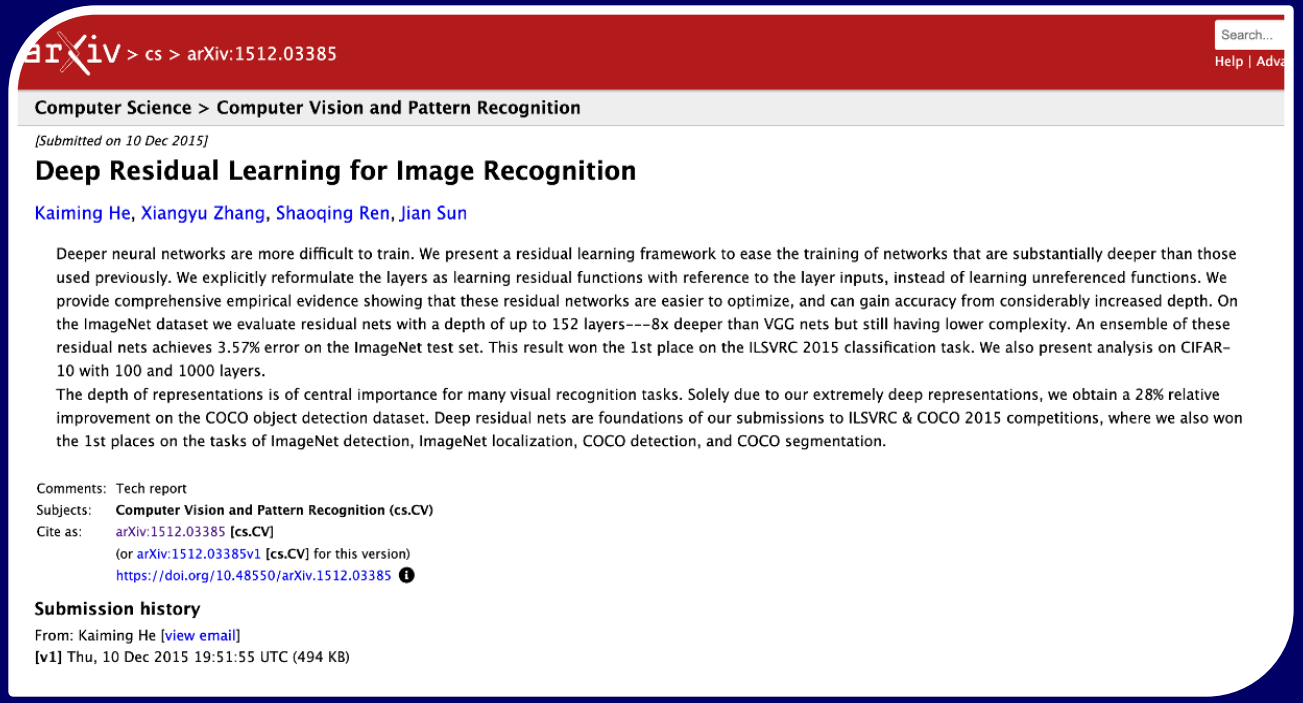
2024.07.26



arXiv > cs > arXiv:1512.03385

Search... Help | Adva

Computer Science > Computer Vision and Pattern Recognition

[Submitted on 10 Dec 2015]

Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers---8x deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.

The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

Comments: Tech report
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: arXiv:1512.03385 **[cs.CV]**
(or arXiv:1512.03385v1 **[cs.CV]** for this version)
https://doi.org/10.48550/arXiv.1512.03385

Submission history

From: Kaiming He [view email]
[v1] Thu, 10 Dec 2015 19:51:55 UTC (494 KB)

ResNet; Deep Residual Learning; Shortcut Connection

Yeon Su Park

Pukyong National University
Division of Computer Engineering And Artificial Intelligence
Medical AI Lab.

Introduction

Problem of exisiting model

- By increasing importance of network depth, people nake and use “very deep” models(13-16 layers) on visual recognition tasks.

- Is learning better networks as easy as stacking more layers? : No, because..
  - There is vanishing/exploding gradients problem which hamper convergence from the beginning.
  - It can be addressed by normalized initialization and intermediate normalization layers, which enable networks with tens of layers to start converging for SGD with backpropagation.
  - When deeper networks are able to start converging, a **degradation** problem has been exposed: with the network depth increasing, accuracy gets saturated and then degrades rapidly. degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error.

