深蓝学院
shenlanxueyuan.com

深度学习理论与实践

第三课：卷积神经网络

主讲人 郑元春

中科院大数据挖掘与知识管理重点实验室
中科院虚拟经济与数据科学研究中心

人类视觉系统

CNN数学原理

CNN推导

CNN发展历史

手写数字识别

1. 人类视觉系统

视觉产生的生物基础

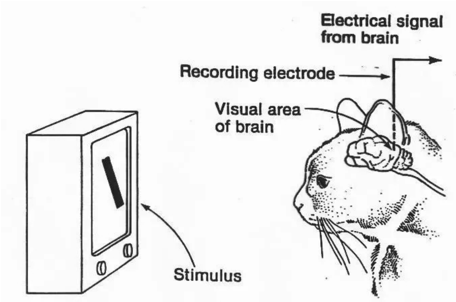
视频[眼睛的进化与视觉系统]

20 世 纪 60 年 代 初 ， David Hubel,Torsten Wiesel和Steven Kuffler来到哈佛大学，在哈佛医学院建立了神经生物学系。

1962 年 《Receptive fields, binocular interaction and functional architecture in the cat's visual cortex》中提出了**Receptive fields**的概念。

Hubel和Wiesel

Hubel和Wiesel记录了猫脑中各个神经元的电活动，通过这些实验系统地创建了视觉皮层的地图。

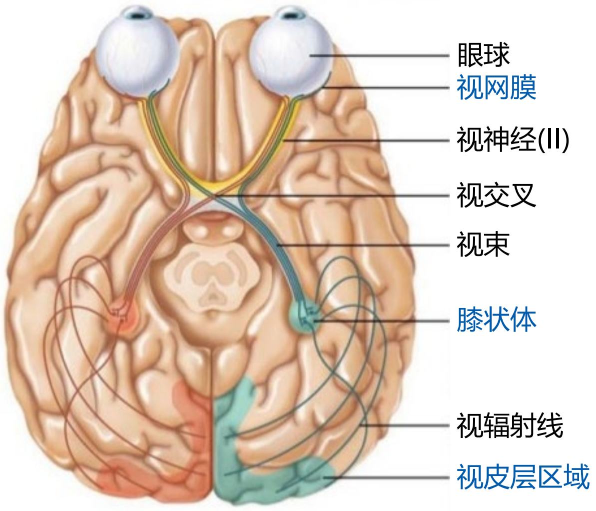


1. 人类视觉系统

视觉产生的生物基础

人眼的视觉系统可以被认为是由3大部分依次链接组合而成，分别是：

- 位于人眼球后壁内部的视网膜（retina）
- 位于丘脑的外侧膝状体（lateral geniculate）
- 以及大脑的视皮层区域（visual cortex）

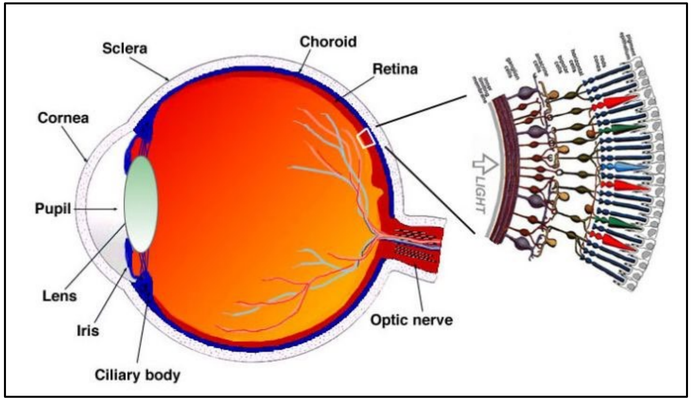


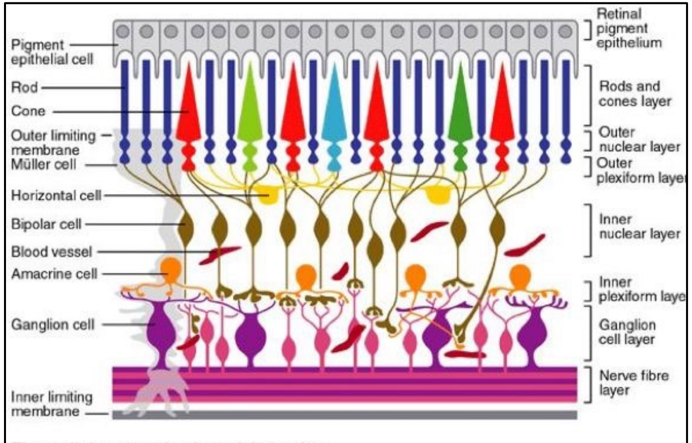
1. 人类视觉系统

视觉产生的生物基础

视觉系统的第一个主要部分就是眼球后壁上的视网膜。

视网膜是位于眼球后方负责将传入眼睛的光信号转化为生物电信号，并初步分析后将处理后的生物电信号通过视神经向外侧膝状体传递的神经网络结构。



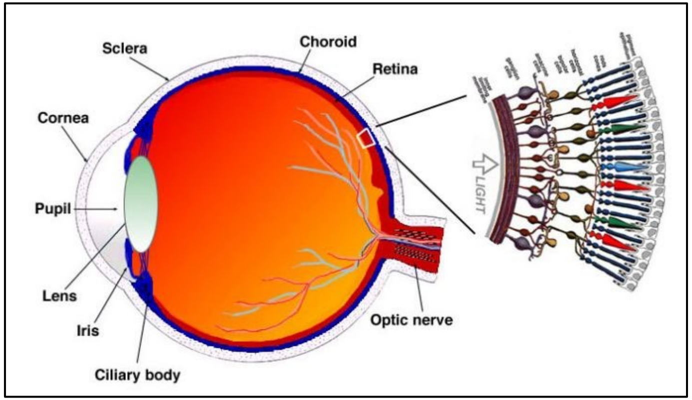


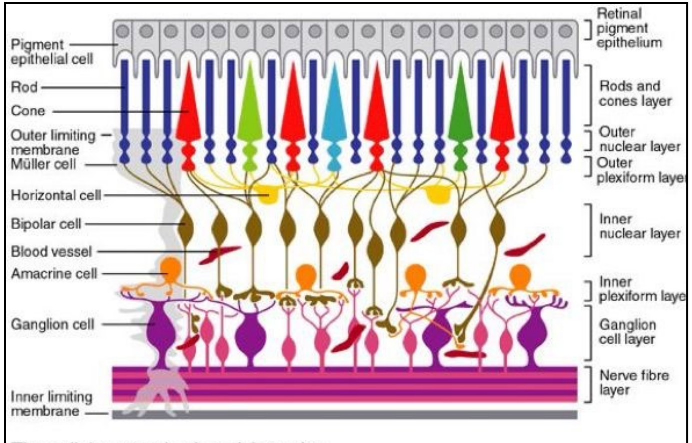
The cellular organisation of the retina

1. 人类视觉系统

视觉产生的生物基础

视网膜中由神经元占据的另外两层分别是外侧核状层（Outer nuclear layer）和内侧核状层（Inner nuclear layer）。





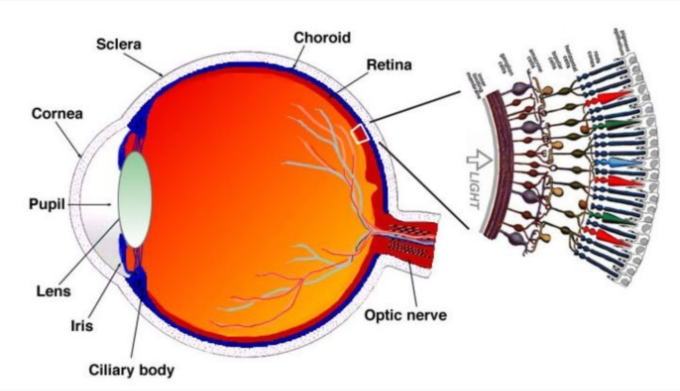
The cellular organisation of the retina

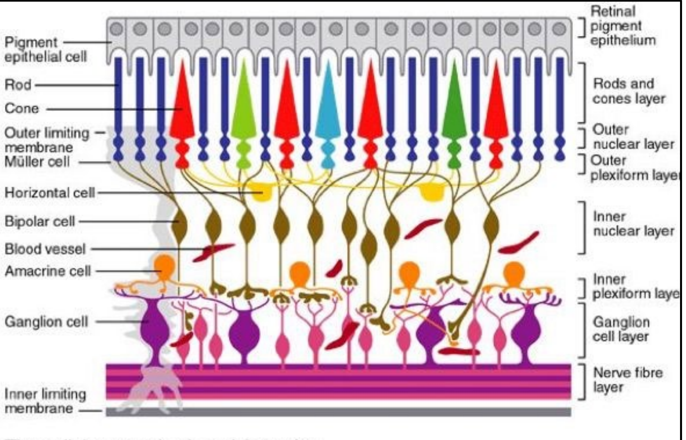
1. 人类视觉系统

视觉产生的生物基础

人类视觉系统与CNN的联系和区别，比如都有感受野、人眼没有卷积操作、CNN没有层内链接等；它们之间还有其他的联系和区别，不单发生在视网膜层次。

- **激活函数与神经元的输出**
- **侧抑制与局部响应归一化**
- **神经元的衰老与drop-out**
- **跨层链接与残差**
- **外侧膝状体处理轮廓、纹理、颜色和运动信息**





The cellular organisation of the retina

人类视觉系统

CNN数学原理

CNN推导

CNN发展历史

手写数字识别

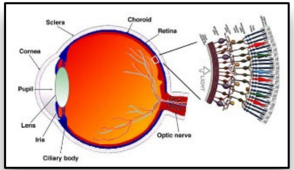
2. CNN数学原理

所见未必所得

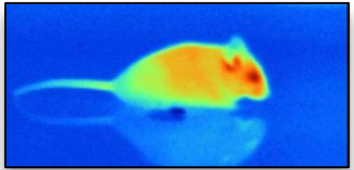
人类的视觉系统：实质上一套精密的光学系统；

动物的视觉系统：基于热成像的视觉系统，比如蛇类；

基于声音的成像系统，比如蝙蝠。

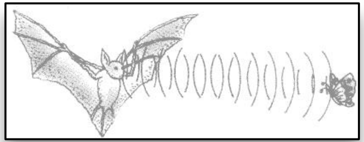


(a) 人类视觉系统



(b) 蛇类的热成像

也就是说，**生物所“看到”的景象并非是这个世界本来的面目，而是长期根据环境进化出来的适合自己生存的一种方式。**



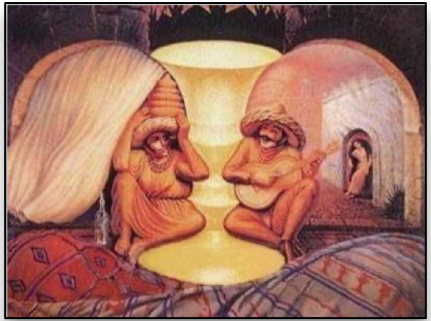
(c) 蝙蝠的回声定位

2. CNN数学原理

所见未必所得

“横看成岭侧成峰，远近高低各不同”

苏轼·题西林壁



对于物体被识别成什么，除了图片本身之外，还取决于你观察图片的方式（提取到了什么特征）。

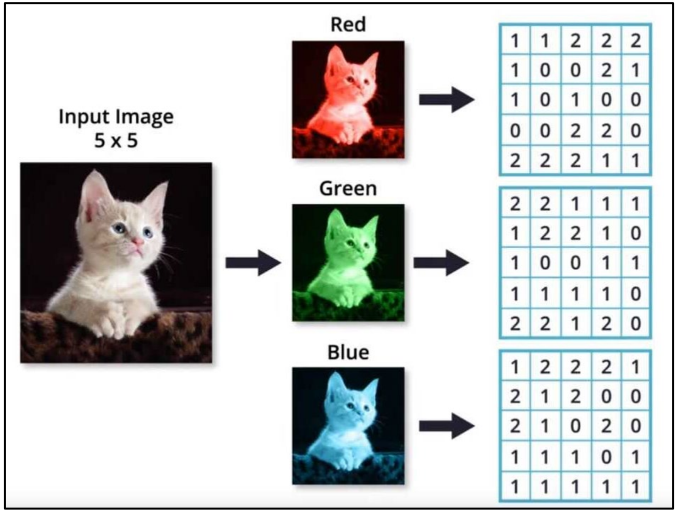
2. CNN数学原理

图像的数字表示

彩色数字图像实际上是一种用0-255的数值表示的二维数字矩阵，分为RGB三个颜色通道。

人类大脑物体识别：较好地处理图像不变性

神经网络结构：能否满足？



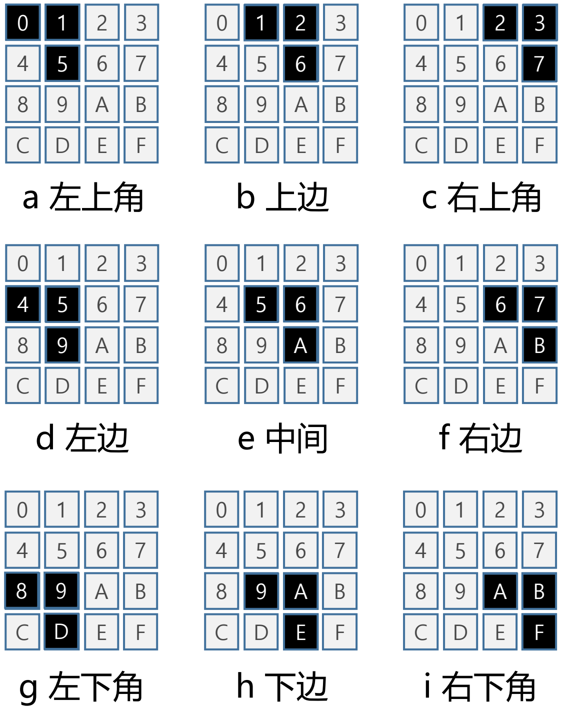
2. CNN数学原理

全连接神经网络

任务：一张4 × 4的图片，检测图像中是否有 形状的轮廓。

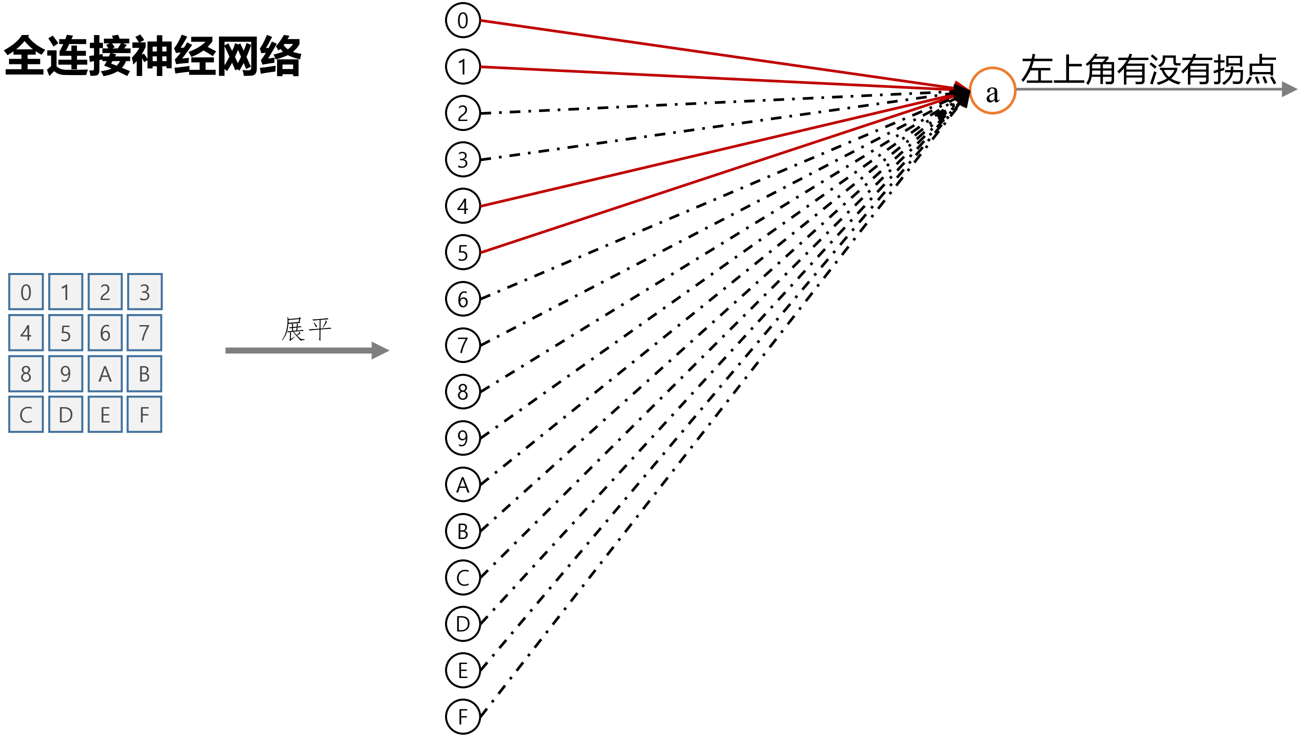
思路：该轮廓可能出现在图片的不同区域（共有9种可能出现的场景），如右图所示。那么，该任务等价于判断9个不同的区域是否存在“ ”形状的轮廓。

