ResNet论文翻译——中文版

| |

ResNet论文翻译——中文版

文章作者：Tyan
博客：noahsnail.com | CSDN | 简书

声明：作者翻译论文仅为学习，如有侵权请联系作者删除博文，谢谢！

翻译论文汇总：https://github.com/SnailTyan/deep-learning-papers-translation

Deep Residual Learning for Image Recognition

Abstract

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——8× deeper than VGG nets [40] 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.

摘要

更深的神经网络更难训练。我们提出了一种残差学习框架来减轻网络训练，这些网络比以前使用的网络更深。我们明确地将层变为学习关于层输入的残差函数，而不是学习未参考的函数。我们提供了全面的经验证据说明这些残差网络很容易优化，并可以显著增加深度来提高准确性。在ImageNet数据集上我们评估了深度高达152层的残差网络——比VGG[40]深8倍但仍具有较低的复杂度。这些残差网络的集合在ImageNet测试集上取得了`3.57%`的错误率。这个结果在ILSVRC 2015分类任务上赢得了第一名。我们也在CIFAR-10上分析了100层和1000层的残差网络。

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.

对于许多视觉识别任务而言，表示的深度是至关重要的。仅由于我们非常深度的表示，我们便在COCO目标检测数据集上得到了28%的相对提高。深度残差网络是我们向ILSVRC和COCO 2015竞赛提交的基础，我们也赢得了ImageNet检测任务，ImageNet定位任务，COCO检测和COCO分割任务的第一名。

1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 49, 39]. Deep networks naturally integrate low/mid/high-level features [49] 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 [40, 43] reveals that network depth is of crucial importance, and the leading results [40, 43, 12, 16] on the challenging ImageNet dataset [35] all exploit “very deep” [40] models, with a depth of sixteen [40] to thirty [16]. Many other non-trivial visual recognition tasks [7, 11, 6, 32, 27] have also greatly benefited from very deep models.

1. 引言

深度卷积神经网络[22, 21]导致了图像分类[21, 49, 39]的一系列突破。深度网络自然地将低/中/高级特征[49]和分类器以端到端多层方式进行集成，特征的“级别”可以通过堆叠层的数量（深度）来丰富。最近的证据[40, 43]显示网络深度至关重要，在具有挑战性的ImageNet数据集上领先的结果都采用了“非常深”[40]的模型，深度从16 [40]到30 [16]之间。许多其它重要的视觉识别任务[7, 11, 6, 32, 27]也从非常深的模型中得到了极大受益。

Driven by the significance of depth, a question arises: Is learning better networks as easy as stacking more layers? An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [14, 1, 8], which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [23, 8, 36, 12] and intermediate normalization layers [16], which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation [22].

在深度重要性的推动下，出现了一个问题：学些更好的网络是否像堆叠更多的层一样容易？回答这个问题的一个障碍是梯度消失/爆炸[14, 1, 8]这个众所周知的问题，它从一开始就阻碍了收敛。然而，这个问题通过标准初始化[23, 8, 36, 12]和中间标准化层[16]在很大程度上已经解决，这使得数十层的网络能通过具有反向传播的随机梯度下降（SGD）开始收敛。

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. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in [10, 41] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

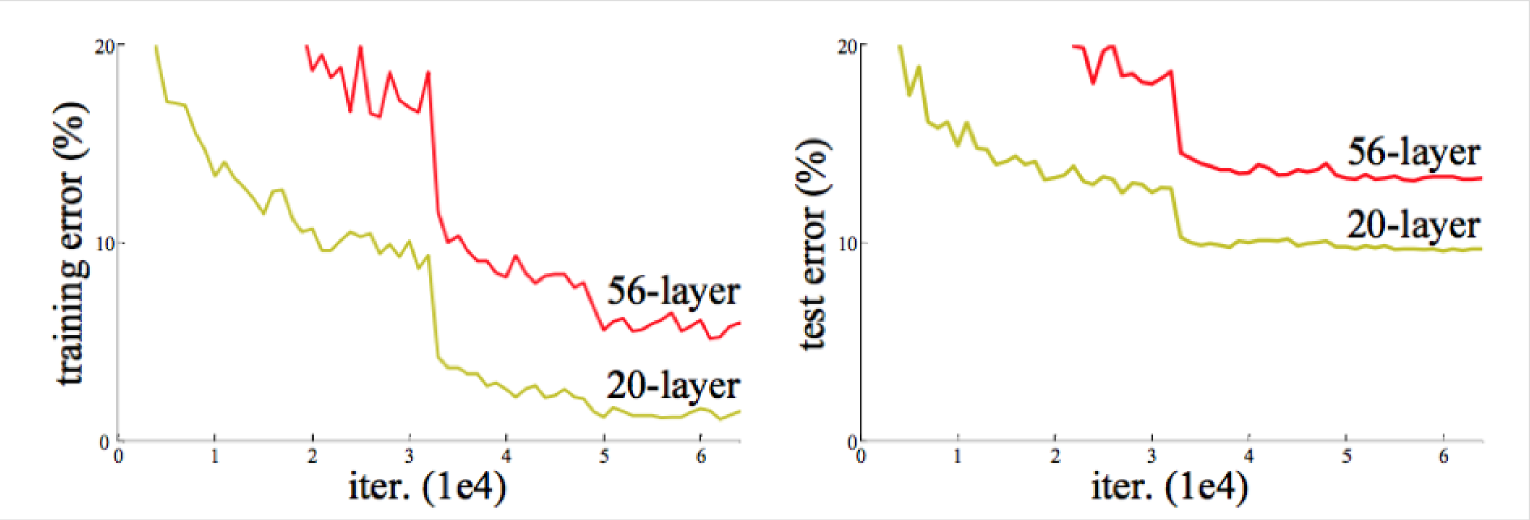


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “ plain ” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

当更深的网络能够开始收敛时，暴露了一个退化问题：随着网络深度的增加，准确率达到饱和（这可能并不奇怪）然后迅速下降。意外的是，这种下降不是由过拟合引起的，并且在适当的深度模型上添加更多的层会导致更高的训练误差，正如[10, 41]中报告的那样，并且由我们的实验完全证实。图1显示了一个典型的例子。

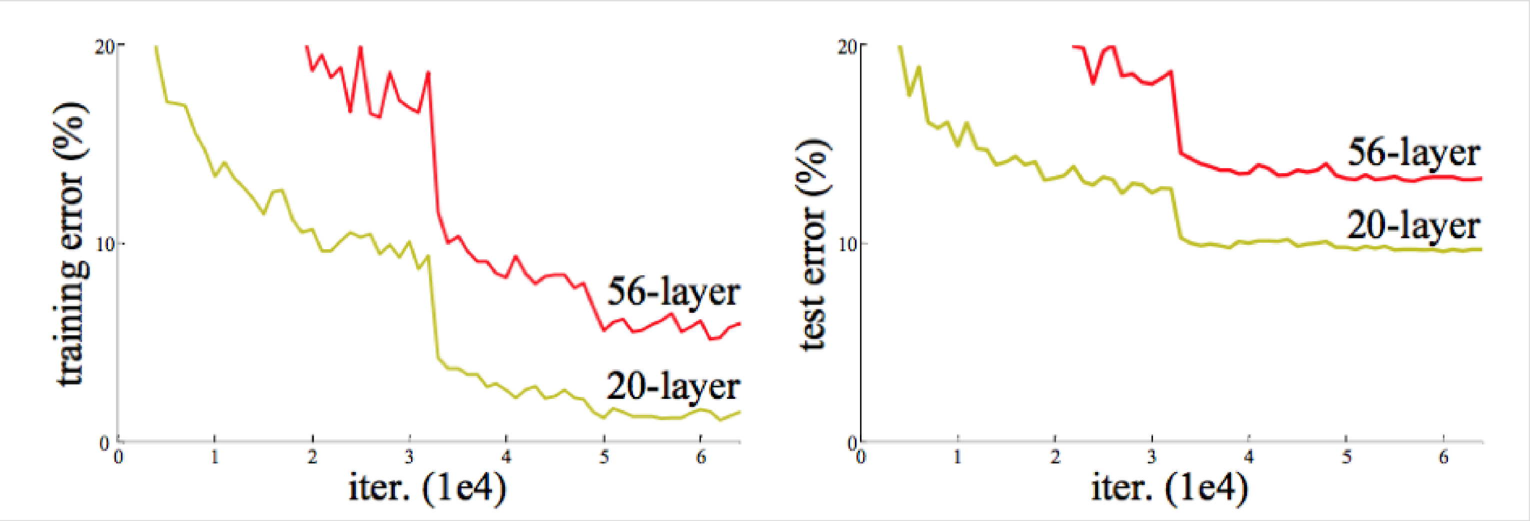


图1 20层和56层的“ 简单 ”网络在CIFAR-10上的训练误差（左）和测试误差（右）。更深的网络有更高的训练误差和测试误差。ImageNet上的类似现象如图4所示。

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution by construction to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers on hand are unable to find solutions that are comparably good or better than the constructed solution (or unable to do so in feasible time).

退化（训练准确率）表明不是所有的系统都很容易优化。让我们考虑一个较浅的架构及其更深层次的对象，为其添加更多的层。存在通过构建得到更深层模型的解决方案：添加的层是恒等映射，其他层是从学习到的较浅模型的拷贝。 这种构造解决方案的存在表明，较深的模型不应该产生比其对应的较浅模型更高的训练误差。但是实验表明，我们目前现有的解决方案无法找到与构建的解决方案相比相

对不错或更好的解决方案（或在合理的时间内无法实现）。

In this paper, we address the degradation problem by introducing a *deep residual learning* framework. Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. Formally, denoting the desired underlying mapping as $H(x)$, we let the stacked nonlinear layers fit another mapping of $F (x) := H(x) − x$. The original mapping is recast into $F(x) + x$. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.

在本文中，我们通过引入*深度残差学习*框架解决了退化问题。我们明确地让这些层拟合残差映射，而不是希望每几个堆叠的层直接拟合期望的基础映射。形式上，将期望的基础映射表示为$H(x)$，我们将堆叠的非线性层拟合另一个映射$F(x) := H(x) − x$。原始的映射重写为$F(x) + x$。我们假设残差映射比原始的、未参考的映射更容易优化。在极端情况下，如果一个恒等映射是最优的，那么将残差置为零比通过一堆非线性层来拟合恒等映射更容易。

The formulation of $F (x) + x$ can be realized by feedforward neural networks with “ shortcut connections ” (Fig. 2). Shortcut connections [2, 33, 48] are those skipping one or more layers. In our case, the shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers (Fig. 2). Identity shortcut connections add neither extra parameter nor computational complexity. The entire network can still be trained end-to-end by SGD with backpropagation, and can be easily implemented using common libraries (e.g., Caffe [19]) without modifying the solvers.

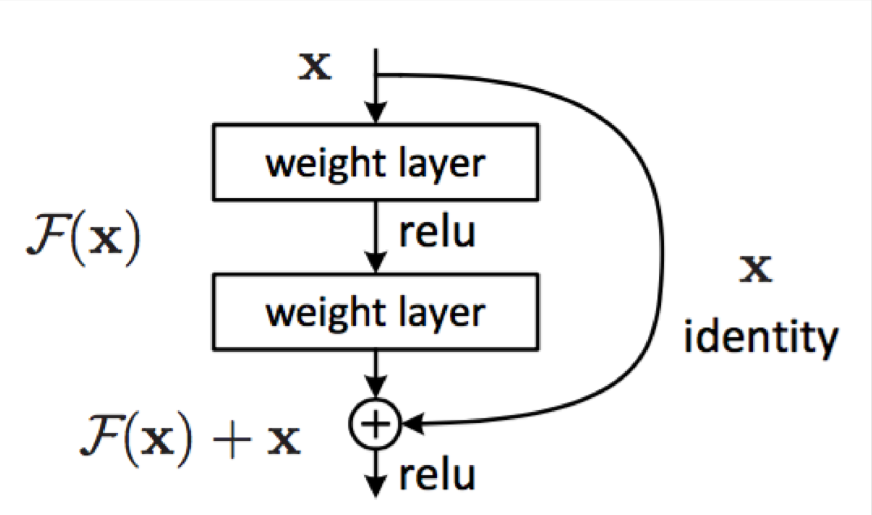

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2. Residual learning: a building block.

公式$F (x) + x$可以通过带有 “ 快捷连接 ” 的前向神经网络（图2）来实现。快捷连接[2, 33, 48]是那些跳过一层或更多层的连接。在我们的案例中，快捷连接简单地执行恒等映射，并将其输出添加到堆叠层的输出（图2）。恒等快捷连接既不增加额外的参数也不增加计算复杂度。整个网络仍然可以由带有反向传播的SGD进行端到端的训练，并且可以使用公共库（例如，Caffe [19]）轻松实现，而无需修改求解器。

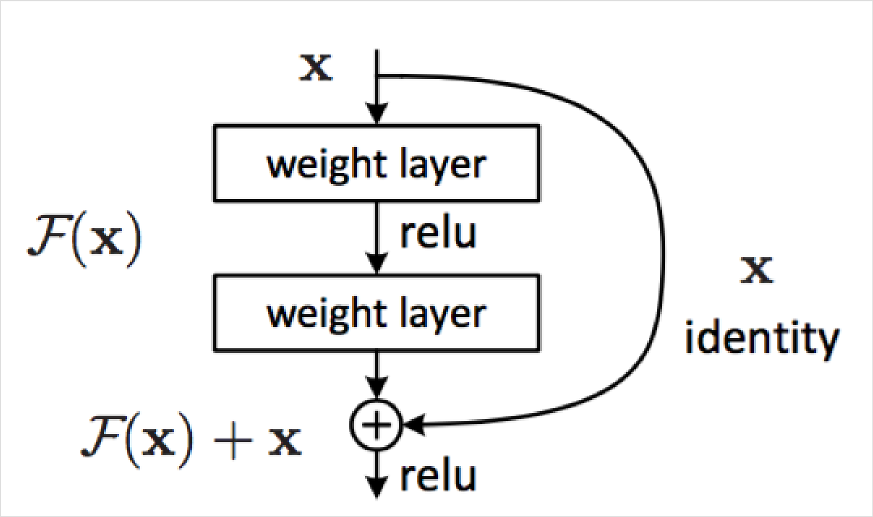


图2. 残差学习：构建块

We present comprehensive experiments on ImageNet [35] to show the degradation problem and evaluate our method. We show that: 1) Our extremely deep residual nets are easy to optimize, but the counterpart “ plain ” nets (that simply stack layers) exhibit higher training error when the depth increases; 2) Our deep residual nets can easily enjoy accuracy gains from greatly increased depth, producing results substantially better than previous networks.

我们在ImageNet[35]上进行了综合实验来显示退化问题并评估我们的方法。我们发现：1）我们极深的残差网络易于优化，但当深度增加时，对应的“ 简单 ”网络（简单堆叠层）表现出更高的训练误差；2）我们的深度残差网络可以从大大增加的深度中轻松获得准确性收益，生成的结果实质上比以前的网络更好。

Similar phenomena are also shown on the CIFAR-10 set [20], suggesting that the optimization difficulties and the effects of our method are not just akin to a particular dataset. We present successfully trained models on this dataset with over 100 layers, and explore models with over 1000 layers.

CIFAR-10数据集上[20]也显示出类似的现象，这表明了优化的困难以及我们的方法的影响不仅仅是针对一个特定的数据集。我们在这个数据集上展示了成功训练的超过100层的模型，并探索了超过1000层的模型。

On the ImageNet classification dataset [35], we obtain excellent results by extremely deep residual nets. Our 152-layer residual net is the deepest network ever presented on ImageNet, while still having lower complexity than VGG nets [40]. Our ensemble has `3.57% top-5` error on the ImageNet test set, and won the 1st place in the ILSVRC 2015 classification competition. The extremely deep representations also have excellent generalization performance on other recognition tasks, and lead us to further win the 1st places on: ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation in ILSVRC & COCO 2015 competitions. This strong evidence shows that the residual learning principle is generic, and we expect that it is applicable in other vision and non-vision problems.

