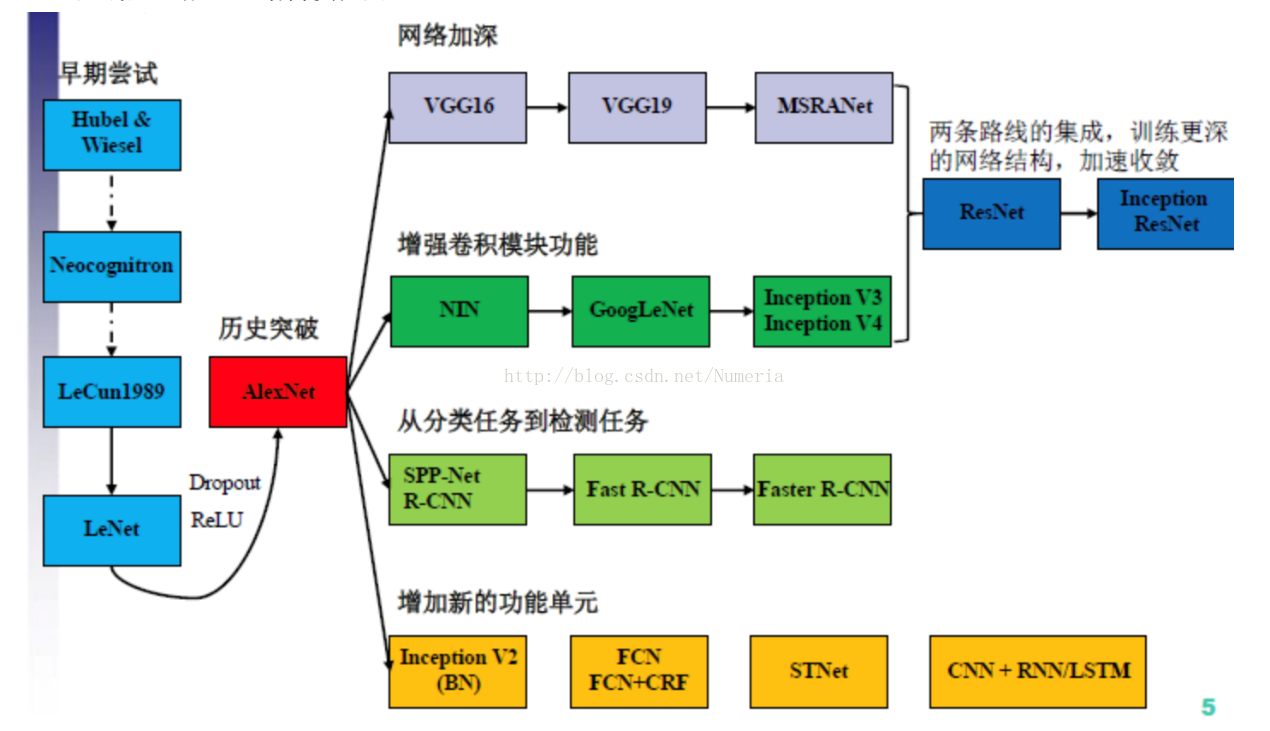
**神经网络发展的总结**



**一、AlexNet：**

**ILSVRC 2012** 大赛冠军，top-5的错误率降低至了16.4%，相比于第二名的成绩26.2%错误率有了很大的提升。

**AlexNet**可以说是在神经网络低估时期后的第一次发声，确立了深度学习（深度卷积网络）在计算机视觉的统治地位，同时也推动了深度学习在语音识别、自然语言处理、强化学习等领域的拓展。

**AlexNet将LeNet的思想发扬光大，把CNN的基本原理应用到了很深很宽的网络中。**

**AlexNet 技术要点**

（1）**成功使用ReLU作为CNN的激活函数**，并验证其效果在较深的网络超过了Sigmoid，成功解决了Sigmoid在网络较深时的梯度弥散问题。虽然ReLU激活函数在很久之前就被提出了，但是直到AlexNet的出现才将其发扬光大。

（2）**训练时使用Dropout随机忽略一部分神经元，以避免模型过拟合**。Dropout虽有单独的论文论述，但是AlexNet将其实用化，通过实践证实了它的效果。在AlexNet中主要是最后几个全连接层使用了Dropout。

（3）**在CNN中使用重叠的最大池化**。此前CNN中普遍使用平均池化，**AlexNet全部使用最大池化，避免平均池化的模糊化效果**。并且AlexNet中提出让**步长比池化核的尺寸小**，这样池化层的、输出之间会有重叠和覆盖，提升了特征的丰富性。

