|  |
| --- |
| **论文名称：** **ImageNet Classification with Deep Convolutional Neural Networks** **使用深度卷积神经网络的 ImageNet 分类** **作者： Alex Krizhevsky Ilya Sutskever** |
| **按照论文顺序展开**  **Abstract**  AlexNet是大型的**深度卷积神经网络**（具有 6000 万个参数和 650,000 个神经元），包含卷积层、最大池化层、全连接层、softmax。在相关比赛（**ILSVRC-2010/2012**）中成绩优异。作者使用了非饱和神经元（non-saturating neurons）、卷积操作的高效GPU实现（effificient GPU implemenation of the convolution operation）、dropout等（减少全连接层的过拟合） 操作来优化网络。  **Introduction**  物体在现实环境中可能表现出相当大的变化性，所以要学会识别它们，就必须使用更大的训练集。（LabelMe ImageNet）  CNNs在处理大规模的图像识别任务的优势：   * 极强的学习能力 * 有先验知识：引入了很多图像处理的先验知识 * 能力可控：可通过改变其网络的广度和深度来控制 * 结构优势：统计的稳定性和像素依赖的局部性（见第五周学习中的解释） * 更少的连接和参数，最佳性能相差不多：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   虽然在处理高分辨率图像时CNN的应用成本很昂贵，但GPU可加速大型CNN的训练以及大型数据集足够多的标记样本使得训练这些网络成为可能，且不易过拟合。  本文的主要贡献是：  一个大型的、性能优异的神经网络，该网络在ImageNet数据集上取得了卓越的结果，并公开了其高度优化的2D卷积GPU实现；  减少训练时间并提高性能（第三节）和防止过拟合（第四节）的新技术；  深度对于网络性能的重要性。  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。硬件配置：两个 GTX 580 3GB GPU ，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。  **The Dataset**  ILSVRC-2010为主要实验版本：   ILSVRC使用的是ImageNet的一个子集，有1000个类别，每个类别中大约有1000个图像。总共有大约120万张训练图像，50,000张验证图像和150,000张测试图像。  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百分率。  为了适应需要固定输入尺寸的系统，ImageNet数据库中的图像被下采样到256×256的分辨率。这一过程包括将矩形图像缩放至短边为256像素，然后裁剪中心的256×256区域。在训练网络时，除了减去像素的平均值，没有对图像进行其他预处理，直接使用图像的原始RGB值。  **The Architecture**  包含八个学习层——五个卷积层和三个全连接层（按照重要性递减排布）  **ReLU非线性单元**   * 非饱和非线性函数：f(x)=max(0,x)的梯度下降法训练时间比tanh和sigmoid的时间快得多。——修正非线性单元（Rectified Linear Units） * 该图展示了对于一个特定的四层CNN，在CIFAR-10数据集训练中的误差率达到25%所需要的迭代次数。使用ReLUs做为激活函数的卷积神经网络比起使用tanh单元作为激活函数的训练起来快了好几倍。 * 实线为ReLUs，虚线为tanh单元 * 并不是第一个替换传统神经元模型的：f(x)=|tanh(x)|在对比度归一化上效果很好 * 在这个数据集上重点是防止过拟合，所以和上面这种方法无法对比加速能力，不过加快训练速度对大型数据集上训练的大型模型的性能有很大的影响。   **在多个GPU上训练**   * 内存限制：提到的 GTX 580 GPU 拥有 3GB 的内存，这限制了可以在单个GPU上训练的神经网络的大小。 * 训练样本数量：作者指出，他们有 120万个训练样本，这些网络因为规模太大而不能在单个GPU上训练。 * 多GPU并行化：为了解决单个GPU内存不足的问题，作者将神经网络分布在两个GPU 上进行训练。现代GPU设计得非常适合于跨GPU并行化操作，因为它们可以直接读写对方的内存，无需通过主机内存，这大大提高了并行计算的效率。