深入研究整流器：
在 ImageNet 分类方面超越人类水平的性能

何开明 张翔宇 任少青 孙健

Microsoft Research

{Kahe，V-Siangz，V-shren，Jiansun}@微软研究院.com

抽象

整流激活单元（整流器）对于以下设备至关重要

最先进的神经网络。在这项工作中，我们从两个方面研究了用于图像分类的整流神经网络。首先，我们提出了一个参数化整流线性单元（PReLU），它推广了传统的整流单元。PReLU 改进了模型拟合，额外的计算成本几乎为零，过拟合风险很小。其次，我们开发了一种稳健的初始化方法，该方法特别考虑了整流器非线性。这种方法使我们能够直接从从头开始训练极深的整流模型，并研究更深或更广泛的网络架构。基于我们的 PReLU 网络（PReLU-nets），我们在 ImageNet 2012 分类数据集上实现了 4.94% 的前 5 名测试误差。这比 ILSVRC 2014 获胜者（GoogLeNet，6.66% [29]）相对提高了 26%。据我们所知，在这项视觉识别挑战中，我们的结果首次超过了人类水平的性能（5.1%，[22]）。

1. 引言

卷积神经网络（CNN）[17，16] 在多项视觉识别任务中表现出优于人类或与人类相当的识别准确性，包括识别交通标志 [3]、人脸 [30，28] 和手写数字 [3，31]。在这项工作中，我们提出了一个结果，该结果在更通用和更具挑战性的识别任务上超越了人类水平的表现——1000 类 ImageNet 数据集中的分类任务 [22]。

在过去几年中，我们目睹了识别性能的巨大进步，这主要是由于两个技术方向的进步。构建更强大的模型和设计防止过拟合的有效策略。一方面，由于复杂性的增加（例如，深度增加 [25，29]、宽度扩大 [33，24]、以及使用更小的步幅 [33，24，2，25]）、新的非线性激活 [21，20，34，19，27，9] 和追辩层设计 [29，11]。另一方面，更好的泛化是通过有效的正则化技术 [12，26，9，31]、积极的数据增强 [16，13，25，29] 和大规模数据 [4，22] 实现的。

在这些进展中，整流神经元 [21，8，20，34]，例如整流线性单元（ReLU），是最近深度网络成功的几个关键之一 [16]。它促进了训练过程的收敛 [16]，并导致比传统的 S 状单元更好的解决方案 [21，8，20，34]。尽管整流器网络很普遍，但最近对模型 [33，24，11，25，29] 和训练它们的理论指南 [7，23] 的改进很少关注整流器的特性。

在本文中，我们从两个方面研究神经网络，特别是由整流器驱动的。首先，我们提出了 ReLU 的新泛化，我们称之为参数校正线性单元（PReLU）。此激活函数自适应地学习整流器的参数，并以可忽略不计的额外计算成本提高精度。其次，我们研究了训练非常深入的修正模型的难度。通过显式模拟整流器的非线性（ReLU/PReLU），我们推导出了一种理论上合理的初始化方法，这有助于收敛直接从从头开始训练的非常深的模型（例如，具有 30 个权重层）。这为我们提供了更大的灵活性来探索更强大的网络架构。

在 1000 类 ImageNet 2012 数据集上，我们的 PReLU 网络（PReLU-net）导致 5.71% 前 5 个误差的单模型结果，这超过了所有现有的多模型结果。此外，我们的多模型结果在测试集上实现了 4.94% 的前 5 名误差，这比 ILSVRC 2014 年获胜者（GoogLeNet，6.66% [29]）相对提高了 26%。据我们所知，在这次视觉识别挑战中，我们的结果首次超过了报告的人类水平表现（[22] 中的 5.1%）。

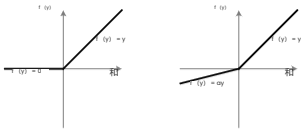

图 1.ReLU 与 PReLU。对于 PReLU，负部分的系数不是恒定的，而是自适应学习的。

2. 方法

在本节中，我们首先介绍 PReLU 激活函数（第 2.1 节）。然后，我们推导出深度整流器网络的初始化方法（第 2.2 节）。最后，我们讨论我们的架构设计（第 2.3 节）。

2.1. 参数整流器

我们表明，用学习的参数激活单元替换无参数的 ReLU 激活可以提高分类精度1。

定义

正式地，我们认为激活函数定义为：

$$f(y_i) = \begin{cases} y_i, & \text{如果 } y_i > 0 \\ a_i y_i, & \text{如果 } y_i \le 0 \end{cases} \quad (1)$$

这里 yi 是第 i 个通道上非线性激活 f 的输入，而 ai 是控制负部分斜率的系数。ai 中的下标 i 表示我们允许非线性激活在不同通道上变化。当 ai = 0 时，它变为 ReLU;当 ai 是一个可学习的参数时，我们将方程 （1） 称为参数 ReLU （PReLU）。图 1 显示了 ReLU 和 PReLU 的形状。方程 （1） 等价于 f （yi） = max （0， yi） + ai min （0， yi）。

如果 ai 是一个较小的固定值，则 PReLU 将成为 [20] 中的泄漏 ReLU （LReLU）（ai = 0.01）。LReLU 的动机是避免零梯度。[20] 中的实验表明，与 ReLU 相比，LReLU 对准确性的影响可以忽略不计。相反，我们的方法与整个模型一起自适应地学习 PReLU 参数。我们希望端到端的培训能够带来更专业的激活。

PReLU 引入了极少数额外的参数。额外参数的数量等于通道的总数，在考虑权重总数时可以忽略不计。因此，我们预计不会有额外的过度拟合风险。我们还考虑一个频道共享的变体：

在我们的工作期间时，Agostinelli 等人 [1] 还研究了学习激活功能，并显示出其他任务的改进。

f （yi） = max （0， yi） + a min （0， yi），其中系数由一层的所有通道共享。此变体仅在每个层中引入一个额外的参数。

优化

PReLU 可以使用反向传播 [17] 进行训练，并与其他层同时优化。{ai} 的更新形式只是从链式规则推导出来的。一层 ai 的梯度为：

$$\frac{\partial E}{\partial a_i} = \sum_{\text{集}} \frac{\partial E}{\partial f(y_i)} \frac{\partial f(y_i)}{\partial a_i}, \quad (2)$$

