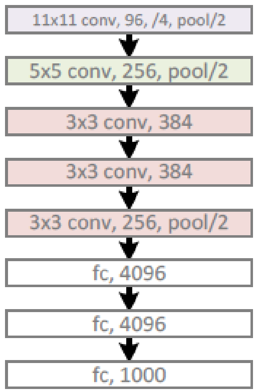
Deep Residual Learning for Image Recognition

Kaiming He - CVPR(2016)

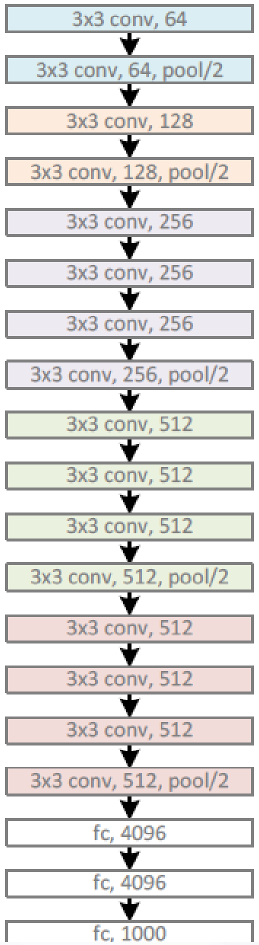
CONTENTS

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

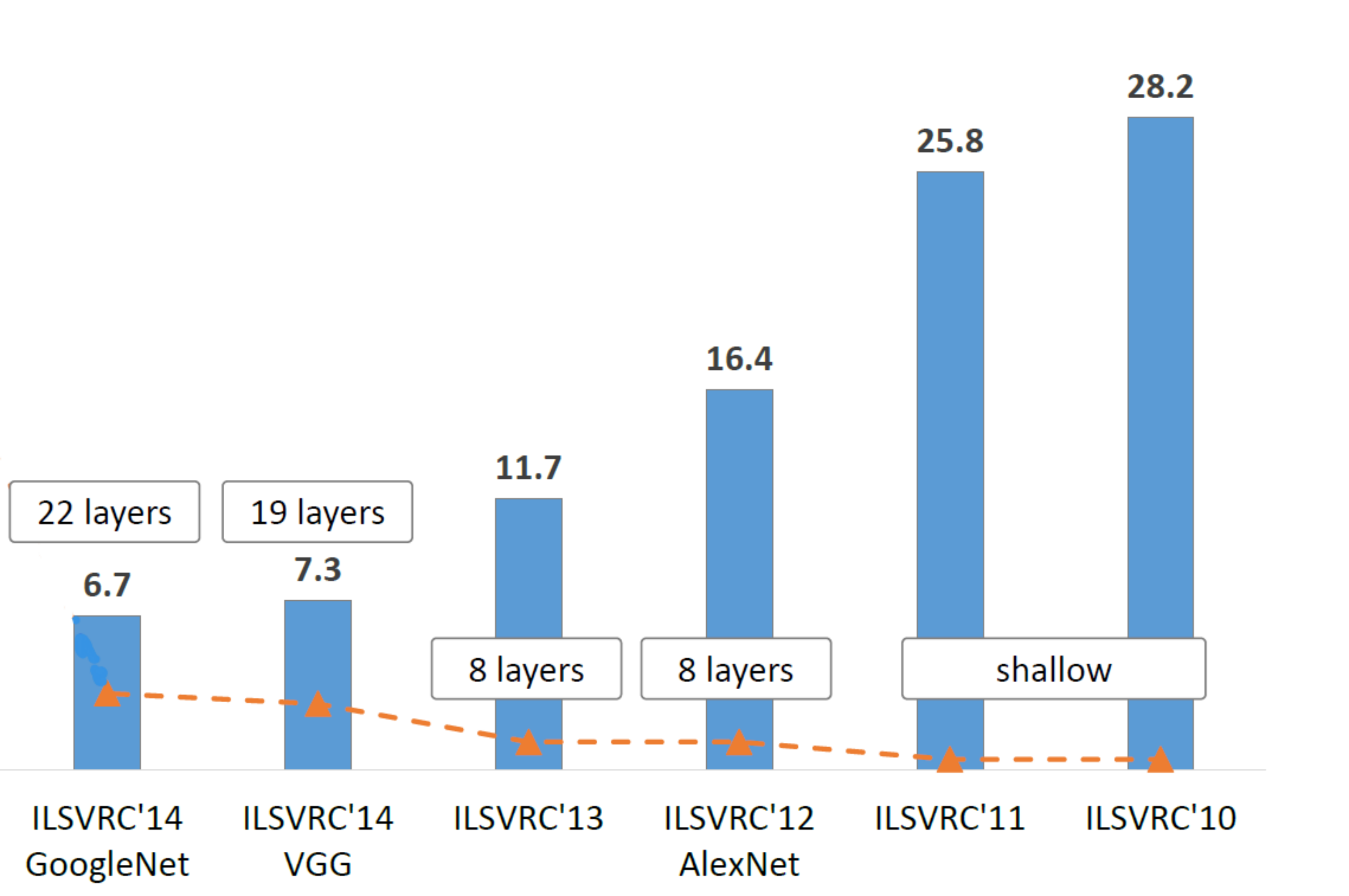


VGG, 19 layers
(ILSVRC 2014)



