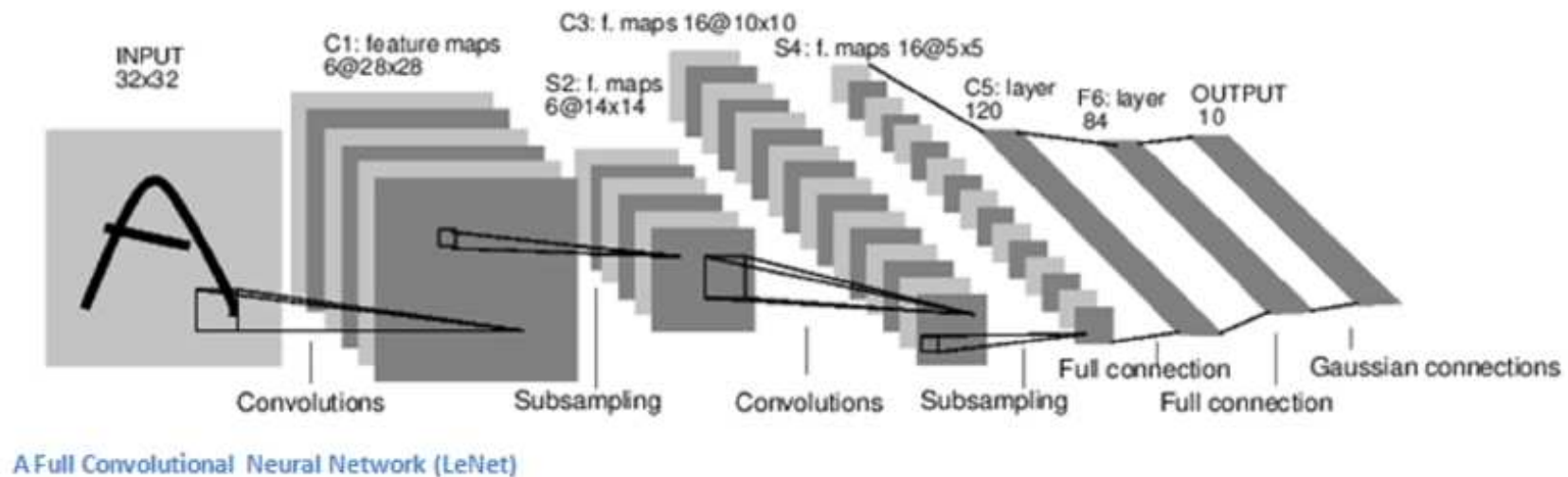


# Архитектуры CNN

# Свёрточная нейронная сеть



- Комбинация сверточных слоев, pooling-слоев, и полносвязных слоев

# Архитектура LeNet-5 (1998)

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

# ImageNet



flamingo



cock



ruffed grouse



quail



partridge

..



Egyptian cat



Persian cat



Siamese cat



tabby



lynx

..