其中 E 表示目标函数。术语 $\frac{\partial E}{\partial f(y_i)}$ 是从较深层传播的梯度。激活的信号由下式给出：

$$\frac{\partial f(y_i)}{\partial a_i} = \begin{cases} 0, & \text{如果 } y_i > 0 \\ y_i, & \text{如果 } y_i \le 0 \end{cases} \quad (3)$$

求和 ∑yi 将覆盖特征的所有位置

地图。对于通道共享变体，a 的梯度为 ∂E ∂a = ∑i∑yi ∂E ∂f （yi） ∂f （yi） ∂a，其中 ∑i 对所有 chan- 求和

层的 nels 的 Nels 的 Nels 的 NicolsPReLU 导致的时间复杂度对于前向和后向传播都可以忽略不计。

我们在更新 ai 时采用 momentum 方法：

$$\Delta a_i := \mu \Delta a_i + \frac{\partial E}{\partial a_i}. \quad (4)$$

这里 μ 是动量，是学习率。值得注意的是，我们在更新 ai 时不使用权重衰减（l2 正则化）。权重衰减往往会将 ai 推到零，从而使 PReLU 偏向于 ReLU。即使没有正则化，在我们的实验中，学习到的系数也很少具有大于 1 的倍数。此外，我们不限制 ai 的范围，因此激活函数可以是非单调的。在本文中，我们使用 ai = 0.25 作为初始化。

比较实验

我们对一个深入但高效的模型进行了比较

有 14 个权重层。该模型在 [10] 中进行了研究（[10] 的模型 E）。其结构在 Ta- ble 1 中进行了描述。我们选择这个模型是因为它足以代表一类非常深的模型，以及使实验可行。

作为基线，我们在卷积 （conv） 层和前两个全连接 （fc） 层中使用 ReLU 来训练这个模型。训练实施遵循 [10]。使用 10 视图测试，ImageNet 2012 的前 1 和前 5 错误分别为 33.82% 和 13.34%（表 2）。

		学习系数	
层		频道共享	通道
卷积1	7×7, 64, /2	0.681	0.596
矿池 1	3×3, /3		
卷积21	2×2, 128	0.103	0.321
卷积22	2×2, 128	0.099	0.204
conv23	2×2, 128	0.228	0.294
conv24	2×2, 128	0.561	0.464
矿池 2	2×2, /2		
conv31	2×2, 256	0.126	0.196
conv32	2×2, 256	0.089	0.152
卷积33	2×2, 256	0.124	0.145
卷积34	2×2, 256	0.062	0.124
卷积35	2×2, 256	0.008	0.134
卷积36	2×2, 256	0.210	0.198
spp	{6, 3, 2, 1}		
FC4096		0.063	0.074
FC4096		0.031	0.075
FC1000			

表 1.一个小而深的 14 层模型 [10]。将列出每个层的过滤器大小和过滤器编号，数字 /s 表示使用的步幅 s。还显示了 PReLU 的学习系数。对于通道情况，显示了每一层的通道上 {ai} 的平均值。

	第一名	前 5 名
ReLU 系列	33.82	13.34
PReLU，通道共享	32.71	12.87
PReLU，通道	32.64	12.75

表 2.ReLU 和 PReLU 在小模型上的比较。错误率适用于使用 10 视图测试的 ImageNet 2012。在训练和测试期间，调整图像的大小，使较短的边为 256。每个视图为 224×224。所有模型都使用 75 个 epoch 进行训练。

然后，我们从头开始训练相同的架构，所有 ReLU 都替换为 PReLU（表 2）。前 1 个错误降低到 32.64%。这比 ReLU 基线高出 1.2%。表 2 还显示，通道/通道共享 PReLU 的性能相当。对于频道共享版本，与 ReLU 版本相比，PReLU 仅引入了 13 个额外的免费参数。但是，这少量的自由参数起着关键作用，这比基线高出 1.1% 就证明了这一点。这意味着适应性学习激活功能形状的重要性。

表 1 还显示了每层的 PReLU 学习系数。在 Ta- ble 1 中有两个有趣的现象。首先，第一个 conv 层 （conv1） 的系数 （0.681 和 0.596） 明显大于 0。由于 conv1 的过滤器大多是类似 Gabor 的滤波器，例如边缘检测器或纹理检测器，因此学习结果表明，滤波器的正响应和负响应都得到了尊重。我们是-

相信这是利用低级信息的一种更经济的方式，因为过滤器的数量有限（例如，64 个）。其次，对于通道版本，较深的 conv 层通常具有较小的系数。这意味着在增加的深度上，激活逐渐变得“更加非线性”。换句话说，学习的模型倾向于在早期阶段保留更多信息，并在更深的阶段变得更具区分性。

2.2. 整流器滤波器权重的初始化

与传统的 S 形激活网络相比，整流器网络更容易训练 [8， 16， 34]。但是错误的初始化仍然会阻碍高度非线性系统的学习。在本小节中，我们提出了一种健壮的初始化方法，它消除了训练极深整流器网络的障碍。

最近的深度 CNN 主要由从高斯分布中提取的随机权重初始化 [16]。在固定标准差（例如，[16]中的0.01）下，非常深的模型（例如，>8 个卷积层）很难收敛，正如VGG团队[25]所反映的那样，在我们的实验中也观察到了这一点。为了解决这个问题，在 [25] 中，他们预先训练了一个具有 8 个 conv 层的模型来初始化更深的模型。但这种策略需要更多的训练时间，并且还可能导致较差的局部最优值。在 [29， 18] 中，辅助分类器被添加到中间层以帮助收敛。

Glorot 和 Bengio [7] 提议采用适当缩放的均匀分布进行初始化。这在 [14] 中称为“Xavier”初始化。它的推导是基于激活是线性的假设。此假设对 ReLU 和 PReLU 无效。

在下文中，我们通过考虑 ReLU/PReLU 来推导出理论上更合理的初始化。在我们的实验中，我们的初始化方法允许极深的模型（例如，30 个 conv/fc 层）收敛，而“Xavier”方法 [7] 则不能。