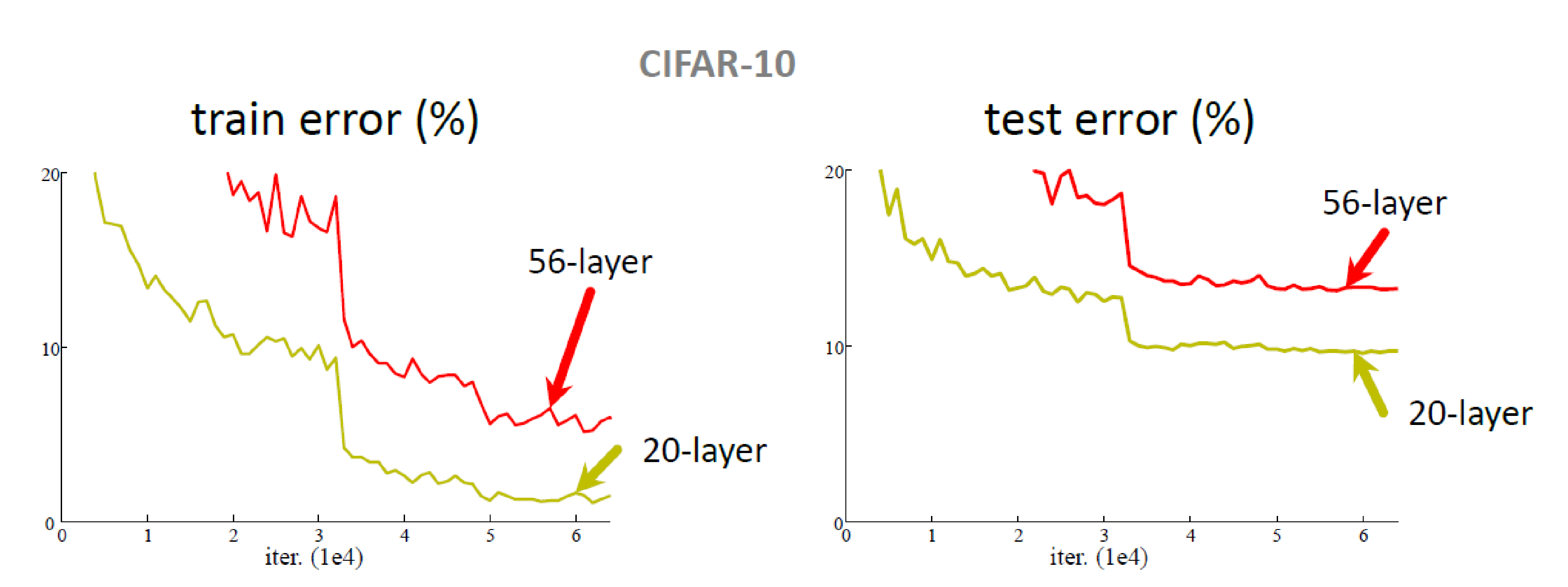
GoogleNet, 22 layers
(ILSVRC 2014)

Revolution of Depth



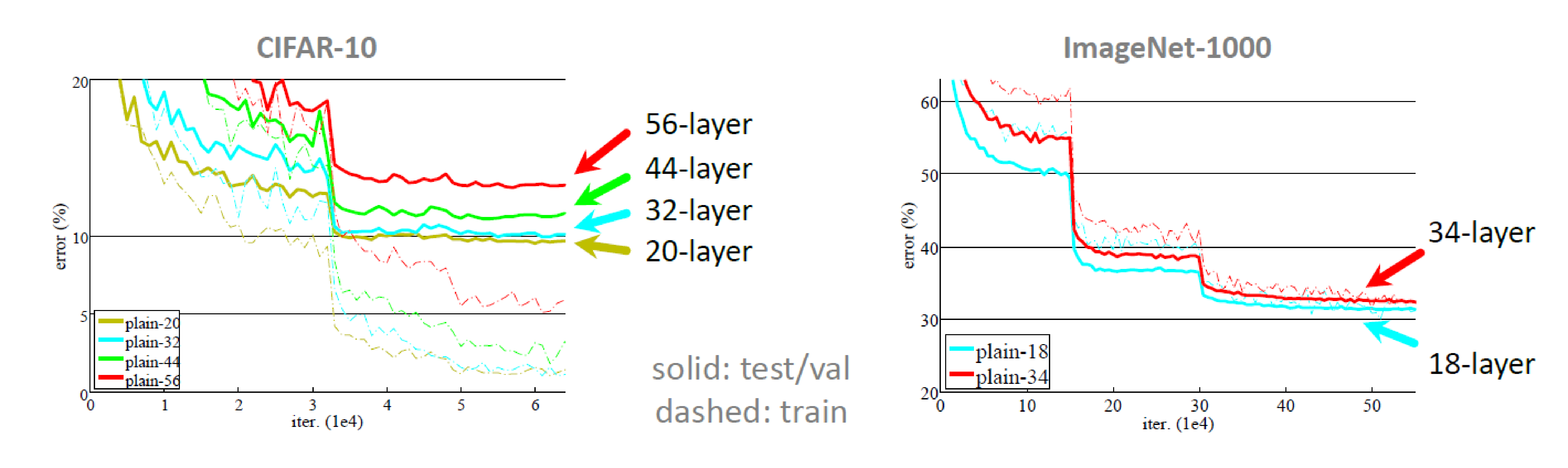
Is learning better networks as simple as stacking more layers?

Simply stacking layers?

