

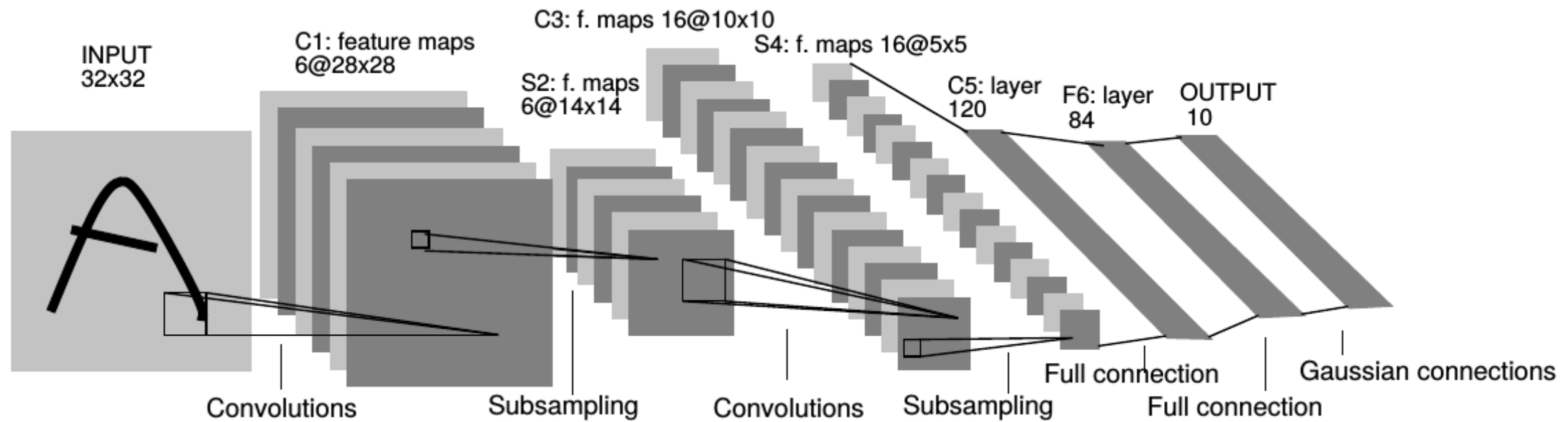
курс «Глубокое обучение»

Архитектуры свёрточных нейронных сетей

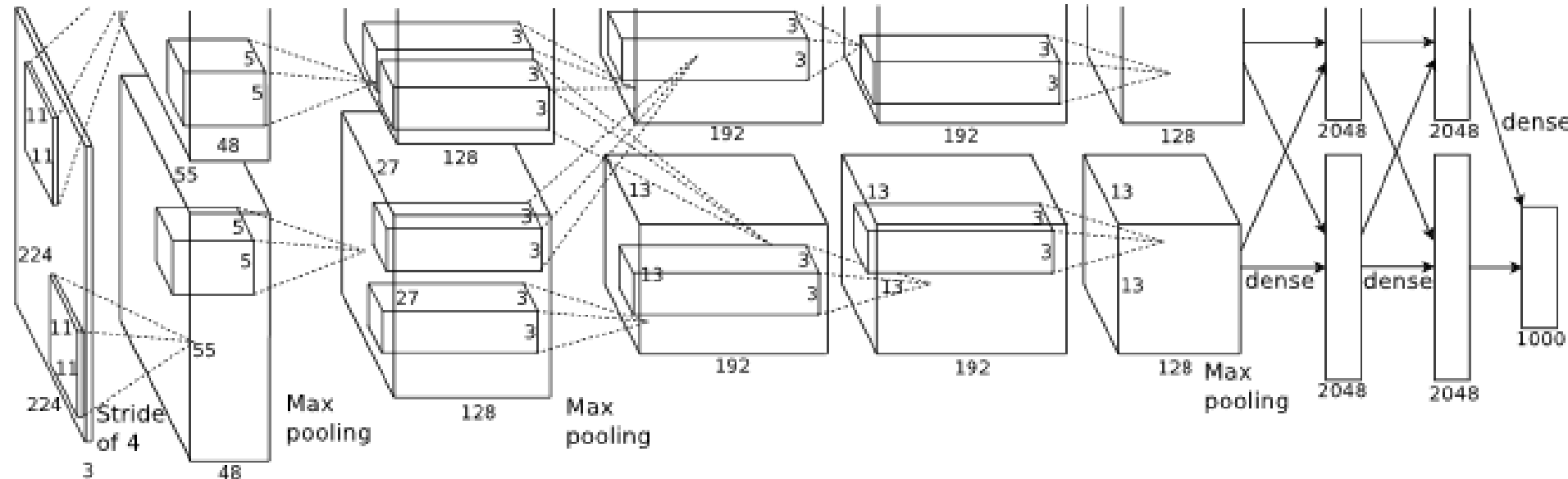
Александр Дьяконов

02 марта 2020 года

LeNet (1998)

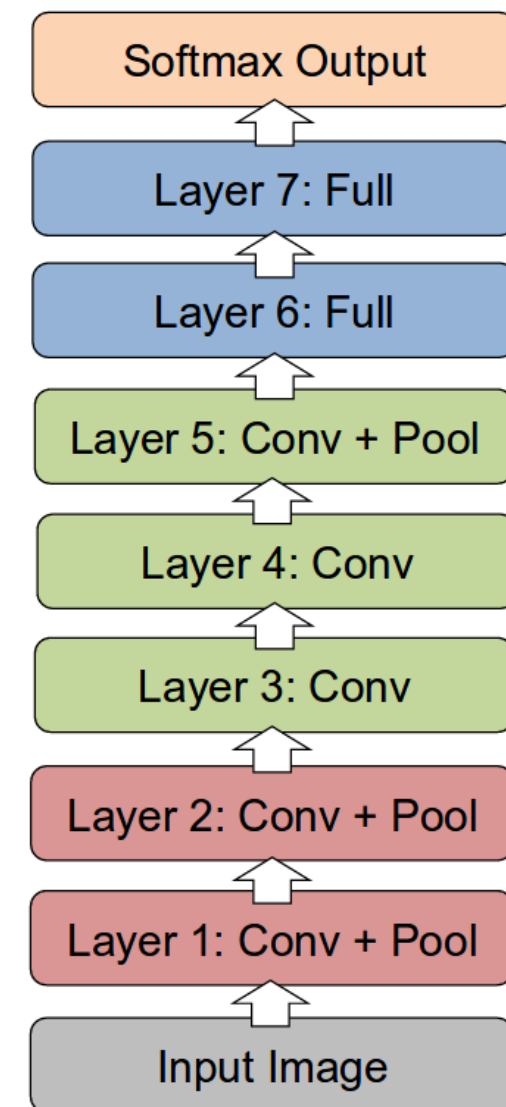


AlexNet (2012)



ReLU (скорость 6×)

- **Max-pooling, полно-связные слои**
 - **Data augmentation**
- **Dropout 0.5 (но и время обучения 2×)**
 - **Batch size = 128**
 - **SGD Momentum 0.9**
- **60M параметров / 650K нейронов**
- **1 неделя на 2 GPU (50x над CPU)**
 - **7 скрытых слоёв**



AlexNet (2012)

Dropout – перед 1м и 2м полносвязными слоями

свёртки 3×3, 5×5, 11×11

7 CNN ансамбль: 18.2% → 15.4%

Интересно

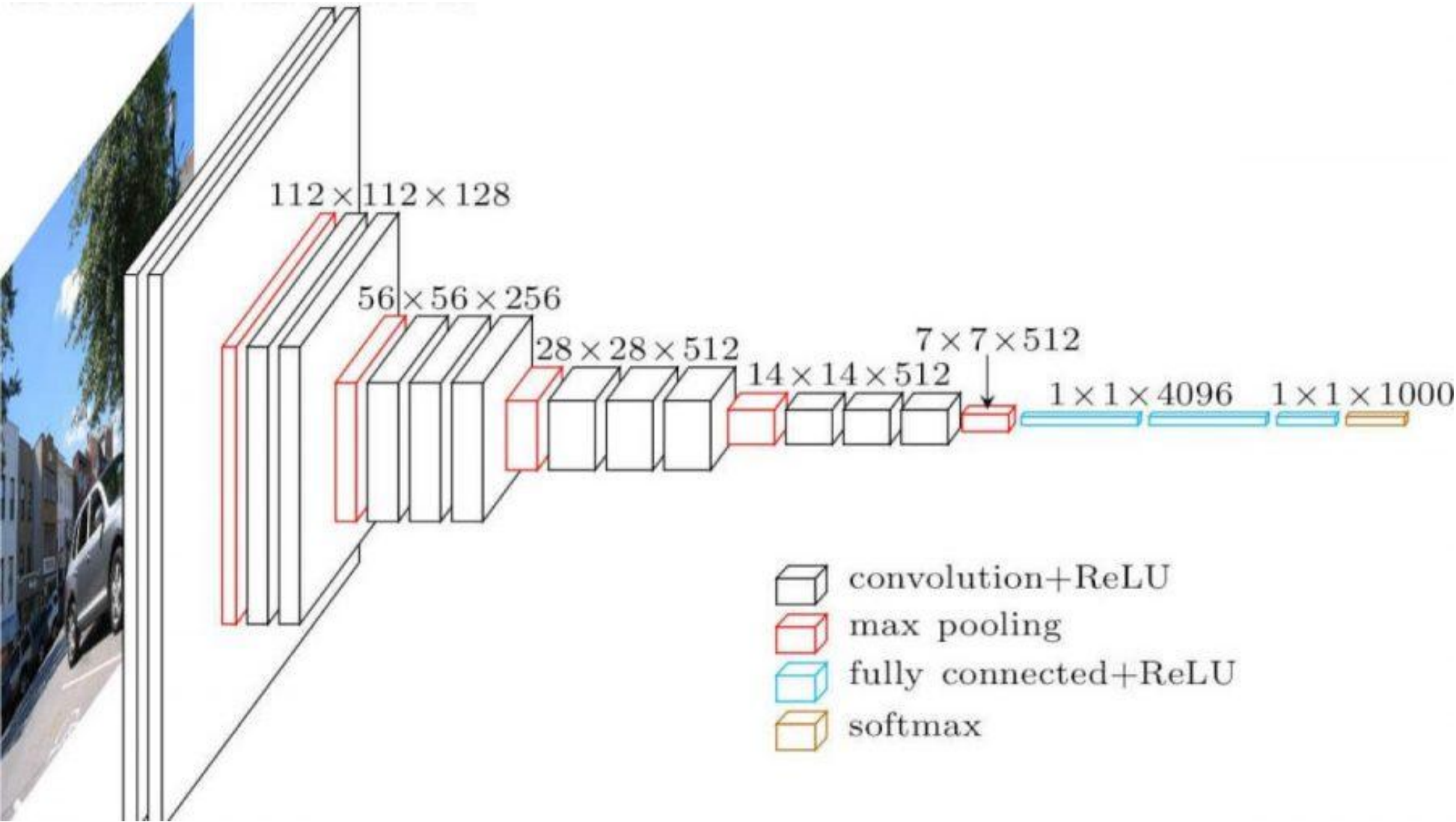
Убрать 16М параметров (последний полносвязный слой) – качество 1.1% ↓

Убрать 50М параметров (2 последних полносвязных слоя) – качество 5.7% ↓

Убрали 1М параметр (3 и 4 слои) – качество 3.0% ↓

Убрать несколько слоёв (3, 4, 6, 7) – качество 33.5% ↓

VGG (2014)

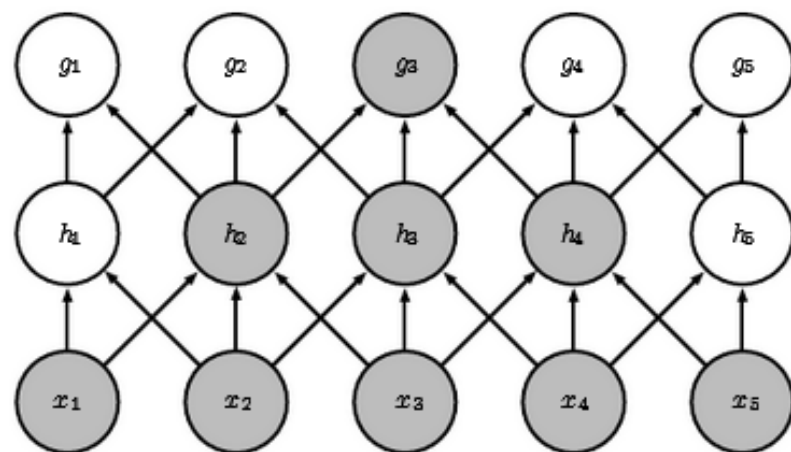


K. Simonyan, A. Zisserman «Very Deep Convolutional Networks for Large-Scale Image Recognition» <https://arxiv.org/pdf/1409.1556.pdf>

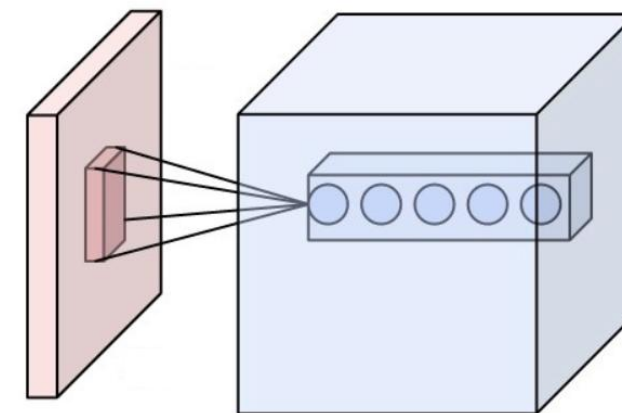
VGG (2014)

- вход = 256×256 (здесь 224×224)
- ReLu
- каскад 3×3 свёрток (замена 7×7)
- несколько стадий обучения
- 138M параметров (очень тяжеловесная)
- обучение сетей разных глубин
- 3 недели 4 GPU, тоже использовали ансамбль

Идея каскада свёрток



«Receptive field»

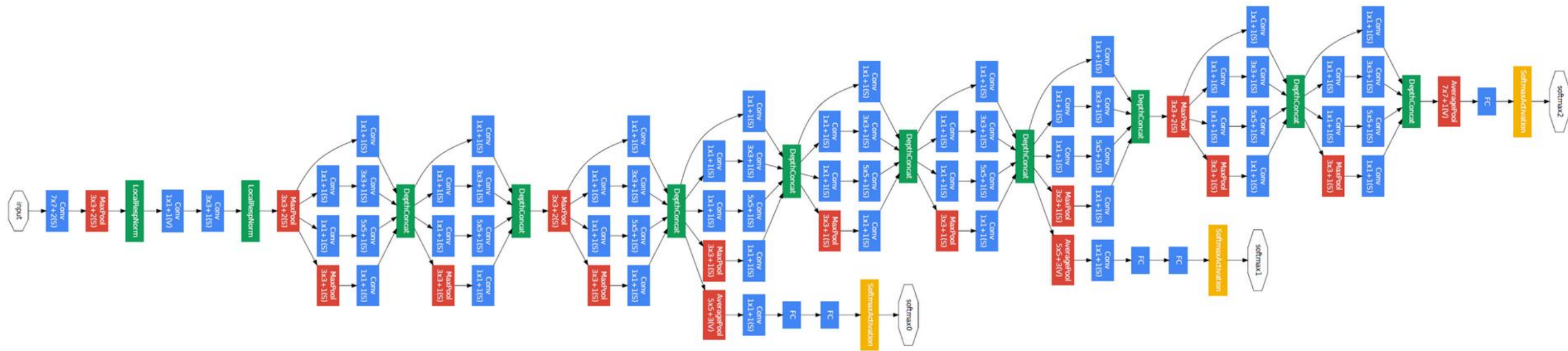


VGG 16 vs 19

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 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	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 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

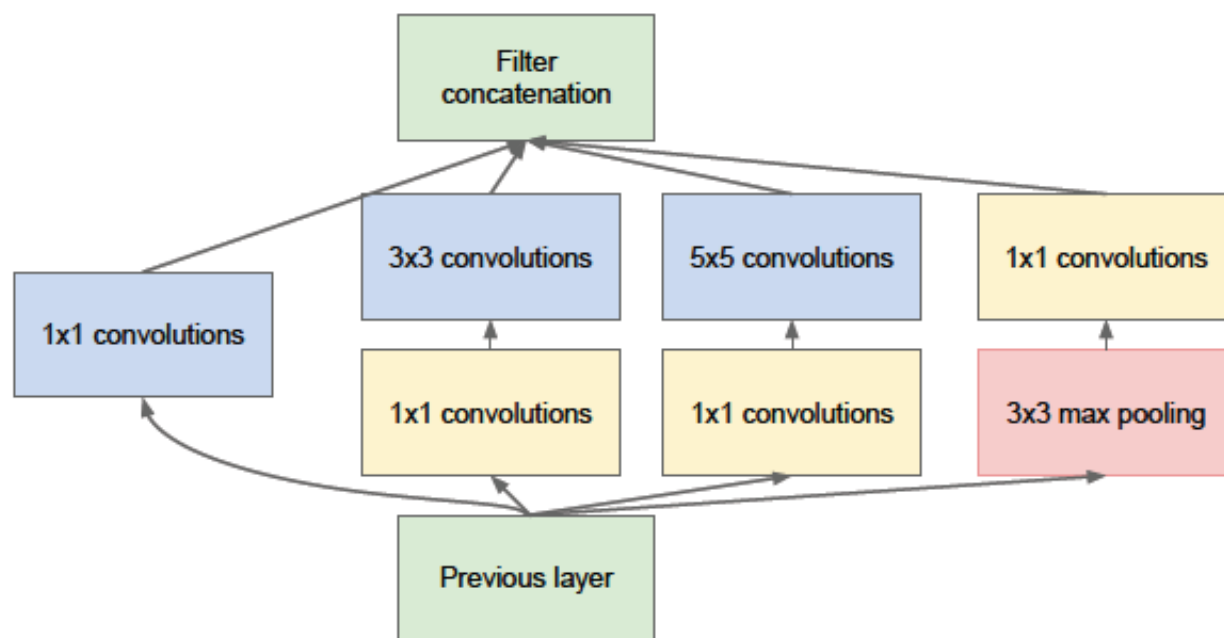
см. колонку D и E

GoogLeNet (2014)



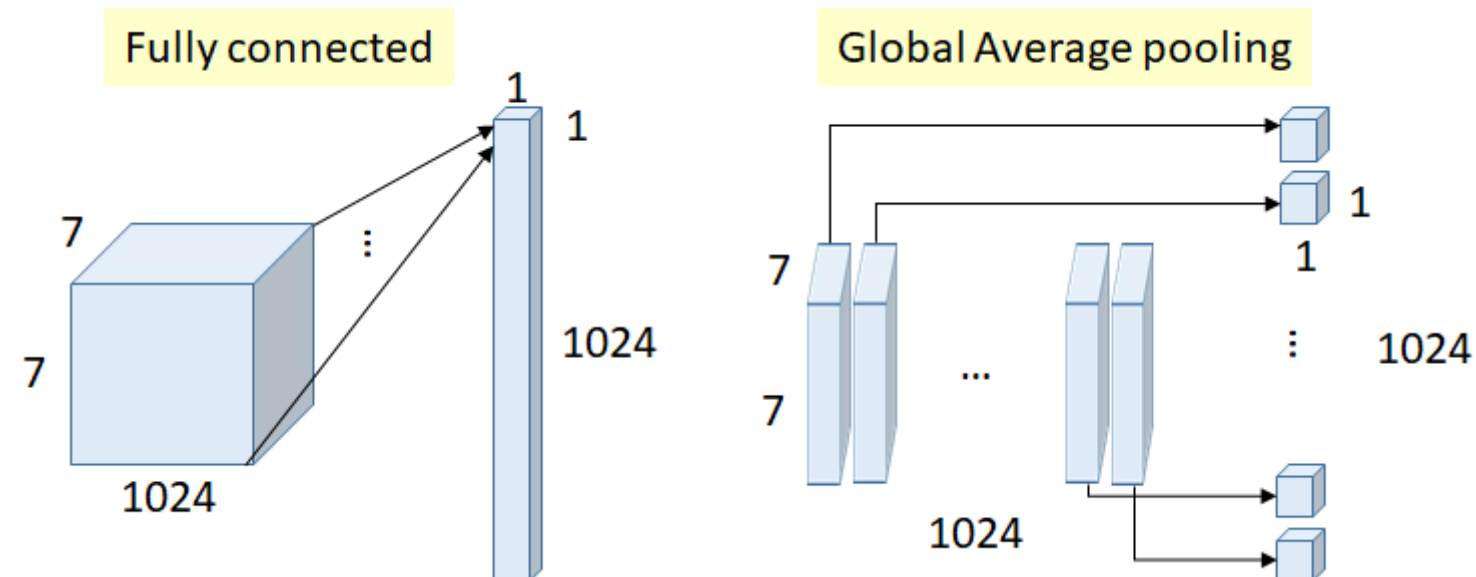
- «конструктор» НС
- 22 слоя – нет полносвязных
- Модуль «Inception», 1×1 -свёртки,
 - 5M параметров (меньше!)
- дополнительные выходы классификации
(с весом 0.3 к общей ошибке, для протекания градиента)
- тоже ансамбль (из 7)

Модуль «Inception»