前向传播情况

我们的推导主要遵循 [7]。中心思想是调查每层中响应的方差。

对于 conv 层，响应为：

$$y_l = W_l x_l + b_l. \quad (5)$$

其中，x 是一个 k2c×1 向量，表示 c 输入通道中共址的 k×k 像素。k 是图层的空间过滤器大小。其中 n = k2c 表示响应的连接数，W 是 d×n 矩阵，其中 d 是滤波器的数量，W 的每一行表示滤波器的权重。b 是偏差向量，y 是输出映射像素处的响应。我们使用 l 来索引一个层。我们有 xl = f （yl-1），其中 f 是活化。我们也有 cl = dl-1。

我们让 Wl 中初始化的元素相互独立并共享相同的分布。与 [7] 中所示，我们假设 xl 中的元素也是相互独立的，并且具有相同的分布，并且 xl 和 Wl 彼此独立。然后我们有：

$$\mathrm{Var}[y_l] = n_l \mathrm{Var}[w_l x_l], \tag{6}$$

其中现在 yl、xl 和 wl 分别表示 yl、Wl 和 xl 中每个元素的随机变量。我们让 wl 的均值为零。然后，自变量乘积的方差得到：

$$\mathrm{Var}[y_l] = n_l \mathrm{Var}[w_l] E[x_l^2]. \tag{7}$$

这里 E[x2l] 是 xl 平方的期望值。值得注意的是，=f Var[xl] 除非 xl 的均值为零。对于 ReLU 激活，xl = max（0，yl-1），因此它的均值不为零。这将导致与 [7] 不同的结论。

如果我们让 wl-1 在 0 附近具有对称分布且 bl-1 = 0，那么 yl-1 的均值为零，在 0 附近具有对称分布。这导致当 f 为 ReLU 时，E[x2l] = 12 Var[yl-1]。将其放入 Eqn.（7）中，我们得到：

$$\mathrm{Var}[y_l] = \tfrac{1}{2} n_l \mathrm{Var}[w_l]\mathrm{Var}[y_{l-1}]。 \tag{8}$$

将 L 层放在一起，我们得到了：

$$\mathrm{Var}[y_L] = \mathrm{Var}[y_1] \left(\prod_{l=2}^{L} \tfrac{1}{2} n_l \mathrm{Var}[w_l] \right). \tag{9}$$

这个产品是初始化设计的关键。正确的初始化方法应避免以指数方式减小或放大 input 信号的幅度。因此，我们将上述乘积计算为适当的标量（例如，1）。充分条件是：

$$\tfrac{1}{2} n_l \mathrm{Var}[w_l] = 1, \quad \forall l. \tag{10}$$

这导致零均值高斯分布，其标准偏差（std）✓ 2/nl。这是我们的初始化方式——

tion.我们还初始化 b = 0。

对于第一层（l = 1），我们应该有 nlVar[wl] = 1，因为输入信号上没有应用 ReLU。但是，如果因子 1/2 只存在于一层上，则无关紧要。因此，为了简单起见，我们还在第一层采用了 Eqn.（10）。

反向传播情况

对于反向传播，卷积层的梯度由下式计算：

$$\Delta x_l = \hat{W}_l \Delta y_l. \tag{11}$$

这里我们使用 Δx 和 Δy 来表示渐变（ ∂E∂x 和 ∂E∂y ）为了简单。Δy 表示 d 通道中的 k×k 像素，

并重塑为 k2d×1 向量。我们表示 ˆn = k2d。请注意，ˆn 6 = n = k2c。ˆW 是一个 c×ˆn 矩阵，其中滤波器以反向传播的方式重新排列。请注意，W 和 ˆW 可以彼此重塑。Δx 是一个 c×1 向量，表示该层像素处的渐变。如上所述，我们假设 wl 和 Δyl 彼此独立，那么当 wl 由围绕零的对称分布初始化时，Δxl 对所有 l 的均值为零。

在反向传播中，我们也有 Δyl = f ′（yl）Δxl+1，其中 f ′是 f 的导数。对于 ReLU 情况，f ′（yl）为 0 或 1，并且它们的概率相等。我们假设 f ′（yl）和 Δxl+1 彼此独立。因此，我们有 E[Δyl] = E[Δxl+1]/2 = 0，还有 E[（Δyl）2] = Var[Δyl] = 12 Var[Δxl+1]。然后我们计算 Eqn.（11）中梯度的方差：

$$\begin{aligned} \text{其中}[\Delta x_l] &= \hat{n}_l \mathrm{Var}[w_l]\mathrm{Var}[\Delta y_l] \\ &= \tfrac{1}{2} \hat{n}_l \mathrm{Var}[w_l]\mathrm{Var}[\Delta x_{l+1}]. \end{aligned} \tag{12}$$

方程（12）和方程（8）中的标量 1/2 是 ReLU 的结果，尽管推导不同。将 L 层放在一起，我们得到了：

$$\mathrm{Var}[\Delta x_2] = \mathrm{Var}[\Delta x_{L+1}] \left(\prod_{l=2}^{L} \tfrac{1}{2} \hat{n}_l \mathrm{Var}[w_l] \right). \tag{13}$$

我们认为梯度不是指数大/小的充分条件：

$$\tfrac{1}{2} \hat{n}_l \mathrm{Var}[w_l] = 1, \quad \forall l. \tag{14}$$

这个方程与方程（10）之间的唯一区别是 ˆnl = k2l dl 而 nl = k2l cl = k2l dl-1。方程（14）得到一个零均值高斯分布，其 std 为 ✓ 2/ˆnl。

对于第一层（l = 1），我们不需要计算 Δxl

，因为它代表图像域。但是我们仍然可以在第一层采用 Eqn.（14），原因与前向传播情况相同 - 单层的因子不会使整个乘积呈指数级大/小。

我们注意到，单独使用 Eqn.（14）或 Eqn.（10）就足够了。例如，如果我们使用方程（14），那么在方程（13）中，乘积 ΠLl=2 12 ˆnlVar[wl] = 1，在方程（9）中，乘积 ΠLl=2 12 nlVar[wl] = ΠLl=2 nl/ˆnl = c2/dL，这在常见的网络符号中不是一个递减数。这意味着，如果初始化正确地缩放了 backward 信号，那么 for- ward 信号也是如此;反之亦然。对于本文中的所有模型，两种形式都可以使它们收敛。