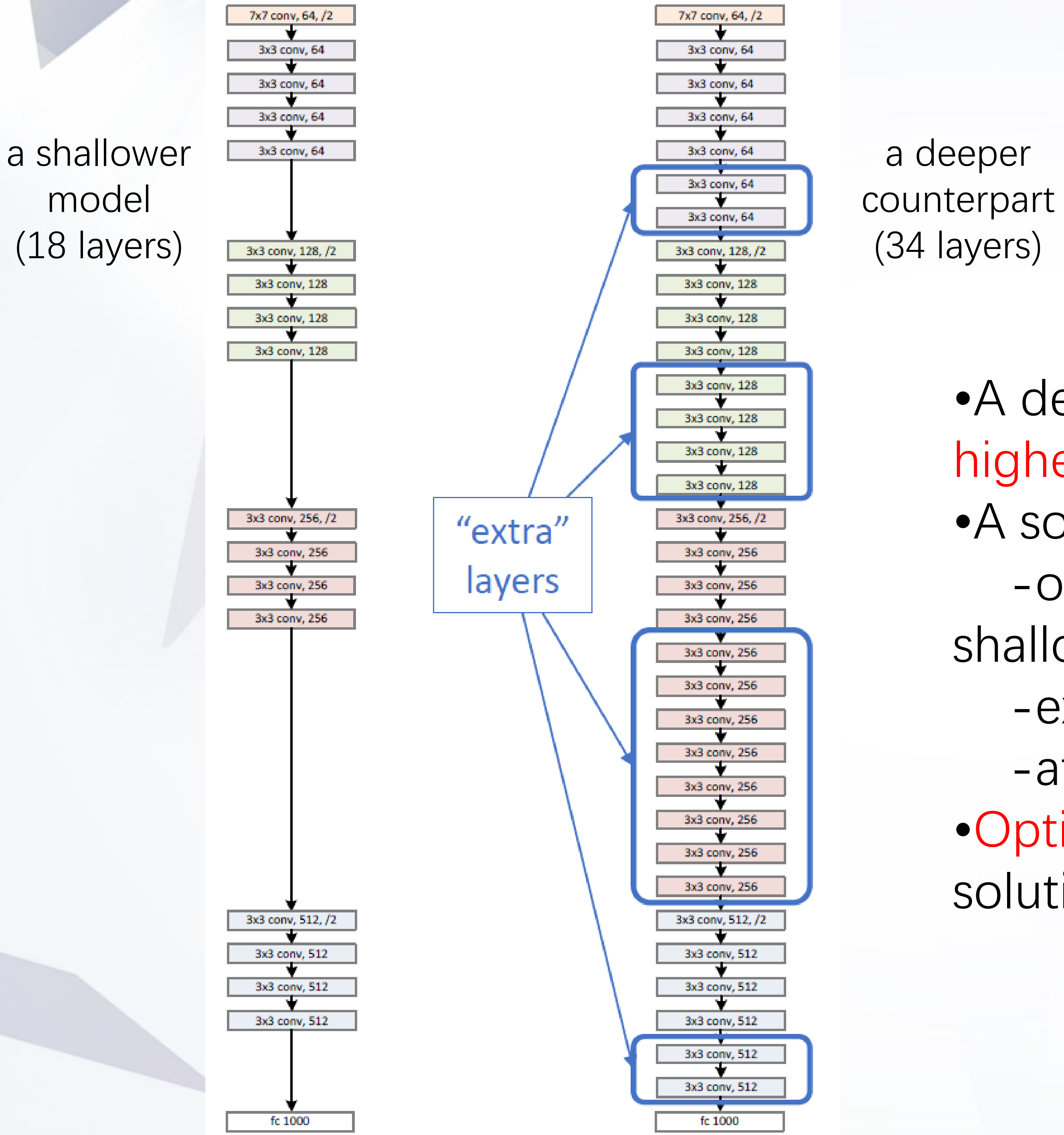


- plainnets: stacking 3x3 conv layers...
- 56-layer net has **higher training error** and test error than 20-layer net

Simply stacking layers?

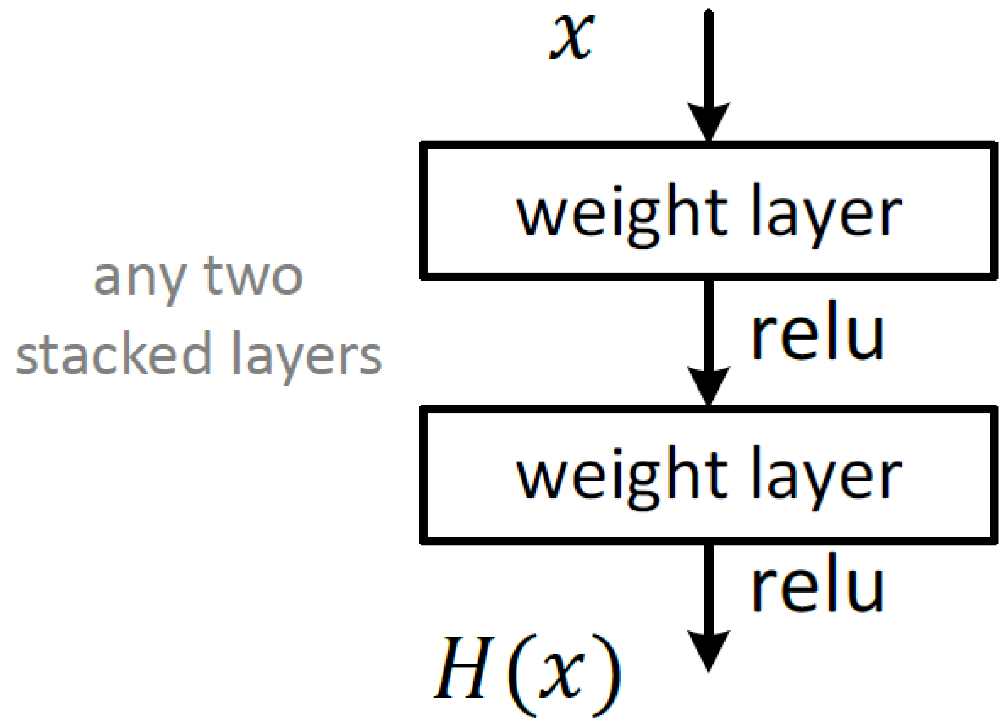


- “Overly deep” plain nets have **higher training error**
- A general phenomenon, observed in many datasets