如果采用全连接神经网络模型的话，该任务对应的神经网络是输入层16个神经元，输出层9个神经元。每个输出层分别判断一个区域是否存在“ ”形状的轮廓。



2. CNN数学原理

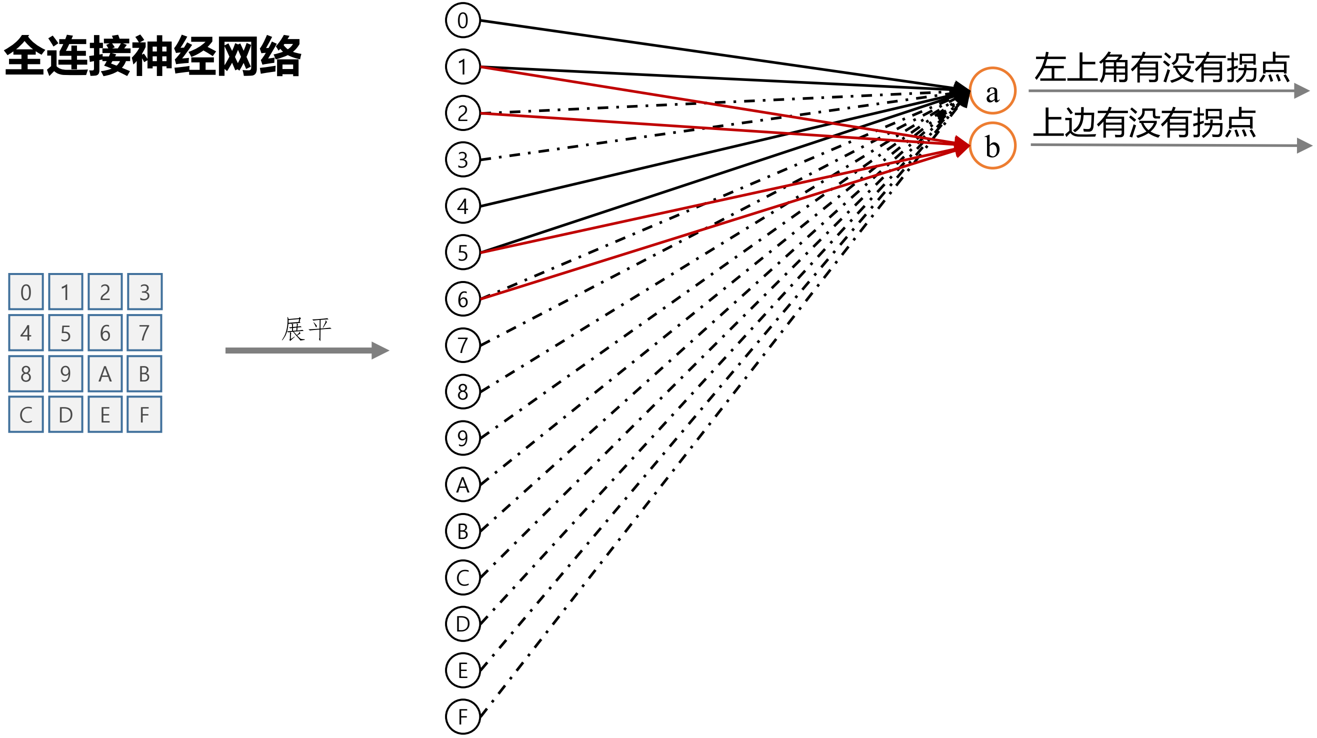
全连接神经网络



$$z_1 = W_1 x + b_1$$

2. CNN数学原理

全连接神经网络

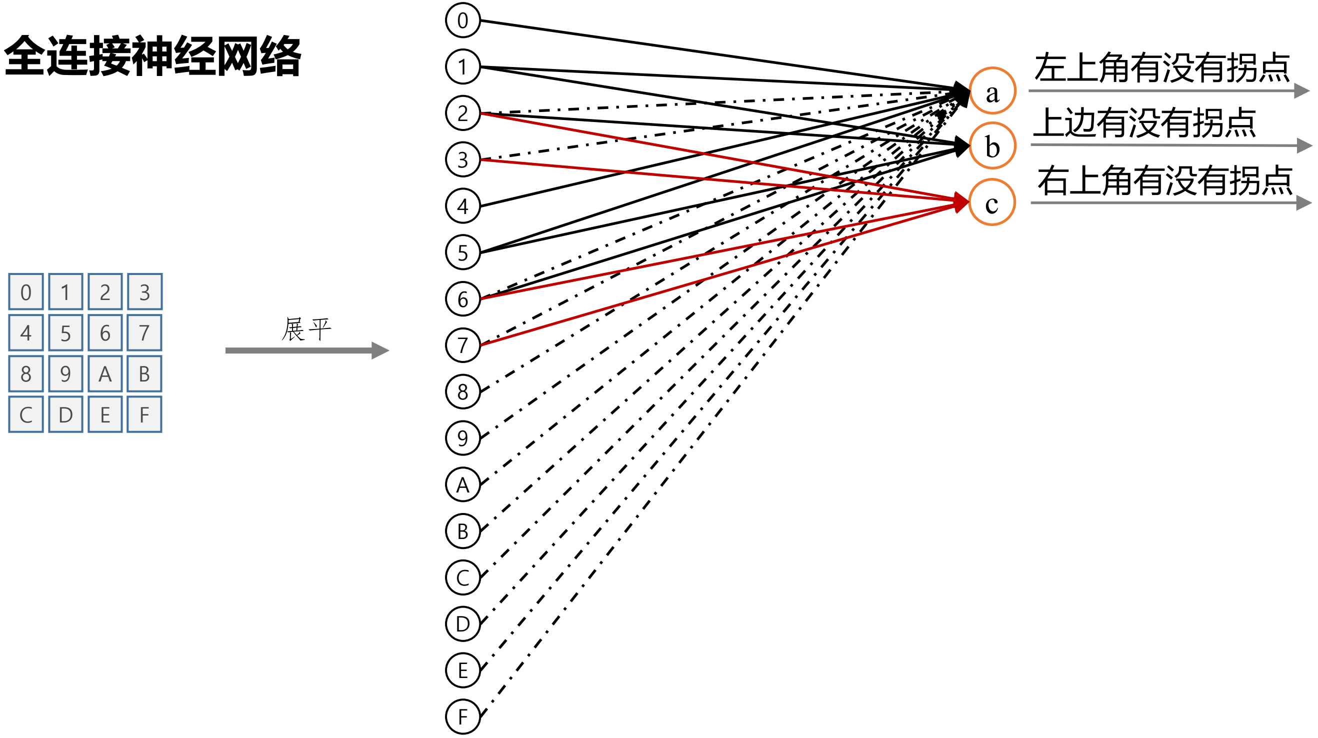


$$z_1 = W_1 x + b_1$$

$$z_2 = W_2 x + b_2$$

2. CNN数学原理

全连接神经网络



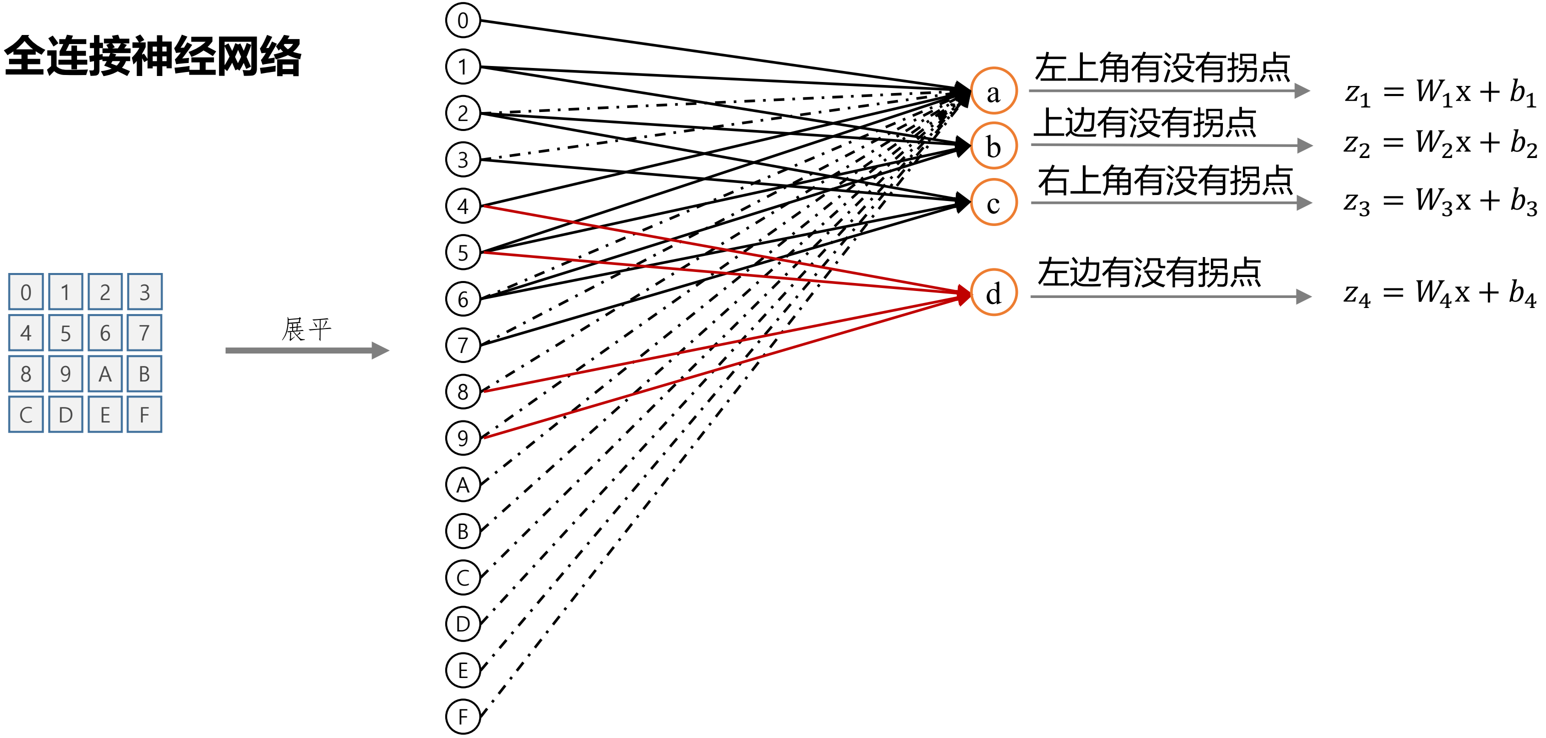
$z_1 = W_1 x + b_1$

$z_2 = W_2 x + b_2$

$z_3 = W_3 x + b_3$

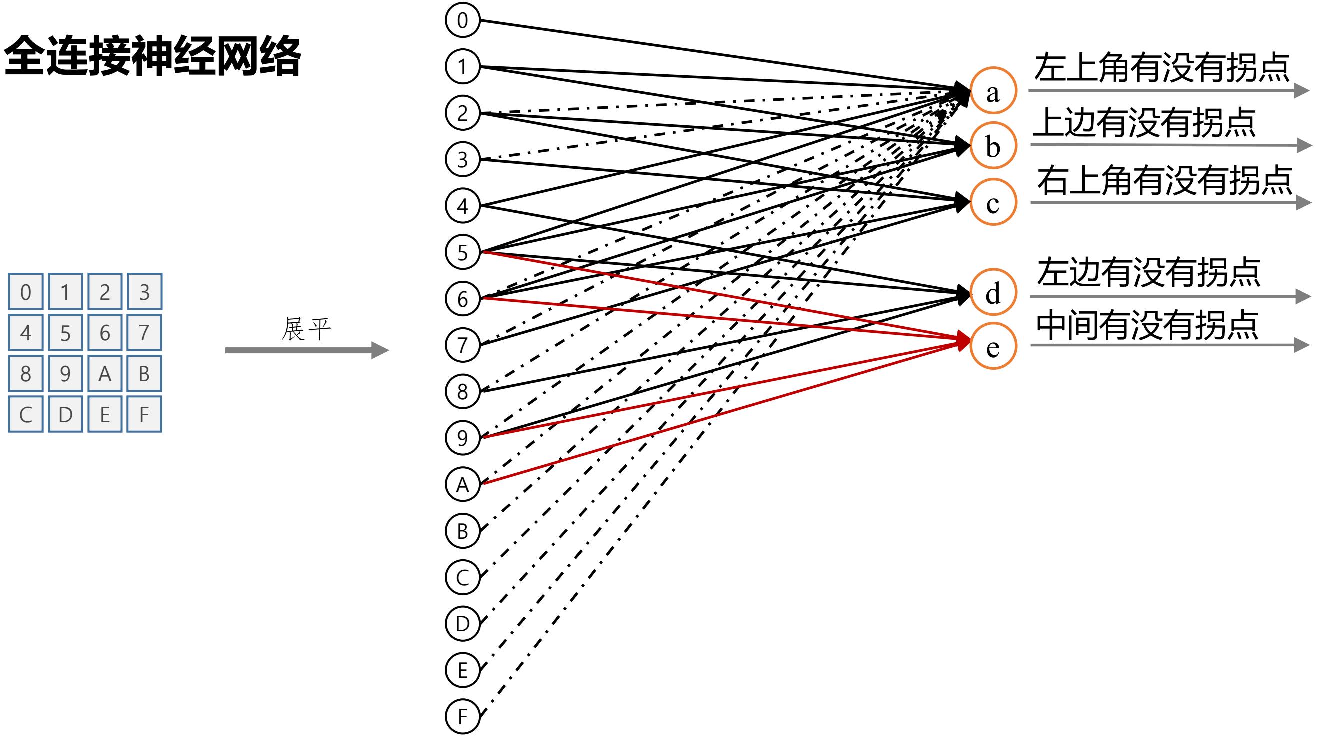
2. CNN数学原理

全连接神经网络



2. CNN数学原理

全连接神经网络



$$z_1 = W_1 x + b_1$$

$$z_2 = W_2 x + b_2$$

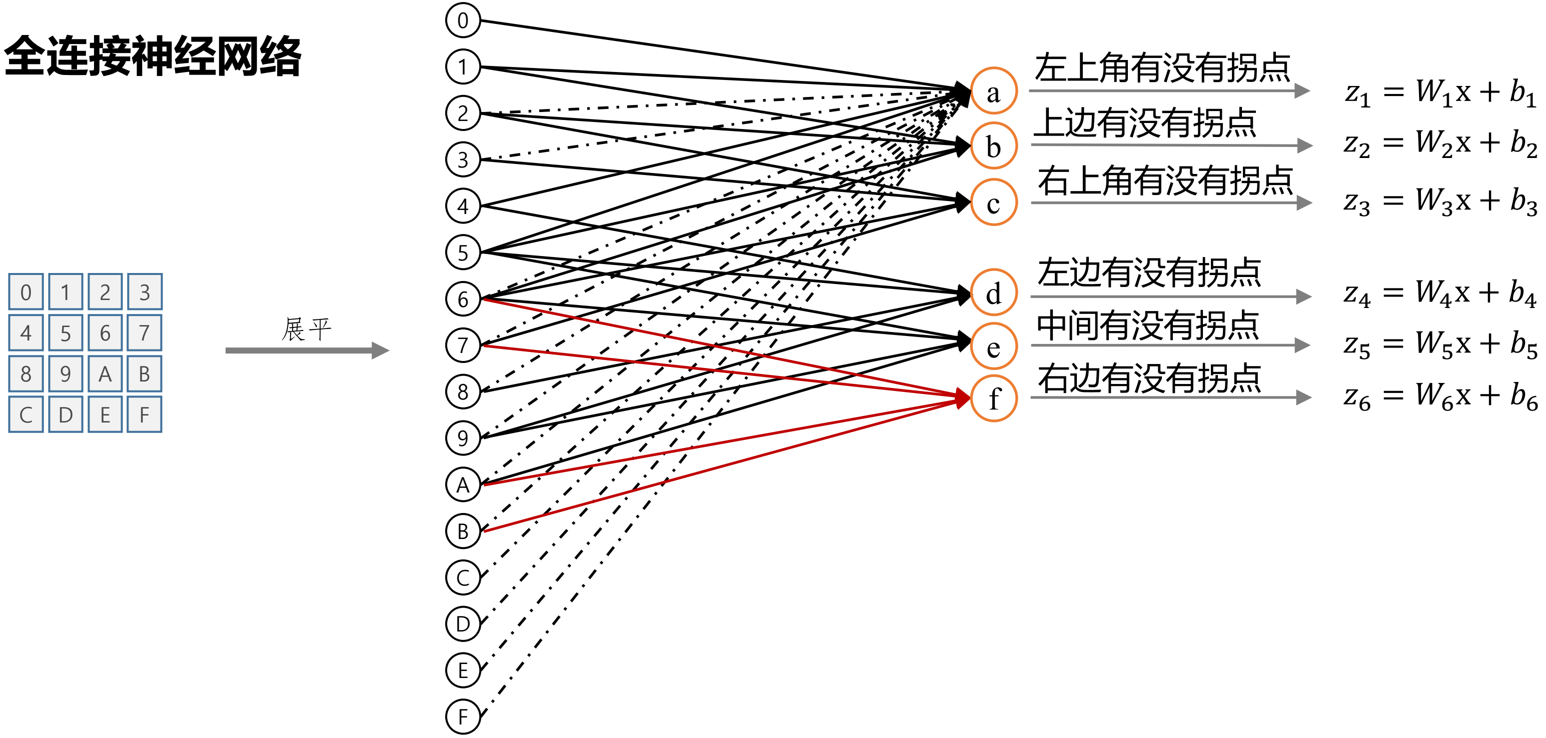
$$z_3 = W_3 x + b_3$$

$$z_4 = W_4 x + b_4$$

$$z_5 = W_5 x + b_5$$

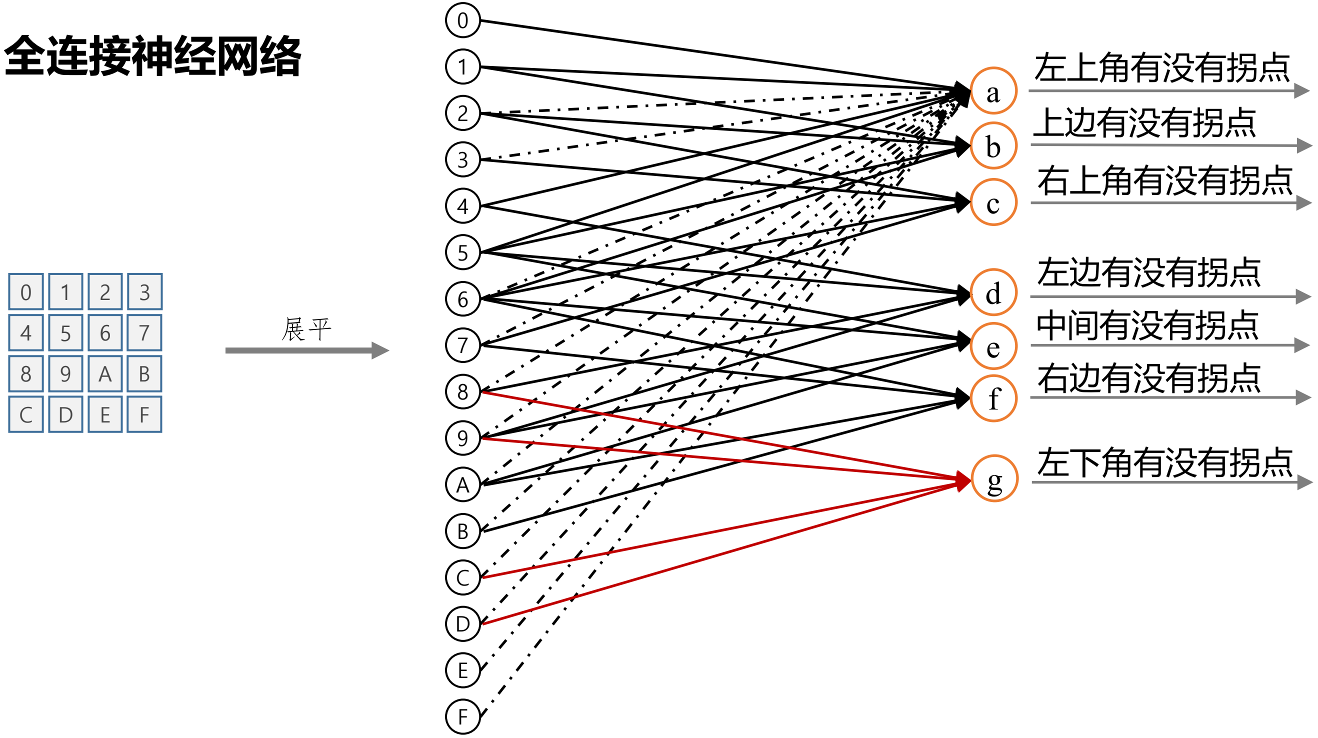
2. CNN数学原理

全连接神经网络



2. CNN数学原理

全连接神经网络



$$z_1 = W_1 x + b_1$$

$$z_2 = W_2 x + b_2$$

$$z_3 = W_3 x + b_3$$

$$z_4 = W_4 x + b_4$$

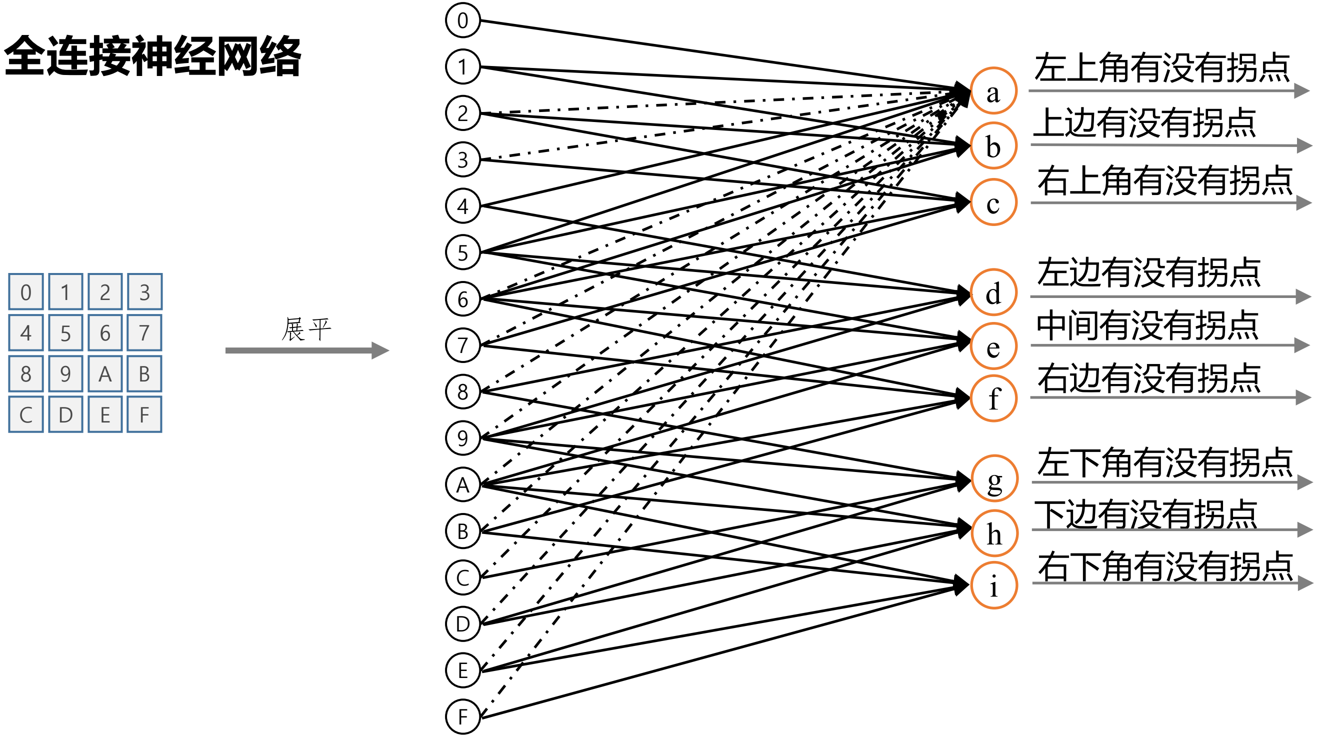
$$z_5 = W_5 x + b_5$$

$$z_6 = W_6 x + b_6$$

$$z_7 = W_7 x + b_7$$

2. CNN数学原理

全连接神经网络



$z_1 = W_1 x + b_1$

$z_2 = W_2 x + b_2$

$z_3 = W_3 x + b_3$

$z_4 = W_4 x + b_4$

$z_5 = W_5 x + b_5$

$z_6 = W_6 x + b_6$

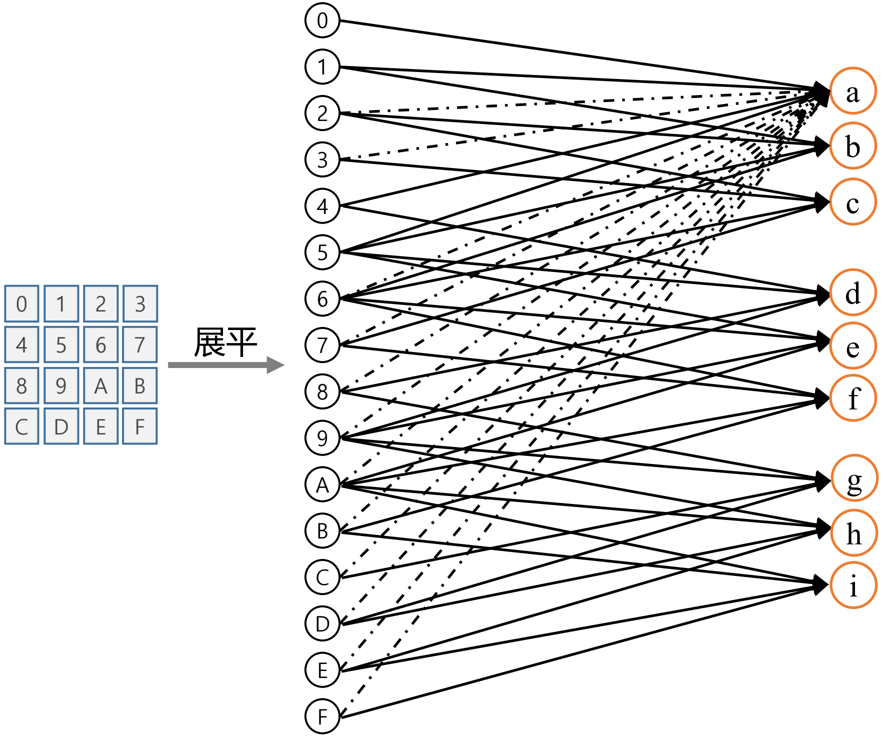
$z_7 = W_7 x + b_7$

$z_8 = W_8 x + b_8$

$z_9 = W_9 x + b_9$

2. CNN数学原理

全连接神经网络



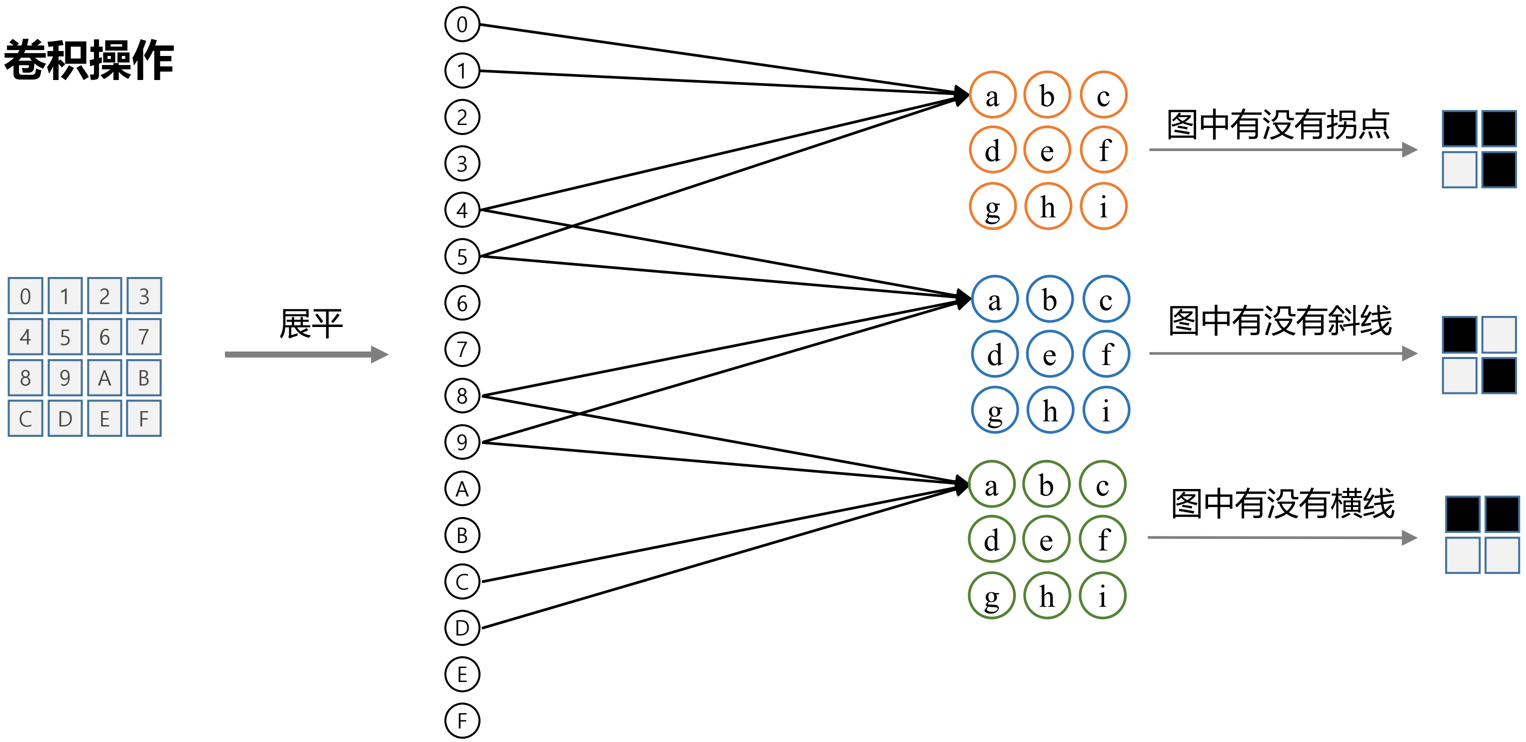
$$Wx + b = z$$

