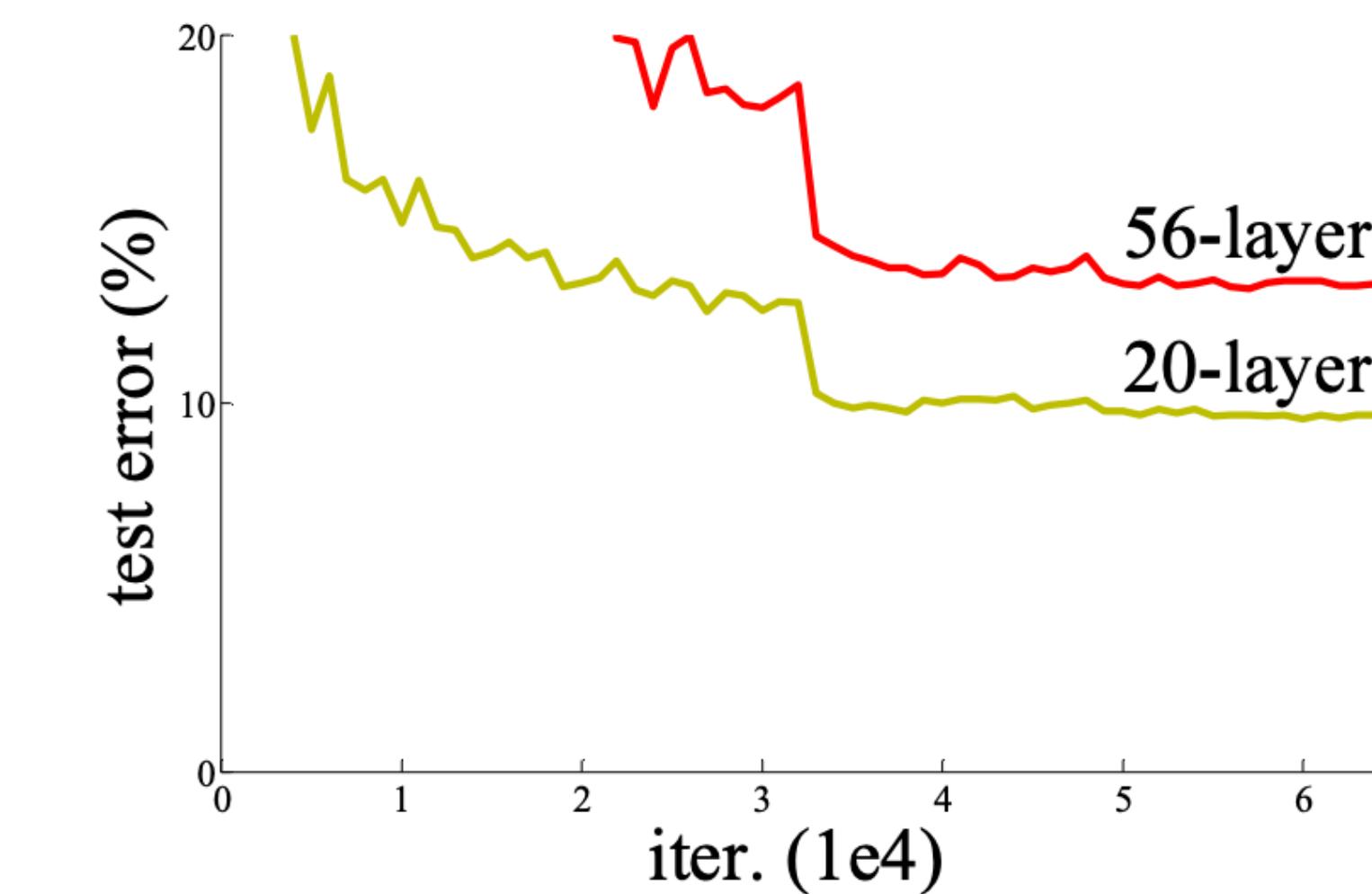
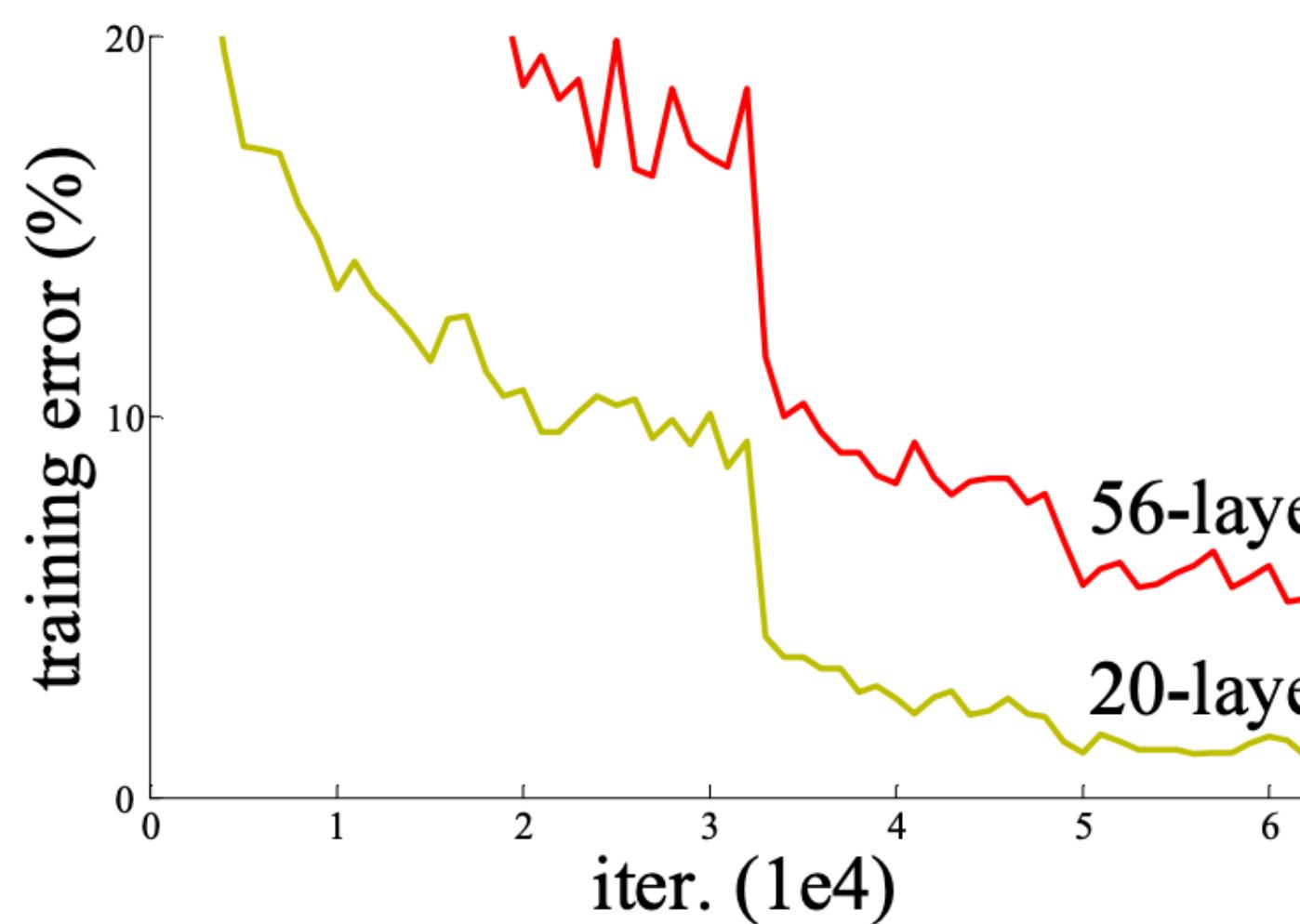


