**背景**

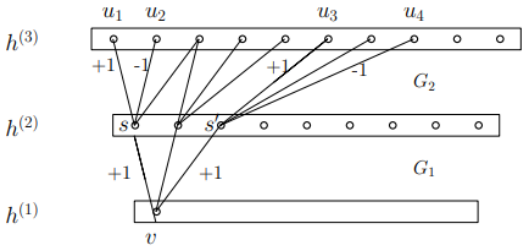
常用更深的网络提升准确率，但是这会导致网络需要更新的参数爆炸式增长，导致两个严重的问题：

①网络更容易过拟合，尤其是应用较小的数据集，过拟合更容易发生，于是我们需要大量的数据，但是制作数据集的时间成本较大；

②需要更新的参数数量大就会导致需要大量的计算资源，费用高昂；

③由于数据稀疏以及网络结构利用不充分（很多权重接近 0），都会导致大量计算的浪费。

解决以上问题的方法就是**把全连接的网络变为稀疏连接**（卷积层其实就是一个稀疏连接），“Provable Bounds for Learning Some Deep Representations”这篇论文里提出，如果可以用一个稀疏且大的深度神经网络将数据的分布表达出来，那么搭建这个网络的最佳方式是，通过分析前层神经元激活值的相关性，并将高度相关的神经元聚类连接到一起，输出到下一层（过程示意图如下所示），即 Hebbian 原则（neurons that fire together, wire together）




而 CPU 和 GPU **硬件更适合稠密矩阵的运算**，所以接下来就思考如何在保证网络稀疏性的同时还可以利用硬件设备加速计算？

“On two-dimensional sparse matrix partitioning:Models, methods, and a recipe.”提出聚类稀疏矩阵得到相关的稠密子矩阵可以加速稀疏矩阵的运算。

**由此就提出了 Inception 的结构，旨在解决如何利用稠密子模块来近似稀疏的网络结构**

补充说明：

什么是稀疏性？

当整个特征空间是非线性甚至不连续时：

学好局部空间的特征集更能提升性能，类似于 Maxout 网络中使用多个局部线性函数的组合来拟合非线性函数的思想；

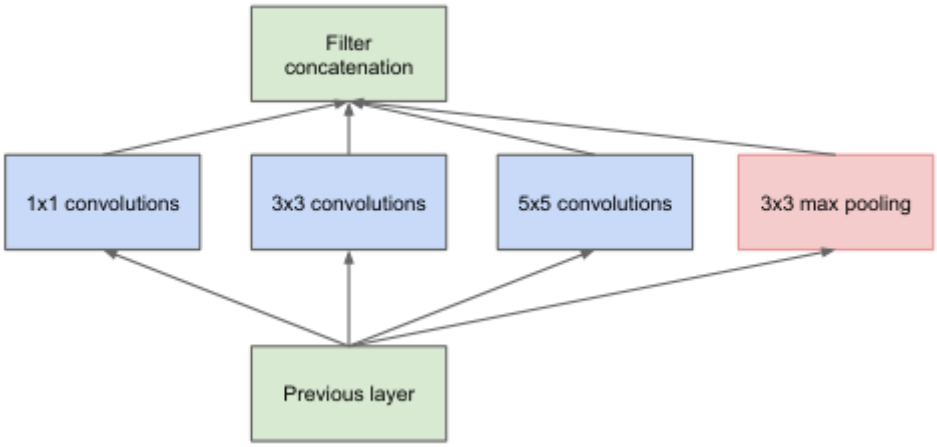
假设整个特征空间由 N 个不连续局部特征空间集合组成，任意一个样本会被映射到这 N 个空间中并激活/不激活相应特征维度，如果用 C1 表示某类样本被激活的特征维度集合，用 C2 表示另一类样本的特征维度集合，当数据量不够大时，要想增加特征区分度并很好的区分两类样本，就要降低 C1 和 C2 的重合度（比如可用 Jaccard 距离衡量），即缩小 C1 和 C2 的大小，意味着相应的特征维度集会变稀疏。（参考）

方法

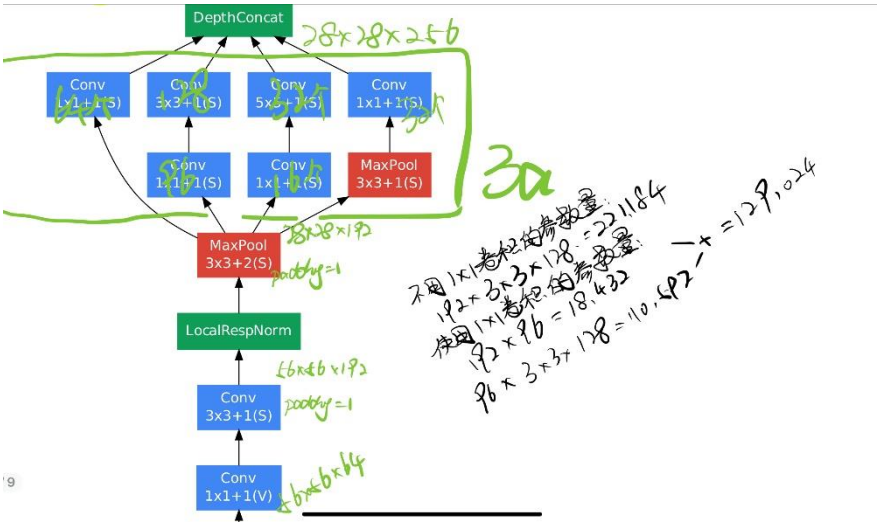
①出于方便，使用 1×1、3×3、5×5 的卷积得到稀疏的特征，然后通过 DepthConcat 将这些稀疏特征融合起来得到相对稠密的特征表达；