**1×1-свёртки существенно уменьшают
число параметров!
... а идея была (синие блоки) – разные
свёртки + пулинг**

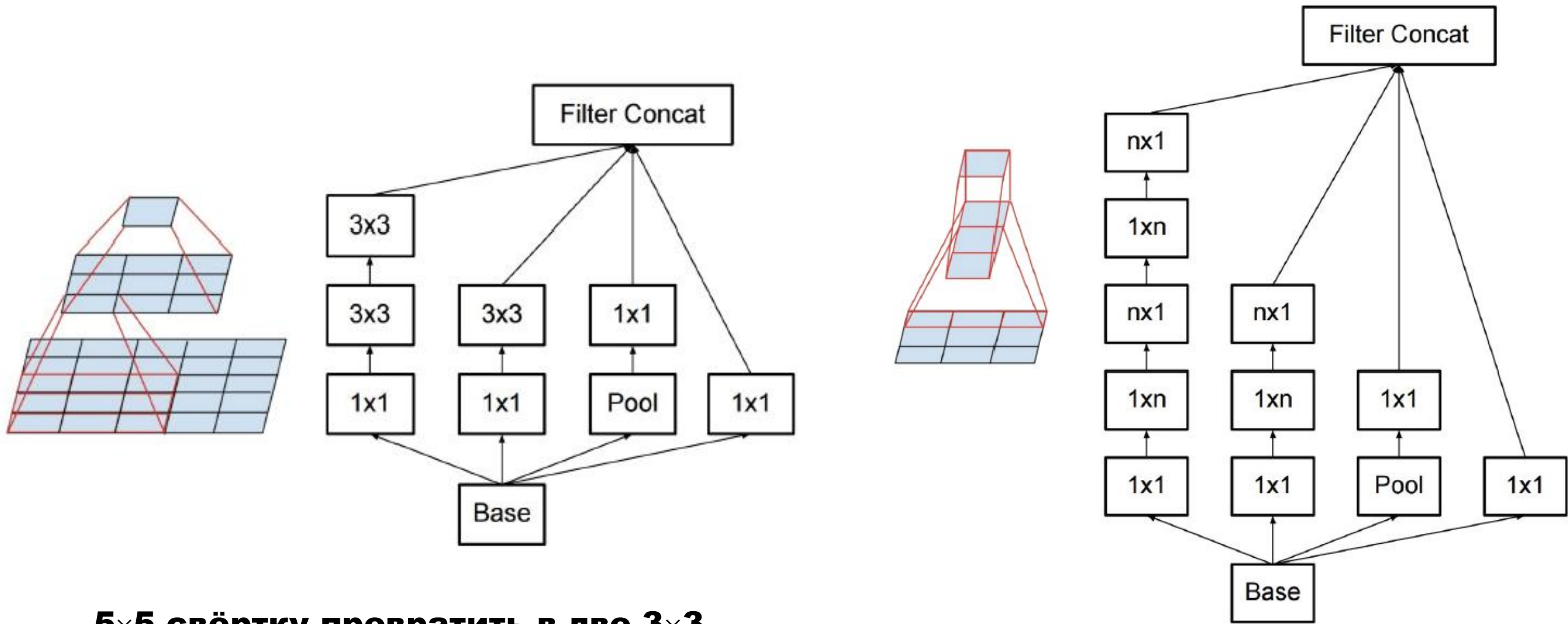
Global Average Pooling



Немного улучшает качество

Inception v2, v3

Другое строение модулей (+batchnorm)



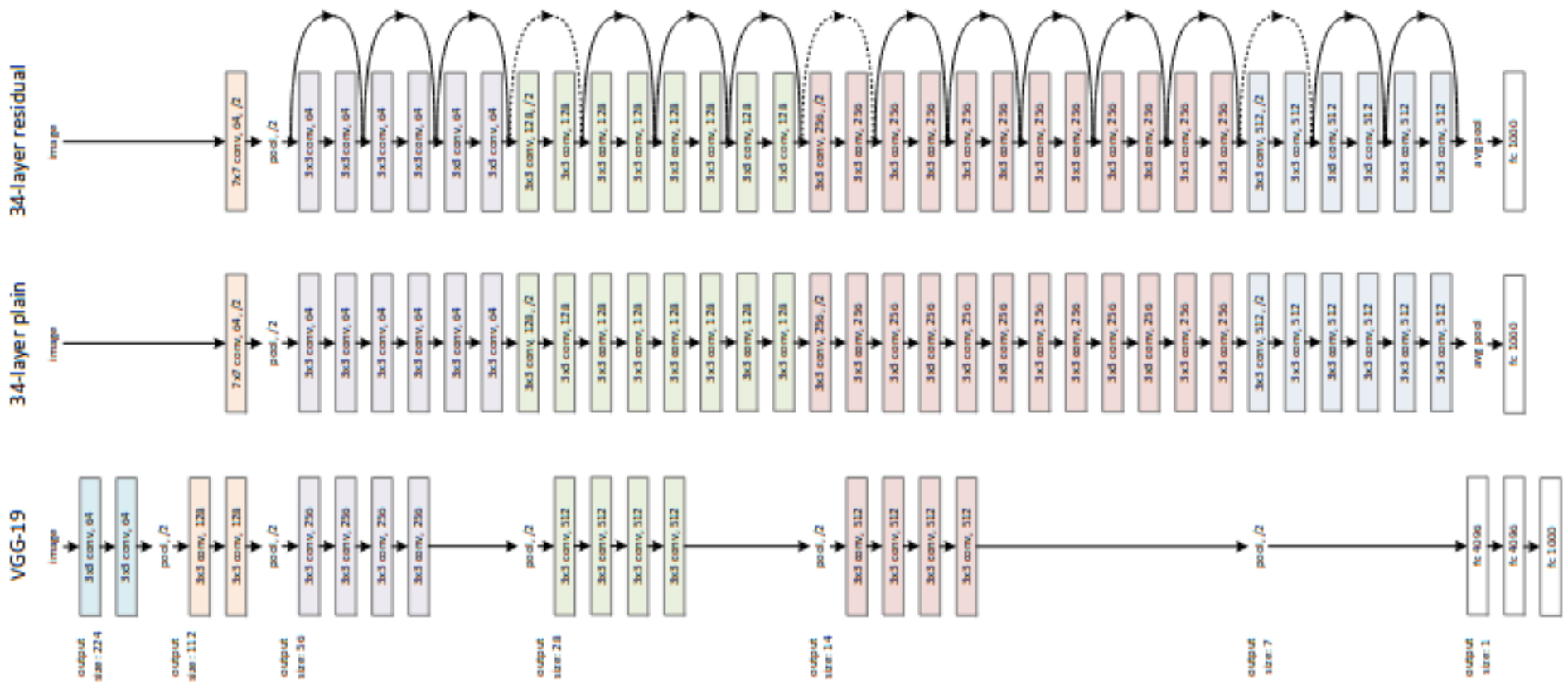
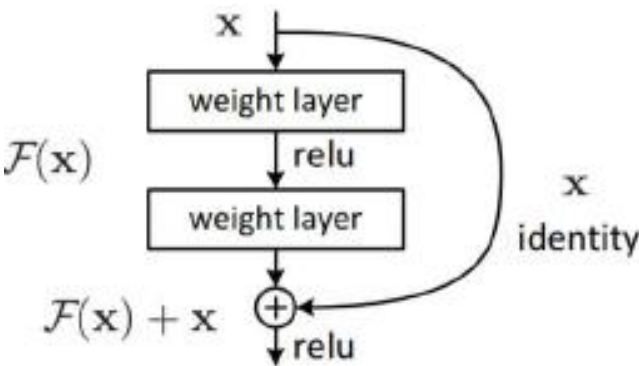
5×5 свёртку превратить в две 3×3

дальнейшая факторизация

ResNet = Residual Network (2014)

$$y = f(x) + x$$
$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} f'(x) + \frac{\partial L}{\partial y}$$

He, Zhang, Ren, Sun, CVPR 2016



ResNet = Residual Network (2014)

skip (shortcut) connections

упрощение реализации тождественной функции,
по крайней мере, через два слоя

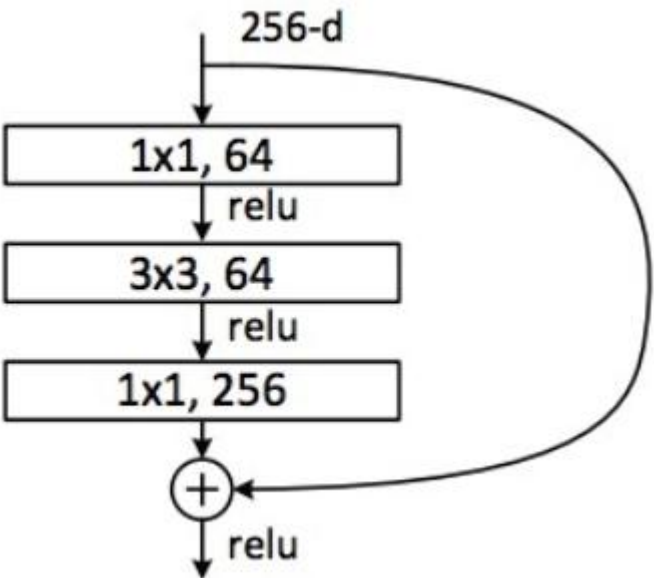
- **152 слоя**
- **связи проходят через слои**
- **Batch Normalization после каждого CONV-layer**
- **Умные инициализации весов**
 - **SGD + Momentum (0.9)**
 - **Mini-batch size = 256**
 - **Нет Dropout!**

Просто добавление слоёв не помогает!

Добавлять надо по-умному...

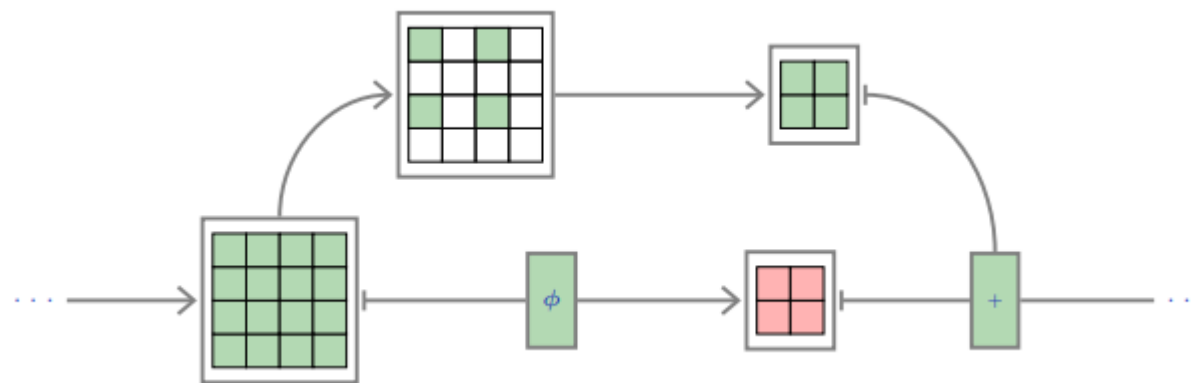
He et al. «Deep Residual Learning for Image Recognition» <https://arxiv.org/pdf/1512.03385.pdf>

ResNet = Residual Network (2014)
Deeper residual module (bottleneck)



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

Проблемы с прокидыванием в свёрточных слоях



1) размеры уменьшаются, поэтому не совсем прямая связь

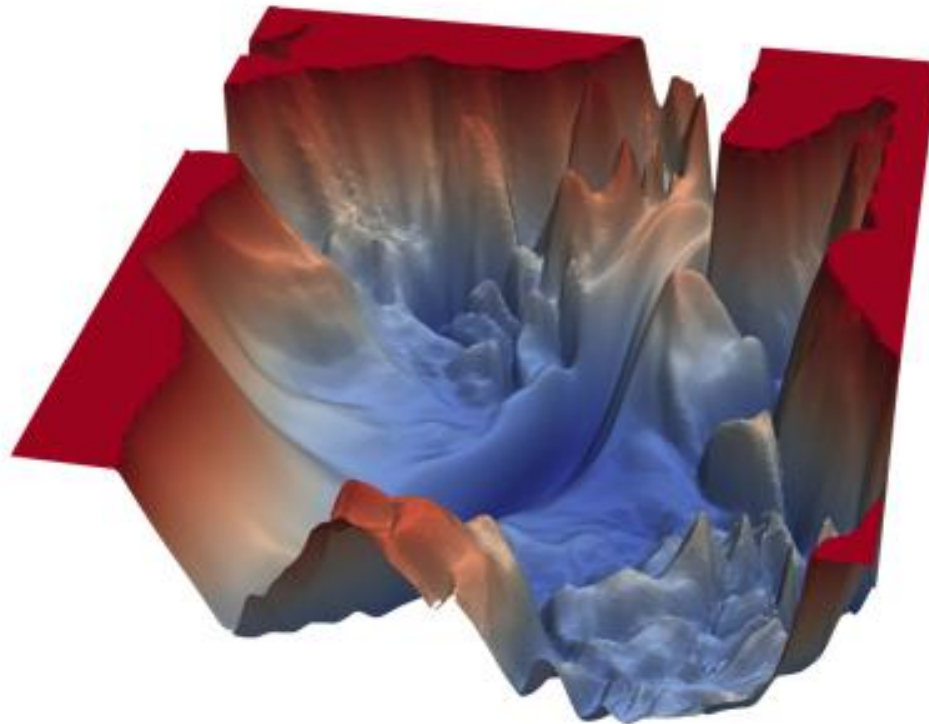
2) изменяется число каналов!

добавить нули
есть ещё способ... ;)

