Learning Deep Representations for Visual Recognition

CVPR18/ECCV18 Tutorial

Kaiming He

Facebook AI Research (FAIR)

Deep Learning is Representation Learning

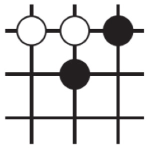
Representation Learning: worth a conference name ☺ (ICLR)

Represent (raw) data for machines to perform tasks:

- Vision: pixels, …
- Language: letters, …
- Speech: waves, …
- Games: status, …

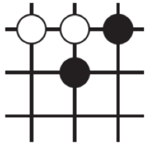
Representation Learning: AlphaGo

3^{361} states?



Representation Learning: AlphaGo

3^{361} states?

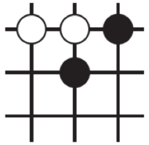


Bad representations

$256^{3*640*480}$?

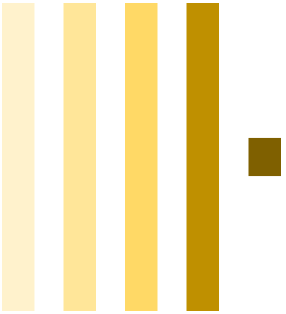
Representation Learning: AlphaGo

3^{361} states?

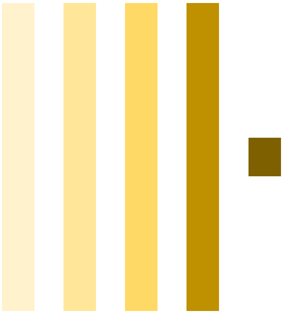


Bad representations

$256^{3*640*480}$?

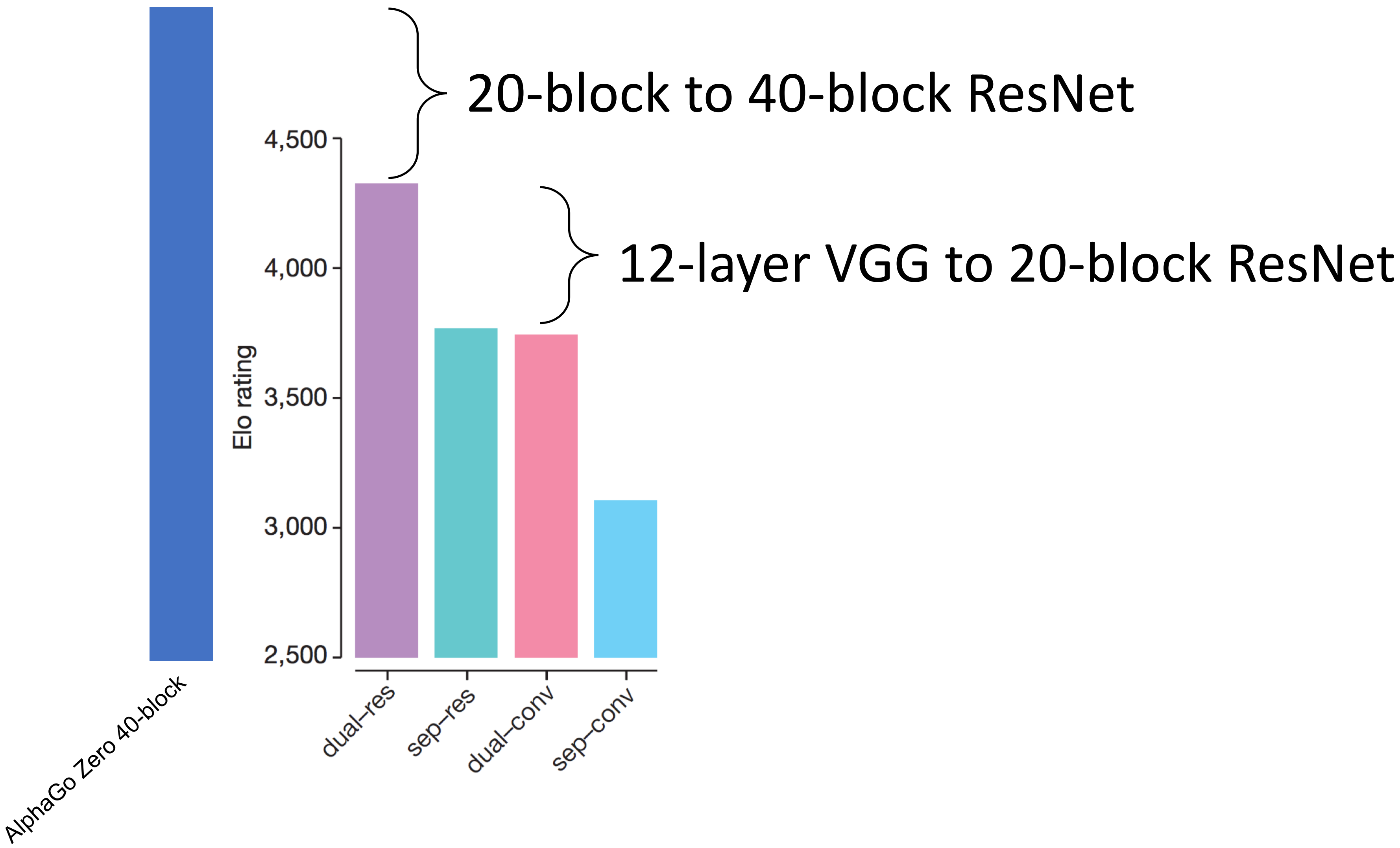


models
(now, neural nets)



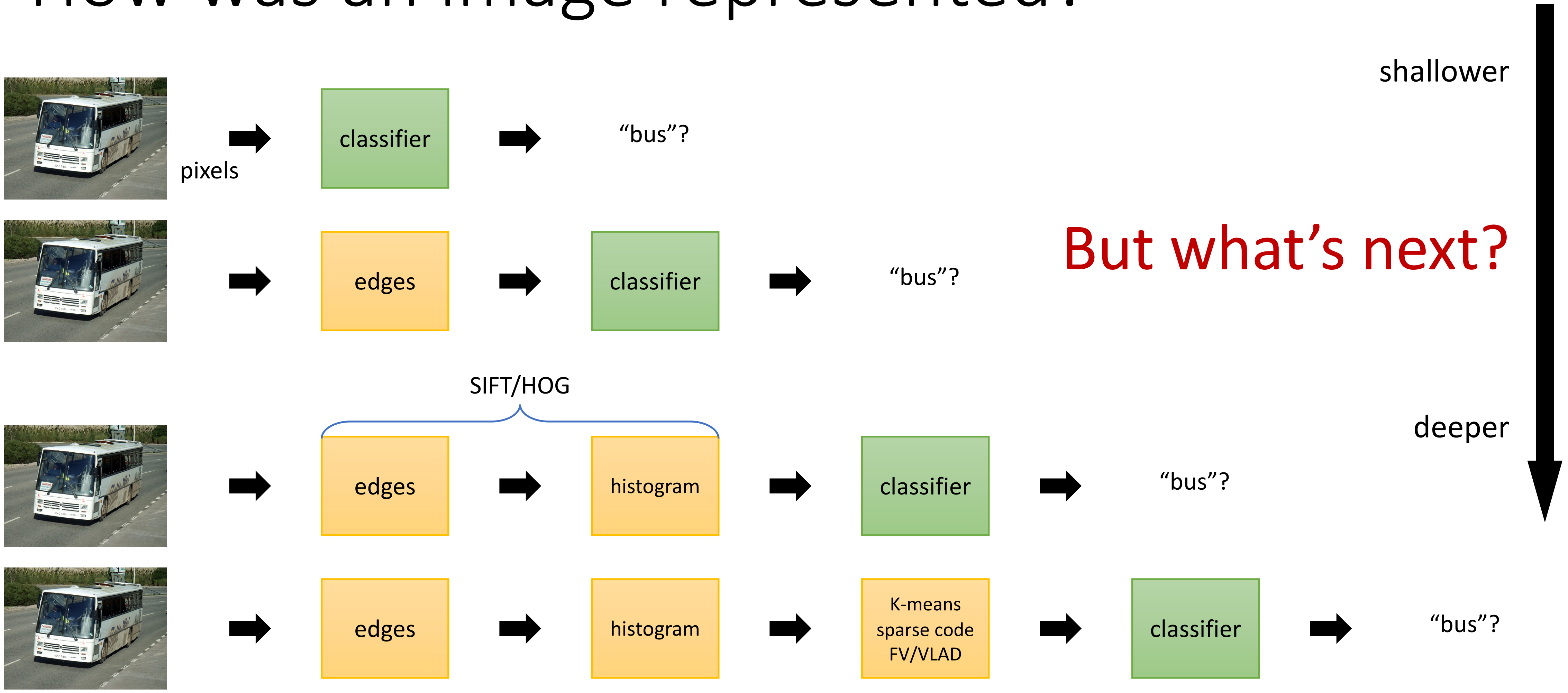
Good representations

Representation Learning: AlphaGo



"Mastering the game of Go without human knowledge", Silver et al. Nature 2017

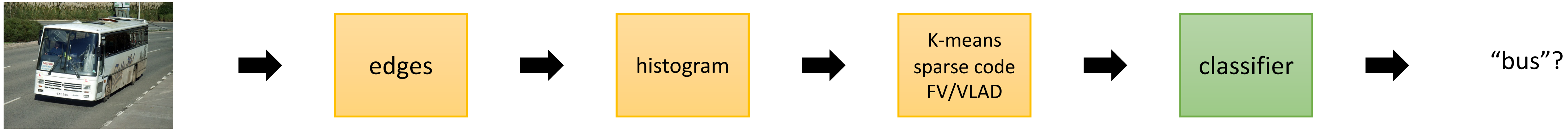
How was an image represented?