（4）提出了LRN层，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。

（5）使用CUDA加速深度卷积网络的训练，利用GPU强大的并行计算能力，处理神经网络训练时大量的矩阵运算。AlexNet使用了***两块***GTX 580 GPU进行训练，单个GTX 580只有3GB显存，这限制了可训练的网络的最大规模。因此作者将AlexNet分布在两个GPU上，在每个GPU的显存中储存一半的神经元的参数。因为GPU之间通信方便，可以互相访问显存，而不需要通过主机内存，所以同时使用多块GPU也是非常高效的。同时，AlexNet的设计让GPU之间的通信**只在网络的某些层进行**，控制了通信的性能损耗。

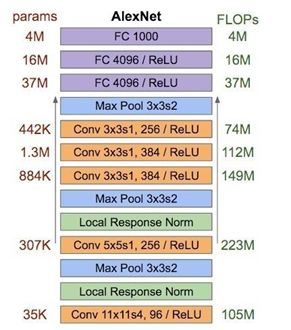
（6）数据增强，随机地从256\*256的原始图像中截取224\*224大小的区域（以及水平翻转的镜像），相当于增加了(256-224)2 \*2=2048倍的数据量。**如果没有数据增强，仅靠原始的数据量，参数众多的CNN会陷入过拟合中**，**使用了数据增强后可以大大减轻过拟合，提升泛化能力**。进行预测时，则是取图片的四个角加中间共5个位置，并进行左右翻转，一共获得10张图片，对他们进行预测并对10次结果求均值。同时，AlexNet论文中提到了**会对图像的RGB数据进行PCA处理，并对主成分做一个标准差为0.1的高斯扰动，增加一些噪声，这个Trick可以让错误率再下降1%。**

**AlexNet 网络结构分析**

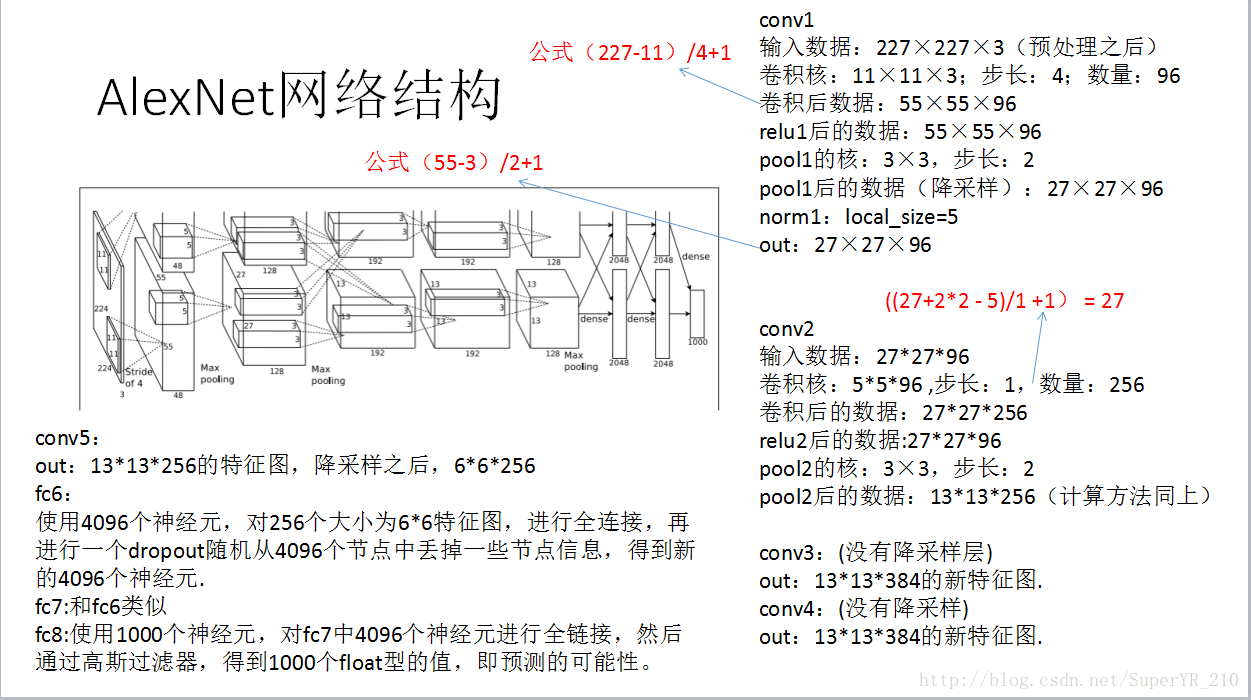
整个AlexNet有8个需要训练参数的层（不包括池化层和LRN层），前5层为卷积层，后3层为全连接层，如图4所示。AlexNet最后一层是有1000类输出的Softmax层用作分类。 LRN层出现在第1个及第2个卷积层后，而最大池化层出现在两个LRN层及最后一个卷积层后。ReLU激活函数则应用在这8层每一层的后面。因为AlexNet训练时使用了两块GPU，因此这个结构图中不少组件都被拆为了两部分。现在我们GPU的显存可以放下全部模型参数，因此只考虑一块GPU的情况即可（发展的力量嘛）

