CNN Architectures

Neural Networks Design And Application

LetNet-5

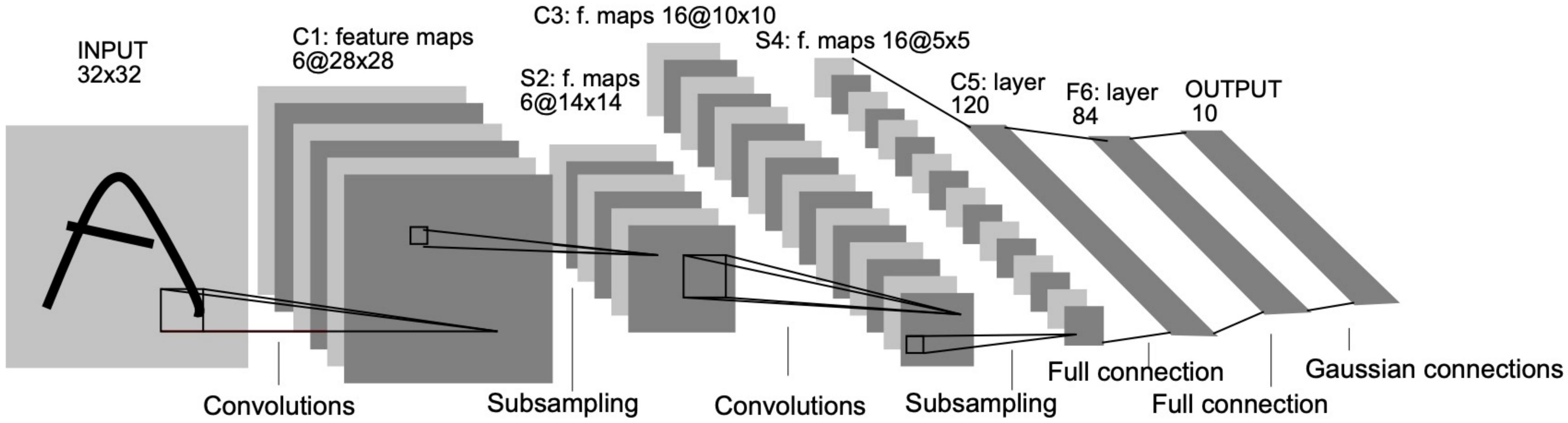
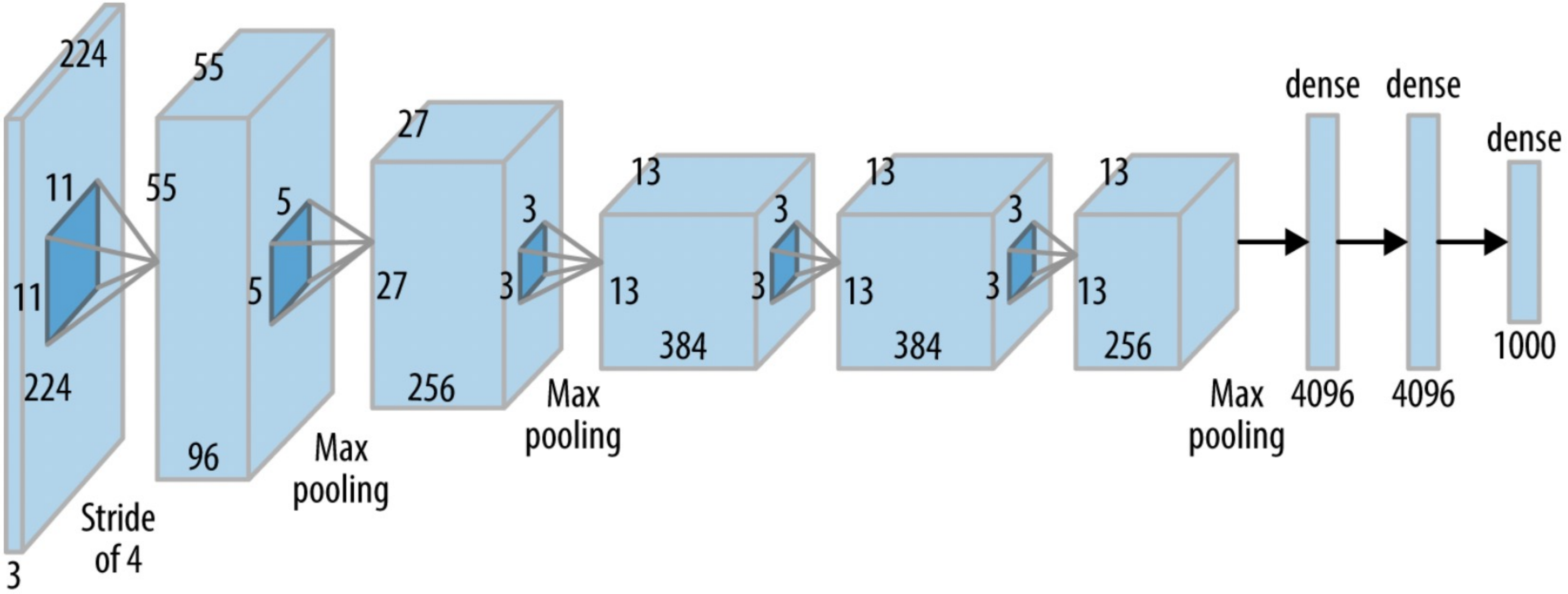


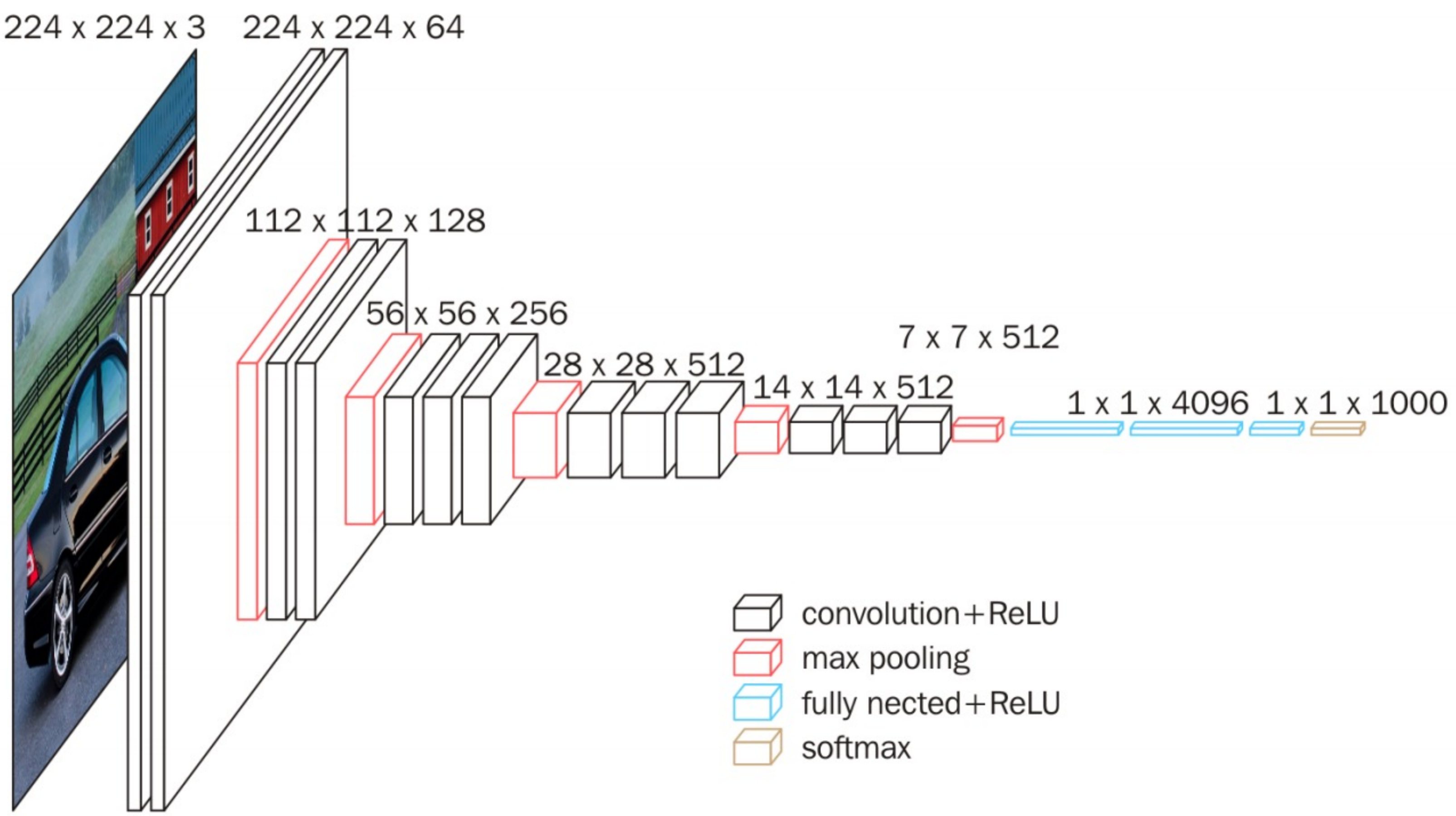
Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

AlexNet

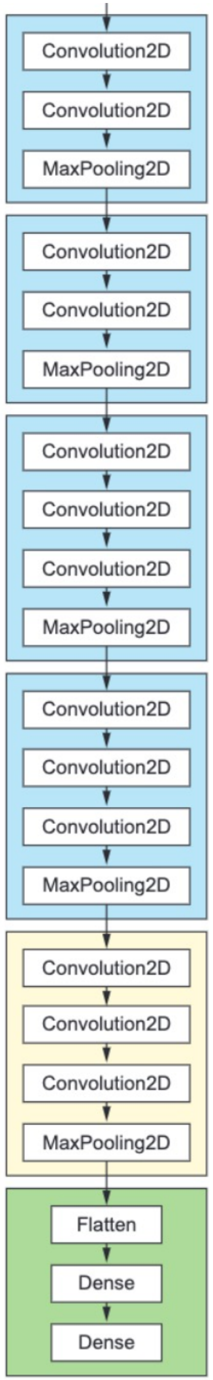


224
55
27
13
13
13
dense
dense
dense
11
55
5
3
3
3
11
5
27
3
13
3
13
3
13
384
384
256
1000
224
256
Max
pooling
Max
pooling
4096
4096
96
Max
pooling
Stride
of 4
3

VGG-16



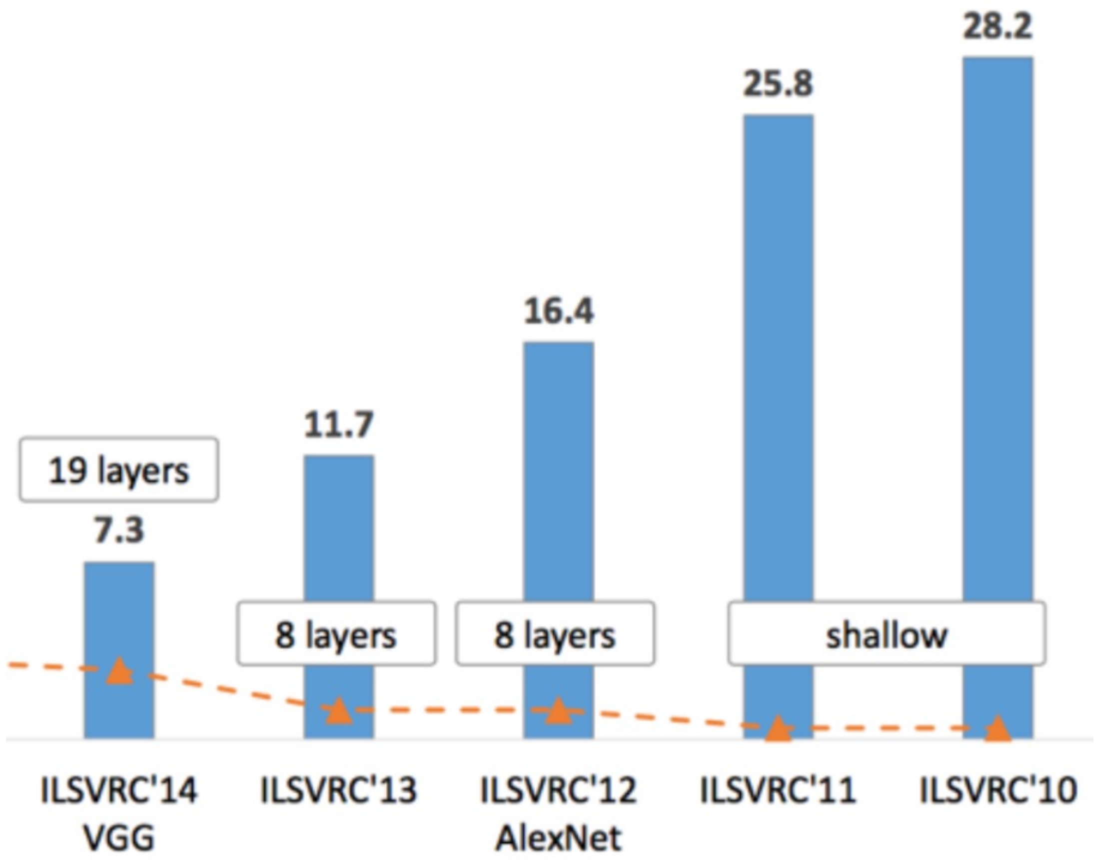
224 x 224 x 3
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
1 x 1 x 1000
convolution+ReLU
max pooling
fully nected+ReLU
softmax



Convolution2D
Convolution2D
MaxPooling2D
Convolution2D
Convolution2D
MaxPooling2D
Convolution2D
Convolution2D
Convolution2D
MaxPooling2D
Convolution2D
Convolution2D
Convolution2D
MaxPooling2D
Convolution2D
Convolution2D
Convolution2D
MaxPooling2D
Flatten
Dense
Dense

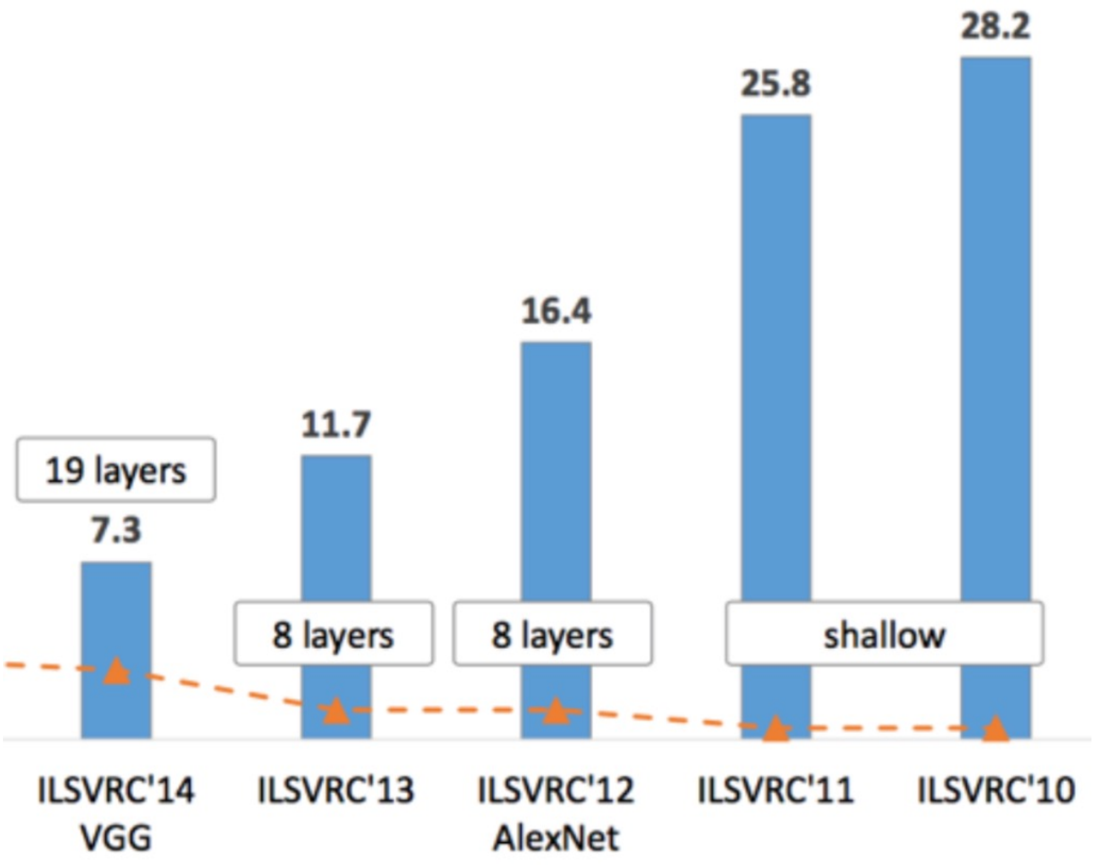
[VGG]

ImageNet competition



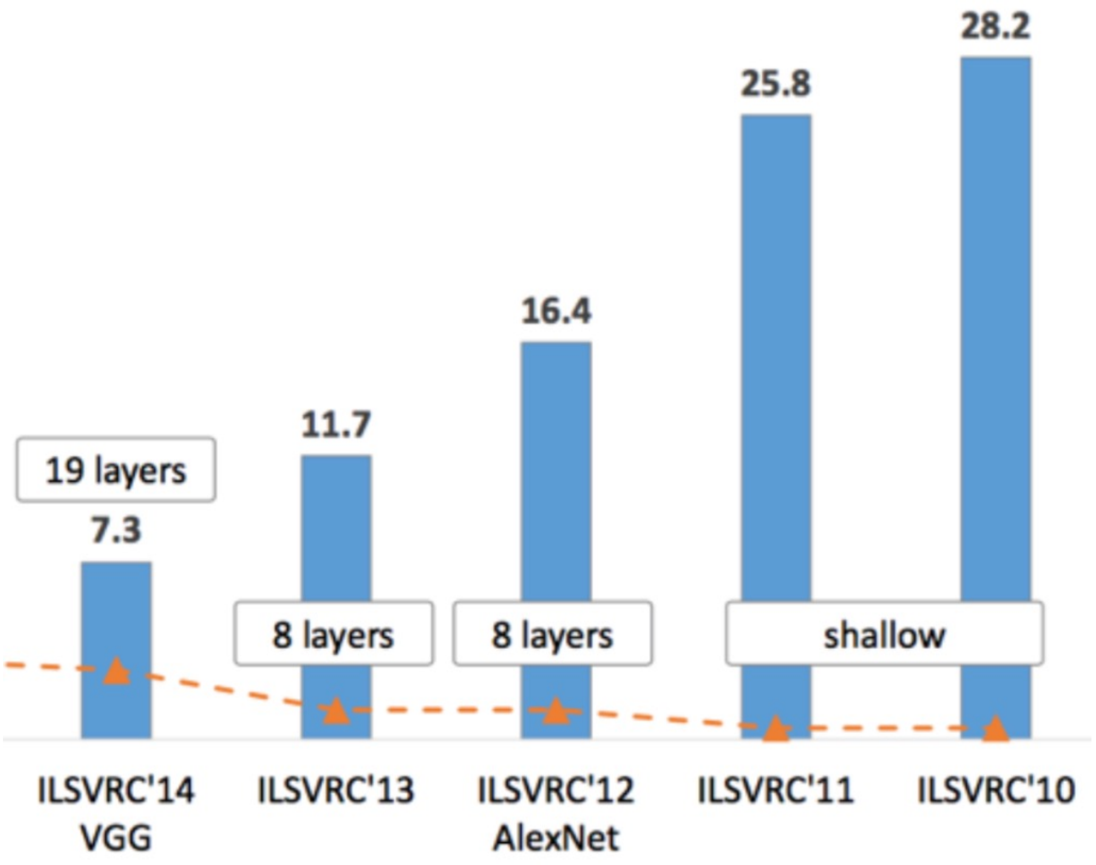
ImageNet competition

More layers



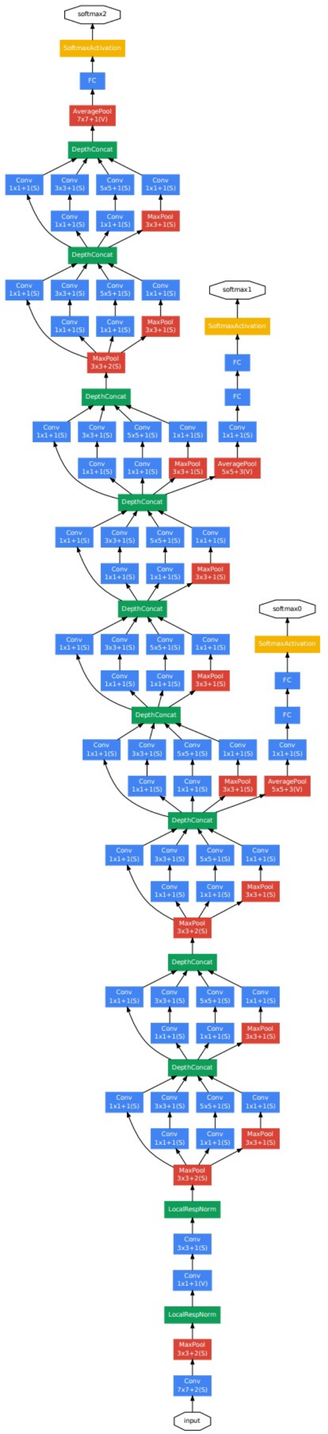
ImageNet competition

More layers

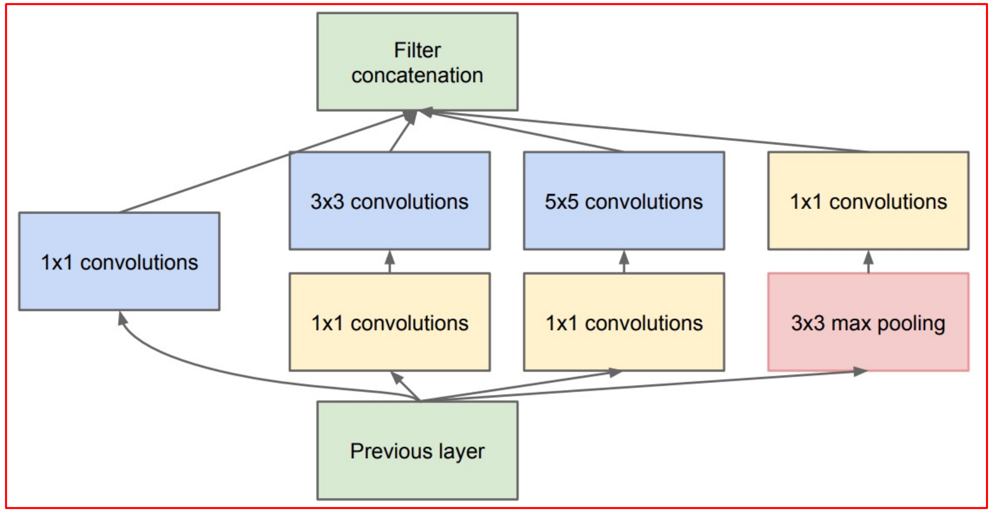


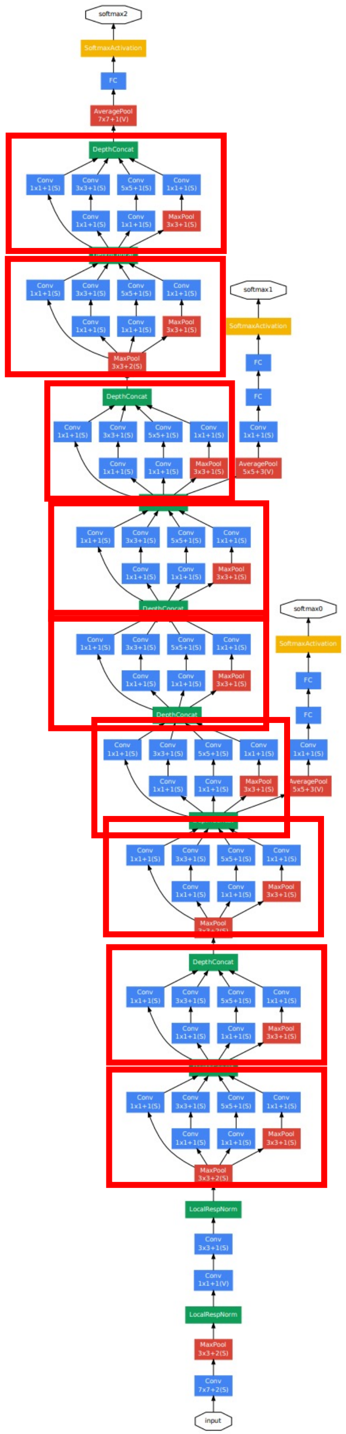
Better performance

Inception (GoogLeNet)

