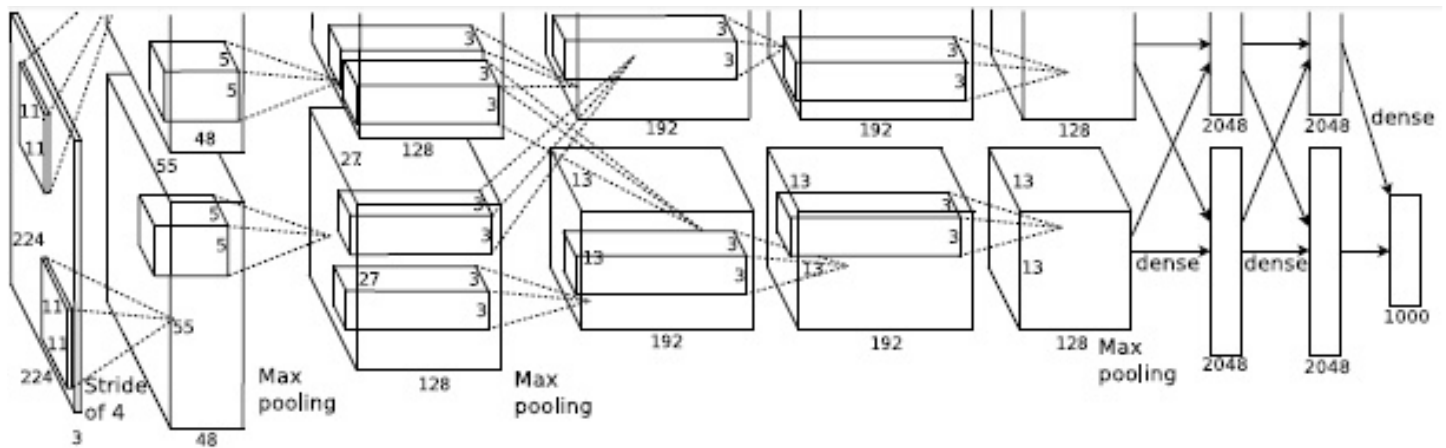


知乎



[原创翻译] 运用CNN对ImageNet进行图像分类



卓柳舟
中国早起要饭第一人

73 人赞了该文章

译者注：本篇翻译自Alex Krizhevsky, Ilya Sutskever以及Geoffrey E.Hinton的论文 *ImageNet Classification with Deep Convolutional Neural Networks*. 该论文在智能单元专栏文章《CS231n课程笔记翻译：神经网络笔记1（上）》中有提到，因此打算翻译一下，一来强化自己的理解，二来供读者参考。文中涉及的名词翻译主要是参考智能单元系列专栏文章《CS231n课程笔记翻译》中的名词翻译以及百度和谷歌。其实对于名词，我偏向于保留其原貌，因为有的名词本身很直观，所以在翻译后面会跟上原文。

摘要

我们利用ImageNet LSVRC-2010比赛的数据训练了一个庞大、深度的卷积神经网络，将一百二十万张高分辨率图片分为1000个类别。将模型运用于测试数据时，我们得到喜人的成果：Top1错误率及Top5错误率（即预测的第一个或前五个类别中不包含正确类别的比例）分别为37.5%和17.0%，优于过往最好的测试结果。我们的模型包含6000万个参数和65万个神经元（neuron），由5个卷积层构成，部分卷积层附带一个最大值汇合层（max pooling），以及3个全连接层（fully-connected layers），最后一层是1000-way softmax评分函数（1000个output score）。为了使训练过程更加迅速，我们使用非饱和神经元(non-saturating neurons)以及在GPU上高效地训练CNN。而为了降低过拟合程度，在全连接层中我们使用了近期发展出的正则化方法-“随机失活（Dropout）”，在实际训练中，随机失活的效果很好。我们还用我们的CNN模型的一个变体对ImageNet ISVRC-2012比赛的数据进行训练，获得了Top5错误率15.3%的骄人成绩，而比赛的亚军的错误率是26.2%。

1 引言



知乎

- 大约是以万计（比如，NORB [16], Caltech-101/256 [8, 9], 以及CIFAR-10/100 [12]）。这样的数据量可以完成简单的识别任务，尤其是当数据经过保留标签转换（label-preserving transformations）处理后。例如，目前为止MNIST数字识别项目最低的错误率（ $<0.3\%$ ）基本达到人类的识别水平[4]。但是现实环境中的物体展示出多样性，所以想要获得更好的识别能力需要更加庞大的训练数据。其实，数据量小的缺点已经是众所周知，但一直到最近才可以收集到百万级的带标签图像数据集。最新的大型数据库包括LabelMe [23] - 其内含数十万张完全分块化图像，以及ImageNet [6] - 其内含超过1500万张、共22000多类带标签的高分辨率图像。