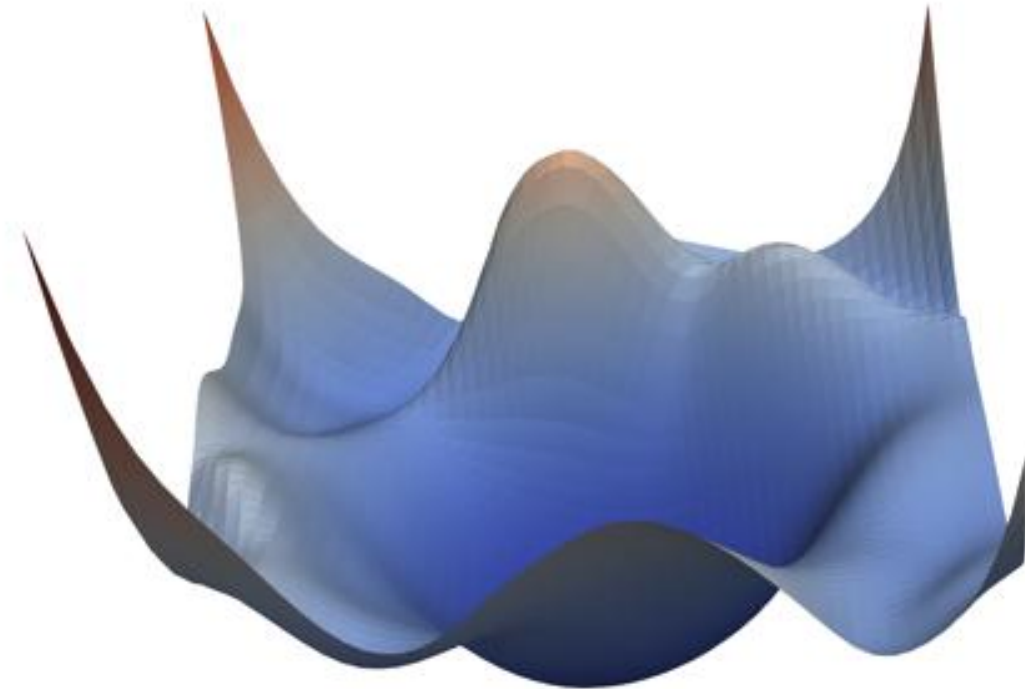
<https://fleuret.org/ee559/>

Эффект прокидывания связей

No residual connections



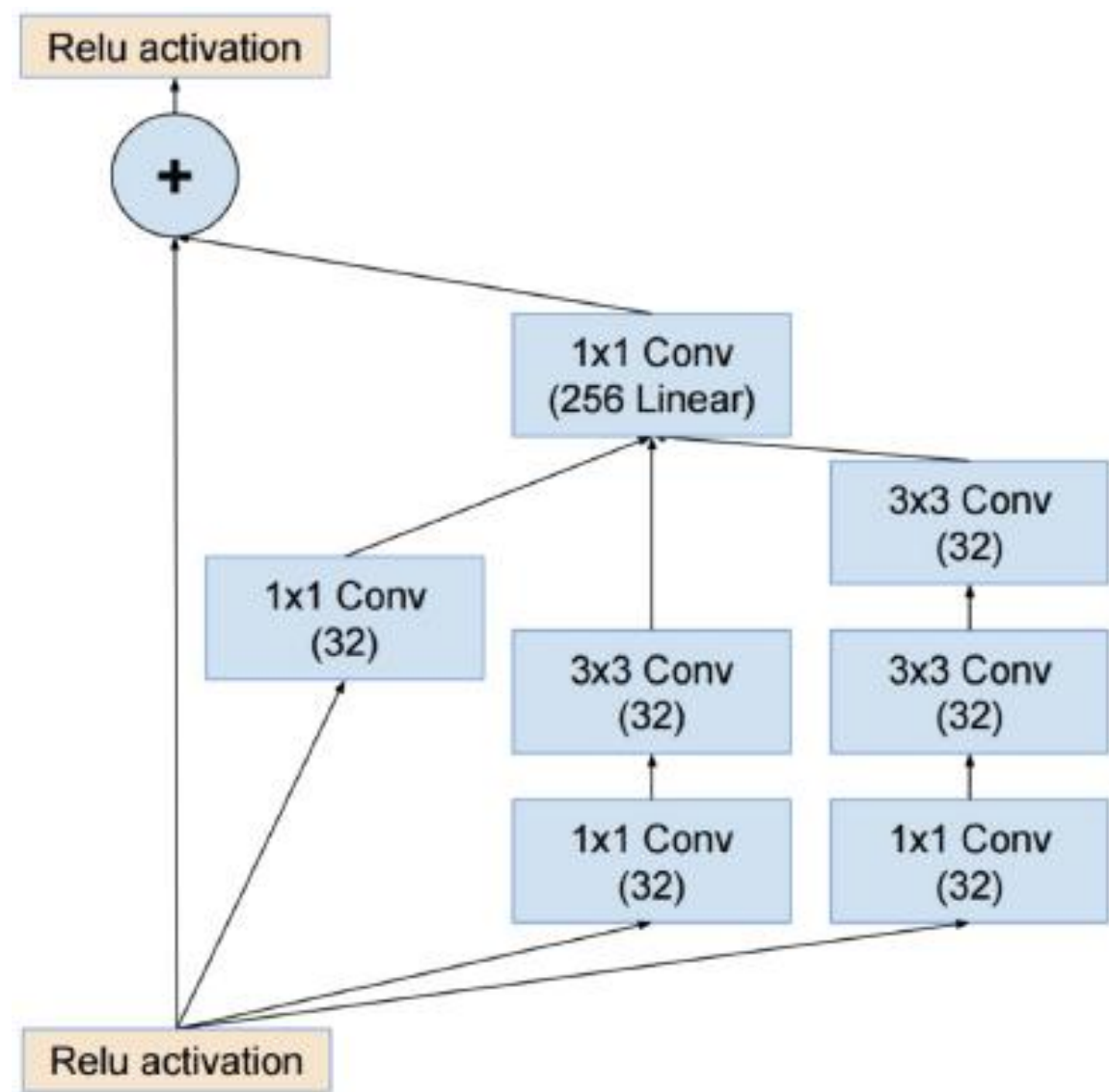
With residual connections



Same general network architecture

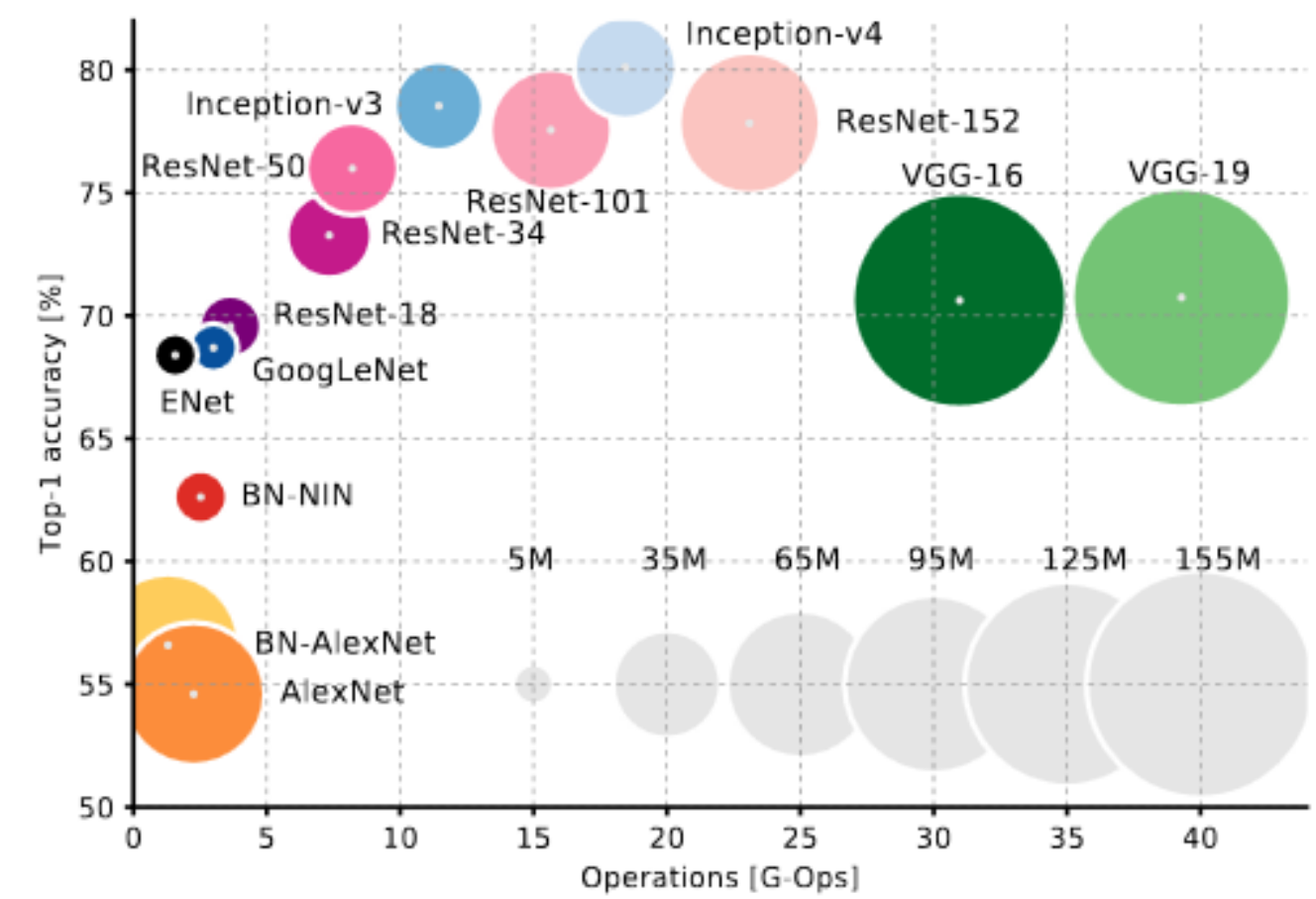
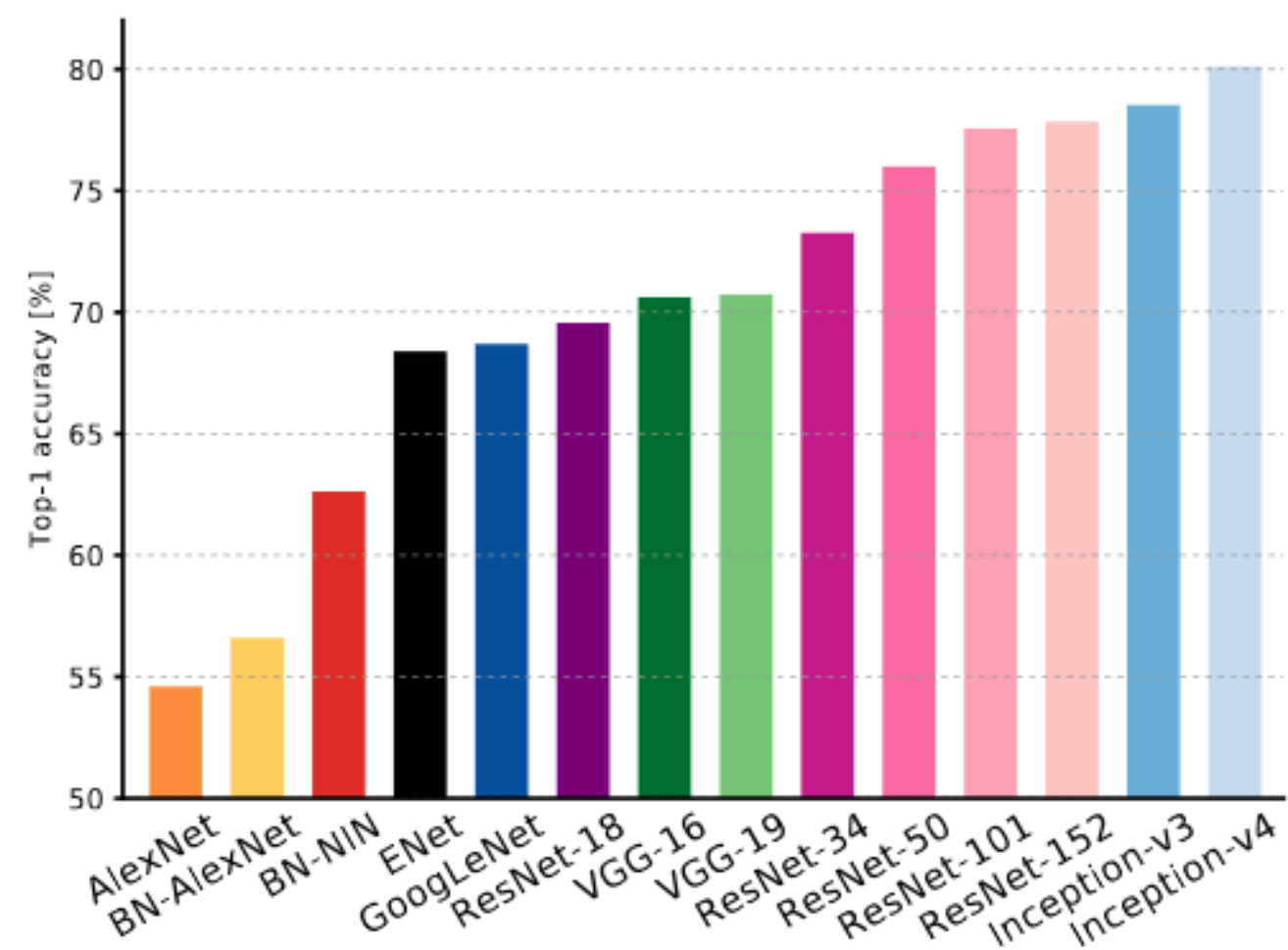
«Visualizing the Loss Landscape of Neural Nets» <https://arxiv.org/abs/1712.09913>

Inception-v4 (Inception-Resnet)



совместить две архитектуры...

Inception-v4 (Inception-Resnet)

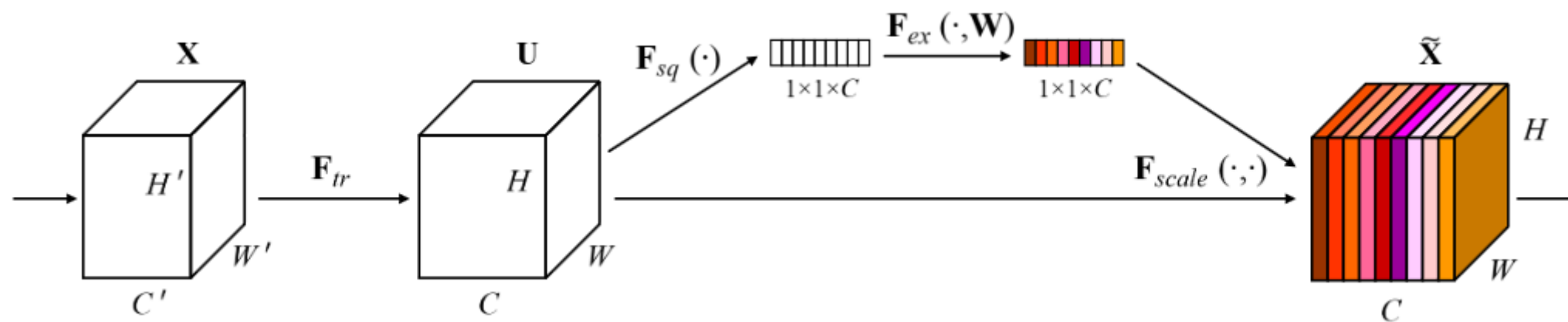


A. Canziani, A. Paszke, E. Culurciello, « An Analysis of Deep Neural Network Models for Practical Applications», 2017 <https://arxiv.org/pdf/1605.07678.pdf>

SENet (Squeeze-and-Excitation Network, 2017)

Раньше: трансформация $F_{tr} : X_{H' \times W' \times C'} \rightarrow U_{H \times W \times C}$
(например, свёртка)

Теперь: «Squeeze-and-Excitation» (SE) block $F_{tr} \oplus \dots$



сжатие (squeeze) – агрегация по каналам

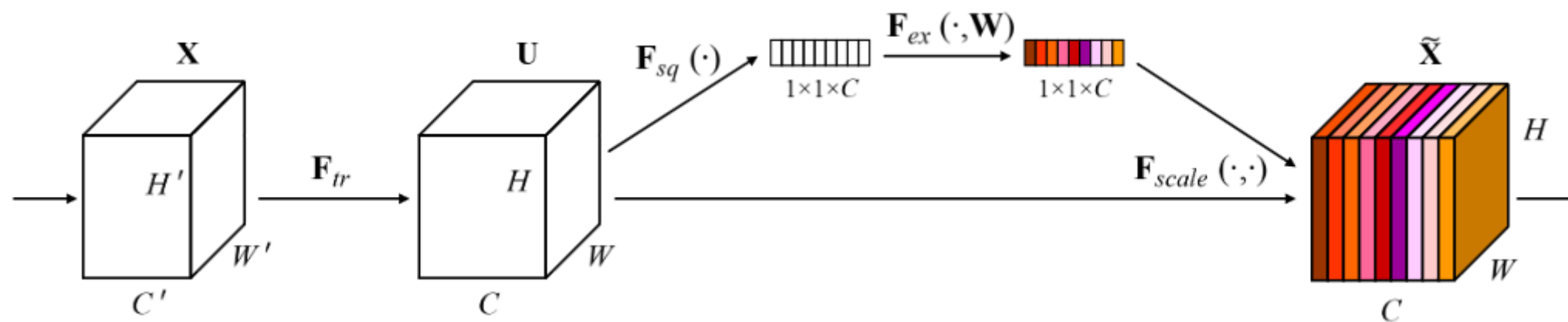
$$F_{sq} : \|u_{h,w,c}\|_{H \times W \times C} \rightarrow \left\| \frac{1}{HW} \sum_{w=1}^W \sum_{h=1}^H u_{h,w,c} \right\|_C$$

Ж. Ху и др. «Squeeze-and-Excitation Networks», 2018 <https://arxiv.org/pdf/1709.01507.pdf>

SENet (Squeeze-and-Excitation Network, 2017)

Раньше: трансформация $F_{tr} : X_{H' \times W' \times C'} \rightarrow U_{H \times W \times C}$
(например, свёртка)

Теперь: «Squeeze-and-Excitation» (SE) block $F_{tr} \oplus \dots$



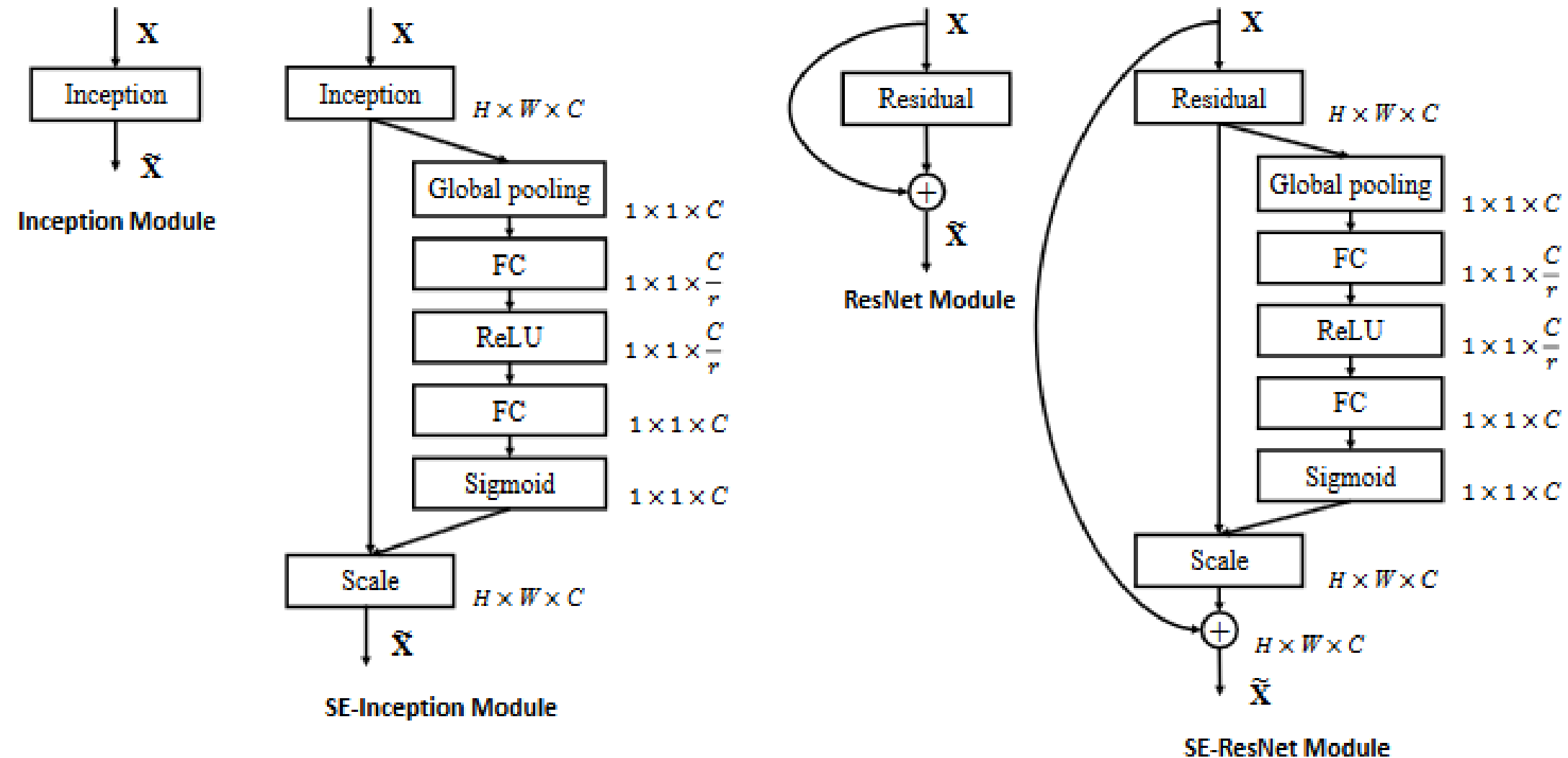
возбуждение (excitation) – адаптивная калибровка

$$F_{ex} = \sigma(W_{C \times k} \text{ReLu}(V_{k \times C} z_C))$$

$$F_{scale} : \|u_{h,w,c}\|_{H \times W \times C} \rightarrow \|u_{h,w,c} F_{ex}(z)_c\|_C$$

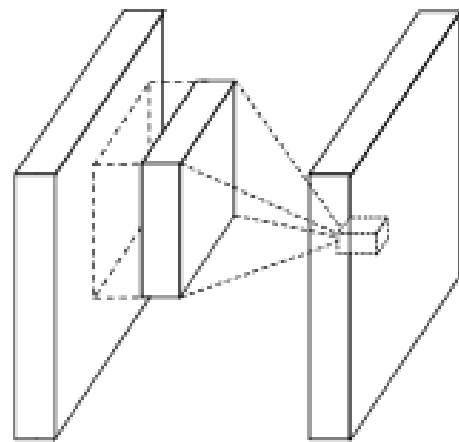
SENet (Squeeze-and-Excitation Network, 2017)

Можно переделать «старые сети»

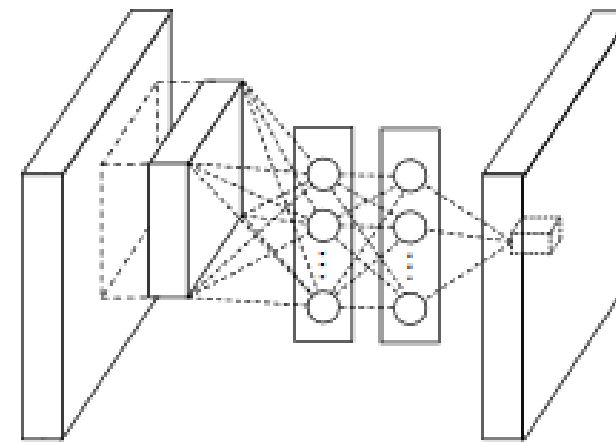


**Динамическая перекалибровка признаков позволяет
«увеличивать» важные признаки и «уменьшать» неважные**

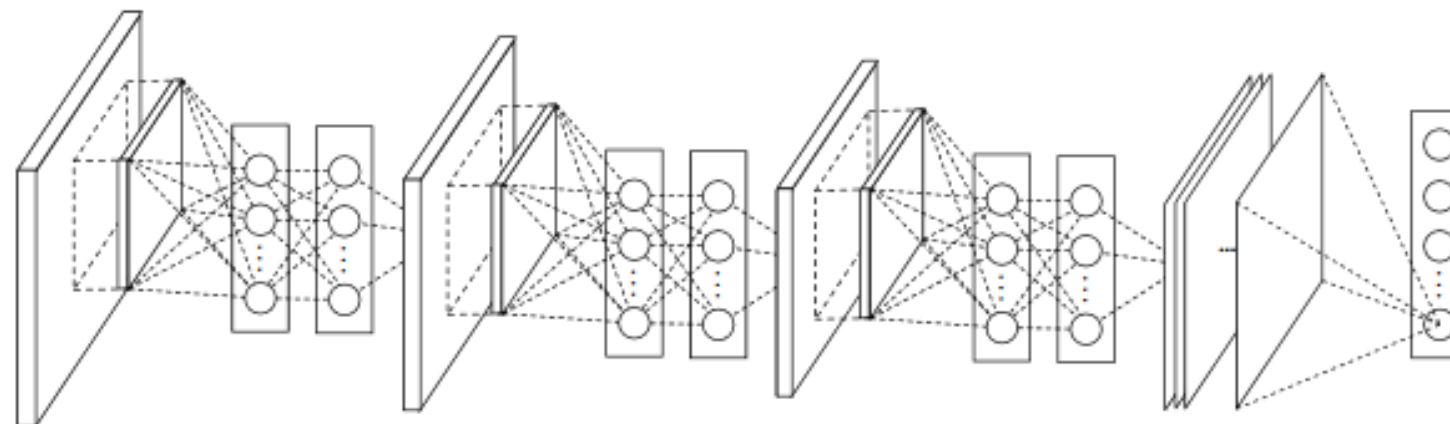
Какие архитектуры ещё надо знать... Network in Network (NiN)



(a) Linear convolution layer



(b) Mlpconv layer



полносвязность ~ свёртки 1×1 внутри свёртки
глобальный пулинг

Min Lin « Network in Network (NiN)» 2014, <https://arxiv.org/pdf/1312.4400.pdf>

Deep Networks with Stochastic Depth

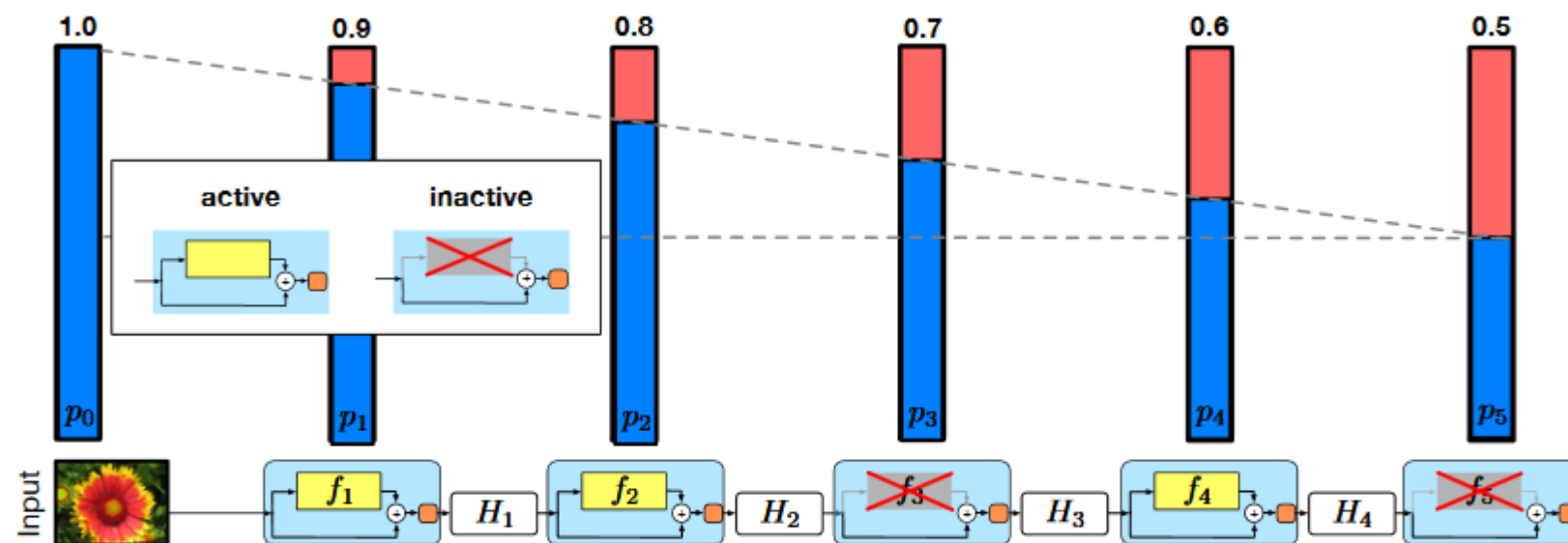


Fig. 2. The linear decay of p_ℓ illustrated on a ResNet with stochastic depth for $p_0=1$ and $p_L=0.5$. Conceptually, we treat the input to the first ResBlock as H_0 , which is always active.