我们可以发现一个比较有意思的现象，在前几个卷积层，虽然计算量很大，但参数量很小，都在1M左右甚至更小，只占AlexNet总参数量的很小一部分。这就是卷积层有用的地方，可以通过较小的参数量提取有效的特征。而如果前几层直接使用全连接层，那么参数量和计算量将成为天文数字。虽然每一个卷积层占整个网络的参数量的1%都不到，但是如果去掉任何一个卷积层，都会使网络的分类性能大幅地下降。

**(C-L-M)-🡪(C-L-M)🡪(C-C-C)🡪M🡪(F-F-F)**



**AlexNet中图像Size变化**



**AlexNet 中60M参数**

AlexNet只有8层，但是它需要学习的参数有60000000个，相比如他的层数，这是一个很可怕的数字了，我们来计算下这些参数都是怎么来的：

首先，Alexnet的输入部分是224\*224\*3的图片，经过预处理之后，变为227\*227\*3的大小，作为网络的输入的部分。输入层：227×227×3

-------------------------------------------------------------------------------------------------------------------------

| 参数 | 图片尺寸

-------------------------------------------------------------------------------------------------------------------------  
C1：96×11×11×3 (卷积核个数/宽/高/厚度) 34848个 55\*55\*96  
C2：256×5×5×48（卷积核个数/宽/高/厚度） 307200个 27\*27\*256  
C3：384×3×3×256（卷积核个数/宽/高/厚度） 884736个 13\*13\*384  
C4：384×3×3×192（卷积核个数/宽/高/厚度） 663552个 13\*13\*384  
C5：256×3×3×192（卷积核个数/宽/高/厚度） 442368个 13\*13\*384  
R1：4096×6×6×256（卷积核个数/宽/高/厚度） 37748736个 6\*6\*256--4096  
R2：4096×4096 16777216个 4096  
R3：4096×1000 4096000个 1000

总共加起来：C1+C2+C3+C4+C5+F1+F2+F3=60954656=60M左右

**参考：https://blog.csdn.net/chaipp0607/article/details/72847422/**

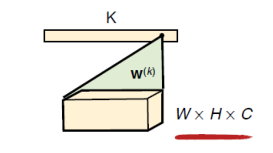
**注意的地方：**

在CNN中，**卷积核的尺寸是人为指定的，但是卷积核内的数全部都是需要不断学习得到的。**比如一个卷积核的尺寸为3×3×3，分别是宽，高和厚度，那么这一个卷积核中的参数有27个。

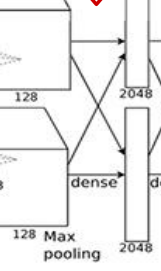
池化操作的降维**改变的是图像的宽高，而不改变通道数。**  
**在这里需要说明一点：**   
**卷积核的厚度=被卷积的图像的通道数**   
**卷积核的个数=卷积操作后输出的通道数**

# 全连接层

## 全连接层的作用

CNN中的全连接层与浅层神经网络中的作用是一样的，负责逻辑推断，所有的参数都需要学习得到。有一点区别在于第一层的全连接层用于链接卷积层的输出，它还有一个作用是去除空间信息（通道数），是一种将三维矩阵变成向量的过程（一种全卷积操作），其操作如下：   
  
输入图像是*W*×*H*×*C*，那么卷积核的尺寸为*W*×*H*×*C*，这样的话整个输入图像就变成了一个数，一共有k个数（第一层全连接层后的神经元个数），就有K个这样的*W*×*H*×*C*的卷积核。所以全连接层（尤其是第一层）的参数量是非常可怕的，也是由于这个弊端，后来的网络将全连接取消了