为了从百万计的图像中训练适用于数千个图像的分类模型，我们需要一个具有很大训练容量的模型。但是，图像识别问题本身极大的复杂性使得即使利用ImageNet这样庞大的数据库也还是难以完美地解释该问题。所以为了弥补数据的不足，我们的模型需要基于许多前人研究。CNN就是符合我们要求的一类模型 [16,11,13,18,15,22,26]。通过调整CNN模型的深度(depth)和宽度(breadth)，我们可以控制其容量。另外CNN模型对图像性质的假设全面且基本正确（即统计量的定态假设以及像素局部依赖性假设）。因此，相比于传统的前馈神经网络（feedforward neural networks）使用相同大小的中间层，CNN使用更少的连接数和参数，所以能够更好地训练模型，同时其理论最优识别能力仅稍弱于传统模型。

即使CNN具有良好的性质以及相对高效的架构，应用于大规模的高清图像时还是很耗资源。幸运的是，现在的显卡（GPU）配合高度优化的二维卷积，已经可以用于训练出乎意料的大型CNN模型。另外像ImageNet这样最新的数据集包含足够的标签样本，可以很好地训练CNN模型的同时不引起严重过拟合问题。

本文的主要贡献是：我们基于ILSVRC-2010及ILSVRC-2012的ImageNet数据集，训练了目前最大的CNN，并且取得了目前为止对该数据集最好的分类结果。我们编写了一个高度优化的在GPU环境执行的二维卷积过程，以及其他训练CNN的相应操作（内容发表于[此处](#)）。我们的网络包含了一些新颖的特征，不仅提高了分类效果还减少了训练耗时，具体详见第三节。由于我们的模型过于庞大，即使有120万训练标签样本，还是会面临过拟合的问题，这在第四节中有具体讨论。我们最后训练出的网络包含五个卷积层和三个全连接层，这个深度设置非常关键：我们发现移除任何一个卷积层（虽然只包含不超过全局1%的参数）都会导致较劣的分类效果。

最后，神经网络的大小主要是受限于所使用的GPU的显存，以及我们所能接受的训练耗时。我们的CNN训练使用了两个GTX580 3GB GPU，总共耗时5至6天。实验结果还表明，只要有更快的GPU和更大的数据集可使用，模型的分类效果还可以提高。

2 数据集

ImageNet内含超过1500万张、共22000多类带标签的高分辨率图像。图像是从网络上收集并通过亚马逊土耳其机器人（[Mechanical Turk](#)）众包工具（crowd-sourcing）人工添加标签。ILSVRC - ImageNet Large-Scale Visual Recognition Challenge是一年一度的比赛，起始于2010年，属于Pascal Visual Object Challenge比赛的一部分。ILSVRC用ImageNet的部分数据集，包含100



知乎

ILSVRC-2010是所有ILSVRC系列比赛中唯一提供测试图像标签的比赛，所以我们大部分的实验是基于该数据集。我们还将模型应用于ILSVRC-2012，在第六节中我们会提供训练结果，不过该数据集没有提供测试图像标签。使用ImageNet训练模型，一般大家都会计算两个错误率，即前文提到的Top1错误率及Top5错误率，其中Top5错误率指的是测试图像中，未能被模型的前五个预测结果正确分类的图像占有所有图像的比例。

ImageNet包含各种清晰度的图像，但我们的模型需要固定的输入维度。因此我们将所有图像都进行下采样处理，统一调整为 256×256 规格。具体步骤是，对一张长方形图像，我们先将其短边长度缩小至256像素，然后截取图像中央 256×256 大小的部分作为最终使用图像。除此以外，我们还对图像的每一个像素进行**中心化处理**。

译者注：原文是*subtracting the mean activity over the training set from each pixel*，通过谷歌，觉得比较靠谱的解释是将训练集每个图像对应像素的R、G、B三个值分别求平均数，然后每个图像的每个像素的R值减去R平均，G值减去G平均，B值减去B平均。如有知友知道正确解释请在评论处指出，非常感谢。

所以神经网络训练使用的是图像像素中心化后的RGB值

3 架构

我们的网络架构总结在图2中。它包含了八个训练层 - 五个卷积层和三个全连接层。以下我将介绍我们的网络架构中所引入的新颖的特征。下文3.1-3.4的顺序依照其重要性排列，其中3.1最重要。

3.1 ReLU 非线性化神经元

神经元根据输入值 x 的输出函数 f 一般采用tanh作为激活函数，即 $f(x) = \tanh(x)$ 或者 $f(x) = (1 + e^{-x})^{-1}$ 。当使用梯度下降法寻找最优解时，这类**饱和**非线性激活函数，相比于**非饱和**非线性激活函数，耗费更长的训练时间。借鉴Nair和Hinton [20]的研究，我们采用ReLU非线性化神经元。采用ReLU的Deep-CNN比采用tanh的CNN训练起来快很多，如图一所示。我们比较了采用不同激活函数的四层神经网络对CIFAR-10数据集训练，记录分别达到25%训练错误率所需要的循环数。此图表明，如果使用传统的饱和激活函数神经网络模型的话，我们无法完成如此大型的神经网络训练项目。