dalmatian



keeshond



miniature schnauzer

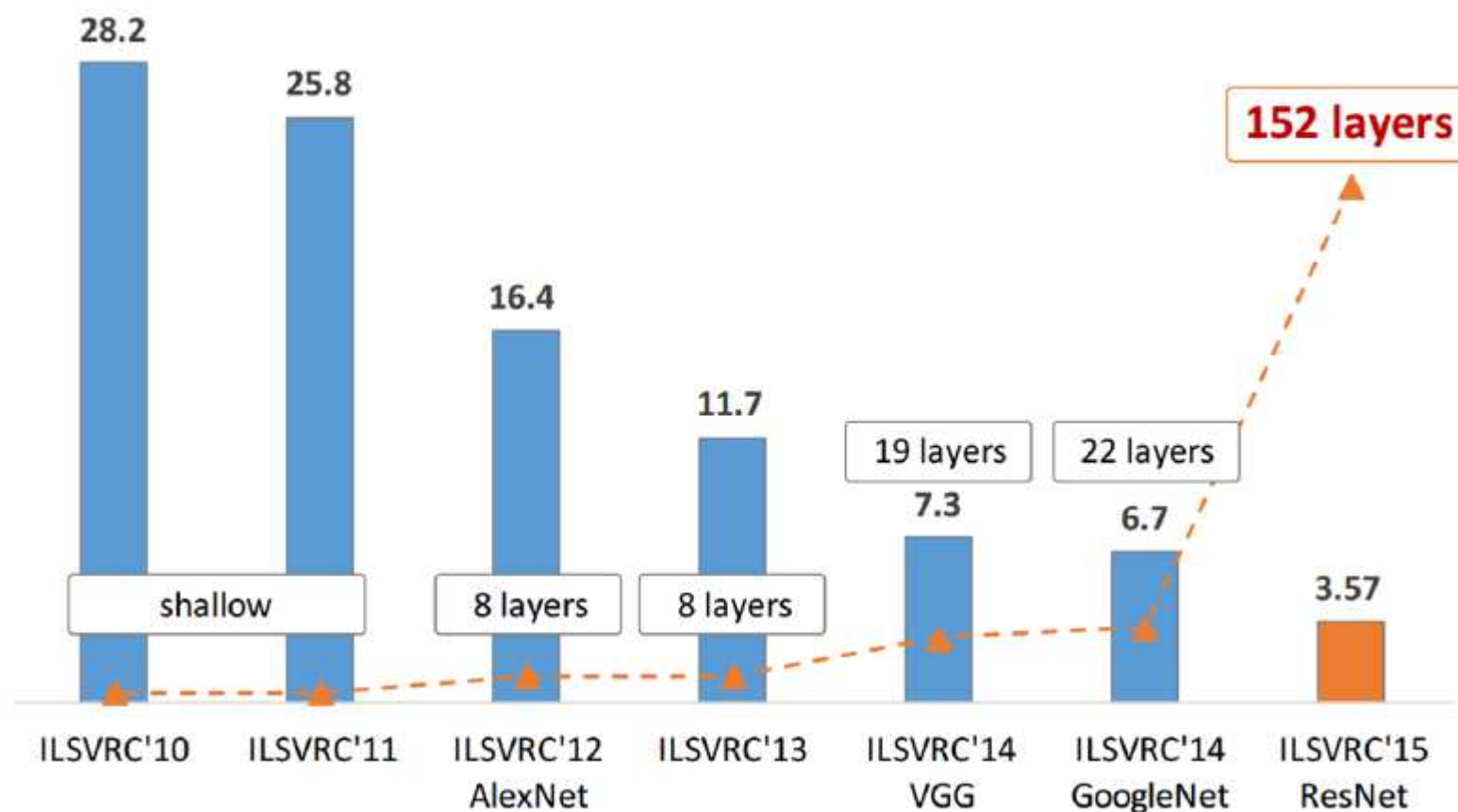


standard schnauzer



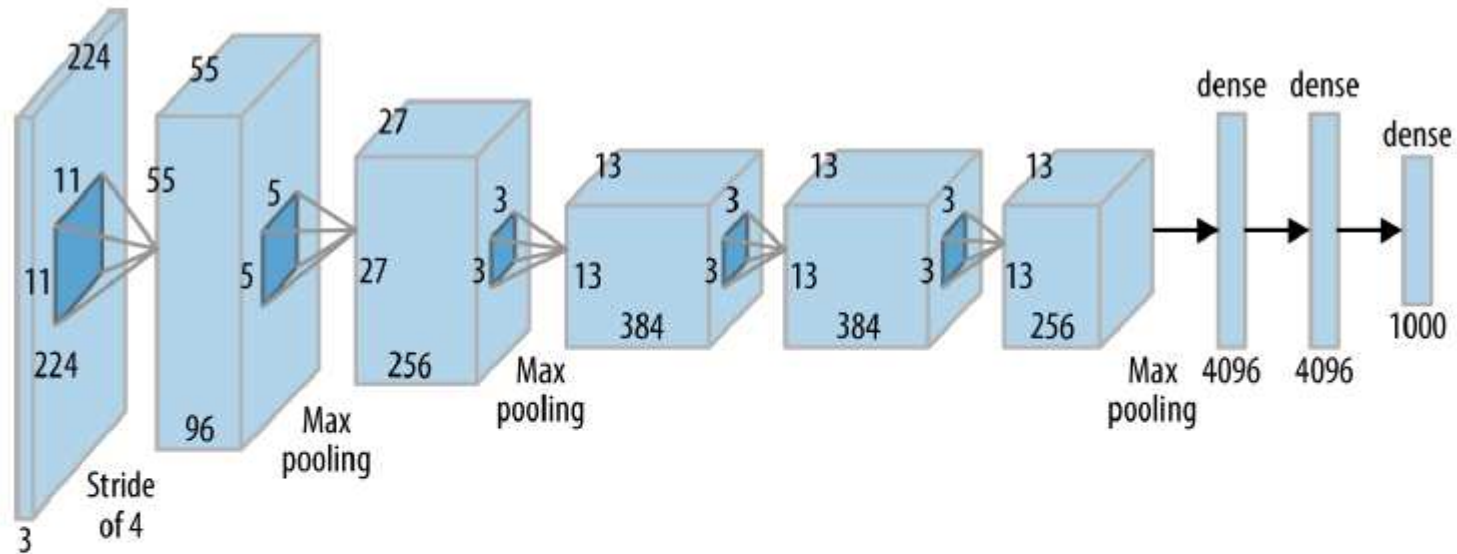
giant schnauzer

# ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



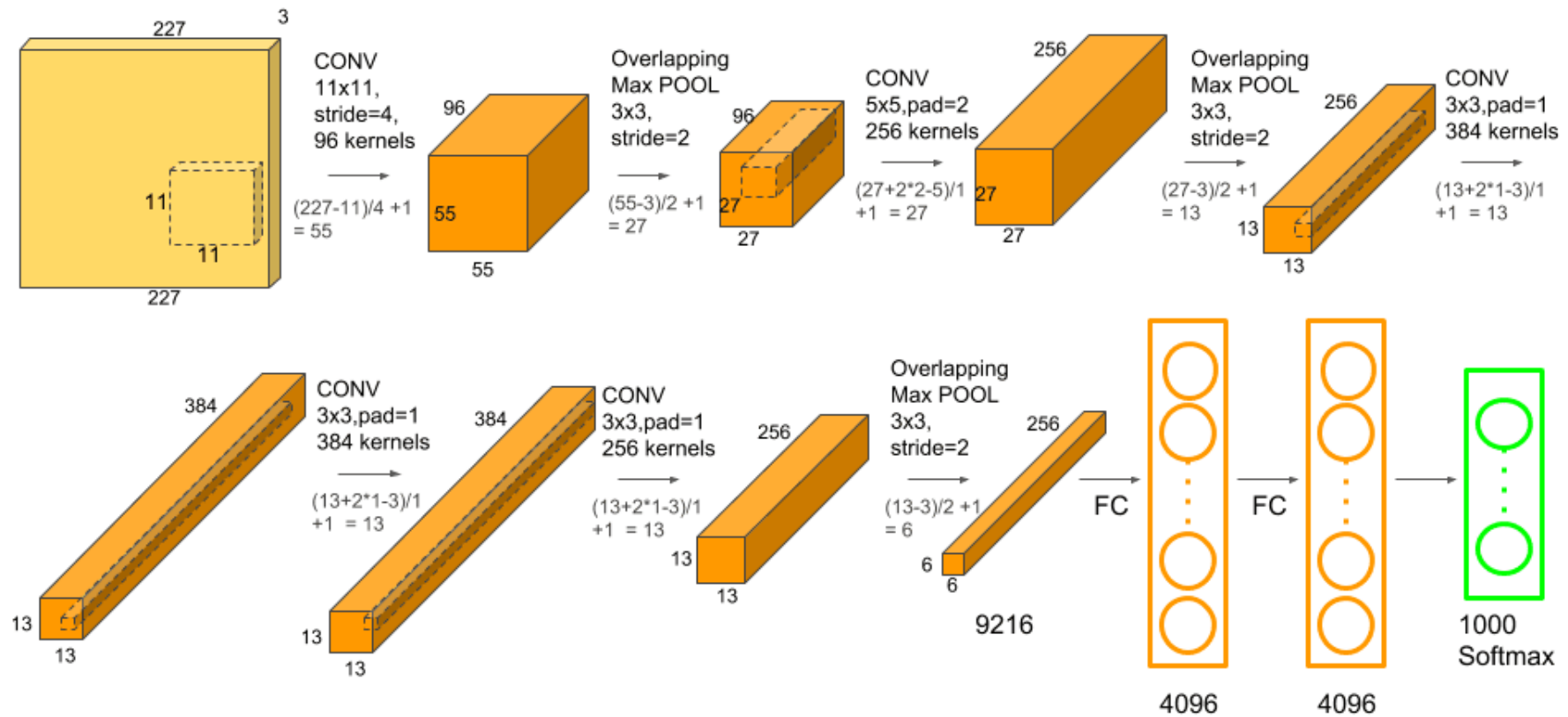
