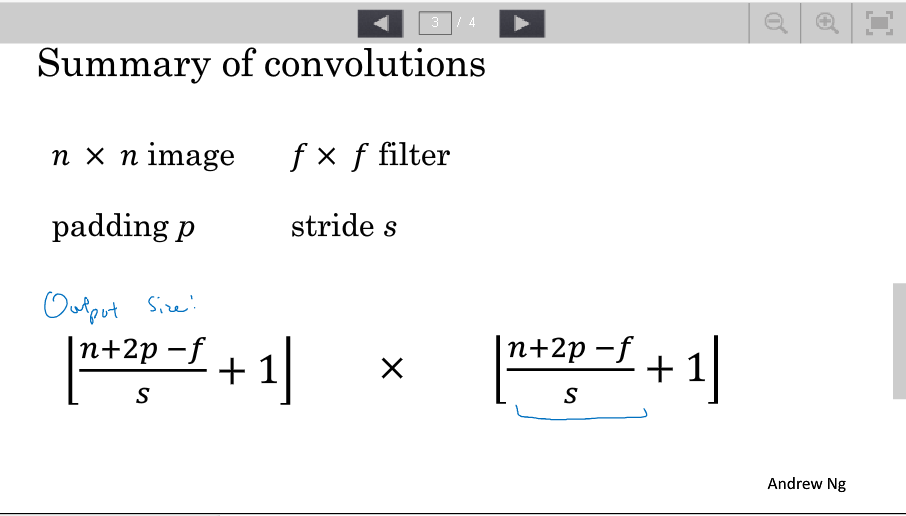
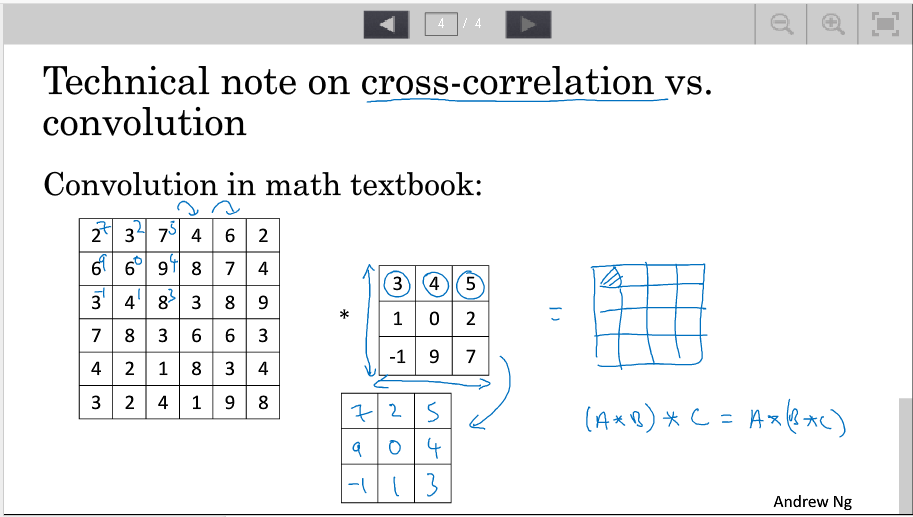
**2018-4-16**

[**来自吴恩达老师的卷积神经网络课程**](http://mooc.study.163.com/learn/2001281004?tid=2001392030%23/learn/content?type=detail&id=2001728686&cid=2001725121)