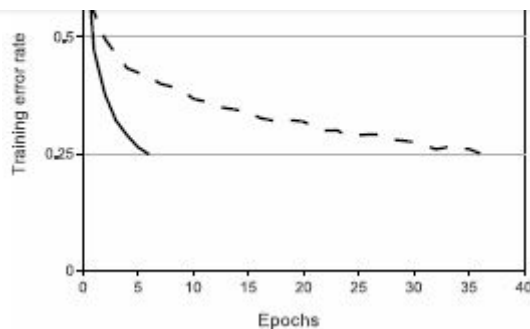


图1：采用ReLU的四层CNN（实线）对CIFAR-10数据集达到25%训练错误率的速度是采用tanh的CNN（虚线）的六倍。每个网络的学习率（learning rate）都是独立选取以使其训练速度最大化，且都没有经过正则化处理。当然，两种CNN的训练时间上的差距依不同架构会有所不同，但采用ReLU的CNN总是快过用饱和激活函数的CNN。

（译者注：epoch指所有训练数据完成一次前向和后向传递，与batch大小有关）

不过，我们也并不是最早想到使用非传统神经元模型的团队。Jarrett等人 [11] 曾提出将非线性激活函数 $f(x) = |\tanh(x)|$ 运用于他们所构建的模型对Caltech-101数据集的训练取得了良好的效果，该模型包含了对比度归一化层（contrast normalization）及局部平均汇合层（local average pooling）。但训练Caltech-101数据集主要关注的是过拟合问题，所以他们所谓的“良好效果”与我们采用ReLU提到的“训练速度提升效果”是有所区别的。更快的训练速度对大型数据集的模型训练效果有很大的影响。

3.2 使用多个GPU训练

单个GTX580 GPU只有3GB显存，限制了其所能训练的神经网络的最大规模，而且实践证明120万张训练图像所需要的神经网络对于单个GPU来说过于庞大。因此，我们将神经网络搭建于两个GPU上。我们使用的GPU特别适合并行作业，因为他们能够直接互相读取和写入显存而不需要经过主机内存。通过并行作业，我们在每个GPU上各搭载了一半的神经元，并且设置了一个机制，即两个GPU只在特定的层级中互相“交流”。举例说明，就是某层级接收前一层级所有神经元的输出结果，而另一层级只接收搭载于同一GPU上的上一层级神经元的输出结果。具体连接模式可以通过交叉验证来调整，但这一机制使得我们可以精确地调整连接数，使得其占总计算量的比例达到我们可以接受的程度。

最后我们搭建的架构有一些类似Ciresan等人[5]提出的“柱状”CNN，不过我们的CNN网络的columns之间是非独立的（见图2）。使用两个GPU的神经网络通过运用这一机制与使用单个GPU且卷积层神经元数减半的神经网络相比，Top1错误率及Top5错误率分别低了1.7%和1.5%。而双GPU模型比单GPU模型所用的训练时间还稍短一些。（单GPU模型和双GPU模型的神经元数量其实差不多，因为神经网络大部分的参数集中在第一个全连接层，其接收的是最后一个卷积层的输出结果。所以为了使两种模型具有大致相同数量的参数，我们没有将最后一个卷积层的规模减半，其后的全连接层也不用。这样的处理导致两者的分类效果对比其实是有利于单GPU模型的，因为神经元数量比双GPU模型的“一半”多一些。）

知乎

文中我推测是指单个GPU里的神经网络。而非独立就是指两个GPU上的网络之间是有连接层的。在文中引用的“柱状”CNN from reference[5] *High-Performance Neural Networks for Visual Object Classification*里没有直接提到independent column, 但估计是指其GPU的implementation是相互独立的。另外一个推测依据是后文对two-GPU和one-GPU的描述, 以及文中提到这是新颖的特征。

3.3 局部响应归一化 (Local Response Normalization)

ReLU有一个良好的性质, 就是它不需要对输入数据进行归一化以防止饱和。如果至少有一部分训练样本提供正的输入值给ReLU, 那么训练就会在该神经元内执行。不过, 我们还是发现下述局部归一化机制可以提高归纳能力。公式中 $a_{x,y}^i$ 表示神经元活动, 即输入数据的局部(x,y)经过卷积核 i 处理然后经过ReLU处理。响应归一化活动 $b_{x,y}^i$ 的计算公式为:

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

其中, 累加项汇总同一空间范围内 n 个“相邻”的卷积核, N 表示该层中的卷积核总数。卷积核map的顺序是随机的且在训练开始前就已定好。这类响应归一化其实是受到真实神经元侧抑制现象的启发在进行类似的操作。其效果就是使不同卷积核计算的神经元输出值之间对计算值比较大的神经元活动(big activities)更为敏感。