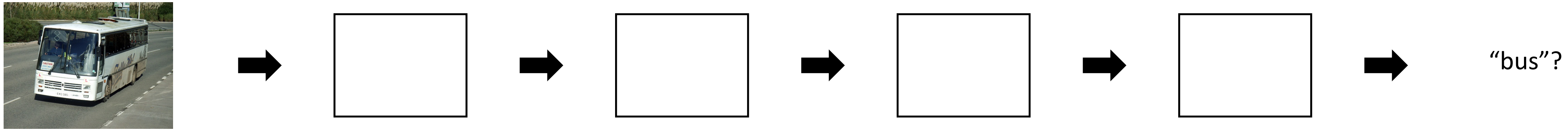
[Lowe 1999, 2004], [Sivic & Zisserman 2003], [Dalal & Triggs 2005], [Grauman & Darrell 2005]
[Lazebnik et al 2006], [Perronnin & Dance 2007], [Yang et al 2009], [Jégou et al 2010], ……

Learning to represent

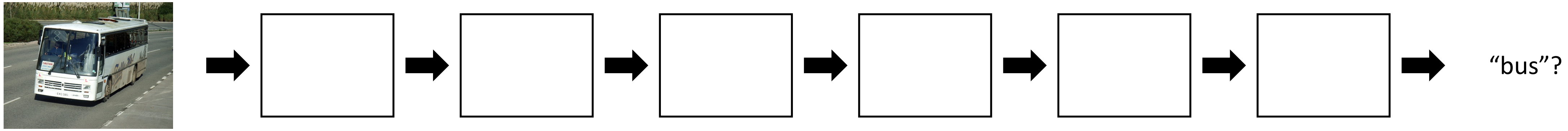
Specialized components, domain knowledge required



Generic components, less domain knowledge

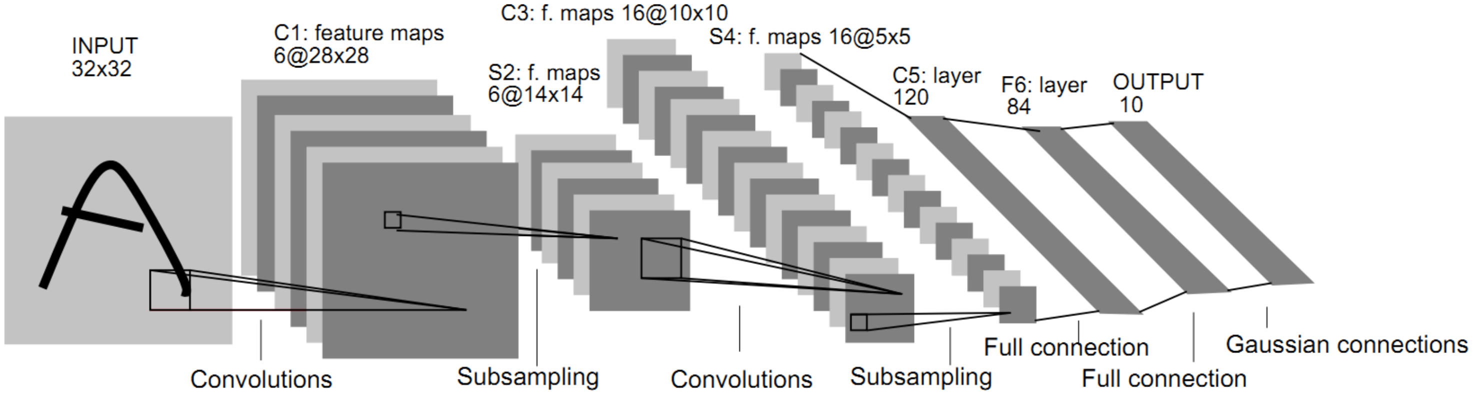


Repeat **elementary** layers: going deeper



- End-to-end by BackProp

LeNet



- Convolution:
  - locally-connected
  - spatially weight-sharing
    - weight-sharing is a key in DL (e.g., RNN shares weights temporally)
- Subsampling
- Fully-connected outputs
- Train by BackProp

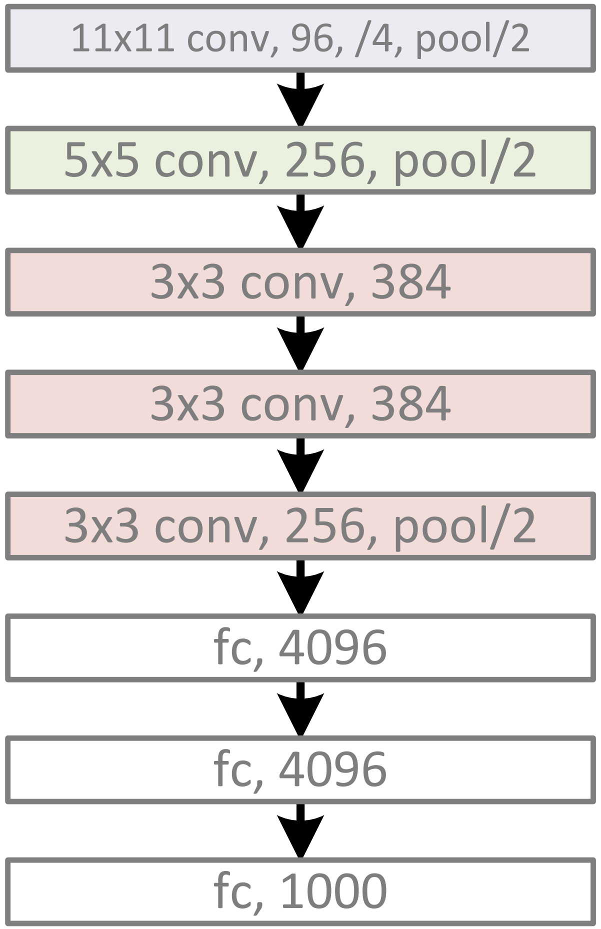
- All are still the basic components of modern ConvNets!

“Gradient-based learning applied to document recognition”, LeCun et al. 1998
“Backpropagation applied to handwritten zip code recognition”, LeCun et al. 1989

AlexNet

LeNet-style backbone, plus:

- ReLU [Nair & Hinton 2010]
  - “RevoLUtion of deep learning”*
  - Accelerate training; better grad prop (vs. tanh)
- Dropout [Hinton et al 2012]
  - In-network ensembling
  - Reduce overfitting (might be instead done by BN)
- Data augmentation
  - Label-preserving transformation
  - Reduce overfitting



*Quote Christian Szegedy

“ImageNet Classification with Deep Convolutional Neural Networks”, Krizhevsky, Sutskever, Hinton. NIPS 2012

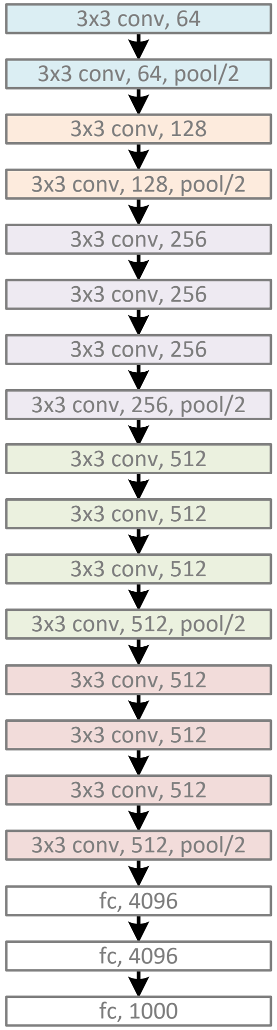
VGG-16/19

"16 layers are beyond my imagination!"
-- after ILSVRC 2014 result was announced.

Simply "Very Deep"!

- Modularized design
  - 3x3 Conv as the module
  - Stack the same module
  - Same computation for each module (1/2 spatial size => 2x filters)

- Stage-wise training
  - VGG-11 => VGG-13 => VGG-16
  - We need a better initialization…



"Very Deep Convolutional Networks for Large-Scale Image Recognition", Simonyan & Zisserman. arXiv 2014 (ICLR 2015)

Initialization Methods

- Analytical formulations of <u>normalizing</u> forward/backward signals
- Based on strong assumptions (like Gaussian distributions)

- Xavier Init (linear): $n \cdot Var[w] = 1$
- MSRA Init (ReLU): $n \cdot Var[w] = 2$

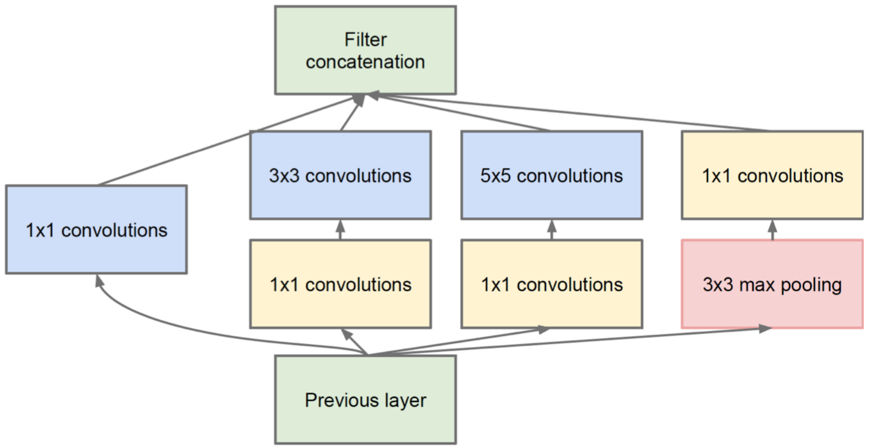
"Efficient Backprop", LeCun et al, 1998
"Understanding the difficulty of training deep feedforward neural networks" Glorot & Bengio, 2010
"Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification" Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun, ICCV 2015

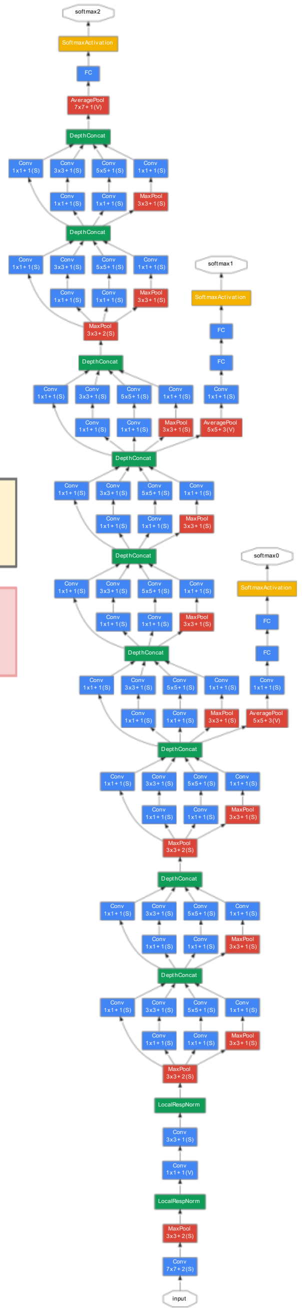
GoogleNet/Inception

Accurate with small footprint.

