

深入卷积网络

Christian Szegedy Google Inc. Wei Liu 北卡罗来纳大学教堂山分校 Yangqing Jia Google Inc. Pierre Sermanet Google Inc. Scott Reed 密歇根大学 Dragomir Anguelov Google Inc. Dumitru Erhan Google Inc. Vincent Vanhoucke Google Inc. Andrew Rabinovich Google Inc.

摘要

我们提出了一种代号为Inception的深度卷积神经网络架构，该架构在2014年ImageNet大规模视觉识别挑战赛（ILSVRC14）中为分类和检测任务创造了新的技术标杆。该架构的主要特点在于提升网络内部计算资源的利用率。这是通过精心设计实现的，使得网络在保持计算量不变的前提下，能够增加深度和宽度。为了优化性能，架构设计基于赫布原理和多尺度处理的直观理念。我们在ILSVRC14中提交的一个具体实现名为GoogLeNet，这是一个22层的深度网络，其性能在分类和检测任务中得到了验证。

1 引言

在过去的三年中，主要由于深度学习的进步，更具体地说是卷积网络[10]的发展，图像识别和目标检测的质量以惊人的速度提升。一个令人鼓舞的消息是，这种进步大多不仅仅是更强大的硬件、更大的数据集和更大的模型的结果，而主要是新思想、算法和改进的网络架构的产物。例如，在ILSVRC 2014竞赛中，除了用于检测目的的同一竞赛分类数据集外，顶级参赛作品并未使用新的数据源。我们在ILSVRC 2014中提交的GoogLeNet实际上比两年前Krizhevsky等人[9]的获奖架构少用了12{v*}个参数，同时显著提高了准确性。目标检测方面的最大收益并非仅仅来自深度网络或更大模型的使用，而是来自深度架构与经典计算机视觉的协同作用，例如Girshick等人[6]提出的R-CNN算法。

另一个值得注意的因素是，随着移动和嵌入式计算的持续发展，我们算法的效率——尤其是其功耗和内存使用——变得愈发重要。值得注意的是，本文提出的深度架构在设计过程中考虑了这一因素，而非仅仅执着于准确率数字。在大多数实验中，模型的设计目标是在推理阶段保持15亿次乘法运算的计算预算，以确保它们不会沦为纯粹的学术探索，而是能够在合理成本下应用于现实世界，即使是处理大规模数据集也不例外。

本文聚焦于一种高效的计算机视觉深度神经网络架构，代号为Inception。该名称源于Lin等人[12]提出的“网络中的网络”论文，并结合了著名的网络流行语“我们需要更深入”[1]。此处“深度”一词具有双重含义：首先指我们通过“Inception模块”形式引入了新的组织层级，同时也更直接地指网络深度的增加。总体而言，可将Inception模型视为[12]研究的逻辑终局，并受到Arora等人[2]理论工作的启发与指导。该架构的优势在ILSVRC 2014分类与检测挑战赛中得到实验验证，其性能显著超越了当时的最先进水平。

2 相关工作

从LeNet-5 [10]开始，卷积神经网络（CNN）通常具有标准结构——堆叠的卷积层（可选地后接对比度归一化和最大池化）之后是一个或多个全连接层。这种基础设计的变体在图像分类文献中十分普遍，并在MNIST、CIFAR数据集上取得了迄今最佳结果，尤其是在ImageNet分类挑战赛上表现突出[9, 21]。对于更大规模的数据集（如ImageNet），近期的趋势是增加网络层数[12]和层尺寸[21, 14]，同时使用dropout[7]来解决过拟合问题。

尽管有人担心最大池化层会导致精确空间信息的丢失，但[9]中采用的相同卷积网络架构已成功应用于定位[9, 14]、目标检测[6, 14, 18, 5]以及人体姿态估计[19]。受灵长类视觉皮层神经科学模型的启发，Serre等人[15]采用了一系列不同尺寸的固定Gabor滤波器来处理多尺度问题，这与Inception模型类似。然而，与[15]中固定的双层深度模型不同，Inception模型中的所有滤波器都是通过学习得到的。此外，Inception层被多次重复堆叠，在GoogLeNet模型中形成了22层的深度架构。