## AlexNet中的全连接层

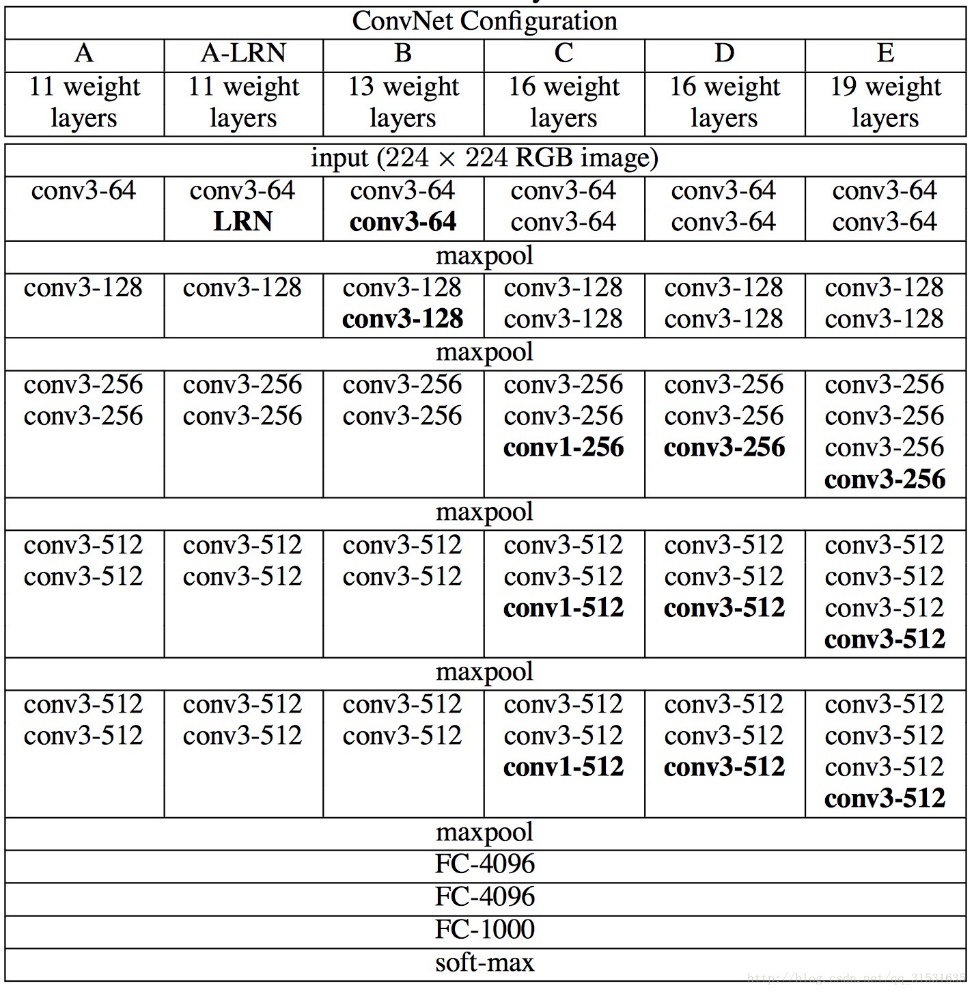
再回到AlexNet结构，R1，R2，R3就是全连接层。R2，R3很好理解，在这里主要说明下R1层：   
输入图像：13×13×256  
卷积核尺寸：13×13×256个数2048×2 （注意这个地方，全连接是用全卷积实现的）  
输出尺寸：4096（列向量）   
从最开始的结构中可以看到，R1中也有通道的交互：   
  
所以串接后的通道数是256，全卷积的卷积核尺寸也就是13×13×256，一个有4096个这样尺寸的卷积核分别对输入图像做4096次的全卷积操作，最后的结果就是一个列向量，一共有4096个数。这些数的排布其实就相当于传统神经网了里面的第一个隐藏层而已,通过R1后，后面的链接方式和ANN就没有区别了。**要学习的参数也从卷积核参数变成了全连接中的权系数。（解释的很到位）**

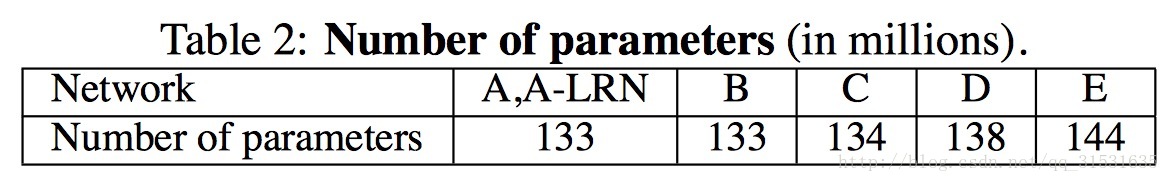
**https://blog.csdn.net/zyqdragon/article/details/72353420**

**二、VGGNet**

**VGG**   
来自牛津大学的 VGG 网络（参见：Very Deep Convolutional Networks for Large-Scale Image Recognition）是第一个在各个卷积层使用更小的 3×3 过滤器（filter），并把它们组合作为一个卷积序列进行处理的网络。

VGG网络结构图：





1、VGG和AlexNet、LeNet

这看来和 LeNet 的原理相反，其中是大的卷积被用来获取一张图像中相似特征。和 AlexNet 的 9×9 或 11×11 过滤器不同，过滤器开始变得更小。**但是 VGG 巨大的进展是通过依次采用多个 3×3 卷积，能够模仿出更大的感受野（receptive field）的效果。**

VGGNet是牛津大学计算机视觉组（Visual Geometry Group）和Google DeepMind公司的研究员一起研发的的深度卷积神经网络。

1. VGGNet探索了卷**积神经网络的深度**与其**性能**之间的关系，通过反复堆叠3\*3的小型卷积核和2\*2的最大池化层，VGGNet成功地构筑了16~19层深的卷积神经网络。
2. VGGNet相比之前state-of-the-art的网络结构，错误率大幅下降，**并取得了ILSVRC 2014比赛分类项目的第2名和定位项目的第1名**。
3. 同时VGGNet的拓展性很强，迁移到其他图片数据上的泛化性非常好。
4. VGGNet的结构非常简洁，整个网络都使用了同样大小的卷积核尺寸（3\*3）和最大池化尺寸（2\*2）。
5. 到目前为止，VGGNet依然经常被用来提取图像特征。VGGNet训练后的模型参数在其官方网站上开源了，可用来在domain specific的图像分类任务上进行再训练（相当于提供了非常好的初始化权重），因此被用在了很多地方。