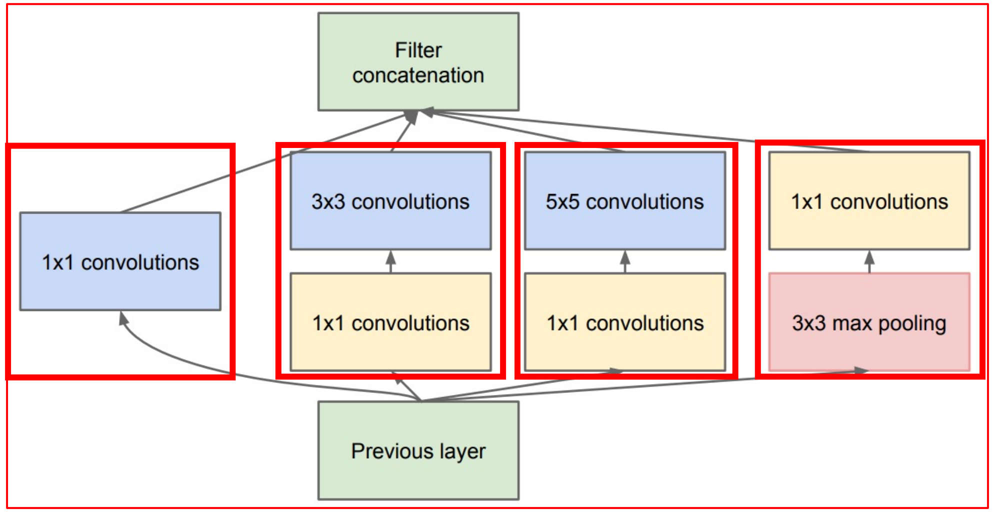


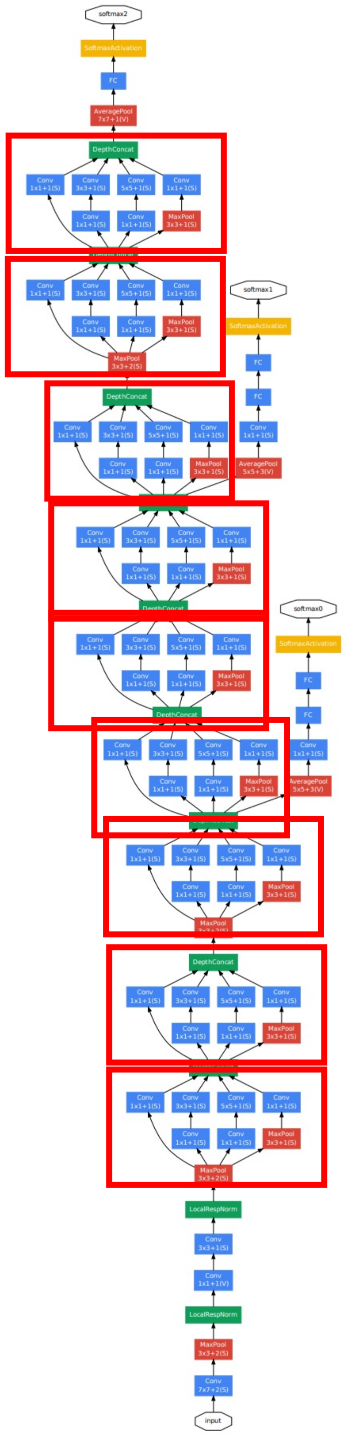
Inception (GoogLeNet)



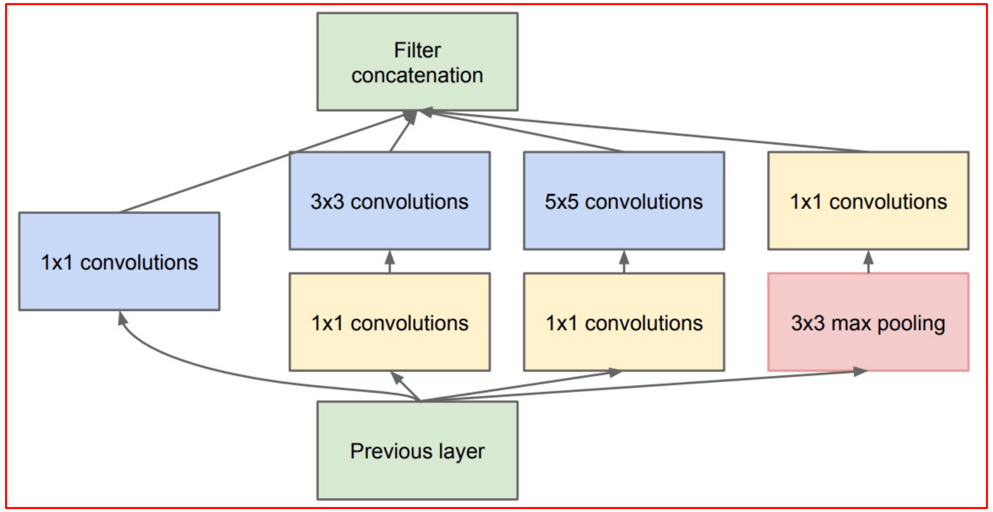


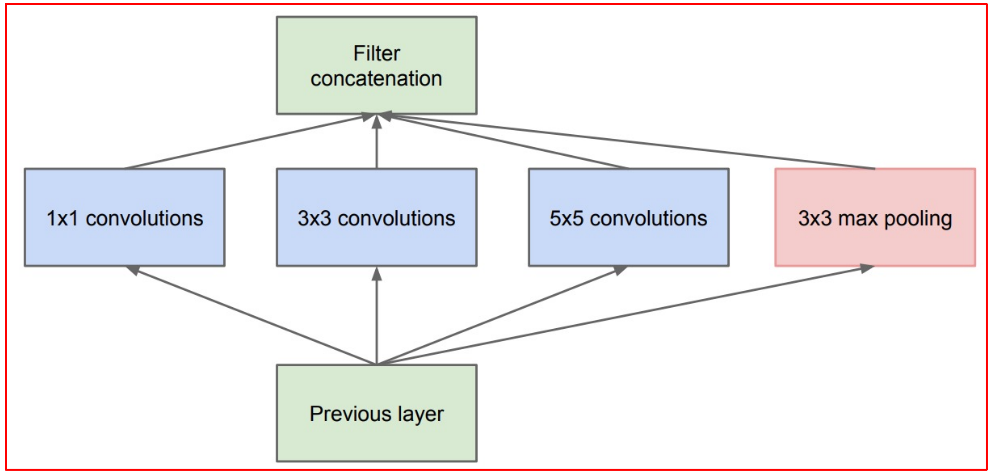
Inception (GoogLeNet)

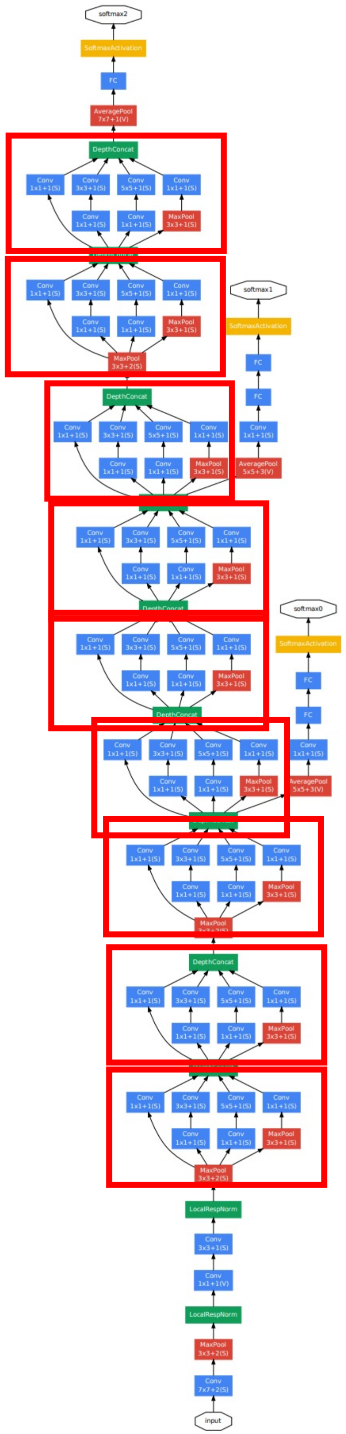




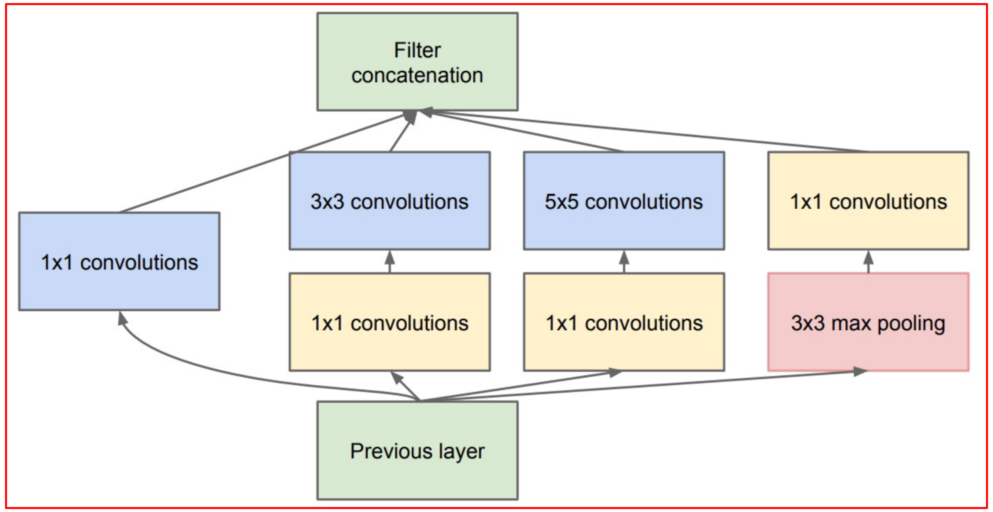
Inception (GoogLeNet)



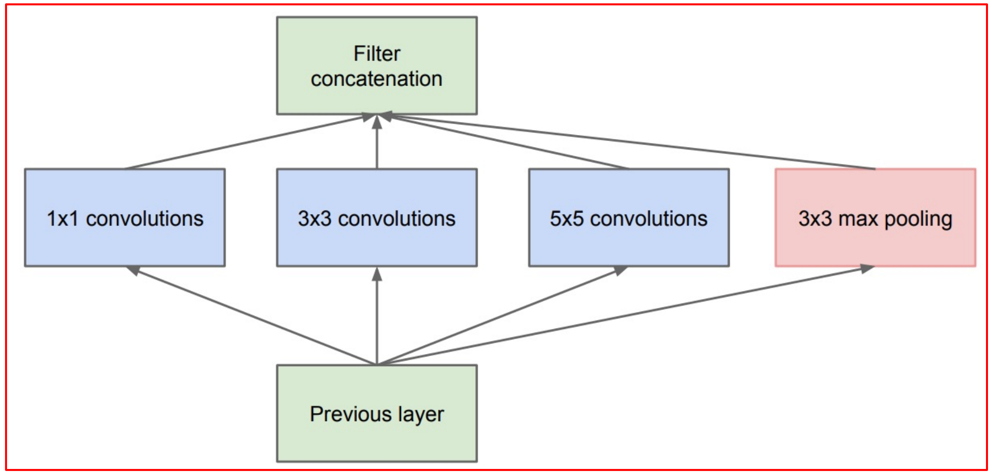


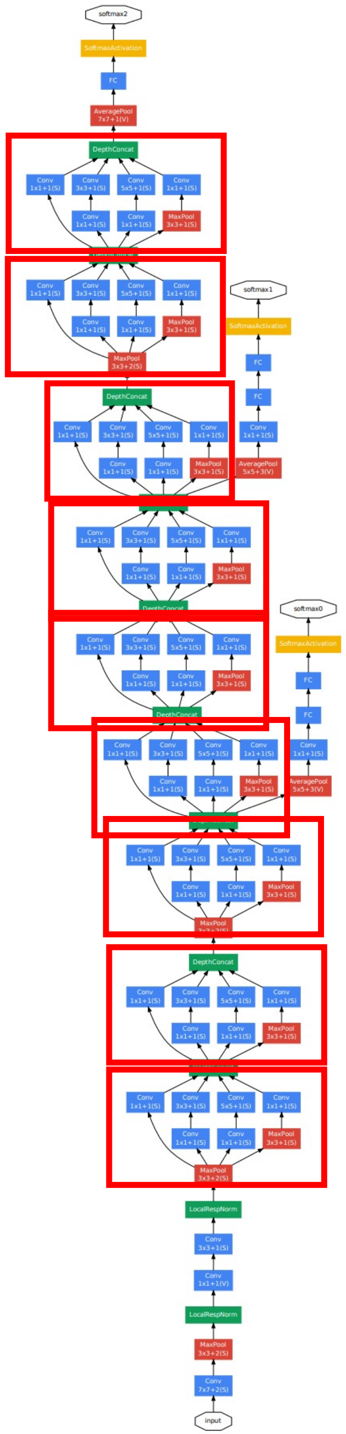


Inception (GoogLeNet)

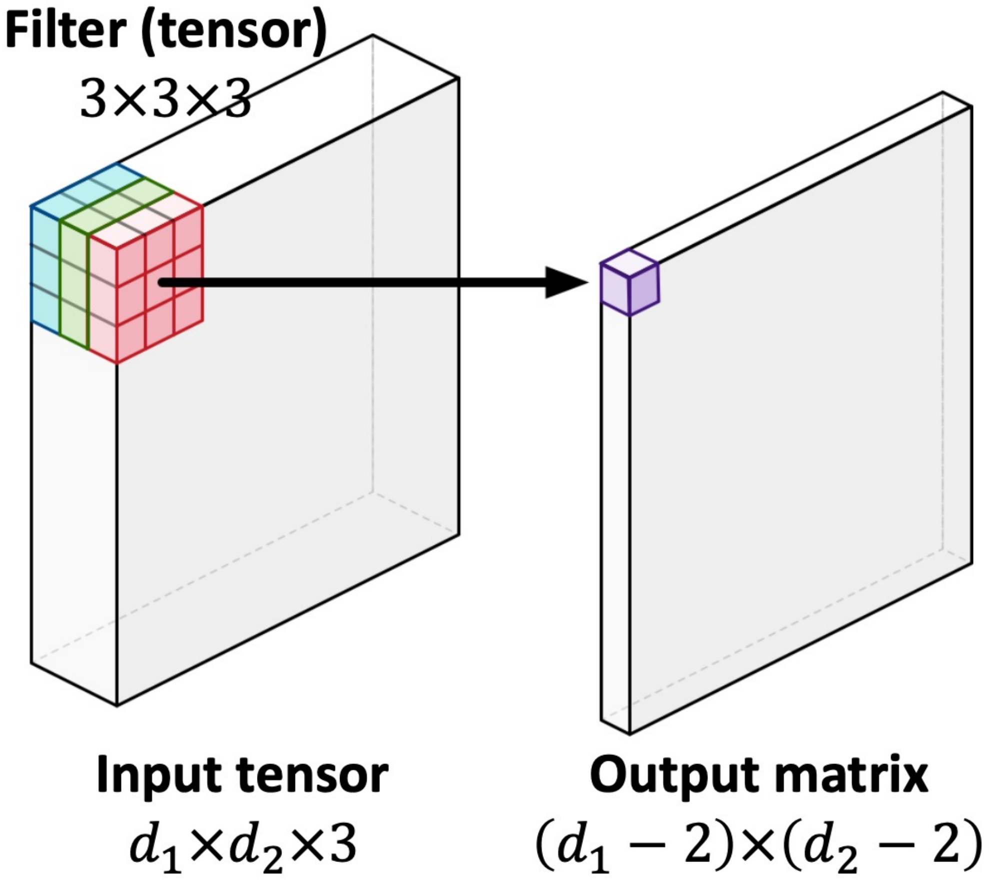


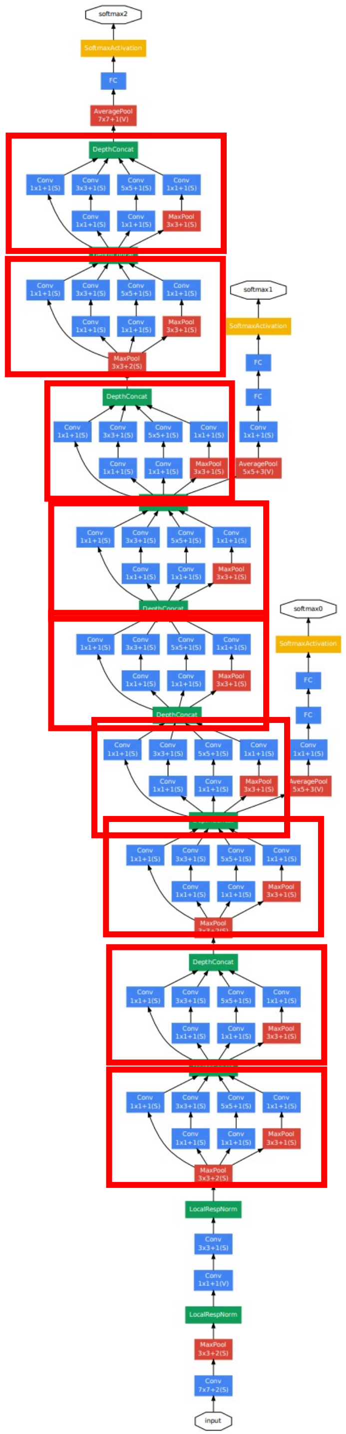
Q: difference between those two variants?



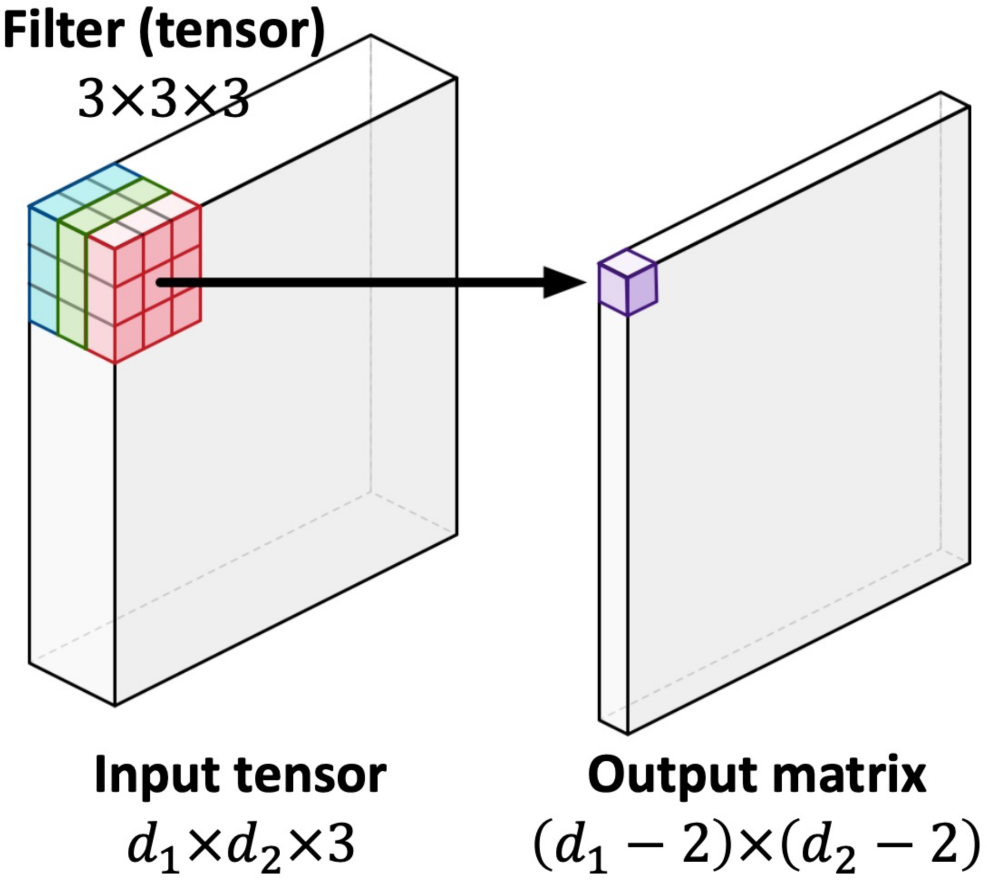


Inception (GoogLeNet)



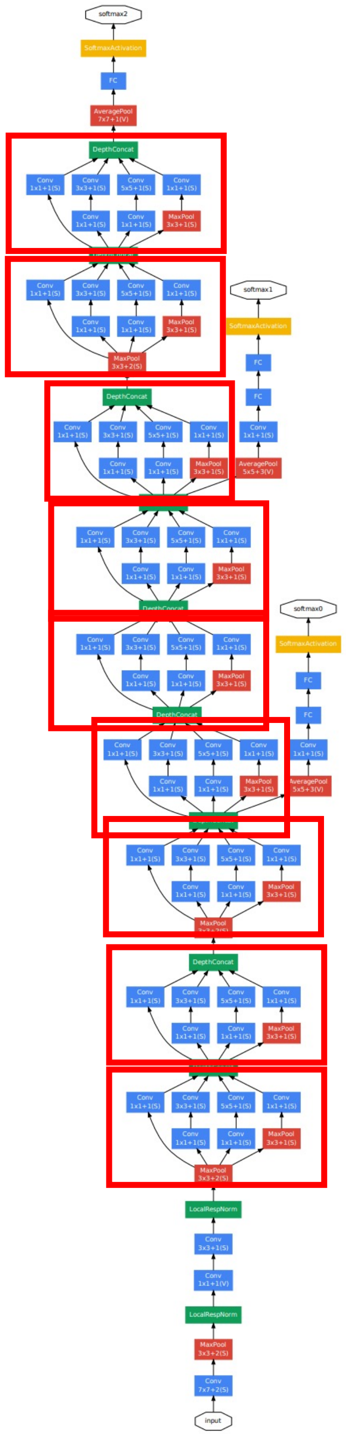


Inception (GoogLeNet)

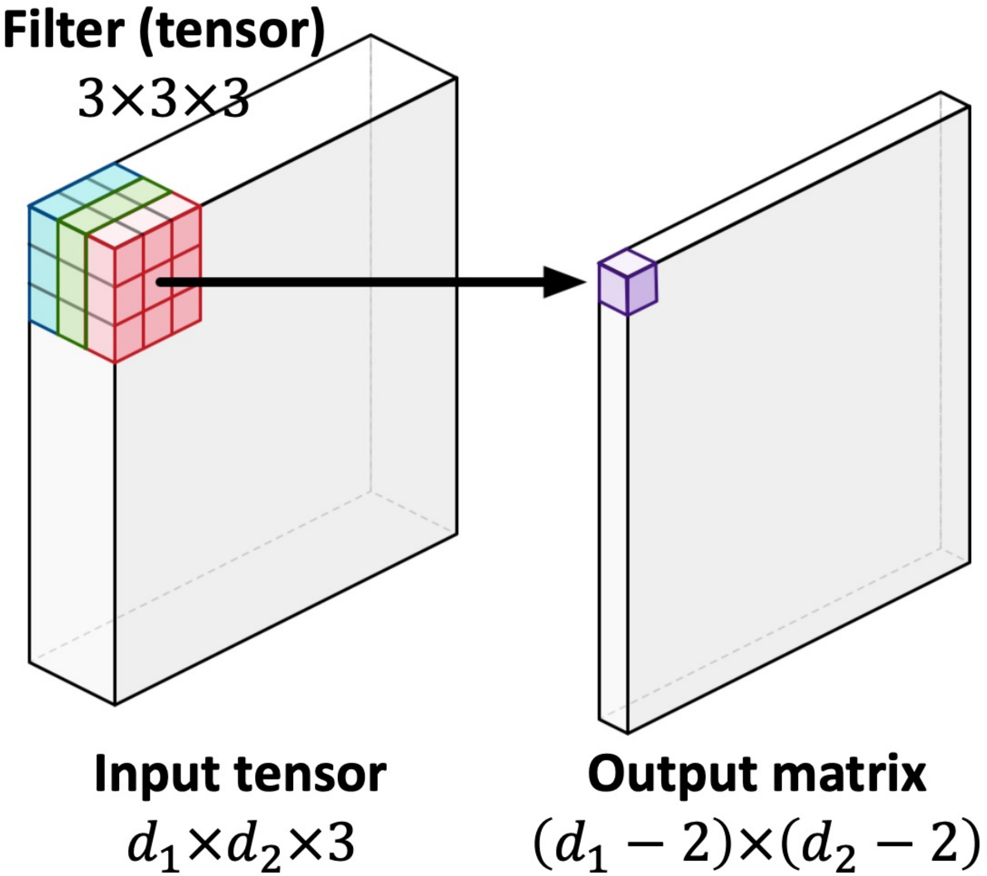


a tensor → a matrix (channel)

a filter/kernel

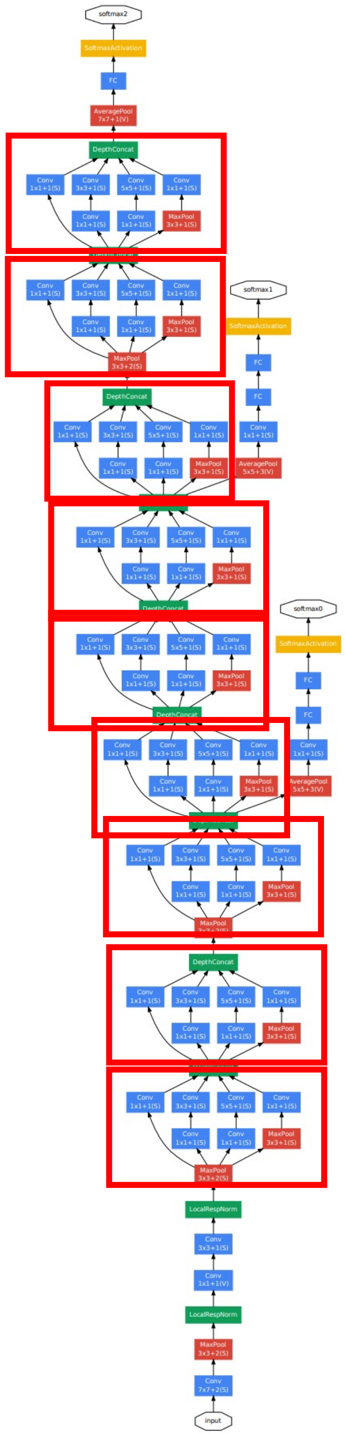


Inception (GoogLeNet)



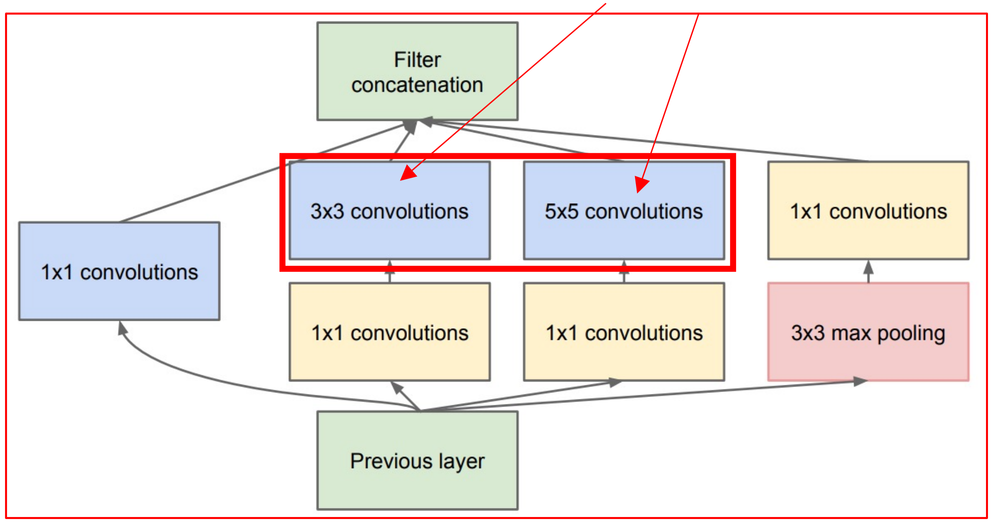
a tensor → m matrices (channels)