译者注1: lateral inhibition:相近的神经元彼此之间发生的抑制作用, 即在某个神经元受指刺激而产生兴奋时, 再刺激相近的神经元, 则后者所发生的兴奋对前者产生的抑制作用。

译者注2: Normalization是CNN里一个很重要的层, 尤其是运用ReLU的CNN, 因为其没有boundary。而LRN的一个优点就是文中提到的侧抑制。我找到一篇对LRN的解释比较易懂的文献。What Is Local Response Normalization In Convolutional Neural Networks。大致意思就是, 真实的神经元利用侧抑制这一性质可以强化局部对比度从而强化识别能力。

原文中使用的是competition for big activities among neuron outputs...

式中常数 k, n, α 以及 β 都是超参数, 用验证集调试得出; 我们的模型使用

$k = 2, n = 5, \alpha = 10^{-4}$ 以及 $\beta = 0.75$ 。我们只对某些层 (详见3.5小节) 在经过ReLU处理后进行上述归一化处理。

这个方法和Jarrett [11]等人使用的局部对比度归一化有一些类似, 但是我们的模型更符合“亮度归一化 (brightness normalization)”这一范畴, 因为我们没有减去均值。响应归一化使得我们模型的Top1错误率及Top5错误率分别降低了1.4%和1.2%。我们还用CIFAR-10数据集验证了响应归一化的效率: 四层CNN没有归一化层时错误率为13%, 而有归一化层时错误率为11%。(因为篇幅限制所以本文没有详细介绍该模型, 感兴趣的可以参考[这里](#))



CNN里的汇合层对同一卷积核map内邻近的神经元的输出值进行总结。传统方法里汇合层单元处理后的邻近神经元是不重叠的（比如[17,11,4]）。汇合层，具体来说，可以表示为一个网格里相互间隔s个像素的一个个汇合单元，每一个单元处理邻近的以该单元为中心，大小为 $z \times z$ 的神经元组。如果我们设置 $s = z$,就得到传统的CNN。如果 $s < z$ ，就得到我们模型使用的重叠汇合。我们的模型设置 $s = 2, z = 3$.如此设置使得我们模型的Top1错误率及Top5错误率相比没有使用重叠汇合的模型（设置 $s = 2, z = 2$ ）低了0.4%和0.3%，非重叠汇合模型得出的结果的维度和我们模型的一样。我们还在训练过程中发现使用重叠汇合的模型较不容易产生过拟合问题。

3.5 整体架构

现在，我们开始介绍CNN模型的整体架构。如图2所示，模型包含八个参数层；前五个是卷积层，后面三个是全连接层。最后一个全连接层的输出结果提供给1000-way softmax，并得出1000个分类标签的概率分布。我们的模型训练目标是最大化多元逻辑斯蒂函数，等价于最大化训练集预测分布下正确分类的log概率的均值。

译者注：上述原文是average across training cases of the log-probability of the correct label under the prediction distribution

用公式表示为
$$\operatorname{argmax}_w \left\{ \frac{1}{N} \sum -\log(p(f(x, w) = y(x))) \right\}$$

模型中第二、四和五卷积层的卷积核只接收位于同一GPU上前一层的卷积核输出结果。而第三卷积层的卷积核接收前一层所有的卷积核（两个GPU）。全连接层的神经元也是接收前一层所有的输出结果。第一和第二卷积层后面各附带一个响应归一层。3.4中介绍的汇合层位于每一个响应归一层之后以及第五个卷积层之后。每一个卷积层和全连接层的输出结果都会经过ReLU非线性化处理。

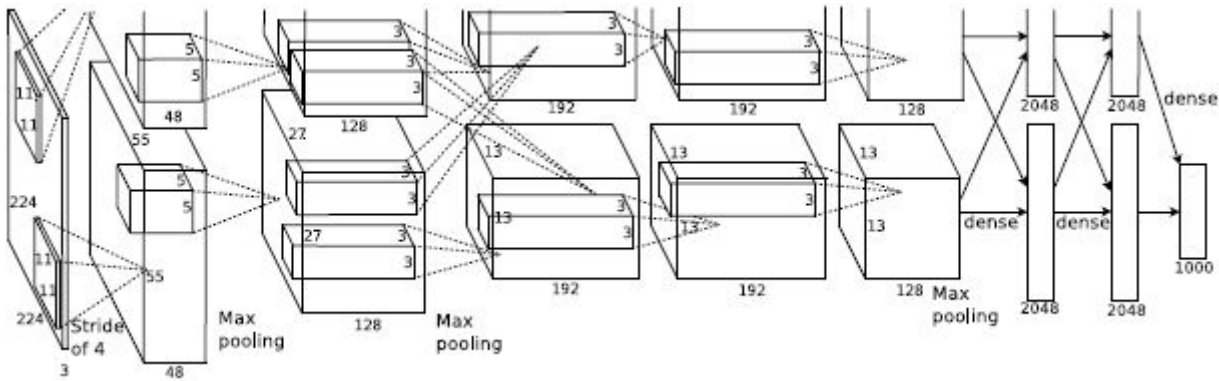


图2：我们的CNN模型的一个图像简述，直观地展示了两个GPU分别负责的任务。图中上下两个部分的卷积层各自搭载于一块GPU上。两个GPU的网络只在部分层次中有交互。神经网络的输入层维数是150,528，网络中剩下的层级的神经元数量分别是253,440、186,624、64,896、64,896、43264、4096、4096、1000。