**2、VGG卷积**

**结构A：**和AlexNet类似，卷积层分为了5个stage，全连接层还是3层。只不过卷积层用的都是**3x3大小的filter**，共8个卷积层，总计11层。

**结构A-LRN：**保留AlexNet里面LRN操作，其他与结构A无区别，总计11层。（3全联）

**结构B：**在A的stage1和stage2分别增加**一个3x3**的卷积层，共有10个卷积层，总计13层

**结构C：**在B的基础上，stage3，stage4，stage5分别增加**一个1x1**的卷积层，有13个卷积层，总计16层。

**结构D：**在B的基础上，stage3，stage4，stage5分别增加**一个3x3**的卷积层，有13个卷积层，总计16层。

**结构E：**在D的基础上，stage3，stage4，stage5分别再增加**一个3x3**的卷积层，有16个卷积层，总计19层。

VGGNet拥有5段卷积，每一段内有2~3个卷积层，同时每段尾部会连接一个最大池化层用来缩小图片尺寸。每段内的卷积核数量一样，越靠后的段的卷积核数量越多：64 – 128 – 256 – 512 – 512。其中经常出现多个完全一样的3\*3的卷积层堆叠在一起的情况，这其实是非常有用的设计。

**2.1 3\*3卷积核堆叠优点**

**所有卷积层都是同样大小的filter！尺寸3x3，卷积步长Stirde = 1，填充Padding = 1**！

**A**、3x3是最小的能够捕获左、右、上、下和中心概念的尺寸；

**B**、**两个3x3**的卷积层连在一起可**视为5x5的filter**，**三个**连在一起**可视为一个7x7**的

   这是卷积的性质，受过#信号系统#这门课摧残的同学应该记忆犹新…

**C**、多个3x3的卷积层比一个大尺寸的filter卷积层**有更多的非线性**，使得判决函数更加具有判断性。

**D**、多个3x3的卷积层笔一个大尺寸的filter具有**更少的参数**。

**2.2 1\*1卷积的意义**

C很有意思，相比B多了几个1\*1的卷积层，1\*1卷积的意义主要在于线性变换，而输入通道数和输出通道数不变，没有发生降维。

**3、VGG训练技巧**

1）网络预训练

VGGNet在训练时有一个小技巧，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模型，这样**训练收敛的速度更快**。

2）Multi-Scale方法。

在**预测**时，VGG采用Multi-Scale的方法，将图像scale到一个尺寸Q，并将图片输入卷积网络计算。然后在最后一个卷积层使用滑窗的方式进行分类预测，将不同窗口的分类结果平均，再将不同尺寸Q的结果平均得到最后结果，这样可提高图片数据的利用率并提升预测准确率。

同时在**训练**中，VGGNet还使用了Multi-Scale的方法做数据增强，将原始图像缩放到不同尺寸S，然后再随机裁切224\*224的图片，这样能增加很多数据量，对于防止模型过拟合有很不错的效果。实践中，作者令S在[256,512]这个区间内取值，使用Multi-Scale获得多个版本的数据，**并将多个版本的数据合在一起进行训练**。

