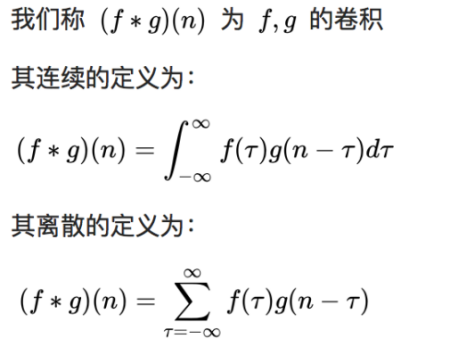
Pytorch实现卷积神经网络一---卷积神经网络基础

这次的文章我们开始接触一下大名鼎鼎的神经网络，神经网络的种类有很多包含全连接的神经网络、卷积神经网络和循环神经网络等等，本次文章我们直接从卷积神经网络开始。

卷积神经网络（CNN,Convolutional neural network,）的文章网络上已经有很多了，那么作者为什么还要自己写呢？主要原因是想从一个小白的角度和大家分享一下卷积神经网络的相关知识，在写作的过程中还可以发现一些不一样的东西和巩固自己已经掌握的知识点。

1. **什么是卷积**

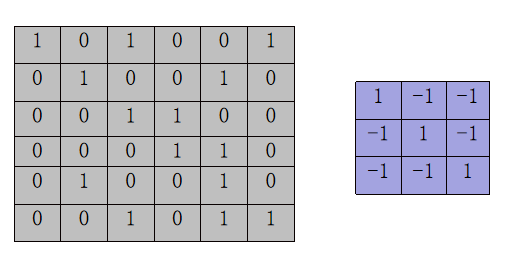
个人对卷积的理解就是一种数学运算，一种加权求和的数学运算。为避免引入更加复杂的概念，这里就不在引入信号处理中相关的知识，大多数作者都赞同卷积是一个函数和另一个函数在某个维度上的加权“叠加”作用。那么数学中对卷积的定义为：



可以看出其就是一种运算罢了，那么卷积的物理意义就是：**一个函数（如：单位响应）在另一个函数（如：输入信号）上的加权叠加。更多的关于卷积的理解大家可以参考相应的数学教材。**

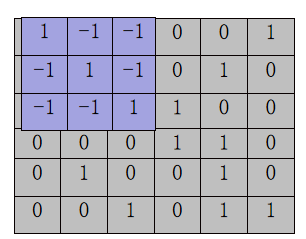
1. **CNN中的卷积**

如何理解CNN中的卷积呢？我们先举一个简单的例子。假设我们现在有一个6×6大小的图片和一个3×3的卷积核（后面将介绍什么是卷积核），如下所示：

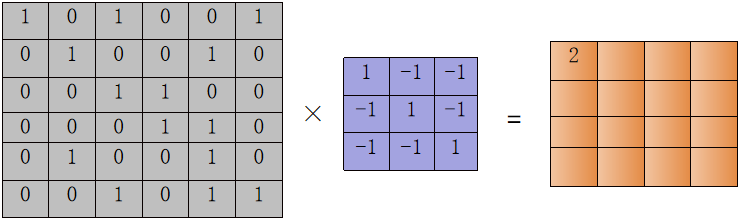


接下来我们开始进行卷积的操作。

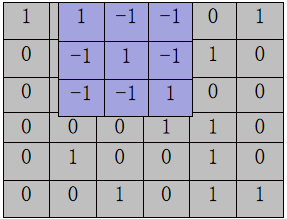
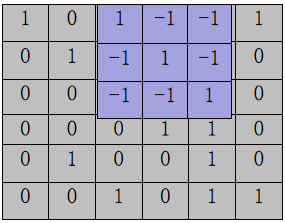
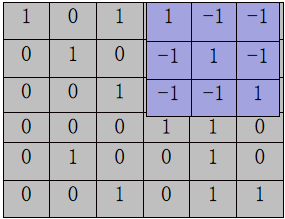
1. 我们将卷积核元素与原图像中的元素对应进行相乘后相加的操作：