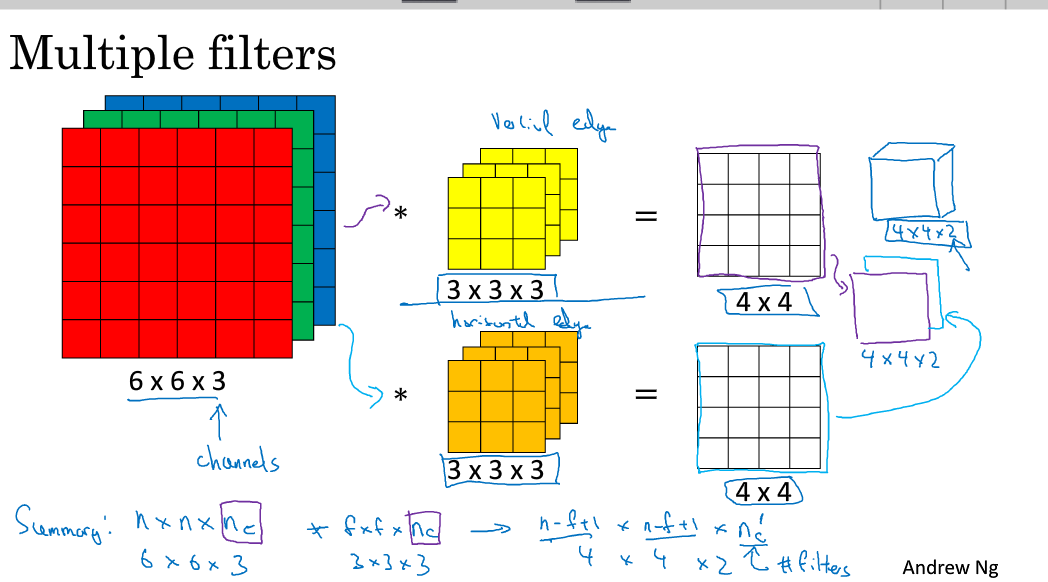
1. 卷积运算的输入与输出size：



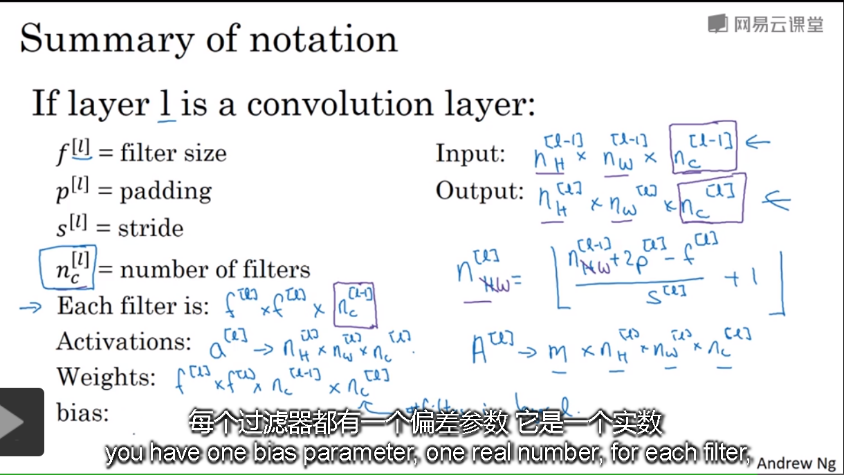
2. 数学中的互相关（神经网络中的卷积）vs数学中的卷积（需要做一个镜像）：



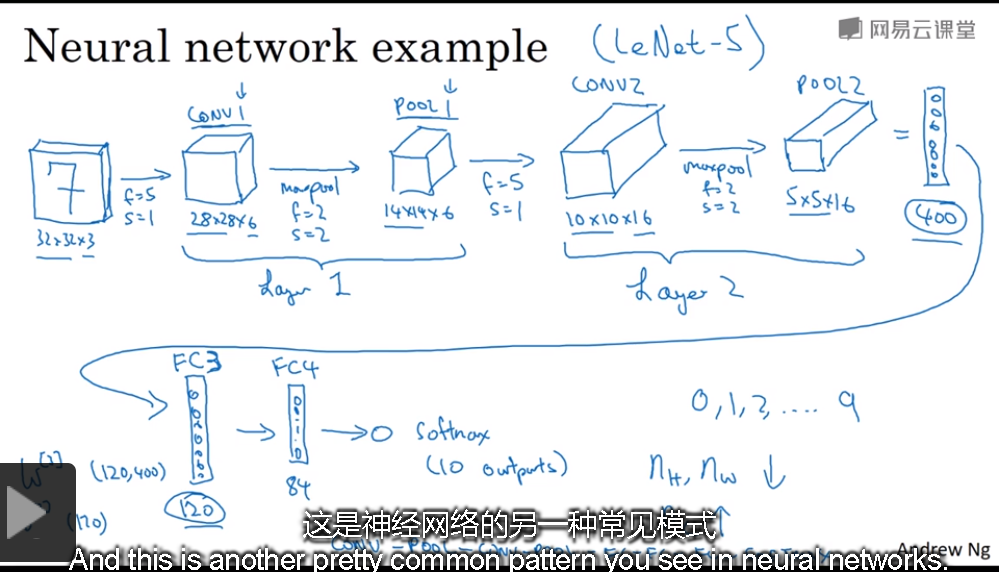
1. 原图与每个filter的通道数（channels或说depth）需要一致，结果为一个通道。当使用多个filter,最后结果的通道数=filter的个数：