**4、作者总结观点**

（1）LRN层作用不大。

（2）越深的网络效果越好。

（3）1\*1的卷积也是很有效的，但是没有3\*3的卷积好，大一些的卷积核可以学习更大的空间特征。

**三、GoogleNet & Inception**

**跳出加深网络层数的思路。通过增加网络“宽度”的方式增加网络的复杂度。**

与VGG同在2014年出现，取得了ILSVRC 2014比赛第一名。

令人感兴趣的是，在GoogLeNet出来之前，主流的网络结构突破大致是网络更深（即层数增加），网络更宽（即神经元数目增多），但这样做存在一些缺点：

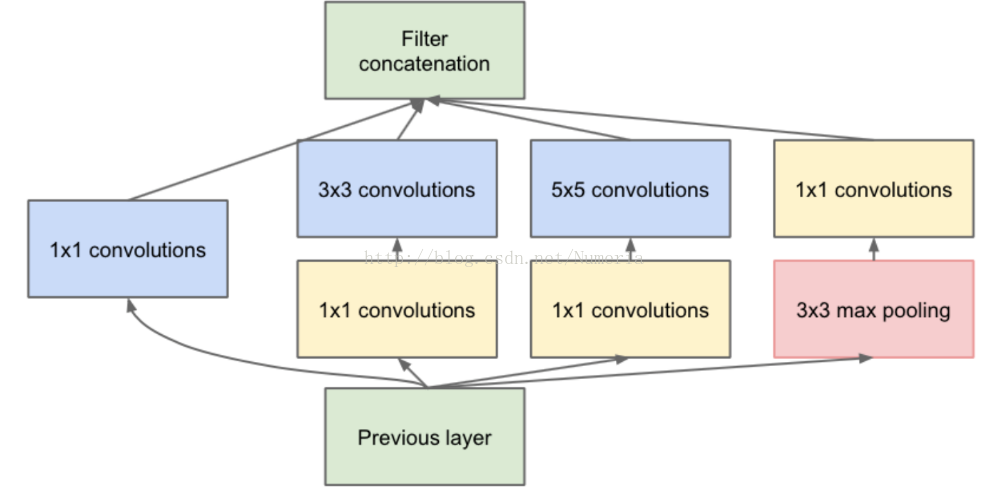
1. 当训练集有限时，参数过多，模型会出现过拟合；

2. 网络越大，计算复杂度越大，设计起来越困难；

3. 当层数增多时，梯度越往后越容易消失；

**Inception V1**

**Inception V1 图**



(2) Inception V1有22层深，控制参数量的同时提高性能。控制参数量的原因：

* 参数越多模型越庞大，需要供模型学习的数据量就越大，而目前高质量的数据非常昂贵
* 参数越多，耗费的计算资源也会更大。

(3) 参数少、模型深、表达能力强：

* 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AlexNet 90%的参数，去除后训练快、减轻过拟合。
* Inception Module提高了参数的利用效率

(4) 一般，卷积层要提升表达能力，主要依靠增加输出通道数，副作用：计算量增大和过拟合。

* 一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。
* 可以在输出通道之间进行信息组合。

(5) Inception Module

* 包含3种不同尺寸的卷积核1个最大池化，增加了网络对不同尺度的适应性，与Multi-scale类似。
* Inception Net的主要目标是找到最优的稀疏结构单元，即Inception Module
* 一个好的稀疏结构，应该把相关性高的一簇神经元节点连接在一起。普通数据中通过聚类，图像中临近区域的数据相关性高。
* 而且可能有多个卷积核，同一空间位置但在不同通道的卷积核的输出结果相关性极高。
* 1\*1的卷积可以把这些相关性很高的、在同一个空间位置但是不同通道的特征连接在一起
* 1\*1连接的节点相关性最高，使用大一点的卷积核3\*3/5\*5连接节点的相关性也很高，增加多样性。
* 整个网络中有多个堆叠的Inception Module，希望靠后的Module能捕获更高阶的抽象特征，因此靠后的Module的卷积的空间集中度应该逐渐降低，3\*3和5\*5大面积的卷积核的占比(输出通道数)应该更多。

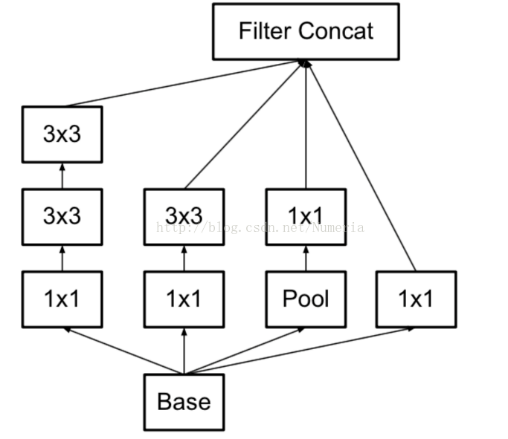
(6) Inception Net 有22层深，使用辅助分类节点(auxiliary classifiers)，将中间某一层的输出用作分类，用较小的权重(0.3)加到最终分类中。

(7) Inception 不同版本

* Inception V1，Going Deeper with Convolutions，2014年9月（top5 error = 6.67%）
* Inception V2，Batch normalization：Accelerating Deep Network Training by Reducing Internal Covariate,5\*5🡪 3\*3 2015年2月 (top5 error = 4.8%)
* Inception V3, Rethinking the Inception Architecture for Computer Vision, 2015年12月 (top5 error = 3.5%)
* Inception V4, Inception-ResNet and the Impact of Residual Connections on Learning, 2016年2月 (top5 error = 3.08%)

