Lecture 6: Deep CNN Architectures (ResNet)

Before ChatGPT was published :-)

Why should we study ResNet and Self-Supervised Learning?

Slides Credits: from Dr. Kaiming He

Overview

- Introduction
- Background
  - From shallow to deep
- Deep Residual Networks
  - From 10 layers to 100 layers
  - From 100 layers to 1000 layers
- Applications

Introduction

Introduction

Deep Residual Networks (ResNets)

- "Deep Residual Learning for Image Recognition". CVPR 2016
- A simple and clean framework of training "very" deep nets
- State-of-the-art performance for
  - Image classification
  - Object detection
  - Semantic segmentation
  - and more…

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

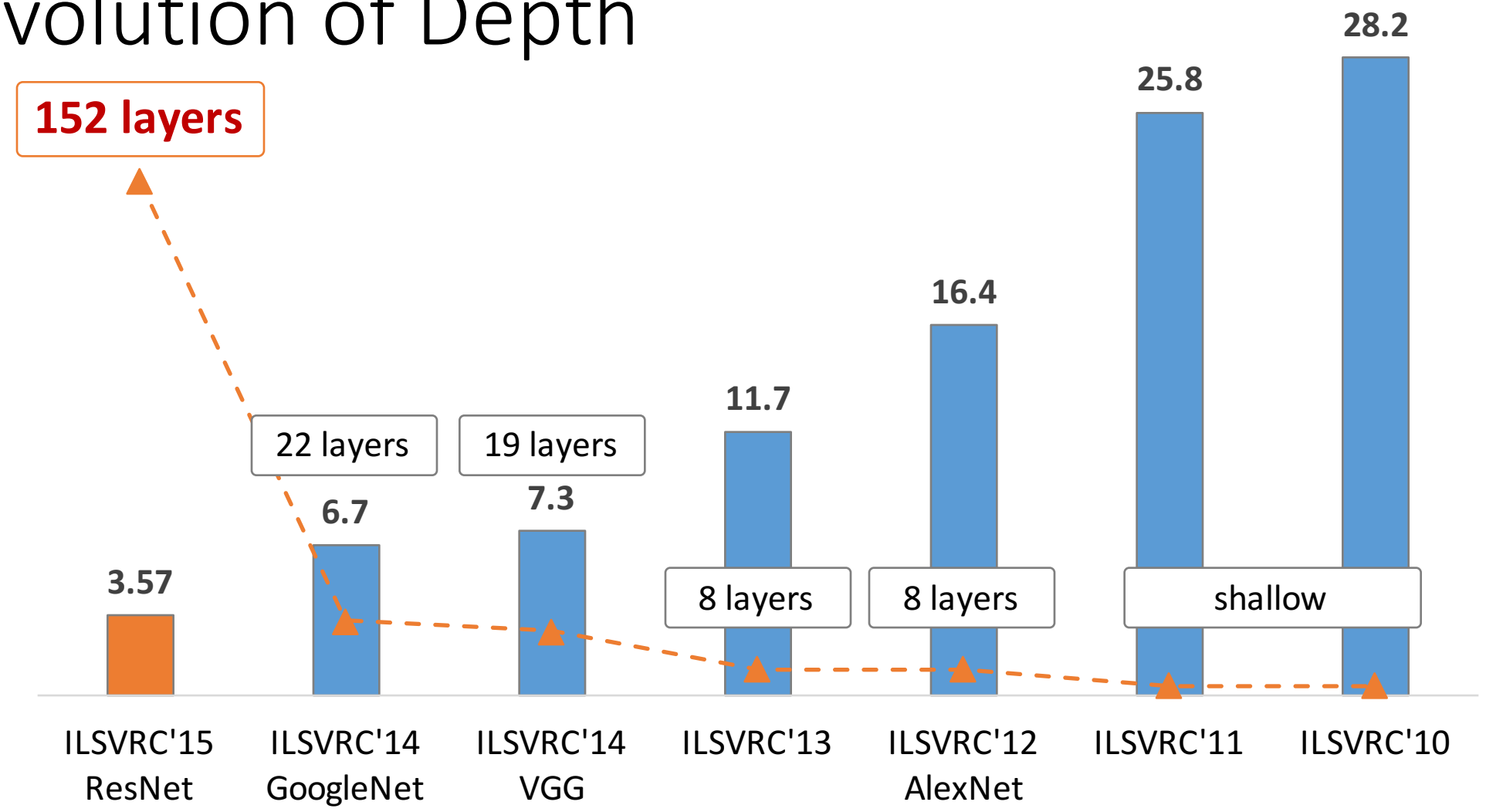
ResNets @ ILSVRC & COCO 2015 Competitions

- **1st places in all five main tracks**
  - ImageNet Classification: "*Ultra-deep*" 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

*improvements are relative numbers

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

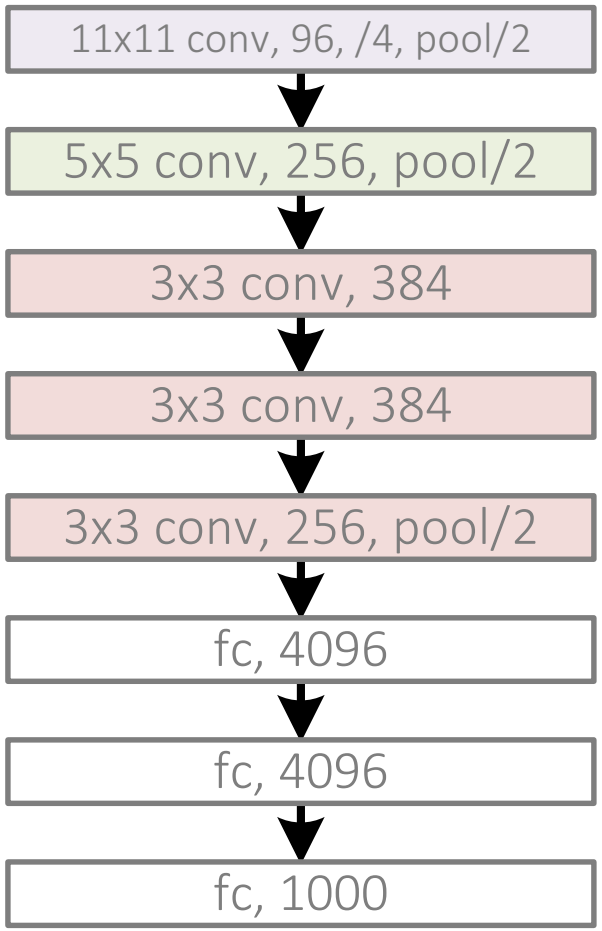
Revolution of Depth



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

Revolution of Depth

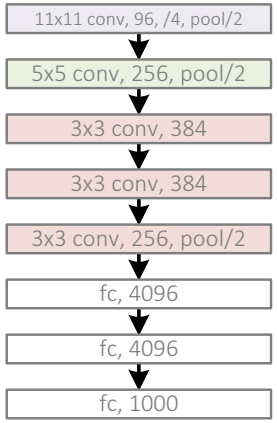
AlexNet, 8 layers
(ILSVRC 2012)



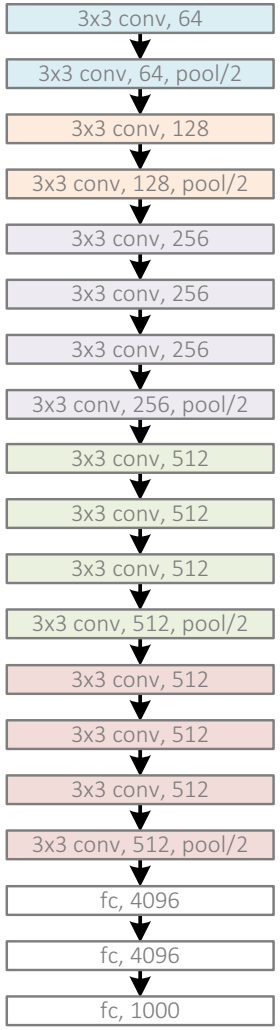
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

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

Revolution of Depth

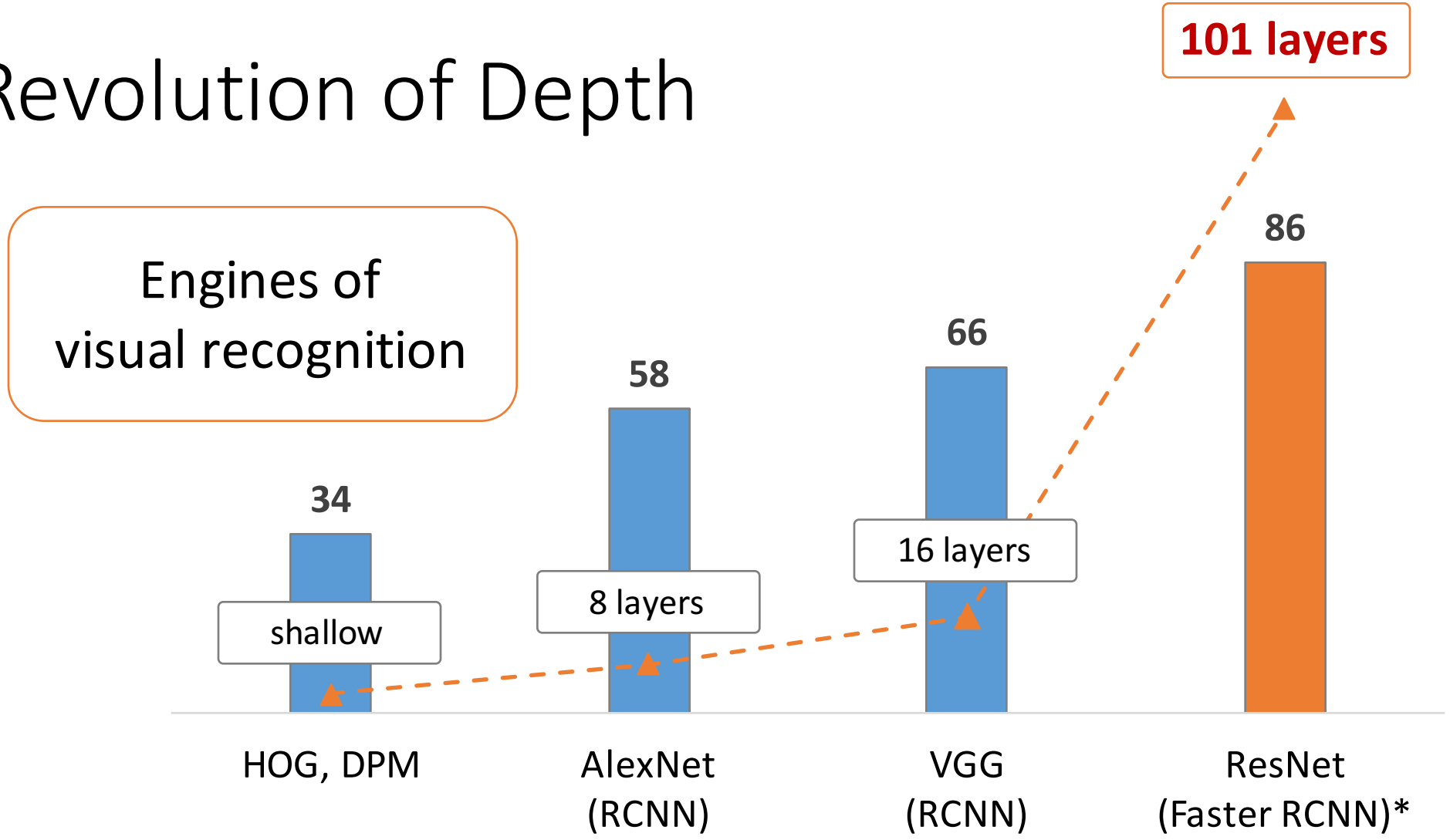
AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

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

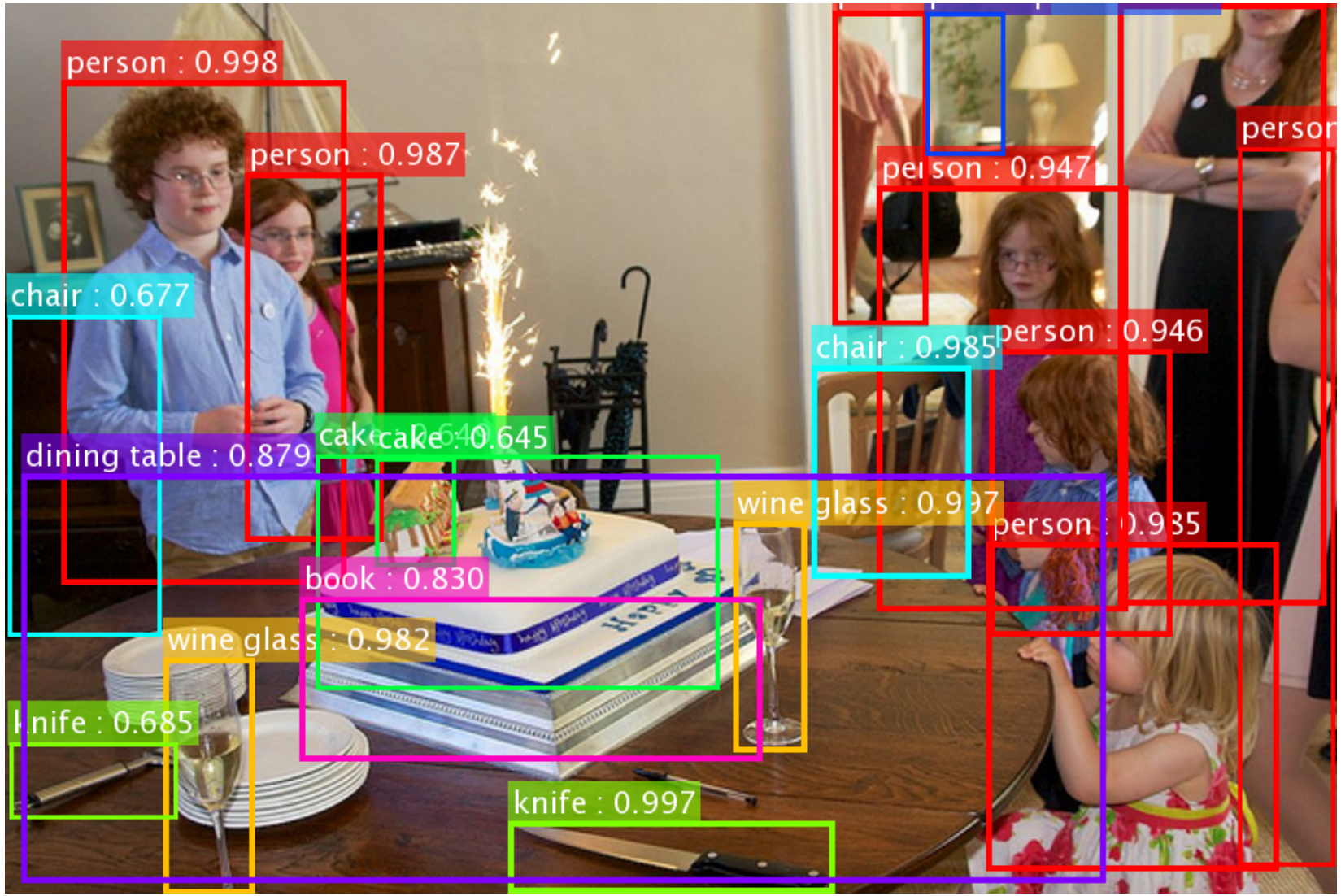
Revolution of Depth



PASCAL VOC 2007 **Object Detection** mAP (%)

*w/ other improvements & more data

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



ResNet's object detection result on COCO

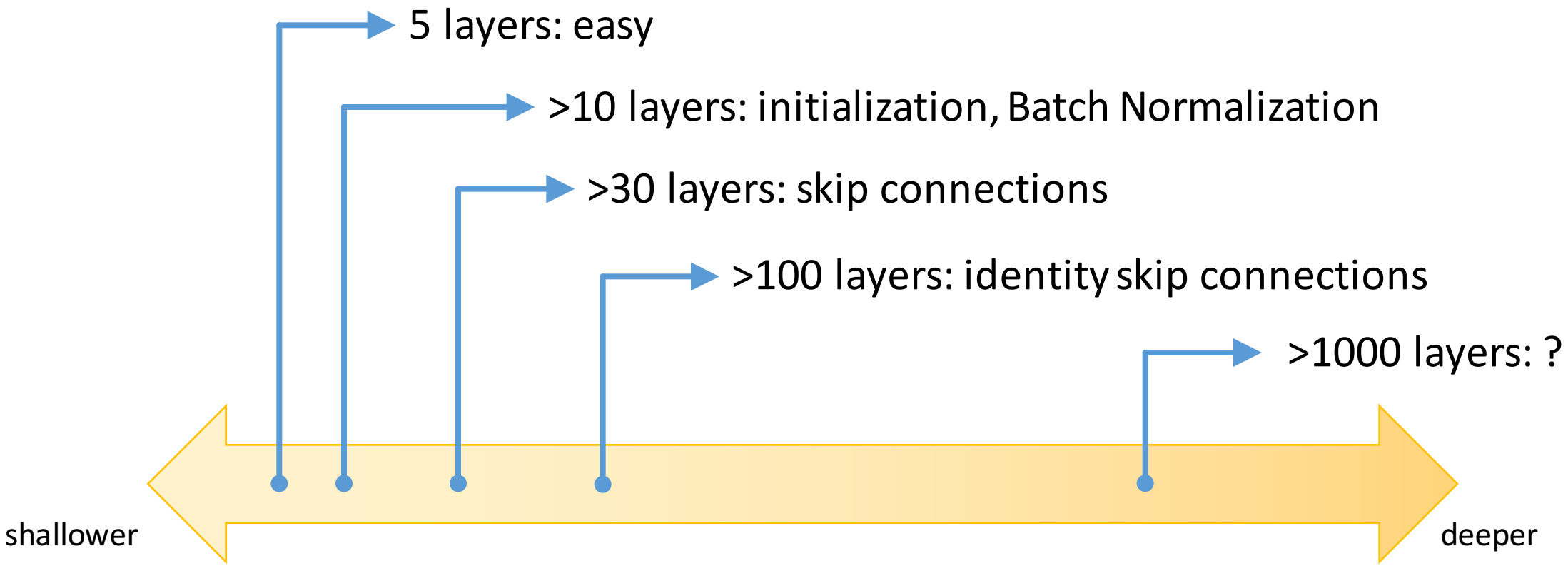
*the original image is from the COCO dataset

