图像分类经典论文翻译汇总：[翻译汇总]

翻译 pdf 文件下载：[下载地址]

此版为纯中文版，中英文对照版请稳步：[ResNet 中英文对照版]

Deep Residual Learning for Image Recognition
深度残差学习的图像识别

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
何恺明 张翔宇 任少卿 孙剑
Microsoft Research (微软研究院)
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

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。

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. 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。

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等同于堆叠更多的层呢？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。

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。

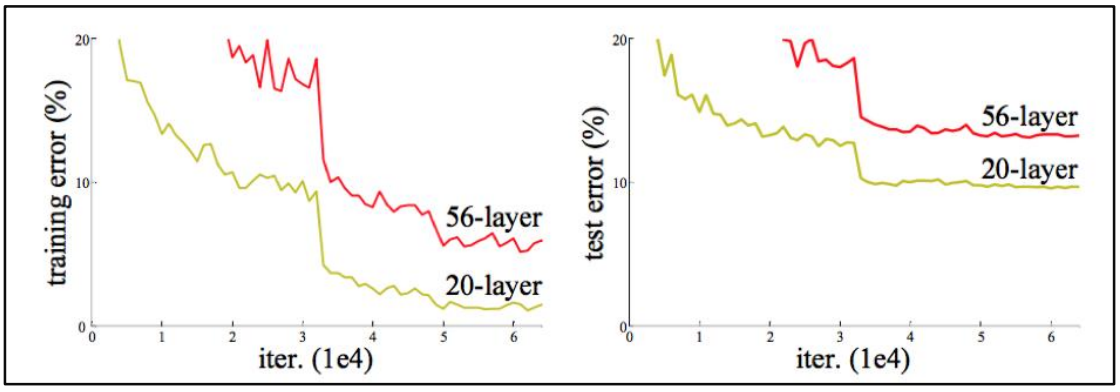

图 1. 20 层和 56 层的“简单”网络在 CIFAR-10 上的训练误差（左）和测试误差（右）。更深的网络有更高的训练误差和测试误差。如图 4 所示，ImageNet 上具有类似现象。

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）。

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 $H(\mathrm{x})$，我们将堆叠的非线性层拟合另一个映射 $F(\mathrm{x}) := H(\mathrm{x}) - \mathrm{x}$。原始的映射重写为 $F(\mathrm{x}) + \mathrm{x}$。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。

公式 $F(\mathrm{x}) + \mathrm{x}$ 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。

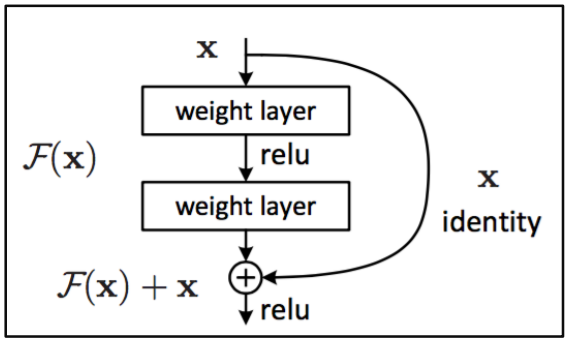

图 2. 残差学习：构建块

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。

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。

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. 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。

在低层次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。

快捷连接。实践和理论导致快捷连接[2,33,48]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。

和我们同时进行的工作，“highway networks” [41, 42]提出了门控功能[15]的快捷连接。这些门是数据依赖且有参数的，与我们不具有参数的恒等快捷连接不同。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. 深度残差学习

3.1. 残差学习

我们把 $H(x)$看作几个堆叠层（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。如果假设多个非线性层可以渐近地近似复杂函数，那么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的）。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目标函数（如假设），但学习的难易程度可能是不同的。

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。

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）得到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。

