基于卷积神经网络的ImageNet分类器

作者：

Alex Krizhevsky-多伦多大学（加拿大）

Ilye Sutskever-多伦多大学

Geoffrey E. Hinton-多伦多大学

摘要

我们训练了一个大型的深度卷积神经网络去将2010年ILSVRC挑战杯包含的120万高分辨率图像分类成1000种不同的类别。在测试数据方面，我们取得了远超过去最佳水平的效果，分别为17%和37.5%的top-5和top-1错误率。有着6000万参数和65万神经元的神经网络由5个部分连接Max池化层的卷积层和3个全连接层连带着1000路softmax组成。为了加快训练速度。我们采用非饱和神经元和一个高效的卷积操作的GPU执行器。为了降低全连接层的过拟合，我们采用了一项近期发展的已被证明有效的名为dropout的正则化方法。

1 引言

解决物体识别的最新方法必不可少的使用机器学习方法。为了提高他们的表现，我们可以收集更大的数据集，训练更有效的模型，并且使用更先进的技术去阻止过拟合。直到近期，有标识的图像数据集相当的小——大约数万张图片的状况才改变。简单的识别任务能够被有效的解决好在这一规模的数据集上，特别是如果他们采用了数据增强。例如，MNIST数字识别任务的最新错误率（0.3%）已接近人类表现。但现实场景中的对象表现出相当大的变异性，所以为了学习识别它们，使用更大的训练集是非常必要的。事实上，小规模的图像数据集的缺陷已经被广泛的认识到，近来收集更大规模有标识的数据集变得有可能了。新的更大的数据集包括LabelMe[23]和ImageNet。

为了从数百万的图像中学习上千种对象，我们需要有强大学习能力的模型。尽管图像识别任务巨大的复杂度意味着这一问题无法被如ImageNet大的数据集规定，所以我们的模型也应该有大量的先验知识来补偿我们没有的所有数据。卷积神经网络就是这样一类模型。他们的能力可以通过改变他们的深度和广度来控制，他们也对图像的性质做出了强有力的、基本上正确的假设（即统计平稳性和像素依赖性的局部性）。因此，相比具有同样层数的标准的前馈神经网络，CNN有更少的连接和参数使得它们更容易培养，而理论上的最佳性能可能略差。

尽管CNN有诱人的质量，尽管它们的架构相对效高效，它们在高分辨率图像的大规模应用上的代价昂贵。幸运的是，目前的GPU，搭配一个高度优化的二维卷积实现方式，足够有力的去帮助大型神经网络的训练，和最近如ImageNet的包含足够的标记的例子的数据来培养没有严重的过拟合的模型。

本文的具体贡献如下：我们训练一个基于ImageNet的用于ILSVRC-2010和ILSVRC-2012比赛的数据子集的最大的卷积神经网络并且在这些数据集上取得了迄今为止最好的结果。我们编写了一个高度优化的2D卷积的GPU实现和训练卷积神经网络我们公开使用的所固有的所有其他操作。我们的网络包含一些改善其性能和降低其训练时间不寻常的新特征，在第3节中详细。我们的网络规模使得过拟合成为一个重要的问题，即使有120万个标记的训练实例，所以我们用了几个有效的方法防止过拟合，在第4节中描述。我们的最终网络包含五个卷积和三个完全连接的层，这个深度似乎很重要：我们发现去掉任何卷积层（每一层包含不超过模型参数的1%）会导致性能低下。

最后，网络的规模主要受限于可用内存量在当前GPU和我们愿意容忍的训练时间。我们的网络需要五至六天时间去训练在两个GTX 580 3GB 的GPU。我们的实验表明，我们的结果可以通过等待更快的GPU和更大的数据集来实现简单的改进。

2 数据集

ImageNet 是一个属于大约有22000类别的超过1500万标记的高分辨率图像的数据集。图片是从网上收集的并且被人工添加标识。从2010开始，作为帕斯卡视觉对象挑战杯的一部分，名为ImageNet大规模视觉识别的挑战（ILSVRC）每年举行一次。ILSVRC使用ImageNet的一个子集，每1000个类别中大约有1000个图像。总共有大约120万个训练图像，50,000个验证图像和150,000个测试图像。