$$\begin{bmatrix} 1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1 \end{bmatrix} \times \begin{bmatrix} 0\\1\\2\\3\\4\\5\\6\\7\\8\\9\\A\\B\\C\\D\\E\\F \end{bmatrix} + b = \begin{bmatrix} a\\b\\c\\d\\e\\f\\g\\h\\i \end{bmatrix}$$

1表示对应位置的权重向量起作用；
0表示对应位置的权重向量不起作用。

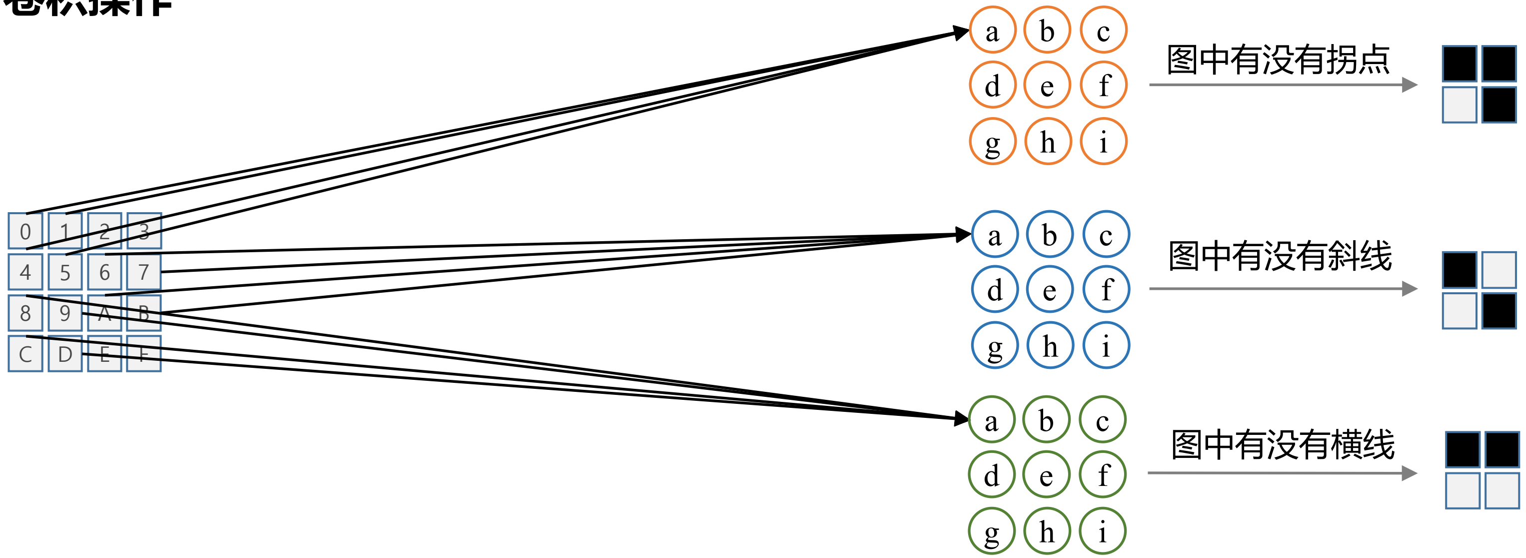
2. CNN数学原理

卷积操作



2. CNN数学原理

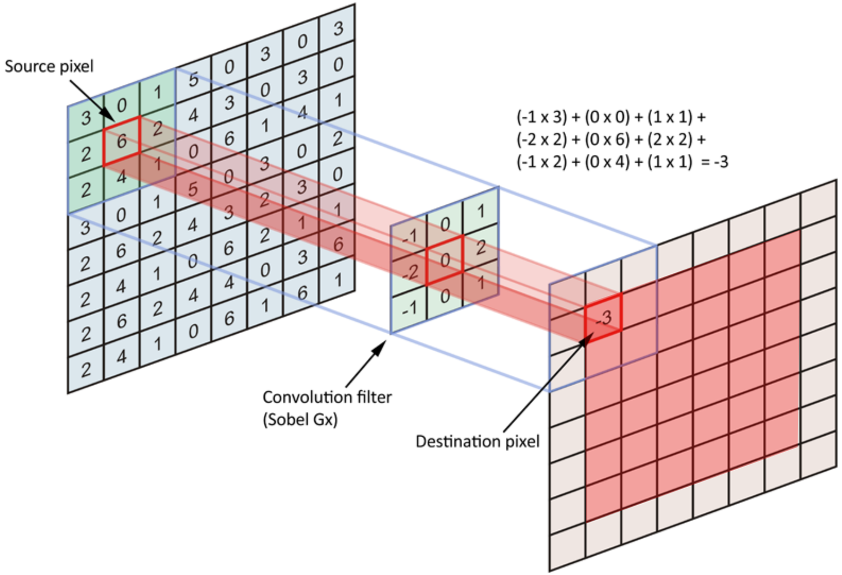
卷积操作

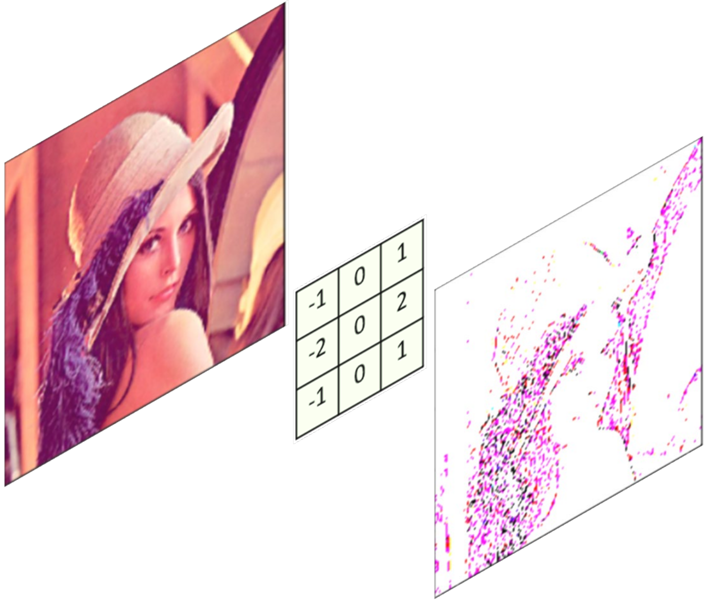


4×4的图像，最后得到3×3的结果

2. CNN数学原理

卷积操作

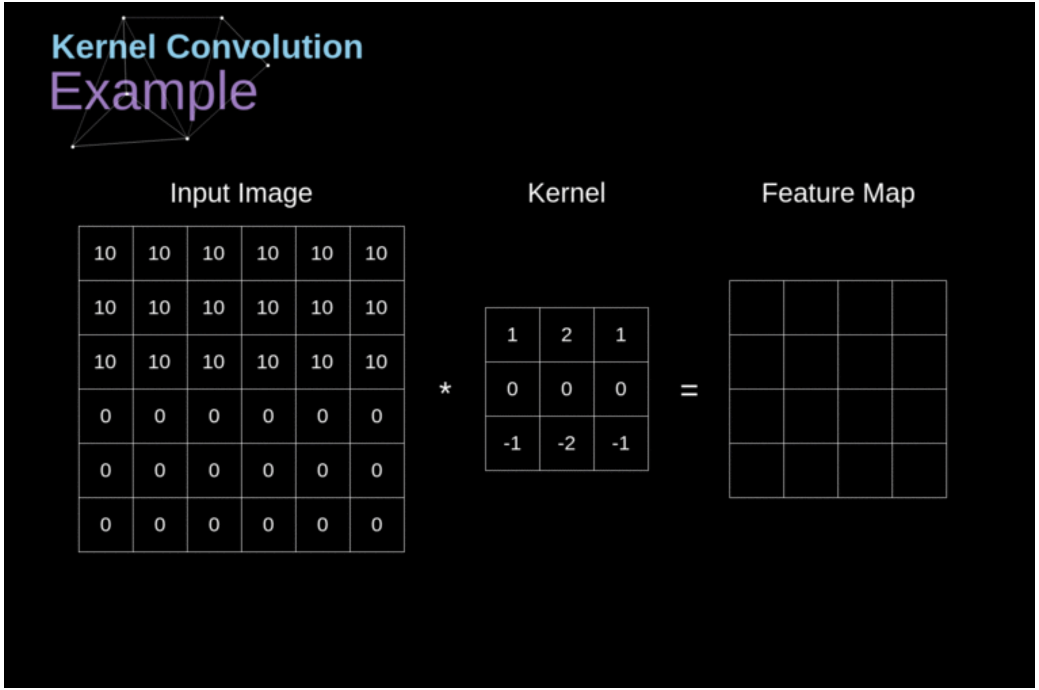




感受野(Local Receptive Fields)：卷积神经网络每一层输出的特征图(feature map)上的像素点在原始图像上映射的区域大小。

2. CNN数学原理

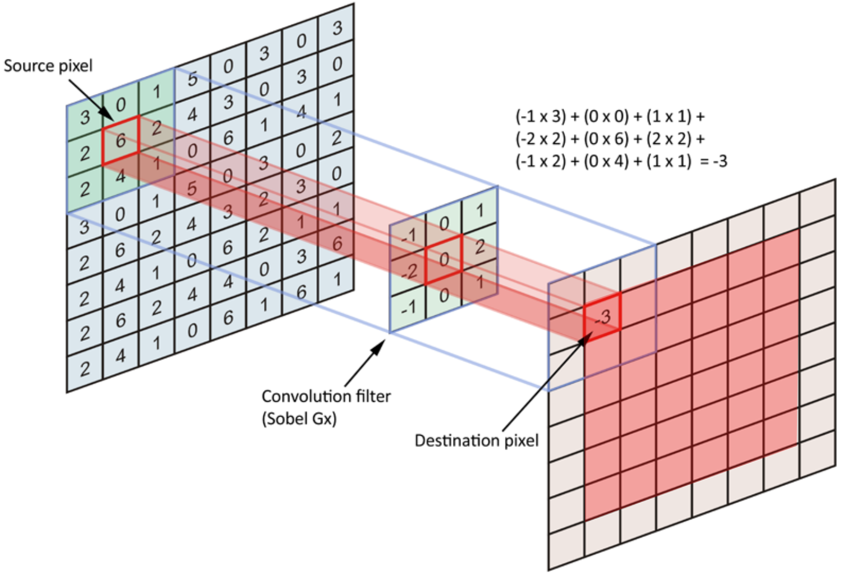
卷积操作



参考资料：[Gentle Dive into Math Behind Convolutional Neural Networks]

2. CNN数学原理

卷积操作



如果将特征图作为下一次卷积的输入，那么需要的卷积次数还是与特征图的特征数目（特征图像素数）差不多。

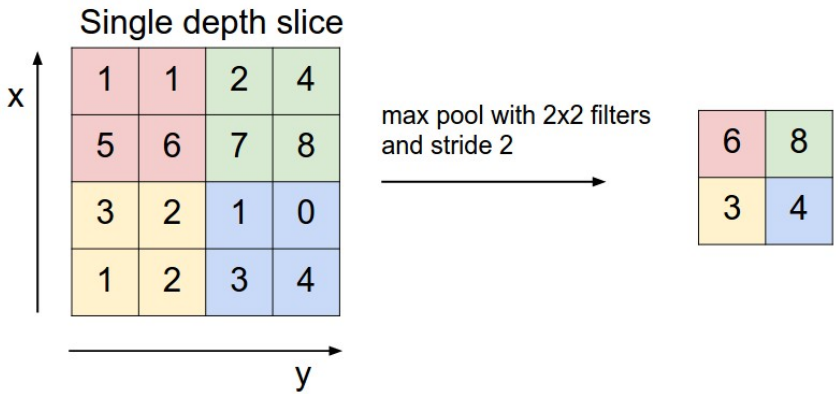
有没有一种有效地减少卷积次数的方法？

2. CNN数学原理

池化操作

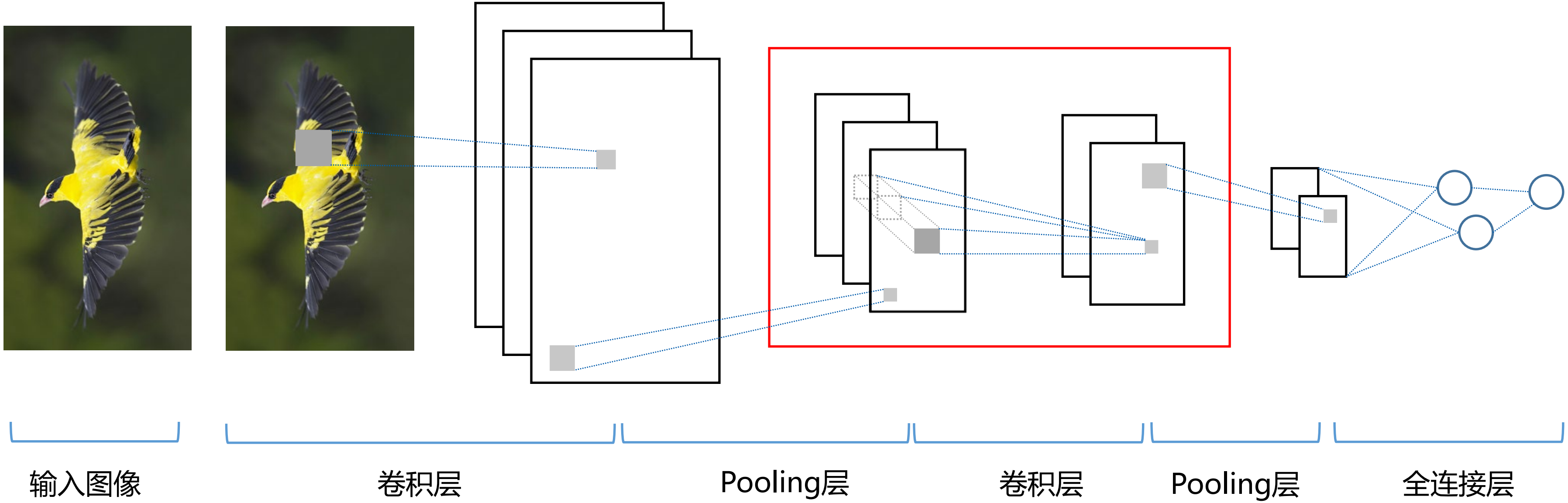
降低卷积次数，同时也降低了图像分辨率

- Max Pooling : 4个领域像素值取最大值
- Mean Pooling：4个像素值的平均值
- Gaussian Pooling: 使用高斯方法
- Trainable Pooling: 训练一个函数



2. CNN数学原理

卷积神经网络一般结构

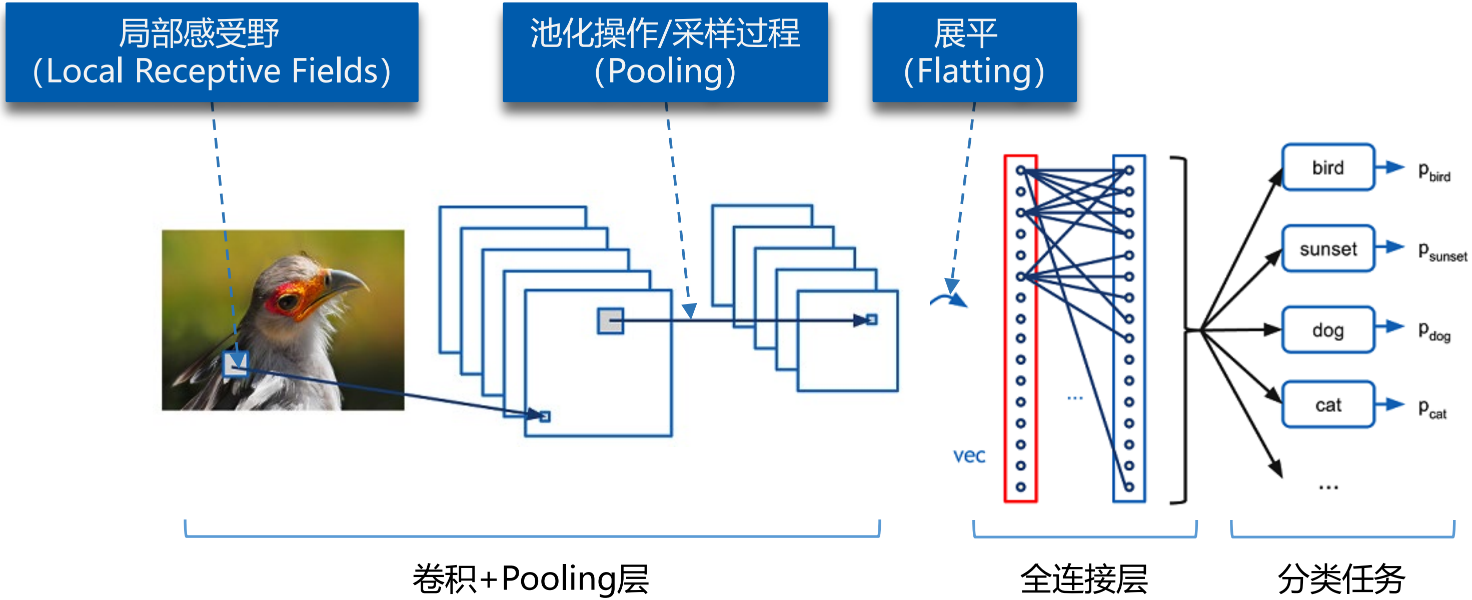


2. CNN数学原理

卷积神经网络一般结构

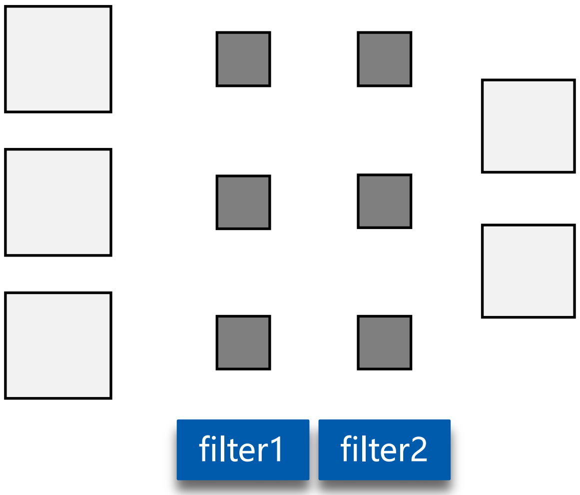
$$\textbf{Input} \rightarrow [\textbf{Conv} \times N \rightarrow \textbf{Pool} \times M] \times L \rightarrow \textbf{FC} \times K$$

