CS698U: Topics in Computer Vision

Jan—May 2017

Lecture 3

Gaurav Sharma

Indian Institute of Technology Kanpur

www.grvsharma.com

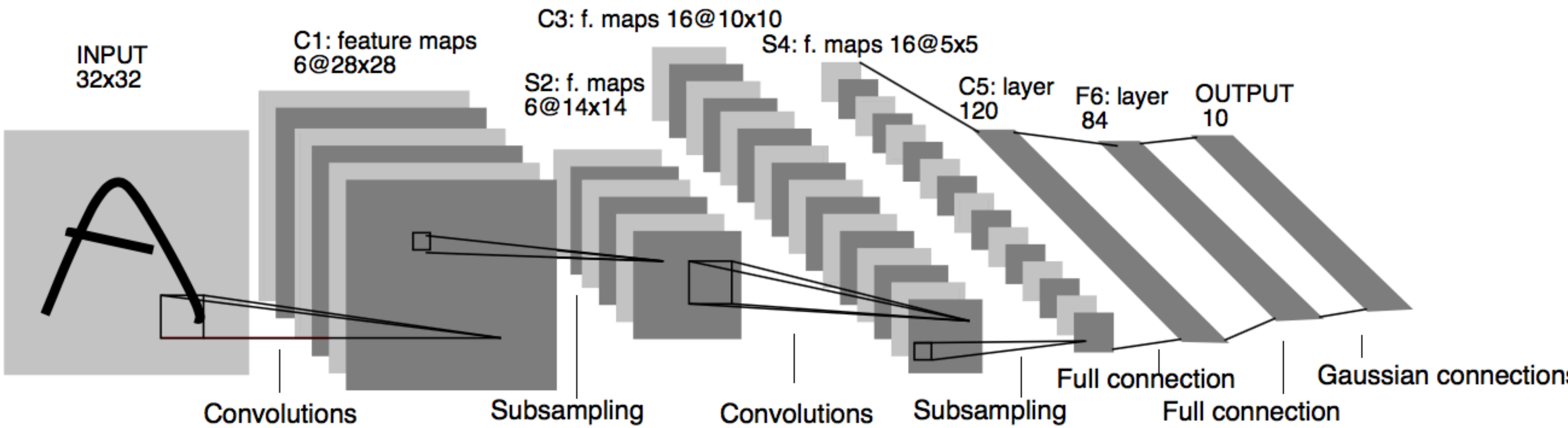
State-of-the-art CNNs in Computer Vision

Image Classification

- Close look at LeNet5 and the four popular/successful CNN architectures for image classification today
  - AlexNet
  - VGG16 & VGG19
  - GoogLeNet
  - ResNet

Krizhevsky et al., Imagenet classification with deep convolutional neural networks, NIPS 2012
Simonyan and Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR 2015
Szegedy et al.,Going deeper with convolutions, CVPR 2015
He et al., Deep residual learning for image recognition, CVPR 2016

LeNet-5



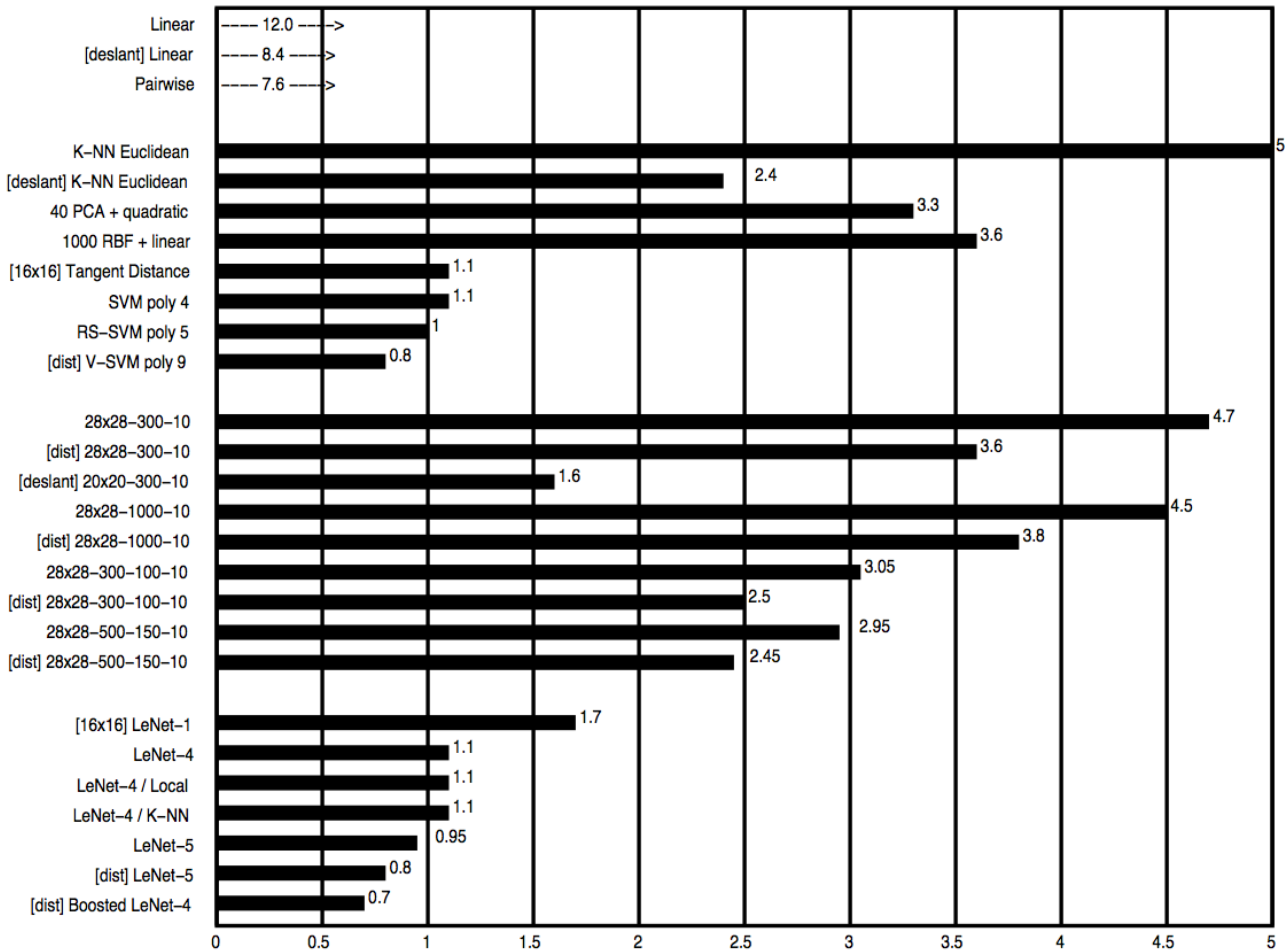
Subsampling: 2x2 window — averaging followed by multiplication by trainable coeff. and addition with a trainable bias

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, November 1998a.

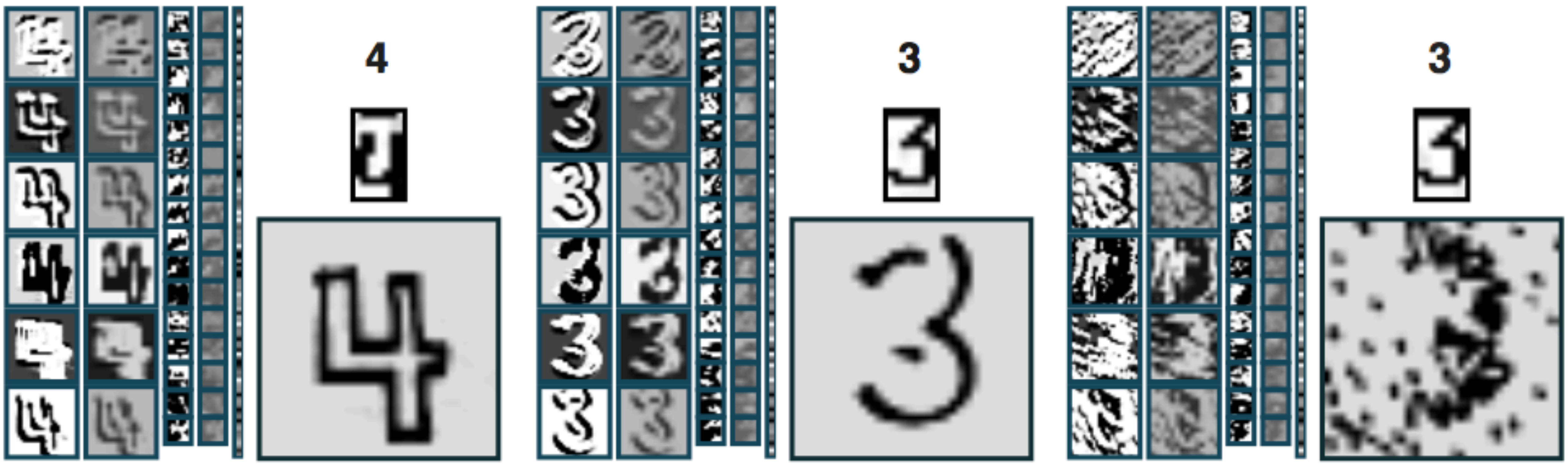
Handwritten Digit Classification

Misclassifications

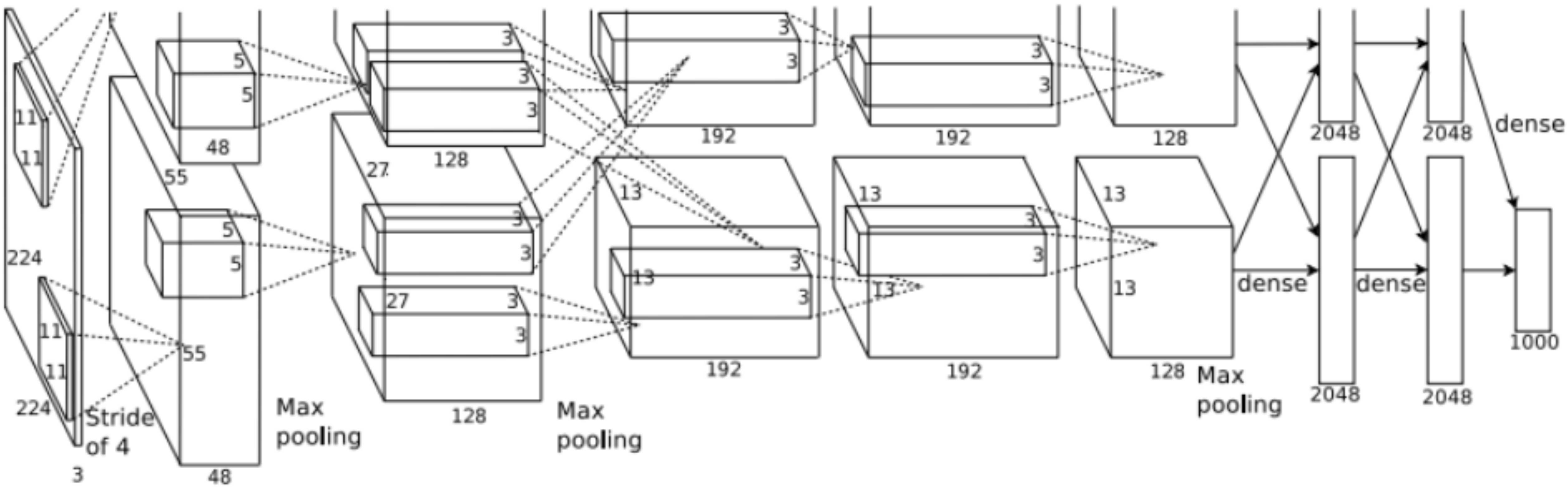
Quantitative Results (Error Rate)