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

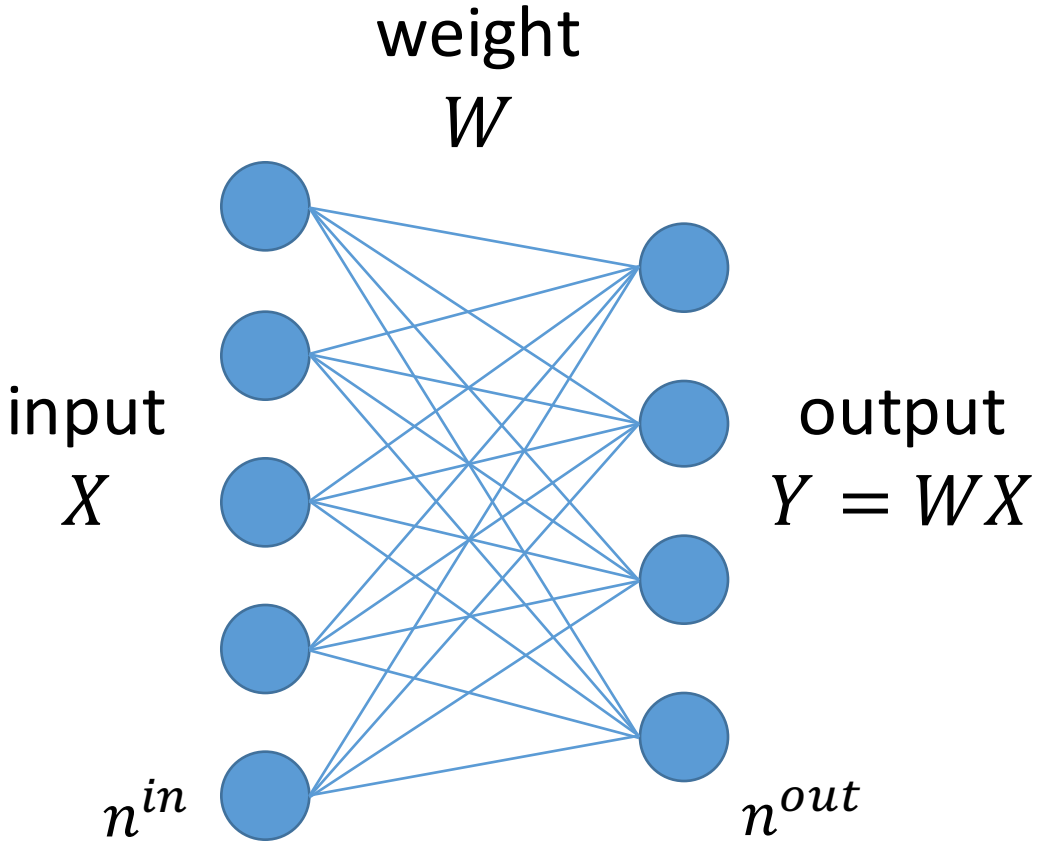
Background

From shallow to deep

Spectrum of Depth



Initialization



If:

- Linear activation
- x, y, w: independent

Then:

1-layer:

$$Var[y] = (n^{in} Var[w]) Var[x]$$

Multi-layer:

$$Var[y] = (\prod_d n_d^{in} Var[w_d]) Var[x]$$

LeCun et al 1998 “Efficient Backprop”

Glorot & Bengio 2010 “Understanding the difficulty of training deep feedforward neural networks”

Initialization

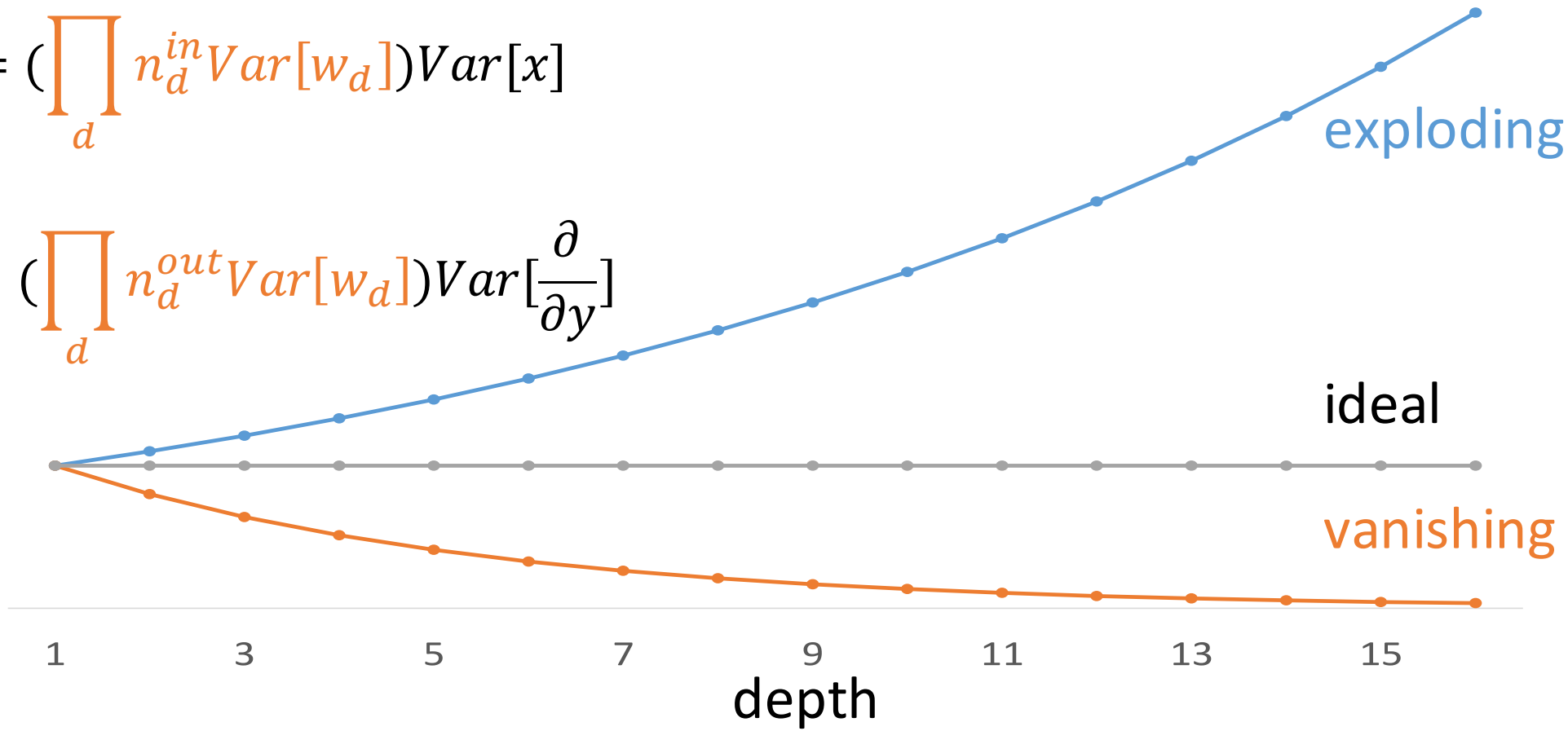
Both forward (response) and backward (gradient) signal can vanish/explode

Forward:

$$Var[y] = (\prod_d n_d^{in} Var[w_d])Var[x]$$

Backward:

$$Var\left[\frac{\partial}{\partial x}\right] = (\prod_d n_d^{out} Var[w_d])Var[\frac{\partial}{\partial y}]$$



LeCun et al 1998 “Efficient Backprop”

Glorot & Bengio 2010 “Understanding the difficulty of training deep feedforward neural networks”

Initialization

- Initialization under **linear** assumption

$$\prod_d n_d^{in} Var[w_d] = const_{\text{fw}} \text{ (healthy forward)}$$

and

How can we make it become a constant?

$$\prod_d n_d^{out} Var[w_d] = const_{\text{bw}} \text{(healthy backward)}$$

➡ $n_d^{in} Var[w_d] = 1$ or* $n_d^{out} Var[w_d] = 1$

We make each part as 1

*: $n_d^{out} = n_{d+1}^{in}$, so $\frac{const_{\text{bw}}}{const_{\text{fw}}} = \frac{n_{\text{last}}^{out}}{n_{\text{first}}^{in}} < \infty$.

It is sufficient to use either form.

"*Xavier*" init in Caffe

LeCun et al 1998 "Efficient Backprop"

Glorot & Bengio 2010 "Understanding the difficulty of training deep feedforward neural networks"

Initialization

- Initialization under **ReLU** activation

$$\prod_d \frac{1}{2} n_d^{in} Var[w_d] = const_{\text{fw}} \text{ (healthy forward)}$$

and

$$\prod_d \frac{1}{2} n_d^{out} Var[w_d] = const_{\text{bw}} \text{(healthy backward)}$$

half of the ReLU elements are zero (-inf, 0)

➡ $\frac{1}{2} n_d^{in} Var[w_d] = 1$ or $\frac{1}{2} n_d^{out} Var[w_d] = 1$

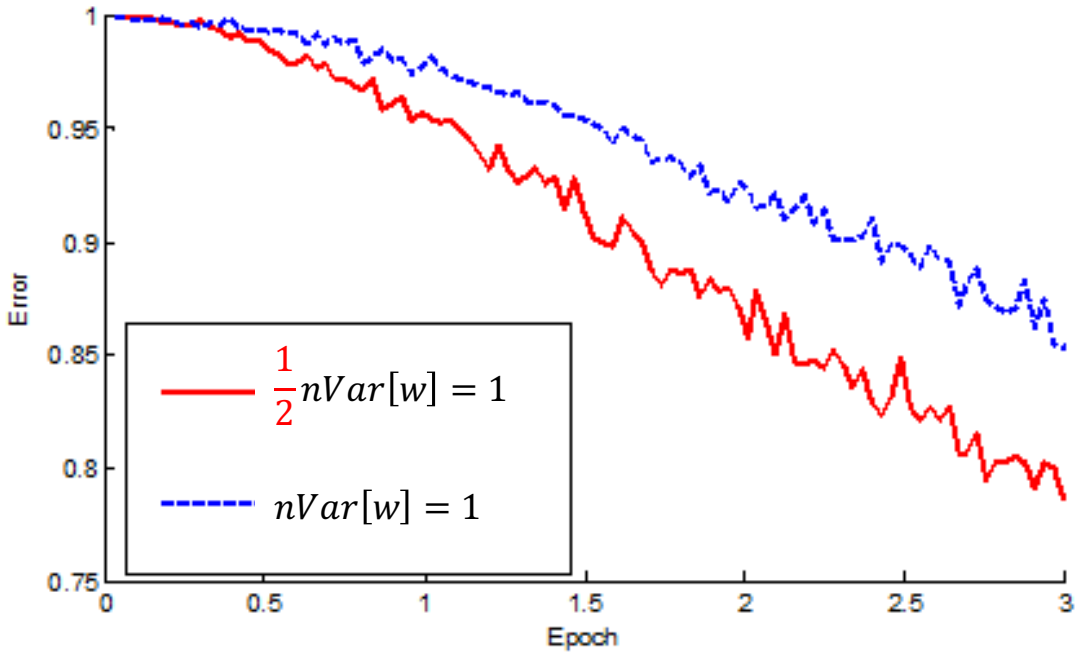
With D layers, a factor of 2 per layer has exponential impact of 2^D

"*MSRA*" init in Caffe

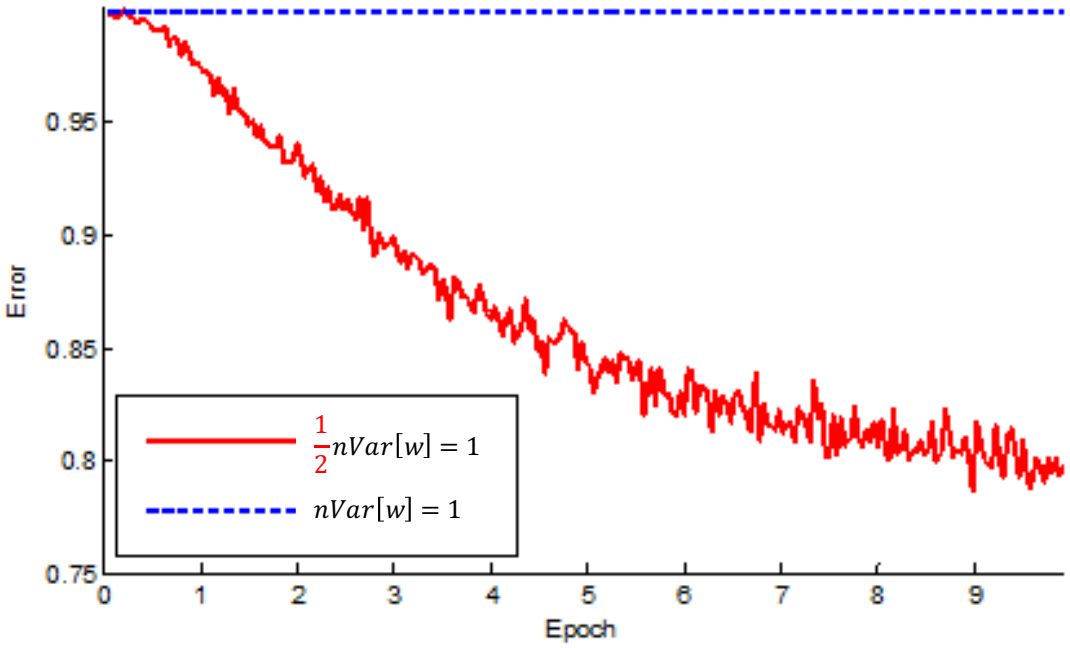
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

Initialization

22-layer ReLU net:
good init converges faster



30-layer ReLU net:
good init is able to converge



*Figures show the beginning of training

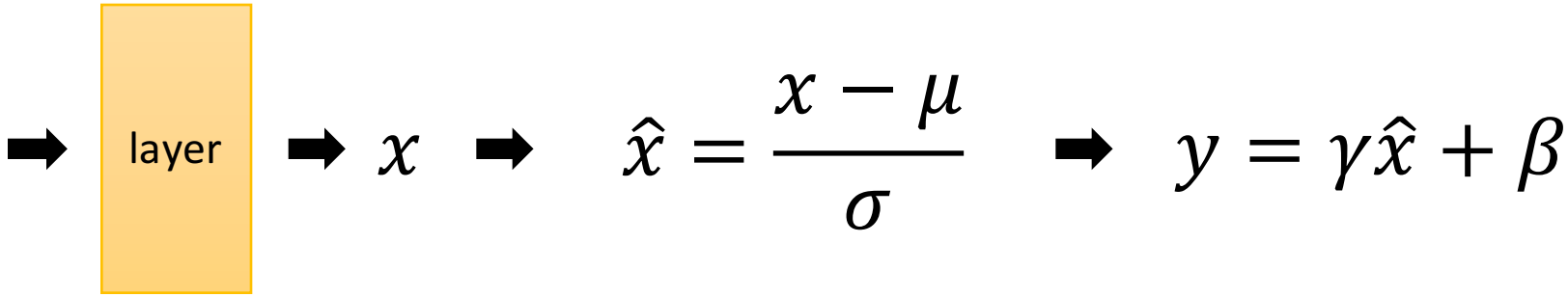
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

Batch Normalization (BN)

- Normalizing input (LeCun et al 1998 “Efficient Backprop”)
- BN: normalizing each layer, for each mini-batch
- Greatly accelerate training
- Less sensitive to initialization
- Improve regularization

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Batch Normalization (BN)

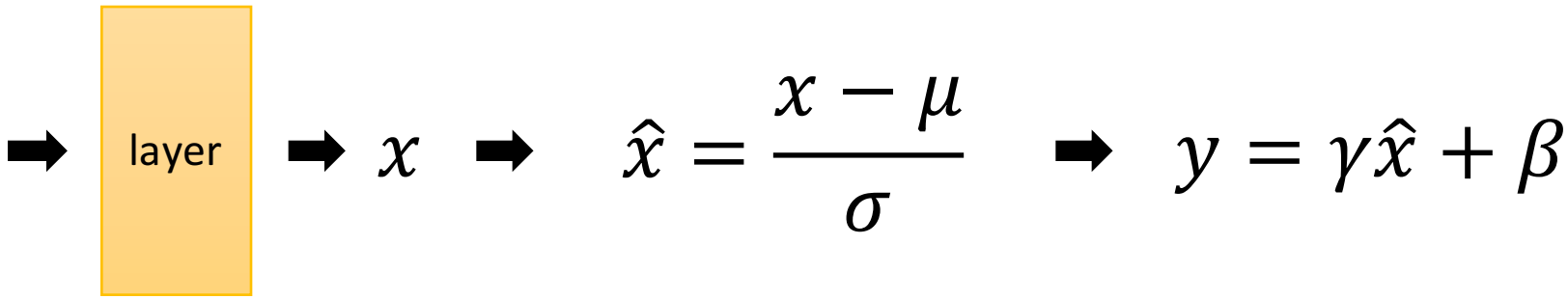


- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

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

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Batch Normalization (BN)



2 modes of BN:

- Train mode:
  - μ, σ are functions of x; backprop gradients
- Test mode:
  - μ, σ are pre-computed* on training set

Caution: make sure your BN is in a correct mode

running_mean = momentum * running_mean + (1 - momentum) * sample_mean
running_var = momentum * running_var + (1 - momentum) * sample_var

*: by running average, or post-processing after training

https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Batch Normalization (BN)

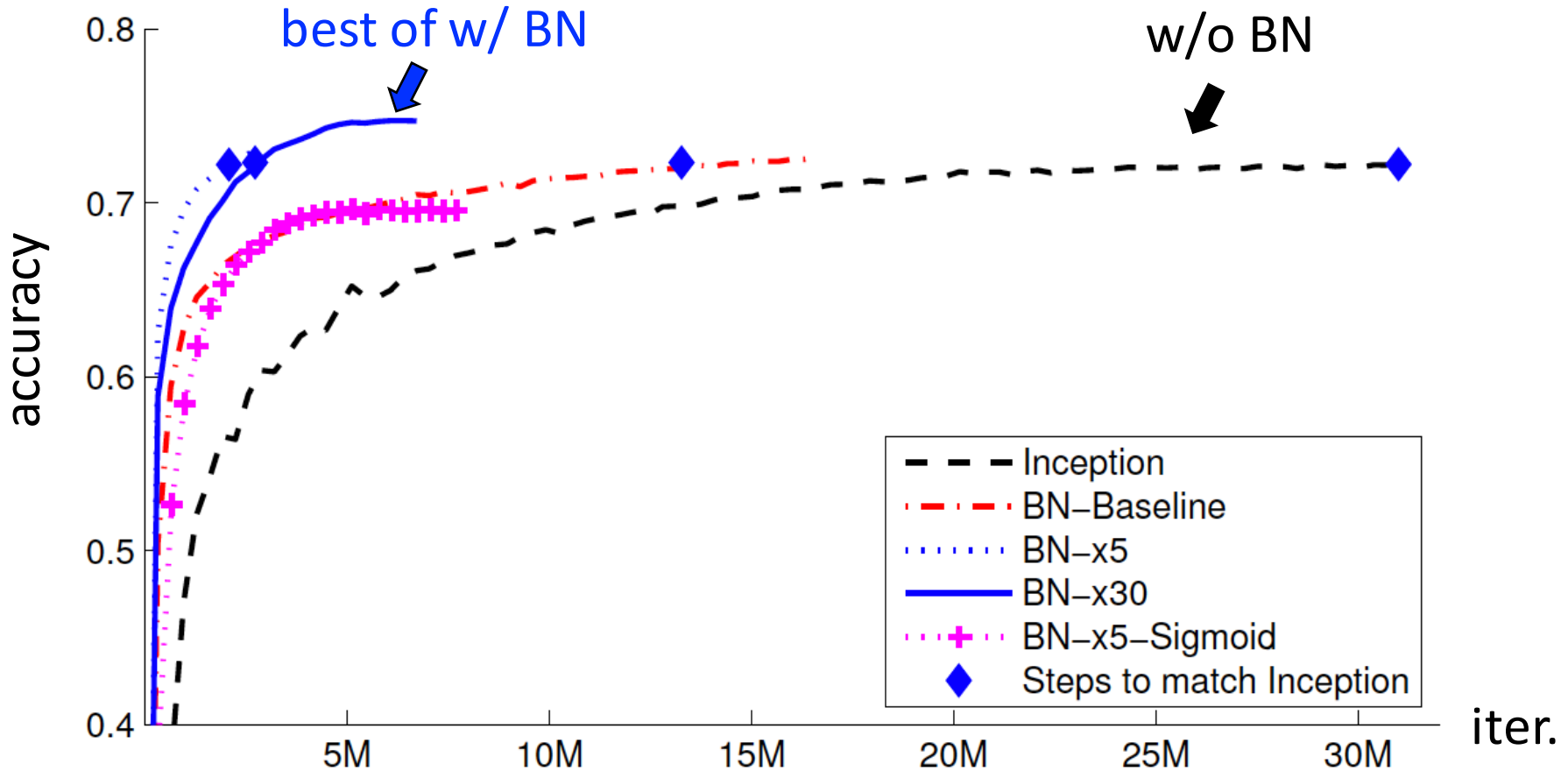


Figure taken from [S. Ioffe & C. Szegedy]

