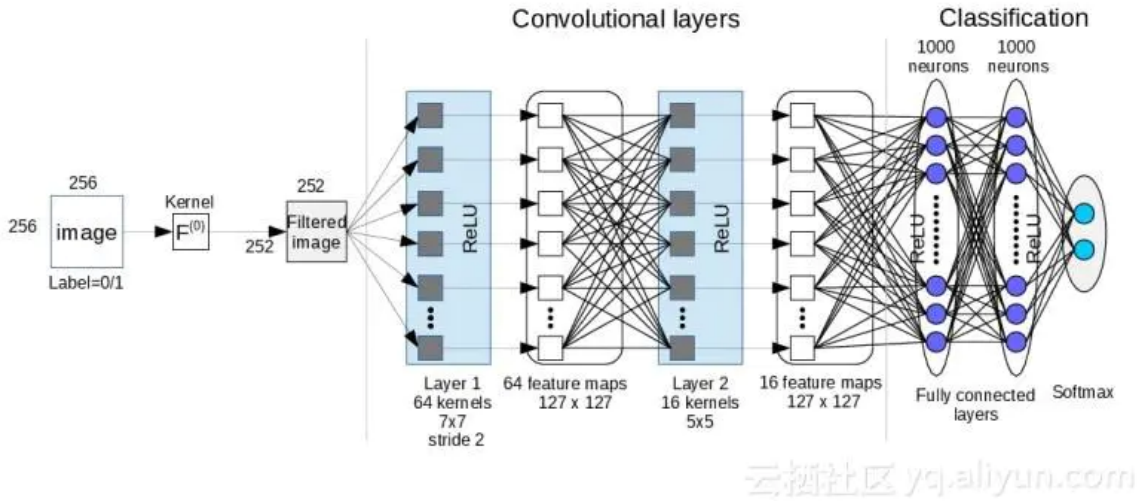
14.深度卷积神经网络

卷积神经网络的架构



如上图所示，卷积神经网络架构与常规人工神经网络架构非常相似，特别是在网络的最后一层，即全连接。此外，还注意到卷积神经网络能够**接受多个特征图**作为输入，而不是向量。

卷积神经网络的层级结构

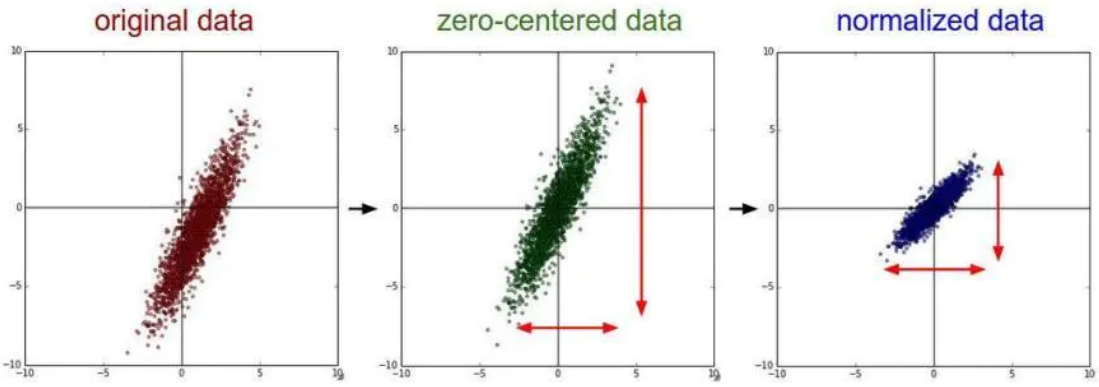
- 数据输入层/ Input layer
- 卷积计算层/ CONV layer
- ReLU激励层 / ReLU layer
- 池化层 / Pooling layer
- 全连接层 / FC layer