**Inception V2**

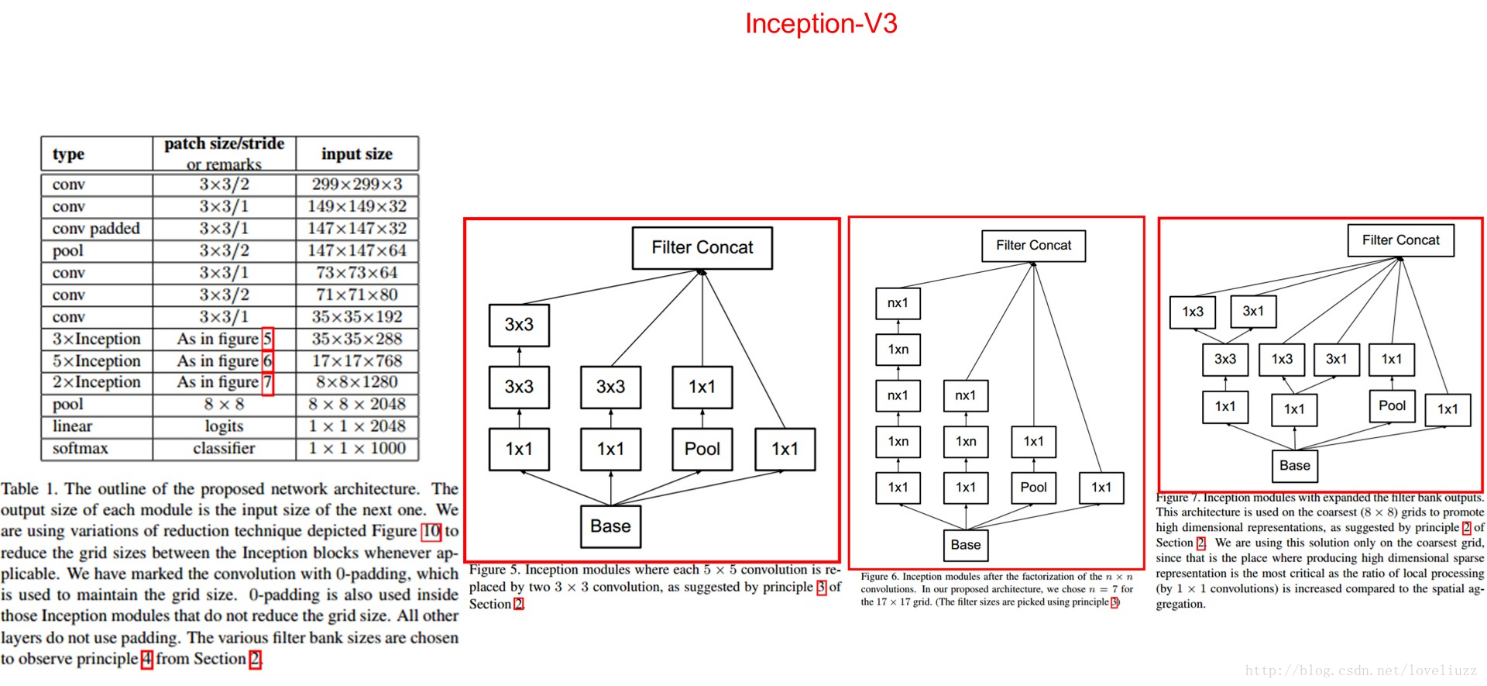
**Inception V2 图**



1. 学习VGG，用两个3\*3的卷积替代5\*5的大卷积，降低参数量并减轻过拟合
2. 提出了Batch normalization，有效的正则化方法，让大型卷机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。
   * BN在用于神经网络某层时，会对每一个mini-batch数据的内部进行标准化处理，输出规范到 N(0,1)的正态分布，减少了Internal Covariate Shift（内部神经元分布的改变）。
   * 传统的深度神经网络在训练时，每一层的输入的分布都在变化，导致训练变得困难，只能使用一个很小的学习速率解决这个问题。对每层使用BN后就可以有效解决这个问题。
   * BN还起到了正则化的作用，可以减少或取消Dropout，简化网络结构。
3. 除了BN外，提高增益的其他调整：
   * 增大学习速率并加快学习衰减速度以适用BN规范后的数据
   * 去除Dropout并减轻 L2 正则
   * 去除 LRN
   * 更彻底地对训练样本进行shuffle
   * 减少数据增强过程中对数据的光学畸变，因为BN训练更快，每个样本被训练的次数更少，因此更真实的样本对训练更有帮助。

