简介

作者发现了一个问题，深层网络学习性能不如浅层，在训练集和测试集上表现均不佳，而且不是过拟合。按理来说深层网络的表现至少不会弱于浅层，因为多出来的层理论上可以学成恒等映射，这样至少和浅层网络性能一致。那么说明深层网络训练要难于浅层，本文的思路是加入残差结构。假设原来block的潜在映射为 $H(x)$ 那么我们现在让其学习的映射为 $F(x) := H(x) - x$。这样如果这一个block的最优映射为恒等映射，那么就很容易学习到。

3.Deep Residual Learning

3.1. Residual Learning

理论上神经网络学习函数 $H(x)$ 或 $H(x) - x$ 没有差别（因为能近似任意函数），但是学习的难易度可能不同。
虽然并不是所有block的最优映射优势恒等映射，但是实验证明将恒等映射作为precondition是有效的，而且实验发现残差函数通常有更小的响应（标准差更小）

3.2.Identity Mapping by Shorcuts

对于一个两层的res block，有：

$$y = F(x, W_i) + x$$

$$F = W_2\,\sigma(W_1 x)$$

其中sigma是RELU，bias项忽略没写，残差连接在第二层的非线性激活之前，加法是element-wise，加数和被加数的维度必须一致，如果不一致的就用一个矩阵进行投影：

$$y = F(x, W_i) + W_s x$$

维度一致其实也可以用Ws但是没有这个必要，本文skip了2到3层，大于3层也可以，但是只跳一层没什么用

$$y = W_1 x + x$$

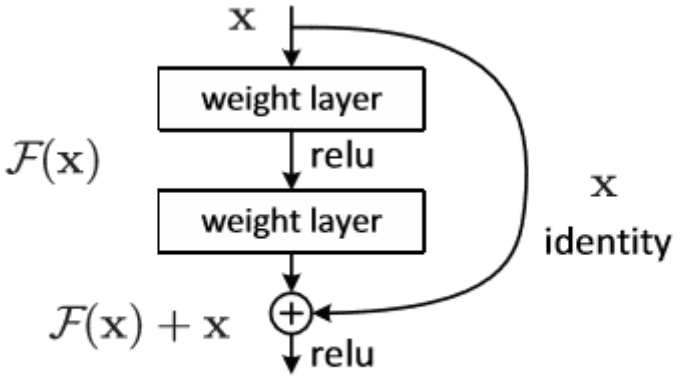


Figure 2. Residual learning: a building block.