- **Во время обучения: случайно удаляем подмножество слоёв**
(используем менее глубокую сеть во время обучения)
 - «Прокидывание» тождественной функции

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, Kilian Q. Weinberger «Deep Networks with Stochastic Depth» <https://arxiv.org/abs/1603.09382>

FractalNet: Ultra-Deep Neural Networks without Residuals

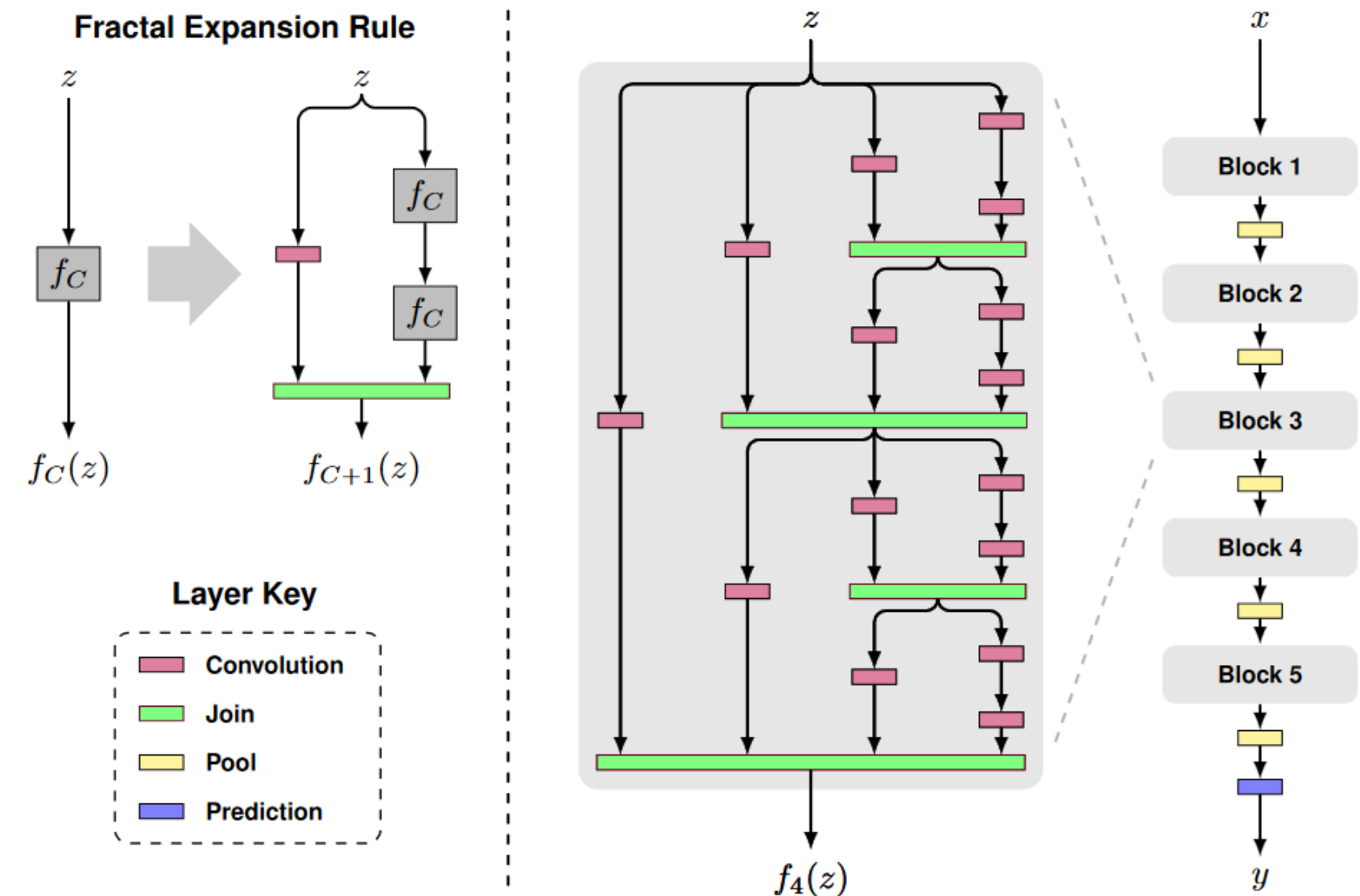


Figure 1: **Fractal architecture.** *Left:* A simple expansion rule generates a fractal architecture with C intertwined columns. The base case, $f_1(z)$, has a single layer of the chosen type (e.g. convolutional) between input and output. Join layers compute element-wise mean. *Right:* Deep convolutional networks periodically reduce spatial resolution via pooling. A fractal version uses f_C as a building block between pooling layers. Stacking B such blocks yields a network whose total depth, measured in terms of convolution layers, is $B \cdot 2^{C-1}$. This example has depth 40 ($B = 5, C = 4$).

Фрактальная архитектура с короткими и длинными связями

[Gustav Larsson 2017 <https://arxiv.org/abs/1605.07648>]

FractalNet: Ultra-Deep Neural Networks without Residuals

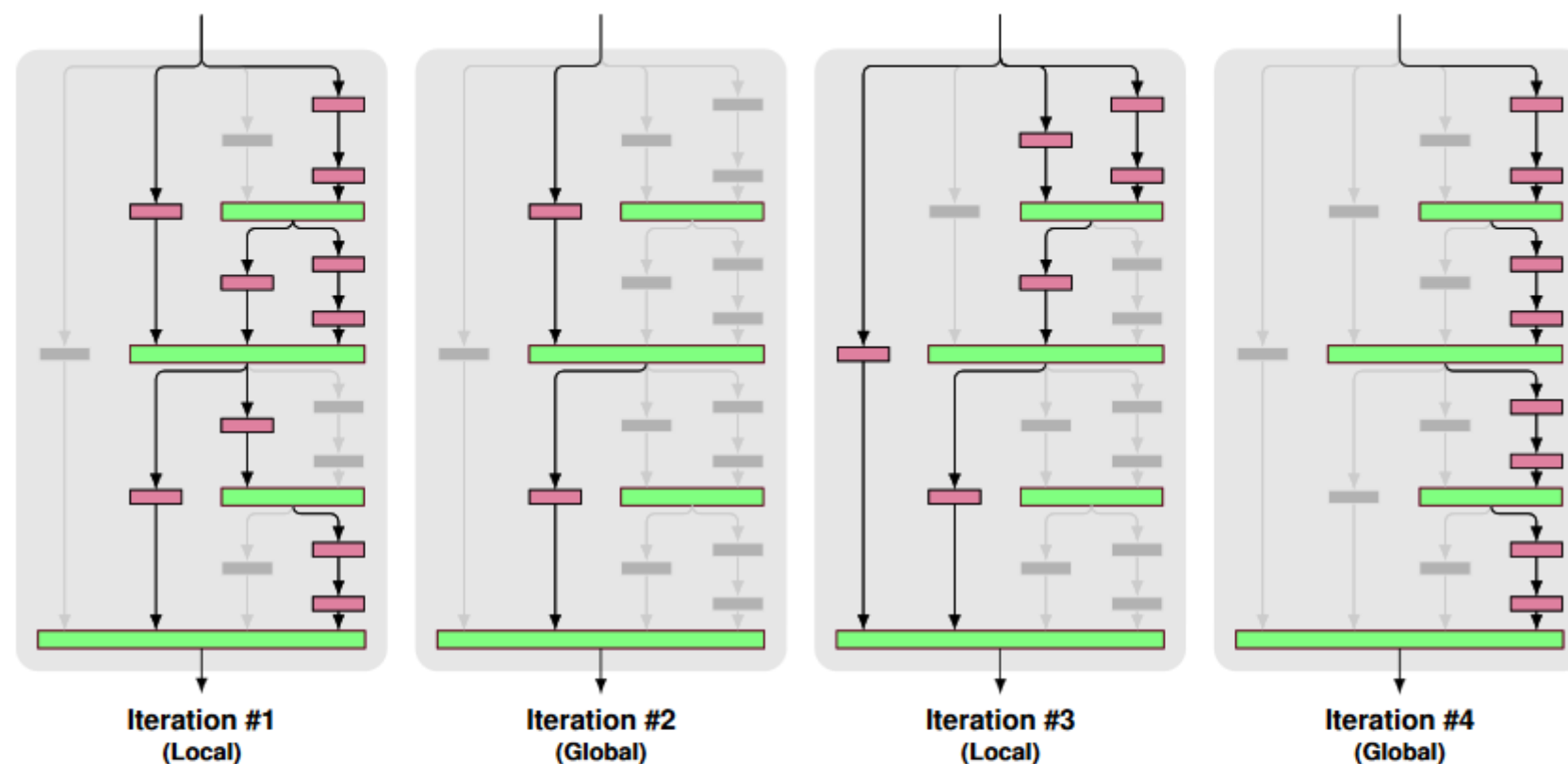


Figure 2: **Drop-path.** A fractal network block functions with some connections between layers disabled, provided some path from input to output is still available. Drop-path guarantees at least one such path, while sampling a subnetwork with many other paths disabled. During training, presenting a different active subnetwork to each mini-batch prevents co-adaptation of parallel paths. A global sampling strategy returns a single column as a subnetwork. Alternating it with local sampling encourages the development of individual columns as performant stand-alone subnetworks.

Обучение со случайным выбрасыванием связей

Fractal of FractalNet (FoF)

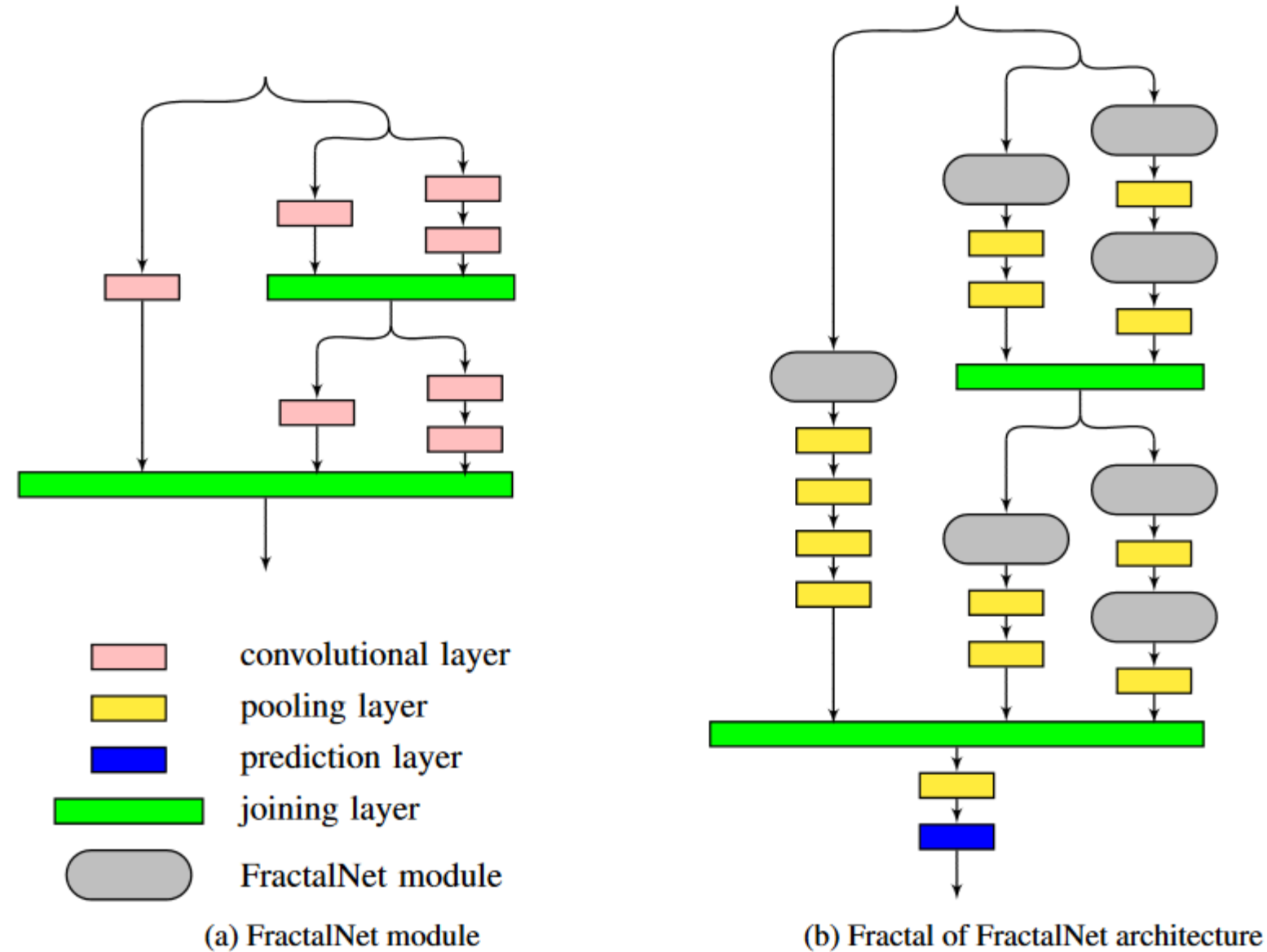


Figure 1: (a) The FractalNet module and (b) the FoF architecture.

Leslie N. Smith, Nicholay Topin, Deep Convolutional Neural Network Design Patterns // <https://arxiv.org/abs/1611.00847>

Densely Connected Convolutional Networks (DenseNets)

Блоки, в которых слой соединён с каждым последующим

Обычная сеть: $z_i = H_i(z_{i-1})$

где z_i выход i -го слоя.

ResNet: $z_i = H_i(z_{i-1}) + z_{i-1}$

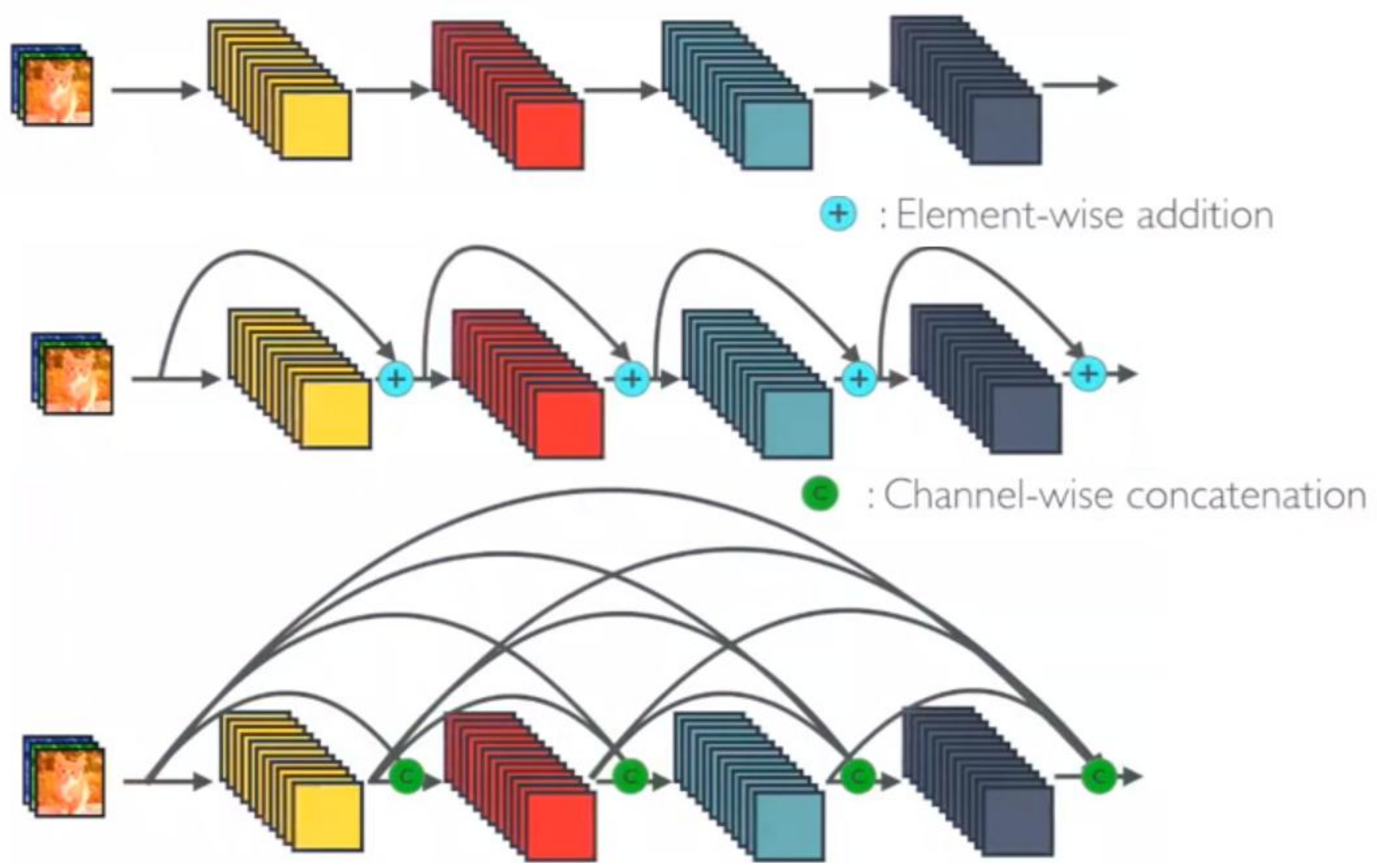
DensNet: $z_i = H_i([z_{i-1}, z_{i-2}, \dots, z_0])$

H = BN + ReLU + convolution + dropout

число признаков линейно вырастает...

Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger «Densely Connected Convolutional Networks» <https://arxiv.org/abs/1608.06993>

Nets → ResNets → DenseNets



Densely Connected Convolutional Networks (DenseNets)

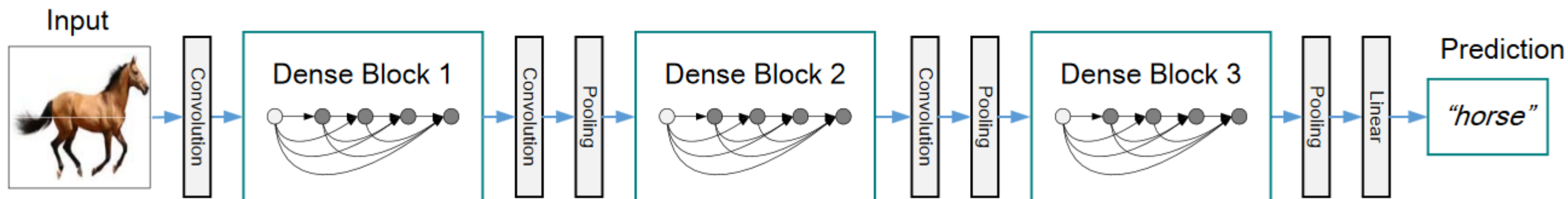


Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

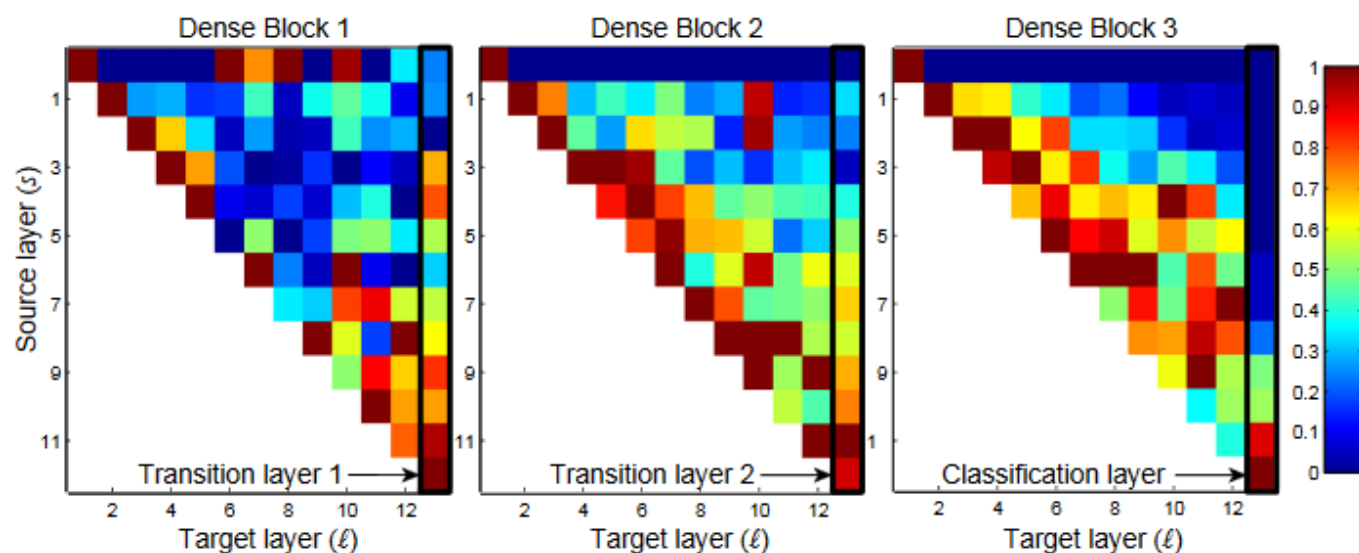


Figure 5: The average absolute filter weights of convolutional layers in a trained DenseNet. The color of pixel (s, ℓ) encodes the average $L1$ norm (normalized by number of input feature-maps) of the weights connecting convolutional layer s to ℓ within a dense block. Three columns highlighted by black rectangles correspond to two transition layers and the classification layer. The first row encodes weights connected to the input layer of the dense block.

ResNeXt

такое же число параметров как в ResNet, но разнести их по cardinality=32 разным путям
тут блок – conv + ИТ + ReLU
используется bottleneck!!!

