使用综合卷积神经网络对大规模图像进行识别

白稹 青海民族大学 [galijiangzhi@163.com](mailto:galijiangzhi@163.com)

# 摘要

在这项工作中，我主要整合了自2010年以来，对大规模图像识别任务具有重大提升的卷积神经网络模型，主要包括在ImageNet大规模视觉挑战识别赛中取得优秀成绩的AlexNet，vgg，GoogLeNet，ResNet。这些网络结构从各个方面解决了卷积神经网络参数数量多，模型特征提取结果表现差，模型网络退化等问题，我通过参考这些模型的论文，设计了一个综合这些模型的优点的网络，其中训练结果最好的一个版本被我命名为PhantomNet，这个网络的深度在我做完实验后我会给出。

# 1.引言

自2012年AlexNet卷积神经网络问世以来，使用卷积神经网络进行图像识别的效果在以惊人的速度进步，这一现象不仅仅是因为计算机性能的提升，更是因为深度学习技术的迅速发展和广泛应用。卷积神经网络作为一种经典的深度学习模型，在图像识别领域展现出了强大的特征提取和分类能力，使得其成为了图像识别领域的主流技术。除了AlexNet之外，Vgg、GoogLeNet、ResNet等一系列卷积神经网络模型的提出和不断改进也为图像识别的进步贡献良多。举例来说：在2014年，vgg团队提出了“使用多个小卷积核堆叠替代大卷积核”的思路，减少了大感受野带来的参数量暴涨问题，同年，GoogLeNet首次提出的inception结构解决了不同大小卷积核对特征提取不全面的问题，同时，该团队根据新加坡国立大学发表的“network in network”论文中的方法，在网络中加入1\*1的卷积减少网络参数的数量，使得网络大小较vgg的网络缩小了十二倍，而且更加精确。

值得注意的是，这些网络都有很好的效果，但是这些网络之间同样也缺少了一些关联性，比如在ResNet中，团队没有考虑使用多个尺寸的卷积核并行提取特征，而是选择在保留当前特征图的同时，对当前特征图进行优化，将优化前后的特征图进行相加输出，又或者，在以优化算法效率为目标之一的GoogLeNet中，团队也没有考虑使用多个小卷积核来代替大卷积核。因此，我萌生了创造一个综合这些主流网络优点的网络的想法。

在本文中我将重点讲述一个集多种网络结构与一身的高效深度卷积神经网络，相比经典的卷积神经网络模型，该模型的参数量更少，特征提取手段更多，同时也在一定程度上解决了网络退化的问题，这得益于将残差模块和inception模块进行结合。同时我也很重视模型的效率，我认为这是非常重要的，就像GoogLeNet中提到的，模型的价值是投入实际应用，而不是成为学术奇观。

# 2.相关工作

手机屏幕的截图

中度可信度描述已自动生成**小卷积核叠加：**在卷积神经网络结构中，小卷积核叠加是一种通过叠加小卷积核以提高感受野的的方法，图1展示了使用两个3x3卷积核与一个5x5卷积核在感受野上的区别，使用多个较小的3x3卷积核堆叠来模拟大卷积核的效果在vgg网络中被证明是可行的，同时根据”参数数量=输入维度x卷积核面积x输出维度”公式可以算出，这种方法在可以显著减少参数数量。

**图1.使用多个3x3卷积核在特征提取上与5x5卷积核的区别**

图示, 工程绘图

描述已自动生成

**网络中的网络：**该方法是由新加坡国立大学研究人员提出的一种用于提高神经网络表征能力的方法，原文中指出，通过在每次卷积之后添加线性层，以增加模型的非线性，图2展示了该方法的实现过程，在实际应用中，可以通过添加1x1卷积层来实现。我在模型中大量使用了这种结构。除了增加模型的非线性外，使用该结构还有降维的作用，通过降维可以减少模型的参数和计算量，为后续增加模型的宽度和深度做铺垫。

**图2.通过在卷积层后添加线性层增加非线性**

**图示

描述已自动生成Inception结构：**Inception 架构的主要思想是找出如何在卷积视觉网络中逼近和覆盖一个最优的局部稀疏结构，GoogLeNet团队在这个问题上使用的实现方式是通过多个不同尺寸的卷积核对图像进行特征提取，同时在特征提取之前增加1x1卷积核以减少参数数量和计算量，希望在能捕获到更多的特征的同时减少计算量，团队把这个捕获特征的结构成为inception结构，我根据小卷积核叠加理论对该结构进行了改进（如图3所示），使用多个小卷积核叠加来代替不同尺寸的卷积核，已达到进一步减少参数数量的目的。

**图3.inception结构**

残差学习

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Type** | **Patch size/**  **stride** | **Output**  **size** | **#1x1** | **#3x3**  **reduce** | **#3x3** | **#3x3->3x3**  **reduce** | **#3x3->3x3** | **Pool**  **proj** | **Params** |
| Convolution | 1x1/1 | 224x224x3 |  |  |  |  |  |  | 0.03K |
| Max pool | 3x3/2 | 112x112x3 |  |  |  |  |  |  |  |
| Convolution(2a) | 3x3/2 | 56x56x32 |  |  |  |  |  |  | 0.85k |
| Convolution(2b) | 3x3/2 | 56x56x32 |  |  |  |  |  |  | 0.85k |
| Convolution(2b) | 3x3/1 | 56x56x32 |  |  |  |  |  |  | 9k |
| 将2a和2b两条线的输出叠加在一起，输出出56\*56\*64 | | | | | | | | | |
| Max poll | 3x3/2 | 28x28x64 |  |  |  |  |  |  |  |
| Inception(3) |  | 28x28x128 | 32 | 48 | 64 | 8 | 16/16 | 16 | 46k |
|  |  |  |  |  |  | 0 |  |  |  |