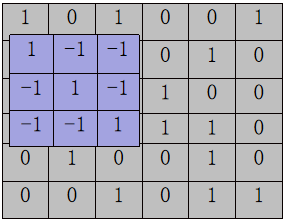
即：1×1+0×(-1)+1×(-1)+0×(-1)+1×1+0×(-1)+0×(-1)+0×(-1)+1×1 =2



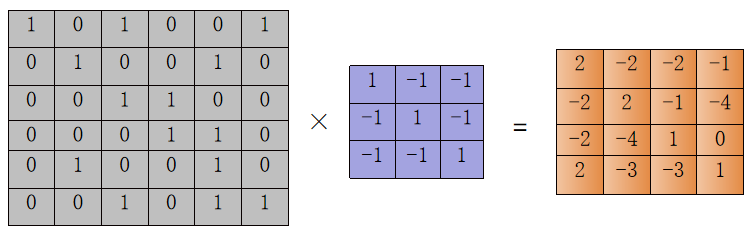
1. 接着我们卷积核的第一次，第二次，第三次的滑动：

1. 在滑动到第一行结尾的时候，进行下移操作：



1. 直到我们将所有的滑动和计算完成，这样就得到如下结果：



可以看出我们原来的6×6大小的图片转化为4×4大小的图片。其中卷积核的滑动有多种方式，这里就使用不超过原图片的方式一步一步的滑动。

下面我们使用Python来实现上述的计算结果：

import numpy as np  
from scipy import signal  
my\_pic = np.array([[1, 0, 1, 0, 0, 1],  
 [0, 1, 0, 0, 1, 0],  
 [0, 0, 1, 1, 0, 0],  
 [0, 0, 0, 1 ,1, 0],  
 [0, 1, 0, 0, 1, 0],  
 [0, 0, 1, 0, 1, 1]  
 ])  
kernal = np.array([[1, -1, -1],  
 [-1, 1, -1],  
 [-1, -1, 1]  
 ])  
*# 卷积计算，参数mode=valid表示卷积核不超过原图片的范围。*res = signal.convolve2d(my\_pic,kernal,mode='valid')  
print(res)

# 输出

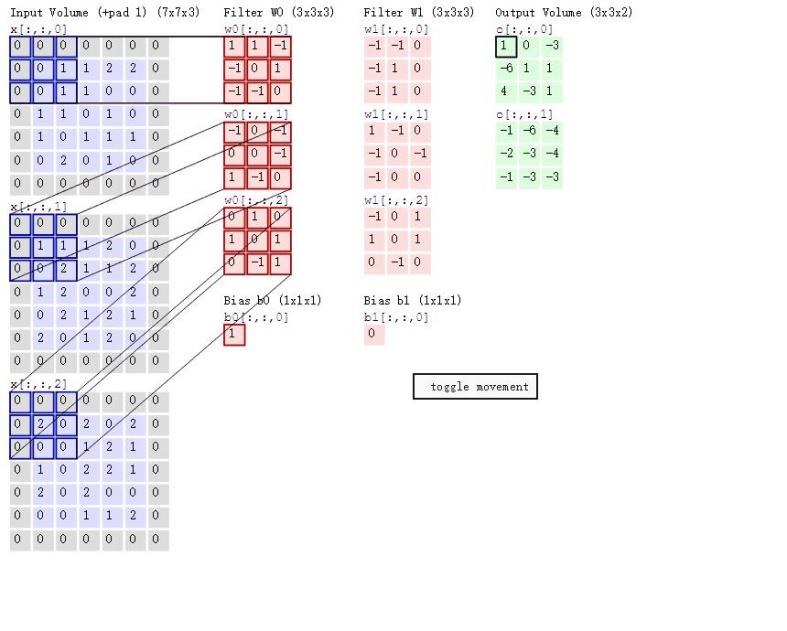
[[ 2 -2 -2 -1]

[-2 2 -1 -4]

[-2 -4 1 0]