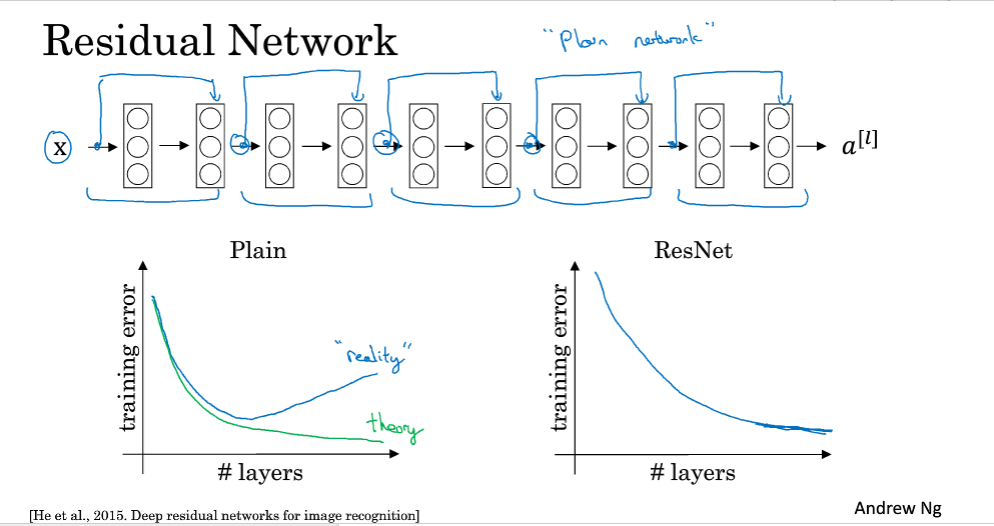
1. 卷积神经网络中的L层的变量维度：



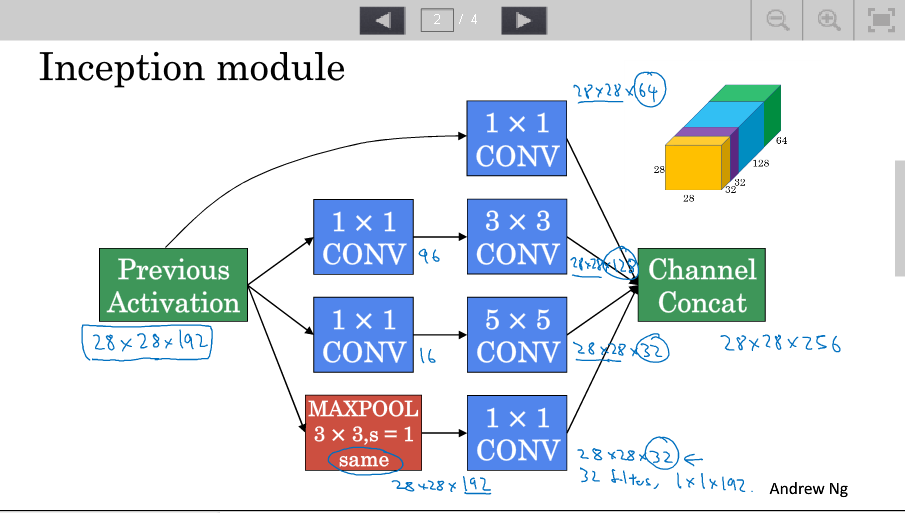
1. 卷积神经网络：卷积层，池化层，全连接层。池化层可以缩减模型大下，提高计算速度。提高锁区特征的鲁棒性（max pooling反向传递的时候没有参数需要学习，s,f等超参数需要手动设置）。随着层数的增大，w,h减小而channels增大：



