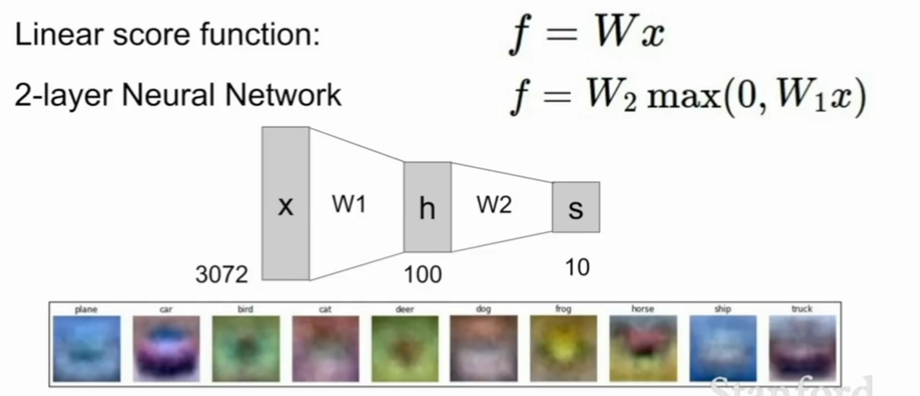
Lecture 5:

Training Neural Networks,

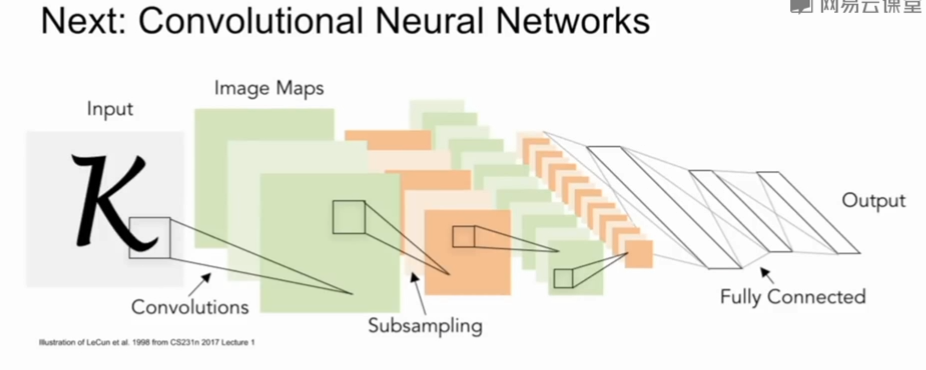
# 卷积神经网络

# Part I

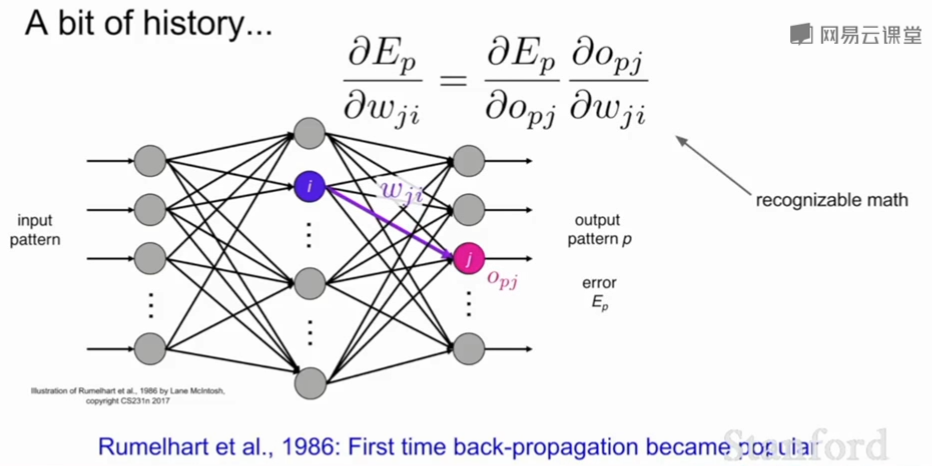
上节回顾：



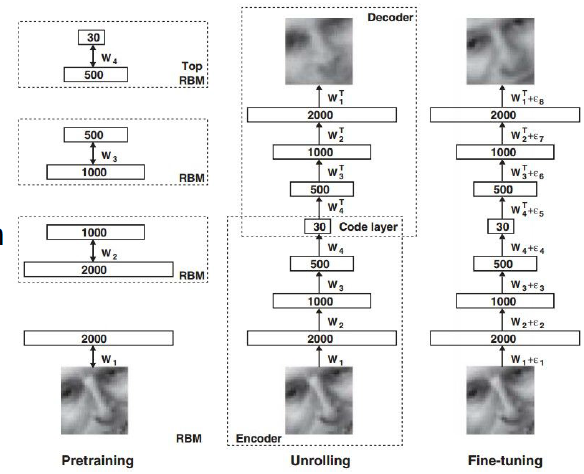