top-5 – одно из 5 предположений верно

# AlexNet



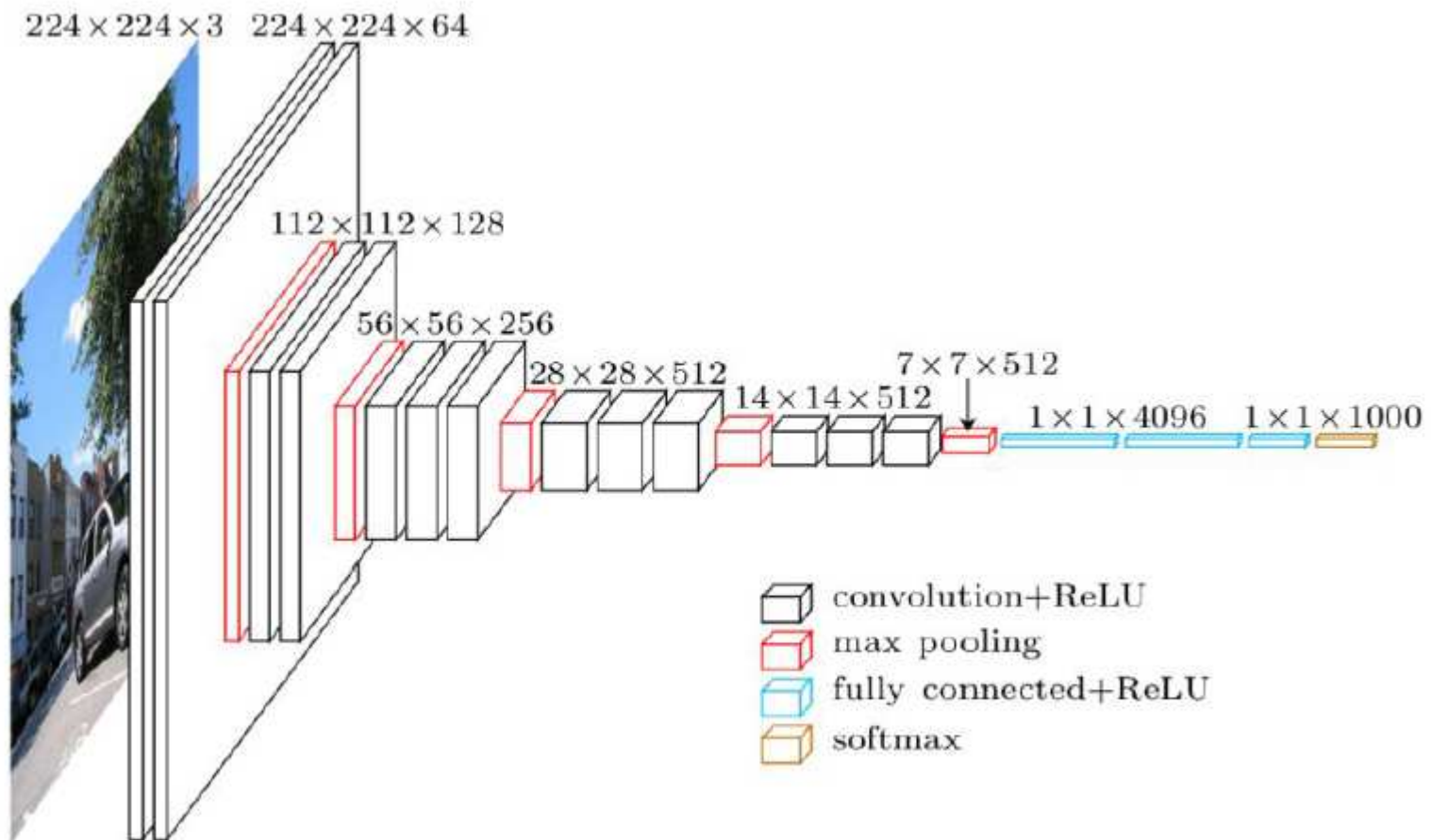
- ReLU + Dropout + пополнение выборки
- 60 млн параметров (в основном в полносвязных слоях)
- Подбор размеров фильтров и пулинга
- GPU

