

# CNN超简洁版整理

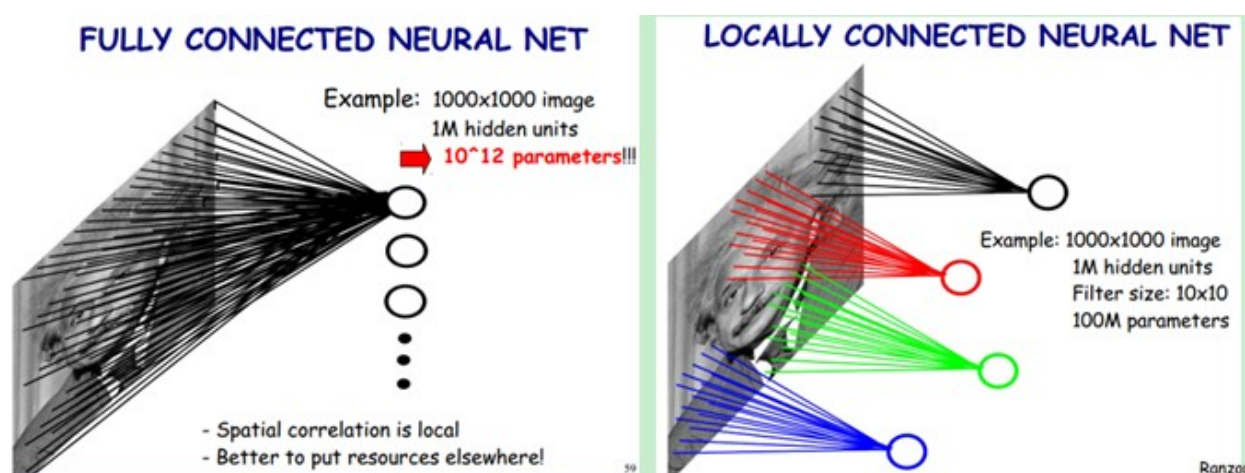
嗯，这个文档是用来整理一下CNN到底是什么东西来的嘻嘻。不包含反向求导的细节毕竟我们有轮子可以用求导就不用自己做啦～在阅读下面的内容的时候建议先别问为什么这样，先接受会舒服很多hhh。

## CNN来历

首先抛出问题为什么需要CNN？为什么不用传统神经网络而要用CNN这个东西？

那么就得先说到我们人类有关视觉感官的机制，研究感官的科学家认为人类对于视觉认知的过程是从局部到整体的，意思就是对于我们眼前的东西，先认出一篇文章再认出一部手机或者电脑，然后知道我们在看东西之类的。所以提出了一个强并且合理的假设，图像中的像素点是局部关联性极强的，我们并不需要找出所有像素点之间的关系，因为他们有一些根本就没有关系。而传统神经网络的做法其实就是要找所有像素点之间的关系，因为隐藏层的每一个神经元都是和输入层的所有节点相连的，所有像素点的信息都通过了这个神经元。这样不仅需要学习的参数过多，而且一定程度上降低了对特征的提取功能。

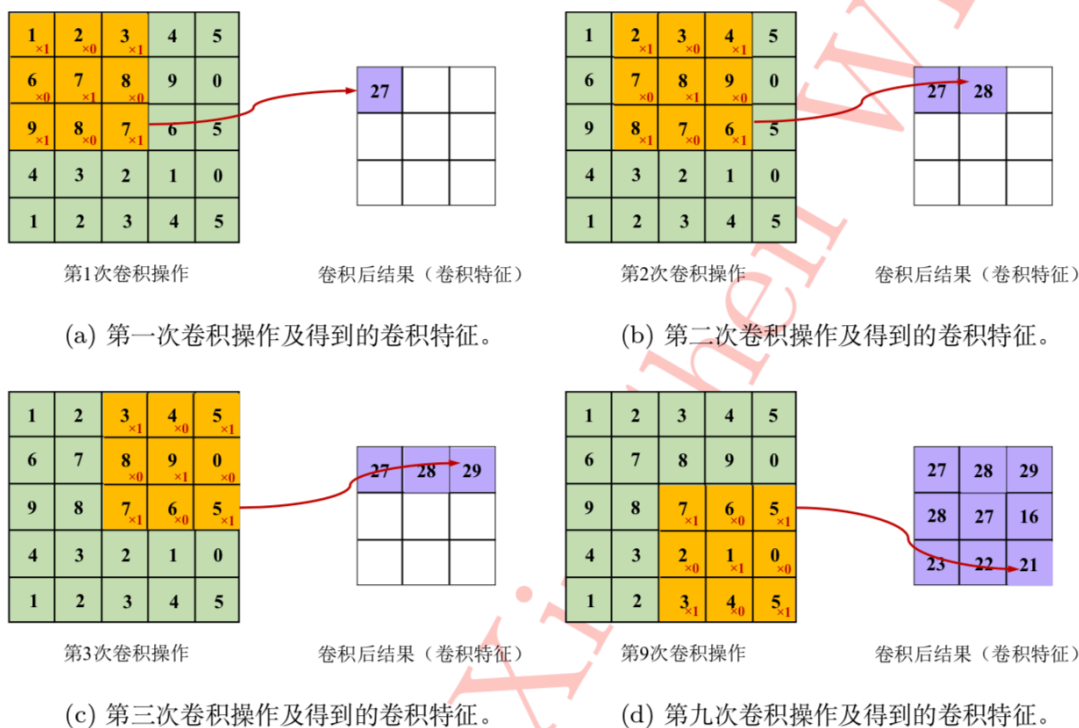
嗯，然后CNN是怎么做的呢？CNN就做到了**局部连接**，意思就是隐藏层的每一个神经元不必在意像素点之间的关系，而是在意一个个区域之间的关系。讲到这里有点迷，必须上图。



