更深层次的卷积

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

Vincent Vanhoucke
Google Inc.

Andrew Rabinovich
Google Inc.

Abstract

本文提出一种代号为“Inception”的深度卷积神经网络架构，负责在2014年ImageNet大规模视觉识别挑战赛(ILSVRC’14)中设置分类和检测的新技术水平。这种架构的主要特点是提高了网络内部计算资源的利用率。这是通过精心设计的设计实现的，该设计允许增加网络的深度和宽度，同时保持计算预算不变。为了优化质量，架构决策基于Hebbian原理和多尺度处理的直觉。在我们提交的ILSVRC’14中使用的一个特殊的例子是GoogLeNet，一个22层的深度网络，其质量在分类和检测的背景下进行了评估。

1 简介

在过去的三年里，主要由于深度学习的进步，更具体地说，卷积网络[10]，图像识别和目标检测的质量一直在以惊人的速度进步。一个令人鼓舞的消息是，大部分进展不仅是更强大的硬件、更大的数据集和更大的模型的结果，而是主要是新想法、算法和改进的网络架构的结果。例如，ILSVRC 2014竞赛中的top entries除了用于检测目的的分类数据集外，没有使用新的数据源。我们向ILSVRC 2014提交的GoogLeNet实际上使用的参数比两年前Krizhevsky等人的获奖架构[9]少了12×，同时也明显更准确。目标检测的最大收益并不是单独来自深度网络或更大模型的利用，而是来自深度架构和经典计算机视觉的协同作用，如Girshick等人的R-CNN算法[6]。

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，我们的算法的效率——特别是它们的功率和内存使用——变得重要。值得注意的是，导致本文提出的深度架构设计的考虑因素包括这个因素，而不是完全专注于准确性数字。对于大多数实验，这些模型被设计为在推理时保持1.5亿乘加的计算预算，这样它们最终不会成为纯粹的学术好奇心，而是可以在现实世界中使用，即使是在大型数据集上，成本合理。

在本文中，我们将重点介绍一种用于计算机视觉的高效深度神经网络架构，代号为“Inception”，其名称源自Lin等人的“network In network”论文[12]以及著名的“we need to go deep”互联网meme [1]。在我们的案例中，“深度”一词有两种不同的含义:首先，在某种意义上，我们以“Inception模块”的形式引入了一个新的组织水平，同时在更直接的意义上，增加了网络深度。一般来说，人们可以将Inception模型视为[12]的逻辑顶点，同时从Arora等人[2]的理论工作中获得灵感和指导。该架构的优势在ILSVRC 2014分类与检测挑战赛上进行了实验验证，其性能明显优于当前技术水平。

(a) Siberian husky (b) Eskimo dog

Figure 1: ILSVRC 2014分类挑战赛的1000个类别中两个不同的类别。

2 相关工作

从LeNet-5 [10]开始，卷积神经网络(CNN)通常具有一个标准结构——堆叠的卷积层(可选地其次是对比度归一化和最大池化)，然后是一个或多个全连接层。这种基本设计的变体在图像分类文献中普遍存在，并在MNIST、CIFAR上产生了迄今为止最好的结果，最明显的是在ImageNet分类挑战[9, 21]上。对于更大的数据集，如Imagenet，最近的趋势是增加层的数量[12]和层的大小[21, 14]，同时使用dropout [7]来解决过拟合的问题。

尽管人们担心最大池化层会导致准确的空间信息丢失，但与[9]相同的卷积网络架构也已成功用于定位[9, 14]，目标检测[6, 14, 18, 5]和人体姿态估计[19]。受灵长类动物视觉皮层神经科学模型的启发，Serre等人[15]使用一系列不同大小的固定Gabor滤波器来处理多个尺度，类似于Inception模型。然而，与[15]的固定2层深度模型相反，Inception模型中的所有过滤器都是学习的。此外，Inception层被多次重复，在GoogLeNet模型中形成了22层的深度模型。