My take on GoogleNets:

- Multiple branches
  - e.g., 1x1, 3x3, 5x5, pool
- Shortcuts
  - stand-alone 1x1, merged by concat.
- Bottleneck
  - Reduce dim by 1x1 before expensive 3x3/5x5 conv



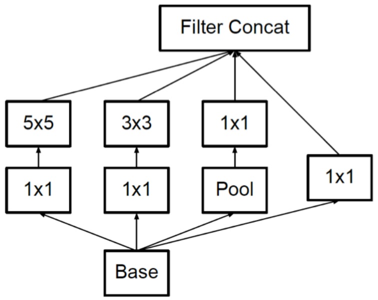


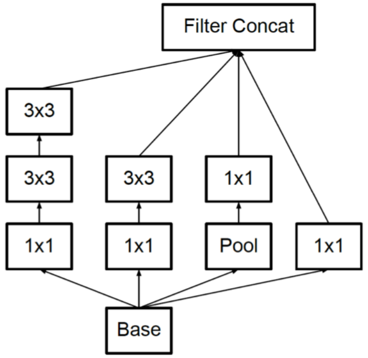
Szegedy et al. “Going deeper with convolutions”. arXiv 2014 (CVPR 2015)

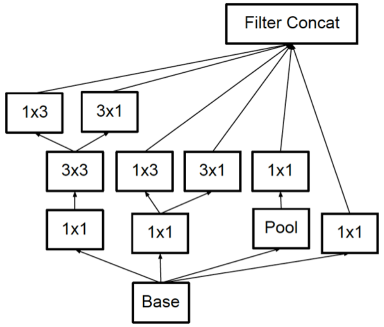
GoogleNet/Inception v1, v2, v3, …

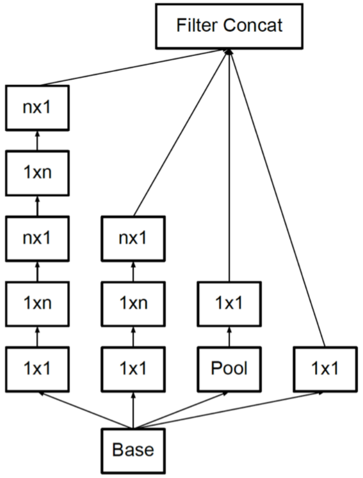
More templates, but the same 3 main properties are kept:

- Multiple branches
- Shortcuts (1x1, concate.)
- Bottleneck







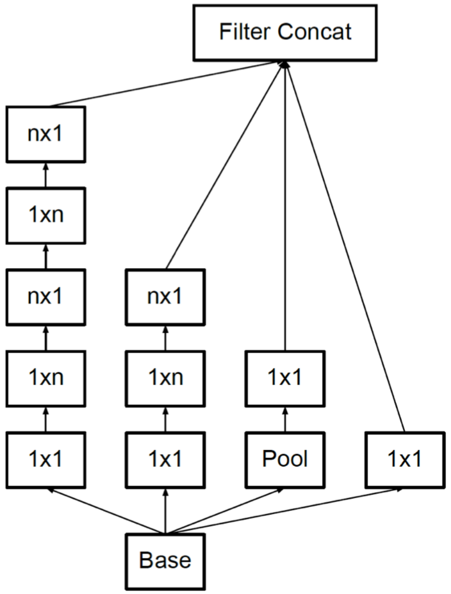


Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". arXiv 2015 (CVPR 2016)

Batch Normalization (BN)

- Xavier/MSRA init are not directly applicable for multi-branch nets

- Optimizing multi-branch ConvNets largely benefits from BN
  - including all Inceptions and ResNets



Ioffe & Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML 2015

Batch Normalization (BN)

- Recap: Normalizing image input (LeCun et al 1998 “Efficient Backprop”)
- Xavier/MSRA init: Analytic normalizing each layer
- BN: data-driven normalization, for each layer, for each mini-batch
  - Greatly accelerate training
  - Less sensitive to initialization
  - Improve regularization

Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015

Batch Normalization (BN)

$$x \quad ➡ \quad \hat{x} = \frac{x - \mu}{\sigma} \quad ➡ \quad y = \gamma \hat{x} + \beta$$

- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

- μ, σ: functions of x, analogous to responses
- γ, β: parameters to be learned, analogous to weights

Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015

Batch Normalization (BN)

$$x \;\Rightarrow\; \hat{x} = \frac{x - \mu}{\sigma} \;\Rightarrow\; y = \gamma \hat{x} + \beta$$

2 modes of BN:

- Train mode:
  - μ, σ are functions of a batch of x
- Test mode:
  - μ, σ are pre-computed on training set

Caution: make sure your BN usage is correct!
(this causes many of my bugs in my research experience!)

Ioffe & Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML 2015

Batch Normalization (BN)

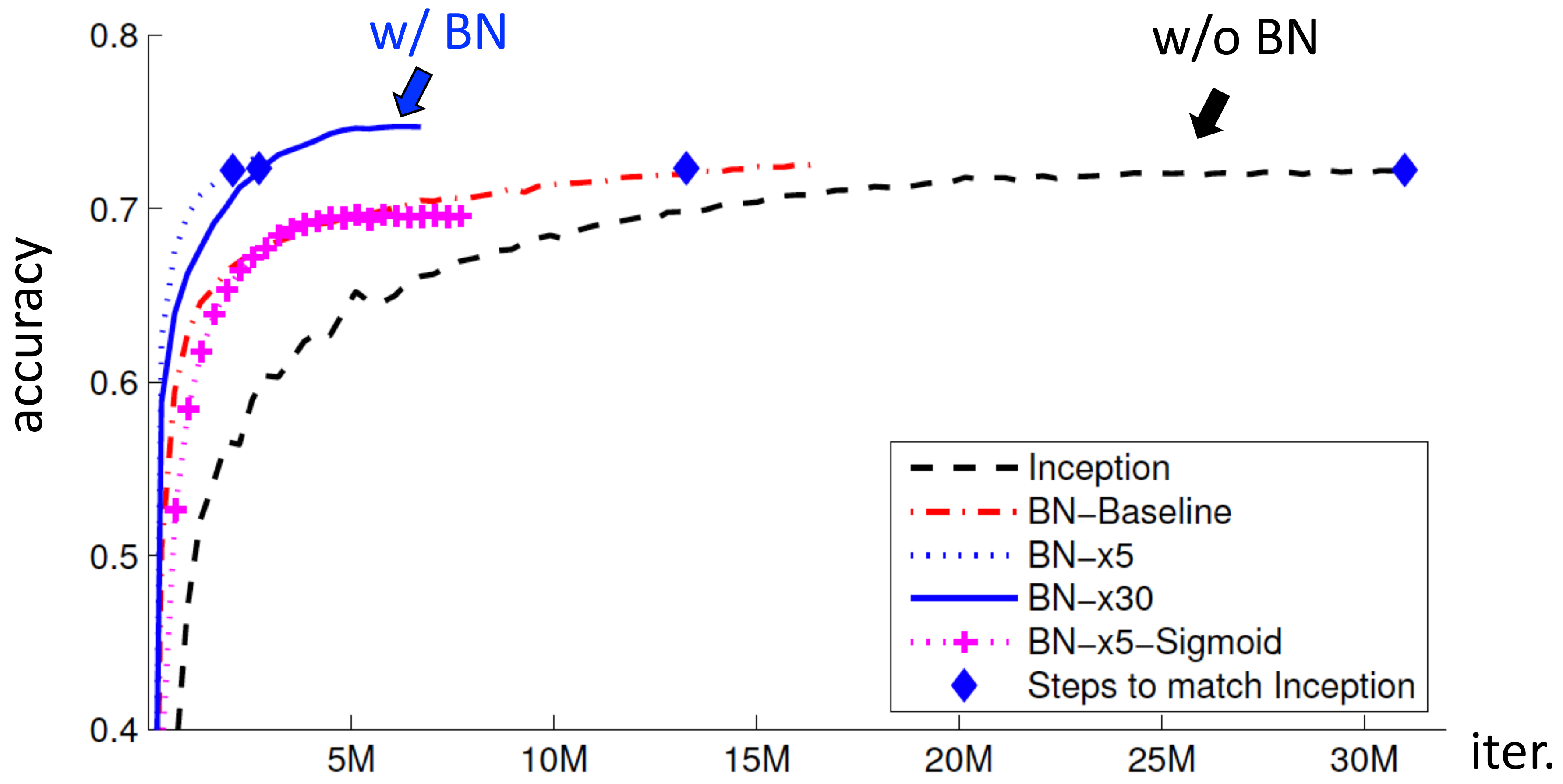


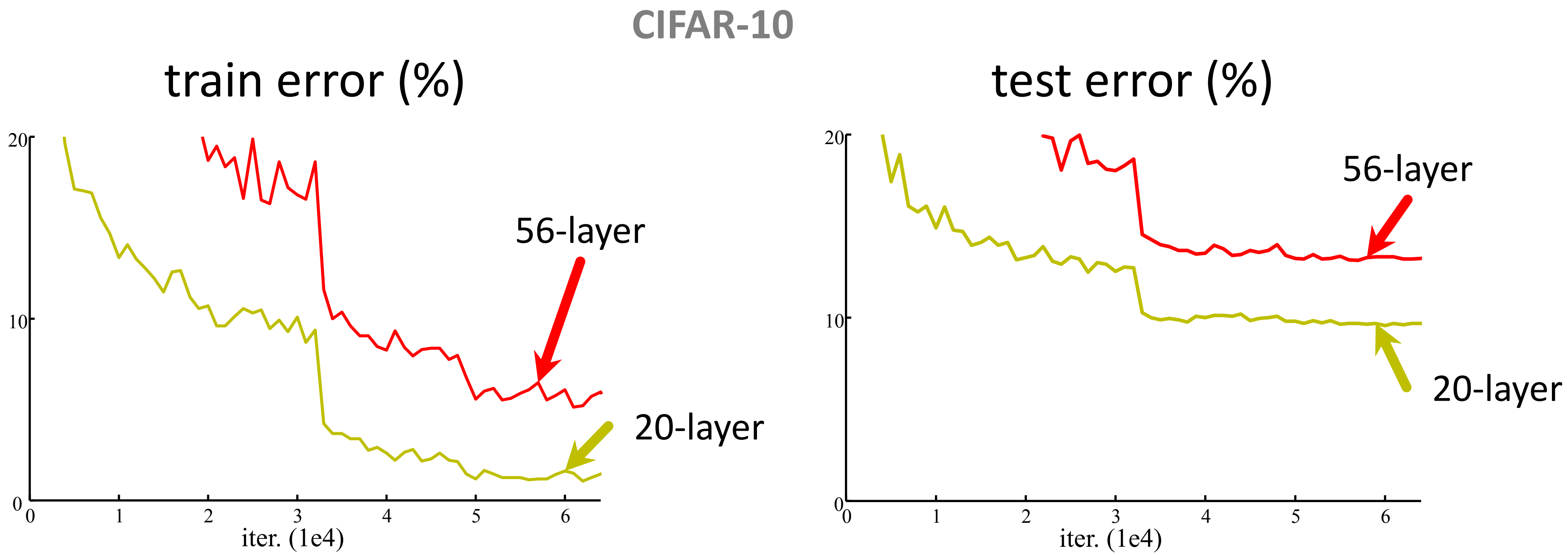
Figure credit: Ioffe & Szegedy

Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015

ResNets

Credit: ???