# AlexNet





# VGG (K. Simonyan, A. Zisserman)



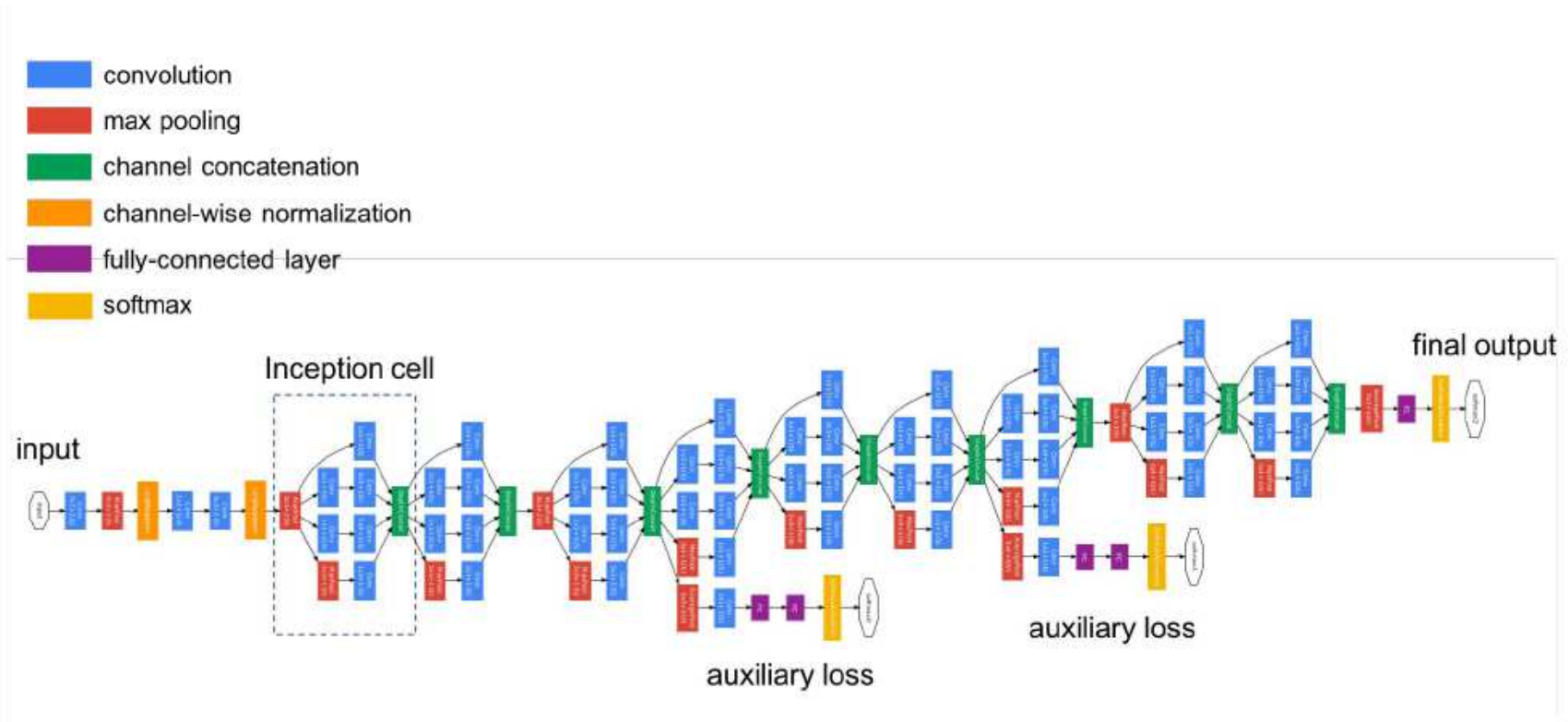


ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

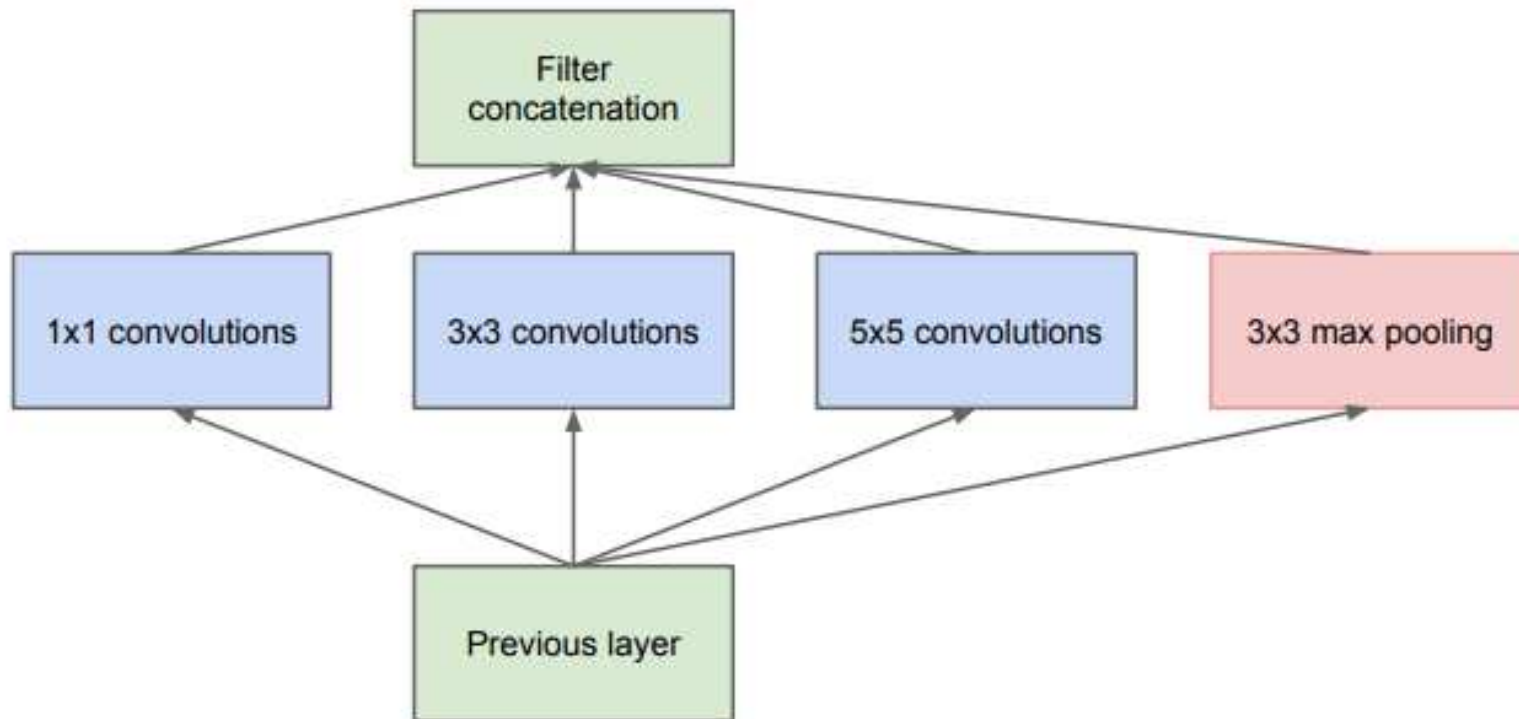
# VGG-16

- Свертки с ядром 3x3
- 130 миллионов параметров
- Обучение 3 недели на 4-х видеокартах  
NVIDIA TITAN BLACK