Qualitative Results



AlexNet



input conv1 conv2 conv3 conv4 conv5 fc6 fc7 fc8

conv1	conv2	conv3	conv4	conv5	
11x11x3	5x5x48	3x3x128	3x3x192	3x3x192	Filter sizes
48x2	128x2	192x2	192x2	128x2	Num filters

Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

VGG Nets

- Main difference cf. AlexNet
- Use very small convolutional kernels of 3x3
- Increase the depth of the network
- Tested multiple architectures

Simonyan and Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR 2015

Small convolution kernels

- Receptive fields

Receptive Field	Conv. layers
3x3	conv3
5x5	conv3+conv3
7x7	conv3+conv3+conv3

- Why not conv7 instead of 3 conv3 layers ?

Simonyan and Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR 2015

Small convolution kernels

- Nonlinearities
  - Three nonlinear transformations between three conv3 layers
  - Cf. one for conv7
- #Parameters (assuming C channels)
  - 3 x (3C)^2 = 27C^2 for three conv3
  - (7C)^2 = 49C^2 for conv7 (81% more)

Architectures Tested

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**

Common

maxpool
FC-4096
FC-4096
FC-1000
soft-max

Number of Parameters

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Layers 11 13 16 16 19

For comparison
AlexNet has 8 layers and 60 million parameters

Simonyan and Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR 2015

Evaluations

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

Simonyan and Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR 2015

Generalization

Features from networks trained on ImageNet, tested on standard benchmarks

Method	VOC-2007 (mean AP)	VOC-2012 (mean AP)	Caltech-101 (mean class recall)	Caltech-256 (mean class recall)
Zeiler & Fergus (Zeiler & Fergus, 2013)	-	79.0	86.5 ± 0.5	74.2 ± 0.3
Chatfield et al. (Chatfield et al., 2014)	82.4	83.2	88.4 ± 0.6	77.6 ± 0.1
He et al. (He et al., 2014)	82.4	-	$\mathbf{93.4 \pm 0.5}$	-
Wei et al. (Wei et al., 2014)	81.5 (85.2*)	81.7 (**90.3***)	-	-
VGG Net-D (16 layers)	89.3	89.0	91.8 ± 1.0	85.0 ± 0.2
VGG Net-E (19 layers)	89.3	89.0	92.3 ± 0.5	85.1 ± 0.3
VGG Net-D & Net-E	**89.7**	**89.3**	92.7 ± 0.5	$\mathbf{86.2 \pm 0.3}$

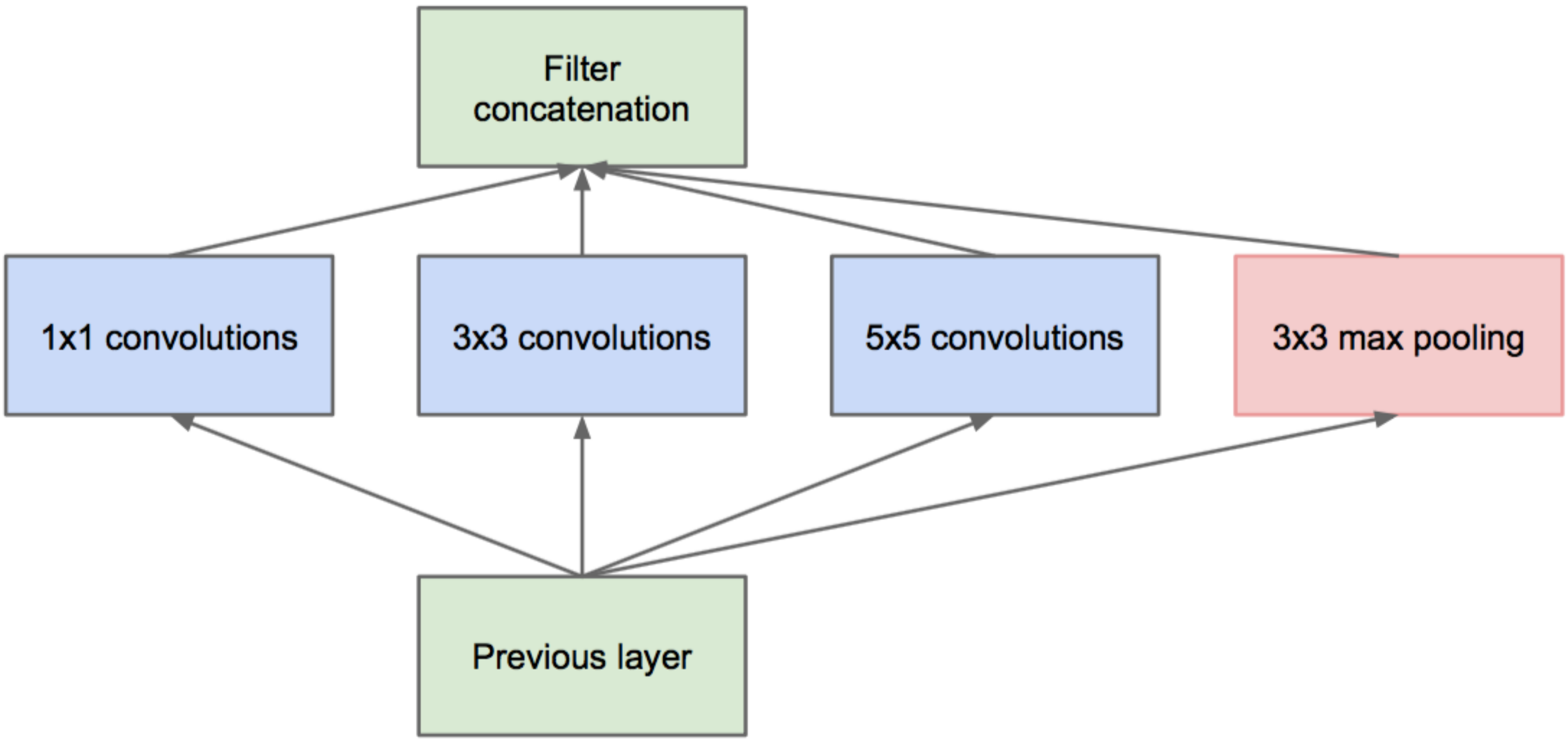
Simonyan and Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR 2015

GoogLeNet

- 22 Layer deep CNN
- 12x less parameters cf. AlexNet
- 1.5 billion multiply-adds at inference time
- ‘Inception module’