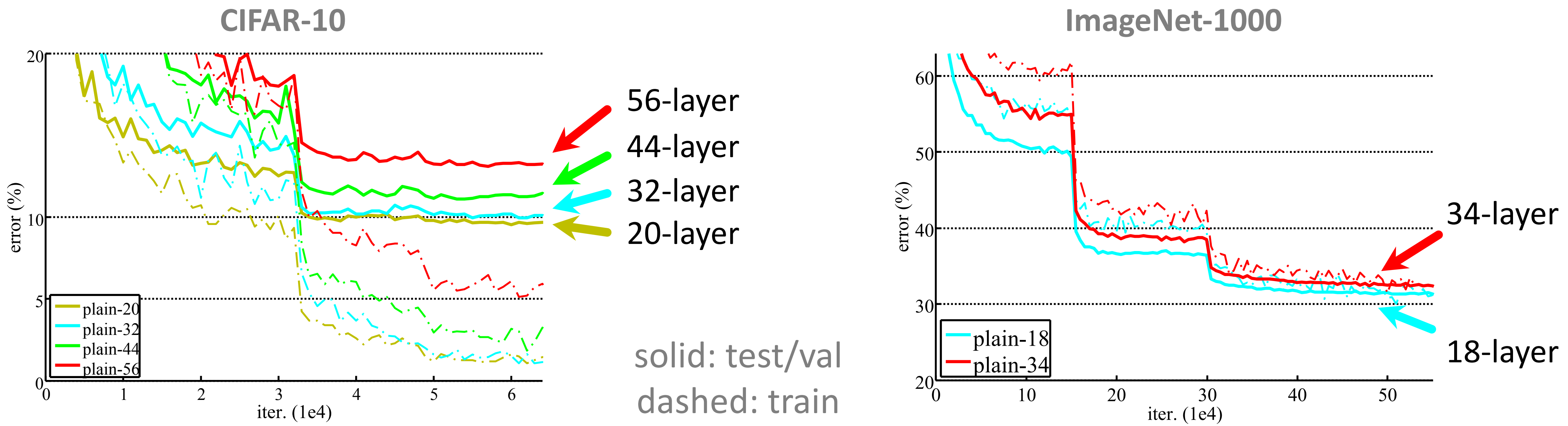
Simply stacking layers?



- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

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

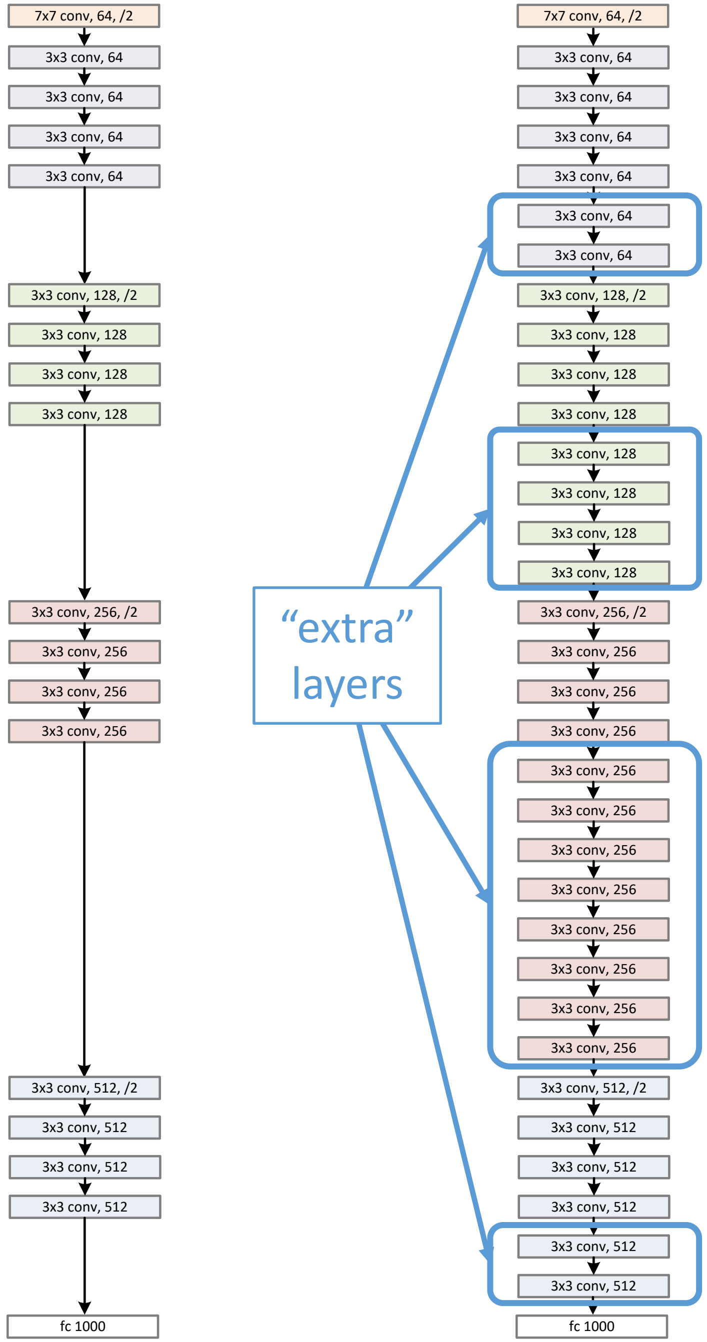
Simply stacking layers?



- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

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

a shallower model
(18 layers)



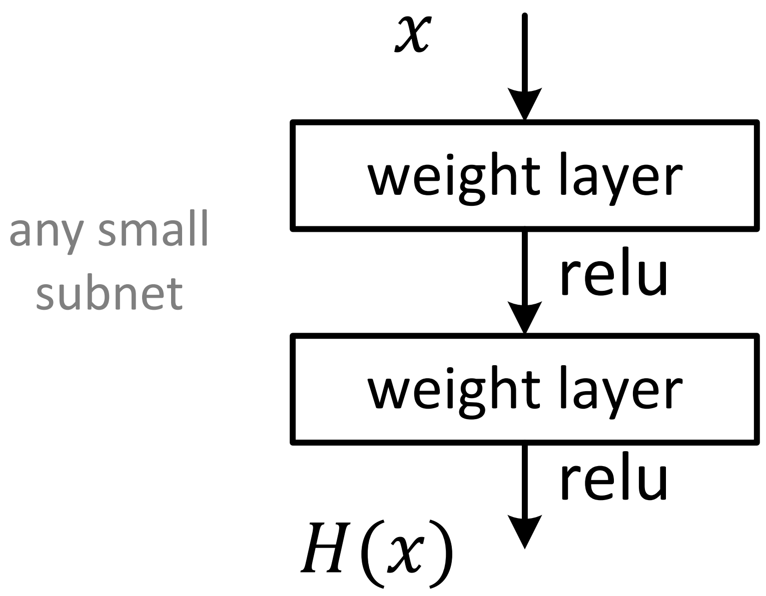
a deeper counterpart
(34 layers)

- Richer solution space
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution when going deeper…

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Deep Residual Learning

- Plain net



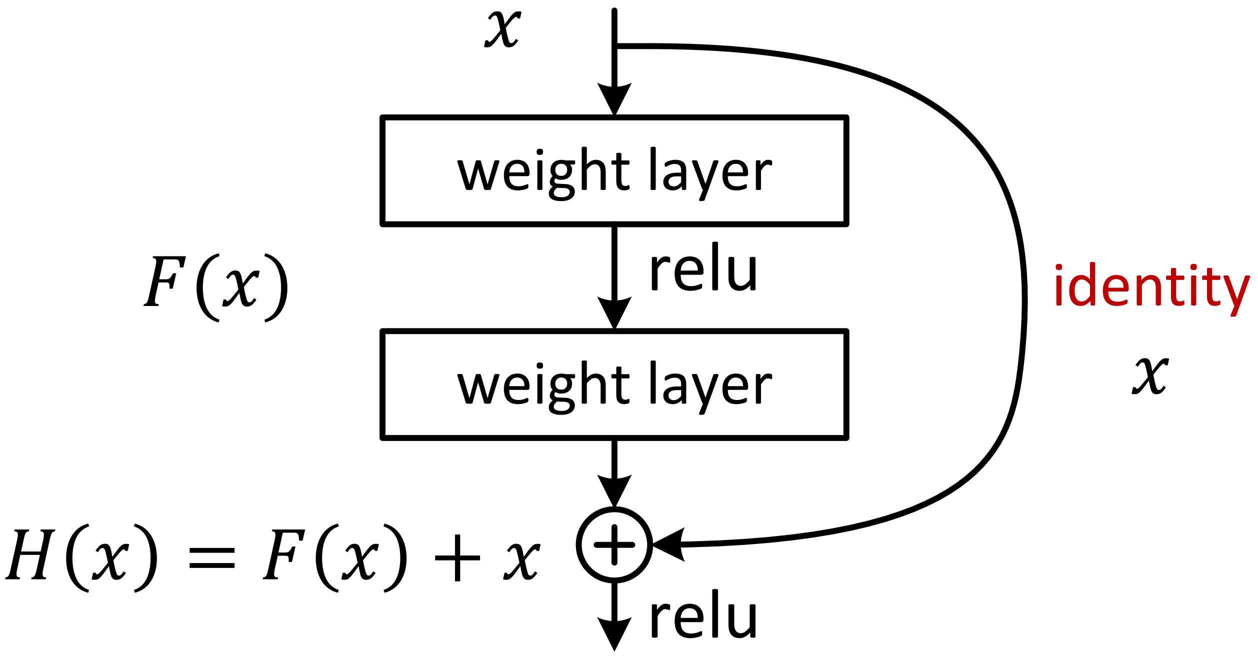
$H(x)$ is any desired mapping,

hope the small subnet fit $H(x)$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

~~hope the small subnet fit $H(x)$~~

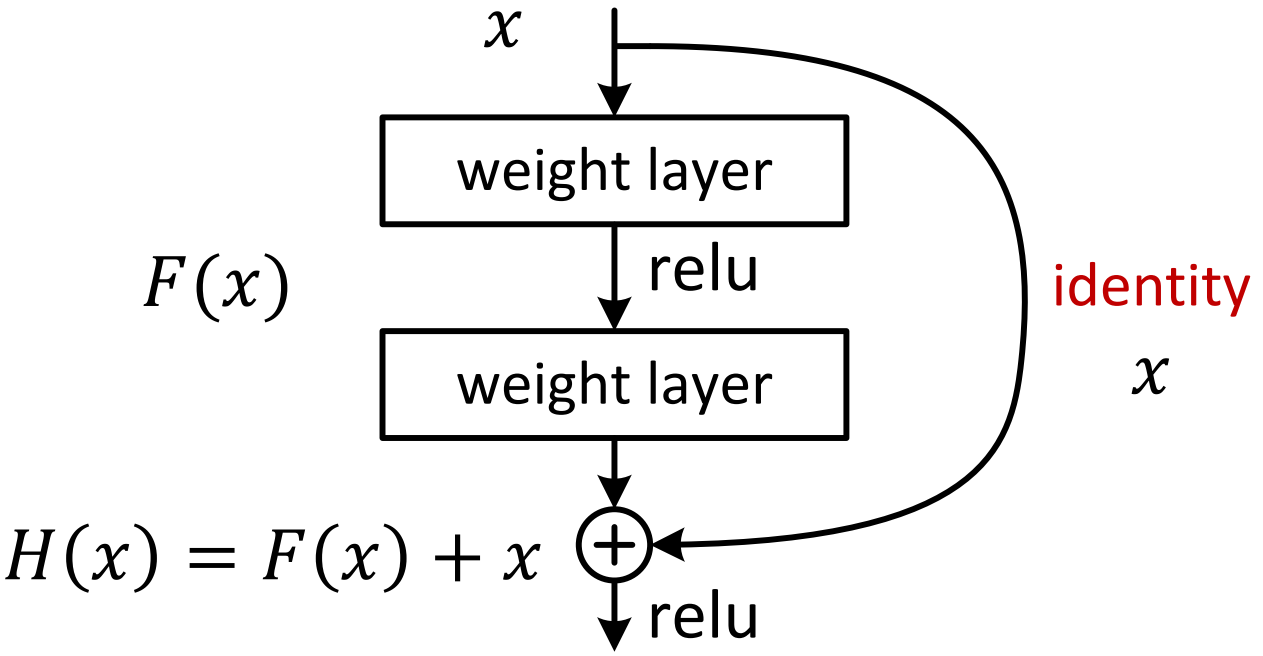
hope the small subnet fit $F(x)$

let $H(x) = F(x) + x$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Deep Residual Learning

