ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
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为 1000 个不同的类别。在测试数据上，我们实现了 top-1 和 top-5 的错误率分别为 37.5%和 17.0%，大大优于之前的最先进方法(state-of-the-art,SOTA)。这个神经网络有 6000 万个参数和 65 万个神经元，由 5 个卷积层组成，其中一些层后伴随了最大池化层，以及 3 个全连接层，最后是 1000 路 softmax 层。为了使训练更快，我们使用了非饱和神经元和一个非常高效的 GPU 实现卷积操作。为了减少全连接层的过拟合，我们采用了最近开发的一种称为“dropout”的正则化方法，该方法被证明非常有效。我们还将这个模型的一个变种参加了 ILSVRC-2012 竞赛，并获得了 15.3%的 top-5 测试错误率，而第二名的错误率为 26.2%。

1 引言

当前的目标识别方法基本上都使用了机器学习方法。为了提高目标识别的性能，我们可以收集更大的数据集，学习更强大的模型，使用更好的技术来防止过拟合。直到最近，标注图像的数据集都相对较小--在几万张图像的数量级上（例如，NORB[16]，Caltech-101/256 [8, 9]和 CIFAR-10/100 [12]）。简单的识别任务在这样大小的数据集上可以被解决的相当好，尤其是如果通过标签保留变换进行数据增强的情况下。例如，目前在 MNIST 数字识别任务上（<0.3%）的最好准确率已经接近了人类水平[4]。但真实环境中的对象表现出了相当大的可变性，因此为了学习识别它们，有必要使用更大的训练数据集。实际上，小图像数据集的缺点已经被广泛认识到（例如，Pinto et al. [21]），但收集上百万图像的标注数据仅在最近才变得的可能。新的更大的数据集包括 LabelMe [23]，它包含了数十万张完全分割的图像，ImageNet[6]，它包含了 22000 个类别上的超过 1500 万张标注的高分辨率的图像。

为了从数百万张图像中学习几千个对象，我们需要一个有很强学习能力的模型。物体识别任务的巨大复杂性意味着即使是像 ImageNet 这样的大型数据集也无法完全描述这个问题，因此我们的模型应该也有许多先验知识来补偿我们所没有的数据。卷积神经网络(CNNs)构成了一个这样的模型[16, 11, 13, 18, 15, 22, 26]。它们的能力可以通过改变它们的广度和深度来控制，它们也可以对图像的本质进行强大且通常正确的假设（也就是说，统计的稳定性和像素依赖的局部性）。因此，与具有层次大小相似的标准前馈神经网络，CNNs 有更少的连接和参数，因此它们更容易训练，而它们理论上的最佳性能可能仅比标准前馈神经网络差一点。

尽管 CNN 具有引人注目的质量，尽管它们的局部架构相当有效，但将它们大规模的应用到到高分辨率图像中仍然是极其昂贵的。幸运的是，目前的 GPU，搭配了高度优化的 2D 卷积实现，强大到足够促进大型 CNN 的训练，最近的数据集例如 ImageNet 包含足够的标注样本来训练这样的模型而没有严重的过拟合。

本文具体的贡献如下：我们在 ILSVRC-2010 和 ILSVRC-2012[2]的 ImageNet 子集上训练了到目前为止最大的神经网络之一，并取得了迄今为止在这些数据集上报道过的最好结果。

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所有其他操作的GPU实现，并将其公开。我们的网络包含许多新的不寻常的特性，这些特性提高了神经网络的性能并减少了训练时间，详见第三节。即使使用了120万标注的训练样本，我们的网络尺寸仍然使过拟合成为一个明显的问题，因此我们使用了一些有效的技术来防止过拟合，详见第四节。我们最终的网络包含5个卷积层和3个全连接层，深度似乎是非常重要的：我们发现移除任何卷积层（每个卷积层包含的参数不超过模型参数的1%）都会导致更差的性能。

最后，网络尺寸主要受限于目前GPU的内存容量和我们能忍受的训练时间。我们的网络在两个GTX 580 3GB GPU上训练五六天。我们的所有实验表明我们的结果可以简单地通过等待更快的GPU和更大的可用数据集来提高。