卷积神经网络：



1986年反向传播的思想才被提出



2006年 Geoff Hinton & Ruslan Salakhutdinov 的论文表明神经网络不仅可以训练，还可以高效的训练



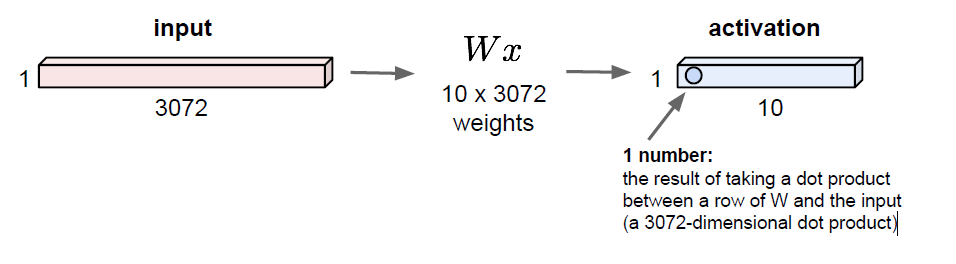
他们的方法是先经过一个预训练阶段，对每一个隐层使用受限玻尔兹曼机来建模，这样通过迭代训练每一层，得到一些初始化权重，得到所有隐层之后，以此来初始化一个完整的神经网络，然后再做反向传播，优化参数