第一个卷积层用96个大小为 $11 \times 11 \times 3$ 的卷积核过滤维度为 $224 \times 224 \times 3$ 的输入图像，步长为4像素（步长是指卷积核接收的相邻神经元组感受野中心之间的距离，也就是卷积核每次移动的像素数，用于决定输出结果的尺寸）。第一个卷积层的输出结果经过响应归一以及汇合

知乎

连接第二卷积层的输出结果（归一+汇合处理后）。第四卷积层使用384个卷积核，大小为 $3 \times 3 \times 192$ ，第五卷积层使用256个卷积核，大小为 $3 \times 3 \times 192$ 。每一个全连接层都有4096个神经元。

卷积核大小的概念可以参考[CS231n课程笔记翻译：卷积神经网络笔记](#)

4 降低过拟合

我们的神经网络总共有6000万个参数。虽然ILSVRC的数据集仅有1000个标签类别使得每一个样本对其到所属标签的映射施加了10bits的限制，但即便如此，训练含有如此多参数的模型时，还是会出现较严重的过拟合现象。

译者注：对“10 bits的限制”的理解，我搜到了一篇[Modelling the Manifolds of Images of Handwritten Digits](#)，里面有提到类似的话语。

It is possible to fit far more parameters before overfitting occurs because the input vectors contain much more information than the class label. Assuming the same number of examples in each class, a class label only contains $\log_2 10$ bits so each example only provides 3.3 bits of constraint on the function that maps inputs to class labels¹

. However, it takes many more bits to specify the input image, so each example provides far more constraint on the parameters.

输入数据的维度很复杂，其包含的信息很多，而输出结果，即分类标签是很简单的语句，如此结构的样本可以训练包含更多的参数的模型而不产生过拟合。

4.1 数据增量 (Data Augmentation)

最简单也最常见的减少过拟合的方法就是通过**保留标签转换**人为地扩大数据集（例如[25,4,5]）。我们运用两种数据增量方式，计算量都很小，所以转换得到的新图像不用存在硬盘中。我们的转换操作是在CPU上用python实现的，而GPU专门用于训练模型。所以实际训练中，数据增量操作对我们的CNN训练的总计算量没有影响。

第一种数据增量方式是图像变换和水平翻转。具体操作是从原本大小为 256×256 的图像中提取所有大小为 224×224 的子图像（以及他们的水平翻转图像），然后将这些子图像作为我们CNN的输入图像。（这解释了为什么图二中我们模型的输入层的大小是 $224 \times 224 \times 3$ ）。经过如此操作，我们的训练数据集变为了原来的2048倍（ $(256 - 224)^2 \times 2$ ）。虽然扩大后的数据之间的相关性非常大，但如果不这样操作，我们的网络会出现严重的过拟合现象，可能会迫使我们使用规模更小的网络。在测试的时候，模型对每个输入图像提取五个 224×224 子图像（四个角落和中心）以及他们分别的水平翻转图像（总共10个），通过softmax层进行预测，并将10个预测值平均。



知乎

3) 一个随机量的乘积。也就是对于某图像的每一个像素 $I_{xy} = [I_{xy}^R, I_{xy}^G, I_{xy}^B]$ 加上以下算式的结果：

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

其中 \mathbf{p}_i 和 λ_i 是图像RGB值计算的 3×3 协方差矩阵的第 i 个特征向量和特征值，而 α_i 就是前面提到的随机量，服从均值为0，标准差为0.1的正态分布。随机产生的一组 α_i 将用于某张图的所有像素，直到该图再次被训练时才会重新产生新的。这一调整是为了突出自然图像的一个重要性质，就是对物体图像的识别不应该受到其表面色彩的强度和颜色的影响。通过该操作，我们CNN的Top1错误率降低了1个百分点。