2 数据集

ImageNet是一个包含超过1500万张高分辨率图像的数据集，涵盖了大约22,000个类别。这些图像是从网络上收集而来，并由人工标注者使用亚马逊的Mechanical Turk众包工具进行标注。从2010年开始，作为Pascal视觉对象挑战的一部分，每年都会举办一个名为ImageNet大规模视觉识别挑战(ILSVRC)的比赛。ILSVRC使用ImageNet的一个子集，每个类别大约有1000张图像。总共，训练集包含大约120万张图像，验证集包含5万张图像，测试集包含15万张图像。

ILSVRC-2010是ILSVRC竞赛中唯一可以获得测试集标签的版本，因此我们大多数实验都是在这个版本上运行的。由于我们也使用我们的模型参加了ILSVRC-2012竞赛，因此在第六节我们也报告了模型在这个版本的数据集上的结果，这个版本的测试标签是不可获得的。在ImageNet上，按照惯例报告两个错误率：top-1和top-5，top-5错误率是指测试图像的正确标签不在模型认为的五个最可能的便签之中。

ImageNet包含各种分辨率的图像，而我们的系统要求不变的输入维度。因此，我们将图像进行下采样到固定的256×256分辨率。给定一个矩形图像，我们首先缩放图像短边长度为256，然后从结果图像中裁剪中心的256×256大小的图像块。除了在训练集上对像素减去平均活跃度外，我们不对图像做任何其它的预处理。因此我们在原始的RGB像素值（中心的）上训练我们的网络。

3 架构

我们的网络架构概括为图2。它包含八个学习层--5个卷积层和3个全连接层。下面，我们将描述我们网络结构中的一些新奇的不寻常的特性。3.1-3.4小节按照我们对它们评估的重要性进行排序，最重要的最优先。

3.1 ReLU非线性

将神经元的输出f作为输入x的函数进行建模的标准方式是使用$f(x)=\tanh(x)$或$f(x)=(1+e^{-x})^{-1}$。考虑到梯度下降的训练时间，这些饱和的非线性比非饱和非线性$f(x)=\max(0,x)$更慢。根据Nair和Hinton[20]的说法，我们将这种非线性神经元称为修正线性单元(Rectified Linear Units,ReLU)。采用ReLU的深度卷积神经网络训练时间比等价的tanh单元要快几倍。在图1中，对于一个特定的四层卷积网络，在CIFAR-10数据集上达到25%的训练误差所需要的迭代次数可以证实这一点。这幅图表明，如果我们采用传统的饱和神经元模型，我们将不能在如此大的神经网络上实验该工作。

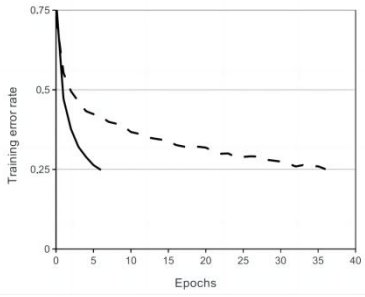

图 1：使用 ReLU(实线）的四层卷积神经网络在 CIFAR-10 数据集上达到 25%的训练误差比使用 tanh 神经元的等价网络(虚线)快六倍。为了使训练尽可能快，每个网络的学习率是单独选择的。没有采用任何形式的正则化。这里展示的影响程度因网络架构而异，但是使用 ReLUs 的网络始终比使用饱和神经元的等效网络学习速度快好几倍。

我们不是第一个考虑替代 CNN 中传统神经元模型的人。例如，Jarrett 等人声称非线性函数 $f(x)=|\tanh(x)|$ 在他们的类型的对比度归一化后跟局部平均池化的 Caltech-101 数据集上表现得特别好。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与我们使用 ReLU 时报告的加速拟合训练集的能力是不同的。更快的学习对于在大型数据集上训练的大型模型的性能有很大影响。

