1. RCNN是目前目标检测技术的最新发展

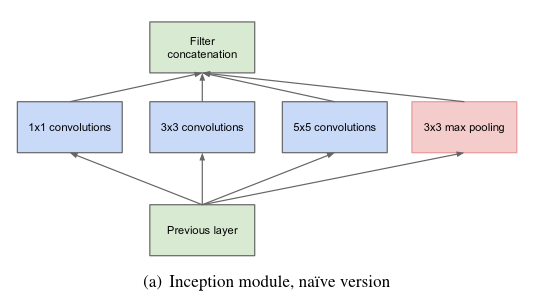
RCNN将整体检测问题划分为两个子问题：

1. 利用低级线索，如颜色、纹理，以一种与类别无关的方式生成对象定位建议
2. 使用CNN分类器在这些位置识别对象类别
3. 增加模型深度的两个缺点：
4. 更大的尺寸通常意味着更多的参数，这使得扩大的网络更容易过度拟合
5. 网络大小均匀增加的另一个缺点是计算资源的使用量急剧增加

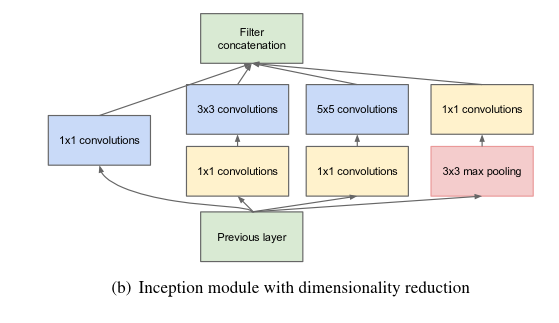
引入稀疏性解决这两个不足，用稀疏层代替完全连接的层，主要思想如下：如果数据集的概率分布可以用一个大的、非常稀疏的深度神经网络来表示，那么通过分析前一层激活的相关统计量，并将输出高度相关的神经元聚类，可以构建一个又一个层次的最优拓扑网络。（**先看最后，5.关于Inception模型的优越性**）

1. 深层次网络结构容易出现梯度弥散，模型性能下降。
2. Inception module（V1）

http://lib.csdn.net/article/aimachinelearning/66253



Inception module 的提出主要考虑多个不同 size 的卷积核能够增强网络的适应力，paper 中分别使用1\*1、3\*3、5\*5卷积核，同时加入3\*3 max pooling。  
随后文章指出这种 naive 结构存在着问题：每一层 Inception module 的 filters 参数量为所有分支上的总数和，多层 Inception 最终将导致 model 的参数数量庞大，对计算资源有更大的依赖。



Inception Module 的4个分支在最后通过一个聚合操作合并（在输出通道数这个维度上聚合，在 TensorFlow 中使用 tf.concat(3, [], []) 函数可以实现合并）。

完整的 GoogLeNet 结构在传统的卷积层和池化层后面引入了 Inception 结构，对比 AlexNet 虽然网络层数增加，但是参数数量减少的原因是绝大部分的参数集中在全连接层，最终取得了 ImageNet 上 6.67% 的成绩。

Inception model的其它版本：（附上下载链接）

1. Inception[**V1**]: [Going Deeper with Convolutions](https://arxiv.org/abs/1409.4842)

2. Inception[**V2**]: [Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift](https://arxiv.org/abs/1502.03167)

3. Inception[**V3**]: [Rethinking the Inception Architecture for Computer Vision](https://arxiv.org/abs/1512.00567)

4. Inception[**V4**]: [Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning](https://arxiv.org/abs/1602.07261)

Inception V1——构建了1x1、3x3、5x5的 conv 和3x3的 pooling 的分支网络，同时使用 MLPConv 和全局平均池化，扩宽卷积层网络宽度，增加了网络对尺度的适应性；

Inception V2——提出了 Batch Normalization，代替 Dropout 和 LRN，其正则化的效果让大型卷积网络的训练速度加快很多倍，同时收敛后的分类准确率也可以得到大幅提高，同时学习 VGG 使用两个3´3的卷积核代替5´5的卷积核，在降低参数量同时提高网络学习能力；

Inception V3——引入了 Factorization，将一个较大的二维卷积拆成两个较小的一维卷积，比如将3´3卷积拆成1´3卷积和3´1卷积，一方面节约了大量参数，加速运算并减轻了过拟合，同时增加了一层非线性扩展模型表达能力，除了在 Inception Module 中使用分支，还在分支中使用了分支（Network In Network In Network）；