Просто увеличить число слоев не поможет

ResNet (He et al., 2015)



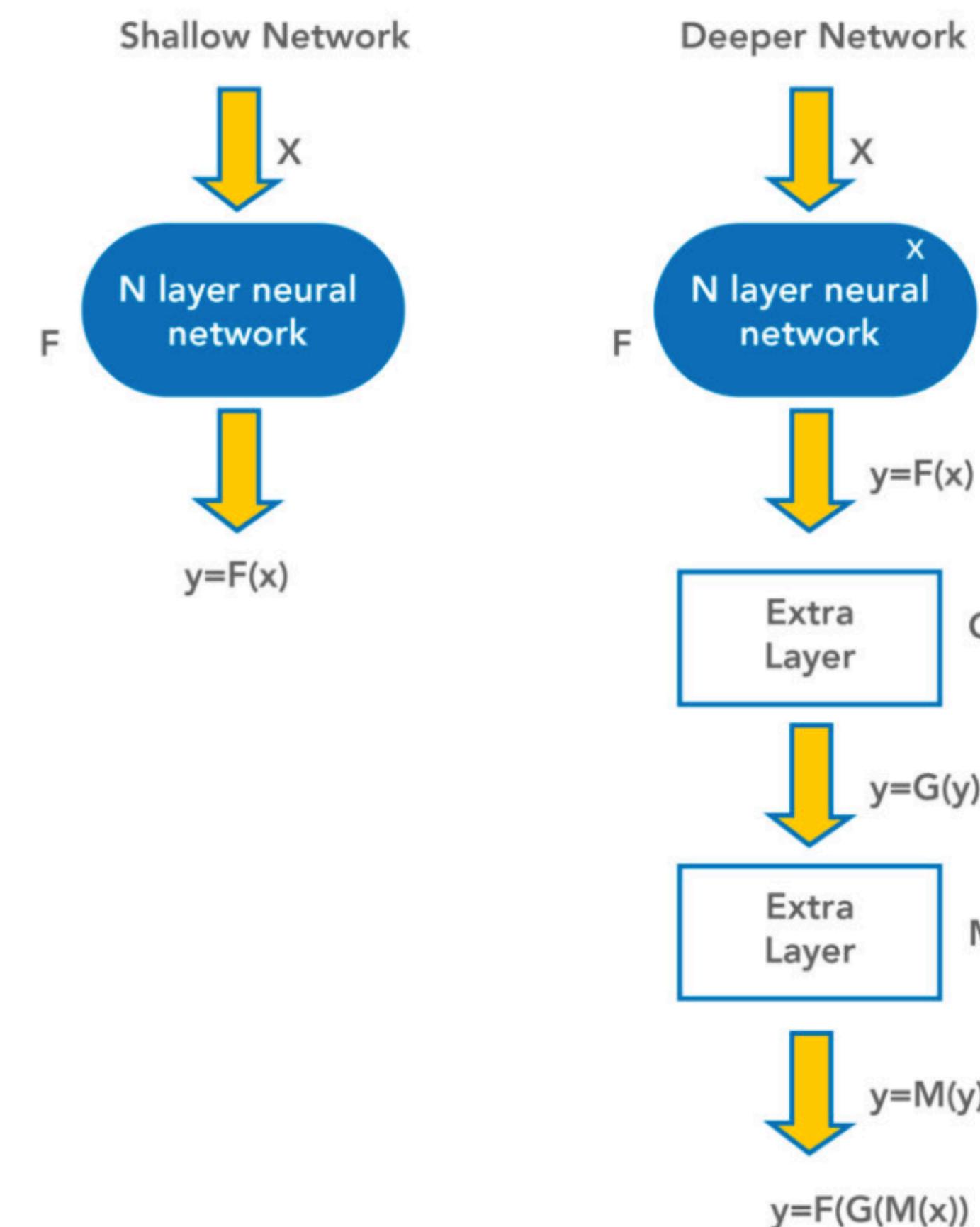
152 слоя!