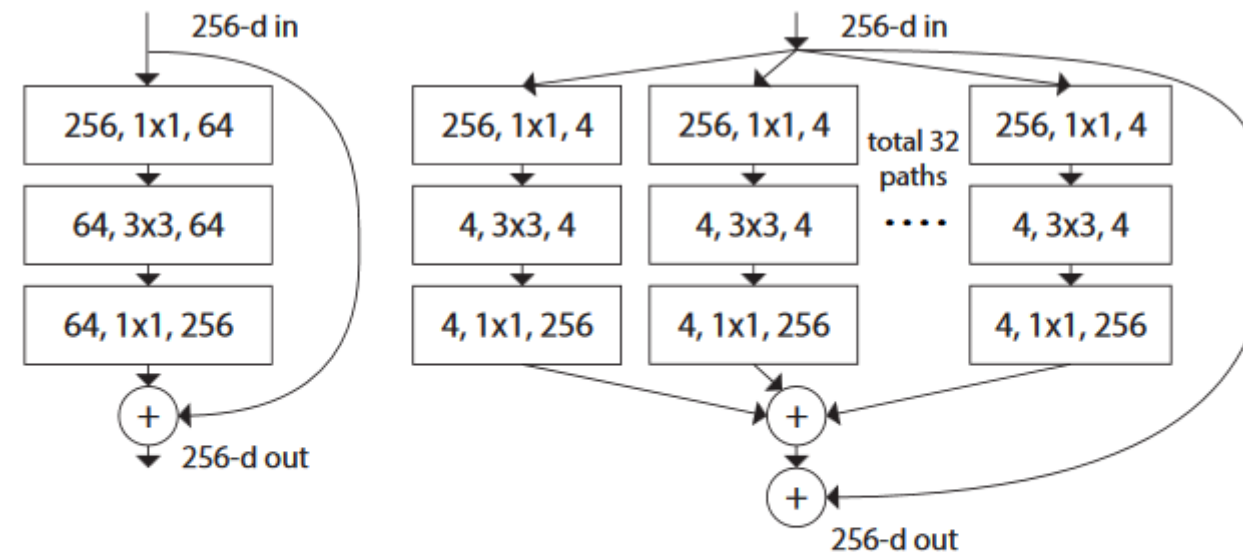


Figure 1. **Left:** A block of ResNet [14]. **Right:** A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He

Aggregated Residual Transformations for Deep Neural Networks // <https://arxiv.org/abs/1611.05431>

ResNeXt

Похожие блоки: (a) ResNeXt Block, (b) Inception-ResNet Block, (c) Grouped Convolution

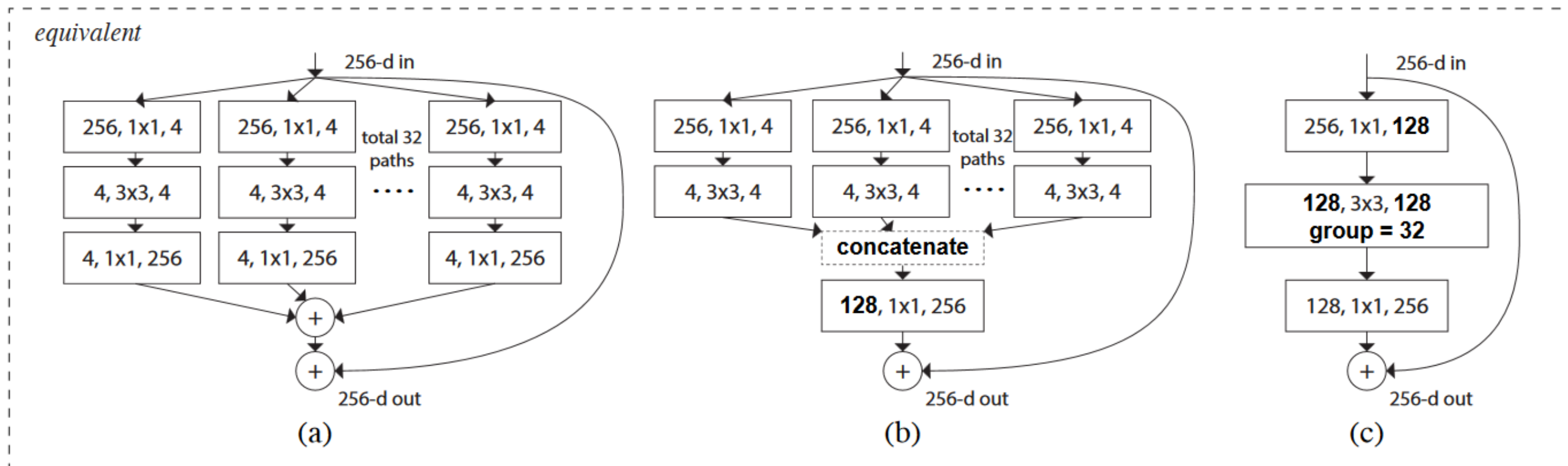


Figure 3. Equivalent building blocks of ResNeXt. (a): Aggregated residual transformations, the same as Fig. 1 right. (b): A block equivalent to (a), implemented as early concatenation. (c): A block equivalent to (a,b), implemented as grouped convolutions [24]. Notations in **bold** text highlight the reformulation changes. A layer is denoted as (# input channels, filter size, # output channels).

Ещё разделения на ветви: ResNeXt, MultiResNet, PolyNet

Стратегия «split-transform-aggregate»

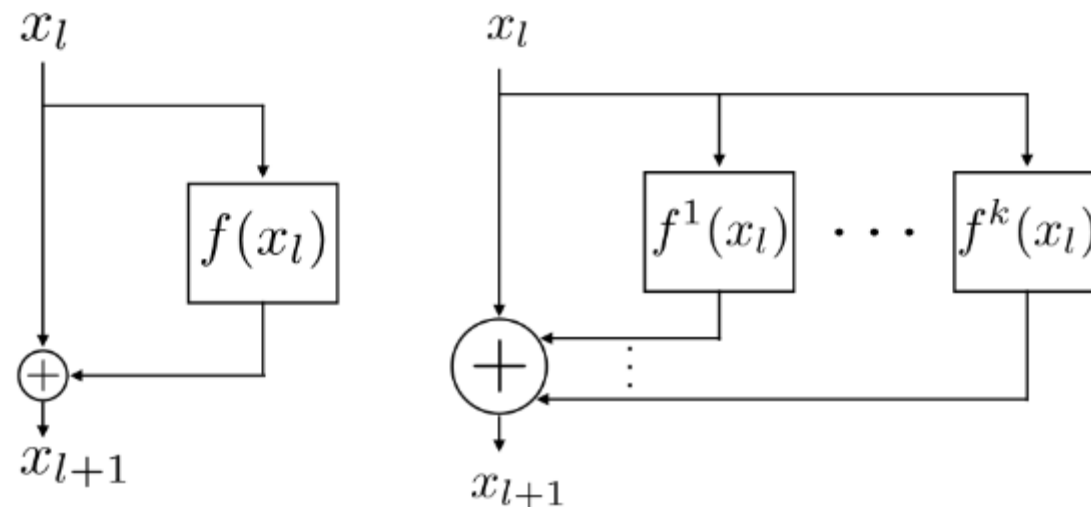


Fig. 2: A residual block (left) versus a multi-residual block (right).

Masoud Abdi, Saeid Nahavandi. Multi-Residual Networks: Improving the Speed and Accuracy of Residual Networks // <https://arxiv.org/abs/1609.05672>

Ещё разделения на ветви: ResNeXt, MultiResNet, PolyNet

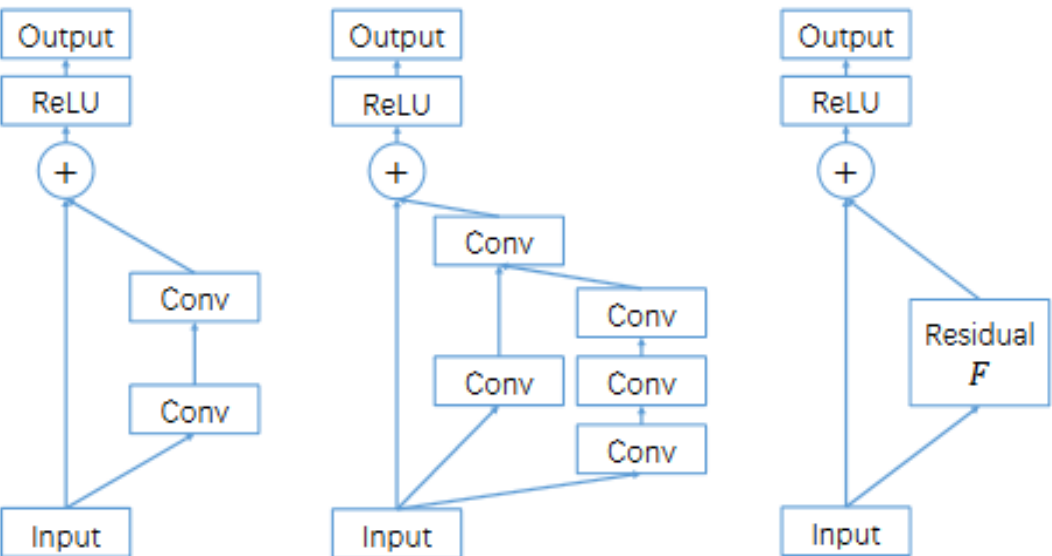
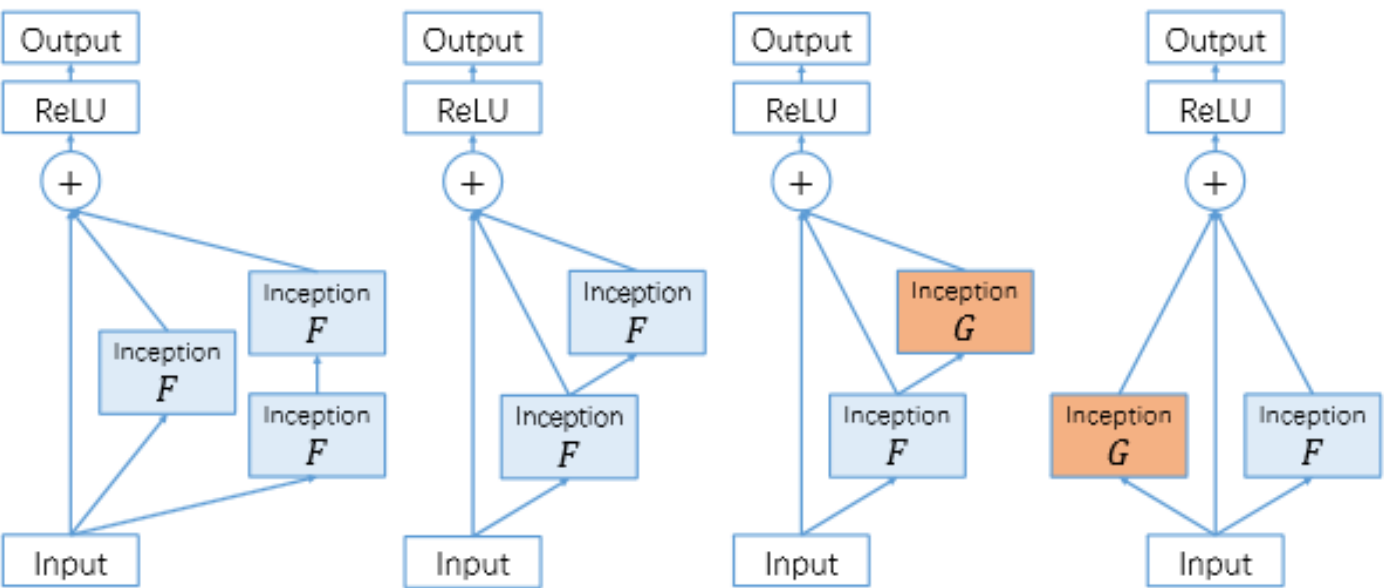


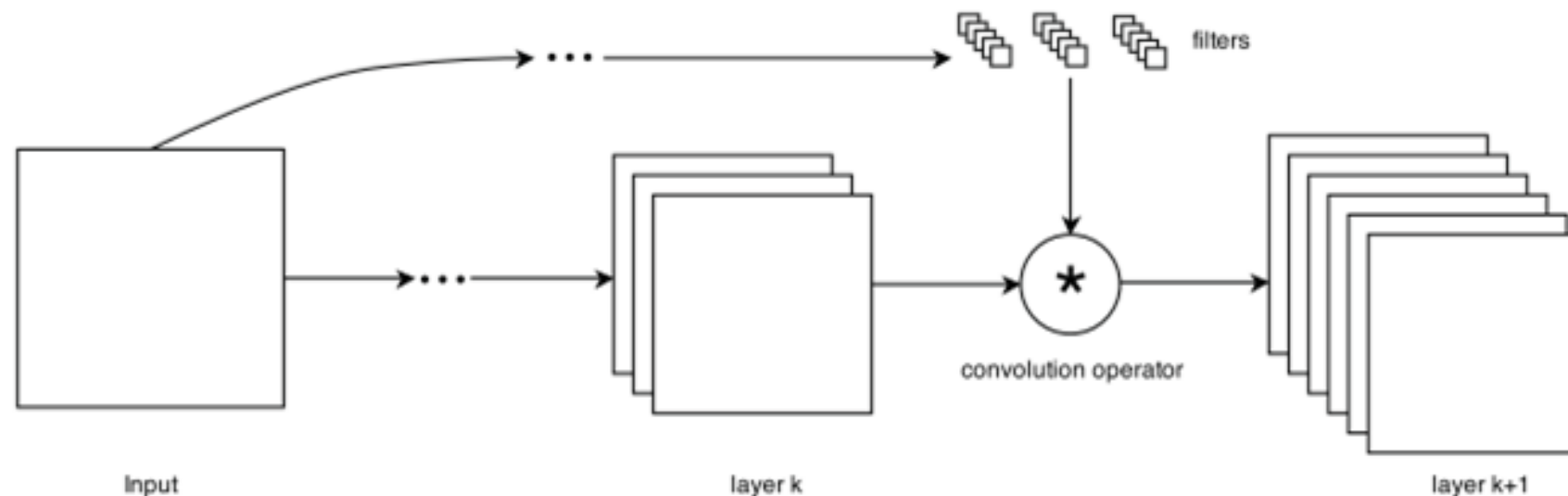
Figure 3: **Left:** residual unit of ResNet [9]. **Middle:** type-B Inception residual unit of Inception-ResNet-v2 [25]. **Right:** abstract residual unit structure where the residual block is denoted by F .



(a) *poly-2* (b) *poly-2* (c) *mpoly-2* (d) *2-way*

Figure 4: Examples of PolyInception structures.

Xingcheng Zhang, Zhizhong Li, Chen Change Loy, Dahua Lin. PolyNet: A Pursuit of Structural Diversity in Very Deep Networks // <https://arxiv.org/abs/1611.05725>

HyperNets термин

**динамическая свёртка – результат действия мини-сети «Dynamic Convolutional Layer»
потом эта идея – SENet, Attention**

Klein, B., Wolf, L., & Afek, Y. (2015). A dynamic convolutional layer for short range weather prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(pp. 4840-4848)

свёртка может зависеть от позиции «Dynamic local filtering»

потом: Jia, X., De Brabandere, B., Tuytelaars, T., & Gool, L. V. (2016). Dynamic filter networks. In Advances in Neural Information Processing Systems(pp. 667-675)

для RNN

Ha, D., Dai, A., and Le, Q. V. (2016). Hypernetworks. arXiv preprint arXiv:1609.09106

MobileNet

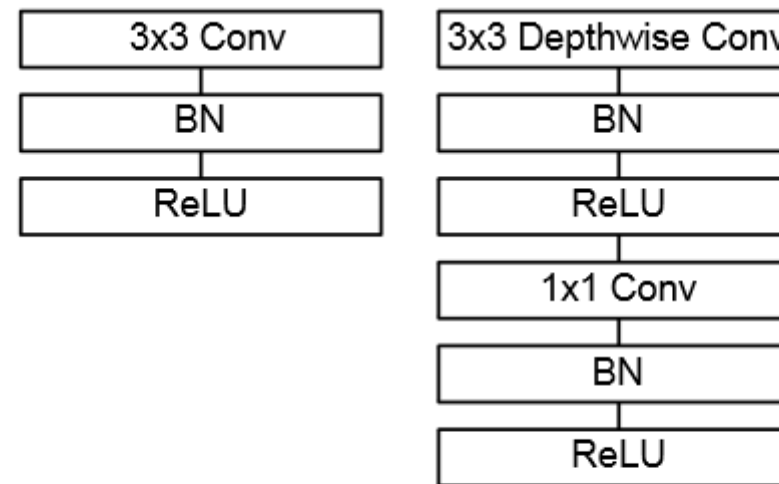


Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

использование Depthwise separable convolution (уже было)

MobileNetv2 – потом вернёмся

Andrew G. Howard et. al. «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications» <https://arxiv.org/pdf/1704.04861.pdf>

EfficientNet: Масштабирование (scaling up) моделей

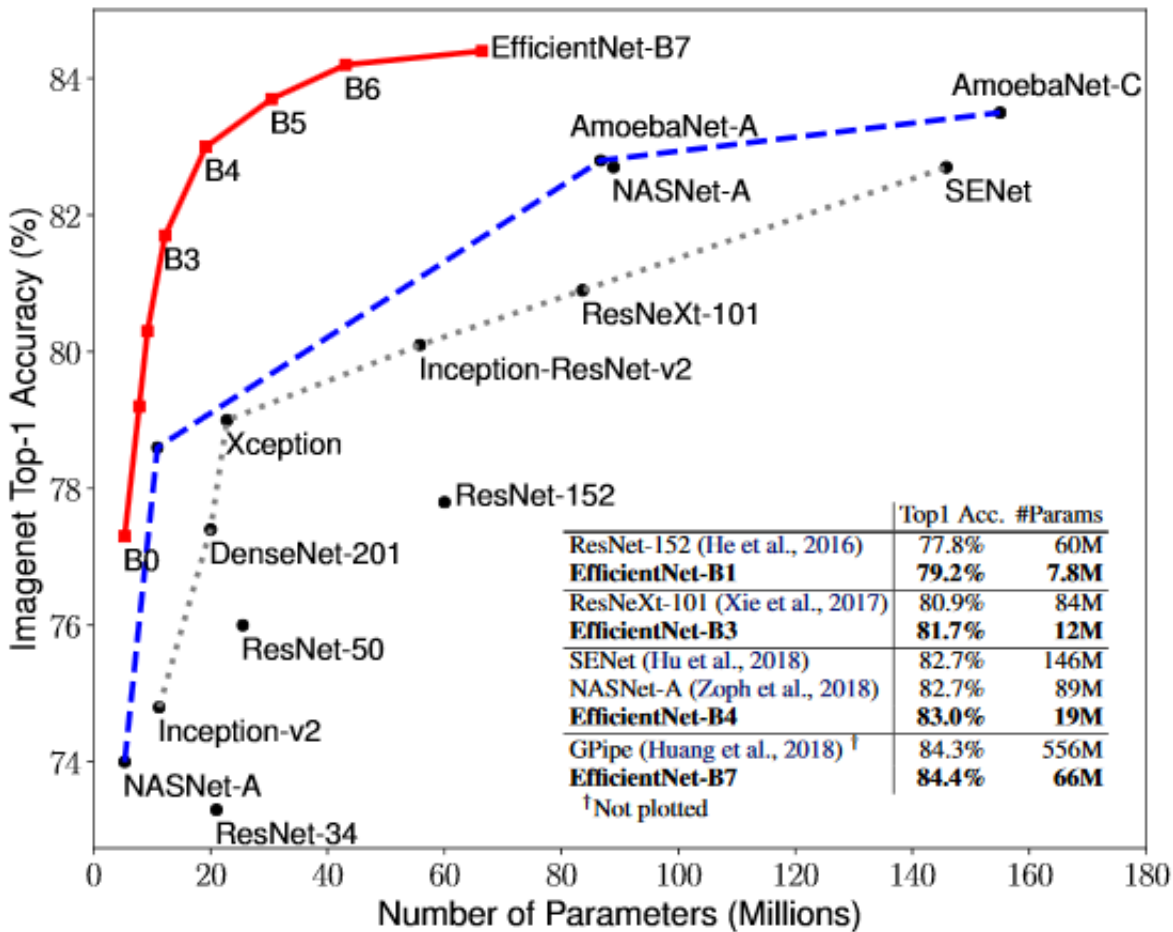


Figure 1. Model Size vs. ImageNet Accuracy. All numbers are for single-crop, single-model. Our EfficientNets significantly outperform other ConvNets. In particular, EfficientNet-B7 achieves new state-of-the-art 84.4% top-1 accuracy but being 8.4x smaller and 6.1x faster than GPipe. EfficientNet-B1 is 7.6x smaller and 5.7x faster than ResNet-152. Details are in Table 2 and 4.