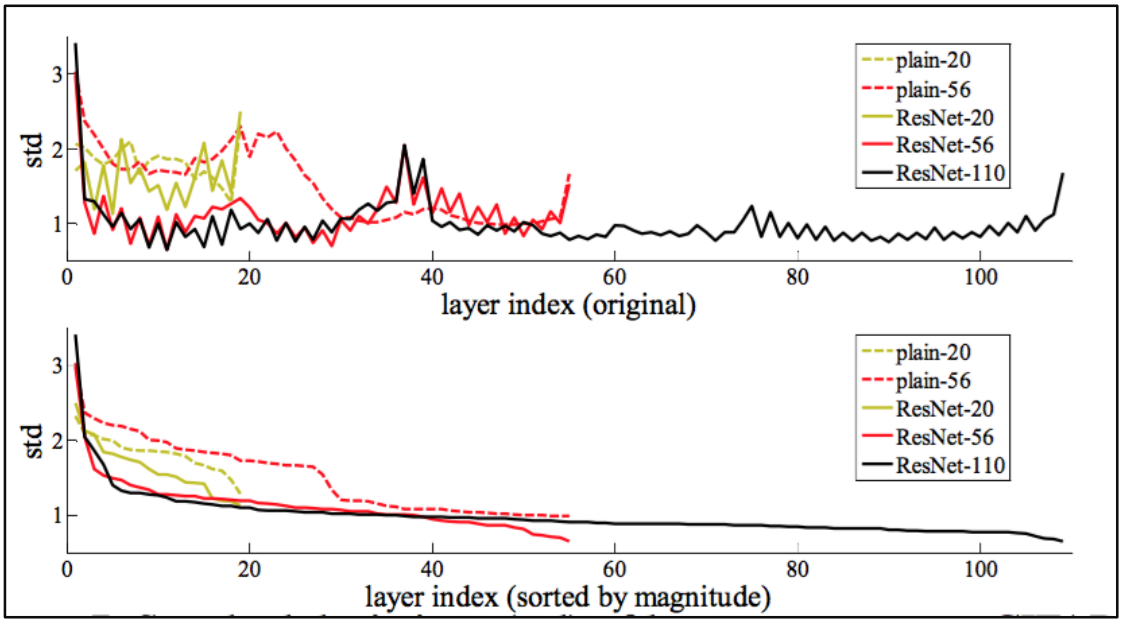

图 7. CIFAR-10 上层响应的标准差（std）。这些响应是每个 3×3 层的输出，在 BN 之后非线性之前。**上图**：以原始顺序显示层。**下图**：响应按降序排列。

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：

$$y = F(x, W_i) + x \qquad (1)$$

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 函数[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 σ(y)，看图 2）。

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（除了可忽略不计的元素加法之外）。

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：

$$y = F(x, \{W_i\}) + W_s x. \quad (2)$$

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。

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，当然更多层也是可以的。但如果 F 只有一层，方程（1）类似于线性层：$y = W_1 x + x$，我们没有看到优势。

