ImageNet分类与深度卷积神经网络

亚历克斯Krizhevsky
多伦多大学kriz@cs.utoronto.caIlya

Sutskever
多伦多大学ilya@cs.utoronto.caGeoff

rey E. Hinton
多伦多大学hinton@cs.utoronto.ca

摘要

我们训练了一个大型的深度卷积神经网络，将ImageNet LSVRC-2010竞赛中的120万张高分辨率图像分类为1000个不同的类别。在测试数据上，我们实现了37.5%和17.0%的top-1和top-5错误率，这大大优于之前的最先进技术。这个拥有6000万个参数和65万个神经元的神经网络由5个卷积层组成，其中一些层之后是最大池化层，以及3个具有最终1000路softmax的全连接层。为了使训练更快，我们使用了非饱和神经元和非常高效的卷积操作的GPU实现。为了减少全连接层中的过拟合，我们采用了最近开发的称为“dropout”的正则化方法，该方法被证明是非常有效的。我们还在ILSVRC-2012竞赛中输入了该模型的一个变体，并实现了15.3%的获胜top-5测试错误率，相比之下，第二好的输入实现了26.2%。

1 介绍

目前的物体识别方法必不可少地使用了机器学习方法。为了提高它们的性能，我们可以收集更大的数据集，学习更强大的模型，并使用更好的技术来防止过拟合。直到最近，带标签图像的数据集都相对较小——在数万张图像的量级上(例如，NORB [16]，Caltech-101/256[8,9]，和CIFAR-10/100[12])。使用这种大小的数据集可以相当好地解决简单的识别任务，特别是如果它们用标签保持变换进行增强。例如，在MNIST数字识别任务上的当前最佳错误率(<0.3%)接近人类性能[4]。但现实环境中的物体表现出相当大的可变性，因此要学会识别它们，就必须使用更大的训练集。而且确实，小型图像数据集的缺点已经被广泛认识(例如Pinto等人[21])，但直到最近才有可能收集数百万张图像的标记数据集。新的更大的数据集包括LabelMe[23]，它由数十万张完全分割的图像组成，以及ImageNet[6]，它由超过22,000个类别的超过1500万张标记的高分辨率图像组成。

要从数百万张图像中了解数千个对象，我们需要一个具有大学习能力的模型。然而，物体识别任务的巨大复杂性意味着即使像ImageNet这样大的数据集也不能指定这个问题，所以我们的模型也应该有大量的先验知识来弥补我们没有的所有数据。卷积神经网络(cnn)就是这样一类模型[16,11,13,18,15,22,26]。它们的容量可以通过改变其深度和广度来控制，而且它们还对图像的性质(即统计的平稳性和像素依赖的局部性)做出强大且基本正确的假设。因此，与具有类似大小层的标准前馈神经网络相比，cnn的连接和参数要少得多，因此更容易训练，而它们的理论最佳性能可能只会稍微差一点。

尽管cnn具有吸引人的品质，尽管其局部架构相对高效，但大规模应用于高分辨率图像的成本仍然高得令人望而却步。幸运的是，当前的gpu，加上高度优化的2D卷积实现，足够强大，可以促进有趣的大型cnn的训练，并且最近的数据集(如ImageNet)包含足够的标记示例来训练这样的模型，而不会出现严重的过拟合。
本文的具体贡献如下:我们在ILSVRC-2010和ILSVRC-2012比赛中使用的ImageNet子集上训练了迄今为止最大的卷积神经网络之一[2]，并取得了迄今为止在这些数据集上报道的最佳结果。我们编写了一个高度优化的2D卷积和训练卷积神经网络中固有的所有其他操作的GPU实现，并将其公开[1]。我们的网络包含了许多新的和不寻常的特征，这些特征提高了它的性能，减少了它的训练时间，详情见第3节。我们网络的规模使得过拟合成为一个显著的问题，即使有120万个带标签的训练示例，因此我们使用了几种有效的技术来防止过拟合，这些技术在第4节中描述。我们最终的网络包含5个卷积层和3个全连接层，这个深度似乎很重要:我们发现删除任何卷积层(每个卷积层包含的模型参数不超过1%)都会导致性能下降。

