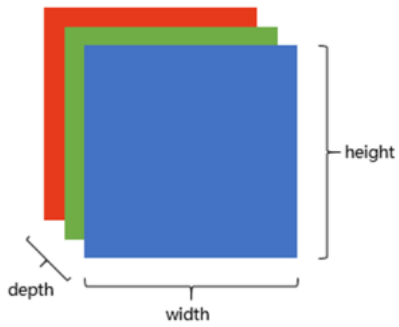


卷积神经网络

一、卷积神经网络的优势

卷积神经网络主要用于图像识别、目标检测、图像生成等领域，其对比全连接神经网络的优势在哪里呢？

首先我们考虑一下图像的原理，一张灰度图像实际上在计算机中是一堆按顺序排列的数字，数值为0到255，0表示最暗，255表示最亮；而对于RGB图像实际上就是三个颜色的矩阵数据，每个矩阵就叫这个图片的通道，如下图所示：

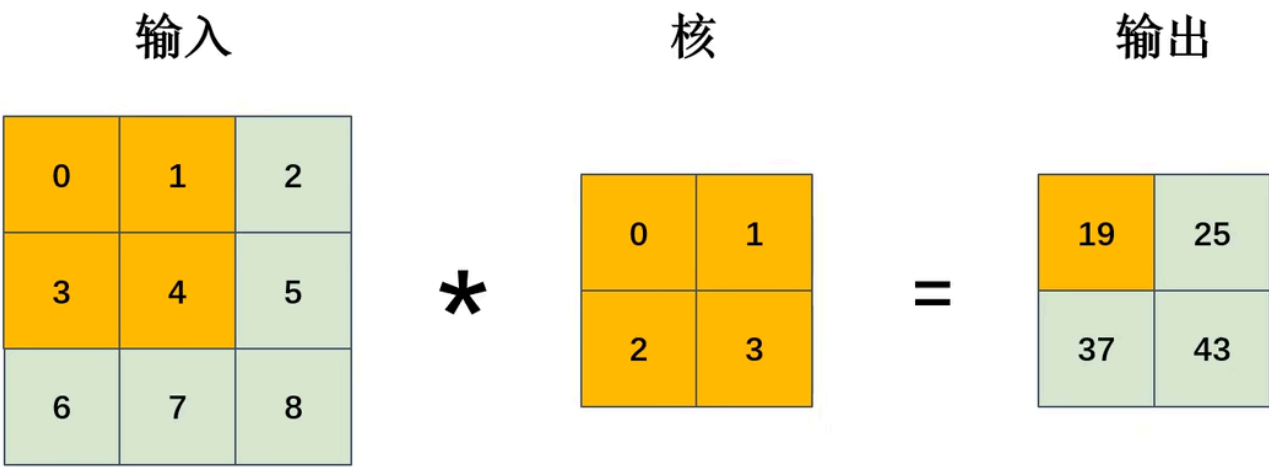


在全连接神经网络中，我们需要的输入是一维的向量，而图片的每一个通道都是二维的矩阵数据，如果直接将数据展开为一维度的向量就会丢失图像中物体的位置关系，因为图像的一个像素点不仅具有数值的信息，还与周围的像素点存在位置距离信息，因此我们引入卷积神经网络来捕捉图像中的局部位置信息。