ILSVRC-2010是ILSVRC的唯一可用的有标识的测试集版本，因此这是我们执行大部分实验的版本。因为我们也搭建了我们的模型在2012年的ILSVRC挑战杯，我们会在第6节报告在这个版本的数据集上的的结果，而它的测试集标签无法获取的。在ImageNet上，习惯上报告两种错误率：top-1和top-5，其中top-5错误率是正确标签不在被模型认为最可能的五个标签之中的测试图像的分数。

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度。因此，我们将图像欠采样到256*256的固定分辨率。给定一个矩形图像，我们首先重新缩放图像，使得短边的长度为256，然后从结果图像中裁剪出中心256 * 256的部分。除了从每个像素中减去训练集上的平均激活值之外，我们没有以任何其他方式预处理图像。 所以我们在像素的（中心化的）原始的RGB值上训练了我们的网络。

3 构架

图2总结了我们网络的体系结构。它包含八个已学习的层——五个卷积和三个完全连接。下面，我们描述一些我们网络架构的新颖或不寻常的特征。3.1-3.4节按照我们对它们重要性的估计进行排序，其中最重要的是第一个。

3.1：ReLU的非线性

去建模一个神经元的输出——作为以x为输入的函数f的标准方式是f(x)=tanh(x)或f(x)=sigmoid(x)。就梯度下降训练时间而言，这些饱和非线性即f(x) = max(0; x)比非饱和非线性慢得多。遵循Nair和Hinton，我们将具有这种非线性的神经元称为整流线性单位（ReLUs）。使用ReLU的深度卷积神经网络的训练速度比使用tanh同等规模的神经网络快上几倍。图1展示了这一点，图中显示了特定四层卷积网络在CIFAR-10数据集上达到25%训练误差所需的迭代次数。

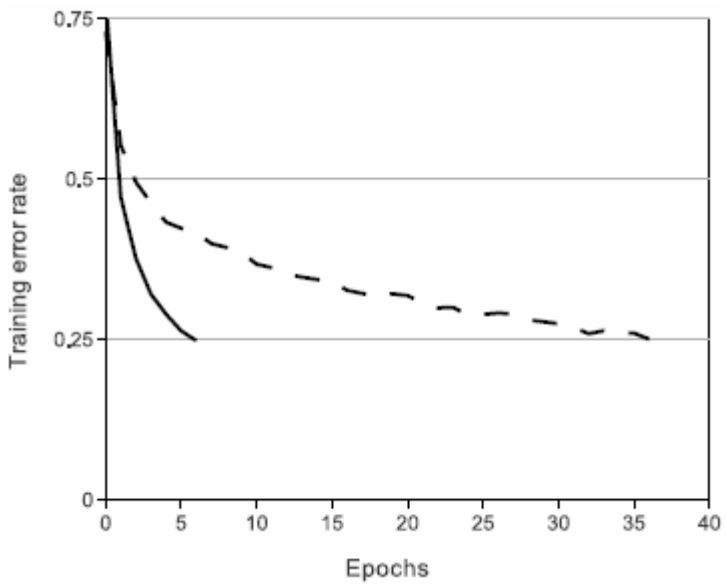

图1：具有ReLUs（实线）的四层卷积神经网络在CIFAR-10上的训练错误率达到25%，比具有tanh神经元（虚线）的网络快6倍。每个网络的学习率都是独立选择的以尽可能快地训练。 没有采用任何形式的正规化。 这里展示的效应的大小随网络架构而变化，但具有ReLU的网络一直学习速度比饱和神经元快几倍。

我们不是第一个在CNN中考虑传统神经元模型的替代方案。例如，Jarrett等人 声称非线性f（x）=| tanh(x)|它们的对比度归一化类型特别适用在Caltech-101数据集上并连接本地平均池化层。然而，在这个数据集中，主要关注的是防止过度拟合，所以他们观察到的效果与我们在使用ReLU时报告的训练集合的加速能力不同。加快学习对大型数据集上训练的大型模型的性能有很大的影响。