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

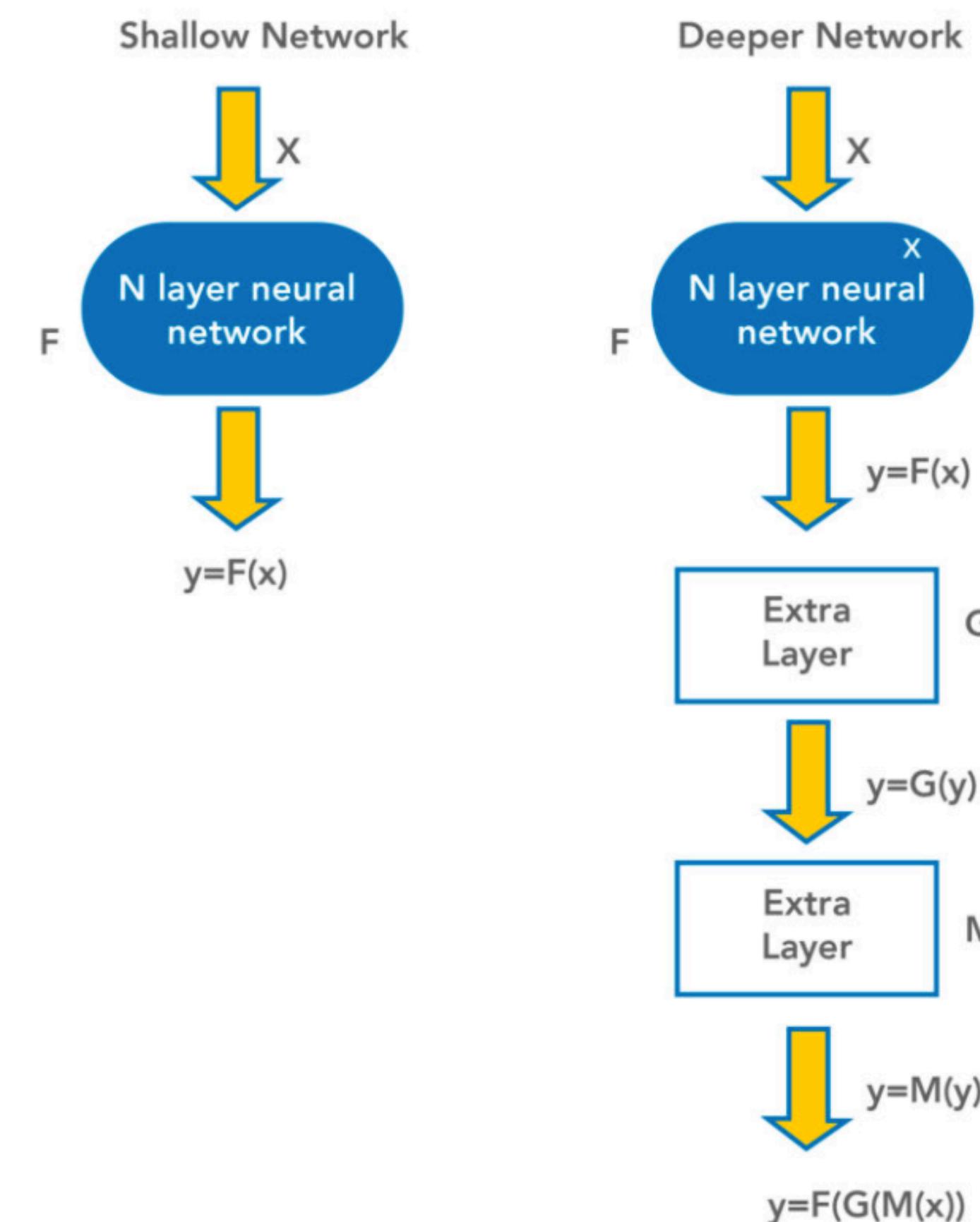
ResNet (He et al., 2015)

Как не испортить неглубокую нейросеть, добавляя слои?



