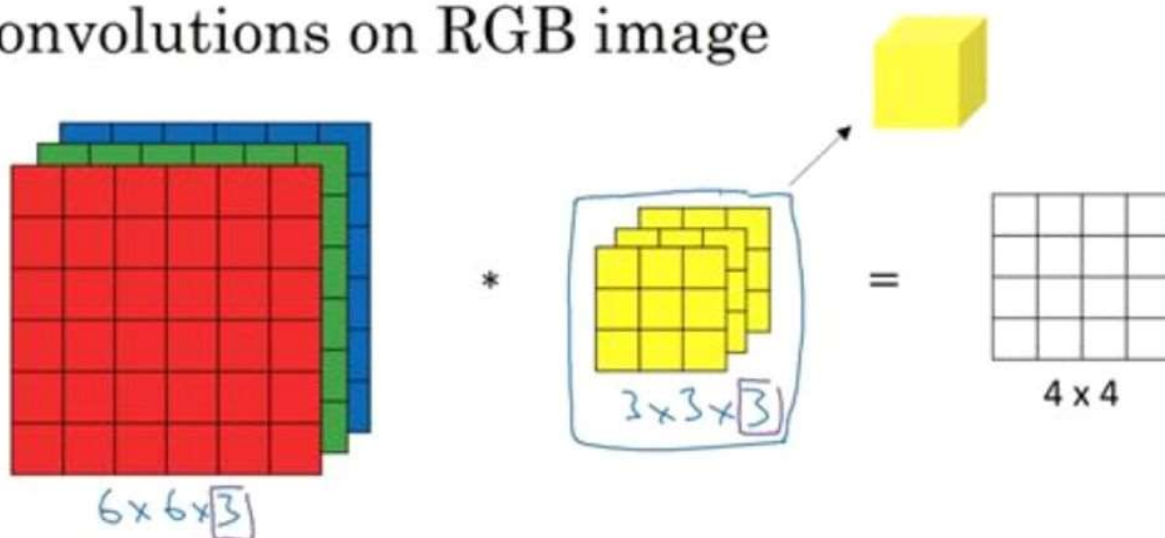


三维卷积、池化与全连接

在上一讲中，我们对卷积神经网络中的卷积操作进行了详细的解读，但仅仅是对二维图像，即通道数为 1 的图像（灰度图）进行了卷积，对于三维或者更高维的图像卷积显然并没有这么简单。

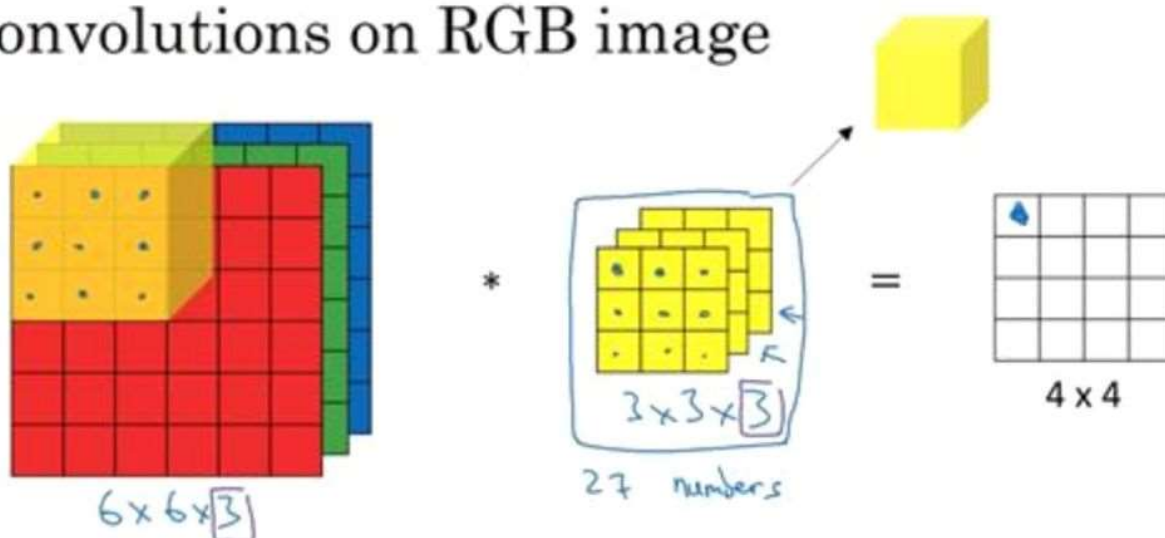
如果图像大小为 $6 \times 6 \times 3$ ，那么这里的 3 指的是三个颜色通道，你可以将其理解为 3 个 6×6 的图像的堆叠，如果要对此图像进行卷积的话，我们的滤波器也需要变成三维的，假设采用一个 $3 \times 3 \times 3$ 的滤波器对其进行卷积（这里需要注意的是，滤波器的通道数一定要等于输入图像的通道数），根据 $((n+2p-f)/s)+1$ 的计算公式我们可以知道输出图像的大小为 $4 \times 4 \times 1$ ，由一个滤波器进行卷积之后输出图像的通道不是 3 而是 1 了。