Szegedy et al.,Going deeper with convolutions, CVPR 2015

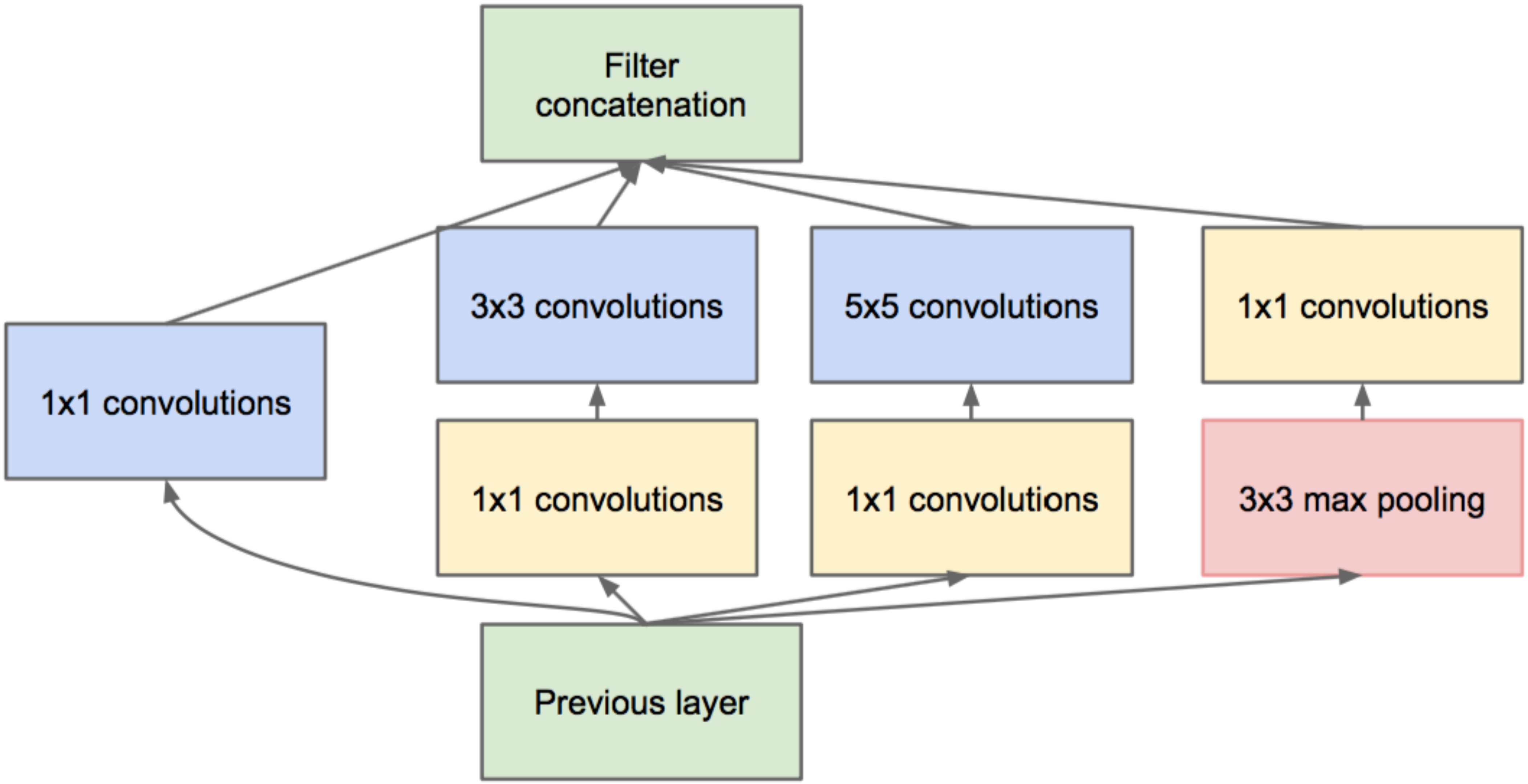
Inception Module



(a) Inception module, naïve version

Szegedy et al.,Going deeper with convolutions, CVPR 2015

Inception Module



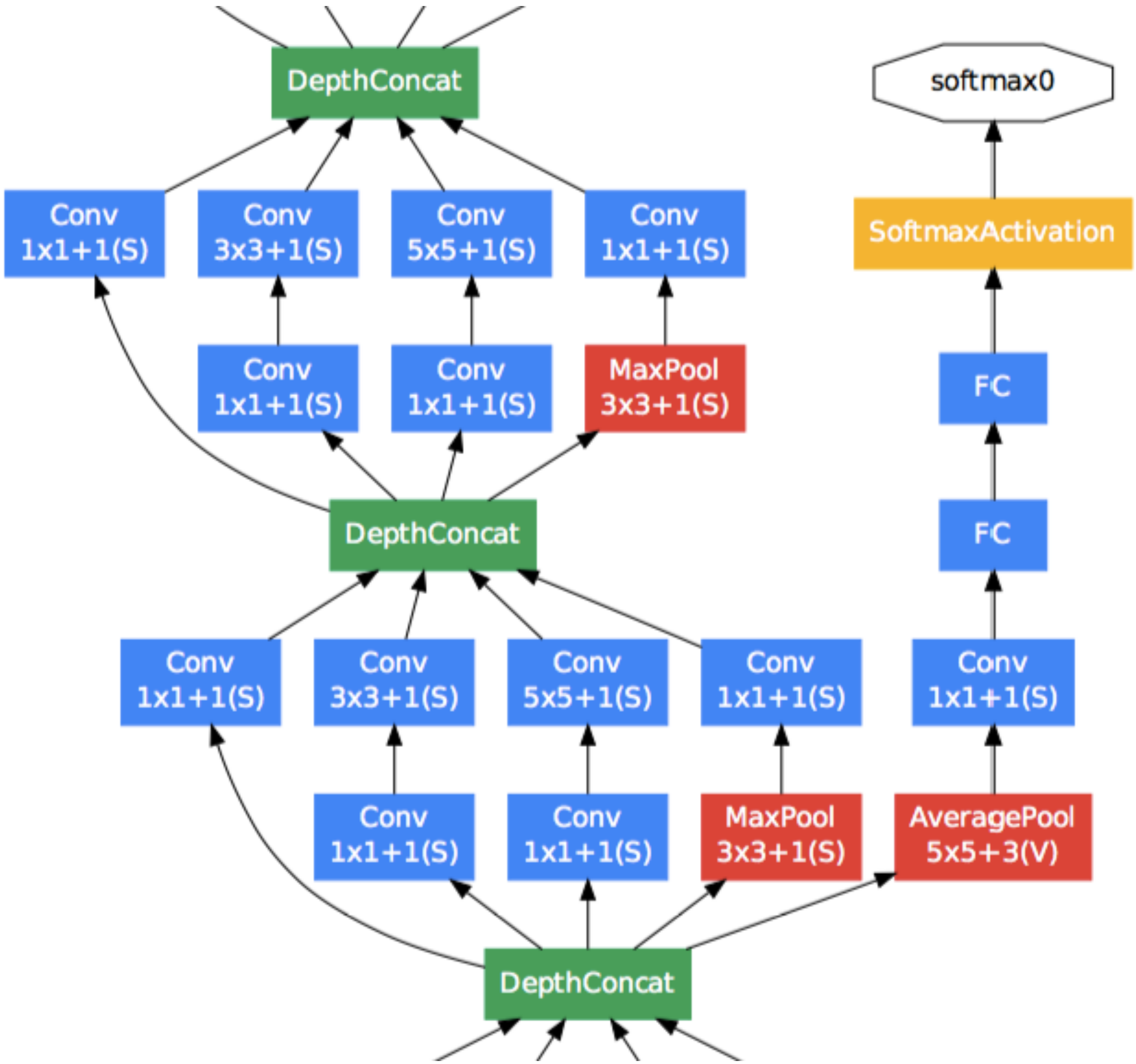
(b) Inception module with dimension reductions

Szegedy et al.,Going deeper with convolutions, CVPR 2015

GoogLeNet

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								

Intermediate Supervision



Auxiliary classifiers added (at two layers in the middle)

Performances

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance

Average over 7 independently trained GoogLeNets

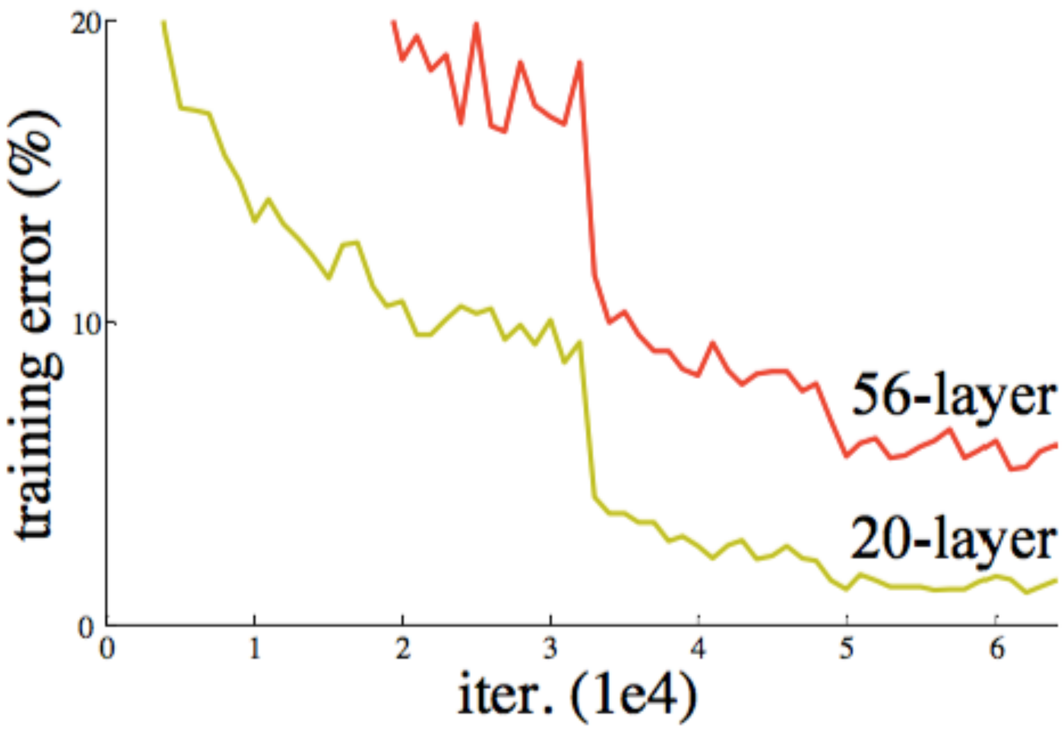
Szegedy et al.,Going deeper with convolutions, CVPR 2015

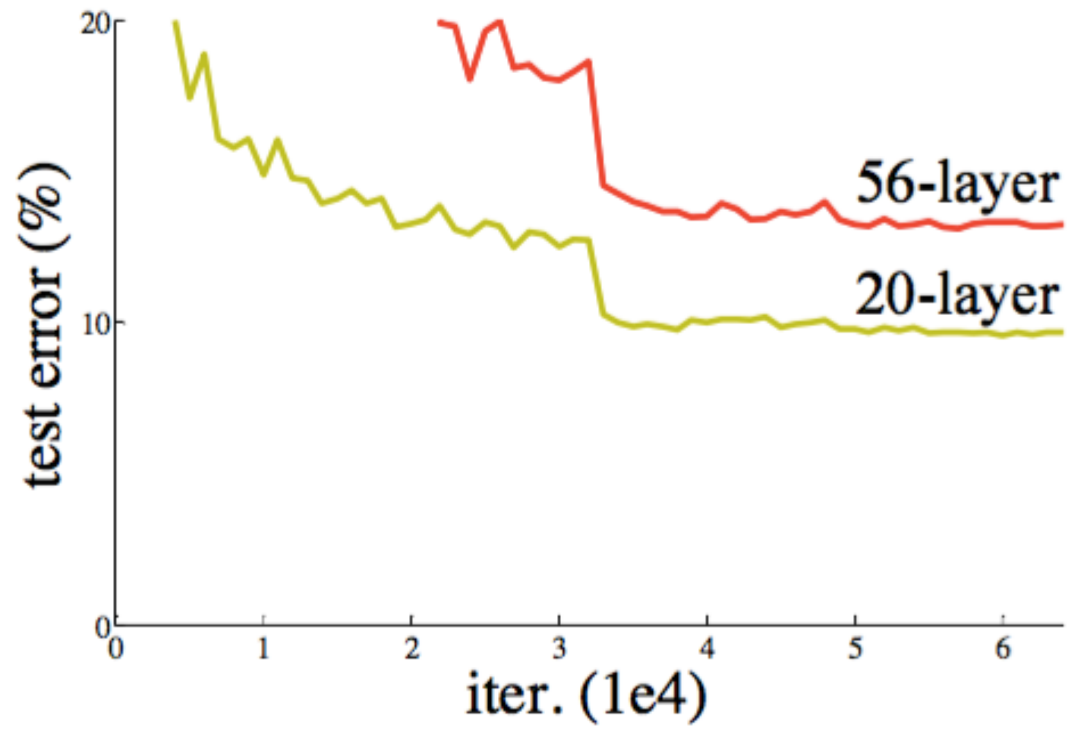
ResNet (MSR)

- 152 Layer network
- Cf. VGG nets
  - 8x deeper
  - Lower complexity

He et al., Deep residual learning for image recognition, CVPR 2016

Deeper nets are hard to train





- Training error increases as well — not overfitting !