直接的GPU到GPU的通信方式被称为 GPU对等访问（GPU Peer-to-Peer Access）或 直接内存访问（Direct Memory Access, DMA） * 并行化策略：作者采用的并行化方案是将网络的一半内核（或神经元）放在一个GPU上，另一半放在另一个GPU上。目的是为了充分利用每个GPU的计算资源，同时避免单个GPU内存不足的问题。此外，他们使用了一个技巧，即GPU只在某些网络层之间进行通信。例如，第3层的内核会从第2层的所有内核映射中获取输入，而第4层的内核则只从位于同一GPU上的第3层内核映射中获取输入。      * 连接模式和交叉验证：选择哪些层应该在GPU之间进行通信是一个重要的决策，因为它影响交叉验证的过程。交叉验证是一种评估模型泛化能力的技术。通过调整连接模式，精确控制通信量，直到计算量达到可接受的水平。      * 这种方法的效果自然是有进步的，top-1和top-5错误率分别降低了1.7％和1.2％。 * 在最终的卷积层中，单GPU网络和双GPU网络有相同数量的卷积核，为了让两种网络配置具有大致相同的参数数量，作者没有减少最终卷积层及其后全连接层的大小。 * 单GPU网络的最终卷积层没有被缩小，这使得单GPU网络在参数数量上比双GPU网络的一半还要多，从而可能在性能比较中更有利。但在最后的方法效果对比中双GPU并行化结果优于单GPU——这更能说明这项方法的正确性   **局部响应归一化（Local Response Normalization）**   * ReLU具有理想的属性，它们不需要对输入进行归一化来防止它们饱和。如果至少有一些训练实例为ReLU产生了正的输入，那么这个神经元就会学习。ReLU不饱和性与稀疏激活的特性 * 局部响应归一化有助于泛化(现在不如BN) * 常数 k、n、 α和β都是超参数 * 论文：k = 2, n = 5, α= 10^−4 , and β= 0.75 * LRN对局部神经元的活动创建竞争机制，使得其中响应比较大的值变得相对更大，并抑制其他反馈较小的神经元，增强了模型的泛化能力。 * top-1和top-5的错误率分别降低了1.4％和1.2％，LRN是否使用在CIFAR-10数据集上让测试错误率降低了2%（基础模型为四层CNN）   **重叠池化**   * 池化层的窗口之间有重叠的部分。这与传统的池化层不同，在传统的池化层中，每次移动的步长等于池化窗口的大小，因此不会有重叠。 * k为stride kernel size:n\*n 论文中使用的为 2 3\*3 * top-1和top-5的错误率分别降低了0.4％和0.3％   **整体结构**     * 上面这个采用双GPU，下面的图是后来单GPU      * 前五个是卷积层，其余三个为全连接层 * 输入层：输入为224×224×3 三通道的图像，实际大小为227\*227 * C1的基本结构为：卷积-->ReLU-->（LRN）-->池化   卷积：输入227 × 227 × 3的图像，96个kernel size为11×11×3的filter，不扩充边缘padding = 0，步长stride = 4，因此其FeatureMap大小为(227-11+0×2+4)/4 = 55，即55×55×96;  激活函数：ReLU；  池化：池化核大小3 × 3，不扩充边缘padding = 0，步长stride=2因此其FeatureMap输出大小(55-3+0×2+2)/2=27, 即C1输出为27×27×96（此处未将输出分到两个GPU中，若按照论文将分成两组，每组为27×27×48）；   * C2的基本结构为：卷积-->ReLU-->（LRN）-->池化 * 卷积：输入27×27×96，256个kernel size为5×5×96的filter，扩充边缘padding = 2， 步长stride = 1，因此其FeatureMap大小为(27-5+2×2+1)/1 = 27，即27×27×256; * 激活函数：ReLU； * 池化：池化核大小3 × 3，不扩充边缘padding = 0，步长stride = 2，因此其FeatureMap输出大小为(27-3+0+2)/2=13, 即C2输出为13×13×256（此处未将输出分到两个GPU中，若按照论文将分成两组，每组为13×13×128）； * C3的基本结构为：卷积–>ReLU。   卷积：输入13×13×256，384个kernel size为3×3×256的filter，扩充边缘padding = 1，步长stride=1，因此其FeatureMap大小为(13-3+1×2+1)/1 = 13，即13×13×384;  激活函数：ReLU，即C3输出为13×13×384（此处未将输出分到两个GPU中，若按照论文将分成两组，每组为13×13×192）；   * C4的基本结构为：卷积–>ReLU。   