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Deep Residual Networks

From 10 layers to 100 layers

Going Deeper

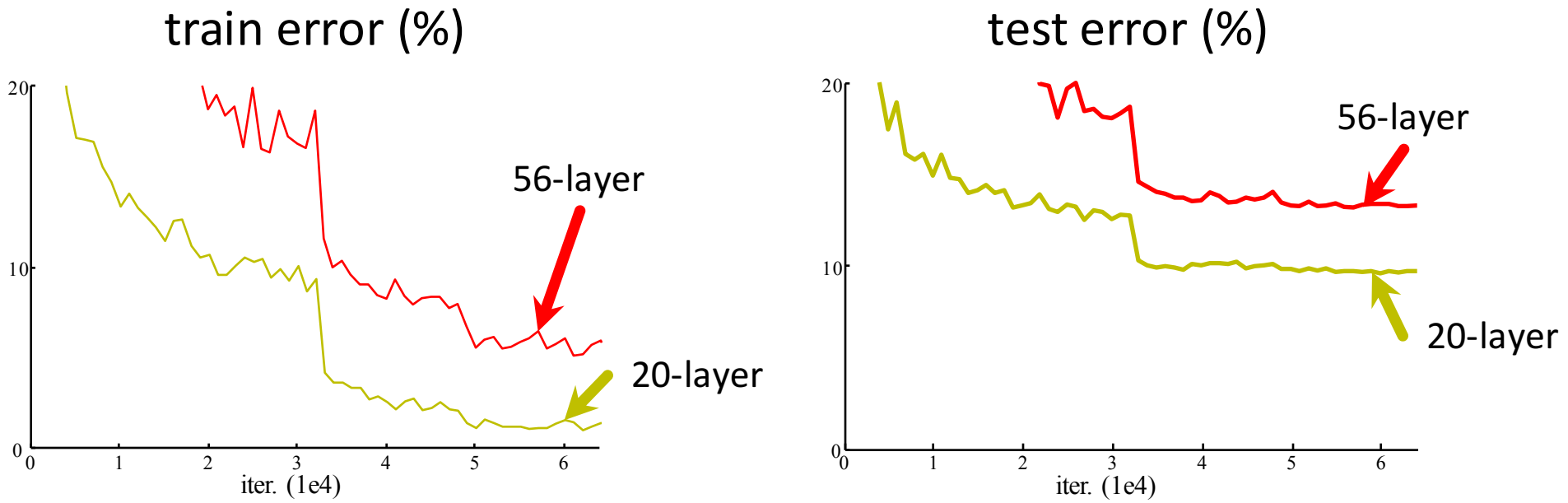
- Initialization algorithms ✓
- Batch Normalization ✓

- **Is learning better networks as simple as stacking more layers?**

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

Simply stacking layers?

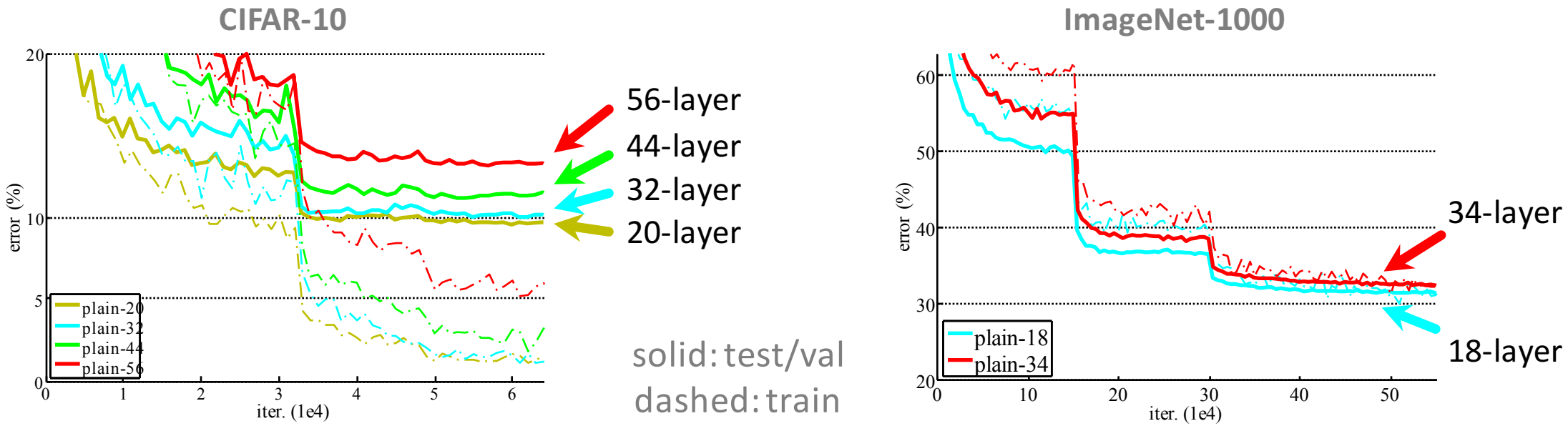
CIFAR-10



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

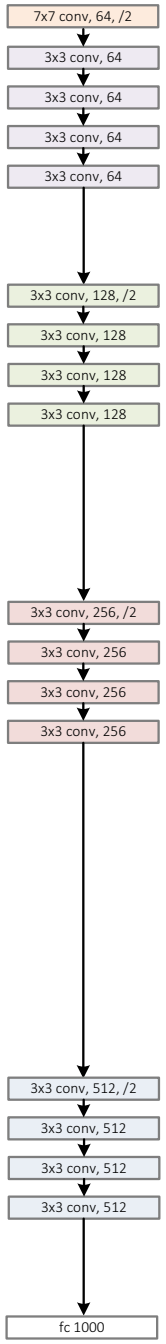
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

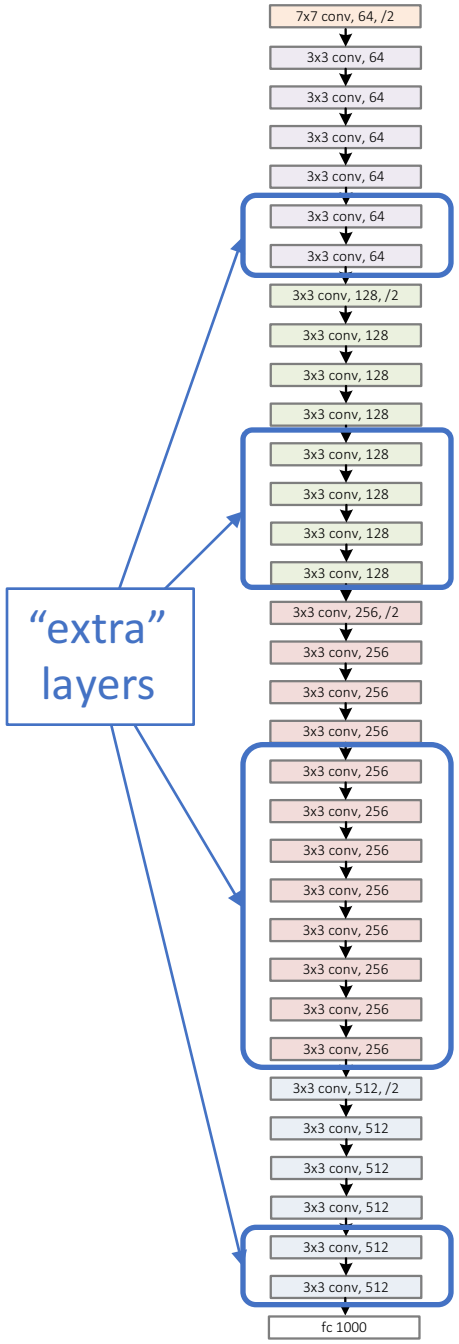
Simply stacking layers?



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

a shallower model (18 layers)





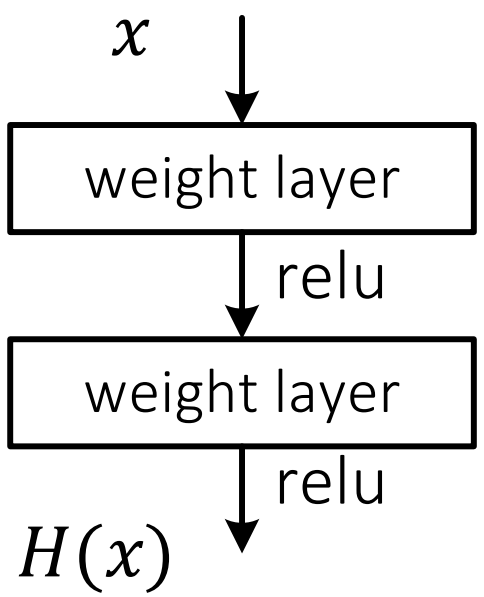
a deeper counterpart (34 layers)

- 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

- Plaint net



any two
stacked layers

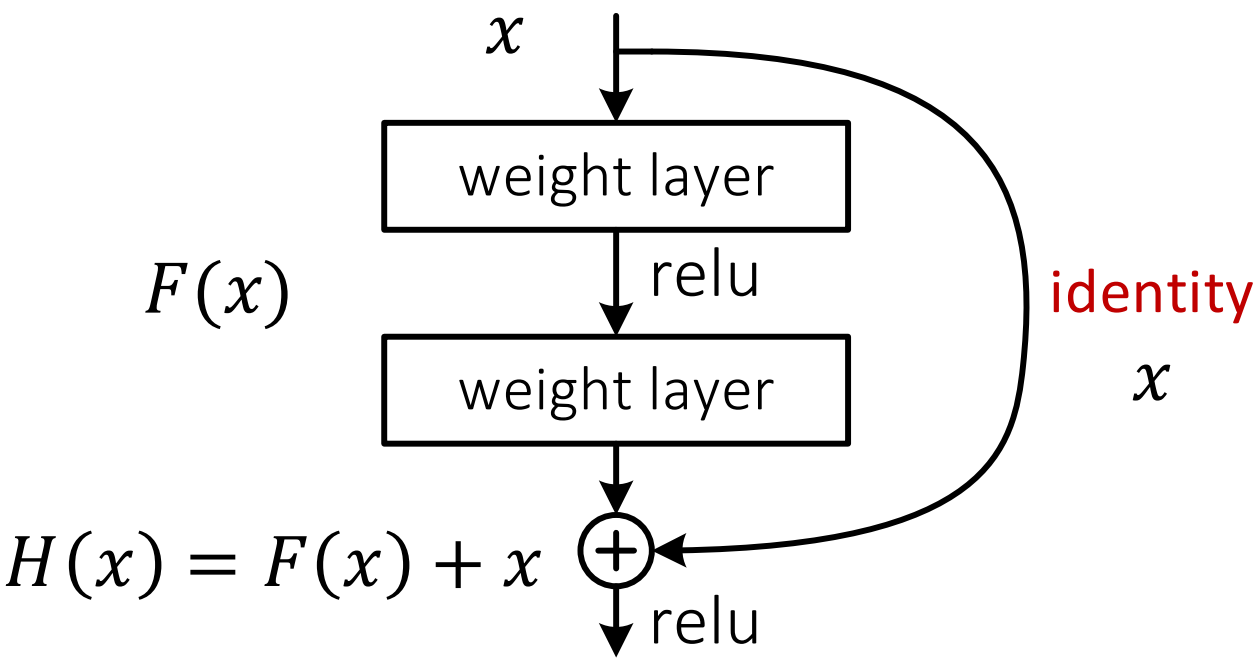
$H(x)$ is any desired mapping,

hope the 2 weight layers 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 2 weight layers fit $H(x)$~~

hope the 2 weight layers fit $F(x)$

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

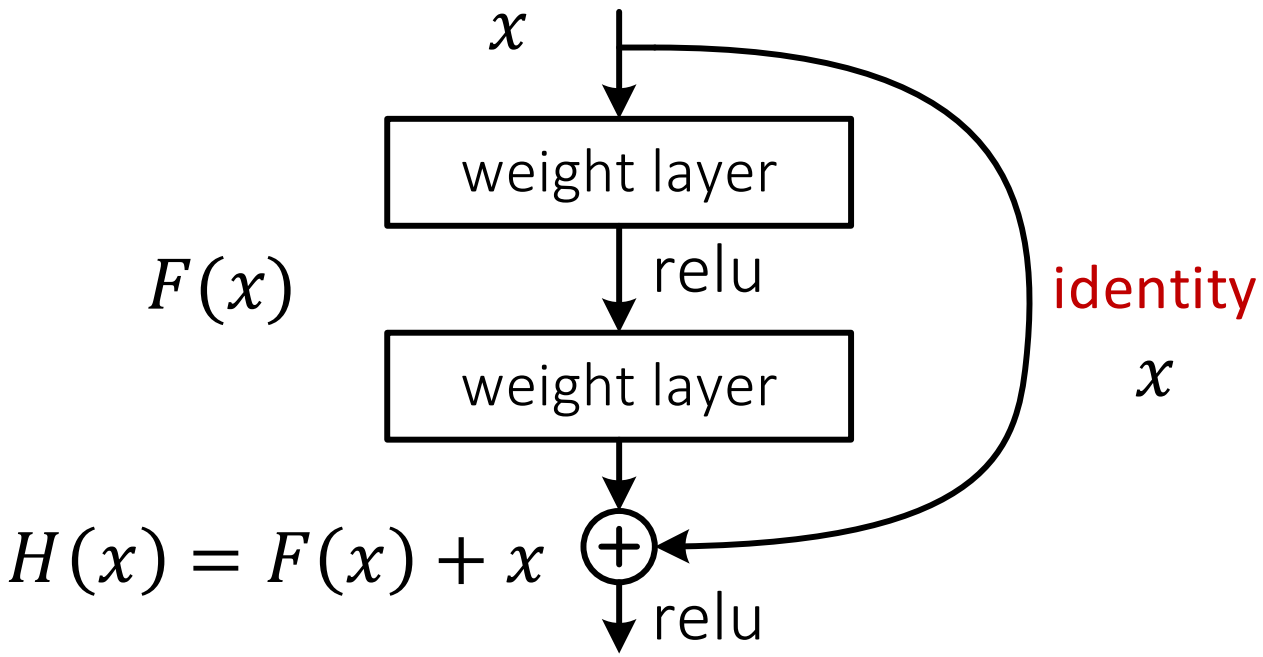
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Residual Learning

Why should we do this?

Make it easier for optimization

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



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

  Optimizer does not need these 2 layers (i.e. F(x))

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

H(x) is close to x

small fluctuations mean weights (i.e. F(x)) are small

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

Related Works – Residual Representations

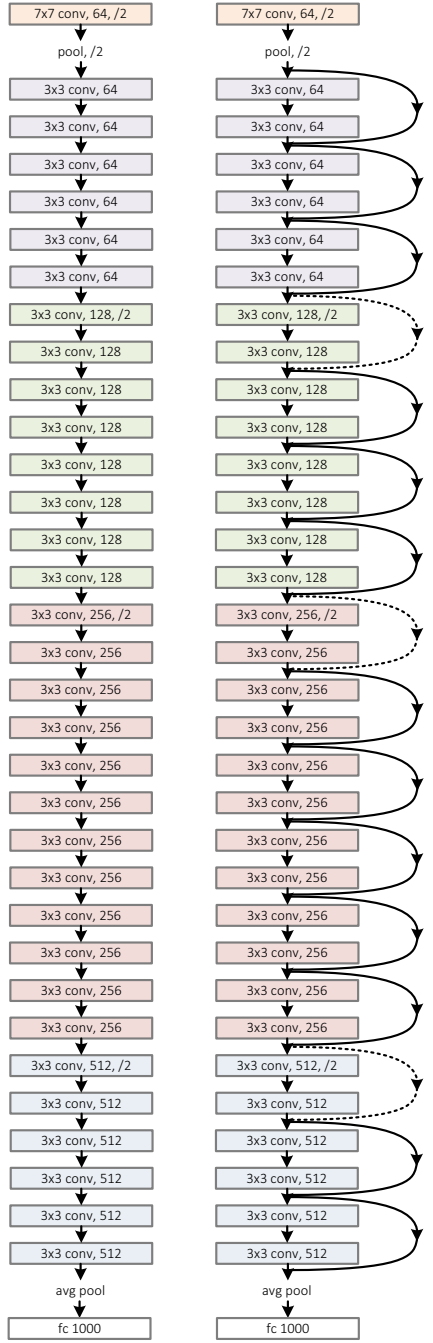
- VLAD & Fisher Vector [Jegou et al 2010], [Perronnin et al 2007]
  - Encoding residual vectors; powerful shallower representations.
- Product Quantization (IVF-ADC) [Jegou et al 2011]
  - Quantizing residual vectors; efficient nearest-neighbor search.
- MultiGrid & Hierarchical Precondition [Briggs, et al 2000], [Szeliski 1990, 2006]
  - Solving residual sub-problems; efficient PDE solvers.

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

Network “Design”

plain net

ResNet



What does “/2” mean here?

- Keep it simple
- Our basic design (VGG-style)
  - all 3x3 conv (almost)
  - spatial size /2 => # filters x2 (~same complexity per layer)
  - Simple design; just deep!

GPT: simple design; just large (larger models & dataset)

- Other remarks:
  - no hidden fc
  - no dropout

The dotted line is there, precisely because there has been a change in the dimension of the input volume (of course a reduction because of the convolution).

