

经典卷积神经网络总结

本文总阅读量83次

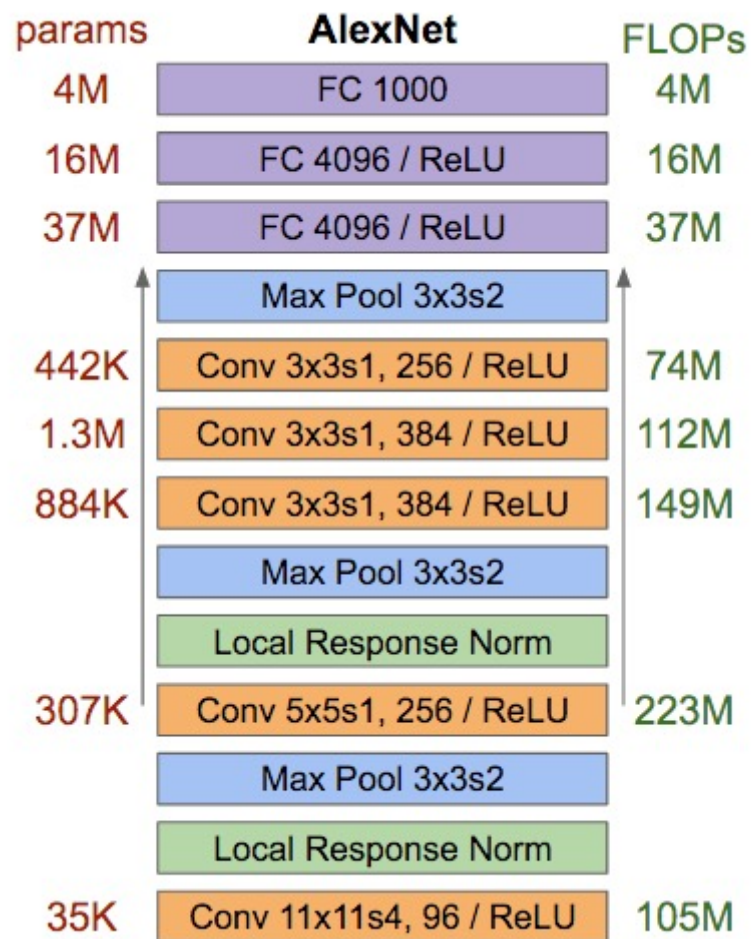
2017-04-18

一 AlexNet

2012年由Hinton的学生Alex Krizhevsky提出。以Top-5的错误率为16.4%赢得ILSVRC 2012年的比赛。它做出了如下创新：

- 首次使用ReLU作为CNN激活函数，解决了Sigmoid激活函数的梯度弥散问题。
- 使用Dropout随机丢弃部分神经元，可以避免模型的过拟合。AlexNet的最后几个全连接层使用了Dropout
- 使用重叠的最大池化，之前使用的都是平均池化。最大池化可以避免平均池化的模糊效果。同时，步长比卷积核的尺寸小，这样池化层的输出之间会有重叠，提升了特征的丰富性。
- 提出了LRN层（局部相应一体化），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较大，并抑制其他反馈较小的神经元，增强模型泛化能力。
- 使用CUDA加速深度卷积网络的训练，使用了GPU的并行计算能力。
- 数据增强，随机从256x256的原始图中截取224x224大小的区域，再做水平翻转，相当于增加了 $(256 - 224)^2 \times 2 = 2048$ 倍的数据量。仅靠原始的数据量，参数众多的CNN会陷入过拟合。预测时，取图片四个角和中间共5个位置，再加上翻转，共10个位置，对它们的预测结果求均值。

https://shartoo.github.io/classic_cnn/



二 VGGNet

VGGNet是牛津大学计算机视觉组(Visual Geometry Group)和Google DeepMind一起研发的深度卷积神经网络。通过反复堆叠3x3的小型卷积核和2x2的最大池化层，VGG成功的构筑了16-19层卷积神经网络。获得 ILSVRC 2014分类项目第二名和定位项目第一名。整个网络使用了同样大小的卷积核尺寸(3x3)和最大池化尺寸(2x2)。

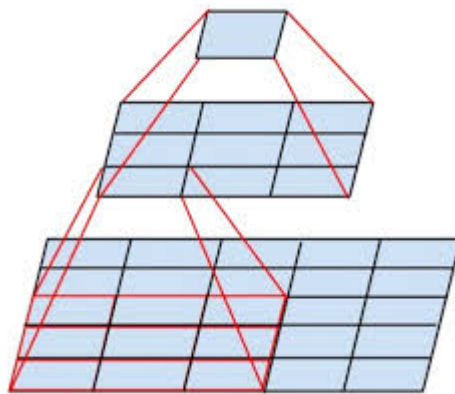
网络结构和参数

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

The 6 different architectures of VGG Net. Configuration D produced the best results

