CS688 Student Presentation

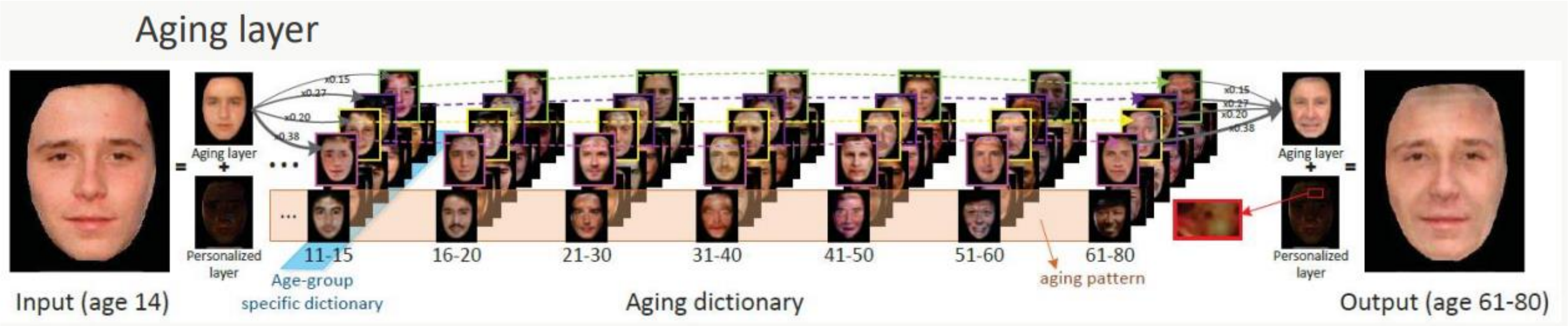
Deep Residual Learning for Image Recognition (CVPR16)

18.11.01

Youngbo Shim

Review: Personalized Age Progression with Aging Dictionary

- Speaker: Hyunyul Cho
- Problem
  - Prev. works of age progression didn't considered personalized facial characteristics
  - Prev. works required dense long-term face aging sequences
- Idea
  - Build two layers (aging/personalized) to retain personal characteristics
  - Construct an aging dictionary



- From Hyunyul Cho's presentation slides

CS688 Student Presentation

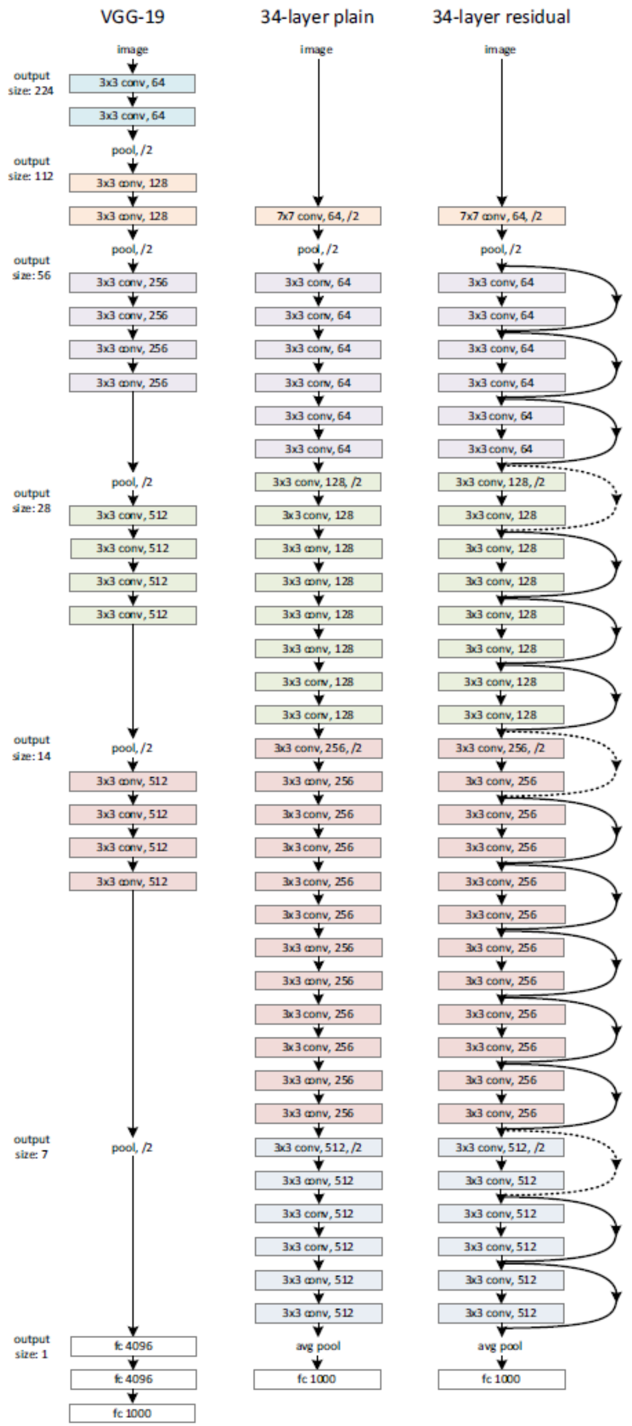
Deep Residual Learning for Image Recognition (CVPR16)

18.11.01

Youngbo Shim

Brief introduction

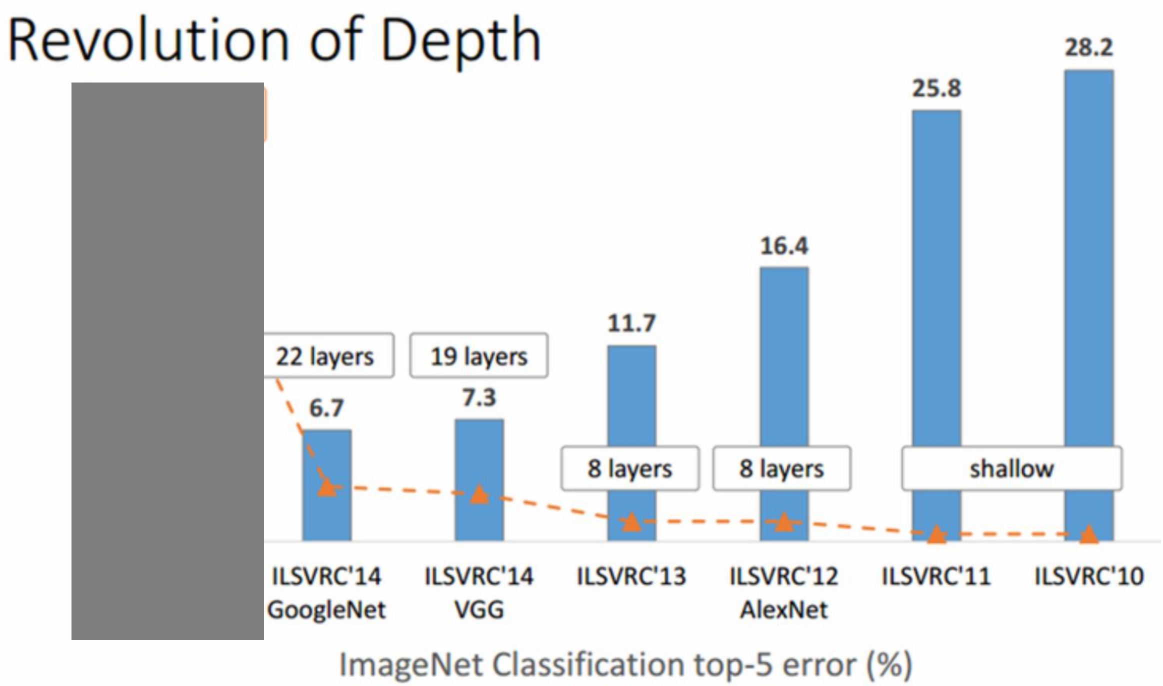
- One of the best CNN architecture
- Exploited over a wide area
  - Image classification (ILSVRC'15 classification 1st place)
  - Image detection (ILSVRC'15 detection 1st place)
  - Localization (ILSVRC'15 localization 1st place)



