卷积运算简介

参考：

http://blog.csdn.net/anan1205/article/details/12313593
http://blog.csdn.net/u014114990/article/details/51125776

本文档主要介绍两个方面：

1. 矩阵卷积如何转化为矩阵相乘的形式；
2. 多通道卷积过程。

矩阵卷积、矩阵相乘的转化

两个矩阵卷积转化为矩阵相乘形式——Matlab应用(这里考虑二维矩阵，在图像中对应)两个图像模糊（边缘）操作，假设矩阵A、B，A代表源图像，B代表卷积模板，那么B的取值决定最后运算的结果。

Matlab中的应用函数——conv2（二维卷积，一维对应conv）

函数给出的公式定义为：

$$c(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} a(k_1, k_2)\, b(n_1 - k_1, n_2 - k_2)$$

http://blog.csdn.net/anan1205

同一维数据卷积一样，它的实质在于将卷积模板图像翻转（旋转180），这里等同于一维信号的翻转，然后将卷积模板依次从上到下、从左到右滑动，计算在模板与原始图像交集元素的乘积和，该和就作为卷积以后的数值。

为了验证后续矩阵卷积转化为矩阵相乘，这里给出的conv2的实例描述：

假设矩阵A（4*3）、B（2*3）如下：

	1	2	3
1	1	4	5
2	2	5	6
3	3	5	6
4	7	8	9

	1	2	3
1	1	2	3
2	5	6	7

http://blog.csdn.net/anan1205

首先，B需要旋转180，

命令旋转2次90即可：

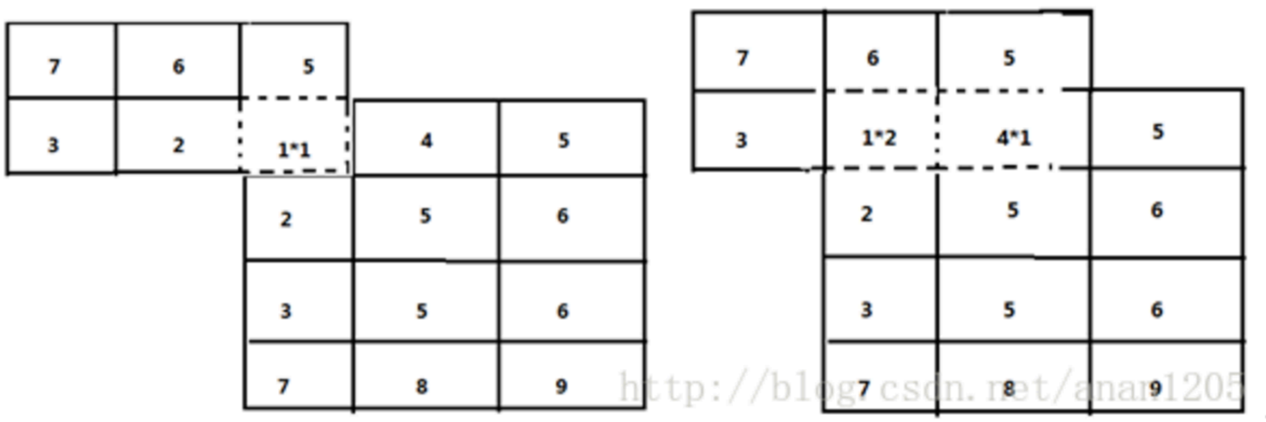
B = rot90(rot90(B));或者B = rot90(h,2); 结果为：

	1	2	3
1	7	6	5
2	3	2	1

http://blog.csdn.net/anan1205

其次：命令conv2函数：

C = conv2(A,B，‘shape’),该函数的具体操作图示：



依次计算直至结束，结果数据为：

	1	2	3	4	5
1	1	6	16	22	15
2	7	35	78	85	53
3	13	48	99	98	60
4	22	65	127	113	69
5	35	82	142	110	63

http://blog.csdn.net/anan1205

shape的取值有三种，full代表返回卷积以后的全部数据，size为(mA+mB-1,nA+nB-1)的数据；same代表返回卷积以后的原图size (mA,nA)的部分数据；valid返回size为(mA-mB+1,nA-nB+1)的数据，指的是模板元素全部参加运算的结果数据，即源图像和模板的交集为模板。