4.2 随机失活（指在模型训练时随机让网络某些隐含层节点的权重不工作）

结合多个不同模型的预测结果可以降低测试错误率 [1,3]，但对于本身就需要数天时间训练的大型神经网络而言，这是很奢侈的。然而，还是有很高效的方法能够结合模型的预测结果，而且只耗费大约两倍的训练时间。其中一种最新的方法叫“随机失活” [10]，将隐含层神经元的输出结果依0.5概率设置为0。被随机失活“脱离”的神经元因此不提供有效输出予前向信息传递也不参与后向传播。因此，每一次训练一个图像时，神经网络就会随机生成一个新的架构，但这些架构中使用的权重是一样的。通过随机失活减少了神经元之间复杂的互相适应性（co-adaptation），因为通过随机失活，神经元无法过分依赖于某个有输出结果的前一神经元（译者注：因为没有输出结果的神经元可能是因为被随机“失活”了，而不是因为其对输入特征解释能力不佳）。在随机神经元组的配合下，这个神经元也因此被迫去学习更加鲁棒且有用的特征。在测试时，我们使用所有的神经元，将他们的输出结果乘以0.5，这其实是由极多的经过随机失活的神经网络产生的平均分类结果的一个合理近似值。

我们在图二中的前两个全连接层运用随机失活。否则，神经网络训练就会出现很严重的过拟合。但随机失活几乎使得模型收敛所需要的循环翻倍。

5 训练细节

我们用随机梯度下降来训练模型，每一个批量有128个样本，动量为0.9，权值衰减为0.0005。我们发现小权值衰减对模型的训练是很重要的。也就是说，权值衰减在模型中不单单起到正则化作用；它还协助降低模型的训练错误率。权重的更新方法如下：

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

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



所有目标函数在 w_i 处对权重的偏导数的均值。

我们将每一个层级的权重初始化为均值0，标准差0.01的正态随机量。第二、四核五卷积层以及全连接层的偏差系数（bias）设置为1。这样可以在训练初期给ReLU单元提供正的输入值，从而加快训练速度。其他层级的偏差系数初始设为0。

所有的层级我们都使用相同的学习率，具体数值是我们在训练过程中不断调整得出的。主要调整方法是每当模型在当前的学习率下验证错误率不再降低时，我们就把学习率除以10。初始学习率是0.01，在完成训练过程中总共减少了三次。我们对120万张图像训练了约90个周期(cycle，也可称为epoch)，使用两块NVIDIA GTX 580 3GB GPU，总共花费5-6天。

6 训练结果

模型对ILSVRC-2010的训练结果总结在表1。Top1和Top5错误率分别为**37.5%**和**17%**（若没有按4.1介绍的对10个预测值取平均的话，错误率是39%和18.3%）。而当时比赛最佳结果是47.1%和28.2%，是由六个稀疏编码(sparse coding)模型的预测值取平均得到，且所用的训练特征都不同[2]。比赛后至今最佳结果是45.7%和25.7%，是两个分类器的预测值的平均值，分类器的训练基于由不同类型的密集取样特征计算出的Fisher Vector（FV）[24]。

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

表1：ILSVRC-2010结果对比，斜体数字是其他模型取得的最佳结果。

我们还将模型应用在ILSVRC-2012比赛上，并将训练结果列于表2。由于ILSVRC-2012没有提供测试集的标签，我们无法计算测试错误率。在本段落接下来的篇幅内，验证错误率和测试错误率代表同一个东西，因为表2显示它们的差距不超过0.1%。我们的CNN的Top5错误率是18.2%。五个结构类似的CNN（5 CNNs）的预测结果取均值得到16.4%的错误率。另外图二第四个CNN模型，事先训练过整个ImageNet 2011秋季 数据集（1500万张图像共22000种类别），其内部位于最后一个汇合层之后还有第六个卷积层。该模型经过“微调”后，用于训练ILSVRC-2012数据集得到的错误率是16.6%。将事先训练2011秋季 数据集的两个CNN和前述的5 CNNs模型的预测值取平均，得到错误率为**15.3%**。比赛中亚军预测结果错误率是26.2%，其模型是数个分类器的预测值的平均值，分类器的训练基于由不同类型的密集取样特征计算出的Fisher Vector（FV）。

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
<i>SIFT + FVs [7]</i>	—	—	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%

最后，我们还对ImageNet 2009秋季 数据集进行训练，该数据集包含890万张图像共10184个标签类别。对该数据集，我们依照以往研究的规矩，将其对半分，一半是训练集，一半是测试集。因为没有给定的测试集，所以我们的分法与其他作者的分法肯定有区别，但还好这并不影响结果。我们最终得到Top1和Top5错误率分别为**67.4%**和**40.9%**，但使用的模型是前述CNN的基础上再最后一个汇合层后增加了第六卷积层。目前为止，对该数据集的最佳预测错误率是78.1%和60.9%[19]。

6.1 量化评估

图3展示了神经网络两个数据连接层（输入层和第一卷积层）训练的卷积核的输出结果。神经网络训练出了丰富的卷积核 - 频率选择型，方向选择型（frequency- and orientation-selective kernels）以及多种色块（colored blobs）。另外，也可以看出两块GPU的明确分工，这是通过3.5节中介绍的**部分连接型**实现的。显卡一上的卷积核基本上是不识别颜色的，而显卡二上的卷积核大部分是能识别颜色的。这样的分工效果在每一次训练中都会出现，而且与任一权重的随机初始化是不相关的（GPU的重编号）