- He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Motivation

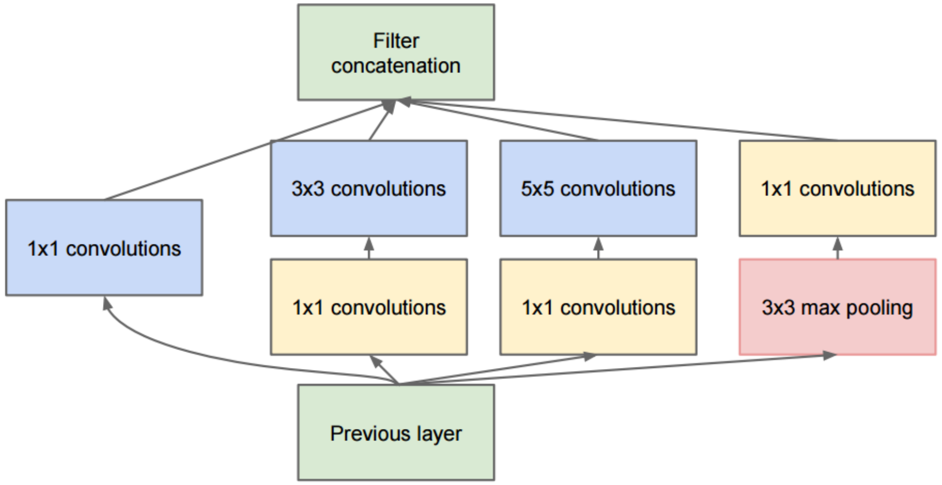
- At the moment (~2015)



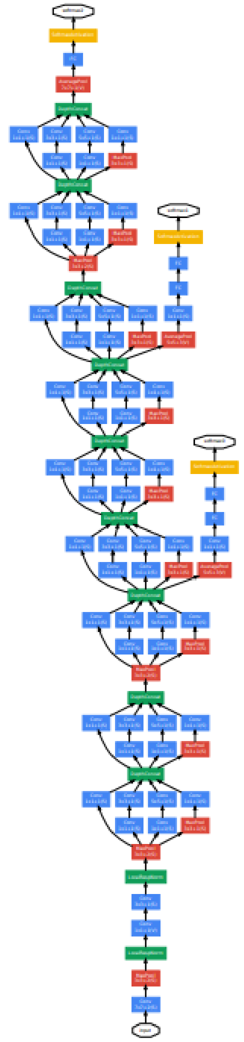
- From Kaiming He slides "Deep residual learning for image recognition." ICML. 2016.

Related work

- GoogLeNet (2015)
  - Inception module
    - Reduced parameters and FLOPs by dimension reduction
  - auxiliary classifier
    - Avoid vanishing gradient problem



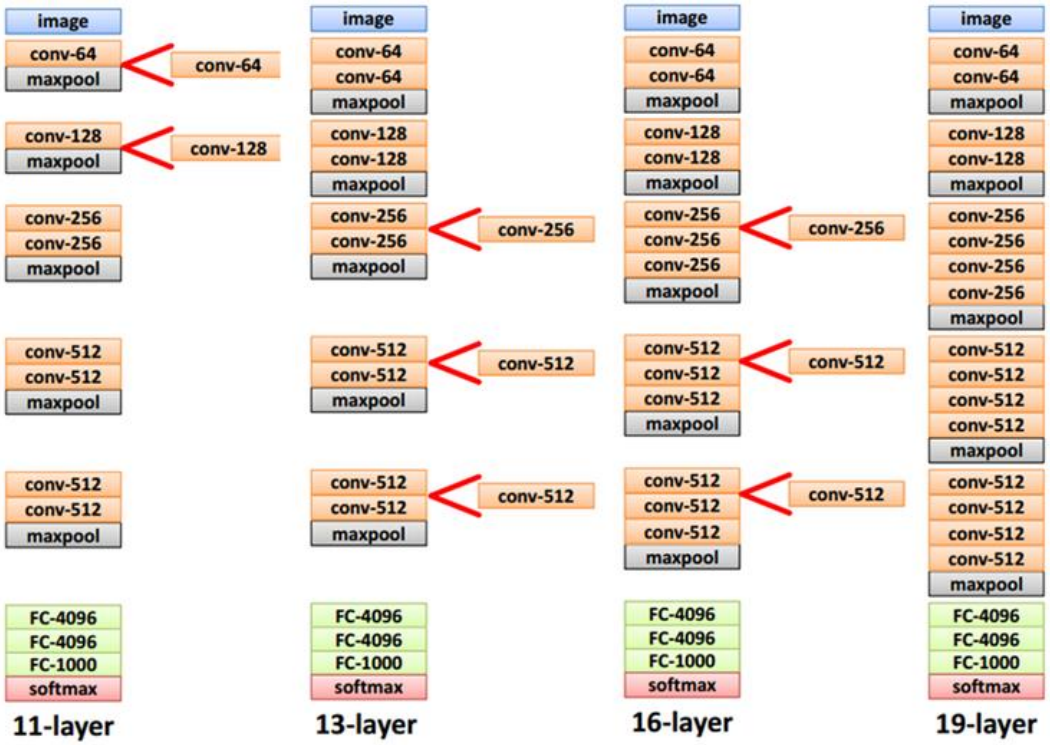
Inception module



- Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Related work

- VGG (2015)
  - Explored the ability of network depth
  - 3×3 Convolution kernels

