Deep Residual Learning for Image Recognition

深度网络有什么好处？

(1) 特征的“等级”随增网络深度的加深而变高

(2) 极其深的深度使该网络拥有极强大的表达能力

50, 40]. Deep networks naturally integrate low/mid/high-level features [50] and classifiers in an end-to-end multi-layer fashion, and the “levels” of features can be enriched by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance,

Question 1:

Driven by the significance of depth, a question arises: *Is learning better networks as easy as stacking more layers?*

训练深度网络的一个重要问题：

- 梯度弥散（梯度爆炸）vanishing/exploding gradients

An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hamper convergence from the beginning.

什么是梯度弥散:

基于反向传播法计算梯度优化的神经网络，由于反向传播求隐藏层梯度时利用了链式法则，梯度值会进行一系列的连乘，导致浅层隐藏层的梯度会出现剧烈的衰减，这也就是梯度消失问题的本源.（当然这是主要原因，随着网络结构的加深，一些更深层次的原因也被发现了）

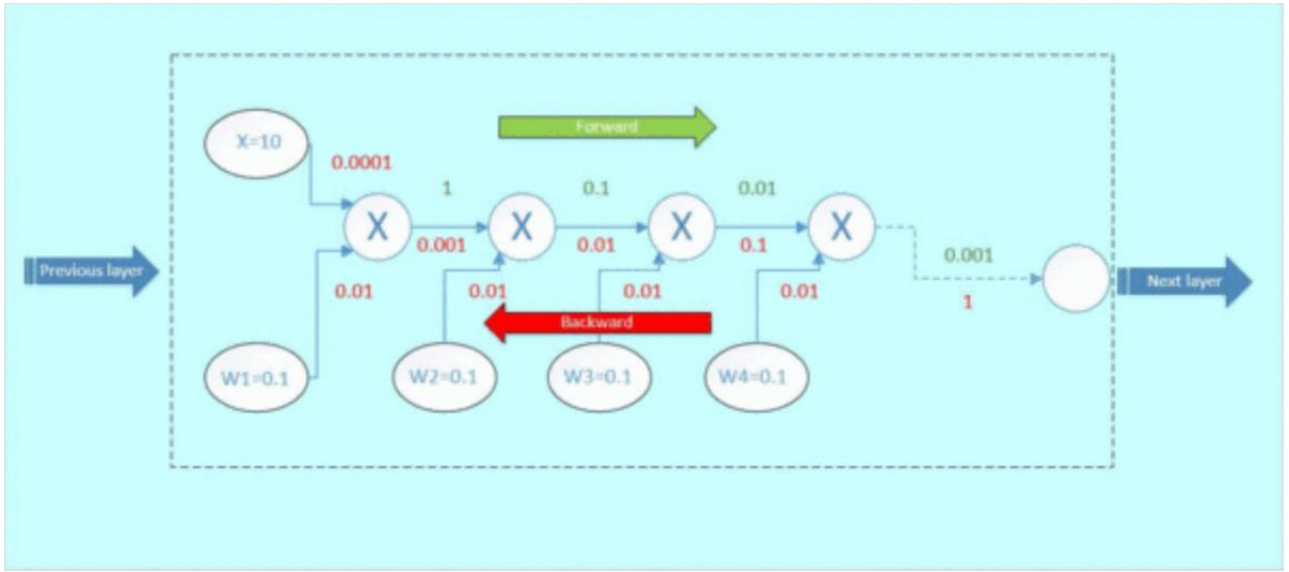


图1 后向传播

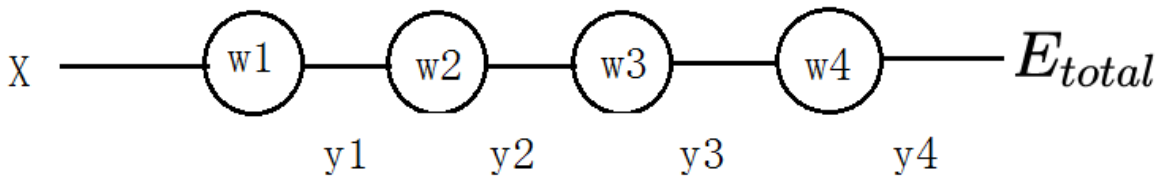
解决办法:

however, has been largely addressed by normalized initial-
ization [23, 9, 37, 13] and intermediate normalization layers

什么是梯度弥散（爆炸）简单例子

$$w_1^+ = w_1 - \eta \frac{\partial E_{total}}{\partial w_1}$$

$$y_i = \sigma(z_i) = \sigma(w_i x_i + b_i)$$



$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial y_4}\frac{\partial y_4}{\partial z_4}\frac{\partial z_4}{\partial x_4}\frac{\partial x_4}{\partial z_3}\frac{\partial z_3}{\partial x_3}\frac{\partial x_3}{\partial z_2}\frac{\partial z_2}{\partial x_2}\frac{\partial x_2}{\partial z_1}\frac{\partial z_1}{\partial w_1}$$

$$= \frac{\partial E_{total}}{\partial y_4}\sigma'(z_4)\, w_4 \sigma'(z_3)\, w_3 \sigma'(z_2)\, w_2 \sigma'(z_1) x_1$$

因此对于上面的链式求导，层数越多:
(1) 我们初始化的网络权值 w 通常都在0 附近，W2*W3*W4 越小，因而导致梯度消失的情况出现。
(2) 如果w 都大于 1，或者部分比较大的情况下，就会产生梯度爆炸。