He et al., Deep residual learning for image recognition, CVPR 2016

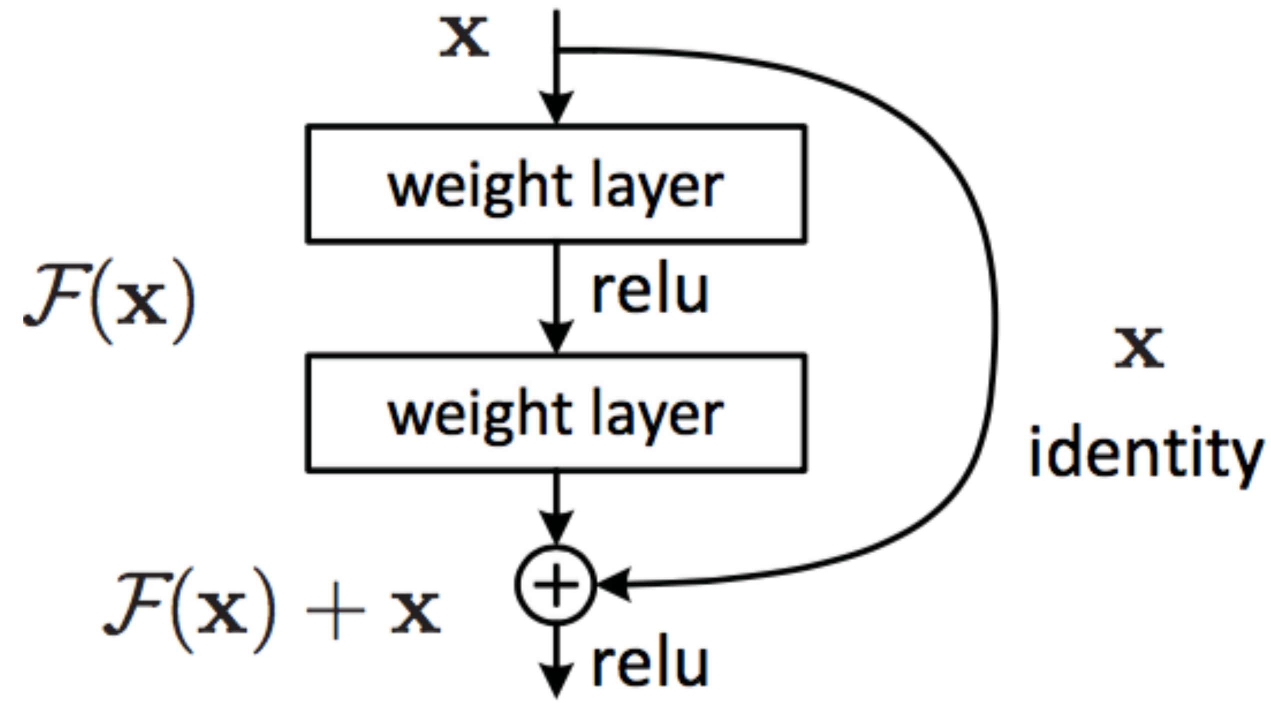
Deeper nets shouldn't be bad

- Given a shallow net with certain performance
- Make a deep net by adding identity layers
- In the worst case
    - Performances of shallow and deep nets same

He et al., Deep residual learning for image recognition, CVPR 2016

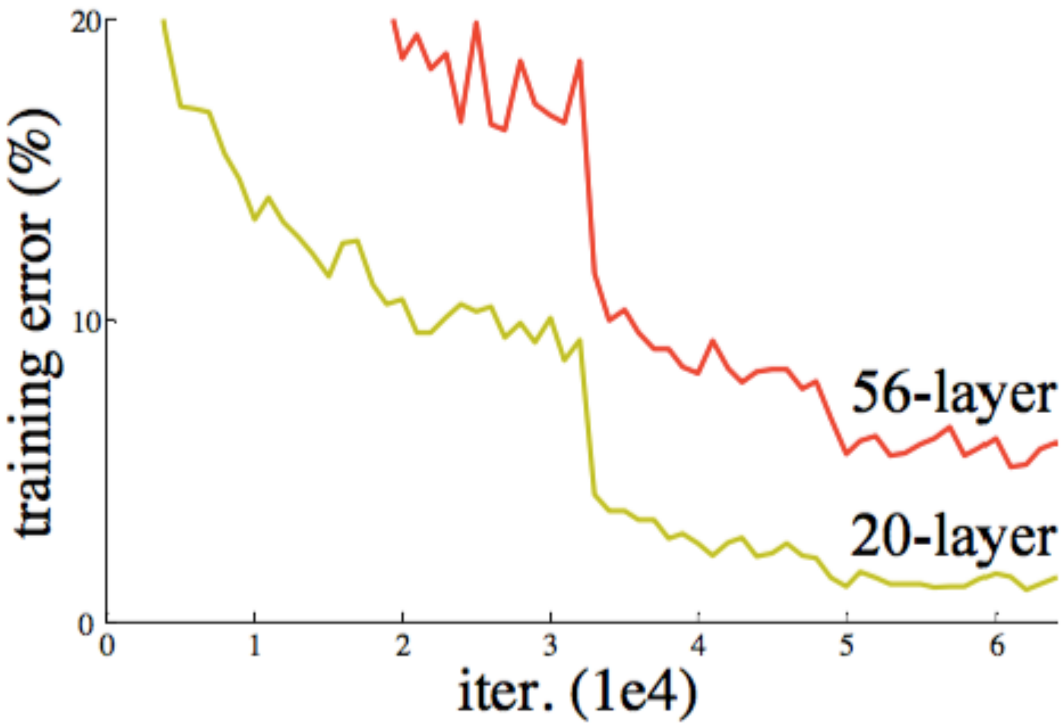
Residual Learning

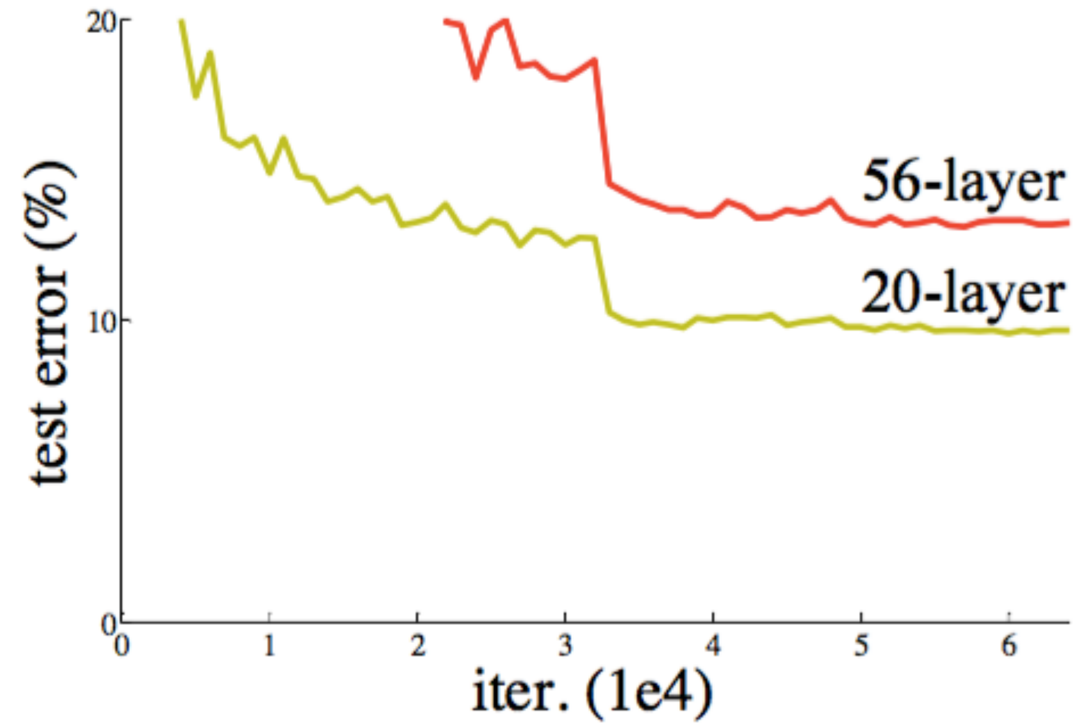
- Learn residual mapping
- If identity was optimal
  - F(x) should be zero
  - … easier to learn
- Shortcut connections



He et al., Deep residual learning for image recognition, CVPR 2016

Intuition/justification





- Degradation suggests problems approximating identity mappings with multiple non-linear layers

- Preconditioning in case optimal mappings are close to 1

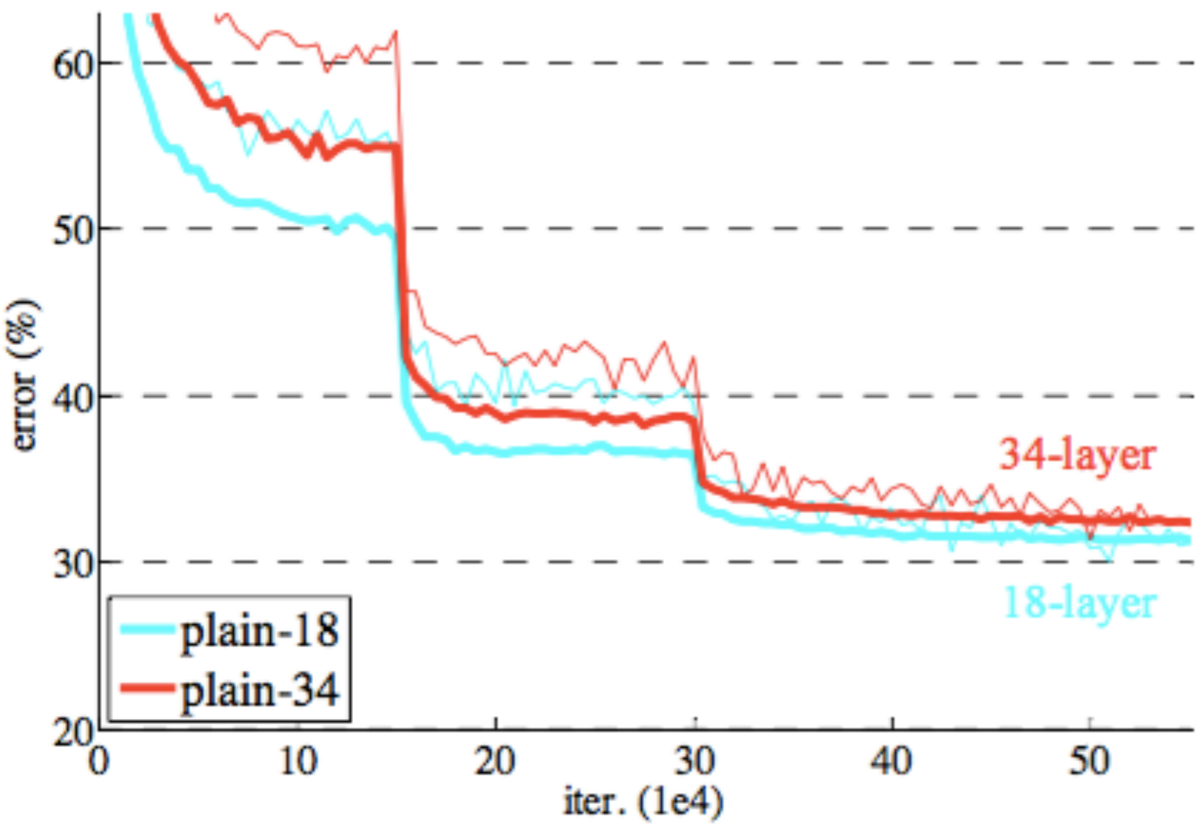
He et al., Deep residual learning for image recognition, CVPR 2016