3.2：在多GPU上训练

单个GTX 580 GPU只有3GB内存，这限制了可以在其上训练的网络的最大尺寸。事实证明，120万个训练样例足以训练那些尺寸太大而不适合一个GPU的网络。因此，我们将网络分布在两个GPU上。目前的GPU特别适合于跨GPU并行化，因为它们能够直接读写对方的内存，而无需通过主机内存。我们所采用的并行化方案基本上在每个GPU上放置了一半的内核（或神经元），还有一个额外的技巧：GPU仅在某些层次上进行通信。这意味着，例如，第3层的内

核从第2层的所有内核映射中获取输入。但是，第4层中的内核只能从驻留在同一GPU上的第3层中的那些内核映射接收输入。选择连通性模式是交叉验证的一个问题，但这使我们能够精确调整通信量，直到它达到计算量的可接受部分。

除了我们的输入值不是独立的（见图2）之外，由此产生的架构有点类似于Cire¸san等人所使用的“柱状”CNN。与一个GPU上训练的每个卷积层内核数量减少一半的网络相比，这个方案分别将我们的top-1和top-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网络的训练时间稍少。

3.3：局部响应归一化

ReLU具有理想的属性，它们不需要输入规范化来防止它们饱和。如果至少有一些训练实例为ReLU提供了正输入值，则学习将会如此发生在那个神经元。然而，我们仍然发现以下的局部标准化方案有助于泛化。式中，ax,yi表示内核i在位置（x,y）上计算的激活值，然后通过ReLU非线性变化得到响应的标准化激活值bx,yi：

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

式中，总和运行在n个映射在相同空间位置上的“相邻的”内核，N是该层中核的总数。内核映射的排序当然是任意的，并且在训练开始之前确定。这种响应归一化实现了一种受真实神经元中发现的类型所激发的横向抑制形式，造成神经元输出的大数值的激活值的竞争使用不同的内核。常量k、n、α和β是超参数，其值是使用验证集确定的; 我们取k = 2, n = 5,α= 10-4和β= 0.75。我们在应用某些层的ReLU非线性后应用了这种规范化（参见第3.5节）。

该方案与Jarrett等人的局部对比归一化方案有一些相似之处。但是我们的将被更准确地称为“亮度标准化”，因为我们不会减去平均激活值。响应规范化将我们的top-1和top-5的错误率分别降低1.4%和1.2%。我们还验证了这种方案在CIFAR-10数据集上的有效性：没有标准化的四层CNN实现了13%的测试错误率，而有标准化的为11%。

3.4：重叠池化

CNN中的池化层概括了相同内核映射中相邻神经元组的输出。一般地，被邻接的池化单元总结的邻居节点是没有重复的。更准确地说，池化层可以被认为是由间隔s个像素的池化单元的网格组成，每个总结值以集中单元的位置为中心的大小为z * z的邻域。如果我们设置s = z，我们就可以获得CNN中常用的传统局部池。如果我们设置s <z，我们获得重叠池。这是我们在整个网络中使用的，s = 2和z = 3。与产生等效尺寸的输出的非重叠方案s = 2和z = 2相比，该方案分别将top-1和top-5的错误率分别降低了0.4%和0.3%。我们通常在训练期间观察到重叠池的模型发现稍微难以过拟合。

3.5总体构架

现在我们准备好描述CNN的整体架构。 如图2所示，这个网络包含八个带权重的层; 前五个是卷积层，其余三个全连接层。最后全连接层的输出被馈送到1000路softmax，其产生1000个类别标签上的分布。 我们的网络最大化多项逻辑回归函数，这相当于在预测分布下最大化正确标签的对数概率的训练案例的平均值。

