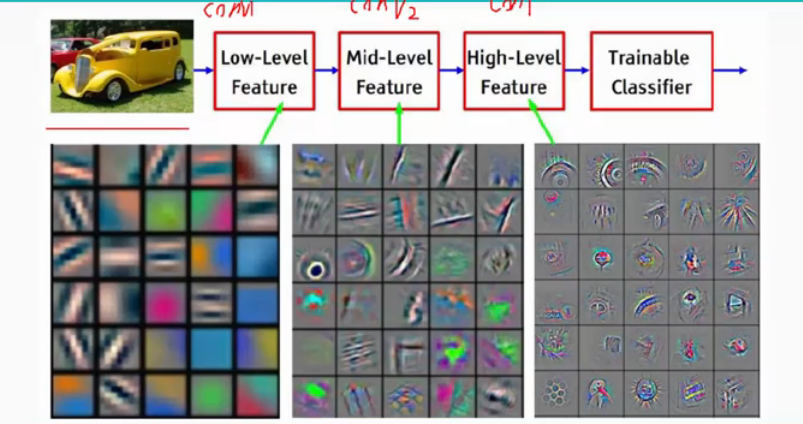
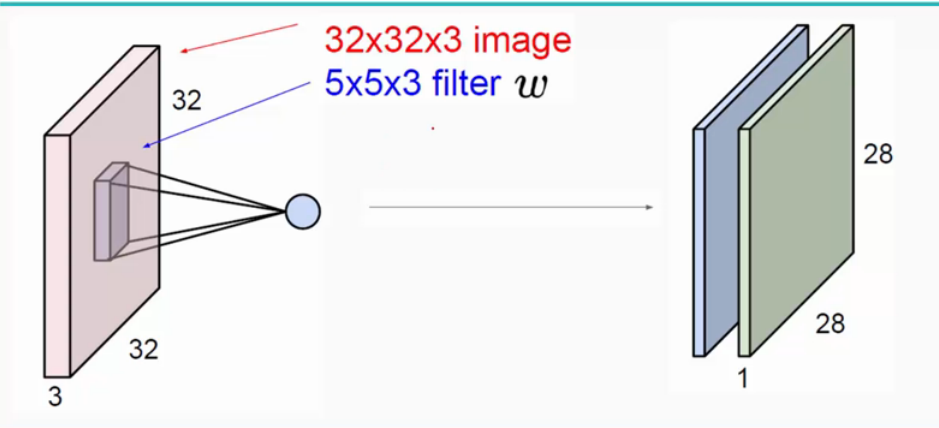
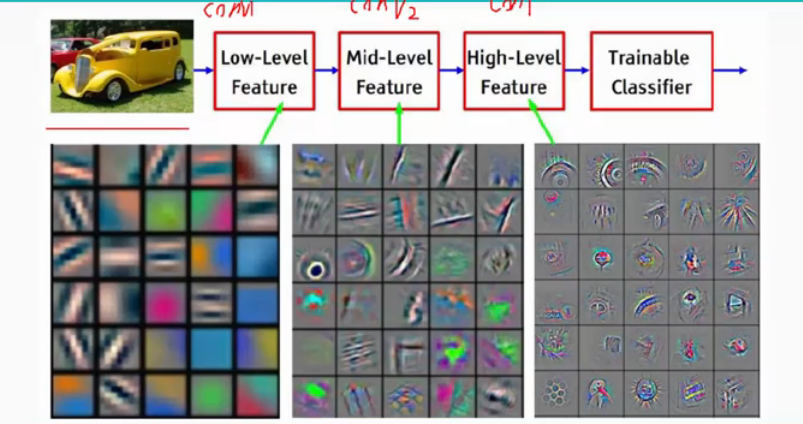
在提取信息的时候，在[池化](http://blog.yinfupai.com/tag/%E6%B1%A0%E5%8C%96)的时候，如果取区域均值（mean-pooling），往往能保留整体数据的特征，能凸出背景的信息，而如果取区域最大值（max-pooling），则能更好保留纹理上的特征，但这些应该都不如小波变换那样，可以保留更多的细节特征，整体上也应该更加细微。  
卷积神经网络（CNN）由输入层、卷积层、激活函数、池化层、全连接层组成，即INPUT-CONV-RELU-POOL-FC