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

In the structure diagram of ResNet (Residual Network)
"3x3 conv, 128, /2": a configuration of a convolutional layer
"3x3 conv": a convolution kernel of size 3x3
"128": the number of output channels is 128
"/2": the stride is 2

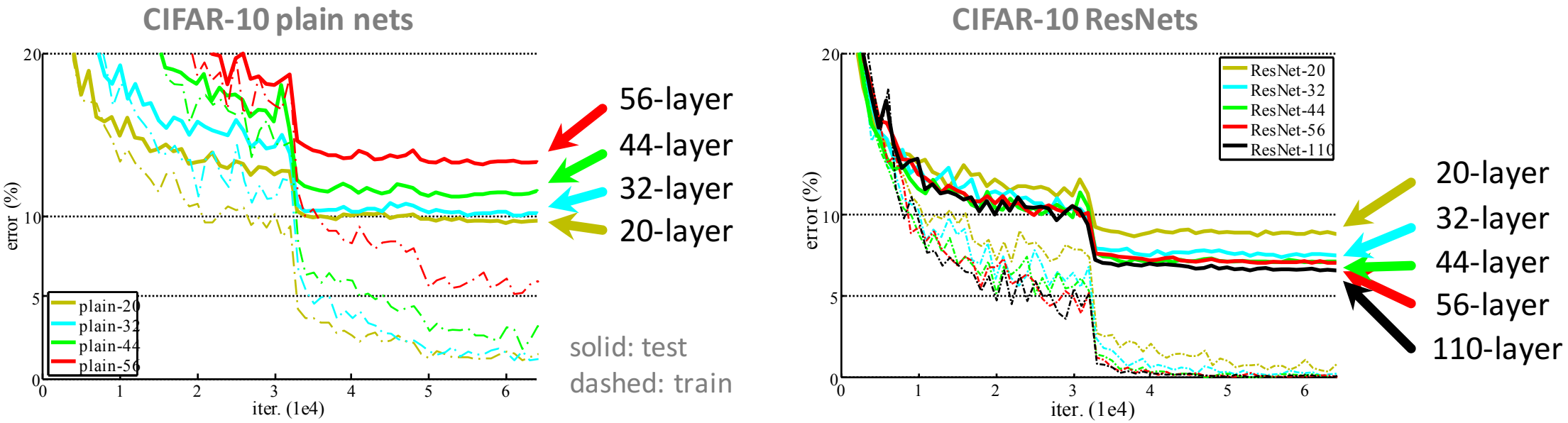
Training

- All plain/residual nets are trained from scratch
- All plain/residual nets use Batch Normalization
- Standard hyper-parameters & augmentation

Make sure both of them are well tuned

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

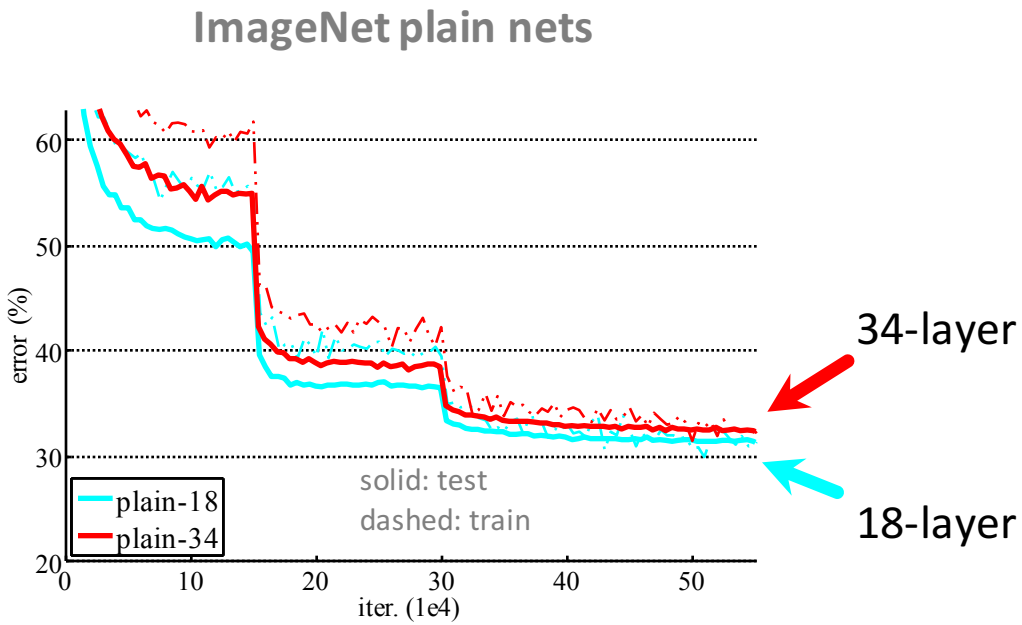
CIFAR-10 experiments

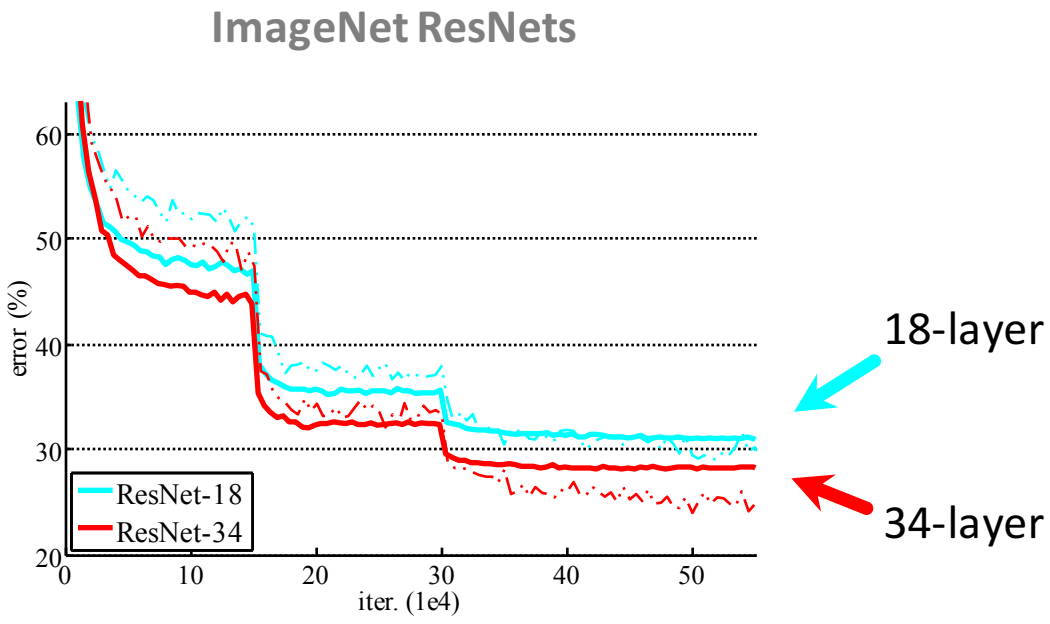


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

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

ImageNet experiments



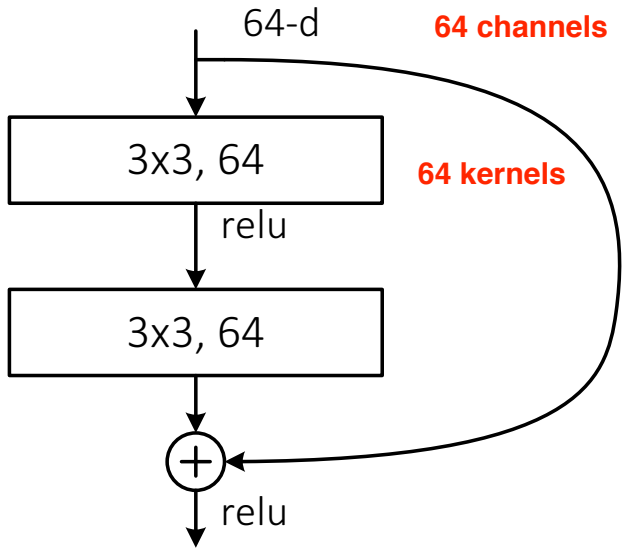


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

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

ImageNet experiments

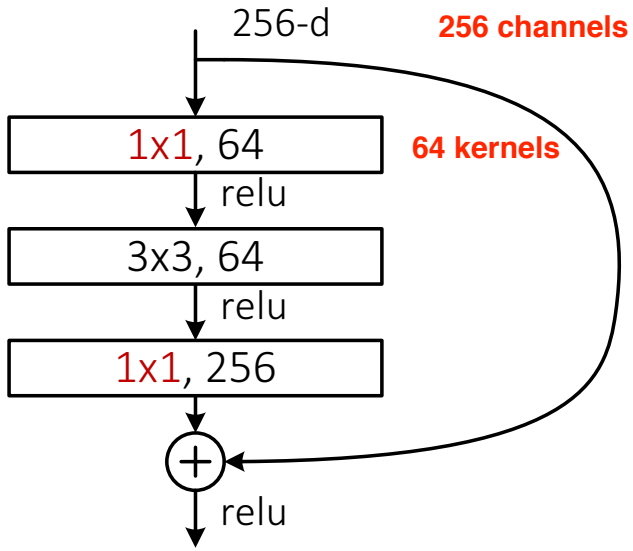
- A practical design of going deeper



Quiz: How many parameters in this block?

all-3x3

similar complexity



Quiz: How many parameters in this block?

bottleneck

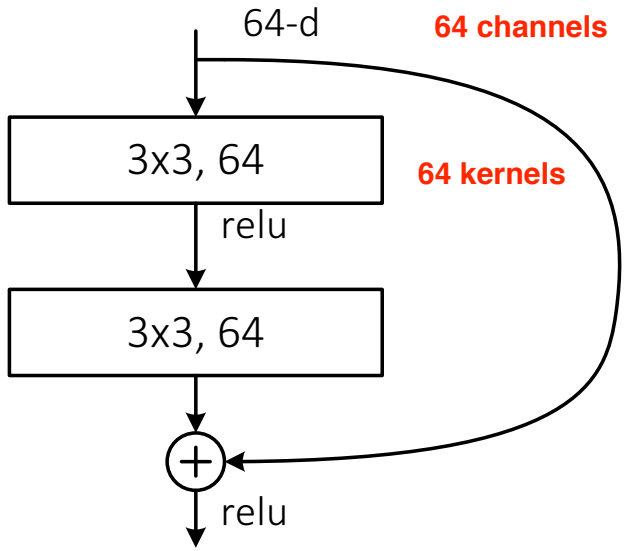
(for ResNet-50/101/152)

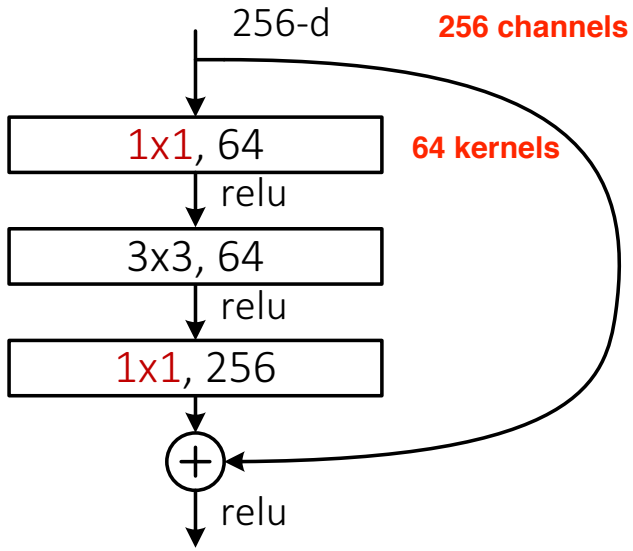
Why do we want to do this?

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

ImageNet experiments

- A practical design of going deeper





3x3x64x64x2 = 73728 parameters

all-3x3

similar complexity

1x1x256x64 + 3x3x64x64 + 1x1x64x256 = 69632 parameters

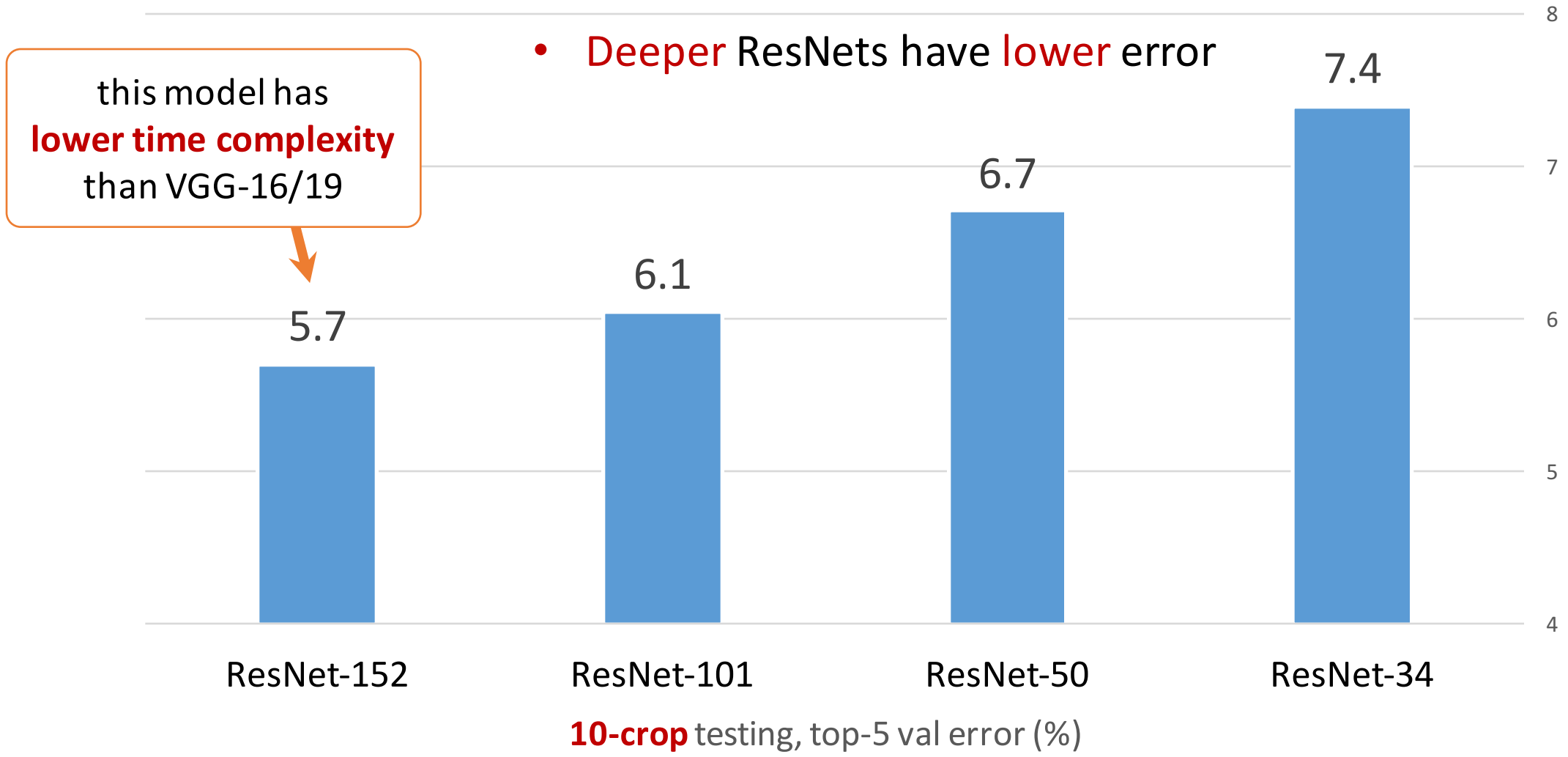
bottleneck

(for ResNet-50/101/152)

higher dimension with fewer parameters

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

ImageNet experiments



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

Why ResNet-152 (more layers) has lower time complexity than VGG-19?

Why ResNet-152 (more layers) has lower time complexity than VGG-19?

Parameter Count:

VGG-19: ~143 million parameters.
ResNet-152: ~60 million parameters.

FLOPs (Floating-Point Operations, a measure of computational complexity):

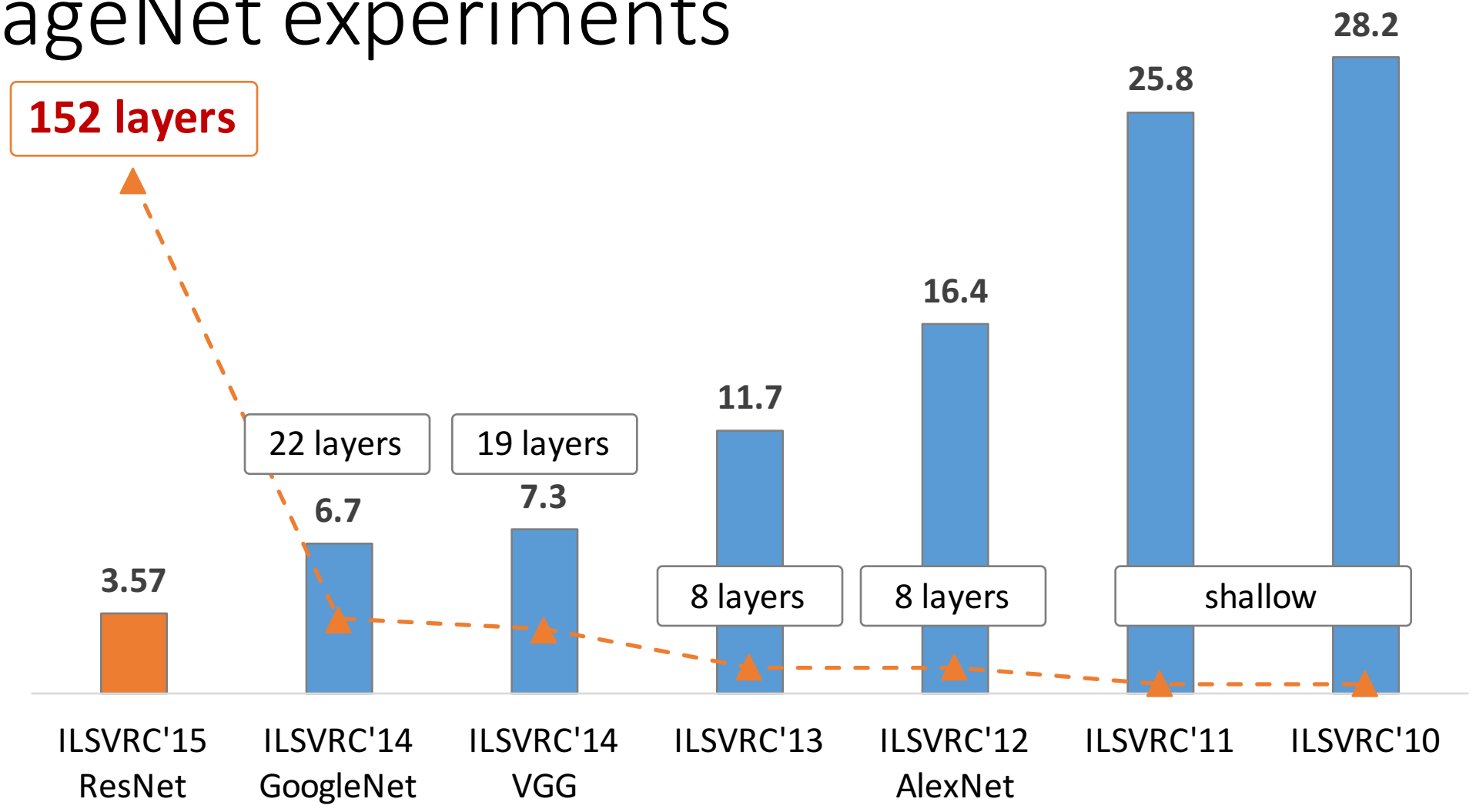
VGG-19: ~19.6 billion FLOPs.
ResNet-152: ~11.3 billion FLOPs.

Why ResNet-152 (more layers) has fewer parameters than VGG-19?

Global Average Pooling: replacing the fully connected layers, which are parameter-heavy, with global average pooling, significantly reducing the total parameter count.

Shortcut Connections: reusing features without additional parameters, further reducing the network's parameter count while maintaining or enhancing performance.

ImageNet experiments



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

Discussions

Representation, Optimization, Generalization

Issues on learning deep models

- **Representation** ability

  How good is the training?

  - Ability of model to fit training data, if optimum could be found
  - If model A's solution space is a superset of B's, A should be better.

- **Optimization** ability

  How easy is the training?

  - Feasibility of finding an optimum
  - Not all models are equally easy to optimize

- **Generalization** ability

  How good is the testing?