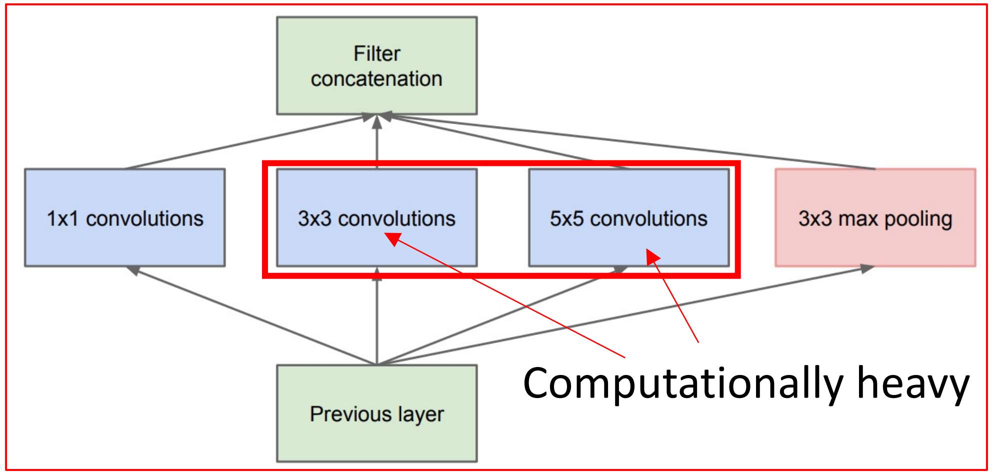
m filter/kernel

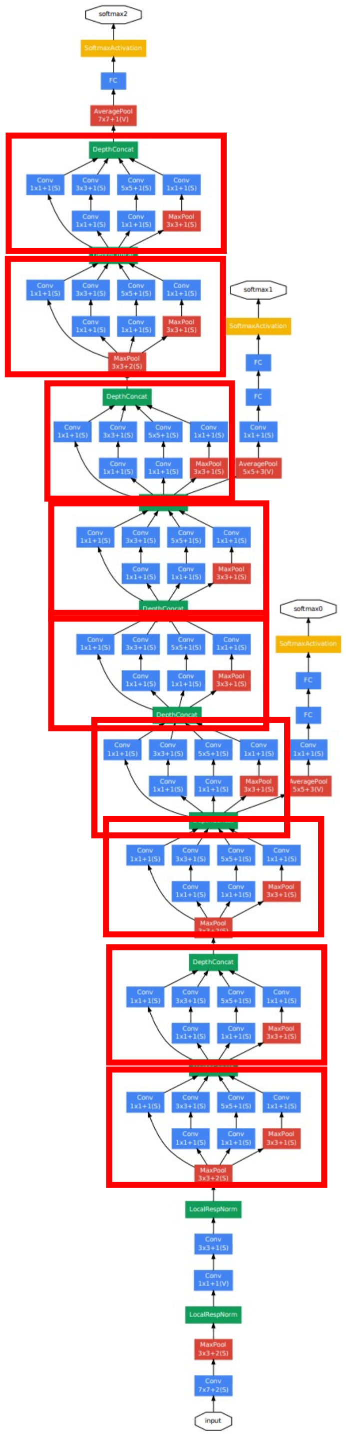


Inception (GoogLeNet)

Computationally heavy

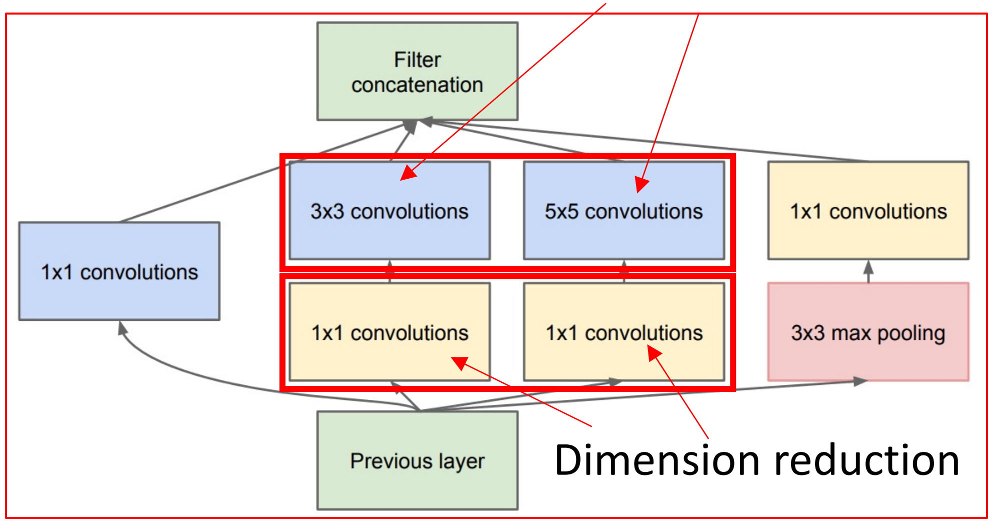


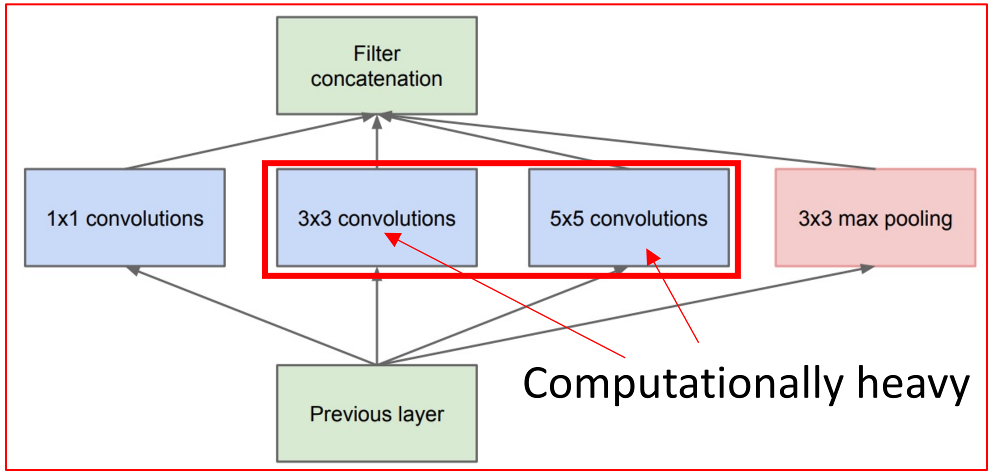


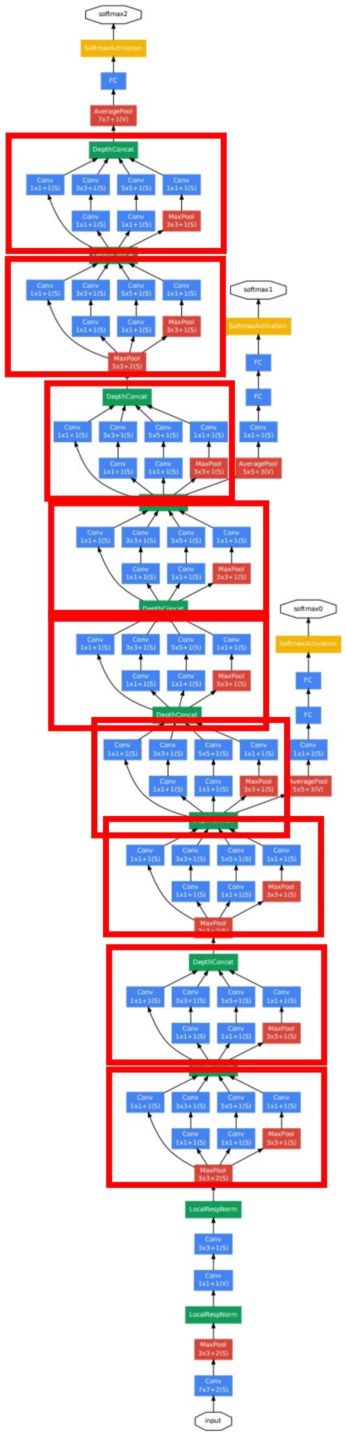


Inception (GoogLeNet)

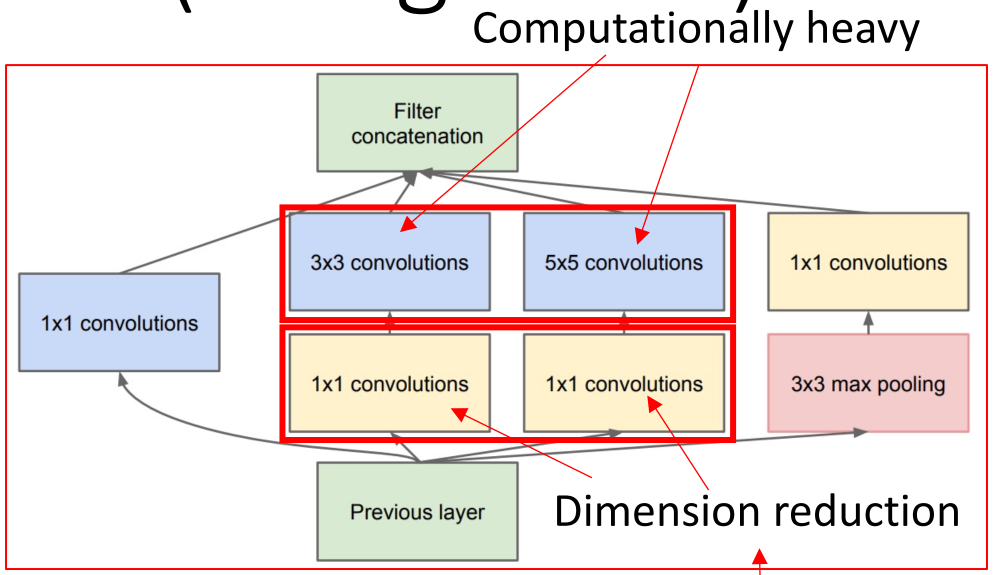
Computationally heavy



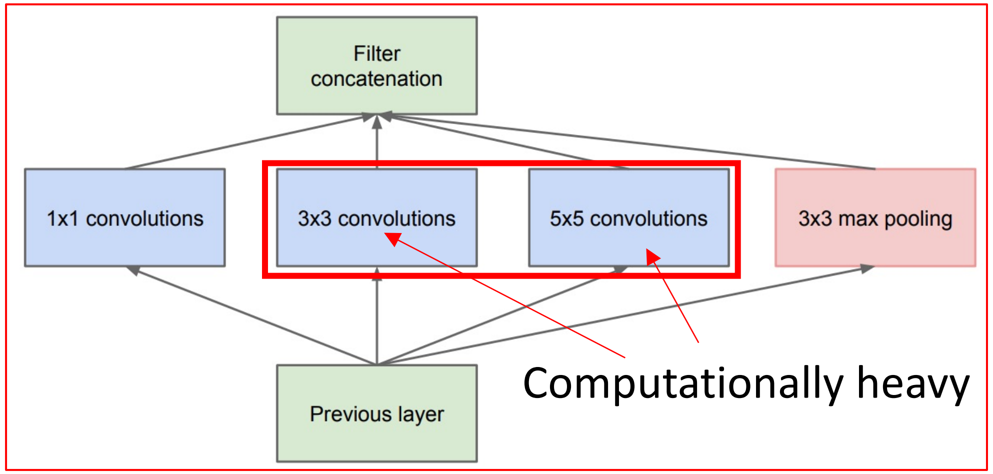


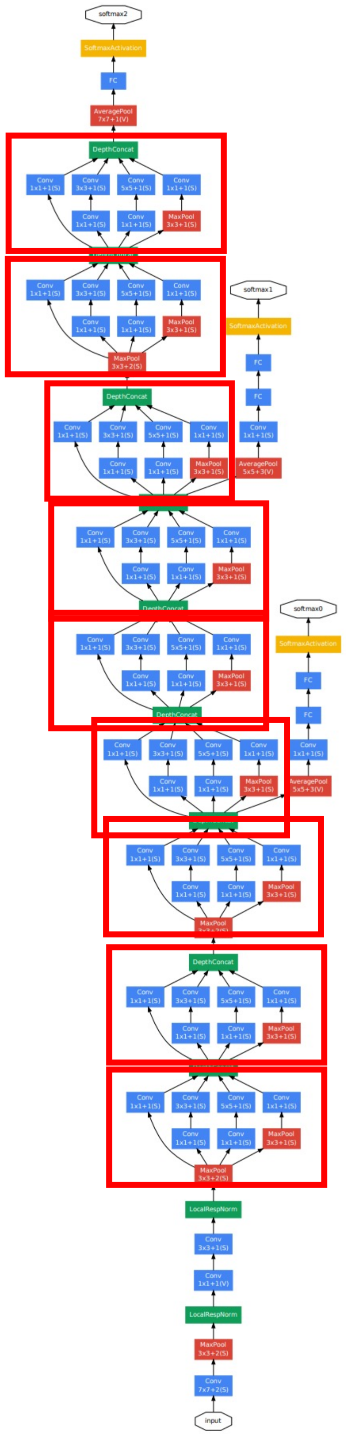


Inception (GoogLeNet)

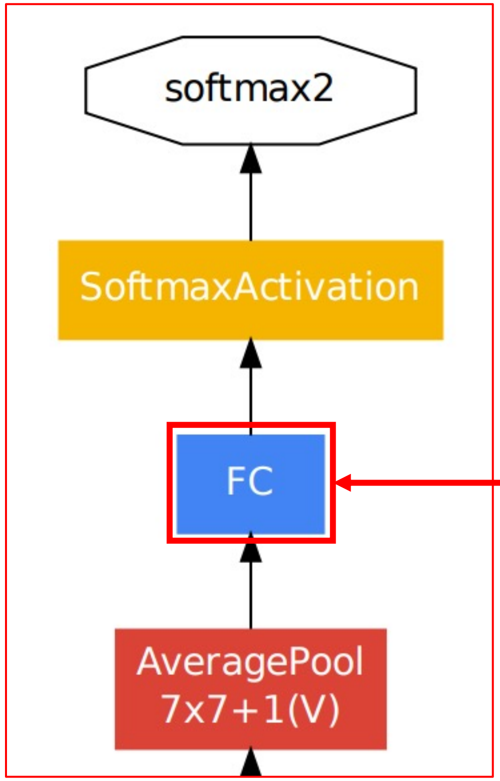


Less channels

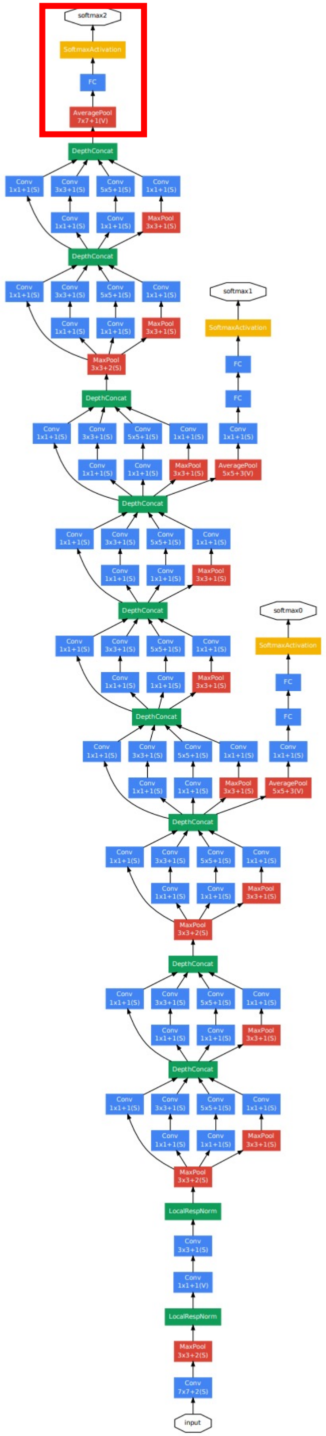




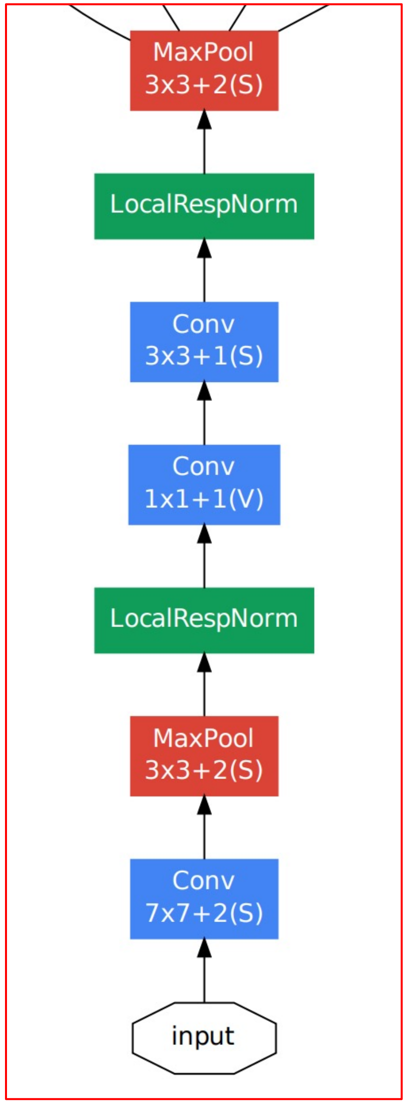
Inception (GoogLeNet)

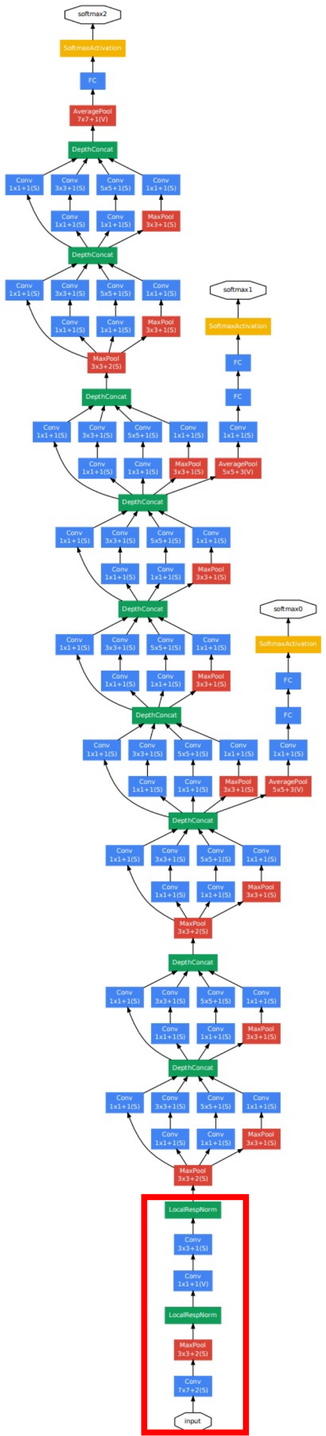


a linear model

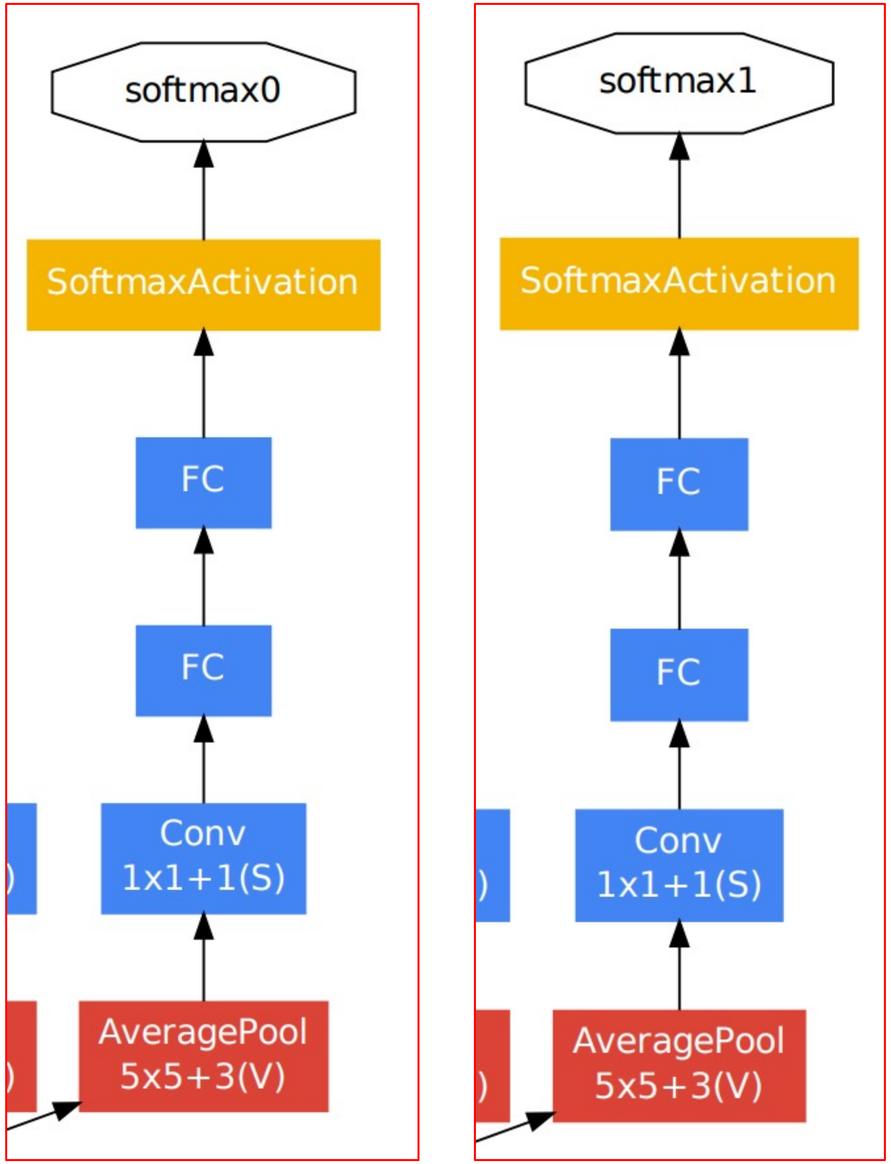


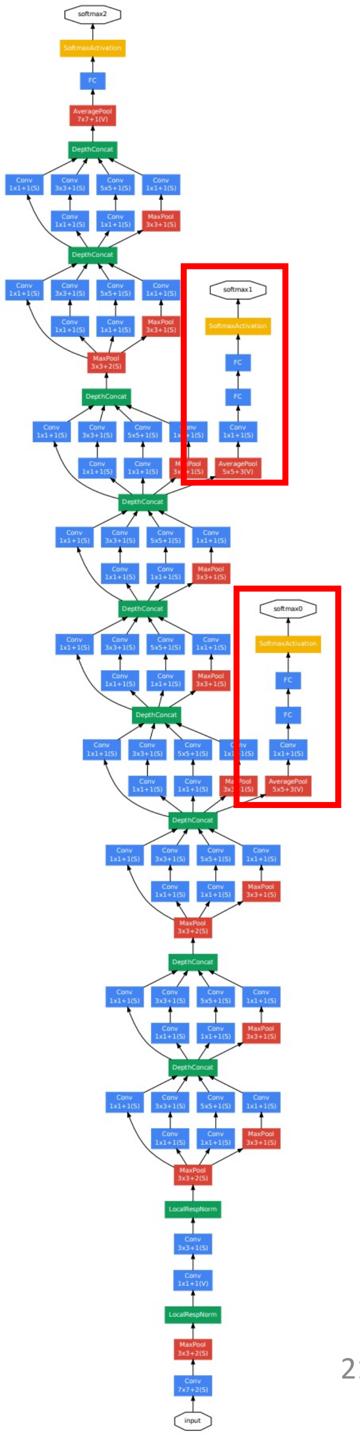
Inception (GoogLeNet)



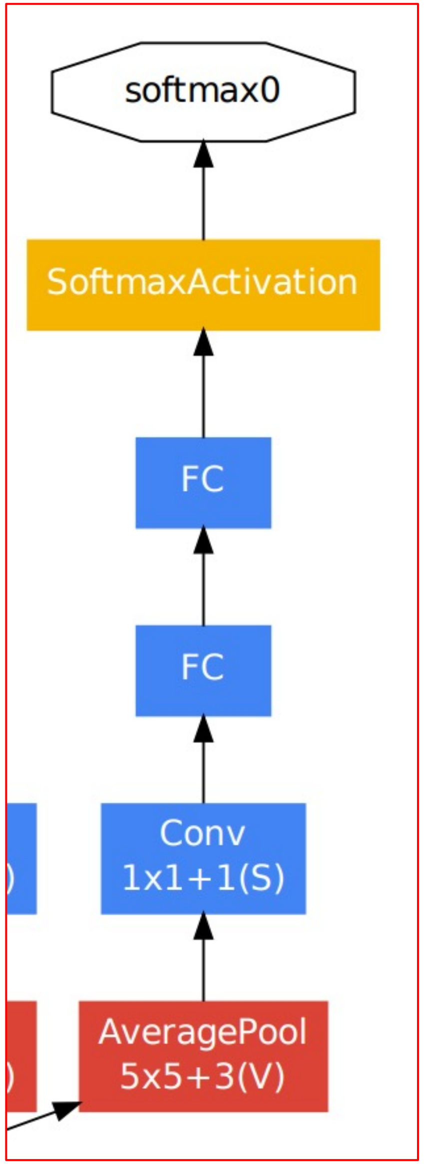


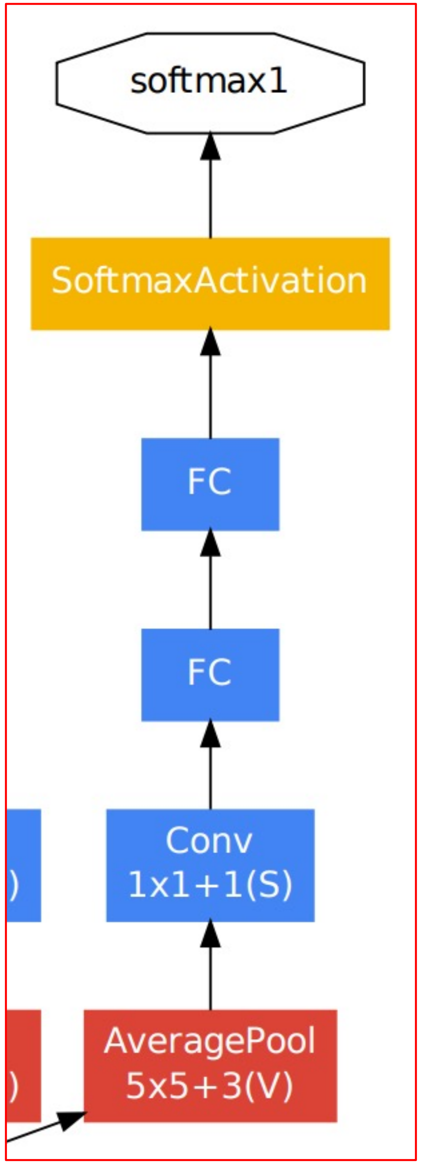
Inception (GoogLeNet)