讨论

如果前向/后向信号在每层中按因子 β 不适当地缩放，则最终传播的信号

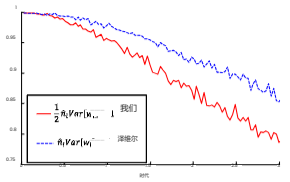

图 2.一个 22 层大型模型的收敛（Ta- ble 3 中的 B）。x 轴是训练 epoch 的数量，y 轴是 3,000 个随机 val 样本的前 1 个误差，在中心裁剪上评估。我们使用 ReLU 作为这两种情况的激活。我们的初始化（红色）和 “Xavier”（蓝色）[7] 都会导致收敛，但我们的初始化更早开始减少误差。

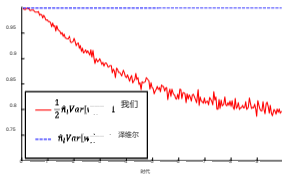

图 3.一个 30 层小模型的收敛（见正文）。我们使用 ReLU 作为这两种情况的激活。我们的初始化（红色）能够使其收敛。但是 “Xavier”（蓝色）[7] 完全停滞了 - 我们还验证了它的梯度都在变小。即使给定更多的纪元，它也不会收敛。

将在 L 层之后按 βL 因子重新缩放，其中 L 可以表示部分或全部层。当 L 很大时，如果β > 1，则会导致信号被极度放大，算法输出为无穷大;如果β < 1，则会导致 sig- nals 递减2。在任何一种情况下，算法都不会收敛 - 它在前一种情况下发散，在后一种情况下停滞。

我们的推导还解释了为什么 0.01 的常数标准差会使一些更深的网络停滞 [25]。我们从 VGG 团队论文 [25] 中的 “模型 B” 为例。此模型有 10 个 conv 层，所有层都有 3×3 个过滤器。第 1 层和第 2 层的滤波器编号（dl）为 64，第 3 层和第 4 层为 128，第 5 层和第 6 层为 256，其余层为 512。由 Eqn.（14）（√2/ nl）计算的 std 为

当过滤器编号分别为 64、128、256 和 512 时，为 0.059、0.042、0.029 和 0.021。如果 std 已初始化

2在存在权重衰减（权重的 l2 正则化）的情况下，当 Logistic 损失函数贡献的梯度减小时，总梯度不会因为权重衰减而减小。诊断递减梯度的一种方法是检查梯度是否仅由权重衰减调制。

当 0.01 时，从 conv10 传播到 conv2 的梯度的标准差是 1/（5.9 × 4.22 × 2.92 × 2.14）= 1/（1.7 × 104）我们得出的。这个数字可以解释为什么在实验中观察到梯度递减。

还值得注意的是，输入信号的方差可以从第一层到最后一层大致保留。在输入信号未归一化的情况下（例如，它在 [-128，128] 范围内），其幅度可能非常大，以至于 softmax 运算符会溢出。解决方案是使输入信号归一化，但这可能会影响其他超参数。另一种解决方案是在所有或部分层之间的权重上包含一个小因子，例如，

L √
L 层为 1/128。在实践中，我们对前两个 fc 层使用 std 0.01，对最后一个 fc 层使用 0.001。这些数字小于应有的大小（例如，√ 2/4096），并且将

解决了范围约为 [-128， 128] 的图像的归一化问题。

对于 PReLU 情况下的初始化，很容易表明 Eqn.（10）变为：

$$\frac{1}{2}(1+a2)\,nlVar[wl] = 1, \qquad \forall l, \qquad (15)$$

其中 a 是系数的初始化值。如果 a = 0，则变为 ReLU 情况;如果 a = 1，则变为线性情况（与 [7] 相同）。同样，方程（14）变为 12（1 + a2）^nlVar[wl] = 1。

与 “Xavier” 初始化 [7] 的比较

我们的推导与“Xavier”初始化 [7] 之间的主要区别在于，我们解决了整流器非线性3。[7] 中的推导只考虑了线性情况，其结果由 nlVar[wl] = 1（前向情况）给出，它可以实现为标准√ 1/nl 的零均值高斯分布。当有

√
L 层，则 std 将为 1/ 2L 我们衍生的标准。这

然而，数字并不小到足以完全阻止我们论文中实际使用的模型的收敛（表 3，最多 22 层），如实验所示。图 2 比较了 22 层模型的收敛性。两种方法都能够使它们收敛。但是我们的 S 更早地开始减少错误。我们还研究了对准确性的可能影响。对于表 2 中的模型（使用 ReLU），“Xavier” 初始化方法导致 33.90/13.44 top- 1/top-5 误差。而我们的初始化方法导致 33.82/13.34 误差。我们没有观察到一个在准确性上明显优于另一个。

接下来，我们在多达 30 层（27 conv 和 3 fc）的极深模型上比较了这两种方法。我们在模型中添加多达 16 个 conv 层和 256 个 2×2 滤波器

3还有其他细微的考虑。在 [7] 中，采用源生方差进行均匀分布，并对前向和后向情况进行平均。但是，对于成熟分布以及仅针对前向或后向情况，采用他们的结论是很简单的。

表 1.图 3 显示了 30 层的收敛

型。我们的初始化能够使极深模型收敛。相反，“Xavier” 方法完全停滞了学习，并且 梯度在实验中监测到的正在缩小。

这些研究表明，我们已经准备好通过使用更原始的初始化方法来研究极其深入的、修正的模型。但是在我们目前在 ImageNet 上的实验中，我们还没有观察到训练极深模型的好处。例如，上述 30 层模型的 top-1/top-5 误差为 38.56/16.59，明显比表 2 中 14 层模型的误差（33.82/13.34）更严重。在小模型的研究 [10]、VGG 的大型模型 [25] 和语音识别 [34] 中也观察到准确性饱和或下降。这可能是因为增加深度的方法不合适，或者识别任务不够复杂。