Network-in-Network 是由 Lin 等人 [12] 提出的一种方法，旨在增强神经网络的表示能力。当应用于卷积层时，该方法可被视为添加额外的 1×1 卷积层，通常后接线性整流激活函数 [9]。这使得它能够轻松集成到当前的 CNN 流程中。我们在架构中大量使用了这种方法。然而，在我们的设置中， 1×1 卷积具有双重作用：最关键的是，它们主要被用作降维模块，以消除计算瓶颈，否则这些瓶颈会限制我们网络的规模。这不仅允许我们增加网络的深度，还能增加网络的宽度，而不会造成显著的性能损失。

当前物体检测的主流方法是Girshick等人提出的基于区域的卷积神经网络（R-CNN）[6]。R-CNN将整体检测问题分解为两个子问题：首先利用颜色和超像素一致性等低级线索，以类别无关的方式生成潜在物体候选区域；随后使用CNN分类器对这些区域进行物体类别识别。这种两阶段方法既利用了低级线索在边界框分割上的准确性，又发挥了先进CNN强大的分类能力。我们在检测任务中采用了类似流程，但对两个阶段均进行了改进探索：例如采用多框预测[5]提升物体边界框召回率，并运用集成学习方法优化候选边界框的分类效果。

3 动机与高层考量

提升深度神经网络性能最直接的方法是增加其规模。这既包括增加网络的深度——即层数，也包括增加其宽度：即每层的单元数量。这是一种简单且安全的训练高质量模型的方法，尤其是在有大量标注训练数据可用的情况下。然而，这种简单的解决方案存在两个主要缺点。

更大的规模通常意味着更多的参数，这使得扩大后的网络更容易过拟合，尤其是在训练集中有标签样本数量有限的情况下。这可能成为一个主要瓶颈，因为创建高质量的训练集可能很棘手。



图1: ILSVRC 2014分类挑战赛1000个类别中的两个不同类别。

且成本高昂，尤其是在需要专家人工评估员来区分细粒度视觉类别（如图像网络中的那些类别，即使在1000类的ILSVRC子集中也是如此）时，如图1所示。

网络规模均匀增大的另一个缺点是计算资源的使用急剧增加。例如，在一个深度视觉网络中，如果两个卷积层串联，其滤波器数量的任何均匀增加都会导致计算量的二次增长。如果新增的容量被低效利用（例如，如果大多数权重最终接近于零），那么大量计算就被浪费了。由于实际计算预算总是有限的，即使主要目标是提高结果质量，也应优先考虑计算资源的有效分配，而非盲目扩大规模。

解决这两个问题的根本方法，最终在于从全连接架构转向稀疏连接架构，即使在卷积内部也是如此。除了模仿生物系统外，由于Arora等人[2]的开创性工作，这种方法还具有更坚实的理论基础。他们的主要结果表明：如果数据集的概率分布可以通过一个大型、高度稀疏的深度神经网络表示，那么可以通过分析最后一层激活值的相关统计量，并将输出高度相关的神经元聚类，从而逐层构建出最优的网络拓扑结构。尽管严格的数学证明需要非常强的条件，但这一结论与著名的赫布原理——“同时激活的神经元会相互连接”——产生共鸣的事实表明，其核心思想在实践中即使在不那么严格的条件下也适用。

从不利的一面来看，当今的计算基础设施在处理非均匀稀疏数据结构的数值计算时效率极低。即使算术运算的数量减少了 $100\times$ ，查找和缓存未命中的开销仍然占据主导地位，以至于转向稀疏矩阵并不会带来回报。随着持续改进、高度优化的数值库的使用，这一差距进一步扩大，这些库能够实现极快的稠密矩阵乘法，并充分利用底层CPU或GPU硬件的细微特性[16, 9]。此外，非均匀稀疏模型需要更复杂的工程和计算基础设施。当前大多数面向视觉的机器学习系统仅通过采用卷积来利用空间域的稀疏性。然而，卷积是通过与前一层的图像块建立稠密连接集合来实现的。自[11]以来，卷积神经网络传统上在特征维度使用随机和稀疏的连接表，以打破对称性并改善学习效果；但为了更好优化并行计算，[9]之后趋势又回归到全连接。结构的均匀性、大量的滤波器以及更大的批处理规模，使得高效的稠密计算得以充分利用。