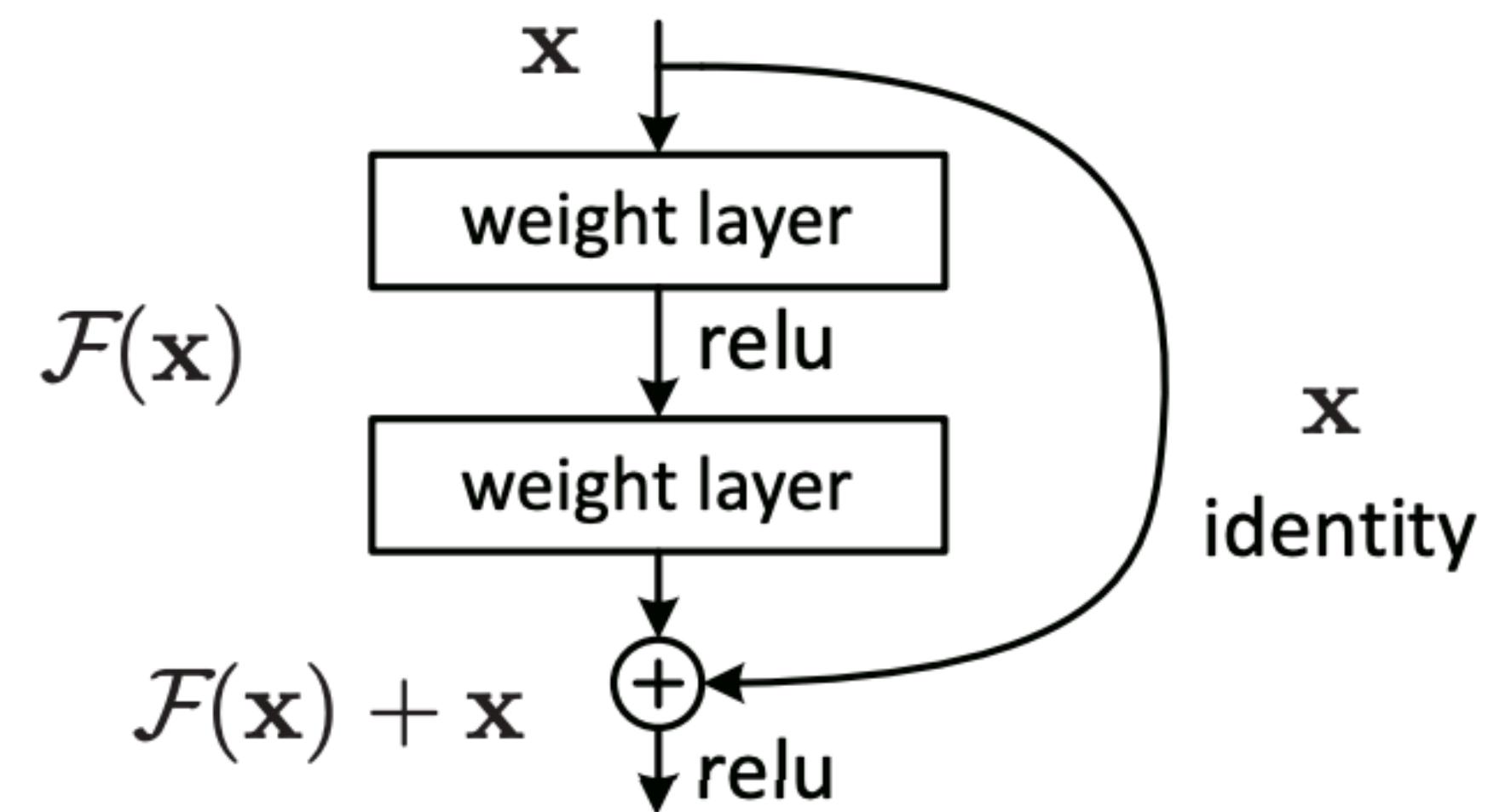
ResNet (He et al., 2015)

Как не испортить неглубокую нейросеть, добавляя слои? Identity blocks



ResNet (He et al., 2015)