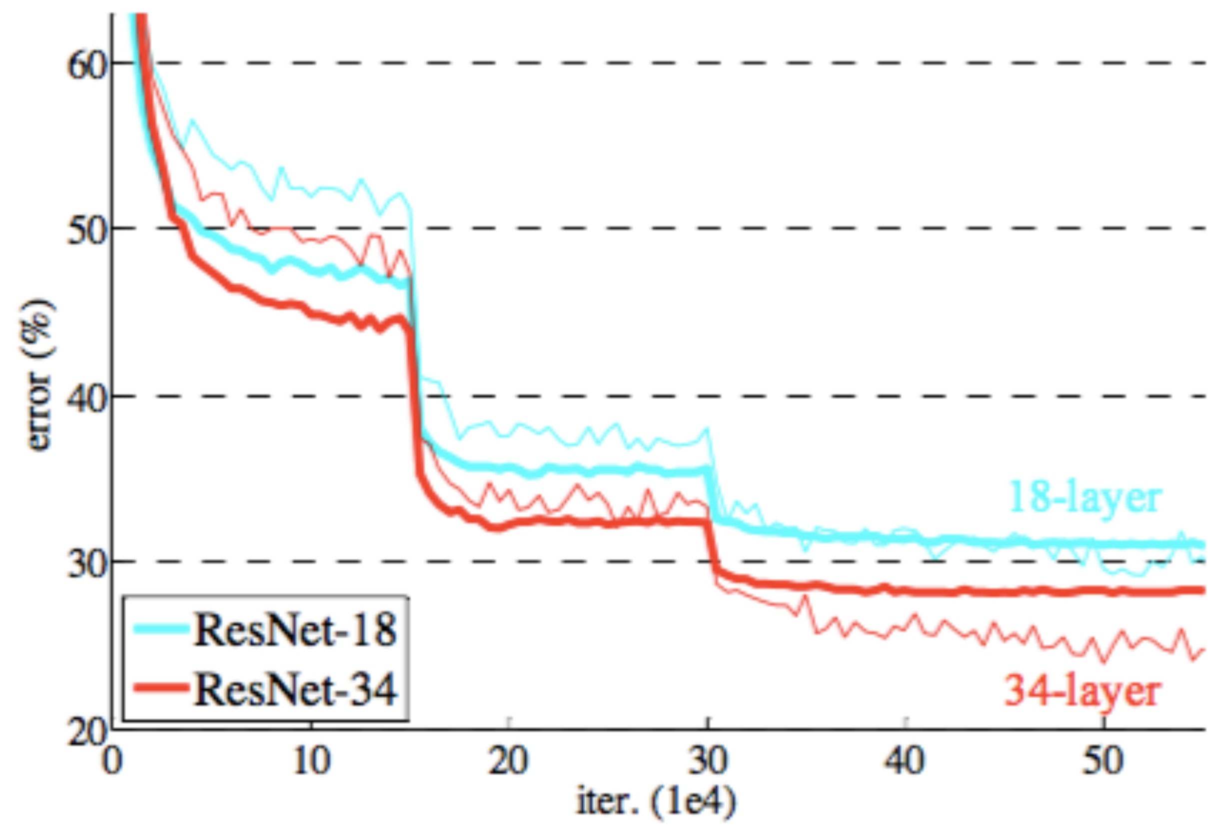
Architecture

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				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

He et al., Deep residual learning for image recognition, CVPR 2016

Training and Validation





- Plain (left) vs. ResNet (right)
- Thin (train) and thick (val)

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

10 crop testing

Comparisons

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

10 crop testing

Comparisons — Ensemble

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Are ResNets really deep ?

Exp Ensembles of Shallow Nets

- Interpret them as ensembles of relatively shallow networks
  - ResNet 110 = ensembles of 10-34 layer nets
- Network params: width, depth and
  - *Multiplicity*: size of the implicit ensemble

Veit et al., Residual Networks are Exponential Ensembles of Relatively Shallow Networks, NIPS 2016

Recursive expansion

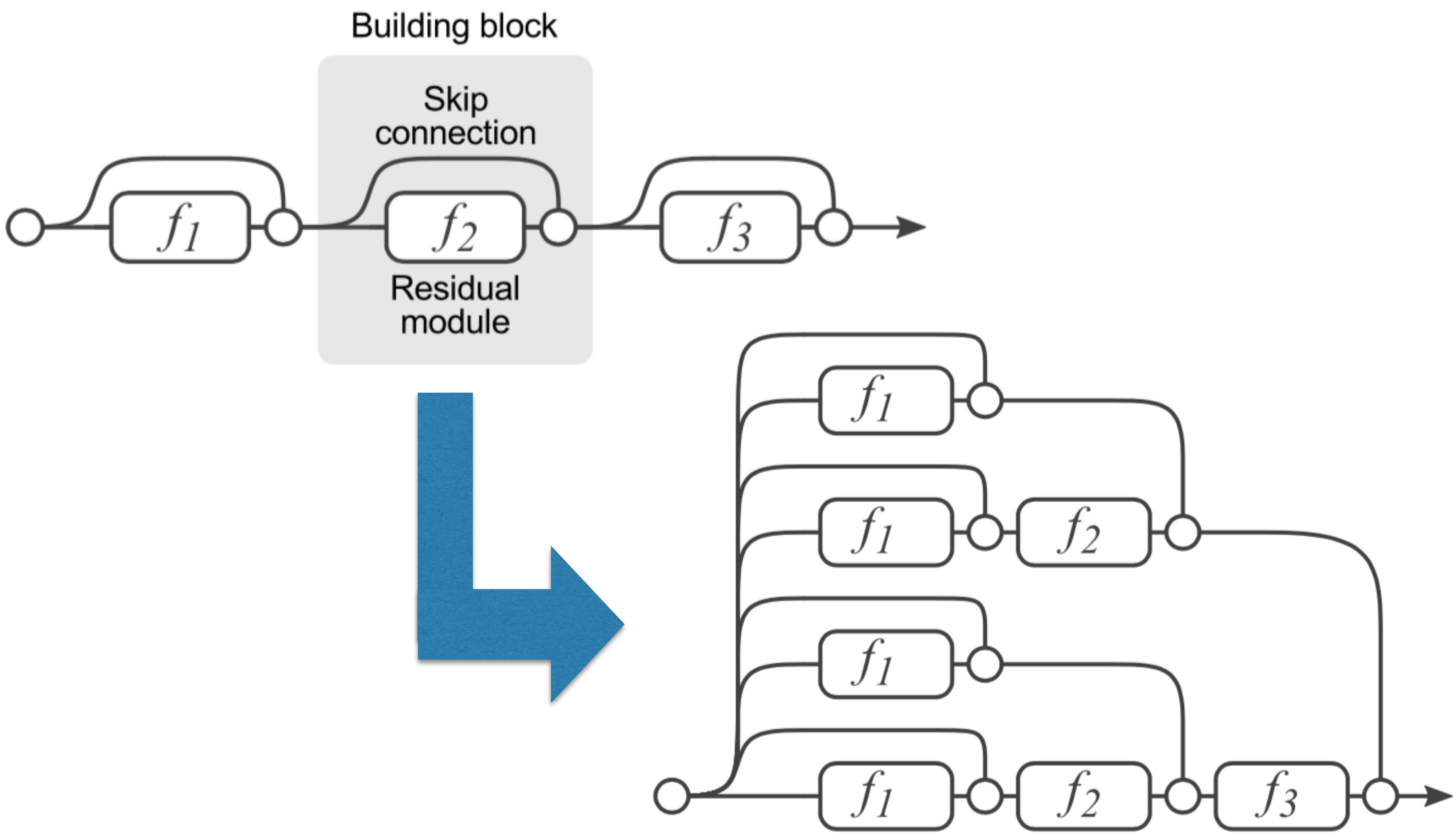
- Basic residual block

$$y_{i+1} \equiv f_{i+1}(y_i) + y_i$$

- Expansion

$$\begin{aligned} y_3 &= y_2 + f_3(y_2) \\ &= [y_1 + f_2(y_1)] \;+\; f_3(y_1 + f_2(y_1)) \\ &= \Big[y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\Big] \;+\; f_3\Big(y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\Big) \end{aligned}$$

Unraveled view



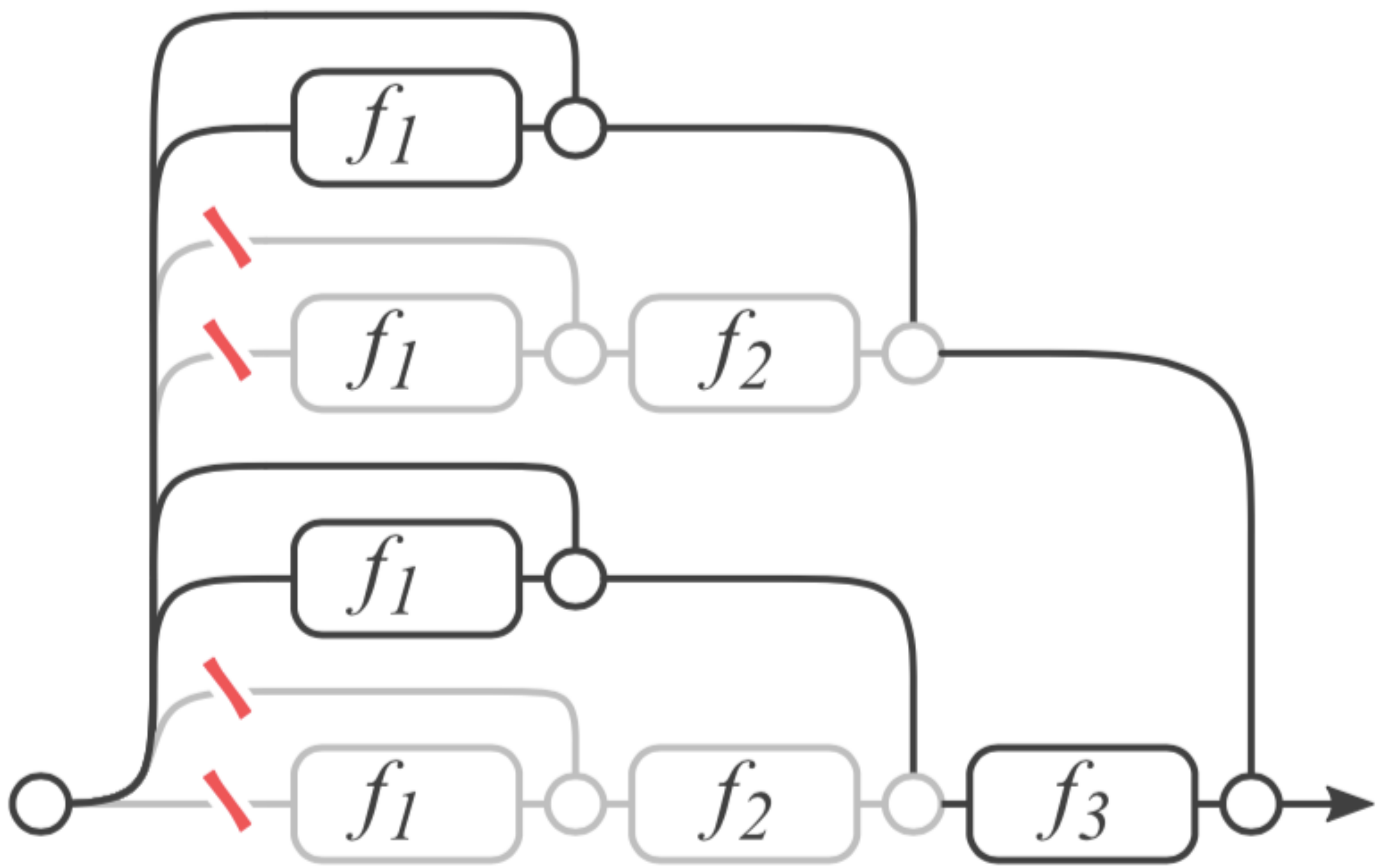
Exponential Multiplicity

- For each residual unit, either
  - information flows through it
  - or not — goes through the skip connection
- For N residual units, # paths = 2^N
- Hypothesis: ResNets = Exponential ensemble of shallow nets