尽管我们对极深模型的尝试没有显示出好处，但我们的初始化方法为进一步研究增加深度奠定了基础。我们希望这对其他更复杂的任务有所帮助。

2.3. 架构

以上调查提供了设计我们架构的指南，介绍如下。

我们的基线是表 3 中的 19 层模型（A）。为了更好地进行比较，我们还列出了 VGG-19 模型 [25]。我们的模型 A 对 VGG-19 进行了以下修改：（i）在第一层中，我们使用 7×7 的滤波器大小和 2 的步幅；(ii) 我们将两个最大特征图（224，112）上的其他三个 conv 层移动到较小的特征图（56，28，14）上。时间复杂度（表 3，最后一行）大致保持不变，因为较深的层具有更多的滤波器；(iii) 我们在第一个 FC 层之前使用空间金字塔池化（SPP）[11]。金字塔有 4 个级别 - 条柱的数量为 7×7、3×3、2×2 和 1×1，总共有 63 个条柱。

值得注意的是，我们没有证据表明我们的模型 A 是比 VGG-19 更好的架构，尽管我们的模型 A 的结果比 [25] 报告的 VGG-19 的结果要好。在我们早期使用较小规模增强的实验中，我们观察到我们的模型 A 和我们复制的 VGG-19（使用 SPP 和我们的初始化）是可比的。使用模型 A 的主要目的是提高运行速度。当它们的时间复杂度相同时，较大特征图上的卷 层 层 器的实际运行时间比较小特征图上的卷 层 层 器慢。在我们的四 GPU 实现中，我们的模型 A 每个小批量需要 2.6 秒（128），而我们复制的 VGG-19 需要 3.0 秒，在四个 Nvidia K20 GPU 上进行评估。

在表 3 中，我们的模型 B 是 A 的更深版本。它有三个额外的 conv 层。我们的模型 C 是 B 的更宽（具有更多过滤器）版本。宽度大大增加了复杂性，其时间复杂度约为 B 的 2.3×（第 3 卷，最后一行）。在 4 个 K20 GPU 上训练 A/B 或在 8 个 K40 GPU 上训练 C 大约需要 3-4 周。

我们选择增加模型宽度而不是深度，因为更深的模型只会减少准确性的改进甚至下降。在最近对小型模型的实验中 [10]，已经发现积极增加深度会导致精度饱和或降低。在 VGG 论文 [25] 中，16 层和 19 层模型的性能相当。在 [34] 的语音识别研究中，当使用超过 9 个隐藏层（全部为 fc）时，深度模型会退化。我们推测，类似的降级也可能发生在 ImageNet 的较大模型上。我们监测了一些极深模型的训练过程（在表 3 的 B 上增加了 3 到 9 层），发现训练和测试错误率在前 20 个 epoch 中都有所下降（但由于时间预算有限，我们没有运行到最后，因此目前还没有确凿的证据表明这些大型且过深的模型最终会退化）。由于可能的性能下降，我们选择不进一步增加这些大型模型的深度。

另一方面，最近对小型数据集的研究 [5] 表明，卷积层中参数数量的增加应该会提高准确性。此数字取决于深度和宽度。因此，我们选择增加 conv 层的宽度以获得更高容量的模型。

虽然表 3 中的所有模型都非常大，但我们没有观察到严重的过拟合。我们将其归因于整个训练过程中使用的侵略性数据增强，如下所述。

3. 实施细节

训练

我们的训练算法主要遵循 [16，13，2，11，25]。从较短边为 s 的调整大小图像中，随机采样 224×224 裁剪，减去每像素平均值。标度 s 在 [256，512] 范围内随机抖动，紧随 [25]。一半的随机样本水平翻转 [16]。还使用了随机颜色更改 [16]。

与 [25] 仅在微调期间应用刻度抖动不同，我们从训练开始就应用它。此外，与 [25] 使用较浅的模型初始化更深的模型不同，我们使用第 2.2 节中描述的初始化直接训练非常深的模型（我们使用方程（14））。我们的端到端训练可能有助于提高准确性，因为它可以避免较差的局部最优值。

其他可能很重要的超参数如下。权重衰减为 0.0005，动量为 0.9。Dropout（50%）用于前两个 fc 层。小批量大小固定为 128。学习率为 1e-2、1e-3、

输入大小	VGG-19 [25]	型号 A	型号 B	型号 C
224	3×3, 64 3×3、64 2×2 maxpool、/2	7×7, 96, /2	7×7, 96, /2	7×7, 96, /2
112	3×3、128、 3×3、128 2×2 maxpool、/2	2×2 maxpool，/2	2×2 maxpool，/2	2×2 maxpool，/2
56	3×3, 256 3×3, 256 3×3, 256 3×3, 256 2×2 maxpool，/2	3×3, 256 3×3, 256 3×3, 256 3×3, 256 3×3, 256 2×2 maxpool，/2	3×3, 256 3×3, 256 3×3, 256 3×3, 256 3×3, 256 3×3, 256 2×2 maxpool，/2	3×3, 384 3×3, 384 3×3, 384 3×3, 384 3×3, 384 3×3, 384 2×2 maxpool，/2
28	3×3, 512 3×3, 512 3×3, 512 3×3, 512 2×2 maxpool，/2	3×3, 512 3×3, 512 3×3, 512 3×3, 512 3×3, 512 2×2 maxpool，/2	3×3, 512 3×3, 512 3×3, 512 3×3, 512 3×3, 512 3×3, 512 2×2 maxpool，/2	3×3, 768 3×3, 768 3×3, 768 3×3, 768 3×3, 768 3×3, 768 2×2 maxpool，/2
14	3×3, 512 3×3, 512 3×3, 512 3×3, 512 2×2 maxpool，/2	3×3, 512 3×3, 512 3×3, 512 3×3, 512 3×3, 512 spp，{7，3，2，1}	3×3, 512 3×3, 512 3×3, 512 3×3, 512 3×3, 512 3×3, 512 spp，{7，3，2，1}	3×3, 896 3×3, 896 3×3, 896 3×3, 896 3×3, 896 3×3, 896 spp，{7，3，2，1}
FC1	4096			
FC2	4096			
FC3	1000			
深度（conv+fc）	19	19	22	22
复杂性（Ops.，×1010）	1.96	1.90	2.32	5.30