在ImageNet分类数据集[35]中，我们通过非常深的残差网络获得了很好的结果。我们的152层残差网络是ImageNet上最深的网络，同时还具有比VGG网络[40]更低的复杂性。我们的模型集合在ImageNet测试集上有`3.57% top-5`的错误率，并在ILSVRC 2015分类比赛中获得了第一名。极深的表示在其它识别任务中也有极好的泛化性能，并带领我们在进一步赢得了第一名：包括ILSVRC & COCO 2015竞赛中的ImageNet检测，ImageNet定位，COCO检测和COCO分割。坚实的证据表明残差学习准则是通用的，并且我们期望它适用于其它的视觉和非视觉问题。

2. Related Work

Residual Representations. In image recognition, VLAD [18] is a representation that encodes by the residual vectors with respect to a dictionary, and Fisher Vector [30] can be formulated as a probabilistic version [18] of VLAD. Both of them are powerful shallow representations for image retrieval and classification [4, 47]. For vector quantization, encoding residual vectors [17] is shown to be more effective than encoding original vectors.

2. 相关工作

残差表示。在图像识别中，VLAD[18]是一种通过关于字典的残差向量进行编码的表示形式，Fisher矢量[30]可以表示为VLAD的概率版本[18]。它们都是图像检索和图像分类[4,47]中强大的浅层表示。对

于矢量量化，编码残差矢量[17]被证明比编码原始矢量更有效。

In low-level vision and computer graphics, for solving Partial Differential Equations (PDEs), the widely used Multigrid method [3] reformulates the system as subproblems at multiple scales, where each subproblem is responsible for the residual solution between a coarser and a finer scale. An alternative to Multigrid is hierarchical basis preconditioning [44, 45], which relies on variables that represent residual vectors between two scales. It has been shown [3, 44, 45] that these solvers converge much faster than standard solvers that are unaware of the residual nature of the solutions. These methods suggest that a good reformulation or preconditioning can simplify the optimization.

在低级视觉和计算机图形学中，为了求解偏微分方程（PDE），广泛使用的Multigrid方法[3]将系统重构为在多个尺度上的子问题，其中每个子问题负责较粗尺度和较细尺度的残差解。Multigrid的替代方法是层次化基础预处理[44,45]，它依赖于表示两个尺度之间残差向量的变量。已经被证明[3,44,45]这些求解器比不知道解的残差性质的标准求解器收敛得更快。这些方法表明，良好的重构或预处理可以简化优化。

Shortcut Connections. Practices and theories that lead to shortcut connections [2, 33, 48] have been studied for a long time. An early practice of training multi-layer perceptrons (MLPs) is to add a linear layer connected from the network input to the output [33, 48]. In [43, 24], a few intermediate layers are directly connected to auxiliary classifiers for addressing vanishing/exploding gradients. The papers of [38, 37, 31, 46] propose methods for centering layer responses, gradients, and propagated errors, implemented by shortcut connections. In [43], an “ inception ” layer is composed of a shortcut branch and a few deeper branches.

快捷连接。导致快捷连接[2,33,48]的实践和理论已经被研究了很长时间。训练多层感知机（MLP）的早期实践是添加一个线性层来连接网络的输入和输出[33,48]。在[43,24]中，一些中间层直接连接到辅助分类器，用于解决梯度消失/爆炸。论文[38,37,31,46]提出了通过快捷连接实现层间响应，梯度和传播误差的方法。在[43]中，一个“ inception ”层由一个快捷分支和一些更深的分支组成。

Concurrent with our work, “ highway networks ” [41, 42] present shortcut connections with gating functions [15]. These gates are data-dependent and have parameters, in contrast to our identity shortcuts that are parameter-free. When a gated shortcut is “ closed ” (approaching zero), the layers in highway networks represent non-residual functions. On the contrary, our formulation always learns residual functions; our identity shortcuts are never closed, and all information is always passed through, with additional residual functions to be learned. In addition, highway networks have not demonstrated accuracy gains with extremely increased depth (e.g., over 100 layers).

和我们同时进行的工作，“ highway networks ” [41, 42]提出了门功能[15]的快捷连接。这些门是数据相关且有参数的，与我们不具有参数的恒等快捷连接相反。当门控快捷连接“ 关闭 ”（接近零）时，高速网络中的层表示非残差函数。相反，我们的公式总是学习残差函数；我们的恒等快捷连接永远不会关闭，所有的信息总是通过，还有额外的残差函数要学习。此外，高速网络还没有证实极度增加的深度（例如，超过100个层）带来的准确性收益。

3. Deep Residual Learning

3.1. Residual Learning

Let us consider $H(x)$ as an underlying mapping to be fit by a few stacked layers (not necessarily the entire net), with $x$ denoting the inputs to the first of these layers. If one hypothesizes 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., $H(x) − x$ (assuming that the input and output are of the same dimensions). So rather than expect stacked layers to approximate $H(x)$, we explicitly let these layers approximate a residual function $F(x) := H(x) − x$. The original function thus becomes $F(x) + x$. Although both forms should be able to asymptotically approximate the desired functions (as hypothesized), the ease of learning might be different.

3. 深度残差学习

3.1. 残差学习

我们考虑$H(x)$作为几个堆叠层（不必是整个网络）要拟合的基础映射，$x$表示这些层中第一层的输入。假设多个非线性层可以渐近地近似复杂函数，它等价于假设它们可以渐近地近似残差函数，即$H(x) − x$(假设输入输出是相同维度)。因此，我们明确让这些层近似参数函数 $F(x) := H(x) − x$，而不是期望堆叠层近似$H(x)$。因此原始函数变为$F(x) + x$。尽管两种形式应该都能渐近地近似要求的函数（如假设），但学习的难易程度可能是不同的。

This reformulation is motivated by the counterintuitive phenomena about the degradation problem (Fig. 1, left). As we discussed in the introduction, if the added layers can be constructed as identity mappings, a deeper model should have training error no greater than its shallower counterpart. 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.

关于退化问题的反直觉现象激发了这种重构（图1左）。正如我们在引言中讨论的那样，如果添加的层可以被构建为恒等映射，更深模型的训练误差应该不大于它对应的更浅版本。退化问题表明求解器通过多个非线性层来近似恒等映射可能有困难。通过残差学习的重构，如果恒等映射是最优的，求解器可能简单地将多个非线性连接的权重推向零来接近恒等映射。

In real cases, it is unlikely that identity mappings are optimal, but our reformulation may help to precondition the problem. If the optimal function is closer to an identity mapping than to a zero mapping, it should be easier for the solver to find the perturbations with reference to an identity mapping, than to learn the function as a new one. We show by experiments (Fig. 7) that the learned residual functions in general have small responses, suggesting that identity mappings provide reasonable preconditioning.

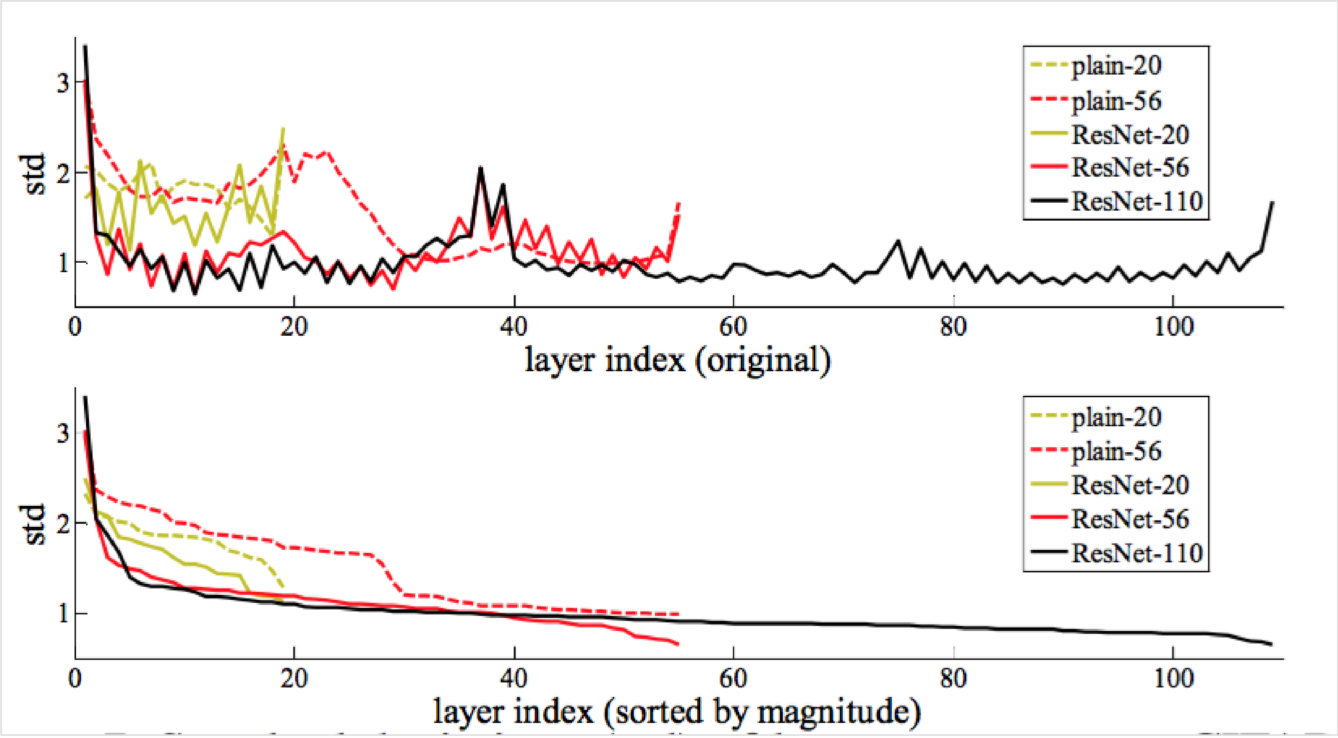


Figure 7. Standard deviations (std) of layer responses on CIFAR-10. The responses are the outputs of each 3×3 layer, after BN and before nonlinearity. Top: the layers are shown in their original order. Bottom: the responses are ranked in descending order.

在实际情况下，恒等映射不太可能是最优的，但是我们的重构可能有助于对问题进行预处理。如果最优函数比零映射更接近于恒等映射，则求解器应该更容易找到关于恒等映射的抖动，而不是将该函数作为新函数来学习。我们通过实验（图7）显示学习的残差函数通常有更小的响应，表明恒等映射提供了合理的预处理。

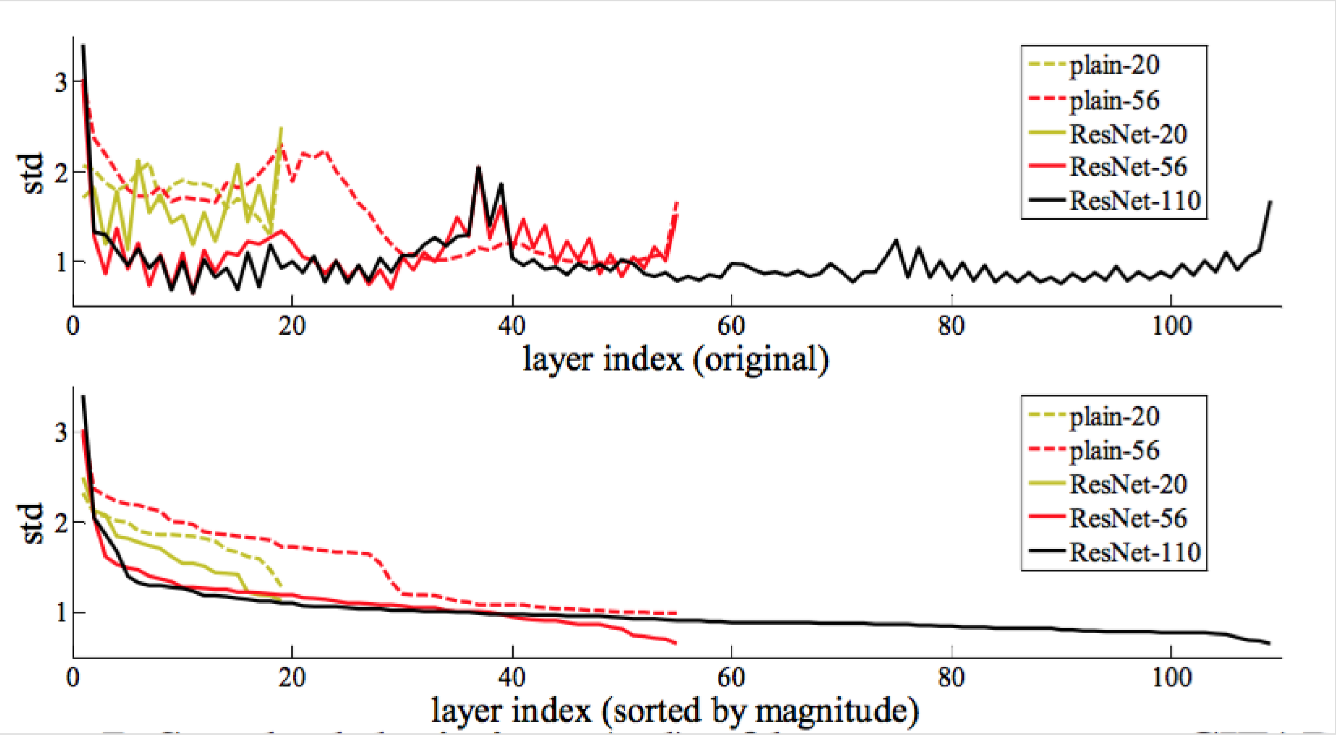


图7。层响应在CIFAR-10上的标准差（std）。这些响应是每个3×3层的输出，在BN之后非线性之前。上面：以原始顺序显示层。下面：响应按降序排列。

3.2. Identity Mapping by Shortcuts

We adopt residual learning to every few stacked layers. A building block is shown in Fig. 2. Formally, in this paper we consider a building block defined as:

$$y = F(x, {W_i}) + x$$ (1)

Here x and y are the input and output vectors of the layers considered. The function $F(x, {W_i})$ represents the residual mapping to be learned. For the example in Fig. 2 that has two layers, $F = W_2 \sigma(W_1x)$ in which σ denotes ReLU [29] and the biases are omitted for simplifying notations. The operation $F + x$ is performed by a shortcut connection and element-wise addition. We adopt the second nonlinearity after the addition (i.e., $\sigma(y)$, see Fig. 2).

3.2. 快捷恒等映射

我们每隔几个堆叠层采用残差学习。构建块如图2所示。在本文中我们考虑构建块正式定义为：

$$y = F(x, {W_i}) + x$$ (1)

x和y是考虑的层的输入和输出向量。函数$F(x, {W_i})$表示要学习的残差映射。图2中的例子有两层，$F = W_2 \sigma(W_1x)$中σ表示ReLU[29]，为了简化写法忽略偏置项。$F + x$操作通过快捷连接和各个元素相加来执行。在相加之后我们采纳了第二种非线性（即$\sigma(y)$，看图2）。

The shortcut connections in Eqn.(1) introduce neither extra parameter nor computation complexity. This is not only attractive in practice but also important in our comparisons between plain and residual networks. We can fairly compare plain/residual networks that simultaneously have the same number of parameters, depth, width, and computational cost (except for the negligible element-wise addition).

方程(1)中的快捷连接既没有引入外部参数又没有增加计算复杂度。这不仅在实践中有吸引力，而且在简单网络和残差网络的比较中也很重要。我们可以公平地比较同时具有相同数量的参数，相同深度，宽度和计算成本的简单/残差网络（除了不可忽略的元素加法之外）。

The dimensions of x and F must be equal in Eqn.(1). 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:

$$y = F(x, {W_i }) + W_sx.$$ (2)

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

方程(1)中x和F的维度必须是相等的。如果不是这种情况（例如，当更改输入/输出通道时），我们可以通过快捷连接执行线性投影W_s来匹配维度：

$$y = F(x, {W_i }) + W_sx.$$ (2)

我们也可以使用方程(1)中的方阵W_s。但是我们将通过实验表明，恒等映射足以解决退化问题，并且是合算的，因此W_s仅在匹配维度时使用。

The form of the residual function F is flexible. Experiments in this paper involve a function F that has two or three layers (Fig. 5), while more layers are possible. But if F has only a single layer, Eqn.(1) is similar to a linear layer: $y = W_1x + x$, for which we have not observed advantages.