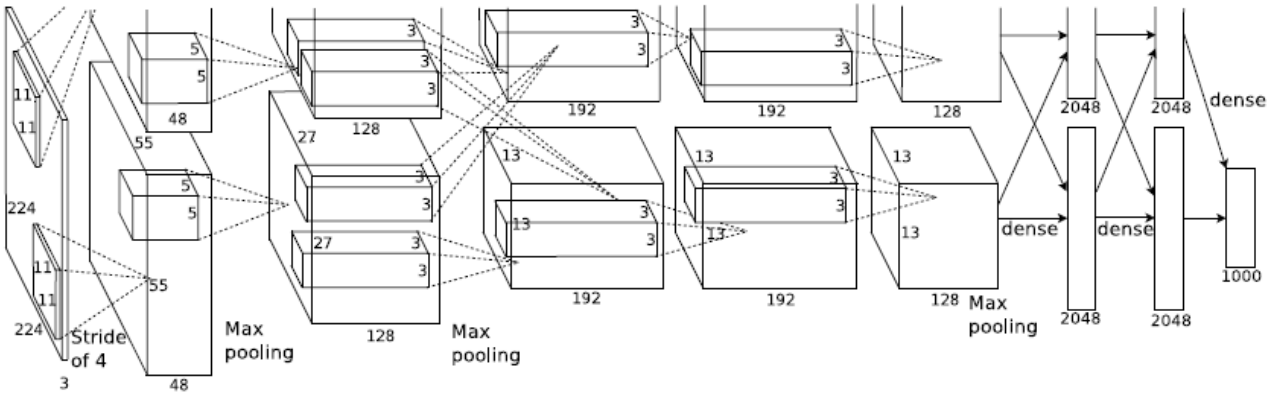

图2：总体架构图，明确显示两个GPU之间职责的划分。一个GPU运行图形顶部部分，另一个运行图形底部部分。GPU仅在特定层通信。网络的输入是150,528维，网络剩余层中的神经元数为253,440-186,624-64,896-64,896-43,264-4096-4096-1000。

第二，第四和第五卷积层的内核仅与位于同一GPU上的前一层中的那些内核映射相连。第三卷积层的内核连接到第二层中的所有内核映射。全连接层中的神经元连接到前一层中的所有神经元。响应标准化层连在第一和第二卷积层。3.4节中描述的最大池化层连在响应规范化层以及第五卷积层。将ReLU非线性应用于每个卷积和完全连接层的输出。

第一卷积层用96个大小为11 * 11 * 3的过滤器以4个像素的步幅卷积224 * 224 * 3输入图像。第二卷积层将第一卷积层的（响应归一化和池化）输出作为输入，并用大小为256×5×48的过滤器对其进行卷积。第三，第四和第五卷积层彼此连接而没有任何池化层或标准化层。第三卷积层具有连接到第二卷积层的（响应归一化和池化）输出的大小为3 * 3 * 256的384个过滤器。第四卷积层具有384个大小为3 * 3 * 192的过滤器，并且第五卷积层具有大小为3 * 3 * 192的256个过滤器。全连接层各有4096个神经元。

4 减少过拟合

我们的神经网络拥有6000万的参数，虽然ILSVRC的1000个类别将从图片到标签的映射限制在10个bits，这依然不足以训练这么多的参数而不造成过拟合。下面，我们将介绍两种处理过拟合的基本方法。

4.1：数据增强

最简单最常用的减少过拟合的方法就是利用标签保存变形技术人工放大数据集。我们采取了两种不同形式的数据放大，它们都允许在仅对原图做少量计算的情况下产生变形的新图，所以变形后的新图无需存储在硬盘中。在我们的实现中，变形的新图由Python在CPU上计算产生，与此同时，GPU仍在计算其他的之前批次的图片。所以这种放大数据集的方式是很高效很节省计算资源的。

第一种形式的数据增强包括生成图像平移和水平反射。我们通过从256 * 256个图像中提取随机224 * 224块（及其水平反射）并在这些提取的块4上训练我们的网络来实现这一点。这使得我们的训练集的规模增加了2048倍，尽管由此产生的训练样例当然是高度相互依赖的。如果没有这个方案，我们的网络会遭受大量的过度训练，这将迫使我们使用更小的网络。在测试时间，网络通过提取五个224*224方块（四个角方块和中心方块）以及它们的水平反射（因此共有十个方块），并对网络的softmax层对十个方块进行的预测进行平均。

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，我们在整个ImageNet训练集的RGB像素值集上执行PCA。 对于每个训练图像，我们添加多个找到的主成分，大小与相应的特征值成比例，乘以从均值为零和标准差为0.1的高斯绘制的随机变量。因此，对于每个RGB像素 Ixy =[IxyR , IxyG , IxyB]T我们加入的值如下：