Validation of Hypothesis

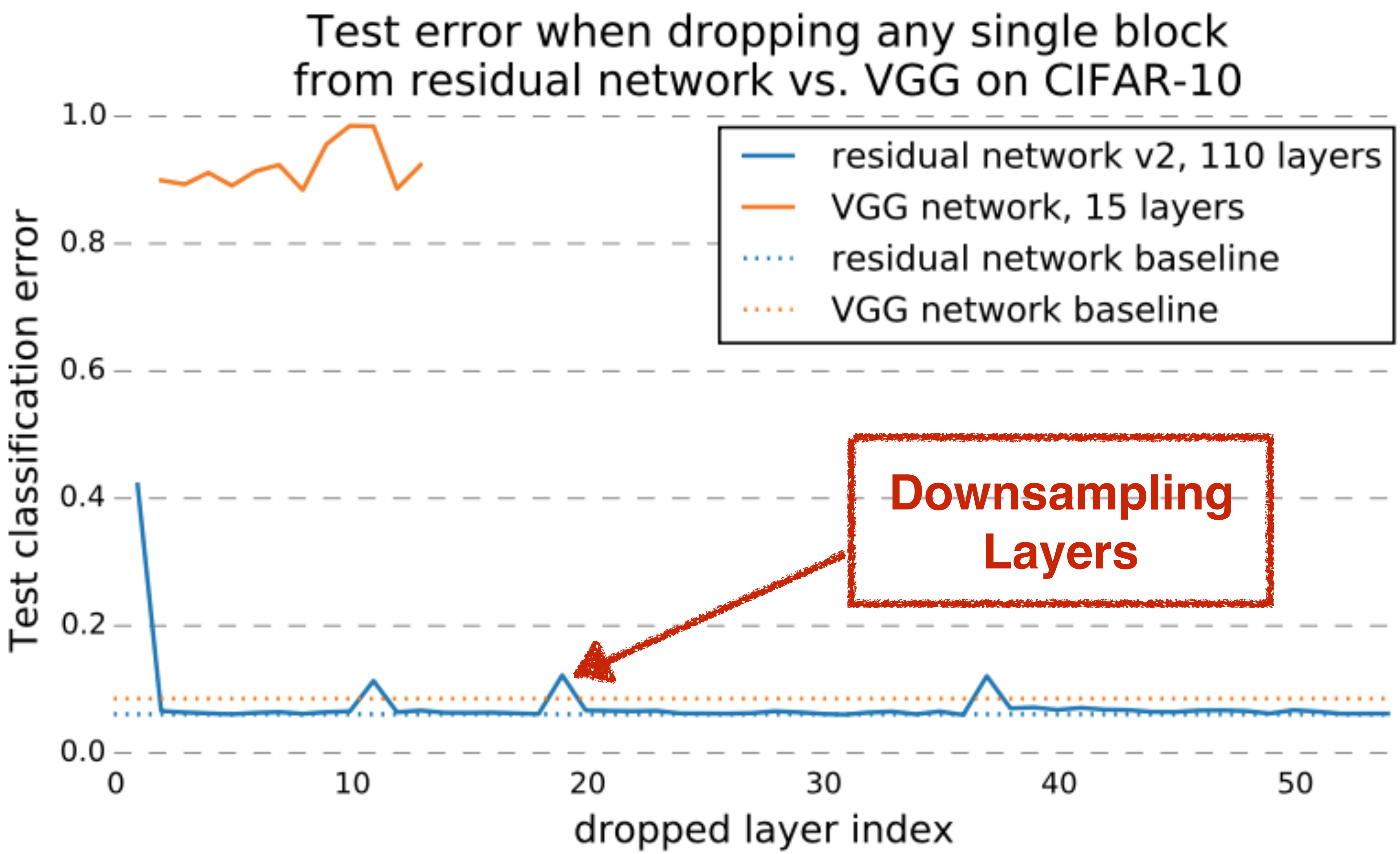
- Lesion studies
- Deleting layers at test time
- Deleting multiple residual modules at test time
- Reordering modules at test time