# GoogLeNet (Inception-v1)

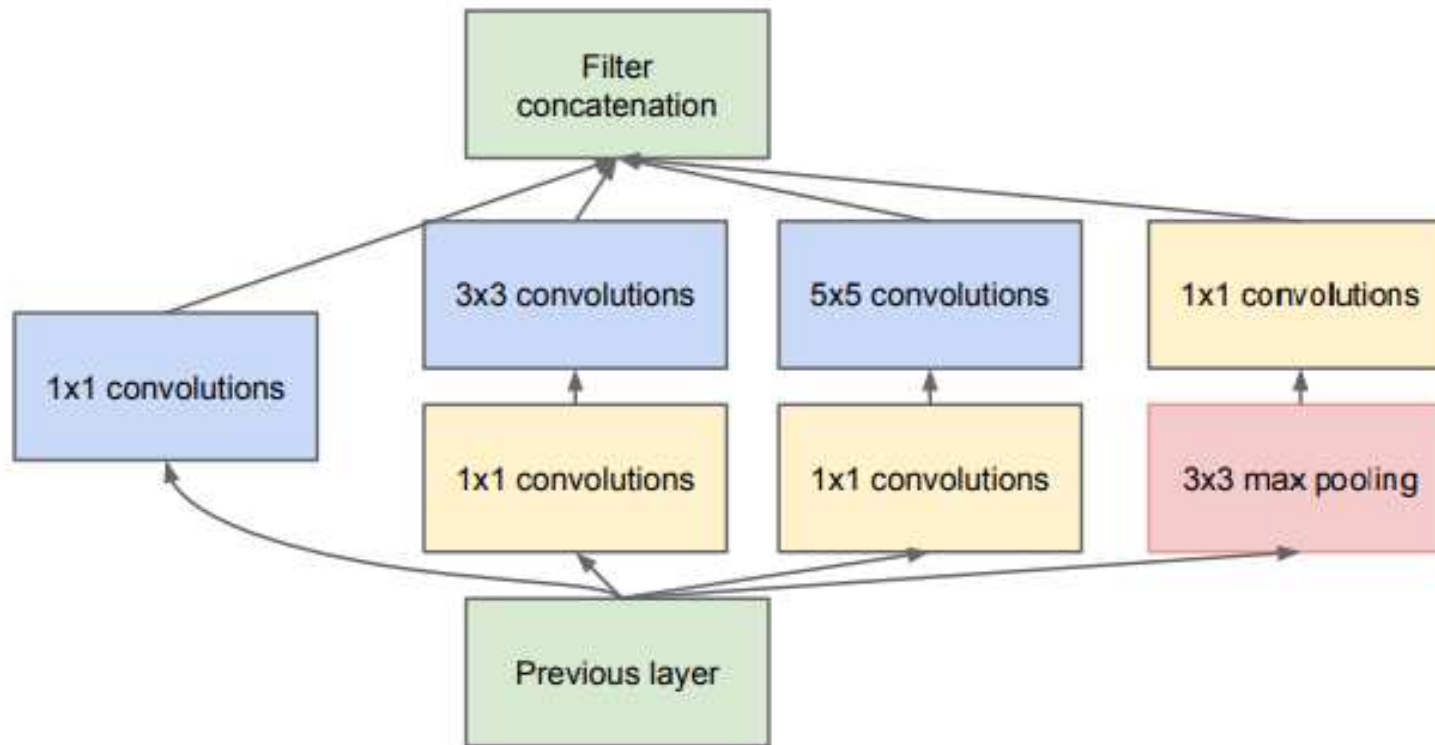


# Inception



(a) Inception module, naïve version

# Inception



(b) Inception module with dimension reductions

# GoogLeNet (Interception-v1)

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

# GoogLeNet

- 1x1 convolution = уменьшение размерности «пикселя»
- Inception
- Дополнительные выходы
- Только 1 полносвязный слой



# ResNet

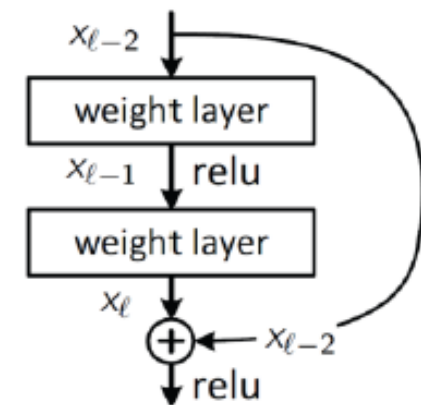


# ResNet

Сквозная связь (skip connection) слоя  $\ell$  с предшествующим слоем  $\ell - d$ :

$$x_\ell = \sigma(Wx_{\ell-1}) + x_{\ell-d}$$

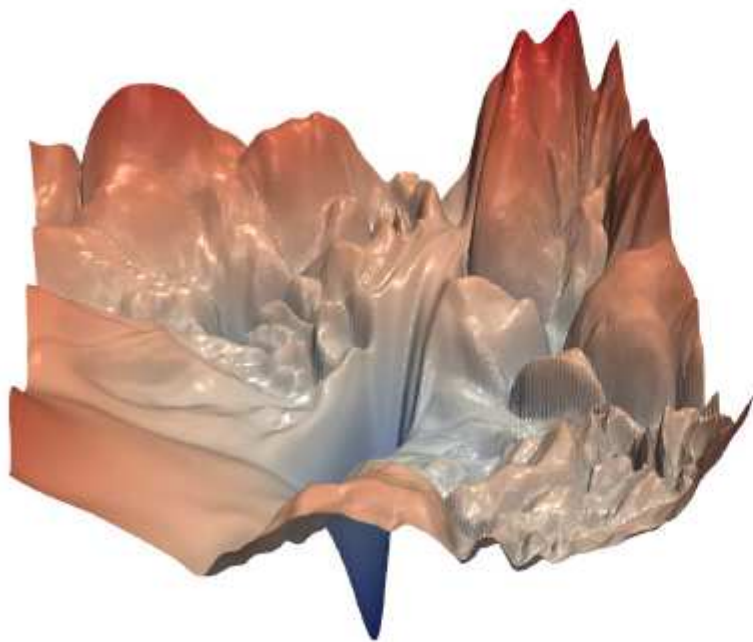
Слой  $\ell$  выучивает не новое векторное представление  $x_\ell$ , а его приращение  $x_\ell - x_{\ell-d}$