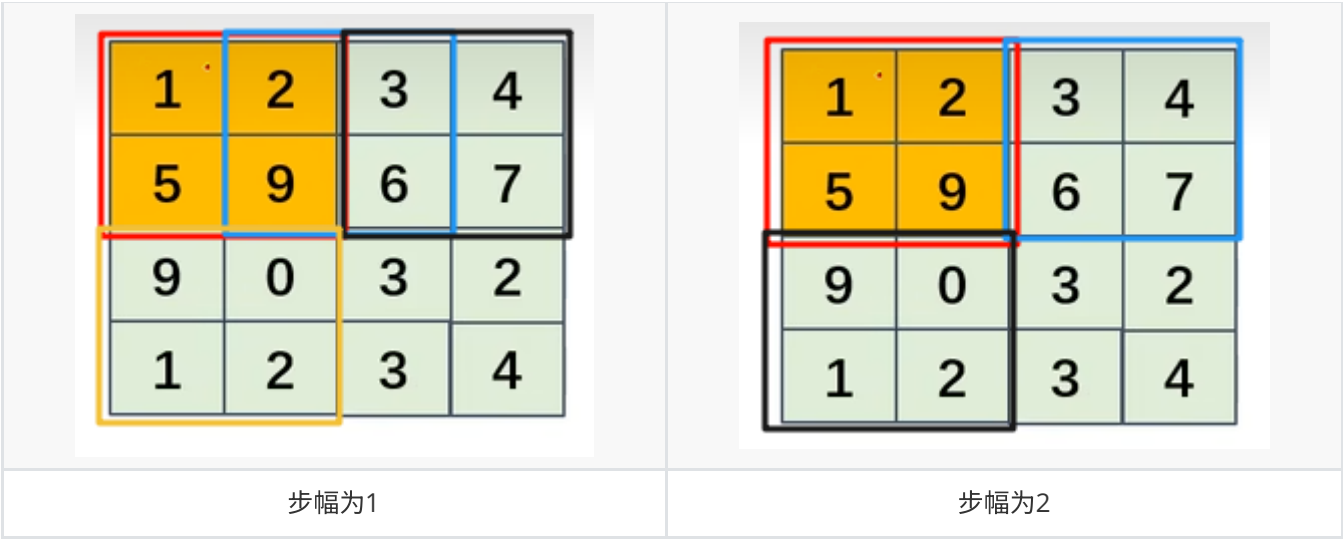
二、卷积运算和池化运算

1.卷积运算

在卷积神经网络中，卷积操作是指将一个可移动的小窗口（一组固定的权重）与图像进行逐元素相乘然后相加的操作，这个矩形小窗口被称作卷积核（一般为N阶的方阵）。具体的运算过程如下图所示：



对于黄色区域的计算就是： $0 * 0 + 1 * 1 + 3 * 2 + 3 * 4 = 19$,接下来将黄色小窗口按照步幅依次划过整个输入矩阵数据进行计算即可得到卷积输出矩阵。**步幅就是小窗口单次移动的距离**，下图显示了不同的步幅滑动：



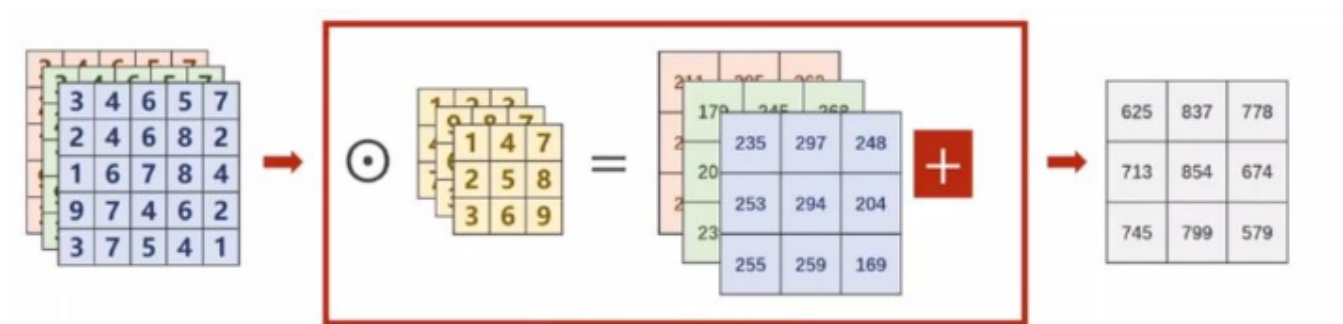
显然经过卷积运算后数据的（大小）高和宽就发生了变化，具体计算经过卷积运算后特征图大小的公式如下：

$$\begin{cases} OH = \frac{H+2P-FH}{S} + 1 \\ OW = \frac{W+2P-FW}{S} + 1 \end{cases} \tag{1}$$

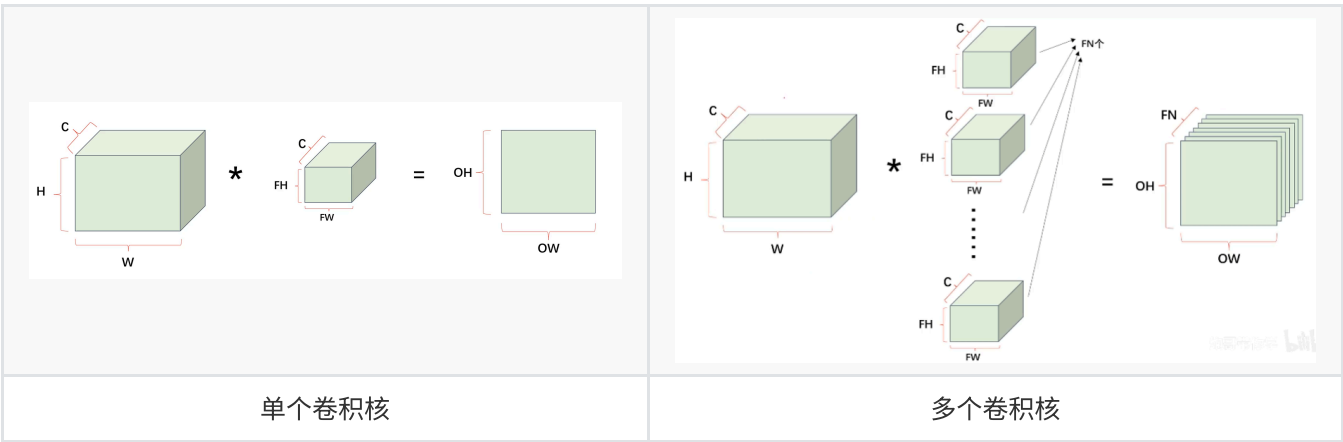
其中H、W为输入图像的高和宽，P为填充的大小，FH、FW为卷积核的高和宽，OH、OW为输出图像的高和宽，S为步幅。这里填充值就是在外围边缘补充若干0，方便从初始位置以步长为单位可以刚好滑到末尾位置，通俗地讲就是为了总长能被步长整除。填充后外围的填充值由于是0，因此不会改变图像原有的局部特征。