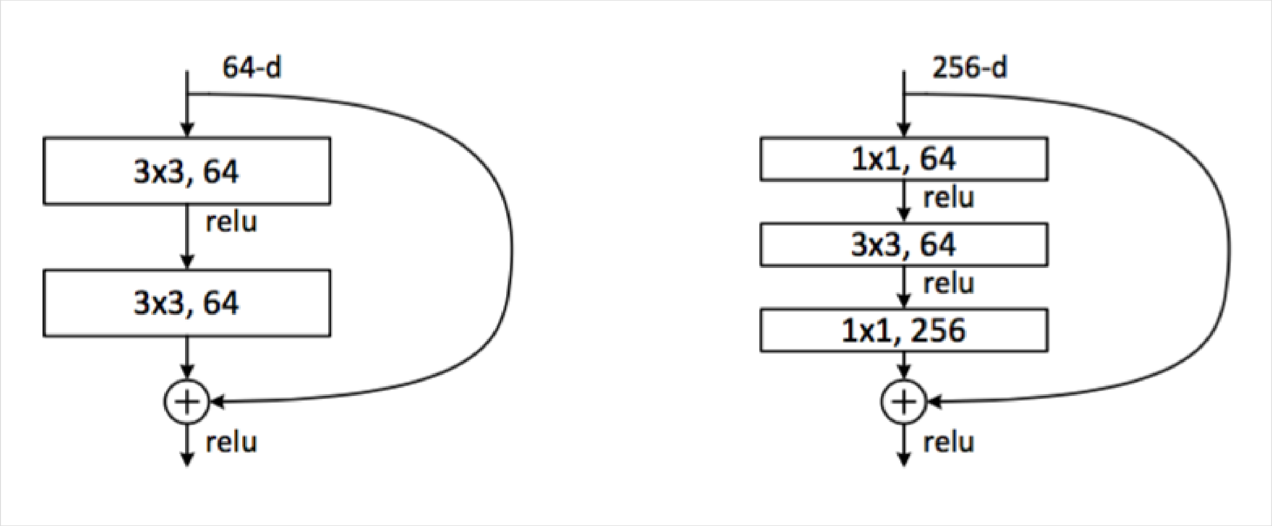


Figure 5. A deeper residual function F for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a “ bottleneck ” building block for ResNet-50/101/152.

残差函数F的形式是可变的。本文中的实验包括有两层或三层（图5）的函数F，同时可能有更多的层。但如果F只有一层，方程(1)类似于线性层：$y = W_1x + x$，我们没有看到优势。

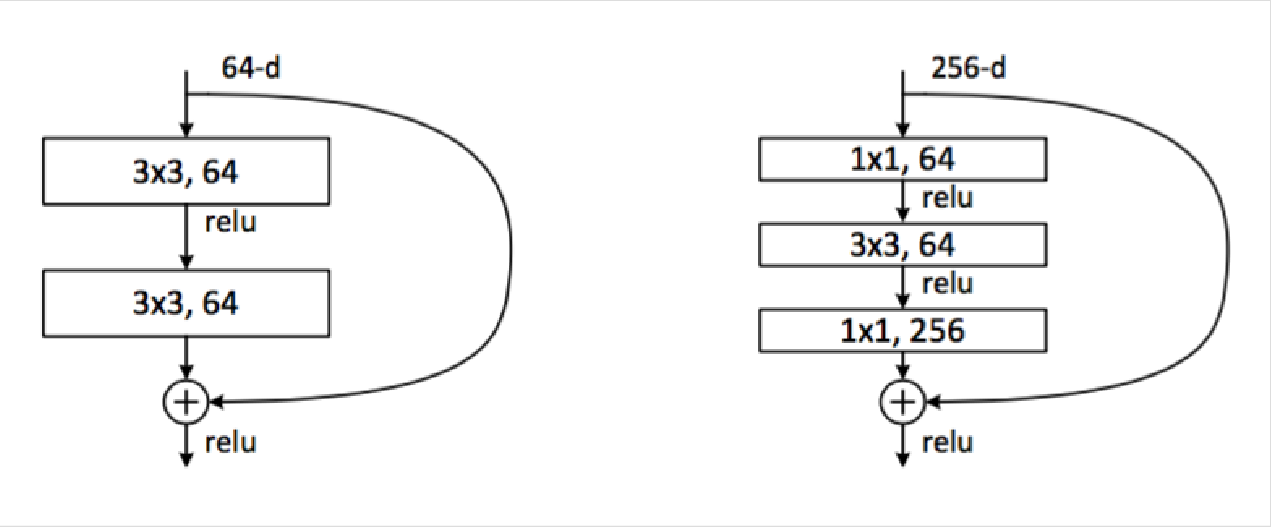


图5。ImageNet的深度残差函数F。左：ResNet-34的构建块（在56×56的特征图上），如图3。右：ResNet-50/101/152的 “ bottleneck ” 构建块。

We also note that although the above notations are about fully-connected layers for simplicity, they are applicable to convolutional layers. The function $F(x,{W_i})$ can represent multiple convolutional layers. The element-wise addition is performed on two feature maps, channel by channel.

我们还注意到，为了简单起见，尽管上述符号是关于全连接层的，但它们同样适用于卷积层。函数$F(x，{W_i})$可以表示多个卷积层。元素加法在两个特征图上逐通道进行。

3.3. Network Architectures

We have tested various plain/residual nets, and have observed consistent phenomena. To provide instances for discussion, we describe two models for ImageNet as follows.

3.3. 网络架构

我们测试了各种简单/残差网络，并观察到了一致的现象。为了提供讨论的实例，我们描述了ImageNet的两个模型如下。

Plain Network. Our plain baselines (Fig. 3, middle) are mainly inspired by the philosophy of VGG nets 40. The convolutional layers mostly have 3×3 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer. We perform downsampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax. The total number of weighted layers is 34 in Fig. 3 (middle).

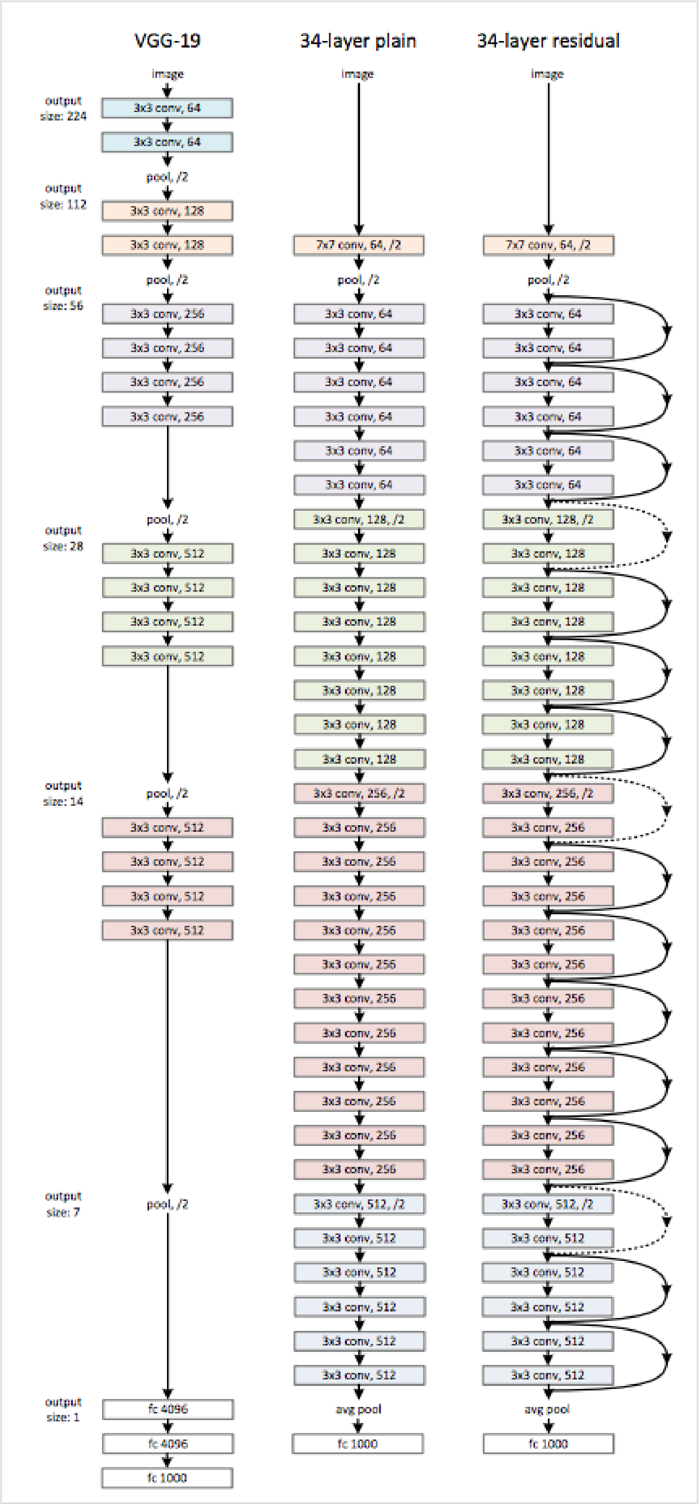
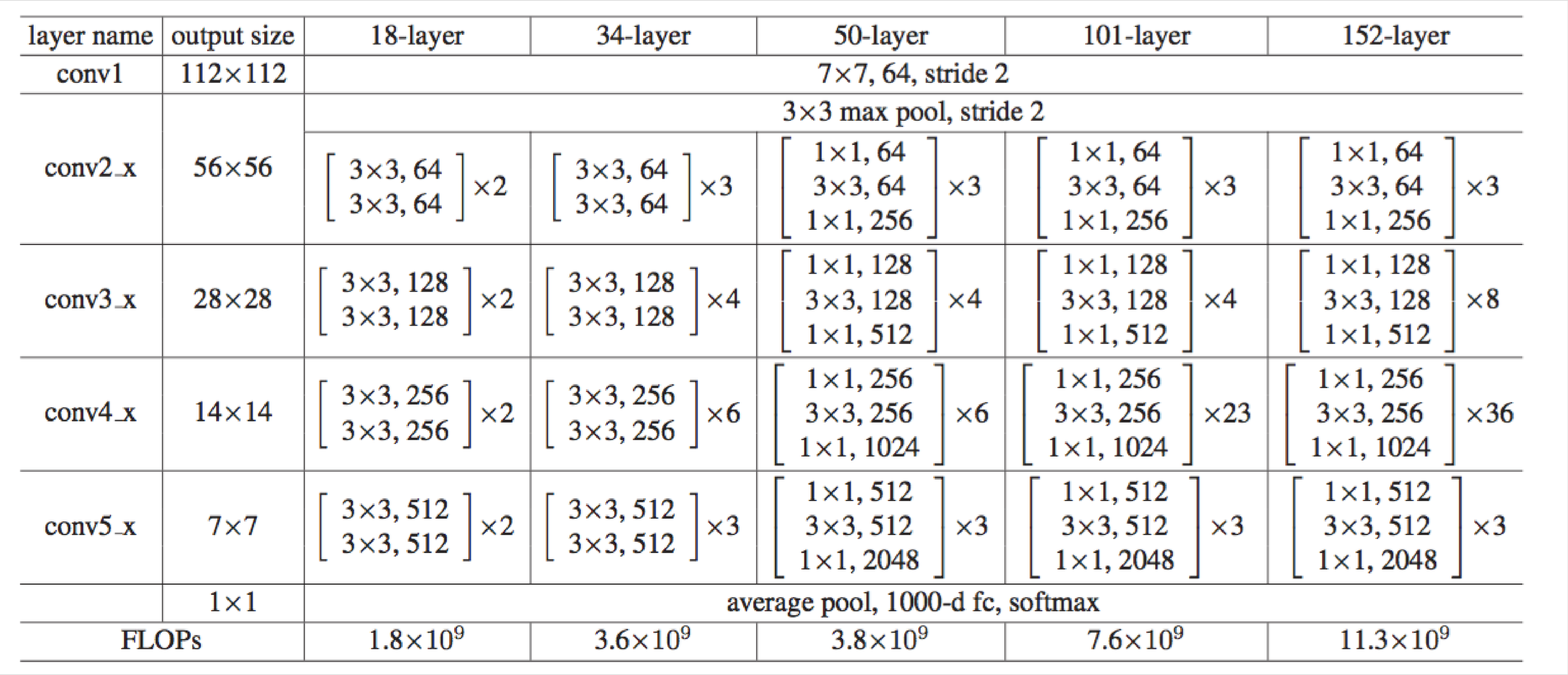

VGG-19
34-layer plain
34-layer residual
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1
image
3x3 conv, 64
3x3 conv, 64
pool, /2
3x3 conv, 128
3x3 conv, 128
pool, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
fc 4096
fc 4096
fc 1000
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000

Figure 3. Example network architectures for ImageNet. Left: the VGG-19 model [40] (19.6 billion FLOPs) as a reference. Middle: a plain network with 34 parameter layers (3.6 billion FLOPs). Right: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. Table 1 shows more details and other variants.



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.

简单网络。 我们简单网络的基准（图3，中间）主要受到VGG网络[40]（图3，左图）的哲学启发。卷积层主要有3×3的滤波器，并遵循两个简单的设计规则：（i）对于相同的输出特征图尺寸，层具有相同数量的滤波器；（ii）如果特征图尺寸减半，则滤波器数量加倍，以便保持每层的时间复杂度。我们通过步长为2的卷积层直接执行下采样。网络以全局平均池化层和具有softmax的1000维全连接层结束。图3（中间）的加权层总数为34。

