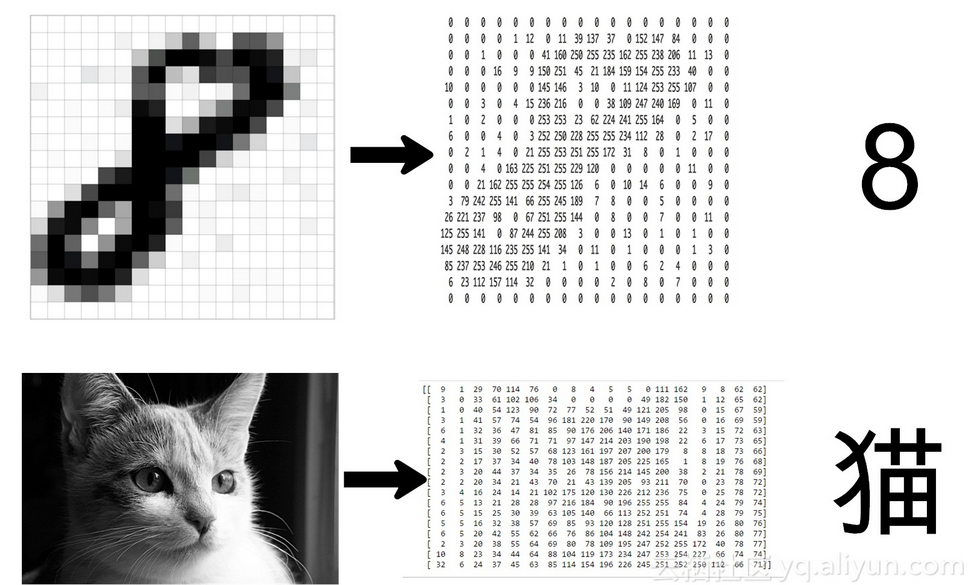
<https://yq.aliyun.com/articles/156269?spm=a2c4e.11153940.0.0.3a5073dds9Wdnu>