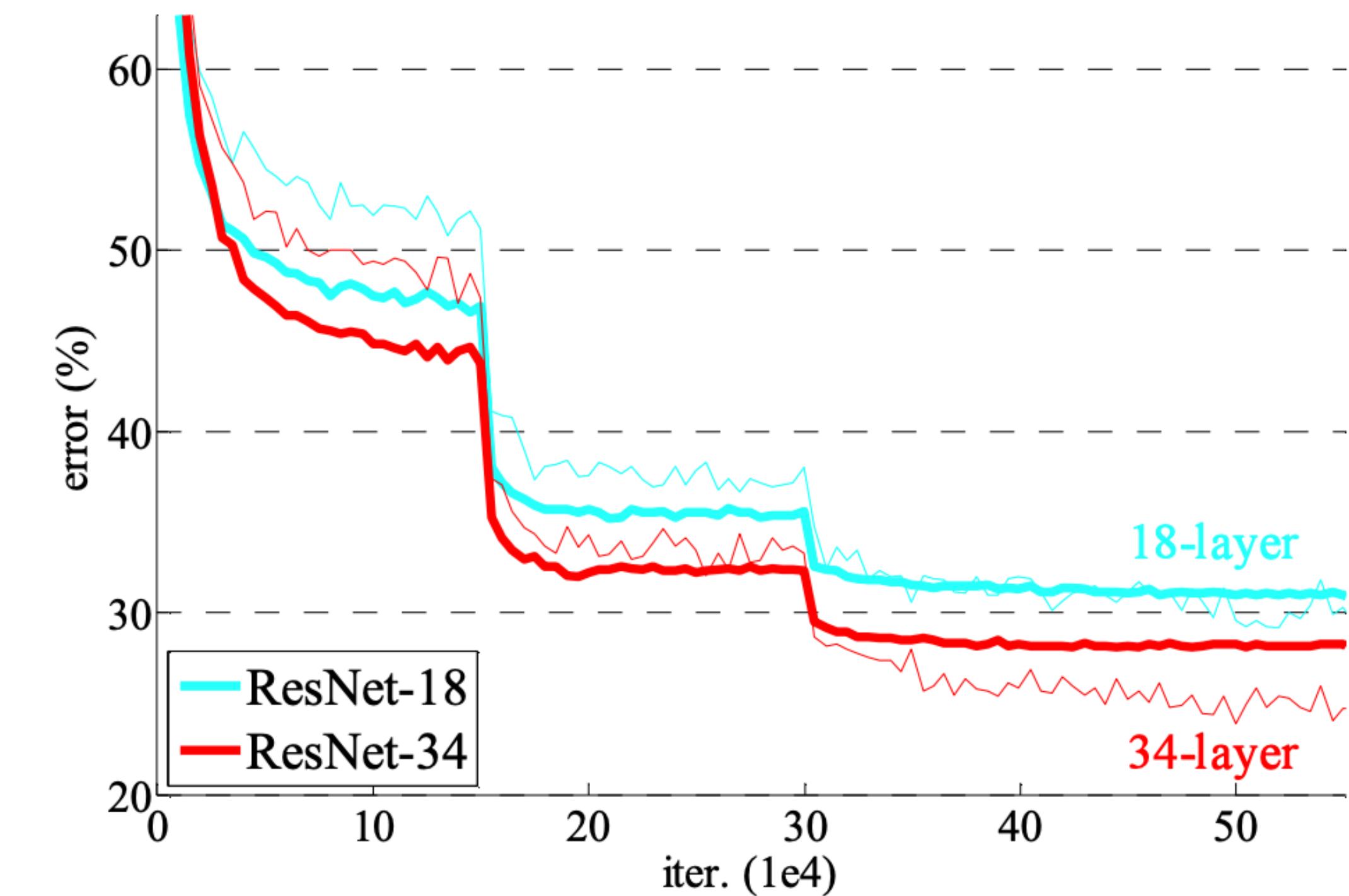