②为了在训练过程中高效率使用内存，增加了最大池化操作；

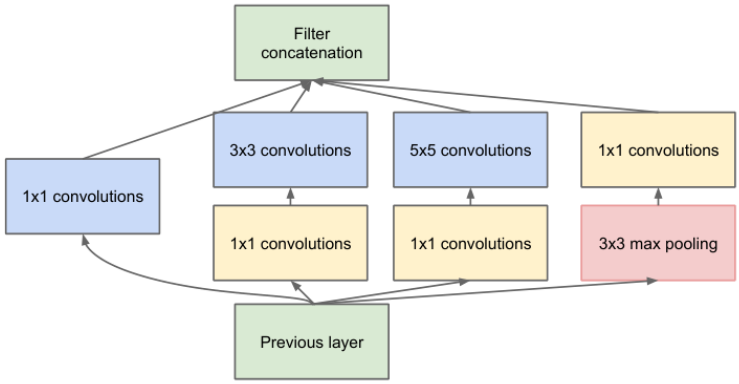
有了以下最初版本的 Inception 结构



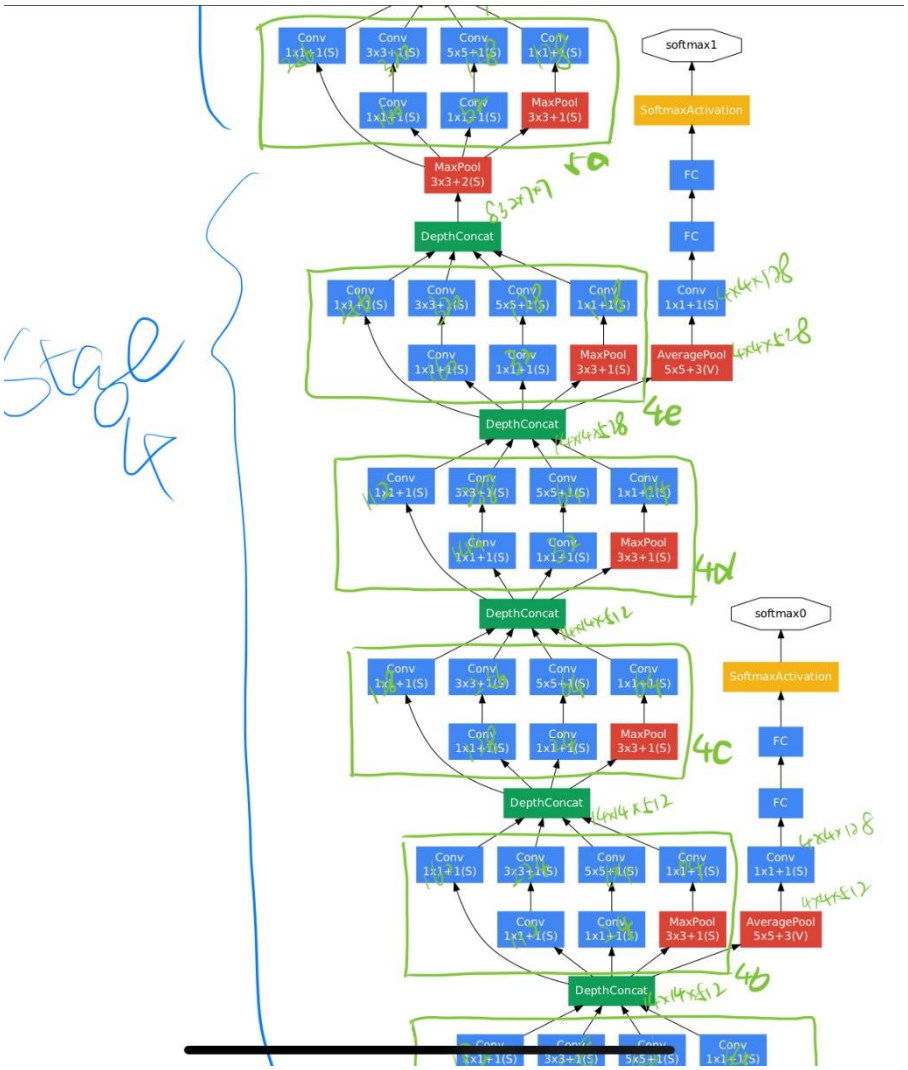

③然后又考虑到参数量的问题，借鉴 Network in Network 的思想，又在 3×3 和 5×5 卷积前加了 1×1 卷积降维，降维后还增加了 ReLU 激活函数。




有了以下版本的 Inception 结构



④考虑到这么深的网络在反向传播时的效率问题，增加了一个辅助分类器。因为相对较浅的层有较强的分类效果，训练阶段通过对 Inception(4b、4e)增加两个额外的分类器来增强反向传播时的梯度信号，避免梯度消失。




**总结**

①在分析每一层通道数的发现在每个 Inception 结构的 1×1 降维操作时，通道数减少的太多了，个人认为会造成很多信息的丢失，所以在后续可以考虑在降维时不要一次减少这么多；

②网络的深度很深，后续可以考虑将深度折合到宽度上，将网络的深度和宽度同时兼顾。