  - Once training data is fit, how good is the test performance

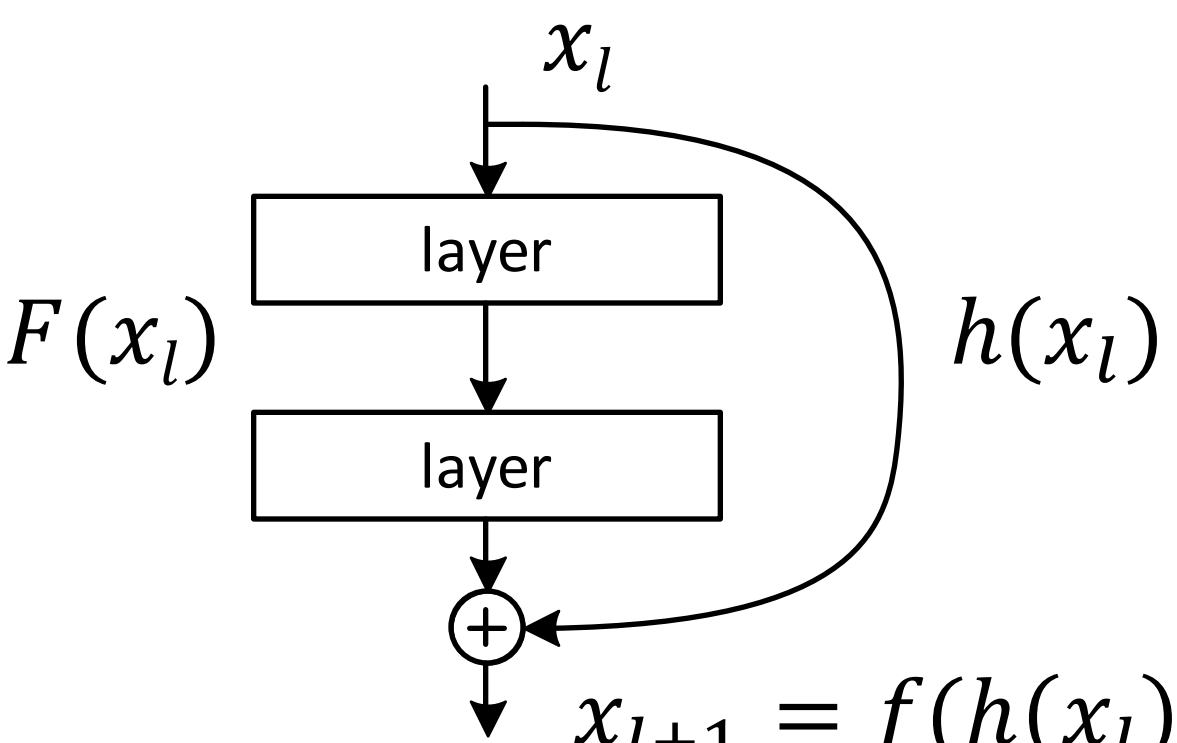
How do ResNets address these issues?

- **Representation** ability
  - No explicit advantage on representation (only re-parameterization), but
  - Allow models to go deeper
- **Optimization** ability
  - Enable very smooth forward/backward prop
  - Greatly ease optimizing deeper models
- **Generalization** ability
  - Not explicitly address generalization, but
  - Deeper+thinner is good generalization

On the Importance of Identity Mapping

From 100 layers to 1000 layers

On identity mappings for **optimization**

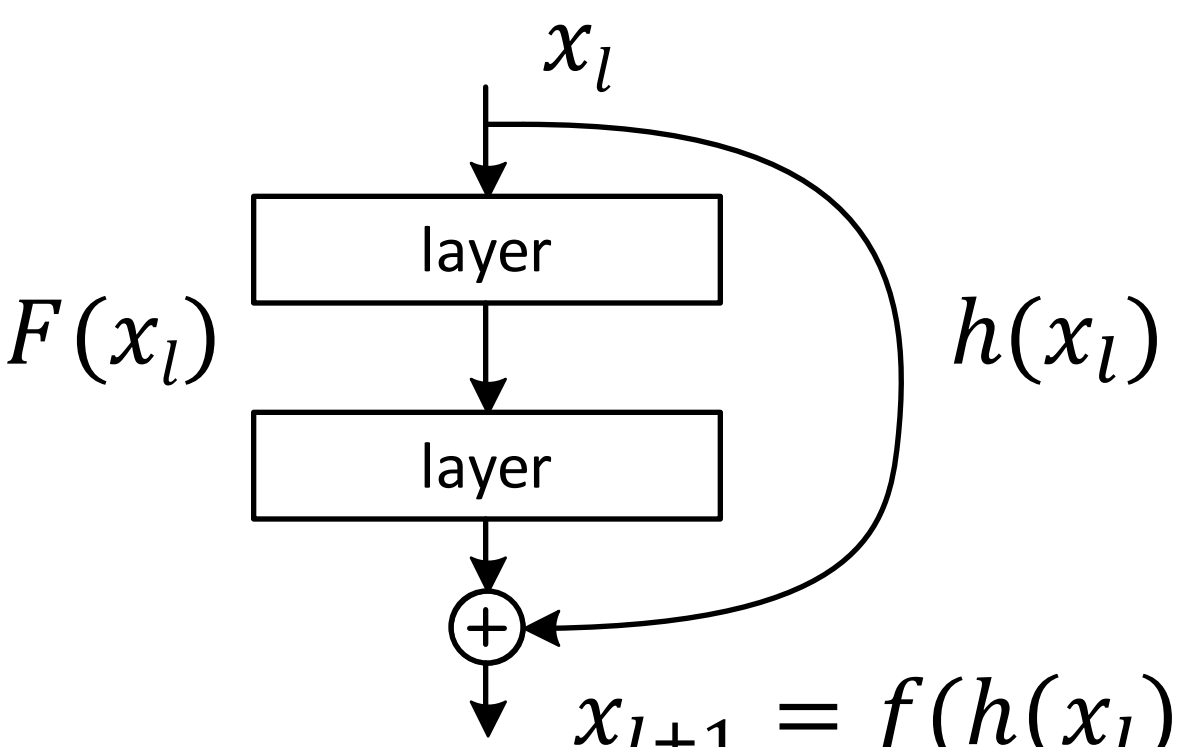


- shortcut mapping: h = identity
- after-add mapping: f = ReLU

$$x_{l+1} = f(h(x_l) + F(x_l))$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

On identity mappings for **optimization**

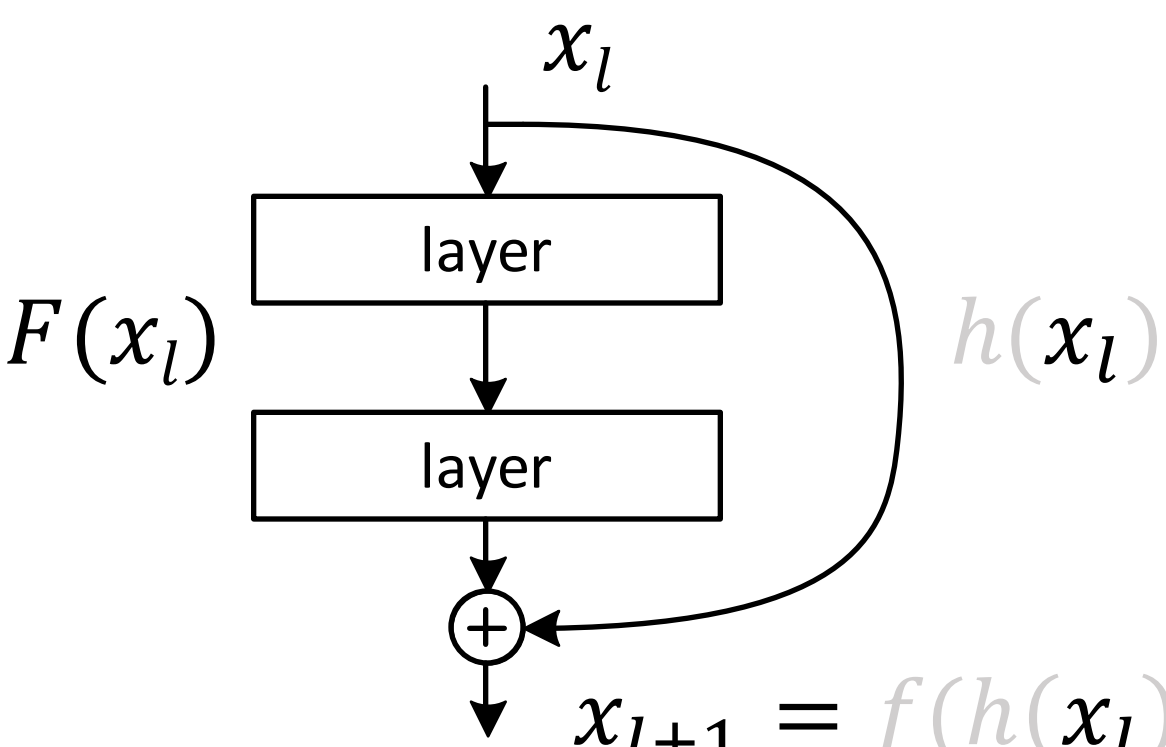


- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

$$x_{l+1} = f(h(x_l) + F(x_l))$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

On identity mappings for **optimization**



- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

$$x_{l+1} = f(h(x_l) + F(x_l))$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

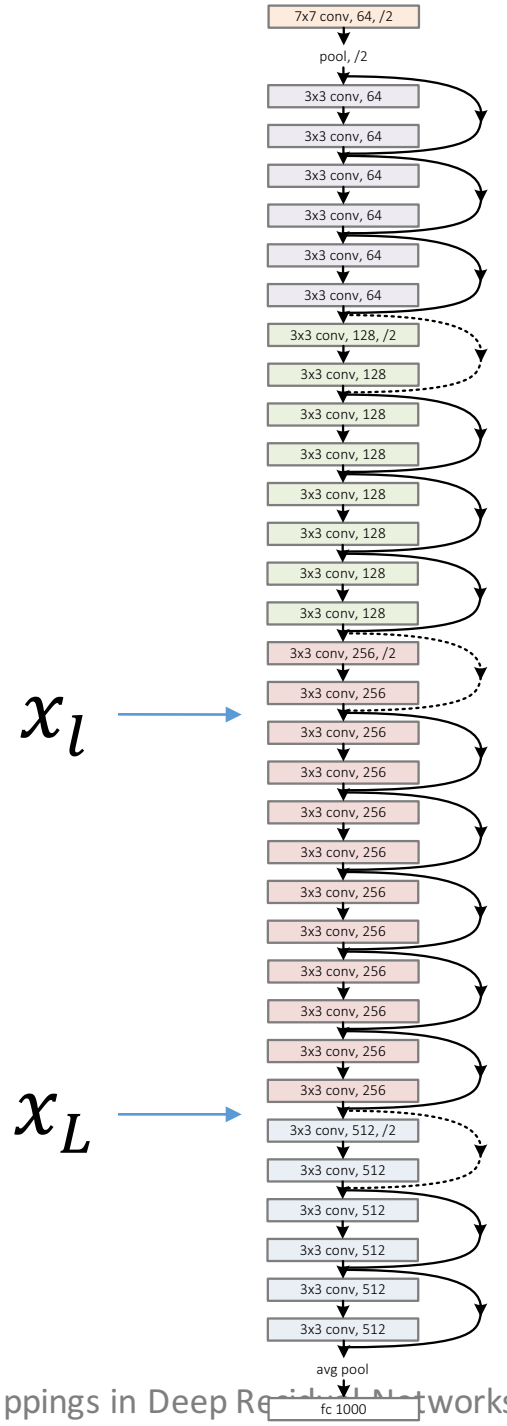
$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Very smooth forward propagation

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

- Any x_l is directly forward-prop to any x_L, plus residual.
- Any x_L is an additive outcome.
  - in contrast to multiplicative: $x_L = \prod_{i=l}^{L-1} W_i x_l$

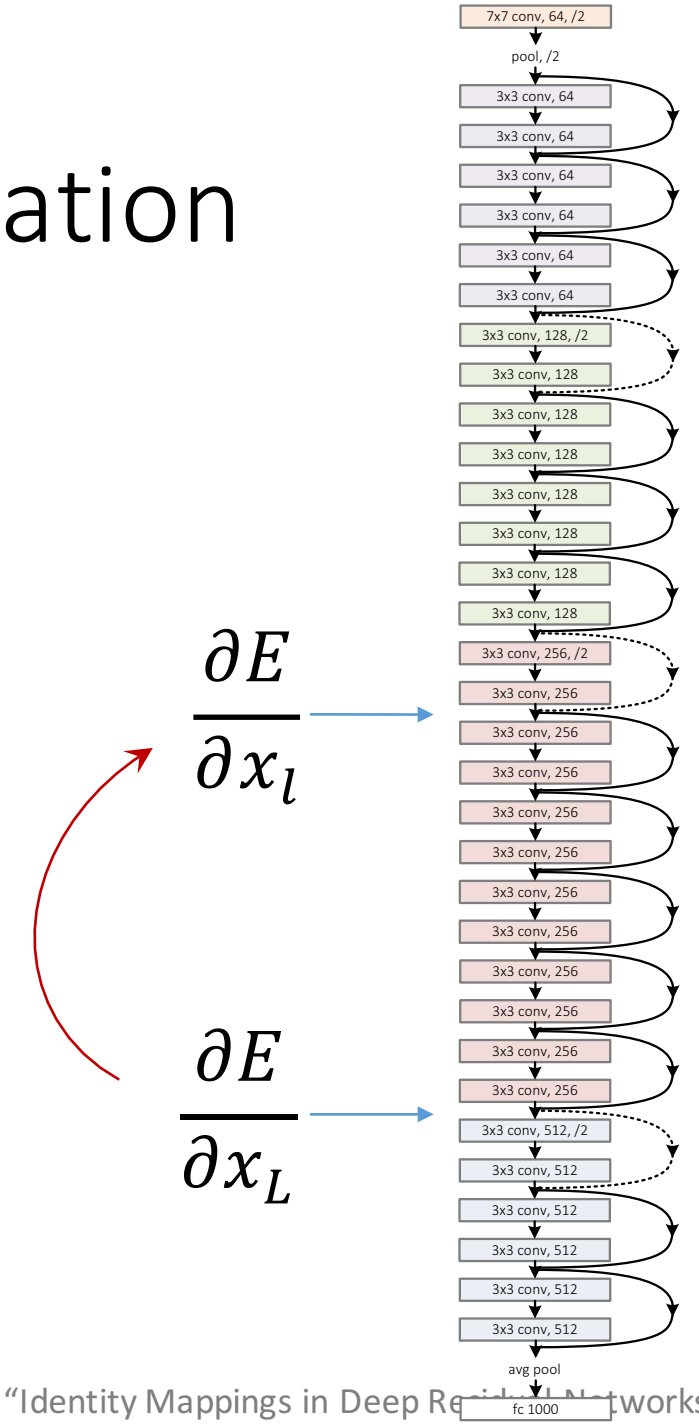


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Very smooth backward propagation

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$$

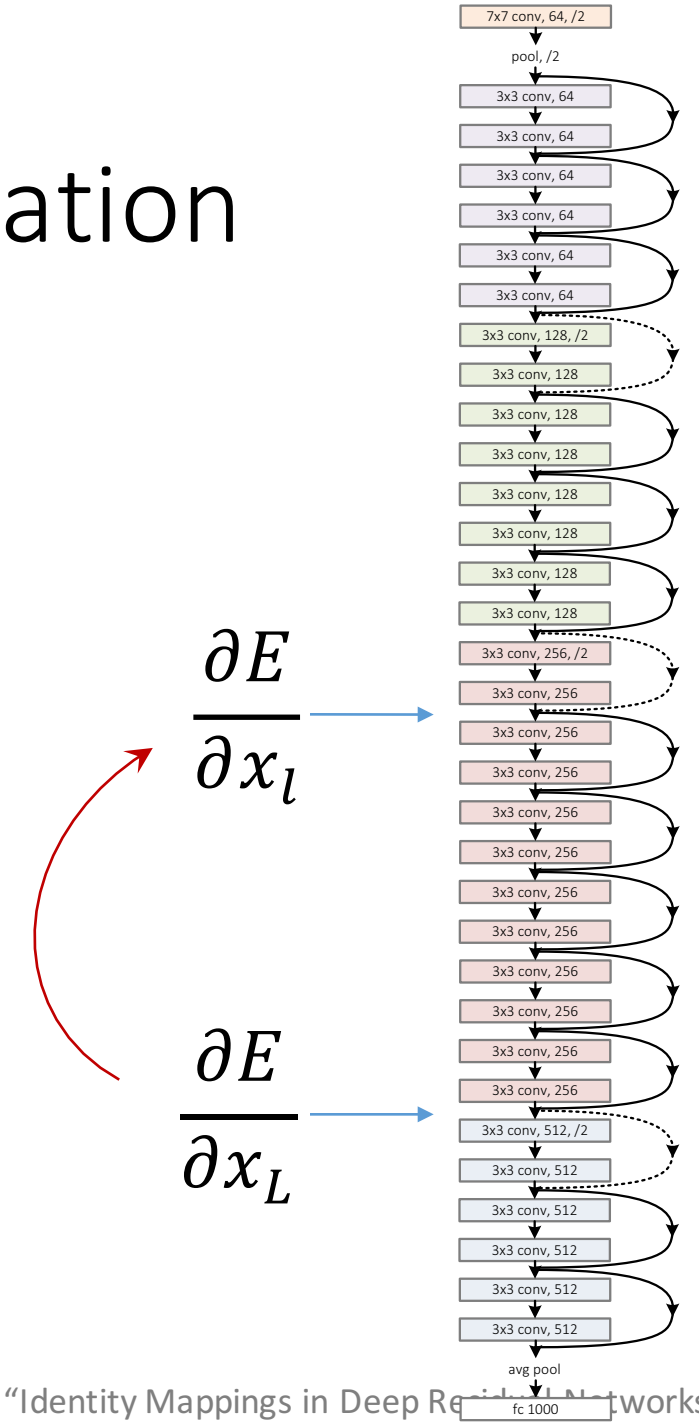


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Very smooth backward propagation

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$$

- Any $\frac{\partial E}{\partial x_L}$ is directly back-prop to any $\frac{\partial E}{\partial x_l}$, plus residual.
- Any $\frac{\partial E}{\partial x_l}$ is additive; unlikely to vanish
  - in contrast to multiplicative: $\frac{\partial E}{\partial x_l} = \prod_{i=l}^{L-1} W_i \frac{\partial E}{\partial x_L}$



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Residual for every layer

forward: $x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$

Enabled by:

- shortcut mapping: h = identity
- after-add mapping: f = identity

backward: $\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Experiments

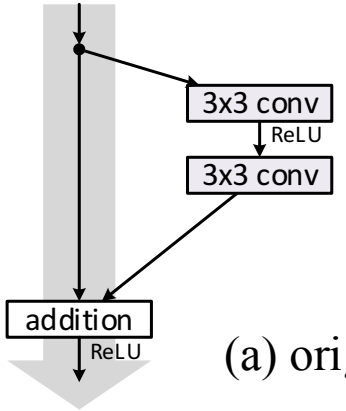
- Set 1: what if shortcut mapping $h \neq$ identity
- Set 2: what if after-add mapping f is identity
- Experiments on ResNets with more than 100 layers
  - deeper models suffer more from optimization difficulty

Experiment Set 1:
what if shortcut mapping h ≠ identity?