**Inception V3**

**Inception V3 图**



* 引入了Factorization into small convolutions，将一个较大的二维卷积拆成两个较小的一维卷积，

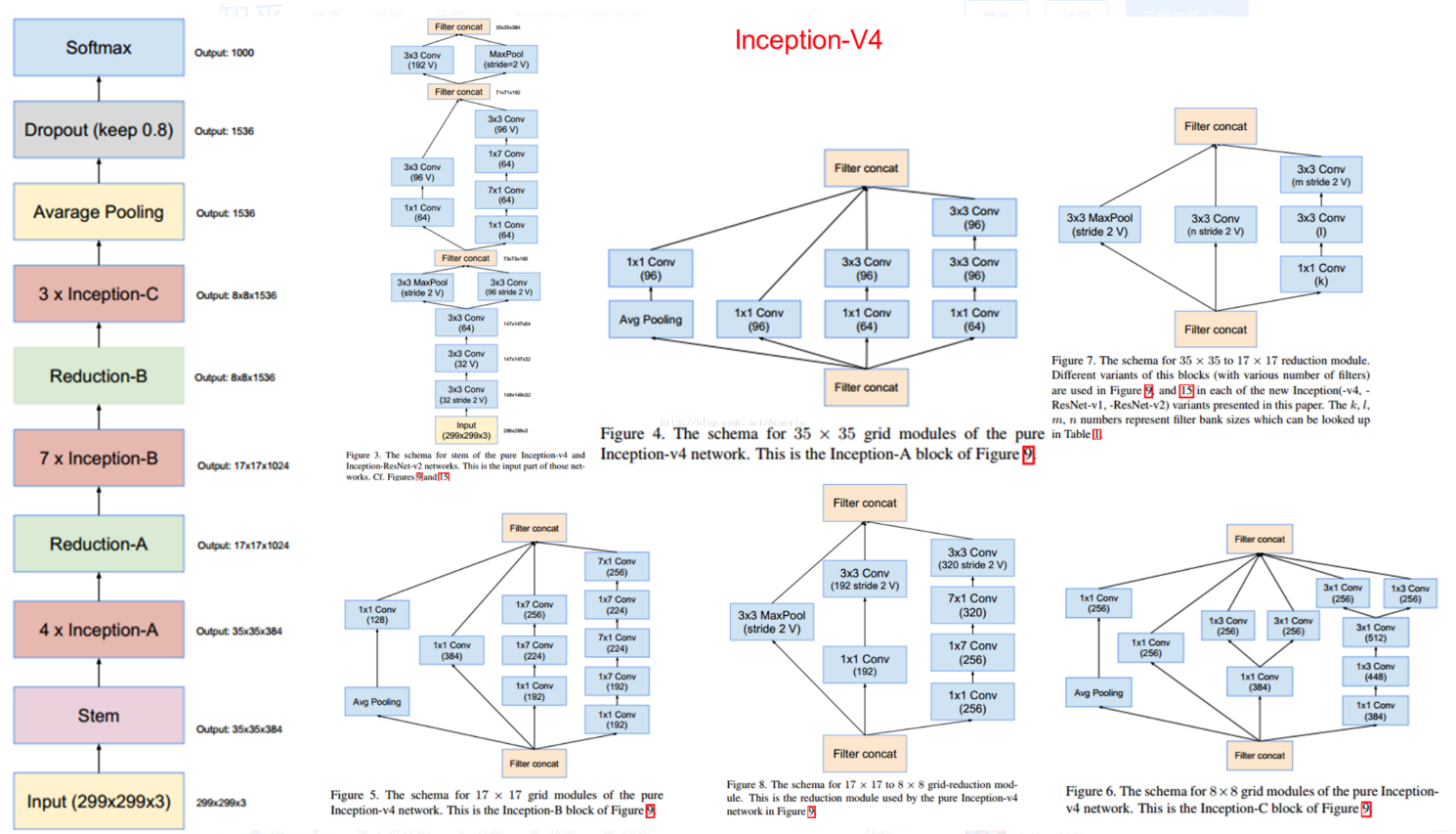
eg：7\*7拆成1\*7和7\*1的卷积，3\*3拆成1\*3和3\*1节约了参数，减轻过拟合，同时增加了一层非线性扩展模型表达能力。非对称结构，比对称拆分为相同的小卷积效果更明显，可以处理更多/更丰富的空间特征，增加特征多样性。

* 优化了Inception Module的结构，有35\*35 17\*17 和8\*8三种不同结构。只在网络后部出现，前部还是普通卷积层。在Inception Module使用分支，分支中又使用分支。

**Inception V4**

**Inception V4 图**

* 将Inception Module与ResNet结合。



**四、ResNet**

**网络中的网络（Network-in-network）**   
网络中的网络（NiN，参见论文：Network In Network）的思路简单又伟大：使用 1×1 卷积为卷积层的特征提供更组合性的能力。

NiN 架构在各个卷积之后使用空间 MLP 层，以便更好地在其他层之前组合特征。同样，你可以认为 1×1 卷积与 LeNet 最初的原理相悖，但事实上它们可以以一种更好的方式组合卷积特征，而这是不可能通过简单堆叠更多的卷积特征做到的。这和使用原始像素作为下一层输入是有区别的。其中 1×1 卷积常常被用于在卷积之后的特征映射上对特征进行空间组合，所以它们实际上可以使用非常少的参数，并在这些特征的所有像素上共享！

MLP 的能力能通过将卷积特征组合进更复杂的组（group）来极大地增加单个卷积特征的有效性。这个想法之后被用到一些最近的架构中，例如 ResNet、Inception 及其衍生技术。

NiN 也使用了平均池化层作为最后分类器的一部分（这里作者说的有异议，全连接层替换成平均池化层，NIN提出的），这是另一种将会变得常见的实践。这是通过在分类之前对网络对多个输入图像的响应进行平均完成的。