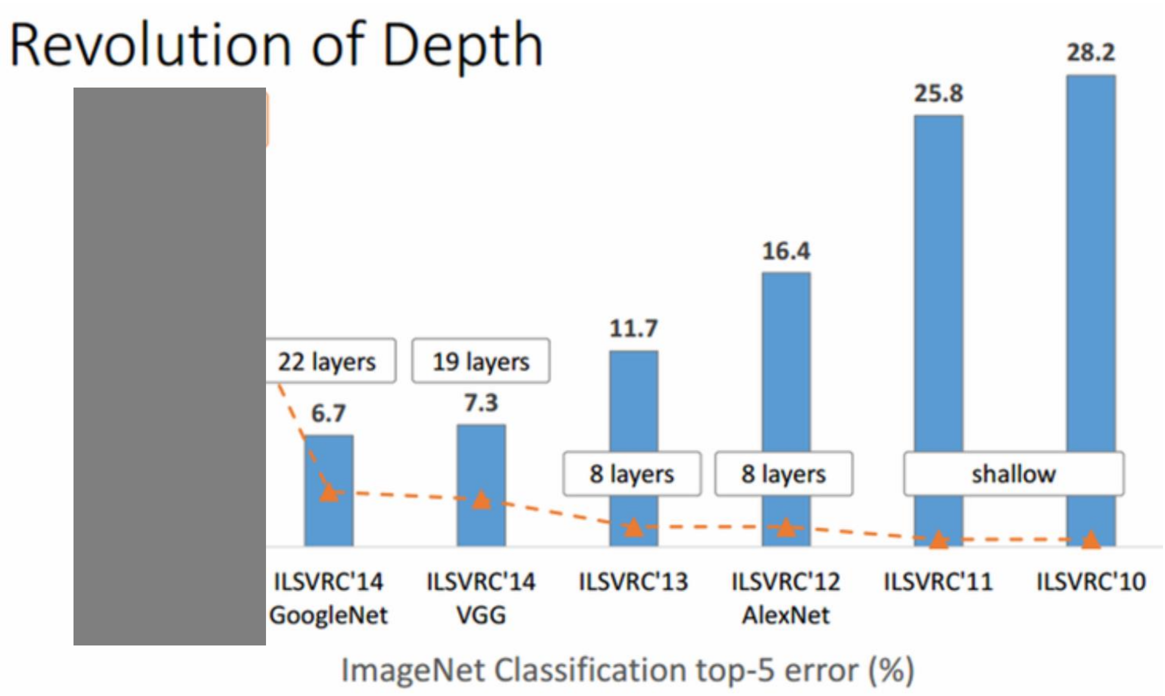


VGG networks

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015., from https://laonple.blog.me/220749876381

Motivation

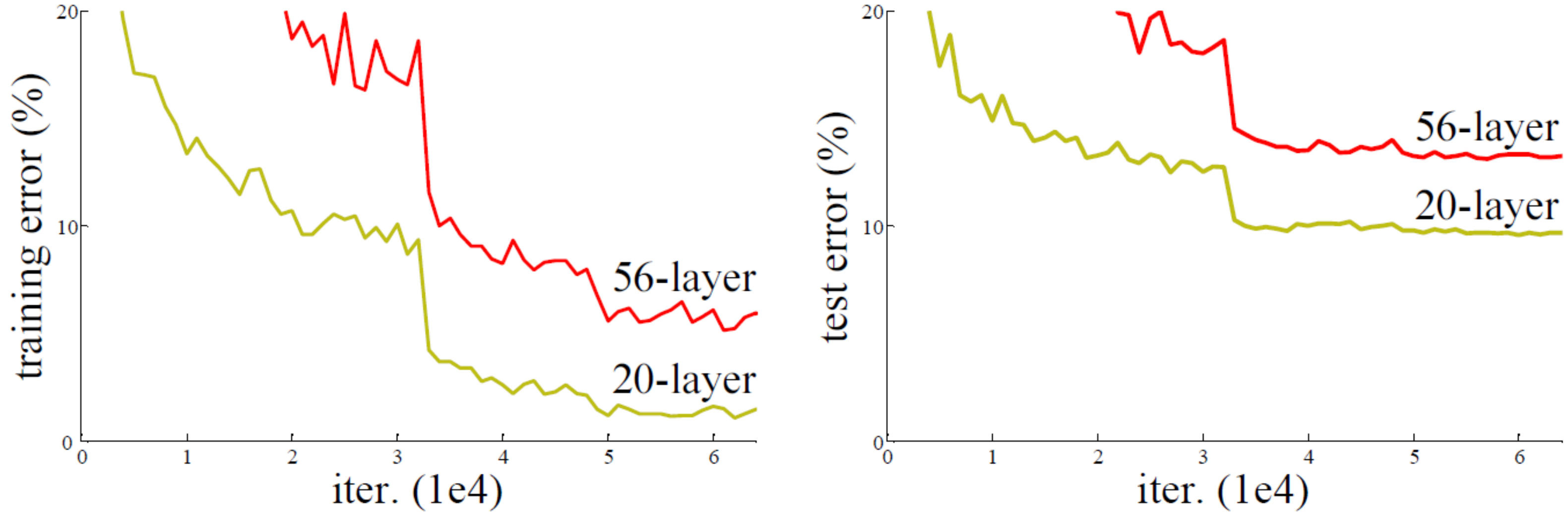
- At the moment (~2015)
- Could we dig deeper?



- From Kaiming He slides "Deep residual learning for image recognition." ICML. 2016.

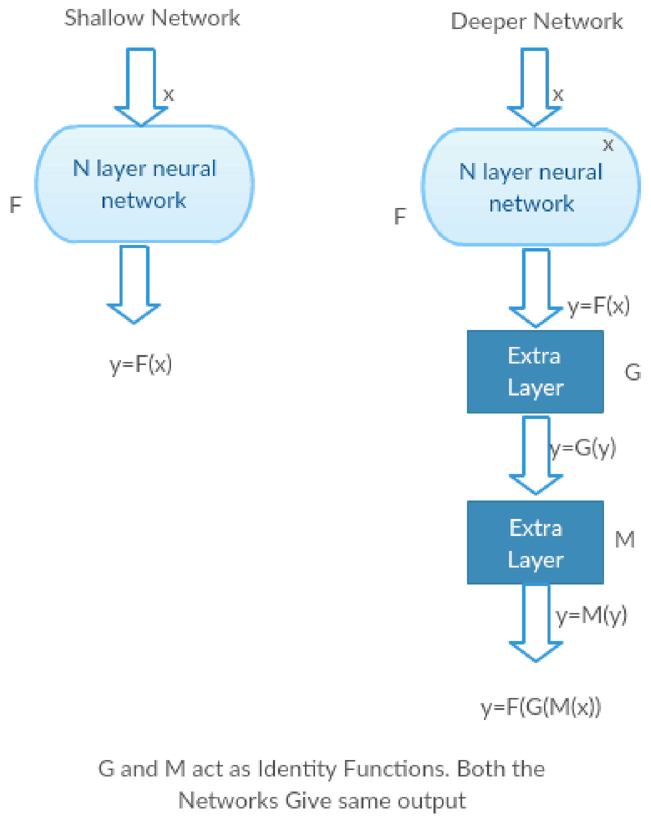
Motivation

- Degradation problem
  - Not caused by overfitting
  - Hard to optimize due to large parameter set



- From Kaiming He slides "Deep residual learning for image recognition." ICML. 2016.