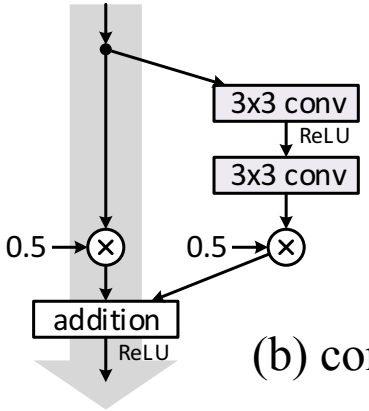
* ResNet-110 on CIFAR-10

$h(x) = x$

error: 6.6%



(a) original



(b) constant scaling

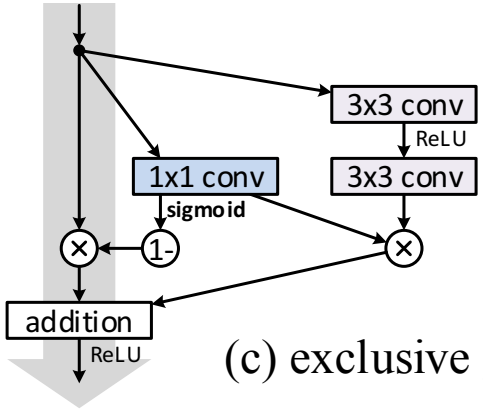
$h(x) = 0.5x$

error: 12.4%

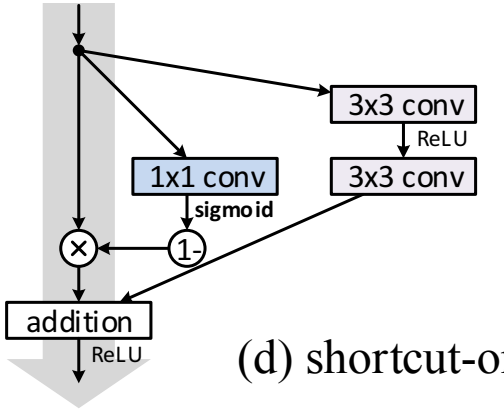
$h(x) = \text{gate} \cdot x$

error: 8.7%

*similar to "Highway Network"



(c) exclusive gating



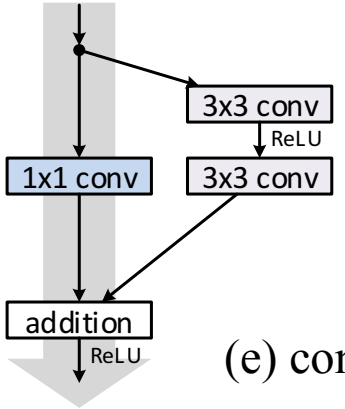
(d) shortcut-only gating

$h(x) = \text{gate} \cdot x$

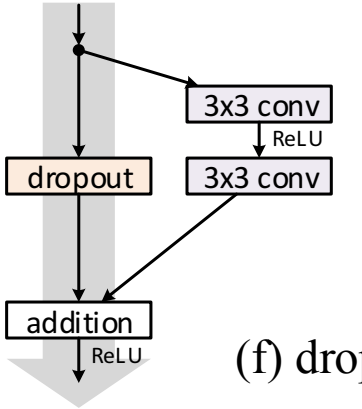
error: 12.9%

$h(x) = \text{conv}(x)$

error: 12.2%



(e) conv shortcut



(f) dropout shortcut

$h(x) = \text{dropout}(x)$

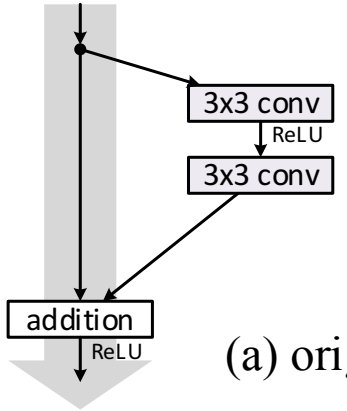
error: > 20%

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

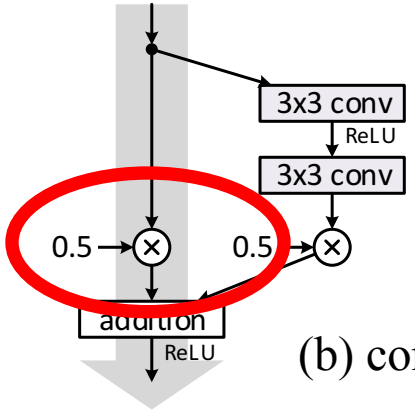
* ResNet-110 on CIFAR-10

$h(x) = x$

error: 6.6%



(a) original



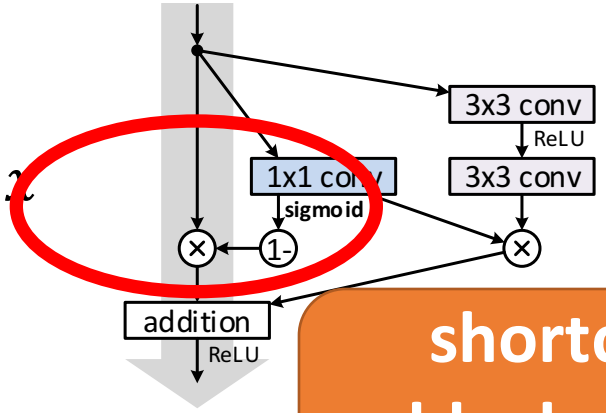
(b) constant scaling

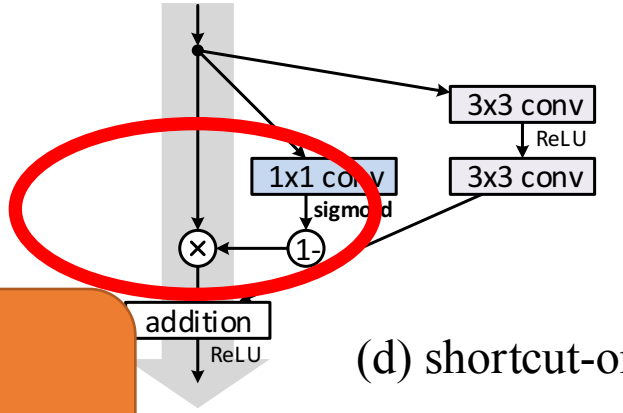
$h(x) = 0.5x$

error: 12.4%

$h(x) = \text{gate} \cdot x$

error: 8.7%





(d) shortcut-only gating

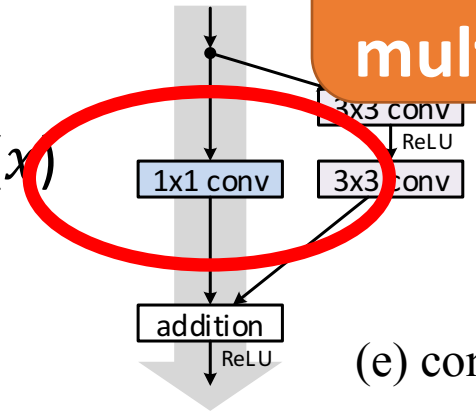
$h(x) = \text{gate} \cdot x$

error: 12.9%

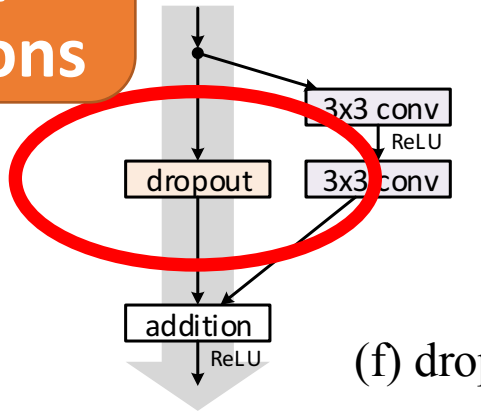
shortcuts blocked by multiplications

$h(x) = \text{conv}(x)$

error: 12.2%



(e) conv shortcut



(f) dropout shortcut

$h(x) = \text{dropout}(x)$

error: > 20%

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

If h is multiplicative, e.g. $h(x) = \lambda x$

forward: $x_L = \lambda^{L-l} x_l + \sum_{i=l}^{L-1} \hat{F}(x_i)$

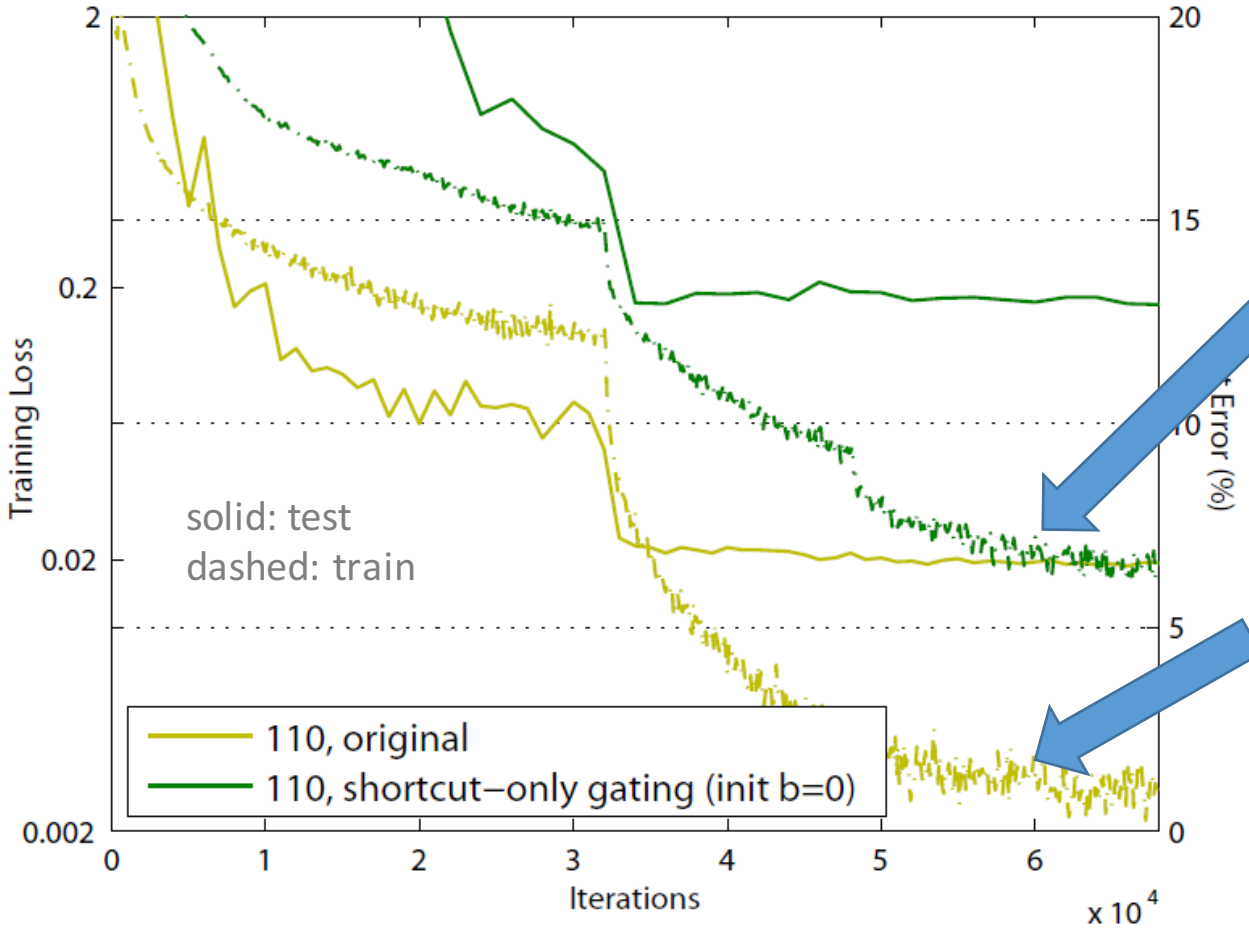
- if h is multiplicative, shortcuts are blocked
- direct propagation is decayed

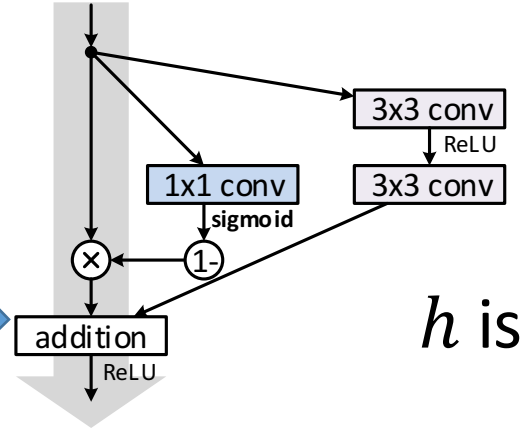
backward: $\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(\lambda^{L-l} + \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} \hat{F}(x_i)\right)$

typo: 1 should be l

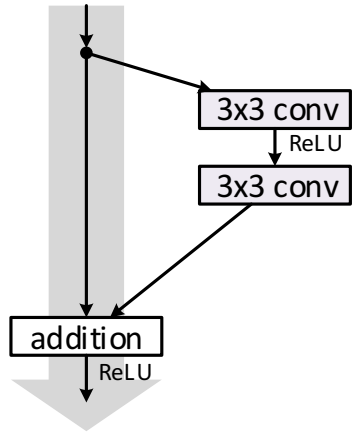
*assuming f = identity

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.





h is gating

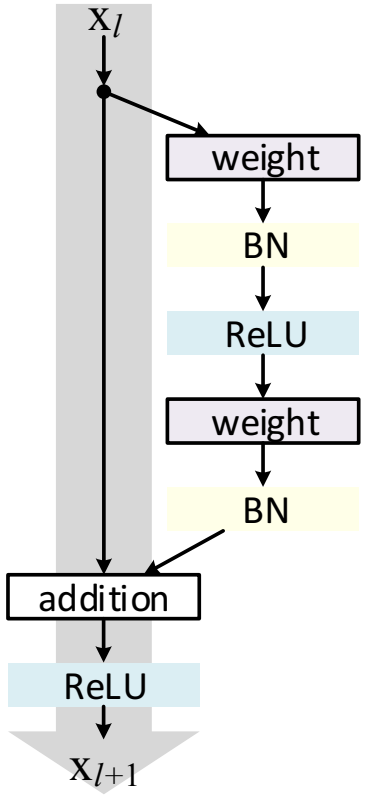


h is identity

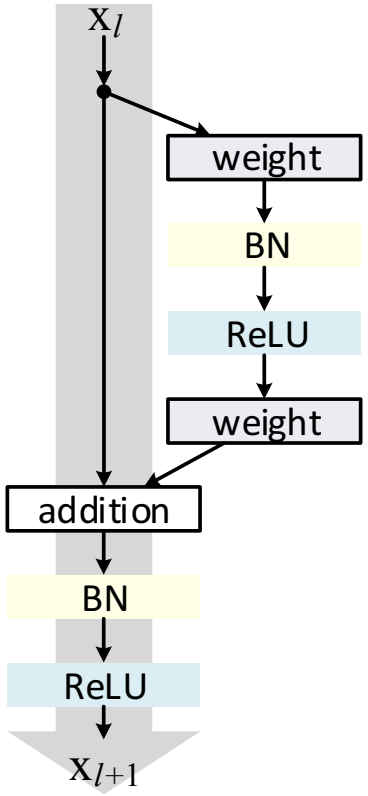
- gating should have better representation ability (identity is a special case), but
- optimization difficulty dominates results

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

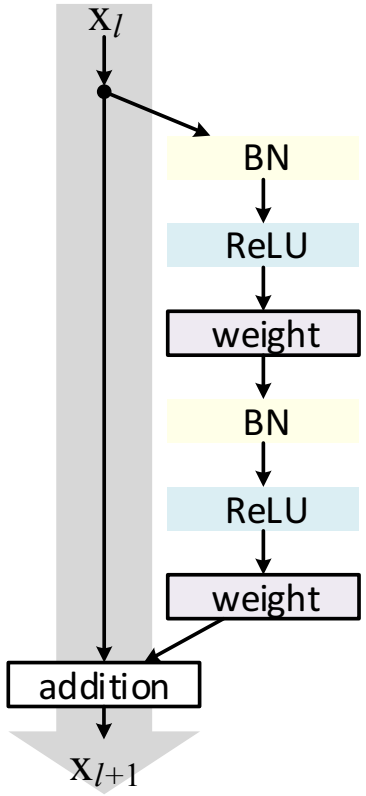
Experiment Set 2:
what if after-add mapping f is identity



f is ReLU
(original ResNet)

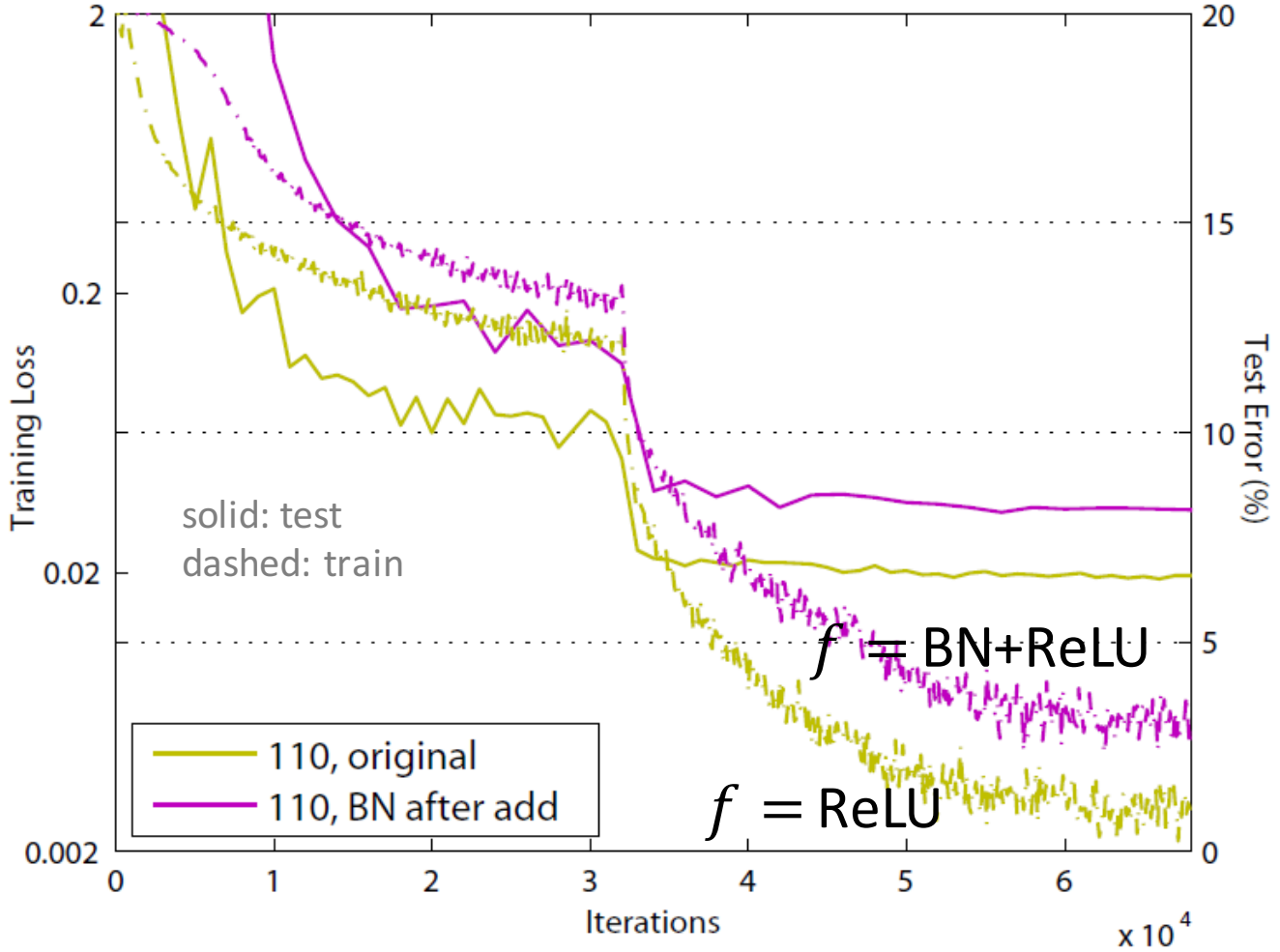


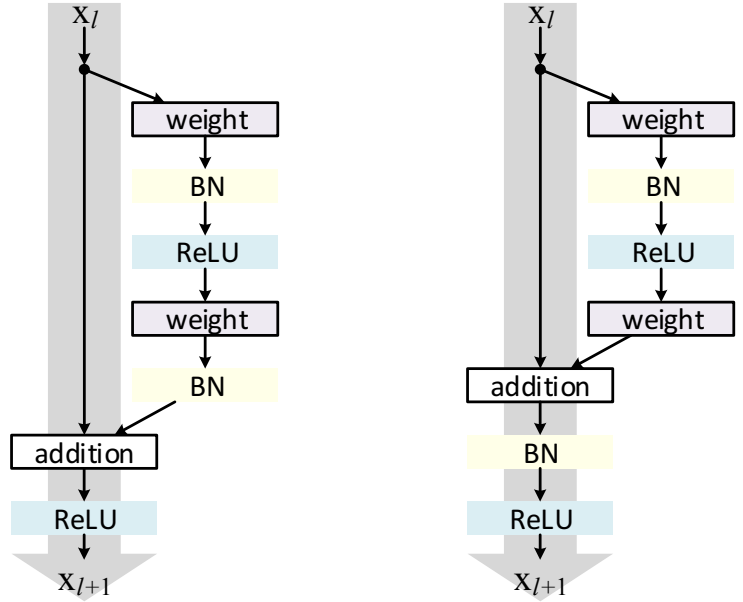
f is BN+ReLU



f is identity
(**pre-activation** ResNet)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.