- $F(x)$ is a residual mapping w.r.t. identity



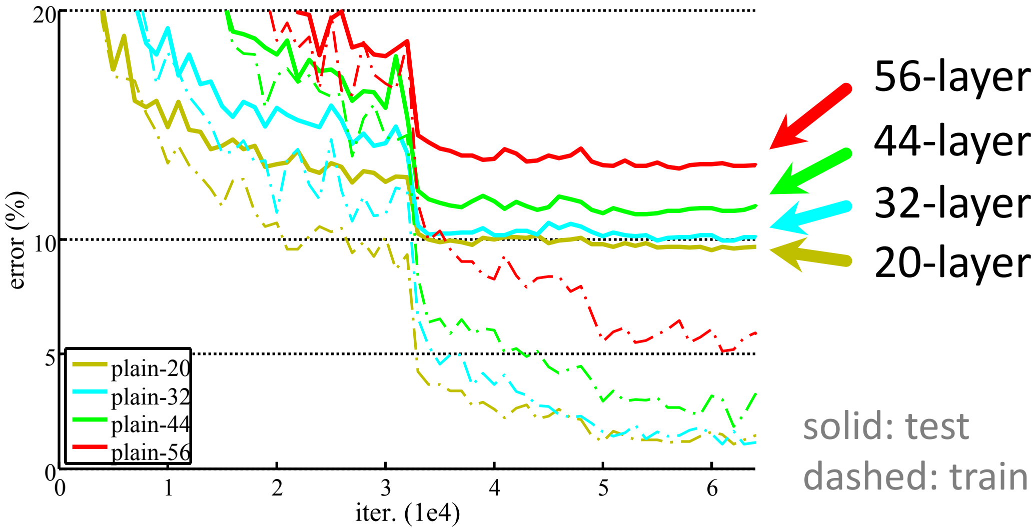
- If identity were optimal, easy to set weights as 0

- If optimal mapping is closer to identity, easier to find small fluctuations

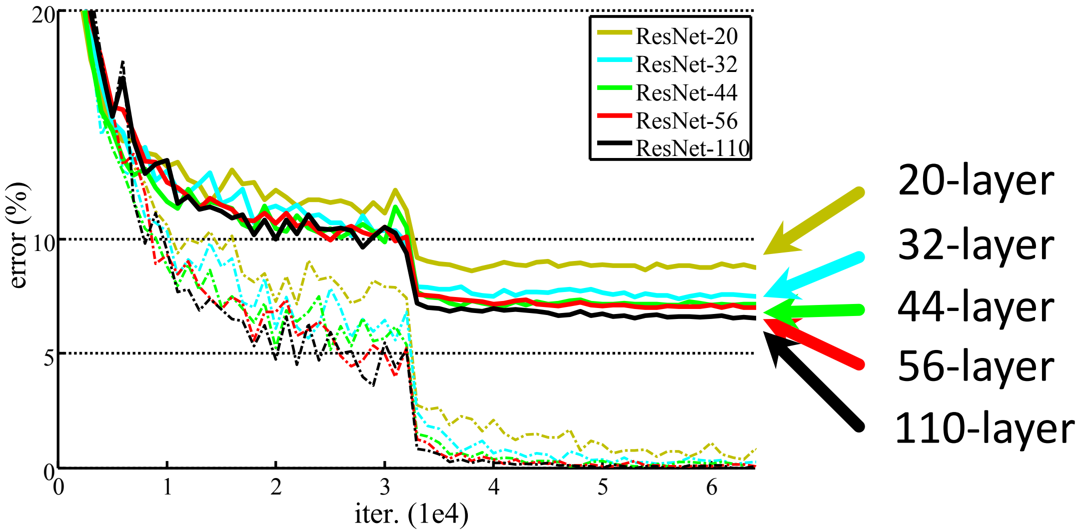
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