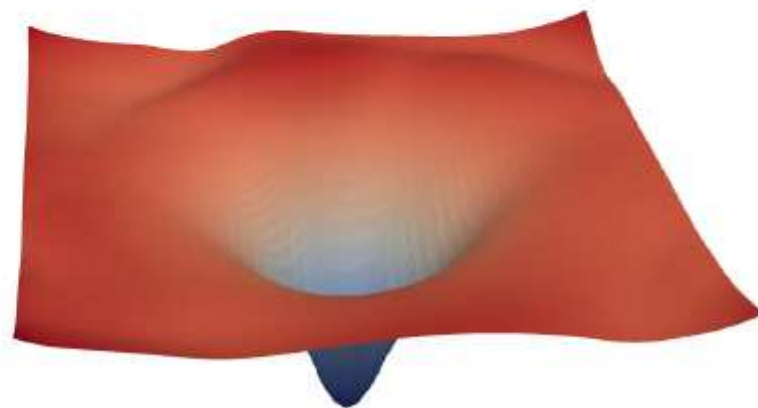
- Приращения более устойчивы  $\Rightarrow$  улучшается сходимость
- Появляется возможность увеличивать число слоёв
- Обобщение — Highway Networks:

$$x_\ell = \sigma(Wx_{\ell-1}) \underbrace{\tau(W'x_{\ell-1})}_{\text{transform gate}} + x_{\ell-d} \underbrace{(1 - \tau(W'x_{\ell-1}))}_{\text{carry gate}}$$

Сквозные связи упрощают оптимизируемый критерий,  
устраняя локальные экстремумы и седловые точки:



without skip connections



with skip connections

# Техника bottleneck

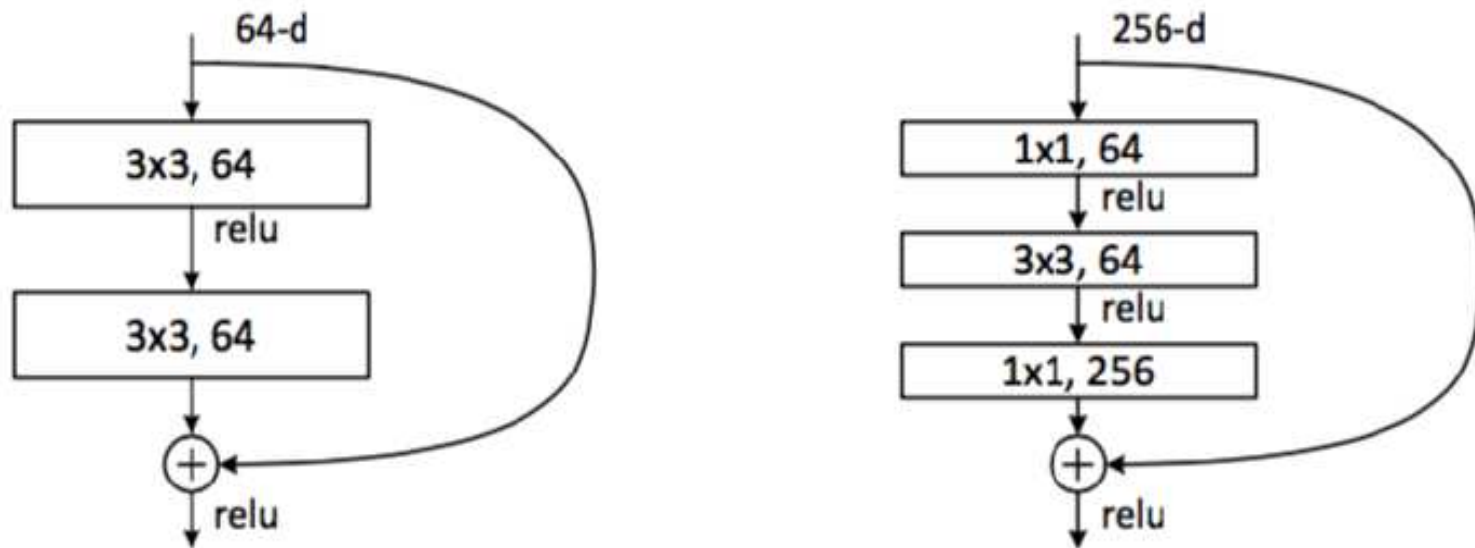
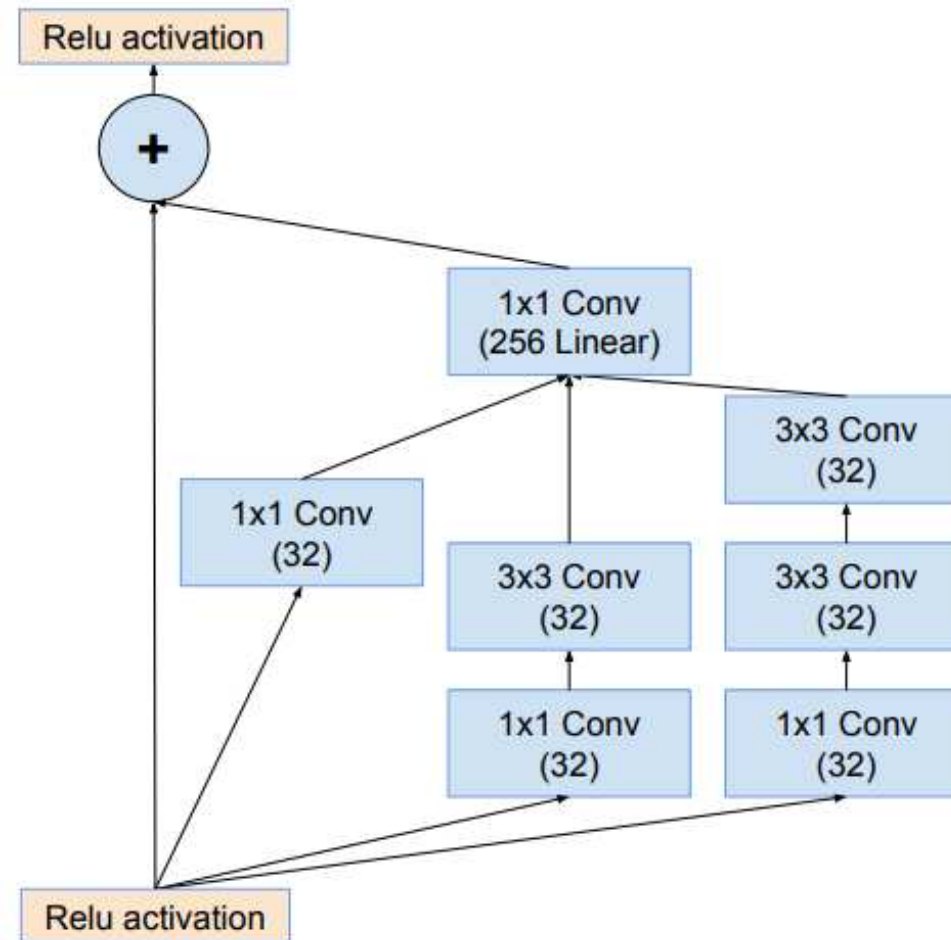