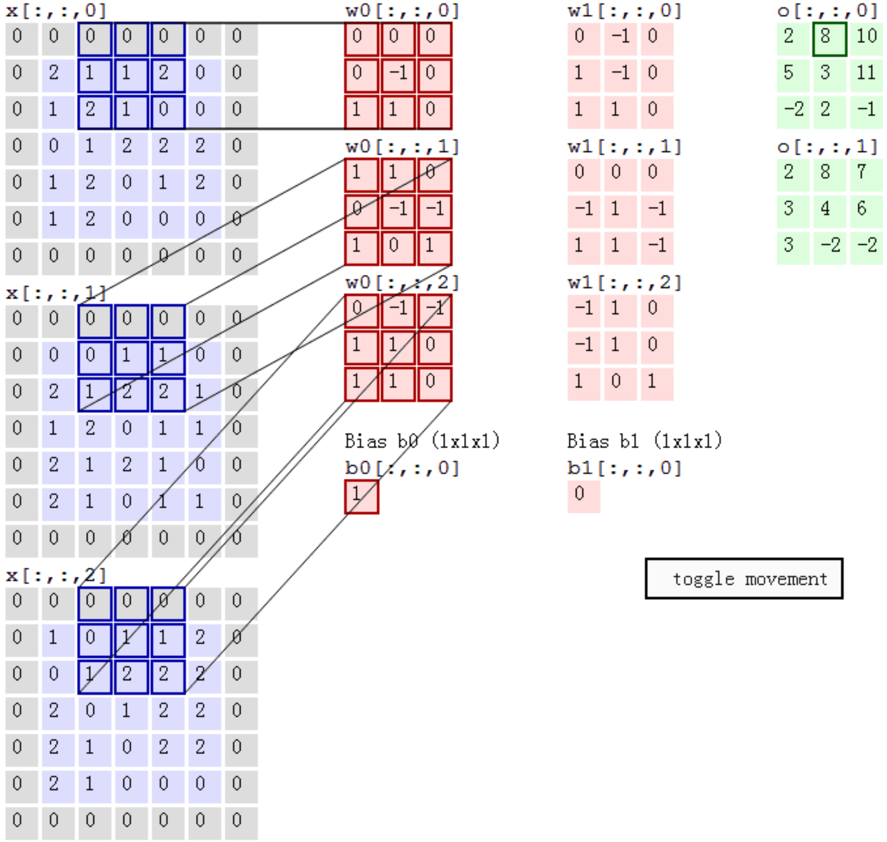
$$N \geq 1 \quad M \geq 0 \quad L \geq 0 \quad K \geq 1$$



2. CNN数学原理

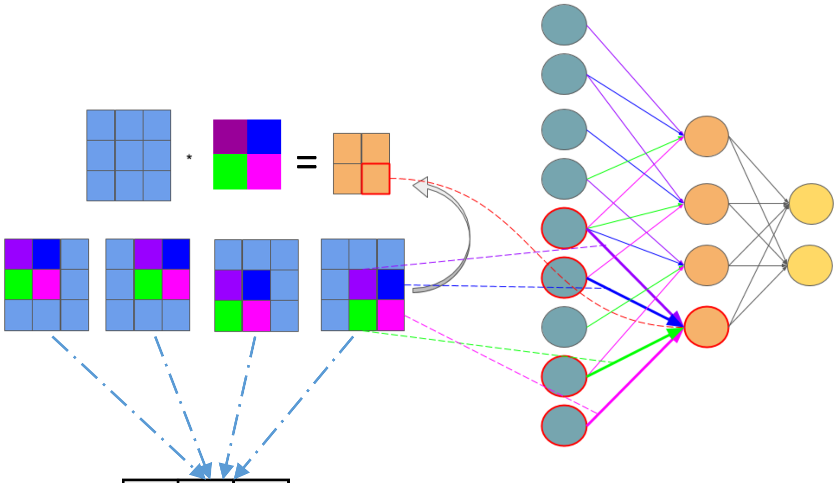
卷积神经网络一般结构





2. CNN数学原理

卷积神经网络一般结构



1	2	1
2	4	2
1	2	1

右图数字代表着这个位置的元素在Filter移动的时候参与了几次计算，也就是前向传播时，原始图像（特征图）像素在不同的位置中起过几次作用。

梯度反向传播的时候有用处

人类视觉系统

CNN数学原理

CNN推导

CNN发展历史

手写数字识别

3. CNN推导

绿色的：输入的图像/特征图，用$x_{i,j}$表示第i行第j列的像素值；

黄色的：某一个filter，权值W，用$w_{\mathrm{m,n}}$表示第m行第n列权值，w_b表示的是filter的偏置项；

红色的：卷积后的结果，用$a_{i,j}$表示第i行第j列的像素值。

$$a_{i,j} = f\left(\sum_{m=0}^{2}\sum_{n=0}^{2} w_{m,n} x_{i+m,j+n} + w_b\right)$$

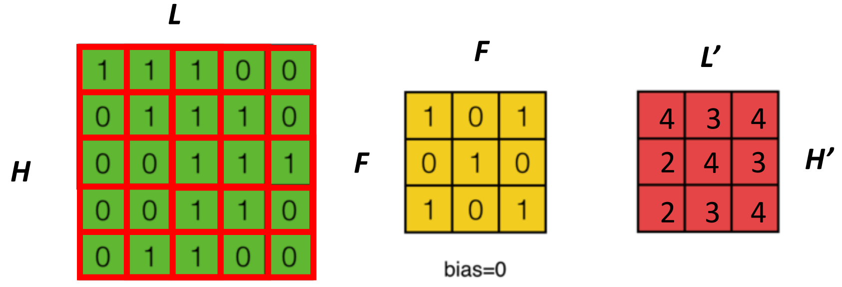


image 5*5 filter 3*3 feature map 3*3

卷积核每次在图像(特征图)上面按照从左向右从上到下的顺序扫描。每次扫描的时候可以看做是将整个卷积核范围的所有像素按照对应权重加权求和之后的结果放到了新生成的图像上同位置。

但是，这样就会带来像素的损失（原先图像的最外层一圈像素没有当成中心点）。

$$L' = (L - F) + 1$$
$$H' = (H - F) + 1$$

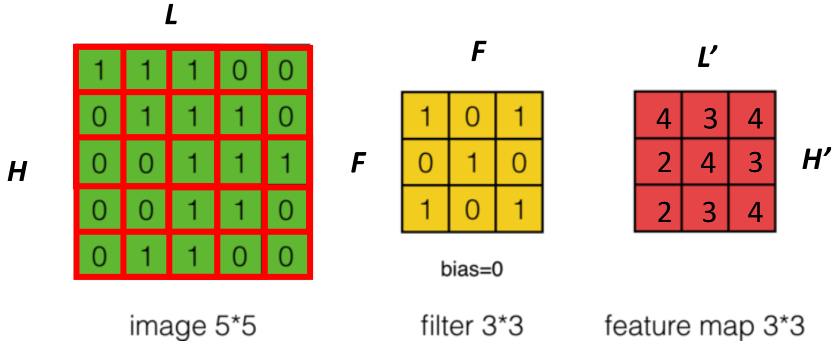
3. CNN推导

绿色的： 输入的图像/特征图，用$x_{i,j}$表示第i行第j列的像素值；

黄色的： 某一个filter，权值W，用$w_{m,n}$表示第m行第n列权值，w_b表示的是filter的偏置项；

红色的： 卷积后的结果，用$a_{i,j}$表示第i行第j列的像素值。

$$a_{i,j} = f(\sum_{m=0}^{2}\sum_{n=0}^{2} w_{m,n}x_{i+m,j+n} + w_b)$$



步长(Step)： 当卷积核扫描按照原先的顺序，但是每次移动的步长(S)不是1个像素的时候。此种情况下卷积后的feature map相比之前步长为1的时候会缺少一些取值。

缺少的这些值就是卷积核跨过的那些位置所能卷积出来的结果。

$$L' = (L - F)/S + 1$$
$$H' = (H - F)/S + 1$$

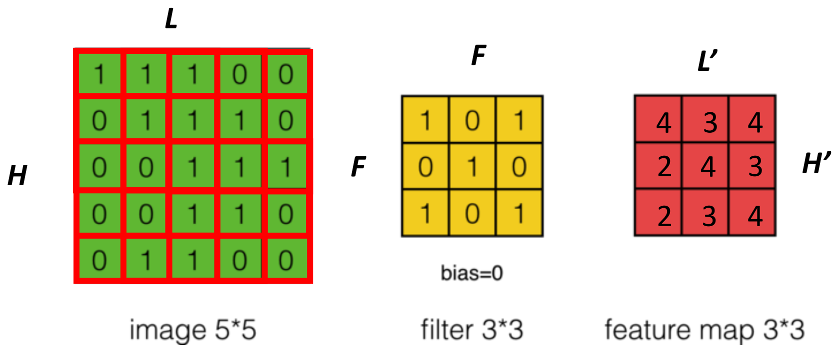
3. CNN推导

绿色的： 输入的图像/特征图，用$x_{i,j}$表示第i行第j列的像素值；

黄色的： 某一个filter，权值W，用$w_{\mathrm{m,n}}$表示第m行第n列权值，w_b表示的是filter的偏置项；

红色的： 卷积后的结果，用$a_{i,j}$表示第i行第j列的像素值。

$$a_{i,j} = f\left(\sum_{m=0}^{2}\sum_{n=0}^{2} w_{m,n} x_{i+m,j+n} + w_b\right)$$



补齐(Padding)： 对于损失的最外圈的元素来说，如果我们想让卷积之后的feature map的大小和之前的图像(feature map)的大小一致的话，那么通过公式我们知道长宽还需要增加 $F-1$个像素。这些像素可以在图像的四周补齐。Padding的大小为 $P = (F-1)/2$

$$L' = (L - F + 2P) + 1$$
$$H' = (H - F + 2P) + 1$$

3. CNN推导

绿色的： 输入的图像/特征图，用$x_{i,j}$表示第i行第j列的像素值；

黄色的： 某一个filter，权值W，用$w_{\mathrm{m,n}}$表示第m行第n列权值，w_b表示的是filter的偏置项；

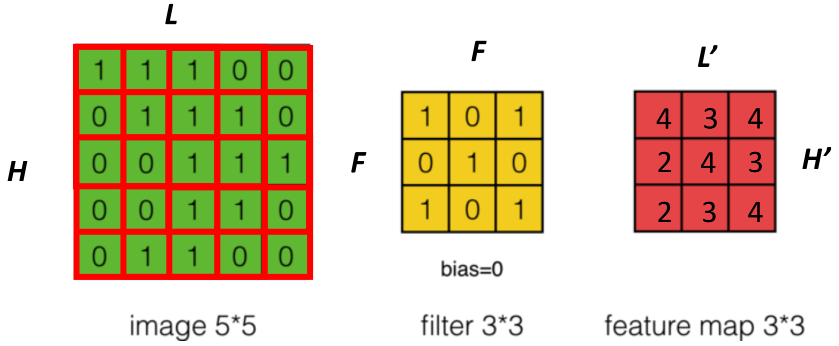
红色的： 卷积后的结果，用$a_{i,j}$表示第i行第j列的像素值。

$\boldsymbol{L/H}$: 原图的像素宽/长度

$\boldsymbol{F}$: 卷积核的大小

$\boldsymbol{P}$: 原图padding的个数

$\boldsymbol{S}$: 卷积核移动步长



卷积后feature map的大小：
$$L' = (L - F + 2P)/S + 1$$
$$H' = (H - F + 2P)/S + 1$$

$$a_{i,j} = f\left(\boxed{\sum_{d=0}^{D-1}}\sum_{m=0}^{F-1}\sum_{n=0}^{F-1} w_{d,m,n}\, x_{d,i+m,j+n} + w_b\right)$$

本层是feature map的个数，也就是卷积核的深度

3. CNN推导

绿色的：输入的图像/特征图，用$x_{i,j}$表示第i行第j列的像素值；

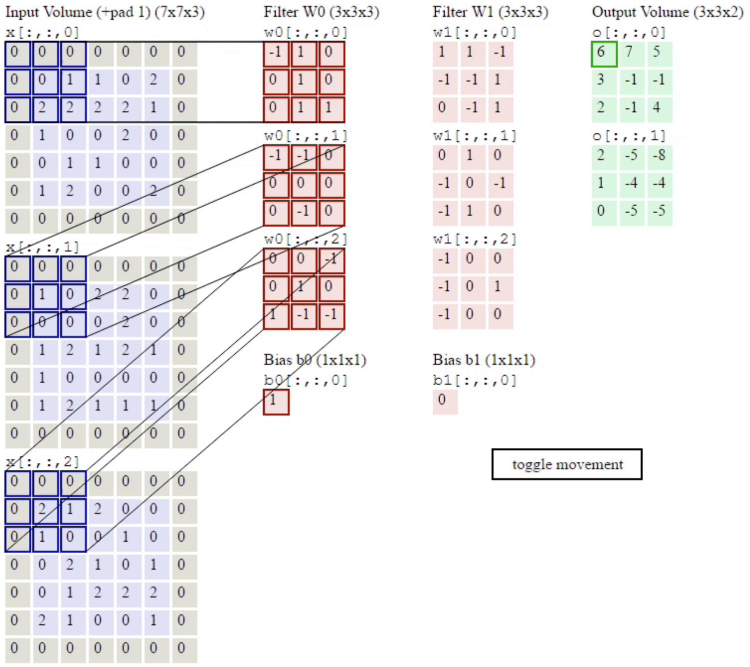
黄色的：某一个filter，权值W，用$w_{m,n}$表示第m行第n列权值，w_b表示的是filter的偏置项；

红色的：卷积后的结果，用$a_{i,j}$表示第i行第j列的像素值。

L/H: 原图的像素宽/长度；F: 卷积核的大小

P: 原图padding的个数；S: 卷积核移动步长

卷积后feature map的大小：
$$L' = (L - F + 2P)/S + 1$$
$$H' = (H - F + 2P)/S + 1$$

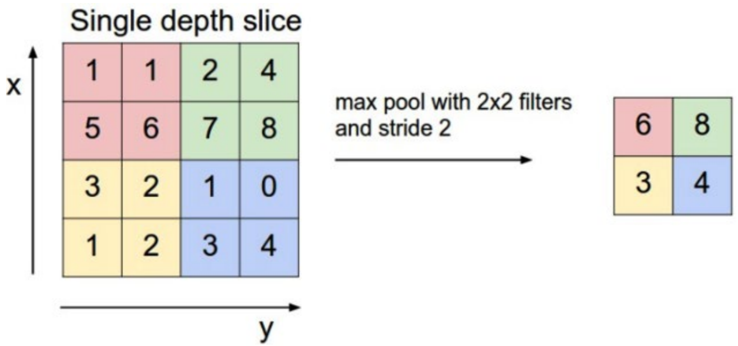


$$a_{i,j} = f\left(\boxed{\sum_{d=0}^{D-1}}\sum_{m=0}^{F-1}\sum_{n=0}^{F-1} w_{d,m,n}\, x_{d,i+m,j+n} + w_b\right)$$

本层是feature map的个数，也就是卷积核的深度

3. CNN推导

Pooling 层的作用：在保持图像信息的同时降低图像的分辨率，从而降低参数的个数。同时，pooling之后，在以后的卷积层中，每一个卷积能够探测到原来的图像区域加倍（如果使用的是2x2的pooling方式）。



Pooling 只会改变同一个feature map的大小，并不会对同一层其他的feature map产生影响。所以，pooling之后的feature map的深度还是相同的。

3. CNN推导

反向传播算法

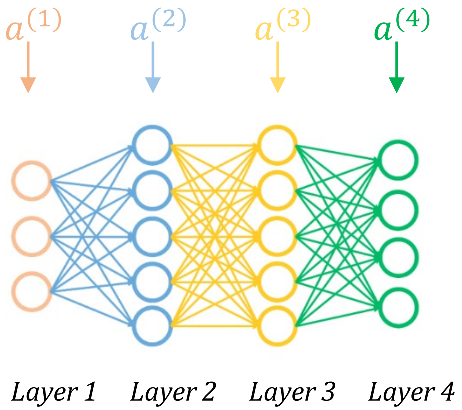
- 前向计算每个神经元的**输出值**$\boldsymbol{a_j}$（$\boldsymbol{j}$表示某层网络的第$\boldsymbol{j}$个神经元）。
- 反向计算每个神经元的**误差项**$\boldsymbol{\delta_j}$，也叫做**敏感度**(sensitivity)，其实际上是网络的损失E_d函数对神经元**加权输入**net_j的偏导数，即$\delta_j = \frac{\partial E_d}{\partial net_j}$。
- 计算每个神经元连接权重w_{ji}的**梯度**（w_{ji}表示从某层神经元i连接到下一层神经元j的权重），公式为 $\frac{\partial E_d}{\partial w_{ji}^{(l)}} = \delta_j^{(l)} a_i^{(l-1)}$

最后，根据梯度下降法则更新每个权重即可。

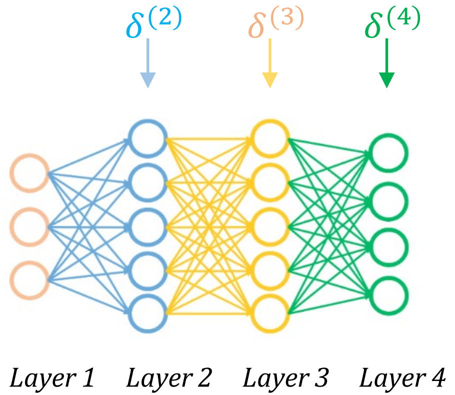
对于卷积神经网络，由于涉及到**局部连接**、**下采样**的等操作，影响到了第二步**误差项**的具体计算方法，而**权值共享**影响了第三步**权重的梯度**的计算方法。

卷积层和**Pooling层的训练算法。**

前向传播过程



反向传播过程



卷积层误差反传—最简单情况

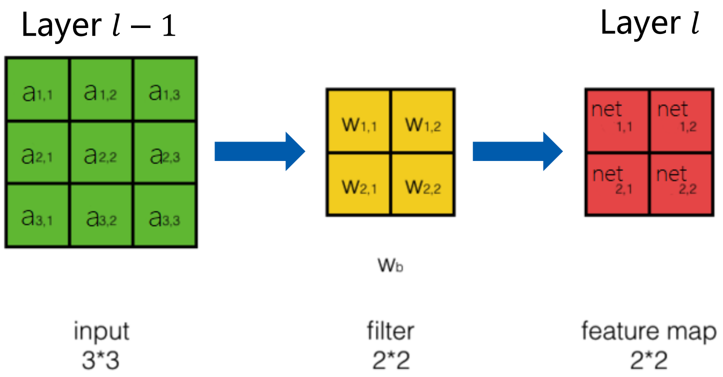
3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

- 前向计算

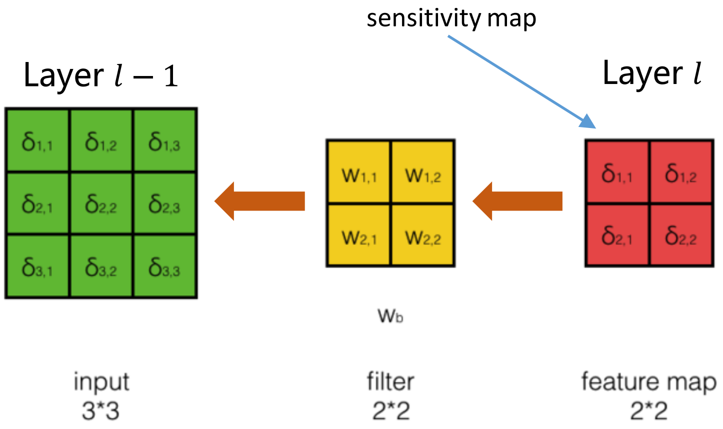
$$a_{i,j}^{l-1} = f^{l-1}(net_{i,j}^{l-1})$$

$$net^{l} = conv(W^{l}, a^{l-1}) + w_b$$



- 误差后向传播

$$\delta_{i,j}^{l-1} = \frac{\partial E_d}{\partial net_{i,j}^{l-1}} = \frac{\partial E_d}{\partial a_{i,j}^{l-1}} \frac{\partial a_{i,j}^{l-1}}{\partial net_{i,j}^{l-1}}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

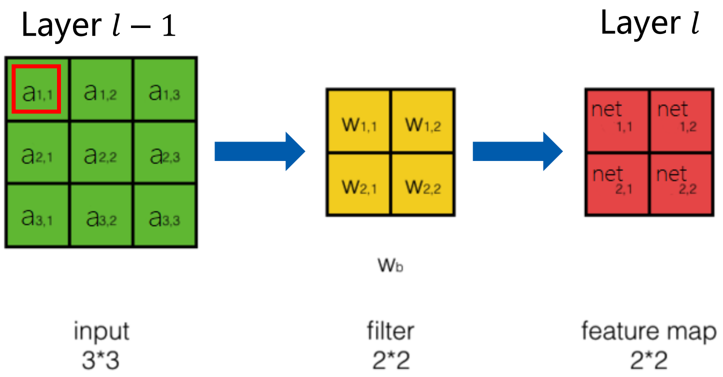
➢ 前向计算

$$net^{l}_{1,1} = w_{1,1}\boxed{a^{l-1}_{1,1}} + w_{1,2}a^{l-1}_{1,2} + w_{2,1}a^{l-1}_{2,1} + w_{2,2}a^{l-1}_{2,2} + w_b$$

$$net^{l}_{1,2} = w_{1,1}a^{l-1}_{1,2} + w_{1,2}a^{l-1}_{1,3} + w_{2,1}a^{l-1}_{2,2} + w_{2,2}a^{l-1}_{2,3} + w_b$$

$$net^{l}_{2,1} = w_{1,1}a^{l-1}_{2,1} + w_{1,2}a^{l-1}_{2,2} + w_{2,1}a^{l-1}_{3,1} + w_{2,2}a^{l-1}_{3,2} + w_b$$