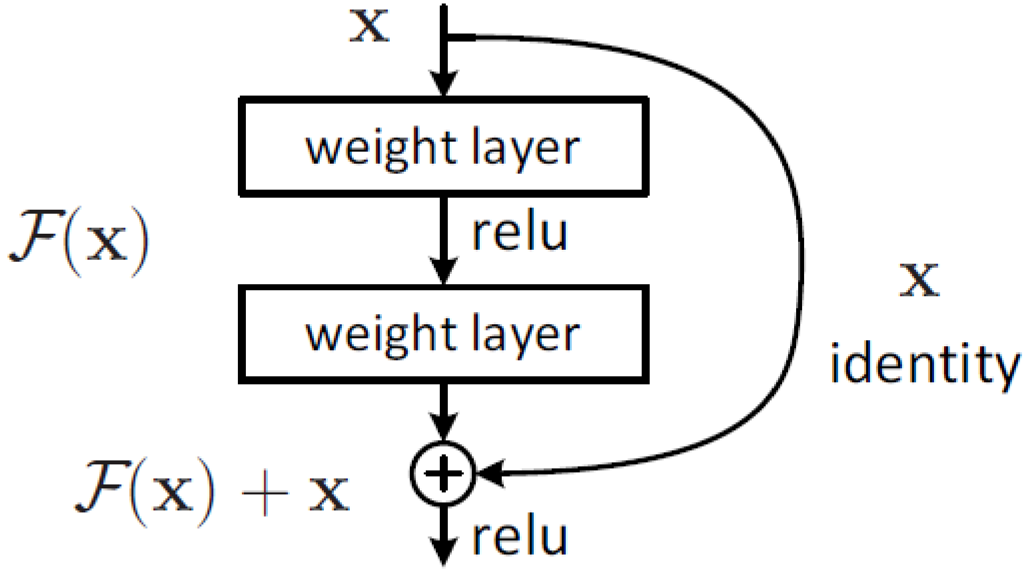
Idea



- Deep network should work well at least as shallow one does.
- If extra layers' are identity mappings.

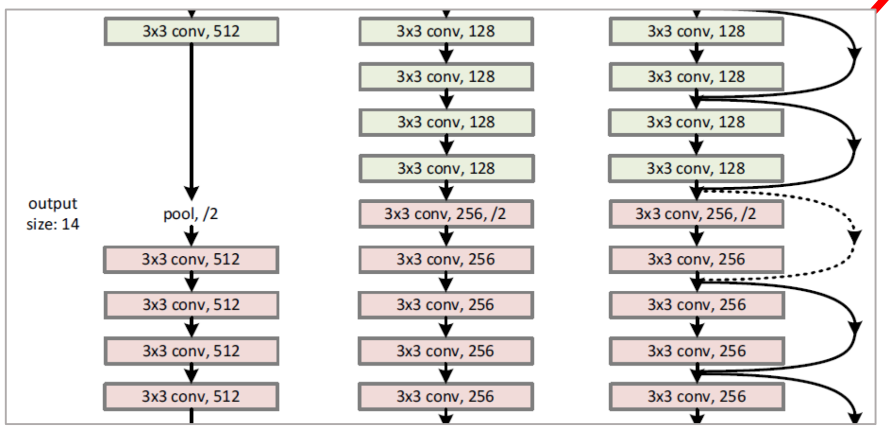
Idea

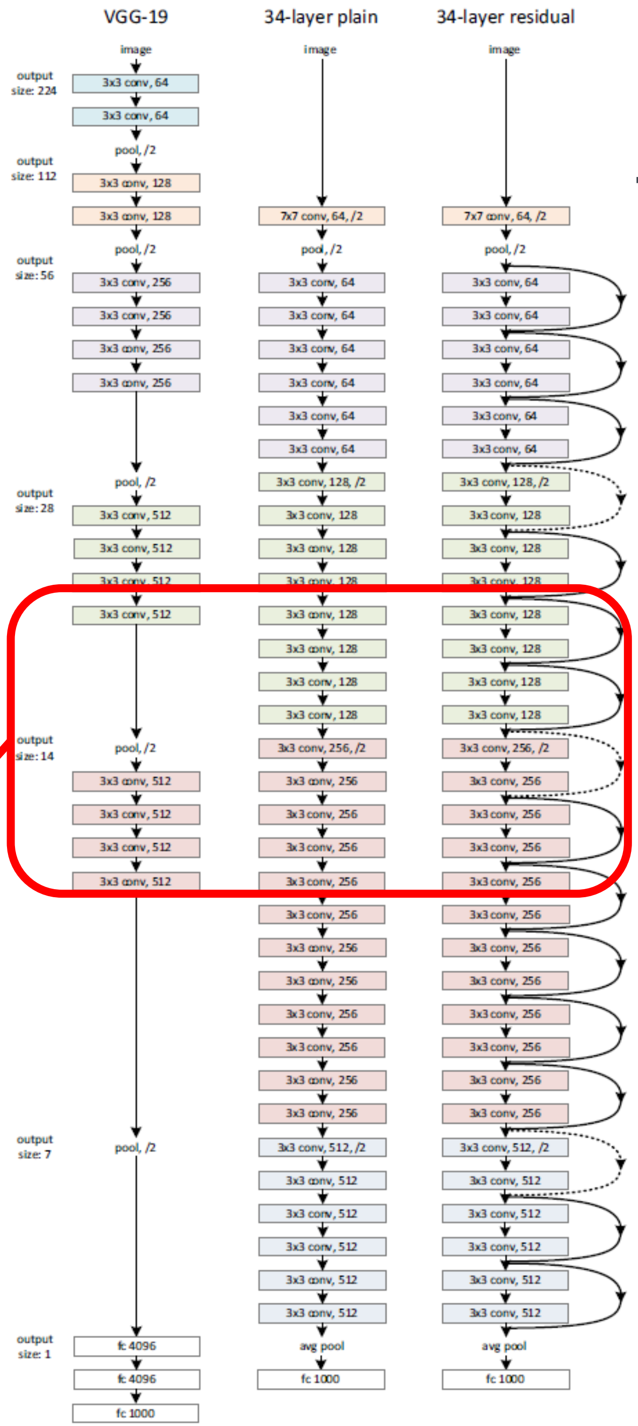
- Residual Learning
  - Shortcut connections with identity mapping reference
  - $F(x) \coloneqq H(x) - x$ (Residual function)
  - If identity mapping is optimal for the case, $F(x)$'s weight will converge to zero.