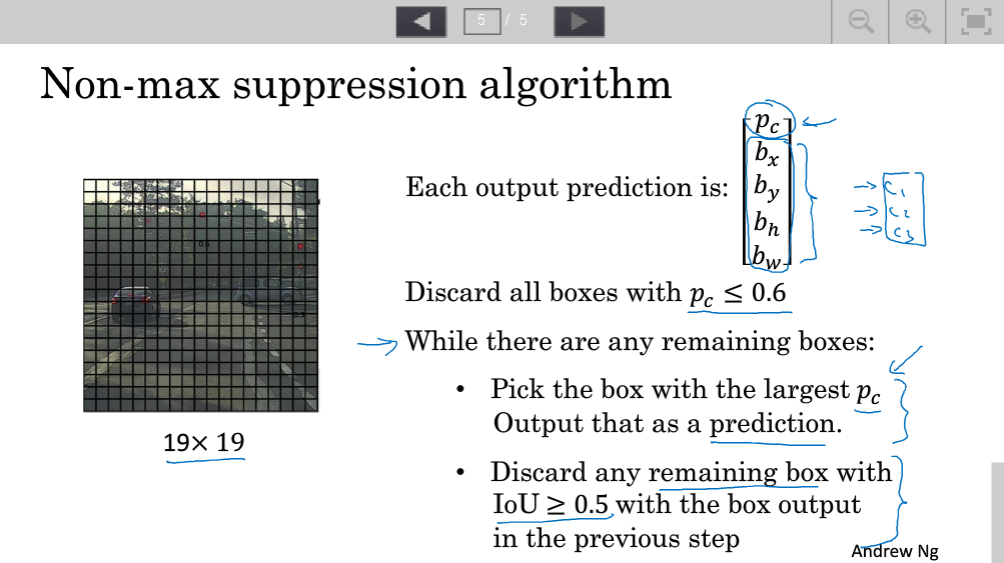
1. 相比全连接，卷积运算可以共享参数(提取某种特征的filter)和稀疏连接，所有需要的参数比较少。
2. LeNet-5
3. AlexNet
4. VGG-16(16层卷积层和全连接层，网络参数多，但是结构比较一致简单)
5. ResNet（加上skip connection或说Residual block残差块）。普通的网络层数增加，性能可能会变差，但是残差网络部不会。



1. 1\*1\*nc filter（bottle neck layer）可以缩减通道数，降低计算复杂度（只要合理使用，是不会降低网络性能的）。
2. Inception module，不需要人为确定filter的大小以及是否需要池化，而是由网络自行确定，你可以提前为网络添加所有可能的filter的池化层，并把它们的输出连接在一起，然后在训练的时候由网络自行确定选择哪个或哪几个filter组合以及是否池化，如下图，同一层中有多个大小不同的filter和一个池化层（inception network->GooLeNet）



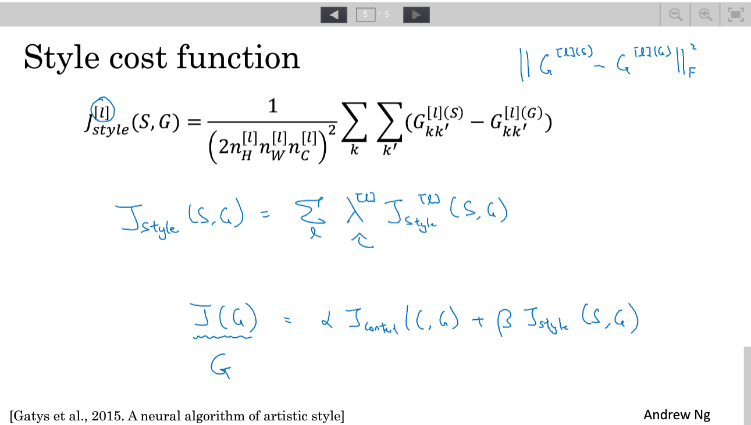
1. 迁移学习(transferring learning).
2. 数据增强(data agumentation)
3. 全连接层和卷积层其实是一样的，在数学层面，因此可以把全连接层当成卷积层来做。
4. 目标检测：
5. 目标定位（bounding box）,特征点检测（landmark）
6. Sliding window dection 细粒度小步幅，计算成本比较高（也有很多重复计算）。滑动窗口的卷积实现，若将整个图片进行卷积，则可以避免重复计算，但是对于不属于任何滑动窗口的物体，任然可能检测不出来
7. YOLO可以得到更准确的bbox，将图片划分格子，若物体的中心点在某一格子，则该格子负责预测它的bbox
8. **交并比IoU（Intersection over Union）**,用来衡量定位精度。
9. **非极大值抑制NMS**，剔除与得分较高的bbox IoU比较高的其他bbox，可以保证每个物体只被检测一次。（当检测多类别物体时c1,c2,c3，需要进行三次非极大值抑制抑制）



1. **Anchor Box**（一般检测物体的形状手工指定形状，或者通过聚类算法得到一组具有代表性anchor box） 一个格子可以检测多个物体。另一方面，因为anchor boxed 的形状，可以使算法更具针对性，比如针对一些又高又瘦的物体，或又矮又胖的物体。
2. RCNN:运行图像分割算法得到region proposal,每个region作为CNN的输入

FastRCNN: region proposal的卷积实现

FasterRCNN引入RPN网络，共享CNN部分卷积层得到候选区域提供给RoI层

1. 人脸识别：one-shot学习，Siamese网络，Triplet损失
2. 神经风格转移：内容代价函数+风格（通道之间的相关性有多相似）损失函数
3. 1D 2D 3D的卷积运算差不多