3.2 多 GPU 训练

单个 GTX580 GPU 只有 3G 内存，这限制了可以在 GTX580 上进行训练的网络最大尺寸。事实证明, 120 万图像用来进行网络训练是足够的，但网络太大因此不能在单个 GPU 上进行训练。因此我们将网络分布在两个 GPU 上。目前的 GPU 非常适合跨 GPU 并行，因为它们可以直接互相读写内存，而不需要通过主机内存。我们采用的并行方案基本上每个 GPU 放置一半的核（或神经元），还有一个额外的技巧：只在某些特定的层上进行 GPU 通信。这意味着，例如，第 3 层的卷积核从第 2 层的所有卷积核图中获取输入。然而，第 4 层的卷积核只从与其在同一 GPU 上的第 3 层的卷积核图获取输入。选择连接模式是一个交叉验证的问题，但这使我们能够精确调整通信量，直到它成为计算量的可接受的部分。

除了我们的列不是独立的之外（看图 2），最终的架构有点类似于 Ciresan 等人[5]采用的“columnar” CNN。与每个卷积层一半的核在单 GPU 上训练的网络相比，这个方案降分别低了我们的 top-1 1.7%，top-5 1.2%的错误率。双 GPU 网络比单 GPU 网络稍微减少了训练时间。

3.3 局部响应归一化

ReLUs 具有一个理想的特性，即它们不需要输入标准化来防止它们饱和。如果至少有一些训练例子对 ReLU 产生了积极的输入，学习将在该神经元中发生。然而，我们仍然发现以下的局部归一化方案有助于泛化。用 $a^i_{x,y}$ 表示通过在位置 (x,y) 上应用卷积核 i，然后应用 ReLU 非线性函数计算出的神经元的活动，响应归一化后的活动 $b^i_{x,y}$ 由以下表达式给出。

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

其中，求和运行在相同空间位置的 n 个“相邻”核图上，N 是该层中总核数。核图的排序当然是任意的，并且在训练开始之前确定。这种反应标准化实现了一种侧抑制形式，灵感来自于在真实神经元中发现的类型，在使用不同内核计算的神经元输出之间创造了对大活动的竞争。常数 k、n、α和β是超参数，其值是使用验证集确定的；我们使用 $k=2,n=5,\alpha=10^{-4}$，和 $\beta=0.75$。我们在某些层应用 ReLU 非线性函数之后应用了这种归一化（参见第 3.5 节）。

这个方案与 Jarrett 等人的局部对比度归一化方案有些相似，但我们更正确地称之为“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 错误率分别降低了 1.4%和 1.2%。我们还验证了这个方案在 CIFAR-10 数据集上的有效性：一个四层 CNN 在没有归一化的情况下达到了 13%的测试错误率，而在归一化的情况下达到了 11%。

3.4 重叠池化

CNN 中的池化层归纳了同一特征图上相邻组神经元的输出。一般来说，相邻池化单元归纳的区域是不重叠的（例如[17, 11, 4]）。更准确地说，池化层可以被认为是由间隔 s 个像素的池化单元组成，每个单元总结一个以池化单元位置为中心的大小为 $z \times z$ 的邻域。如果我们设置 $s = z$，我们得到 cnn 中常用的传统局部池。如果我们设置 $s < z$，我们得到重叠池。我们在整个网络中都使用这种方案，其中 $s = 2$ 和 $z = 3$。与不重叠的方案 $s = 2$，$z = 2$ 相比，这种方案将 top-1 和 top-5 错误率分别降低了 0.4%和 0.3%。尽管产生了相同维度的输出，但我们通常观察到，在训练过程中，具有重叠池化的模型稍微更难过拟合。

3.5 整体架构

现在我们准备描述我们的 CNN 的整体架构。如图 2 所示，我们的网络包含 8 个带权重的层；前 5 层是卷积层，剩下的 3 层是全连接层。最后一层全连接层的输出是 1000 维 softmax 的输入，它会产生一个 1000 类标签的分布。我们的网络最大化多项逻辑回归的目标，这等价于最大化预测分布下训练样本正确标签的对数概率的均值。