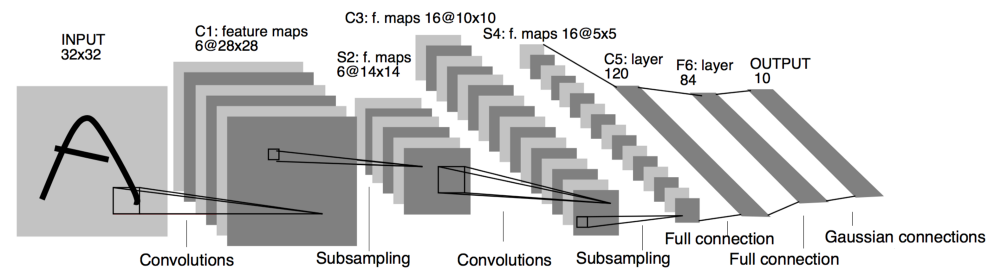
[ 2 -3 -3 1]]

这样的操作就是CNN中的卷积操作，当我们的图像是彩色RGB三个颜色通道的时候，此时我们输入数据维度就从长，宽，通道三个维度来进行描述。假设我们现在输入图片为(7,7,3)即在原始图像上增多了一个channels，此时我们的卷积核的维度就要变成从(3,3)变为(3,3,3)了，**卷积核的最后一维要跟输入的channel维度一致，**这个时候的卷积计是三个channel的所有元素对应相乘后求和，当我们使用2个卷积核进行卷积操作的时候，那么输出的维度就会变成(3,3,2),大家可以参考以下动图，其中b为一偏置量：



1. **卷积神经网络结构**

根据不同的卷积层、采样层的使用不同神经网络也存在着不同的结果，下图是经典的LeNet-5神经网络的结构：



LeNet-5神经网络结构

接下来介绍一下卷积神经网络中主要层的作用：

1. 卷积层（Convolutional）

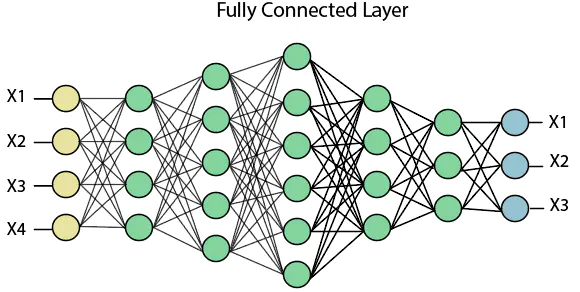
卷积层是卷积神经网络的核心层，由卷积核filter和激活函数构成。在实际工程中，对于这一步骤我们要设置的参数包括：卷积核filter的数量、卷积大小、卷积核移动步长，以及填充方式padding(valid或same)，以及还包括选择什么激活函数。

2. 池化层（Pooling）

池化层也是卷积神经网络中的重要的一个结构，相当于LeNet-5中的Subsampling采样层。 池化层不需要进行参数的训练和学习，仅仅需要设置池化的方式如Maxpooling还是Minpooling等。一般我们需要将原始图像的大小缩小的时候，使用Maxpooling进行池化。

3. 全连接层（Fully-Connected）

在整个[卷积神经网络](https://so.csdn.net/so/search?q=%E5%8D%B7%E7%A7%AF%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_45829462/article/details/_blank)中起到“分类器”的作用。用数学思维的语言概括就是：全连接层则起到将学到的“特征表示”映射到样本空间的作用。从下图中可以看出层中的每个节点都会连接它下一层的所有节点，这就是全连接层。



1. **总结**

本次推文我们讲述了卷积神经网络的卷积意义，用实例和图解的方式讲述了卷积神经网络中的卷积操作，之后讲述了卷积神经网络中各个层的实际作用。下次文章我们进一步详细的讲述卷积层、池化层相关参数的设置，如步长Stride,填充Padding等。

1. **参考资料：**
2. [Convolutional Neural Networks - Basics · Machine Learning Notebook (mlnotebook.github.io)](https://mlnotebook.github.io/post/CNN1/)
3. 《数字信号处理（第二版）》程乾生，北京大学出版社