这就引出了一个问题：是否存在一种有希望的中间步骤：一种架构，它能够利用理论所建议的额外稀疏性，甚至在滤波器级别上，同时充分利用我们的

当前硬件通过利用密集矩阵计算来提升性能。关于稀疏矩阵计算的大量文献（例如[3]）表明，将稀疏矩阵聚类为相对密集的子矩阵往往能为稀疏矩阵乘法带来最先进的实用性能。可以合理推测，类似方法在不久的将来将被用于非均匀深度学习架构的自动化构建中。

Inception架构最初是作为第一作者的一项案例研究，旨在评估一种复杂网络拓扑构建算法的假设输出。该算法试图逼近[2]中为视觉网络所暗示的稀疏结构，并通过密集且易于获取的组件来覆盖这一假设结果。尽管这是一项高度推测性的尝试，但在仅对拓扑结构进行两次精确选择的迭代后，我们已经能看到相对于基于[12]的参考架构所取得的适度收益。经过对学习率、超参数及改进训练方法的进一步调优，我们确认了由此产生的Inception架构在定位和物体检测背景下作为[6]和[5]的基础网络尤为有效。有趣的是，尽管最初的大部分架构选择曾受到质疑并经过彻底测试，但它们最终被证明至少是局部最优的。

不过，我们必须谨慎：尽管所提出的架构在计算机视觉领域取得了成功，但其质量是否可归功于指导其构建的原则仍存疑问。要确证这一点，需要更深入分析和验证：例如，基于下述原则的自动化工具能否为视觉网络找到类似但更优的拓扑结构。最有力的证明将是，一个自动化系统能够使用相同算法，在其他领域创建出带来类似性能提升的网络拓扑，即使其全局架构外观迥异。至少，Inception架构的初步成功为这一方向的未来激动人心的工作提供了坚实的动力。

4 架构细节

Inception架构的核心思想在于探索如何利用现成的密集组件来近似并覆盖卷积视觉网络中最优的局部稀疏结构。需注意，平移不变性的假设意味着我们的网络将由卷积构建模块组成。我们只需找到最优的局部结构并在空间上重复它。Arora等人[2]提出了一种逐层构建方法：分析前一层的相关性统计量，将高相关性的单元聚类成组。这些聚类将形成下一层的单元，并与前一层的单元相连接。我们假设早期层的每个单元对应输入图像的某个区域，这些单元被分组为滤波器组。在靠近输入的底层网络中，相关单元会集中在局部区域。这意味着我们将得到大量集中于单个区域的聚类，它们可以通过下一层的 1×1 卷积层来覆盖，正如文献[12]所建议的。然而，我们也可以预期会出现数量较少、空间分布更广的聚类，这些聚类需要更大尺寸的卷积核来覆盖，且随着区域增大，所需卷积核数量会递减。为避免图像块对齐问题，当前Inception架构的实现将滤波器尺寸限制为 1×1 、 3×3 和 5×5 ，但这更多是出于便利性而非必要性考量。该架构实质上是各卷积层的输出滤波器组拼接为单一输出向量，作为下一阶段的输入。此外，由于池化操作对当前先进卷积网络的成功至关重要，在每一阶段增加并行的池化路径也应能带来额外增益（见图2(a)）。

随着这些“初始模块”层层叠加，它们的输出相关统计量必然发生变化：当更高层次捕捉到更高抽象的特征时，其空间集中度预计会降低，这表明随着向更高层级移动， 3×3 和 5×5 卷积的比例应当增加。

上述模块的一个大问题，至少在这种天真的形式下，是即使数量不多的 5×5 卷积，在具有大量滤波器的卷积层之上也可能成本过高。一旦将池化单元加入其中，这个问题变得更加明显：它们的输出滤波器数量等于前一阶段的滤波器数量。将池化层的输出与卷积层的输出合并，将不可避免地