训练深度网络所遇到的第二个重要的问题

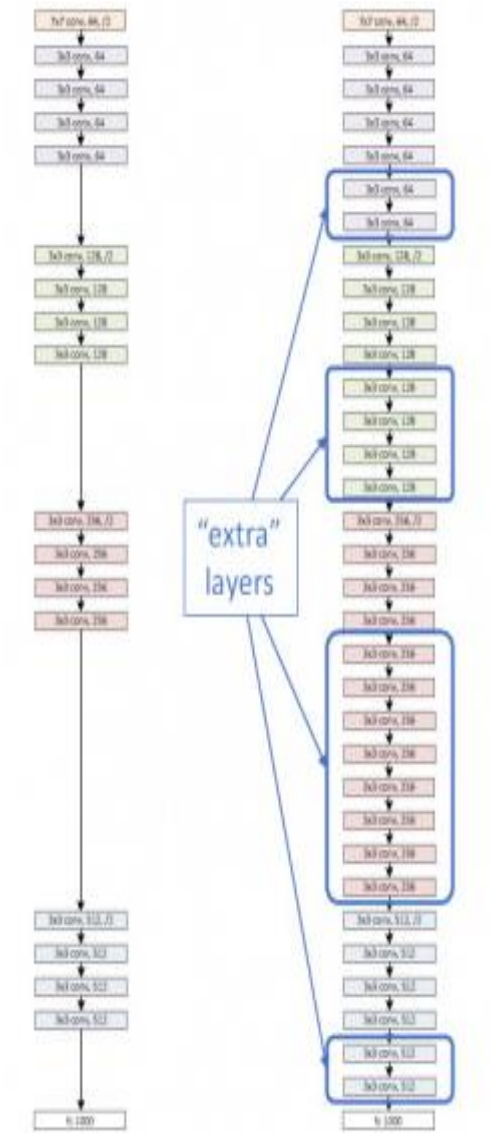


图2 浅层　　深层

•原始层：由一个已经学会的较浅模型复制得来
•附加层：设置为“恒等”
•至少具有相同的训练误差

优化难题：随着网络层数不断加深，求解器不能找到解决途径

实际情况：
很多时候，附加层不能被训练为‘恒等’，导致浅层和深层不具备相同误差

本想美滋滋的开始加深网络开始训练

氧化钙

层数越深，越暴露出一个问题

退化问题：层数过深的平原网络具有更高的训练误差

浅层和深层的神经网络结果对比

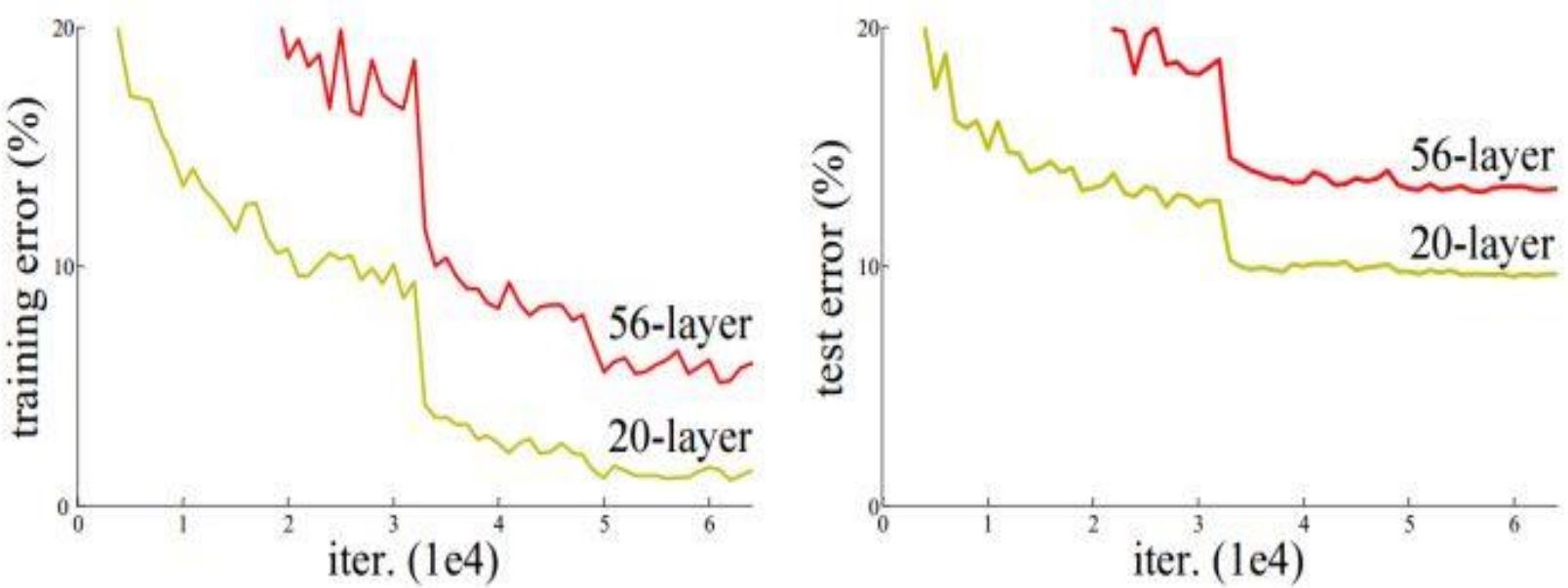


Figure 2. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

图3 浅层和深层训练误差和测试误差对比

When deeper networks are able to start converging, a *degradation* problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly.