CIFAR-10 experiments

CIFAR-10 plain nets



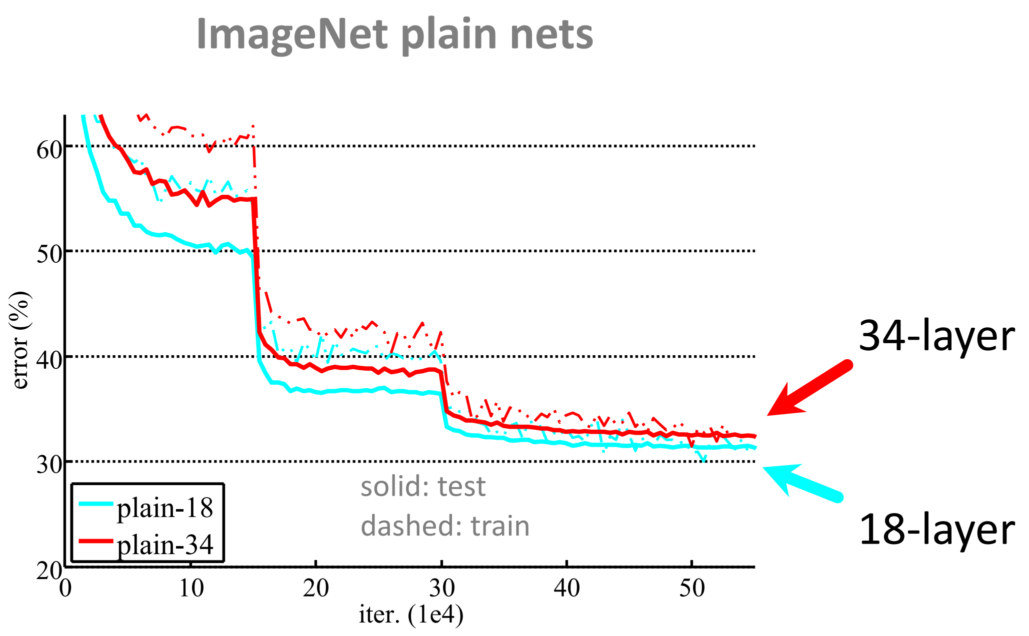
CIFAR-10 ResNets

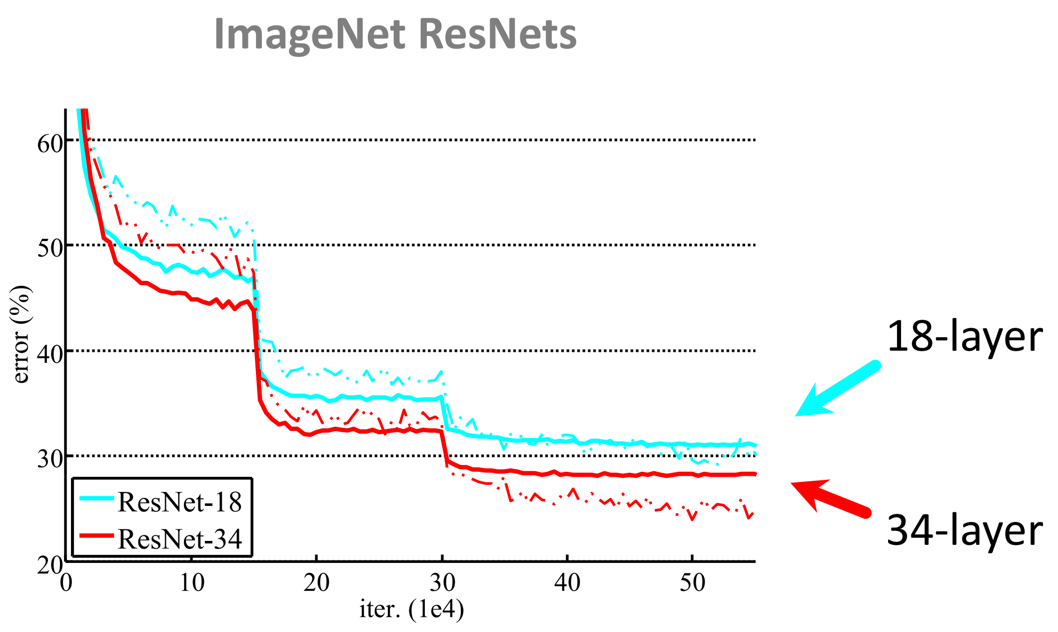


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

ImageNet experiments



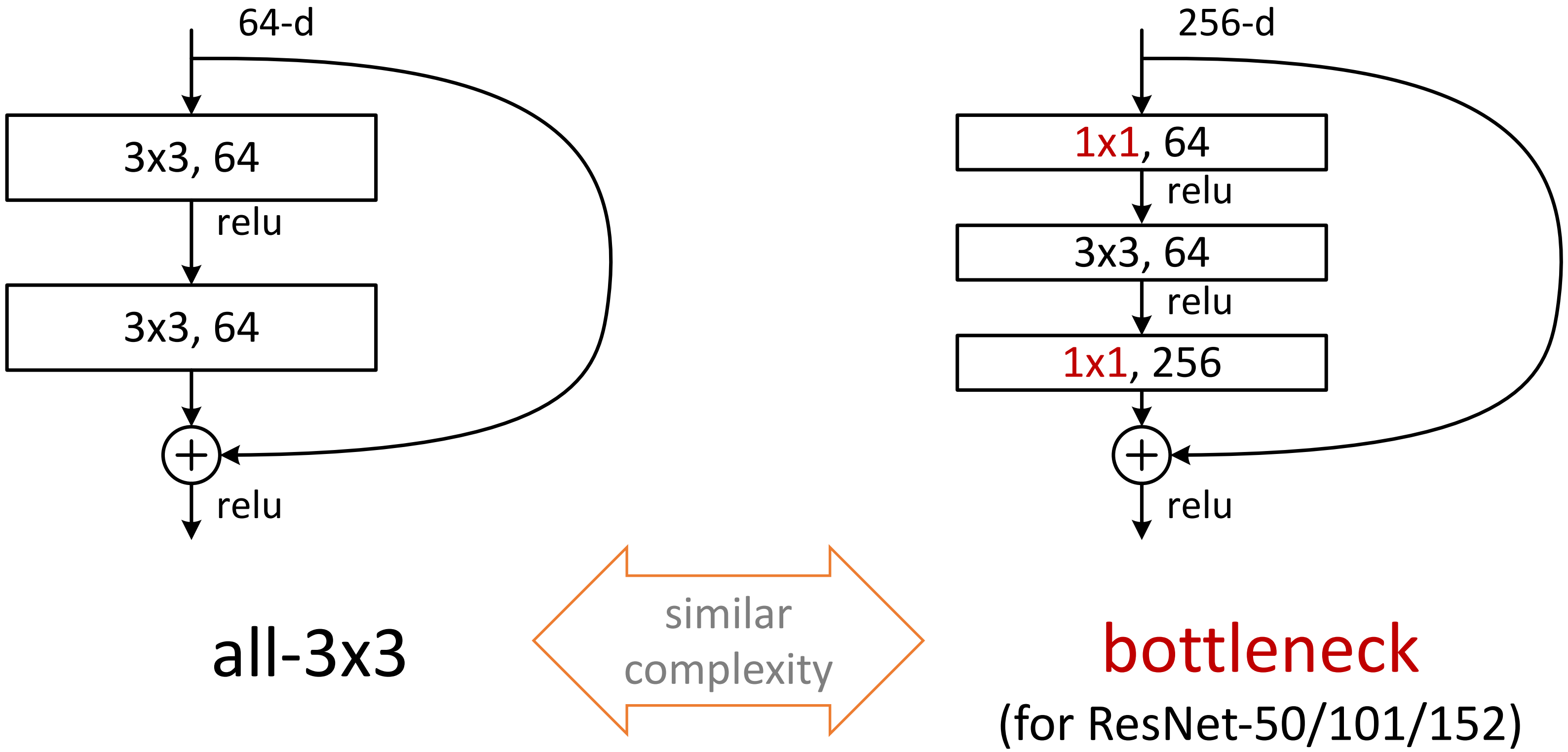


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

ImageNet experiments

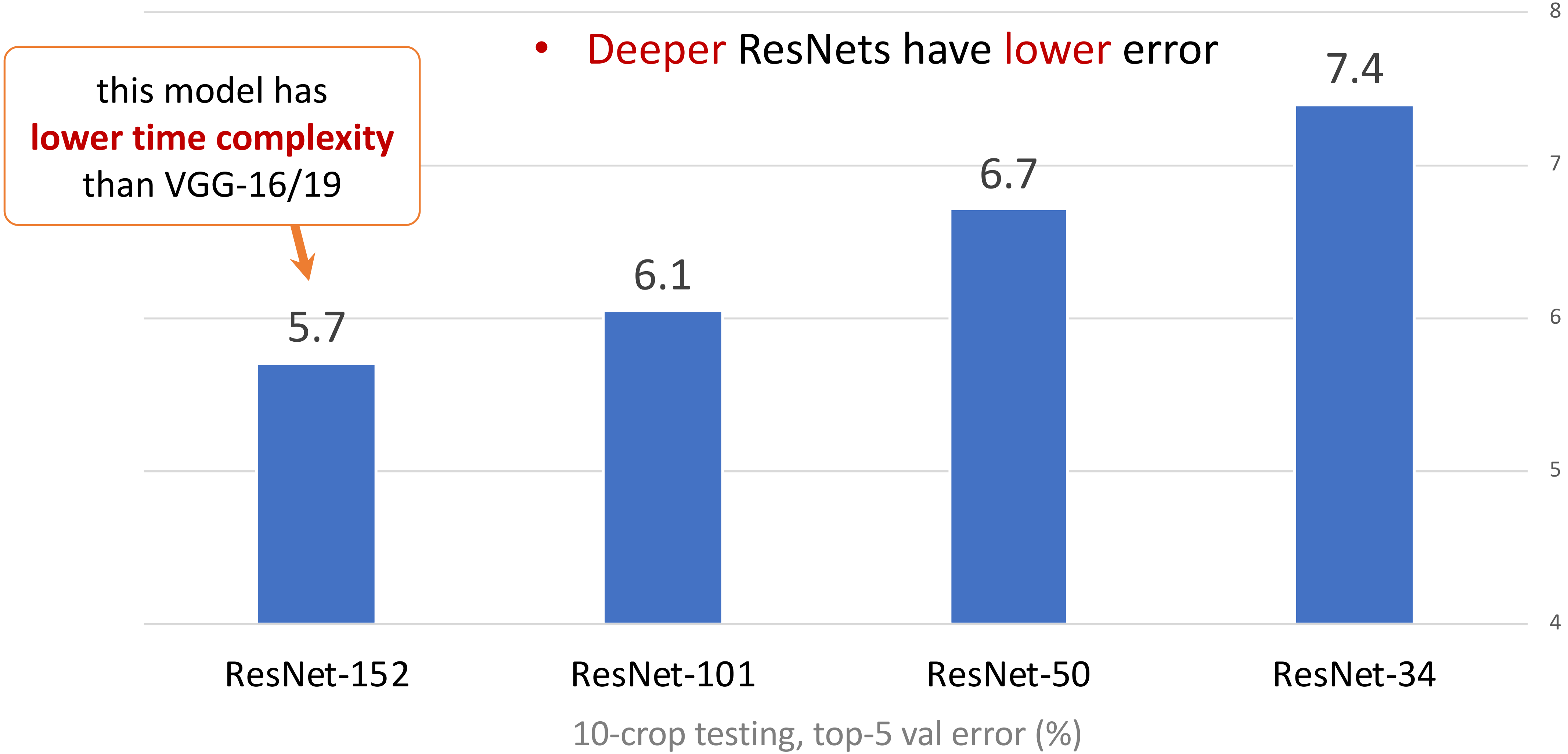
- A practical design of going deeper



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

ImageNet experiments

- Deeper ResNets have lower error

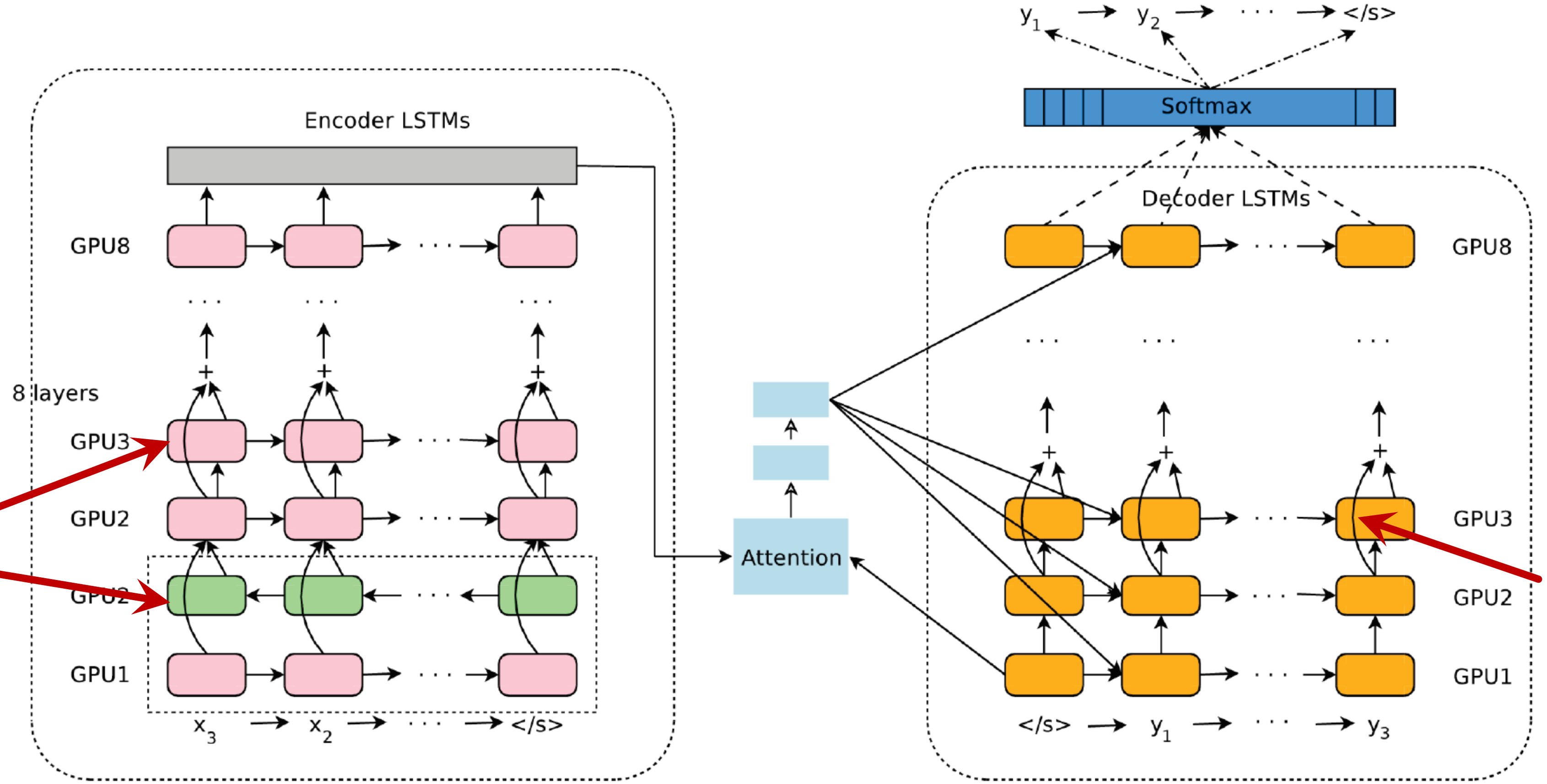


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

ResNet beyond computer vision

- **Neural Machine Translation** (NMT): 8-layer LSTM!

residual connections



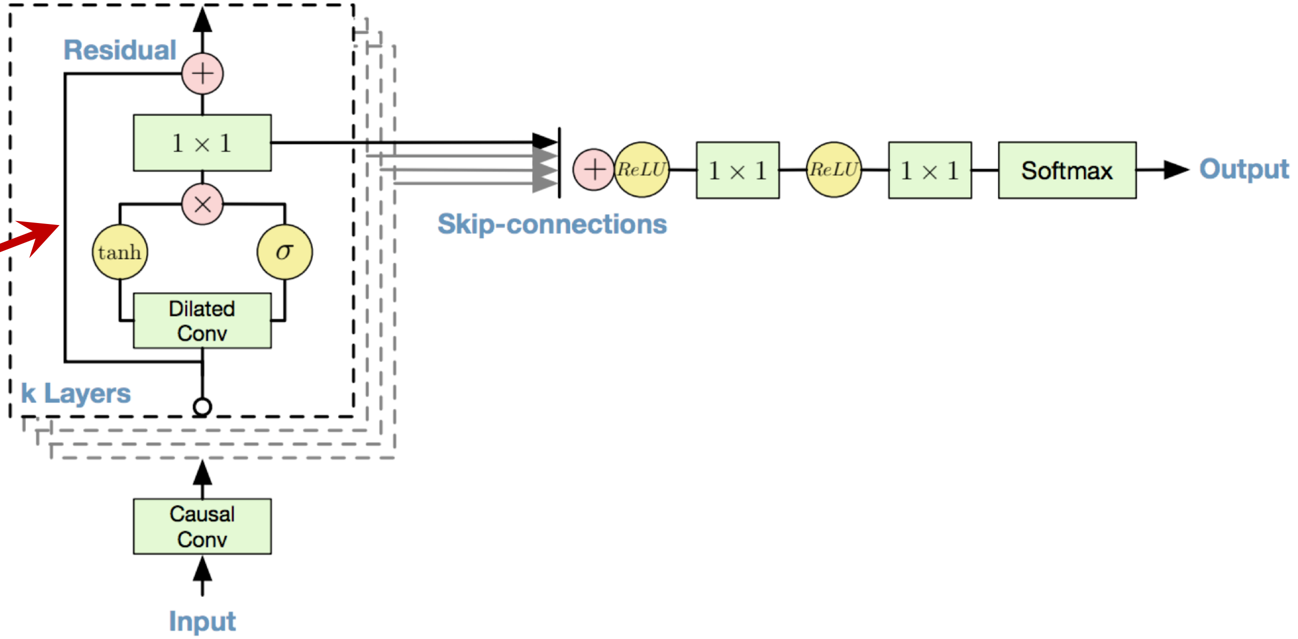
residual connections

Wu et al. “Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. arXiv 2016.