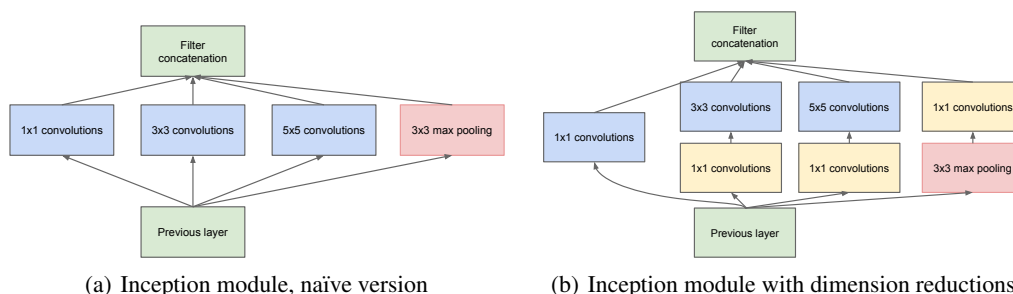


图2: Inception模块

阶段间输出数量的增加。即使这种架构可能覆盖了最优稀疏结构，其实现效率也会非常低下，导致在少数几个阶段内计算量急剧膨胀。

这引出了所提出架构的第二个核心理念：在计算需求可能过度增长的地方，审慎地应用降维和投影操作。这一理念基于嵌入技术的成功实践：即使是低维嵌入，也可能包含相对较大图像块的大量信息。然而，嵌入以密集压缩的形式表示信息，而压缩信息往往更难建模。我们希望保持表征在多数位置具有稀疏性（如文献[2]所述条件的要求），仅当信号需要大规模聚合时才进行压缩。具体而言，在计算密集的 3×3 和 5×5 卷积之前，会使用 1×1 卷积执行降维处理。这些卷积不仅用于降维，还结合了整流线性激活函数，从而实现双重功能。最终结果如图2(b)所示。

一般来说，Inception网络是由上述类型的模块相互堆叠而成的网络，偶尔会使用步长为2的最大池化层来将网格分辨率减半。出于技术原因（训练期间的内存效率），似乎只在较高层开始使用Inception模块，而将较低层保持为传统的卷积形式更为有利。这并非绝对必要，只是反映了我们当前实现中的一些基础设施效率不足。

这种架构的主要优势之一在于，它允许在每个阶段显著增加单元数量，而不会导致计算复杂度失控性激增。通过普遍采用降维操作，能够将最后阶段的大量输入滤波器屏蔽到下一层——先降低其维度，再使用大尺寸卷积核进行卷积运算。该设计另一个具有实际价值的方面是：它符合视觉信息应在多尺度处理后再聚合的直觉，从而使下一阶段能同时从不同尺度中提取特征。

计算资源的优化使用使得在不陷入计算困境的情况下，既能增加每个阶段的宽度，也能增加阶段的数量。利用Inception架构的另一种方式是创建性能稍逊但计算成本更低的版本。我们发现，所有内置的调节机制允许对计算资源进行可控平衡，从而构建出比性能相当的非Inception架构网络快2–至 $3 \times$ 倍的网络，但这目前仍需谨慎的人工设计。

5 GoogLeNet

在ILSVRC14竞赛中，我们选择GoogLeNet作为团队名称。此名旨在致敬Yann LeCun开创性的LeNet 5网络[10]。我们亦用GoogLeNet指代本次竞赛提交方案中采用的Inception架构具体实现。我们还使用了更深更广的Inception网络变体，其质量略逊一筹，但将其加入集成模型后，结果似乎有微弱提升。由于实验表明具体架构参数的影响相对有限，此处不再赘述该网络细节。

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

表1: Inception架构的GoogLeNet实现

次要。这里，为了演示目的，表1描述了最成功的特定实例（名为GoogLeNet）。在我们的集成模型中，有6个模型使用了完全相同的拓扑结构（采用不同的采样方法训练）。

所有卷积，包括Inception模块内部的卷积，均使用修正线性激活函数。我们网络中的感受野大小为 224×224 ，采用经过均值减除处理的RGB色彩通道。“#3×3 reduce”和“#5×5 reduce”表示在3×3和5×5卷积之前使用的降维层中1×1滤波器的数量。在pool proj列中，可以看到内置最大池化后投影层中1×1滤波器的数量。所有这些降维/投影层同样使用修正线性激活函数。