表 3.大型模型的架构。这里的 "/2" 表示 2 的步幅。

和 1e-4，并在误差稳定时切换。每个模型的纪元总数约为 80 个。

测试

我们采用"功能多视图测试"的策略

地图"[11]。我们在 [24， 25] 中使用密集滑动窗口方法进一步验证了这一策略。

我们首先在调整大小后的完整图像上应用卷积层，并获得最后的卷积特征图。在特征图中，每个 14×14 个窗口都使用 SPP 层 [11] 进行池化。然后，将 fc 层应用于池化特征以计算分数。这也是在水平翻转的图像上完成的。所有密集滑动窗的分数是平均的 [24,25]。我们进一步组合了多个尺度的结果，如 [11]。

多 GPU 实现

我们采用 Krizhevsky 方法 [15] 的简单变体在多个 GPU 上进行并行训练。我们在 conv 层上采用 "数据 paral- lelism" [15]。GPU 在第一个 fc 层之前同步。然后，在单个 GPU 上执行 fc 层的正向/向后传播 - 这意味着我们不会并行化 fc 层的计算。fc 层的时间成本较低，因此无需并行化它们。这导致了比 [15] 中的 "模型并行" 更简单的实现。此外，由于滤波器响应的通信，模型并行性会带来一些开销，并且并不比在单个 GPU 上计算 fc 层更快。

我们在 Caffe 库的修改中实现了上述算法 [14]。我们不增加小批量大小（128），因为准确性可能会降低 [15]。对于本文中的大型模型，我们观察到使用 4 个 GPU 的加速提高了 3.8 倍，使用 8 个 GPU 的加速提高了 6.0 倍。

型号 A	ReLU 系列		PReLU	
比例尺 S	第一名	前 5 名	第一名	前 5 名
256	26.25	8.25	25.81	8.08
384	24.77	7.26	24.20	7.03
480	25.46	7.63	24.83	7.39
多尺度	24.02	6.51	22.97	6.28

表 4.使用 Dense testing 在 Ima- geNet 2012 中模型 A 上 ReLU/PReLU 的比较。

4. ImageNet 上的实验

我们在 1000 类 ImageNet 2012 数据集 [22] 上进行了实验，该数据集包含大约 120 万张训练图像、50,000 张验证图像和 100,000 张测试图像（没有已发布的标签）。结果通过前 1/前 5 错误率来衡量 [22]。我们只使用提供的数据进行训练。所有结果都在验证集上进行评估，但表 7 中的最终结果除外，这些结果在测试集上进行评估。前 5 名错误率是官方用于对分类挑战中方法进行排名的指标 [22]。

ReLU 和 PReLU 之间的比较

在表 4 中，我们比较了大型模型 A 上的 ReLU 和 PReLU。我们使用 PReLU 的通道版本。为了公平地进行比较，两个 ReLU/PReLU 模型都是使用相同的 epoch 总数进行训练的，并且学习率也在运行相同数量的 epoch 后切换。

表 4 显示了三个尺度和多

秤组合。最好的单标度是 384，可能是因为它处于抖动范围的中间 [256，512]。对于多尺度组合，与 ReLU 相比，PReLU 将 top-1 误差降低了 1.05%，top-5 误差降低了 0.23%。表 2 和表 4 中的结果一致表明，PReLU 改进了小型和大型模型。这种改进几乎不需要计算成本。

单模型结果的比较

接下来，我们比较单一模型的结果。我们首先在表 5 中展示了 10 视图测试结果 [16]。在这里，每个视图都是 224 裁剪。VGG-16 的 10 视图结果基于我们使用公开发布的模型 [25] 进行的测试，因为 [25] 中没有报道。我们最好的 10 次观看结果是 7.38%（表 5）。我们的其他模型也优于现有结果。

表 6 显示了单模型结果的比较，这些结果都是使用多尺度和多视图（或密集）测试获得的。我们的结果表示为 MSRA。我们的基线模型（A+ReLU，6.51%）已经大大优于 VGG-19 的最新更新 [25]（arXiv v5）中报告的 7.1% 的现有最佳单一模型结果。我们是-

相信这种收益主要是由于我们的端到端训练，不需要预训练的浅层模型。

此外，我们最好的单一模型（C，PReLU）有 5.71%

top-5 错误。这个结果甚至比以前的所有多模型结果都要好（表 7）。将 A+PReLU 与 B+PReLU 进行比较，我们可以看到 19 层模型和 22 层模型的性能相当。另一方面，增加宽度（C vs. B，表 6）仍然可以提高准确性。这表明，当模型足够深时，宽度成为准确性的重要因素。

多模型结果的比较

我们结合了 6 个模型，包括表 6 中的模型。目前，我们只用架构 C 训练了一个模型。其他模型的精度不如 C，差 出 可观的边际。我们推测，通过使用更少的更强模型，我们可以获得更好的结果。

多模型结果见表 7。我们的结果是测试集上 4.94% 的前 5 个错误。此数字由 ILSVRC 服务器评估，因为测试集的标签未发布。我们的结果比 ILSVRC 2014 获胜者（GoogLeNet，6.66% [29]）高出 1.7%，这意味着相对提高了 ~26%。这也比最新结果（百度，5.98% [32]）相对提高了 ~17%。

结果分析

图 4 显示了一些示例验证图像 - 通过我们的方法成功完全分类。除了正确排列的标签外，我们还关注前 5 名结果中的其他 4 个预测。这四个标签中的一些是多对象图像中的其他对象，例如，“马车”图像（图 4，第 1 行，第 1 列）包含一个“迷你总线”，它也可以被算法识别。这四个标签中的一些是由于相似类别之间的不确定性，例如，“粘稠”图像（图 4，第 2 行，第 1 列）预测了其他鸟类的标签。