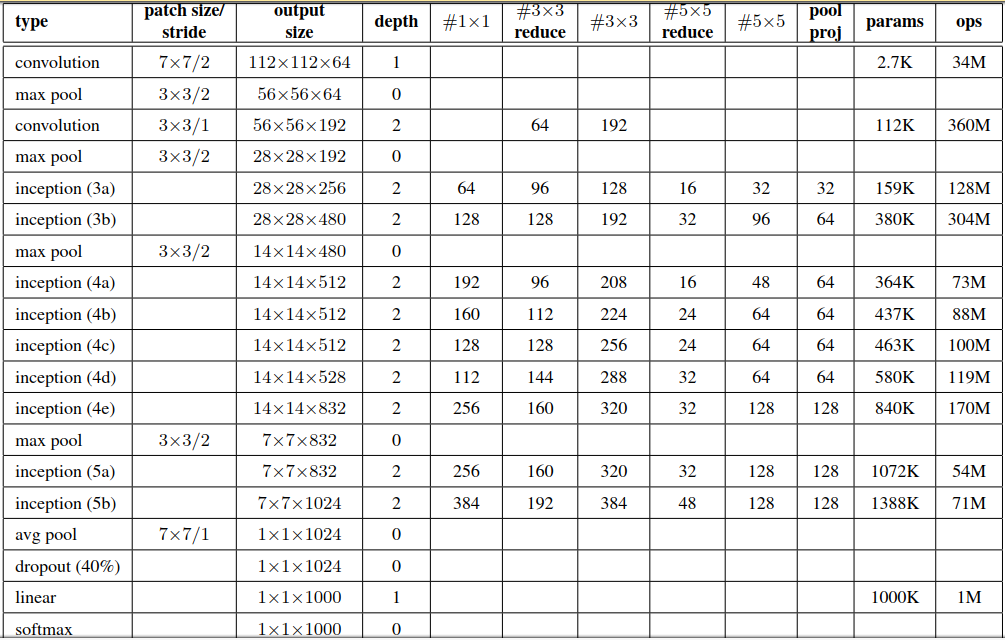
Inception V4——研究了 Inception Module 结合 Residual Connection，结合 ResNet 可以极大地加速训练，同时极大提升性能，在构建 Inception-ResNet 网络同时，还设计了一个更深更优化的 Inception v4 模型，能达到相媲美的性能。

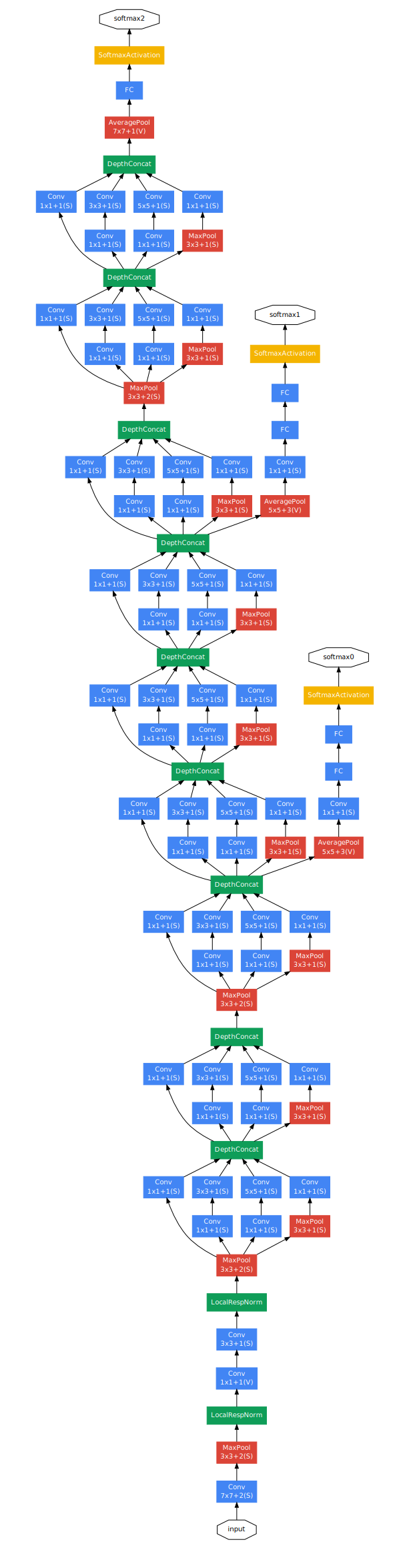
1. GoogleNet模型

原始图像是，224 \* 224 的RGB图像

“#3×3 reduce” and “#5×5 reduce” stands for the number of 1×1 ﬁlters in the reduction layer used before the 3×3 and 5×5 convolutions