该网络在设计时充分考虑了计算效率和实用性，以便推理过程能够在包括计算资源有限的设备上运行，尤其是内存占用较低的情况。仅计算带参数的层时，网络深度为22层（若计入池化层则为27层）。用于构建网络的整体层数（独立构建模块）约为100层，但具体数值取决于所使用的机器学习基础设施系统。分类器前的平均池化操作借鉴了文献[12]，但我们的实现方式有所不同——我们增加了一个额外的线性层。这使得网络能够轻松适配和微调至其他标签集，虽然这主要是出于便利性考虑，我们预计其不会产生重大影响。实验发现，将全连接层替换为平均池化能使Top-1准确率提升约0.6%，但即使移除全连接层后，dropout技术的使用仍然至关重要。

考虑到网络深度相对较大，如何以有效方式将梯度反向传播至所有层是一个值得关注的问题。一个有趣的发现是：相对较浅的网络在此任务上表现优异，这表明网络中间层生成的特征应具备极强的区分性。通过在中间层添加辅助分类器，我们期望能够增强分类器低层阶段的判别能力，增加反向传播的梯度信号，并提供额外的正则化效果。这些分类器采用小型卷积网络的形式，附加在Inception（4a）和（4d）模块的输出端。训练过程中，其损失值会以折扣权重（辅助分类器损失权重设为0.3）加入网络总损失。在推理阶段，这些辅助网络将被移除。

的确切结构

侧边的额外网络，包括辅助分类器

如下：

- 一个平均池化层，使用5×5的滤波器尺寸和步幅3，为(4a)阶段生成 $4 \times 4 \times 512$ 的输出，为(4d)阶段生成 $4 \times 4 \times 528$ 的输出。

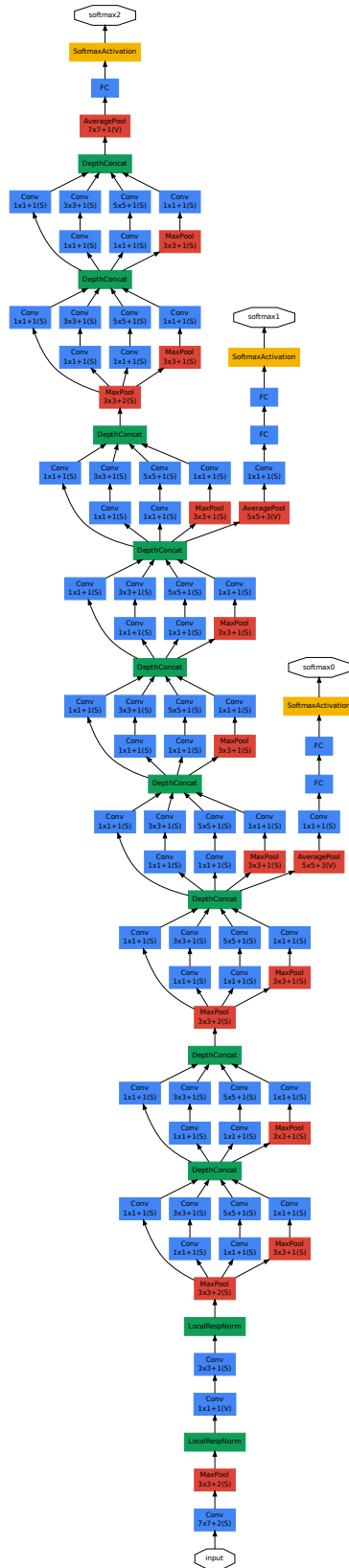


图3: 配备所有附加功能的GoogLeNet网络

- 一个具有128个滤波器用于降维和修正线性激活的 1×1 卷积。
- 一个具有1024个单元和修正线性激活的全连接层。
- 丢弃率为70%的dropout层。
- 一个带有softmax损失的线性层作为分类器（预测与主分类器相同的1000个类别，但在推理时被移除）。