我们知道

一个两层神经网络就可以拟合任意一个函数

二层平原网络（Plain Net）

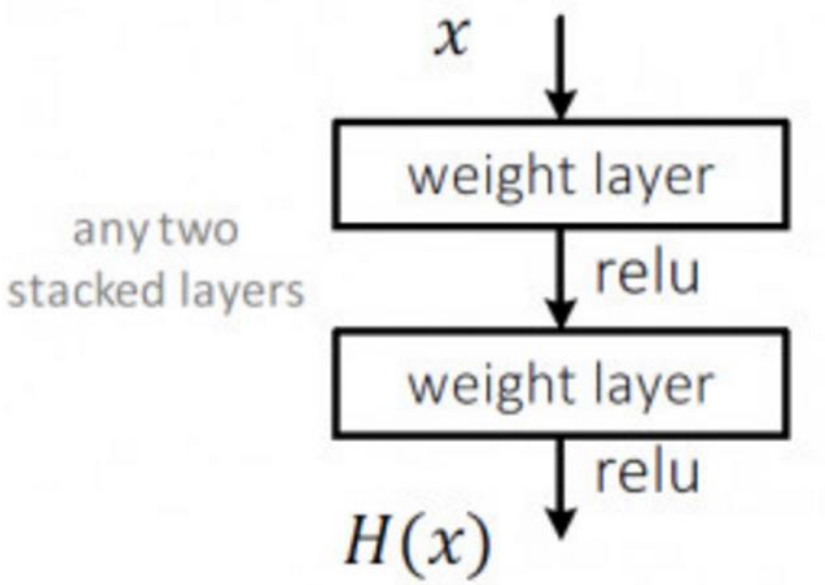
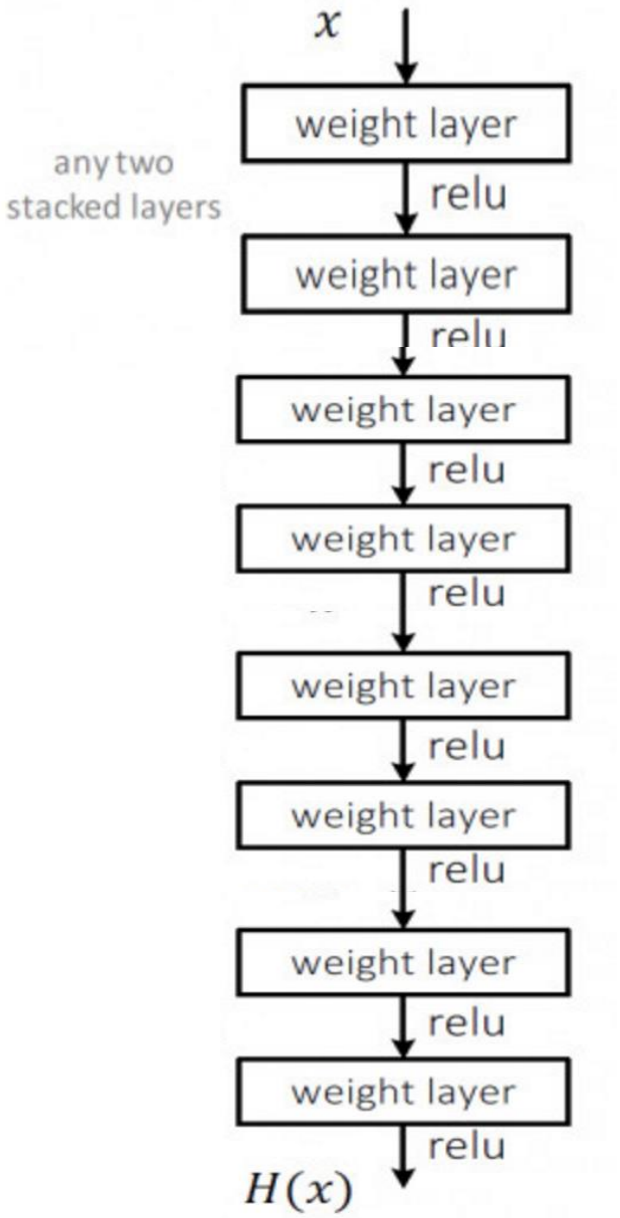


图4 一个平原网络的其中两层

（1）H(x)是任意一种理想的映射

（2）希望第2权重层输出能够与理想H(x)拟合

多层平原网络（Plain Net）

- 

如果我们前面6个layers 已经很好的拟合数据分布了

那7,8 layer 有啥用呢？

图5 一个平原网络中的八个weight layers

解决办法：Residual Net

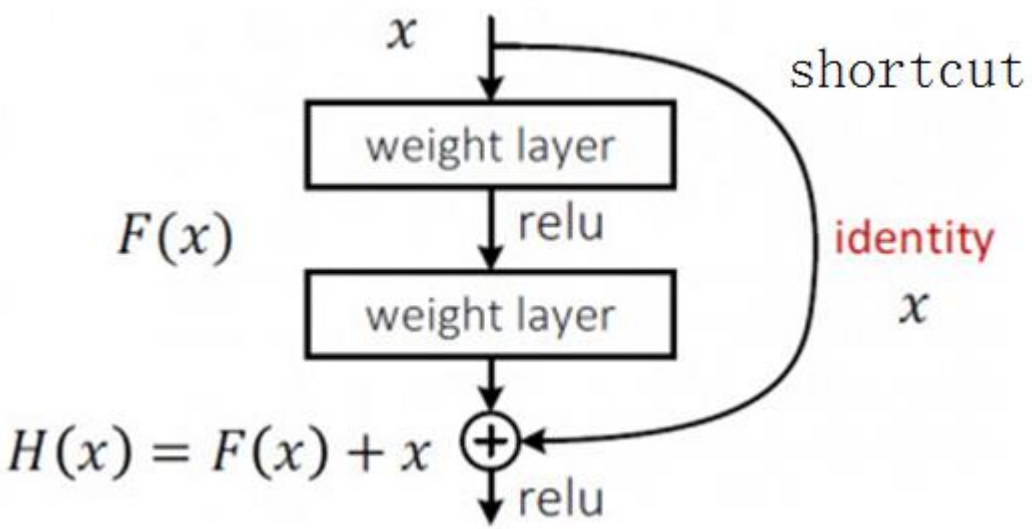


图6 残差模块

The degradation problem suggests that the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers. With the residual learning reformulation, if identity mappings are optimal, the solvers may simply drive the weights of the multiple nonlinear layers toward zero to approach identity mappings.