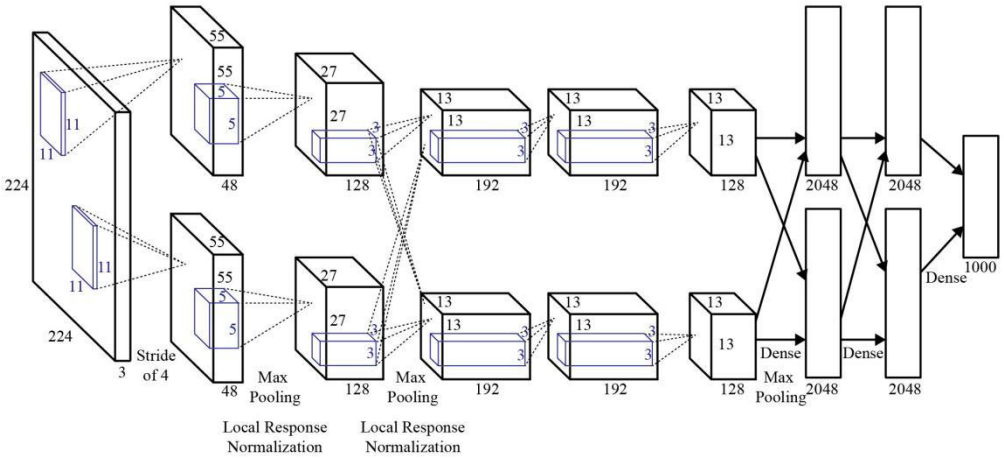

图 2：我们 CNN 的架构示意图，明确显示了两个 GPU 之间责任的划分。一个 GPU 在图的顶部运行层的部分，而另一个 GPU 在底部运行层的部分。GPU 之间只在某些特定的层进行通信。网络的输入是 150,528 维，网络剩余层中的神经元数量为 253,440-186,624-64,896-64,896-43,264-4096-4096-1000。

第 2，4，5 卷积核只与位于同一 GPU 上的前一层的特征图相连接（看图 2）。第 3 卷积层的核与第 2 层的所有核映射相连。全连接层的神经元与前一层的所有神经元相连。第 1，2 卷积层之后是响应归一化层。3.4 节描述的这种最大池化层在响应归一化层和第 5 个卷积层之后。ReLU 非线性应用在每个卷积层和全连接层的输出上。

第 1 个卷积层使用 96 个卷积核对 224 × 224 × 3 的输入图像进行滤波，核大小为 11 × 11 × 3，步长是 4 个像素（核映射中相邻神经元感受野中心之间的距离）。第 2 个卷积层使用第 1 个卷积层的输出（响应归一化和池化）作为输入，并使用 256 个卷积核进行滤波，核大小为 5 × 5 × 48。第 3，4，5 卷积层互相连接，中间没有接入池化层或归一化层。第 3 个卷积层有 384 个卷积核，核大小为 3 × 3 × 256，与第 2 个卷积层的输出（归一化，池化）相连。第 4 个卷积层有 384 个卷积核，核大小为 3 × 3 × 192，第 5 个卷积层有 256 个核，核大小为 3 × 3 × 192。每个全连接层有 4096 个神经元。

4 减少过拟合

我们的神经网络架构有 6000 万参数。尽管 ILSVRC 的 1000 类使每个训练样本从图像到标签的映射上强加了 10 比特的约束，但事实证明，这对于学习这么多参数而言是不足的，会导致严重的过拟合。下面，我们将描述我们应对过拟合的两种主要方法。

4.1 数据增强

图像数据上最简单常用的用来减少过拟合的方法是使用标签保留变换（例如[25, 4, 5]）来人工增大数据集。我们使用了两种独特的数据增强方式，这两种方式都可以从原始图像通过非常少的计算量产生变换的图像，因此变换图像不需要存储在硬盘上。在我们的实现中，转换后的图像是在 CPU 上以 Python 代码生成的，而此时 GPU 正在训练前一批图像。因此，实际上这些数据增强方案是计算免费的。

第一种数据增强的方式是生成图像的平移和水平翻转。我们通过从 256×256 的图像中提取随机的 224×224 的图像块（以及它们的水平翻转）来实现这一点，并在这些提取的图像块上训练我们的网络。这将训练集的大小增加了 2048 倍，尽管结果训练样本之间高度相关。如果没有这个方案，我们的网络将会受到严重的过拟合问题，这将迫使我们使用更小的网络。在测试时，网络通过提取五个 224×224 的图像块（四个角落图像块和中心图像块）以及它们的水平翻转（总共十个图像块），并对这十个图像块上的网络的 softmax 层的预测进行平均，来进行预测。