译者注：括号内的文字原文是：modulo a renumbering of the GPUs。不知道是啥意思，估计是GPU的内部操作，希望有识之士给予帮助。



图3：第一个卷积层的96个 $11 \times 11 \times 3$ 卷积核对输入的 $224 \times 224 \times 3$ 图像训练后的输出结果。前48个和后48个运算分别在两个GPU上进行。

下图4中左半部分是我们模型对八幅测试图像计算的前五个分类，作为我们评估模型训练效果的定性分析基础。值得一提的是，部分不在图像中心的物体，比如左上角的小虫子都可以被模型准确辨识。而且大部分预测都是比较靠谱的。比如，对猎豹的前五个分类中，除了第一个正确分类外，其他分类结果都至少将其归类为猫科动物。不过有些情况下（比如汽车和樱桃），图像内容和真实标签的对应本身就有很大的歧义。



知乎



图4：（左边）八幅ILSVRC-2010测试图像以及我们的模型对其做出的前五个分类。图片的正确标签列于图片下方，而正确分类的概率也用红色条带标识（如果前五个分类结果包含正确标签）（右边）第一列是五张ILSVRC-2010测试图像。剩下的六列图像是训练集图像中通过模型最后的隐含层（也就是输出层的前一层）后得出的特征向量与第一列测试图像的相应的特征向量距离（欧式距离）最近的图像。

另一个探索神经网络视觉识别能力的方法是研究图像在最后一个层级，即维度为4096的隐含层上产生的特征激活状态（feature activation）（译者注：其实就是通过最后一个隐含层的输出结果）。如果两个图像的特征激活状态向量之间的欧式距离比较小，那么就代表神经网络内部较高层次认为这两张图是类似的。图4右半部分展示了五张ILSVRC-2010测试图像以及分别计算出六张最为接近的训练集图像（译者注：类似从训练集中找出第一列测试集图像的6-Nearest-Neighbors，不过是基于最后一个隐含层的输出结果）。可以观察到，在像素层面上，返回的几张训练图像其实与第一列的测试图像的L2距离不是特别接近。例如，第二行和第五行用大象和狗的测试图像返回的训练图像里姿势多种多样。我们在补充材料里提供了更多的测试图像与训练图像匹配结果。

计算4096维实向量之间的欧式距离是很低效的，但可以通过训练一个自动编码器将向量压缩成较短的二进制码，从而提高效率。对这些向量进行计算返回的训练集图像比直接将自动编码器运用在输入图形上返回的结果要好得多[14]，因为直接计算像素而不使用其标签会使得计算偏向于在图像边缘寻找模式的相似性，不管它们实际图片内容上是否相似。

7 讨论

我们的结果显示大型深度CNN可以在比较困难的数据集上单纯使用监督学习方法取得突破性进展。值得一提的是若移除我们模型中的任何一个卷积层，训练效果都会大打折扣。例如，去除任何一个中间层都会导致Top1分类结果损失增大约2%。所以我们设置的深度对于取得优良的结果是非常关键的。

为了简化我们的实验，我们没有进行非监督的预训练即使我们觉得这会有所帮助，尤其是当我们获得足够的计算能力得以增大神经网络的规模，却没有相应的更大的带标签图像集。目前为止，我们的训练结果通过不断扩大神经网络以及不断增加训练时间而有所优化。但是，如果想要接近人视觉系统的infero-temporal路径还是有很长的路要走。我们的最终目标是用非常大且深的神经

知乎

译者注：关于infero-temporal，根据后面提到的人类视觉系统推测是视觉系统中的一部分生理结构，在谷歌上只能搜到inferior temporal。另外这个地方也提到It is crucial for visual object recognition

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. IMAGENET.ORG. 2010.
- [3] L. Breiman. Random forests. Machine learning, 45(1):5–32, 2001.
- [4] D. Cireşan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745, 2012.
- [5] D.C. Cireşan, 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 ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012).
- [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 authors.library.caltech.edu....
- [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.



知乎

- [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



知乎

[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

译者反馈

- 1. 第一篇翻译文章，希望大家批评指导；
- 2. 感谢知友[杜客](#)、[申尚昆](#)的校对和点评。

编辑于 2017-07-26

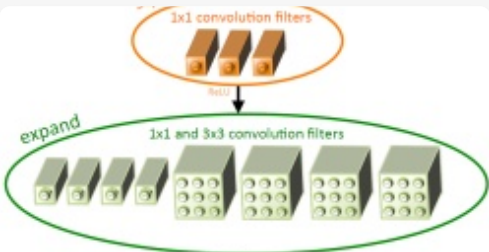
机器学习

推荐阅读

CNN-反卷积（1）

本文收录在无痛的机器学习第一季。关于CNN的内容已经说了很多，虽然我们无法把这个长得像黑盒的东西完全摸清楚，但是我们多多少少也对它的外部结构有了一定的了解。下面我们来看看大神们…

冯超 发表于无痛的机器…



CNN模型之SqueezeNet

我是小将



一种新的CNN网络可地区分自然图像&生成

AI科技大... 发表于

34 条评论

切换为时间排序

写下你的评论...

