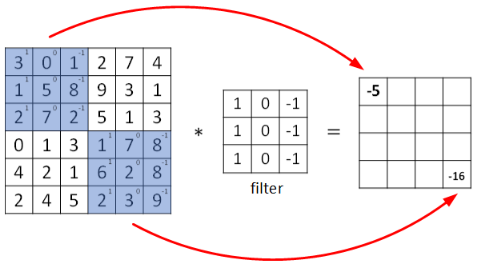
卷积神经网络基础

http://blog.csdn.net/red\_stone1/article/details/78651063

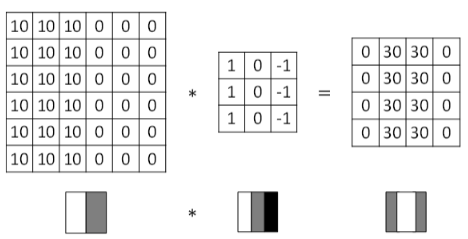
卷积神经网络多用于机器视觉，常见场景为图像分类，目标检测和风格迁移

在机器视觉中，使用CNN替代NN是因为图片输入通常变量巨多，导致权重参数和网络模型都非常复杂。进而造成样本数量相对于网络和参数复杂度显得不足（尽管图片已经很多了），以及巨大的算力要求。CNN因为共享参数，有效避免了上述问题。

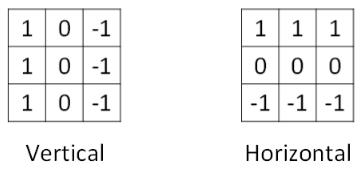
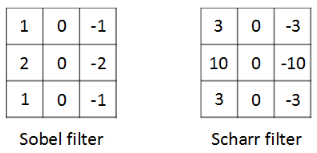
卷积运算过程。中图filter每一元素与左图蓝色部分对应位置相乘后全部相加，为有图结果-5。然后filter在左图中滑动，与右移1格下移0格（filter尺寸为3，右移3此后就会下移1次）的方块重复前一步骤。直至滑动到左图右下角最后一块蓝色区域，计算得到-16



CNN能够实现垂直边缘检测的例子。构建不同的filter参数，能够实现不同的边缘类别检测。Filter参数也正是CNN需要梯度下降学习的。

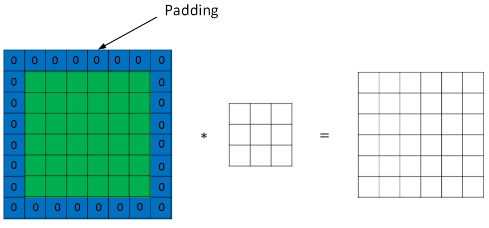


常见边缘检测filter

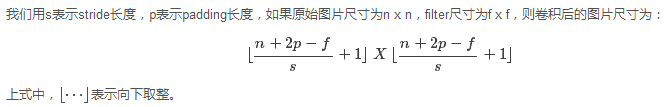
Sobel filter和Scharr filter可以增加图片中心区域的权重

Padding。卷积之后图像输入的尺寸会减小，也为着原始输入边缘信息的丢失。padding通过在原始输入边缘补0后再进行卷积，解决此问题。因此，padding之后的卷积，也称为same convolution



卷积步长convolution stride。卷积运算中，filter在输入中每次向右向下移动的格数。

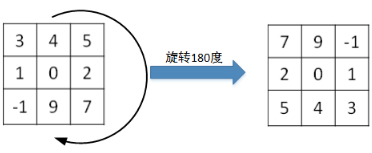
卷积运算输入输出尺寸变化关系



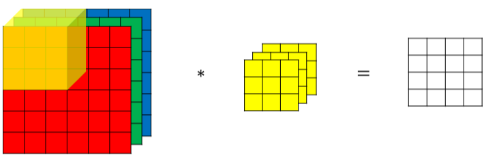
中间看着像X的符号，其实是乘号-\_-!