Network-in-Network是Lin等人提出的一种提高神经网络表征能力的方法[12]。当应用于卷积层时，该方法可以被视为额外的1×1卷积层，通常随后是修正线性激活[9]。这使得它可以很容易地集成到当前的CNN管道中。我们在我们的架构中大量使用这种方法。然而，在我们的设置中，1×1卷积有双重目的:最重要的是，它们主要用作降维模块，以消除计算瓶颈，否则将限制我们的网络的大小。这不仅可以增加网络的深度，还可以在不影响性能的情况下增加网络的宽度。

目前目标检测的主要方法是由Girshick等人提出的区域卷积神经网络(R-CNN) [6]。R-CNN将整个检测问题分解为两个子问题:首先以类别无关的方式利用颜色和超像素一致性等低级线索来建议潜在的目标，然后使用CNN分类器来识别这些位置的目标类别。这种两阶段的方法利用了低水平线索的边界框分割的准确性，以及最先进的cnn的强大分类能力。我们在提交的检测中采用了类似的管道，但在两个阶段都探索了增强功能，例如多框[5]预测以获得更高的对象边界框召回率，以及集成方法以更好地对边界框建议进行分类。

3 动机和高层次的考虑

提高深度神经网络性能最直接的方法是增加它们的大小。这包括增加网络的深度(层次的数量)和宽度(每个层次的单位数量)。这是一种简单且安全的训练高质量模型的方法，特别是在有大量标记训练数据的情况下。然而，这个简单的解决方案有两个主要缺点。

更大的规模通常意味着更多的参数，这使得扩大的网络更容易过拟合，特别是在训练集中标记样本的数量有限的情况下。这可能成为一个主要的瓶颈，因为创建高质量的训练集可能是棘手和昂贵的，特别是如果需要专家人工评分人员来区分细粒度的视觉类别，如ImageNet中的那些(即使是在1000类的ILSVRC子集中)，如图1所示。

统一增加网络规模的另一个缺点是计算资源的使用急剧增加。例如，在深度视觉网络中，如果两个卷积层被链接起来，其滤波器数量的任何均匀增加都会导致计算的二次增长。如果增加的容量没有得到有效利用(例如，大多数权重最终都接近于0)，则会浪费大量计算。由于在实践中计算预算总是有限的，所以即使主要目标是提高结果的质量，有效地分配计算资源也比随意增加大小更可取。

解决这两个问题的基本方法将是最终从完全连接转移到稀疏连接架构，甚至在卷积内部。除了模拟生物系统，这还将具有更坚实的理论基础的优势，因为Arora等人的开创性工作[2]。他们的主要结果是，如果数据集的概率分布由一个大型的、非常稀疏的深度神经网络表示，则可以通过分析最后一层激活值的相关性统计，并对输出高度相关的神经元进行聚类，逐层构建最优的网络拓扑。尽管严格的数学证明需要非常强的条件，但这一陈述与著名的赫布尼亚原理(神经元一起点火，连接在一起)产生共鸣的事实表明，即使在不那么严格的条件下，这一基本思想也适用于实践。

缺点是，当涉及到非均匀稀疏数据结构的数值计算时，当今的计算基础设施非常低效。即使通过100×减少了算术操作的数量，但查找和缓存缺失的开销仍然非常大，因此切换到稀疏矩阵将无法获得回报。通过使用稳定改进的、高度调整的数字库，这一差距甚至进一步扩大，这些数字库允许极快的密集矩阵乘法，利用底层CPU或GPU硬件的细微细节[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。目前大多数面向视觉的机器学习系统仅通过使用卷积来利用空间域的稀疏性。然而，卷积被实现为与前一层中的补丁的密集连接的集合。自[11]以来，ConvNets传统上在特征维度上使用随机和稀疏连接表，以打破对称性并提高学习，趋势改为使用[9]的全连接，以便更好地优化并行计算。结构的均匀性、大量的滤波器和更大的批量大小允许利用高效的密集计算。