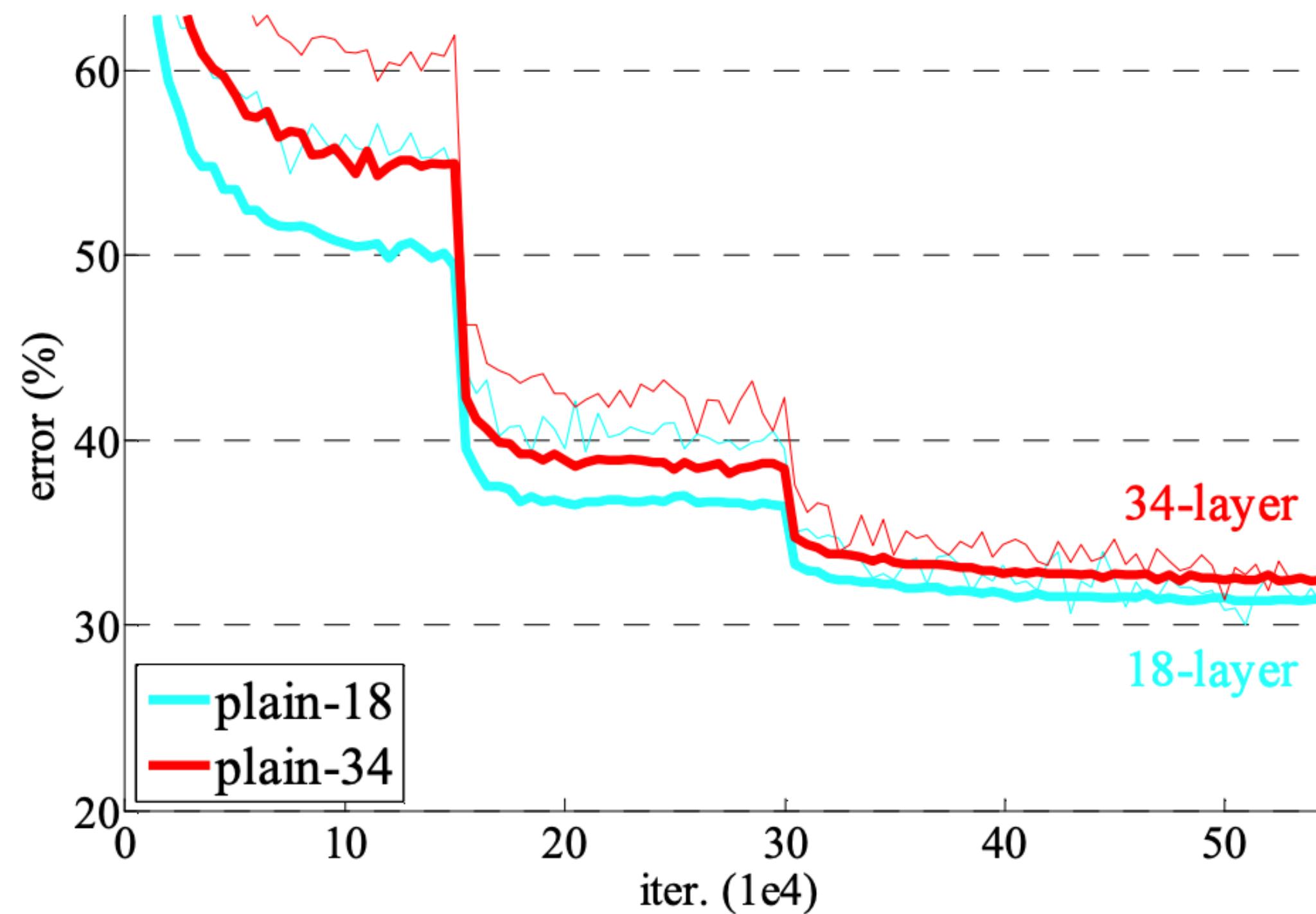
Residual block