<https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet>

Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML 2019. <https://arxiv.org/abs/1905.11946>

EfficientNet: Масштабирование (scaling up) моделей

- увеличение глубины
- увеличение числа каналов
- увеличение разрешения
- **compound scaling method** (увеличение всего с опред. пропорциями)

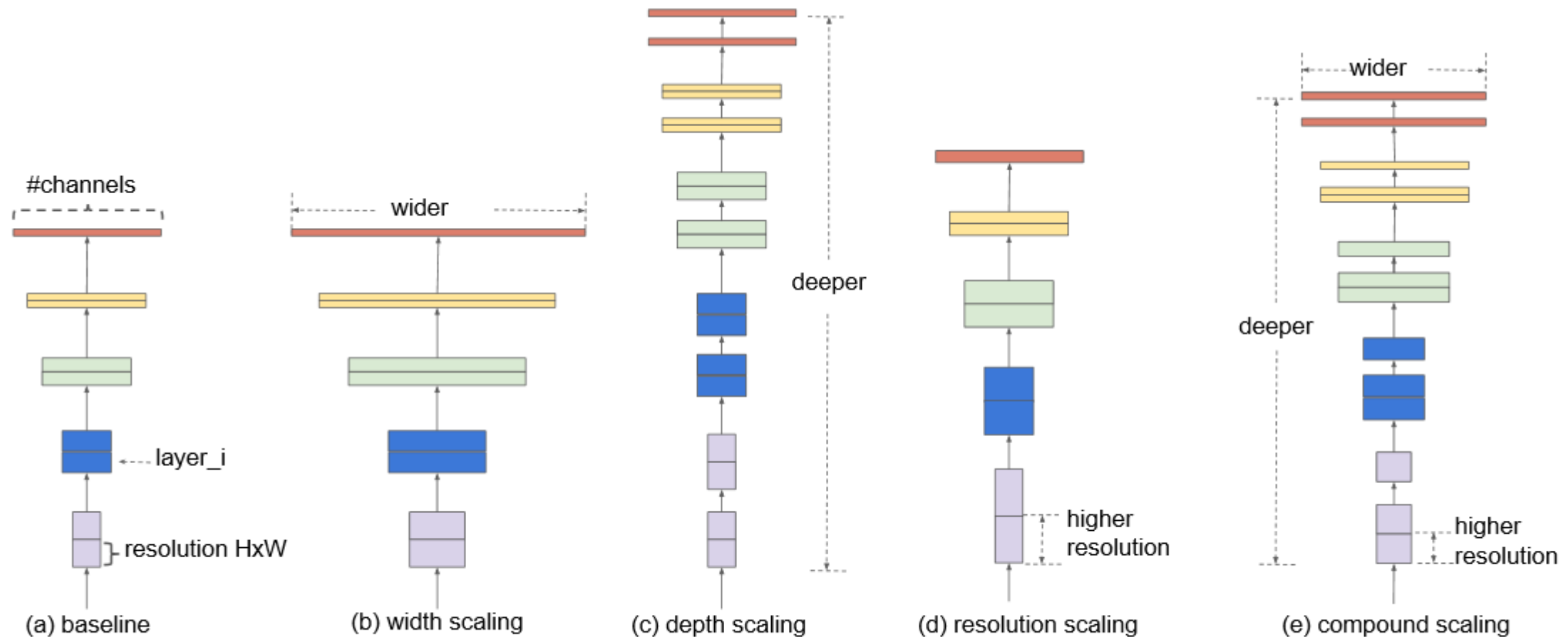


Figure 2. Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

EfficientNet: Масштабирование (scaling up) моделей

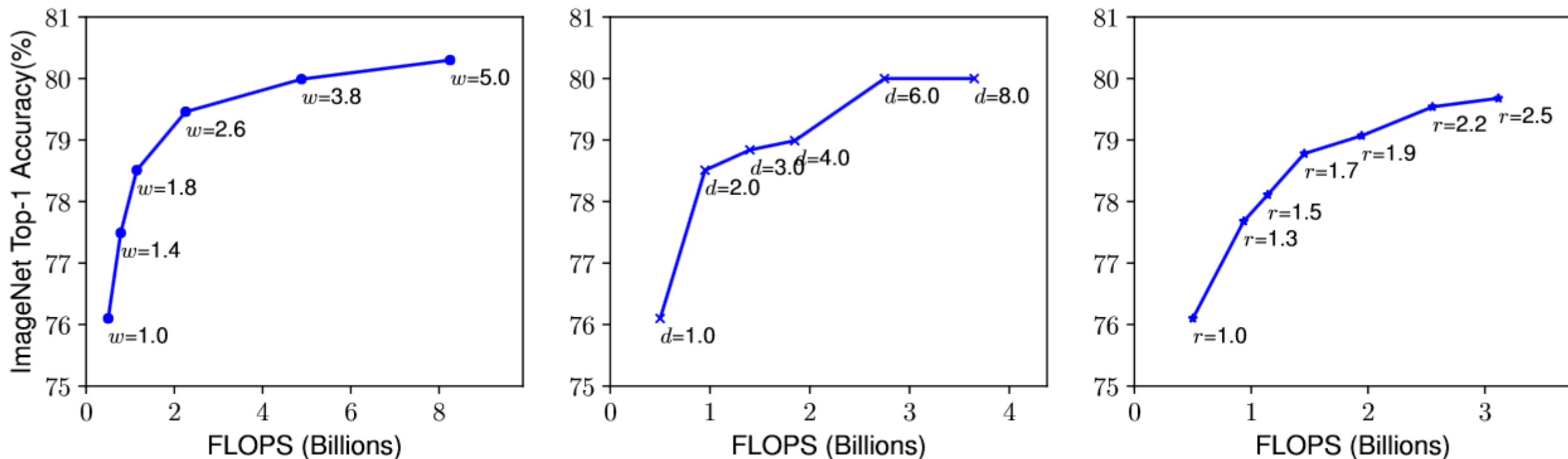


Figure 3. Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients. Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single dimension scaling. Baseline network is described in Table 1.

**разные масштабирования не являются независимыми
нужно учитывать, что меняется и время обучения сети!**

EfficientNet: compound scaling method

depth: $d = \alpha^\phi$

width: $w = \beta^\phi$

resolution: $r = \gamma^\phi$

s.t. $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$

$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$

ограничение =2 для ограничения числа операций

Table 3. Scaling Up MobileNets and ResNet.

Model	FLOPS	Top-1 Acc.
Baseline MobileNetV1 (Howard et al., 2017)	0.6B	70.6%
Scale MobileNetV1 by width ($w=2$)	2.2B	74.2%
Scale MobileNetV1 by resolution ($r=2$)	2.2B	72.7%
compound scale ($d=1.4, w=1.2, r=1.3$)	2.3B	75.6%
Baseline MobileNetV2 (Sandler et al., 2018)	0.3B	72.0%
Scale MobileNetV2 by depth ($d=4$)	1.2B	76.8%
Scale MobileNetV2 by width ($w=2$)	1.1B	76.4%
Scale MobileNetV2 by resolution ($r=2$)	1.2B	74.8%
MobileNetV2 compound scale	1.3B	77.4%
Baseline ResNet-50 (He et al., 2016)	4.1B	76.0%
Scale ResNet-50 by depth ($d=4$)	16.2B	78.1%
Scale ResNet-50 by width ($w=2$)	14.7B	77.7%
Scale ResNet-50 by resolution ($r=2$)	16.4B	77.5%
ResNet-50 compound scale	16.7B	78.8%

EfficientNet: compound scaling method

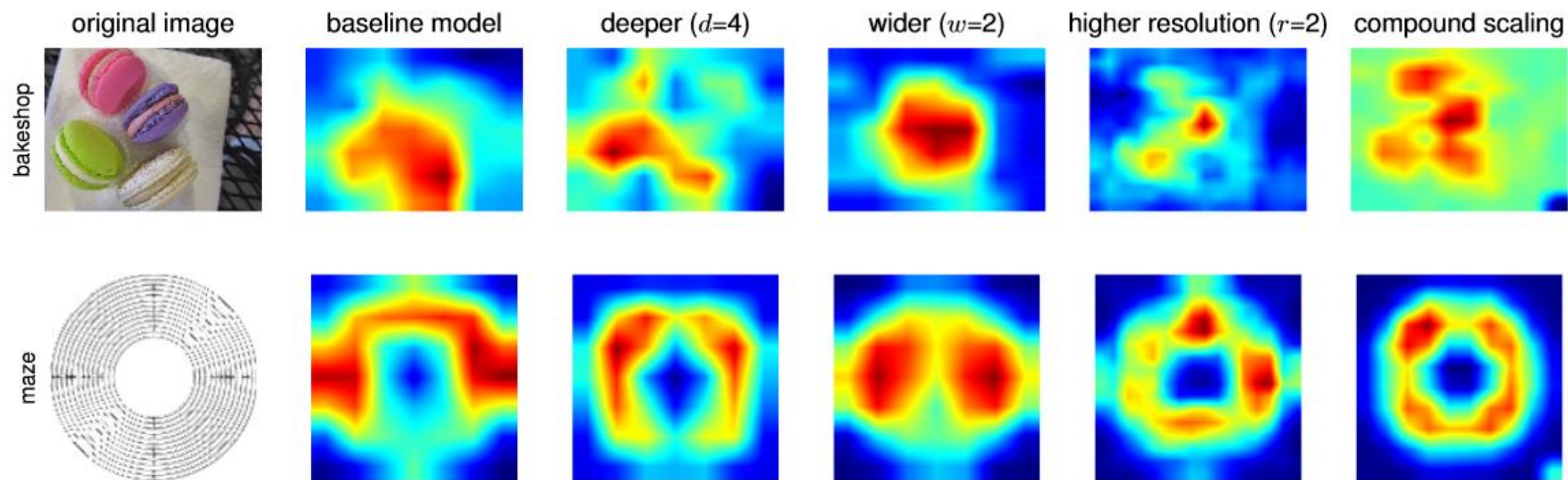


Figure 7. Class Activation Map (CAM) (Zhou et al., 2016) for Models with different scaling methods- Our compound scaling method allows the scaled model (last column) to focus on more relevant regions with more object details. Model details are in Table 7.

SqueezeNet – маленькая сеть с качеством AlexNet

Table 2: Comparing SqueezeNet to model compression approaches. By *model size*, we mean the number of bytes required to store all of the parameters in the trained model.

CNN architecture	Compression Approach	Data Type	Original → Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
AlexNet	None (baseline)	32 bit	240MB	1x	57.2%	80.3%
AlexNet	SVD (Denton et al., 2014)	32 bit	240MB → 48MB	5x	56.0%	79.4%
AlexNet	Network Pruning (Han et al., 2015b)	32 bit	240MB → 27MB	9x	57.2%	80.3%
AlexNet	Deep Compression (Han et al., 2015a)	5-8 bit	240MB → 6.9MB	35x	57.2%	80.3%
SqueezeNet (ours)	None	32 bit	4.8MB	50x	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	8 bit	4.8MB → 0.66MB	363x	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	6 bit	4.8MB → 0.47MB	510x	57.5%	80.3%

в 3 раза быстрее

Fully Convolutional Network (FCN) – нет полносвязных слоёв

Forrest N. Iandola « SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size»2017 <https://arxiv.org/abs/1602.07360>

SqueezeNet – маленькая сеть с качеством AlexNet

замена 3×3 -свёрток на 1×1 – squeeze – см. дальше
уменьшаем в 9 раз число параметров

уменьшаем число каналов перед 3×3 -свёртками
опять же – меньше параметров

делать побольше stride ближе к концу сети
есть гипотеза, что повышает качество

SqueezeNet – маленькая сеть с качеством AlexNet

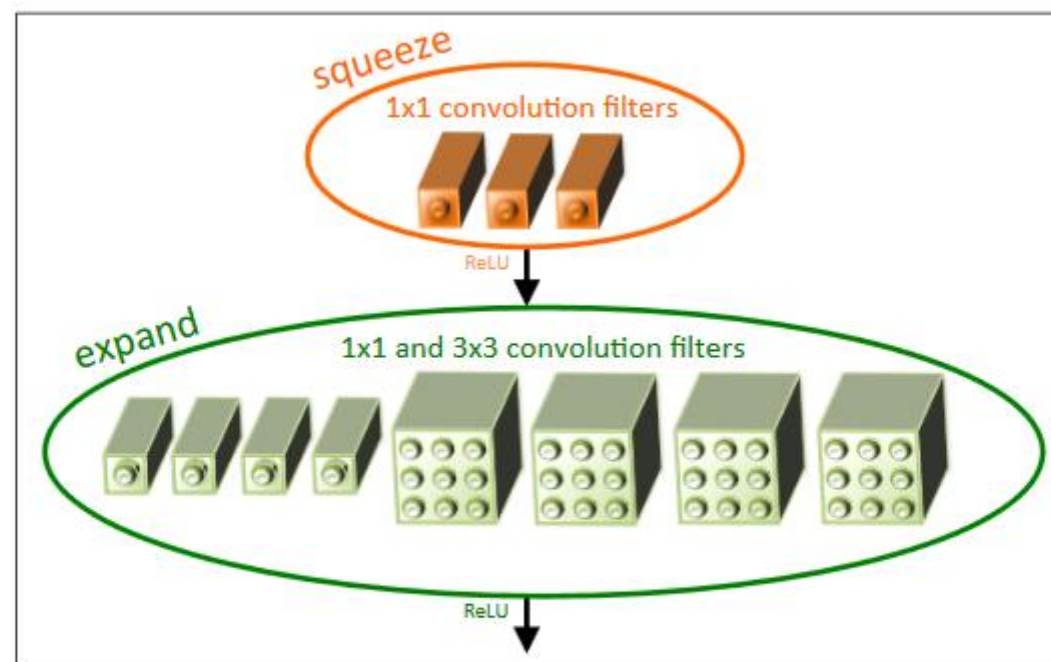


Figure 1: Microarchitectural view: Organization of convolution filters in the **Fire module**. In this example, $s_{1 \times 1} = 3$, $e_{1 \times 1} = 4$, and $e_{3 \times 3} = 4$. We illustrate the convolution filters but not the activations.

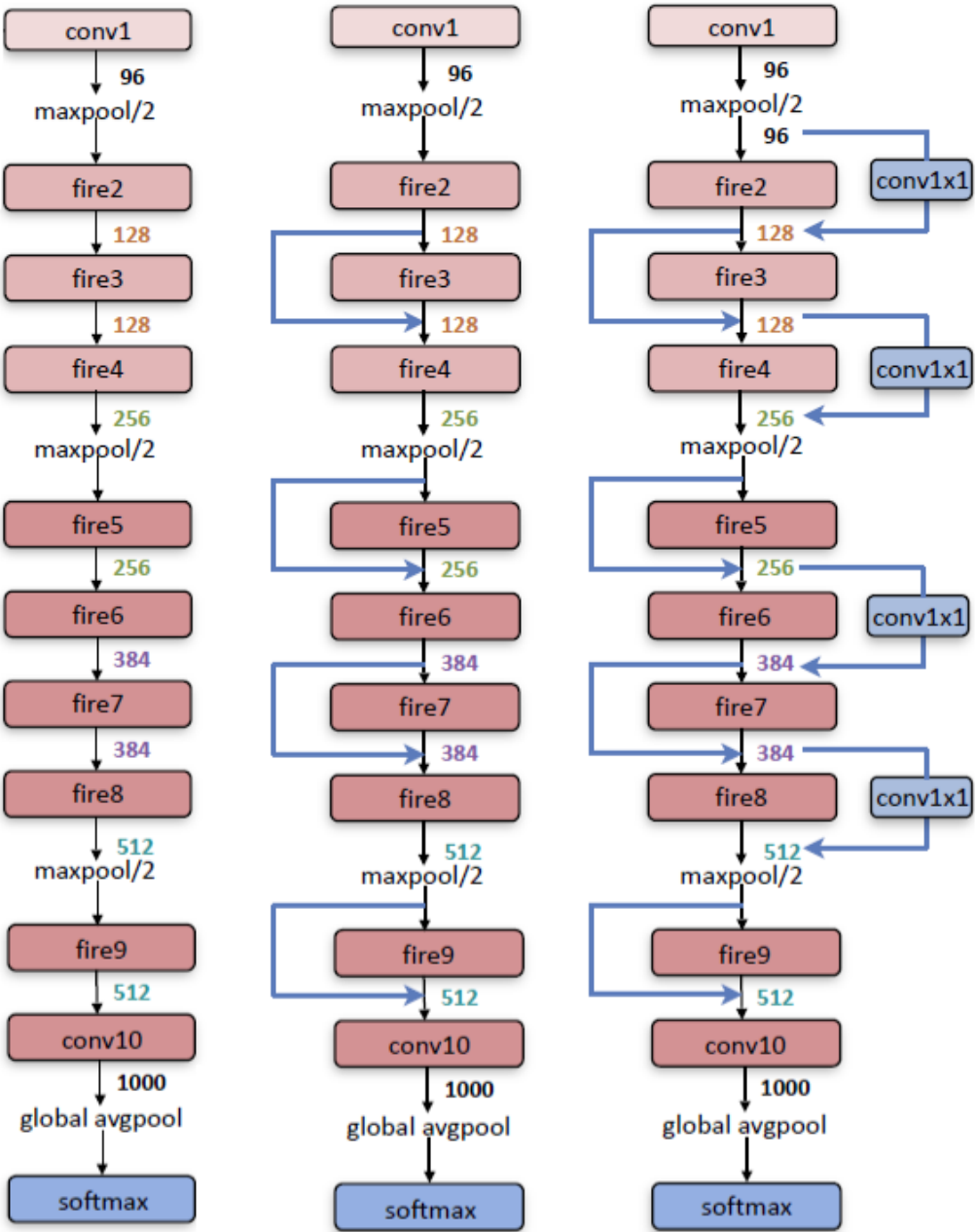
«Fire-модуль», его параметры (здесь):

$s_{1 \times 1} = 3$ – число 1×1 -свёрток в сжимающей части

$e_{1 \times 1} = 4$ – число 1×1 -свёрток в расширяющей части

$e_{3 \times 3} = 4$ – число 3×3 -свёрток в расширяющей части

SqueezeNet – три варианта – SN, SN + simple bypass, SN + complex bypass



Architecture	Top-1 Accuracy	Top-5 Accuracy	Model Size
Vanilla SqueezeNet	57.5%	80.3%	4.8MB
SqueezeNet + Simple Bypass	60.4%	82.5%	4.8MB
SqueezeNet + Complex Bypass	58.8%	82.0%	7.7MB

max-pooling stride = 2
после conv1, fire4, fire8, conv10

ShuffleNet – перемешивание каналов для мобильных устройств (низкая вычислительная мощность) идея перемешивание каналов (в групповых свёртках)

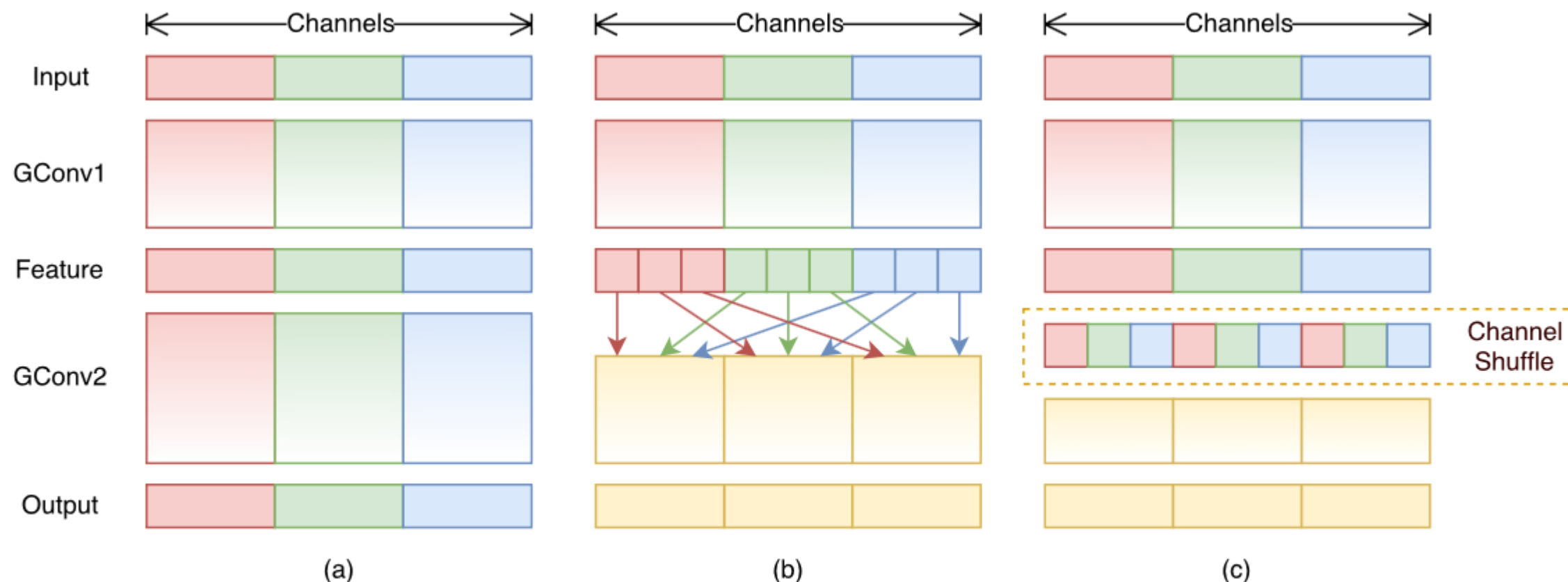


Figure 1. Channel shuffle with two stacked group convolutions. GConv stands for group convolution. a) two stacked convolution layers with the same number of groups. Each output channel only relates to the input channels within the group. No cross talk; b) input and output channels are fully related when GConv2 takes data from different groups after GConv1; c) an equivalent implementation to b) using channel shuffle.

Xiangyu Zhang «ShuffleNet: An Extremely Efficient Convolutional Neural Network for MobileDevices»

<https://arxiv.org/pdf/1707.01083.pdf>

ShuffleNet – перемешивание каналов

группы нужны, чтобы свёртка работала только на каналах группы (быстрее)

Complexity (MFLOPs)	VGG-like	ResNet	Xception-like	ResNeXt	ShuffleNet (ours)
140	50.7	37.3	33.6	33.3	32.4 (1×, $g = 8$)
38	-	48.8	45.1	46.0	41.6 (0.5×, $g = 4$)
13	-	63.7	57.1	65.2	52.7 (0.25×, $g = 8$)

Table 4. Classification error vs. various structures (% , smaller number represents better performance). We do not report VGG-like structure on smaller networks because the accuracy is significantly worse.