1.数据输入层

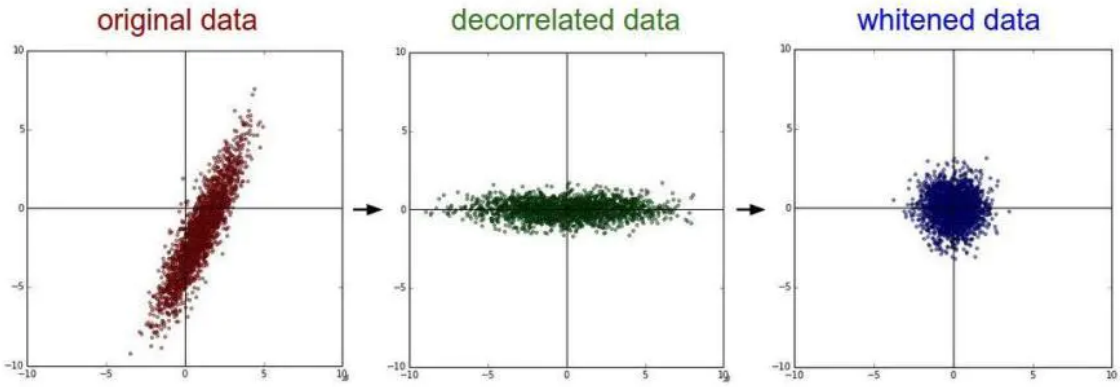
该层要做的处理主要是对原始图像数据进行预处理，其中包括：

- 去均值：把输入数据各个维度都中心化为0，如下图所示，其目的就是把样本的中心拉回到坐标系原点上。
- 归一化：幅度归一化到同样的范围，如下所示，即减少各维度数据取值范围的差异而带来的干扰，比如，我们有两个维度的特征A和B，A范围是0到10，而B范围是0到10000，如果直接使用这两个特征是有问题的，好的做法就是归一化，即A和B的数据都变为0到1的范围。
- PCA/白化：用PCA降维；白化是对数据各个特征轴上的幅度归一化

去均值与归一化效果图：



去相关与白化效果图：



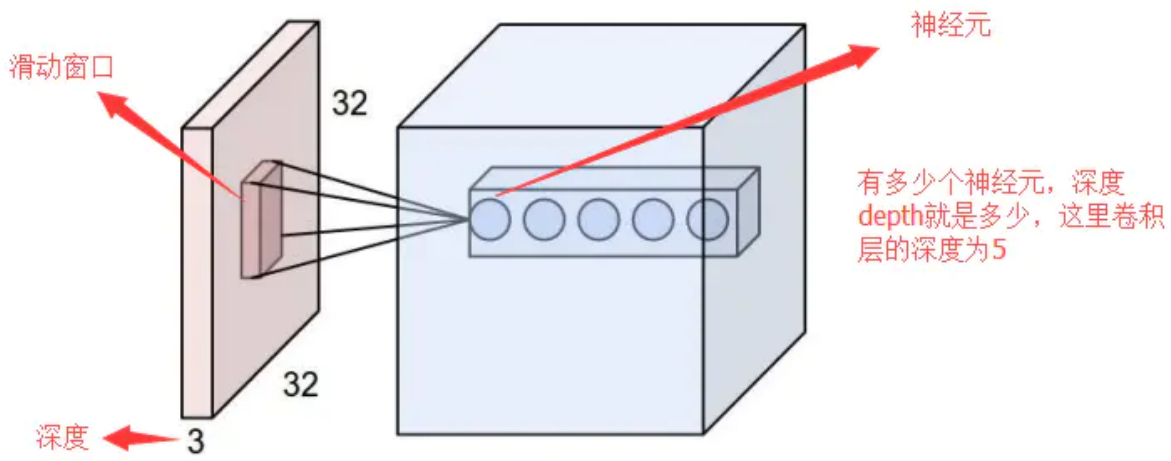
2.卷积计算层

这一层就是卷积神经网络最重要的一个层次，也是“卷积神经网络”的名字来源。
在这个卷积层，有两个关键操作：

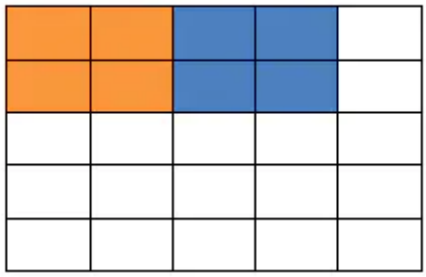
- 局部关联。每个神经元看做一个滤波器(filter)
- 窗口(receptive field)滑动， filter对局部数据计算