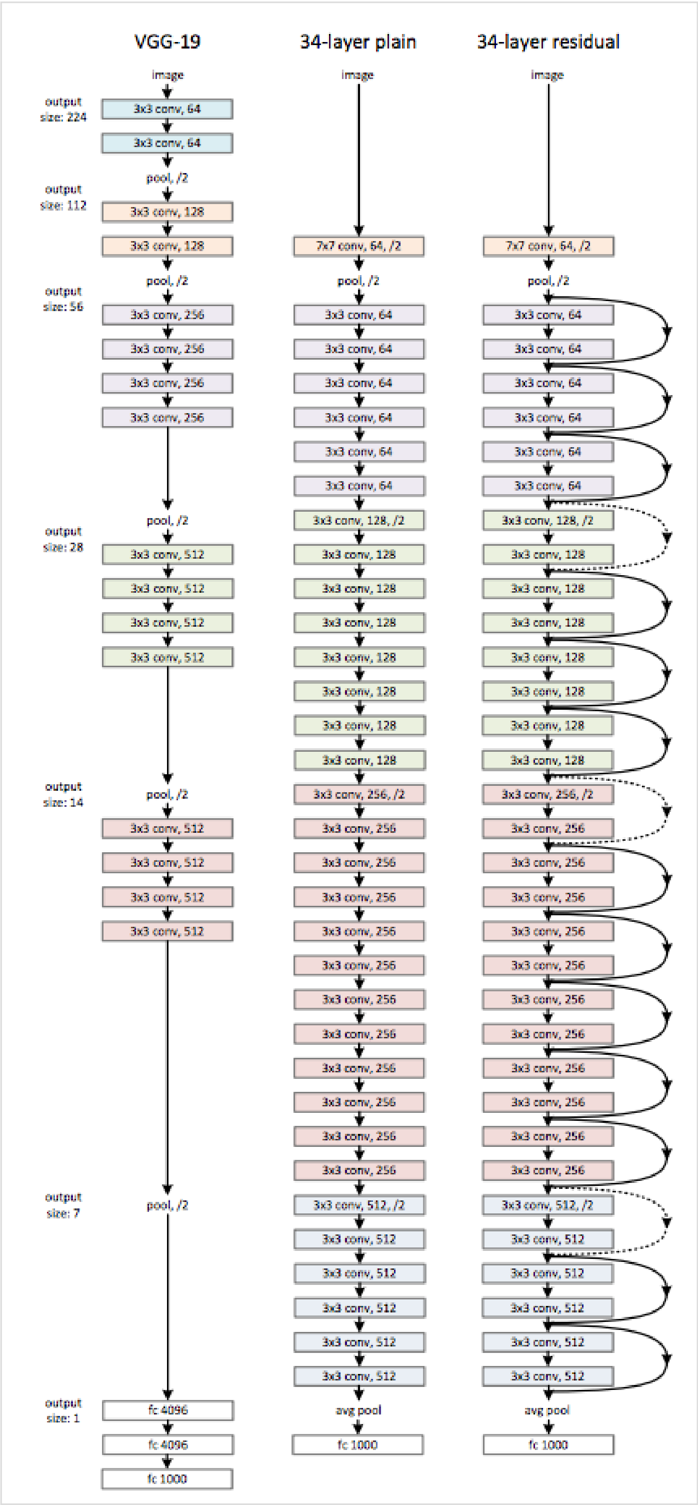
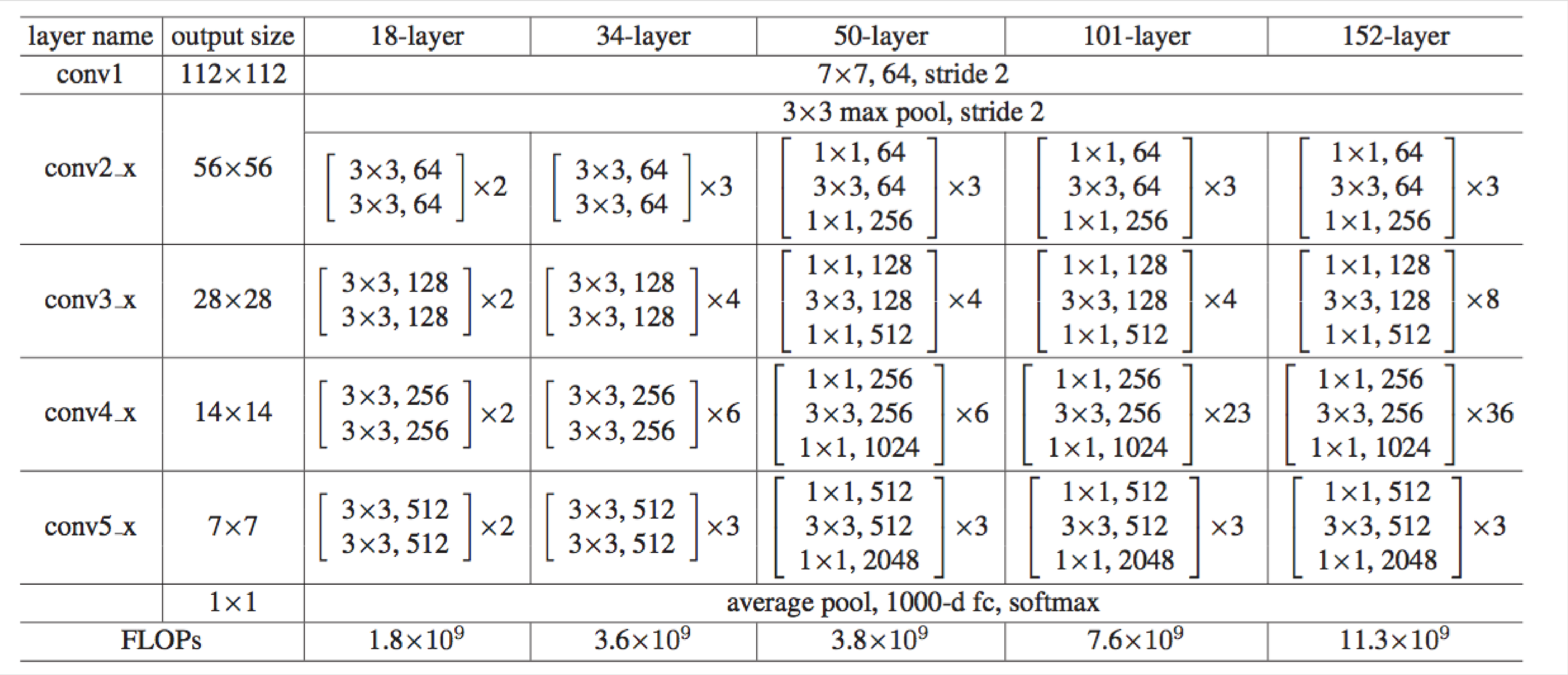

VGG-19
34-layer plain
34-layer residual
image
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1
3x3 conv, 64
3x3 conv, 64
pool, /2
3x3 conv, 128
3x3 conv, 128
pool, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
fc 4096
fc 4096
fc 1000
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000

图3。ImageNet的网络架构例子。左：作为参考的VGG-19模型40。中：具有34个参数层的简单网络（36亿FLOPs）。右：具有34个参数层的残差网络（36亿FLOPs）。带点的快捷连接增加了维度。表1显示了更多细节和其它变种。



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

表1。ImageNet架构。构建块显示在括号中（也可看图5），以及构建块的堆叠数量。下采样通过步长为2的conv3_1, conv4_1和conv5_1执行。

It is worth noticing that our model has *fewer* filters and *lower* complexity than VGG nets 40. Our 34-layer baseline has 3.6 billion FLOPs (multiply-adds), which is only 18% of VGG-19 (19.6 billion FLOPs).

值得注意的是我们的模型与VGG网络（图3左）相比，有更少的滤波器和更低的复杂度。我们的34层基准有36亿FLOP(乘加)，仅是VGG-19（196亿FLOP）的18%。

Residual Network. Based on the above plain network, we insert shortcut connections (Fig. 3, right) which turn the network into its counterpart residual version. The identity shortcuts (Eqn.(1)) can be directly used when the input and output are of the same dimensions (solid line shortcuts in Fig. 3). When the dimensions increase (dotted line shortcuts in Fig. 3), we consider two options: (A) The shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions. This option introduces no extra parameter; (B) The projection shortcut in Eqn.(2) is used to match dimensions (done by 1×1 convolutions). For both options, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2.

残差网络。 基于上述的简单网络，我们插入快捷连接（图3，右），将网络转换为其对应的残差版本。当输入和输出具有相同的维度时（图3中的实线快捷连接）时，可以直接使用恒等快捷连接（方程（1））。当维度增加（图3中的虚线快捷连接）时，我们考虑两个选项：（A）快捷连接仍然执行恒等映射，额外填充零输入以增加维度。此选项不会引入额外的参数；（B）方程（2）中的投影快捷连接用于匹配维度（由1×1卷积完成）。对于这两个选项，当快捷连接跨越两种尺寸的特征图时，它们执行时步长为2。

3.4. Implementation

Our implementation for ImageNet follows the practice in [21, 40]. The image is resized with its shorter side randomly sampled in [256, 480] for scale augmentation [40]. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted [21]. The standard color augmentation in [21] is used. We adopt batch normalization (BN) [16] right after each convolution and before activation, following [16]. We initialize the weights as in [12] and train all plain/residual nets from scratch. We use SGD with a mini-batch size of 256. The learning rate starts from 0.1 and is divided by 10 when the error plateaus, and the models are trained for up to $60 × 10^4$ iterations. We use a weight decay of 0.0001 and a momentum of 0.9. We do not use dropout [13], following the practice in [16].

3.4. 实现

ImageNet中我们的实现遵循[21，40]的实践。调整图像大小，其较短的边在[256,480]之间进行随机采样，用于尺度增强[40]。224×224裁剪是从图像或其水平翻转中随机采样，并逐像素减去均值[21]。使用了[21]中的标准颜色增强。在每个卷积之后和激活之前，我们采用批量归一化（BN）[16]。我们按照[12]的方法初始化权重，从零开始训练所有的简单/残差网络。我们使用批大小为256的SGD方法。学习速度从0.1开始，当误差稳定时学习率除以10，并且模型训练高达$60 × 10^4$次迭代。我们使用的权重衰减为0.0001，动量为0.9。根据[16]的实践，我们不使用丢弃[13]。

In testing, for comparison studies we adopt the standard 10-crop testing [21]. For best results, we adopt the fully-convolutional form as in [40, 12], and average the scores at multiple scales (images are resized such that the shorter side is in {224, 256, 384, 480, 640}).

在测试阶段，为了比较学习我们采用标准的10-crop测试[21]。对于最好的结果，我们采用如[40, 12]中的全卷积形式，并在多尺度上对分数进行平均（图像归一化，短边位于{224, 256, 384, 480, 640}中）。

4. Experiments

4.1. ImageNet Classification

We evaluate our method on the ImageNet 2012 classification dataset [35] that consists of 1000 classes. The models are trained on the 1.28 million training images, and evaluated on the 50k validation images. We also obtain a final result on the 100k test images, reported by the test server. We evaluate both top-1 and top-5 error rates.

4. 实验

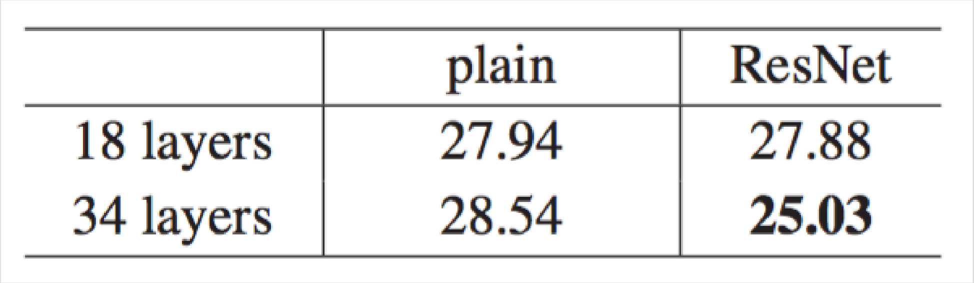
4.1. ImageNet分类

我们在ImageNet 2012分类数据集[35]对我们的方法进行了评估，该数据集由1000个类别组成。这些模型在128万张训练图像上进行训练，并在5万张验证图像上进行评估。我们也获得了测试服务器报告的在10万张测试图像上的最终结果。我们评估了top-1和top-5错误率。

Plain Networks. We first evaluate 18-layer and 34-layer plain nets. The 34-layer plain net is in Fig. 3 (middle). The 18-layer plain net is of a similar form. See Table 1 for detailed architectures.

简单网络。我们首先评估18层和34层的简单网络。34层简单网络在图3（中间）。18层简单网络是一种类似的形式。有关详细的体系结构，请参见表1。

The results in Table 2 show that the deeper 34-layer plain net has higher validation error than the shallower 18-layer plain net. To reveal the reasons, in Fig. 4 (left) we compare their training/validation errors during the training procedure. We have observed the degradation problem —— the 34-layer plain net has higher training error throughout the whole training procedure, even though the solution space of the 18-layer plain network is a subspace of that of the 34-layer one.



	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.

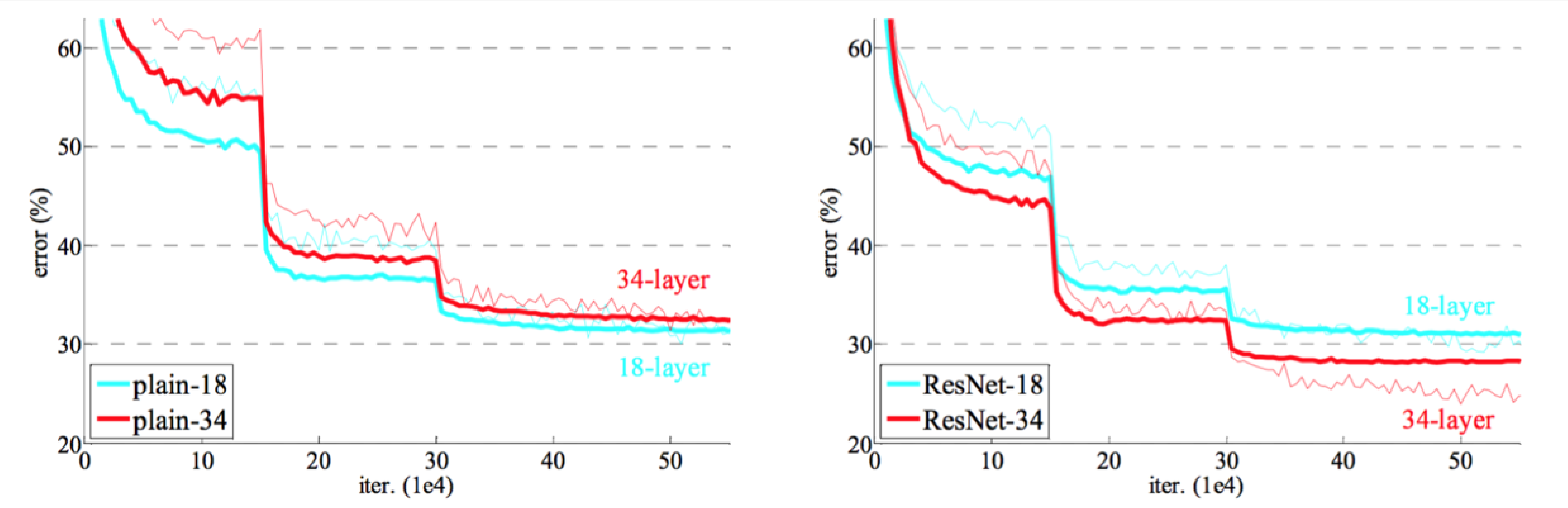
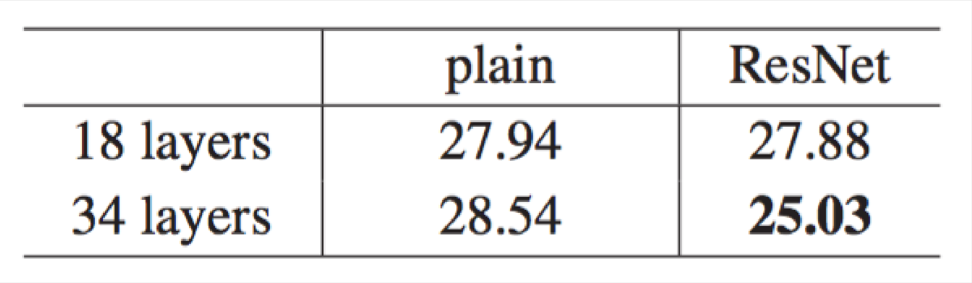


Figure 4. Training on ImageNet. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

表2中的结果表明，较深的34层简单网络比较浅的18层简单网络有更高的验证误差。为了揭示原因，在图4（左图）中，我们比较训练过程中的训练/验证误差。我们观察到退化问题——虽然18层简单网络的解空间是34层简单网络解空间的子空间，但34层简单网络在整个训练过程中具有较高的训练误差。