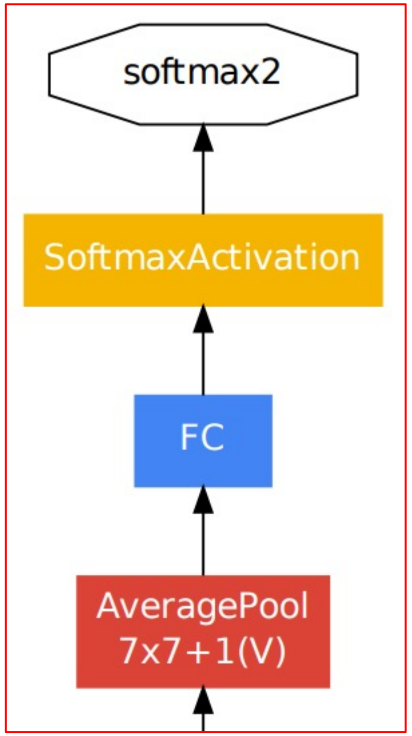


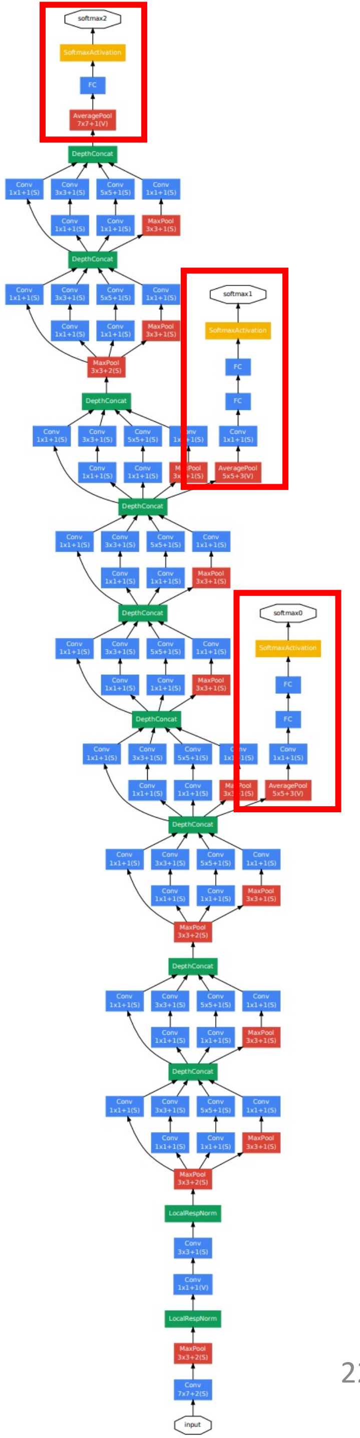


Inception (GoogLeNet)



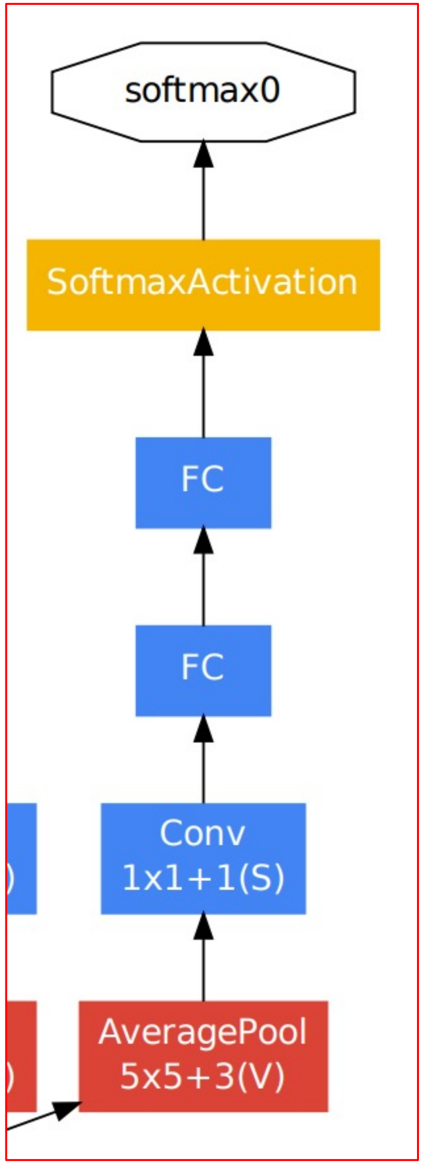


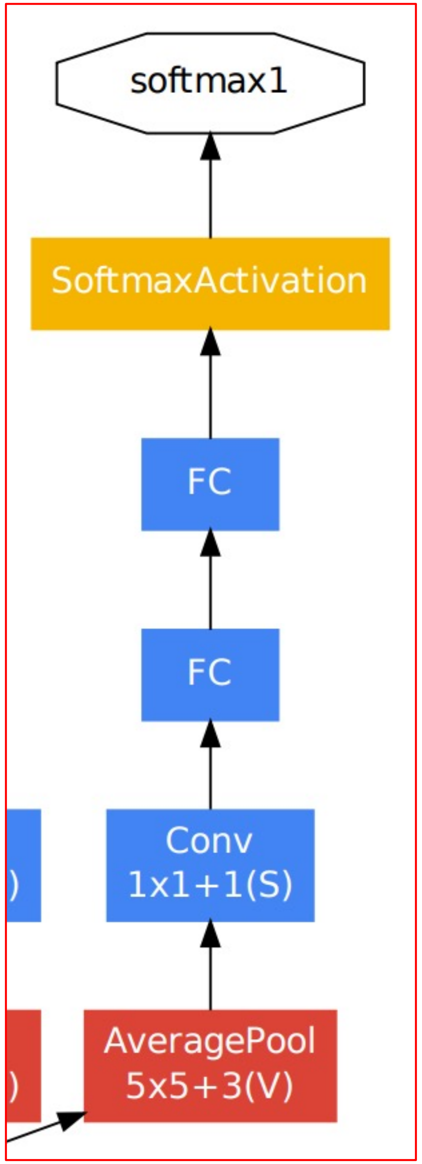


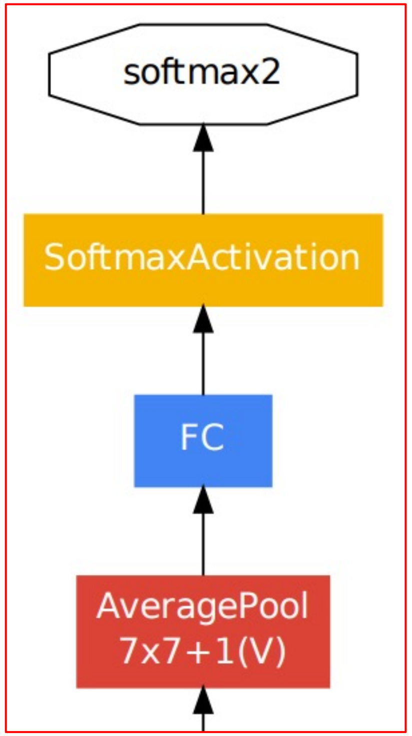


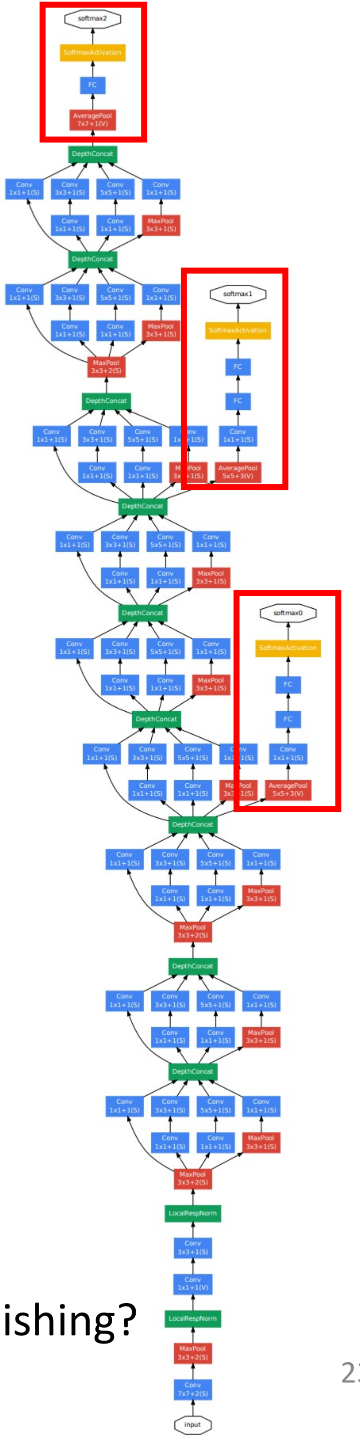
Q: why output prediction from lower layers?

Inception (GoogLeNet)









Q: why output prediction from lower layers?

Hint: remember gradient vanishing?

of layers

- LeNet-5: 3 conv + 2 fc
- AlexNet: 5 conv + 2 fc
- VGG-16: 13 conv + 2 fc

More conv layers

of layers

- LeNet-5: 3 conv + 2 fc
- AlexNet: 5 conv + 2 fc
- VGG-16: 13 conv + 2 fc

More conv layers

Q: why they did not develop some architecture with more layers?

of layers

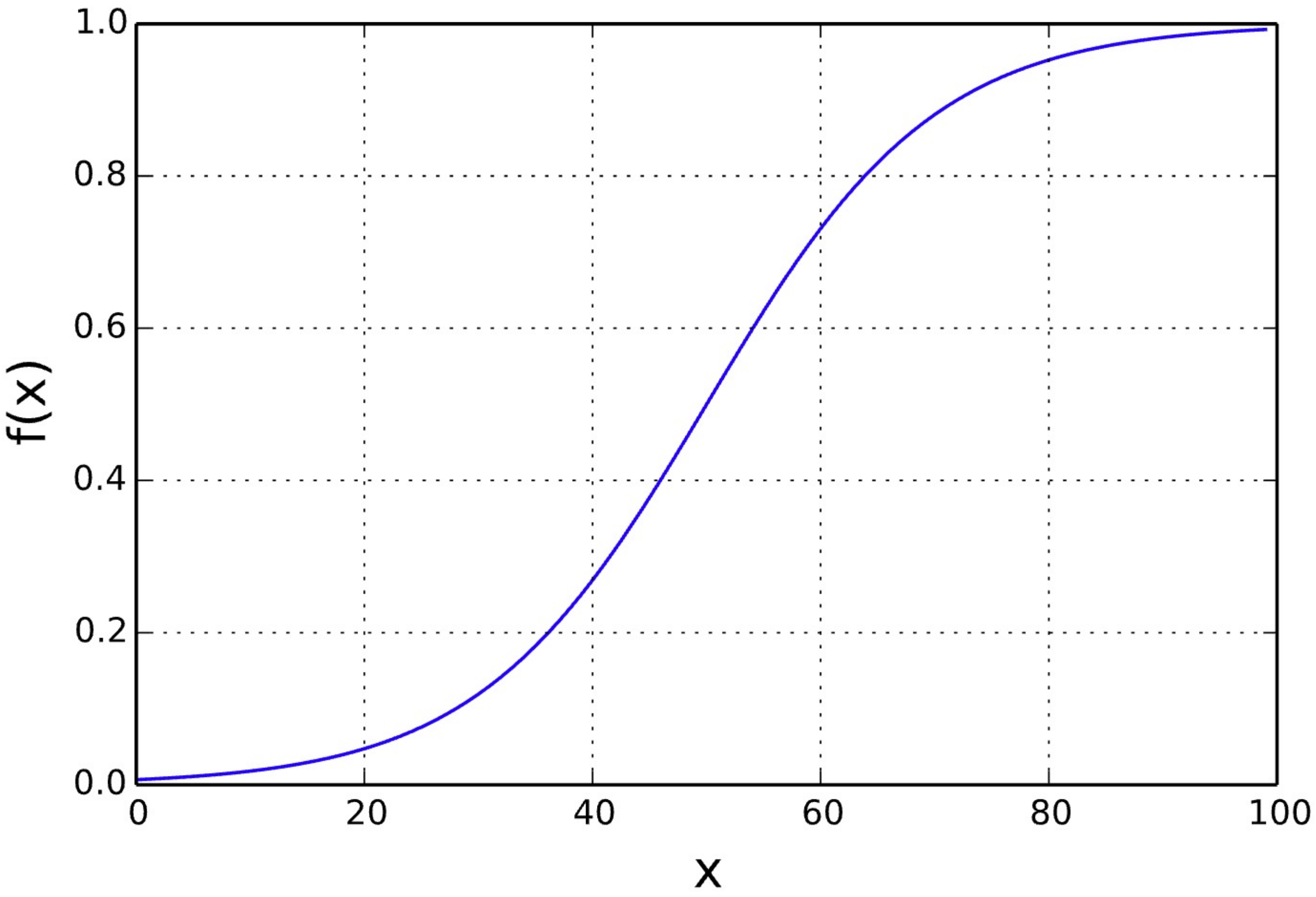
- LeNet-5: 3 conv + 2 fc
- AlexNet: 5 conv + 2 fc
- VGG-16: 13 conv + 2 fc

More conv layers

Q: why they did not develop some architecture with more layers?

Hint (again): remember gradient vanishing?

Gradient vanish



Sigmoid function

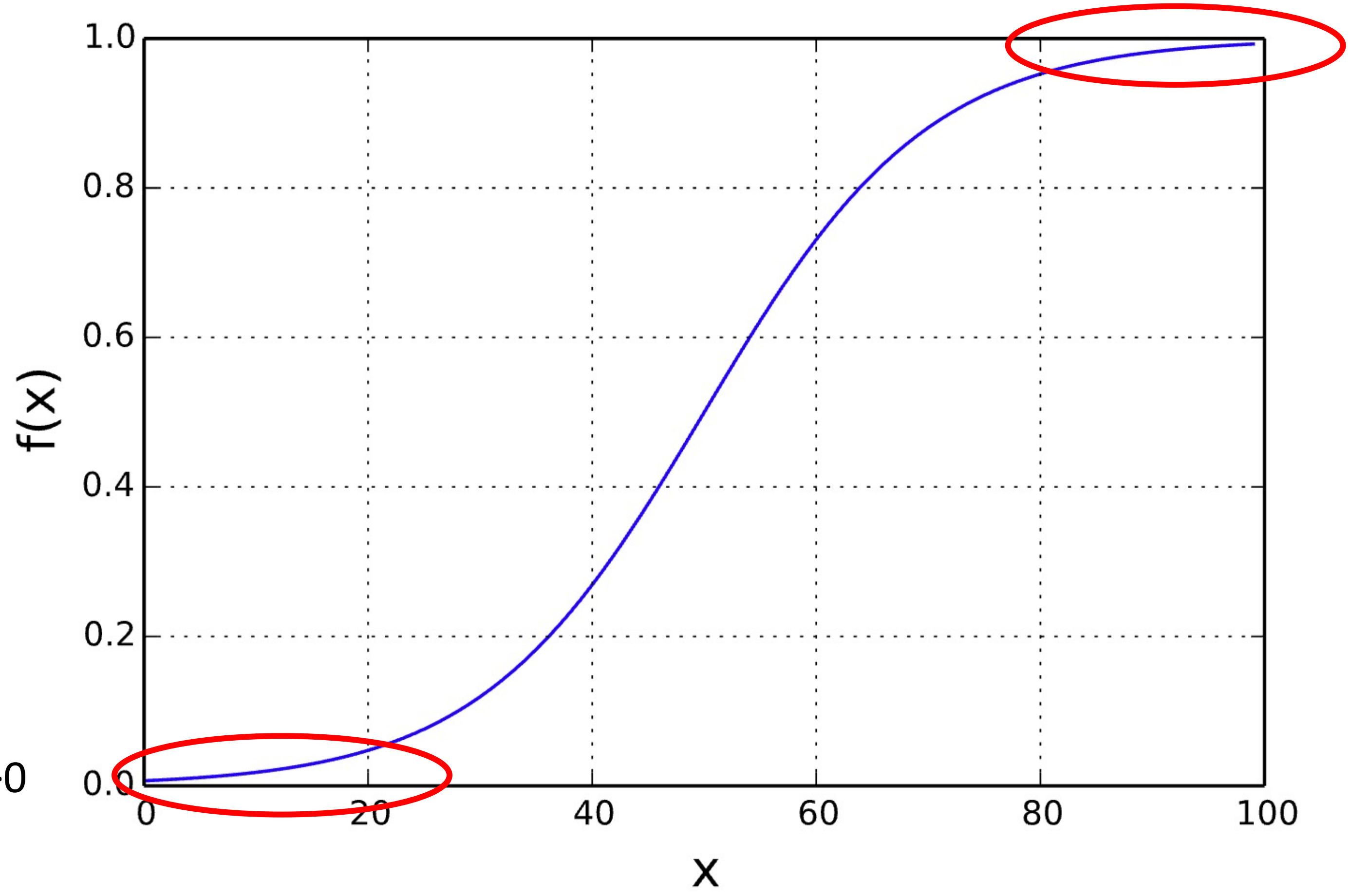
Gradient vanish

$$f_n\left(\dots\left(f_2\left(f_1(x)\right)\right)\right) \to ?$$

$$f_i \to x_i$$

$$\frac{dx_n}{dx_1} = \frac{dx_n}{dx_{n-1}} \cdot \dots \cdot \frac{dx_2}{dx_1} \cdot \frac{dx_1}{dx}$$

gradients->0



gradients->0

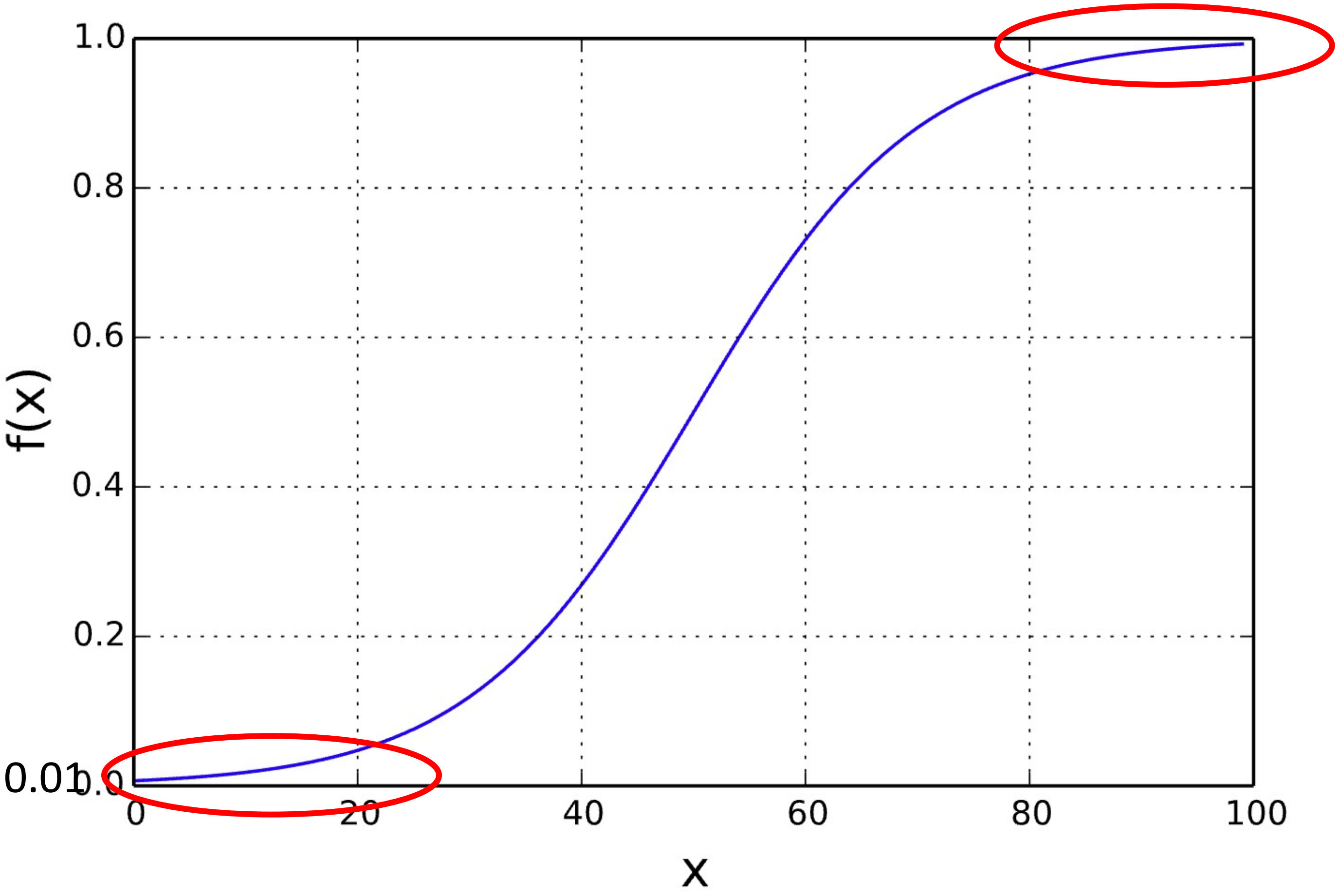
Sigmoid function

Gradient vanish

$$f_n\left(\dots\left(f_2\left(f_1(x)\right)\right)\right) \rightarrow ?$$

$$f_i \rightarrow x_i$$

$$\frac{dx_n}{dx_1} = \frac{dx_n}{dx_{n-1}} \cdot \dots \cdot \frac{dx_2}{dx_1} \cdot \frac{dx_1}{dx}$$



gradients-> 0.01

gradients-> 0.01

Sigmoid function

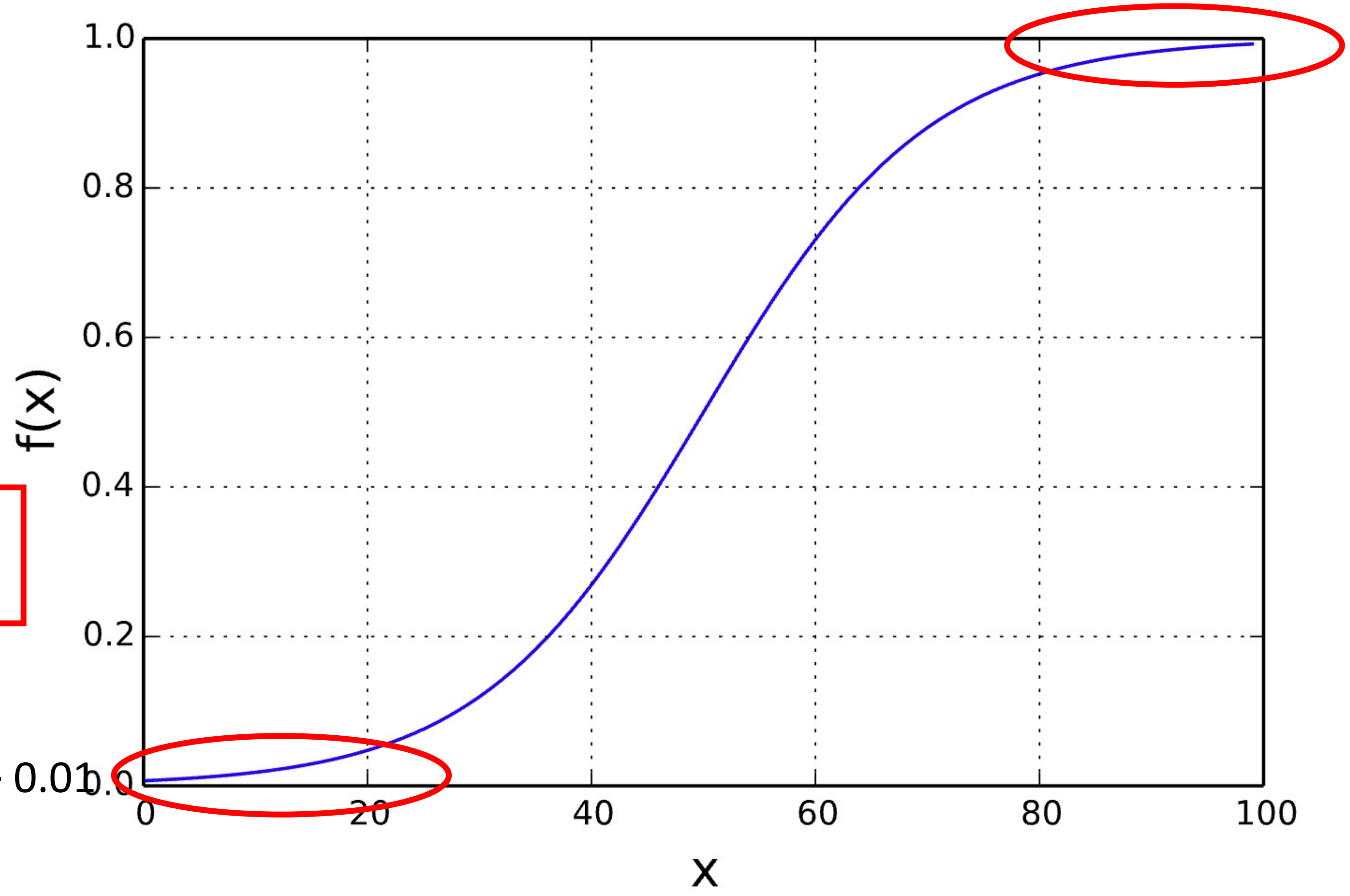
Gradient vanish

$$f_n\left(\dots\left(f_2\left(f_1(x)\right)\right)\right) \to ?$$

$$f_i \to x_i$$

$$\frac{dx_n}{dx_1} = \boxed{\frac{dx_n}{dx_{n-1}}} \cdot \dots \cdot \boxed{\frac{dx_2}{dx_1}} \cdot \boxed{\frac{dx_1}{dx}}$$