Related work

Residual Representations and Shortcut connections

- **Residual Representations**
  - Image recognition : VLAD, Fisher Vector(probabilistic version of VLAD), vector quantization, etc …
  - Low-level vision, computer graphics : Multigrid, hierarchical basis preconditioning, etc …

- **Shortcut connections**
  - MLP, inception, highway networks, etc …

Deep Residual Learning

Residual Learning

- H(x) = **underlying mapping** to be fit by a few stacked layers
- x = **the inputs** to the first of these layers.
- Hypothesize = that multiple nonlinear layers can asymptotically approximate complicated functions, then it is equivalent to hypothesize that they can asymptotically approximate the residual functions, i.e., H(x) − x
- F(x) := H(x)−x
- F(x) +x := H(x)

Formula

$$r = x - x_0$$

r = residual

x = measure variable

x_0 = approximate variable

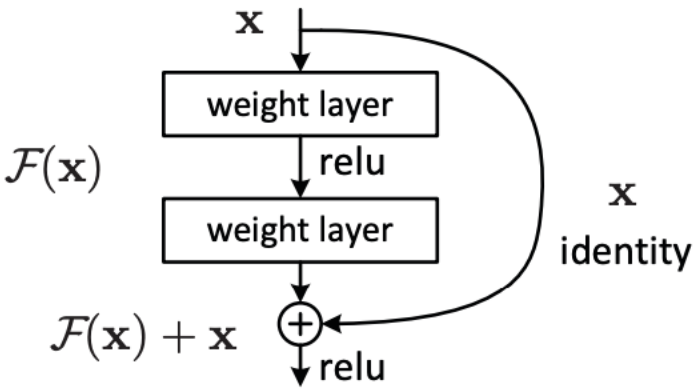


Figure 2. Residual learning: a building block.

Deep Residual Learning

Identity mapping by shortcuts

- Block : few stacked layer which adopt residual learning.
- $Y = F(x, \{W_i\}) + x$ Eqn.(1)
- $F = W_2\sigma(W_1x)$ multiple convolutional layers
- σ = ReLU

- 1. F + x = element-wise addition by shortcut connection
- 2. $\sigma(y)$

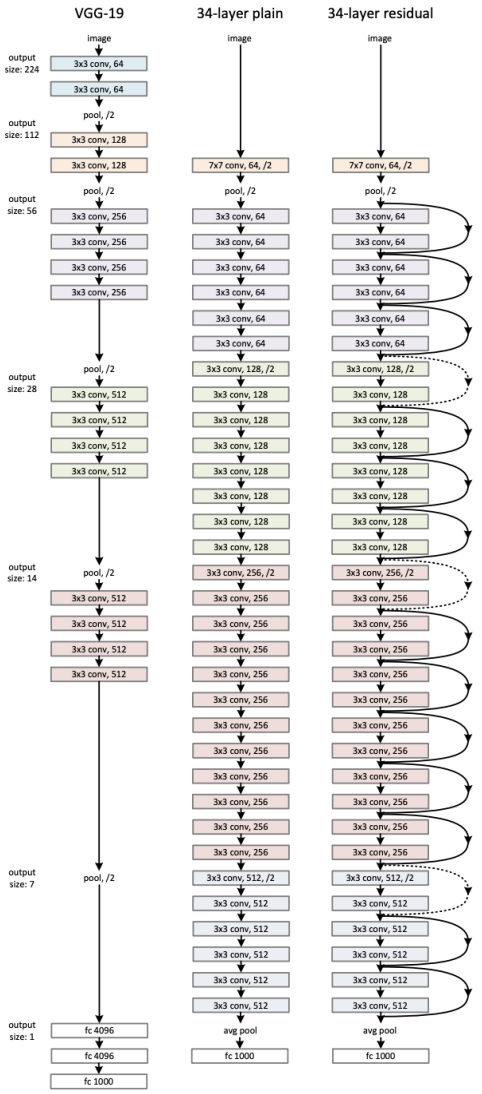
- The dimensions of x and F must be equal. If this is not the case (e.g., when changing the input/output channels), we can perform a linear projection W_s by the shortcut connections to match the dimensions.
- $y = F(x, \{W_i\}) + W_sx$ Eqn.(2)

- We can also use a square matrix W_s in Eqn.(1). But according to experiments that the identity mapping is sufficient for addressing the degradation problem and is economical, and thus Ws is only used when matching dimensions.

Deep Residual Learning

Network architecture

- ImageNet
  - image resized : with its shorter side randomly sampled scale augmentation
  - crop : 224×224
  - weights : scratch
  - SGD with a mini-batch size of 256
  - learning rate : 0.1 and is divided by 10 when the error plateaus, and the models are trained for up to 60 × 104 iterations.
  - weight decay : 0.0001 and a momentum of 0.9.
  - random horizontal flip : True
  - The standard color augmentation : True
  - batch normalization : True
  - dropout : False
- Testing
  - we adopt the standard 10-crop testing.
  - fullyconvolutional form
  - average the scores at multiple scales
  - images resized : the shorter side {224, 256, 384, 480, 640}.