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

表2。ImageNet验证集上的Top-1错误率(%，10个裁剪图像测试)。相比于对应的简单网络，ResNet没有额外的参数。图4显示了训练过程。

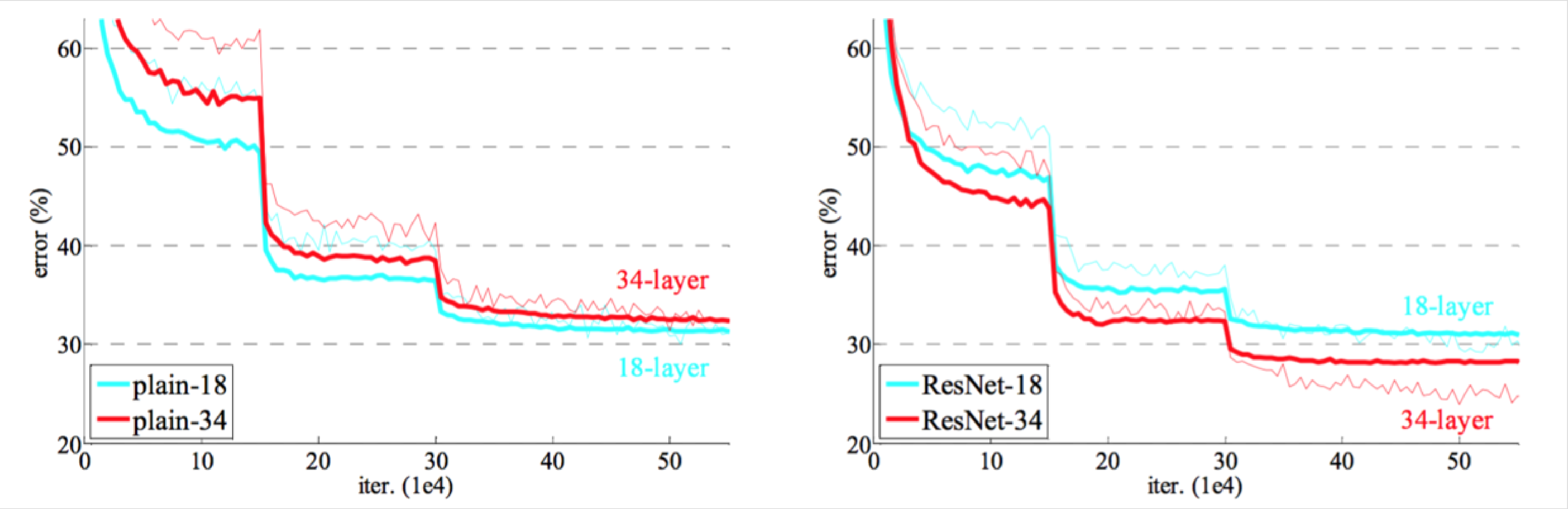
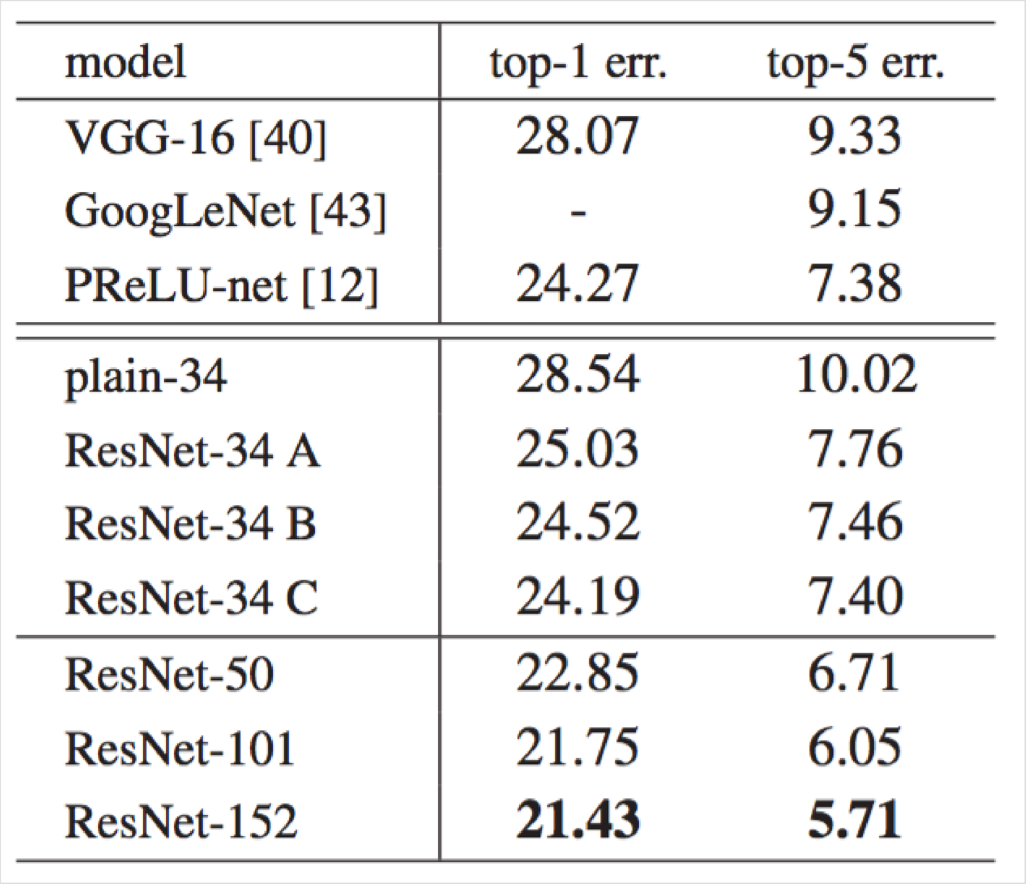


图4。在ImageNet上训练。细曲线表示训练误差，粗曲线表示中心裁剪图像的验证误差。左：18层和34层的简单网络。右：18层和34层的ResNet。在本图中，残差网络与对应的简单网络相比没有额外的参数。

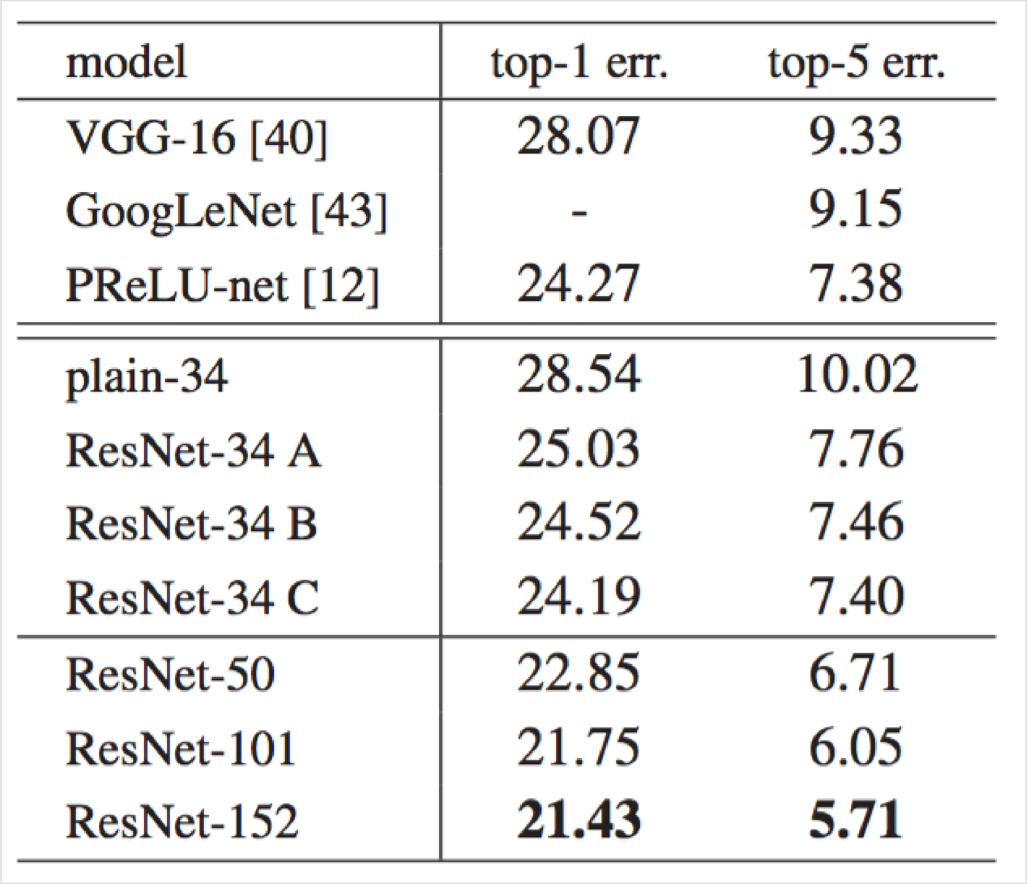
We argue that this optimization difficulty is unlikely to be caused by vanishing gradients. These plain networks are trained with BN [16], which ensures forward propagated signals to have non-zero variances. We also verify that the backward propagated gradients exhibit healthy norms with BN. So neither forward nor backward signals vanish. In fact, the 34-layer plain net is still able to achieve competitive accuracy (Table 3), suggesting that the solver works to some extent. We conjecture that the deep plain nets may have exponentially low convergence rates, which impact the reducing of the training error. The reason for such optimization difficulties will be studied in the future.



model	top-1 err.	top-5 err.
VGG-16 [40]	28.07	9.33
GoogLeNet [43]	-	9.15
PReLU-net [12]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. Error rates (%, 10-crop testing) on ImageNet validation. VGG-16 is based on our test. ResNet-50/101/152 are of option B that only uses projections for increasing dimensions.

我们认为这种优化难度不可能是由于梯度消失引起的。这些简单网络使用BN[16]训练，这保证了前向传播信号有非零方差。我们还验证了反向传播的梯度，结果显示其符合BN的正常标准。因此既不是前向信号消失也不是反向信号消失。实际上，34层简单网络仍能取得有竞争力的准确率（表3），这表明在某种程度上来说求解器仍工作。我们推测深度简单网络可能有指数级低收敛特性，这影响了训练误差的降低。这种优化困难的原因将来会研究。



model	top-1 err.	top-5 err.
VGG-16 [40]	28.07	9.33
GoogLeNet [43]	-	9.15
PReLU-net [12]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

表3。ImageNet验证集错误率（%，10个裁剪图像测试）。VGG16是基于我们的测试结果的。ResNet-50/101/152的选择B仅使用投影增加维度。

Residual Networks. Next we evaluate 18-layer and 34-layer residual nets (ResNets). The baseline architectures are the same as the above plain nets, expect that a shortcut connection is added to each pair of 3×3 filters as in Fig. 3 (right). In the first comparison (Table 2 and Fig. 4 right), we use identity mapping for all shortcuts and zero-padding for increasing dimensions (option A). So they have no extra parameter compared to the plain counterparts.

残差网络。接下来我们评估18层和34层残差网络（ResNets）。基准架构与上述的简单网络相同，如图3（右）所示，预计每对3×3滤波器都会添加快捷连接。在第一次比较（表2和图4右侧）中，我们对所有快捷连接都使用恒等映射和零填充以增加维度（选项A）。所以与对应的简单网络相比，它们没有额外的参数。

We have three major observations from Table 2 and Fig. 4. First, the situation is reversed with residual learning —— the 34-layer ResNet is better than the 18-layer ResNet (by 2.8%). More importantly, the 34-layer ResNet exhibits considerably lower training error and is generalizable to the validation data. This indicates that the degradation problem is well addressed in this setting and we manage to obtain accuracy gains from increased depth.

我们从表2和图4中可以看到三个主要的观察结果。首先，残留学习的情况变了——34层ResNet比18层ResNet更好（2.8%）。更重要的是，34层ResNet显示出较低的训练误差，并且可以泛化到验证数据。这表明在这种情况下，退化问题得到了很好的解决，我们从增加的深度中设法获得了准确性收益。

Second, compared to its plain counterpart, the 34-layer ResNet reduces the top-1 error by 3.5% (Table 2), resulting from the successfully reduced training error (Fig. 4 right vs. left). This comparison verifies the effectiveness of residual learning on extremely deep systems.

第二，与对应的简单网络相比，由于成功的减少了训练误差，34层ResNet降低了3.5%的top-1错误率。这种比较证实了在极深系统中残差学习的有效性。

Last, we also note that the 18-layer plain/residual nets are comparably accurate (Table 2), but the 18-layer ResNet converges faster (Fig. 4 right vs. left). When the net is “ not overly deep ” (18 layers here), the current SGD solver is still able to find good solutions to the plain net. In this case, the ResNet eases the optimization by providing faster convergence at the early stage.

最后，我们还注意到18层的简单/残差网络同样地准确（表2），但18层ResNet收敛更快（图4右和左）。当网络“ 不过度深 ”时（18层），目前的SGD求解器仍能在简单网络中找到好的解。在这种情况下，ResNet通过在早期提供更快的收敛简便了优化。

Identity vs. Projection Shortcuts. We have shown that parameter-free, identity shortcuts help with training. Next we investigate projection shortcuts (Eqn.(2)). In Table 3 we compare three options: (A) zero-padding shortcuts are used for increasing dimensions, and all shortcuts are parameter-free (the same as Table 2 and Fig. 4 right); (B) projection shortcuts are used for increasing dimensions, and other shortcuts are identity; and (C) all shortcuts are projections.

恒等和投影快捷连接我们已经表明没有参数，恒等快捷连接有助于训练。接下来我们调查投影快捷连接（方程2）。在表3中我们比较了三个选项：(A) 零填充快捷连接用来增加维度，所有的快捷连接是没有参数的（与表2和图4右相同）；(B)投影快捷连接用来增加维度，其它的快捷连接是恒等的；（C）所有的快捷连接都是投影。

Table 3 shows that all three options are considerably better than the plain counterpart. B is slightly better than A. We argue that this is because the zero-padded dimensions in A indeed have no residual learning. C is marginally better than B, and we attribute this to the extra parameters introduced by many (thirteen) projection shortcuts. But the small differences among A/B/C indicate that projection shortcuts are not essential for addressing the degradation problem. So we do not use option C in the rest of this paper, to reduce memory/time complexity and model sizes. Identity shortcuts are particularly important for not increasing the complexity of the bottleneck architectures that are introduced below.

表3显示，所有三个选项都比对应的简单网络好很多。选项B比A略好。我们认为这是因为A中的零填充确实没有残差学习。选项C比B稍好，我们把这归因于许多（十三）投影快捷连接引入了额外参数。但A/B/C之间的细微差异表明，投影快捷连接对于解决退化问题不是至关重要的。因为我们在本文的剩余部分不再使用选项C，以减少内存/时间复杂性和模型大小。恒等快捷连接对于不增加下面介绍的瓶颈结构的复杂性尤为重要。

Deeper Bottleneck Architectures. Next we describe our deeper nets for ImageNet. Because of concerns on the training time that we can afford, we modify the building block as a bottleneck design. For each residual function F , we use a stack of 3 layers instead of 2 (Fig. 5). The three layers are 1×1, 3×3, and 1×1 convolutions, where the 1×1 layers are responsible for reducing and then increasing (restoring) dimensions, leaving the 3×3 layer a bottleneck with smaller input/output dimensions. Fig. 5 shows an example, where both designs have similar time complexity.

更深的瓶颈结构。接下来我们描述ImageNet中我们使用的更深的网络网络。由于关注我们能承受的训练时间，我们将构建块修改为瓶颈设计。对于每个残差函数F，我们使用3层堆叠而不是2层（图5）。三层是1×1，3×3和1×1卷积，其中1×1层负责减小然后增加（恢复）维度，使3×3层成为具有较小输入/输出维度的瓶颈。图5展示了一个示例，两个设计具有相似的时间复杂度。

The parameter-free identity shortcuts are particularly important for the bottleneck architectures. If the identity shortcut in Fig. 5 (right) is replaced with projection, one can show that the time complexity and model size are doubled, as the shortcut is connected to the two high-dimensional ends. So identity shortcuts lead to more efficient models for the bottleneck designs.

无参数恒等快捷连接对于瓶颈架构尤为重要。如果图5（右）中的恒等快捷连接被投影替换，则可以显示出时间复杂度和模型大小加倍，因为快捷连接是连接到两个高维端。因此，恒等快捷连接可以为瓶颈设计得到更有效的模型。

50-layer ResNet: We replace each 2-layer block in the 34-layer net with this 3-layer bottleneck block, resulting in a 50-layer ResNet (Table 1). We use option B for increasing dimensions. This model has 3.8 billion FLOPs.