Рис. 11: Блок ResNet до и после применения bottleneck

# ResNet

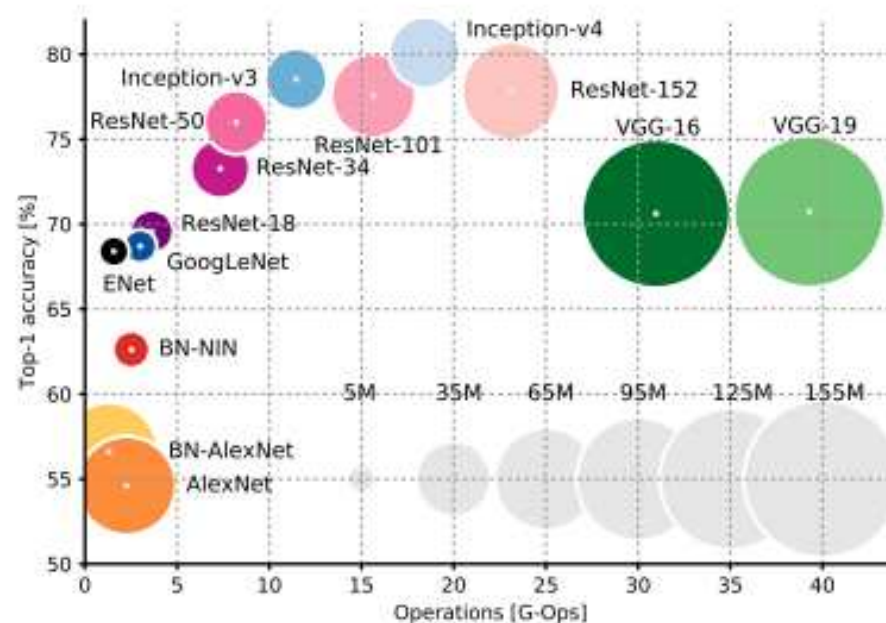
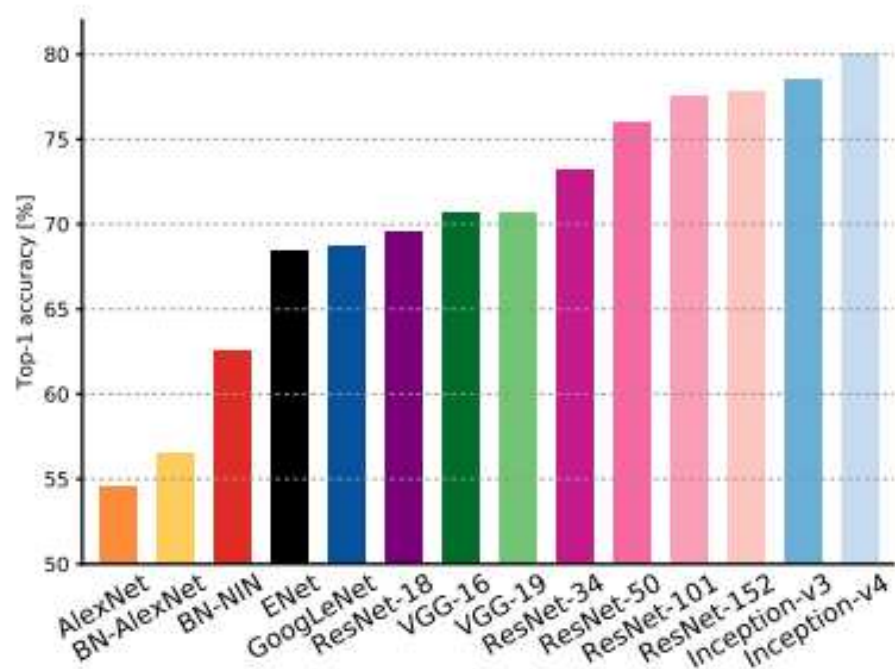
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