最后，网络的大小主要受到当前gpu上可用的内存量和我们愿意容忍的训练时间的限制。我们的网络需要五到六天的时间来训练两个GTX 580 3GB gpu。我们所有的实验都表明，只要等待更快的gpu和更大的数据集可用，我们的结果就可以得到改善。

2 数据集

ImageNet是一个超过1500万张标记高分辨率图像的数据集，属于大约22,000个类别。这些图像是从网络上收集的，由人工标记人员使用亚马逊的Mechanical Turk众包工具进行标记。从2010年开始，作为Pascal视觉对象挑战赛的一部分，一年一度的ImageNet大规模视觉识别挑战赛(ILSVRC)已经举行。ILSVRC使用ImageNet的一个子集，在1000个类别中每个类别中大约有1000张图像。总的来说，大约有120万张训练图像，5万张验证图像和15万张测试图像。

ILSVRC-2010是唯一一个提供测试集标签的ILSVRC版本，所以我们在这个版本上进行了大多数实验。由于我们的模型也参加了ILSVRC-2012竞赛，因此在第6节中，我们也报告了该版本数据集的结果，其中测试集标签不可用。在ImageNet上，习惯上报告两个错误率:top-1和top-5，其中top-5错误率是正确标签不在模型认为最可能的五个标签中的测试图像的比例。

ImageNet由可变分辨率的图像组成，而我们的系统需要恒定的输入维数。因此，我们将图像降采样到固定分辨率256 ×256。给定一个矩形图像，我们首先重新缩放图像，使较短的边的长度为256，然后从结果图像中裁剪出中心256×256补丁。除了从每个像素减去训练集上的平均活动，我们没有以任何其他方式对图像进行预处理。因此，我们在像素的(居中)原始RGB值上训练我们的网络。

3 架构

我们网络的架构总结在图2中。它包含8个学习层——5个卷积层和3个全连接层。下面，我们描述了我们的网络架构的一些新颖或不寻常的特征。3.1-3.4节根据我们对其重要性的估计进行排序，最重要的排在前面。

[1] http://code.google.com/p/cuda-convnet/

3.1 ReLU非线性

将神经元的输出f建模为其输入x的函数的标准方法是:$f(x) = \tanh(x)$或$f(x) = (1 + e^{-x})^{-1}$。就梯度下降的训练时间而言，这些饱和非线性比非饱和非线性$f(x) = \max(0, x)$慢得多。根据Nair和Hinton[20]，我们将具有这种非线性的神经元称为Rectified Linear Units (ReLUs)。具有ReLUs的深度卷积神经网络的训练速度比具有tanh单元的等效网络快几倍。图1展示了这一点，图中显示了对于一个特定的四层卷积网络，在CIFAR-10数据集上达到25%的训练误差所需的迭代次数。这张图显示，如果我们使用传统的饱和神经元模型，我们将无法为这项工作进行如此大的神经网络实验。

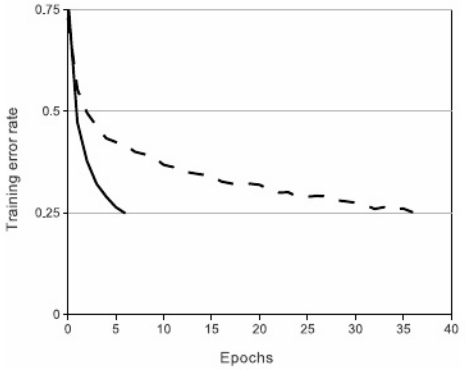

图1:四层卷积神经网络

具有ReLUs的网络(实线)在CIFAR-10上达到25%的训练错误率，比具有tanh神经元的等效网络(虚线)快6倍。每个网络的学习率都是独立选择的，以使训练尽可能快。没有采用任何类型的正则化。这里所展示的效果的大小随网络结构的不同而不同，但具有ReLUs的网络始终比具有饱和神经元的网络学习速度快几倍。

我们不是第一个考虑cnn中传统神经元模型的替代方案的人。例如，Jarrett等人[11]声称非线性$f(x) = |\tanh(x)|$特别适用于他们的对比归一化类型，然后在Caltech-101数据集上进行局部平均池化。然而，在这个数据集上，主要关注的是防止过拟合，所以他们观察到的效果不同于我们在使用ReLUs时报告的训练集的加速拟合能力。更快的学习对在大型数据集上训练的大型模型的性能有很大的影响。