虽然从A到E每一级网络逐渐变深，但是网络参数数量没有太多增长，因为参数数量主要消耗在最后的三个全连接。前面卷积虽然很深，但是消耗的参数量不大，不过训练时比较耗时的还是卷积部分。上图中的D和E就是VGGNet-16和VGGNet-19。

技巧 网络中经常出现的多个完全一样的3x3的卷积串联相当于1个5x5的卷积层，即一个像素跟周围5x5像素产生关联，可以感受野大小为5x5。而3个3x3卷积串联效果等同于1个7x7的卷积层。同时3个串联的3x3的卷积拥有比1个7x7的卷积更少的参数量，只有后者的 $3 \times 3 \times 3 / 7 \times 7 = 55\%$ 。同时3个3x3的卷积层拥有比一个7x7的卷积层更多的非线性变换（前者使用了3次ReLU激活，而后者只使用了一次），使得CNN对特征的学习能力更强。



训练技巧，先训练级别A的简单网络再复用A网络的权重来初始化后面的几个复杂网络，这样收敛速度更快。 **预测时**：VGG采用Multi-scale方法，将图像scale到一个尺寸Q，并将图片输入卷积网络计算，再将不同尺寸的Q的结果平均得到最后结果，这样可以提高图片数的利用率并提升预测准确率。

结论

- LRN层作用不大
- 越深的网络效果越好
- 1x1的卷积也是有效的，但是没有3x3的卷积好，大的卷积核可以学习更大的空间特征。

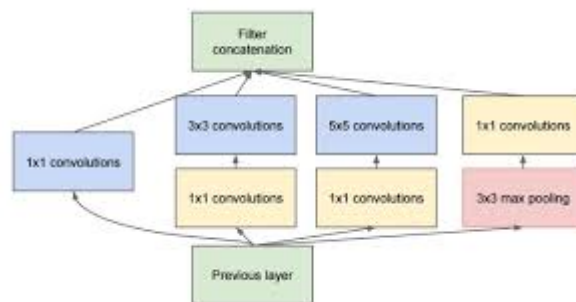
三 Google Inception Net

2014年ILSVRC 冠军，最大的特点是控制住计算量和参数量的同时，获得了很好的分类性能，top-5错误率 6.7%。Inception v1有22层，比AlexNet的8层或VGGNet的19层要深，但是 计算量只有15亿次，500万参数量，仅为AlexNet的1/12(6000万)。

特点

- 去除了最后的全连接层，用全局平均池化层（将图片尺寸变为 1×1 ）来取代它。全连接层占据了AlexNet或VGGNet的90%的参数量，而且会引起过拟合。用全局平均池化层取代全连接层的做法借鉴了Network in Network。
- 其精心设计的Inception Module 提高了参数的利用率。一般来说，卷积层要提升表达能力，主要依靠增加输出通道数，但是副作用是计算量增大和过拟合。每个输出通道对应一个滤波器，同一个滤波器共享参数，只能提取一类特征，因此一个输出通道只能做一种特征处理。而NIN中的MLPConv则拥有更强大的能力，允许在输出通道之间组合信息。MLPConv基本等效于普通卷积层后再连接 1×1 的卷积和ReLU激活函数。

Inception Module的基本结构如下：



其中的 1×1 卷积可以以很小的计算量就增加一层特征变换和非线性化，它可以跨通道组织信息，提高网络的表达能力，同时对输出通道升维和降维。

人脑神经元是稀疏激活的，模拟的神经网络也是类似。应该把相关性高的一簇神经元节点连接在一起。图片数据中，临近区域的数据相关性高，因此相邻像素点被卷积操作连接在一起。因此，一个 1×1 的卷积就可以自然地把这些相关性很高的、在同一空间位置但是不同通道的特征连接在一起，这就是为什么 1×1 的卷积反复被应用在Inception Net中的原因。