第二种数据增强的方式是改变训练图像中 RGB 通道的强度。具体而言，我们对整个 ImageNet 训练集的 RGB 像素值进行主成分分析（PCA）。对于每个训练图像，我们添加一定倍数的主成分，其大小与相应的特征值乘以从均值为零、标准差为 0.1 的高斯分布中随机抽取的随机变量成比例。因此，对于每个 RGB 图像像素 $I_{xy}=[I_{xy}^R, I_{xy}^G, I_{xy}^B]^T$，我们添加以下数量：

$$[p_1, p_2, p_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

其中 p_i 和 λ_i 分别是 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 是前面提到的随机变量。每个 α_i 仅在特定训练图像的所有像素中抽取一次，直到该图像再次用于训练时，才重新抽取。这个方案近似捕捉到了自然图像的一个重要特性，即物体的身份对照明的强度和颜色的变化是不变的。这个方案将 Top-1 错误率降低了 1%以上。

4.2 Dropout

将许多不同模型的预测结合起来是降低测试误差[1, 3]的一个非常成功的方法，但对于需要花费几天来训练的大型神经网络来说，这似乎太昂贵了。然而，有一个非常有效的模型结合版本，它只花费两倍的训练成本。这种最近引入的技术，叫做“dropout”[10]，它会以 0.5 的概率对每个隐层神经元的输出设为 0。那些“失活的”的神经元不再进行前向传播并且不参与反向传播。因此每次输入时，神经网络会采样一个不同的架构，但所有架构共享权重。这个技术减少了复杂的神经元互适应，因为一个神经元不能依赖特定的其它神经元的存在。因此，神经元被强迫学习更鲁棒的特征，它在与许多不同的其它神经元的随机子集结合时是有用的。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，这是对由指数级别的 dropout 网络产生的预测分布取几何平均的合理近似。

我们在图 2 中的前两个全连接层使用 dropout。如果没有 dropout，我们的网络表现出大量的过拟合。dropout 大致上使要求收敛的迭代次数翻了一倍。

5 学习细节

我们使用随机梯度下降来训练我们的模型，每批大小为 128 个实例，动量为 0.9，权重衰减为 0.0005。我们发现，这少量的重量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。重量 w 的更新规则为

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

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

其中 i 为迭代索引，u 为动量变量，ϵ是学习率，$\left\langle \frac{\partial L}{\partial w}|_{w_i} \right\rangle_{D_i}$是对于第 i 个批次$D_i$中目标函数关于$w$

的导数在w_i处的平均值。

我们使用均值为 0，标准差为 0.01 的高斯分布对每一层的权重进行初始化。我们在第 2，4，5 个卷积层和全连接隐层将神经元偏置初始化为常量 1。这个初始化通过为 ReLU 提供正输入加速了学习的早期阶段。我们在剩下的层将神经元偏置初始化为常数 0。

我们对所有层使用相同的学习率，并在训练过程中手动调整。我们遵循的启发式方法是当验证错误率在当前学习率下不再改善时，将学习率除以 10。学习率初始值为 0.01，在终止之前降低三次。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，这需要在两个 NVIDIA GTX 580 3GB GPU 上耗时五到六天。

6 结果

我们在 ILSVRC-2010 上的结果概括为表 1。我们的神经网络取得了 top-1 37.5%，top-5 17.0%的错误率。在 ILSVRC-2010 竞赛中最佳结果是 top-1 47.1%，top-5 28.2%，使用的方法是对 6 个在不同特征上训练的稀疏编码模型生成的预测进行平均，此后，最佳的已发布结果是通过对两种密集采样特征计算得到的 Fisher 向量(Fisher Vectors,FVs)训练的两个分类器的预测结果进行平均，其错误率分别为 45.7%和 25.7%[24]。