所得网络的示意图如图 {v*} 所示。

关于第3点。

6 训练方法

我们的网络是使用DistBelief [4]分布式机器学习系统进行训练的，采用了适度的模型并行与数据并行策略。尽管我们仅使用了基于CPU的实现，但粗略估计表明，GoogLeNet网络可以在大约一周内通过少量高端GPU训练至收敛，主要限制在于内存使用量。我们的训练采用异步随机梯度下降法，动量为0.9 [17]，并采用固定的学习率调整策略（每8轮迭代将学习率降低4%）。在推理阶段使用的最终模型通过Polyak平均法 [13] 生成。

我们的图像采样方法在比赛前的几个月里发生了显著变化，已经收敛的模型曾使用其他选项进行训练，有时还结合了改变的超参数（如dropout和学习率），因此很难给出一个明确的指导来确定训练这些网络最有效的单一方法。更复杂的是，部分模型主要训练于相对较小的裁剪区域，另一些则基于较大的区域，这受到了[8]的启发。尽管如此，比赛后验证效果极佳的一种方案包括：采样不同尺寸的图像块，其大小均匀分布在图像面积的8%到100%之间，长宽比则在3/4和4/3之间随机选择。此外，我们发现Andrew Howard[8]提出的光度失真方法在一定程度上有助于对抗过拟合。另外，我们较晚才开始使用随机插值方法（双线性、区域、最近邻和立方插值，每种方法等概率选择）进行尺寸调整，并且与其他超参数更改结合使用，因此无法确定最终结果是否因其使用而获得了积极影响。

7 国际ILSVRC 2014 分类挑战赛设置与 R

结果

ILSVRC 2014分类挑战赛的任务是将图像分类到ImageNet层次结构中的1000个叶节点类别之一。该赛事提供约120万张训练图像、5万张验证图像和10万张测试图像。每张图像对应一个真实类别标签，评估标准基于分类器预测得分最高的类别。通常报告两个指标：top-1准确率（将真实类别与预测最高概率的类别比较）和top-5错误率（将真实类别与预测概率前五的类别比较：只要真实类别出现在前五预测中即视为正确分类，不要求具体排名）。本挑战赛采用top-5错误率作为排名依据。

我们参与挑战时未使用任何外部数据进行训练。除了本文前述的训练技术外，我们在测试阶段还采用了一系列技术以获取更高性能，具体内容如下详述。

1. 我们独立训练了7个相同版本的GoogLeNet模型（包括一个更宽的版本），并对它们进行了集成预测。这些模型采用相同的初始化（甚至使用相同的初始权重，主要是由于疏忽）和学习率策略，它们仅在采样方法以及输入图像的随机顺序上有所不同。
2. 在测试过程中，我们采用了比Krizhevsky等人[9]更激进的裁剪方法。具体来说，我们将图像缩放到4个尺度，其中较短边（高度或宽度）分别为256、288、320和352，取这些缩放后图像的左、中、右正方形区域（对于纵向图像，则取上、中、下正方形区域）。接着，对每个正方形区域，我们取4个角落和中心 224×224 的裁剪区域，以及

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

表2：分类性能

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

表3：GoogLeNet分类性能细分

将图像调整为 224×224 的正方形，并生成其镜像版本。这样每张图像可得到 $4 \times 3 \times 6 \times 2 = 144$ 个裁剪区域。Andrew Howard [8] 在去年的参赛方案中采用了类似方法，我们通过实验验证其效果略逊于本文提出的方案。需要指出的是，在实际应用中可能无需如此密集的裁剪，因为当裁剪数量达到合理范围后，继续增加裁剪区域带来的收益会逐渐递减（后续将对此进行说明）。

3. 通过对多个裁剪区域和所有独立分类器的softmax概率进行平均，以获得最终预测。在我们的实验中，我们在验证数据上分析了其他方法，例如对裁剪区域进行最大池化并对分类器进行平均，但这些方法的性能均低于简单的平均方法。