gradients-> 0.01



gradients-> 0.01

Sigmoid function

Gradient vanish

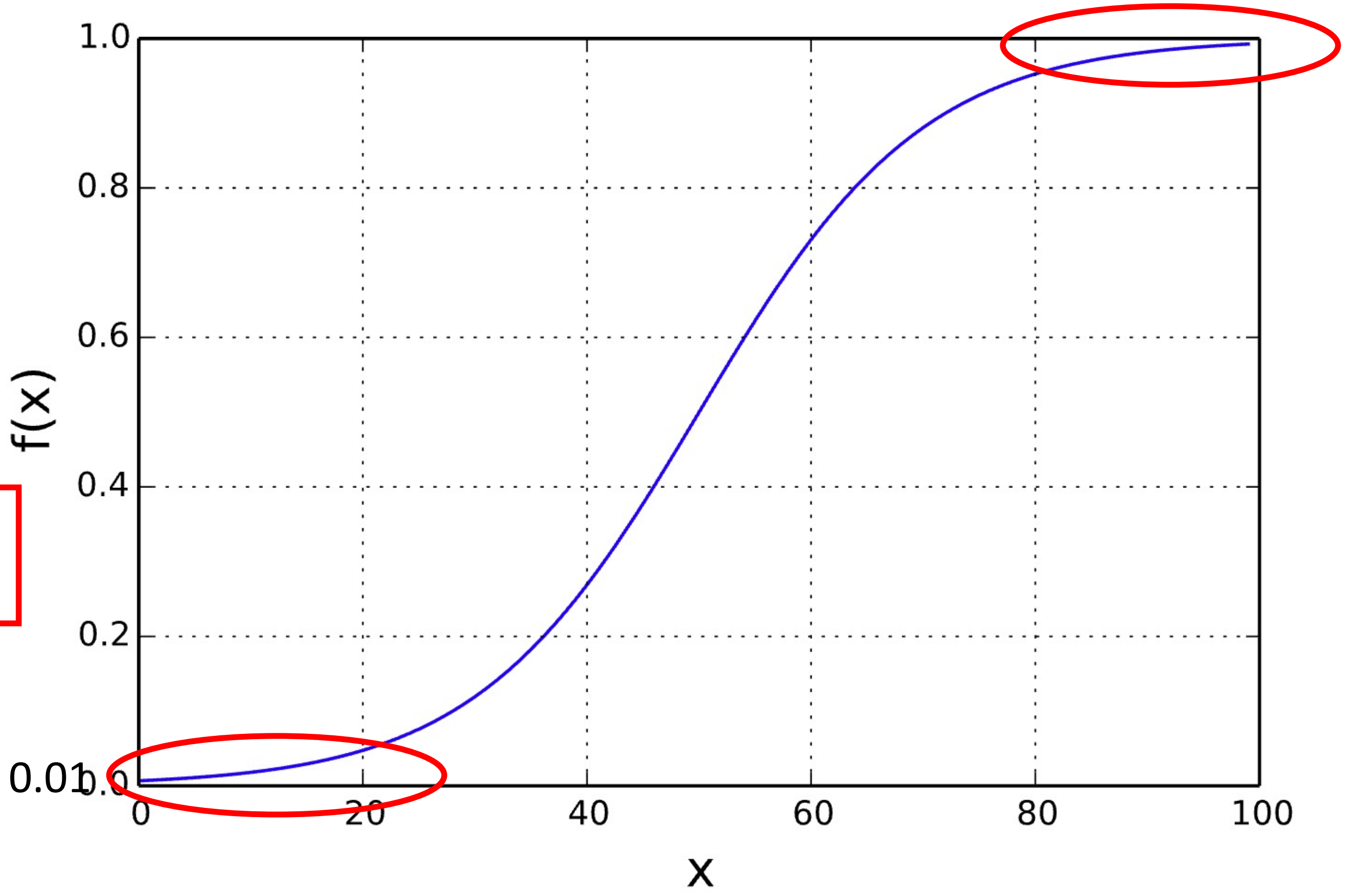
$$f_n\left(\dots\left(f_2(f_1(x))\right)\right) \to ?$$

$$f_i \to x_i$$

$$\frac{dx_n}{dx_1} = \boxed{\frac{dx_n}{dx_{n-1}}} \cdot \dots \cdot \boxed{\frac{dx_2}{dx_1}} \cdot \boxed{\frac{dx_1}{dx}}$$

$\to 0.01^n$

gradients-> 0.01



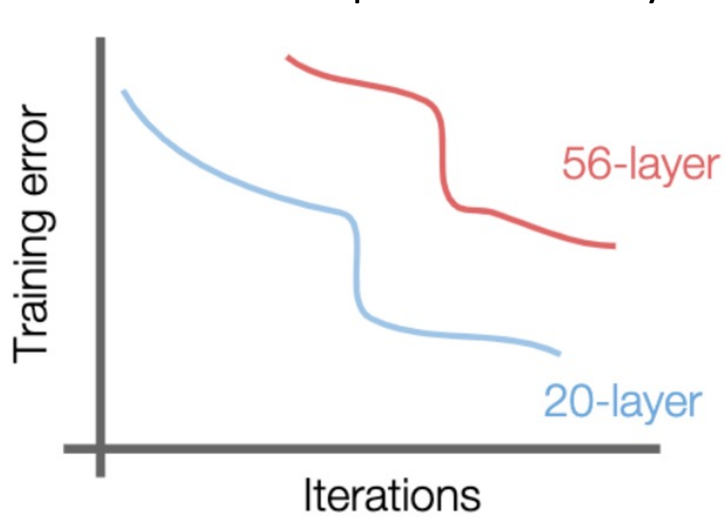
gradients-> 0.01

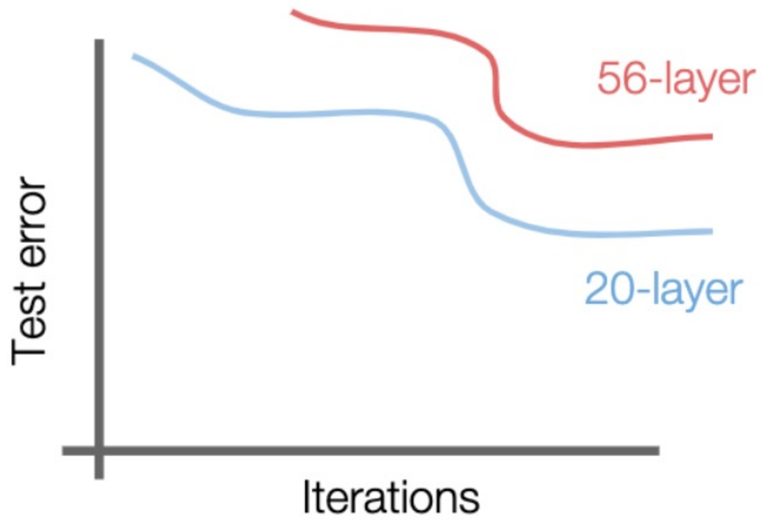
Sigmoid function

of layers

- LeNet-5: 3 conv + 2 fc
- AlexNet: 5 conv + 2 fc
- VGG-16: 13 conv + 2 fc

Q: why they did not develop some architecture with more layers?
Optimization may be difficult.



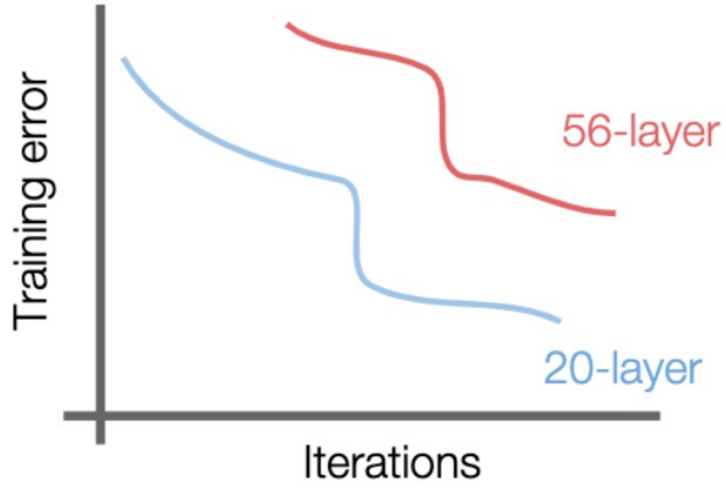


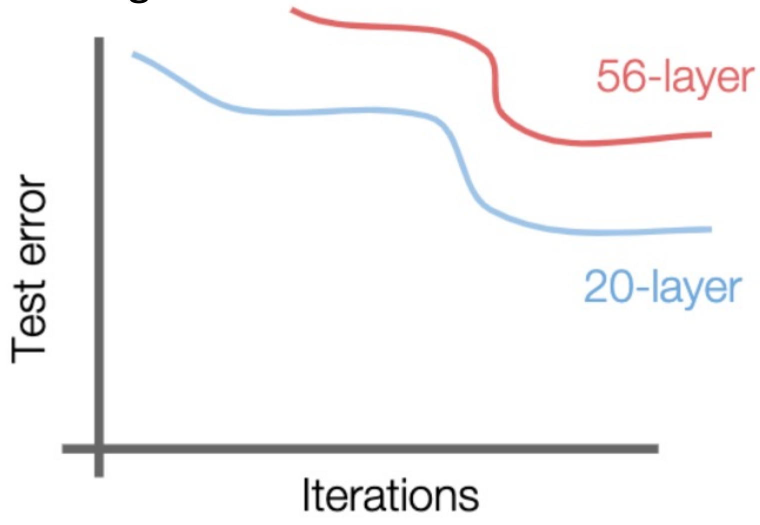
of layers

- LeNet-5: 3 conv + 2 fc
- AlexNet: 5 conv + 2 fc
- VGG-16: 13 conv + 2 fc

Q: why they did not develop some architecture with more layers?

Optimization may be difficult. We do not have a good solution as our model.



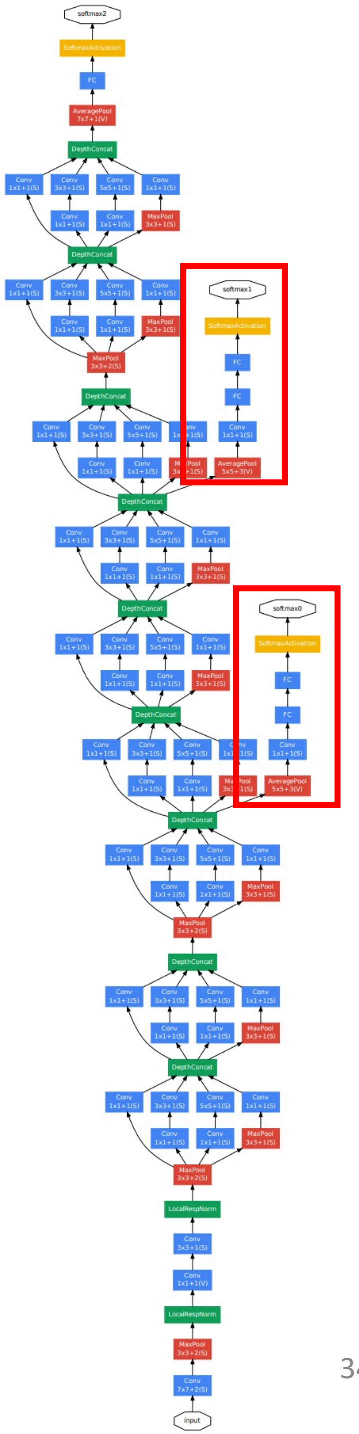


Inception

$$f_n\left(\dots\left(f_2\left(f_1(x)\right)\right)\right) \rightarrow ?$$

$$\frac{dx_n}{dx_1} = \boxed{\frac{dx_n}{dx_{n-1}}} \cdot \dots \cdot \boxed{\frac{dx_2}{dx_1}} \cdot \boxed{\frac{dx_1}{dx}}$$

$\rightarrow 0.01^n$

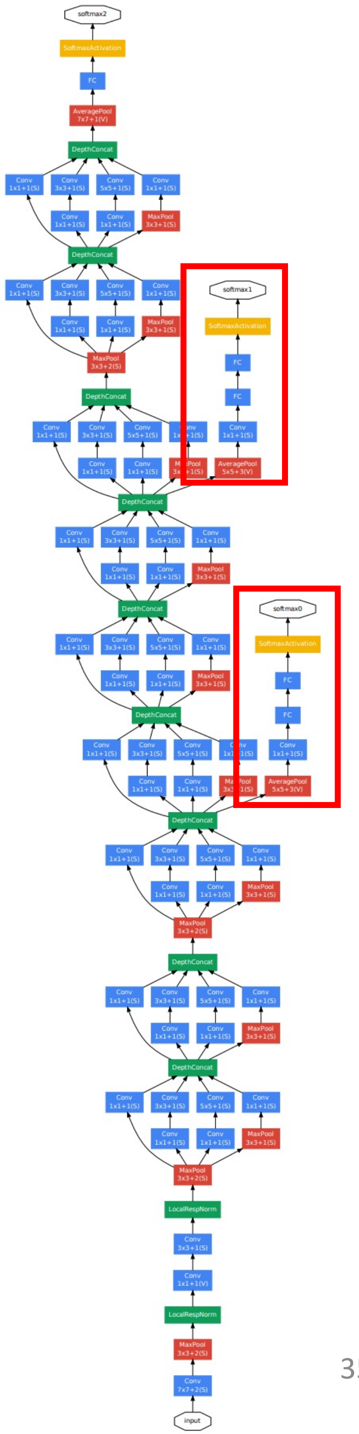


Inception

$$f_n\left(\dots\left(f_2\left(f_1(x)\right)\right)\right) + f_m\left(\dots\left(f_{n+2}\left(f_{n+1}(x)\right)\right)\right)$$

$$\frac{dx_n}{dx_1} = \boxed{\frac{dx_n}{dx_{n-1}}} \cdot \dots \cdot \boxed{\frac{dx_2}{dx_1}} \cdot \boxed{\frac{dx_1}{dx}}$$

$\rightarrow 0.01^n$

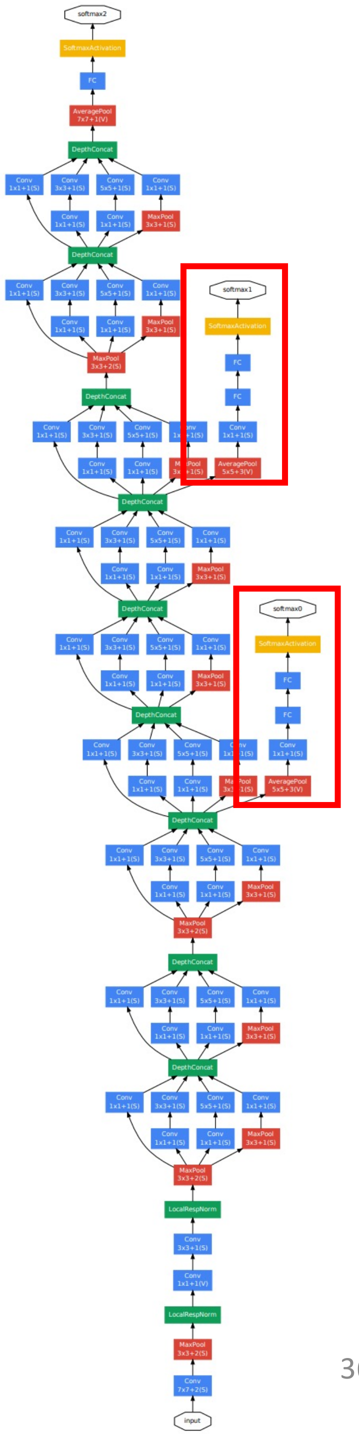


Inception

$$f_n\left(\cdots\left(f_2\left(f_1(x)\right)\right)\right) + f_m\left(\cdots\left(f_{n+2}\left(f_{n+1}(x)\right)\right)\right)$$

$$\boxed{\frac{dx_n}{dx_1}} = \underbrace{\boxed{\frac{dx_n}{dx_{n-1}} \cdot \ldots \cdot \frac{dx_2}{dx_1} \cdot \frac{dx_1}{dx}}}_{\to 0.01^n} + \underbrace{\boxed{\frac{dx_m}{dx_{m-1}} \cdot \ldots \cdot \frac{dx_{n+2}}{dx_{n+1}} \cdot \frac{dx_{n+1}}{dx}}}_{>> 0}$$

Will not be very small



Residual neural networks (ResNet)

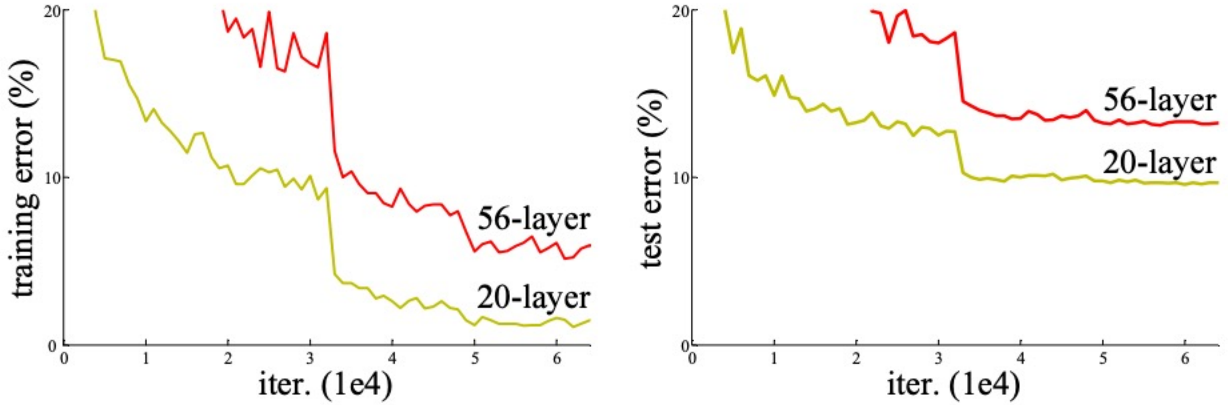


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Residual neural networks (ResNet)

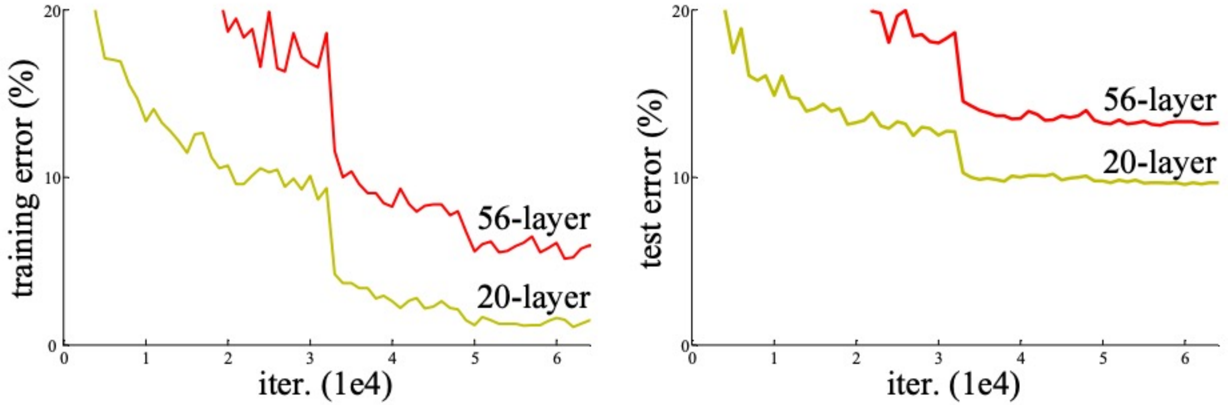


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Without special structure other than conv/fc layers

ResNet: shortcut connection

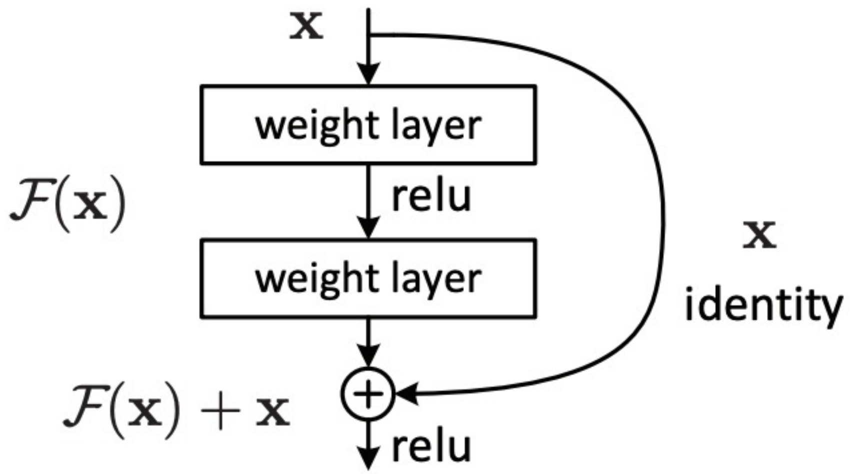


Figure 2. Residual learning: a building block.

ResNet: shortcut connection

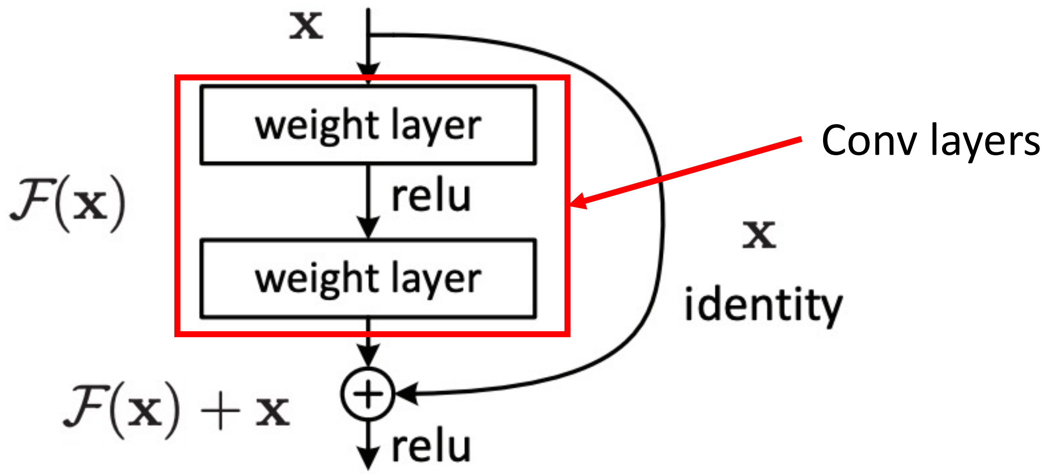


Figure 2. Residual learning: a building block.

ResNet: shortcut connection

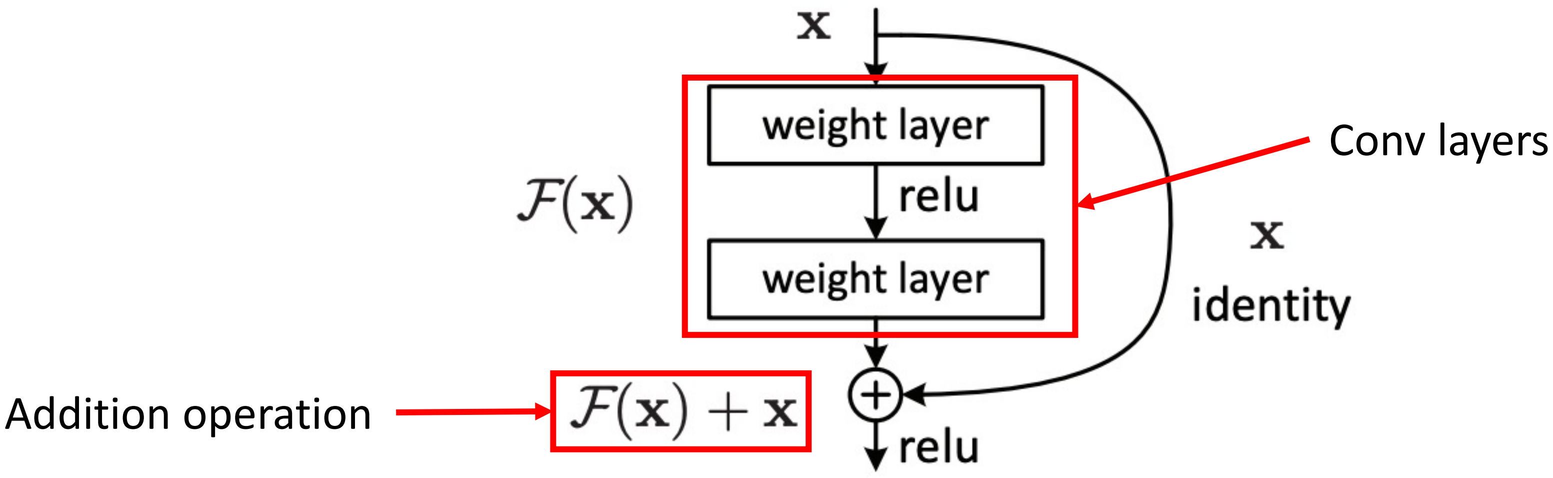
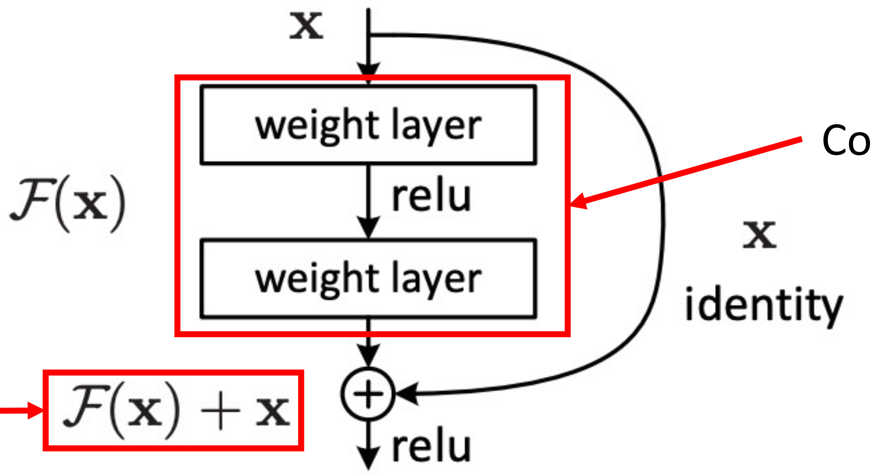


Figure 2. Residual learning: a building block.