3.2 多GPU训练

单个GTX 580 GPU只有3GB的内存，这限制了可以在其上训练的网络的最大大小。事实证明，120万个训练样本足以训练一个GPU无法容纳的网络。因此，我们将网络分散到两个GPU上。当前的GPU特别适合跨gpu并行化，因为它们能够直接从彼此的内存中读取和写入，而无需通过主机内存。我们采用的并行化方案实际上是将一半的内核(或神经元)放在每个GPU上，还有一个额外的技巧:GPU只在某些层进行通信。这意味着，例如，第3层的核从第2层的所有核映射中获取输入。然而，第4层的内核只从位于同一GPU上的第3层的内核映射中获取输入。选择连接的模式对于交叉验证来说是一个问题，但这允许我们精确地调整通信量，直到它是计算量的一个可接受的部分。

所得到的架构有点类似于cirelsan等人使用的“柱状”CNN[5]，除了我们的列不是独立的(见图2)。与在一个GPU上训练的每个卷积层的核数减半的网络相比，该方案将我们的前1和前5的错误率分别降低了1.7%和1.2%。双gpu网络的训练时间比单gpu网络稍微少一点[2]。

[2] The one-GPU net actually has the same number of kernels as the two-GPU net in the final convolutional layer. This is because most of the net’s parameters are in the first fully-connected layer, which takes the last convolutional layer as input. So to make the two nets have approximately the same number of parameters, we did not halve the size of the final convolutional layer (nor the fully-conneced layers which follow). Therefore this comparison is biased in favor of the one-GPU net, since it is bigger than “half the size” of the two-GPU net.

3.3 局部响应规范化

ReLUs具有理想的特性，即不需要输入归一化来防止它们饱和。如果至少有一些训练样例对ReLU产生正输入，则学习将在该神经元中进行。然而，我们仍然发现下面的局部归一化方案有助于泛化。用aix,y表示在(x, y)位置应用核i计算的神经元的活动，然后应用ReLU非线性，响应归一化的活动bix,y由表达式给出

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

其中，和在相同空间位置的n个“相邻”核映射上运行，n是层中核的总数。核映射的排序当然是任意的，在训练开始之前就确定了。这种响应归一化实现了一种受真实神经元中发现的类型启发的侧抑制形式，在使用不同核计算的神经元输出之间创建了大型活动的竞争。常数k, n， α和β是超参数，它们的值是使用验证集确定的;我们使用k = 2, n = 5， $\alpha=10^{-4}$，β= 0.75。我们在某些层中应用ReLU非线性后应用这种归一化(参见第3.5节)。

该方案与Jarrett等人[11]的局部对比度归一化方案有一些相似之处，但我们的方案更准确地称为“亮度归一化”，因为我们没有减去平均活动。响应归一化将我们的top-1和top-5错误率分别降低1.4%和1.2%。我们还在CIFAR-10数据集上验证了该方案的有效性:四层CNN在没有归一化的情况下实现了13%的测试错误率，在归一化的情况下实现了11%的测试错误率[3]。

3.4 重叠池

cnn中的池化层对同一核映射中相邻神经元组的输出进行汇总。传统上，相邻池化单元汇总的邻域不重叠(例如[17,11,4])。更准确地说，池化层可以被认为是由间隔为s像素的池化单元网格组成，每个池化单元汇总一个以池化单元位置为中心的大小为z ×z的邻域。如果我们设s = z，我们就得到了cnn中常用的传统局部池化。如果我们设置s < z，我们获得重叠池化。这就是我们在整个网络中使用的，s = 2, z = 3。与产生等效维度输出的非重叠方案s = 2, z = 2相比，这个方案分别降低了0.4%和0.3%的top-1和top-5错误率。我们在训练过程中通常观察到，具有重叠池化的模型发现过拟合稍微困难一些。

3.5 总体架构

现在我们准备描述CNN的整体架构。如图2所示，网络包含8层权重;前五层是卷积层，其余三层是全连接层。最后一个全连接层的输出被馈送到1000路softmax，该softmax在1000个类标签上产生分布。我们的网络最大化多项逻辑回归目标，这相当于最大化预测分布下正确标签的对数概率的跨训练案例的平均值。