- 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…

Plaint net



$H(x)$ is any desired mapping,

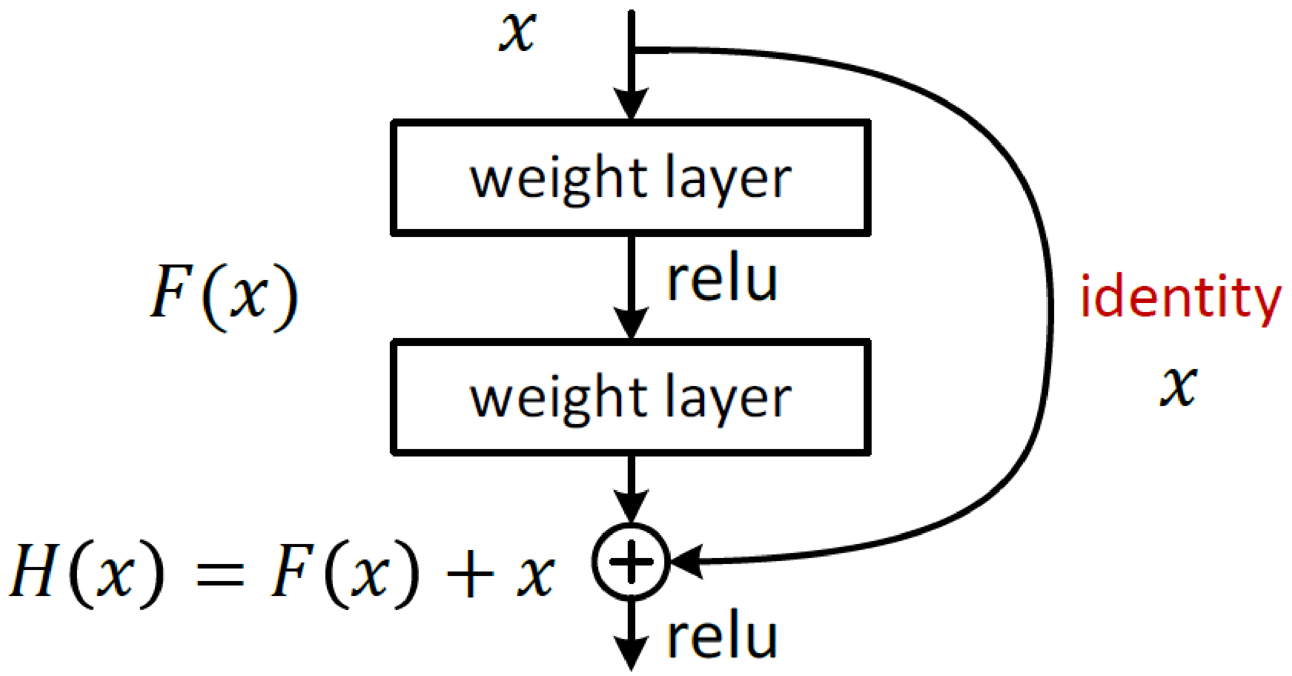
hope the 2 weight layers fit $H(x)$

What is the residual?

Definition(statistics)：
The difference between results obtained by observation and by computation from a formula or between the mean of several observations and any one of them

Example：
Suppose we want to find an x such that $f(x)=b$,given an estimate x_0 of x, the residual is $b-f(x_0)$