 章子誉

翻译了alexnet那篇？

2 年前



知乎



杜客

2 年前

1.dropout这个名词我们当时也是反复讨论，最后是翻译成了“随机失活”。随机是对应了他的工作原理，失活是和activation（激活）对应；

赞



杜客

2 年前

2.pooling我们是采用了周志华教授《机器学习》里面的说法，翻译成了采样或者汇合层；

赞



杜客

2 年前

3.Mechanical Turk个人是翻译成土耳其机器人，具体的典故在知乎上有篇《人肉计算》的文章中有，供参考。

赞



杜客

2 年前

4.Data Augmentation个人建议是可以用数据增加或者增量？因为Data Augmentation实际上要做的事情就是通过一些翻转之类的手段让训练数据能够增加，比如一张猫的图，我把它左右翻转后，就等于增加了一张猫的图片，所以个人觉得增量或者增加比较合适。

赞



杜客

2 年前

目前的校对建议就是这些，后续我再对照原文仔细校对，建议仅供参考，还是以你自己的理解为主。nice work!

赞



卓柳舟 (作者) 回复 杜客

2 年前

很感谢，我翻译的时候也是对这些名词比较头疼，大部分都是百度的。

赞 查看对话



卓柳舟 (作者) 回复 章子誉

2 年前

应该是

赞 查看对话



polossk

2 年前

平行作业……这个明显是 并行作业 吧

赞



知乎

👍 赞 💬 查看对话



KAKAC

2 年前

关于“subtracting the mean activity over the training set from each pixel”的翻译我认为正确的。有两点证据可以支持这种理解。

1. [Caffe | ImageNet tutorial](#)在Compute Image Mean部分中提到，一个重要的预处理手段就是subtract the image mean from each image，这个处理方法与楼主所理解的处理方法是一样的。
2. [caffe/compute_image_mean.cpp · GitHub](#)中代码所实现的功能正是楼主所理解的预处理操作。

👍 1



卓柳舟 (作者) 回复 KAKAC

2 年前

感谢！

👍 赞 💬 查看对话



柒柚秋

1 年前

做毕设导师让我翻译这个文章，结果发现了现成的哈哈。

👍 赞



洛必达

1 年前

同毕设翻译。。。自己纠结于各个名词解释，很多概念也不知道，翻不下去了，来向大神学习下，顺便从这里也去看了解释CNN的那篇翻译！点赞！太棒啦！

👍 赞



夏晓玲

1 年前

“subtracting the mean activity over the training set from each pixel”

深度学习Imagenet caffe AlexNet 实验步骤在这篇博客里操作是：

(2) 减去像素平均值：所有图片的每个像素都减去所有训练集图片的平均值。训练集图片的平均值存储于data/ilsrv12/imagenet_mean.binaryproto。如果没有该文件，执行./examples/imagenet/make_imagenet_mean.sh可以生成该文件。

我也不知道他的这个文件和生成是怎么做的，因为我没用caffe打算用keras复现。如果只是减去当前图片的均值好像更容易实现一点。

👍 赞



陈钊2048

1 年前

3.2节里面“具体连接模式可以通过交叉验证来调整”的原文是” Choosing the pattern of



知乎

👍 赞



卓柳舟 (作者) 回复 陈钊2048

1 年前

我自己理解：从意义上看，CV确实是用来调节参数的，所以说选择哪种方式交给CV来处理是比较合理的，你提供的版本意思感觉不是很顺，感觉像是若选择了某种方式可能CV做不下去了；另外也搜了一下，A is a problem for B，我看一般意思都是说A是B需要考虑的问题。

👍 赞

💬 查看对话



陈钊2048 回复 卓柳舟 (作者)

1 年前

我重复看了几遍，的确是啊。这里应该说明的是怎么选择连接方式就不关我什么事了，这个是xxx的问题（浓浓的甩手掌柜气息，大雾）。这样的话，翻译成“而选择哪种连接方式就是交叉验证需要解决的问题”，意译和直译结合，信达雅

👍 赞

💬 查看对话



Refine 回复 杜客

1 年前

数据扩增会不会更好，译成动词比名词要好些，数据增加的话又有些直白。

👍 赞

💬 查看对话



何夕

1 年前

第三卷积层使用384个卷积核，大小为3X3X256 这里是不是应该是3X3X128

👍 赞

1 2 下一页