第二层、第四层和第五层卷积层的内核只连接到位于同一GPU上的前一层的内核映射(见图2)。第三层卷积层的内核连接到第二层的所有内核映射。全连接层中的神经元与前一层中的所有神经元相连。响应归一化层紧随第一层和第二层卷积层。3.4节中描述的那种最大池化层，既遵循响应规范化层，也遵循第五卷积层。ReLU非线性应用于每个卷积层和全连接层的输出。

第一个卷积层以4像素的步幅(这是邻近感受野中心之间的距离)对224×224×3输入图像进行96个大小为11×11×3的核滤波

[3] We cannot describe this network in detail due to space constraints, but it is specified precisely by the code and parameter files provided here: http://code.google.com/p/cuda-convnet/.

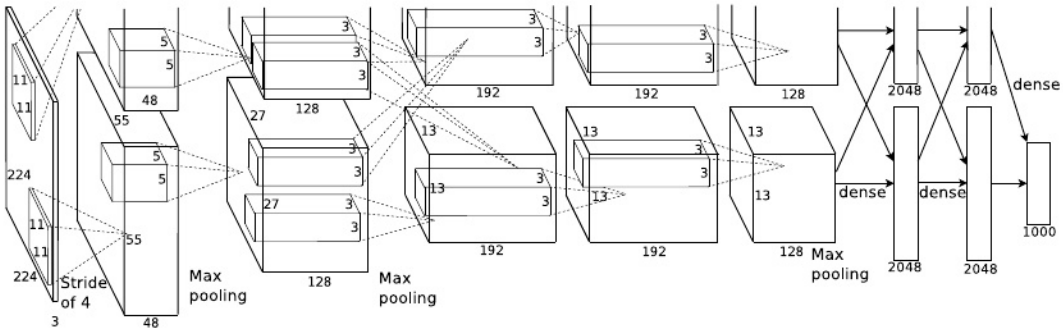

图2:我们的CNN架构示意图，明确显示了两个gpu之间的职责划分。一个GPU在图的顶部运行层部件，而另一个在图的底部运行层部件。gpu只在某些层通信。网络的输入为150,528维，网络剩余层的神经元数量为253,440-186,624-64,896-64,896-43,264 - 4096-4096-1000。

核图中的神经元)。第二个卷积层将第一个卷积层的输出(响应归一化和池化)作为输入，并用256个大小为5的核对其进行过滤×5 ×48。第三层、第四层和第五层卷积层相互连接，没有任何中间的池化或规范化层。第三层卷积层有384个大小为3 ×3 × 256的核连接到第二层卷积层的输出(归一化，池化)。第四个卷积层有384个大小为3 ×3 ×192的卷积核，第五个卷积层有256个大小为3 ×3 ×192的卷积核。全连接层每个有4096个神经元。

4 减少过度拟合

我们的神经网络架构有6000万个参数。虽然1000类ILSVRC使每个训练示例对从图像到标签的映射施加了10位约束，但事实证明，在没有相当大的过拟合的情况下，学习这么多参数是不够的。下面，我们描述了我们对抗过拟合的两种主要方法。

4.1 数据增加

减少图像数据上过拟合最简单和最常见的方法是使用标签保持变换人为地扩大数据集(例如，[25,4,5])。我们采用了两种不同的数据增强形式，这两种形式都允许从原始图像中以很少的计算产生转换图像，因此转换图像不需要存储在磁盘上。在我们的实现中，转换后的图像是在CPU上用Python代码生成的，而GPU在前一批图像上进行训练。因此，这些数据增强方案实际上在计算上是免费的。

数据增强的第一种形式是生成图像平移和水平反射。我们通过从256×256图像中随机提取224个×224补丁(以及它们的水平反射)，并在这些提取的补丁上训练我们的网络[4]。这使我们的训练集的大小增加了2048倍，尽管由此产生的训练示例当然是高度相互依赖的。如果没有这个方案，我们的网络会遭受严重的过拟合，这将迫使我们使用更小的网络。在测试时，网络通过提取5个224 ×224补丁(四个角补丁和中心补丁)以及它们的水平反射(因此总共有10个补丁)来进行预测，并将网络的softmax层在10个补丁上所做的预测进行平均。

第二种形式的数据增强包括改变训练图像中RGB通道的强度。具体来说，我们对整个ImageNet训练集的RGB像素值集执行PCA。对于每个训练图像，我们添加找到的主成分的倍数，

[4] This is the reason why the input images in Figure 2 are 224 ×224 ×3-dimensional.