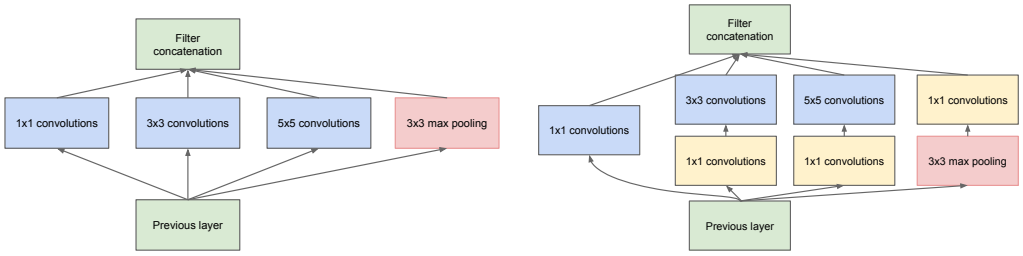
这提出了一个问题，即是否有希望实现下一个中间步骤:正如理论所建议的，一种利用额外稀疏性的架构，即使在滤波器级别，但通过利用稠密矩阵上的计算来利用我们当前的硬件。关于稀疏矩阵计算的大量文献(例如[3])表明，将稀疏矩阵聚类为相对稠密的子矩阵往往可以提供最先进的稀疏矩阵乘法的实际性能。在不久的将来，类似的方法将被用于非均匀深度学习架构的自动化构建，这似乎不是遥不可及的。

Inception架构开始作为第一作者的一个案例研究，用于评估复杂网络拓扑结构构造算法的假设输出，该算法试图近似由[2]为视觉网络隐含的稀疏结构，并通过密集的、现成的组件覆盖假设的结果。尽管这是一个高度推测性的任务，但仅仅在拓扑的准确选择上进行了两次迭代之后，我们已经可以看到相对于基于[12]的参考体系结构的适度改进。在进一步调整学习率、超参数和改进的训练方法之后，我们确定由此产生的Inception架构在定位和目标检测的背景下特别有用，作为[6]和[5]的基础网络。有趣的是，虽然大多数最初的架构选择都经过了彻底的质疑和测试，但它们至少是局部最优的。

然而，人们必须谨慎:尽管所提出的架构已经成为计算机视觉的成功，但其质量是否可以归因于导致其建设的指导原则仍然存在疑问。确保这一点需要更彻底的分析和验证:例如，如果基于下面描述的原则的自动化工具可以为视觉网络找到类似但更好的拓扑结构。最有说服力的证明是，自动化系统是否可以使用相同的算法在其他领域创建导致类似收益的网络拓扑，但外观非常不同的全局架构。至少，Inception架构的最初成功产生了坚定的动机，为这个方向令人兴奋的未来工作提供了动力。

4 建筑细节

Inception架构的主要思想是基于找出卷积视觉网络中最优的局部稀疏结构如何被现成的密集组件逼近和覆盖。注意，假设平移不变性意味着我们的网络将由卷积构建模块构建。我们所需要的就是找到最优的局部结构并在空间上重复它。Arora et al. [2]建议一种逐层构建，其中应该分析最后一层的相关性统计数据，并将它们聚类为具有高度相关性的单元组。这些簇形成下一层的单元，并与前一层的单元相连。我们假设前一层的每个单元对应于输入图像的某个区域，这些单元被分组到滤波器组中。在较低的层(靠近输入的层)中，相关单元将集中在局部区域。这意味着，我们最终将有许多集群集中在单个区域，它们可以被下一层中的1×1卷积层覆盖，如[12]所建议的。然而，人们也可以预期，将会有更少的空间分布更分散的集群，这些集群可以被更大的patch上的卷积覆盖，并且在越来越大的区域上的patch数量将会越来越少。为了避免patch-alignment问题，当前的Inception架构被限制为过滤大小1×1, 3×3和5×5，然而这个决定更多是基于方便而不是必要。这也意味着所建议的架构是所有这些层的组合，其输出滤波器组连接成单个输出向量，形成下一阶段的输



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception模块

入。此外，由于池化操作对当前最先进的卷积网络的成功至关重要，这表明在每个这样的阶段添加一个替代的并行池化路径也应该具有额外的有益效果(参见图2(a))。

由于这些"Inception模块"彼此堆叠在一起，它们的输出相关性统计必然会变化:随着更高抽象的特征被更高的层捕获，它们的空间集中度预计会降低，这表明3×3和5×5卷积的比例应该随着我们移动到更高的层而增加。