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 测试集上的结果对比。*斜体*是其它人取得的最好结果。

我们还将我们的模型参加了 ILSVRC-2012 竞赛，并在表 2 中报告了我们的结果。由于 ILSVRC-2012 测试集的标签不公开，我们无法报告我们尝试的所有模型的测试错误率。在本段剩余部分中，我们将验证错误率和测试错误率互换使用，因为根据我们的经验，它们之间的差异不超过 0.1%（见表 2）。本文描述的 CNN 实现了 18.2%的 top-5 错误率。将五个类似的 CNN 的预测结果进行平均，得到错误率为 16.4%。训练一个 CNN，在最后一个池化层之上添加额外的第六个卷积层，对整个 ImageNet Fall 2011 发布的数据集（1500 万张图像，2200 个类别）进行分类，然后在 ILSVRC-2012 上进行“微调”，得到错误率为 16.6%。将在整个 Fall 2011 发布的数据集上预训练的两个 CNN 的预测结果与前述的五个 CNN 进行平均，得到一个错误率为 **15.3%**。第二好的比赛参赛作品采用了一种方法，将从不同类型的密集采样特征计算得到的 Fisher 向量(FVs)训练的多个分类器的预测结果进行平均，其错误率为 26.2%[7]。

最后，我们还报告了我们在 ImageNet Fall 2009 版本上的错误率，该数据集包含 10,184 个类别和 890 万张图像。在这个数据集上，我们遵循文献中使用一半图像进行训练，一半图像进行测试的惯例。由于没有确定的测试集，我们的划分与先前作者使用的划分不同，但这并不明显影响结果。我们在这个数据集上的 top-1 和 top-5 错误率分别为 **67.4%**和 **40.9%**，这是通过上述网络，并在最后一个池化层之上添加了额外的第六个卷积层实现的。这个数据集上的最佳已发布结果是 78.1%和 60.9%[19]。

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%**
1CNN*	39.0%	16.6%	-
7CNNs*	36.7%	15.4%	**15.3%**

表 2：ILSVRC-2012 验证集和测试集的误差对比。斜线部分是其它人取得的最好的结果。带星号的是“预训练的”对 ImageNet 2011 秋季数据集进行分类的模型。更多细节请看第六节。

6.1 定性评估

图 3 显示了网络的两个数据连接层学到的卷积核。网络学到了各种频率选择性和方向选择性的卷积核，以及各种有色斑点。注意到两个 GPU 所展示的特化性，这是由第 3.5 节描述的受限连接性的结果。GPU 1 上的卷积核主要与颜色无关，而 GPU 2 上的卷积核主要与颜色相关。这种特化在每次运行时都会发生，并且与任何特定的随机权重初始化无关（除了 GPU 的重新编号）。

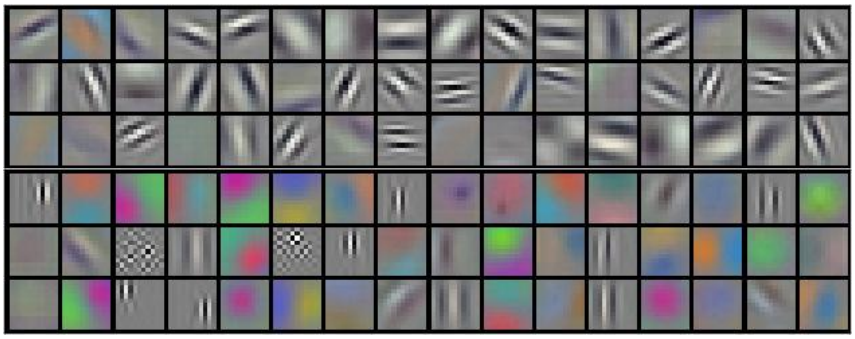

图 3：在 224×224×3 输入图像上，由第一个卷积层学习到的 96 个大小为 11×11×3 的卷积核。前 48 个卷积核是在 GPU 1 上学习的，后 48 个内核是在 GPU 2 上学习的。详见第 6.1 节。

在图 4 的左边部分，我们通过在 8 张测试图像上计算它的 top-5 预测定性地评估了网络学习到的东西。注意即使是不在图像中心的目标也能被网络识别，例如左上角的小虫。大多数的 top-5 标签似乎是合理的。例如，对于美洲豹来说，只有其它类型的猫被认为是看似合理的标签。在某些案例（格栅，樱桃）中，网络在意的图片焦点真的很含糊。