Experiments

Image classification

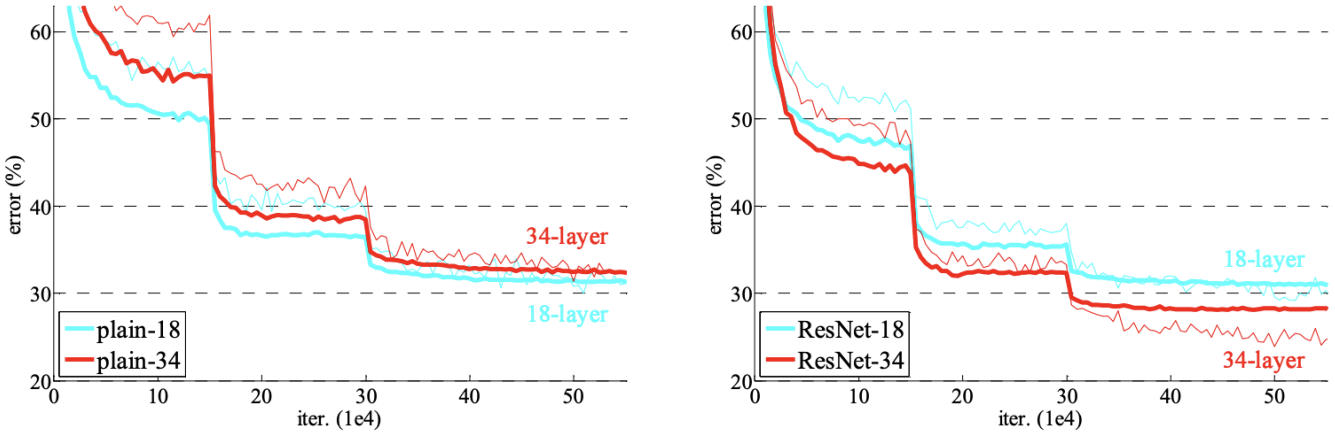


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

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

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

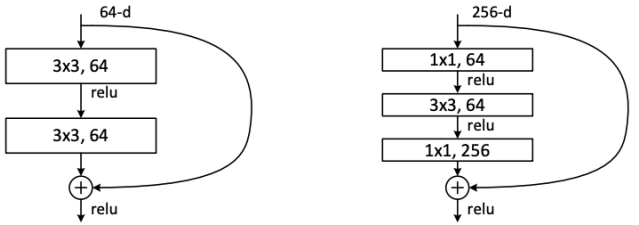


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

Experiments

CIFAR-10 and Analysis

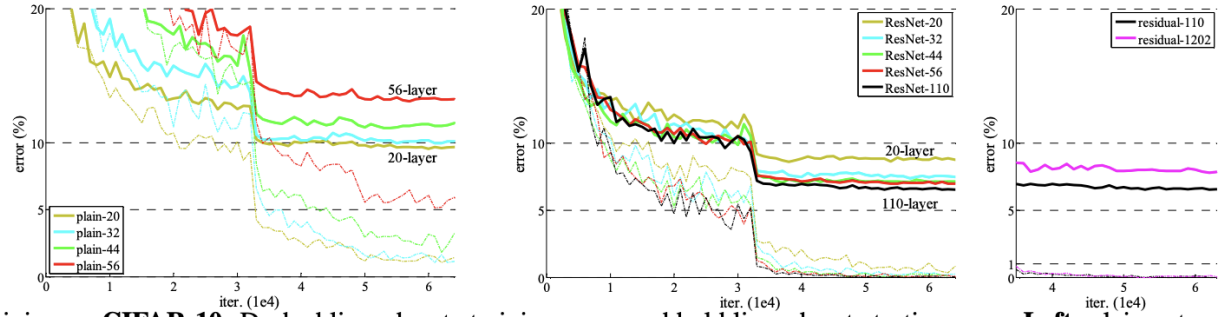


Figure 6. Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing error. **Left**: plain networks. The error of plain-110 is higher than 60% and not displayed. **Middle**: ResNets. **Right**: ResNets with 110 and 1202 layers.

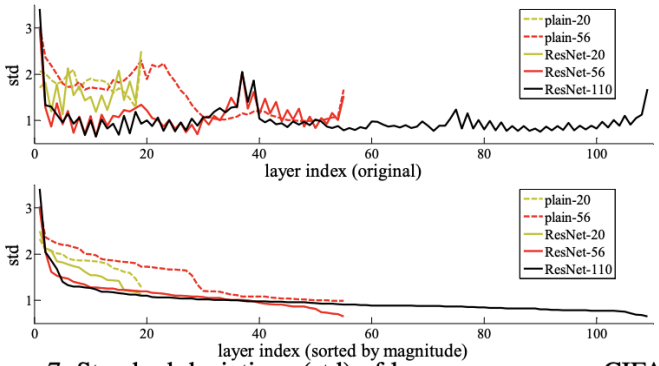


Figure 7. Standard deviations (std) of layer responses on CIFAR-10. The responses are the outputs of each 3×3 layer, after BN and before nonlinearity. **Top**: the layers are shown in their original order. **Bottom**: the responses are ranked in descending order.

training data	07+12	07++12
test data	VOC 07 test	VOC 12 test
VGG-16	73.2	70.4
ResNet-101	**76.4**	**73.8**

Table 7. Object detection mAP (%) on the PASCAL VOC 2007/2012 test sets using **baseline** Faster R-CNN. See also Table 10 and 11 for better results.

metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

Table 8. Object detection mAP (%) on the COCO validation set using **baseline** Faster R-CNN. See also Table 9 for better results.