$$H(x) = \mathcal{F}(x) + x$$

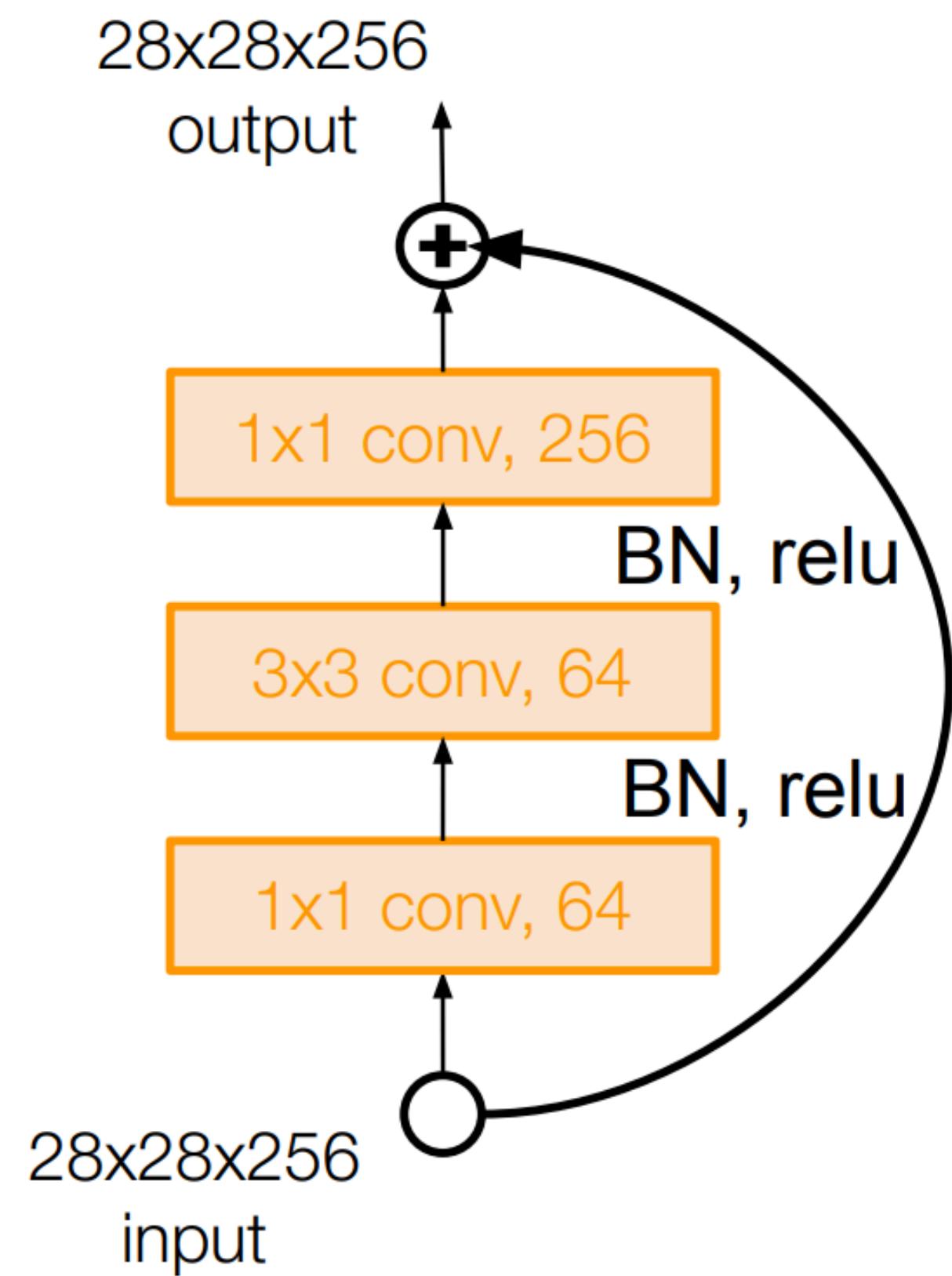
$$H(x) = x \text{ if } \mathcal{F}(x) = 0$$