50层ResNet：我们用3层瓶颈块替换34层网络中的每一个2层块，得到了一个50层ResNet（表1）。我们使用选项B来增加维度。该模型有38亿FLOP。

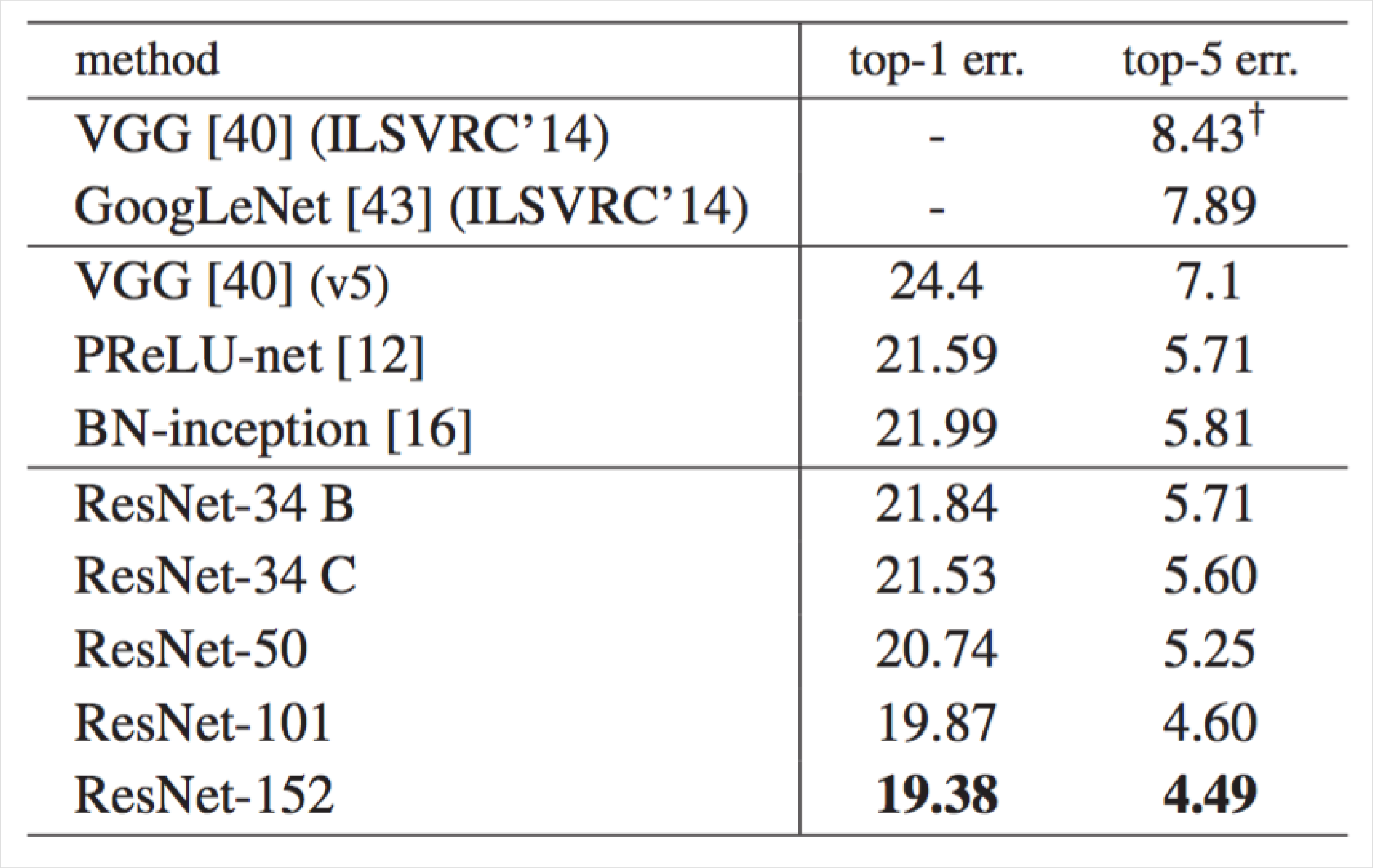
101-layer and 152-layer ResNet: We construct 101-layer and 152-layer ResNets by using more 3-layer blocks (Table 1). Remarkably, although the depth is significantly increased, the 152-layer ResNet (11.3 billion FLOPs) still has lower complexity than VGG-16/19 nets (15.3/19.6 billion FLOPs).

101层和152层ResNet：我们通过使用更多的3层瓶颈块来构建101层和152层ResNets（表1）。值得注意的是，尽管深度显著增加，但152层ResNet（113亿FLOP）仍然比VGG-16/19网络（153/196亿FLOP）具有更低的复杂度。

The 50/101/152-layer ResNets are more accurate than the 34-layer ones by considerable margins (Table 3 and 4). We do not observe the degradation problem and thus enjoy significant accuracy gains from considerably increased depth. The benefits of depth are witnessed for all evaluation metrics (Table 3 and 4).

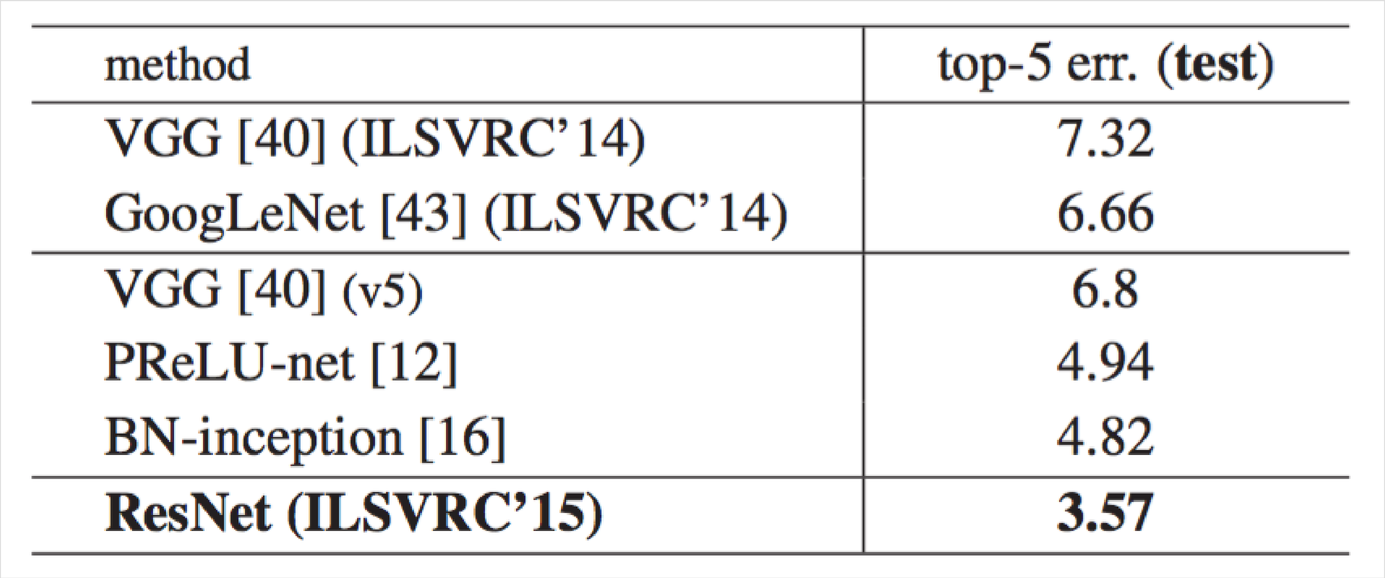
50/101/152层ResNet比34层ResNet的准确性要高得多（表3和4）。我们没有观察到退化问题，因此可以从显著增加的深度中获得显著的准确性收益。所有评估指标都能证明深度的收益（表3和表4）。

Comparisons with State-of-the-art Methods. In Table 4 we compare with the previous best single-model results. Our baseline 34-layer ResNets have achieved very competitive accuracy. Our 152-layer ResNet has a single-model top-5 validation error of 4.49%. This single-model result outperforms all previous ensemble results (Table 5). We combine six models of different depth to form an ensemble (only with two 152-layer ones at the time of submitting). This leads to 3.57% top-5 error on the test set (Table 5). This entry won the 1st place in ILSVRC 2015.



method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43†
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**

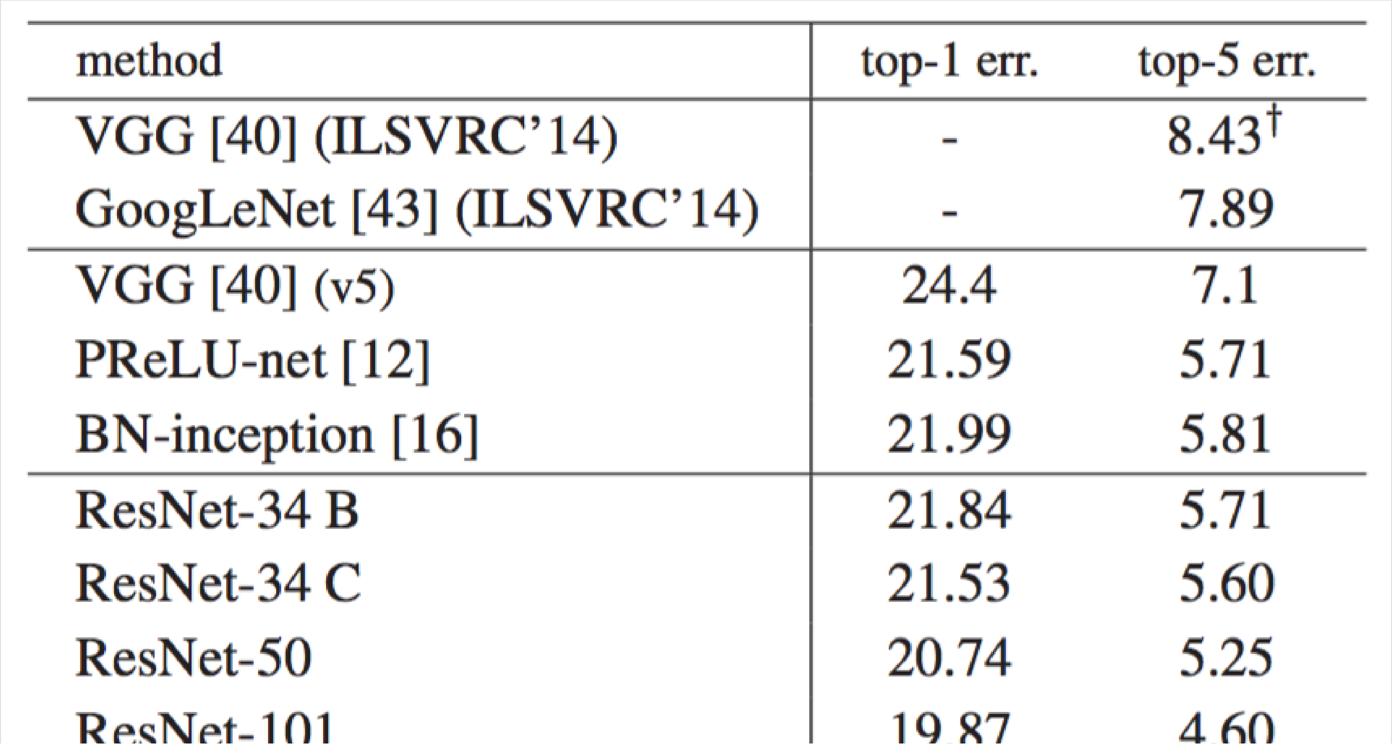
Table 4. Error rates (%) of single-model results on the ImageNet validation set (except reported on the test set).



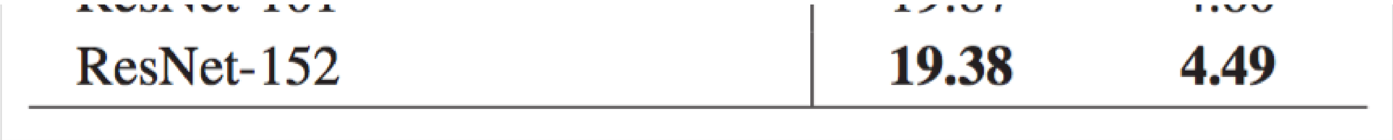
method	top-5 err. (**test**)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. Error rates (%) of ensembles. The top-5 error is on the test set of ImageNet and reported by the test server.

与最先进的方法比较。在表4中，我们与以前最好的单一模型结果进行比较。我们基准的34层ResNet已经取得了非常有竞争力的准确性。我们的152层ResNet具有单模型4.49%的top-5错误率。这种单一模型的结果胜过以前的所有综合结果（表5）。我们结合了六种不同深度的模型，形成一个集合（在提交时仅有两个152层）。这在测试集上得到了3.5%的top-5错误率（表5）。这次提交在2015年ILSVRC中荣获了第一名。

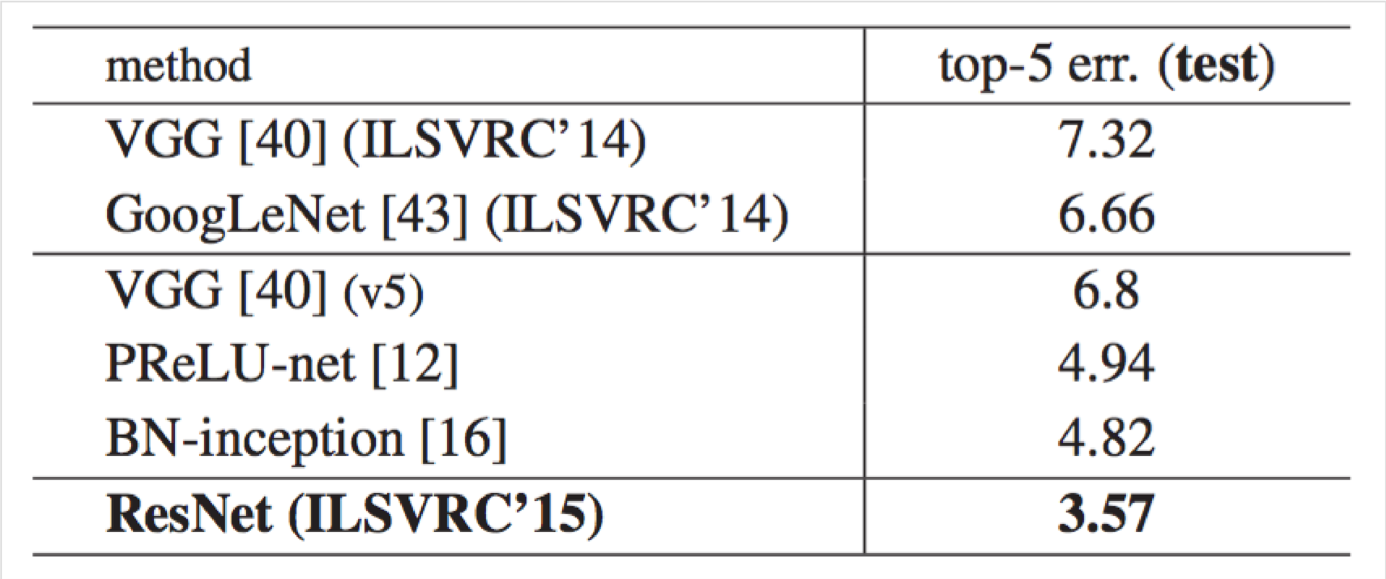


method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43†
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60



ResNet-152	**19.38**	**4.49**

表4。单一模型在ImageNet验证集上的错误率（%）(除了†是测试集上报告的错误率)。



method	top-5 err. (**test**)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

表5。模型综合的错误率(%)。top-5错误率是ImageNet测试集上的并由测试服务器报告的。

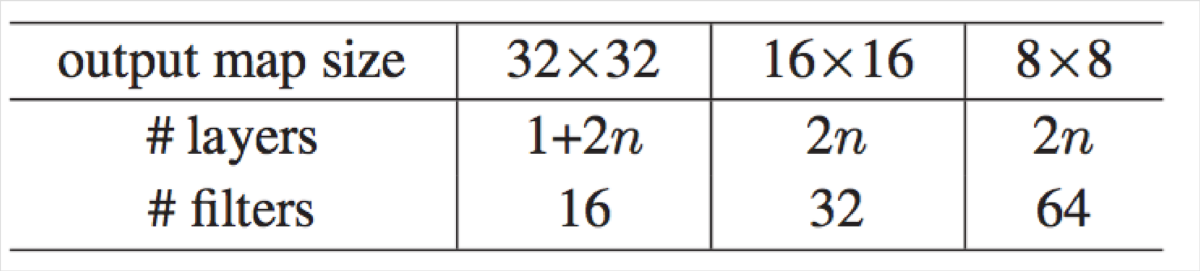
4.2. CIFAR-10 and Analysis

We conducted more studies on the CIFAR-10 dataset [20], which consists of 50k training images and 10k testing images in 10 classes. We present experiments trained on the training set and evaluated on the test set. Our focus is on the behaviors of extremely deep networks, but not on pushing the state-of-the-art results, so we intentionally use simple architectures as follows.