先介绍卷积层遇到的几个名词：

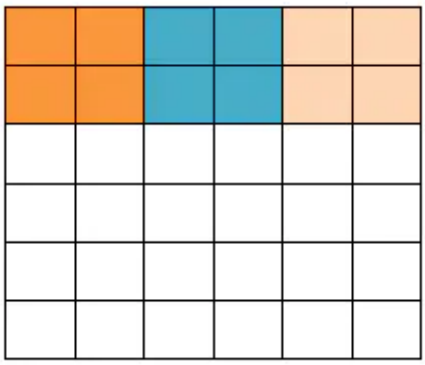
- 深度/depth（解释见下图）
- 步长/stride （窗口一次滑动的长度）
- 填充值/zero-padding



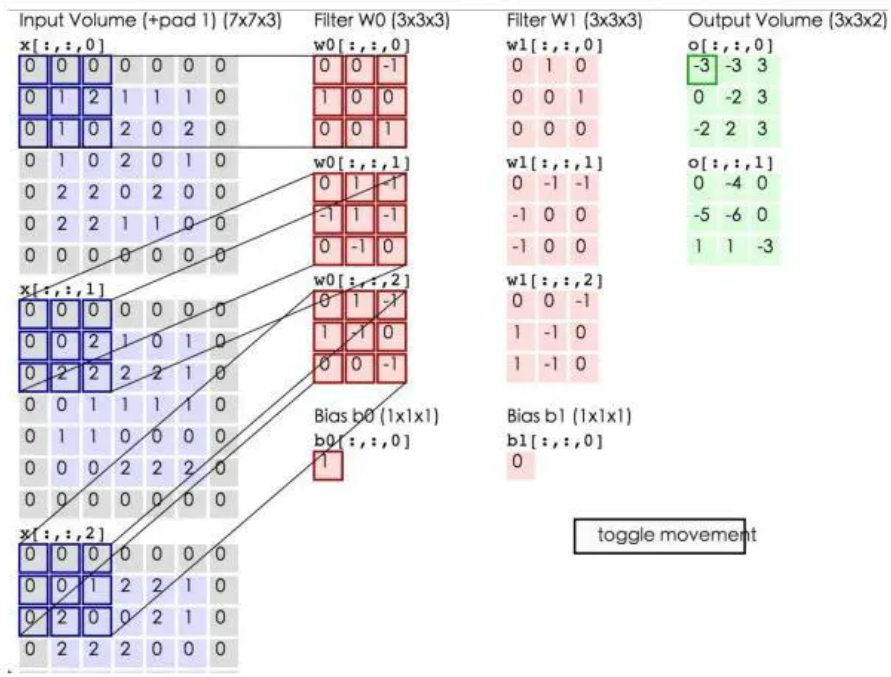
填充值是什么呢？以下图为例子，比如有这么一个5 * 5的图片（一个格子一个像素），我们滑动窗口取2*2，步长取2，那么我们发现还剩下1个像素没法滑完，那怎么办呢？