Network Architecture

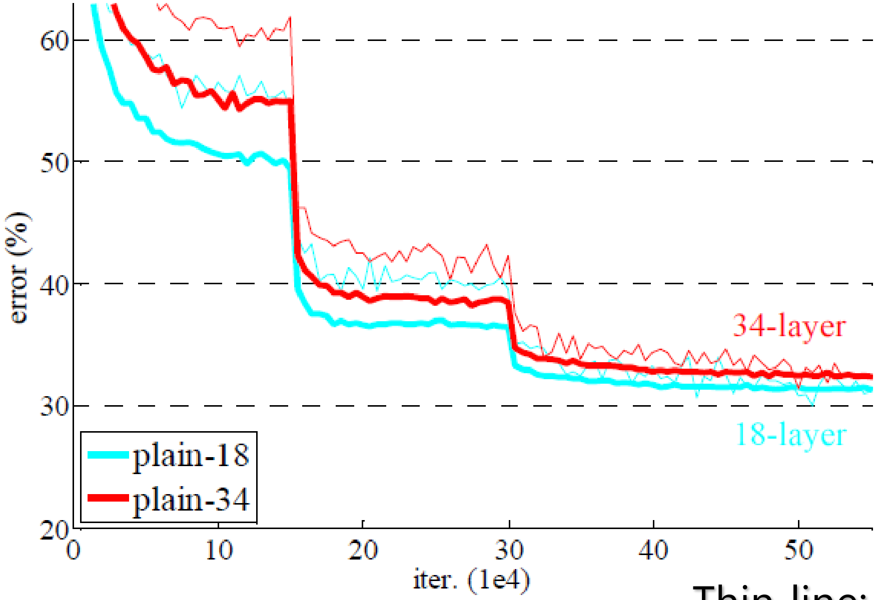
- Exemplar model in comparison with VGG
- Stride, instead of pooling
- Zero padding/Projection to match dimensions

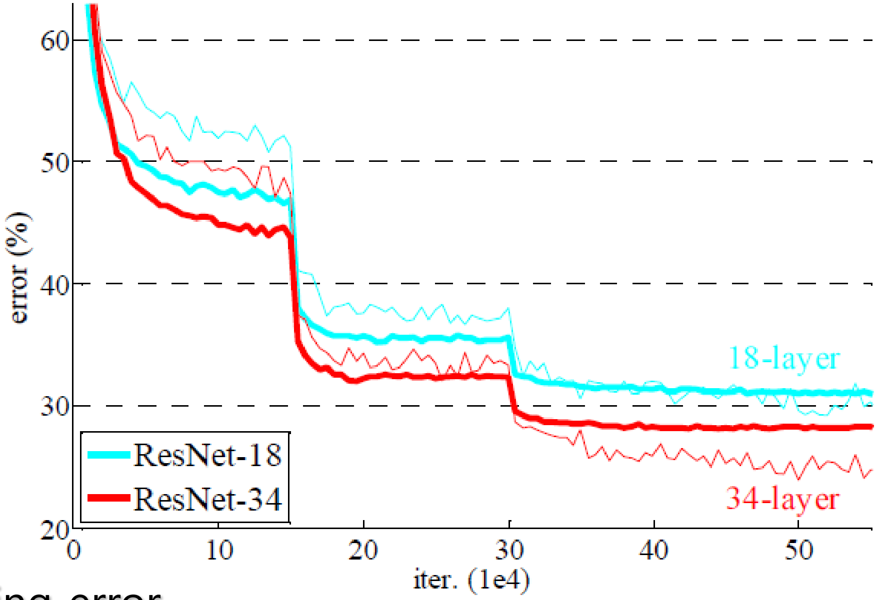




Experiment 1: ImageNet classification

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



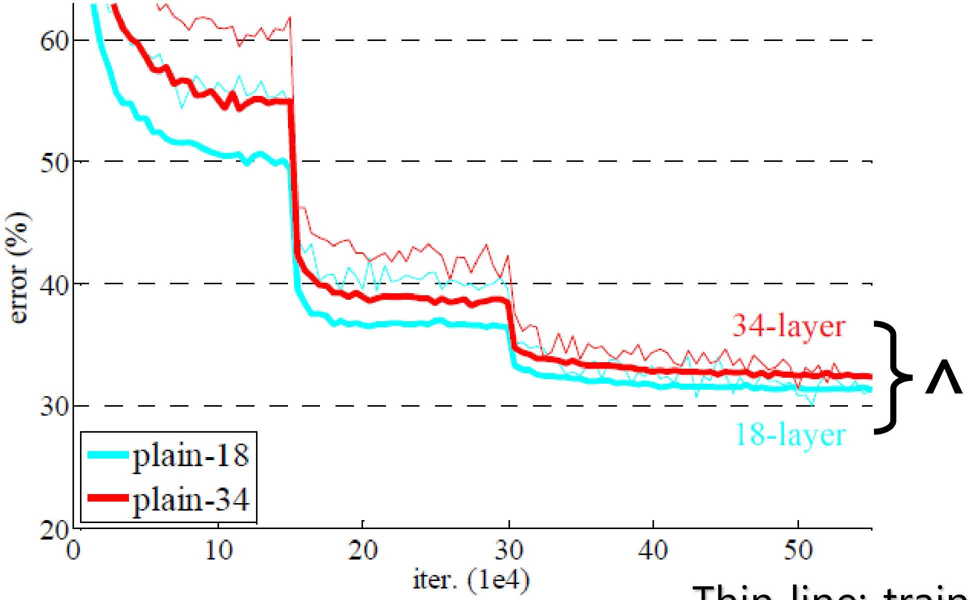


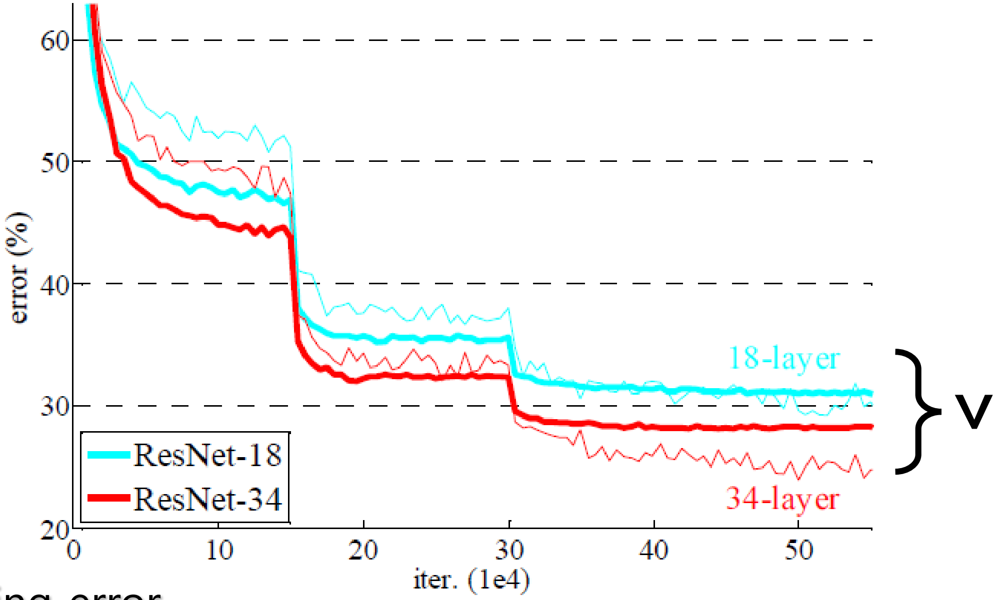
Thin line: training error
Bold line: validation error

Experiment 1: Findings

- plain-18 is better than plain-34
  - degradation
- ResNet-34 is better than ResNet-18
  - Deeper, better!