(1) 增加了一个shortcut
(2) 我们就拟合的对象就变成F（x）,F（x）就是残差
(3) 拟合目标变成使F（x）趋近于0

rather than expect stacked layers to approximate $\mathcal{H}(\mathbf{x})$, we explicitly let these layers approximate a residual function $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$. The original function thus becomes $\mathcal{F}(\mathbf{x})+\mathbf{x}$. Although both forms should be able to asymptotically approximate the desired functions (as hypothesized), the ease of learning might be different.

解决问题1：退化问题

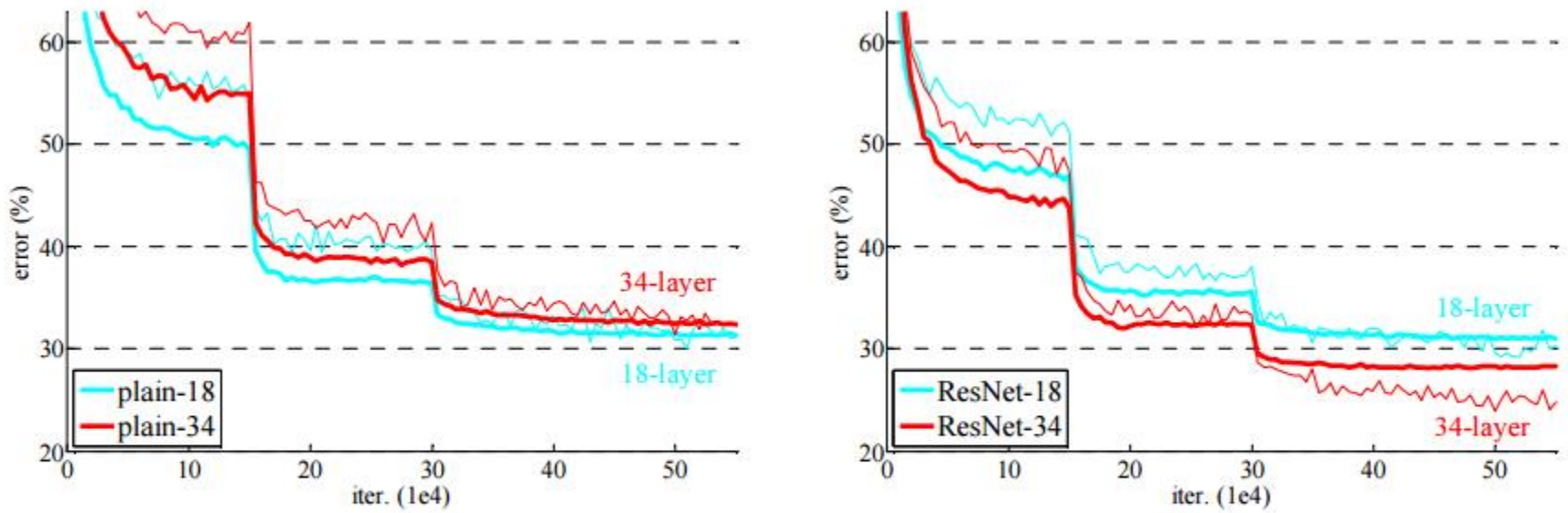


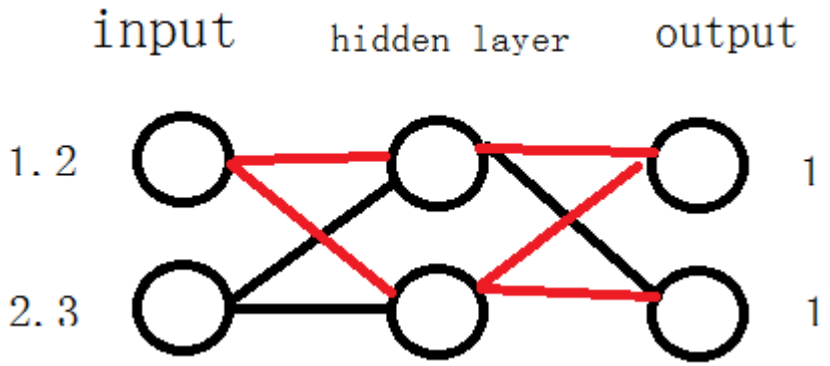
Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

图7 18层和34层网络在平原网络和残差网络对比试验

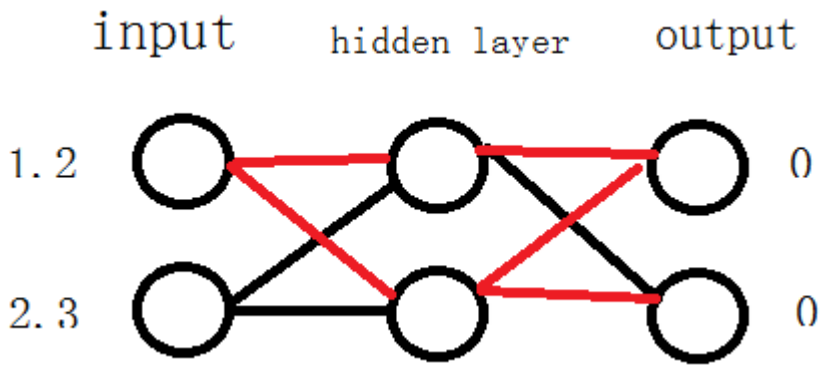
结果：对于Resnet,较深的模型生比较浅的模型有更低的训练误差，说明退化问题得到一定解决