其大小与相应的特征值成正比，乘以从平均值为零，标准差为0.1的高斯中得出的随机变量。因此，对于每个RGB图像像素I_{xy}= [ixr, ixg, ixb]，我们添加以下数量:

$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

式中p_i和λ_i分别为RGB像素值的3 ×3协方差矩阵的第i个特征向量和特征值，α_i为上述随机变量。对于特定训练图像的所有像素，每个α_i只绘制一次，直到该图像被再次用于训练，此时它将被重新绘制。这个方案近似地捕捉了自然图像的一个重要属性，即，对象身份对光照的强度和颜色的变化是不变的。该方案将top-1错误率降低了1%以上。

4.2 辍学

将许多不同模型的预测结合起来是一种非常成功的减少测试误差的方法[1,3]，但对于已经需要几天训练的大型神经网络来说，这似乎太昂贵了。然而，有一种非常高效的模型组合版本，在训练过程中只需要花费大约2倍的成本。最近引入的技术被称为“dropout”[10]，包括以0.5的概率将每个隐藏神经元的输出设置为零。以这种方式被“dropout”的神经元不参与前向传递，也不参与反向传播。所以每次提出一个输入时，神经网络都会采样一个不同的架构，但所有这些架构都共享权重。这种技术减少了神经元复杂的共同适应，因为一个神经元不能依赖于特定的其他神经元的存在。因此，它被迫学习更鲁棒的特征，这些特征与其他神经元的许多不同随机子集相结合是有用的。在测试时，我们使用所有的神经元，但将它们的输出乘以0.5，这是取指数级多个dropout网络产生的预测分布的几何平均值的合理近似。

我们在图2的前两个全连接层中使用dropout。没有dropout，我们的网络抑制了大量的过拟合。Dropout大约使收敛所需的迭代次数翻倍。

5 学习细节

我们使用随机梯度下降训练我们的模型，批量大小为128个示例，动量为0.9，权重衰减为0.0005。我们发现，这少量的权重衰减对模型的学习很重要。换句话说，这里的权重衰减不仅仅是一个正则化器:它减少了模型的训练误差。权重w的更新规则为

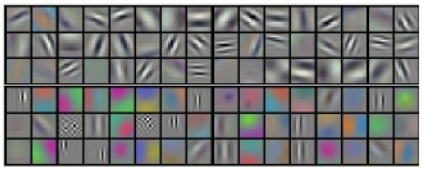

图3:第一个卷积层在224×224×3输入图像上学习到的96个大小的卷积核11×11×3。前48个内核在GPU 1上学习，后48个内核在GPU 2上学习。详情参见6.1节。

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

其中Di是迭代指数，v是动量变量，是学习率，∂L E是$\partial w\, wi\, Di$ —目标对w的导数在第i批中的平均值D_i，在wi进行评估。

我们从标准差为0.01的零均值高斯分布初始化每层的权重。我们用常数1初始化第二层、第四层和第五层卷积层以及全连接隐藏层中的神经元偏差。这种初始化通过为ReLUs提供正输入来加速学习的早期阶段。我们用常数0初始化剩余层中的神经元偏差。

我们对所有层使用了相同的学习率，在整个训练过程中我们手动调整。我们遵循的启发式方法是，当验证错误率停止随着当前的学习率提高时，将学习率除以10。学习率初始化为0.01和

在终止前减少三倍。我们通过120万张图像的训练集对网络进行了大约90个周期的训练，在两块NVIDIA GTX 580 3GB gpu上花费了五到六天的时间。

6 结果

我们对ILSVRC-2010的研究结果总结于表1。我们的网络实现了37.5%和17.0%的top-1和top-5测试集错误率[5]。在ILSVRC-2010竞赛期间，通过对六个基于不同特征训练的稀疏编码模型产生的预测进行平均的方法，取得的最佳表现为47.1%和28.2%[2]，此后，通过对基于两种密集采样特征计算的Fisher向量(fv)训练的两个分类器的预测进行平均的方法，发表的最佳结果为45.7%和25.7%[24]。