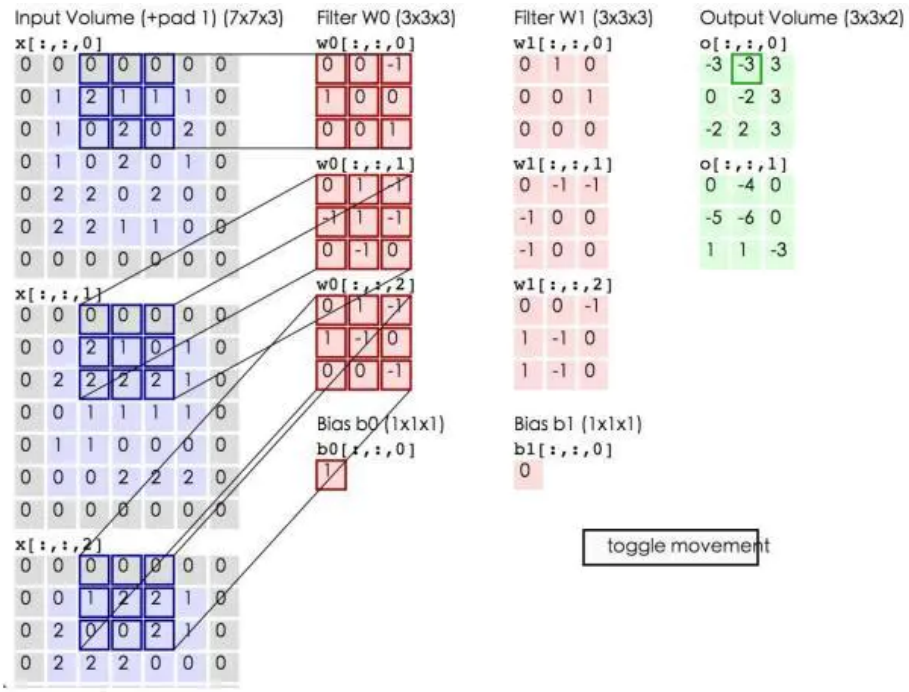
那我们在原先的矩阵加了一层填充值，使得变成6*6的矩阵，那么窗口就可以刚好把所有像素遍历完。这就是填充值的作用。