以上模块(至少在naïve形式中)的一个大问题是，在具有大量滤波器的卷积层之上，即使是数量适中的5×5卷积也可能非常昂贵。一旦加入池化单元，这个问题变得更加明显:它们的输出过滤器数量等于前一阶段的过滤器数量。池化层的输出与卷积层的输出的合并将不可避免地导致每个阶段的输出数量增加。即使这种架构可能覆盖最优的稀疏结构，它的效率也非常低，在几个阶段内就会导致计算爆炸。

这导致了所提出架构的第二个想法:在计算需求会增加太多的地方明智地应用降维和预测。这是基于嵌入的成功:即使是低维嵌入也可能包含有关相对较大的图像块的大量信息。然而，嵌入以密集的、压缩的形式表示信息，压缩信息更难建模。我们希望在大多数地方保持我们的表示稀疏(根据[2]的条件要求)，并仅在必须整体聚合信号时压缩信号。也就是说，1×1卷积用于在昂贵的3×3和5×5卷积之前计算归约。除了被用作还原，它们还包括修正线性激活的使用，这使它们具有双重用途。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堆叠在一起组成的网络，偶尔具有步幅2的最大池化层，以将网格的分辨率减半。出于技术原因(训练期间的记忆效率)，只在较高的层开始使用Inception模块似乎是有益的，同时以传统的卷积方式保持较低的层。这并不是严格必要的，只是反映了我们当前实现中一些基础设施的低效。

这种架构的一个主要好处是，它允许在每个阶段显著增加单元的数量，而不会导致计算复杂度的失控膨胀。降维的普遍使用允许将上一阶段的大量输入滤波器屏蔽到下一层，首先降低它们的维度，然后用较大的patch大小对它们进行卷积。这种设计的另一个实际有用的方面是，它符合视觉信息应该在各种尺度上处理，然后聚合，以便下一阶段可以同时从不同尺度抽象特征的直觉。

改进了对计算资源的使用，可以在不陷入计算困难的情况下增加每个阶段的宽度和阶段数量。利用inception架构的另一种方法是创建稍差的，但计算成本更低的版本。我们发现，所有包含的旋钮和杠杆都允许计算资源的受控平衡，这可以导致网络比具有非inception架构的类似执行网络快$2-3\times$，然而，这需要在这一点上仔细的手动设计。

5 GoogLeNet

在ILSVRC14竞赛中，我们选择GoogLeNet作为我们的团队名称。这个名字是向Yann LeCun的先锋LeNet 5网络[10]致敬。我们还使用GoogLeNet来引用我们在竞赛中使用的Inception架构的特定化身。我们还使用了一个更深更宽的Inception网络，其质量稍差，但将其添加到集成中似乎略微改善了结果。我们省略了该网络的细节，因为我们的实验表明，准确的架构参数的影响相对较小。为了演示，表1中描述了一个最成功的特定实例(名为GoogLeNet)。在我们的集成中，7个模型中的6个使用了完全相同的拓扑结构(用不同的采样方法进行训练)。

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								

Table 1: Inception架构的GoogLeNet化身

所有卷积，包括Inception模块内的卷积，都使用修正线性激活。我们的网络中感受野的大小是224×224使用均值减法的RGB颜色通道。“ #3×3 reduce ‘和“ #5×5 reduce ‘代表在3×3和5×5卷积之前使用的缩减层中1×1过滤器的数量。在“pool proj”列中，可以看到投影层中内置max-pooling后的1×1过滤器数量。所有这些还原/投影层也使用修正线性激活。

该网络在设计时考虑了计算效率和实用性，因此推理可以在单个设备上运行，包括计算资源有限的设备，特别是低内存占用的设备。当只计算带参数的层时，网络有22层(如果我们也计算池化，则为27层)。用于构建网络的“层”(独立的构建块)总数约为100。然而，这个数字取决于所使用的机器学习基础设施系统。在分类器之前使用平均池化是基于[12]的，尽管我们的实现有所不同，因为我们使用了额外的线性层。这使我们能够轻松地为其他标签集调整和微调我们的网络，但这主要是方便的，我们不期望它有重大影响。研究发现，从全连接层移动到平均池化将top-1的准确率提高了约0.6%，然而，即使在删除全连接层后，使用dropout仍然是必不可少的。