实际卷积运算会旋转后再滑动运算。严格意义上讲，没有旋转的“卷积运算”，是“相关系数运算”。

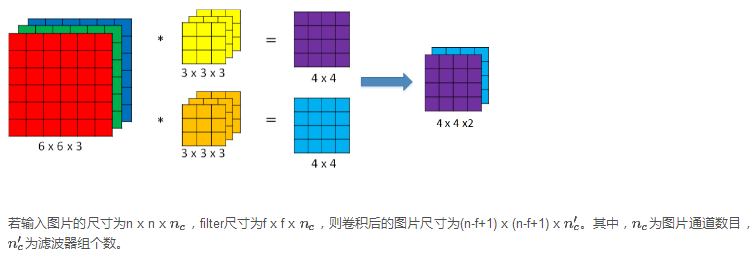
但是考虑到卷积系数的梯度学习过程，将会隐形包含旋转过程，因此CNN中可以认为“卷积运算”等效于“相关系数运算”，从而简化模型。



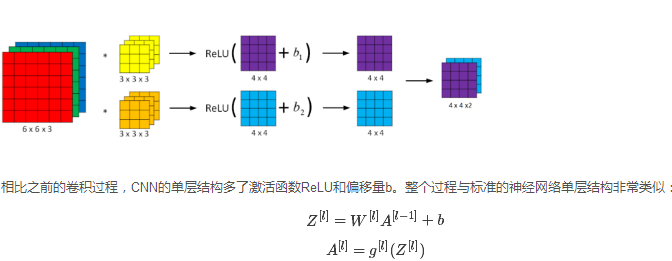
二维以上输入的卷积，filter需要增加相同维度。运算过程与二维一样，都是对应元素相乘后求和，最终结果还是二维



对二维以上输入，通常由不同filter进行多个卷积，并将多个二维结果堆叠为更高维度。作为下一环节的输入。

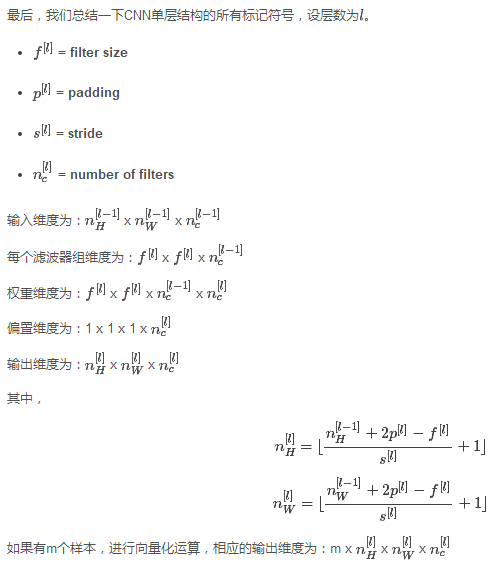


单层CNN

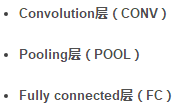


卷积参数与输入无关，在处理变量巨多的输入时，能够明显降低模型和运算复杂度

CNN模型符号约定



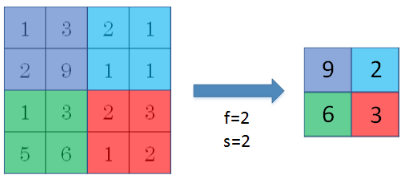
卷积网络结构。

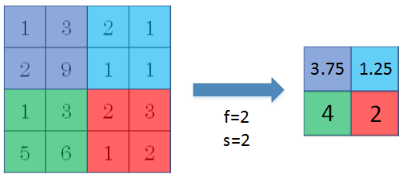


随着CNN层数增加，各层图片输入的高度和宽度逐渐减小，通道数逐渐增加

池化层pooling layer，用来图像减小尺寸，提高运算速度。还能减小noise影响，让各特征更加健壮

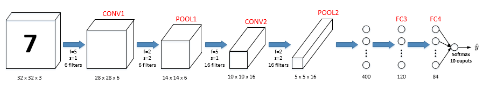
最常见池化是max pooling，也有average pooling。运算过程如下，就是在输入的filter滑动区域内取最大值或均值

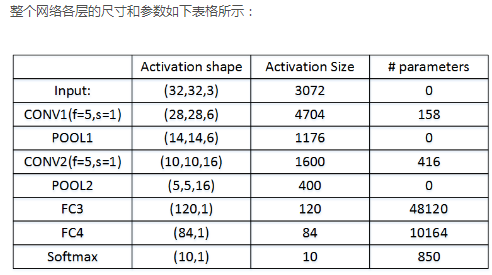




池化层相当于在小范围特征中提取最具代表型的项

CNN样例





Activation size即某层输入总共变量。3072 = 32 \* 32 \*3

中间池化层运算后，输出尺寸变为多少？？？

机器视觉选择CNN而非NN的原因

参数共享，一个特征检测器（即一个filter）可以作用于图片的多个区域

CNN参数较少，需要的样本数量较少，不容易发生过拟合。后面还会看到，CNN能够在一次运算中捕捉物体的位置移动，而不需要NN的若干次运算。