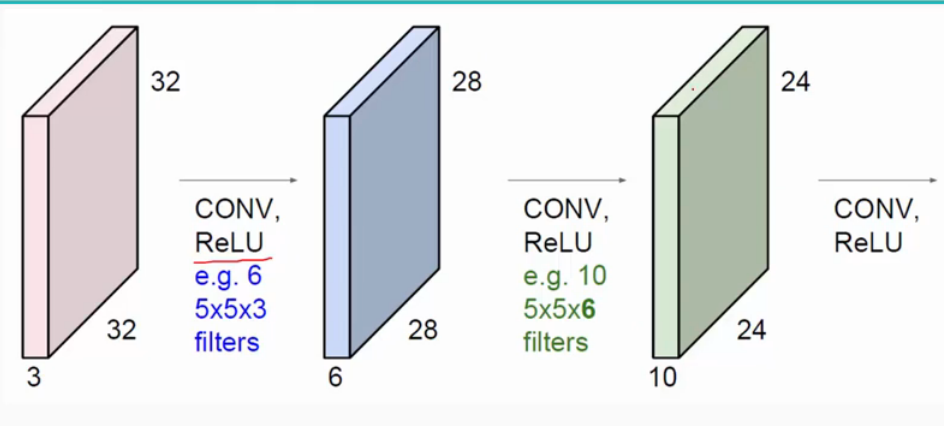
(1)卷积层：用它来进行特征提取，如下：



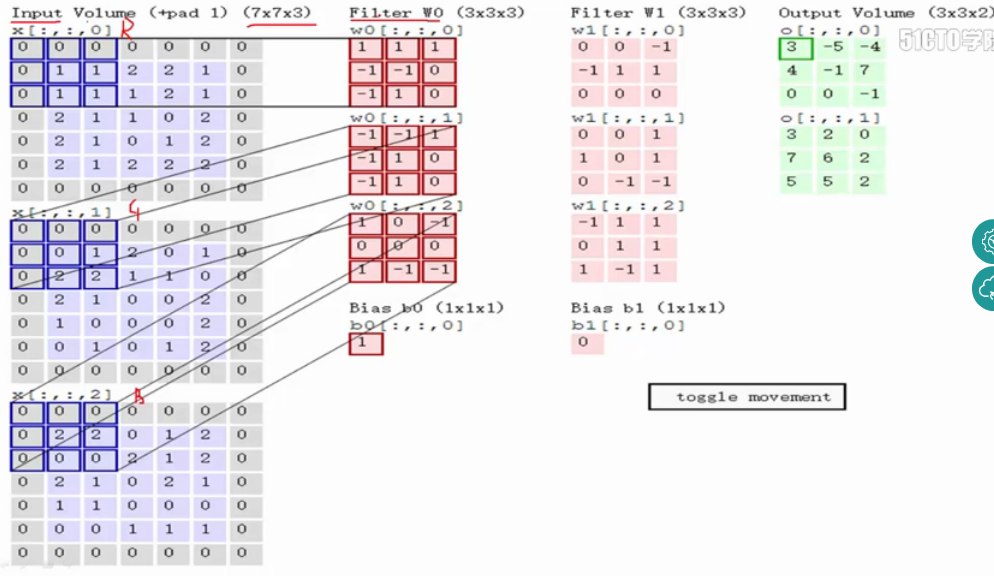
输入图像是32\*32\*3，3是它的深度（即R、G、B），卷积层是一个5\*5\*3的filter(感受野)，这里注意：感受野的深度必须和输入图像的深度相同。通过一个filter与输入图像的卷积可以得到一个28\*28\*1的特征图，上图是用了两个filter得到了两个特征图；