我们的模型还参加了ILSVRC-2012竞赛，并在表2中报告了我们的结果。由于ILSVRC-2012测试集标签不公开，我们无法报告我们尝试的所有模型的测试错误率。在本段的剩余部分，我们使用
验证和测试错误率可以互换，因为在我们的经验中，它们的差异不超过0.1%(见表2)。本文描述的CNN实现了

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

表1:ILSVRC-2010测试集的结果比较。*斜体*为他人取得的最佳结果。

top-5错误率为18.2%。预测的平均值
五个相似的cnn给出了16.4%的错误率。训练一个CNN，在最后一个池化层上额外增加第六个卷积层，对整个ImageNet 2011秋季版本(15M张图像，22K个类别)进行分类，然后在ILSVRC-2012上对其进行“微调”，错误率为16.6%。对两个cnn的预测进行平均，这两个cnn是在2011年秋季发布的整个版本上预先训练的，上面提到的五个cnn的预测错误率为15.3%。第二好的竞赛条目实现了26.2%的错误率，其方法是对从不同类型的密集采样特征计算的fv上训练的几个分类器的预测进行平均[7]。

最后，我们还报告了2009年秋季版本的ImageNet的错误率，其中包含10184个类别和890万张图像。在这个数据集上，我们遵循文献中的惯例，使用一半的图像进行训练，另一半用于测试。因为没有es-
既定的测试集，我们的划分必然不同于之前作者使用的划分，但这不会明显影响结果。我们的top-1和top-5错误率

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

表2:ILSVRC-2012验证和测试集的错误率比较。*斜体*为其他人取得的最佳结果。带有星号*的模型被“预先训练”以对整个ImageNet 2011秋季版本进行分类。详情参见第6节。

在这个数据集上是67.4%和
40.9%，由上面描述的网络获得，但在最后一个池化层之上增加了第6个卷积层。在这个数据集上发表的最佳结果是78.1%和60.9%[19]。

6.1 定性评估

图3显示了网络的两个数据连接层学习到的卷积核。网络学习了各种频率和方向选择的核，以及各种颜色的斑点。请注意两个gpu所表现出的专门化，这是3.5节中描述的受限连接的结果。GPU 1上的内核在很大程度上与颜色无关，而GPU 2上的内核在很大程度上与颜色相关。这种专门化在每次运行期间发生，并且独立于任何特定的随机权重初始化(对gpu的重新编号取模)。

[5] The error rates without averaging predictions over ten patches as described in Section 4.1 are 39.0% and 18.3%.

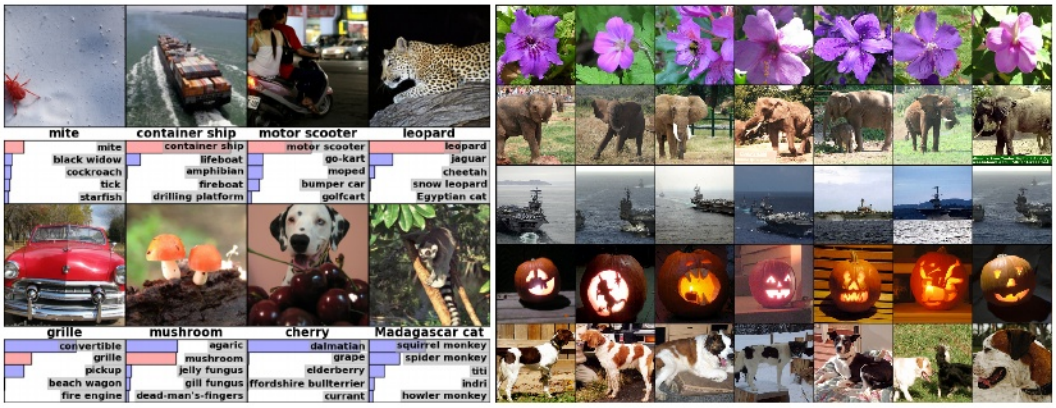

图4:(左)8张ILSVRC-2010测试图像和我们的模型认为最可能的5个标签。正确的标签写在每张图像的下面，分配给正确标签的概率也用红色条表示(如果它恰好在前5位)。(右)第一列中有5张ILSVRC-2010测试图像。剩余的列显示了在最后一个隐藏层中与测试图像的特征向量的欧氏距离最小的6个训练图像。