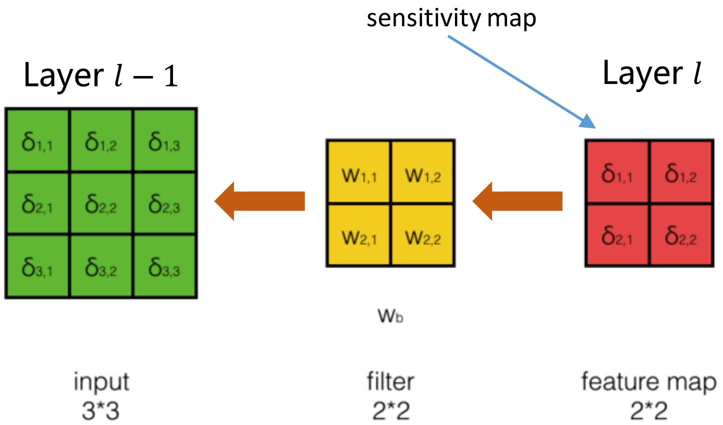
$$net^{l}_{2,2} = w_{1,1}a^{l-1}_{2,2} + w_{1,2}a^{l-1}_{2,3} + w_{2,1}a^{l-1}_{3,2} + w_{2,2}a^{l-1}_{3,3} + w_b$$



➢ 误差后向传播

$$\frac{\partial E_d}{\partial a^{l-1}_{1,1}} = \frac{\partial E_d}{\partial net^{l}_{1,1}} \frac{\partial net^{l}_{1,1}}{\partial a^{l-1}_{1,1}}$$

$$= \delta^{l}_{1,1} w_{1,1}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

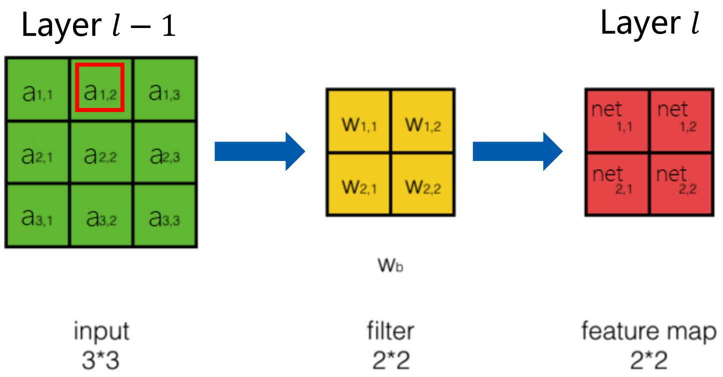
➢ 前向计算

$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}\boxed{a_{1,2}^{l-1}} + w_{2,1}a_{2,1}^{l-1} + w_{2,2}a_{2,2}^{l-1} + w_b$$

$$net_{1,2}^{l} = w_{1,1}\boxed{a_{1,2}^{l-1}} + w_{1,2}a_{1,3}^{l-1} + w_{2,1}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$

$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + w_{1,2}a_{2,2}^{l-1} + w_{2,1}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$

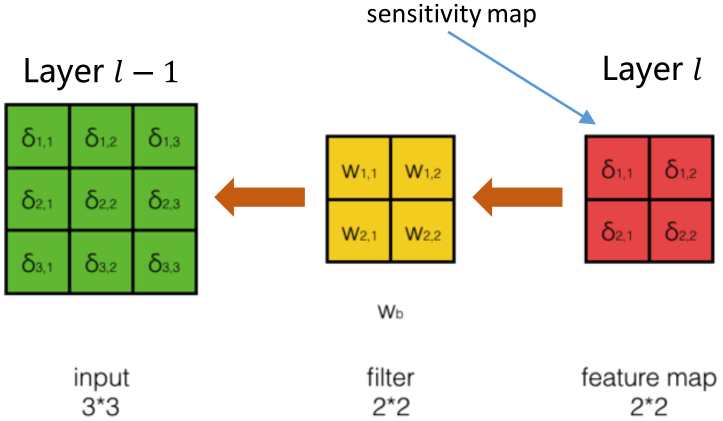
$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + w_{1,2}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$



➢ 误差后向传播

$$\frac{\partial E_d}{\partial a_{1,2}^{l-1}} = \frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial a_{1,2}^{l-1}} + \frac{\partial E_d}{\partial net_{1,2}^{l}}\frac{\partial net_{1,2}^{l}}{\partial a_{1,2}^{l-1}}$$

$$= \delta_{1,1}^{l}w_{1,2} + \delta_{1,2}^{l}w_{1,1}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

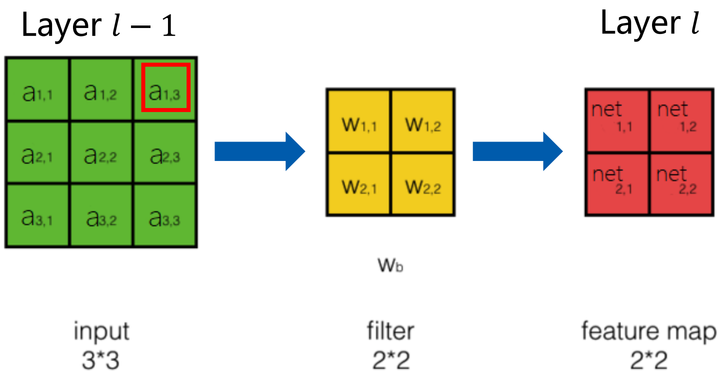
➢ 前向计算

$$net_{1,1}^{l}=w_{1,1}a_{1,1}^{l-1}+w_{1,2}a_{1,2}^{l-1}+w_{2,1}a_{2,1}^{l-1}+w_{2,2}a_{2,2}^{l-1}+w_{b}$$

$$net_{1,2}^{l}=w_{1,1}a_{1,2}^{l-1}+w_{1,2}\boxed{a_{1,3}^{l-1}}+w_{2,1}a_{2,2}^{l-1}+w_{2,2}a_{2,3}^{l-1}+w_{b}$$

$$net_{2,1}^{l}=w_{1,1}a_{2,1}^{l-1}+w_{1,2}a_{2,2}^{l-1}+w_{2,1}a_{3,1}^{l-1}+w_{2,2}a_{3,2}^{l-1}+w_{b}$$

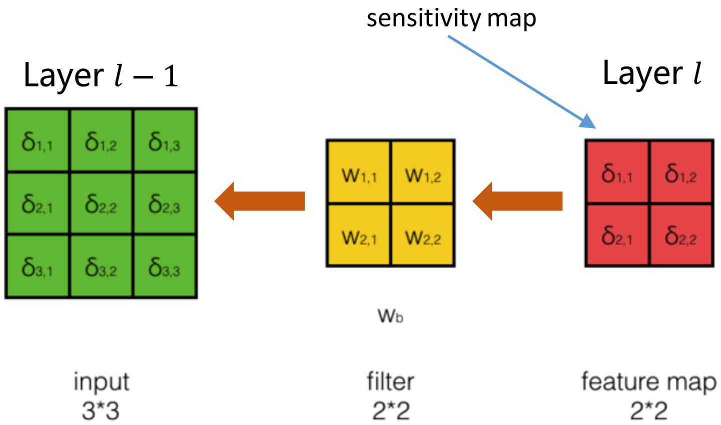
$$net_{2,2}^{l}=w_{1,1}a_{2,2}^{l-1}+w_{1,2}a_{2,3}^{l-1}+w_{2,1}a_{3,2}^{l-1}+w_{2,2}a_{3,3}^{l-1}+w_{b}$$



➢ 误差后向传播

$$\frac{\partial E_d}{\partial a_{1,3}^{l-1}}=\frac{\partial E_d}{\partial net_{1,2}^{l}}\frac{\partial net_{1,2}^{l}}{\partial a_{1,3}^{l-1}}$$

$$=\delta_{1,2}^{l}w_{1,2}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

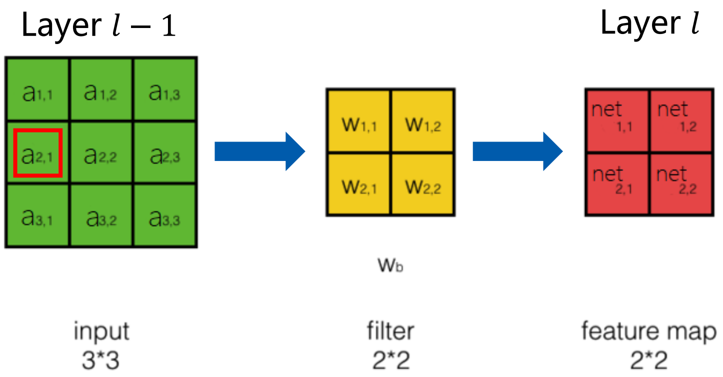
➢ 前向计算

$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}a_{1,2}^{l-1} + w_{2,1}\boxed{a_{2,1}^{l-1}} + w_{2,2}a_{2,2}^{l-1} + w_b$$

$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + w_{1,2}a_{1,3}^{l-1} + w_{2,1}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$

$$net_{2,1}^{l} = w_{1,1}\boxed{a_{2,1}^{l-1}} + w_{1,2}a_{2,2}^{l-1} + w_{2,1}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$

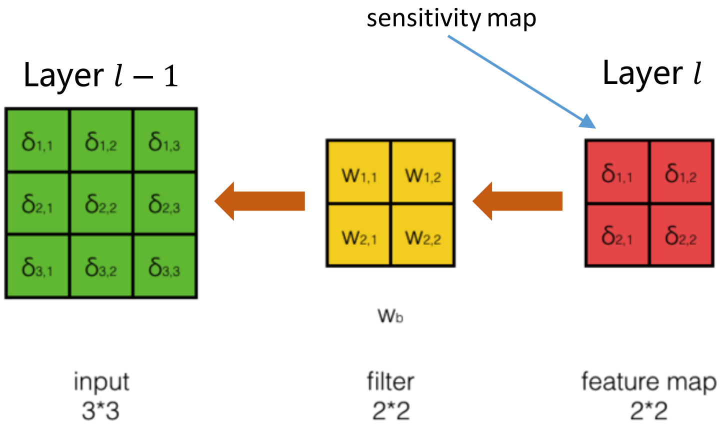
$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + w_{1,2}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$



➢ 误差后向传播

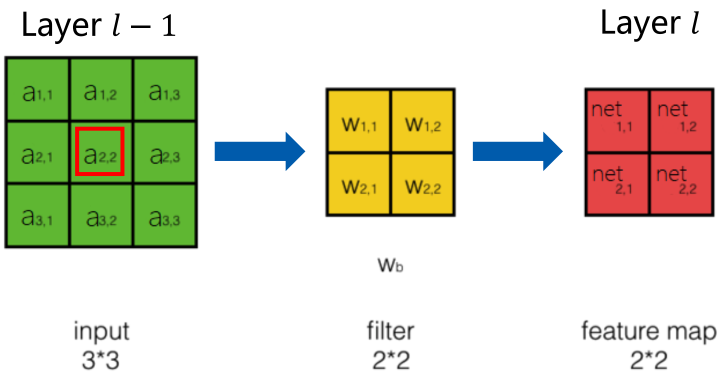
$$\frac{\partial E_d}{\partial a_{2,1}^{l-1}} = \frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial a_{2,1}^{l-1}} + \frac{\partial E_d}{\partial net_{2,1}^{l}}\frac{\partial net_{2,1}^{l}}{\partial a_{2,1}^{l-1}}$$

$$= \delta_{1,1}^{l} w_{2,1} + \delta_{2,1}^{l} w_{1,1}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1



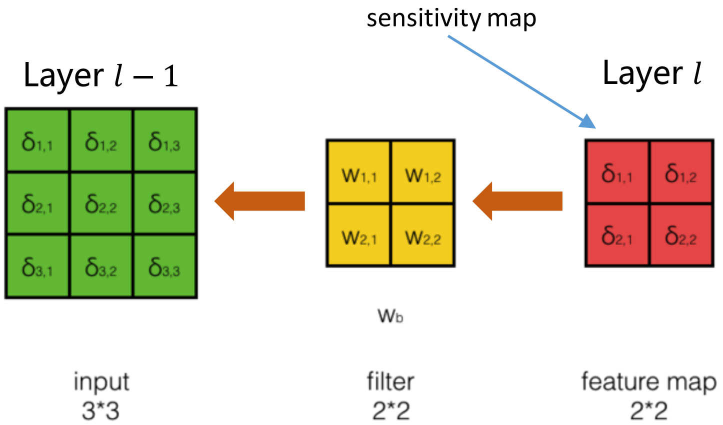
➢ 前向计算

$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}a_{1,2}^{l-1} + w_{2,1}a_{2,1}^{l-1} + w_{2,2}\boxed{a_{2,2}^{l-1}} + w_b$$

$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + w_{1,2}a_{1,3}^{l-1} + w_{2,1}\boxed{a_{2,2}^{l-1}} + w_{2,2}a_{2,3}^{l-1} + w_b$$

$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + w_{1,2}\boxed{a_{2,2}^{l-1}} + w_{2,1}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$

$$net_{2,2}^{l} = w_{1,1}\boxed{a_{2,2}^{l-1}} + w_{1,2}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$



➢ 误差后向传播

$$\frac{\partial E_d}{\partial a_{2,2}^{l-1}}$$

$$= \frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial a_{2,2}^{l-1}} + \frac{\partial E_d}{\partial net_{1,2}^{l}}\frac{\partial net_{1,2}^{l}}{\partial a_{2,2}^{l-1}} + \frac{\partial E_d}{\partial net_{2,1}^{l}}\frac{\partial net_{2,1}^{l}}{\partial a_{2,2}^{l-1}} + \frac{\partial E_d}{\partial net_{2,2}^{l}}\frac{\partial net_{2,2}^{l}}{\partial a_{2,2}^{l-1}}$$

$$= \delta_{1,1}^{l}w_{2,2} + \delta_{1,2}^{l}w_{2,1} + \delta_{2,1}^{l}w_{1,2} + \delta_{2,2}^{l}w_{1,1}$$

3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

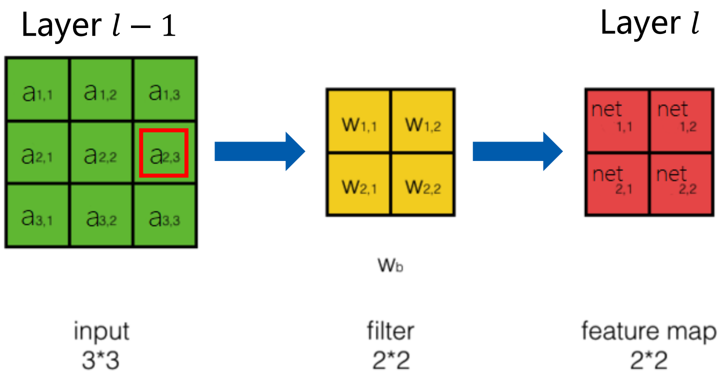
➤ 前向计算

$$net^l_{1,1} = w_{1,1}a^{l-1}_{1,1} + w_{1,2}a^{l-1}_{1,2} + w_{2,1}a^{l-1}_{2,1} + w_{2,2}a^{l-1}_{2,2} + w_b$$

$$net^l_{1,2} = w_{1,1}a^{l-1}_{1,2} + w_{1,2}a^{l-1}_{1,3} + w_{2,1}a^{l-1}_{2,2} + w_{2,2}\boxed{a^{l-1}_{2,3}} + w_b$$

$$net^l_{2,1} = w_{1,1}a^{l-1}_{2,1} + w_{1,2}a^{l-1}_{2,2} + w_{2,1}a^{l-1}_{3,1} + w_{2,2}a^{l-1}_{3,2} + w_b$$

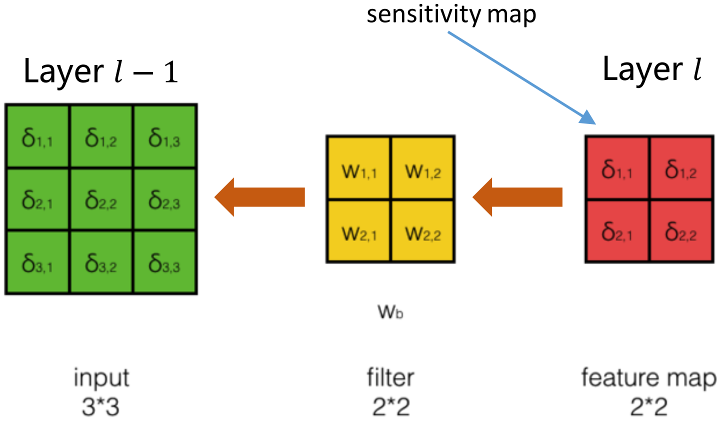
$$net^l_{2,2} = w_{1,1}a^{l-1}_{2,2} + w_{1,2}\boxed{a^{l-1}_{2,3}} + w_{2,1}a^{l-1}_{3,2} + w_{2,2}a^{l-1}_{3,3} + w_b$$



➤ 误差后向传播

$$\frac{\partial E_d}{\partial a^{l-1}_{2,3}} = \frac{\partial E_d}{\partial net^l_{1,2}}\frac{\partial net^l_{1,2}}{\partial a^{l-1}_{2,3}} + \frac{\partial E_d}{\partial net^l_{2,2}}\frac{\partial net^l_{2,2}}{\partial a^{l-1}_{2,3}}$$

$$= \delta^l_{1,2}w_{2,2} + \delta^l_{2,2}w_{1,2}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

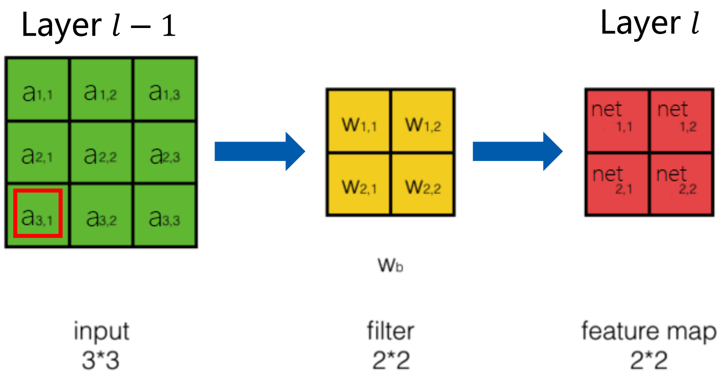
➢ 前向计算

$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}a_{1,2}^{l-1} + w_{2,1}a_{2,1}^{l-1} + w_{2,2}a_{2,2}^{l-1} + w_b$$

$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + w_{1,2}a_{1,3}^{l-1} + w_{2,1}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$

$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + w_{1,2}a_{2,2}^{l-1} + w_{2,1}\boxed{a_{3,1}^{l-1}} + w_{2,2}a_{3,2}^{l-1} + w_b$$

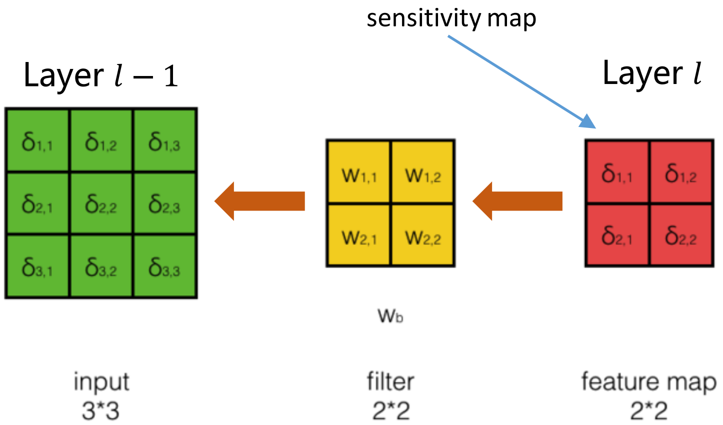
$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + w_{1,2}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$



➢ 误差后向传播

$$\frac{\partial E_d}{\partial a_{3,1}^{l-1}} = \frac{\partial E_d}{\partial net_{2,1}^{l}} \frac{\partial net_{2,1}^{l}}{\partial a_{3,1}^{l-1}}$$

$$= \delta_{2,1}^{l} w_{2,1}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

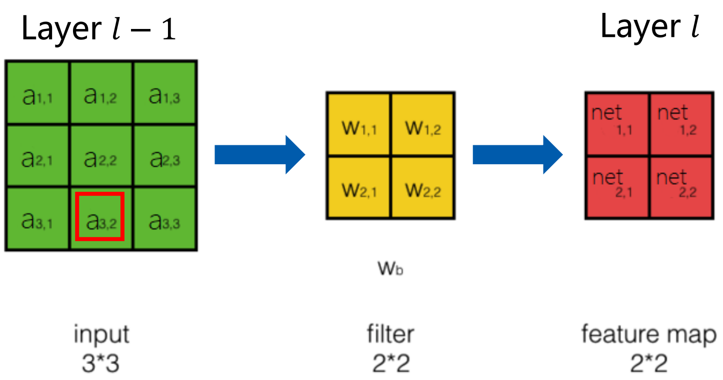
➢ 前向计算

$$net^{l}_{1,1} = w_{1,1}a^{l-1}_{1,1} + w_{1,2}a^{l-1}_{1,2} + w_{2,1}a^{l-1}_{2,1} + w_{2,2}a^{l-1}_{2,2} + w_b$$

$$net^{l}_{1,2} = w_{1,1}a^{l-1}_{1,2} + w_{1,2}a^{l-1}_{1,3} + w_{2,1}a^{l-1}_{2,2} + w_{2,2}a^{l-1}_{2,3} + w_b$$

$$net^{l}_{2,1} = w_{1,1}a^{l-1}_{2,1} + w_{1,2}a^{l-1}_{2,2} + w_{2,1}a^{l-1}_{3,1} + w_{2,2}\boxed{a^{l-1}_{3,2}} + w_b$$

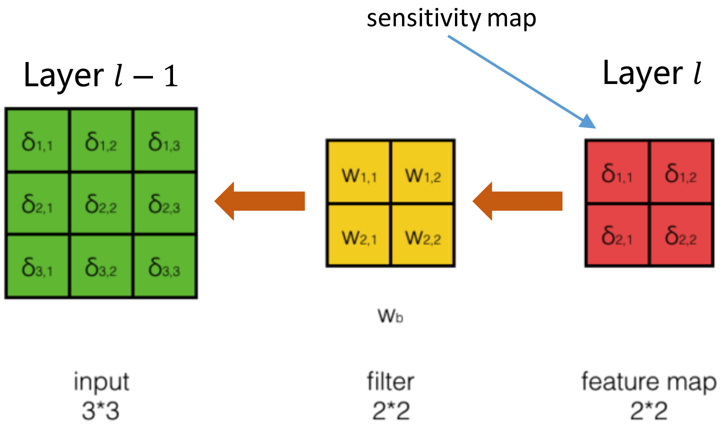
$$net^{l}_{2,2} = w_{1,1}a^{l-1}_{2,2} + w_{1,2}a^{l-1}_{2,3} + w_{2,1}\boxed{a^{l-1}_{3,2}} + w_{2,2}a^{l-1}_{3,3} + w_b$$