Residual net



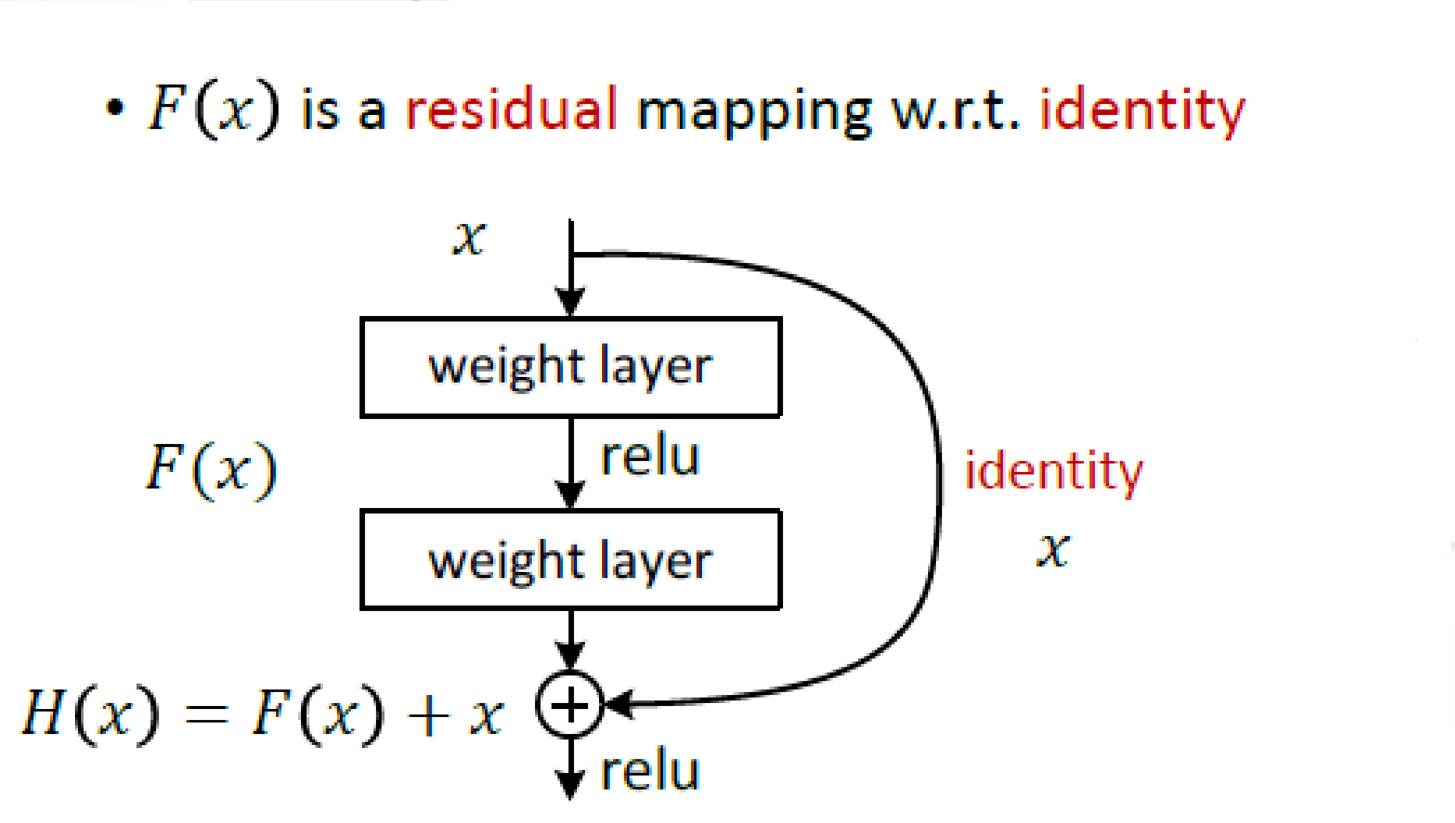
$H(x)$ is any desired mapping,

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

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

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

Residual net

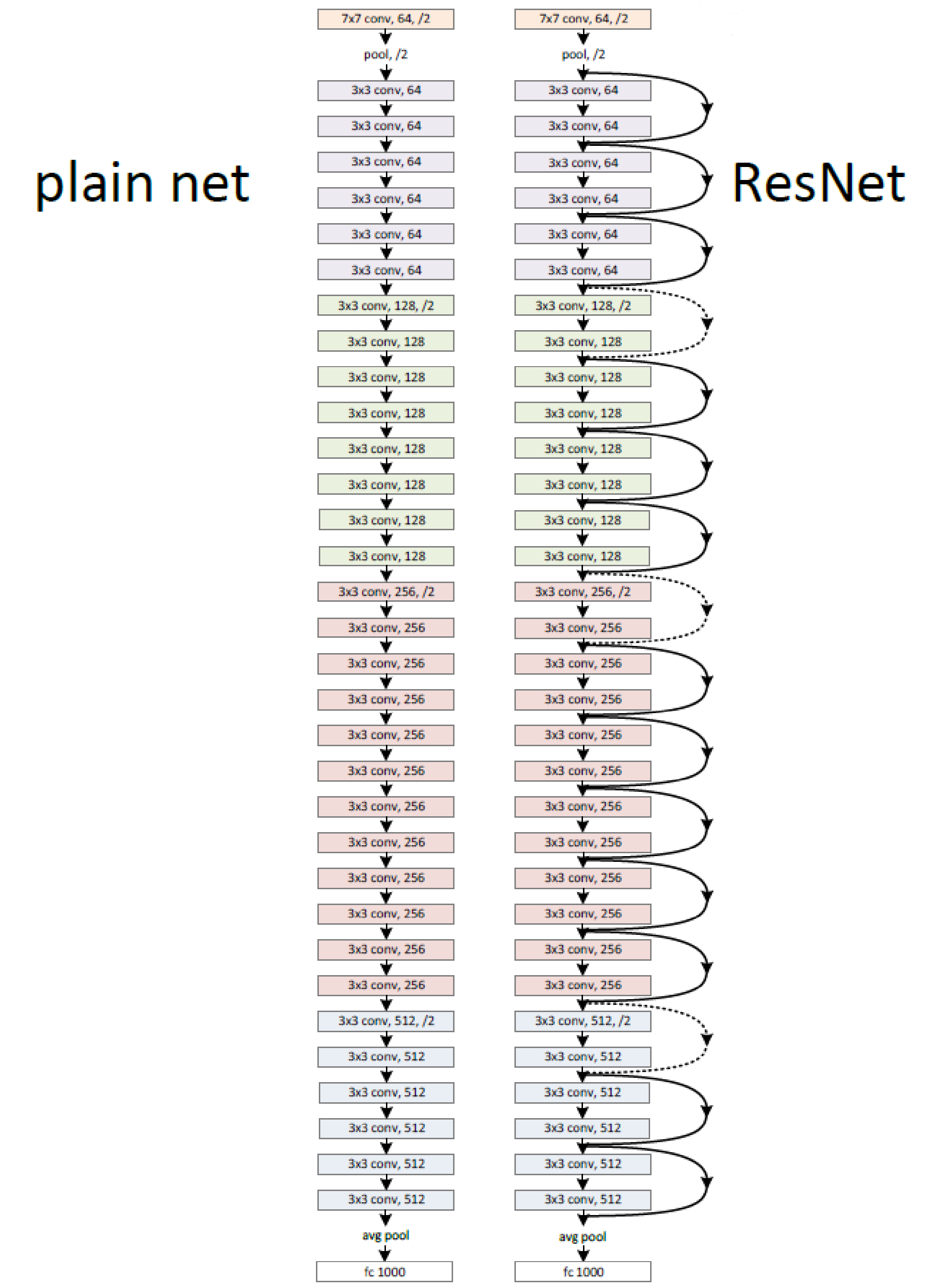


- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

Network "Design"

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

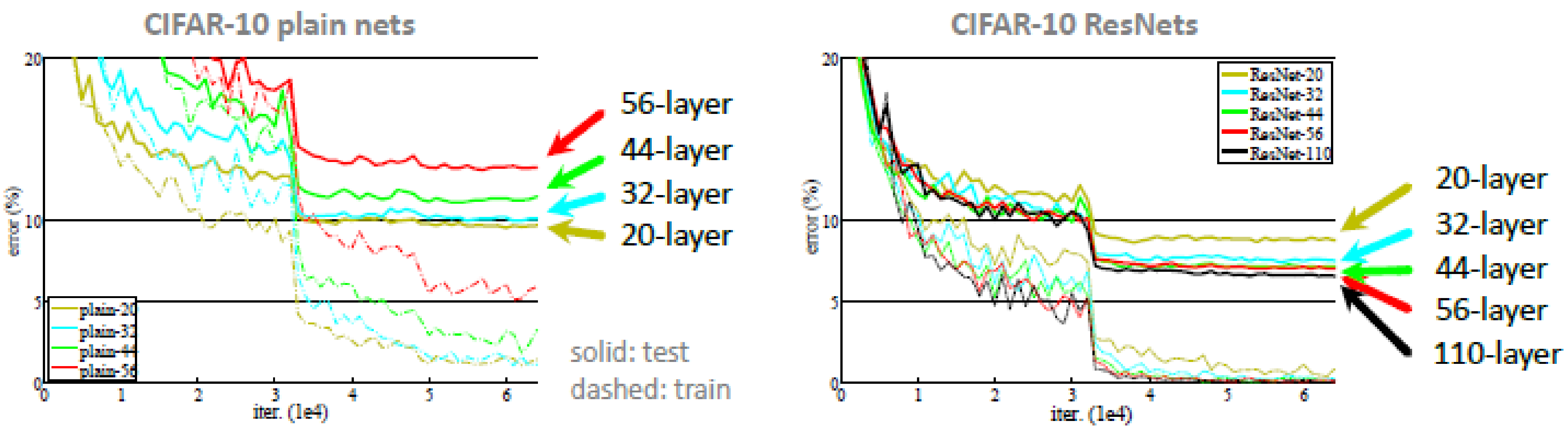
- Other remarks:no max pooling (almost)
  - -no hidden fc
  - -no dropout