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



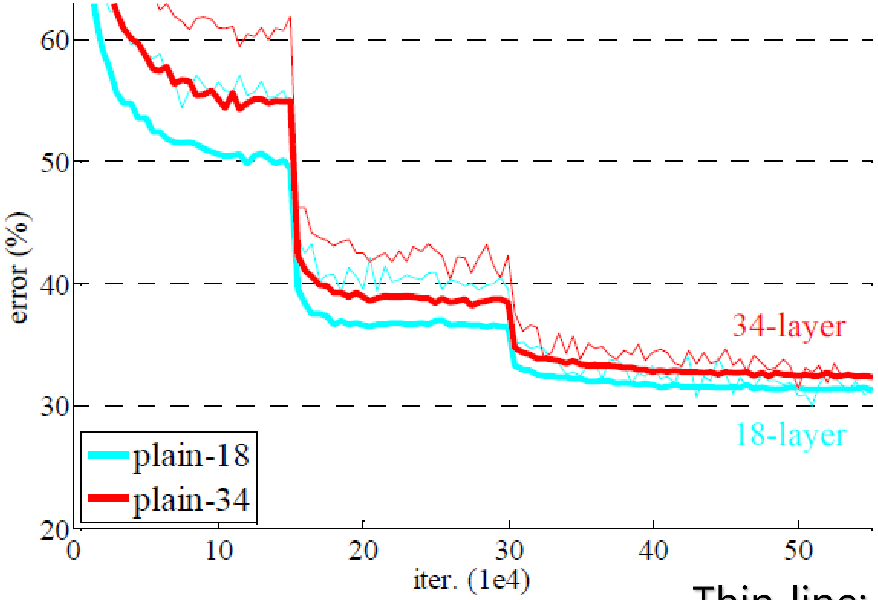


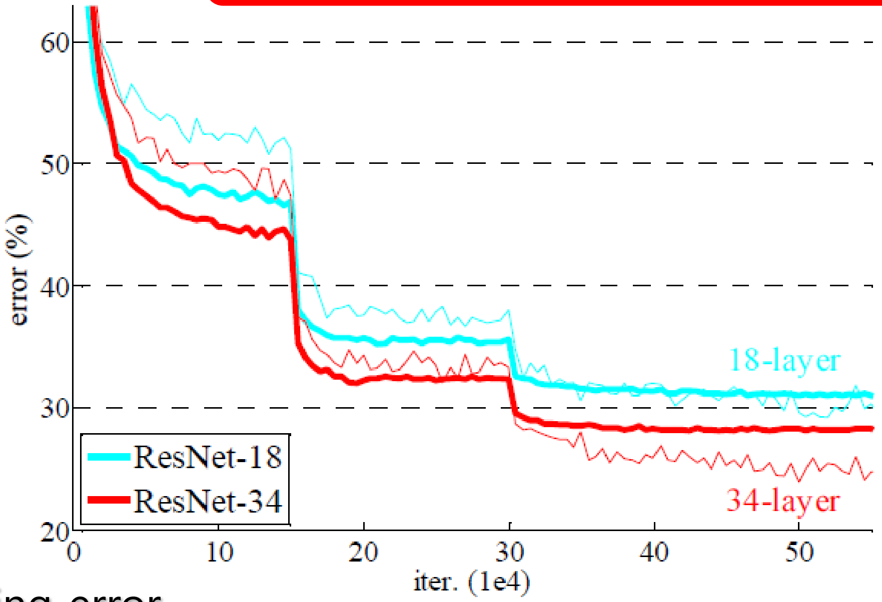
Thin line: training error
Bold line: validation error

Experiment 1: Findings

- ResNet-34 successfully reduces error compared to its counterpart (plain-34)