怎么样去理解所谓的残差F(X) 要比原始期望映射H（x）更容易优化呢？
总即让F(x) 去逼近0

最开始Weight layer 的 初始化是让 weight 在0 附近



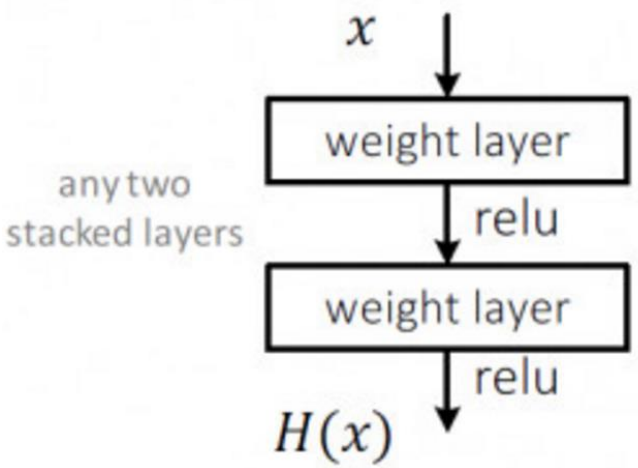
1.2x0.1+2.3x0.1 = 0.35 -------1

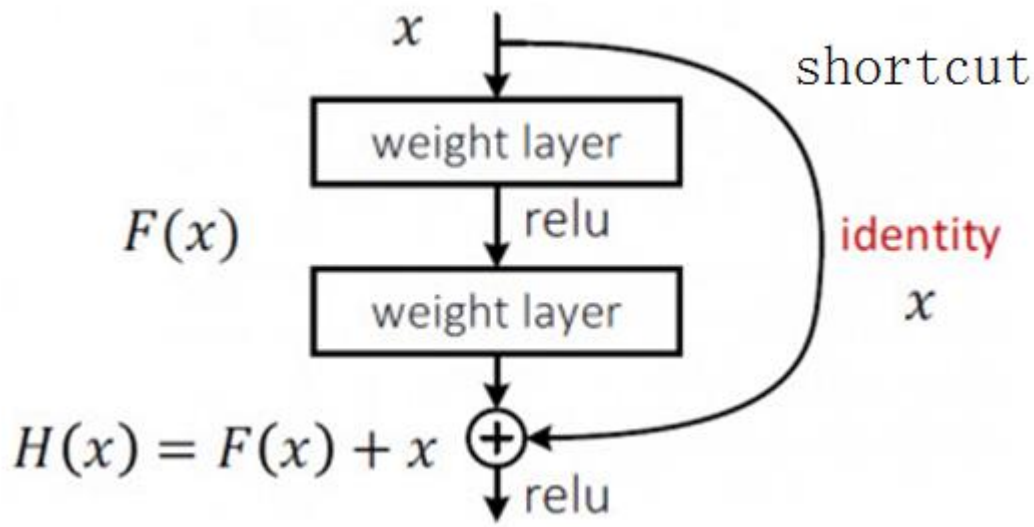


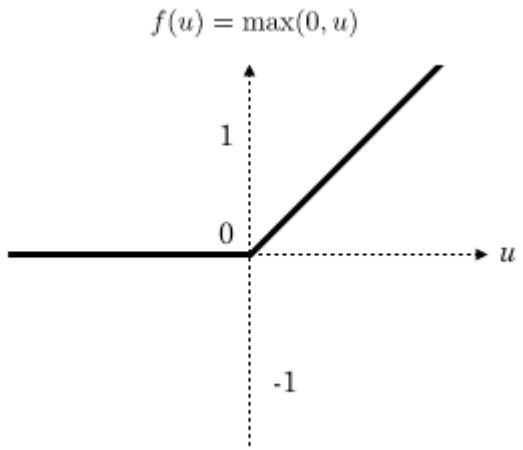
1.2x0.1+2.3x0.1 = 0.35 -------0

W = 0.1

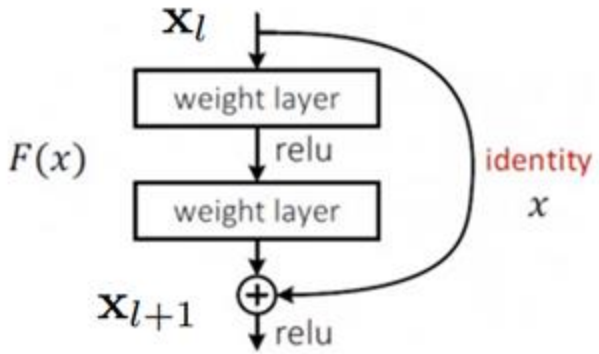
Relu 的影响


x
weight layer
relu
weight layer
relu
H(x)
any two
stacked layers


x
shortcut
weight layer
F(x)
relu
identity
x
weight layer
H(x) = F(x) + x
relu


f(u) = max(0, u)
1
0
u
-1

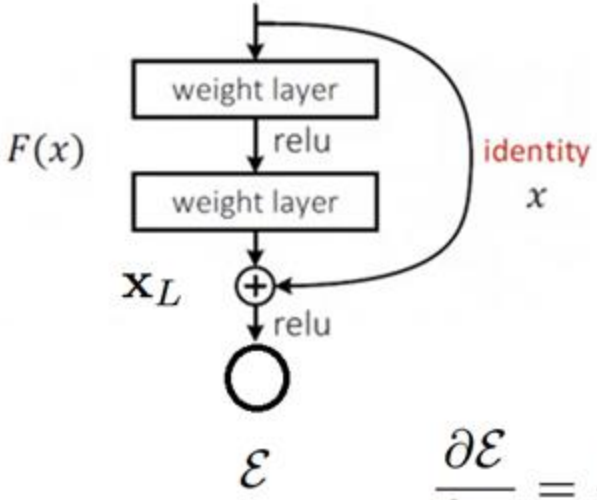
解决问题2：梯度弥散（或者爆炸）



$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l)$$

a term of $\frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}$ that propagates information directly without concerning any weight layers, and another term of $\frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(\frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}\right)$ that propagates through the weight layers. The additive term of $\frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}$ ensures that information is directly propagated back to *any shallower unit* l.