看到每一个神经元其实就只负责一个区域的像素，而这里面涉及到卷积的操作，稍后会细讲。

除了**局部连接**，CNN还使用了**权值共享**(这是CNN最重要的两个思想科科)。用具体的例子来解释。假如我们现在的神经元有10个，每个神经元与图像中10\*10大小的局部区域相连(意思就是卷积核的维度是 $R^{10 \times 10}$ )，那么需要学习的参数总共有10x10x10。嗯，那么权值共享的做法就是在局部连接中让这些神经元的权重都相同，所以最后需要学习的参数就是10x10，刚好就是卷积核的大小。

可能这里又有点迷了，看CNN的时候最容易搞不清楚的就是传统神经网络中的神经元在这里到底扮演什么角色，有什么参数需要学习？在上面的例子中我们先按照正常CNN的流程来说明。



在上面的例子中可以看到一个卷积核（黄色区域是一个卷积核）是在移动的，上面的四个黄色区域其实就是四个传统意义上的神经元，如果没有权值共享的话，神经元是不会共用一个卷积核的。如果权值是共享的话我们可以当作是一个卷积核在移动着识别到各个区域，所以在这里一般我们讨论的是卷积核而不是神经元。那么这个问题解释清楚了以后就看清楚上面的卷积是怎么进行的。很简单，就是对应像素点的值和卷积核上对应位置的值相乘再加起来，比如图a的 $1 \times 1 + 3 \times 1 + 7 \times 1 + 9 \times 1 + 7 \times 1 = 27$ ，那么这个27的值就是这个区域的特征值～这样就完成了特征提取的操作啦。

## 卷积层

上面其实已经简单的说明了卷积的内容了。卷积层是卷积核在输入层上通过逐一滑动串口计算得到的，卷积核中的每一个参数相当于一个权重啦。通常在代码里面我们需要指定卷积核的大小以及滑动的步长(所以卷积核不一定一格一格那样移动的哈)。下面再举一个复杂一点的例子，里面涉及到几个概念。



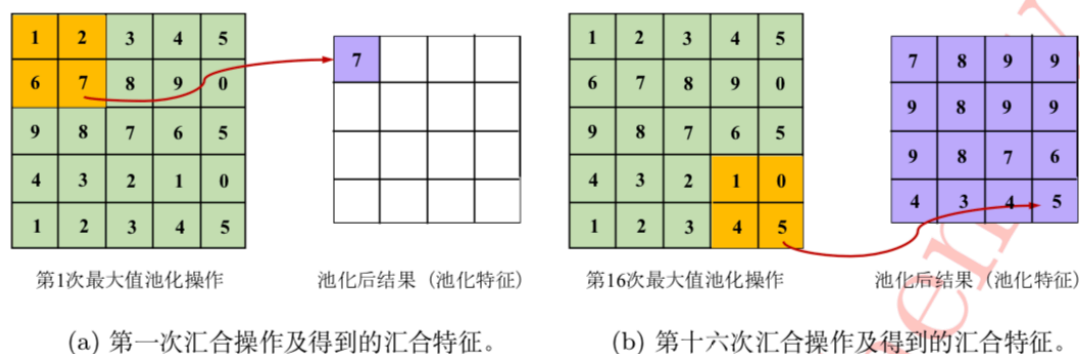
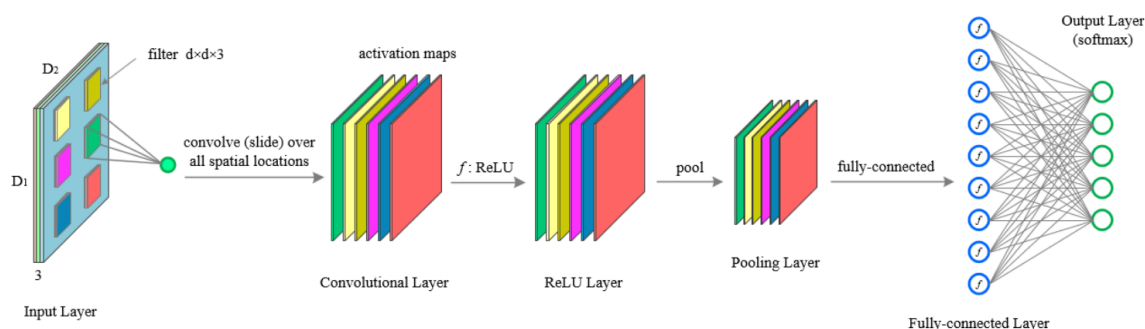


图 2.7: 最大值汇合操作示例。

由于池化层达到的降维效果，这一个步骤也被称为降采样。研究者普遍认为池化层有下面三种功效：

- 特征不变性。汇合操作使模型更关注是否存在某些特征而不是特征具体的位置。可看作是一种很强的先验，使特征学习包含某种程度自由度，能容忍一些特征微小的位移。
- 特征降维。由于汇合操作的降采样作用，汇合结果中的一个元素对应于原输入数据的一个子区域，因此汇合相当于在空间范围内做了维度约减，从而使模型可以抽取更广范围的特征。同时减小了下一层输入大小，进而减小计算量和参数个数。
- 在一定程度防止过拟合,更方便优化。

## 一般的CNN网络架构



注：实际应用的过程中常常不限于C-R-P循环，也有可能是C-C-R-P等等

一般都是输入层扔到卷积层提取特征，然后扔进ReLU层去掉无用的信息，然后再进行一个降采样，这样就完成了特征的提取过程。最后扔进一个全连接层来学习。这是一个最简单的CNN架构～