在本文的剩余部分，我们将分析影响最终提交整体性能的多种因素。

我们在挑战赛中的最终提交在验证和测试数据上均获得了6.67%的前五错误率，在所有参与者中排名第一。与2012年的SuperVision方法相比，这实现了56.5%的相对降低；与上一年度的最佳方法（Clarifai）相比，也实现了约40%的相对降低——这两种方法均使用了外部数据来训练分类器。下表展示了一些表现最佳方法的统计数据。

我们还通过改变模型数量和预测图像时使用的裁剪数量，在下表中分析并报告了多种测试选择的性能。当使用单一模型时，我们选择在验证数据上 top-1 错误率最低的模型。为避免过度拟合测试数据统计特征，所有数据均在验证数据集上报告。

8 ILSVRC 2014 检测挑战赛设置与结果

ILSVRC检测任务是在图像中为200个可能类别中的物体生成边界框。若检测到的物体与真实类别一致，且其边界框重叠率至少达到50%（使用Jaccard指数），则视为正确检测。多余的检测会被视为误报并受到惩罚。与分类任务不同，每张图像可能包含

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

表4：检测性能

Team	mAP	Contextual model	Bounding box regression
Trimps-Soushen	31.6%	no	?
Berkeley Vision	34.5%	no	yes
UvA-Euvision	35.4%	?	?
CUHK DeepID-Net2	37.7%	no	?
GoogLeNet	38.02%	no	no
Deep Insight	40.2%	yes	yes

表5：检测任务的单模型性能

许多物体或没有物体，且它们的尺度可能从大到微小不等。结果使用平均精度均值（mAP）进行报告。

GoogLeNet采用的检测方法与[6]提出的R-CNN相似，但通过引入Inception模型作为区域分类器进行了增强。此外，区域建议步骤通过将选择性搜索[20]方法与multi-box[5]预测相结合得到改进，从而提高了目标边界框的召回率。为减少误报数量，超像素尺寸被增大了2×，这使得选择性搜索算法产生的建议区域减半。我们额外补充了200个来自multi-box[5]的区域建议，最终使用的建议区域总数约为[6]所用数量的60%，同时将覆盖率从92%提升至93%。在提高覆盖率的同时削减建议区域数量，使得单模型情况下的平均精度均值提升了1%。最后，在对每个区域进行分类时，我们采用了6个ConvNet组成的集成模型，将准确率从40%提升至43.9%。需要注意的是，由于时间限制，我们未像R-CNN那样使用边界框回归方法。

我们首先报告了最佳检测结果，并展示了自首届检测任务举办以来的进展。与2013年的结果相比，准确率几乎翻了一番。表现最佳的团队均采用了卷积网络。我们在表4中列出了官方评分及各团队的常用策略：使用外部数据、集成模型或上下文模型。外部数据通常指ILSVRC12分类数据，用于预训练模型，随后再基于检测数据进行微调。部分团队还提及使用了定位数据。由于定位任务中相当一部分边界框未包含在检测数据集中，研究者可借此数据预训练通用边界框回归器，其方式与使用分类数据进行预训练类似。GoogLeNet参赛方案未使用定位数据进行预训练。

在表5中，我们仅对比了使用单一模型的结果。表现最佳的模型来自Deep Insight，令人惊讶的是，其采用3个模型集成后仅提升了0.3分，而GoogLeNet通过集成方法获得了显著更强的结果。

9 结论

我们的结果似乎提供了坚实的证据，表明通过现成的密集构建模块来近似期望的最优稀疏结构，是改进计算机视觉神经网络的可行方法。该方法的主要优势在于，与较浅且较窄的网络相比，它在计算需求适度增加的情况下实现了显著的质量提升。还需注意的是，尽管我们的检测工作既未利用上下文也未执行边界框回归，但仍具有竞争力。

回归, 这一事实进一步证明了Inception架构的强大。尽管预期通过更昂贵、深度和宽度相似的网络可以获得类似质量的结果, 但我们的方法提供了坚实的证据, 表明转向更稀疏的架构总体上是可行且有益的想法。这为未来基于[2]以自动化方式创建更稀疏、更精细的结构提出了有前景的研究方向。