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

其中，pi 和λi分别是第i个特征向量和第i个3x3RGB协方差矩阵的本征值。而αi是前面所述的随机变量。对于一张特定的训练图片的所有像素，每个αi仅被抽取一次，直到这张图像再次被用于训练才会再次提取随机变量。这一方案能够近似地捕捉原始图像的一些重要特征，即那些不随光线强度与颜色变化的物体特质。这一方法把top-1错误降低了1%。

4.2：Dropout

降低测试错误的一种有效方法是联立多种不同模型的预测结果，但这种方法对于大型神经网络来说似乎太昂贵了，需要好几天去训练。然而，有一种非常高效的模型联立方法，只需要在训练过程中消耗一到两个因子。这种新近研究出来的技术叫做“DROPOUT”，它会以50%的概率将每个隐藏层神经元置零。以这种方法被置零的神经元不再参与前馈和BP过程。所以每次一个输入进来之后，这个神经网都会被置于不同的结构，但所有这些结构共享同一套参数。这种技术降低了神经元间相互适应的复杂性，因为每个神经元都不可能依赖其他特定某个神经元的表现。因此，模型被迫学习更加健壮的特征，使之能够被许多不同的随机神经元子集使用。在测试中，我们使用所有的神经元，但是把它们的输出乘以0.5，这是一种对大量dropout网络产生的预测分布的几何均值的合理近似。

我们在图2中的前两个全连接层使用dropout。否则，我们的网络会表现出严重的过拟合。dropout大概会让达到收敛所需要的迭代次数翻倍。

5 训练细节

我们每个训练批次有128个样本，在其上采用随机梯度下降进行训练。设置增量为0.9，权值衰退因子为0.0005。我们发现小的权重衰退因子对于模型学习很重要，换句话说，权重衰退因子在这里不光是个正则化因子，它还可以减少模型错误。权值w的更新规则是：

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

其中，i是迭代次数，v是增量，ε是学习速率。

我们将每一层的权值利用均值为0方差为0.01的高斯分布随机初始化，我们用常数1初始化第2、4、5卷积层和全连接隐藏层的偏置神经元（常数单元）。这种初始化通过向ReLUs提供正输入，加速了学习的早期过程。我们将其它层的偏置神经元初始化为0。

在整个学习过程中，我们在所有层都使用人工调整的相等的学习速率。我们采用的启发式方法是当验证误差不在降低时，就把当前的学习速率除以10。学习速率初始化为0.01，并在结束前减小3次。（做三次除以10）我们大概用120万张图片把我们的网络训练了约90轮，在两个NVIDIA GTX 580 3GB GPU上这大概要5到6天。

6 实验结果

我们在ILSVRC-2010数据集上的实验结果归纳在表1里。我们的网络top-1和top-5测试误差分别是37.5%和17.0%。在此之前ILSVRC- 2010数据集上的最好的比赛纪录是对在不同特征上训练的留个稀疏自编码器取平均，top-1和top-5测试误差分别是47.1%和28.2%。

之后，已出版的最佳结果是一种对两个在不同取样密度的费舍向量上训练的分类器取平均的方法，结果是45.7%和25.7%。

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。斜体字是其他人取得的最好结果。

我们也让我们的模型参加了ILSVRC-2012的比赛，并在表2中展示了我们的结果。因为ILSVRC-2012测试集的标签并未公开，所以我们不能报告我们所有试过的模型的测试错误率。在这一段的余下部分，我们使用验证误差代替测试误差，因为根据我们的经验，它们的差距不会大于0.1%（见表2）。本文描述的CNN实现了18.2%的top-5错误率。五个相似CNN的预测值的平均值的误差率为16.4%。 培训一个CNN，在最后一个池层上增加一个额外的第六卷积层，对整个ImageNet 2011秋季赛发布数据（15M图像，22K类别）进行分类，然后在ILSVRC-2012上对其进行“微调”，则错误率为16.6%。对整个2011年秋季赛发布数据预训练的两个CNN与上述五个CNN进行的平均预测给出了15.3%的错误率。 第二好的比赛录入达到了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出的FV进行训练的几个分类器的预测进行平均。