倒数第二层（linear）：使我们能够很容易地使我们的网络适应其它的标签集合，主要为了方便使用，不希望其产生主要影响



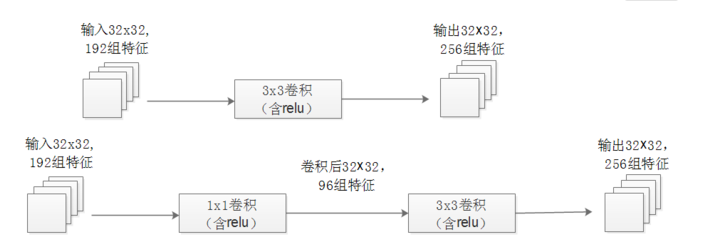


1. 关于Inception模型的优越性
2. **1x1卷积 ：**可以看到图1中有多个黄色的1x1卷积模块，这样的有什么用处呢？

**作用1：**在相同尺寸的感受野中叠加更多的卷积，能提取到更丰富的特征。这个观点来自于Network in Network(NIN, https://arxiv.org/pdf/1312.4400.pdf)

**作用2：**使用1x1卷积进行降维，降低了计算复杂度。

下图是优化前后两种方案的乘法次数比较，同样是输入一组有192个特征、32x32大小，输出256组特征的数据，第一张图直接用3x3卷积实现，需要192x256x3x3x32x32=452984832次乘法；第二张图先用1x1的卷积降到96个特征，再用3x3卷积恢复出256组特征，需要192x96x1x1x32x32+96x256x3x3x32x32=245366784次乘法，使用1x1卷积降维的方法节省了一半的计算量。有人会问，用1x1卷积降到96个特征后特征数不就减少了么，会影响最后训练的效果么？答案是否定的，只要最后输出的特征数不变（256组），中间的降维类似于压缩的效果，并不影响最终训练的结果。



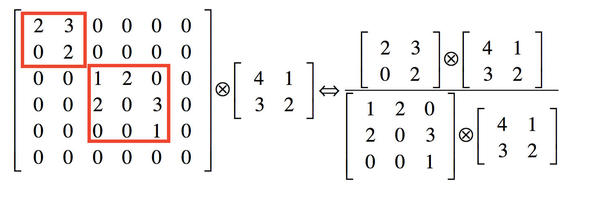
1. **多个尺寸上进行卷积再聚合**

对输入做了4个分支，分别用不同尺寸的filter进行卷积或池化，最后再在特征维度上拼接到一起。这种全新的结构有什么好处呢？

**解释1：**在直观感觉上在多个尺度上同时进行卷积，能提取到不同尺度的特征。特征更为丰富也意味着最后分类判断时更加准确。

**解释2：**利用稀疏矩阵分解成密集矩阵计算的原理来加快收敛速度。

举个例子下图左侧是个稀疏矩阵（很多元素都为0，不均匀分布在矩阵中），和一个2x2的矩阵进行卷积，需要对稀疏矩阵中的每一个元素进行计算；如果像右图那样把稀疏矩阵分解成2个子密集矩阵，再和2x2矩阵进行卷积，稀疏矩阵中0较多的区域就可以不用计算，计算量就大大降低



**解释3：（选择性了解）**Hebbin赫布原理。Hebbin原理是神经科学上的一个理论，解释了在学习的过程中脑中的神经元所发生的变化，用一句话概括就是fire togethter, wire together。赫布认为“两个神经元或者神经元系统，如果总是同时兴奋，就会形成一种‘组合’，其中一个神经元的兴奋会促进另一个的兴奋”。比如狗看到肉会流口水，反复刺激后，脑中识别肉的神经元会和掌管唾液分泌的神经元会相互促进，“缠绕”在一起，以后再看到肉就会更快流出口水。用在inception结构中就是要把相关性强的特征汇聚到一起。这有点类似上面的解释2，把1x1，3x3，5x5的特征分开。因为训练收敛的最终目的就是要提取出独立的特征，所以预先把相关性强的特征汇聚，就能起到加速收敛的作用。