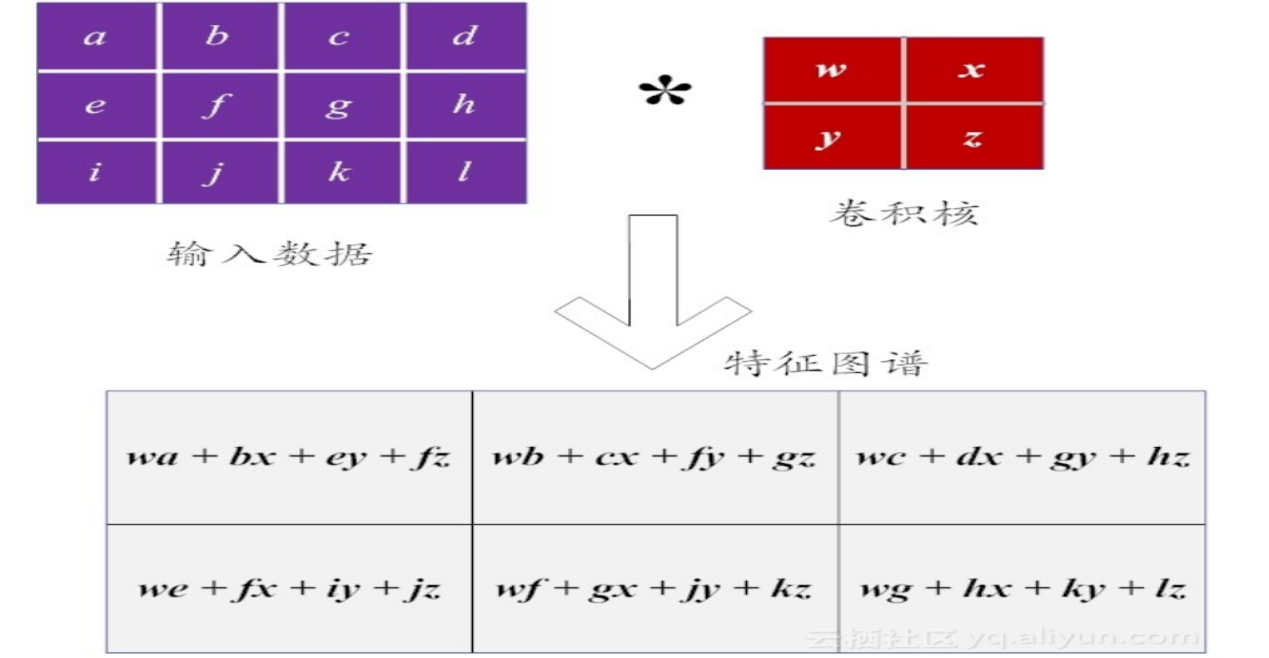
老哥说的非常详尽了

卷积神经网络主要的应用在CV领域，我们先来观察一下图片这种数据的具体数据形态

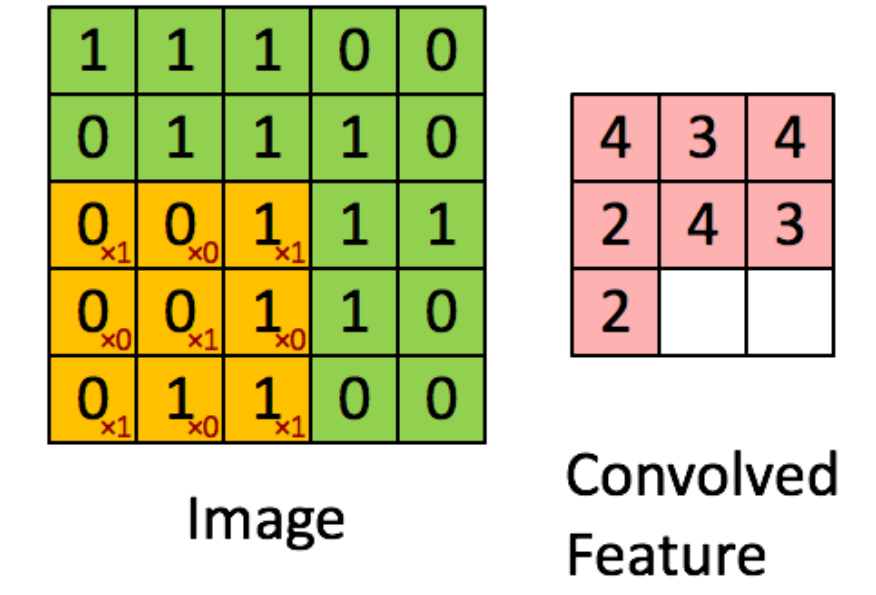
 数字矩阵（每个元素都是0到255之间像素值）

对于黑白图片来说，0代表黑色，255代表白色 像素值越小颜色越深。而对于彩色图片来说，一般有三个通道RGB（红，绿，蓝）。把三个不同通道的像素矩阵堆叠在一起，即可描述彩色图像。在图像处理中应用卷积操作，主要目的就是从图像中提取特征。卷积可以很方便地通过从输入的一小块数据矩阵（也就是一小块图像）中学到图像的特征，并能保留像素间的空间关系。

下面举例说明在二维图像使用卷积过程。

图像数据矩阵的像素值分别用诸如a-b-c-d这样的字母代替，卷积核是一个2×2的小矩阵。需要注意的是，在其他场合，这个小矩阵，也被称为“滤波器（filter）”或“特征检测器（feature detector）”。

在本质上，离散卷积就是一个线性运算。因此，这样的卷积操作也被称为线性滤波。这里的“线性”是指，我们用每个像素的邻域的线性组合来代替这个像素。其实，卷积操作还具有平移不变性（shift-invariant）。这个“平移不变性”是指，在图像的每个位置都执行相同的操作。