ResNet: shortcut connection

Conv layers



Addition operation

Figure 2. Residual learning: a building block.

implication: same dimension

ResNet

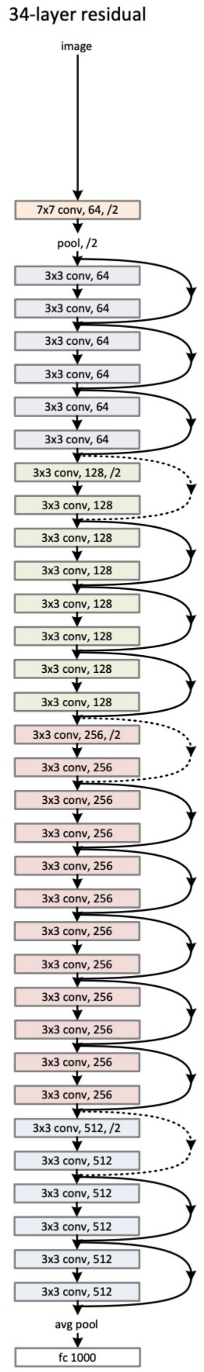


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

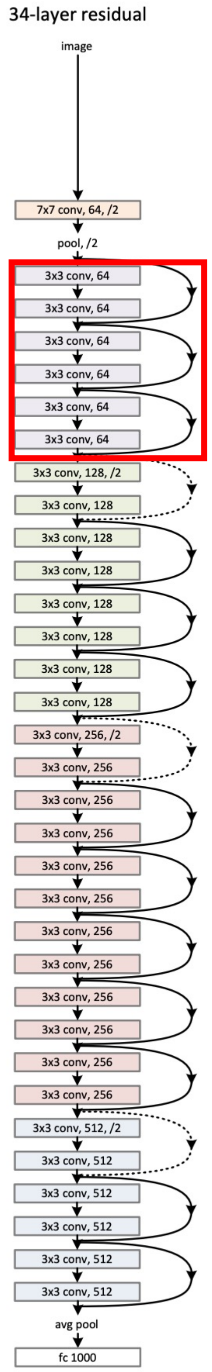
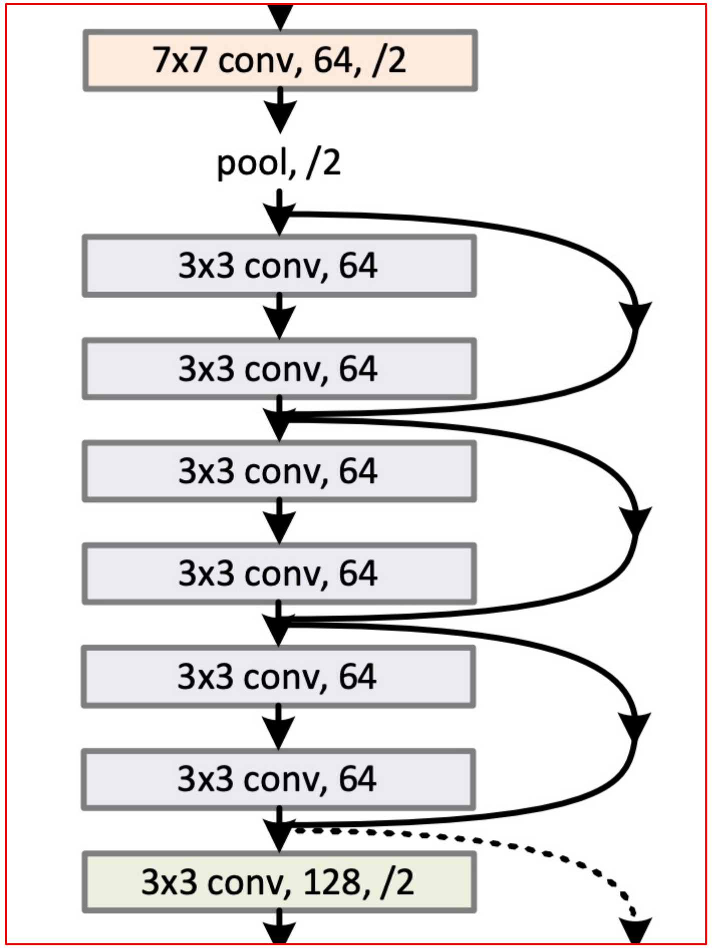


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet



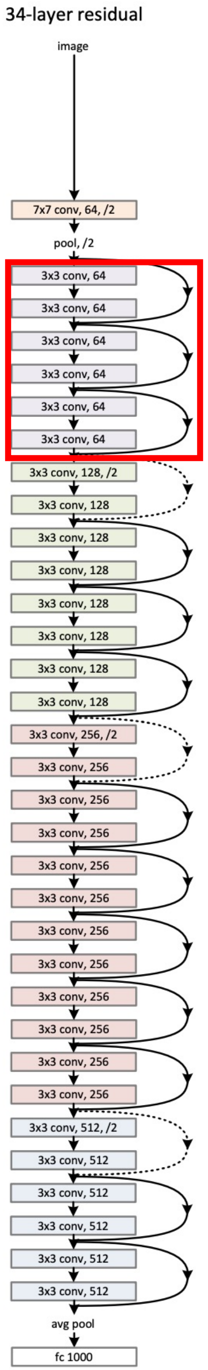
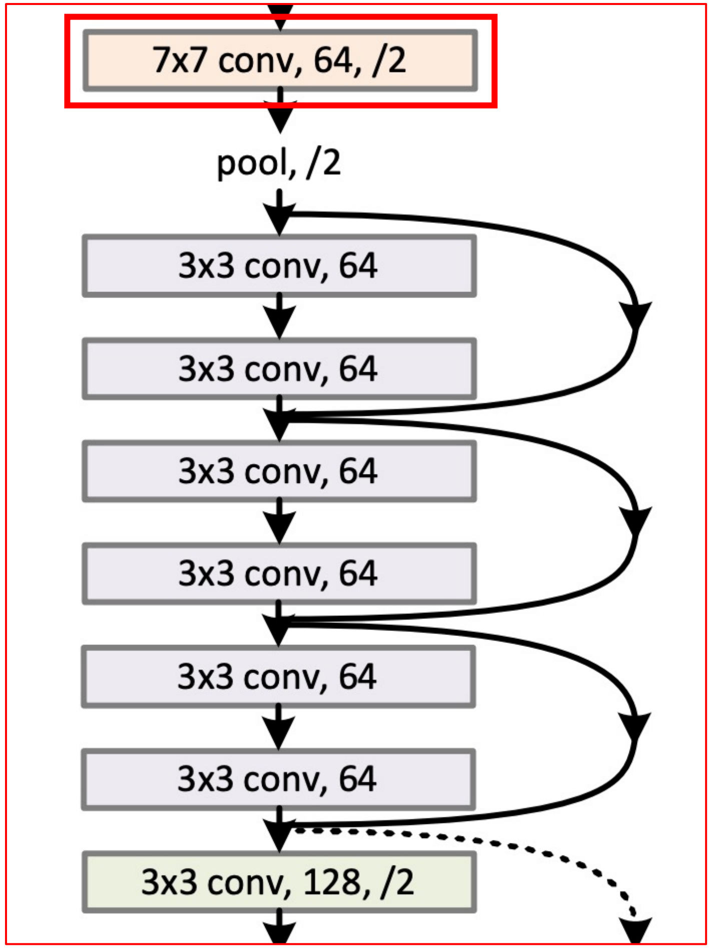


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet



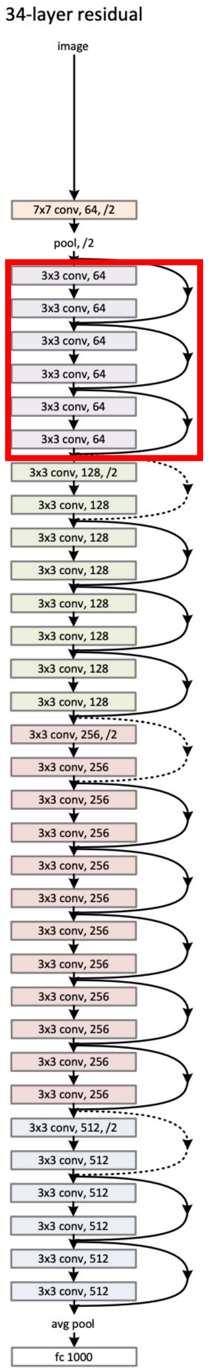
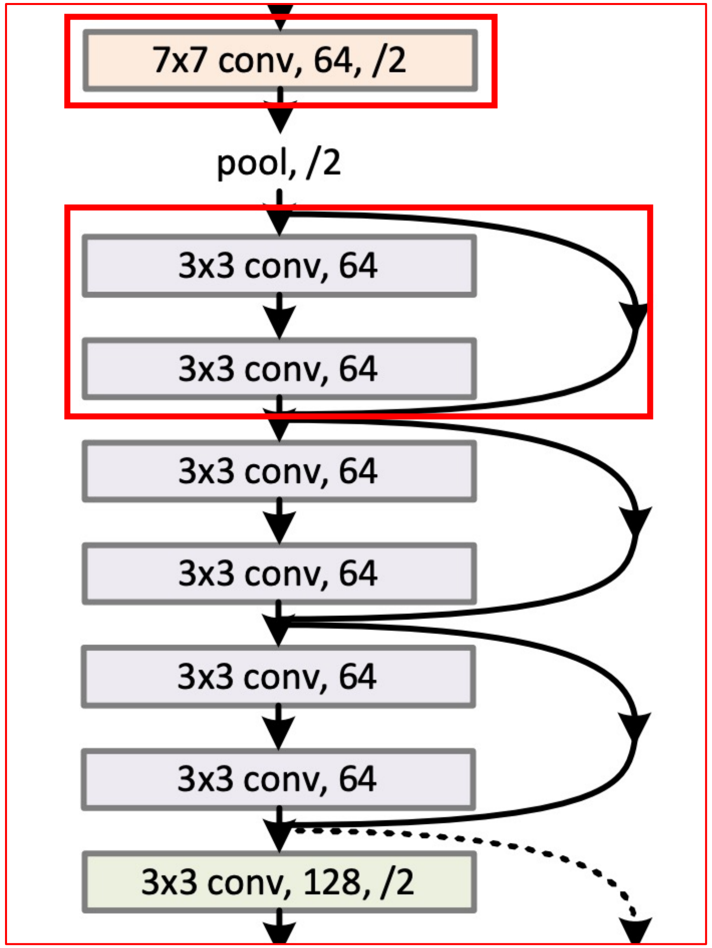


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet



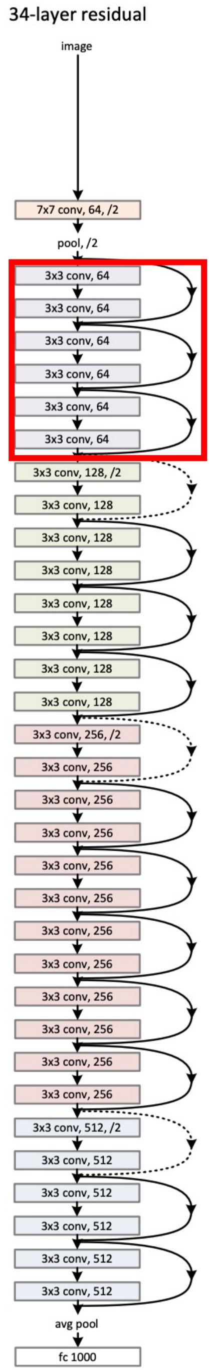
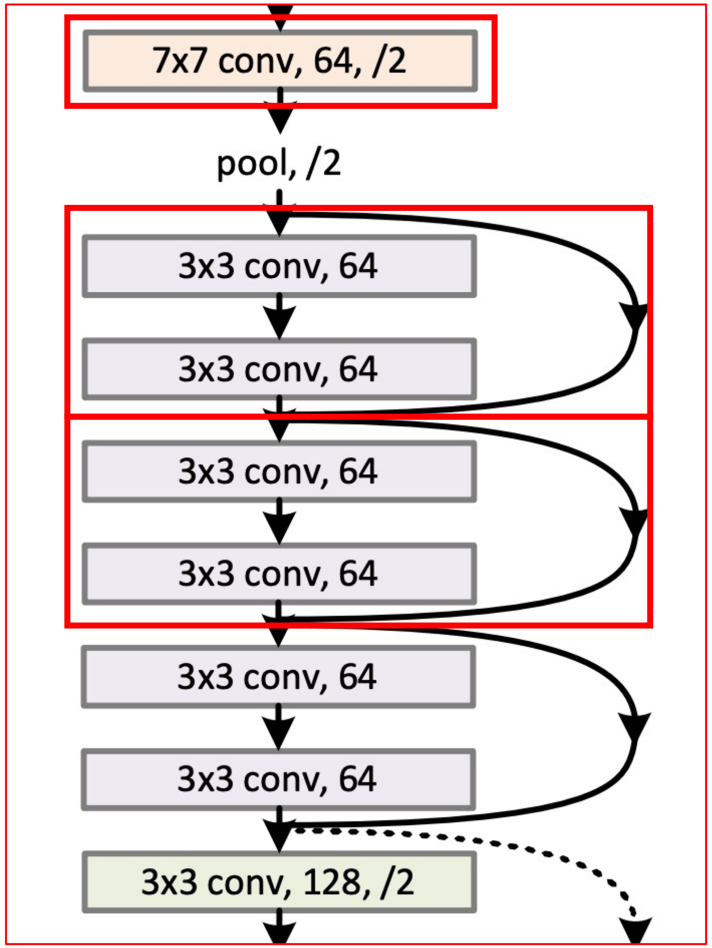


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet



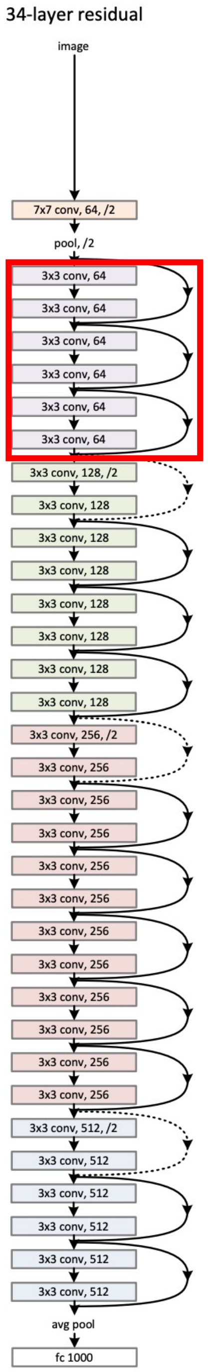
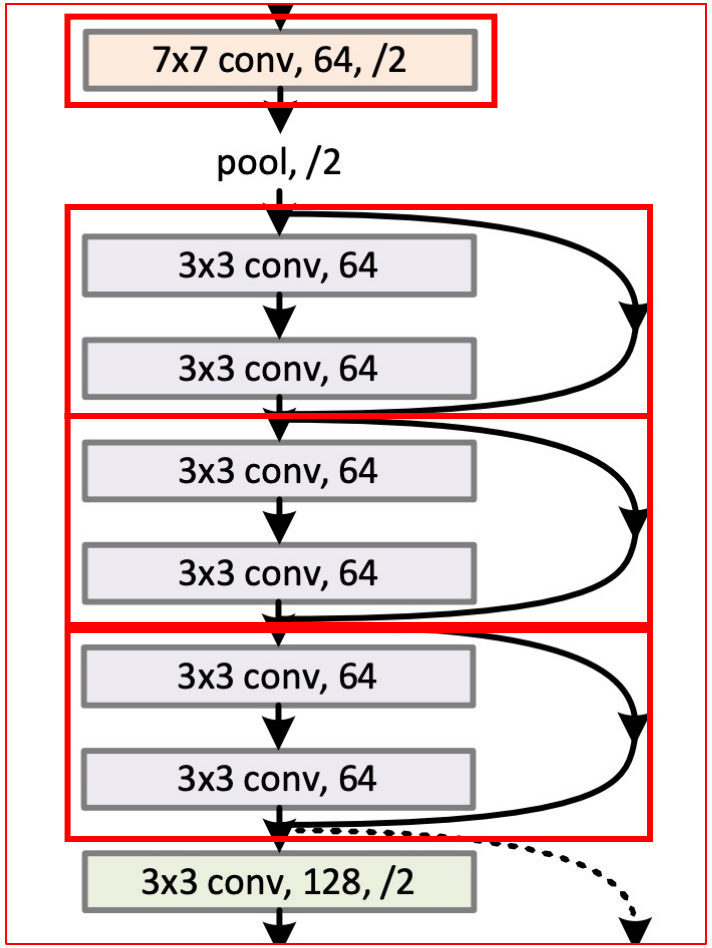


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet



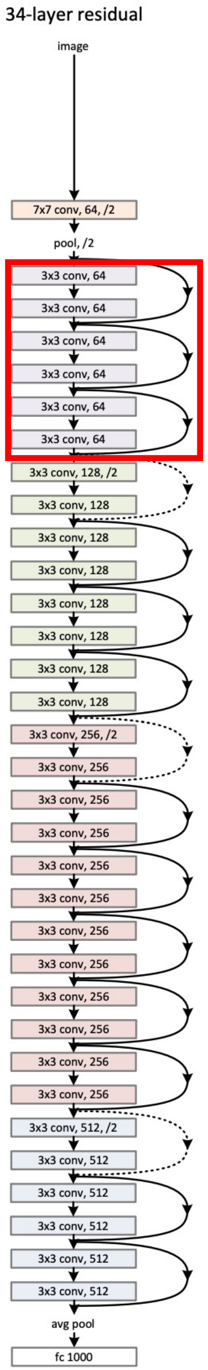
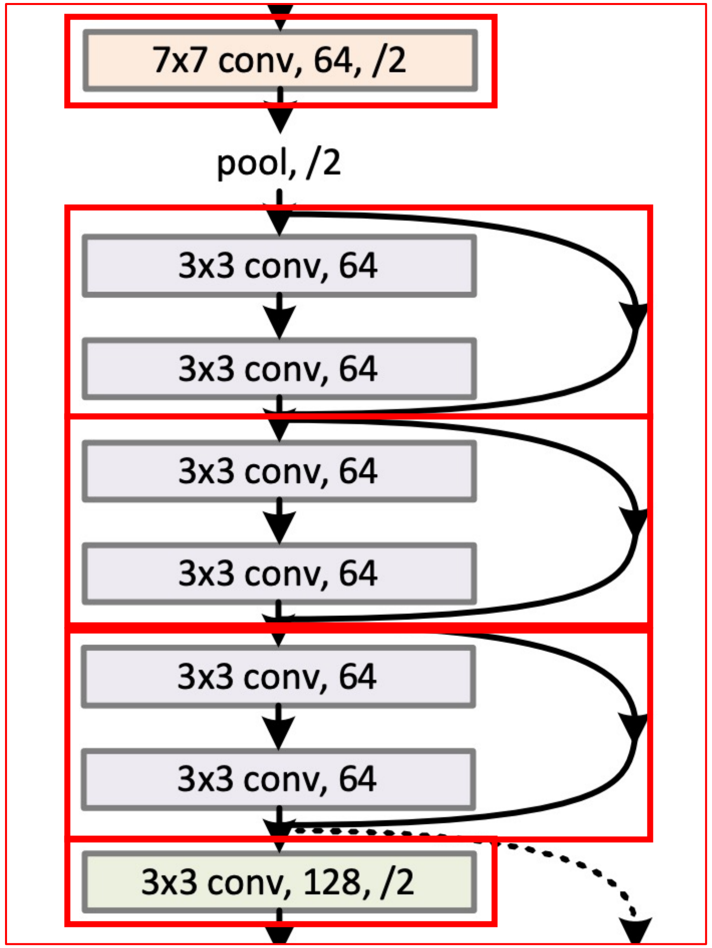