我们通常会使用多层卷积层来得到更深层次的特征图。如下：

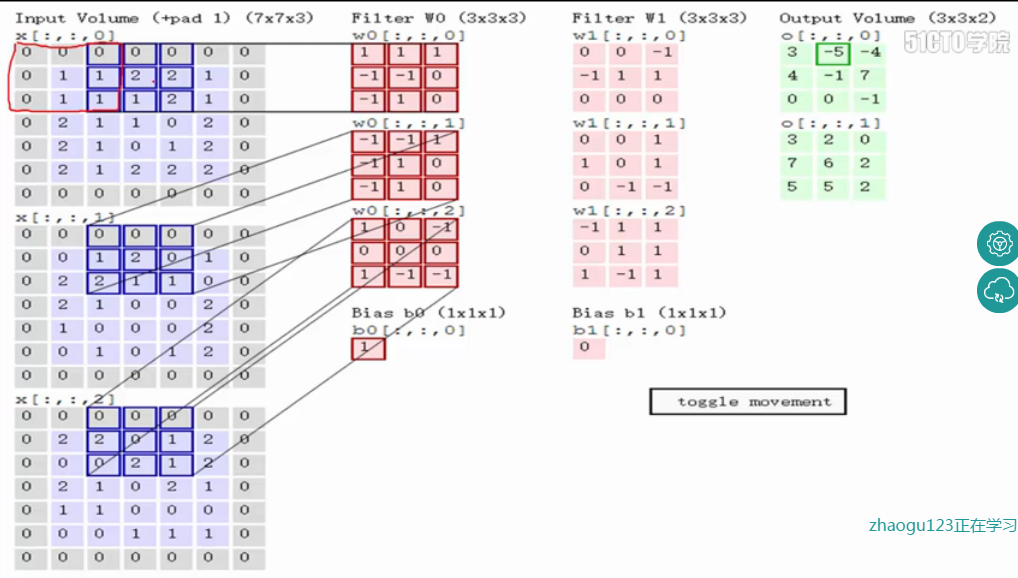




关于卷积的过程图解如下：



输入图像和filter的对应位置元素相乘再求和，最后再加上b,得到特征图。如图中所示，filter w0的第一层深度和输入图像的蓝色方框中对应元素相乘再求和得到0，其他两个深度得到2，0，则有0+2+0+1=3即图中右边特征图的第一个元素3.，卷积过后输入图像的蓝色方框再滑动，stride=2，如下：



如上图，完成卷积，得到一个3\*3\*1的特征图；在这里还要注意一点，即zero pad项，即为图像加上一个边界，边界元素均为0.（对原输入无影响）一般有

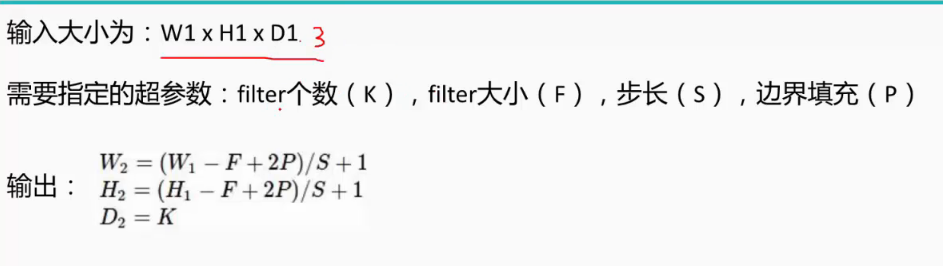
F=3 => zero pad with 1

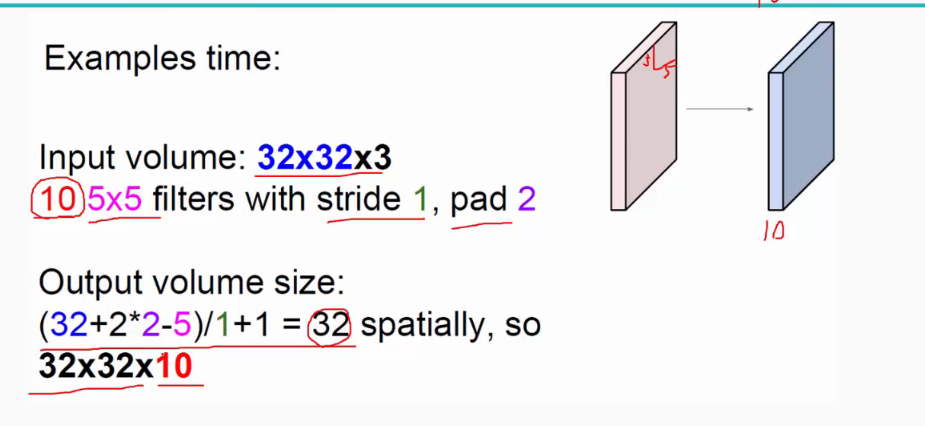
F=5 => zero pad with 2

F=7=> zero pad with 3,边界宽度是一个经验值，加上zero pad这一项是为了使输入图像和卷积后的特征图具有相同的维度，如：

输入为5\*5\*3，filter为3\*3\*3，在zero pad 为1，则加上zero pad后的输入图像为7\*7\*3，则卷积后的特征图大小为5\*5\*1（（7-3）/1+1），与输入图像一样；