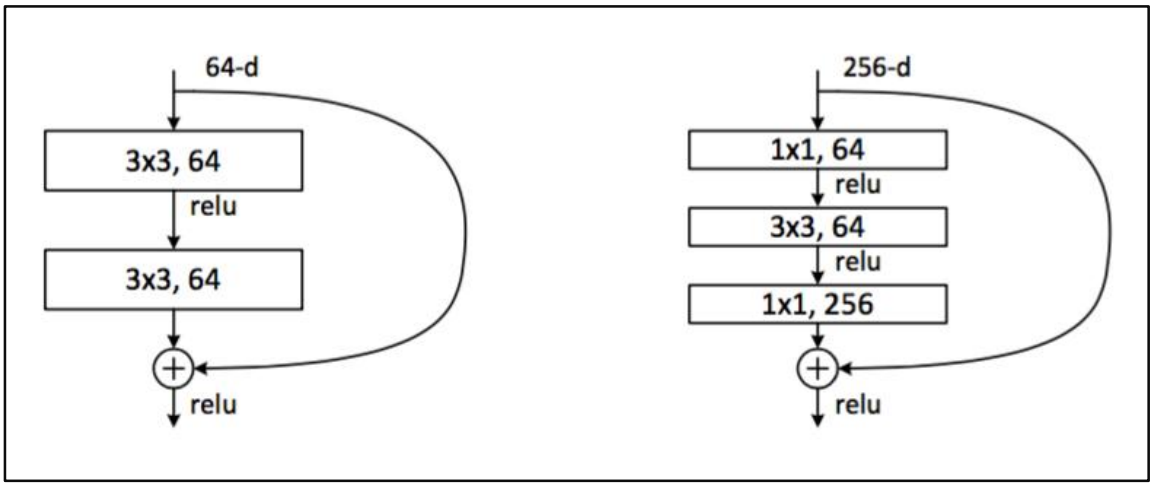

图 5. ImageNet 的深度残差函数 F。**左**：图 3 中 ResNet-34 的构建块（在 56×56 的特征图上）。**右**：ResNet-50/101/152 的“瓶颈”构建块。

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. 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。

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）为了得到相同的输出特征图尺寸，所有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。

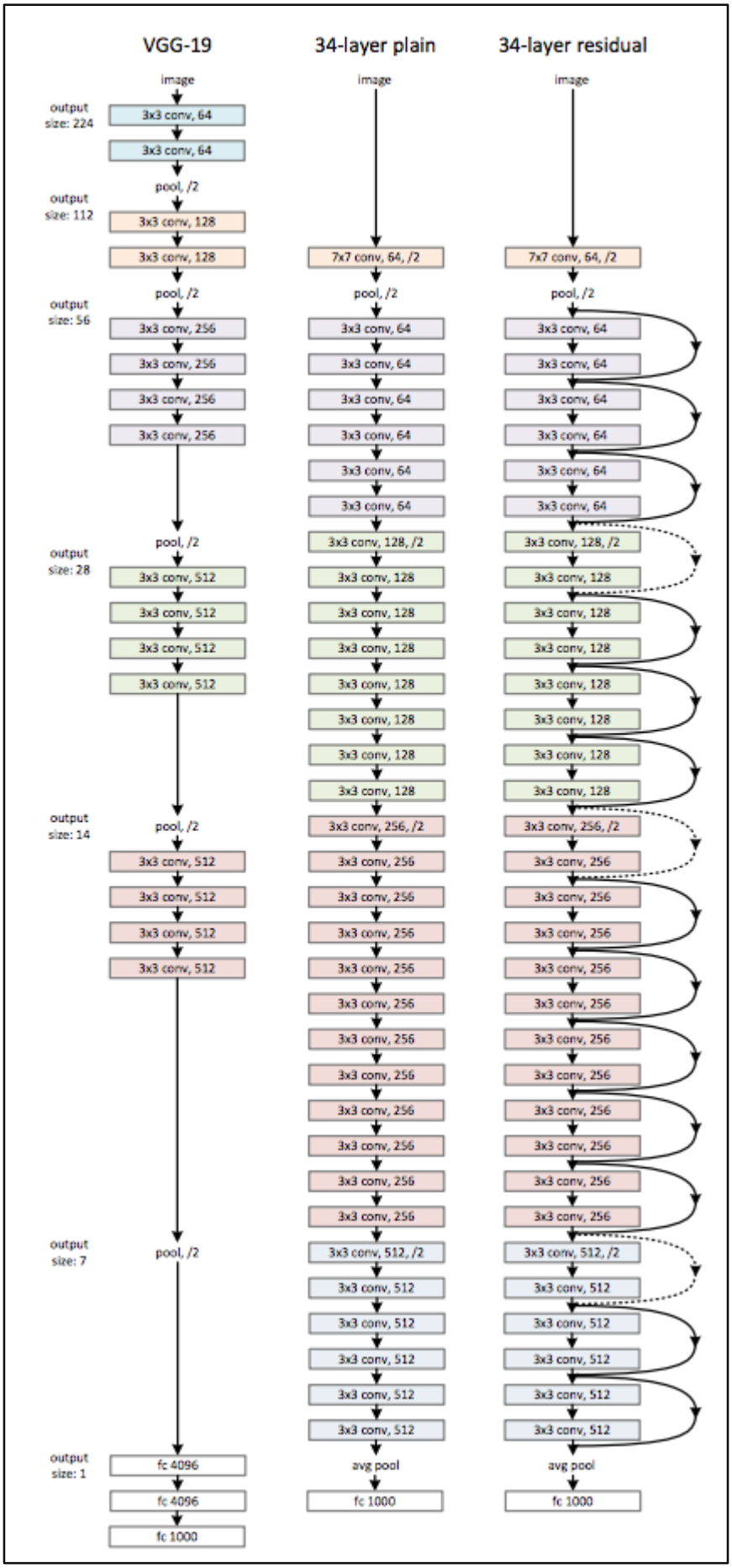
VGG-19
34-layer plain
34-layer residual
image
output size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 14
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 7
fc 4096
fc 4096
fc 1000
output size: 1
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[41]（196 亿 FLOPs）。**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。构建块显示在方括号中（同图 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。