加法也适用于卷积层，如果维度不一致本文实验用1x1卷积进行转换

3.3. Network Architectures

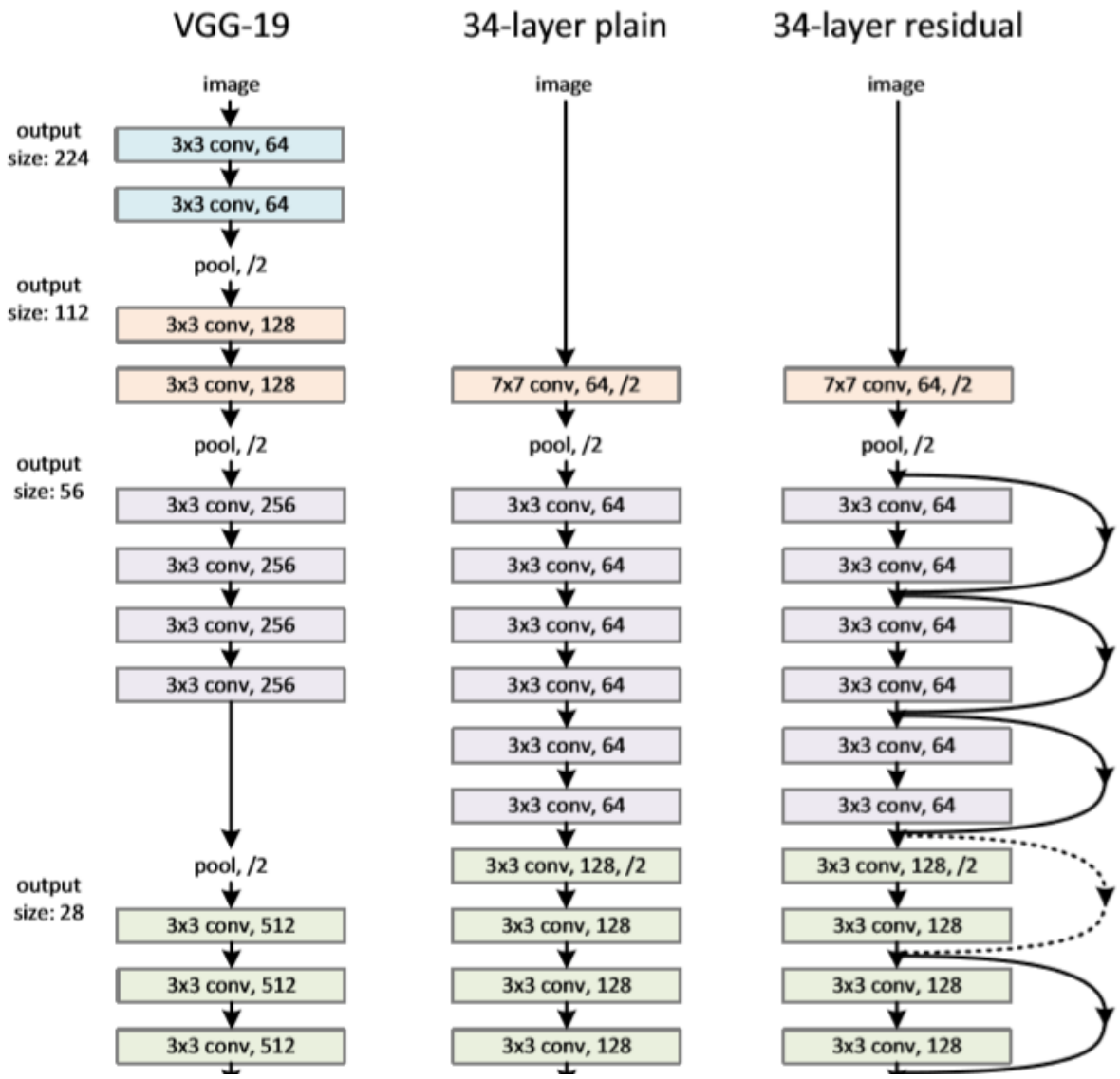
Plain Network

主要灵感来自于VGG
卷积层基本为3x3卷积满足下面两个条件：
i) feature map size相同的层，通道数（filters）也相同
ii) feature map尺寸减半，filters numbers double来保持每层的时间复杂度？？（time complexity per layer）
然后卷积层之间没有用pool层降低feature size，而是用stride 2使得feature map的size减半。最后用global average pooling和1000个units的FC层馈送入softmax，最后有权重的层数为34层。
这个plain network的参数只有VGG19（19.6亿FLOPs）的18%(3.6亿FLOPs)

Resizdual Network

残差版本的Plain网络，残差链接如果维度不同时有下面两种方案：
A) 对于增加的维度采用zero padding B) 1x1卷积投影变换

对用这两种情况，当feature map跨越两个不同sizes的时候，都用stride 2



3.4 Implementation

- random scale aug:将图片短边随机sample为[256, 480]
- 224x224 crop
- horizontal flip
- per-pixel mean subtracted
- stardard color aug
- 每个卷积之后激活之前用BN
- 初始化方法来源于论文[13]
- SGD， minibatch：256， lr:0.1 碰到瓶颈lr除以10，iters： 60x10^4，weight decay:0.0001,momentum: 0.9,no dropout（flow the practice in [16]）
- 测试阶段用了标准的10-crop，采用了全卷积形式?见[41, 13]，对多个尺度取了平均（图像的短边为{224,256,384,480,640}）

plain network训练得不好应该并不是vanishing gradient的原因，因为作者观测了bn层的前项和后向，发现都是健康的，那收敛不充分的原因应该是源自于收敛速度过慢（然而这作者迭代了3x的次数

仍然存在degration的现象，说明迭代更多次数解决不了问题）

bottle neck结构

由于原来的结构中随着channel的增大，维度会越来越大从而大大增加训练时间，作者提出了这种bottle neck结构其实就是用1x1卷积进行降维然后升维。而且作者还提出，如果所有的res-block都用projection的方式作为shot-cut连接，那么其实参数会double

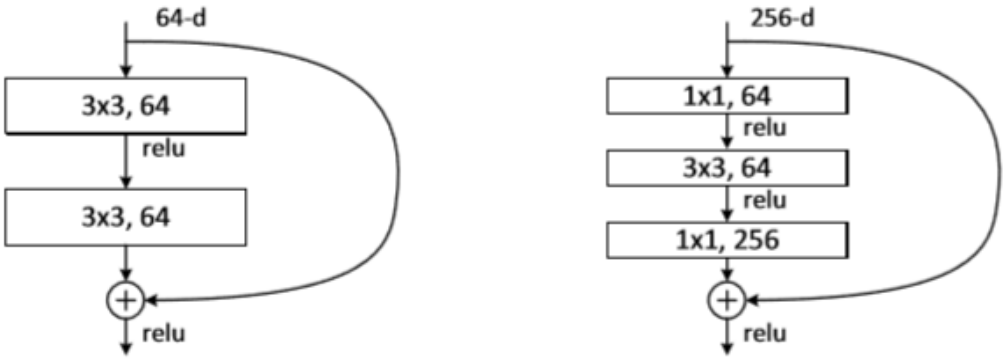


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

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

Table 1. Architectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked. Down-sampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

问题

1. residual block按理来说必须要求前后的h x w必须要一致（c可以通过1x1卷积来扩展），但是resnet的res-block中间跨了一个3x3卷积按理说hxw变了，难道是中间用了padding么?

回答，看原文章和程序后发现确实是使用了padding=1，也就是res-block的时候尺寸是不变的，只有通道变化。需要downsample的时候用专门的down sample模块，具体实现是用卷积stride = 2，注意，在原platin network中，如果3x3的conv block后面出现了/2，那么代表这里用了stride=2的down sample，那么如果res-block跨尺寸连接的话，也需要用1x1,s=2的卷积核。在bottle neck结构中，如果需要down sample，也是由3x3卷积的s=2完成的。

模型训练

训练cifar-10

cifar-10的模型结构和imagenet的结构不同，因为其输入比较小，不适合连续的下采样。
采用6n+2的结构，先是一个3x3卷积，然后是3个2n层的residual结构，最后跟一个avgpool层。下采样是通过stride=2的卷积完成的，只下采样了两次。而且最后的残差连接用的是直接pad 0使得channel一样而不是通过1x1卷积

output map size	32×32	16×16	8×8
# layers	$1+2n$	$2n$	$2n$
# filters	16	32	64

不采用bottle neck结构,直接连续的2个3x3卷积。

注意事项

- 看图，residual block的ReLU是在add之后的