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



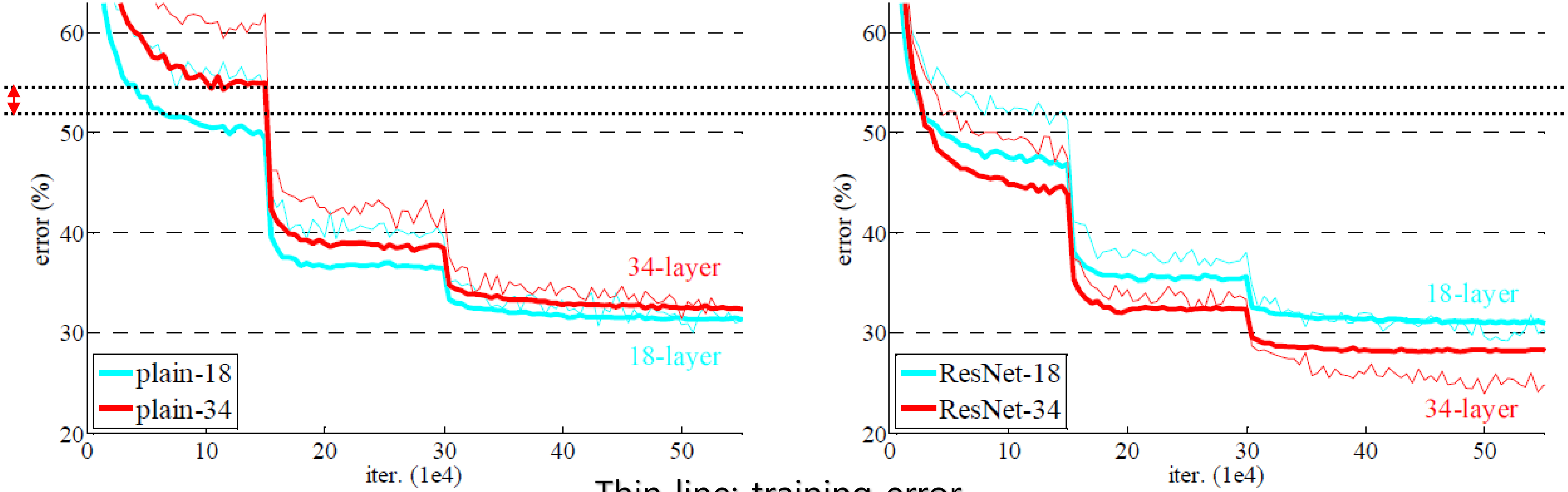


Thin line: training error
Bold line: validation error

Experiment 1: Findings

- ResNet shows faster convergence at the early stage

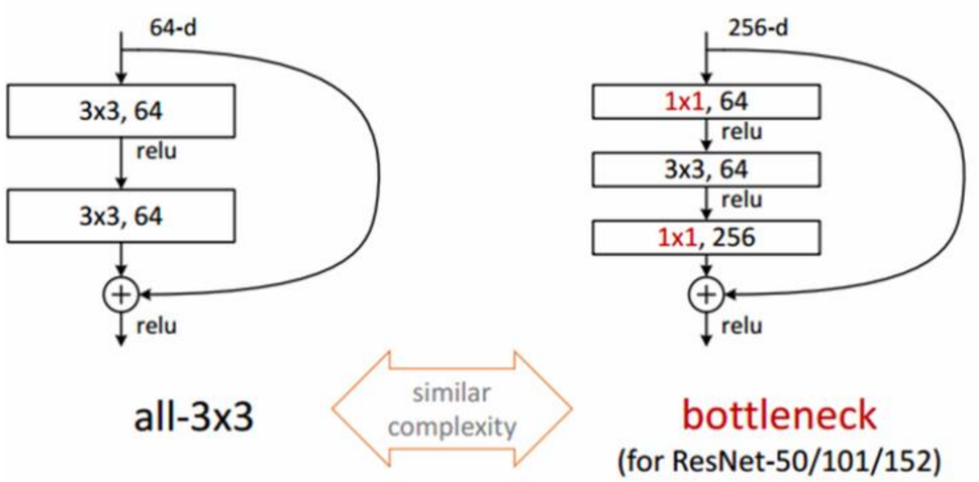
	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

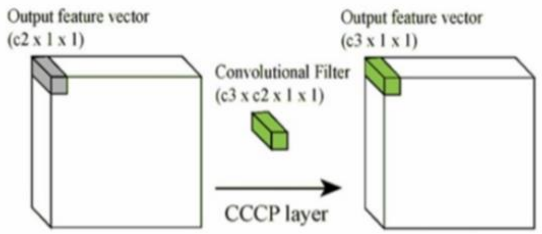


Thin line: training error
Bold line: validation error

Idea

- How could we dive deeper?
  - Practical problem: # of parameters & calculations ∝ training time
- Deeper Bottleneck Architecture



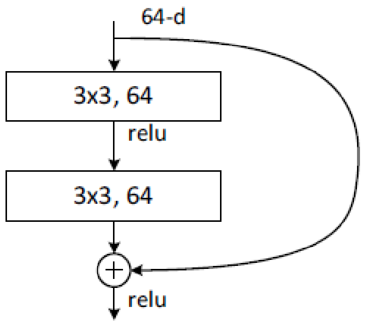


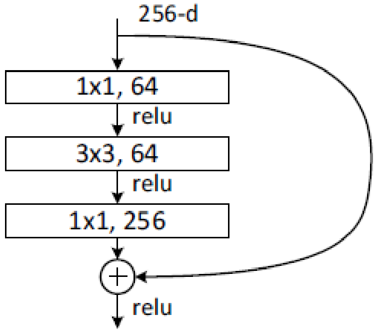
  - 1×1 convolution layer reduces the dimension
  - Similar to GoogLeNet's inception module

- From https://laonple.blog.me/220692793375

Experiment 2: Deeper Imagenet classification

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9





Experiment 2: Result

- Better than state-of-the-art methods
- Still(!) deeper, better
- Low complexity
  - ResNet-152 (11.3b FLOPs) < VGG-16/19 (15.3/19.6b FLOPs)

method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	$8.43^{\dagger}$
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

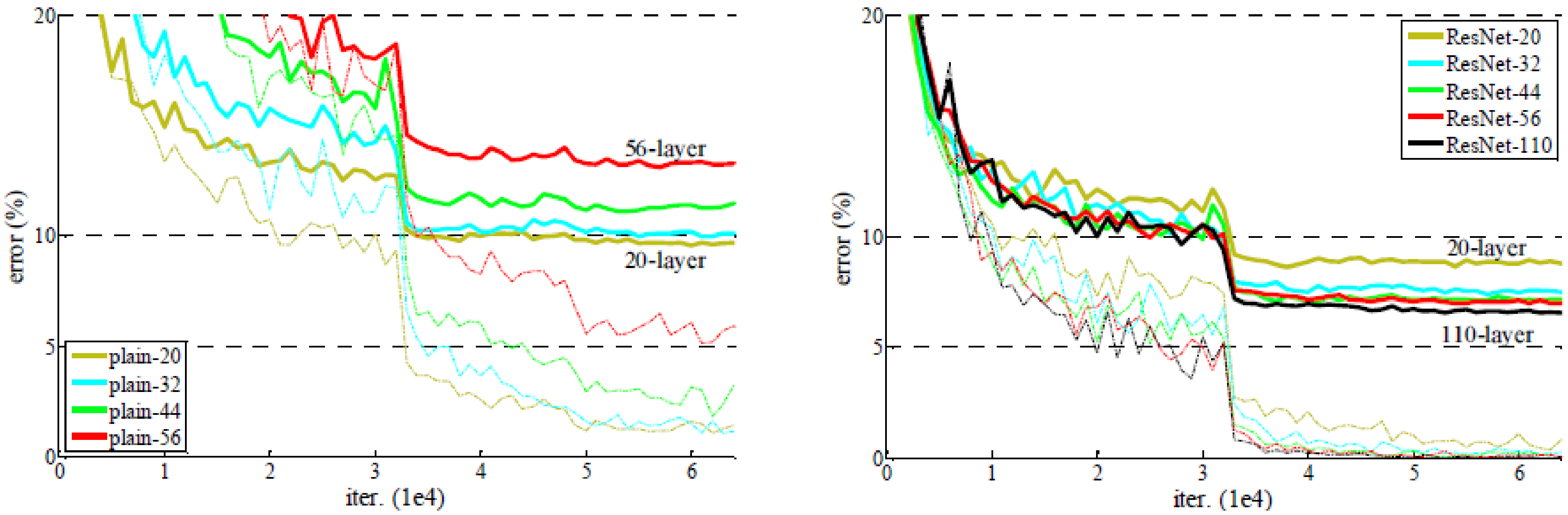
Experiment 3: CIFAR-10 classification

- CIFAR-10 has relatively small input of 32×32
  - Could test extremely deep network (depth: 1202)
- Observe the behavior of networks in relation with depth

method			error (%)
Maxout [9]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [34]	19	2.5M	8.39
Highway [41, 42]	19	2.3M	7.54 (7.72±0.16)
Highway [41, 42]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