➢ 误差后向传播

$$\frac{\partial E_d}{\partial a^{l-1}_{3,2}} = \frac{\partial E_d}{\partial net^{l}_{2,1}}\frac{\partial net^{l}_{2,1}}{\partial a^{l-1}_{3,2}} + \frac{\partial E_d}{\partial net^{l}_{2,2}}\frac{\partial net^{l}_{2,2}}{\partial a^{l-1}_{3,2}}$$

$$= \delta^{l}_{2,1}w_{2,2} + \delta^{l}_{2,2}w_{2,1}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

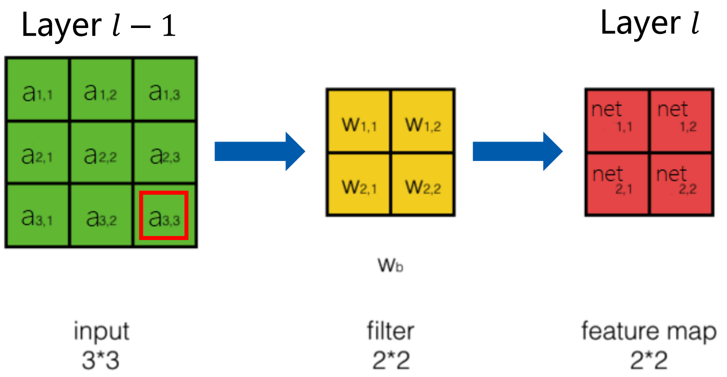
➤ 前向计算

$$net^l_{1,1} = w_{1,1}a^{l-1}_{1,1} + w_{1,2}a^{l-1}_{1,2} + w_{2,1}a^{l-1}_{2,1} + w_{2,2}a^{l-1}_{2,2} + w_b$$

$$net^l_{1,2} = w_{1,1}a^{l-1}_{1,2} + w_{1,2}a^{l-1}_{1,3} + w_{2,1}a^{l-1}_{2,2} + w_{2,2}a^{l-1}_{2,3} + w_b$$

$$net^l_{2,1} = w_{1,1}a^{l-1}_{2,1} + w_{1,2}a^{l-1}_{2,2} + w_{2,1}a^{l-1}_{3,1} + w_{2,2}a^{l-1}_{3,2} + w_b$$

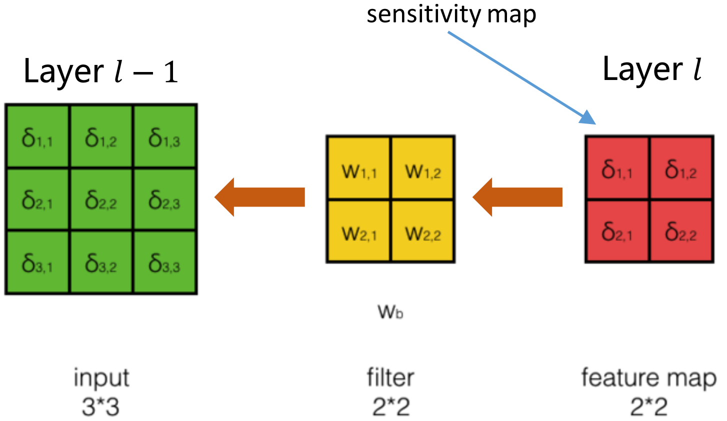
$$net^l_{2,2} = w_{1,1}a^{l-1}_{2,2} + w_{1,2}a^{l-1}_{2,3} + w_{2,1}a^{l-1}_{3,2} + w_{2,2}\boxed{a^{l-1}_{3,3}} + w_b$$



➤ 误差后向传播

$$\frac{\partial E_d}{\partial a^{l-1}_{3,3}} = \frac{\partial E_d}{\partial net^l_{2,2}} \frac{\partial net^l_{2,2}}{\partial a^{l-1}_{3,3}}$$

$$= \delta^l_{2,2} w_{2,2}$$



3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

$$\frac{\partial E_d}{\partial a_{1,1}^{l-1}} = \delta_{1,1}^{l} w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{1,2}^{l-1}} = \delta_{1,1}^{l} w_{1,2} + \delta_{1,2}^{l} w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{1,3}^{l-1}} = \delta_{1,2}^{l} w_{1,2}$$

$$\frac{\partial E_d}{\partial a_{2,1}^{l-1}} = \delta_{1,1}^{l} w_{2,1} + \delta_{2,1}^{l} w_{1,1}$$

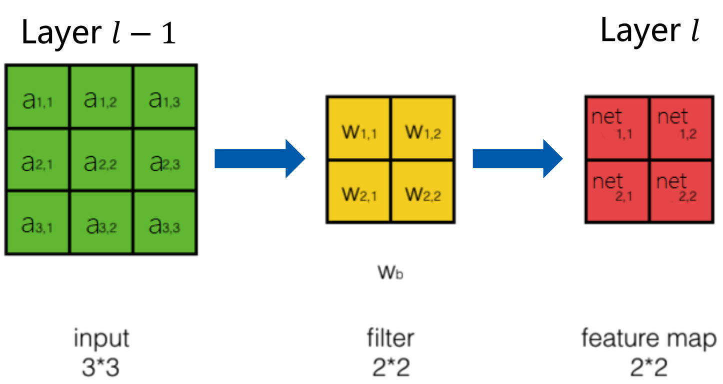
$$\frac{\partial E_d}{\partial a_{2,2}^{l-1}} = \delta_{1,1}^{l} w_{2,2} + \delta_{1,2}^{l} w_{2,1} + \delta_{2,1}^{l} w_{1,2} + \delta_{2,2}^{l} w_{1,1}$$

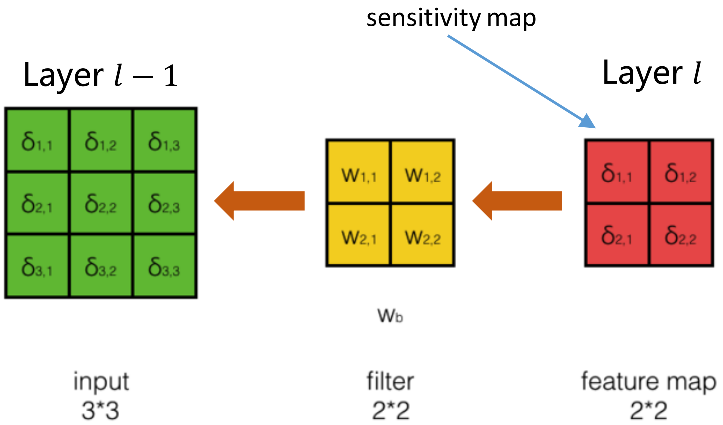
$$\frac{\partial E_d}{\partial a_{2,3}^{l-1}} = \delta_{1,2}^{l} w_{2,2} + \delta_{2,2}^{l} w_{1,2}$$

$$\frac{\partial E_d}{\partial a_{3,1}^{l-1}} = \delta_{2,1}^{l} w_{2,1}$$

$$\frac{\partial E_d}{\partial a_{3,2}^{l-1}} = \delta_{2,1}^{l} w_{2,2} + \delta_{2,2}^{l} w_{2,1}$$

$$\frac{\partial E_d}{\partial a_{3,3}^{l-1}} = \delta_{2,2}^{l} w_{2,2}$$





3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

反向传播中的filter旋转与卷积

$$\frac{\partial E_d}{\partial a_{1,1}^{l-1}} = \delta_{1,1}^l w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{1,2}^{l-1}} = \delta_{1,1}^l w_{1,2} + \delta_{1,2}^l w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{1,3}^{l-1}} = \delta_{1,2}^l w_{1,2}$$

$$\frac{\partial E_d}{\partial a_{2,1}^{l-1}} = \delta_{1,1}^l w_{2,1} + \delta_{2,1}^l w_{1,1}$$

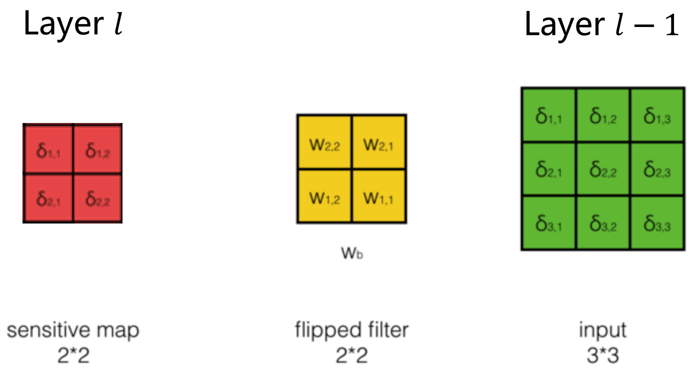
$$\frac{\partial E_d}{\partial a_{2,2}^{l-1}} = \delta_{1,1}^l w_{2,2} + \delta_{1,2}^l w_{2,1} + \delta_{2,1}^l w_{1,2} + \delta_{2,2}^l w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{2,3}^{l-1}} = \delta_{1,2}^l w_{2,2} + \delta_{2,2}^l w_{1,2}$$

$$\frac{\partial E_d}{\partial a_{3,1}^{l-1}} = \delta_{2,1}^l w_{2,1}$$

$$\frac{\partial E_d}{\partial a_{3,2}^{l-1}} = \delta_{2,1}^l w_{2,2} + \delta_{2,2}^l w_{2,1}$$

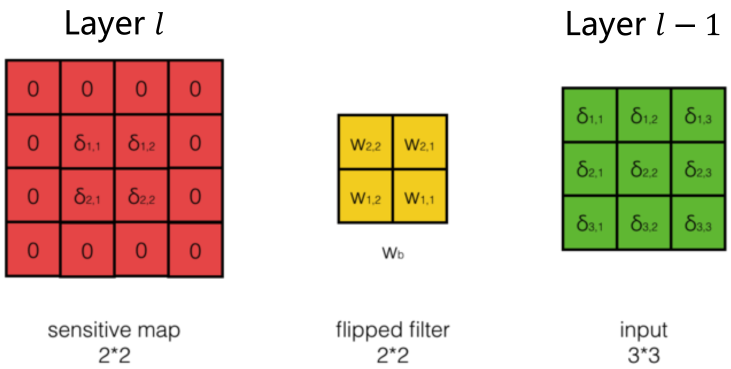
$$\frac{\partial E_d}{\partial a_{3,3}^{l-1}} = \delta_{2,2}^l w_{2,2}$$



$$\frac{\partial E_d}{\partial a^{l-1}} = \delta^l * W^l$$

3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1



反向传播中的filter旋转与卷积

$$\frac{\partial E_d}{\partial a_{1,1}^{l-1}} = \delta_{1,1}^l w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{1,2}^{l-1}} = \delta_{1,1}^l w_{1,2} + \delta_{1,2}^l w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{1,3}^{l-1}} = \delta_{1,2}^l w_{1,2}$$

$$\frac{\partial E_d}{\partial a_{2,1}^{l-1}} = \delta_{1,1}^l w_{2,1} + \delta_{2,1}^l w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{2,2}^{l-1}} = \delta_{1,1}^l w_{2,2} + \delta_{1,2}^l w_{2,1} + \delta_{2,1}^l w_{1,2} + \delta_{2,2}^l w_{1,1}$$

$$\frac{\partial E_d}{\partial a_{2,3}^{l-1}} = \delta_{1,2}^l w_{2,2} + \delta_{2,2}^l w_{1,2}$$

$$\frac{\partial E_d}{\partial a_{3,1}^{l-1}} = \delta_{2,1}^l w_{2,1}$$

$$\frac{\partial E_d}{\partial a_{3,3}^{l-1}} = \delta_{2,2}^l w_{2,2}$$

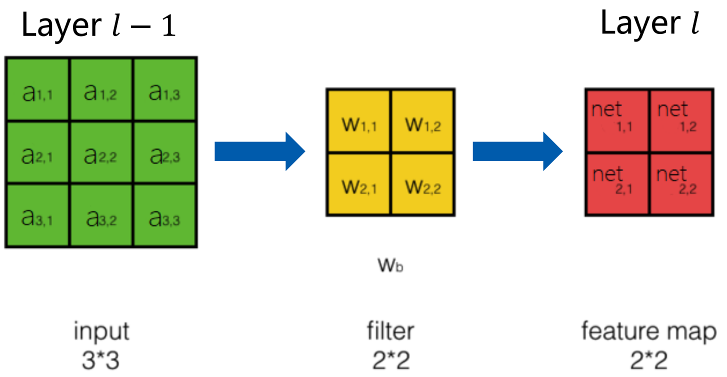
$$\begin{aligned}\delta_{i,j}^{l-1} &= \frac{\partial E_d}{\partial net_{i,j}^{l-1}} \\ &= \frac{\partial E_d}{\partial a_{i,j}^{l-1}} \frac{\partial a_{i,j}^{l-1}}{\partial net_{i,j}^{l-1}} \\ &= \sum_m \sum_n w_{m,n}^l \delta_{i+m,j+n}^l f'(net_{i,j}^{l-1})\end{aligned}$$

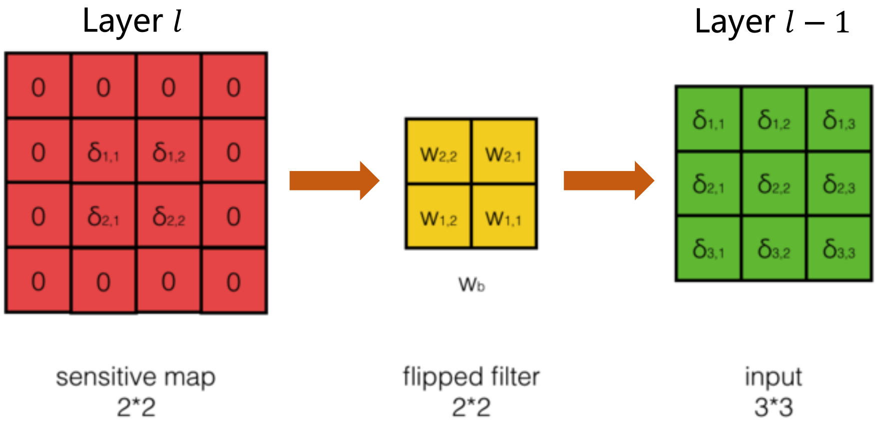
$$\delta^{l-1} = \delta^l * W^l \circ f'(net^{l-1})$$

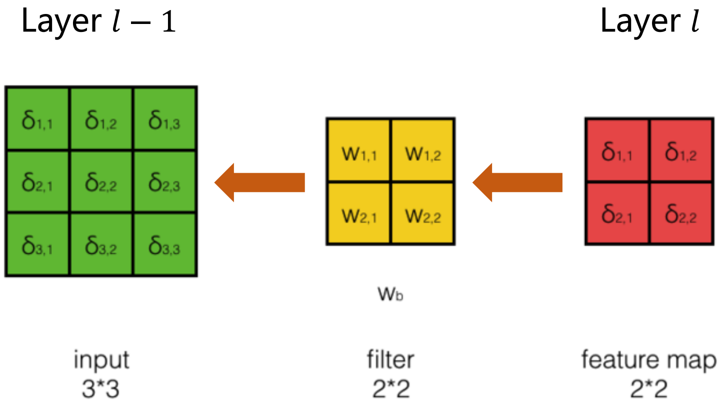
3. CNN推导-卷积层

最简单的情况： Step=1,Depth=1,Filter=1

反向传播中的filter旋转与卷积







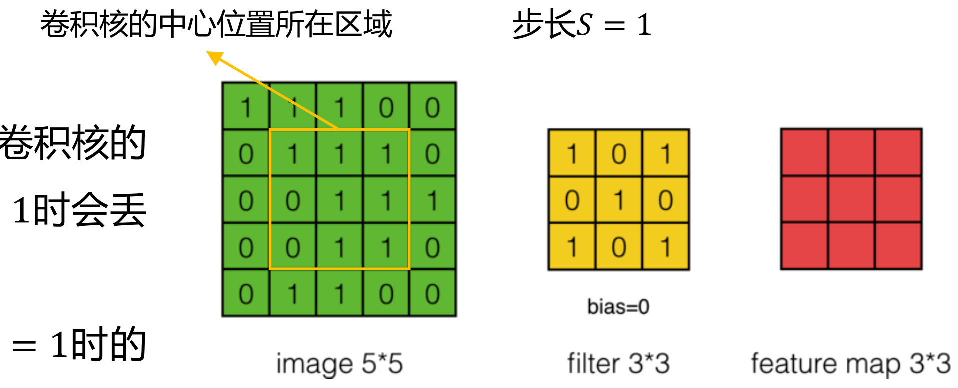
卷积层误差反传—步长为S的情况

3. CNN推导-卷积层

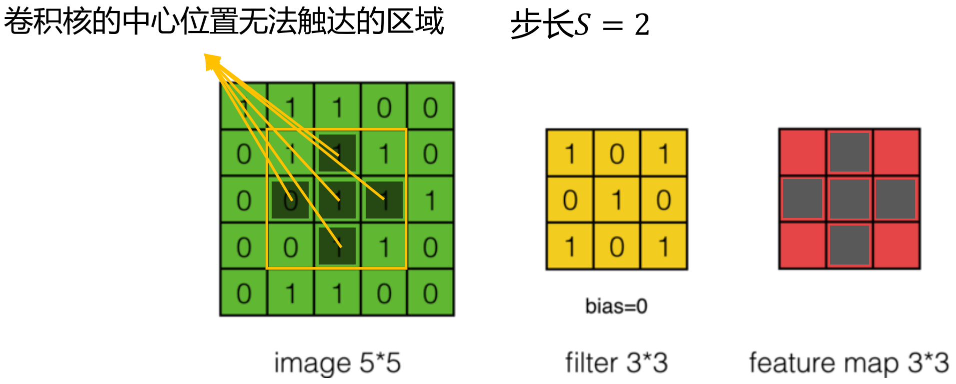
步长为S的情况($S > 1$)

卷积为S时，随着卷积的移动(可以只看卷积核的移动)，得到的feature map相比与$S = 1$时会丢失掉部分的结果(右边暗色标明的位置)。

通过将结果补零之后，再利用前面的$S = 1$时的计算规则就可以计算反向传播的误差了。



$$\delta^{l-1} = \delta^{l} * W^{l} \circ f'(net^{l-1})$$



卷积层误差反传—深度为D时候的误差

3. CNN推导-卷积层

深度为D的情况($D > 1$)

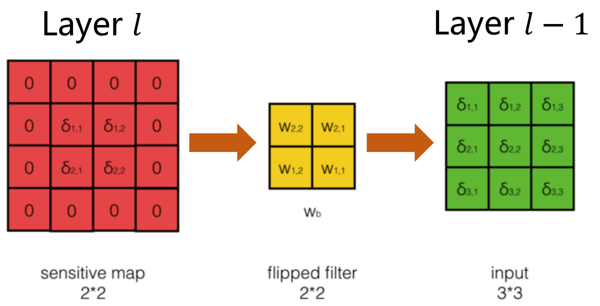
当输入深度为D时，filter的深度也必须为D，$l-1$层的第d通道只与filter的第d通道的权重进行计算。因此，反向计算**误差项**时，还是使用前面最简单的情况的计算方式。

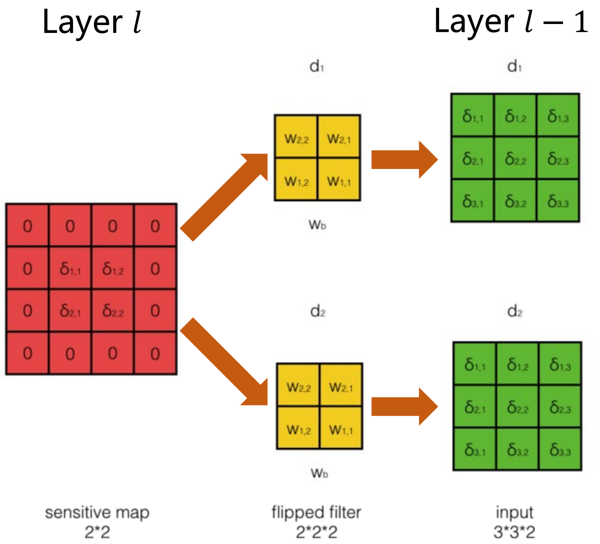
用filter的第d通道权重对第d层sensitivity map进行卷积，得到第$l-1$层通道的sensitivity map。

$$a_{i,j} = f(\boxed{\sum_{d=0}^{D-1}} \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} w_{d,m,n} x_{d,i+m,j+n} + w_b)$$

本层是feature map的个数，也就是卷积核的深度

$$\delta^{l-1} = \delta^l * W^l \circ f'(net^{l-1})$$





卷积层误差反传—filter的数量为N的时候

3. CNN推导-卷积层

filter个数为N的情况

Filter的数量为N的时候，输出层的深度也是N。第n个filter卷积只产生第n个feature map。由于第$l-1$层**每个加权输入** $net_{d,i,j}^{l-1}$ 都同时影响了第l层所有feature map的输出值。因此，反向计算**误差项**时，需要分别计算出每个卷积造成的误差，然后将所有的误差相加。