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验证和测试集错误率的比较。斜体字是其他人取得的最好结果。带有*的是“预先训练好的”，用于对整个ImageNet 2011秋季赛进行分类。 详情请参阅第6节。

最后，我们还在2009年秋季版ImageNet上报告了10,184个类别和890万个图像的错误率。 在这个数据集中，我们遵循文献中使用一半图像进行训练和一半进行测试的惯例。 由于没有建立测试集，我们的分割必然不同于先前作者使用的分割，但这并不会对结果产生显着影响。 在这个数据集中，我们的前1和前5的错误率分别是67.4%和40.9%，通过上面描述的网络获得，但是在最后的池化层上具有额外的第六卷积层。 该数据集的最佳公布结果是78.1%和60.9%。

6.1：定量分析

图3显示了网络的两个数据连接层学习的卷积核。 该网络已经学习了各种频率和方向选择内核，以及各种彩色斑点。请注意两个GPU表现除出了不同的特性，这是3.5节介绍的限制互联方式的结果。GPU 1上的内核基本上不在意颜色，而GPU 2 上的内核就是色彩专家。这种专一性每次都会出现，与权值的随机初始化无关（GPU重新编号）。

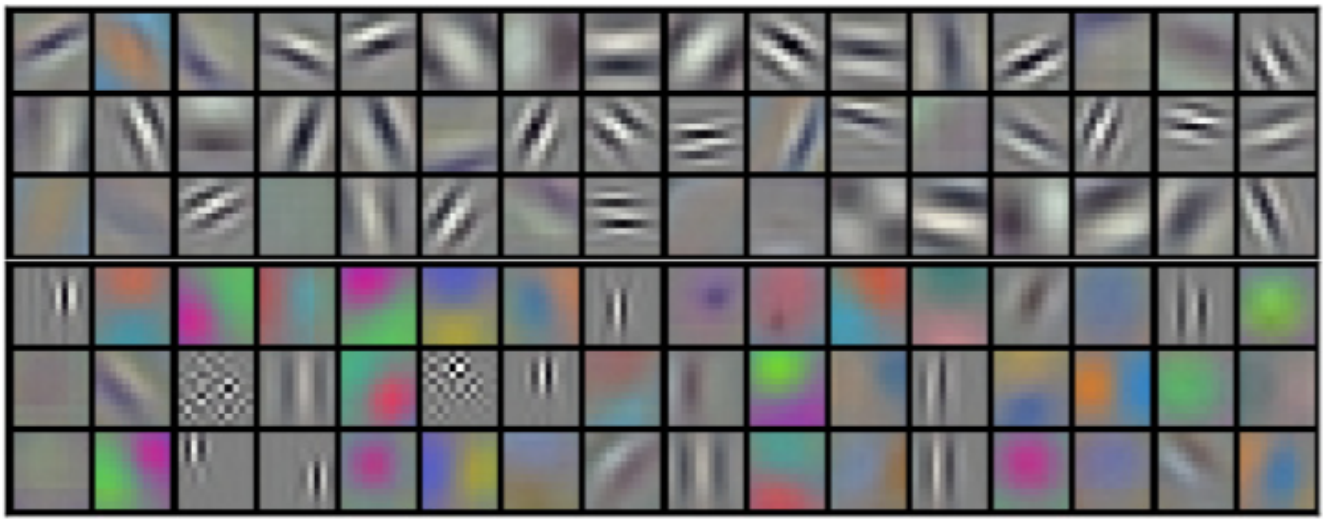

图3：96个通过第一个卷积层学习224x224x3的图片得到的11x11x3的卷积内核。上面48个和下面48个分别由两个GPU学习得到，详见6.1.