$$\mathbf{x}_{l+2} = \mathbf{x}_{l+1} + \mathcal{F}(\mathbf{x}_{l+1}, \mathcal{W}_{l+1}) = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l) + \mathcal{F}(\mathbf{x}_{l+1}, \mathcal{W}_{l+1})$$



$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)$$

因为shortcut 的加入使得gradient通过'1' 流回任意浅层l,避免了经过weight layer 造成的梯度弥散

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right)$$

如果输入x 和输出F（X）的维度不一致怎么办

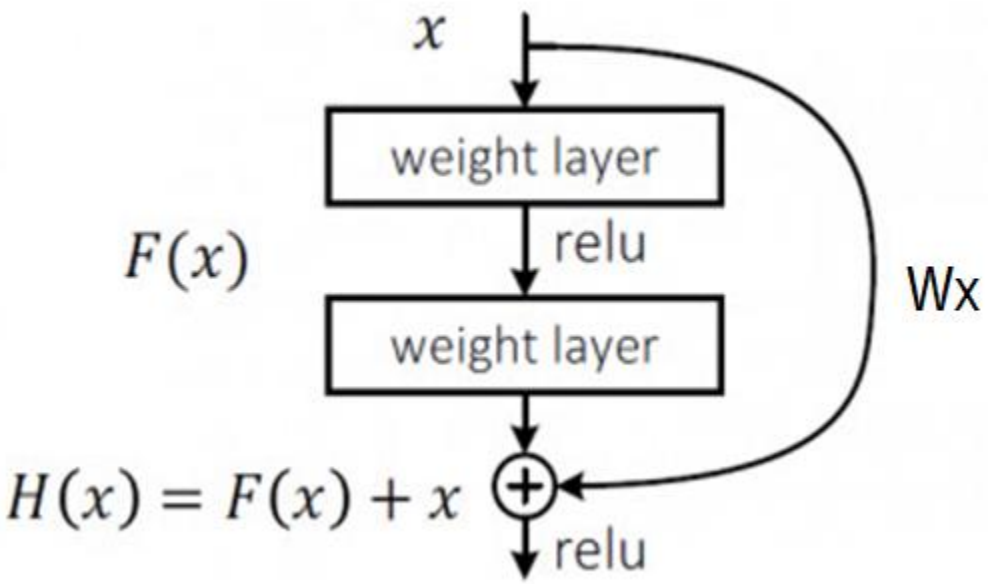


图7 带有线性变换的残差模块

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$

（1）快捷连接仍然使用自身映射，对于维度的增加用零来填补空缺，此策略不会引入额外的参数。