Convolutions on RGB image



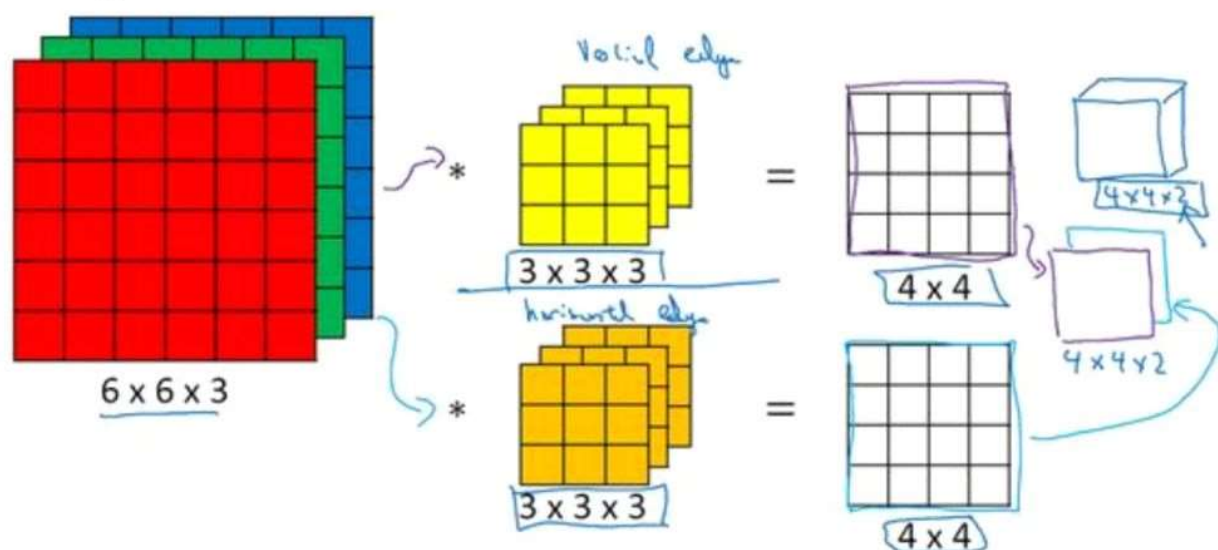
那么究竟如何理解三通道的卷积过程？单通道的卷积操作我们已经知道，就是直接对感受野与滤波器进行元素相乘求和，那三通道呢？我们可以将 $3 \times 3 \times 3$ 的滤波器想象为一个三维的立方体，为了计算立方体滤波器在输入图像上的卷积操作，我们首先将这个三维的滤波器放到左上角，让三维滤波器的 27 个数依次乘以红绿蓝三个通道中的像素数据，即滤波器的前 9 个数乘以红色通道中的数据，中间 9 个数乘以绿色通道中的数据，最后 9 个数乘以蓝色通道中的数据。将这些数据加总起来，就得到输出像素的第一个元素值。如下图所示：

Convolutions on RGB image



这样做可能还不够。在实际图像处理中，仅靠单滤波器很难将复杂的图像特征进行充分提取。所以，通常而言，我们用来进行卷积的滤波器远不止一个。下面我们用两个滤波器对上述图像进行卷积操作。

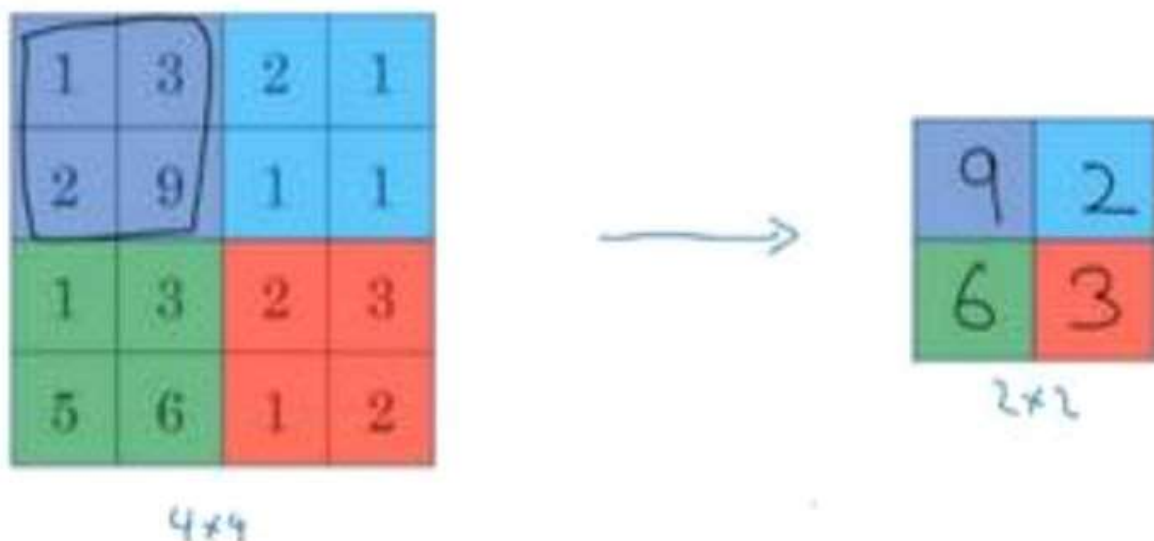
Multiple filters



我们用第一个滤波器进行卷积得到一个 4*4 的输出，然后用第二个滤波器进行卷积又得到一个 4*4 的输出，将这个输出放到第一个输出后面，形成了一个 4*4*2 的输出，这里的 2 的就是滤波器的个数。