也就是，我们先使用第n个filter对第l层相应的第n个误差矩阵进行卷积，得到一组N个$l-1$层的误差矩阵。依次用每个filter做这种卷积，就得到N组偏误差矩阵。最后在各组之间将N个偏误差矩阵**按元素相加**，得到最终的N个$l-1$层的误差项。

$$\delta^{l-1} = \boxed{\sum_{n=0}^{N}} \delta_n^l * W_n^l \circ f'(net^{l-1})$$

对每一个filter

卷积层误差反传—N个filter，Depth为D

3. CNN推导-卷积层

Filter=N, Depth=D

每个depth之间都是相互独立的，所以可以将不同的depth看作是单独的Filter是N的上一种情况。

$$\delta^{l-1}=\boxed{\sum_{n=0}^{N}}\delta_n^l * W_n^l \circ f'(net^{l-1})$$

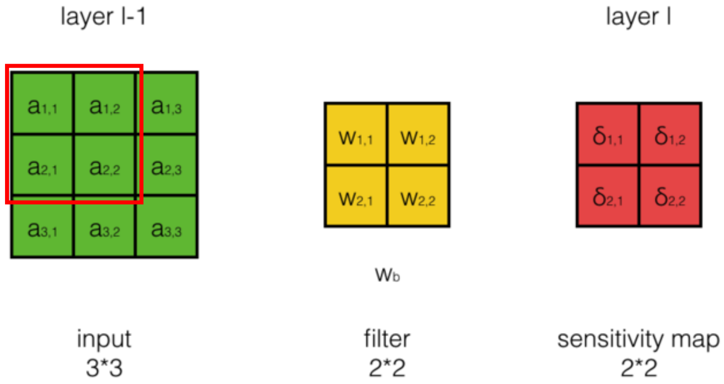
对每一个filter

$$\delta_d^{l-1}=\sum_{n=0}^{N}\delta_n^l * W_{nd}^l \circ f'(net_d^{l-1})$$

卷积层权重更新

3. CNN推导-卷积层权重更新

在得到第l层sensitivity map的情况下，计算filter的权重的梯度，由于卷积层是**权重共享**的，因此梯度的计算稍有不同。



$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}a_{1,2}^{l-1} + w_{2,1}a_{2,1}^{l-1} + w_{2,2}a_{2,2}^{l-1} + w_b$$

$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + w_{1,2}a_{1,3}^{l-1} + w_{2,1}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$

$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + w_{1,2}a_{2,2}^{l-1} + w_{2,1}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$

$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + w_{1,2}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$

$$\begin{aligned}\frac{\partial E_d}{\partial w_{1,1}} &= \frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial w_{1,1}} + \frac{\partial E_d}{\partial net_{1,2}^{l}}\frac{\partial net_{1,2}^{l}}{\partial w_{1,1}} + \frac{\partial E_d}{\partial net_{2,1}^{l}}\frac{\partial net_{2,1}^{l}}{\partial w_{1,1}} + \frac{\partial E_d}{\partial net_{2,2}^{l}}\frac{\partial net_{2,2}^{l}}{\partial w_{1,1}} \\ &= \delta_{1,1}^{l}a_{1,1}^{l-1} + \delta_{1,2}^{l}a_{1,2}^{l-1} + \delta_{2,1}^{l}a_{2,1}^{l-1} + \delta_{2,2}^{l}a_{2,2}^{l-1}\end{aligned}$$

3. CNN推导-卷积层权重更新

在得到第l层sensitivity map的情况下，计算filter的权重的梯度，由于卷积层是**权重共享**的，因此梯度的计算稍有不同。

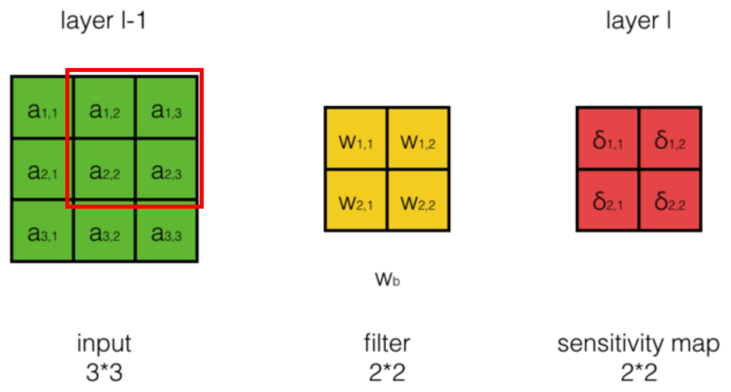
$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + \boxed{w_{1,2}}a_{1,2}^{l-1} + w_{2,1}a_{2,1}^{l-1} + w_{2,2}a_{2,2}^{l-1} + w_b$$

$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + \boxed{w_{1,2}}a_{1,3}^{l-1} + w_{2,1}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$

$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + \boxed{w_{1,2}}a_{2,2}^{l-1} + w_{2,1}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$

$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + \boxed{w_{1,2}}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$

$$\frac{\partial E_d}{\partial w_{1,2}} = \delta_{1,1}^{l}a_{1,2}^{l-1} + \delta_{1,2}^{l}a_{1,3}^{l-1} + \delta_{2,1}^{l}a_{2,2}^{l-1} + \delta_{2,2}^{l}a_{2,3}^{l-1}$$

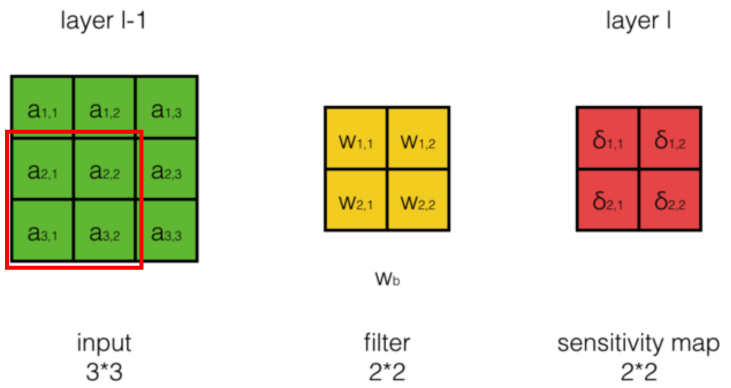


3. CNN推导-卷积层权重更新

在得到第l层sensitivity map的情况下，计算filter的权重的梯度，由于卷积层是**权重共享**的，因此梯度的计算稍有不同。

$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}a_{1,2}^{l-1} + \boxed{w_{2,1}}a_{2,1}^{l-1} + w_{2,2}a_{2,2}^{l-1} + w_b$$
$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + w_{1,2}a_{1,3}^{l-1} + \boxed{w_{2,1}}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$
$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + w_{1,2}a_{2,2}^{l-1} + \boxed{w_{2,1}}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$
$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + w_{1,2}a_{2,3}^{l-1} + \boxed{w_{2,1}}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$

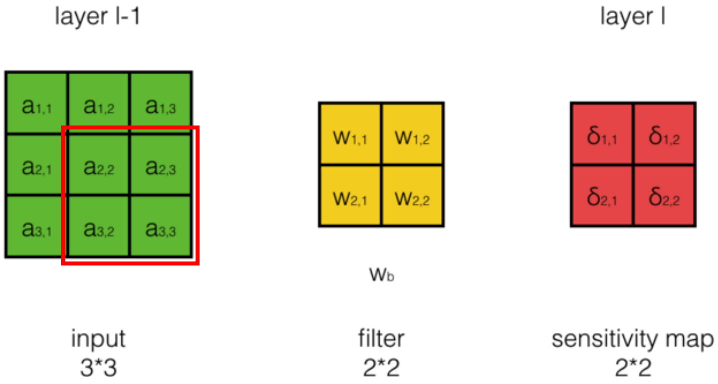
$$\frac{\partial E_d}{\partial w_{2,1}} = \delta_{1,1}^{l}a_{2,1}^{l-1} + \delta_{1,2}^{l}a_{2,2}^{l-1} + \delta_{2,1}^{l}a_{3,1}^{l-1} + \delta_{2,2}^{l}a_{3,2}^{l-1}$$



3. CNN推导-卷积层权重更新

在得到第l层sensitivity map的情况下，计算filter的权重的梯度，由于卷积层是**权重共享**的，因此梯度的计算稍有不同。

$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}a_{1,2}^{l-1} + w_{2,1}a_{2,1}^{l-1} + w_{2,2}a_{2,2}^{l-1} + w_b$$
$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + w_{1,2}a_{1,3}^{l-1} + w_{2,1}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$
$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + w_{1,2}a_{2,2}^{l-1} + w_{2,1}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$
$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + w_{1,2}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$

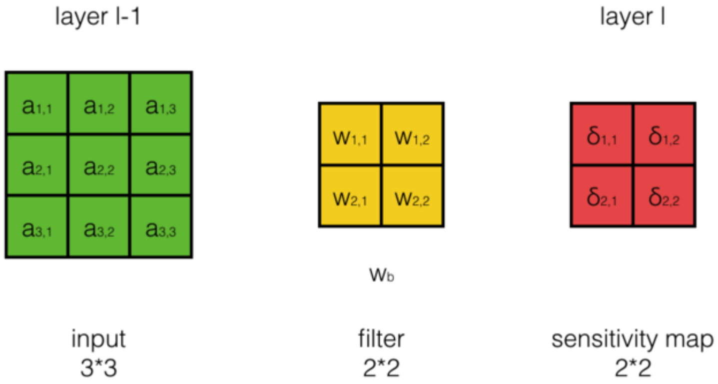


$$\frac{\partial E_d}{\partial w_{2,2}} = \delta_{1,1}^{l}a_{2,2}^{l-1} + \delta_{1,2}^{l}a_{2,3}^{l-1} + \delta_{2,1}^{l}a_{3,2}^{l-1} + \delta_{2,2}^{l}a_{3,3}^{l-1}$$

3. CNN推导-卷积层权重更新

对于偏置w_b的权重更新：

$$net_{1,1}^{l} = w_{1,1}a_{1,1}^{l-1} + w_{1,2}a_{1,2}^{l-1} + w_{2,1}a_{2,1}^{l-1} + w_{2,2}a_{2,2}^{l-1} + w_b$$
$$net_{1,2}^{l} = w_{1,1}a_{1,2}^{l-1} + w_{1,2}a_{1,3}^{l-1} + w_{2,1}a_{2,2}^{l-1} + w_{2,2}a_{2,3}^{l-1} + w_b$$
$$net_{2,1}^{l} = w_{1,1}a_{2,1}^{l-1} + w_{1,2}a_{2,2}^{l-1} + w_{2,1}a_{3,1}^{l-1} + w_{2,2}a_{3,2}^{l-1} + w_b$$
$$net_{2,2}^{l} = w_{1,1}a_{2,2}^{l-1} + w_{1,2}a_{2,3}^{l-1} + w_{2,1}a_{3,2}^{l-1} + w_{2,2}a_{3,3}^{l-1} + w_b$$



$$\begin{aligned}
\frac{\partial E_d}{\partial w_b} &= \frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial w_b} + \frac{\partial E_d}{\partial net_{1,2}^{l}}\frac{\partial net_{1,2}^{l}}{\partial w_b} + \frac{\partial E_d}{\partial net_{2,1}^{l}}\frac{\partial net_{2,1}^{l}}{\partial w_b} + \frac{\partial E_d}{\partial net_{2,2}^{l}}\frac{\partial net_{2,2}^{l}}{\partial w_b} \\
&= \delta_{1,1}^{l} + \delta_{1,2}^{l} + \delta_{2,1}^{l} + \delta_{2,2}^{l} \\
&= \sum_i \sum_j \delta_{i,j}^{l}
\end{aligned}$$

3. CNN推导-卷积层权重更新

对于步长为S的卷积层，处理方法与传递 误差项 是一样的，首先将sensitivity map『还原』成步长为1时的sensitivity map，再用上面的方法进行计算。

获得了所有的梯度之后，就是根据梯度下降算法来更新每个权重。至此，我们已经解决了卷积层的训练问题，接下来是Pooling层的训练。

3. CNN推导-Pooling层权重更新

无论max pooling还是mean pooling，都没有需要学习的参数。因此，在卷积神经网络的训练中，Pooling层需要做的仅仅是将误差项传递到上一层，而没有梯度的计算。

3. CNN推导-Pooling层权重更新

3.1 Max Pooling的计算

$$net_{1,1}^{l} = max(net_{1,1}^{l-1}, net_{1,2}^{l-1}, net_{2,1}^{l-1}, net_{2,2}^{l-1})$$

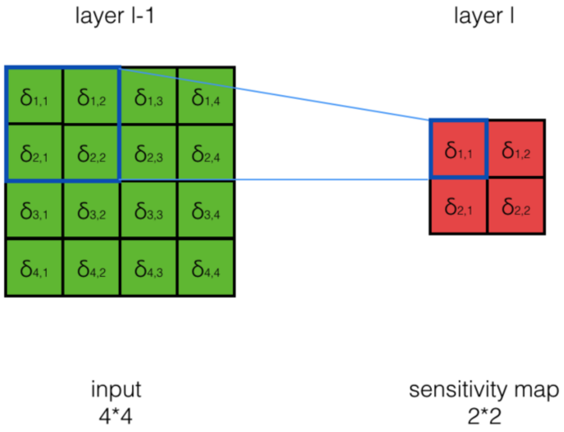
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}} = 1$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} = 0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} = 0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} = 0$$

$$\begin{aligned}\delta_{1,1}^{l-1} &= \frac{\partial E_d}{\partial net_{1,1}^{l-1}} \\ &= \frac{\partial E_d}{\partial net_{1,1}^{l}} \frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}} \\ &= \delta_{1,1}^{l}\end{aligned}$$



3. CNN推导-Pooling层权重更新

3.1 Max Pooling的计算

$$net_{1,1}^{l}=max(net_{1,1}^{l-1},net_{1,2}^{l-1},net_{2,1}^{l-1},net_{2,2}^{l-1})$$

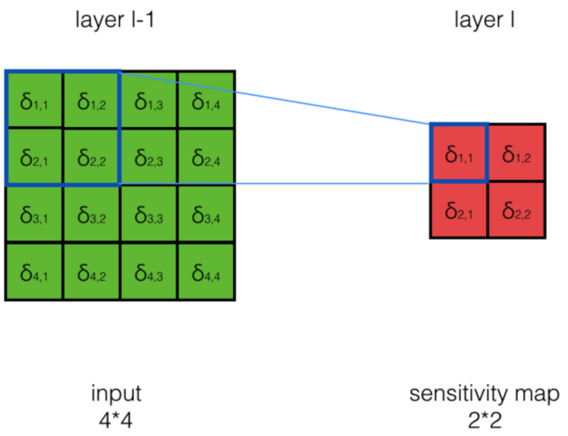
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}}=1$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}}=0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}}=0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}}=0$$

$$\begin{aligned}\delta_{1,2}^{l-1} &= \frac{\partial E_d}{\partial net_{1,2}^{l-1}} \\ &= \frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} \\ &= 0\end{aligned}$$



3. CNN推导-Pooling层权重更新

3.1 Max Pooling的计算

$$net_{1,1}^{l} = max(net_{1,1}^{l-1}, net_{1,2}^{l-1}, net_{2,1}^{l-1}, net_{2,2}^{l-1})$$

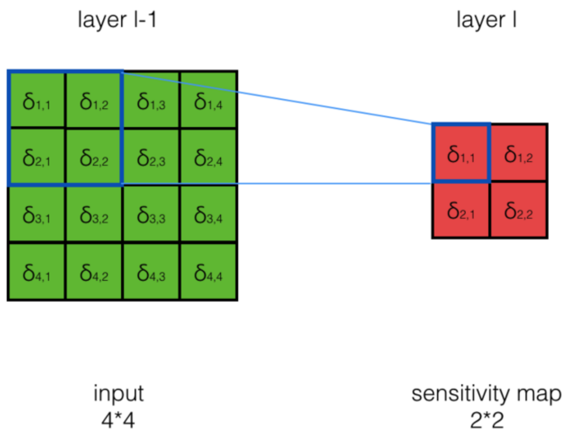
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}} = 1$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} = 0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} = 0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} = 0$$

$$\begin{aligned}\delta_{2,1}^{l-1} &= \frac{\partial E_d}{\partial net_{2,1}^{l-1}} \\ &= \frac{\partial E_d}{\partial net_{1,1}^{l}} \frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} \\ &= 0\end{aligned}$$



3. CNN推导-Pooling层权重更新

3.1 Max Pooling的计算

$$net_{1,1}^{l} = max(net_{1,1}^{l-1}, net_{1,2}^{l-1}, net_{2,1}^{l-1}, net_{2,2}^{l-1})$$

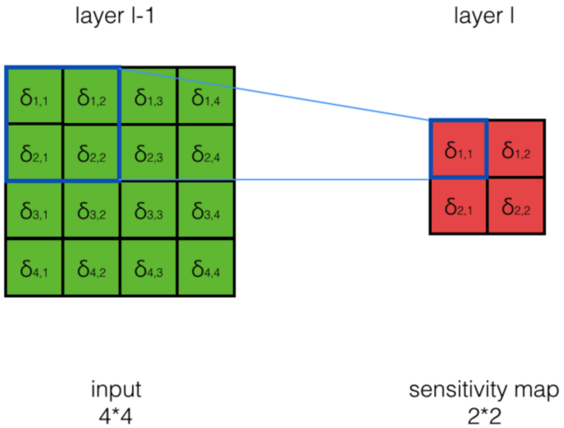
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}} = 1$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} = 0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} = 0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} = 0$$

$$\begin{aligned}\delta_{2,2}^{l-1} &= \frac{\partial E_d}{\partial net_{2,2}^{l-1}} \\ &= \frac{\partial E_d}{\partial net_{1,1}^{l}} \frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} \\ &= 0\end{aligned}$$



对于max pooling，下一层的**误差项**的值会原封不动地传递到上一层对应区块中的最大值所对应的神经元，而其他神经元的**误差项**的值都是0。

如上图所示，layer l层的误差$\delta_{1,1}$会原封不动地传递到layer $l-1$层对应区块$\begin{pmatrix}\delta_{1,1} & \delta_{1,2} \\ \delta_{2,1} & \delta_{2,2}\end{pmatrix}$中的最大值所对应的神经元。

3. CNN推导-Pooling层权重更新

3.1 Max Pooling的计算

$$net_{1,1}^{l}=max(net_{1,1}^{l-1},net_{1,2}^{l-1},net_{2,1}^{l-1},net_{2,2}^{l-1})$$

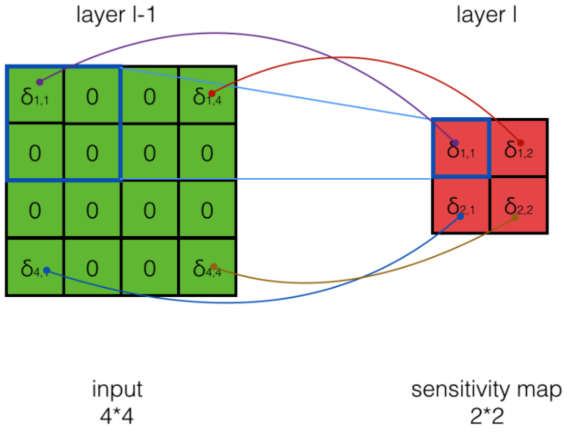
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}}=1$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}}=0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}}=0$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}}=0$$

$$\begin{aligned}\delta_{2,2}^{l-1}&=\frac{\partial E_d}{\partial net_{2,2}^{l-1}}\\&=\frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}}\\&=0\end{aligned}$$



对于max pooling，下一层的**误差项**的值会原封不动的传递到上一层对应区块中的最大值所对应的神经元，而其他神经元的**误差项**的值都是0。

3. CNN推导-Pooling层权重更新

3.2 Mean Pooling的计算

$$net_{1,1}^{l}=\frac{1}{4}(net_{1,1}^{l-1}+net_{1,2}^{l-1}+net_{2,1}^{l-1}+net_{2,2}^{l-1})$$

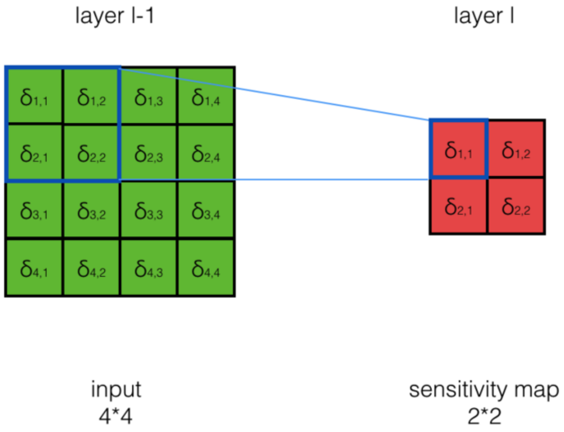
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}}=\frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}}=\frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}}=\frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}}=\frac{1}{4}$$

$$\begin{aligned}\delta_{1,1}^{l-1}&=\frac{\partial E_d}{\partial net_{1,1}^{l-1}}\\&=\frac{\partial E_d}{\partial net_{1,1}^{l}}\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}}\\&=\frac{1}{4}\delta_{1,1}^{l}\end{aligned}$$



3. CNN推导-Pooling层权重更新

3.2 Mean Pooling的计算

$$net_{1,1}^{l} = \frac{1}{4}(net_{1,1}^{l-1} + net_{1,2}^{l-1} + net_{2,1}^{l-1} + net_{2,2}^{l-1})$$

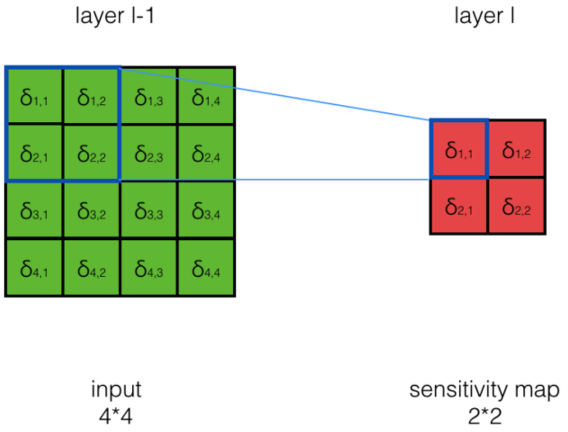
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} = \frac{1}{4}$$

$$\begin{aligned}\delta_{1,2}^{l-1} &= \frac{\partial E_d}{\partial net_{1,2}^{l-1}} \\ &= \frac{\partial E_d}{\partial net_{1,1}^{l}} \frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} \\ &= \frac{1}{4}\delta_{1,1}^{l}\end{aligned}$$



3. CNN推导-Pooling层权重更新

3.2 Mean Pooling的计算

$$net_{1,1}^{l} = \frac{1}{4}(net_{1,1}^{l-1} + net_{1,2}^{l-1} + net_{2,1}^{l-1} + net_{2,2}^{l-1})$$