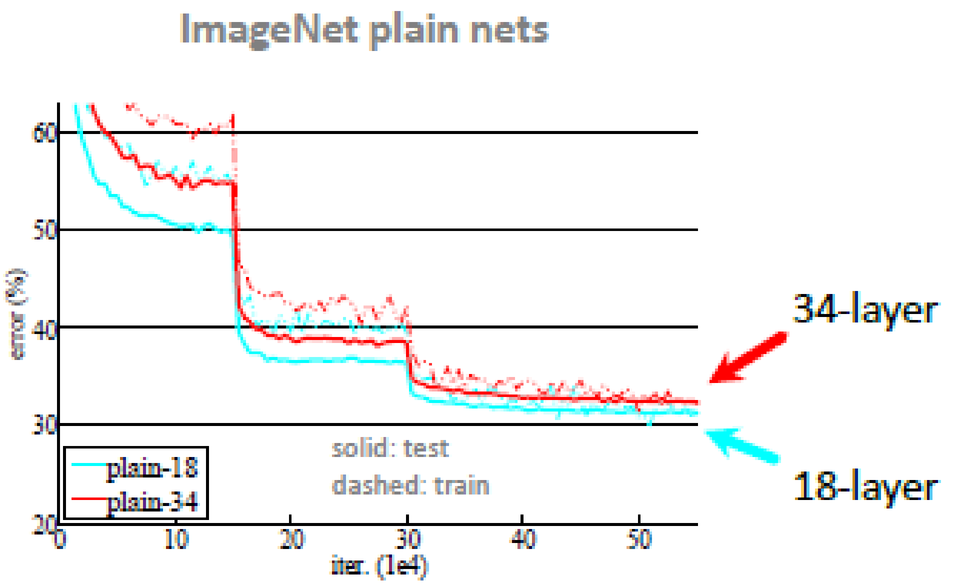
Training

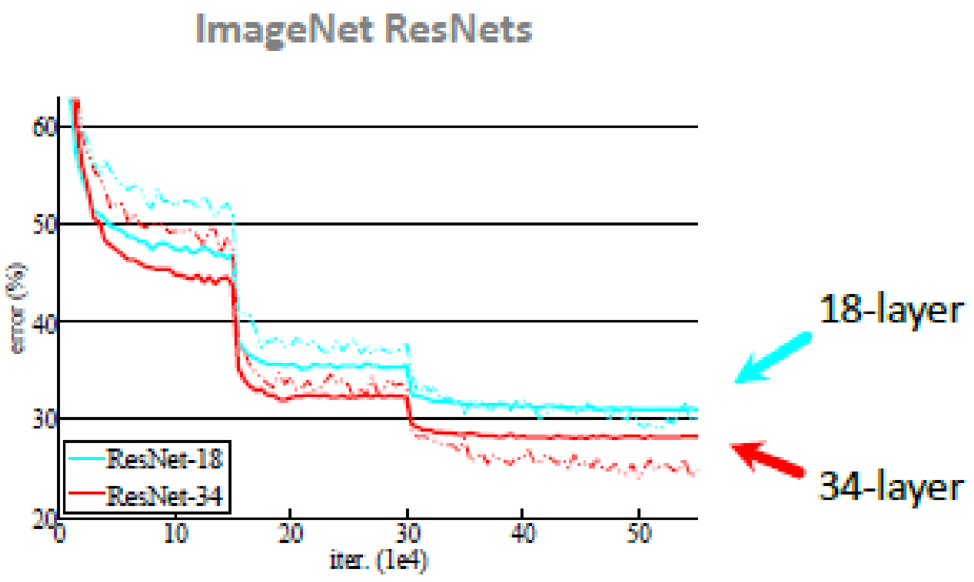
- All plain/residual nets are trained from scratch
- All plain/residual nets use Batch Normalization
- Standard hyper-parameters & augmentation