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



Как обучать нейросети?

Как обучать нейросети?

Основной принцип - от простого к сложному

Как обучать нейросети?

Основной принцип - от простого к сложному

- Для начала берите простой baseline
 - вы должны знать, какой результат ожидать
 - dummy solution/существующая обученная нейросеть

Как обучать нейросети?

Основной принцип - от простого к сложному

- Для начала берите простой baseline
- Разберитесь с чтением/обработкой данных

Как обучать нейросети?

Основной принцип - от простого к сложному

- Для начала берите простой baseline
- Разберитесь с чтением/обработкой данных
- Выберите метрики и loss

Как обучать нейросети?

Основной принцип - от простого к сложному

- Для начала берите простой baseline
- Разберитесь с чтением/обработкой данных
- Выберите метрики и loss
- Настройте логирование и визуализацию: Tensorboard, Weights&Biases

Как обучать нейросети?

Основной принцип - от простого к сложному

- Для начала берите простой baseline
- Разберитесь с чтением/обработкой данных
- Выберите метрики и loss
- Настройте логирование и визуализацию: Tensorboard, Weights&Biases
- Фиксируйте random seed!

Как обучать нейросети?

Основной принцип - от простого к сложному

- Для начала берите простой baseline
- Разберитесь с чтением/обработкой данных
- Выберите метрики и loss
- Настройте логирование и визуализацию: Tensorboard, Weights&Biases
- Фиксируйте random seed!
- Для начала пробуйте обучаться на простых данных (подвыборка/синтетика)

Можно проверить переобучение на один батч/на маленькую подборку

Как обучать нейросети?

Пользуйтесь гитом, сохраняйте код, модель, ее конфигурацию и результаты

Как обучать нейросети?

Одно улучшение за раз!

Как обучать нейросети?

Одно улучшение за раз!

- Регуляризация при переобучении

Как обучать нейросети?

Одно улучшение за раз!

- Регуляризация при переобучении
- Аугментации

Как обучать нейросети?

Одно улучшение за раз!

- Регуляризация при переобучении
- Аугментации
- Качественная предобработка

Как обучать нейросети?

Одно улучшение за раз!

- Регуляризация при переобучении
- Аугментации
- Качественная предобработка
- Loss-функция должна соответствовать метрике

Как обучать нейросети?

Одно улучшение за раз!

- Регуляризация при переобучении
- Аугментации
- Качественная предобработка
- Loss-функция должна соответствовать метрике
- Подбор оптимизаторов, lr scheduler, подбор гиперпараметров

Как обучать нейросети?

Одно улучшение за раз!

- Регуляризация при переобучении
- Аугментации
- Качественная предобработка
- Loss-функция должна соответствовать метрике
- Подбор оптимизаторов, lr scheduler, подбор гиперпараметров

Используйте известные модели и их существующие реализации!

- классические модели для CV - в [torchvision](#)