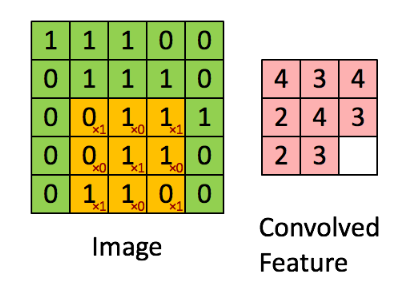
<https://blog.csdn.net/u010089444/article/details/52563213?locationNum=9>

# 1 卷积

**1.1 卷积**

如下图所示，展示了一个3×3的卷积核在5×5的图像上做卷积的过程。左侧的绿色矩阵代表输入的一张图片（例如一张5px\*5px的图片），3×3阶的黄色矩阵就是卷积核，用来提取图像的一种特征，让卷积核在图像上进行从左到右，从上到下移动。每移动一次，让卷积核与相应的图像局部区域做点积，就得到了右边矩阵中对应的一个元素。



每个卷积都是一种特征提取方式，通过一种特征提取可得到一个feature map。上图右边的3×3矩阵就是一个feature map，使用不同的卷积核可得到多种feature map

**1.2 滑动的步长(stride)**

步长(stride)决定了卷积核在移动过程中一次跳过几格。上面那张图片从左到右，每次滑动的时候只移动一格。

**1.3 卷积的边界处理(padding)**

tensorflow中卷积的 padding参数可以设为两个值SAME，VALID

VALID模式如上图所示，对原始图像进行卷积，卷积后的矩阵只有3×3阶，比原来的图片要小了

SAME模式要求卷积后的feature map与输入的矩阵大小相同，因此需要对输入矩阵的外层包裹n层0，然后再按照VALID的卷积方法进行卷积。n的求法如下式：

* SAME：edge\_row = (kernel\_row - 1) / 2; edge\_cols = (kernel\_cols - 1) / 2;
* VALID：edge\_row = edge\_cols = 0;

其中，edge\_row是包裹0的行数，edge\_cols是包裹0的列数 , kernel\_row就卷积核的行数

# 2 池化(Pooling)

通过将卷积层提取到的特征输入至分类器中进行训练，可以实现输出最终的分类结果。理论上可以直接将卷积层提取到的所有特征输入至分类器中，然而这将需要非常大的计算开销。例如：对于一个输入为96×96大小的图像样本，假设在卷积层使用200个8×8大小的卷积核对该输入图像进行卷积运算操作，每个卷积核都输出一个（96-8＋1）×（96-8＋1）=7921维的特征向量，最终卷积层将输出一个7921×200=1584200维的特征向量。将如此高维度的特征输入至分类器中进行训练需要耗费非常庞大的计算资源，同时也会产生严重的过拟合问题。

由于图像具有 “静态性”，在图像的一个局部区域得到的特征极有可能在另一个局部区域同样适用。因此，可以对图像的一个局部区域中不同位置的特征进行聚合统计，这种操作称为“池化”(也称作子采样)。池化分为两种，一种是最大池化，在选中区域中找最大的值作为Pooling后的值，另一种是平均值池化，把选中的区域中的平均值作为Pooling后的值。