同时卷积运算也可以加入偏置，只需要将计算出的每一项数据都加上偏置就行。

对于多通道的卷积操作，具体过程是输入一张三通道的图片，这时有多个卷积核进行卷积，每个卷积核都有三通道，分别对这张输入图片的三通道进行卷积操作然后对应位置累加求和得到一个卷积输出结果，有多少个卷积核，就有多少个卷积输出结果。如果只有一个卷积和的具体计算过程：



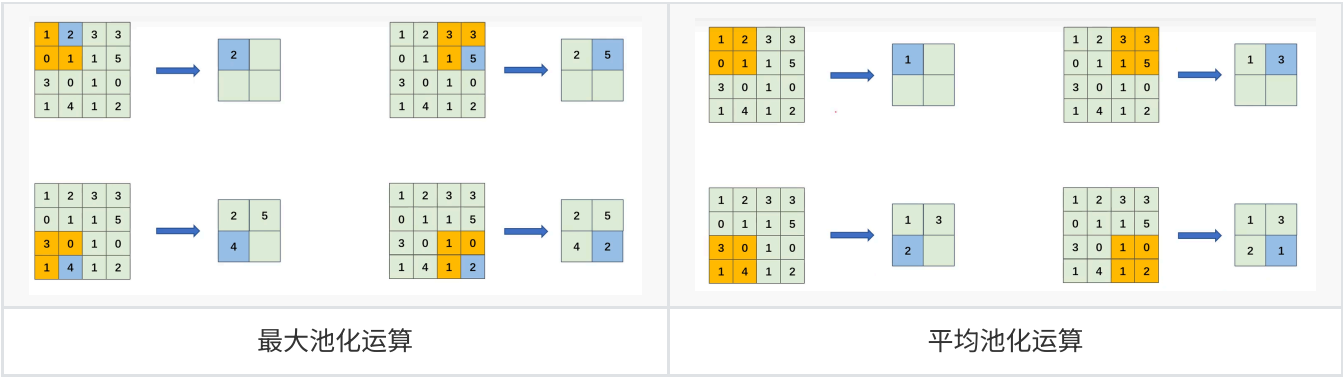
下面是有一个和多个卷积核的立体堆叠表示，其中C是通道数，FN是卷积核个数：



通过卷积的操作，模型提取到了输入图像的局部特征，不同的卷积核大小和参数可以提取到不同层次的特征，同时由于卷积操作存在参数共享，因此相较于全连接神经网络降低了参数数量，对于图像数据的处理更快。

2.池化运算

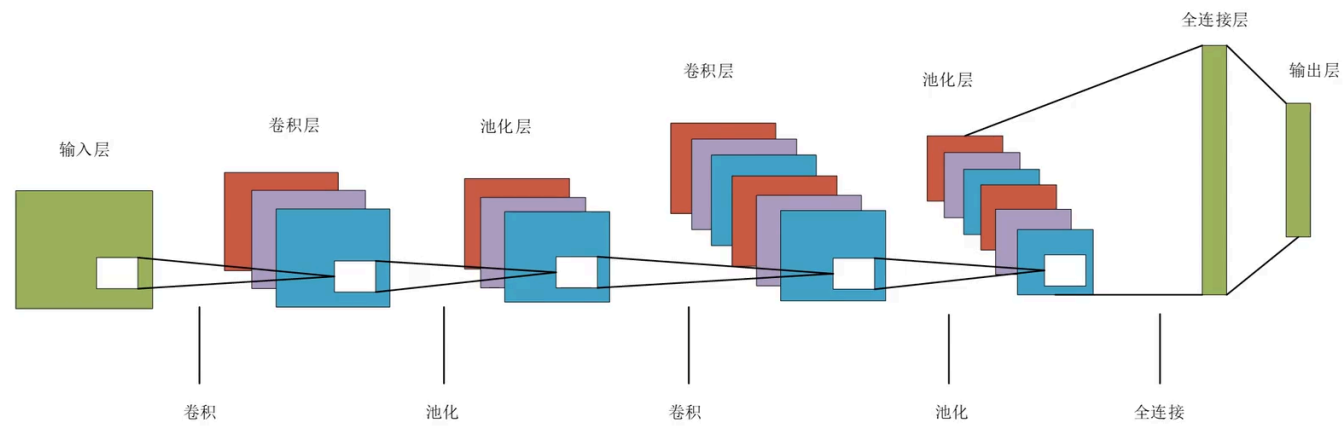
池化操作是卷积神经网络中的一个特殊的操作，主要就是在一定的区域内提出该区域的关键信息，通常分为最大池化运算和平均池化运算，最大池化是保留池化窗口内信息的最大值，这是在提取纹理特征，保留更多的局部细节。平均池化是对池化窗口中的数值进行均值化操作，这能保留模板内的数据的整体特征从而背景信息，具体运算如下图所示：