чтобы не было такого, что зависимость локальна в группах
каналы «перемешивают»

принцип:

`resize([A, A, B, B], [2, 2]).T.flatten = [A, B, A, B]`

основной блок ~ ResNeXt

ShuffleNet – перемешивание каналов

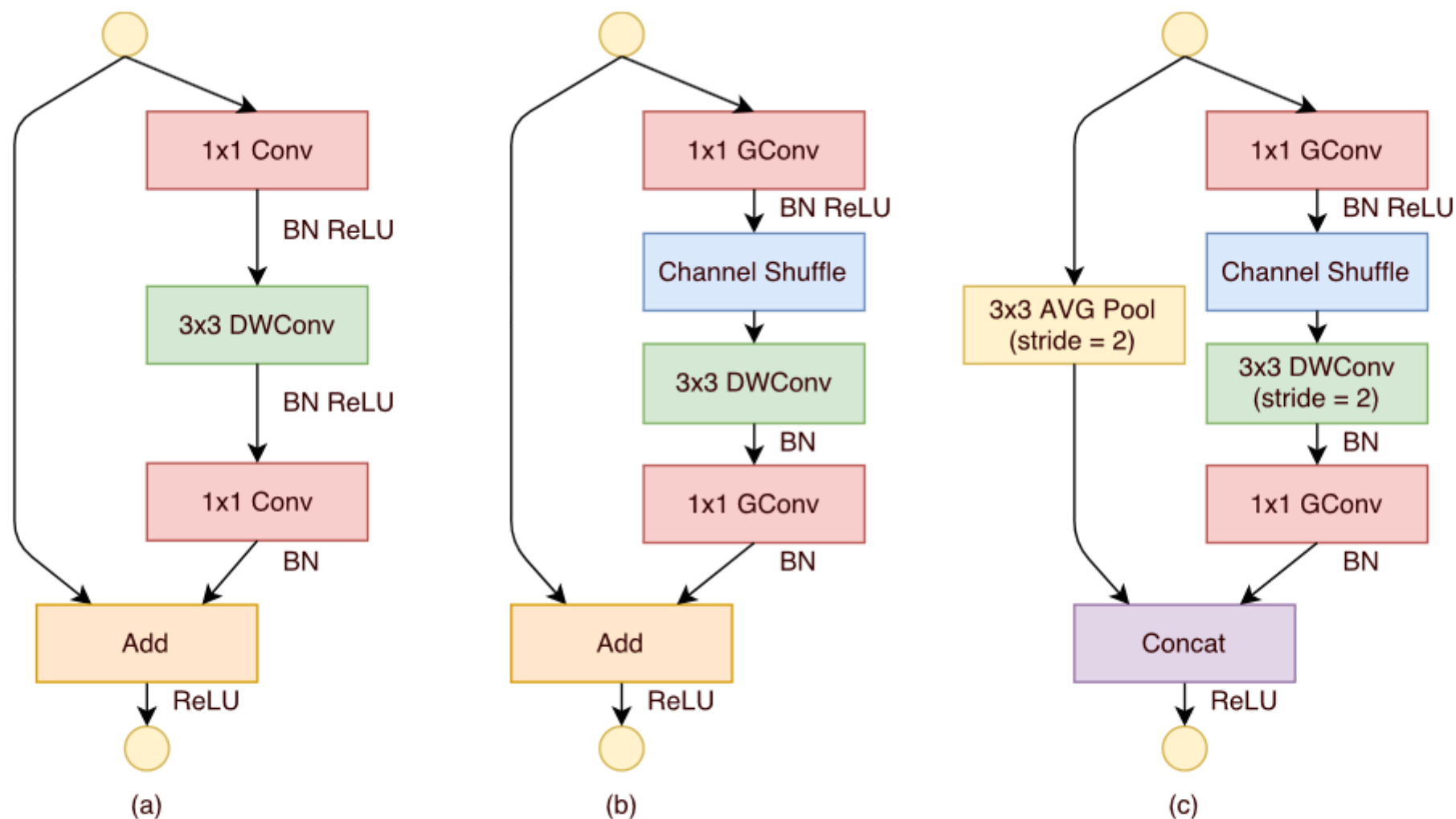


Figure 2. ShuffleNet Units. a) bottleneck unit [9] with depthwise convolution (DWConv) [3, 12]; b) ShuffleNet unit with pointwise group convolution (GConv) and channel shuffle; c) ShuffleNet unit with stride = 2.

последние 1×1 -свёртки для получения нужного числа каналов (для прокидывания связи)

WideResNets – «широкие» сети

можно увеличивать глубину, а что если увеличивать ширину

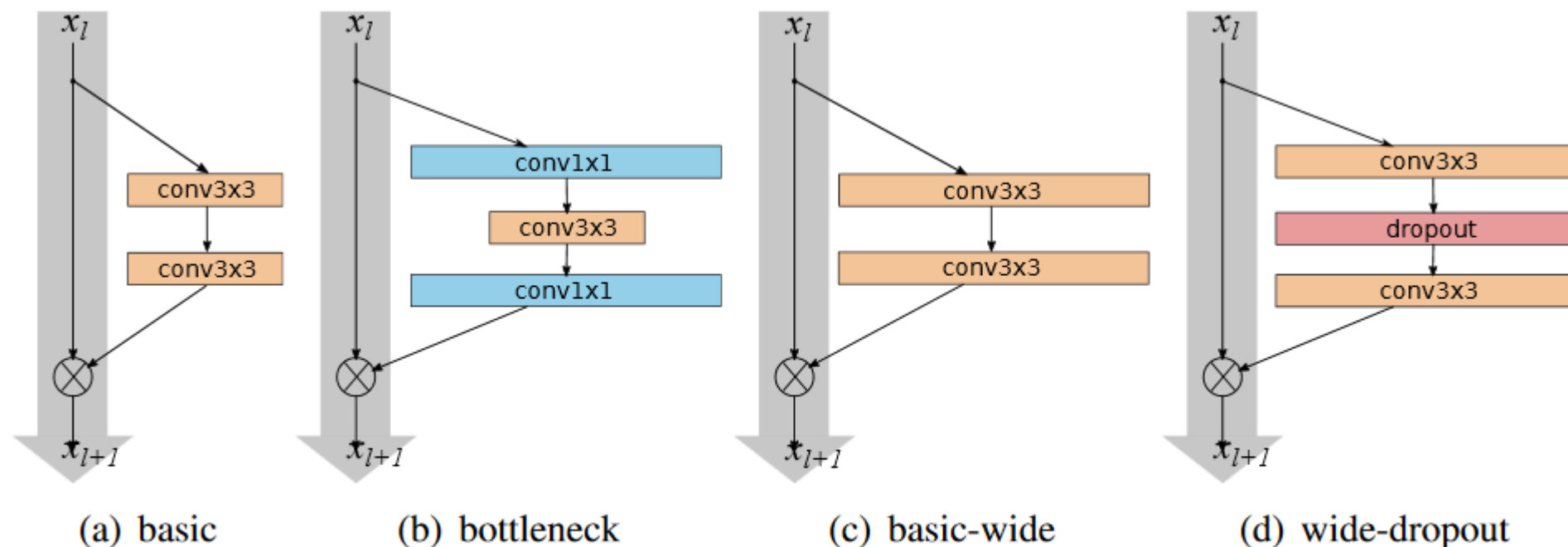


Figure 1: Various residual blocks used in the paper. Batch normalization and ReLU precede each convolution (omitted for clarity)

Sergey Zagoruyko, Nikos Komodakis «Wide Residual Networks» //

<https://arxiv.org/abs/1605.07146>

WideResNets – «широкие» сети

group name	output size	block type = $B(3, 3)$
conv1	32×32	$[3 \times 3, 16]$
conv2	32×32	$\begin{bmatrix} 3 \times 3, 16 \times k \\ 3 \times 3, 16 \times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3 \times 3, 32 \times k \\ 3 \times 3, 32 \times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3 \times 3, 64 \times k \\ 3 \times 3, 64 \times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Table 1: Structure of wide residual networks. Network width is determined by factor k . Original architecture [13] is equivalent to $k = 1$. Groups of convolutions are shown in brackets where N is a number of blocks in group, downsampling performed by the first layers in groups conv3 and conv4. Final classification layer is omitted for clearance. In the particular example shown, the network uses a ResNet block of type $B(3, 3)$.

параметр k – ширина – и оптимизация по нему

WideResNets – «широкие» сети

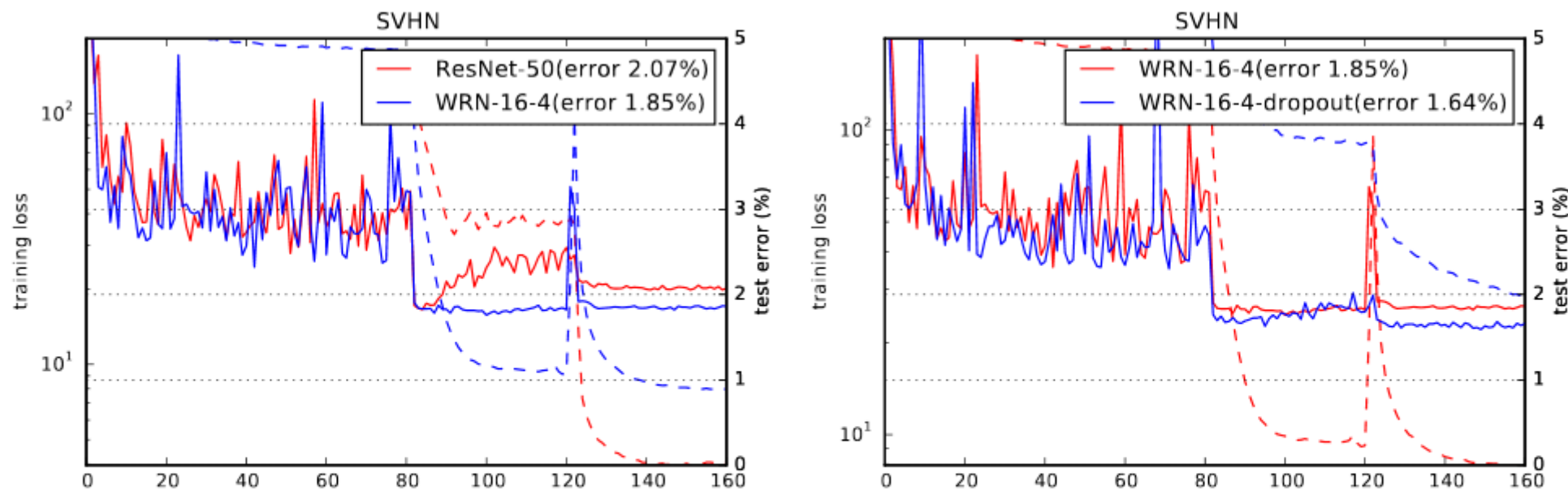
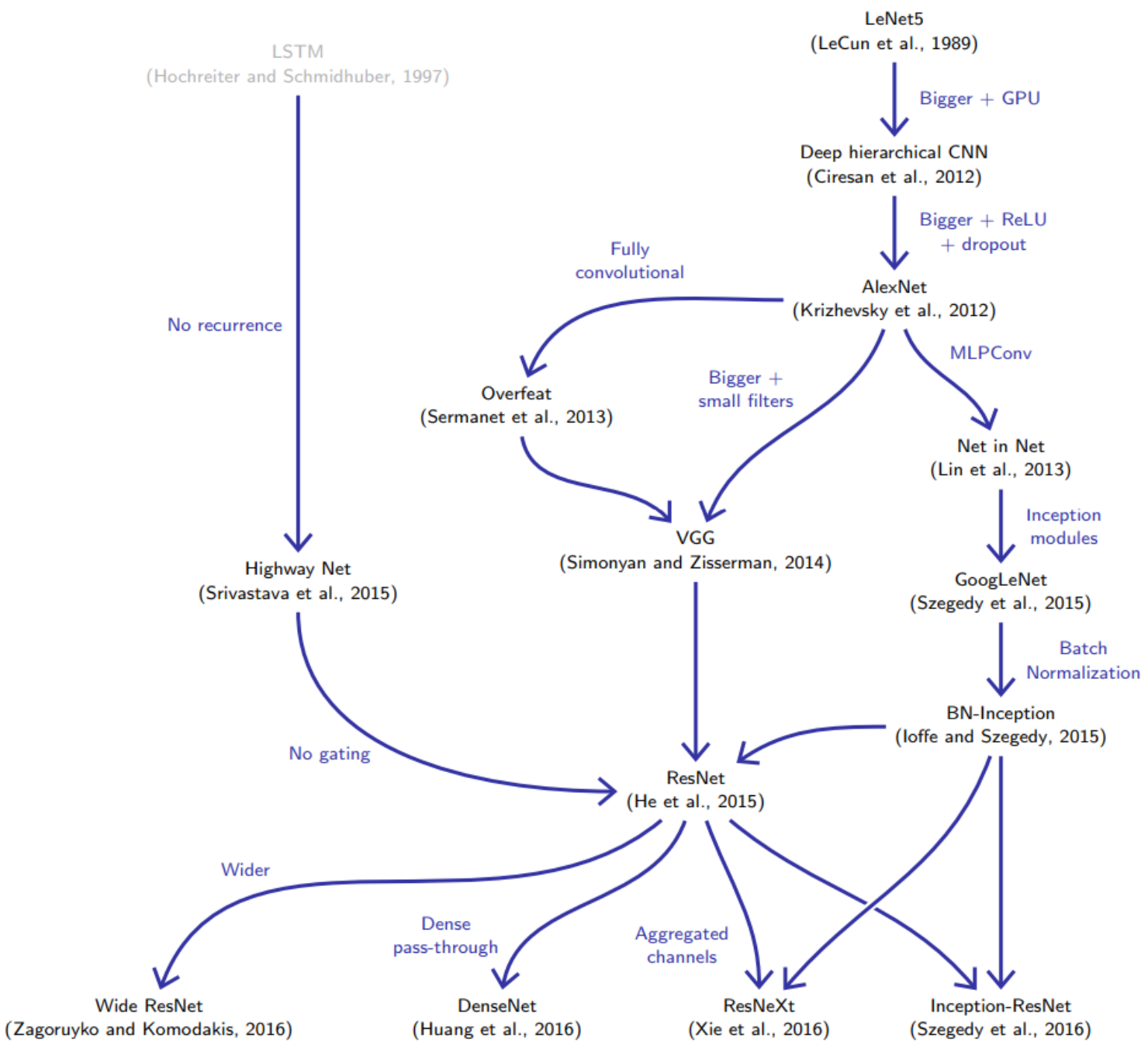


Figure 3: Training curves for SVHN. On the left: thin and wide networks, on the right: effect of dropout. Solid lines denote test error (y-axis on the right), dashed lines denote training loss (y-axis on the left).



<https://fleuret.org/ee559/ee559-slides-7-2-image-classification.pdf>

Применение CNN не только изображения!

везде, где есть локальность...

**Если задача как-то связана с изображениями,
часто берут архитектуру проверенную на ImageNet-е**

- Object detection
- Action recognition
- Human pose estimation
- Semantic segmentation
- Image captioning

И даже, если не связана с изображениями...

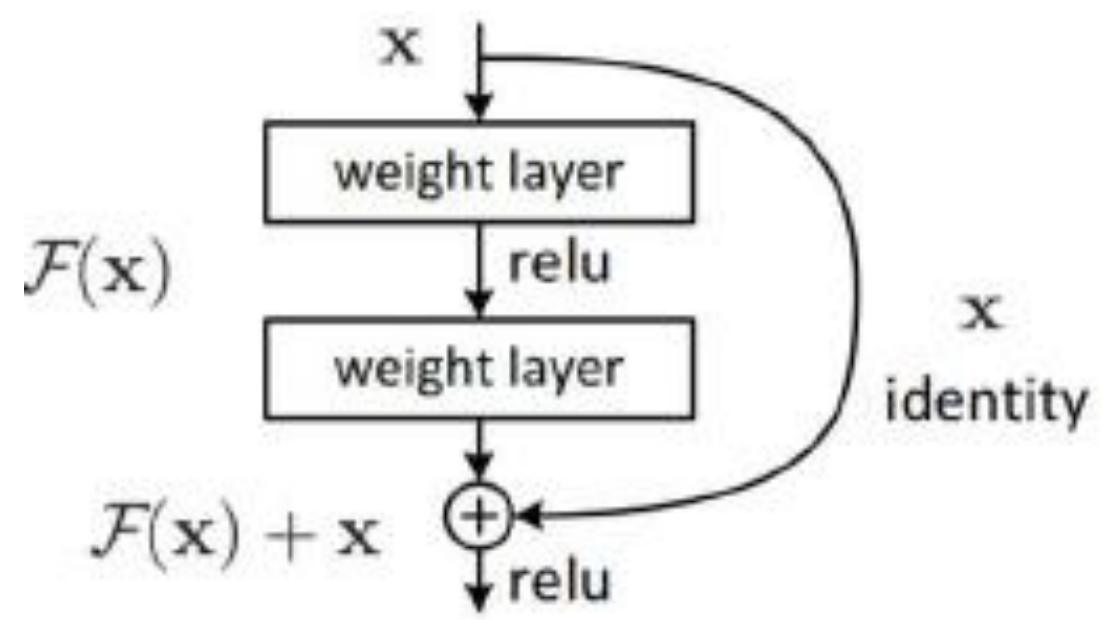
Volumetric Brain Segmentation (VoxResNet)

City-Wide Crowd Flow Prediction: (ST-ResNet)

Generating Realistic Voices (WaveNet)

СМ

ResNet: почему работает



ResNet: почему работает

Рассмотрим сети без прокидывания связей: plain-18 и plain-34 по числу слоёв

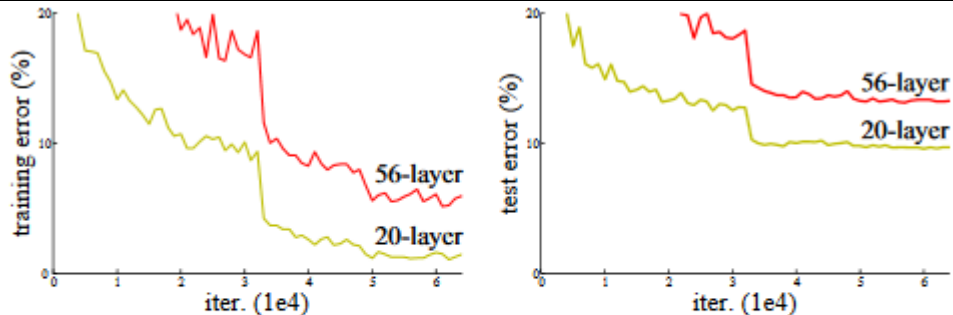
layer name	output size	18-layer	34-layer
conv1	112×112		
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$

plain-18 –лучше!

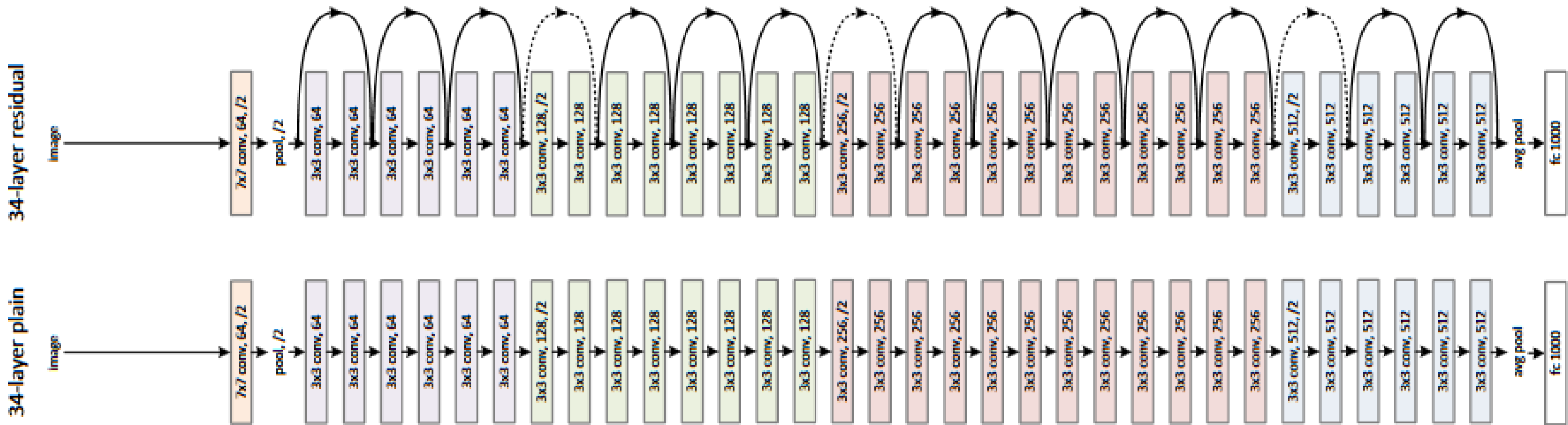
Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

ResNet: почему работает

Причины, почему plain-18 лучше

<div>Исчезающие градиенты (Vanishing Gradients)</div> <div>Нет</div>	<div>«We argue that this optimization difficulty is unlikely to be caused by vanishing gradients. These plain networks are trained with BN, which ensures forward propagated signals to have non-zero variances. We also verify that the backward propagated gradients exhibit healthy norms with BN. So neither forward nor backward signals vanish»</div>
<div>Переобучение (Overfitting)</div> <div>Нет</div>	<div><p>Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.</p></div>
<div>Функциональная мощность (Representation power)</div> <div>нет</div>	