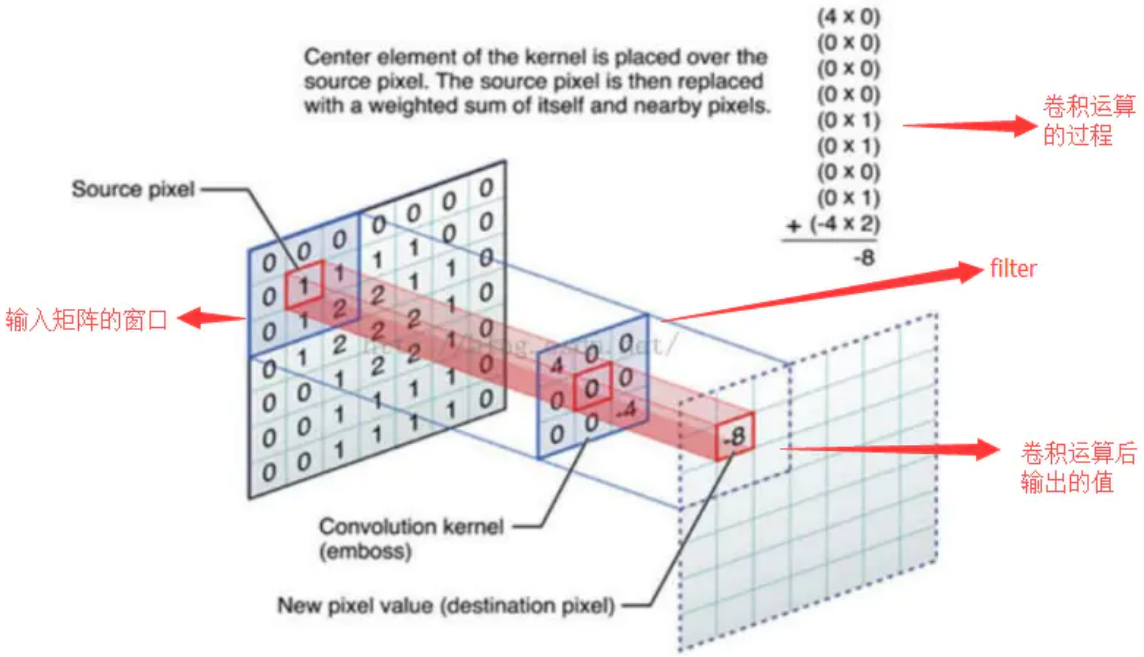
卷积的计算（注意，下面蓝色矩阵周围有一圈灰色的框，那些就是上面所说到的填充值）



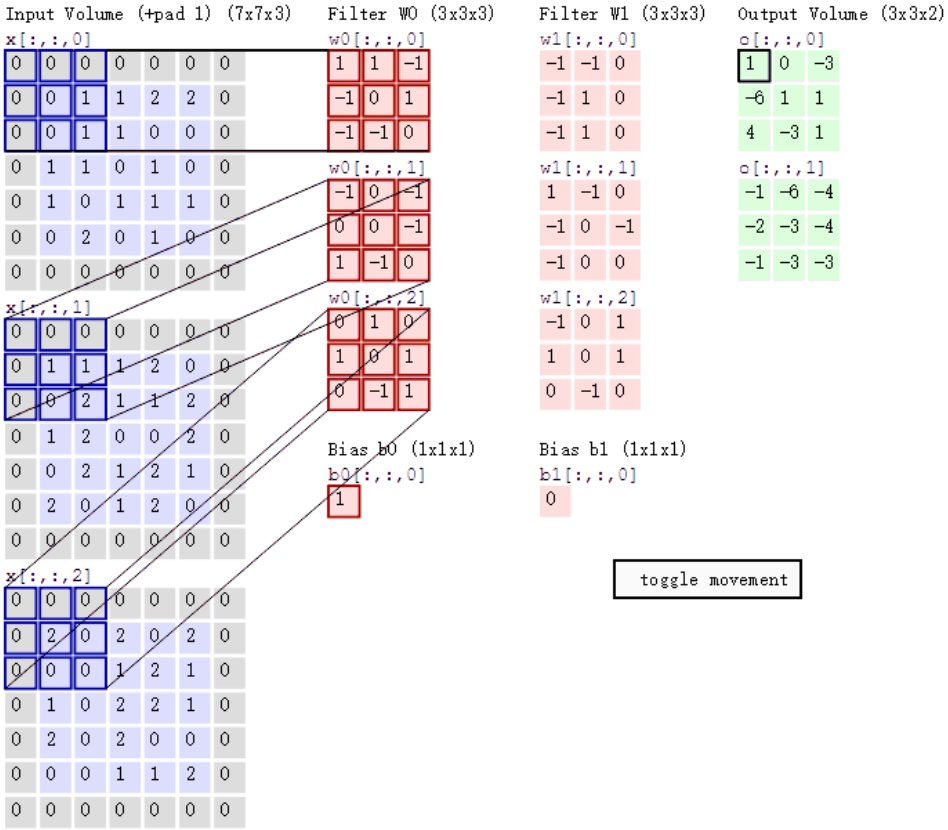
这里的蓝色矩阵就是输入的图像，粉色矩阵就是**卷积层的神经元**，这里表示了有两个神经元（w0,w1）。**绿色矩阵就是经过卷积运算后的输出矩阵**，这里的步长设置为2。



蓝色的矩阵(输入图像)对粉色的矩阵（filter）进行矩阵内积计算并将三个内积运算的结果与偏置值b相加（比如上面图的计算：2+（-2+1-2）+（1-2-2） + 1= 2 - 3 - 3 + 1 = -3），计算后的值就是绿框矩阵的一个元素。