2012年Alex Krizshevsky的 Alex net 将卷积神经网络实现，并取得了很好的成绩

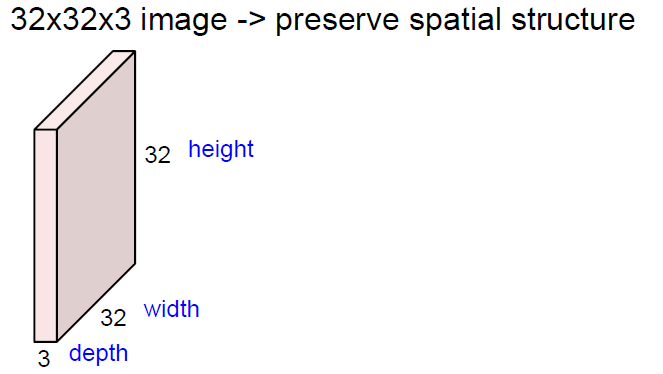
Fully Connected Layer

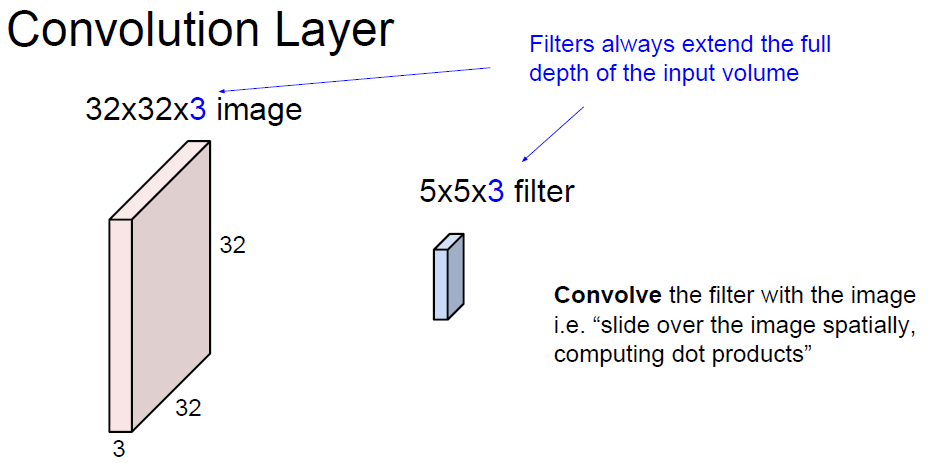
## 例（向量）：

32\*32\*3 张成3072\*1



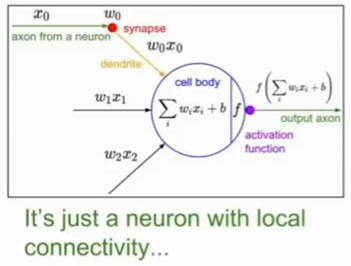
## 例（图像）：

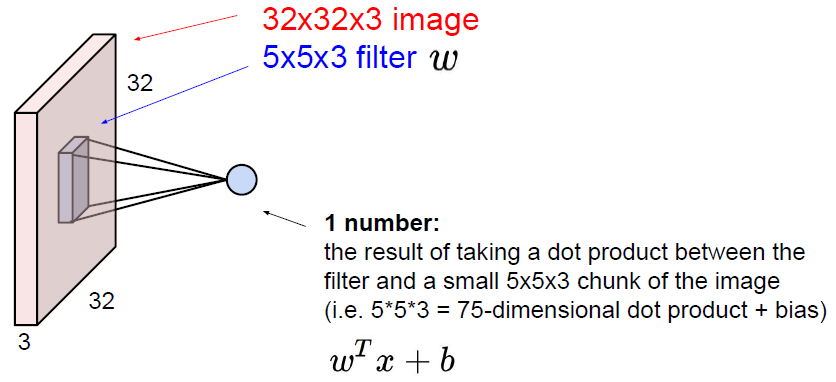




图片中，取一个卷积核w(5,5,3) 与图像特定区域之间做点积，得到一个**数字**

**注意这里是求和，**



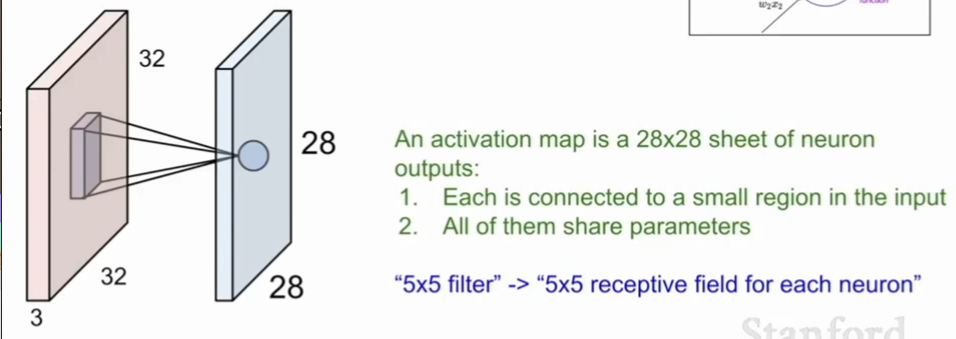


Activation map:

对于每一个5x5 的 filter，我们都让他们划过整个图像空间，同时这些卷积核具有相同的权重

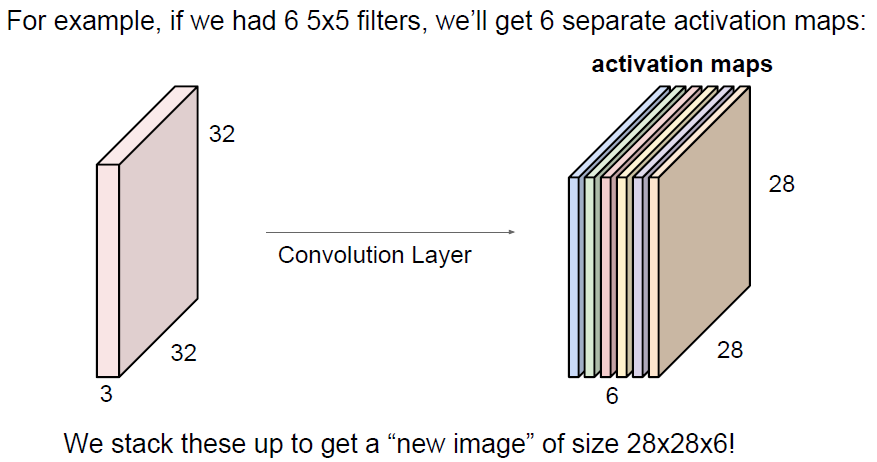
\*\*\* 这里为什么是28呢？ 因为32x32 中取5x5的话，就是从头扫到尾扫28次，横竖都是，

5x5 filter => 5x5 receptive field



Filter的数量就是所谓的深度

不同的层取自图像中的同一个区域，不同的filter就是对应不同的深度

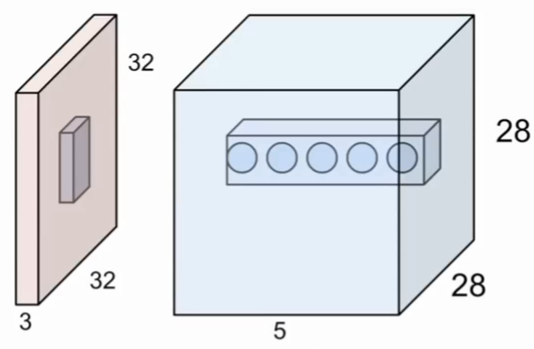


每一个卷积层的运算：（反向求导时会用到）表示filter扫描时第[i,j]个区域

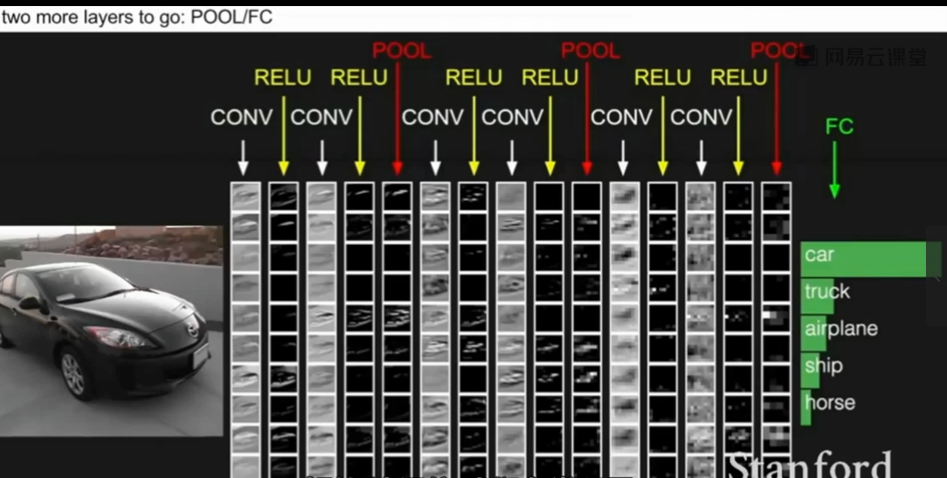
out = 

dw = 





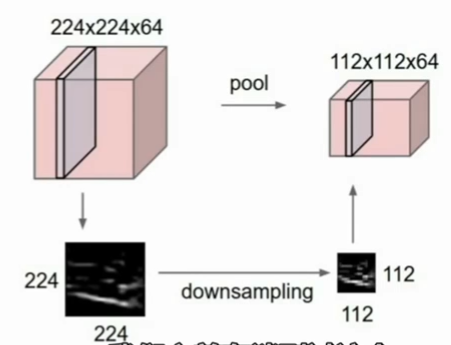
以上就是卷积层CONV，每隔几个CONV就有一个池化层POOL，实际的卷积神经网络：（还有两个层要讲，POOL/FC）