考虑到网络的深度相对较大，以有效的方式将梯度传播回所有层的能力是一个问题。一个有趣的观点是，相对较浅的网络在这项任务上的强大性能表明，网络中间层产生的特征应该非常具有区分性。通过添加连接到这些中间层的辅助分类器，我们期望在分类器的较低阶段鼓励判别，增加反向传播的梯度信号，并提供额外的正则化。这些分类器采用更小的卷积网络的形式，放在Inception (4a)和(4d)模块的输出之上。在训练过程中，它们的损失以折扣权重添加到网络的总损失中(辅助分类器的损失加权为0.3)。在推理时，这些辅助网络被丢弃。

包含辅助分类器的额外网络的确切结构如下所示:

- 具有5×5滤波器大小和步幅3的平均池化层，产生(4a)的4×4×512输出，和(4d)阶段的4×4×528输出。
- 1×1具有128个滤波器的卷积，用于降维和校正线性激活。
- 具有1024个单元和整流线性激活的全连接层。
- 具有70%丢弃输出比率的dropout层。
- 一个以softmax损失作为分类器的线性层(与主分类器预测相同的1000个类别，但在推理时被删除)。

结果网络的示意图如图3所示。

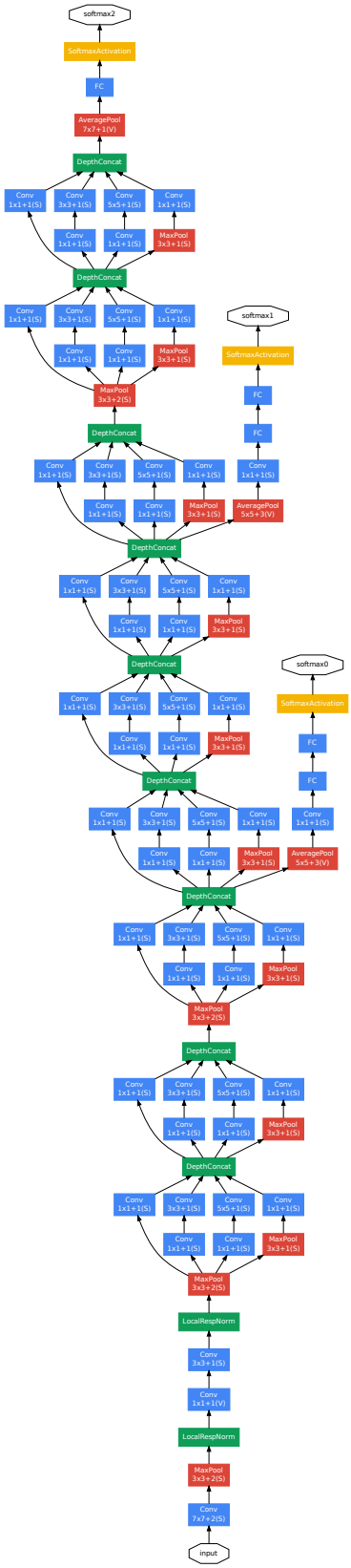


Figure 3: 拥有所有花哨功能的GoogLeNet网络

6 培训方法

我们的网络使用DistBelief [4]分布式机器学习系统进行训练，使用适度的模型和数据并行性。虽然我们只使用基于CPU的实现，但粗略估计表明，可以在一周内使用少量高端gpu训练GoogLeNet网络，使其收敛，主要限制是内存使用。我们的训练使用0.9动量的异步随机梯度下降[17]，固定的学习率调度(每8个周期降低4%的学习率)。Polyak平均[13]用于创建推理时使用的最终模型。