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet



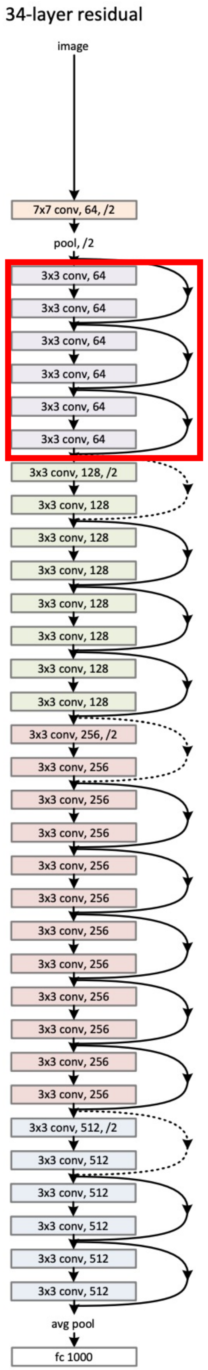


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

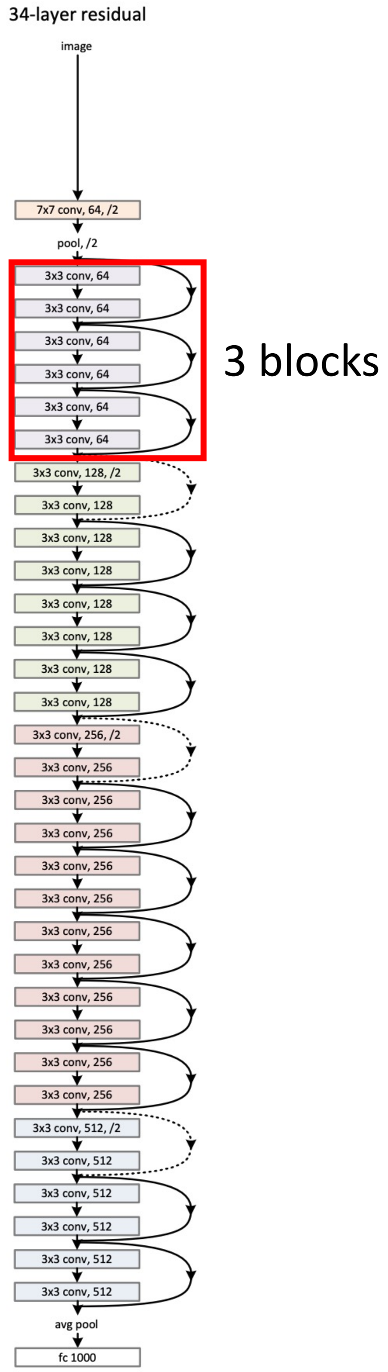


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

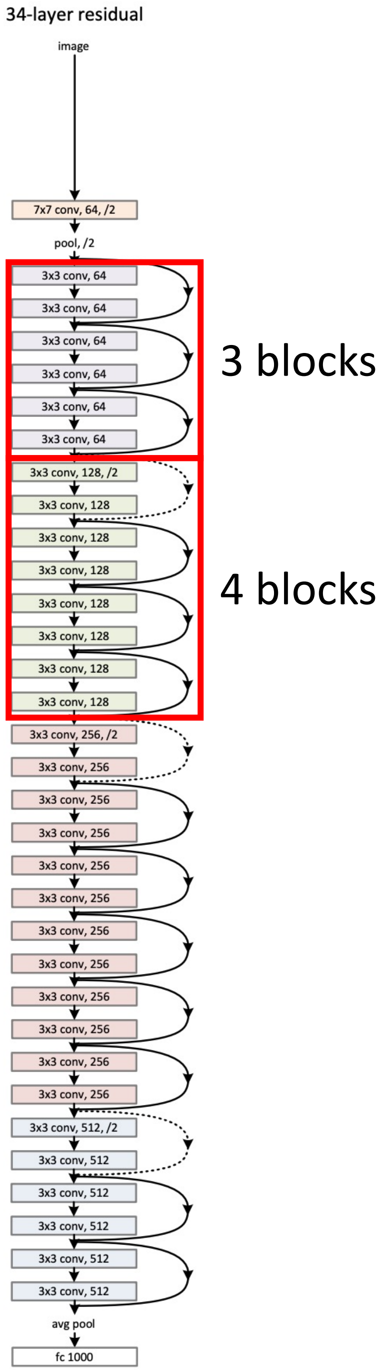


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

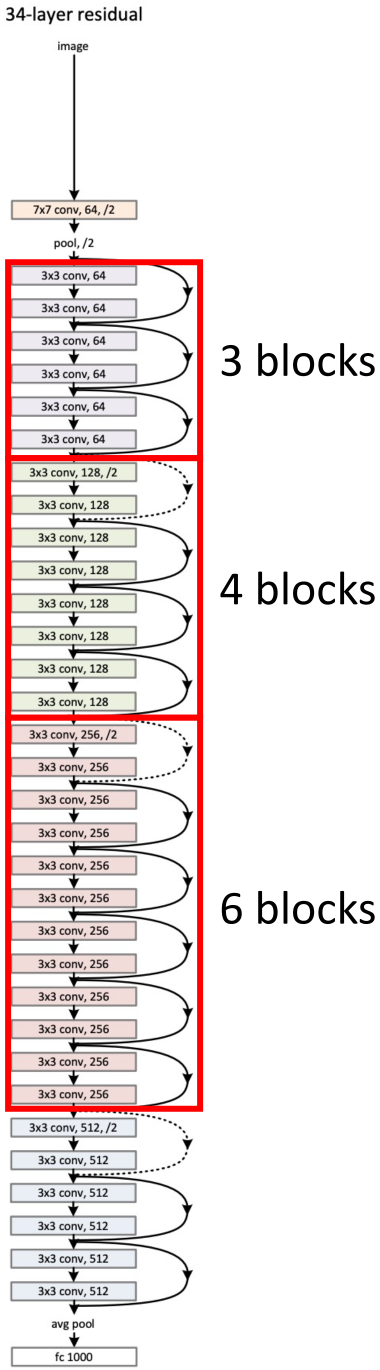


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

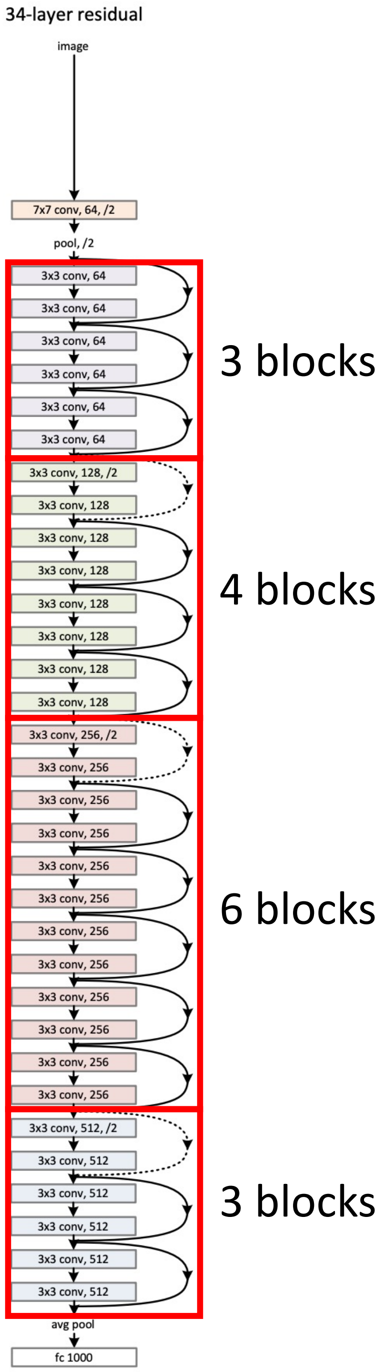


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

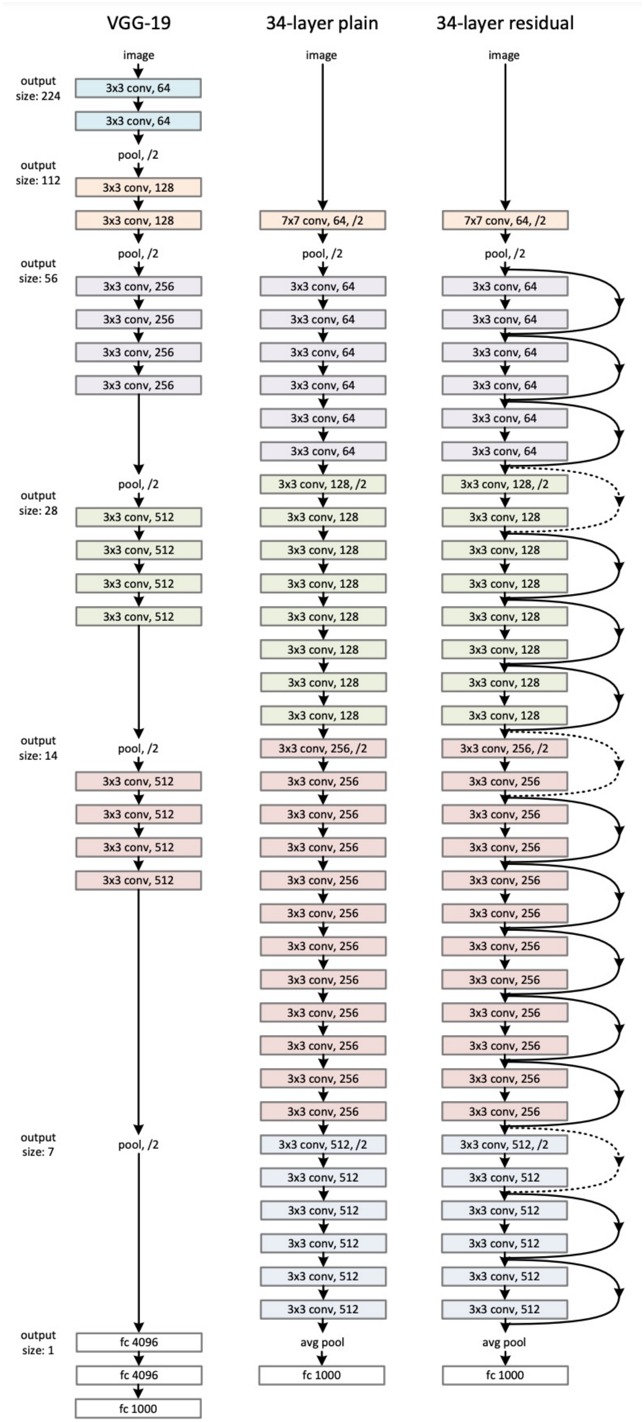


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

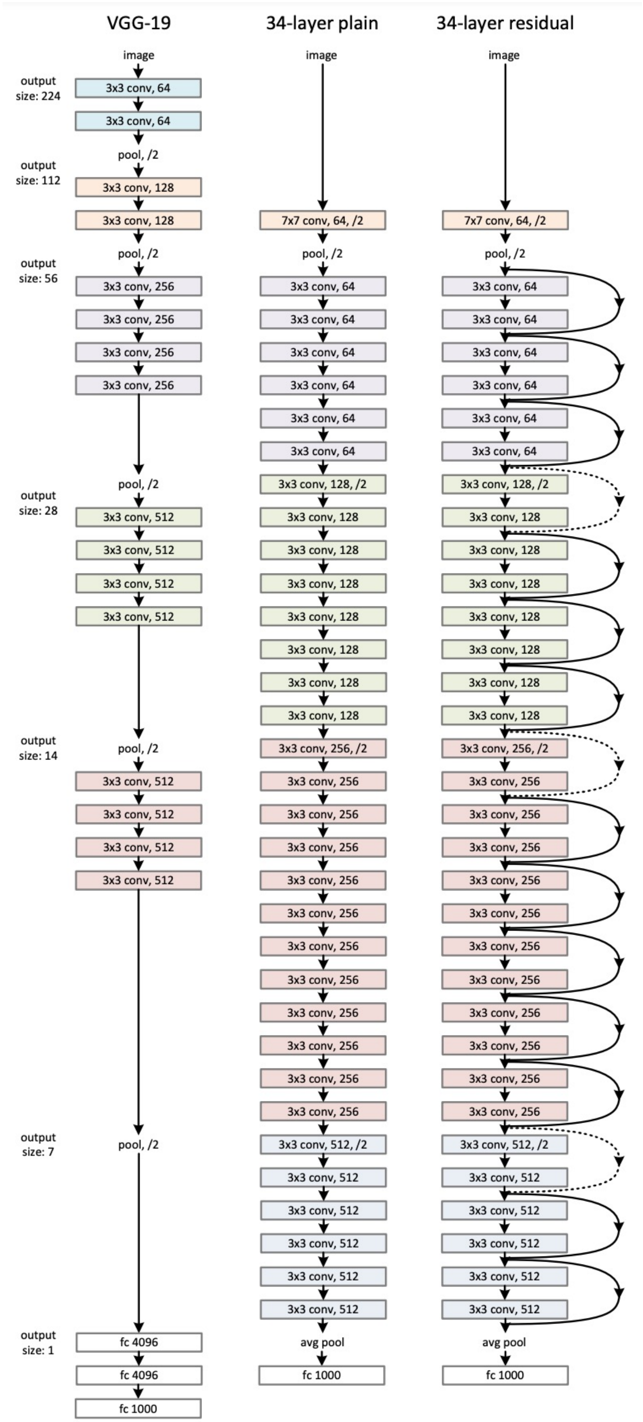
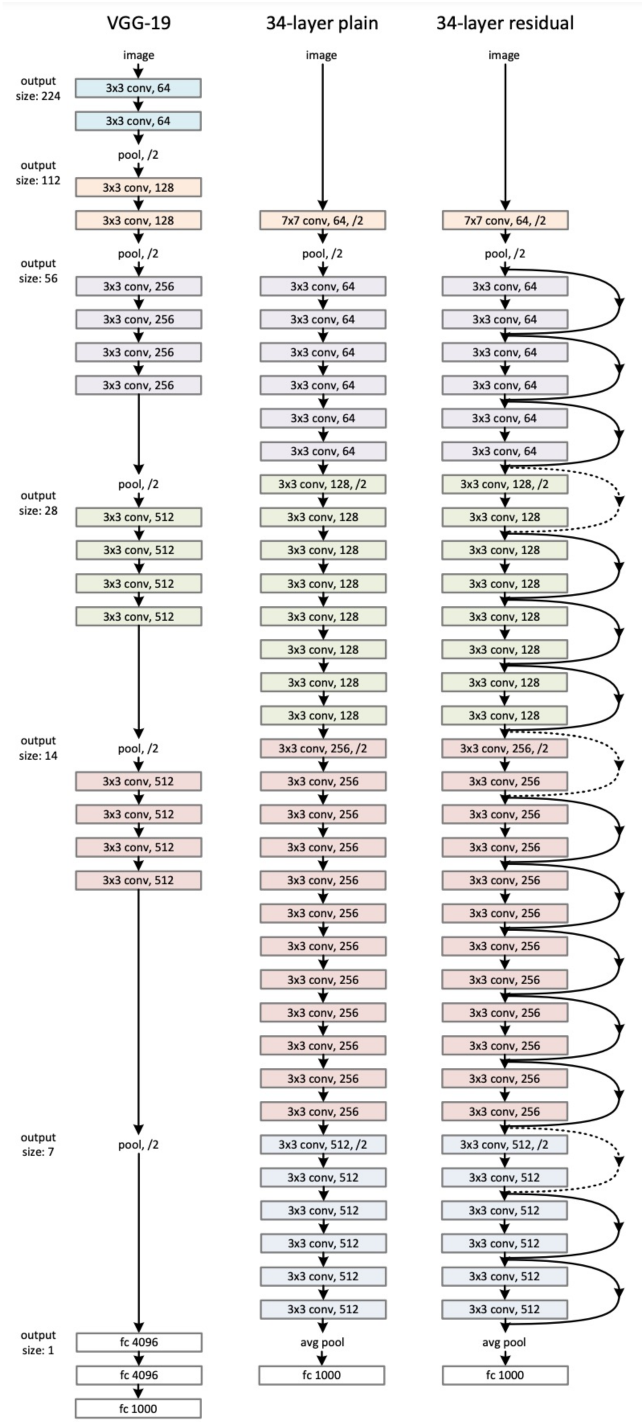


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

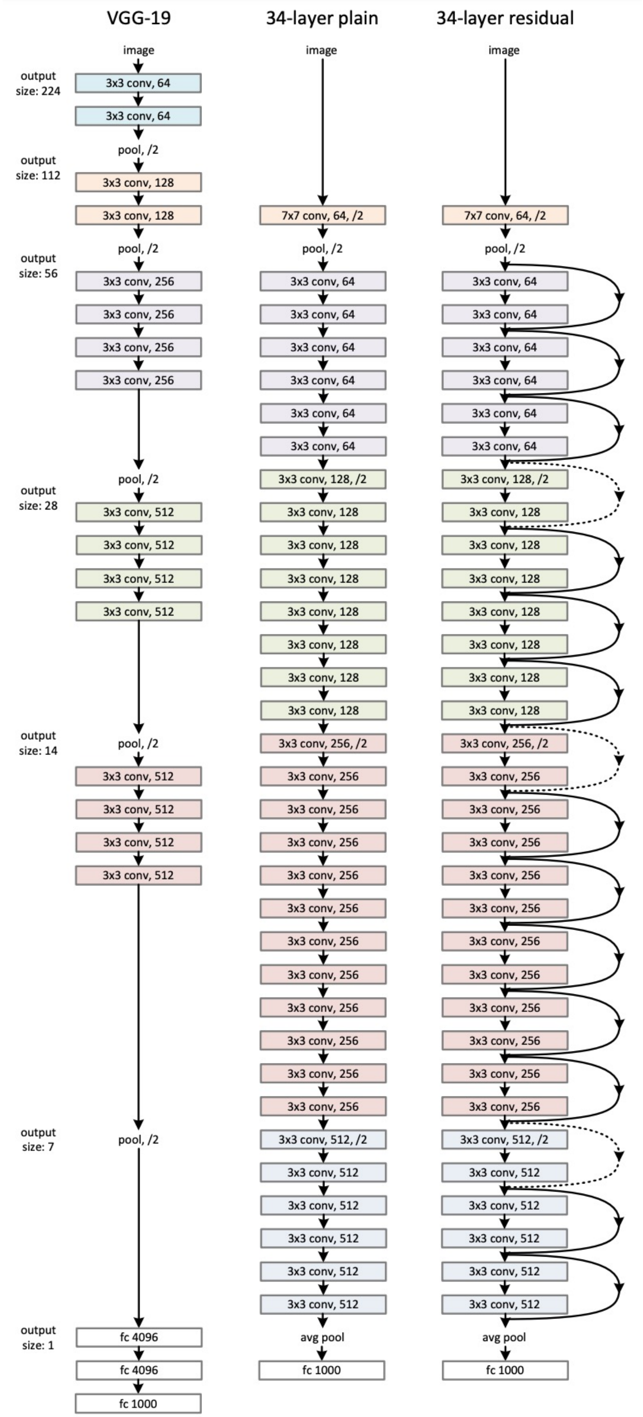
ResNet



floating point operations per second

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

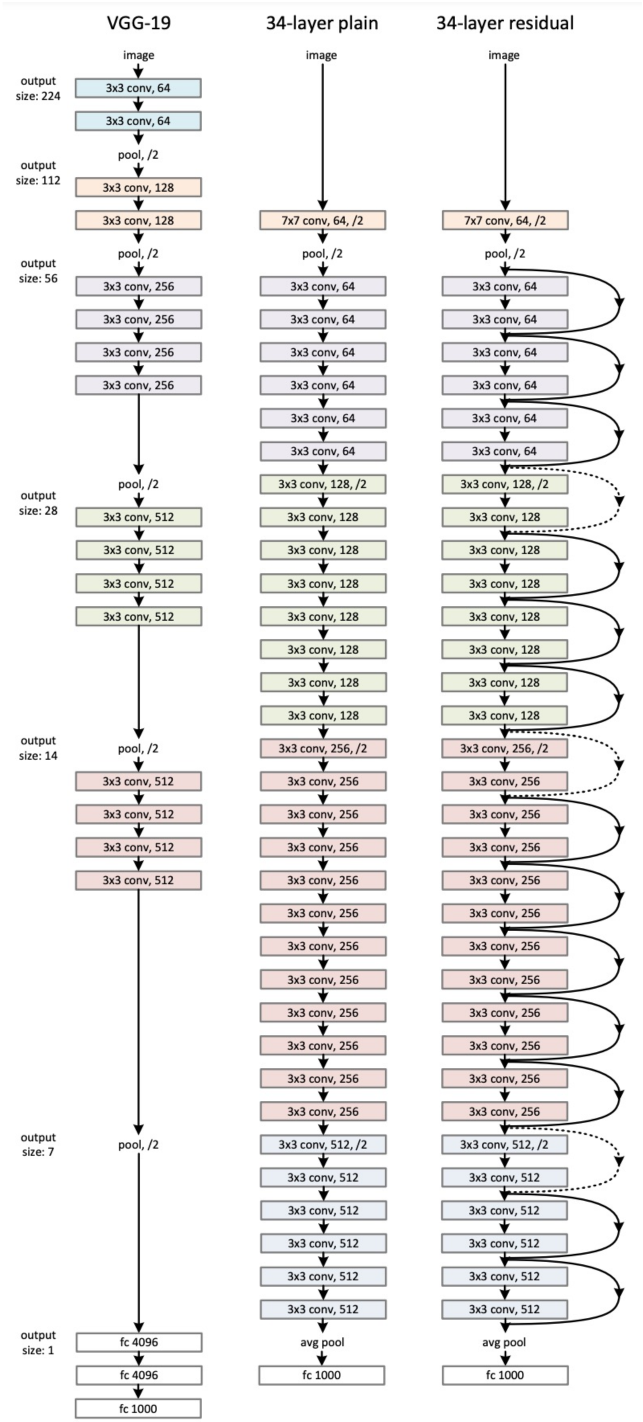
ResNet



Measure: how complicated the model is

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

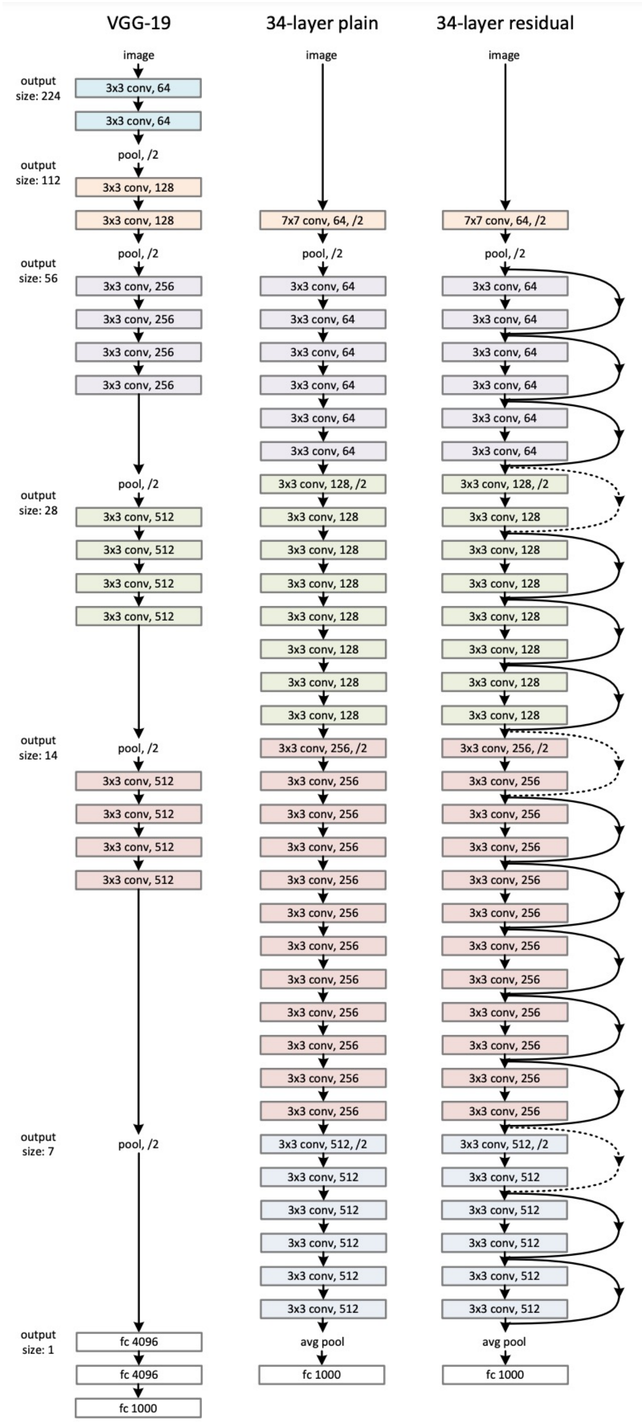
ResNet



Measure: how complicated the model is

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

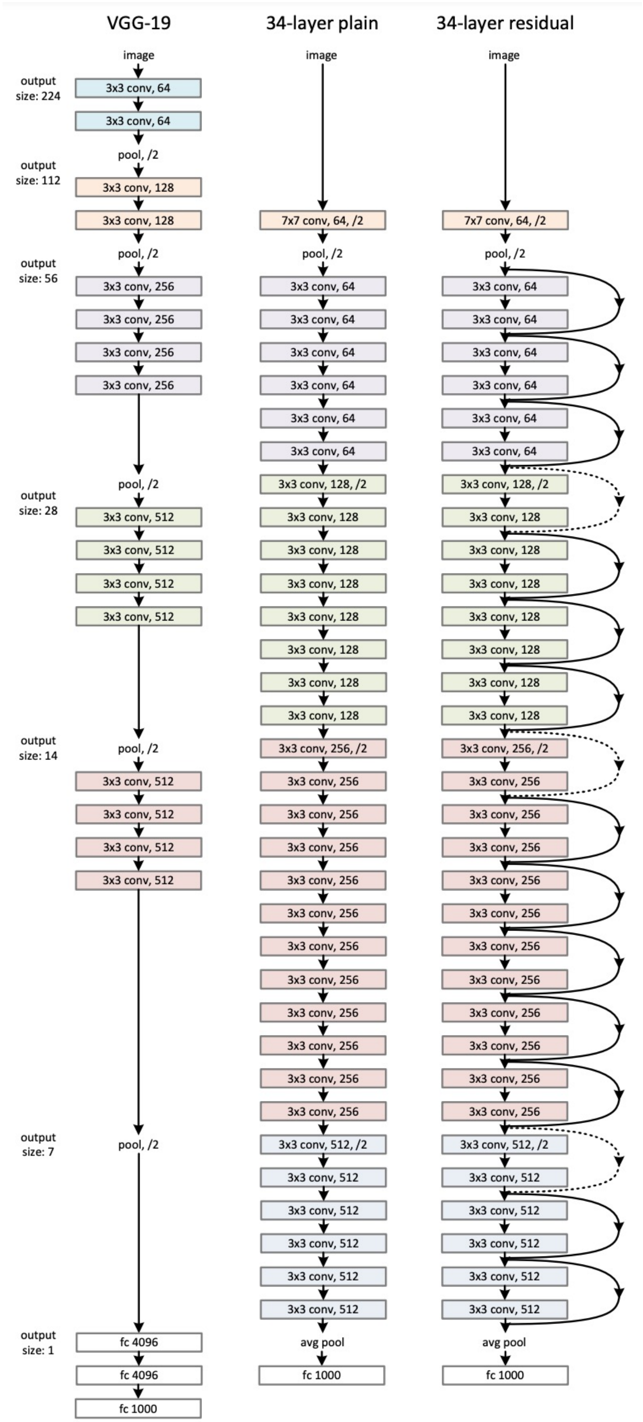
ResNet



Measure: how complicated the model is

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

ResNet

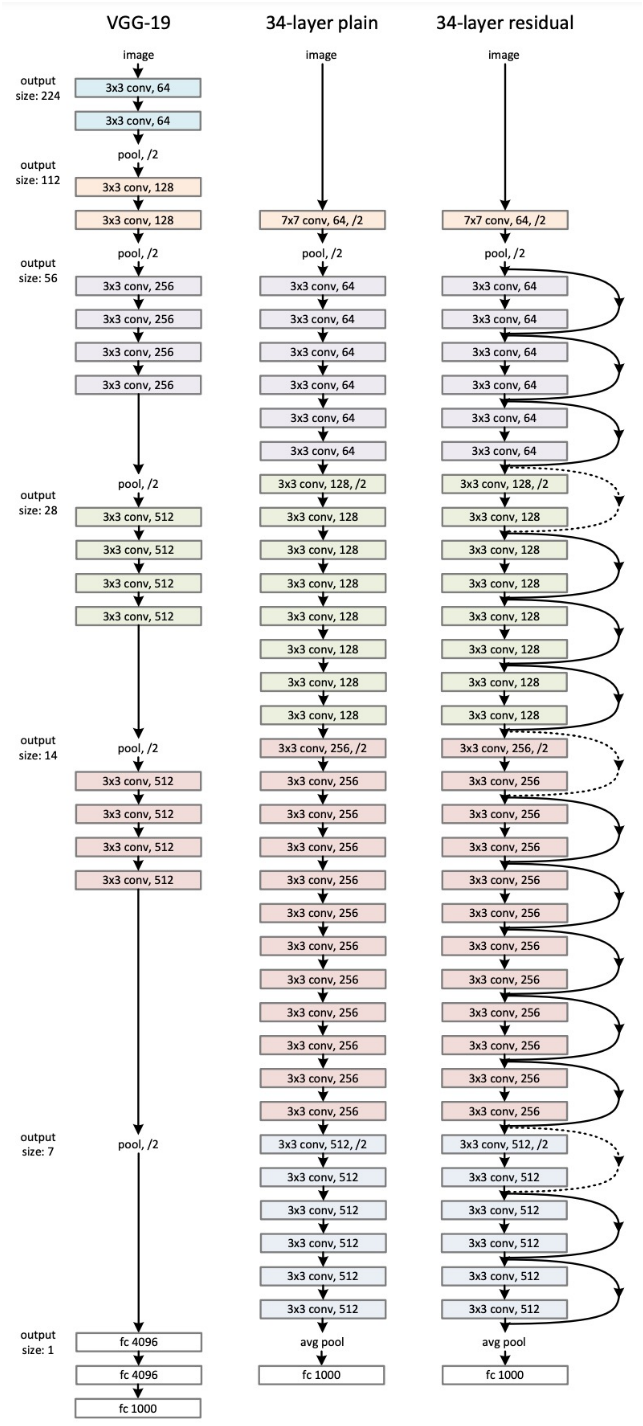


Measure: how complicated the model is

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

Q: VGG-19 has much more FLOPS than 34-layer plain network and 34-layer ResNet?