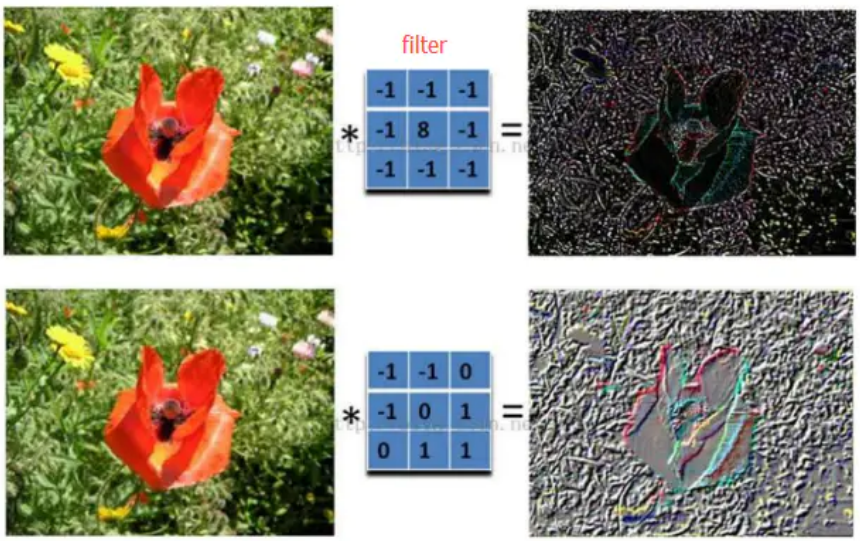


下面的动态图形象地展示了卷积层的计算过程：



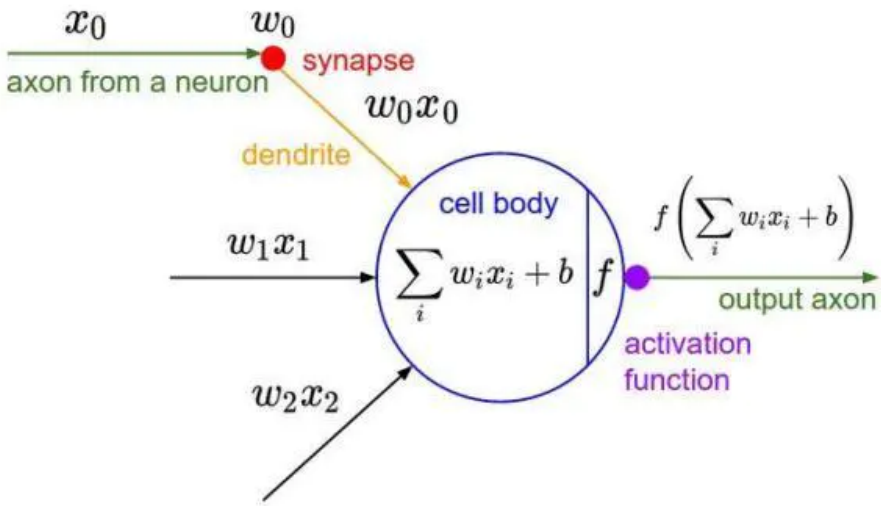
参数共享机制

- 在卷积层中每个神经元连接数据窗的权重是固定的，每个神经元只关注一个特性。神经元就是图像处理中的滤波器，比如边缘检测专用的Sobel滤波器，即卷积层的每个滤波器都会有自己所关注一个图像特征，比如垂直边缘，水平边缘，颜色，纹理等等，这些所有神经元加起来就好比就是整张图像的特征提取器集合。
- 需要估算的权重个数减少: AlexNet 1亿 => 3.5w
- 一组固定的权重和不同窗口内数据做内积: 卷积