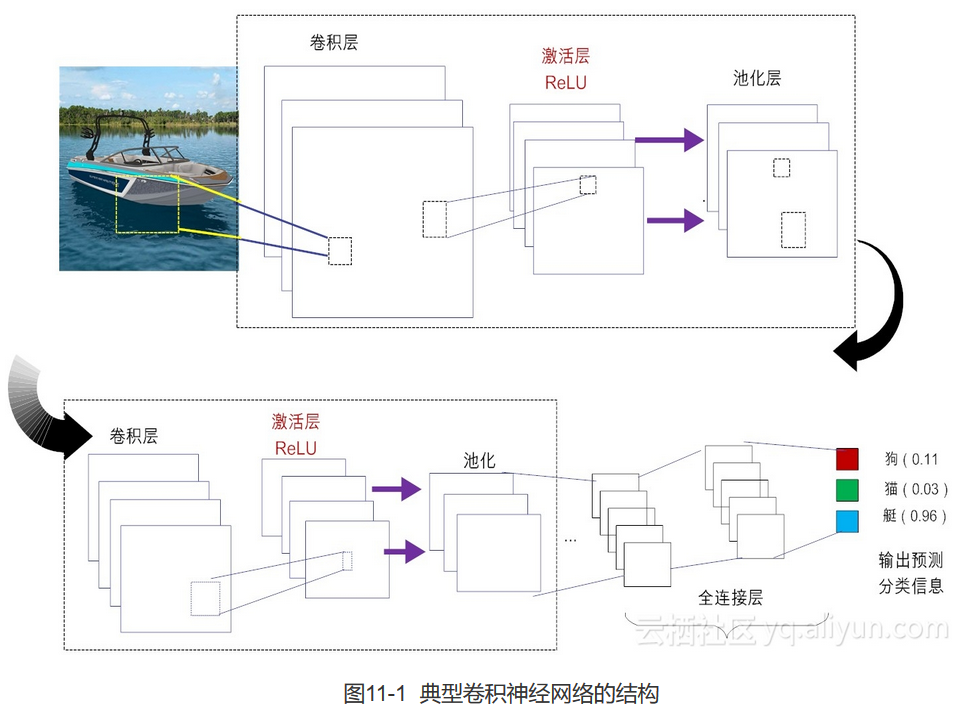
卷积操作在图像上作用之后具体的表现





从上图可以看出，不同的卷积核对图片的特征提取偏好不同，但是可以肯定的是图片数字矩阵元素之间的某些固定的线性组合作用之后，得到的卷积图像不仅仅保留了原图像主要的特征，还凸显了其某些特定方面的特征。

一个典型卷积神经网络的结构



**卷积层：**这个是卷积神经网络的核心所在。在卷积层，通过实现“局部感知”和“权值共享”等系列的设计理念，可达到两个重要的目的：

1.对高维输入数据实施降维处理和

2.实现自动提取原始数据的核心特征。

**激活层：**其作用是将前一层的线性输出，通过非线性激活函数处理，从而可模拟任意函数，进而增强网络的表征能力。在深度学习领域，ReLU（Rectified-Linear Unit，修正线性单元）是目前使用较多的激活函数，原因是它收敛更快，且不会产生梯度消失问题。

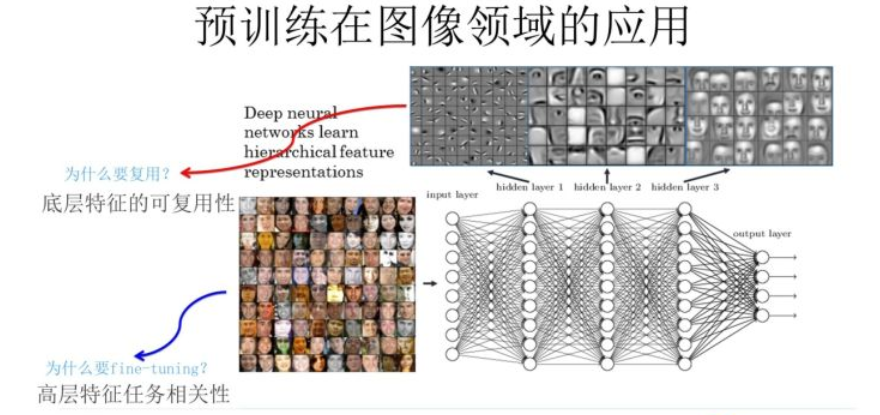
**池化层：**亦称亚采样层（Subsampling Layer）。简单来说，利用局部相关性，“采样”在较少数据规模的同时保留了有用信息。巧妙的采样还具备局部线性转换不变性，从而增强卷积神经网络的泛化处理能力。