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%。

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. 实现

ImageNet 中我们的实现遵循[21，41]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[41]。从图像或其水平翻转中随机采样 224×224 裁剪大小，并逐像素减去均值[21]。使用了[21]中的标准颜色增强。参照[16]，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]。

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, 13]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. 实验

4.1. ImageNet 分类

我们在 ImageNet 2012 分类数据集[36]上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。

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。

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 个裁剪图像进行测试)。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。

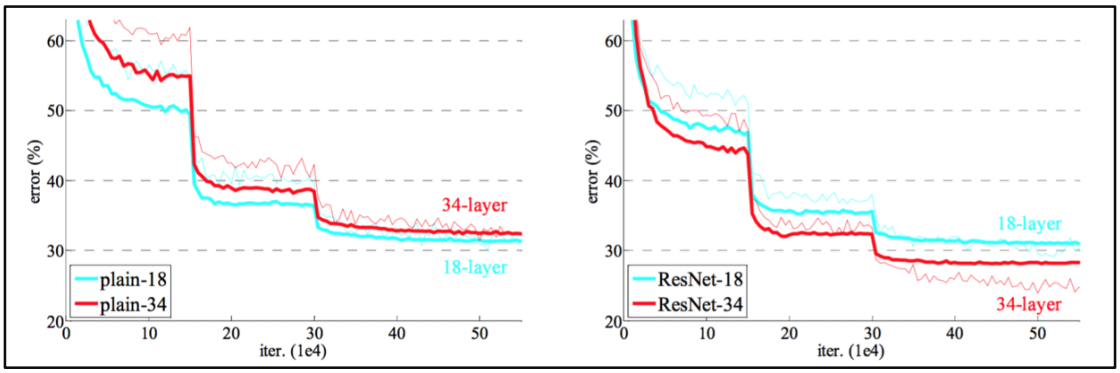

图 4. 在 ImageNet 上进行训练。细曲线表示训练误差，粗曲线表示中心裁剪图像的验证误差。**左**：18 层和 34 层的简单网络。**右**：18 层和 34 层的 ResNet。在本图中，残差网络与对应的简单网络相比没有额外的参数。

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有

效。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 个裁剪图像进行测试）。VGG16 是基于我们的测试结果的。ResNet-50/101/152 都是仅使用投影增加维度的 B 配置。

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。

我们从表 2 和图 4 中可以看到三个主要的观察结果。首先，通过残差学习这种情况发生的逆转——34 层 ResNet 比 18 层 ResNet 更好（2.8%）。更重要的是，34 层 ResNet 显示出相当低的训练误差，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。

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。

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。

恒等与投影快捷连接。我们已经表明没有参数、恒等快捷连接有助于训练。接下来我们探讨投影快捷连接（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。