ResNet beyond computer vision

- **Speech Synthesis** (WaveNet): Residual CNNs on 1-d sequence

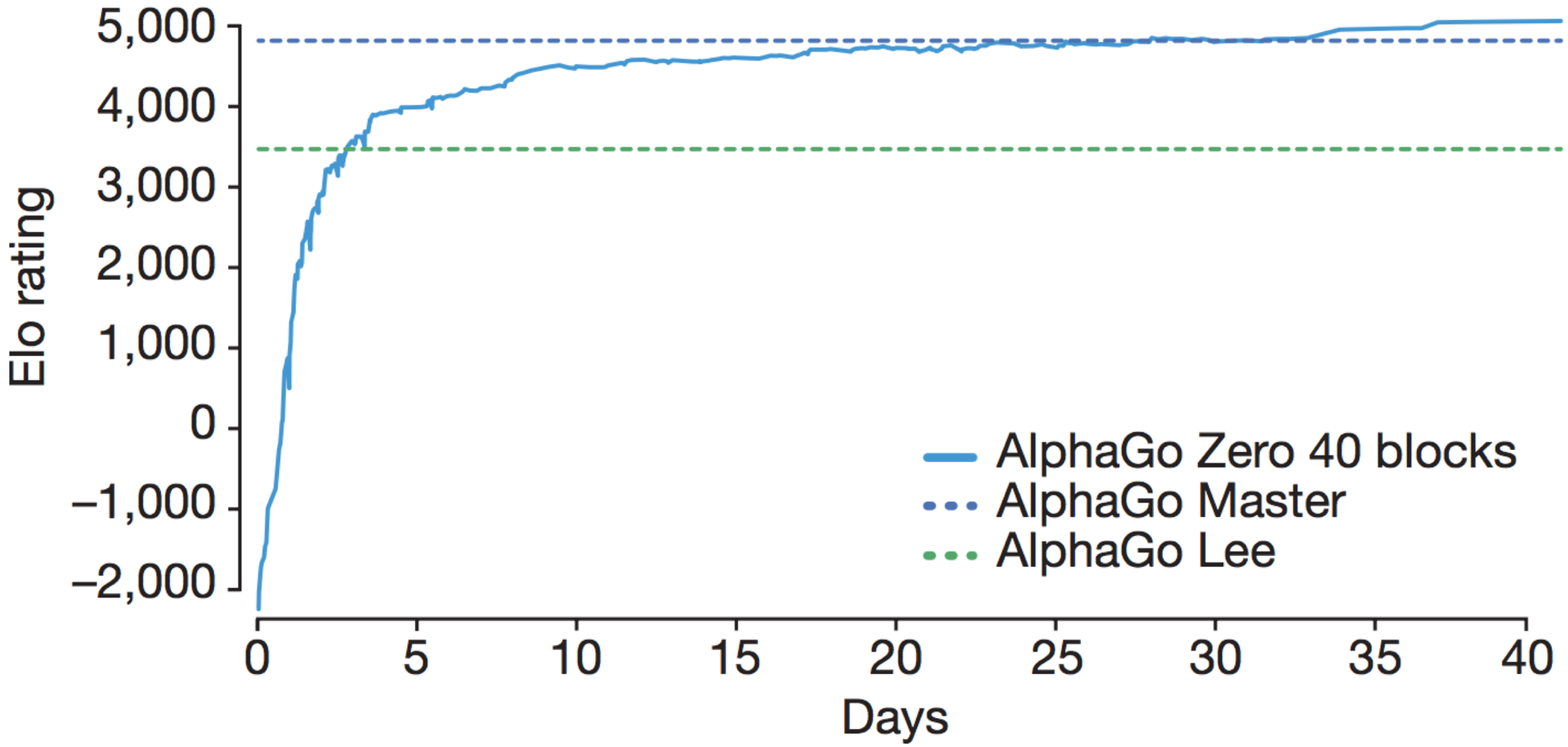
residual
connections



van den Oord et al. "WaveNet: A Generative Model for Raw Audio". arXiv 2016.

ResNet beyond computer vision

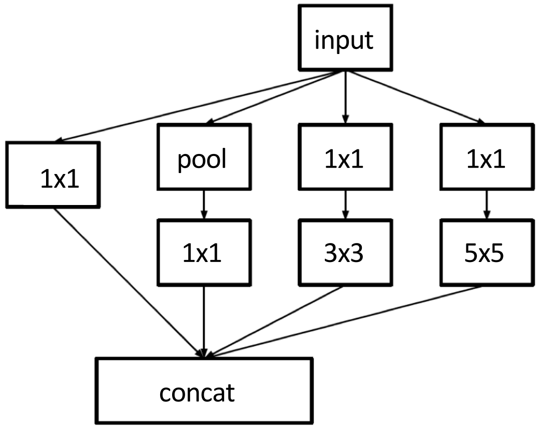
- **AlphaGo Zero**: 40 Residual Blocks



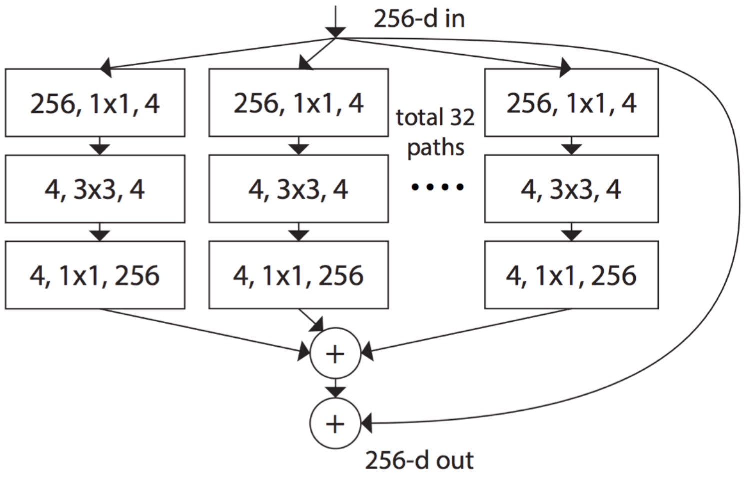
“Mastering the game of Go without human knowledge”, Silver et al. Nature 2017

ResNeXt

- Recap: shortcut, bottleneck, and <u>multi-branch</u>



Inception:
heterogeneous multi-branch

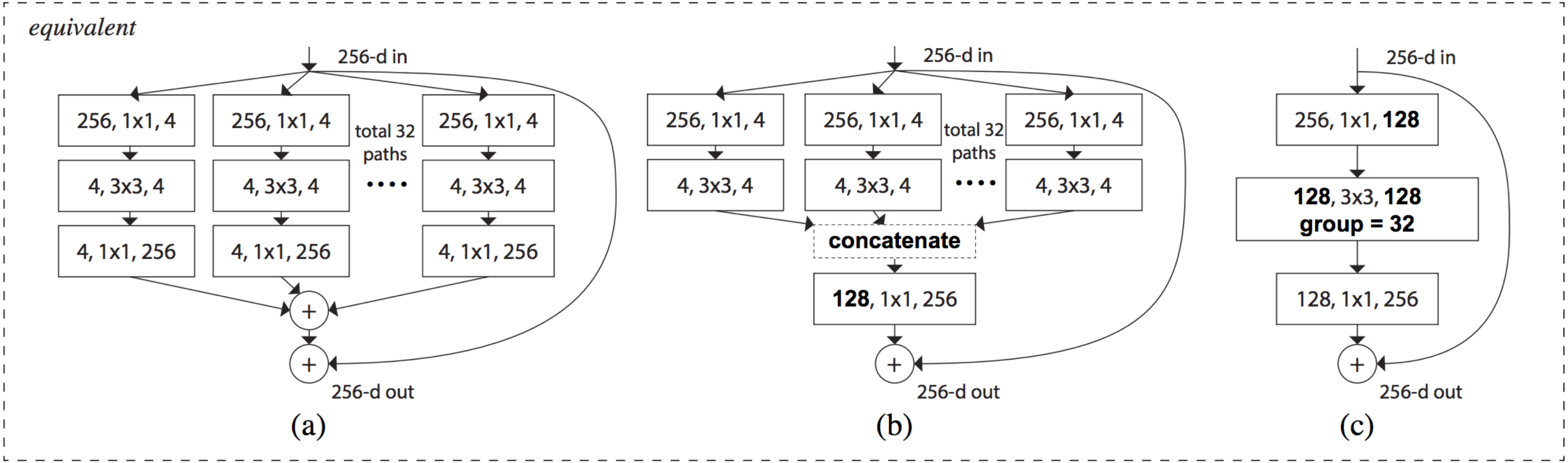


ResNeXt:
uniform multi-branch

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. “Aggregated Residual Transformations for Deep Neural Networks”. arXiv 2016 (CVPR 2017).