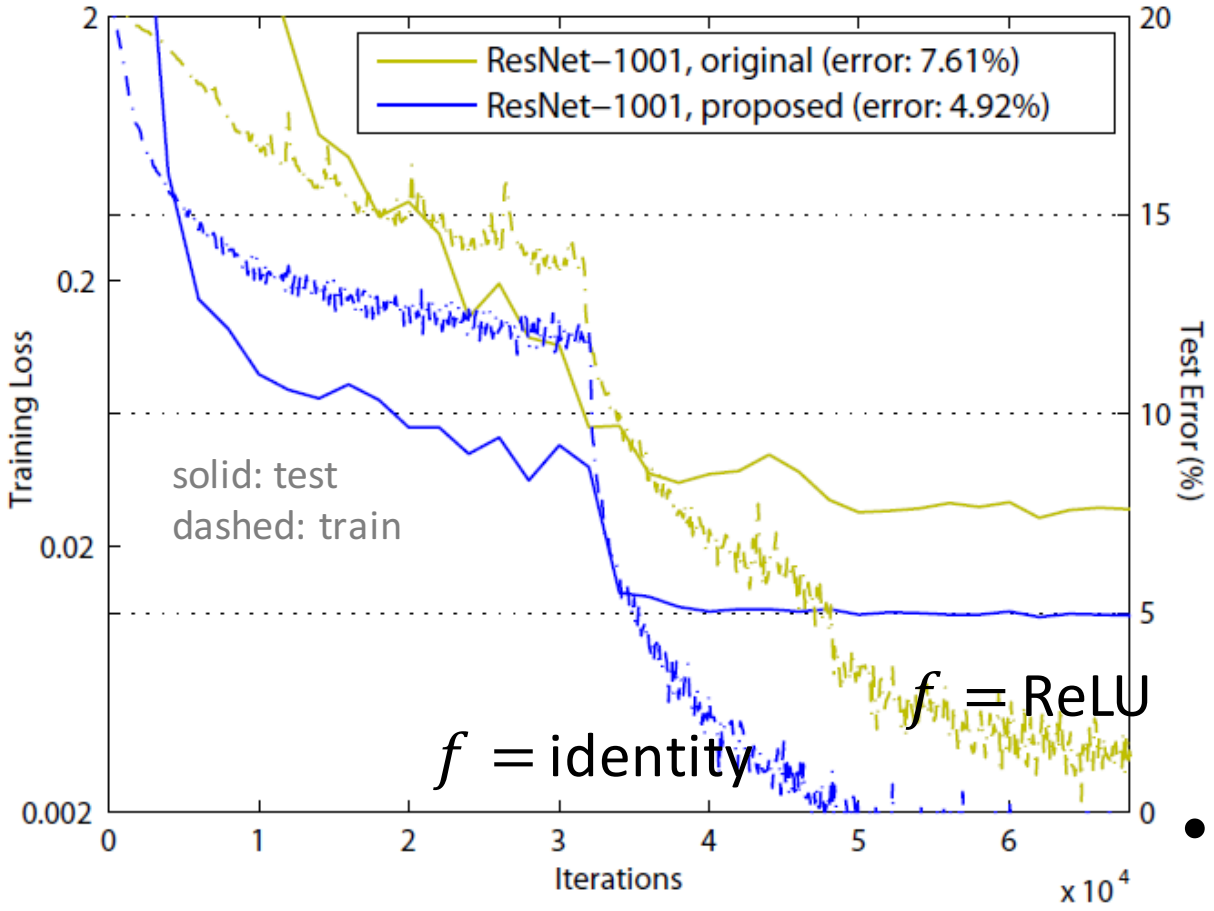


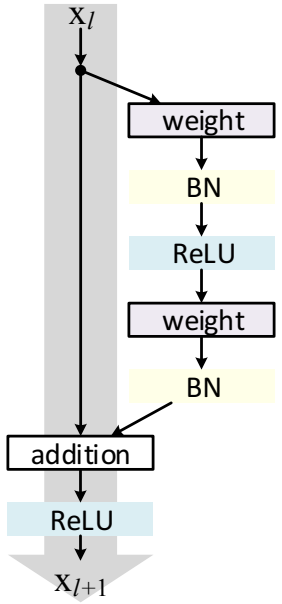
f = ReLU f = BN+ReLU

- BN could block prop
- Keep the shortest pass as smooth as possible

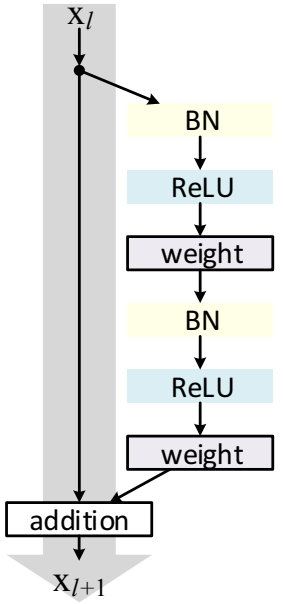
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

1001-layer ResNets on CIFAR-10





f = ReLU



f = identity

- ReLU could block prop when there are 1000 layers
- pre-activation design eases optimization (and improves generalization; see paper)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Comparisons on CIFAR-10/100

What is the difference between CIFAR-10 and CIFAR-100?

CIFAR-10

method	error (%)
NIN	8.81
DSN	8.22
FitNet	8.39
Highway	7.72
ResNet-110 (1.7M)	6.61
ResNet-1202 (19.4M)	7.93
ResNet-164, pre-activation (1.7M)	5.46
ResNet-1001, pre-activation (10.2M)	**4.92** (4.89±0.14)

CIFAR-100

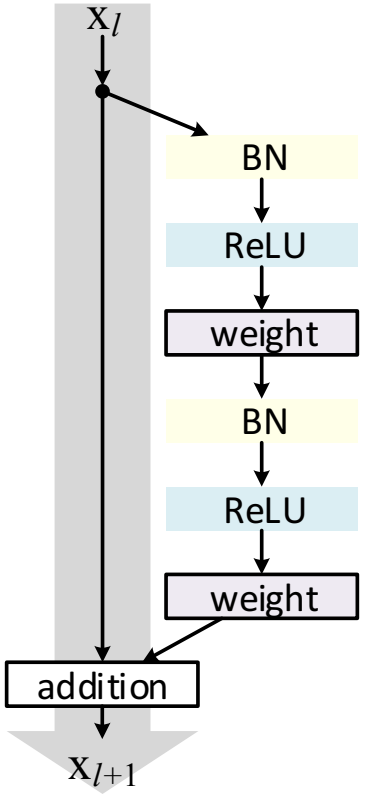
method	error (%)
NIN	35.68
DSN	34.57
FitNet	35.04
Highway	32.39
ResNet-164 (1.7M)	25.16
ResNet-1001 (10.2M)	27.82
ResNet-164, pre-activation (1.7M)	24.33
ResNet-1001, pre-activation (10.2M)	**22.71** (22.68±0.22)

*all based on moderate augmentation

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Summary of observations

- Keep the shortest path as smooth as possible
  - by making h and f identity
  - forward/backward signals directly flow through this path
- Features of any layers are additive outcomes
- 1000-layer ResNets can be easily trained and have better accuracy



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

Applications

“Features matter”

"Features matter." (quote [Girshick et al. 2014], the R-CNN paper)

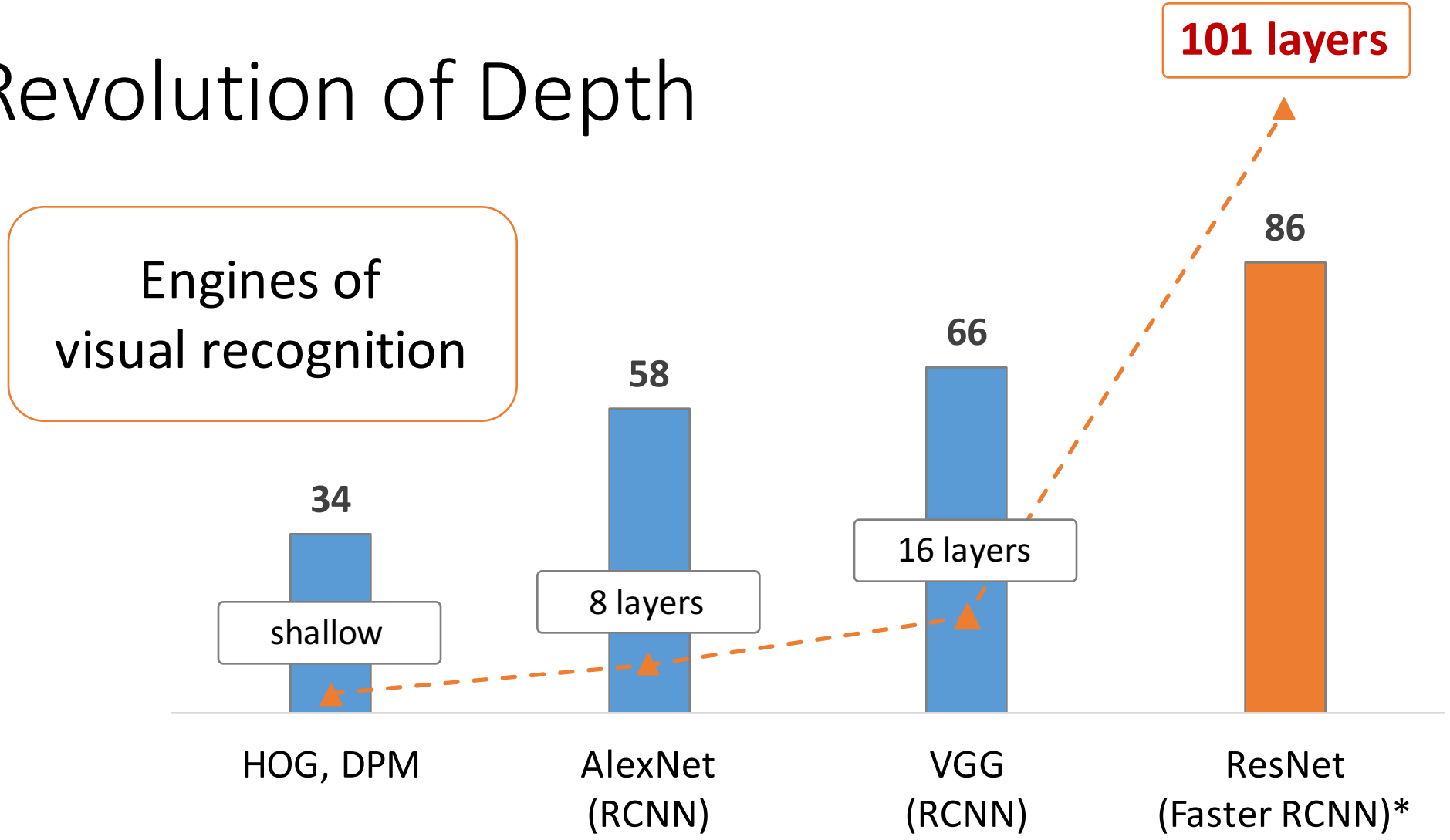
task	2nd-place winner	ResNets	margin (relative)
ImageNet Localization (top-5 error)	12.0	9.0	**27%**
ImageNet Detection (mAP@.5)	53.6	62.1	**16%**
COCO Detection (mAP@.5:.95)	33.5	37.3	**11%**
COCO Segmentation (mAP@.5:.95)	25.1	28.2	**12%**

absolute 8.5% better!

- Our results are all based on ResNet-101
- Deeper features are well transferrable

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

Revolution of Depth

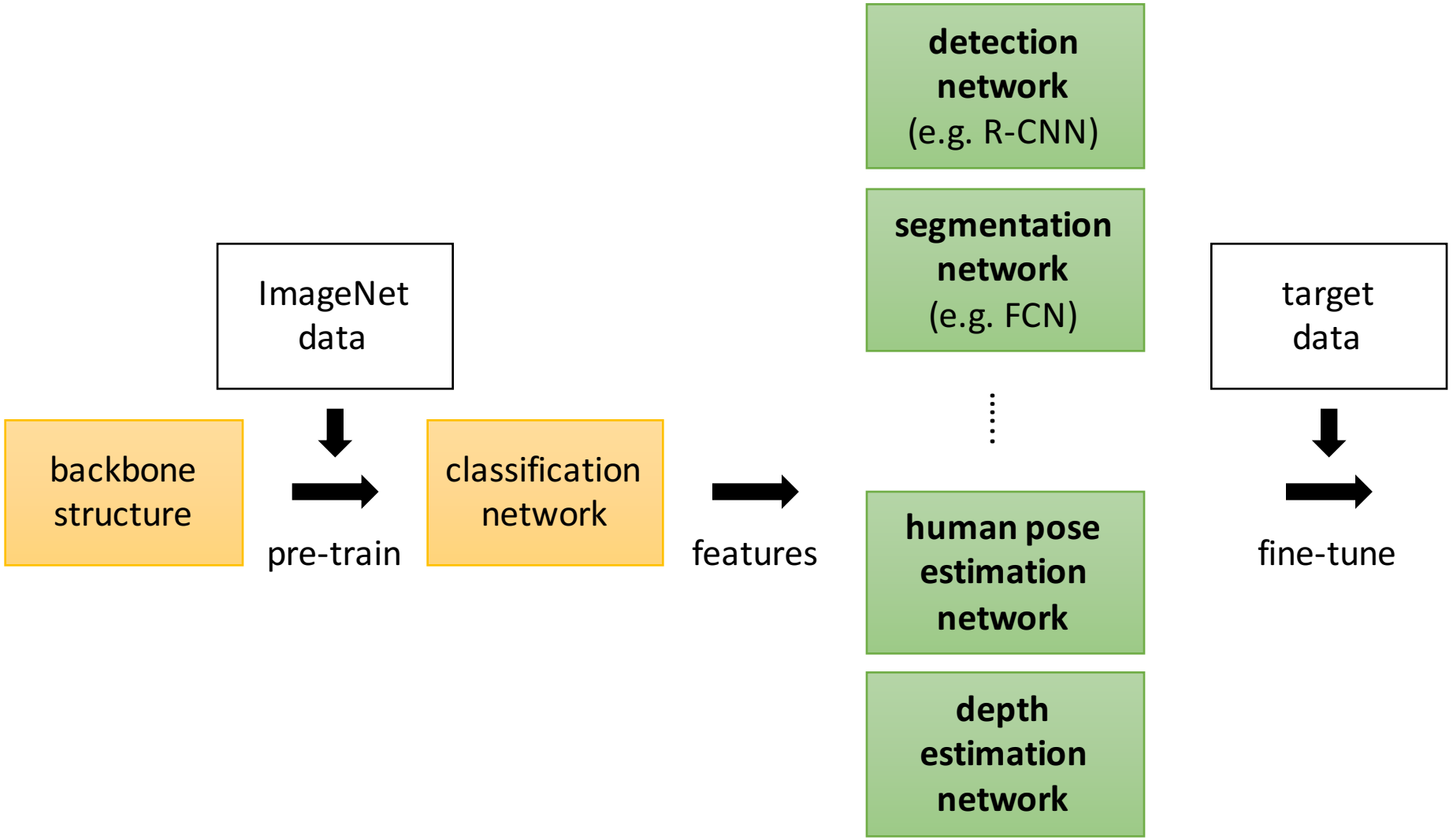


PASCAL VOC 2007 **Object Detection** mAP (%)

*w/ other improvements & more data

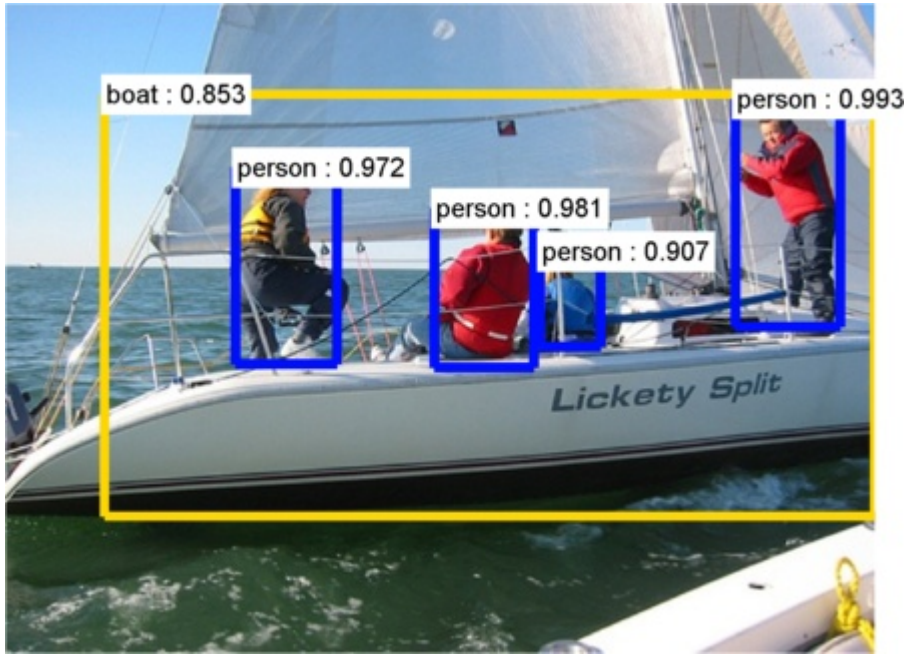
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Learning for Computer Vision