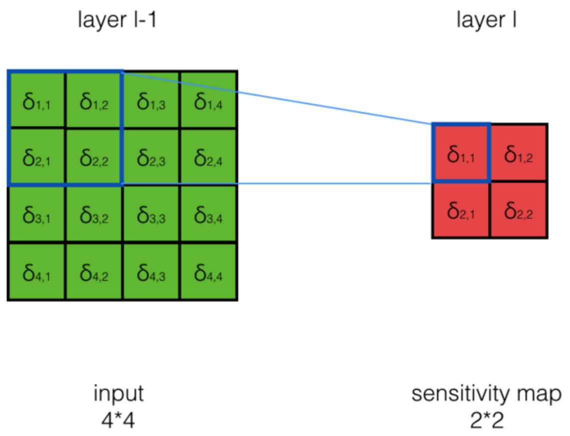
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} = \frac{1}{4}$$

$$\begin{aligned}\delta_{2,1}^{l-1} &= \frac{\partial E_d}{\partial net_{2,1}^{l-1}} \\ &= \frac{\partial E_d}{\partial net_{1,1}^{l}} \frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} \\ &= \frac{1}{4}\delta_{1,1}^{l}\end{aligned}$$



3. CNN推导-Pooling层权重更新

3.2 Mean Pooling的计算

$$net_{1,1}^{l} = \frac{1}{4}(net_{1,1}^{l-1} + net_{1,2}^{l-1} + net_{2,1}^{l-1} + net_{2,2}^{l-1})$$

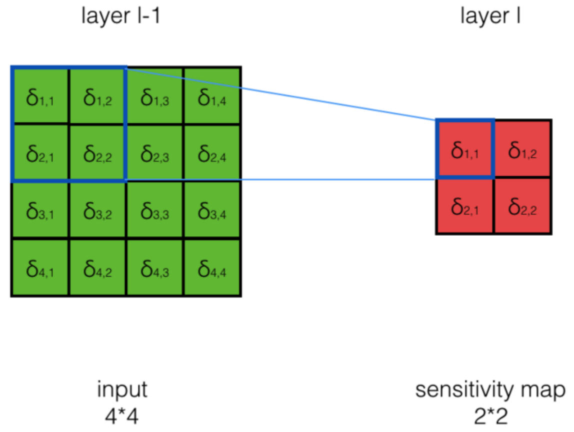
$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,1}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{1,2}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,1}^{l-1}} = \frac{1}{4}$$

$$\frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} = \frac{1}{4}$$

$$\begin{aligned} \delta_{2,2}^{l-1} &= \frac{\partial E_d}{\partial net_{2,2}^{l-1}} \\ &= \frac{\partial E_d}{\partial net_{1,1}^{l}} \frac{\partial net_{1,1}^{l}}{\partial net_{2,2}^{l-1}} \\ &= \frac{1}{4}\delta_{1,1}^{l} \end{aligned}$$



对于mean pooling，下一层的误差项的值会平均分配到上一层对应区块中的所有神经元。

如上图所示，layer l层的误差$\delta_{1,1}$会平均分配到layer $l-1$层对应区块$\begin{pmatrix} \delta_{1,1} & \delta_{1,2} \\ \delta_{2,1} & \delta_{2,2} \end{pmatrix}$中的所有神经元。

3. CNN推导-Pooling层权重更新

3.2 Mean Pooling的计算

$$net^l_{1,1} = \frac{1}{4}(net^{l-1}_{1,1} + net^{l-1}_{1,2} + net^{l-1}_{2,1} + net^{l-1}_{2,2})$$

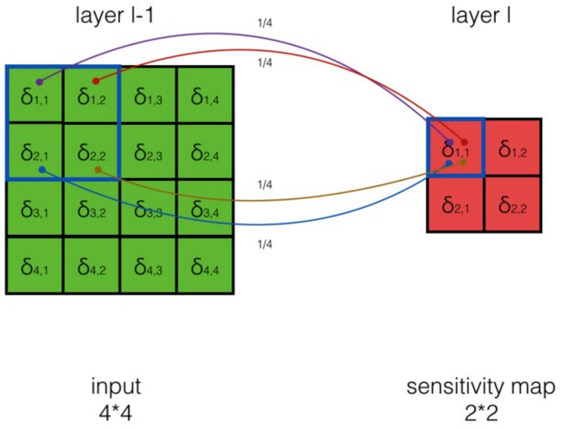
$$\frac{\partial net^l_{1,1}}{\partial net^{l-1}_{1,1}} = \frac{1}{4}$$

$$\frac{\partial net^l_{1,1}}{\partial net^{l-1}_{1,2}} = \frac{1}{4}$$

$$\frac{\partial net^l_{1,1}}{\partial net^{l-1}_{2,1}} = \frac{1}{4}$$

$$\frac{\partial net^l_{1,1}}{\partial net^{l-1}_{2,2}} = \frac{1}{4}$$

$$\begin{aligned}\delta^{l-1}_{2,2} &= \frac{\partial E_d}{\partial net^{l-1}_{2,2}} \\ &= \frac{\partial E_d}{\partial net^l_{1,1}} \frac{\partial net^l_{1,1}}{\partial net^{l-1}_{2,2}} \\ &= \frac{1}{4}\delta^l_{1,1}\end{aligned}$$



对于mean pooling，下一层的误差项的值会平均分配到上一层对应区块中的所有神经元。

如上图所示，layer l层的误差$\delta_{1,1}$会平均分配到layer $l-1$层对应区块$\begin{pmatrix}\delta_{1,1} & \delta_{1,2} \\ \delta_{2,1} & \delta_{2,2}\end{pmatrix}$中的所有神经元。

人类视觉系统

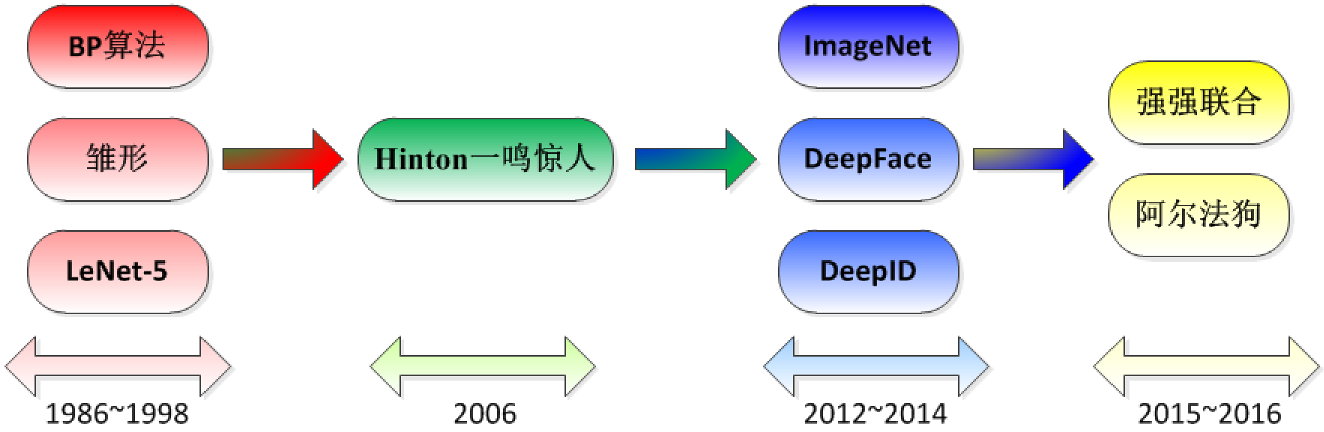
CNN数学原理

CNN推导

CNN发展历史

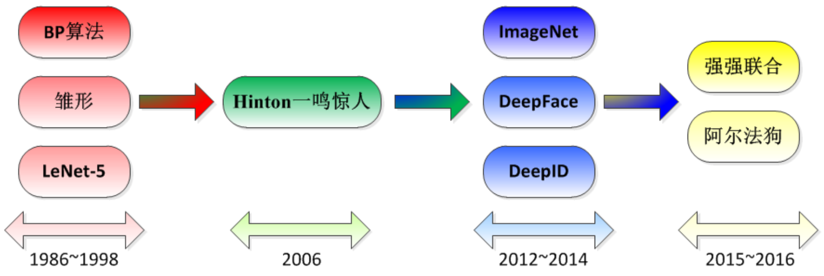
手写数字识别

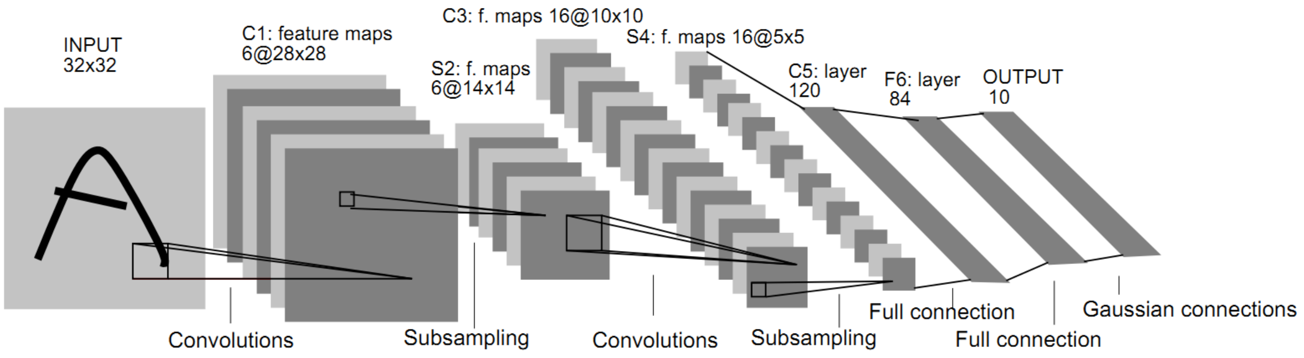
4. CNN发展历史


BP算法
维形
LeNet-5
Hinton一鸣惊人
ImageNet
DeepFace
DeepID
强强联合
阿尔法狗
1986~1998
2006
2012~2014
2015~2016

4. CNN发展历史

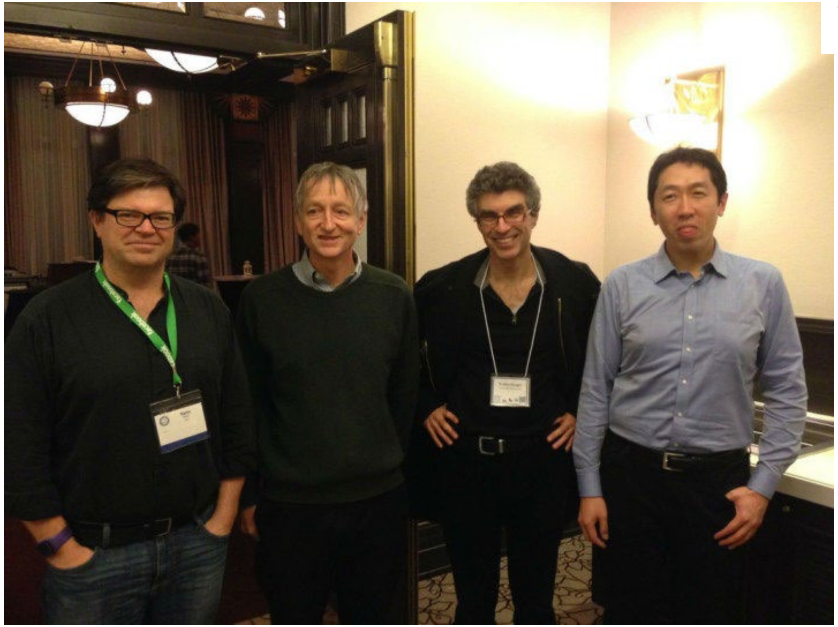
1986-1998

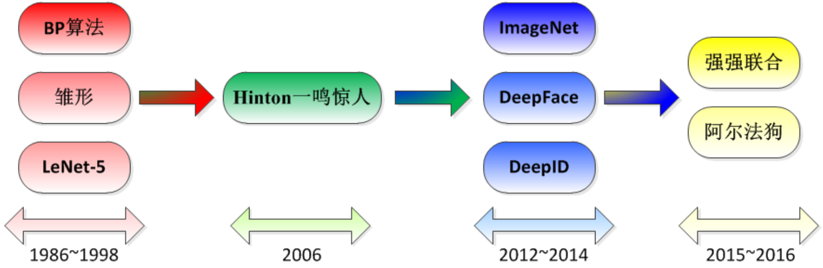




4. CNN发展历史

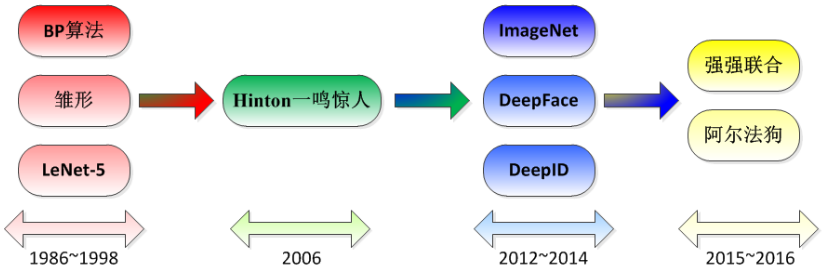
2006

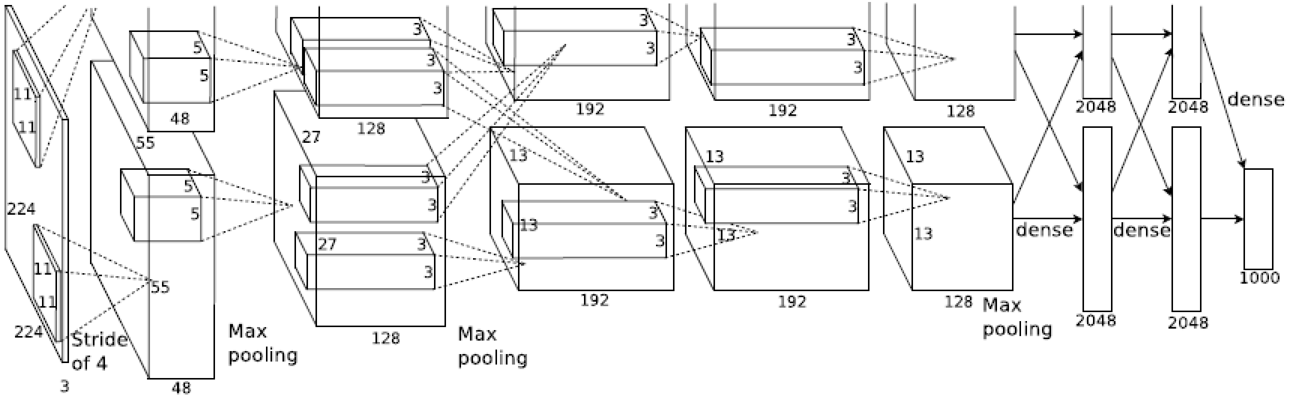




4. CNN发展历史

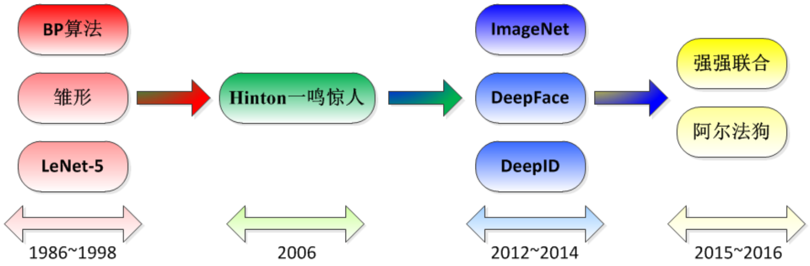
2012-2014

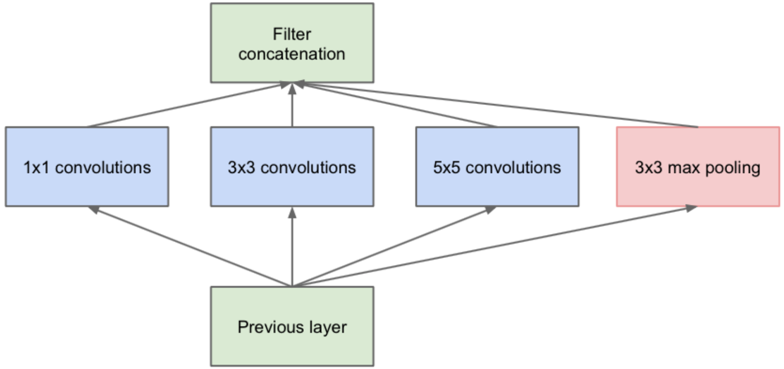


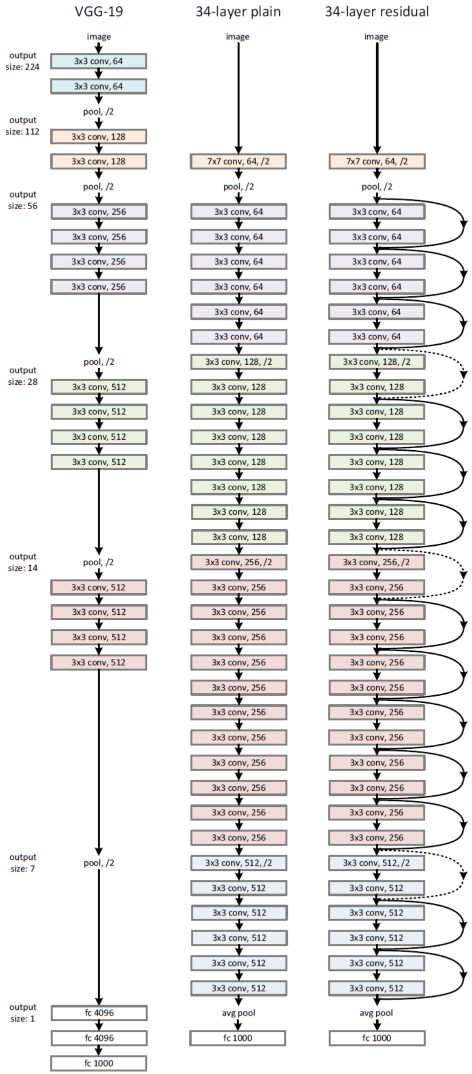


4. CNN发展历史

2015-至今

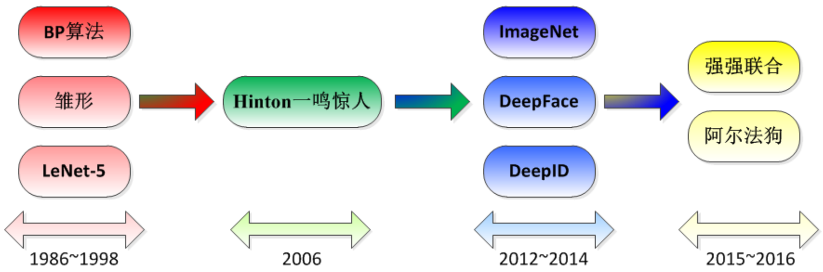


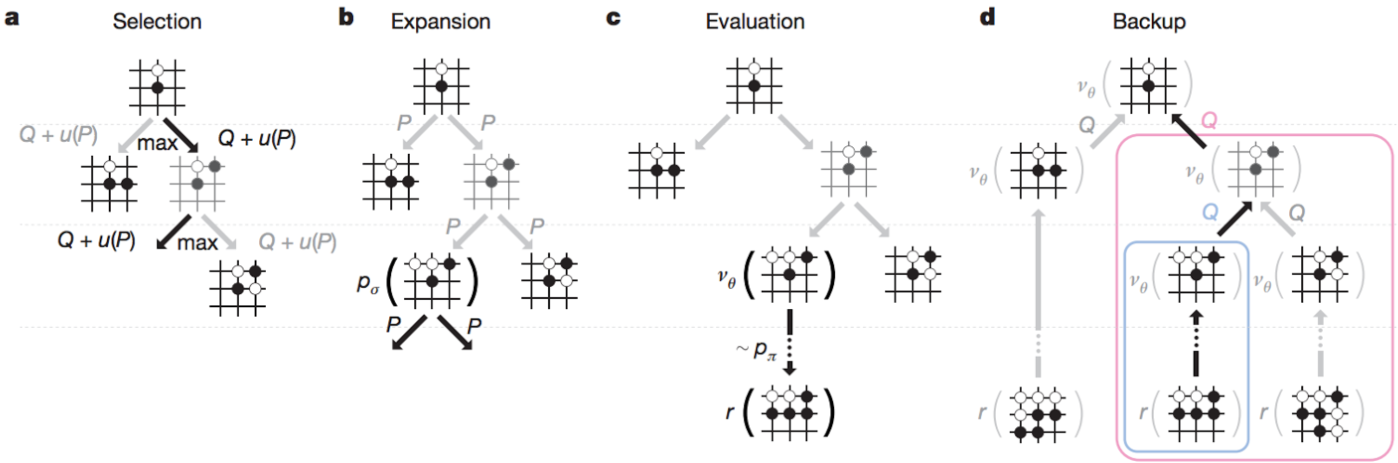




4. CNN发展历史

2015-至今





- 人类视觉系统
- CNN数学原理
- CNN推导
- CNN发展历史
- **手写数字识别**

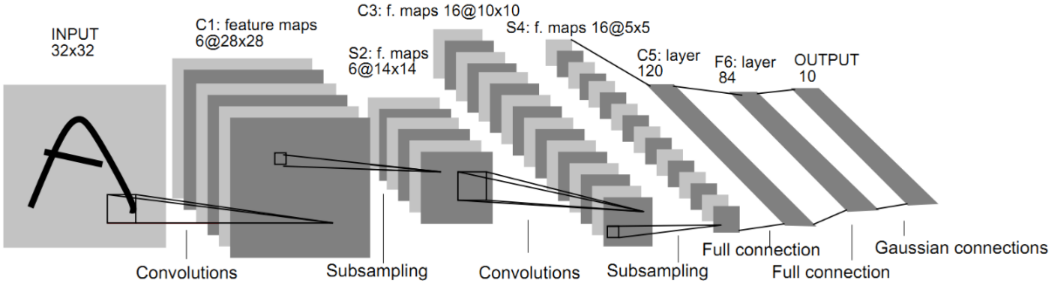
课后作业

必做

1. 对比卷积神经网络与全连接神经网络，在图像分类任务中，原始图像大小的变化将会怎样影响模型可训练参数个数?

2. 卷积神经网络是通过什么方式来完成可训练参数的减少?

3. 如图是LeNet-5的示意图，试着写出每一层的参数个数。



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X