与卷积操作一样，池化操作也有池化核和步幅两个关键概念。经过池化运算后特征图的大小计算公式也和卷积操作一致。不同的是池化操作不会改变通道数，池化操作是按通道独立进行计算的。池化运算对卷积得到的特征图进行下采样，减少了数据维度、参数量和计算量，同时增强特征的鲁棒性。

三、卷积神经网络结构

卷积神经网络的整体结构如下所示：其包括输入层、卷积层、池化层和全连接层。输入层就是输入的数据，其中对于图片输入来说通常是以RGB三通道的形式输入，所以C通常是3；卷积层和池化层就是对应进行卷积和池化操作，卷积层提取不同尺度的图像特征数据。池化层对特征进行下采样，因此一般卷积层后面都会跟上一个池化层，至于具体每个卷积层多少个卷积核，卷积核尺寸以及卷积层和池化层的数量可根据模型自行设计。



其中全连接层就是相当于在最后面加一层或多层传统神经网络层，我们在连接全连接层前，需要把我们的CNN的三维矩阵进行展平成二维，比如说从池化层出来的是 $5 \times 5 \times 3$ 的图像，在输入到全连接层展平成 1×75 以满足全连接层的输入模式。

视频代码实践使用的是LeNet-5卷积神经网络模型，其具体的结构如下所示：

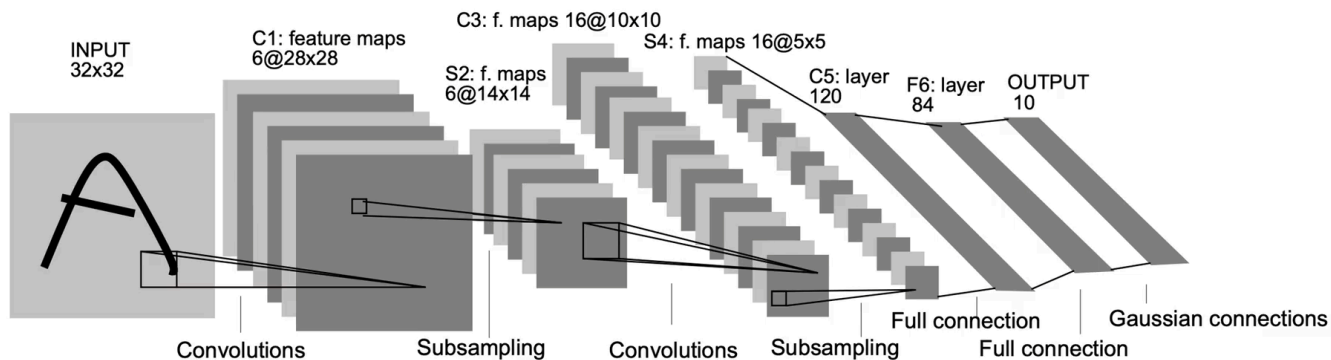


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

四、代码实践问题

1.使用lenet模型时的图像输入

在 TensorFlow / Keras 中，**卷积神经网络 (CNN)** 一般期望输入的形状是：

```
(batch_size, height, width, channels)
```

其中**batch_size**表示输入的要识别的样本数量，**height**是图像高，**width**是图像宽，**channels**是图像通道数。

通过cv2.imread读取的img图像对象的形状是：

```
(height, width, channels)
```

因此需要先将其增加一个维度，batch_size=1才能输入模型进行识别：

```
# 使用模型
model = keras.models.load_model("lenet.h5")
# 缺陷检测
pred = model.predict(img.reshape(1, IMG_HEIGHT, IMG_WIDTH, 3))
print(pred)
print(CLASS_NAMES[np.argmax(pred)])
```