**全连接层：**这个网络层相当于传统的多层感知机（Multi-Layer Perceptron，简称MLP）。通常来说，“卷积-激活-池化”是一个基本的处理栈，通过多个前栈处理之后，待处理的数据特性已有了显著变化：

一方面，输入数据的维度已下降到可用“全连接”网络来处理了；

另一方面，此时全连接层的输入数据已不再是“泥沙俱下、鱼龙混杂”，而是经过反复提纯过的结果，因此最后输出的结果要可控得高。

总体来讲，卷积神经网络通过多层处理，逐渐将初始的“低层”特征表示，转化为“高层”特征表示，之后再用“简单模型”即可完成复杂的分类等学习任务。因此在本质上，深度学习就是一个“特征学习（feature learning）”或“表示学习（representation learning）”[3]。



对于层级的CNN结构来说，不同层级的神经元学习到了不同类型的图像特征，由底向上特征形成层级结构，如上图所示，如果我们手头是个人脸识别任务，训练好网络后，把每层神经元学习到的特征可视化肉眼看一看每层学到了啥特征，你会看到最底层的神经元学到的是线段等特征，图示的第二个隐层学到的是人脸五官的轮廓，第三层学到的是人脸的轮廓，通过三步形成了特征的层级结构，越是底层的特征越是所有不论什么领域的图像都会具备的比如边角线弧线等底层基础特征，越往上抽取出的特征越与手头任务相关。正因为此，所以预训练好的网络参数，尤其是底层的网络参数抽取出特征跟具体任务越无关，越具备任务的通用性，所以这是为何一般用底层预训练好的参数初始化新任务网络参数的原因。而高层特征跟任务关联较大，实际可以不用使用，或者采用Fine-tuning用新数据集合清洗掉高层无关的特征抽取器。

**卷积层的3个核心概念**

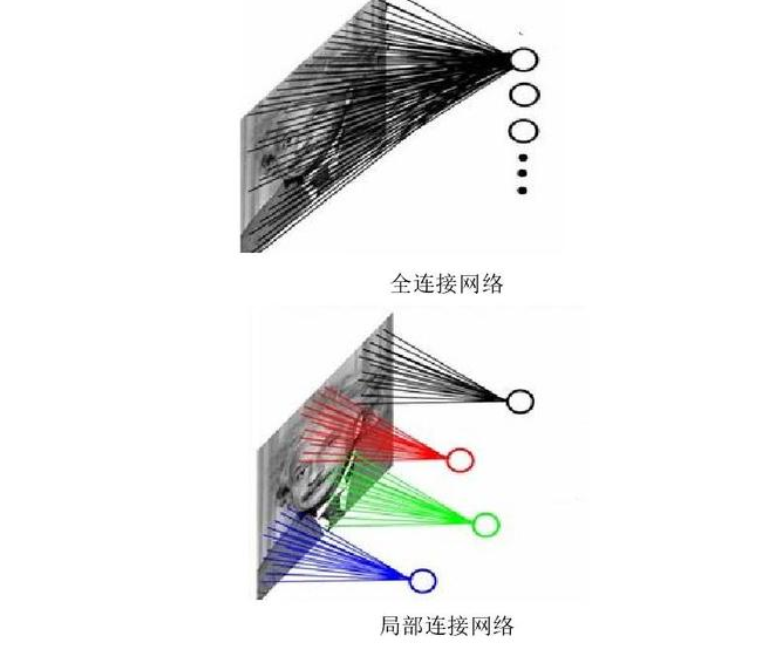
有了上面的工作的铺垫，下面我们来聊聊卷积层的三个核心概念：局部连接、空间位置排列及权值共享。