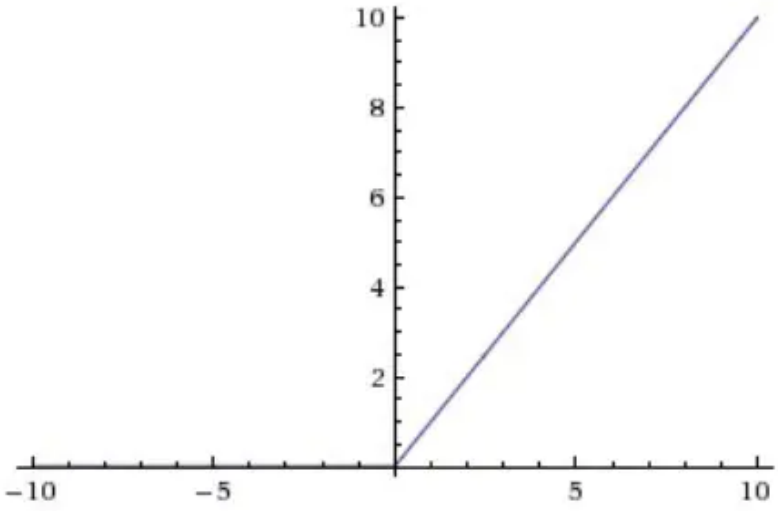


3.激励层

把卷积层输出结果做非线性映射。



CNN采用的激励函数一般为ReLU(The Rectified Linear Unit/修正线性单元)，它的特点是收敛快，求梯度简单，但较脆弱，图像如下。



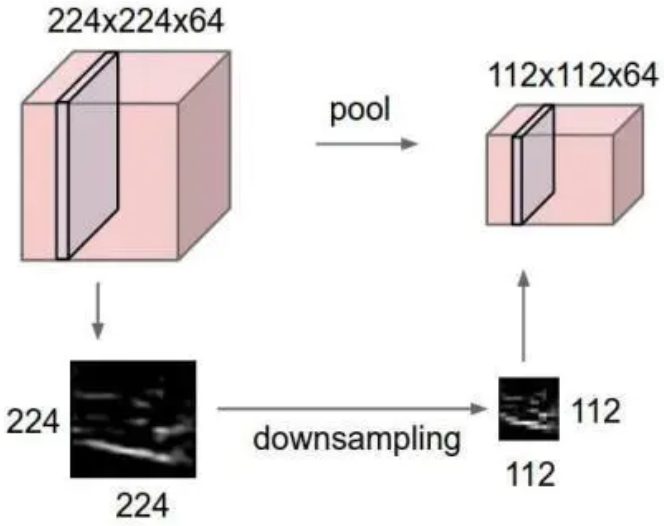
激励层的实践经验：
①不要用sigmoid！不要用sigmoid！不要用sigmoid！
② 首先试RELU，因为快，但要小心点
③ 如果2失效，请用Leaky ReLU或者Maxout
④ 某些情况下tanh倒是有不错的结果，但是很少

4.池化层

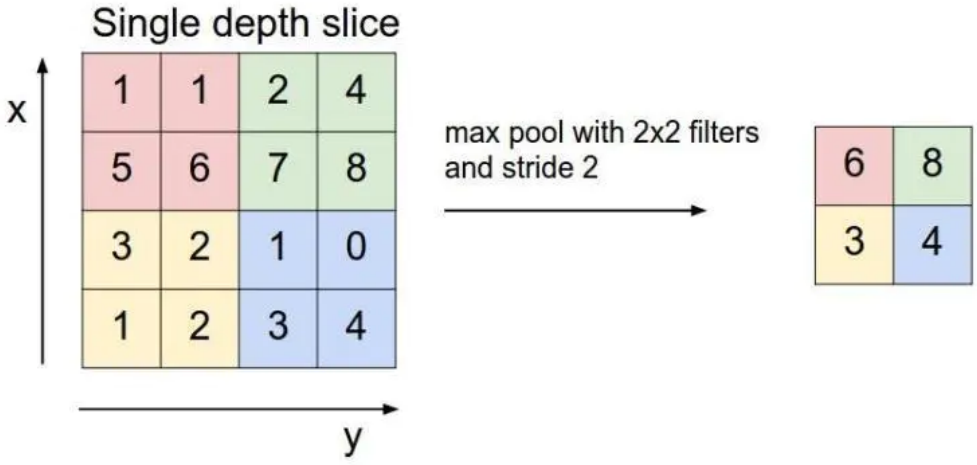
池化层夹在连续的卷积层中间， 用于压缩数据和参数的量，减小过拟合。
简而言之，**如果输入是图像的话，那么池化层的最主要作用就是压缩图像。**

这里再展开叙述池化层的具体作用。

1. 特征不变性，也就是我们在图像处理中经常提到的特征的尺度不变性，池化操作就是图像的resize，平时一张狗的图像被缩小了一倍我们还能认出这是一张狗的照片，这说明这张图像中仍保留着狗最重要的特征，我们一看就能判断图像中画的是一只狗，图像压缩时去掉的信息只是一些无关紧要的信息，而留下的信息则是具有尺度不变性的特征，是最能表达图像的特征。
2. 特征降维，我们知道一幅图像含有的信息是很大的，特征也很多，但是有些信息对于我们做图像任务时没有太多用途或者有重复，我们可以把这类冗余信息去除，把最重要的特征抽取出来，这也是池化操作的一大作用。
3. 在一定程度上防止过拟合，更方便优化。