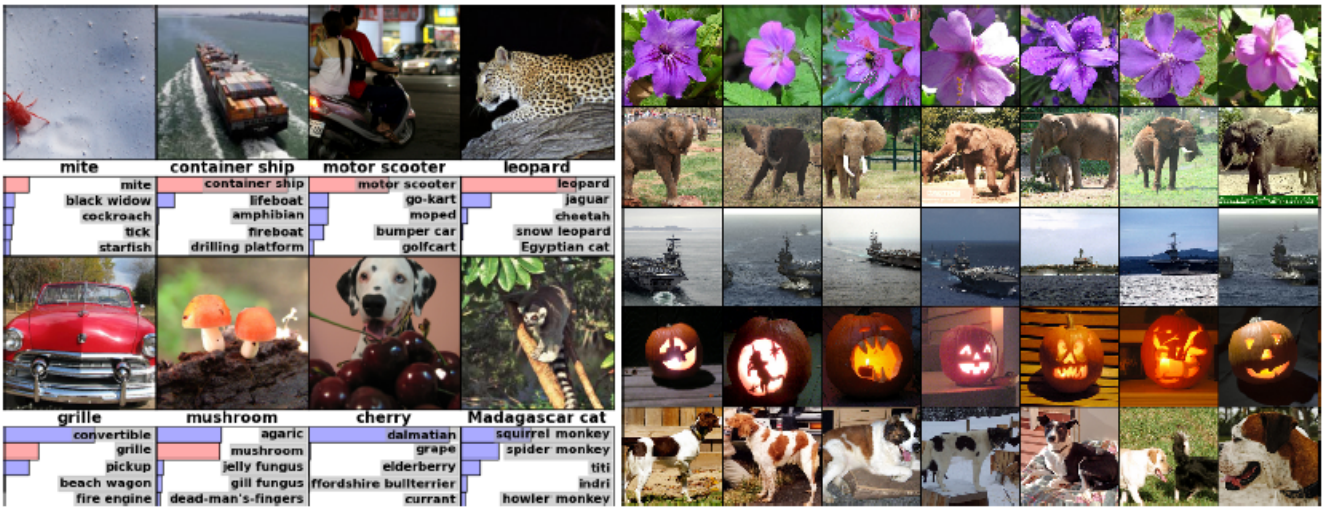

图4 :(左）8个ILSVRC-2010测试图像和我们模型最可能考虑的5个标签。每张图像下面都写有正确的标签，并且标有正确标签的概率也显示为红色条（如果是 恰好在前5名）。（右）第一列有5个ILSVRC-2010测试图像。其余的列显示了六个训练图像，其产生与测试图像的特征向量具有最小欧几里德距离的最后隐藏层中的特征向量。

在图四的左侧，我们定量地展示了对于8张图片网络所学习到的前五个预测。注意对于偏离中心的物体，比如左上角的那只螨虫，网络依然可以识别出来。大多数前五个标签看起来都比较合理，比如，只有其他类别的猫科动物才被判别是豹子的可能标签。在一些例子中，比如栅栏，樱桃，确实对于究竟该关注哪个物体存在歧义。

另一个研究可视化网络所学知识的方法是考虑最后一个4096维隐层所激活的特征向量。如果两张图的向量欧氏距离很小，我们可以说很大程度上神经网络认为它们是相似的。图4展示了五张测试集中的图片，以及按照上述方法找出的分别与这五张图最相似的6张训练集图片。请注意，在像素级别，检索到的训练图像在L2中通常不会与第一列中的查询图像接近。 例如，检索到的狗和大象出现在各种姿势中。 我们在补充材料中提供更多测试图像的结果。

通过使用两个4096维实值向量之间的欧几里得距离来计算相似性效率不高，但通过训练自动编码器将这些向量压缩为短二进制代码，可以使其有效。 与将自动编码器应用于未使用图像标签的原始像素相比，这应该会产生更好的图像检索方法，因此具有检索具有类似边缘图像的图像的趋势，而不管它们是否在语义上相似。

7 讨论

我们的研究结果表明，一个大的深层卷积神经网络能够使用纯监督学习在具有高度挑战性的数据集上实现破纪录的结果。值得注意的是，如果单个卷积层被删除，我们的网络性能就会下降。例如，删除任何中间层导致网络性能前1的性能损失约2%。所以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，即使我们预计它会有所帮助，特别是如果我们获得足够的计算能力来显着增加网络的大小而不会相应增加标记数据的数量。 到目前为止，我们的结果已经改善，因为我们已经使我们的网络更大并且训练了它更长的时间，但是为了匹配人类视觉系统的时间 - 地球路径，我们仍然有很多数量级要去。 最终，我们希望对视频序列使用非常大而深的卷积网络，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不存在或不太明显。