### 局部连接

全连接的前馈神经网络有个非常致命的缺点，那就是可扩展性（Scalability）非常差。原因非常简单，网络规模一大，需要调参的个数以神经元数的平方倍增，导致它难以承受参数太多之痛。局部连接（Local Connectivity）在能某种程度上缓解这个“参数之痛”。

每一幅CIFAR-10图像都是32×32×3的RGB图像。对于隐藏层的某个神经元，如果是全连接前馈网络中，它不得不和前一层的所有神经元（32×32）都保持连接。对于RGB图像而言，如果我们需要在红色、蓝色和绿色等三个通道提取特征，那么卷积核个数就是3）。所以对于隐藏层的某个神经元，它的前向连接个数是由全连接的32×32×3个。

但现在，对于卷积神经网络而言，隐藏层的这个神经元仅仅需要与前向层的部分区域相连接。这个局部连接区域有个特别的名称叫“感知域（receptive field）”，其大小等同于卷积核的大小（比如说5×5）目前看应该就是卷积层，从卷积过程看，卷积层的每一个像素(神经元)都是原图像某一片区域的线性聚合信息组成的，其实就是那一小块和卷积核同样大小的局部区域。卷积层相当于全连接网络中的隐藏层。所以对于隐藏层的某个神经元，它的前向连接个数是由全连接的32×32×3个，通过卷积操作，减少到局部连接的到5×5×3个，这里的权重矩阵变成了卷积核的矩阵，学习的过程就是不断的调整卷积核的值。



对比全连接网络这一层隐藏层表达 W\*X = H

对于某一个隐藏层神经元，连接的权边有W=32×32×3

如果转换为卷积神经网络 W\*C(X(局部)) = H

对于某一个隐藏层神经元，连接的权边有W=5×5×3

这个神经元是原图中局部区域与卷积核线性组合的结果，而不在是和全部像素连接

### 空间排列

下面我们来谈谈决定卷积层的空间排列（Spatial arrangement）的4个参数，它们分别是：卷积核的大小、深度、步幅及补零。

卷积核的大小（通常多是3×3或5×5的方矩阵）

1. 卷积核的深度（depth）：卷积核的深度对应的是卷积核的个数。每个卷积核只能提取输入数据的部分特征。每一个卷积核与原始输入数据执行卷积操作，会得到一个卷积特征，这样的多个特征汇集在一起，我们称为卷积特征。在图11-1所示的示例中（左上部），我们使用三个不同的滤波器（即卷积核）对原始图像进行卷积操作，这样就可以生成三个不同的卷积特征图。你可以把这三个特征图看作是堆叠在一起的2D（二维）矩阵。每个卷积核提取的特征都有各自的侧重点。因此，通常说来，多个卷积核的叠加效果要比单个卷积核的分类效果要好得多。例如在2012年的ImageNet竞赛中，Hinton教授和他的学生Krizhevsky等人打造了第一个“大型的深度卷积神经网络”，也即现在众所周知的AlexNet。在这个夺得冠军的深度学习算法中，他们使用的卷积核高达96个！可以说，自那时起，深度卷积神经网络一战成名，才逐渐被世人瞩目。
2. 步幅（stride）：即滤波矩阵在输入矩阵上滑动跨越的单元个数。设步幅大小为S，当S为1时，滤波器每次移动一个像素的位置。当S为2时，每次移动滤波器会跳过2个像素。S越大，卷积得到特征图就越小。以一维数据为例，当卷积核为[1，0，-1]，输入矩阵为[0, 1, 2, -1, 1, -3, 0]时，图11-5显示了步幅分别为1和2卷积层的神经元分布情况
3. 补零（zero-padding）：补零操作通常用于边界处理。在有些场景下，卷积核的大小并不一定刚好就被输入数据矩阵的维度大小整除。因此，就会出现卷积核不能完全覆盖边界元素的情况。这时，我们就需要在输入矩阵的边缘使用零值进行填充，使得在输入矩阵的边界处的大小刚好和卷积核大小匹配。这样做的结果，相当于对输入图像矩阵的边缘进行了一次滤波。零填充的好处在于，它可以让我们控制卷积特征图的大小。使用零填充的卷积叫做泛卷积（wide convolution），不适用零填充的叫做严格卷积（narrow convolution）。