Deleting a Layer

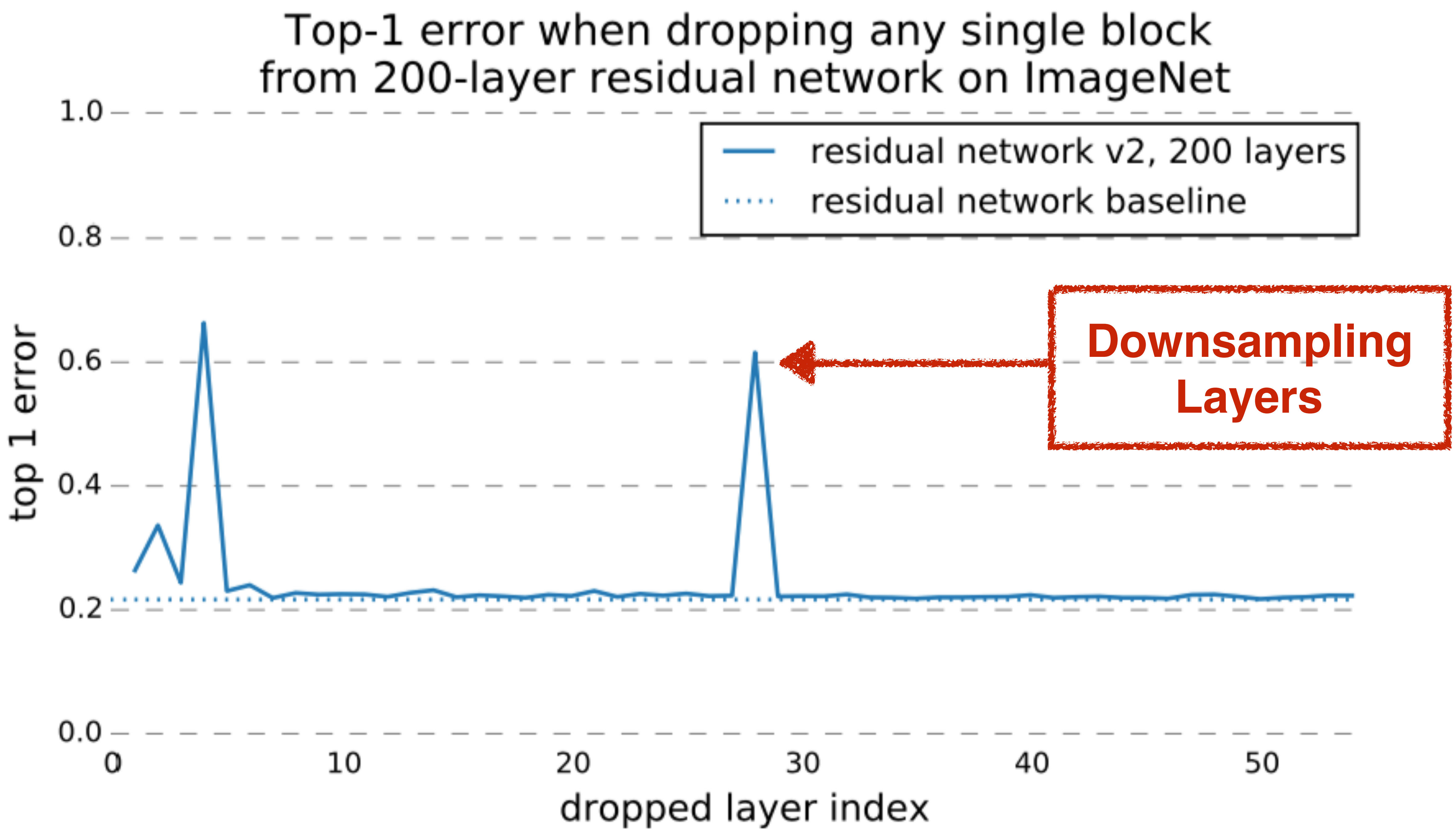


Equivalent to zeroing out half of the paths

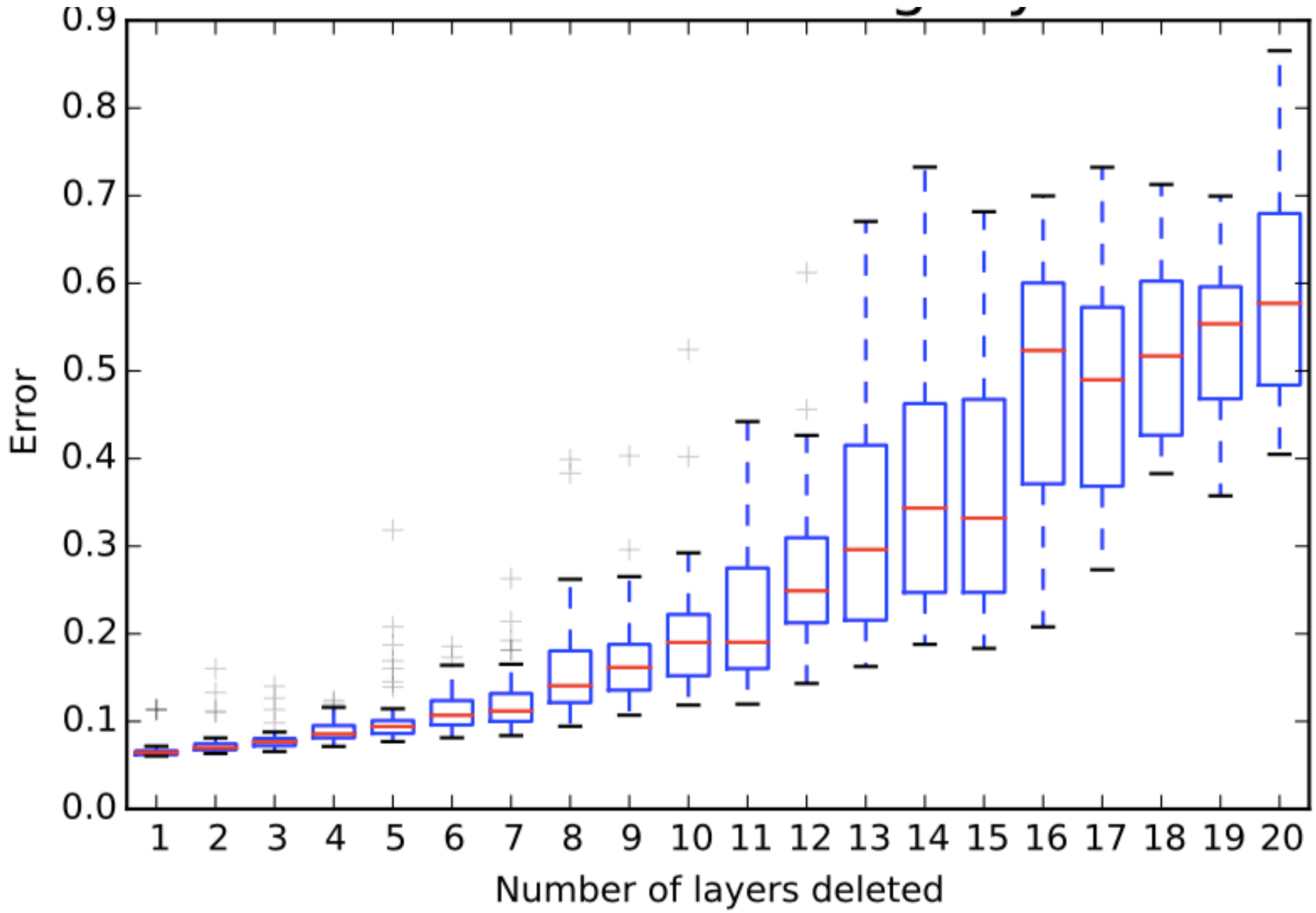
Deleting a Layer/Module resp.