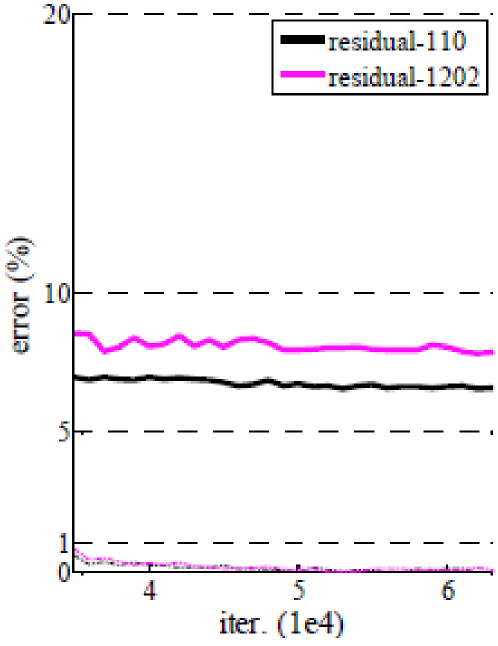
Experiment 3: Result

- Deeper, better until 110 layers…



Experiment 3: Result

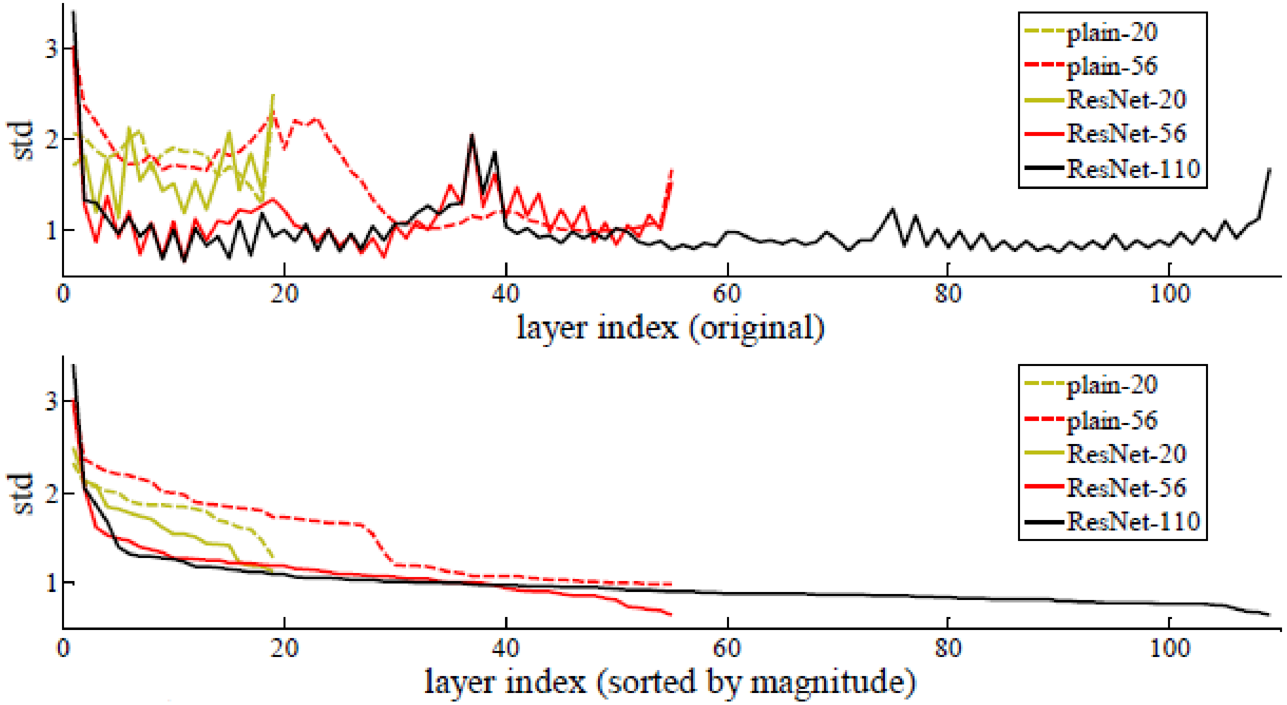
- Deeper, better until 110 layers…
- Not in 1202 layers anymore
  - Both 110 & 1202 optimizes well (training error converges to <0.1%)
  - Overfitting occurs (higher validation error rate)



Dotted line: training error
Bold line: validation error

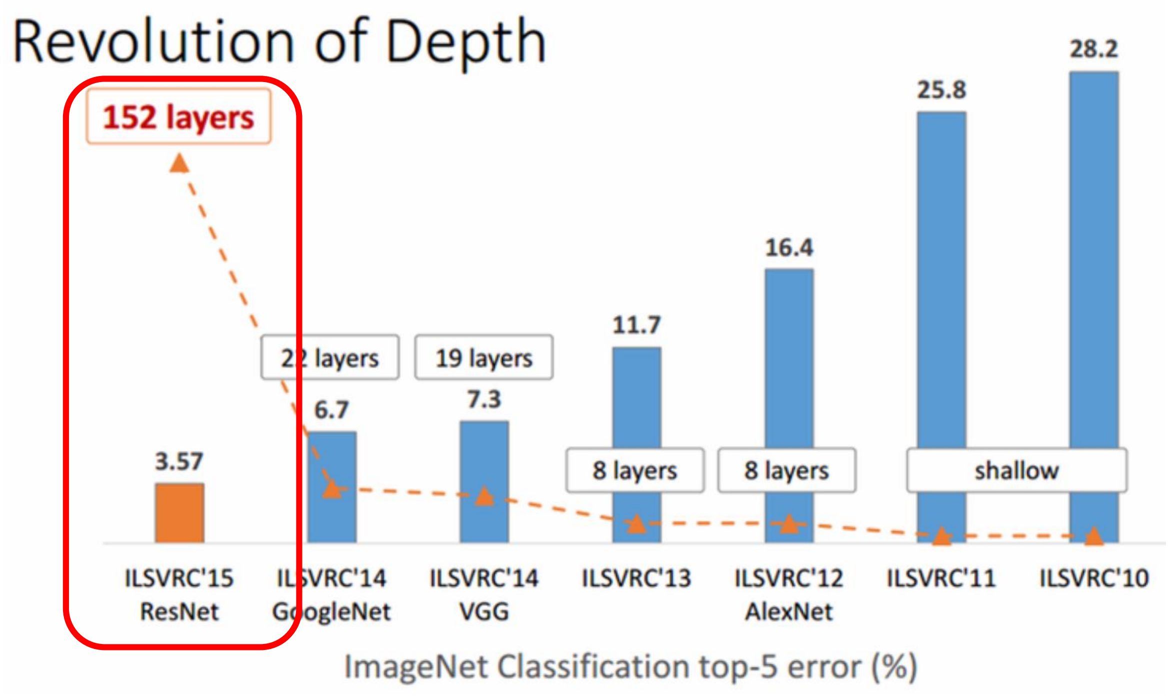
Experiment 3: Result

- Standard deviation of layer responses
- Small responses than their counterparts (plain networks)
    - Residual functions are closer to zero
- Deeper = smaller response



Wrap-up

- ResNet
- Stable layer stacking by residual learning
- Empirical data to show performance and depth's influence



- From Kaiming He slides "Deep residual learning for image recognition." ICML. 2016.

Wrap-up

- ResNet
- Stable layer stacking by residual learning
- Empirical data to show performance and depth's influence

Thank you for listening

- From Kaiming He slides "Deep residual learning for image recognition." ICML. 2016.

Quiz

- Q1. What was the problem of deep CNNs before ResNet?
  1. Degradation problem
  2. Identity mapping
  3. Overfitting

- Q2. What is the name of architecture of ResNet to reduce training time?
  1. Inception module
  2. Deeper bottleneck architecture
  3. Multi-layer perceptron