# Inception-Resnet





# Сравнение моделей по качеству и количеству операций



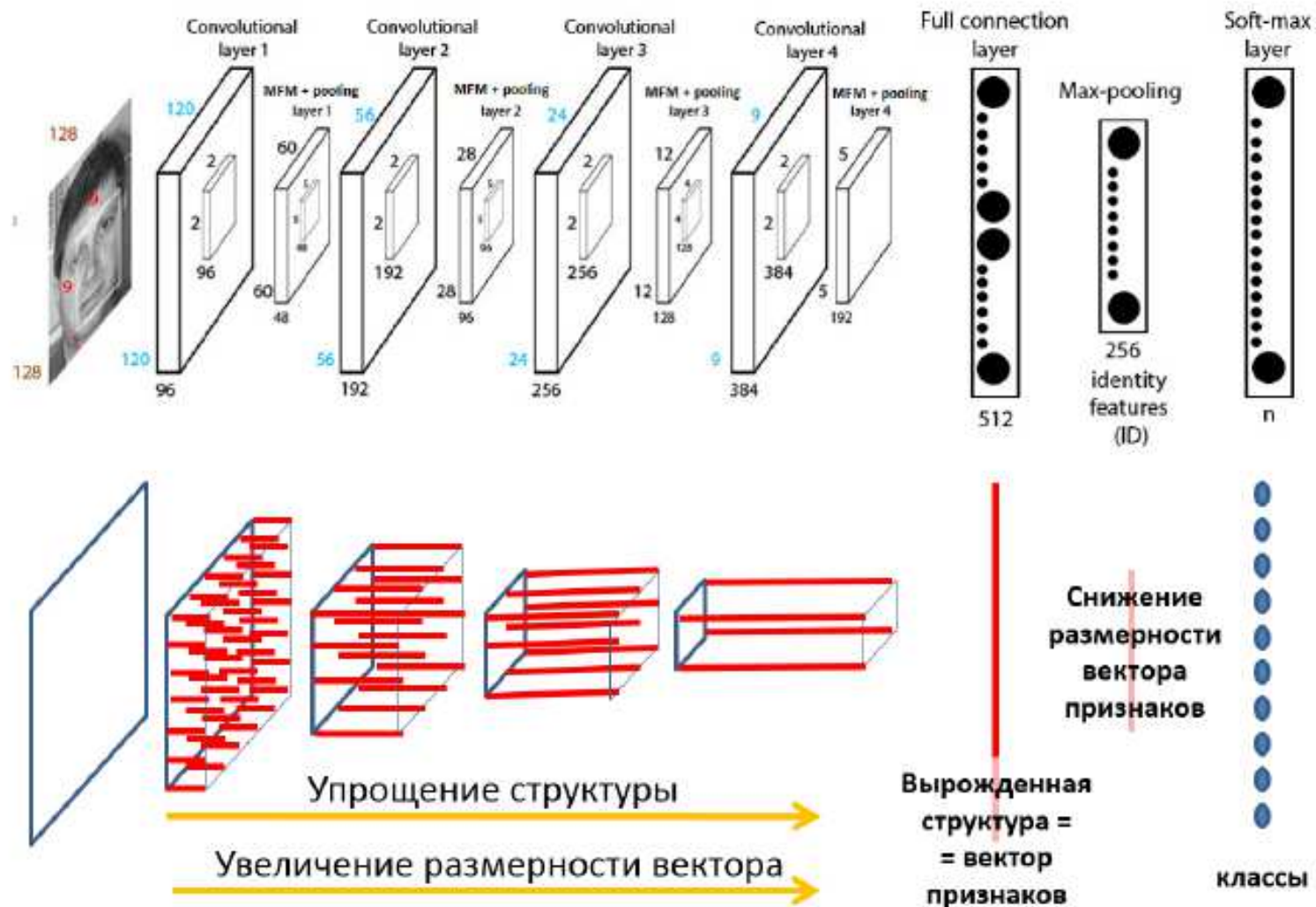


# Успешные приемы в архитектурах CNN

- функции активации без горизонтальных асимптот, типа ReLU
- адаптивные градиентные методы
- dropout
- batch normalization
- остаточные нейронные сети (Residual NN)
- подбор числа слоёв и их размеров
- dataset augmentation — пополнение выборки с помощью преобразований, сохраняющих класс объекта



# Векторизация изображений



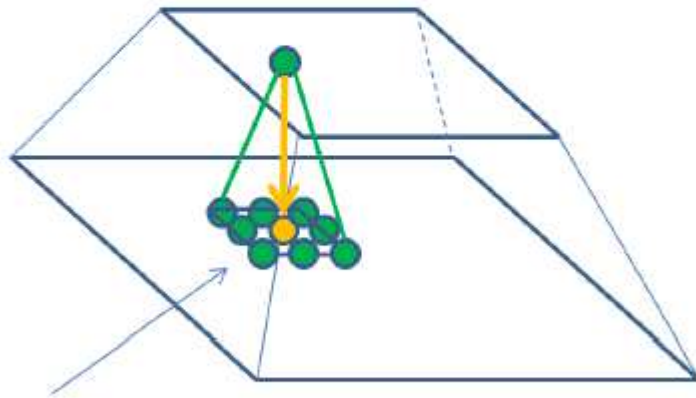
# Обобщение CNN на другие задачи

Допустим, каждый объект имеет структуру, заданную графом

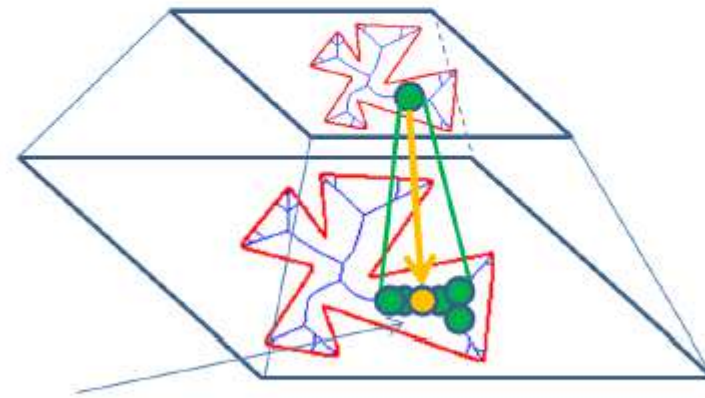
Свёртка определяется по локальной окрестности вершины

Пулинг агрегирует векторы вершин локальной окрестности

Такая сеть обучается находить и классифицировать подграфы



Прямоугольное окно заданного размера с центром в заданной точке + операция свёртки по окну



Локальная окрестность, определяемая для любой вершины графа + операция свёртки по окрестности

# Другие архитектуры

Модели для классификации на ImageNet:

- ResNeXt (<https://arxiv.org/abs/1611.05431>)
- SENet (<https://arxiv.org/abs/1709.01507>)
- DenseNet (<https://arxiv.org/abs/1608.06993>)
- Inception-ResNet-V2 (<https://arxiv.org/abs/1602.07261>)
- Inception-V4 (<https://arxiv.org/abs/1602.07261>)
- Xception (<https://arxiv.org/abs/1610.02357>)

Легковесные модели (для мобильных устройств):

- NASNet (<https://arxiv.org/abs/1707.07012>)
- MobileNetV2 (<https://arxiv.org/abs/1801.04381>)
- ShuffleNet (<https://arxiv.org/abs/1707.01083>)
- SqueezeNet (<https://arxiv.org/abs/1602.07360>)



# Transfer Learning