ResNeXt

- Concatenation and Addition are interchangeable
  - General property for DNNs; not only limited to ResNeXt
- Uniform multi-branching can be done by group-conv

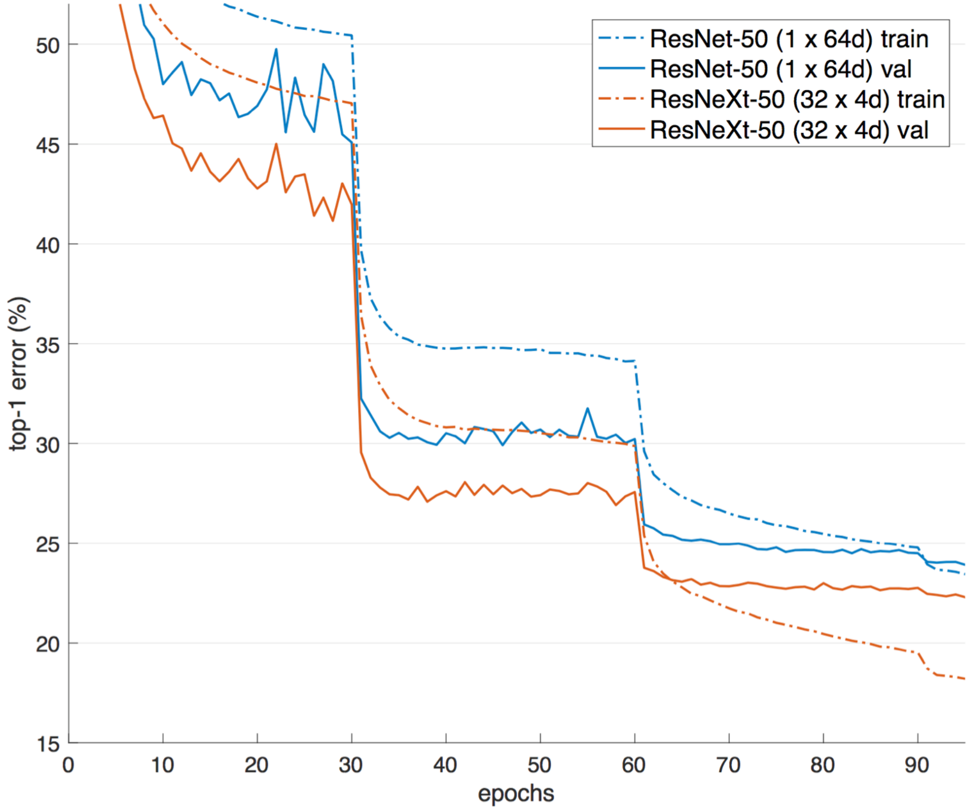


Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated Residual Transformations for Deep Neural Networks". arXiv 2016 (CVPR 2017).

ResNeXt

- Better accuracy
  - when having the same FLOPs/#params as a baseline ResNet

- <u>Better trade-off for high-capacity models</u>



Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. “Aggregated Residual Transformations for Deep Neural Networks”. arXiv 2016 (CVPR 2017).

Competition winners using ResNeXt

ResNeXt is a good trade-off for high-capacity:

- ImageNet Classification 2017, 1st place
  - SE-ResNeXt
- COCO Object Detection 2017, 1st place
  - MegDet + ResNeXt
- COCO Instance Segmentation 2017, 1st place
  - PANet + ResNeXt
- COCO Stuff Segmentation 2017, 1st place
  - FPN + ResNetXt
- …

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. “Aggregated Residual Transformations for Deep Neural Networks”. arXiv 2016 (CVPR 2017).

ResNeXt: higher capacity for billion-scale images

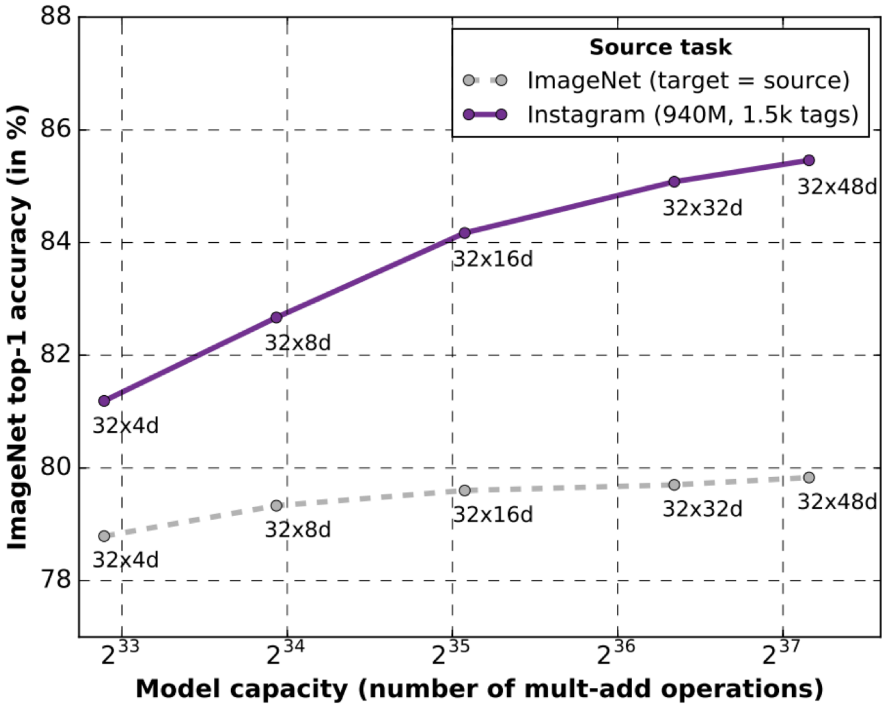
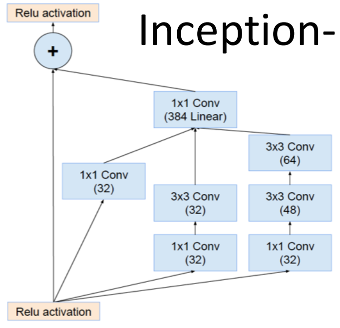


Fig. 5: Classification accuracy on val-IN-1k using ResNeXt-101 32×{4, 8 16, 32, 48}d with and without pretraining on the IG-940M-1.5k dataset.

"Exploring the Limits of Weakly Supervised Pretraining". arXiv 2018.
Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten.

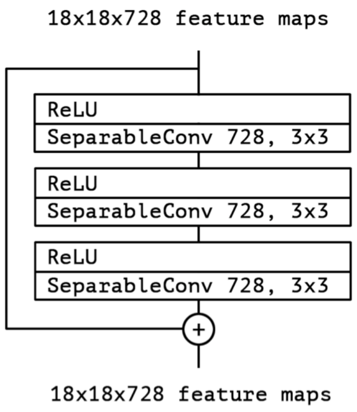
More architectures (not covered in this tutorial)

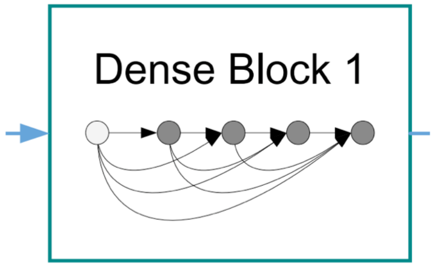
- Inception-ResNet [Szegedy et al 2017]
  - Inception as transformation + residual connection
- DenseNet [Huang et al CVPR 2017]
  - Densely connected shortcuts w/ concat.
- Xception [Chollet CVPR 2017], MobileNets [Howard et al 2017]
  - DepthwiseConv (i.e., GroupConv with #group=#channel)
- ShuffleNet [Zhang et al 2017]
  - More Group/DepthwiseConv + shuffle
- ……



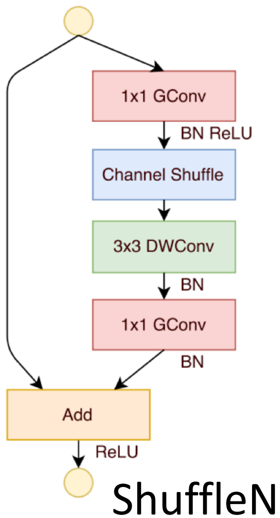
Inception-ResNet

Xception





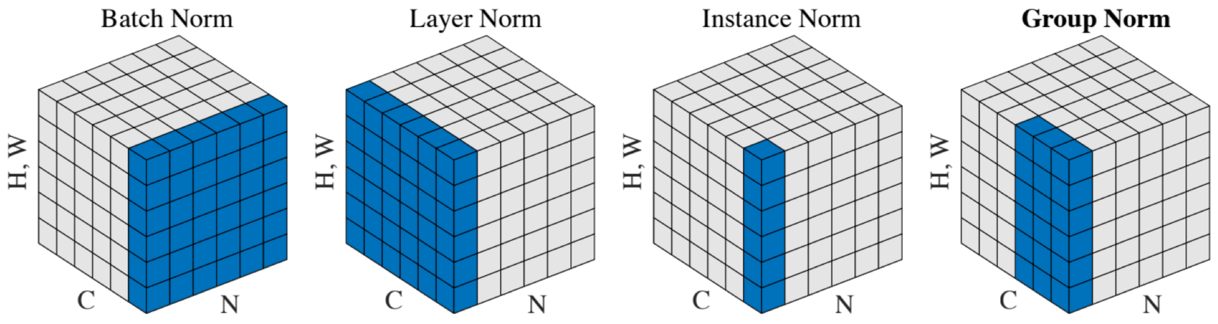
DenseNet

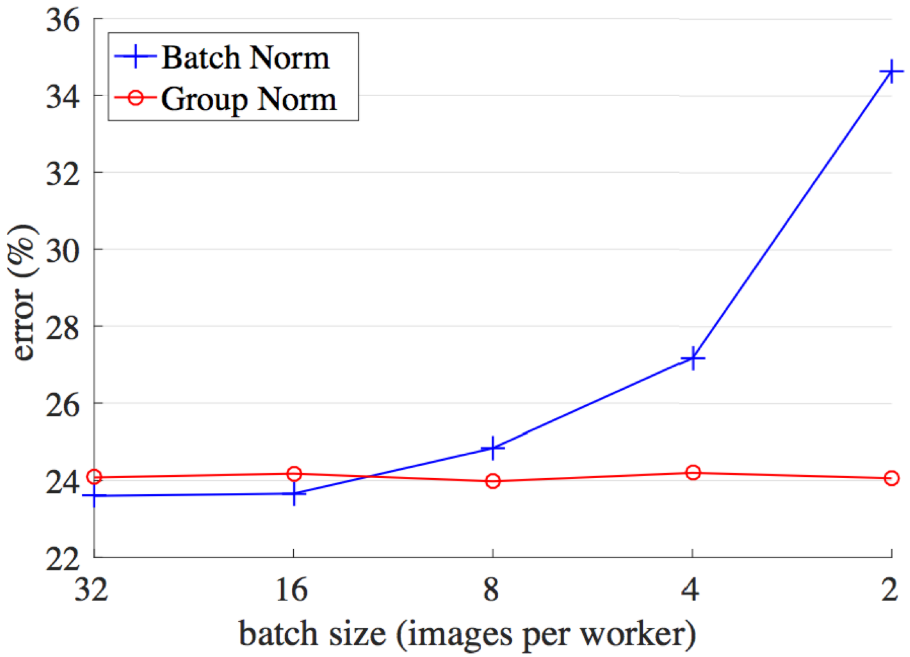


ShuffleNet

Teaser: Group Normalization (GN)

- Independent of batch size
- Robust to small batches
- Enable new scenarios: e.g.: 41 AP on COCO trained from scratch





Yuxin Wu and Kaiming He. "Group Normalization". ECCV 2018

Conclusion

- Deep Learning is Representation Learning
- Represent data for machines to perform tasks (this talk)
- Represent data for machines to perform tasks (next talks)