图 6 显示了我们在测试集上结果的每个班级前 5 名误差（平均 4.94%），按升序显示。我们的结果在 113 个类中没有前 5 个错误 - 这些类中的图像都被正确分类。前 5 名错误最高的三个类别是“letter opener”（49%）、“spotlight”（38%）和“restaurant”（36%）。错误是由于存在多个对象、小对象或较大的类内差异。图 5 显示了被我们的方法错误分类在这三个类别中的一些示例图像。一些预测的标签仍然有一定的意义。

在图 7 中，我们显示了我们的成绩（平均 4.94%）和我们团队在 ILSVRC 2014 中的比赛成绩（平均 8.06%）之间前 5 名错误率的每班级差异。错误率在 824 个类别中降低，在 127 个类别中保持不变，在 49 个类别中增加。

型	第一名	前 5 名
MSRA [11]	29.68	10.95
VGG-16 [25]	28.07†	9.33†
谷歌网 [29]	-	9.15
A，ReLU	26.48	8.59
A，PReLU	25.59	8.23
B，PReLU	25.53	8.13
C，PReLU	24.27	7.38

表 5.ImageNet 2012 val set 的单模型 10 视图结果。†：根据我们的测试。

	团队	第一名	前 5 名
参加比赛 ILSVRC 14	MSRA [11]	27.86	9.08†
	VGG [25]	-	8.43†
	谷歌网 [29]	-	7.89
赛后	VGG [25]（arXiv v2）	24.8	7.5
	VGG [25]（arXiv v5）	24.4	7.1
	百度 [32]	24.88	7.42
	MSRA（A，ReLU）	24.02	6.51
	MSRA（A，PReLU）	22.97	6.28
	MSRA（B，PReLU）	22.85	6.27
	MSRA（C，PReLU）	21.59	5.71

表 6.ImageNet 2012 val set 的单模型结果。†：从测试集评估。

	团队	前 5 名（测试）
参加比赛 ILSVRC 14	MSRA、SPP 网络 [11]	8.06
	VGG [25]	7.32
	谷歌网 [29]	6.66
赛后	VGG [25]（arXiv v5）	6.8
	百度 [32]	5.98
	MSRA、PReLU 网络	4.94

表 7.ImageNet 2012 测试集的多模型结果。

与 [22] 的人类表现的比较

Russakovsky 等人 [22] 最近报道说，在 ImageNet 数据集上，人类性能产生了 5.1% 的前 5 个误差。此数字由人工注释者实现，该注释者在验证图像方面接受过良好的培训，可以更好地了解相关类的存在。在注释测试图像时，人工注释者会得到一个特殊的界面，其中每个类标题都附有一行 13 个示例训练图像。报告的人类性能是根据 1500 张测试图像的随机子集估计的。

我们的结果（4.94%）超过了报告的人类水平表现。据我们所知，我们的结果是首次在视觉识别挑战中超越人类。[22] 中的分析表明，两种主要类型的人为错误来自细粒度的认知和阶级无意识。[22] 中的调查表明一

认为算法可以在细粒度识别方面做得更好（例如，数据集中有 120 种狗）。图 4 的第二行显示了我们的方法成功识别的一些细粒度对象示例 - “coucal”、“komondor” 和 “yellow lady's slipper”。虽然人类可以很容易地将这些物体识别为鸟、狗和花，但对于大多数人来说，分辨他们的物种并非易事。从不利的一面来看，我们的算法仍然会在对人类来说并不困难的情况下犯错误，尤其是对于那些需要上下文理解或高级知识的情况（例如，图 5 中的 “聚光灯” 图像）。

虽然我们的算法在这个特定的数据集上产生了更好的结果，但这并不意味着机器视觉在物体识别方面总体上优于人类视觉。关于识别基本对象类别（即日常生活中的常见对象或概念），例如 Pascal VOC

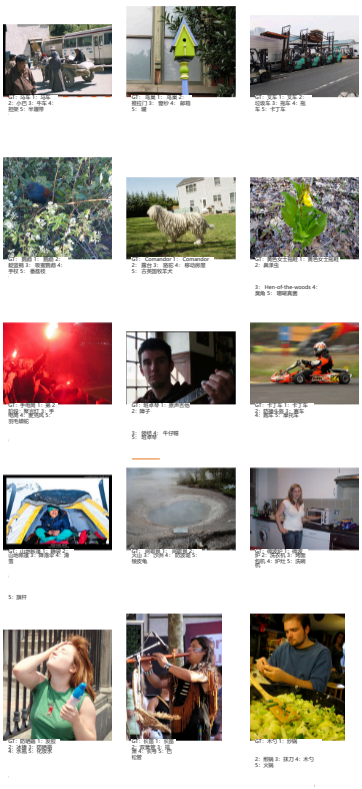

图 4.通过我们的方法成功分类的示例验证图像。对于每张图像，都列出了 ground-truth 标签和我们前方法预测的前 5 个 la- bels。

任务 [6] 中，机器在以下情况下仍然存在明显的错误

对人类来说是微不足道的。尽管如此，我们相信，我们的研究成果显示了机器算法在视觉识别方面与人类水平的性能相匹配的巨大潜力。

图 5.示例验证图像被我们的方法错误地分类。属于测试误差最高的三个类 5 个。顶部："letter opener"（49% 的前 5 名测试错误）、中层："荧光灯"（38%）、底部："餐厅"（36%）。对于每张图像，都会列出我们的方法预测的 ground-truth 标签和前 5 个标签。

引用

[1] F. Agostinelli、M. Hoffman、P. Sadowski 和 P. Baldi。学习激活函数以改进深度神经网络 - 工作。arXiv：1412.6830,2014 年。

[2] K. Chatfield、K. Simonyan、A. Vedaldi 和 A. Zisserman。魔鬼在细节中的回归：深入研究 convo- lutional nets。在 BMVC，2014 年。

[3] D. Ciresan、U. Meier 和 J. Schmidhuber。用于图像分类的多列深度神经网络。在 CVPR，2012 年。

[4] J. 邓、W. Dong、R. Socher、L.-J.李、李国和L. Fei-Fei。Imagenet：大规模分层图像数据库。在 CVPR，2009 年。

[5] D. Eigen、J. Rolfe、R. Fergus 和 Y. LeCun。使用递归卷积网络了解深度架构。arXiv：1312.1847,2013 年。