矩阵卷积转化为矩阵相乘，网上也有很多方法，通俗化表示为：

A×B = B1*A1;

需要针对原始数据与模板数据做变换，变换过程如下：

	1	2	3
1	1	4	5
2	2	5	6
3	3	5	6
4	7	8	9

	1	2	3
1	1	2	3
2	5	6	7

http://blog.csdn.net/anan1205

首先进行周期延拓，补零：

M = mA+mB-1 = 5；N = nA+nB-1 = 5，对应卷积以后full数据大小。

那么初次换换的A和B为：

	1	2	3	4	5
1	1	4	5	0	0
2	2	5	6	0	0
3	3	5	6	0	0
4	7	8	9	0	0
5	0	0	0	0	0

（A1）

	1	2	3	4	5
1	1	2	3	0	0
2	5	6	7	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

http://blog.csdn.net/anan1205

（B1）

其次对A1和B1分别进行变换

转化B1——针对B1以及转换矩阵方法为：

	1	2	3	4	5	
1	1	2	3	0	0	a
2	5	6	7	0	0	e
3	0	0	0	0	0	d
4	0	0	0	0	0	c
5	0	0	0	0	0	b

http://blog.csdn.net/anan1205

$$B = \begin{vmatrix} Ba & Bb & Bc & Bd & Be \\ Be & Ba & Bb & Bc & Bd \\ Bd & Be & Ba & Bb & Bc \\ Bc & Bd & Be & Ba & Bb \\ Bb & Bc & Bd & Be & Ba \end{vmatrix}$$

将B1中的每一行向量依次按照B转化为一个方形矩阵Ba~Be，然后针对于每一个方形矩阵按照B矩阵组合成一个新的矩阵B1。B1矩阵的大小为（（mA+mB-1）*（nA+nB-1），（mA+mB-1）*（nA+nB-1））。

转化A1——堆叠向量式

将上个步骤转换的A1按照行向量顺寻依次转化为一个列向量，那么列向量的大小为（（mA+mB-1）*（nA+nB-1），1）大小。

$$A1 = [1, 4, 5, 0, 0, 2, 5, 6, 0, 0, 3, 5, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$$

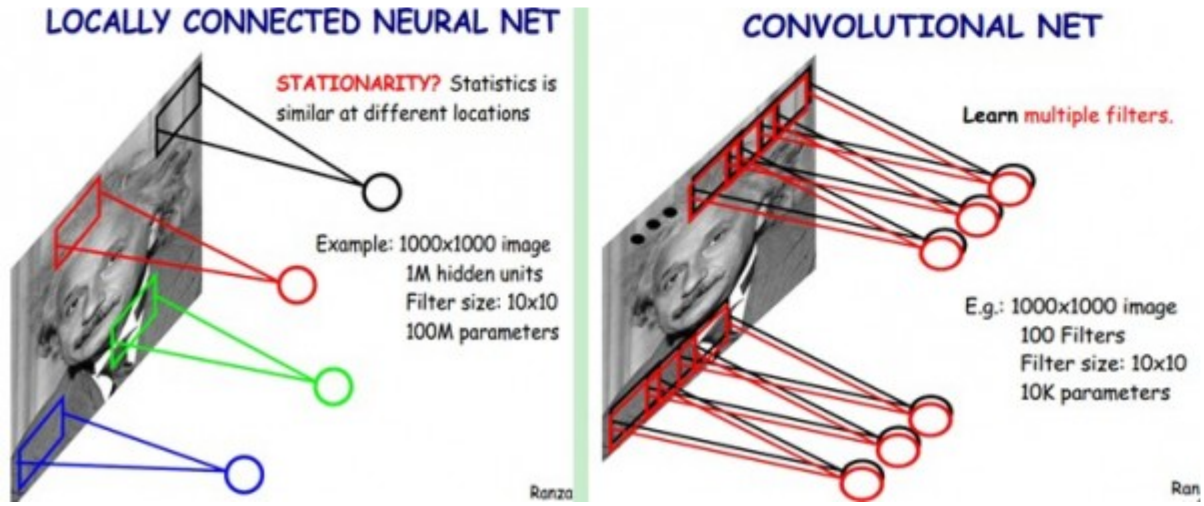
http://blog.csdn.net/anan1205

多通道卷积过程

卷积过程对一个通道的图像进行卷积，比如 10 个卷积核，得到 10 个 feature maps，那么输入图像为 RGB 三个通道呢，输出就为 30 个 feature map 吗， 答案肯定不是的， 输出的个数依然是 卷积核的个数。 可以查看常用模型，比如 lenet 手写体，Alex imagenet 模型， 每一层输出 feature map 个数 就是该层卷积核的个数。