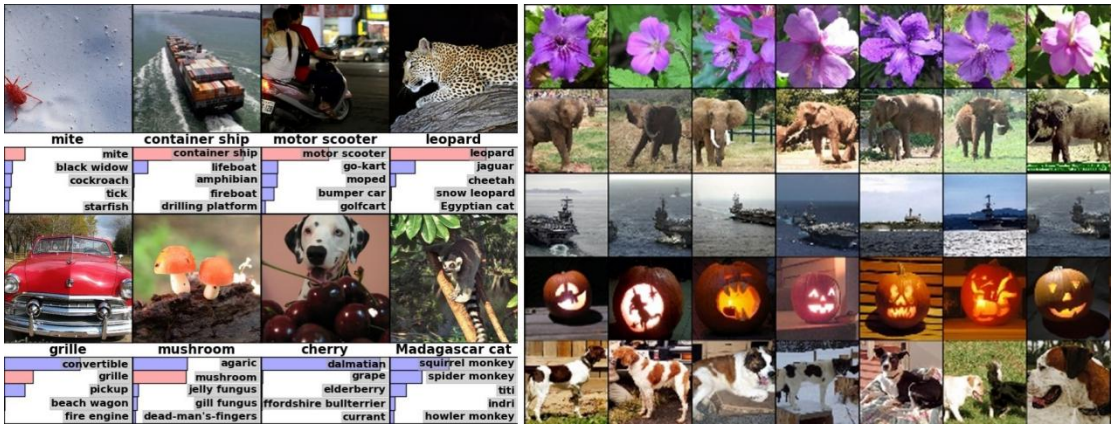

图 4：（左）8 张 ILSVRC-2010 测试图像和我们的模型认为最可能的 5 个标签。每张图像的下面是它的正确标签，正确标签的概率用红条表示（如果正确标签在 top 5 中）。（右）第一列是 5 张 ILSVRC-2010 测试图像。剩下的列展示了 6 张训练图像，这些图像在最后的隐藏层的特征向量与测试图像的特征向量有最小的欧氏距离。

探索网络可视化知识的另一种方式是思考最后的 4096 维隐藏层在图像上得到的特征激活。如果两幅图像生成的特征激活向量之间有较小的欧式距离，我们可以认为神经网络的更高层特征认为它们是相似的。图 4 表明根据这个度量标准，测试集的 5 张图像和训练集的 6 张图像中的每一张都是最相似的。注意在像素级别，检索到的训练图像与第一列的查询图像在 L2 上通常是不接近的。例如，检索的狗和大象似乎有很多姿态。我们在补充材料中对更多的测试图像呈现了这种结果。

通过两个 4096 维实值向量间的欧氏距离来计算相似性是效率低下的，但通过训练一个自动编码器将这些向量压缩为短二值编码可以使其变得高效。这应该会产生一种比将自动编码器应用到原始像素上[14]更好的图像检索方法，自动编码器应用到原始像素上的方法没有

使用图像标签，因此会趋向于检索与要检索的图像具有相似边缘模式的图像，无论它们是否是语义上相似。

7 探讨

我们的结果表明一个大型深度卷积神经网络在一个具有高度挑战性的数据集上使用全监督学习可以取得破纪录的结果。值得注意的是，如果移除一个卷积层，我们的网络性能会降低。例如，移除任何中间层都会引起网络损失大约 2%的 top-1 性能。因此深度对于实现我们的结果非常重要。

为了简化我们的实验，我们没有使用任何无监督的预训练，尽管我们希望它会有所帮助，特别是在如果我们能获得足够的计算能力来显著增加网络的大小而标注的数据量没有对应增加的情况下。到目前为止，我们的结果已经提高了，因为我们的网络更大、训练时间更长，但为了匹配人类视觉系统的下颞线（视觉专业术语）我们仍然有许多数量级要达到。最后我们想在视频序列上使用非常大的深度卷积网络，视频序列的时序结构会提供非常有帮助的信息，这些信息在静态图像上是缺失的或远不那么明显。

References

[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.imagenet.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 Understanding, 106(1):59–70, 2007.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California 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 networks 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. Handwritten 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 unsupervised 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 Classification: 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 computational 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. Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Computation, 22(2):511–538, 2010.