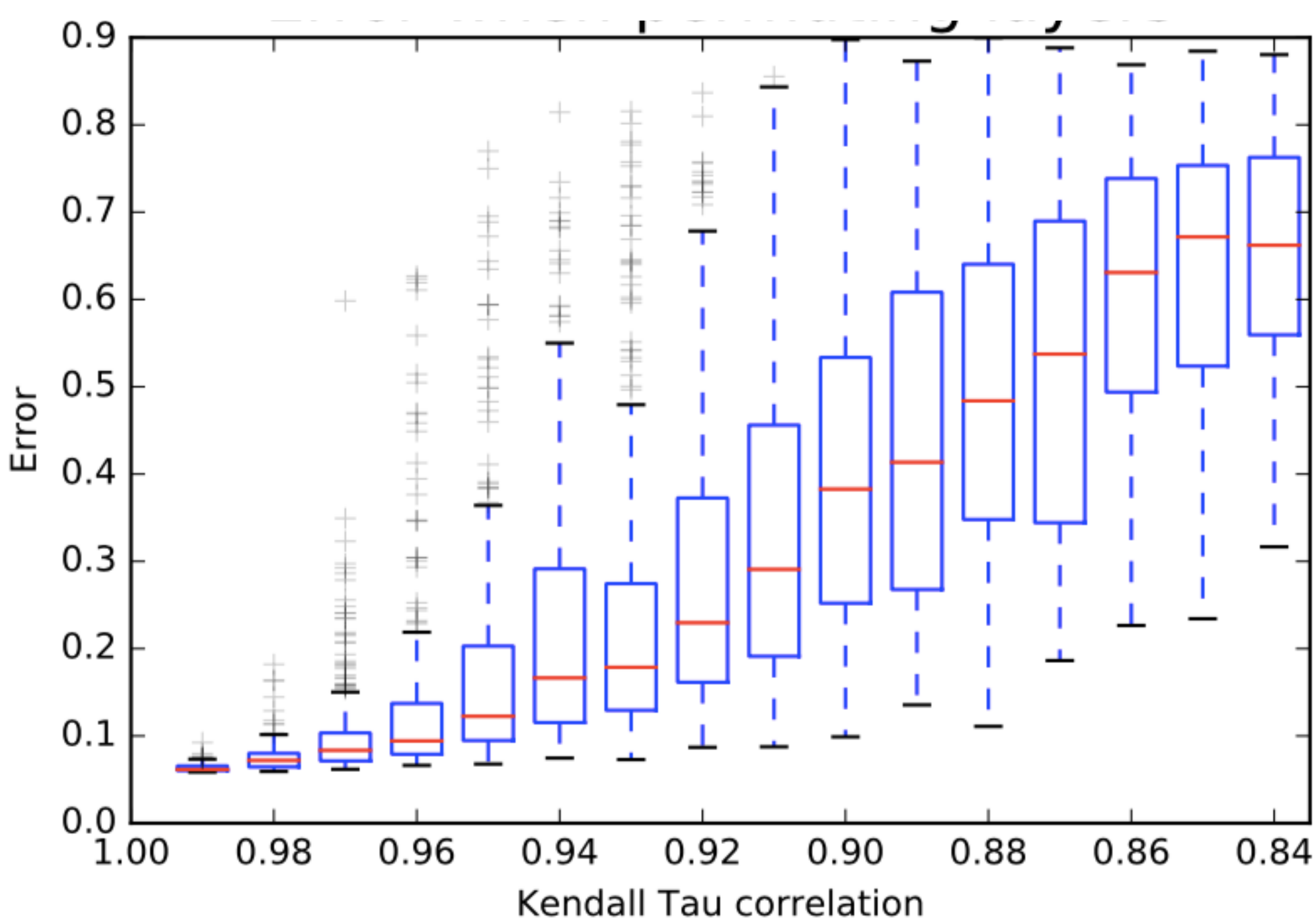
Deleting a Module



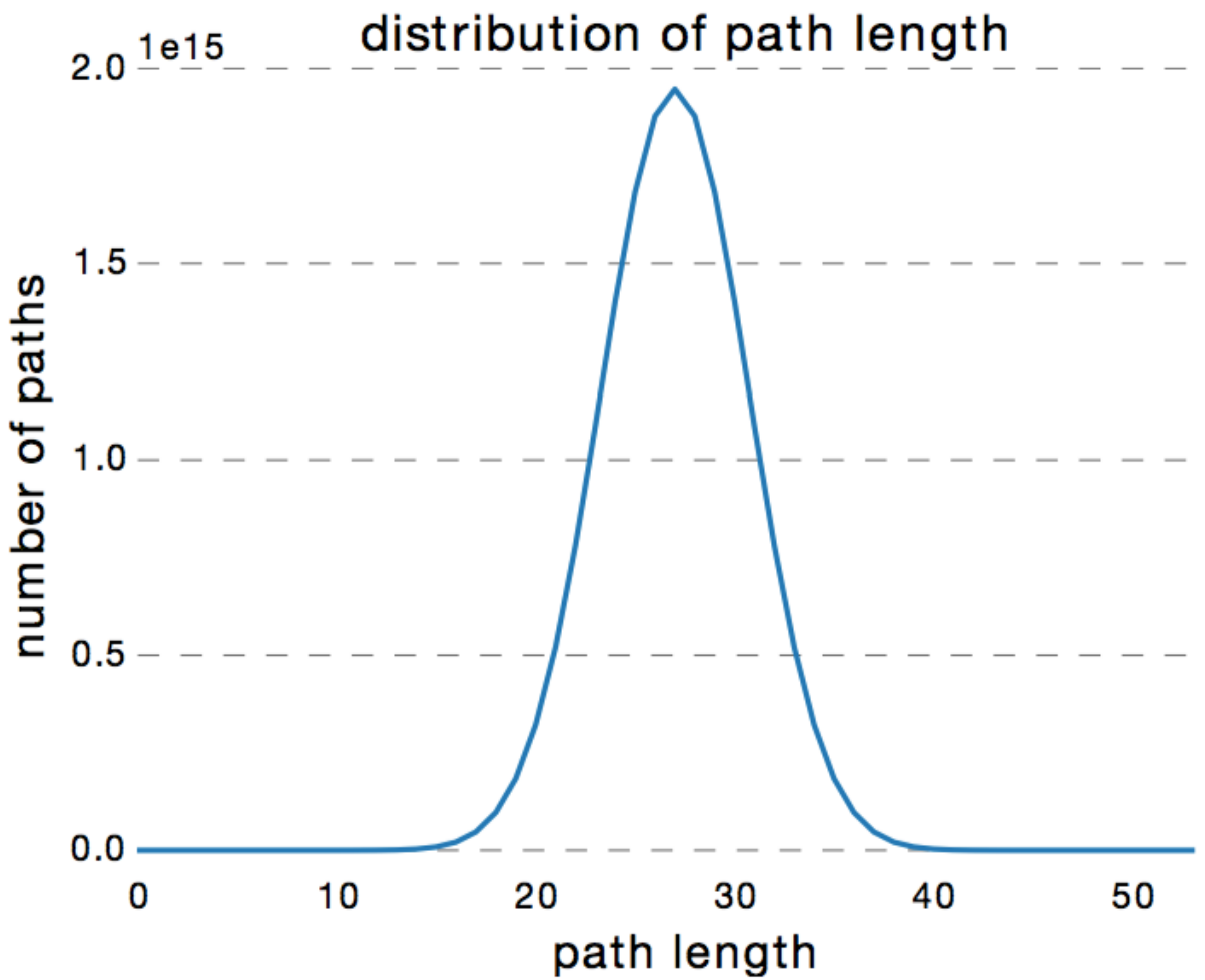
Deleting many Layers



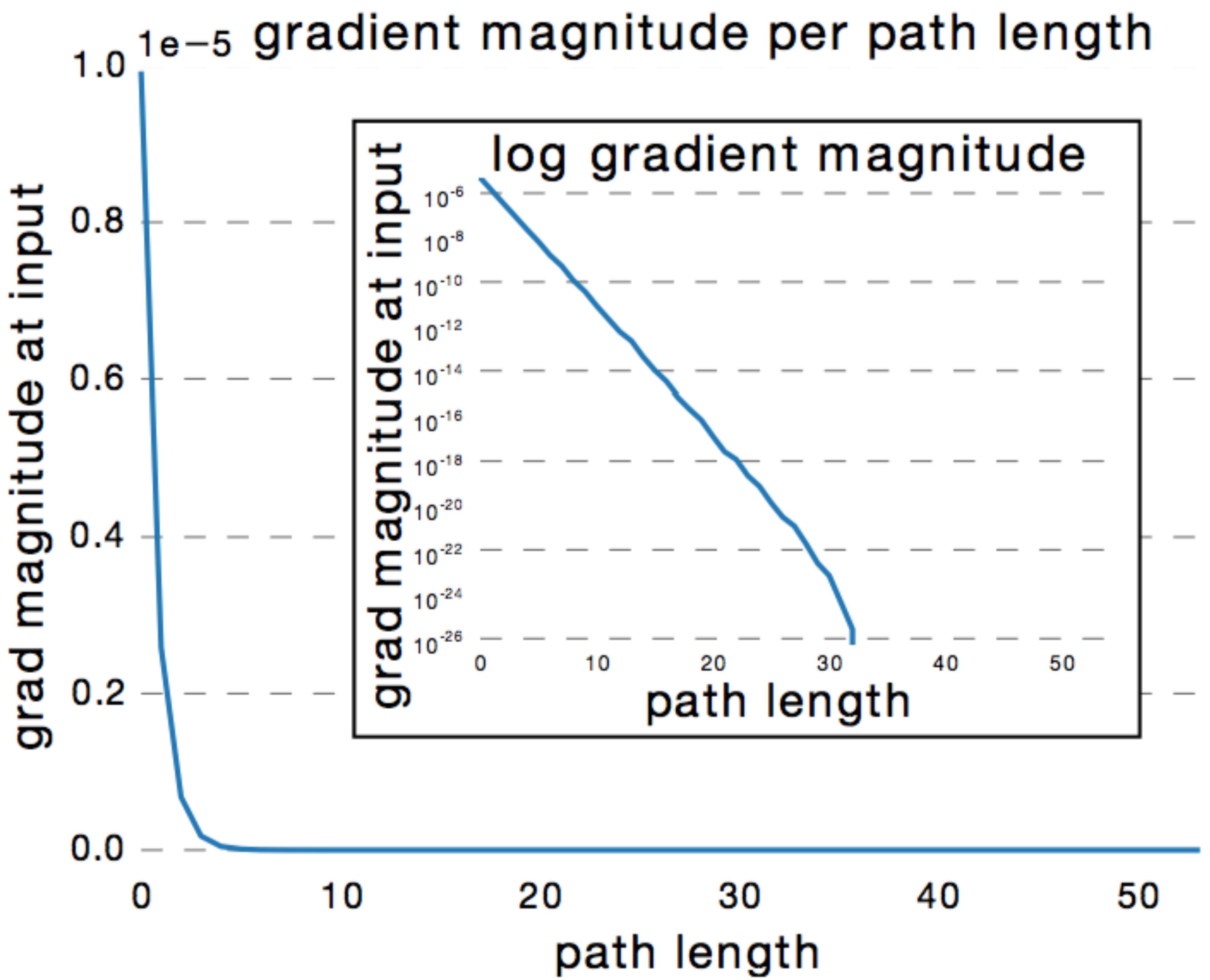
Reordering Layers



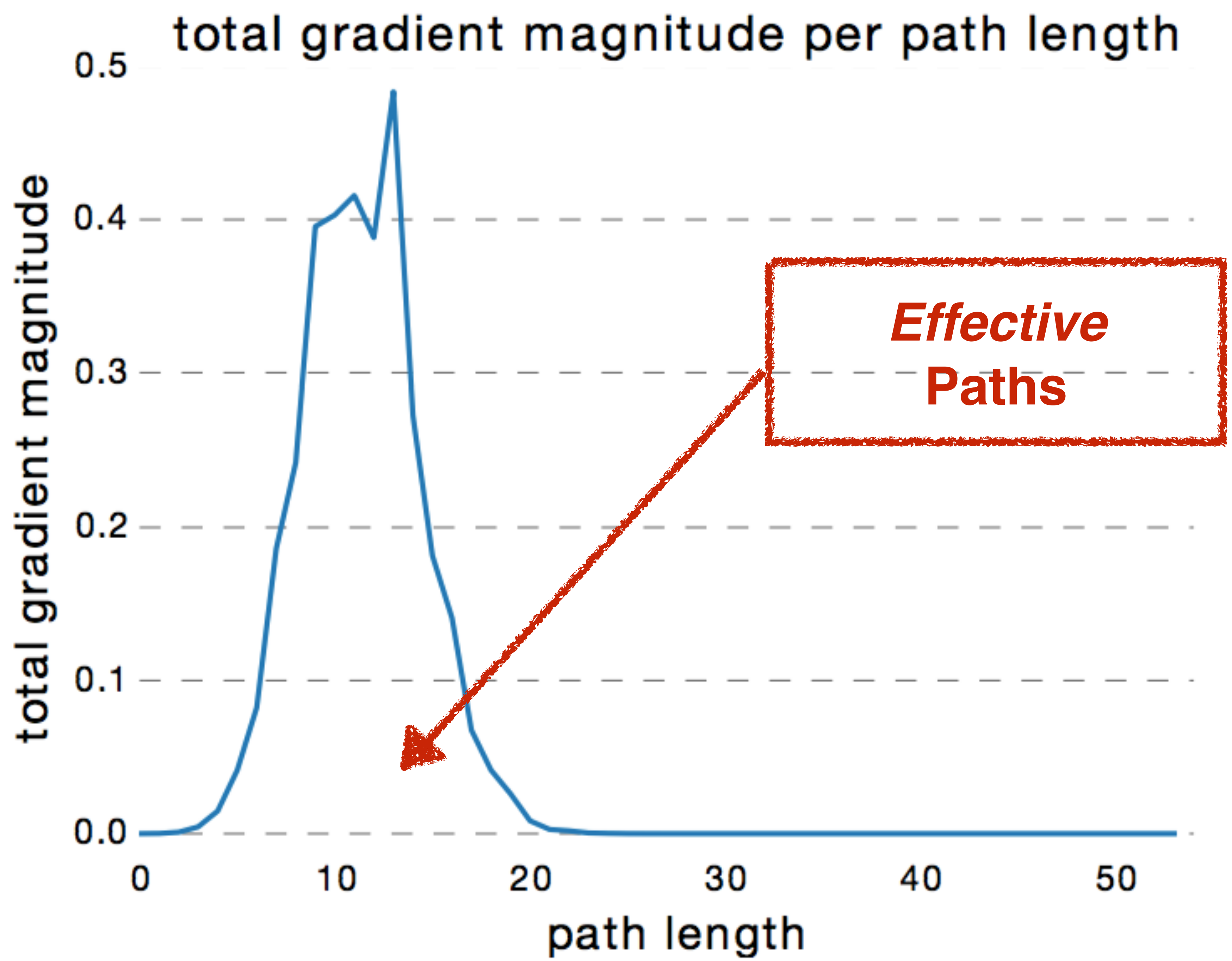
Properties: Path lengths



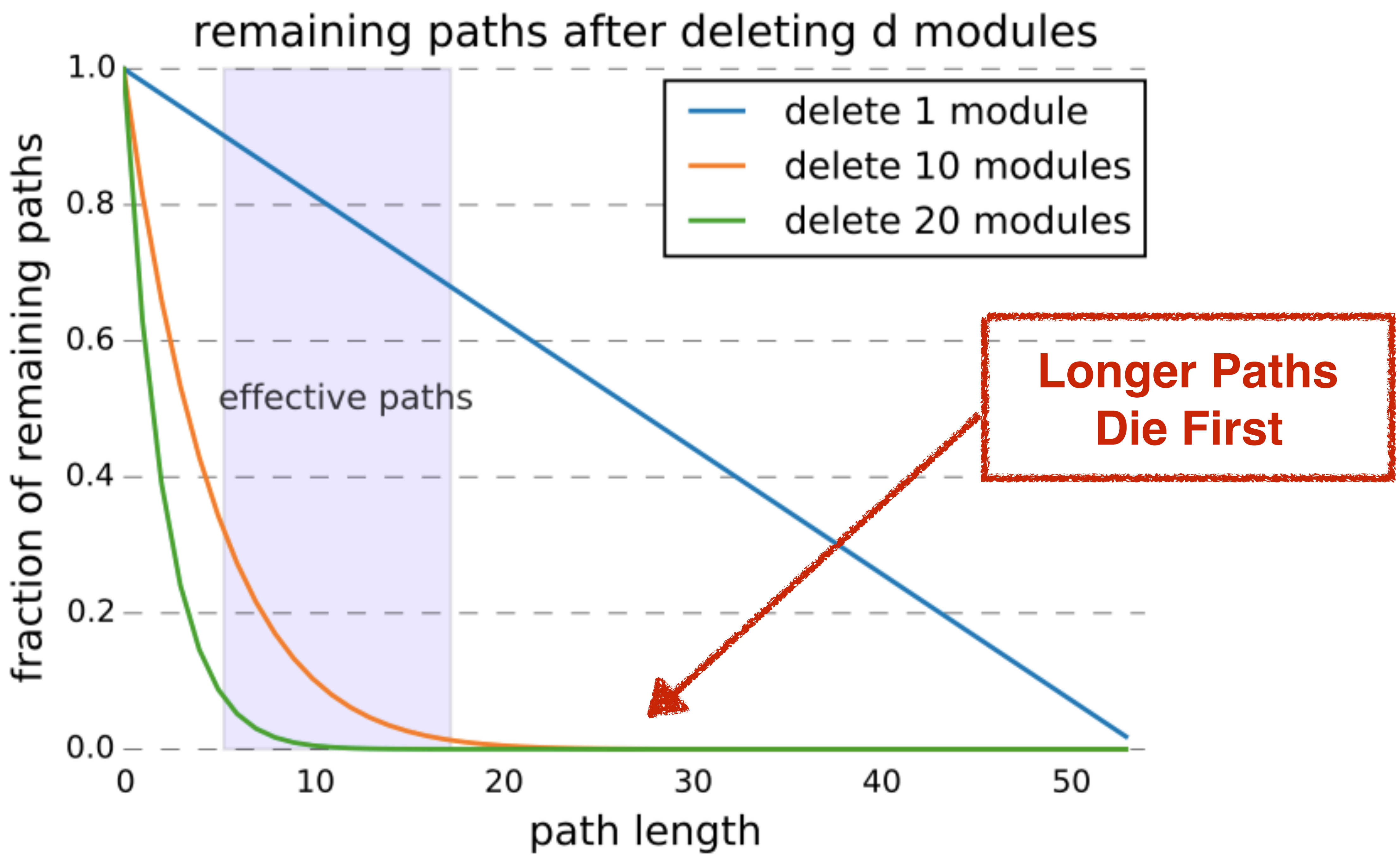
Properties: Gradient flow



Properties: Gradient/Path



Deleting Modules



Depth of ResNet

- Paths that contribute gradients are small
  - cf. full depth of the network
- Not just “going deeper”
- Multiplicity is an important factor
  - (expressibility in terms of # paths)