## 池化POOLING LAYERS

把数据体积降维’，下采样

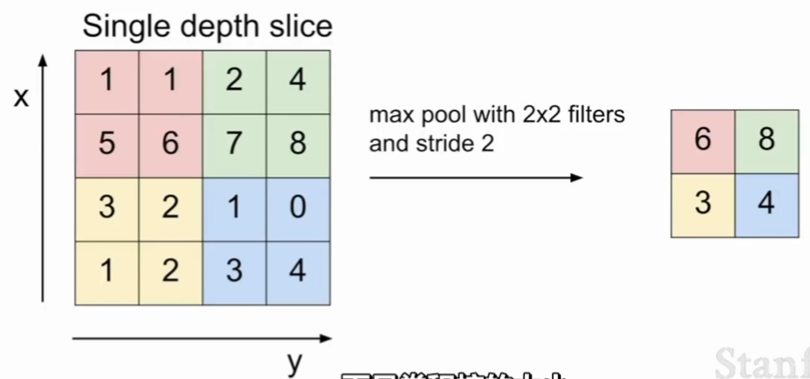
我们只会做超平面上的下采样，不会做深度上的降维，如下图，深度不变还是64



**Maxpooling 目前用得更多，大概的看最大值更好的表示了这个小的区域的情况**

**Stride：**每一个filter间隔的元素

例： 这里的池化(max pooling)是取一个2x2的区域中取最大值，得到[6，8，3，4]



**在一些人工智能算法中，人决已经开始更多的应用stride（滑动）而不用池化去做采样了。因为stride 和 pooling 都能实现下采样**

实际运算过程：

W1,H1,D1为要做filter前的原始输入

W2,H2,D2,为filter后的输出，也就是filter会有多少个不同的输出

Spatial extent 池化的空间范围

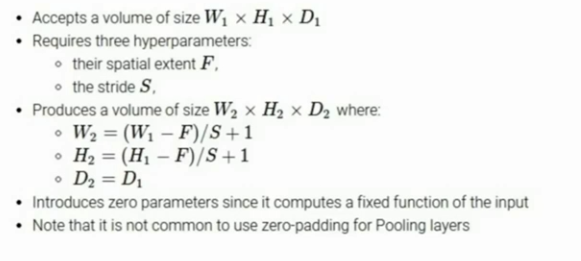
Stride 步长

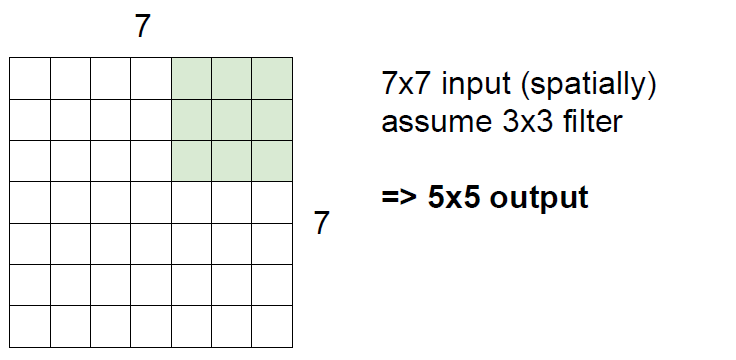
\*\*\* Common settings:

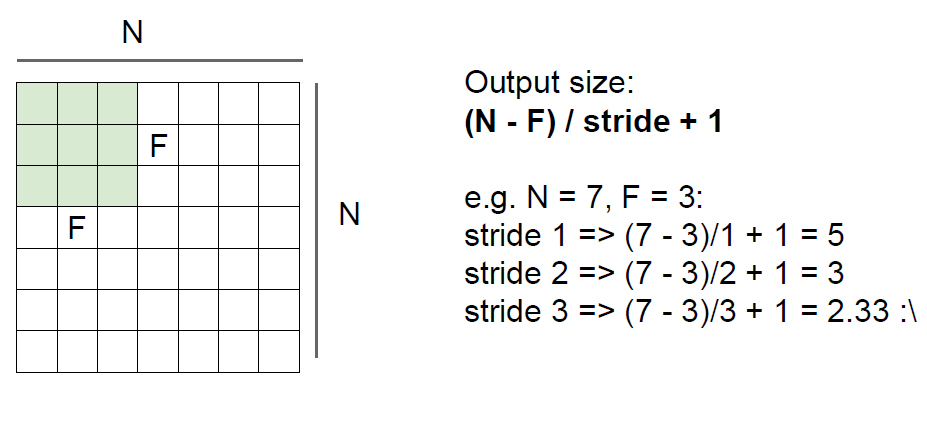
F=2,S=2

F=3,S=2

\*\*\* POOLING超参数 F filter都是正方形

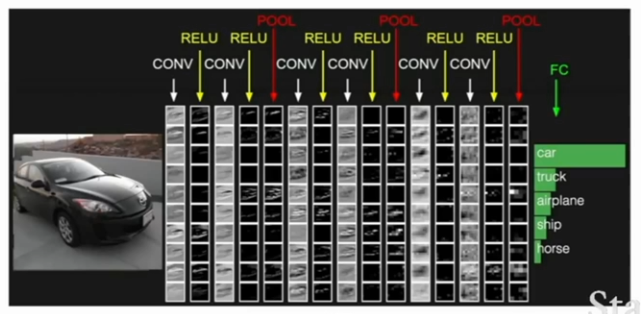


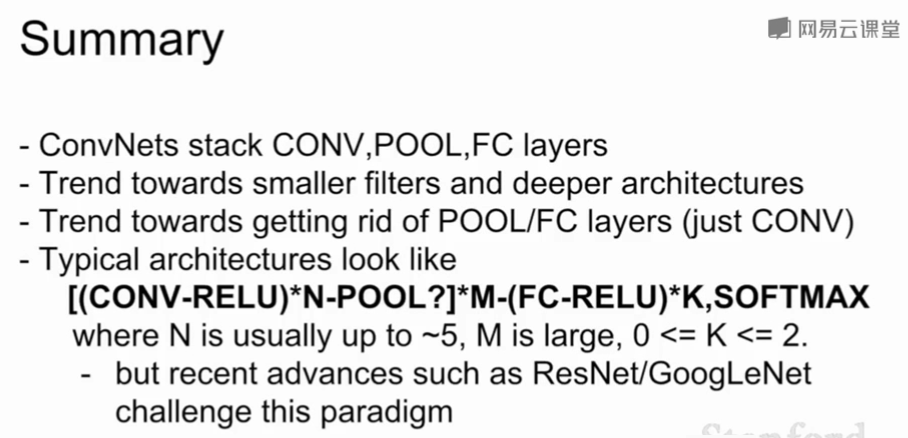


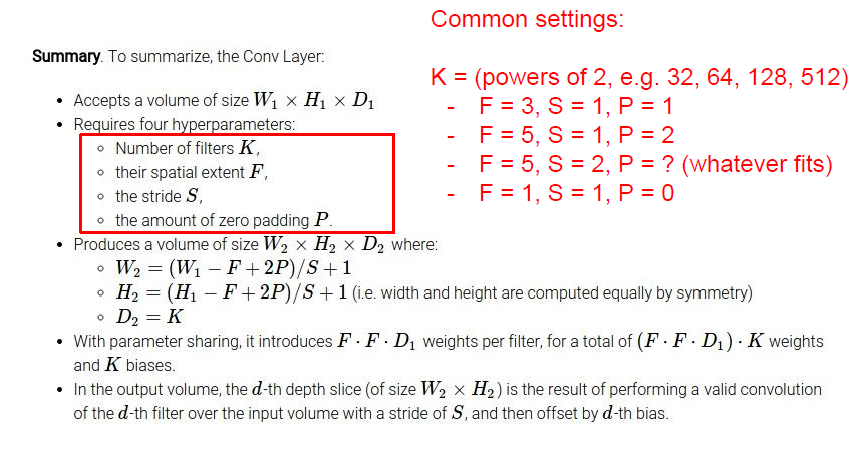


## FC层 全连接层 （Full Connected Layer）

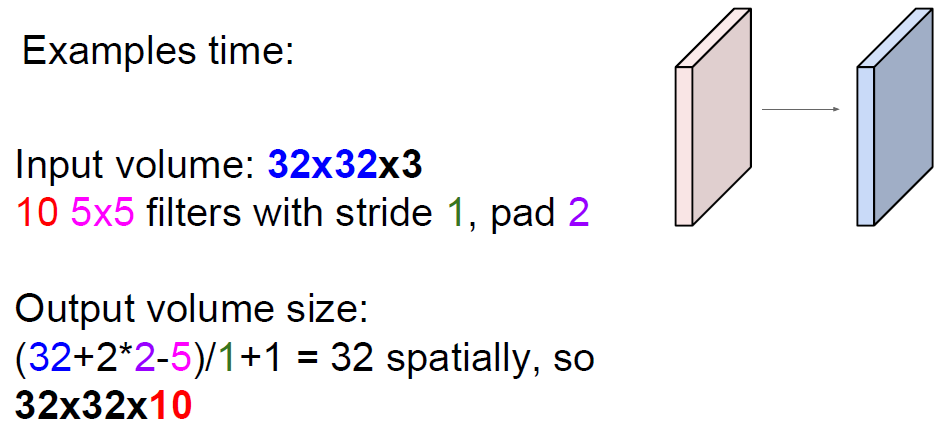