4.2. CIFAR-10和分析

我们对CIFAR-10数据集[20]进行了更多的研究，其中包括10个类别中的5万张训练图像和1万张测试图像。我们介绍了在训练集上进行训练和在测试集上进行评估的实验。我们的焦点在于极深网络的行为，但不是推动最先进的结果，所以我们有意使用如下的简单架构。

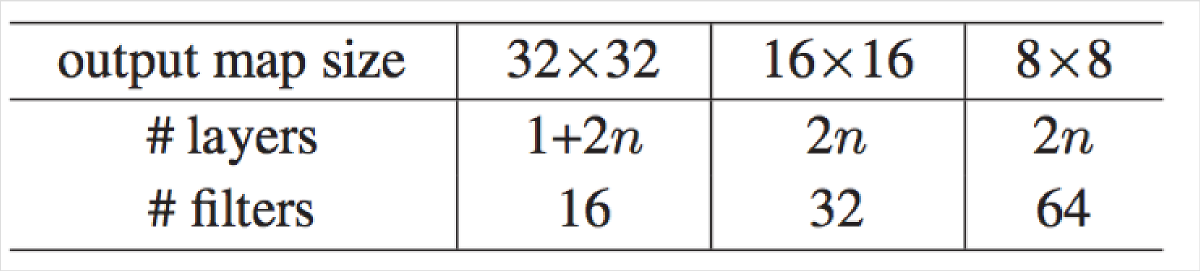
The plain/residual architectures follow the form in Fig. 3 (middle/right). The network inputs are 32×32 images, with the per-pixel mean subtracted. The first layer is 3×3 convolutions. Then we use a stack of 6n layers with 3×3 convolutions on the feature maps of sizes {32, 16, 8} respectively, with 2n layers for each feature map size. The numbers of filters are {16, 32, 64} respectively. The subsampling is performed by convolutions with a stride of 2. The network ends with a global average pooling, a 10-way fully-connected layer, and softmax. There are totally 6n+2 stacked weighted layers. The following table summarizes the architecture:



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

When shortcut connections are used, they are connected to the pairs of 3×3 layers (totally 3n shortcuts). On this dataset we use identity shortcuts in all cases (i.e., option A), so our residual models have exactly the same depth, width, and number of parameters as the plain counterparts.

简单/残差架构遵循图3（中/右）的形式。网络输入是32×32的图像，每个像素减去均值。第一层是3×3卷积。然后我们在大小为{32,16,8}的特征图上分别使用了带有3×3卷积的6n个堆叠层，每个特征图大小使用2n层。滤波器数量分别为{16,32,64}。下采样由步长为2的卷积进行。网络以全局平均池化，一个10维全连接层和softmax作为结束。共有6n+2个堆叠的加权层。下表总结了这个架构：



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

当使用快捷连接时，它们连接到成对的3×3卷积层上（共3n个快捷连接）。在这个数据集上，我们在所有案例中都使用恒等快捷连接（即选项A），因此我们的残差模型与对应的简单模型具有完全相同的深度，宽度和参数数量。

We use a weight decay of 0.0001 and momentum of 0.9, and adopt the weight initialization in [12] and BN [16] but with no dropout. These models are trained with a mini-batch size of 128 on two GPUs. We start with a learning rate of 0.1, divide it by 10 at 32k and 48k iterations, and terminate training at 64k iterations, which is determined on a 45k/5k train/val split. We follow the simple data augmentation in [24] for training: 4 pixels are padded on each side, and a 32×32 crop is randomly sampled from the padded image or its horizontal flip. For testing, we only evaluate the single view of the original 32×32 image.

我们使用的权重衰减为0.0001和动量为0.9，并采用[12]和BN[16]中的权重初始化，但没有使用丢弃。这些模型在两个GPU上进行训练，批处理大小为128。我们开始使用的学习率为0.1，在32k次和48k次迭代后学习率除以10，并在64k次迭代后终止训练，这是由45k/5k的训练/验证集分割决定的。我们按照[24]中的简单数据增强进行训练：每边填充4个像素，并从填充图像或其水平翻转图像中随机采样32×32的裁剪图像。对于测试，我们只评估原始32×32图像的单一视图。

We compare $n = {3, 5, 7, 9}$, leading to 20, 32, 44, and 56-layer networks. Fig. 6 (left) shows the behaviors of the plain nets. The deep plain nets suffer from increased depth, and exhibit higher training error when going deeper. This phenomenon is similar to that on ImageNet (Fig. 4, left) and on MNIST (see [41]), suggesting that such an optimization difficulty is a fundamental problem.

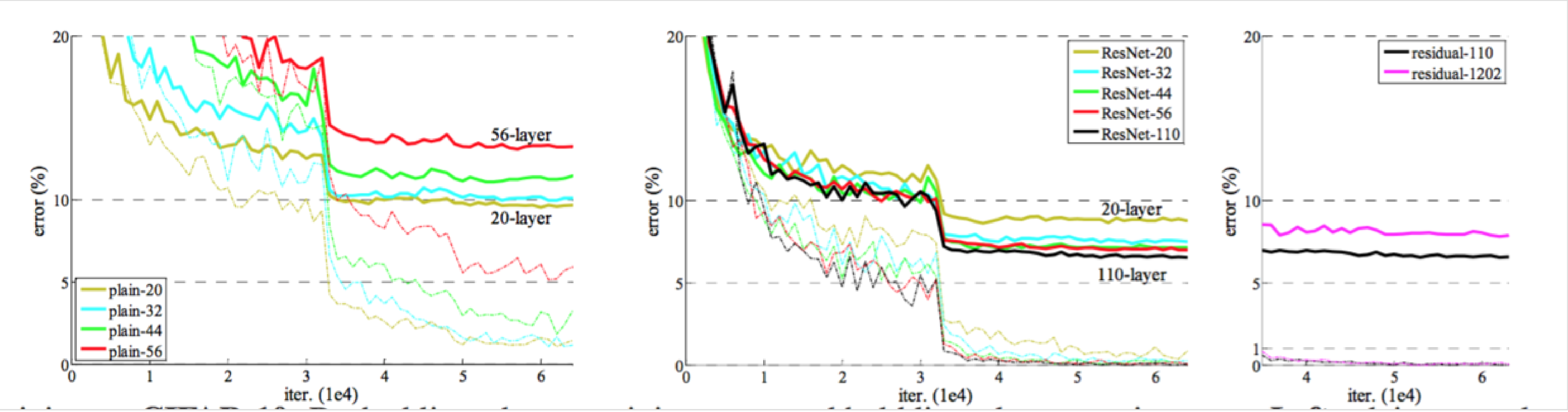


Figure 6. Training on CIFAR-10. Dashed lines denote training error, and bold lines denote testing error. Left: plain networks. The error of plain-110 is higher than 60% and not displayed. Middle: ResNets. Right: ResNets with 110 and 1202 layers.

我们比较了$n = {3, 5, 7, 9}$，得到了20层，32层，44层和56层的网络。图6（左）显示了简单网络的行为。深度简单网络经历了深度增加，随着深度增加表现出了更高的训练误差。这种现象类似于ImageNet中（图4，左）和MNIST中（请看[41]）的现象，表明这种优化困难是一个基本的问题。

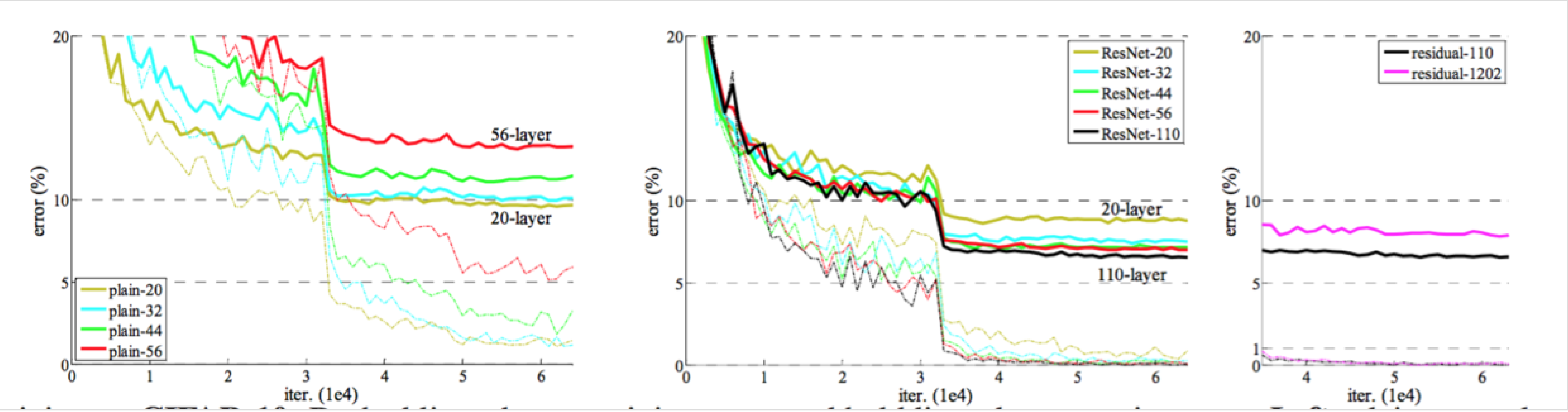
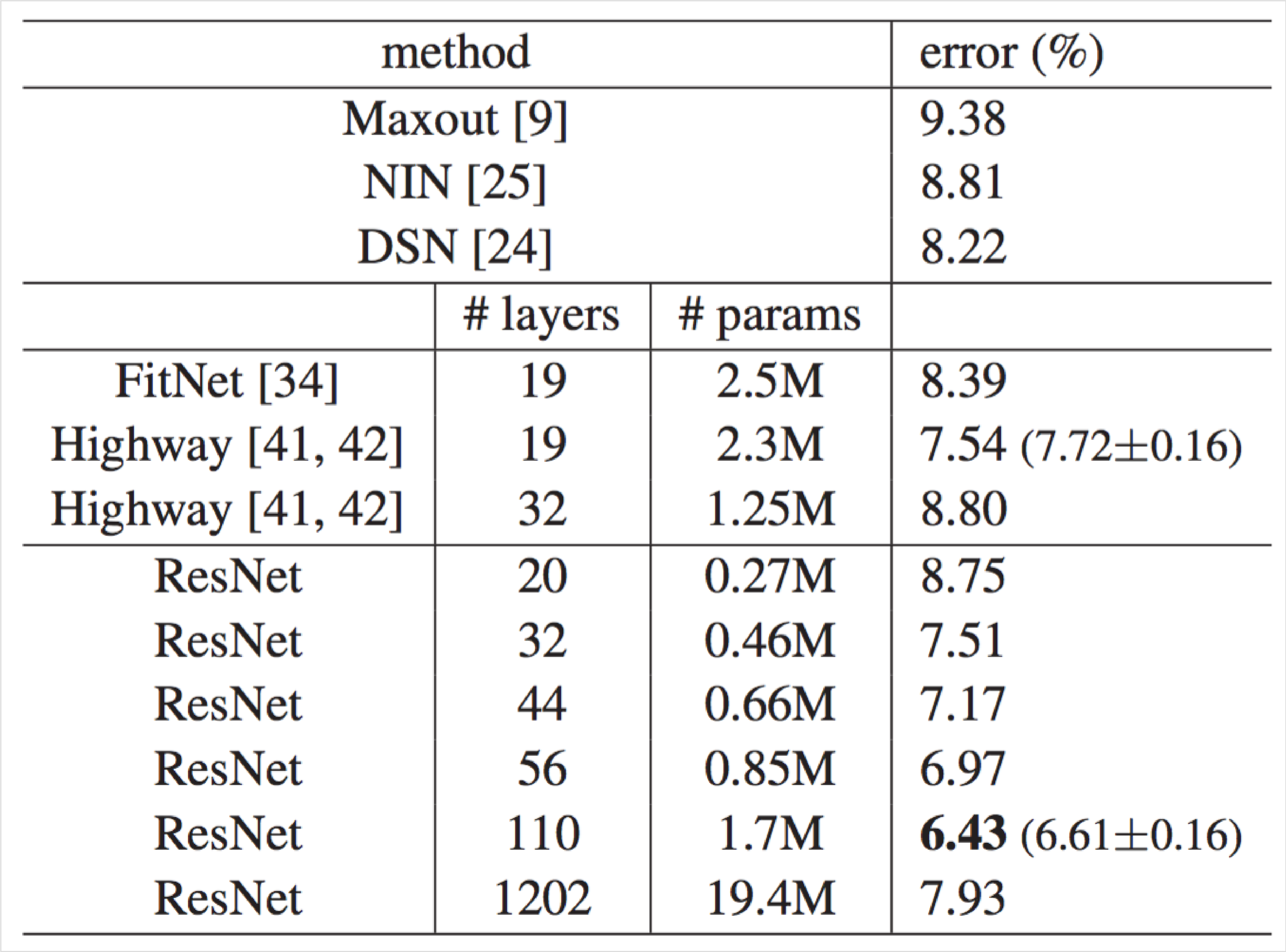


图6。在CIFAR-10上训练。虚线表示训练误差，粗线表示测试误差。左：简单网络。简单的110层网络错误率超过60%没有展示。中间：ResNet。右：110层ResNet和1202层ResNet。

Fig. 6 (middle) shows the behaviors of ResNets. Also similar to the ImageNet cases (Fig. 4, right), our ResNets manage to overcome the optimization difficulty and demonstrate accuracy gains when the depth increases.

图6（中）显示了ResNet的行为。与ImageNet的情况类似（图4，右），我们的ResNet设法克服优化困难并随着深度的增加展示了准确性收益。

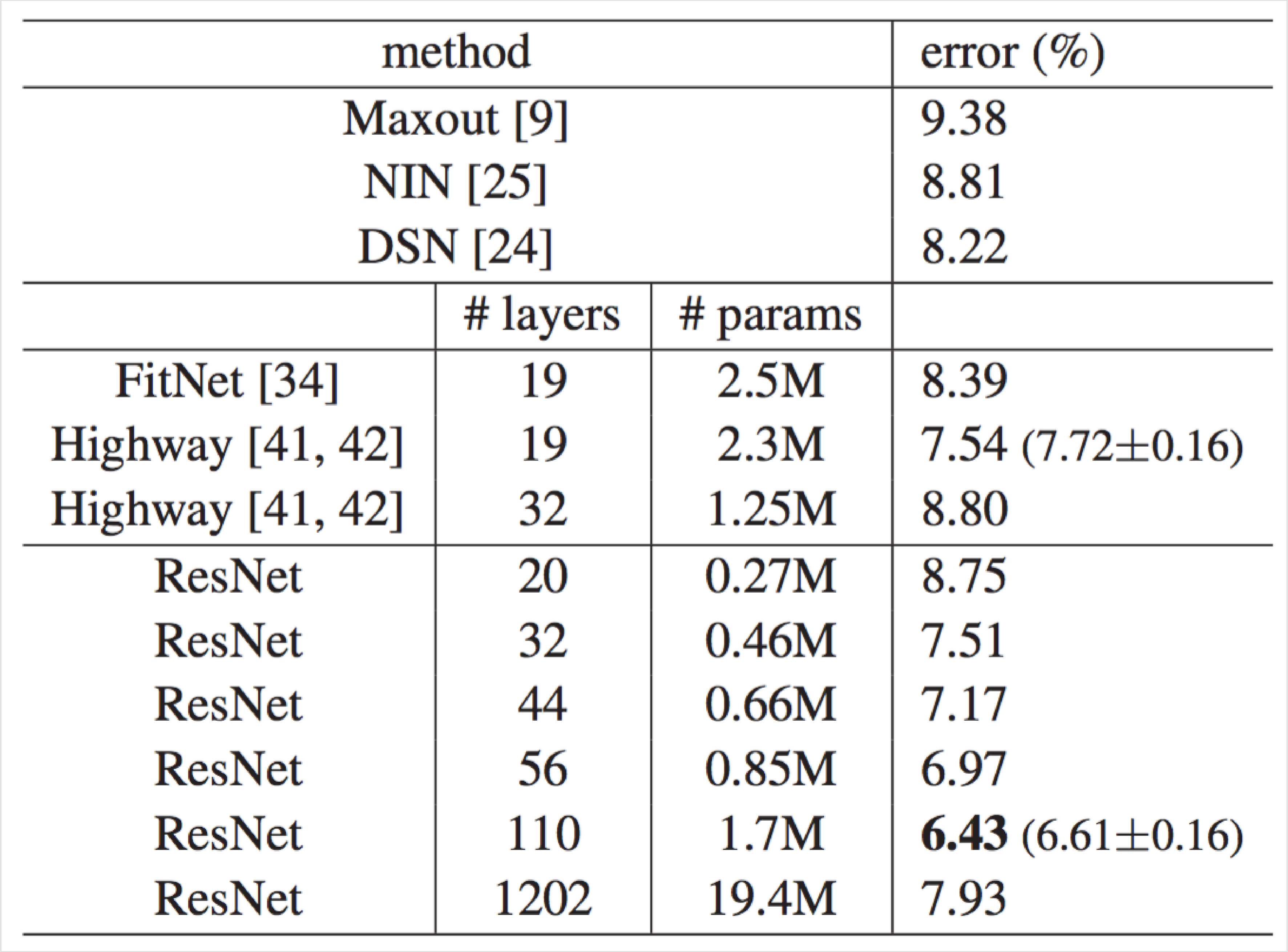
We further explore $n = 18$ that leads to a 110-layer ResNet. In this case, we find that the initial learning rate of 0.1 is slightly too large to start converging. So we use 0.01 to warm up the training until the training error is below 80% (about 400 iterations), and then go back to 0.1 and continue training. The rest of the learning schedule is as done previously. This 110-layer network converges well (Fig. 6, middle). It has fewer parameters than other deep and thin networks such as FitNet [34] and Highway [41] (Table 6), yet is among the state-of-the-art results (6.43%, Table 6).