（2）可以在shortcut时对x做一个线性变换Ws ,相当于加入了1x1卷积层

针对50层以上residual unit 变换

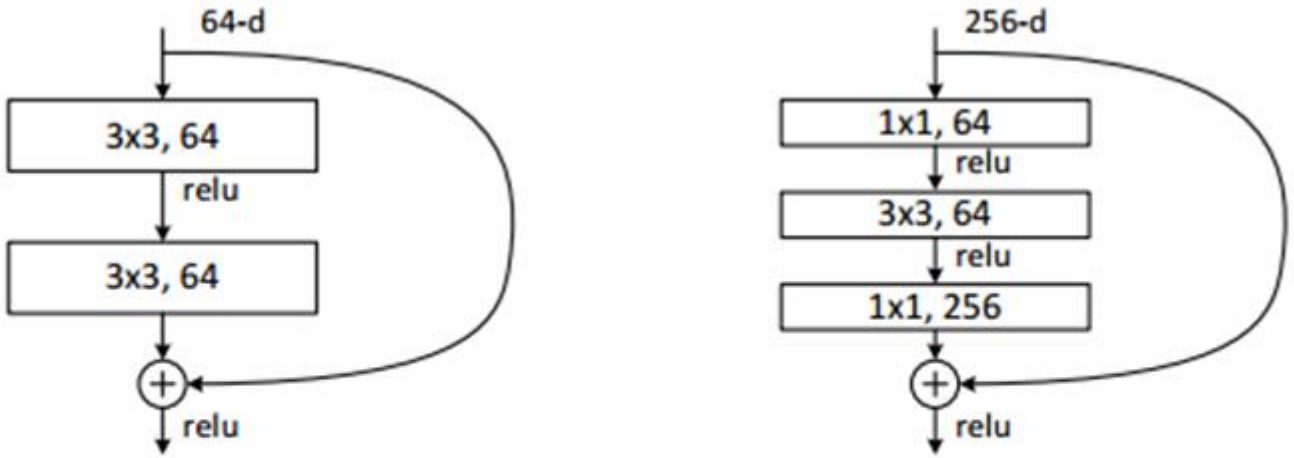


Figure 9. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

传话游戏

Resnet 可以理解为传话游戏

有时候传的人越多

错误率越高

Shortcut就相当于直接把第一个人说的话告诉第三个人

网上的一些理解

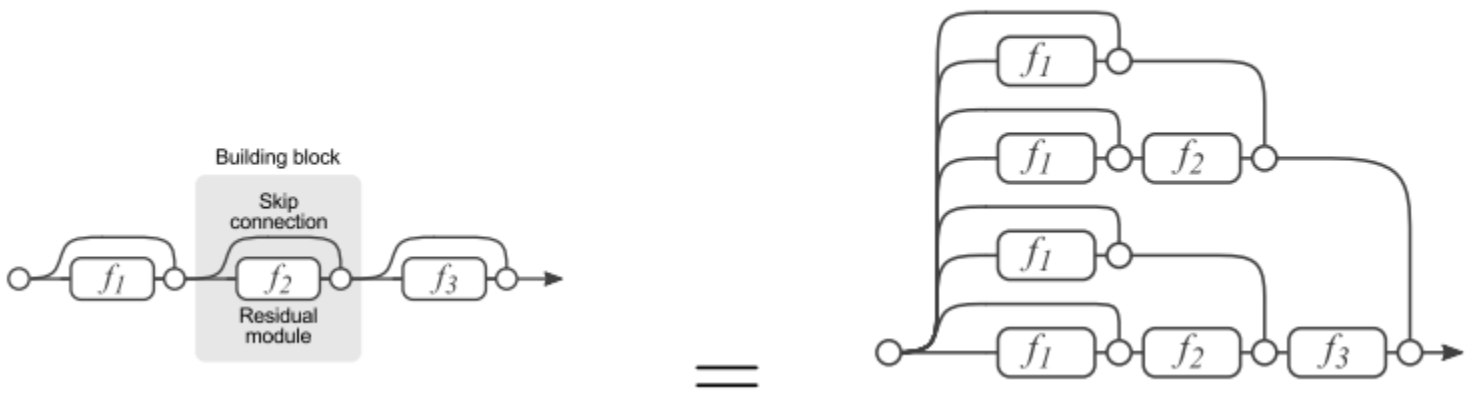


图10 网上借来的对残差模块的理解

- 从这可以看出其实ResNet是由大多数中度网络和一小部分浅度网络和深度网络组成的，说明虽然表面上ResNet网络很深，但是其实起实际作用的网络层数并没有很深

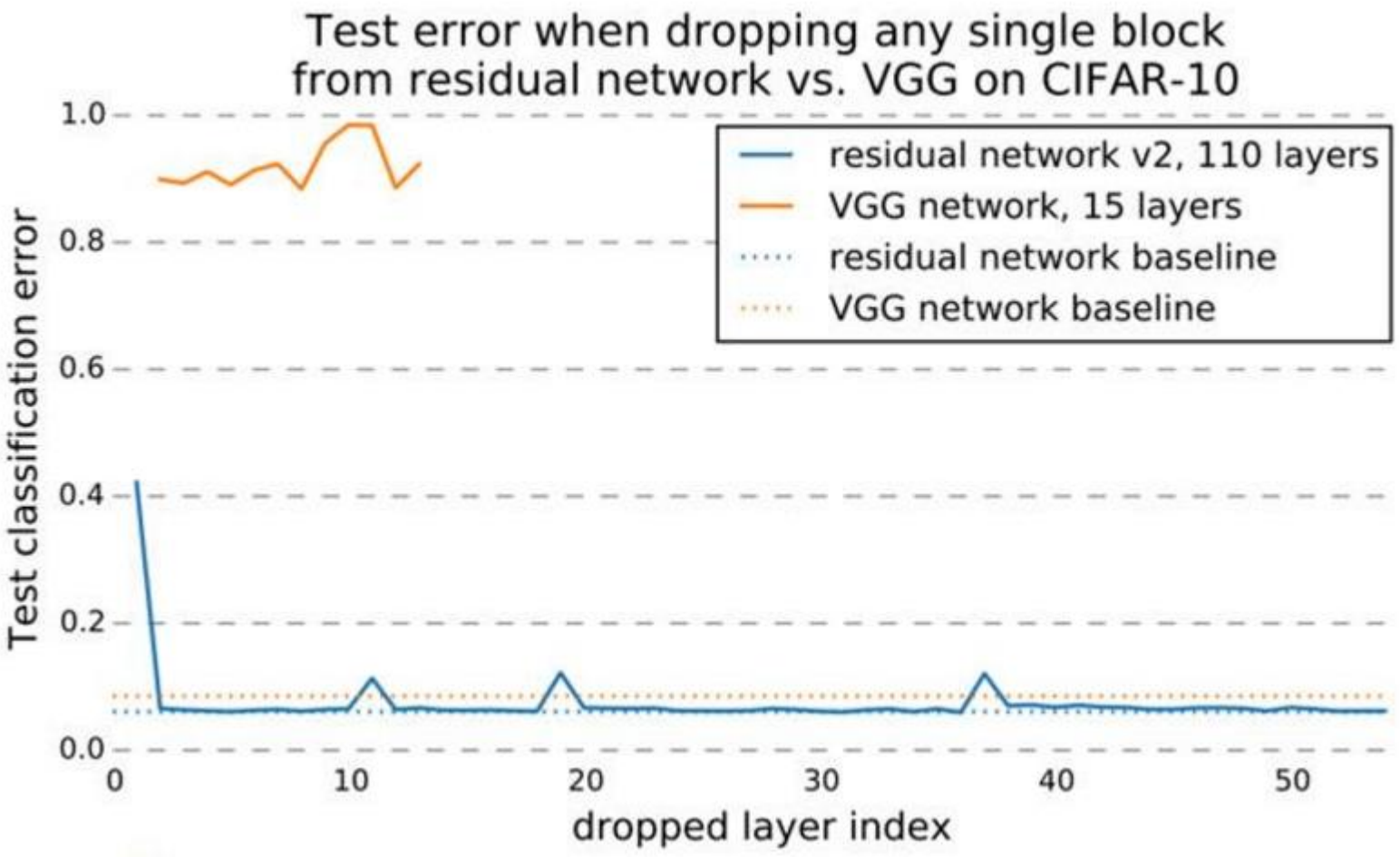


图11 又网上借来的，去掉残差网络和VGG中一些模块的对比试验

网上的一些理解

Microsoft Wins ImageNet 2015 through Highway Net (or Feedforward LSTM) without Gates