ResNet: почему работает



**Сделали прокидывание связей – и ситуация изменилась
(глубокие сети лучше)**

посмотрите, как соотносится картинка и предыдущая табличка

ResNet: почему работает

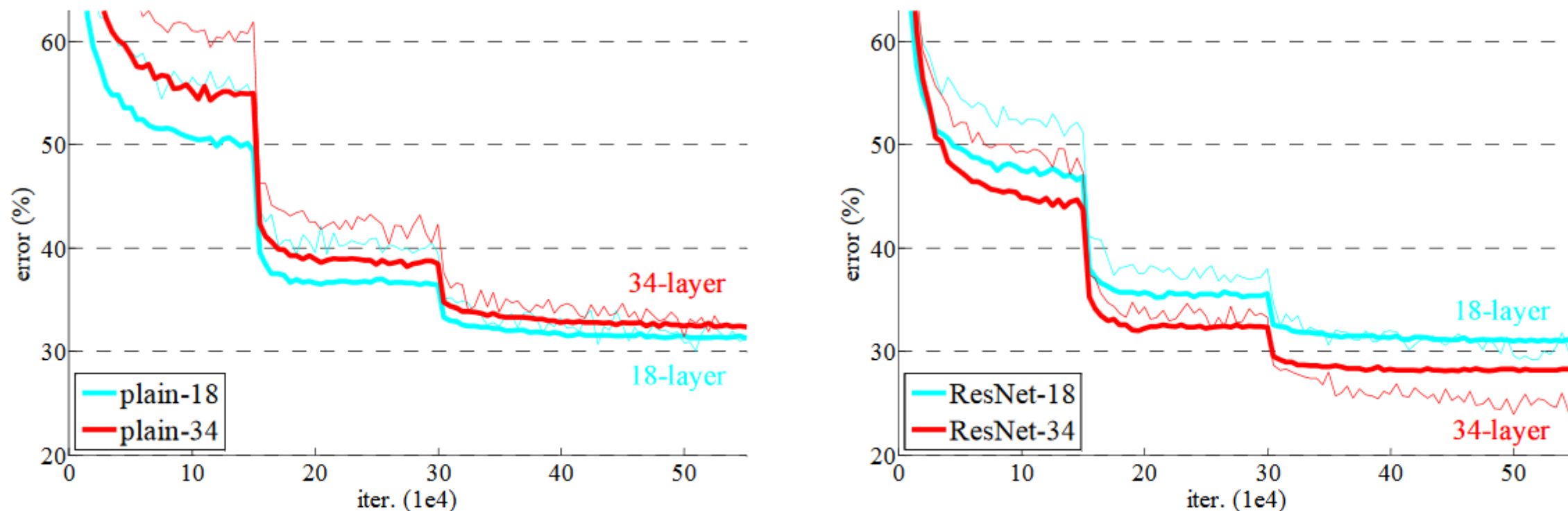


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

**Сделали прокидывание связей – и ситуация изменилась
(глубокие сети лучше)**

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	25.03

Table 2. Top-1 error (% , 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

ResNet: почему работает

Меньше разброс в активациях

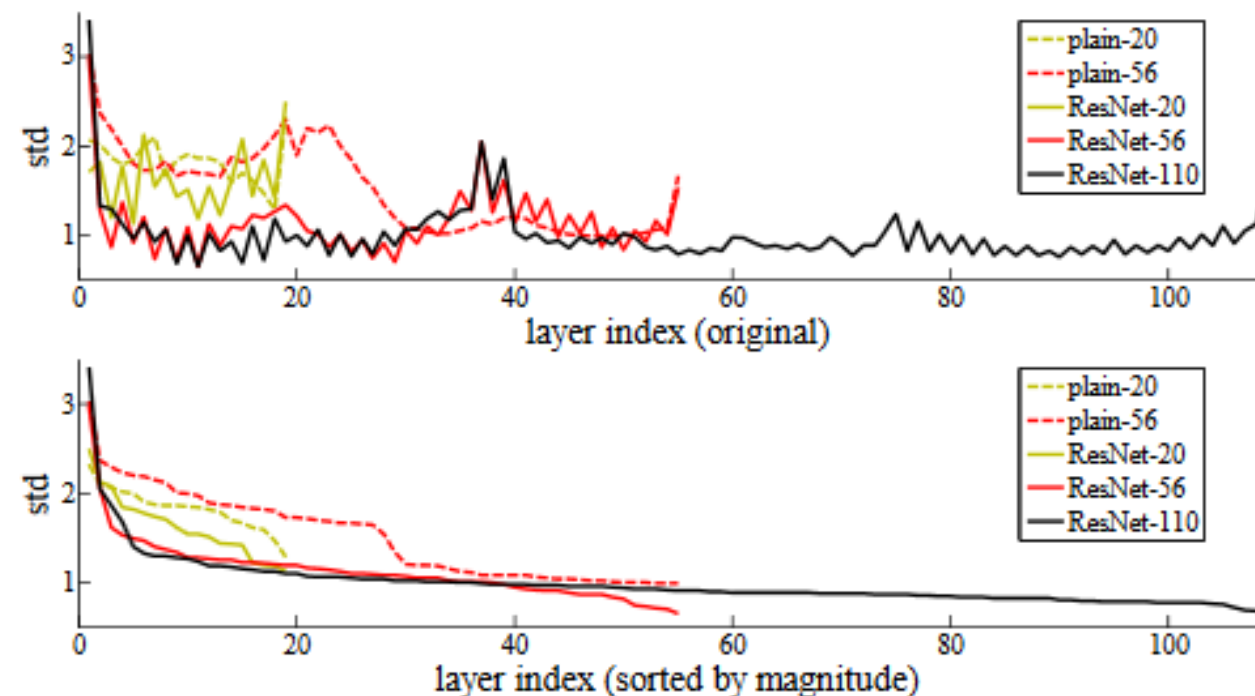


Figure 7. Standard deviations (std) of layer responses on CIFAR-10. The responses are the outputs of each 3×3 layer, after BN and before nonlinearity. **Top:** the layers are shown in their original order. **Bottom:** the responses are ranked in descending order.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016 // <https://arxiv.org/abs/1512.03385>

ResNet: почему работает

Моделирует ансамбль сетей разной глубины

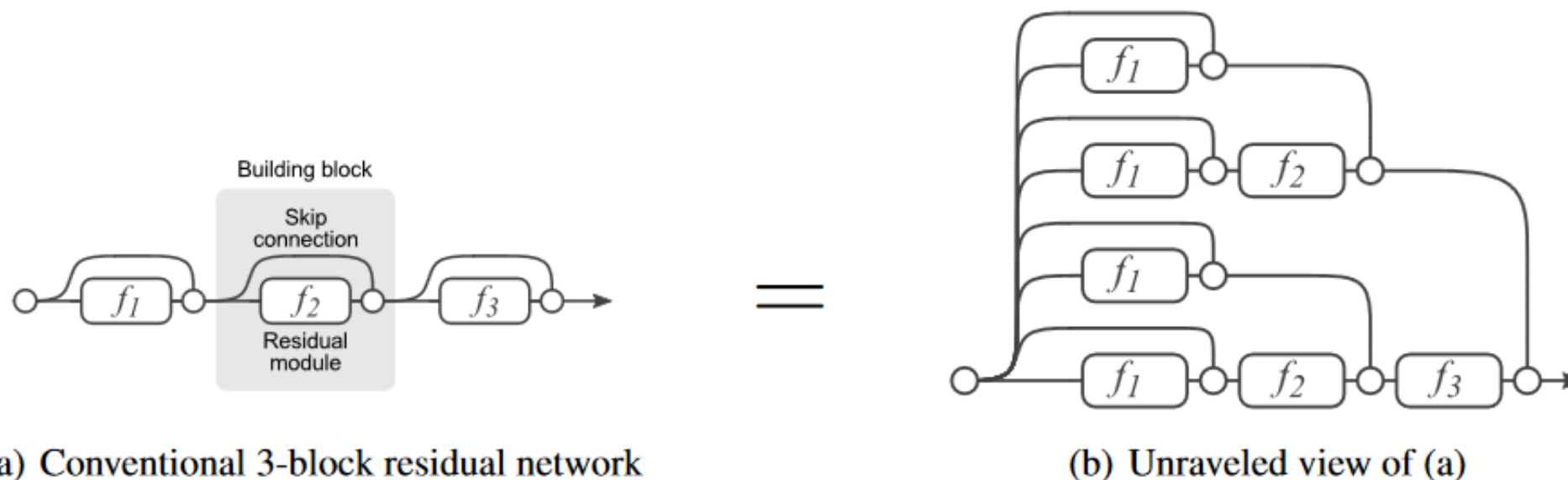


Figure 1: Residual Networks are conventionally shown as (a), which is a natural representation of Equation (1). When we expand this formulation to Equation (6), we obtain an *unraveled view* of a 3-block residual network (b). Circular nodes represent additions. From this view, it is apparent that residual networks have $O(2^n)$ implicit paths connecting input and output and that adding a block doubles the number of paths.

вспомним, что на разных уровнях – разные графические примитивы

Andreas Veit, Michael Wilber, Serge Belongie «Residual Networks Behave Like Ensembles of Relatively Shallow Networks» // <https://arxiv.org/pdf/1605.06431.pdf>

Эксперименты с удалением слоёв

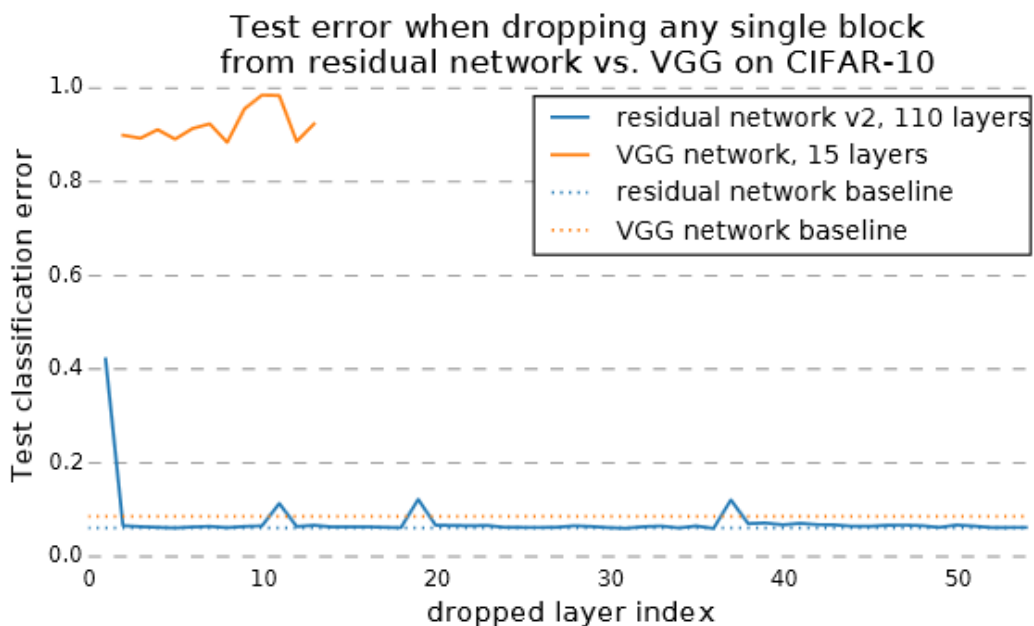


Figure 3: Deleting individual layers from VGG and a residual network on CIFAR-10. VGG performance drops to random chance when any one of its layers is deleted, but deleting individual modules from residual networks has a minimal impact on performance. Removing downsampling modules has a slightly higher impact.

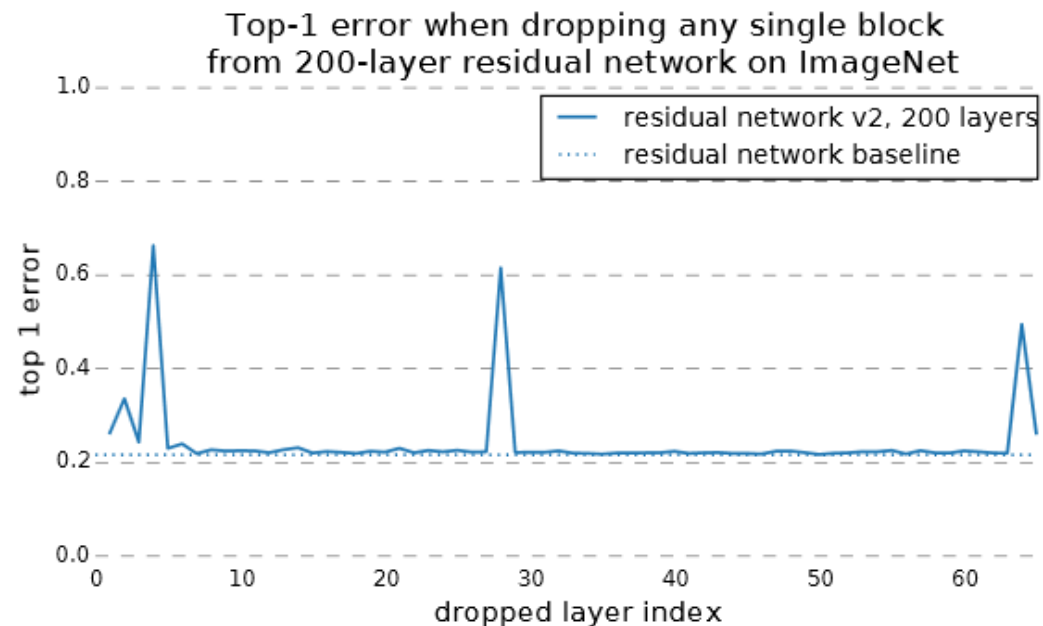


Figure 4: Results when dropping individual blocks from residual networks trained on ImageNet are similar to CIFAR results. However, downsampling layers tend to have more impact on ImageNet.

удаляется residual block, а прямая связь остаётся
пики – понижение размерности

Эксперименты с удалением и перестановкой слоёв

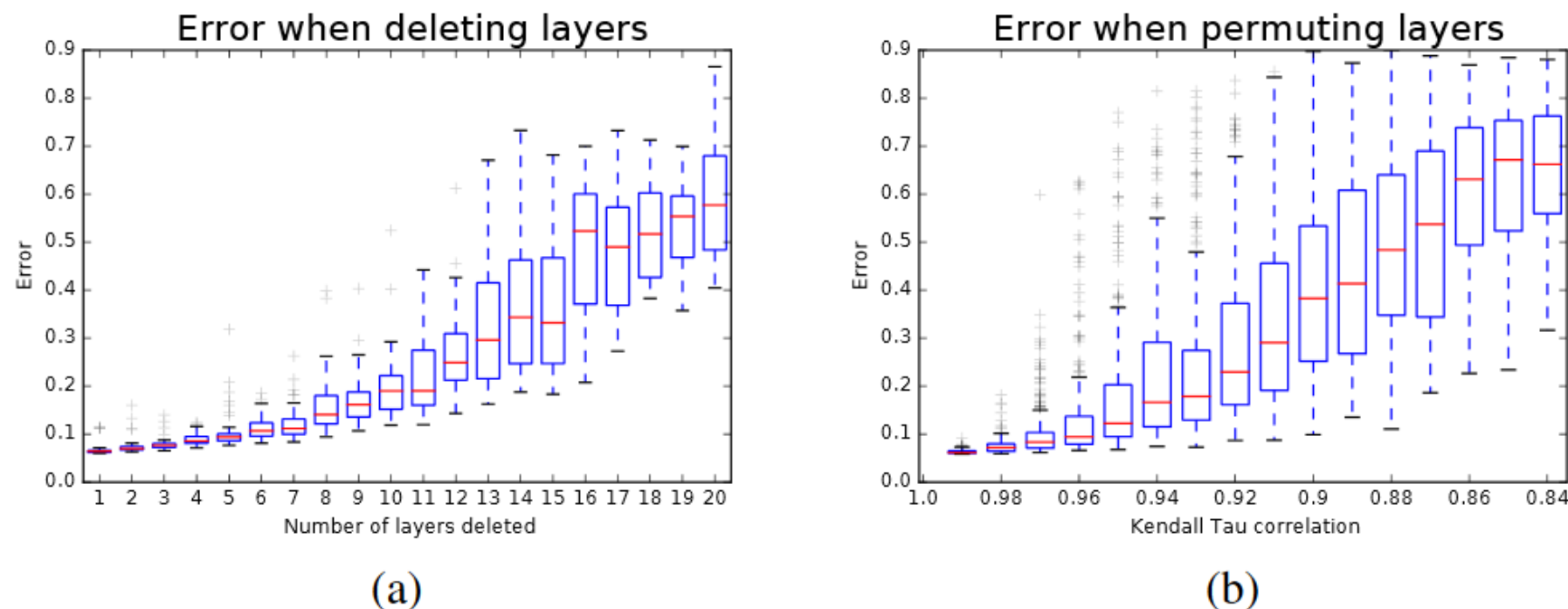


Figure 5: (a) Error increases smoothly when randomly deleting several modules from a residual network. (b) Error also increases smoothly when re-ordering a residual network by shuffling building blocks. The degree of reordering is measured by the Kendall Tau correlation coefficient. These results are similar to what one would expect from ensembles.

**удаление схоже на удаление деревьев в RF – малый эффект
ещё одно подтверждение сходства с ансамблем**

см. ещё рис. в статье

ResNet: число слоёв

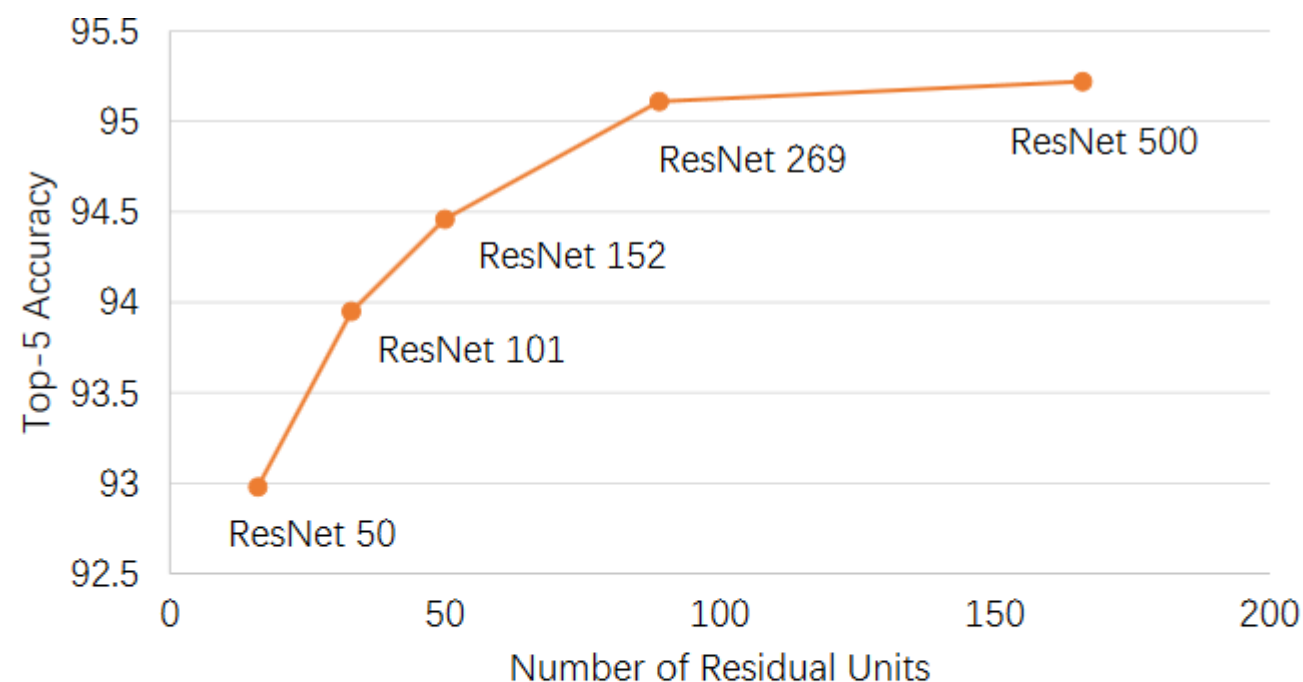
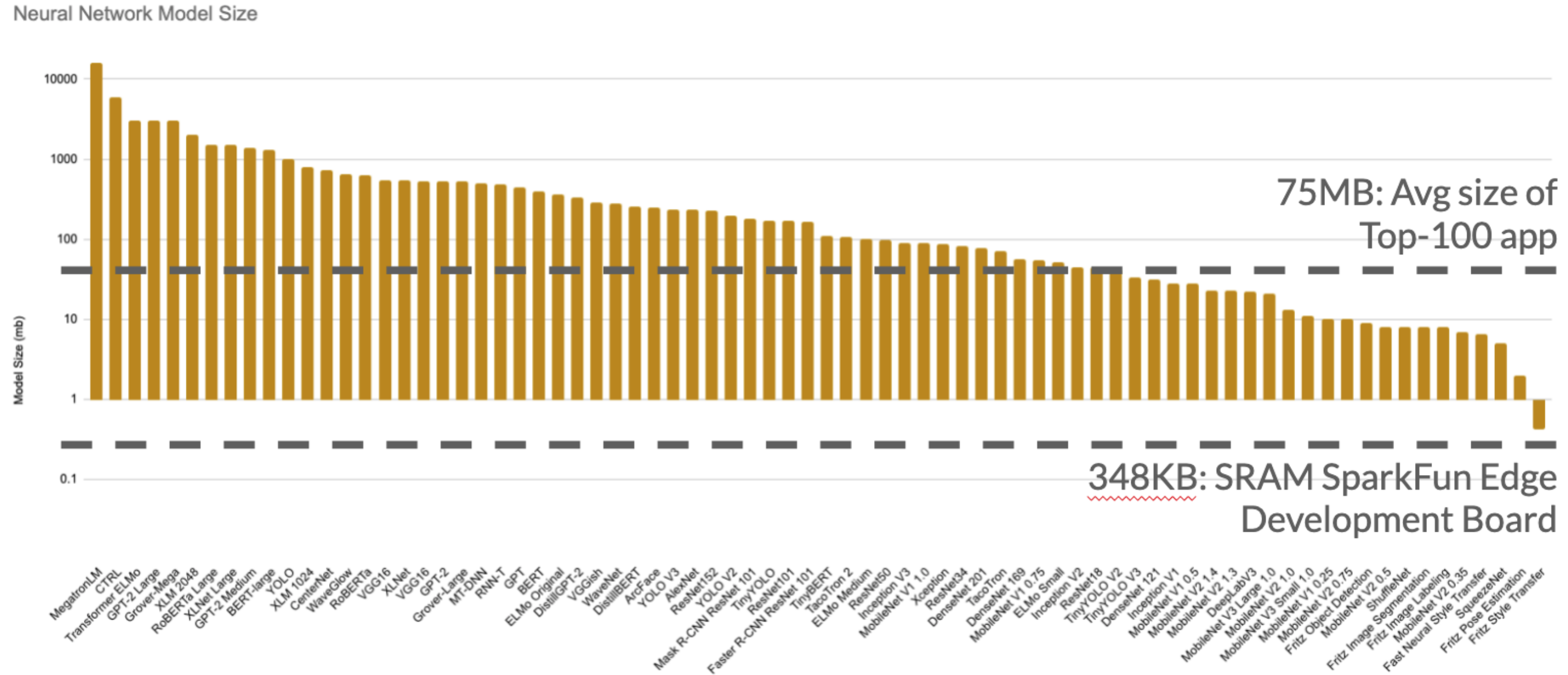


Figure 2: Top-5 single crop accuracies of ResNets [10] with different number of residual units on the ILSVRC 2012 validation set. As the number of residual units increases beyond 100, we can see that the return diminishes.

<https://arxiv.org/pdf/1611.05725.pdf>

Сложность моделей



Итоги

**факторизация (суперпозиция) свёрток
особенно на первых слоях**

1×1-свёртки

прокидывание связей

перебор параметров

масштабирование и сжатие сетей