在图4的左面板中，我们通过计算对8个测试图像的前5个预测来定性地评估网络学习了什么。注意，即使是偏离中心的物体，比如左上角的螨虫，也可以被网络识别。排名前5的标签大部分看起来都是合理的。例如，只有其他类型的猫被认为是豹子的合理标签。在某些情况下(格栅，樱桃)，照片的预期焦点确实是模棱两可的。

另一种探索网络视觉知识的方法是考虑最后4096维隐藏层的图像诱发的特征激活。如果两幅图像产生的特征激活向量具有较小的欧氏分离，我们可以说神经网络的高层认为它们是相似的。图4显示了来自测试集的5张图像和来自训练集的6张图像，根据这个度量与它们每一张最相似。注意，在像素级别上，检索到的训练图像一般在L2上与第一列中的查询图像并不接近。例如，检索到的狗和大象以各种姿势出现。我们在补充材料中展示了更多测试图像的结果。

使用两个4096维实值向量之间的欧氏距离计算相似度是低效的，但通过训练一个自动编码器将这些向量压缩为短二进制代码可以提高效率。这应该会产生一种比将自动编码器应用于原始像素[14]更好的图像检索方法，[14]不利用图像标签，因此有检索具有相似边缘模式的图像的趋势，无论它们在语义上是否相似。

7 讨论

我们的结果表明，一个大型的深度卷积神经网络能够使用纯监督学习在一个极具挑战性的数据集上实现破纪录的结果。值得注意的是，如果去掉单个卷积层，我们的网络性能会下降。例如，删除任何中间层都会导致网络top-1性能损失约2%。因此，深度对于实现我们的结果真的很重要。

为了简化我们的实验，我们没有使用任何无监督的预训练，即使我们期望它会有所帮助，特别是如果我们获得足够的计算能力来显著增加网络的规模，而没有获得相应的标记数据量的增加。到目前为止，我们的结果已经有所改善，因为我们使我们的网络变得更大，训练它的时间更长，但我们仍然有许多数量级要做，以匹配人类视觉系统的下时间路径。最终，我们希望在视频序列上使用非常大和深度的卷积网络，其中时间结构提供了非常有用的信息，这些信息在静态图像中缺失或远不明显。

参考文献

[1] R.M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2): 75–79, 2007.

[2] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge 2010. www. image-net. org/ challenges. 2010.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] D. Cire¸san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *Arxiv preprint arXiv:1202.2745*, 2012.

[5] D.C. Cire¸san, U. Meier, J. Masci, L.M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. *Arxiv preprint arXiv:1102.0183*, 2011.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. *ILSVRC-2012*, 2012. URL http://www.image-net.org/challenges/LSVRC/2012/.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understand-ing*, 106(1):59–70, 2007.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, Cali-fornia Institute of Technology, 2007. URL http://authors.library.caltech.edu/7694.

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

[11] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *International Conference on Computer Vision*, pages 2146–2153. IEEE, 2009.

[12] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[13] A. Krizhevsky. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 2010.

[14] A. Krizhevsky and G.E. Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, 2011.

[15] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, et al. Hand-written digit recognition with a back-propagation network. In *Advances in neural information processing systems*, 1990.

[16] Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–97. IEEE, 2004.

[17] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[18] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsuper-vised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[19] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Metric Learning for Large Scale Image Classifi-cation: Generalizing to New Classes at Near-Zero Cost. In *ECCV - European Conference on Computer Vision*, Florence, Italy, October 2012.

[20] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. 27th International Conference on Machine Learning*, 2010.

[21] N. Pinto, D.D. Cox, and J.J. DiCarlo. Why is real-world visual object recognition hard? *PLoS computa-tional biology*, 4(1):e27, 2008.

[22] N. Pinto, D. Doukhan, J.J. DiCarlo, and D.D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS computational biology*, 5(11):e1000579, 2009.

[23] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[24] J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1665–1672. IEEE, 2011.

[25] P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, volume 2, pages 958–962, 2003.

[26] S.C. Turaga, J.F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H.S. Seung. Con-volutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2): 511–538, 2010.