[6] M. Everingham、L. Van Gool、CK Williams、J. Winn 和 A. Zisserman。Pascal 视觉对象类（VOC）挑战赛。IJCV，第 303-338 页，2010 年。

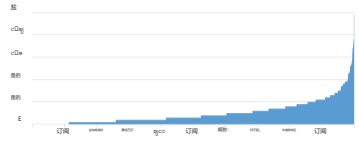

图 6.测试集中我们结果的每个班级前 5 个错误（平均 4.94%）。错误按升序显示。

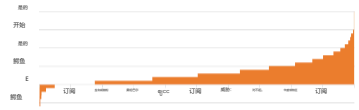

图 7.在测试集中，我们的结果（平均 4.94%）和我们团队在 ILSVRC 2014 的比赛中成绩（平均 8.06%）之间的前 5 名错误率差异，按升序显示。正数表示错误率降低。

[7] X. Glorot 和 Y. Bengio。了解训练深度前馈神经网络的难度。在人工智能与统计国际会议中，第 249-256 页，2010 年。

[8] X. Glorot、A. Bordes 和 Y. Bengio。深度稀疏整流器网络。第 14 届人工智能与统计国际会议论文集，第 315-323 页，2011 年。

[9] I. J. Goodfellow， D. Warde-Farley， M. Mirza， A. Courville，
和 Y. Bengio 的作者。Maxout 网络。arXiv：1302.4389,2013 年。

[10] K.He 和 J. Sun. 卷积神经网络
紧张的时间成本。arXiv：1412.1710,2014 年。[11] K. He， X. Zhang， S. 任， 和 J. Sun. 空间金字塔池-
ing 在深度卷积网络中进行视觉识别。arXiv：1406.4729v2,2014 年。

[12] G. E. Hinton， N. Srivastava， A. Krizhevsky， I. Sutskever，和 R. R. Salakhutdinov.通过预先通风特征检测器的协适应来改进神经网络。arXiv：1207.0580,2012 年。

[13] A. G. 魏华德。对基于深度卷积神经网络的图像分类进行了一些改进。arXiv：1312.5402,2013 年。

[14] Y. Jia， E. Shelhamer， J. Donahue， S. Karayev， J. Long， R. Gir- shick， S. Guadarrama，和 T. Darrell.Caffe：用于快速特征嵌入的卷积架构。arXiv：1408.5093,2014 年。

[15] A. 克里热夫斯基。并行化卷积的一个奇怪技巧 -
tional 神经网络。arXiv：1404.5997,2014 年。[16] A. Krizhevsky、I. Sutskever 和 G. Hinton。Imagenet clas-
使用深度卷积神经网络进行 sification 的分析。在 NIPS 中，2012 年。

[17] Y. LeCun， B. Boser， J. S. Denker， D. Henderson， R. E. Howard， W. Hubbard，和 L. D. Jackel.反向传播
适用于手写邮政编码识别。神经计算，1989 年。

[18] C.-Y.Lee， S. Xie， P. Gallagher， Z. Zhang， and Z. Tu. Deeply-
监督网络。arXiv：1409.5185,2014 年。[19] M. Lin， Q. Chen，和 S. Yan.
网络中的网络。
arXiv：1312.4400,2013 年。[20] AL Maas、AY Hannun 和 AY Ng. 整流器非林
earities 改进了神经网络声学模型。在 ICML，2013 年。

[21] V. Nair 和 GE Hinton。整流线性装置改进了受限玻尔兹曼机。在 ICML 中，第 807-814 页，2010 年。

[22] O. 鲁萨科夫斯基， J. 邓， H. Su， J. 克劳斯， S. Satheesh， S. 马， Z. 黄， A. 卡尔帕西， A. 科斯拉， M. 伯恩斯坦， et al.Imagenet 大规模视觉识别挑战赛。arXiv：1409.0575,2014 年。

[23] AM Saxe、JL McClelland 和 S. Ganguli。深度线性神经网络中学习非线性动力学的精确解决方案。arXiv：1312.6120,2013 年。

[24] P. Sermanet， D. Eigen， X. Zhang， M. Mathieu， R. Fergus， 和 Y. LeCun.Overfeat：使用卷积网络集成识别、定位和检测。2014.

[25] K. Simonyan 和 A. Zisserman。 非常深的
用于大规模图像识别的卷积网络。arXiv：1409.1556,2014 年。

[26] N. Srivastava， G. Hinton， A. Krizhevsky， I. Sutskever，和 R. Salakhutdinov.Dropout：一种防止神经网络过拟合的简单方法。机器学习研究杂志，第 1929-1958 页，2014 年。

[27] R. K. Srivastava， J. Masci， S. Kazerounian， F. Gomez， 和 J. Schmidhuber.竞争计算。在 NIPS 中，第 2310-2318 页，2013 年。

[28] Y. Sun， Y. Chen， X. Wang， 和 X. Tang.通过联合身份验证进行深度学习人脸表示。在 NIPS 中，2014 年。

[29] C. Szegedy， W. Liu， Y. Jia， P. Sermanet， S. Reed， D. Anguelov， D. Erhan， V. Vanhoucke， and A. Rabinovich.更深入地使用卷积。arXiv：1409.4842,2014 年。

[30] Y. Taigman、M. Yang、M. Ranzato 和 L. Wolf。Deepface：缩小与人脸验证中人类水平性能的差距。在 CVPR，2014 年。

[31] L. Wan， M. Zeiler， S. Zhang， Y. L. Cun， 和 R. Fergus.使用 dropconnect 对神经网络进行注册。在 ICML 中，第 1058-1066 页，2013 年。

[32] R. Wu， S. Yan， Y. Shan， Q. Dang， and G. Sun. 深图：
扩大图像识别。arXiv：1501.02876,2015 年。[33] M. D. Zeiler 和 R. Fergus。可视化和理解
卷积神经网络。在 ECCV，2014 年。[34] M. D. Zeiler， M. Ranzato， R. Monga， M. 毛， K. Yang， Q. V.
Le， P. Nguyen， A. Senior， V. Vanhoucke， J. Dean， 和 GE Hinton。在用于语音处理的整流线性单元上。在 ICASSP，2013 年。