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。

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。由于考虑到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）。这三层是 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。

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 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 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。

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）。

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 单模型具有 4.49％的 top-5 错误率。这个单一模型的结果胜过以前的所有集成结果（表 5）。我们结合了六种不同深度的模型，形成一个集成模型（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. 集成模型的错误率(%)。top-5 错误率是 ImageNet 测试集上由测试服务器报告的。

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，而不是产生最先进的结果，所以我们有意使用如下的简单架构。

简单/残差架构如图 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 6*n* 个堆叠层，每个特征图大小使用 2*n* 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 6*n*+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+2*n*	2*n*	2*n*
# 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 3*n* 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。

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 dropout。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。

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[42]）的现象，表明这种优化困难是一个基本的问题。

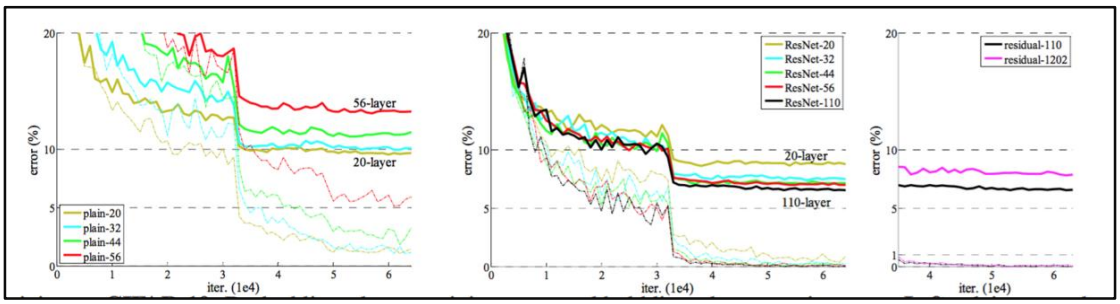

图 6. 在 CIFAR-10 上训练。虚线表示训练误差，粗线表示测试误差。**左**：简单网络。简单的 110 层网络错误率超过 60%，没有展示。**中**：ResNet。**右**：110 层和 1202 层的 ResNet。

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

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[35]和 Highway[42]相比有更少的参数，然而结果仍在目前最好的结果中（6.43%，表 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

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[43]中那样，我们运行了 5 次并展示了“best (mean±std)”。

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），即残差函数通常比非残差函数更接近零。通过比较图 7 中 ResNet-20，ResNet-56 和 ResNet-110，我们还注意到，更深的 ResNet 具有较小的响应幅度。当层数更多时，单层 ResNet 趋向于更少地修改信号。

探索超过 1000 层。我们探索了一个超过 1000 层非常深的模型。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 层的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）。

但是，这种极深的模型仍然存在着一些未解决的问题。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。我们认为这是过拟合造成的。对于这种小型数据集，1202 层网络可能过大（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，我们将来会做研究。

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 Faster 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]。这两种模型的检测方法（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到的表示。

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。

基于深度残差网络，我们在 ILSVRC 和 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。详见附录。

参考文献

[1] Y. Bengio, P. Simard, and P. 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] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware cnn model. In ICCV, 2015.
[7] R. Girshick. Fast R-CNN. In ICCV, 2015.
[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.
[10] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. arXiv:1302.4389, 2013.
[11] K. He and J. Sun. Convolutional neural networks at constrained time cost. In CVPR, 2015.
[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.
[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015.
[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing coadaptation of feature detectors. arXiv:1207.0580, 2012.
[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 handwritten zip code recognition. Neural computation, 1989.

[23] Y. LeCun, L. Bottou, G. B. Orr, and K.-R.M ¨uller. Efficient backprop. 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. Deeplysupervised 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. Doll ´ar, 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. Mont ´ufar, 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] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. arXiv:1504.06066, 2015.

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

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

[36] 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.

[37] 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.

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

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

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

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

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

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

[44] 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.

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

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

[47] 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.

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

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

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