10 致谢

我们要感谢Sanjeev Arora和Aditya Bhaskara就文献[2]进行的富有成果的讨论。同时, 我们感谢DistBelief[4]团队的支持, 特别是Rajat Monga、Jon Shlens、Alex Krizhevsky、Jeff Dean、Ilya Sutskever和Andrea Frome。我们还要感谢Tom Duerig和Ning Ye在光度失真方面提供的帮助。此外, 如果没有Chuck Rosenberg和Hartwig Adam的支持, 我们的工作也无法完成。

参考文献

- [1] 了解你的梗: 我们需要更深入。 <http://knowyourmeme.com/memes/we-need-to-go-deeper>。访问日期: 2014-09-15。[2] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. 学习某些深度表示的可证明边界。 *CoRR*, abs/1310.6343, 2013。[3] Ümit V. Çatalyürek, Cevdet Aykanat, and Bora Uçar. 论二维稀疏矩阵划分: 模型、方法与方案。 *SIAM J. Sci. Comput.*, 32(2):656–683, 2010年2月。[4] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, and Andrew Y. Ng. 大规模分布式深度网络。载于 P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger 编辑的 *Advances in Neural Information Processing Systems 25*, 第1232–1240页。2012年。[5] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. 使用深度神经网络的可扩展目标检测。载于 *Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on*, 2014年。[6] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 用于精确目标检测和语义分割的丰富特征层次结构。载于 *Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on*, 2014年。[7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 通过防止特征检测器的共适应来改进神经网络。 *CoRR*, abs/1207.0580, 2012年。[8] Andrew G. Howard. 基于深度卷积神经网络的图像分类的一些改进。 *CoRR*, abs/1312.5402, 2013年。[9] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. 使用深度卷积神经网络进行ImageNet分类。载于 *Advances in Neural Information Processing Systems 25*, 第1106–1114页, 2012年。
- [10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, 与 L. D. Jackel. 反向传播应用于手写邮政编码识别。 *Neural Comput.*, 1(4):541–551, 1989年12月。
- [11] Yann LeCun, Léon Bottou, Yoshua Bengio 和 Patrick Haffner. 基于梯度的学习应用于文档识别。 *Proceedings of the IEEE*, 86(11):2278–2324, 1998。
- [12] Min Lin, Qiang Chen 和 Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013。
- [13] B. T. Polyak 与 A. B. Juditsky. 通过平均加速随机逼近。 *SIAM J. Control Optim.*, 第30卷第4期, 第838–855页, 1992年7月。
- [14] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: 使用卷积网络进行集成识别、定位与检测。 *CoRR*, abs/1312.6229, 2013。

[15] Thomas Serre, Lior Wolf, Stanley M. Bileschi, Maximilian Riesenhuber, 与 Tomaso Poggio。采用类皮层机制的鲁棒物体识别。 *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):411–426, 2007。 [16] Fengguang Song 与 Jack Dongarra。在具有1000个CPU核心的共享内存众核系统上扩展矩阵计算。收录于 *Proceedings of the 28th ACM International Conference on Supercomputing, ICS '14*, 第333–342页, 美国纽约, 2014。ACM。 [17] Ilya Sutskever, James Martens, George E. Dahl, 与 Geoffrey E. Hinton。论深度学习中初始化和动量的重要性。收录于 *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 第28卷 *JMLR Proceedings*, 第1139–1147页。JMLR.org, 2013。 [18] Christian Szegedy, Alexander Toshev, 与 Dumitru Erhan。用于物体检测的深度神经网络。收录于 Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, 与 Kilian Q. Weinberger 编辑的 *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 第2553–2561页, 2013。 [19] Alexander Toshev 与 Christian Szegedy。Deeppose: 通过深度神经网络进行人体姿态估计。 *CoRR*, abs/1312.4659, 2013。 [20] Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, 与 Arnold W. M. Smeulders。分割作为物体识别的选择性搜索。收录于 *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, 第1879–1886页, 美国华盛顿特区, 2011。IEEE 计算机学会。 [21] Matthew D. Zeiler 与 Rob Fergus。可视化和理解卷积网络。收录于 David J. Fleet, Tomás Pajdla, Bernt Schiele, 与 Tinne Tuytelaars 编辑的 *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, 第868–883页。Springer, 2014。