Jürgen Schmidhuber

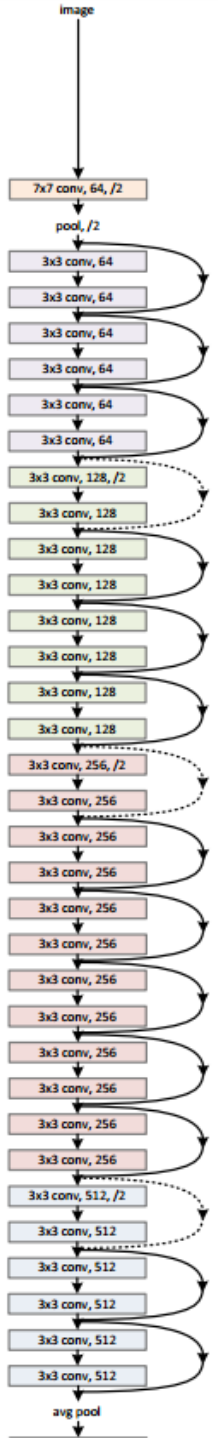
Microsoft Research dominated the ImageNet 2015 contest with a very deep neural network of 150 layers [1]. Congrats to Kaiming He & Xiangyu Zhang & Shaoqing Ren & Jian Sun on the great results [2]!

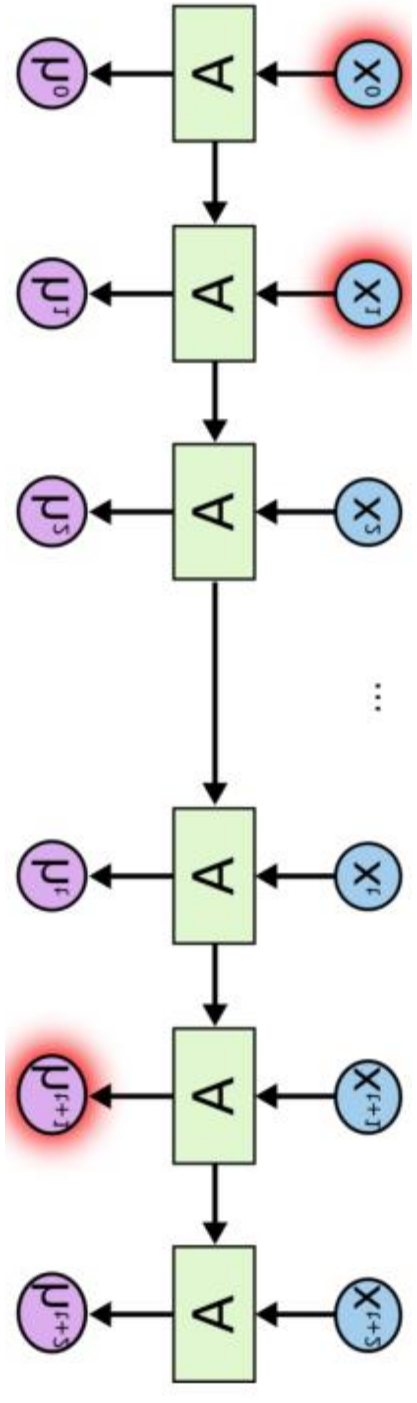
Their Residual Net or **ResNet** [1] of December 2015 is a special case of our **Highway Networks** [4] of May 2015, the first very deep feedforward networks with hundreds of layers. Highway nets are essentially feedforward versions of recurrent Long Short-Term Memory (LSTM) networks [3] **with** forget gates (or gated recurrent units) [5].

Let *g, t, h* denote non-linear differentiable functions. Each non-input layer of a Highway Net computes *g(x)x + t(x)h(x)*, where *x* is the data from the previous layer. (Like LSTM [3] with forget gates [5] for recurrent networks.)

The CNN layers of ResNets [1] do the same with *g(x)=1* (a typical Highway Net initialisation) and *t(x)=1*, essentially like a Highway Net or a feedforward LSTM [3] **without** gates.

本质其实和LSTM类似，通过加入gate的方法保留任何想长期保留的信息





后续推荐阅读：

- 《极深网络（**ResNet/DenseNet**）》 http://blog.csdn.net/malefactor/article/details/67637785
- 《**The Shattered Gradients Problem: If resnets are the answer, then what is the question?** 》 https://arxiv.org/pdf/1702.08591.pdf
- 《 **Densely Connected Convolutional Networks**》 https://arxiv.org/abs/1608.06993
- 《**Training Very Deep Networks**》**https**://arxiv.org/pdf/1507.06228v2.pdf
- 《Deep Residual Learning for age Recognition》 https://arxiv.org/pdf/1512.03385.pdf -
- 《Identity Mappings in Deep Residual Networks》 https://-

下期预告

- 一步一步，从论文到代码，来实现Resnet

更多内容,欢迎关注
本人公众号：随波竺流 （follow_bobo）
知乎：蒋竺波
合作公众号：一度AI （ onedegree_ai）