在比赛前的几个月里，我们的图像采样方法发生了很大的变化，并且已经与其他选项一起训练收敛模型，有时与改变的超参数一起训练，如dropout和学习率，因此很难给训练这些网络的最有效的单一方法提供明确的指导。更复杂的是，一些模型主要在相对较小的作物上训练，其他模型则在较大的作物上训练，其灵感来自[8]。尽管如此，在比赛后被验证非常有效的方法包括对图像的各种大小的块进行采样，这些块的大小均匀分布在图像区域的8%和100%之间，其长宽比在3/4和4/3之间随机选择。此外，我们发现Andrew Howard [8]的光度失真在一定程度上有助于对抗过拟合。此外，我们开始使用随机插值方法(bilinear, area, nearest neighbor, cubic，等概率)进行较晚的缩放，并结合其他超参数的变化，因此我们无法明确地判断它们的使用是否会积极影响最终结果。

7 ILSVRC 2014分类挑战赛设置和结果

ILSVRC 2014分类挑战涉及将图像分类为Imagenet层次结构中的1000个叶节点类别之一的任务。大约有120万张图像用于训练，5万张用于验证，10万张用于测试。每个图像都与一个真实类别相关联，性能是根据得分最高的分类器预测来衡量的。通常报告两个数字:top-1准确率，将地面真实值与第一个预测类别进行比较;top-5错误率，将地面真实值与前5个预测类别进行比较:如果地面真实值在top-5中，则认为图像被正确分类，无论其在其中的排名。该挑战使用前5名的错误率来进行排名。

我们参加了挑战，没有使用外部数据进行训练。除了本文提到的训练技术，我们在测试过程中采用了一套技术来获得更高的性能，我们将在下面详细阐述。

1. 我们独立训练了同一个GoogLeNet模型的7个版本(包括一个更宽的版本)，并对它们进行了集合预测。这些模型使用相同的初始化(甚至使用相同的初始权重，主要是因为疏忽)和学习率策略进行训练，它们只是在采样方法和看到输入图像的随机顺序上有所不同。

2. 在测试中，我们采用了比Krizhevsky等人[9]更激进的裁剪方法。具体来说，我们将图像调整为4种比例，其中较短的尺寸(高度或宽度)分别为256、288、320和352，取这些调整大小的图像的左、中和右正方形(在肖像图像的情况下，我们取顶部、中心和底部正方形)。对于每个正方形，我们然后采取四个角和中心224×224裁剪以及正方形调整大小为224×224，以及它们的镜像版本。这导致每张图像有$4\times3\times6\times2 = 144$个农作物。Andrew Howard [8]在上一年的条目中使用了类似的方法，我们通过经验验证，该方法的性能比所提出的方案稍差。我们注意到，这种激进的种植在实际应用中可能是不必要的，因为在出现合理数量的作物后，更多作物的效益变得微不足道(我们将在稍后展示)。

3. 对多个作物和所有单个分类器的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，如农作物的最大池化和分类器的平均，但它们的性能不如简单平均。

在本文的其余部分，我们分析了影响最终提交的整体性能的多个因素。

我们在挑战赛中的最终提交在验证和测试数据上都获得了6.67%的top-5错误，在其他参与者中排名第一。与2012年的监督方法相比，相对减少了56.5%，与上一年的最佳方法(Clarifai)相比，相对减少了约40%，这两种方法都使用外部数据来训练分类器。下表显示了一些表现最好的方法的统计数据。

通过改变预测图像时使用的模型数量和作物数量，我们还分析和报告了多个测试选择的性能。当我们使用一个模型时，我们选择验证数据上top-1错误率最低的模型。所有的数字都报告在验证数据集上，以避免对测试数据统计过拟合。

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

Table 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%

Table 3: GoogLeNet分类性能分解

8 ILSVRC 2014检测挑战赛设置和结果

ILSVRC的检测任务是在200个可能的类别中为图像中的物体生成边界框。如果检测到的物体与groundtruth的类别匹配，并且它们的边界框重叠至少50%(使用杰卡德指数)，则视为正确。无关的检测将被视为假阳性，并受到惩罚。与分类任务相反，每幅图像可能包含许多物体，也可能不包含，而且它们的尺度可能从大到小。结果使用平均精度均值(mAP)来报告。