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

Table 6. Classification error on the CIFAR-10 test set. All methods are with data augmentation. For ResNet-110, we run it 5 times and show “ best (mean±std) ” as in [42].

我们进一步探索了$n = 18$得到了110层的ResNet。在这种情况下，我们发现0.1的初始学习率对于收敛来说太大了。因此我们使用0.01的学习率开始训练，直到训练误差低于80%（大约400次迭代），然后学习率变回到0.1并继续训练。学习过程的剩余部分与前面做的一样。这个110层网络收敛的很好（图6，中）。它与其它的深且窄的网络例如FitNet[34]和Highway41相比有更少的参数，但结果仍在目前最好的结果之间（6.43%，表6）。



method	# layers	# params	error (%)
Maxout [9]			9.38
NIN [25]			8.81
DSN [24]			8.22
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

表6。在CIFAR-10测试集上的分类误差。所有的方法都使用了数据增强。对于ResNet-110，像论文[42]中那样，我们运行了5次并展示了“ 最好的(mean±std) ”。

Analysis of Layer Responses. Fig. 7 shows the standard deviations (std) of the layer responses. The responses are the outputs of each 3×3 layer, after BN and before other nonlinearity (ReLU/addition). For ResNets, this analysis reveals the response strength of the residual functions. Fig. 7 shows that ResNets have generally smaller responses than their plain counterparts. These results support our basic motivation (Sec.3.1) that the residual functions might be generally closer to zero than the non-residual functions. We also notice that the deeper ResNet has smaller magnitudes of responses, as evidenced by the comparisons among ResNet-20, 56, and 110 in Fig. 7. When there are more layers, an individual layer of ResNets tends to modify the signal less.

层响应分析。图7显示了层响应的标准偏差（std）。这些响应每个3×3层的输出，在BN之后和其他非线性（ReLU/加法）之前。对于ResNets，该分析揭示了残差函数的响应强度。图7显示ResNet的响应比其对应的简单网络的响应更小。这些结果支持了我们的基本动机（第3.1节），残差函数通常具有比非残差函数更接近零。我们还注意到，更深的ResNet具有较小的响应幅度，如图7中ResNet-20，56和110之间的比较所证明的。当层数更多时，单层ResNet趋向于更少地修改信号。

Exploring Over 1000 layers. We explore an aggressively deep model of over 1000 layers. We set $n = 200$ that leads to a 1202-layer network, which is trained as described above. Our method shows no optimization difficulty, and this 10^3-layer network is able to achieve training error <0.1% (Fig. 6, right). Its test error is still fairly good (7.93%, Table 6).

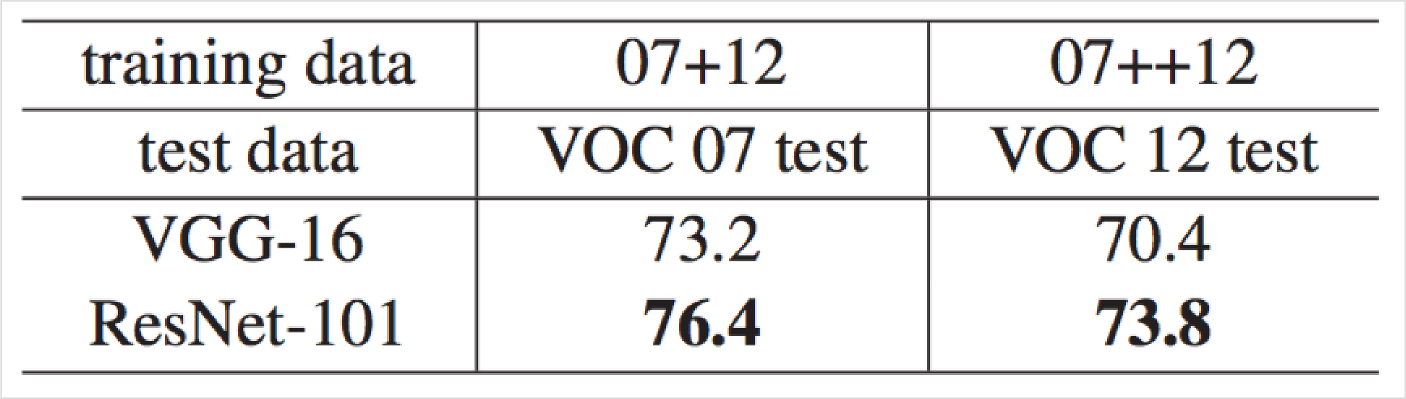
探索超过1000层。我们探索超过1000层的过深的模型。我们设置$n = 200$，得到了1202层的网络，其训练如上所述。我们的方法显示没有优化困难，这个10^3层网络能够实现训练误差<0.1%（图6，右图）。其测试误差仍然很好（7.93%，表6）。

But there are still open problems on such aggressively deep models. The testing result of this 1202-layer network is worse than that of our 110-layer network, although both have similar training error. We argue that this is because of overfitting. The 1202-layer network may be unnecessarily large (19.4M) for this small dataset. Strong regularization such as maxout [9] or dropout [13] is applied to obtain the best results ([9, 25, 24, 34]) on this dataset. In this paper, we use no maxout/dropout and just simply impose regularization via deep and thin architectures by design, without distracting from the focus on the difficulties of optimization. But combining with stronger regularization may improve results, which we will study in the future.

但是，这种极深的模型仍然存在着开放的问题。这个1202层网络的测试结果比我们的110层网络的测试结果更差，虽然两者都具有类似的训练误差。我们认为这是因为过拟合。对于这种小型数据集，1202层网络可能是不必要的大（19.4M）。在这个数据集应用强大的正则化，如maxout[9]或者dropout[13]来获得最佳结果（[9,25,24,34]）。在本文中，我们不使用maxout/dropout，只是简单地通过设计深且窄的架构简单地进行正则化，而不会分散集中在优化难点上的注意力。但结合更强的正规化可能会改善结果，我们将来会研究。

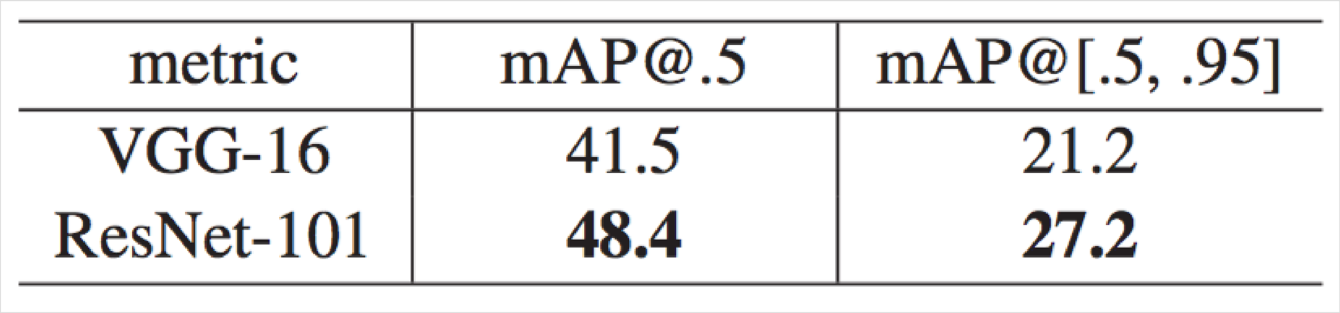
4.3. Object Detection on PASCAL and MS COCO

Our method has good generalization performance on other recognition tasks. Table 7 and 8 show the object detection baseline results on PASCAL VOC 2007 and 2012 [5] and COCO [26]. We adopt Faster R-CNN [32] as the detection method. Here we are interested in the improvements of replacing VGG-16 [40] with ResNet-101. The detection implementation (see appendix) of using both models is the same, so the gains can only be attributed to better networks. Most remarkably, on the challenging COCO dataset we obtain a 6.0% increase in COCO ’ s standard metric (mAP@[.5, .95]), which is a 28% relative improvement. This gain is solely due to the learned representations.



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 appendix for better results.

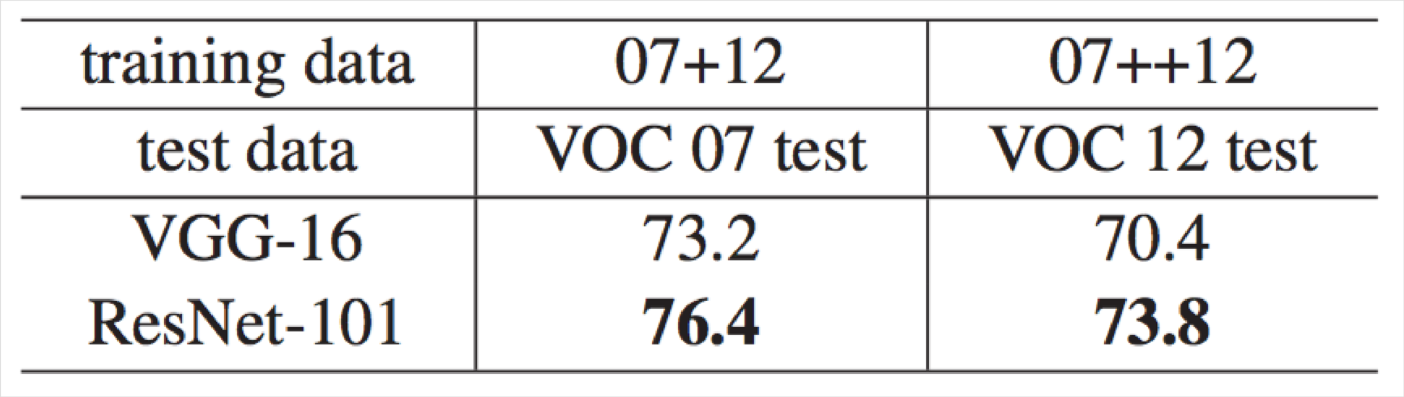


metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

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

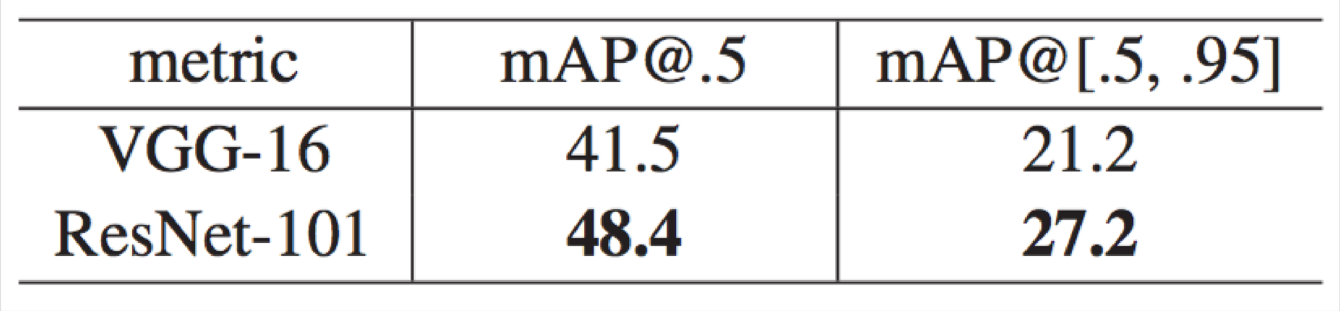
4.3. 在PASCAL和MS COCO上的目标检测

我们的方法对其他识别任务有很好的泛化性能。表7和表8显示了PASCAL VOC 2007和2012[5]以及COCO[26]的目标检测基准结果。我们采用更快的R-CNN[32]作为检测方法。在这里，我们感兴趣的是用ResNet-101替换VGG-16[40]。使用这两种模式的检测实现（见附录）是一样的，所以收益只能归因于更好的网络。最显著的是，在有挑战性的COCO数据集中，COCO的标准度量指标（mAP@[.5，.95]）增长了6.0%，相对改善了28%。这种收益完全是由于学习表示。



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**

表7。在PASCAL VOC 2007/2012测试集上使用基准Faster R-CNN的目标检测mAP(%)。更好的结果请看附录。



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

表8。在COCO验证集上使用基准Faster R-CNN的目标检测mAP(%)。更好的结果请看附录。

Based on deep residual nets, we won the 1st places in several tracks in ILSVRC & COCO 2015 competitions: ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. The details are in the appendix.

基于深度残差网络，我们在ILSVRC & COCO 2015竞赛的几个任务中获得了第一名，分别是：ImageNet检测，ImageNet定位，COCO检测，COCO分割。跟多细节请看附录。

References

[1] Y.Bengio,P.Simard,andP.Frasconi.Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.

[2] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.

[3] W. L. Briggs, S. F. McCormick, et al. A Multigrid Tutorial. Siam, 2000.

[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC, 2011.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. IJCV, pages 303–338, 2010.

[6] R. Girshick. Fast R-CNN. In ICCV, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. arXiv:1302.4389, 2013.

[10] K.Heand J.Sun. Convolutional neural networks at constrained time cost. In CVPR, 2015.

[11] K.He, X.Zhang, S.Ren, and J.Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015.

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, 2012.

[14] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Diploma thesis, TU Munich, 1991.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.

[17] H.Jegou, M.Douze, and C.Schmid. Product quantization for nearest neighbor search. TPAMI, 33, 2011.

[18] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. TPAMI, 2012.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093, 2014.

[20] A. Krizhevsky. Learning multiple layers of features from tiny images. Tech Report, 2009.

[21] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to hand-written zip code recognition. Neural computation, 1989.

[23] Y.LeCun,L.Bottou,G.B.Orr,and K.-R.Muller. Efficient back prop. In Neural Networks: Tricks of the Trade, pages 9–50. Springer, 1998.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. arXiv:1409.5185, 2014.

[25] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv:1312.4400,2013.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: Common objects in context. In ECCV. 2014.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[28] G. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. In NIPS, 2014.

[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.

[30] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In CVPR, 2007.

[31] T. Raiko, H. Valpola, and Y. LeCun. Deep learning made easier by linear transformations in perceptrons. In AISTATS, 2012.

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[33] B. D. Ripley. Pattern recognition and neural networks. Cambridge university press, 1996.

[34] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In ICLR, 2015.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. arXiv:1409.0575, 2014.

[36] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120, 2013.

[37] N.N.Schraudolph. Accelerated gradient descent by factor-centering decomposition. Technical report, 1998.

[38] N. N. Schraudolph. Centering neural network gradient factors. In Neural Networks: Tricks of the Trade, pages 207–226. Springer, 1998.

[39] P.Sermanet, D.Eigen, X.Zhang, M.Mathieu, R.Fergus, and Y.LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[41] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. arXiv:1505.00387, 2015.

[42] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. 1507.06228, 2015.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.

[44] R. Szeliski. Fast surface interpolation using hierarchical basis functions. TPAMI, 1990.

[45] R. Szeliski. Locally adapted hierarchical basis preconditioning. In SIGGRAPH, 2006.

[46] T. Vatanen, T. Raiko, H. Valpola, and Y. LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In Neural Information Processing, 2013.

[47] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[48] W. Venables and B. Ripley. Modern applied statistics with s-plus. 1999.

[49] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In ECCV, 2014.

# Deep Learning

ResNet论文翻译——中文对照

YOLO论文翻译——中文对照

- 文章目录
- 站点概览

Tyan
工作中的技术总结

612
日志
26

分类
42
标签