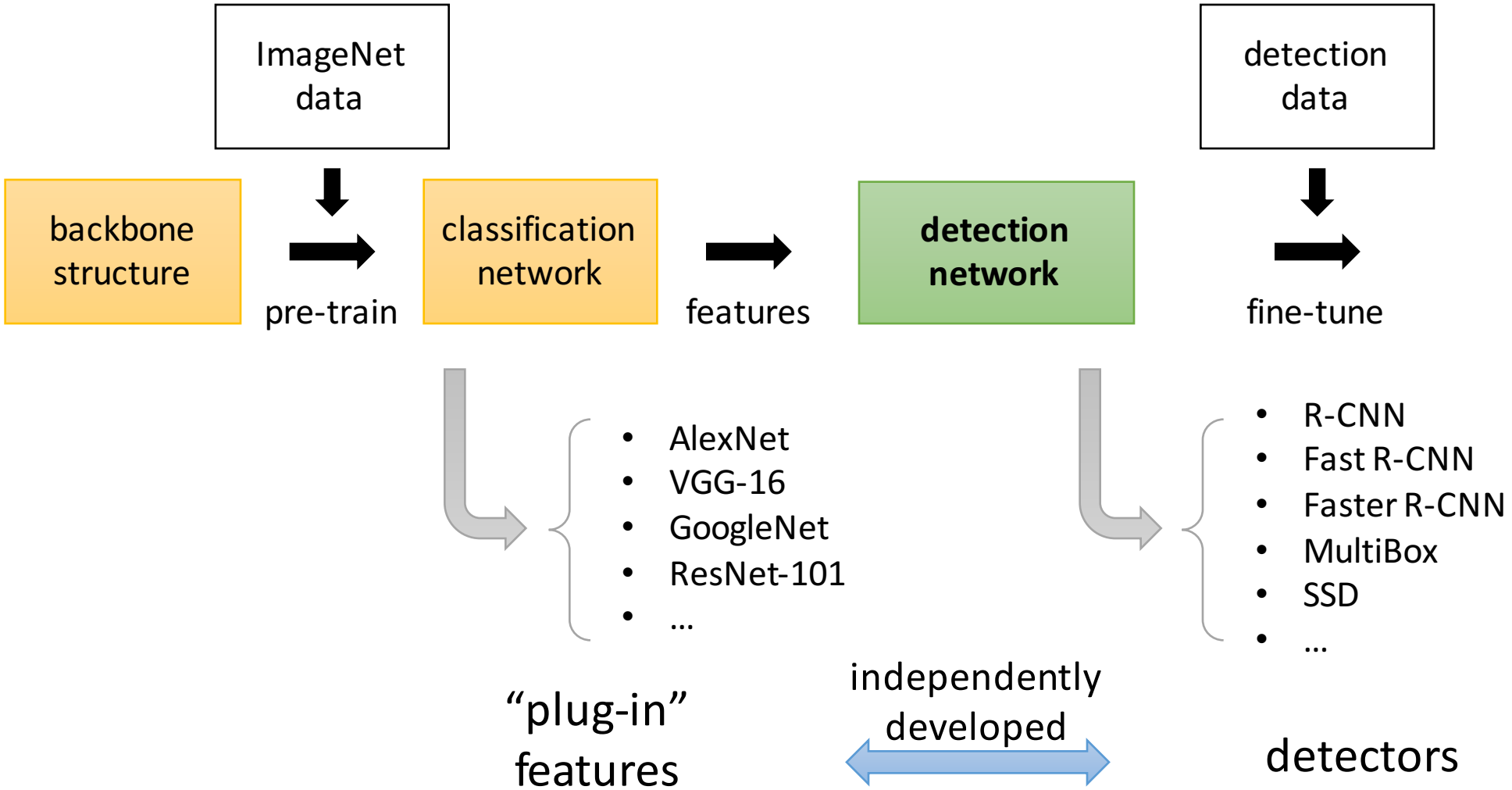
Example: Object Detection

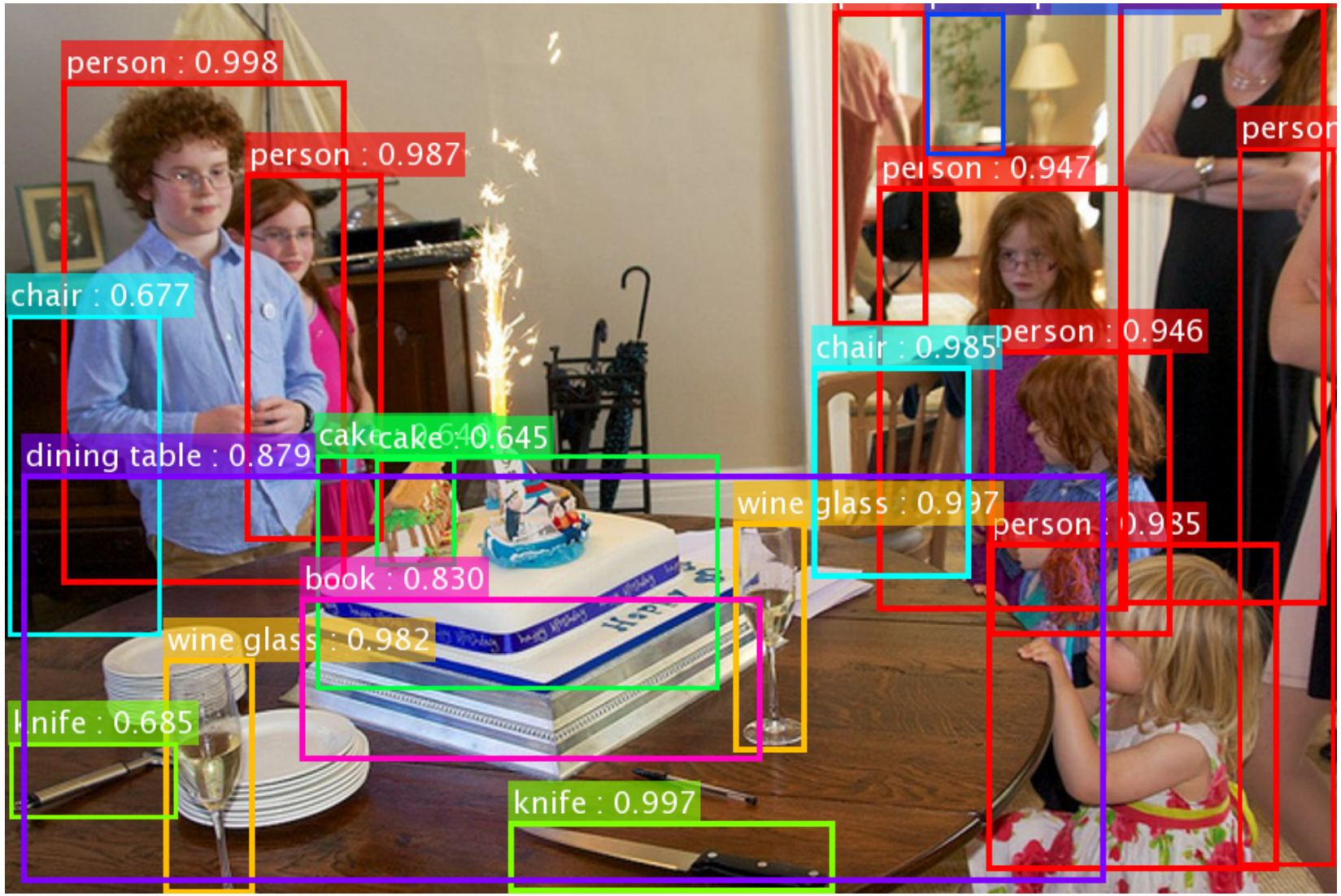
Image Classification
(what?)



Object Detection
(what + where?)

Object Detection

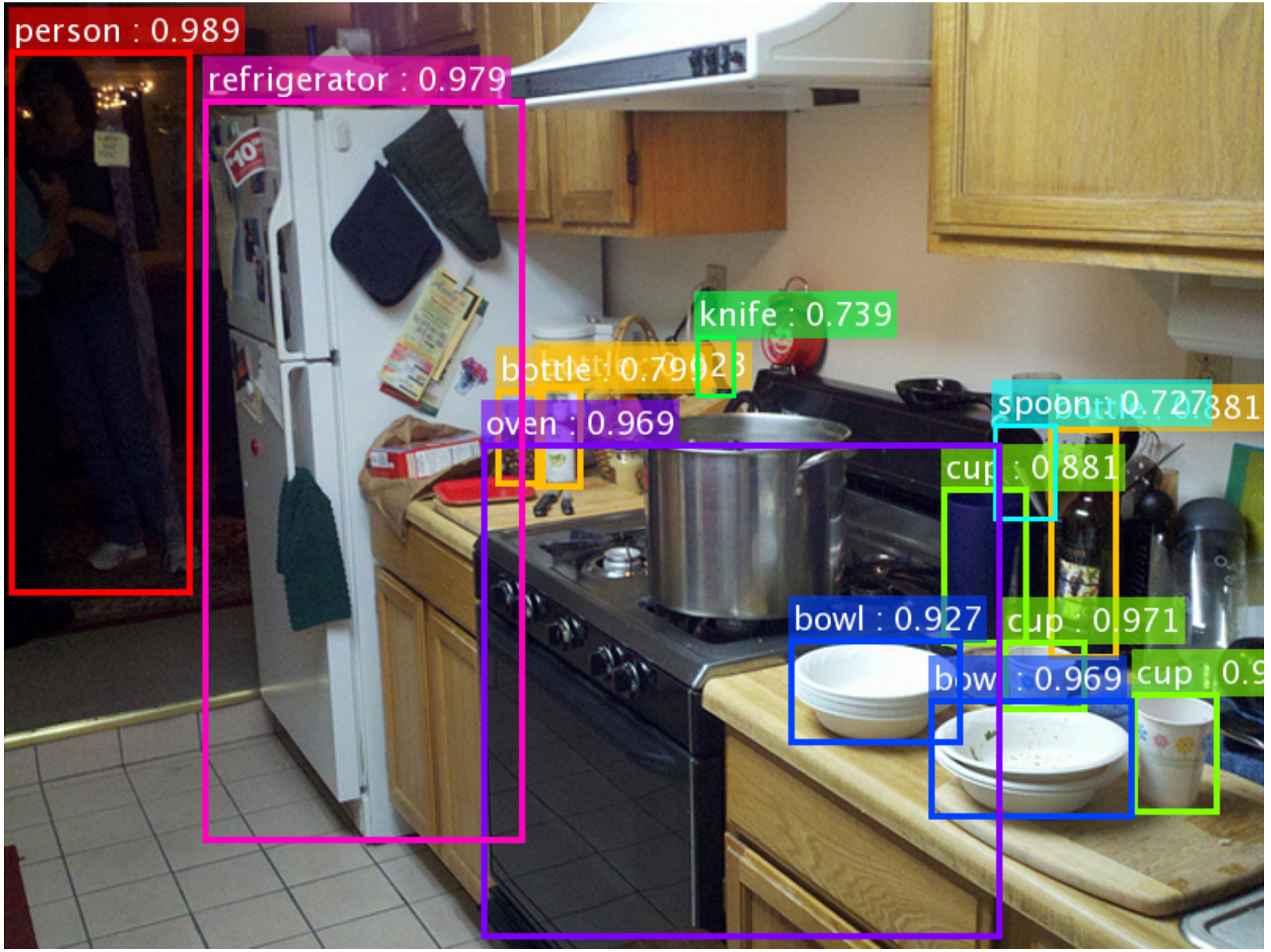




ResNet's object detection result on COCO

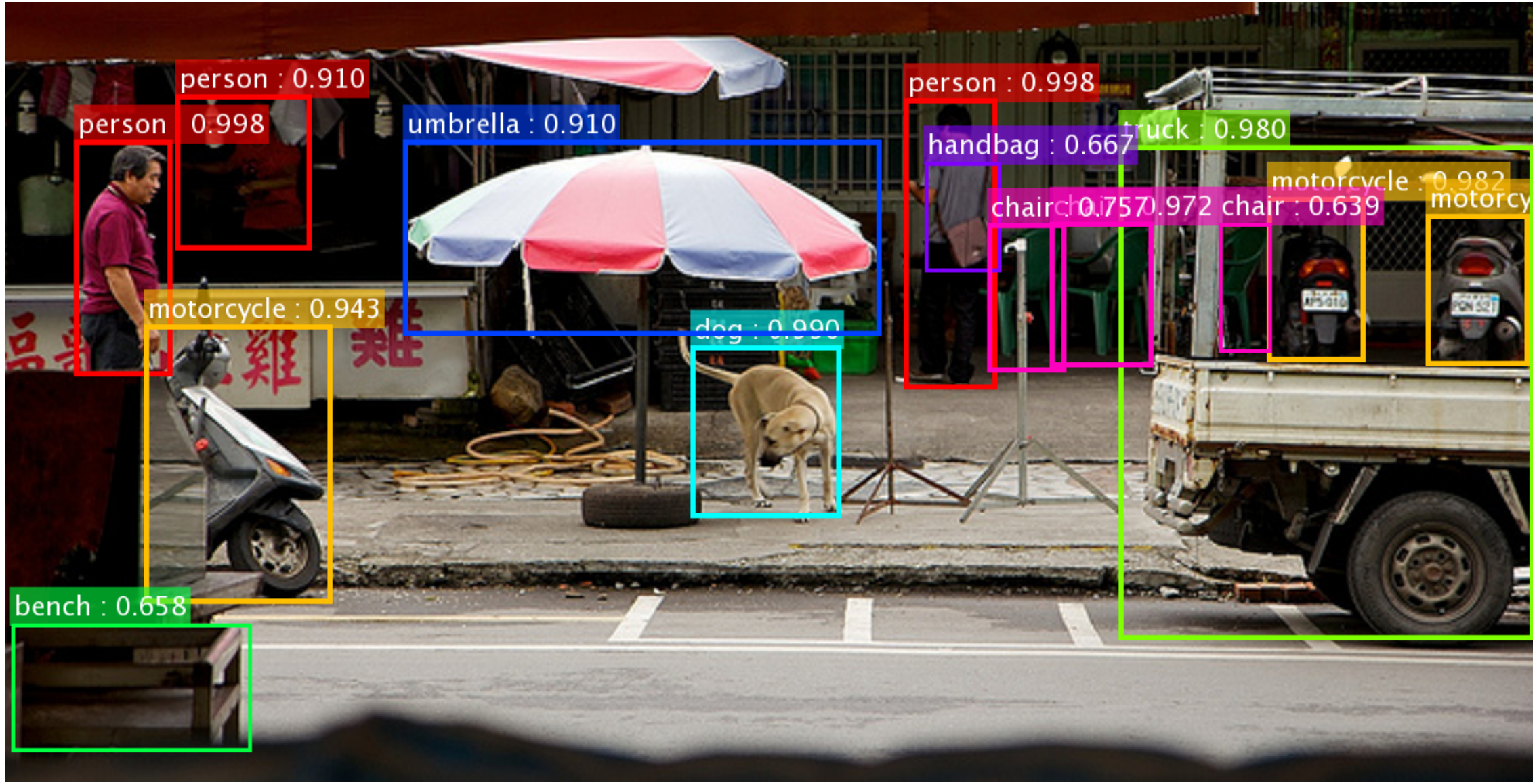
*the original image is from the COCO dataset

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.



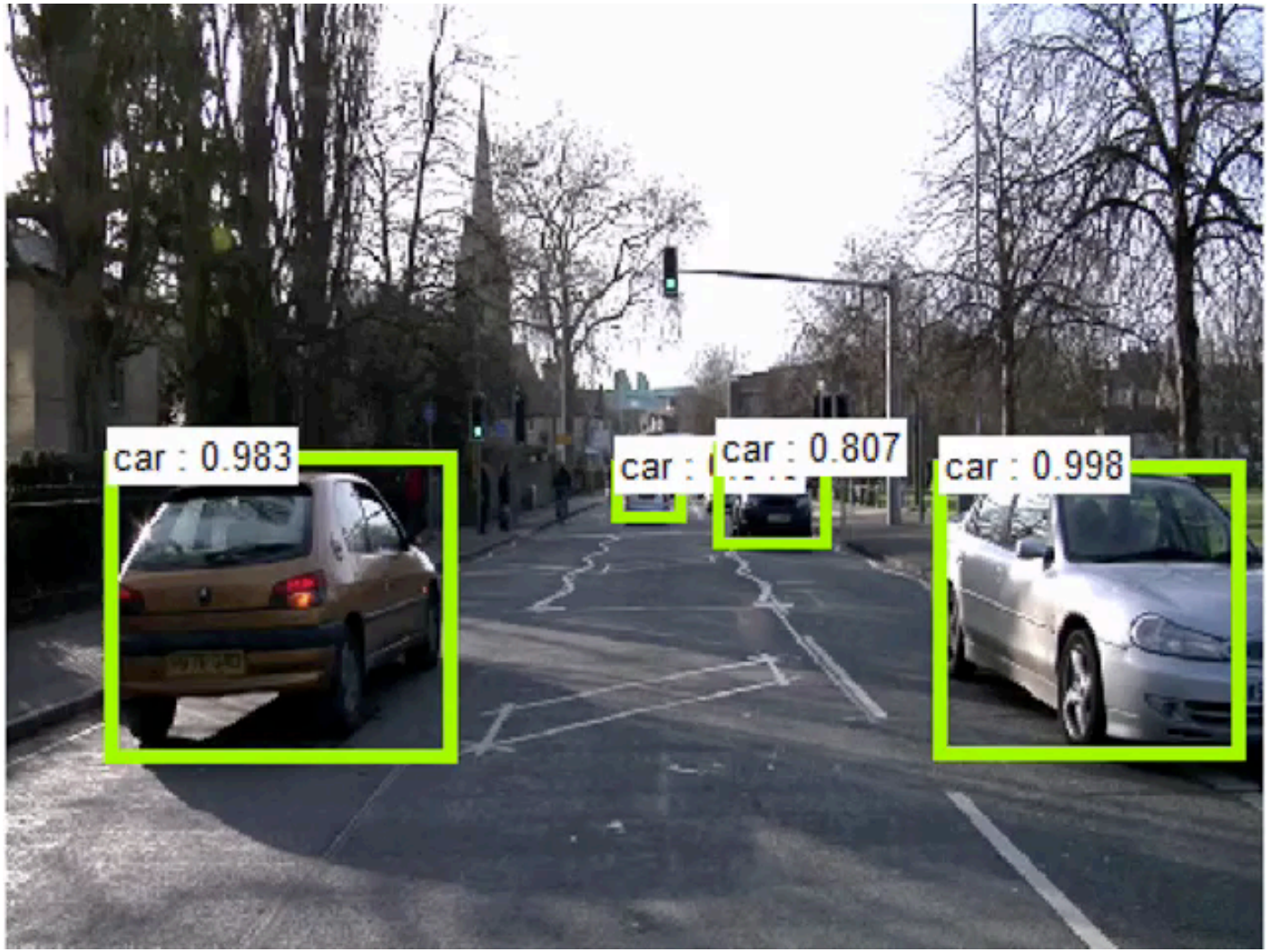
*the original image is from the COCO dataset

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.



*the original image is from the COCO dataset

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.



this video is available online: https://youtu.be/WZmSMkK9VuA

Results on real video. Models trained on MS COCO (80 categories).
(frame-by-frame; no temporal processing)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.

Potential Applications