1、一通道 多个卷积核卷积过程

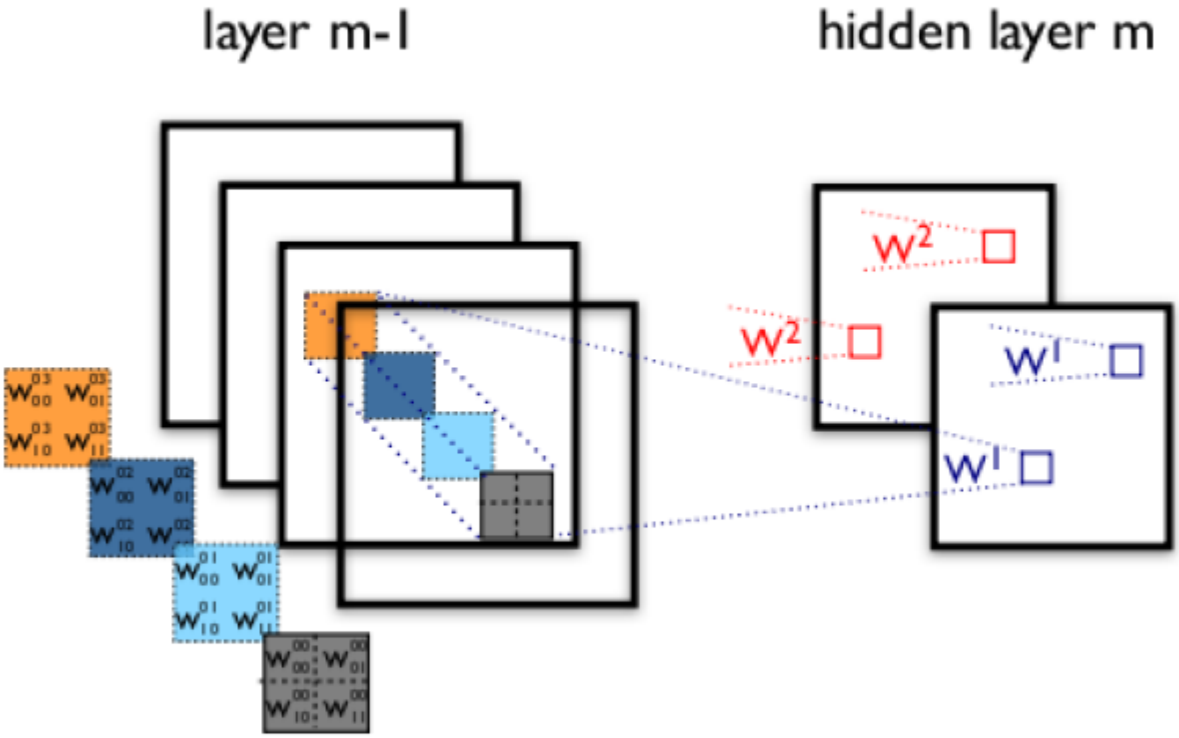
一个卷积核得到的特征提取是不充分的，我们可以添加多个卷积核，比如 32 个卷积核，可以学习 32 种特征。在有多个卷积核时，如下图所示：输出就为 32 个 feature map



2、 多通道的多个卷积核

下图展示了在四个通道上的卷积操作，有两个卷积核，生成两个通道。其中需要注意的是，四个通道上每个通道对应一个卷积核，先将 w2 忽略，只看 w1，那么在 w1 的某位置（i,j）处的值，是由四个通道上（i,j）处的卷积结果相加然后再取激活函数值得到的。 所以最后得到两个 feature map， 即输出层的卷积核核个数为 feature map 的个数。

$$h_{ij}^{k} = \tanh((W^{k} * x)_{ij} + b_{k})$$



所以，在上图由 4 个通道卷积得到 2 个通道的过程中，参数的数目为 4×2×2×2 个，其中 4 表示 4 个通道，第一个 2 表示生成 2 个通道，最后的 2×2 表示卷积核大小。