CIFAR-10 experiments



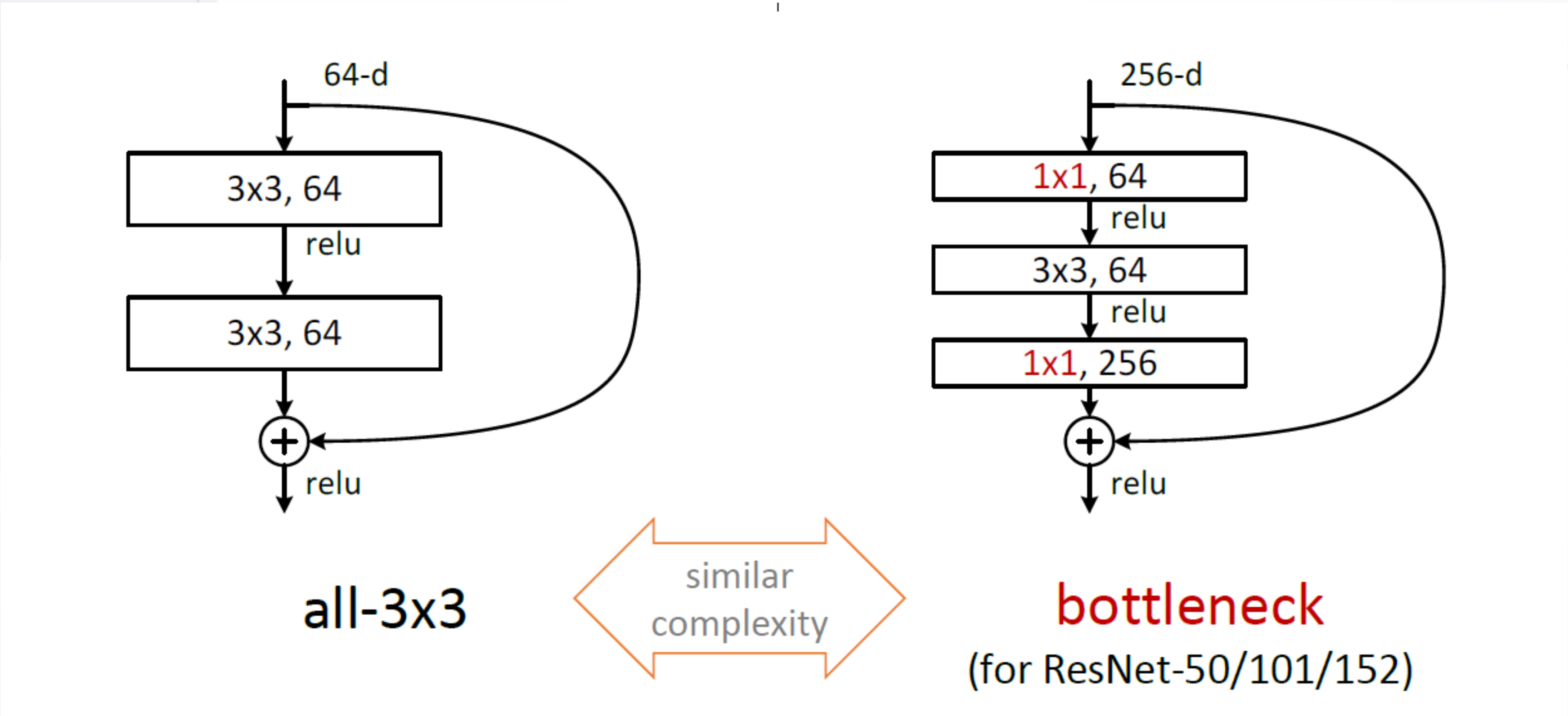
ImageNet experiments



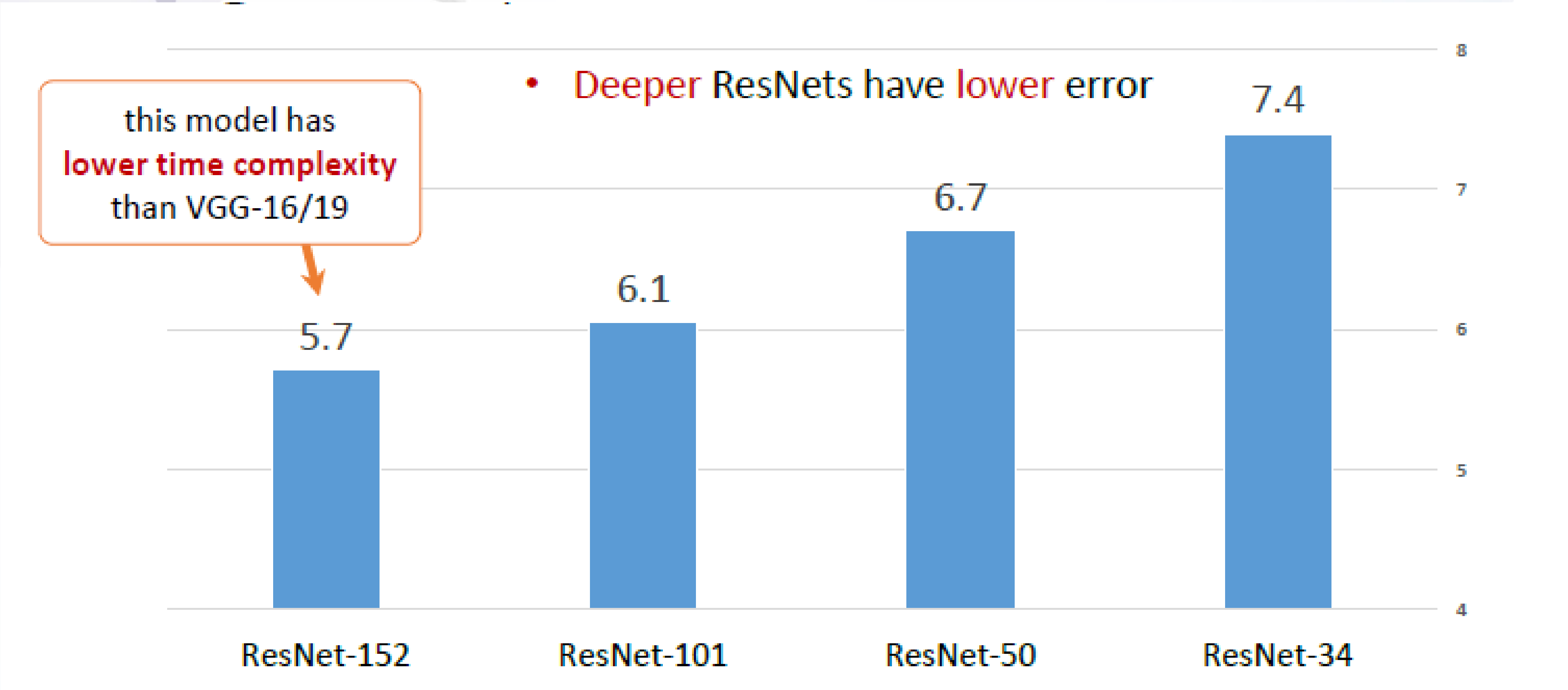


- Deep ResNets can be trained without difficulties
- Deeper ResNets have lower training error, and also lower test error