而关于特征图的大小计算方法具体如下：

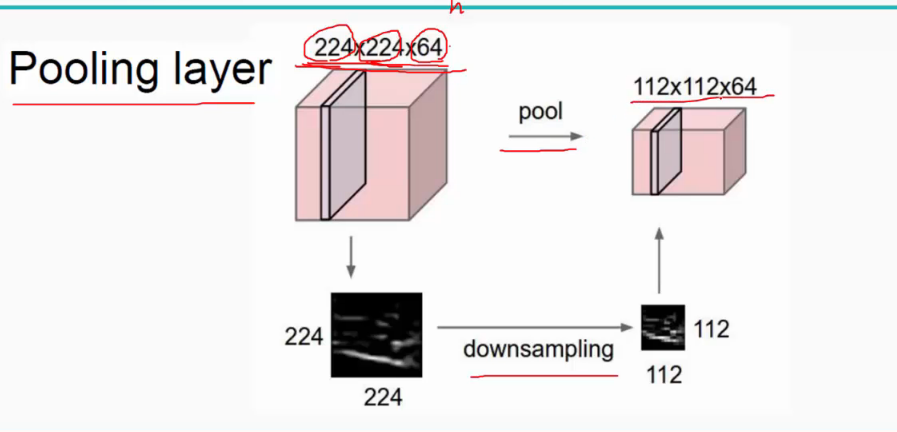




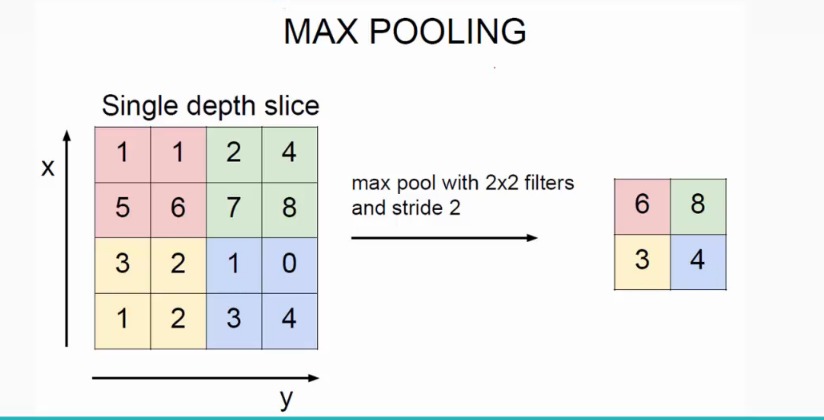
 卷积层还有一个特性就是“权值共享”原则。如下图：

如没有这个原则，则特征图由10个32\*32\*1的特征图组成，即每个特征图上有1024个神经元，每个神经元对应输入图像上一块5\*5\*3的区域，即一个神经元和输入图像的这块区域有75个连接，即75个权值参数，则共有75\*1024\*10=768000个权值参数，这是非常复杂的，因此卷积神经网络引入“权值”共享原则，即一个特征图上每个神经元对应的75个权值参数被每个神经元共享，这样则只需75\*10=750个权值参数，而每个特征图的阈值也共享，即需要10个阈值，则总共需要750+10=760个参数。

池化层：对输入的特征图进行压缩，一方面使特征图变小，简化网络计算复杂度；一方面进行特征压缩，提取主要特征，如下：



池化操作一般有两种，一种是Avy Pooling,一种是max Pooling,如下：



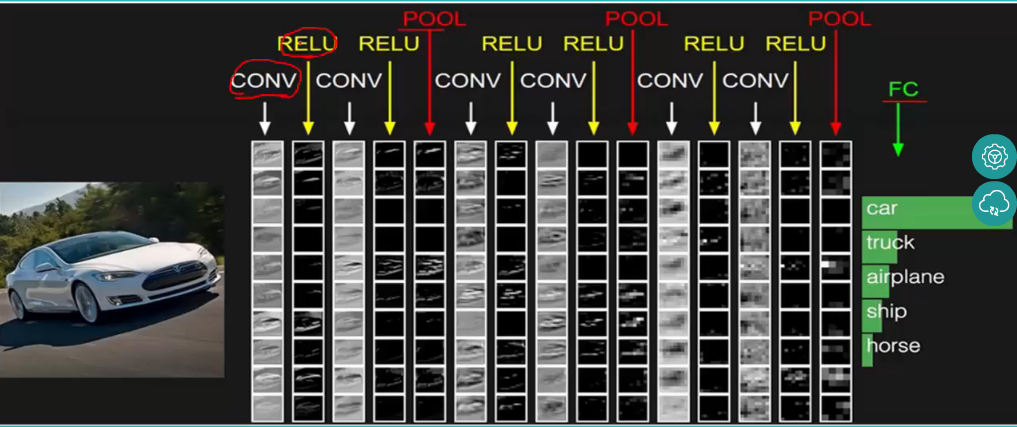
同样地采用一个2\*2的filter,max pooling是在每一个区域中寻找最大值，这里的stride=2,最终在原特征图中提取主要特征得到右图。

（Avy pooling现在不怎么用了，方法是对每一个2\*2的区域元素求和，再除以4，得到主要特征），而一般的filter取2\*2,最大取3\*3,stride取2，压缩为原来的1/4.

注意：这里的pooling操作是特征图缩小，有可能影响网络的准确度，因此可以通过增加特征图的深度来弥补（这里的深度变为原来的2倍）。

全连接层：连接所有的特征，将输出值送给分类器（如softmax分类器）。

总的一个结构大致如下：



另外：CNN网络中前几层的卷积层参数量占比小，计算量占比大；而后面的全连接层正好相反，大部分CNN网络都具有这个特点。因此我们在进行计算加速优化时，重点放在卷积层；进行参数优化、权值裁剪时，重点放在全连接层。