卷积：输入13×13×384，384个kernel size为3×3×384的filter， 扩充边缘padding = 1，步长stride = 1，因此其FeatureMap大小为(13-3+1×2+1)/1 = 13，即13×13×384;  激活函数：ReLU，即C4输出为13×13×384（此处未将输出分到两个GPU中，若按照论文将分成两组，每组为13×13×192）；   * C5的基本结构为：卷积–>ReLU–>池化   卷积：输入13×13×384，256个kernel size为3×3×384的filter，扩充边缘padding = 1，步长stride = 1，因此其FeatureMap大小为(13-3+1×2+1)/1 = 13，即13×13×256;  激活函数：ReLU；  池化：池化核大小3 × 3， 扩充边缘padding = 0，步长stride = 2，因此其FeatureMap输出大小为(13-3+0×2+2)/2=6, 即C5输出为6×6×256（此处未将输出分到两个GPU中，若按照论文将分成两组，每组为6×6×128）；   * FC6的基本结构为：全连接–>>ReLU–>Dropout   全连接：此层的全连接实际上是通过卷积进行的，输入6×6×256，4096个kernel size为6×6×256的filter，扩充边缘padding = 0, 步长stride=1, 因此其FeatureMap大小为(6-6+0×2+1)/1 = 1，即1×1×4096;  激活函数：ReLU；  Dropout：全连接层中去掉了一些神经节点，达到防止过拟合，FC6输出为1×1×4096；   * FC7的基本结构为：全连接–>>ReLU–>Dropout   全连接：此层的全连接，输入1×1×4096;  激活函数：ReLU；  Dropout：全连接层中去掉了一些神经节点，达到防止过拟合，FC7输出为1×1×4096；   * FC8的基本结构为：全连接–>>softmax   全连接：此层的全连接，输入1×1×4096;  softmax：softmax为1000，FC8输出为1×1×1000；   * LRN一般是在激活、池化后进行的一种处理方法。   **Reducing Overfitting**  **数据增强**   * 平移图像和水平映射   从256×256图像中随机提取224×224的图像块（及其水平映射）  水平映射是一种将图像中的像素坐标沿水平方向移动的变换。具体来说，它将图像中的每个像素的坐标分别加上指定的水平偏移量，从而改变像素的位置。（由此产生的训练样本当然还是高度相互依赖的）  在测试时，网络通过提取5个224×224的图像块（四个角块和中心块）以及它们的水平映射（因此总共包括10个块）来进行预测，并求网络的softmax层的上的十个预测结果的均值。   * 改变训练图像中RGB通道的灰度（PCA）      * top-1 error rate 降低 1%   **Dropout层**   * 将隐含层的神经元输出置为0，以这种方法被置0的神经元不参与网络的前馈和反向传播 * nn.Dropout(0.5)——50%将隐含层的神经元输出置为0 * 迭代次数增加 * 每次给网络提供了输入后，神经网络都会采用一个不同的结构，但是这些结构都共享权重 * 神经元无法依赖其它特定的神经元而存在，减少了神经元的复杂适应性 * 它被迫学习更强大更鲁棒的功能，使得这些神经元可以与其他神经元的许多不同的随机子集结合使用。 * 效果：试着使用了所有的神经元，并将它们的输出乘以0.5。（激活强度是平时的一半）这与采用大量dropout的网络产生的预测结果分布的几何均值近似   **Details of Learning**   * 随机梯度下降法，每个batch有128个样本，动量（momentum）为0.9，权重衰减（weight decay）为0.0005 * 较小的权重衰减对于模型的训练很重要（正则化方法，减少训练误差） * i表示当前的迭代次数，v表示动量（momentum），ε表示学习率，最后的是第 i批次的目标函数关于w的导数（wi的偏导数）Di的平均值。 * 标准差为0.01、均值为0的高斯分布来初始化各层的权重 * 使用常数1来初始化了网络中的第二个、第四个和第五个卷积层以及全连接层中的隐含层中的所有偏置参数，使用常数0来初始化剩余层中的偏置参数 * 在训练神经网络时，作者对所有层使用了相同的初始学习率0.01，并在训练过程中手动调整了学习率。他们采用了一种启发式的方法来调整学习率：当验证集上的错误率不再下降时，他们就会将学习率除以10。在整个训练过程中，学习率总共减少了三次。      * 使用120万张图像的训练集对网络进行了大约90次迭代的训练，这在两块NVIDIA GTX 580 3GB GPU上花费了大约5到6天的时间。   **Results**    * 与2010年比赛期间冠军模型（第一行）和赛后公布的冠军模型（第二行）相比，作者的网络的top-1和top-5测试集错误率分别为37.5％和17.0％。 * 2012年的模型未公布测试集的正确标签，作者将验证集和测试集的错误率互换，因为根据作者的经验，它们之间的差值不超过0.1％     本文描述的CNN（第二行）  对五个相似CNN的预测结果计算均值（第三行）  单独一个CNN，在最后一个池化层之后，额外添加第六个卷积层，对整个ImageNet Fall 2011 release(15M images, 22K categories)进行分类，然后在ILSVRC-2012上“微调”（fine-tuning）网络 （第四行）  对整个ImageNet Fall 2011版本的数据集下预训练的两个CNN，求他们输出的预测值与前面提到的5个不同的CNN输出的预测值的均值（第五行）   * 作者在ImageNet Fall 2009版本的数据集上提交了错误率，总共有10,184个类别和890万张图像 * top-1和top-5错误率分别是67.4％和40.9％ * 上述错误率是通过前面描述的网络获得的，但是在最后的池化层上还有额外的第6个卷积层。该数据集此前公布的最佳结果是78.1％和60.9％ * 定性评估（Qualitative Evaluations）      * 图三是由网络的两个数据连接层学习得到的卷积内核 * 该网络已经学习到许多频率和方向提取的内核，以及各种色块。 * GPU1为上四十八个图像，GPU2为下四十八个图像 * GPU1上的内核在很大程度上与颜色无关，然而GPU2上的内核在很大程度上都于颜色有关。 * 即使在不同的训练迭代中使用不同的随机权重初始化，或者即使改变GPU的编号（也就是说，改变它们的工作顺序或者分配的任务），网络中的两个GPU还是会展现出相同的特异性。 * 网络的结构使得每个GPU在训练过程中都会自动地学习到专门化的特征提取方式。      * 在图4的左边，我们通过计算8张测试图像的top-5预测来定性评估网络的训练结果。 * 前六张都成功识别，即便是第一张图片中左下角的螨虫都被正确地识别出来。 * 第七张图片中樱桃识别成斑点狗——文中表示是照片的关注点存在模糊性。 * 第八张图片文中没有解释。 * 在图4的右边，考虑由最后一个4096维隐含层中的图像的特征的激活函数输出值。最左边的一列是2010中的测试图像，剩下六列显示了6个训练图像，它们在最后一个隐藏层中产生特征向量，与测试图像的特征向量的欧氏距离最小。 * 每幅图像都被转换成一个4096维的实值向量，然后计算这些向量之间的欧氏距离来判断图像的相似性。 * 如果两幅图像在高层次的神经网络中产生的向量之间的欧氏距离较小，那么网络认为这两幅图像是相似的。 * 这种方法不仅仅考虑像素层次上的相似性，而是更深层次的特征相似性，这些特征可能包括图像中的模式和边缘等。 * 通过使用欧式距离来计算两个4096维实值向量的相似性，效率不高，但是通过训练自编码器可以将这些向量压缩为较短的二进制码，能够使其更高效。 * 这种方法不依赖于图像标签（无监督学习），因此它更适合于检索在视觉特征上相似的图像，而不是仅仅在语义上相似的图像。   **Discussion**   * 大型深度卷积神经网络 纯有监督学习 * 如果移除一个卷积层，我们的网络性能会降低 深度对于实现作者的结果非常重要 * 作者要将网络在视频序列上实践   **Reference**   * 略 |
| **内容概要：AlexNet是一个深度卷积神经网络（CNN）架构，由Alex Krizhevsky、Ilya Sutskever和Geoffrey Hinton共同设计。它在2012年的ImageNet大规模视觉识别挑战赛中取得了突破性的成绩，大幅度降低了Top-5错误率。**  **总结：AlexNet通过增加网络深度和使用ReLU激活函数，显著提高了图像分类任务的准确性。它的成功证明了深度学习在视觉识别任务中的潜力**  **主要解决问题/应用：AlexNet主要解决的是大规模图像分类问题。它的应用包括但不限于图像识别、物体检测和其他需要视觉理解的任务。**  **主要使用方法/模型：AlexNet模型包含五个卷积层和三个全连接层，使用了ReLU激活函数、局部响应归一化、最大池化和Dropout技术。模型被设计为可以在两个GPU上并行运行。**  **主要实验手段/数据集：AlexNet在ImageNet数据集上进行训练和测试。**  **评判式思考： 虽然AlexNet取得了显著的成绩，但它也存在一些局限性，如对大量计算资源的需求和过拟合的风险。此外，模型的泛化能力和对新领域的适应性也是值得探讨的问题。**  **创造性思考：AlexNet的成功激发了对深度学习模型的进一步探索，促进了更深、更复杂网络架构的发展。它也推动了GPU在科学计算中的广泛应用，以及对模型优化和正则化技术的创新。** |