基本的神经网络都差不多的结构

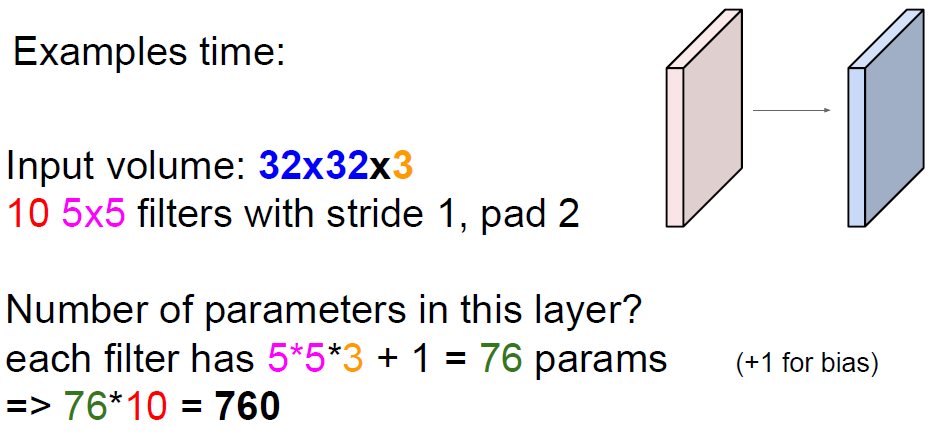






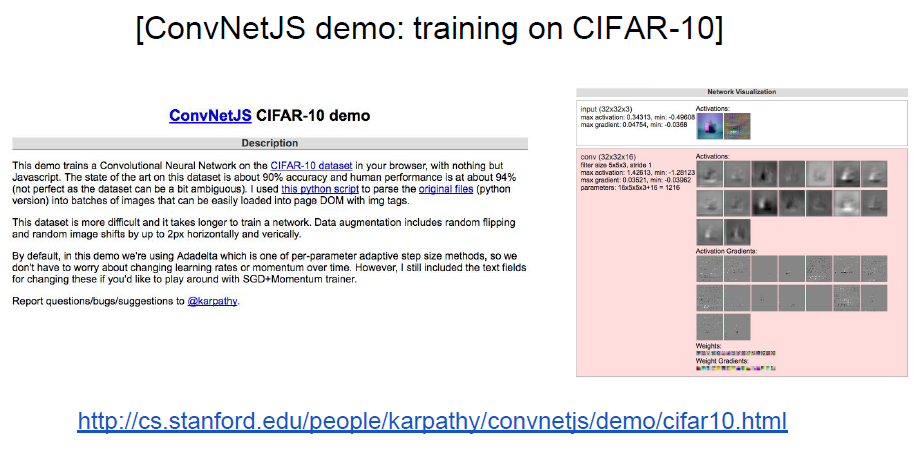
例：





卷积神经网络的Demo:

[**http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html**](http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html)

****