GoogLeNet采取的检测方法类似于[6]的R-CNN，但使用Inception模型作为区域分类器。此外，通过将选择性搜索[20]方法与多框[5]预测相结合来改进区域建议步骤，以获得更高的目标边界框召回率。为了减少假阳性的数量，超像素大小增加了2×。这将来自选择性搜索算法的建议减半。我们添加了来自多框[5]的200个区域提案，总共产生了[6]使用的约60%的提案，同时将覆盖率从92%提高到93%。在增加覆盖率的同时减少建议数量的总体效果是将单个模型的平均精度均值提高1%。最后，在对每个区域进行分类时，使用6个卷积网络的集成，将结果的准确率从40%提高到43.9%。请注意，与R-CNN相反，由于时间不足，我们没有使用边界框回归。

我们首先报告顶部的检测结果，并显示自第一版检测任务以来的进展。与2013年的结果相比，准确率几乎翻了一番。表现最好的团队都使用卷积网络。我们在表4中报告了官方分数和每个团队的常见策略:使用外部数据，集成模型或上下文模型。外部数据通常是用于预训

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

Table 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

Table 5: 用于检测的单模型性能

练模型的ILSVRC12分类数据，稍后在检测数据上进行细化。一些团队还提到了本地化数据的使用。由于定位任务边界框的很大一部分未包含在检测数据集中，因此可以用与分类用于预训练相同的方法，用该数据对通用边界框回归器进行预训练。GoogLeNet条目没有使用本地化数据进行预训练。

在表5中，我们仅使用单个模型比较结果。表现最好的模型是通过深入的洞察力获得的，令人惊讶的是，在3个模型的集成中只提高了0.3个点，而GoogLeNet通过集成获得了更强的结果。

9 结论

我们的结果似乎产生了一个确凿的证据，用现成的密集构建块来近似预期的最优稀疏结构是改进计算机视觉神经网络的一种可行方法。与较浅和较宽的网络相比，该方法的主要优势是在计算需求适度增加的情况下，获得了显著的质量增益。还要注意，我们的检测工作是有竞争力的，尽管既没有利用上下文也没有执行边界框回归，这一事实进一步证明了Inception架构的强度。虽然期望通过深度和宽度相似、代价昂贵得多的网络可以实现相似的结果质量，但该方法产生了可靠的证据，表明迁移到更稀疏的架构通常是可行和有用的想法。这表明，未来有希望在[2]的基础上以自动化方式创建更稀疏和更精细的结构。

10 致谢

我们要感谢Sanjeev Arora和Aditya Bhaskara在[2]上进行的富有成果的讨论。我们还要感谢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在光度失真方面的帮助。此外，如果没有Chuck Rosenberg和Hartwig Adam的支持，我们的工作也不可能完成。

References

[1] Know your meme: We need to go deeper. http://knowyourmeme.com/memes/we-need-to-go-deeper. Accessed: 2014-09-15.

[2] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. *CoRR*, abs/1310.6343, 2013.

[3] Ümit V. Çatalyürek, Cevdet Aykanat, and Bora Uçar. On two-dimensional sparse matrix partitioning: Models, methods, and a recipe. *SIAM J. Sci. Comput.*, 32(2):656–683, February 2010.

[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. Large scale distributed deep networks. In P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1232–1240. 2012.

[5] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on*, 2014.

[6] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on*, 2014.

[7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[8] Andrew G. Howard. Some improvements on deep convolutional neural network based image classification. *CoRR*, abs/1312.5402, 2013.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[10] 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 Comput.*, 1(4):541–551, December 1989.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[13] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992.

[14] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[15] Thomas Serre, Lior Wolf, Stanley M. Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):411–426, 2007.

[16] Fengguang Song and Jack Dongarra. Scaling up matrix computations on shared-memory manycore systems with 1000 cpu cores. In *Proceedings of the 28th ACM International Conference on Supercomputing*, ICS '14, pages 333–342, New York, NY, USA, 2014. ACM.

[17] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Proceedings*, pages 1139–1147. JMLR.org, 2013.

[18] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *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.*, pages 2553–2561, 2013.

[19] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

[20] Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, and Arnold W. M. Smeulders. Segmentation as selective search for object recognition. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1879–1886, Washington, DC, USA, 2011. IEEE Computer Society.

[21] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014.