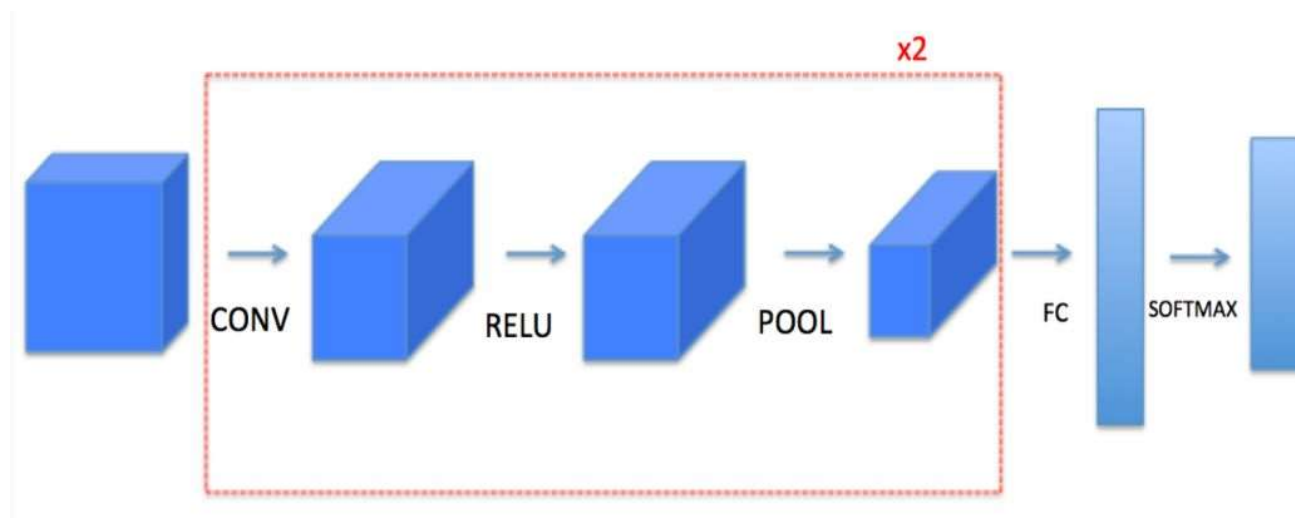
关于多通道卷积的另一种理解类似于 DNN 中的标准神经网络中的 $Z=Wx+b$ 的线性计算，其中 x 为输入的原始图像， W 为滤波器， b 偏差项，卷积过程就类似与标准神经网络的线性计算，完了之后利用激活函数进行激活。还有一个关键问题就是卷积层有多少参数，卷积层的参数通常在于滤波器 W ，根据滤波器的大小，我们可以计算一个滤波器的参数数量为 $f*f*nc$ ，其中 nc 为通道数，那 k 个滤波器所含的参数数量则为 $f*f*nc*k$ 。

卷积过程到此就差不多解释完了。我们再看 CNN 的另一个重要操作——池化 (pooling)。简单来说，池化层是用来缩减模型大小，提高模型计算速度以及提高所提取特征的鲁棒性。池化操作通常有两种，一种是常用的最大池化 (max pooling)，另一种是不常用的平均池化 (average pooling)。池化操作过程也非常简单，假设池化层的输入为一个 4*4 的图像，我们用最大池化对其进行池化，执行最大池化的核是一个 2*2 的矩阵，执行过程就是将输入矩阵拆分为不同区域，对于 2*2 的输出而言，输出的每个元素都是其对应区域的最大元素值。



最大池化过程就像是应用了一个 2×2 的滤波器以步幅 2 进行区域最大化输出操作。所以简单而言，池化的参数就是滤波器的大小 f 和步幅 s ，池化的效果就相当于对输入图像的高度和宽度进行缩小。值得注意的是，最大池化只是计算神经网络某一层的静态属性，中间并没有什么学习过程。

池化完成之后就是标准神经网络中的全连接层了。全连接层我们在 DNN 中有详细介绍，这里就不再赘述。总之，一个典型的卷积层通常包括卷积层-池化层和全连接层。



下一讲中，我们将一起尝试如何利用 `numpy` 实现一个简单的典型的卷积神经网络。

作者：**鲁伟**，热爱数据，坚信数据技术和代码改变世界。R语言和Python的忠实拥趸，为成为一名未来的数据科学家而奋斗终生。

个人公众号：机器学习实验室（微信ID: louwill12）

参考资料：

<https://www.coursera.org/learn/machine-learning>

<https://www.deeplearning.ai/>