综上所述，在构造卷积层时，对于给定的输入数据，如果确定了卷积核的大小，卷积核的深度（个数）、步幅以及补零个数，那么卷积层的空间安排就能确定下来。以一维数据为例，假设数据的大小（数据元素的长度）为W，卷积核的深度为F，步幅大小为S，补零的数目为P，那么对于每个卷积核，在它与输入数据实施卷积操作后得到特征图谱，它包含的神经元个数N可以用（11-1）公式计算得到。

卷积层特征个数就是神经元的个数。

### 权值共享

卷积层设计的第三个核心概念就是权值共享（Shared Weights），由于这些权值实际上就是不同神经元之间的连接参数，所以有时候，也将权值共享称为参数共享（Parameter Sharing）。为什么要设置权值共享呢？其实这也是无奈之举。前文我们提到，通过局部连接处理后，神经元之间的连接个数已经有所减少。可到底减少多少呢？还以CIFAR-10数据集合为例，一个原始的图像大小为32×32×3，假设我们有100个卷积核，每个卷积核的大小为5×5×3，步幅为1，没有补零。先单独考虑一个卷积核，将公式11-1扩展到二维空间，可以很容易计算得到每一个卷积核对应的特征图谱大小是28×28。也就是说，这个特征图谱对应有28×28神经元。而每个神经元以卷积核大小（5×5×3）连接前一层的“感知域（receptive field）”，也就是说，它的连接参数个数为（28×28）×（5×5×3）。如果考虑所有的100个卷积核，（在不考虑偏置参数的情况下）连接的参数个数为（5×5×3）×（28×28）×100 = 5,880,000。

那么全连接的参数个数又是多少呢？仅仅考虑两层网络的情况下，其连接个数为（32×32×3）×（32×32×3）=9,437,184。对比这二者的数字可以发现，局部连接虽然降低了连接的个数，但整体幅度并不大，需要调节的参数个数依然非常庞大，因此还是无法满足高效训练参数的需求。

而权值共享就是来解决这个问题的，它能显著降低参数的数量。该如何理解权值共享呢？首先从生物学意义上来看，相邻神经元的活性相似，从而可以它们共享相同的连接权值。其实就是说，每一个卷积层的神经元的值都是由输入层局部空间（感知域）与卷积核线性作用的结果，每一个值就是卷积核权值的线性组合，所以才有（5×5×3）×（28×28）×100个连接，权值共享的策略就是同属于一个特征图谱的神经元，将共用一个权值参数矩阵，也就是一个卷积核对全部的卷积层神经元只对应有一套权重参数，你们这28\*28个神经元共享。

而权值共享就是来解决这个问题的，它能显著降低参数的数量。该如何理解权值共享呢？首先从生物学意义上来看，相邻神经元的活性相似，从而可以它们共享相同的连接权值。其次单从数据特征上来看，我们可以把每个卷积核（即过滤核）当作一种特征提取方式，而这种方式与图像等数据的位置无关。这就意味着，对于同一个卷积核，它在一个区域提取到的特征，也能适用于于其他区域。基于权值共享策略，将卷积层神经元与输入数据相连，同属于一个特征图谱的神经元，将共用一个权值参数矩阵，如图11-7所示。经过权值共享处理后，CIFAR-10的连接参数一下子锐减为5×5×3×1×100 = 7500。这样就不用把每一个和卷积层神经元相连的权重都做更新了，参数空间立马下降了很多