Inception Net是一个大家族，包括了以下系列。

网络	年代	错误率	创新
Inception V1	2014年9月	Top-5 6.67%	使用了Network In Network的思想
Inception V2	2015年2月	Top-5 4.8%	(1)使用了两个3x3来替代5x5的大卷积。(2)提出了Batch Normalization，正则化方法，加速大型卷积神经网络的训练速度，同时提升收敛后的分类准确率。
Inception V3	2015年12月	Top-5 3.5%	(1)引入Factorization into small convolutions，将较大卷积拆分为两个小卷积，比如7x7拆成1x7和7x1卷积，节省了大量参数，加速运算同时减轻过拟合，同时增加一层非线性拓展模型表达能力 (2)在Inception V3中使用了分枝，还在分枝之中使用了分枝
Inception V4	2016年2月	Top-5 3.08%	结合了微软的ResNet

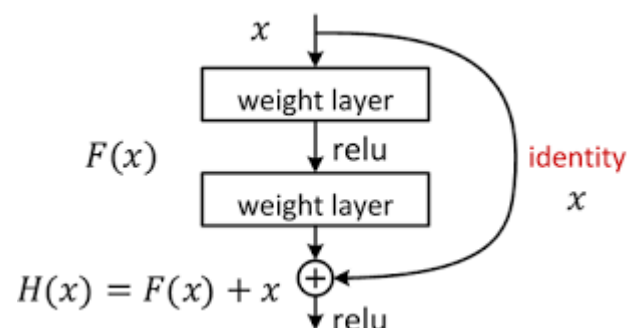
四 ResNet

ResNet(Residual Neural Network)由微软研究院Kaiming He等四位华人提出，通过使用Residual Unit成功训练152层深的神经网络，在ILSVRC 2015比赛中获得冠军，获得3.57%的Top-5准确率，同时参数量比VGGNet低。

ResNet源于**Highway Network**，通常认为神经网络的深度对齐性能非常重要，但是网络越深其训练难度越大，Highway Network的目标就是解决极深网络的难以训练的问题。

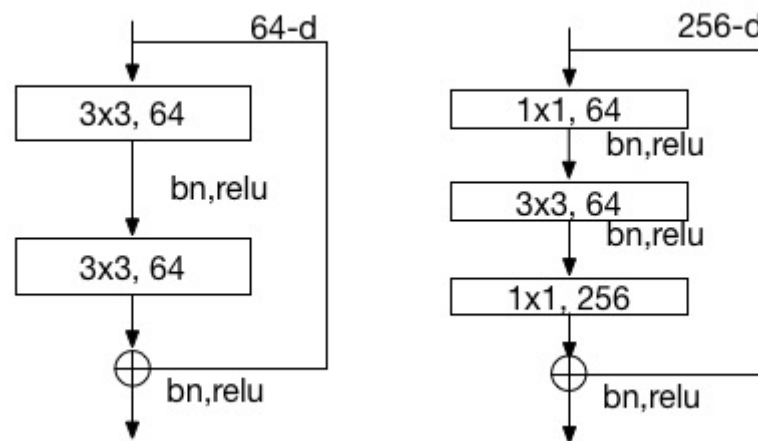
Highway Network相当于修改了每一层的激活函数，此前的激活函数只是对输入做一个非线性变换 $y = H(x, W_H)$ ，Highway Network则允许保留一定比例的原始输入 x ，即 $y = H(x, W_H)T(x, W_T) + xC(x, W_C)$ ，其中T为变换系数，C为保留系数。论文中令 $C = 1 - T$ 。这样前面一层的信息，有一定比率可以不经过矩阵乘法和非线性变换，直接传输到下一层，仿佛一条高速公路。

假定某段神经网络的输入是 x ,期望输出是 $H(x)$,如果直接把输入 x 传都输出作为初始结果,那么此时我们需要学习的目标就是 $F(x) = H(x) - x$ 。如下图所示,这就是一个ResNet的残差学习单元(Residual Unit),ResNet相当于将学习目标改变了,不再是学习一个完整的输出 $H(x)$,只是输出和输入的差别 $H(x) - x$ 即残差。



传统卷积层或全连接层在信息传递时,或多或少会存在信息丢失、损耗等问题。ResNet在某种程度上解决了这个问题,通过直接将输入信息绕道到输出,保护信息的完整性,整个网络则只需要学习输入、输出差别的那一部分,简化学习目标和难度。

下图是两层或三层的ResNet残差学习模块。



下图是ResNet不同层数时的网络配置

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	$7 \times 7, 64, \text{stride } 2$				
conv2_x	56×56	$3 \times 3 \text{ max pool, stride } 2$				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9



撰写评论

使用社交网站账户登录

或使用来必力便捷评论

邮件

写评论

总评论数 0

按时间正序

还没有评论，快来抢沙发吧！

来必力是？ | 询问