ResNet



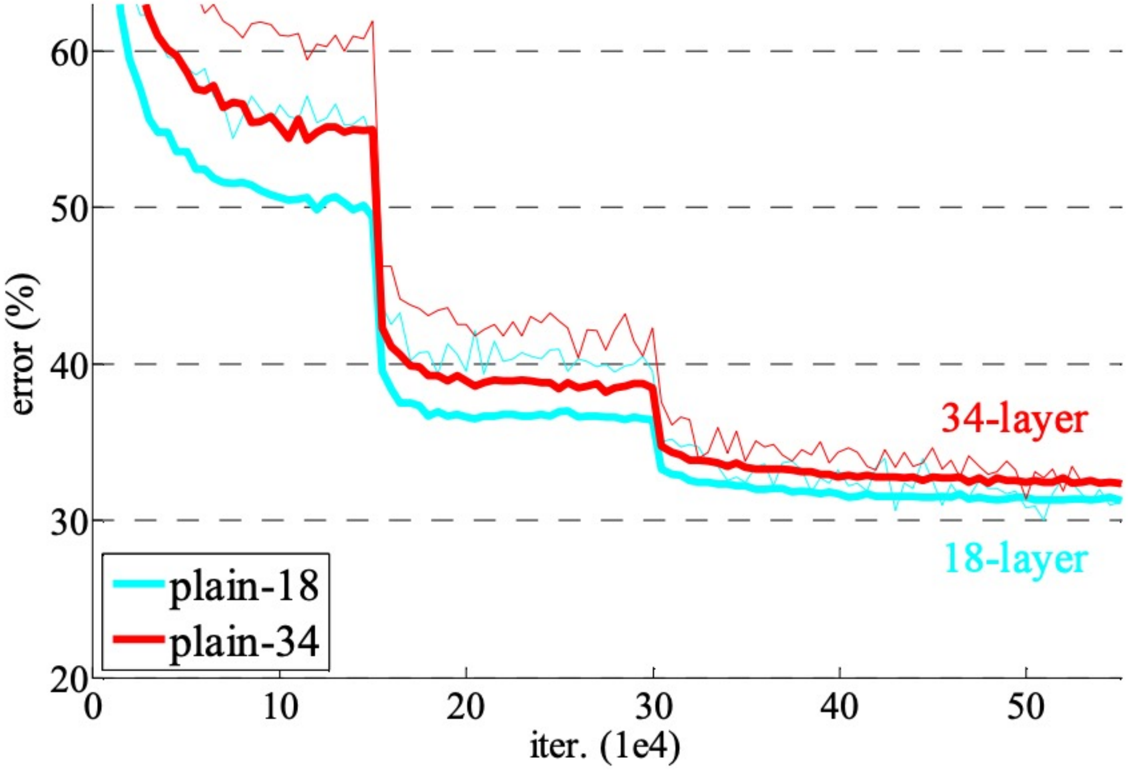
Measure: how complicated the model is

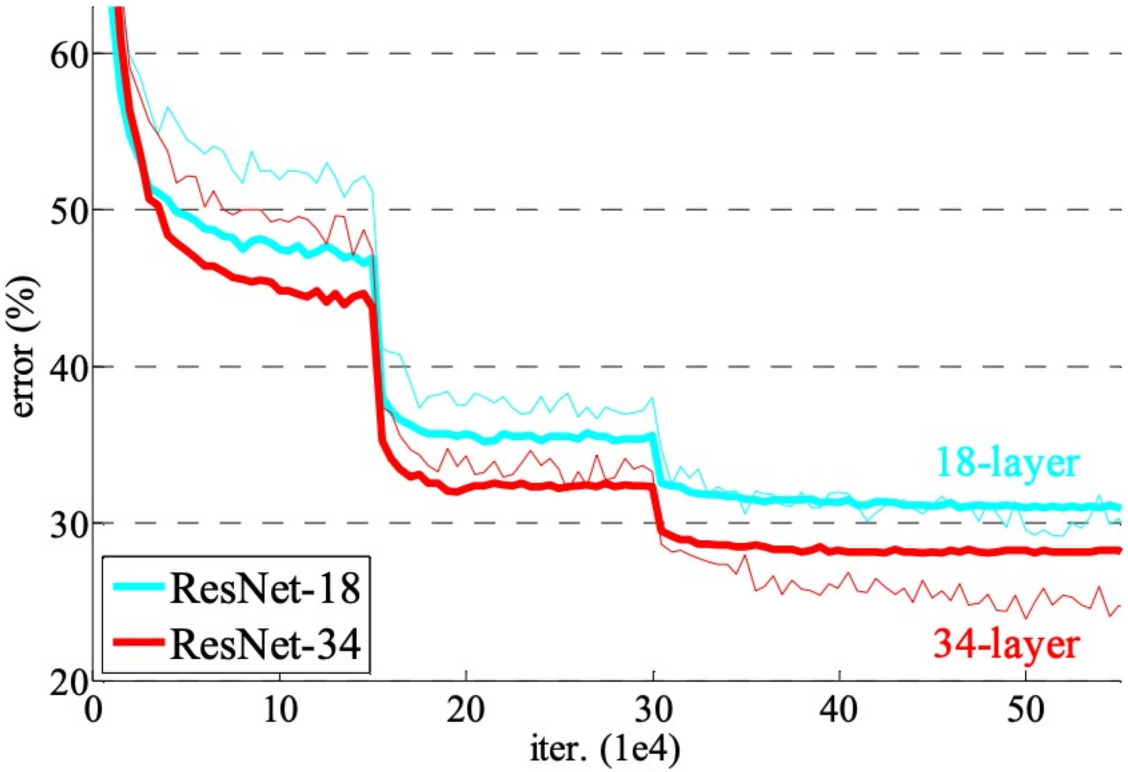
Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

Q: VGG-19 has much more FLOPS than 34-layer plain network and 34-layer ResNet?

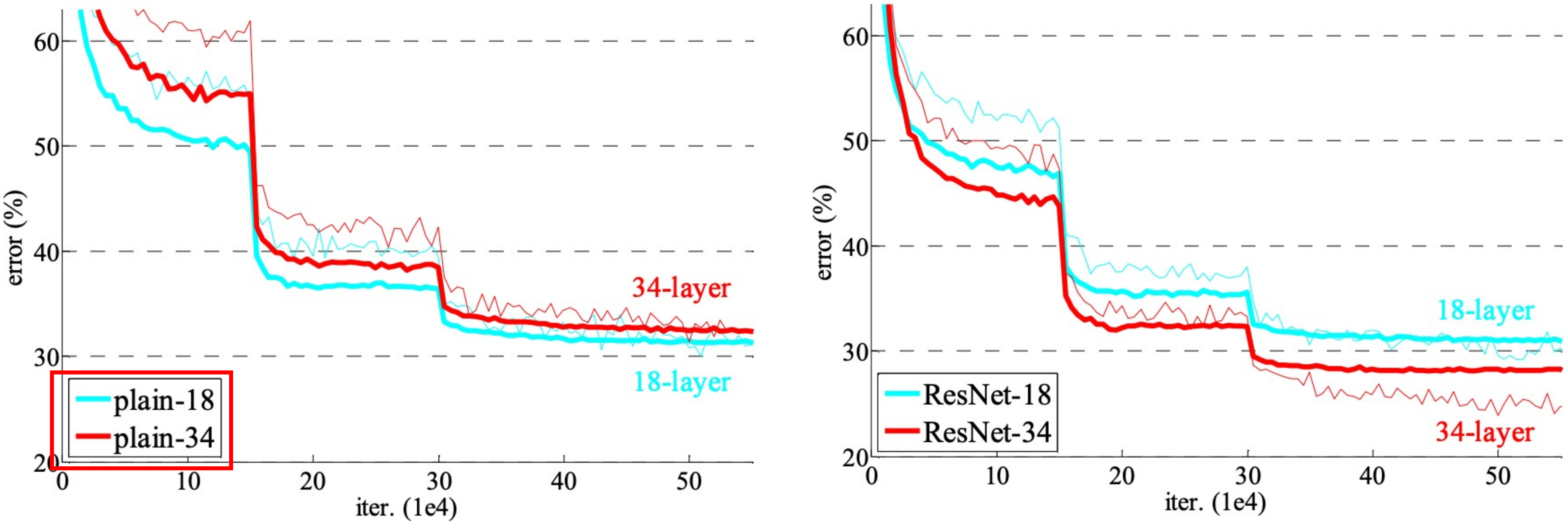
Reading material

ResNet

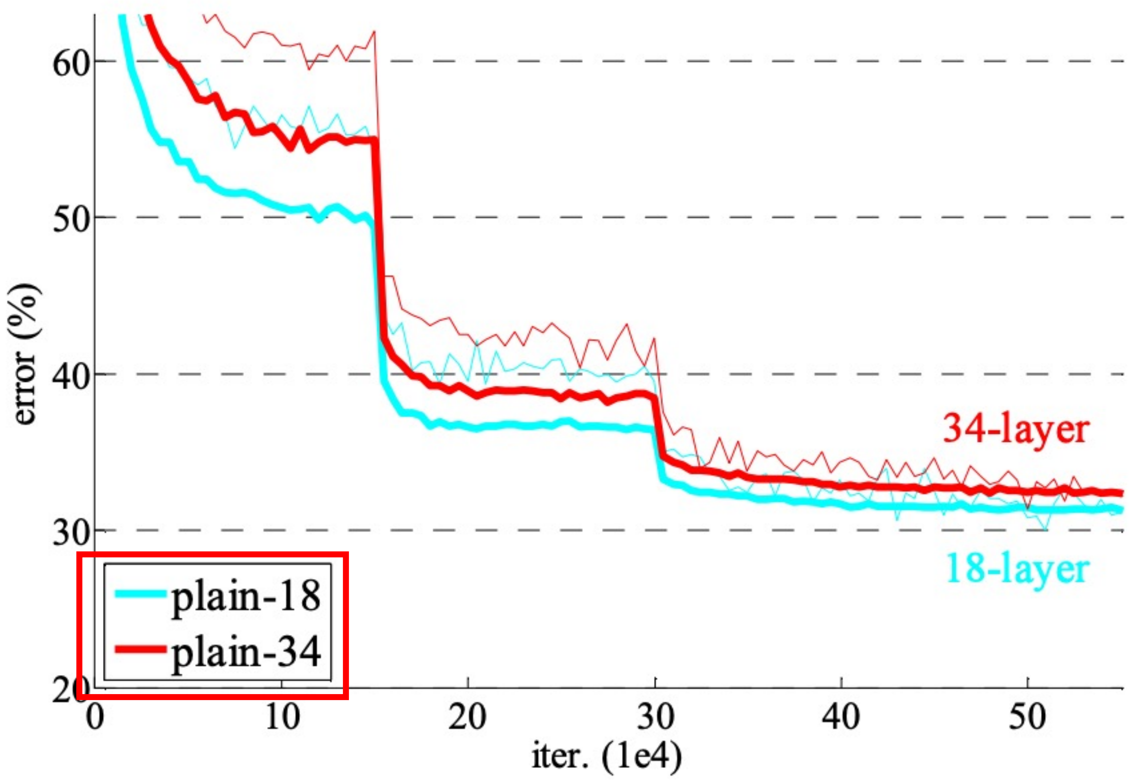

60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
34-layer
18-layer
plain-18
plain-34

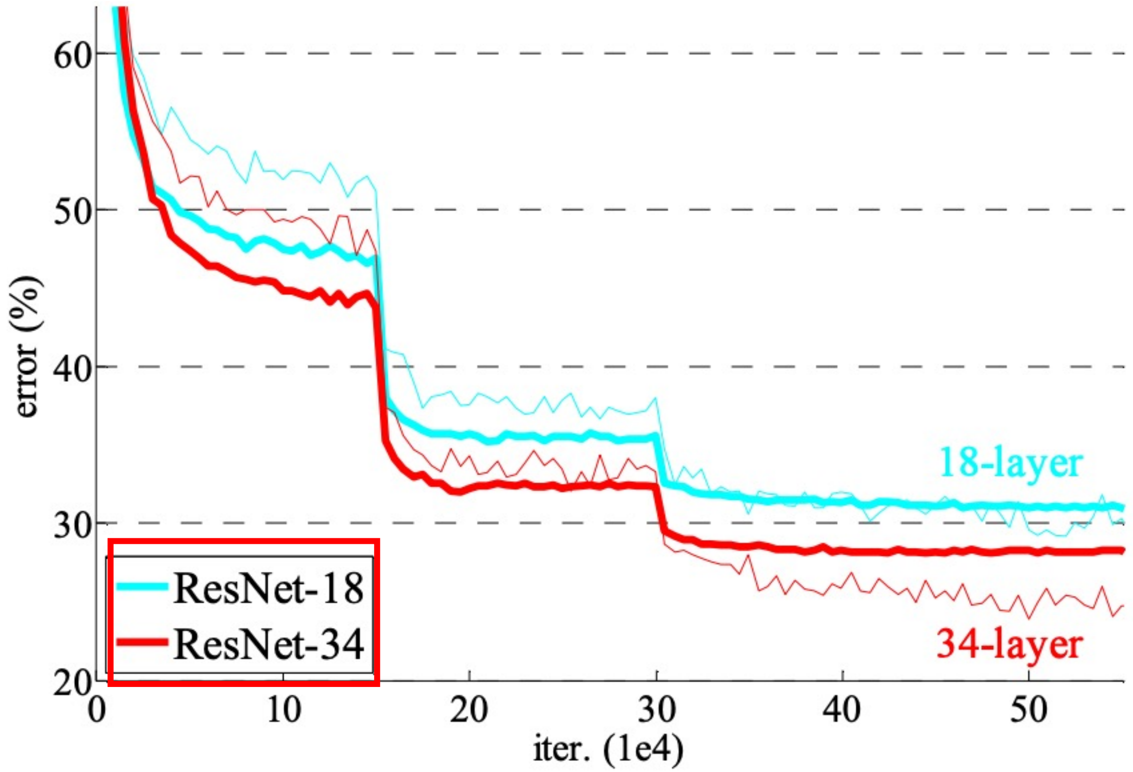

60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
18-layer
34-layer
ResNet-18
ResNet-34

ResNet

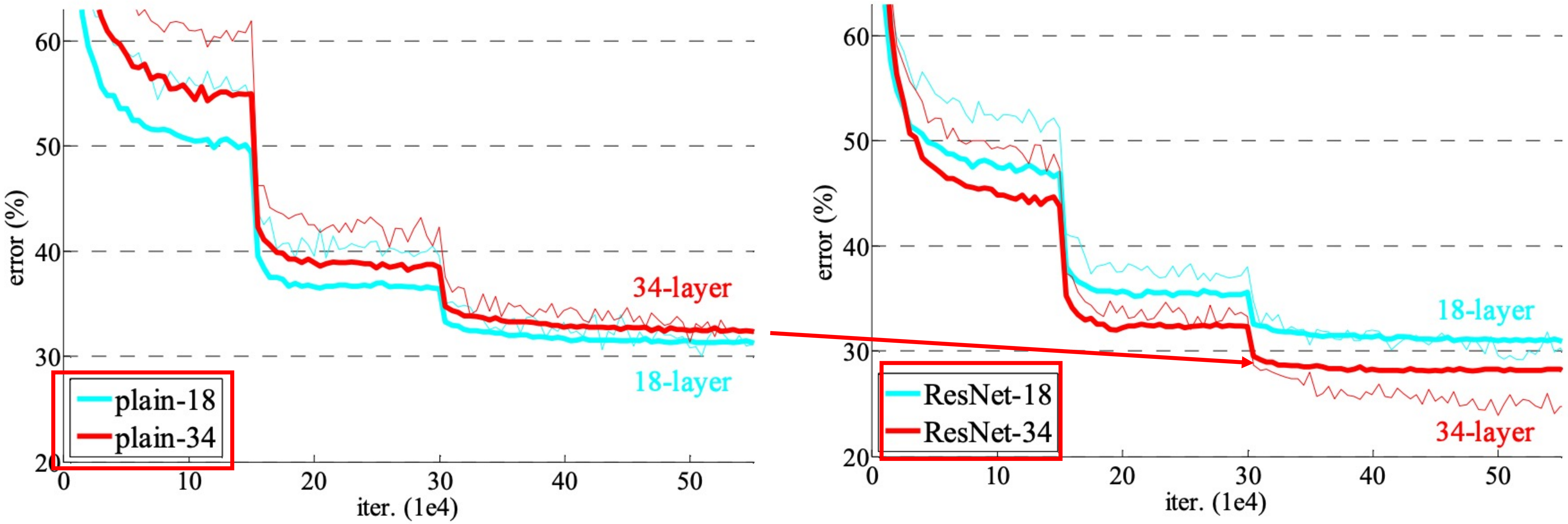

60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
34-layer
18-layer
plain-18
plain-34
60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
18-layer
34-layer
ResNet-18
ResNet-34

ResNet

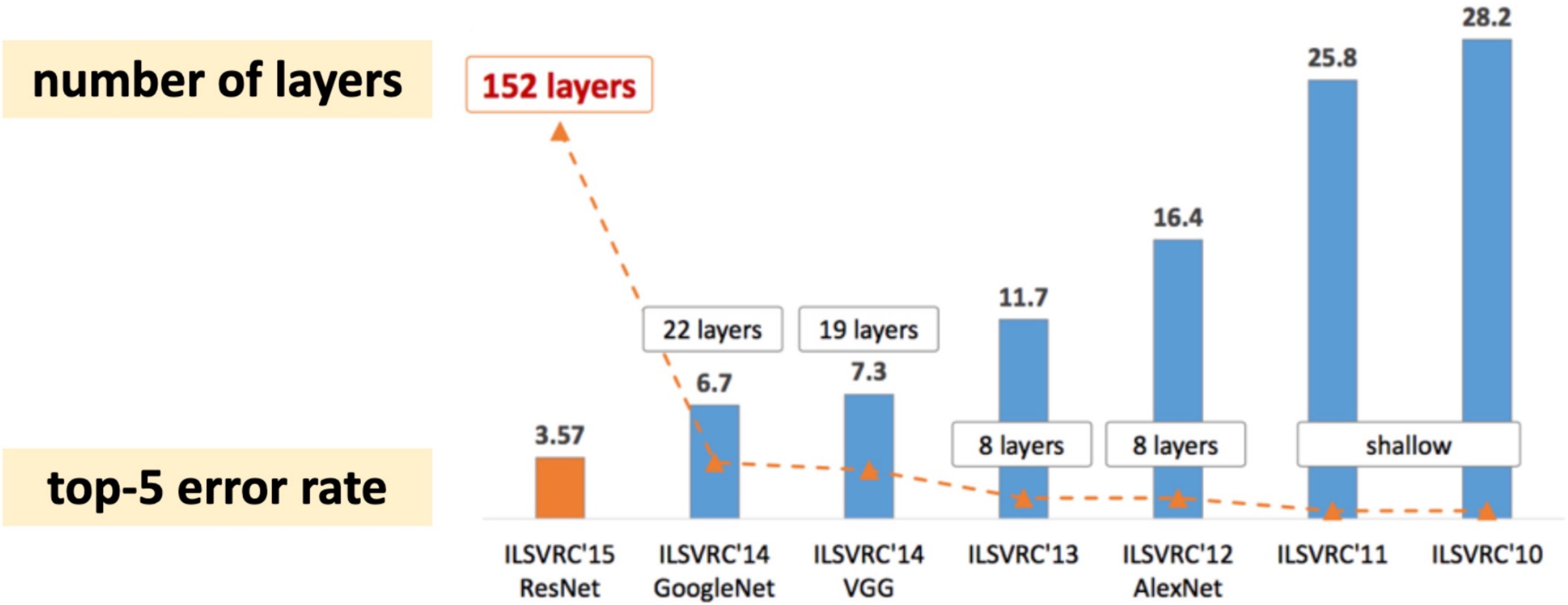

60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
34-layer
18-layer
plain-18
plain-34


60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
18-layer
34-layer
ResNet-18
ResNet-34

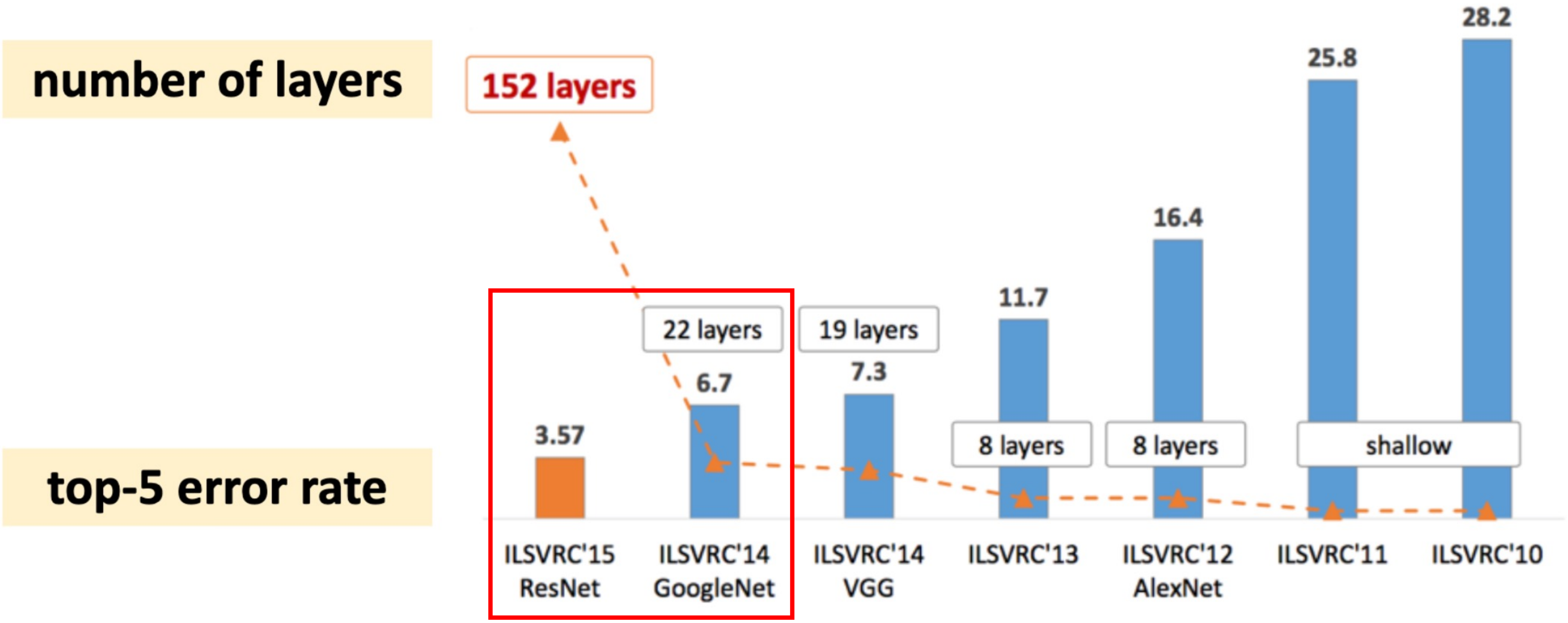
ResNet


60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
34-layer
18-layer
plain-18
plain-34
60
50
40
30
20
error (%)
0
10
20
30
40
50
iter. (1e4)
18-layer
34-layer
ResNet-18
ResNet-34

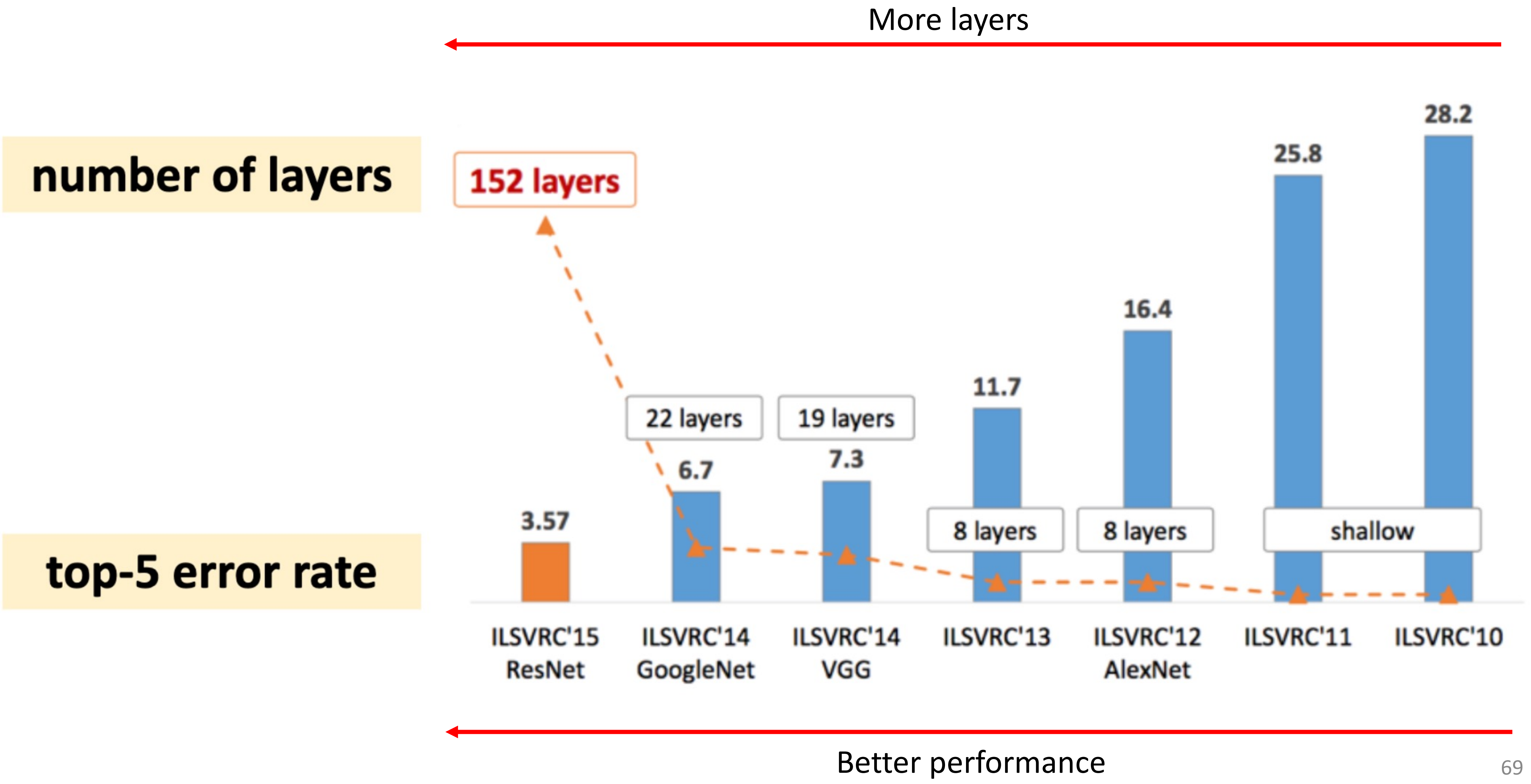
ImageNet competition winners



ImageNet competition winners



ImageNet competition winners



References

- [LetNet-5] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324.
- [AlexNet] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012): 1097-1105.
- [VGG] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014). Arxiv at https://arxiv.org/pdf/1409.1556.pdf

References

- [Inception] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9. 2015. ArXiv at https://arxiv.org/pdf/1409.4842.pdf (*Section 4 and 5)*
- [ResNet] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016. ArXiv at https://arxiv.org/pdf/1512.03385.pdf (*Section 3.1, 3.2 and 3.3)*