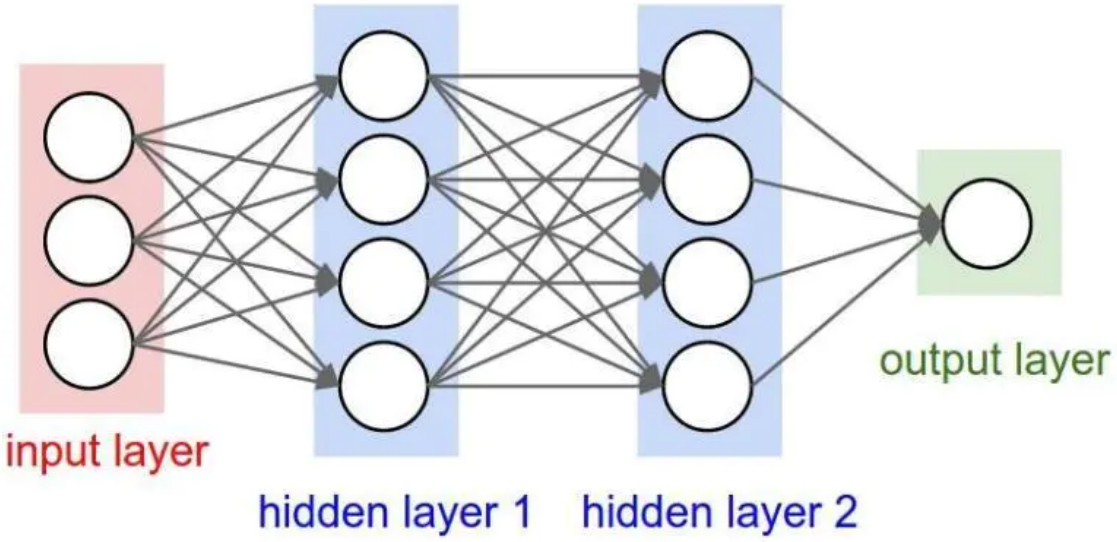
池化层用的方法有Max pooling 和 average pooling，而实际用的较多的是Max pooling。
这里就说一下Max pooling，其实思想非常简单。



对于每个2 * 2的窗口选出最大的数作为输出矩阵的相应元素的值，比如输入矩阵第一个2 * 2窗口中最大的数是6，那么输出矩阵的第一个元素就是6，如此类推。

5.全连接层

两层之间所有神经元都有权重连接，通常全连接层在卷积神经网络尾部。也就是跟传统的神经网络神经元的连接方式是一样的：



一般CNN结构依次为

1.INPUT
2.[[CONV -> RELU]*N* -> *POOL?*]M
3.[FC -> RELU]*K
4.FC

卷积神经网络之训练算法

1.同一般机器学习算法，先定义Loss function，衡量和实际结果之间差距。
2.找到最小化损失函数的W和b， CNN中用的算法是SGD（随机梯度下降）。

卷积神经网络之优缺点

优点
•共享卷积核，对高维数据处理无压力
•无需手动选取特征，训练好权重，即得特征分类效果好
缺点
•需要调参，需要大样本量，训练最好要GPU
•物理含义不明确（也就说，我们并不知道没个卷积层到底提取到的是什么特征，而且神经网络本身就是一种难以解释的“黑箱模型”）

术语解释

池化（pooling）

见池化层

dropout

Dropout可以作为训练深度神经网络的一种trick供选择。在每个训练批次中，通过忽略一半的特征检测器（让一半的隐层节点值为0），可以明显地减少过拟合现象。这种方式可以减少特征检测器（隐层节点）间的相互作用，检测器相互作用是指某些检测器依赖其他检测器才能发挥作用。

Dropout说的简单一点就是：我们在前向传播的时候，让某个神经元的激活值以一定的概率p停止工作，这样可以使模型泛化性更强，因为它不会太依赖某些局部的特征，如图1所示。

当前Dropout被大量利用于全连接网络，而且一般认为设置为0.5或者0.3，而在卷积网络隐藏层中由于卷积自身的稀疏化以及稀疏化的ReLu函数的大量使用等原因，Dropout策略在卷积网络隐藏层中使用较少。总体而言，Dropout是一个超参，需要根据具体的网络、具体的应用领域进行尝试。