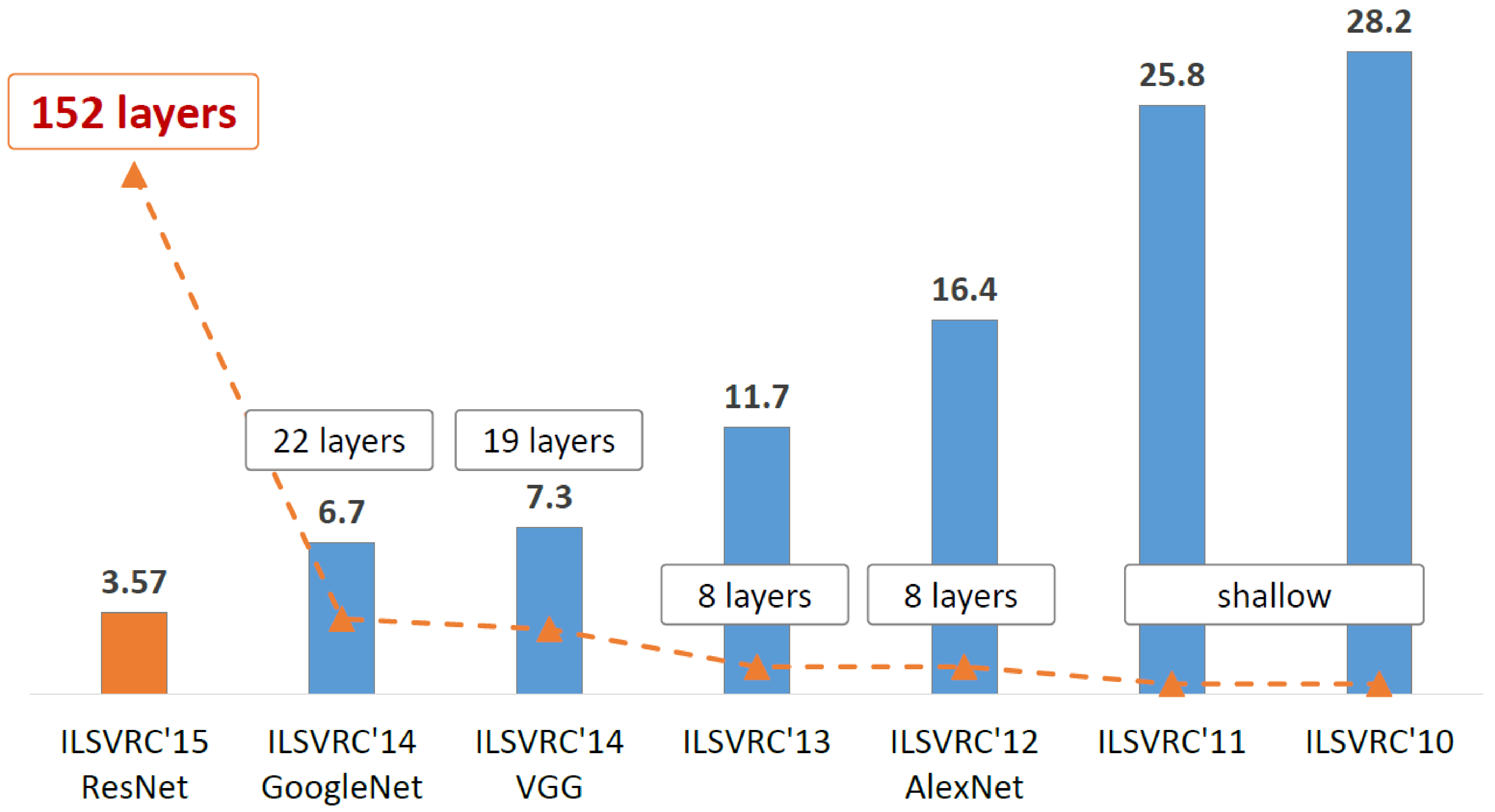
A practical design of going deeper



ImageNet experiments



ImageNet experiments



Why residual works?

The main reason the residual network works is that it's so easy for these extra layers to learn the identity function that you're kind of guaranteed that it doesn't hurt performance. And then lot of time you maybe get lucky and even helps performance, or at least is easier to go from a decent baseline of not hurting performance, and then creating the same can only improve the solution from there.

$$z^{[l+1]}=W^{[l+1]}a^{[l]}+b^{[l+1]}$$

$$a^{[l+1]}=g(z^{[l+1]})$$

$$z^{[l+2]}=W^{[l+2]}a^{[l+1]}+b^{[l+2]}$$

$$a^{[l+2]}=g(z^{[l+2]}) \Longrightarrow a^{[l+2]}=g(W^{[l+2]}a^{[l+1]}+b^{[l+2]}+a^{[l]})$$

Personal experiment

```
Test - > Loss: 0.414 | Acc: 86.237% (6554/7600)
Test - > Loss: 0.415 | Acc: 86.260% (6642/7700)
Test - > Loss: 0.417 | Acc: 86.154% (6720/7800)
Test - > Loss: 0.416 | Acc: 86.152% (6806/7900)
Test - > Loss: 0.418 | Acc: 86.112% (6889/8000)
Test - > Loss: 0.416 | Acc: 86.148% (6978/8100)
Test - > Loss: 0.417 | Acc: 86.146% (7064/8200)
Test - > Loss: 0.418 | Acc: 86.145% (7150/8300)
Test - > Loss: 0.417 | Acc: 86.179% (7239/8400)
Test - > Loss: 0.418 | Acc: 86.129% (7321/8500)
Test - > Loss: 0.419 | Acc: 86.093% (7404/8600)
Test - > Loss: 0.419 | Acc: 86.023% (7484/8700)
Test - > Loss: 0.420 | Acc: 86.000% (7568/8800)
Test - > Loss: 0.419 | Acc: 85.989% (7653/8900)
Test - > Loss: 0.419 | Acc: 85.956% (7736/9000)
Test - > Loss: 0.418 | Acc: 85.967% (7823/9100)
Test - > Loss: 0.416 | Acc: 86.022% (7914/9200)
Test - > Loss: 0.416 | Acc: 86.032% (8001/9300)
Test - > Loss: 0.415 | Acc: 85.979% (8082/9400)
Test - > Loss: 0.414 | Acc: 85.979% (8168/9500)
Test - > Loss: 0.414 | Acc: 85.990% (8255/9600)
Test - > Loss: 0.414 | Acc: 85.979% (8340/9700)
Test - > Loss: 0.415 | Acc: 85.959% (8424/9800)
Test - > Loss: 0.416 | Acc: 85.949% (8509/9900)
 [============================================
Test - > Loss: 0.416 | Acc: 85.940% (8594/10000)
```

Epoches:5000
ResNet-18
Simple preprocess

Futher study

http://kaiminghe.com/icml16tutorial/index.html

Publications:

- [a] *Deep Residual Learning for Image Recognition*
  Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. CVPR 2016.
- [b] *Identity Mappings in Deep Residual Networks*
  Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Technical report, arXiv 2016.

Resources:

- tutorial slides
- slides and video for the talk at ICCV 2015 ImageNet and COCO joint workshop.
- code/models of 50, 101, and 152-layer ResNets pre-trained on ImageNet.
- code of 1001-layer ResNet on CIFAR.
- list of third-party ResNet implementations on ImageNet, CIFAR, MNIST, etc.

Thanks