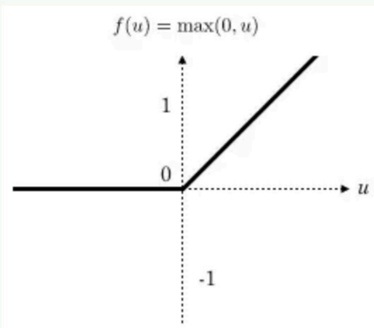
全新激活函数 Relu

近几年不再选择 sigmoid 或 tanh，而是 relu

$f(x)=\max(0,x)$

Relu函数图像如下图所示：



优点：1. 速度快

2. 减轻梯度消失问题

$\nabla = \sigma' \delta x$

sigmoid 函数的导数

即每经过一个 sigmoid 神经元就要乘一个 σ'

这样因为 σ' 最大值为 $1/4$，这样越来越小.

但是 Relu 的导数值为1. 不会有影响

不会导致梯度变小

- **稀疏性** 通过对大脑的研究发现，大脑在工作的时候只有大约5%的神经元是激活的，而采用sigmoid激活函数的人工神经网络，其激活率大约是50%。有论文声称人工神经网络在15%–30%的激活率时是比较理想的。因为relu函数在输入小于0时是完全不激活的，因此可以获得一个更低的激活率。

全连接网络 VS 卷积网络

全连接神经网络之所以不太适合图像识别任务，主要有以下几个方面的问题：

- **参数数量太多** 考虑一个输入1000*1000像素的图片(一百万像素，现在已经不能算大图了)，输入层有1000*1000=100万节点。假设第一个隐藏层有100个节点(这个数量并不多)，那么仅这一层就有(1000*1000+1)*100=1亿参数，这实在是太多了！我们看到图像只扩大一点，参数数量就会多很多，因此它的扩展性很差。
- **没有利用像素之间的位置信息** 对于图像识别任务来说，每个像素和其周围像素的联系是比较紧密的，和离得很远的像素的联系可能就很小了。如果一个神经元和上一层所有神经元相连，那么就相当于对于一个像素来说，把图像的所有像素都等同看待，这不符合前面的假设。当我们完成每个连接权重的学习之后，最终可能会发现，有大量的权重，它们的值都是很小的(也就是这些连接其实无关紧要)。努力学习大量并不重要的权重，这样的学习必将是非常低效的。
- **网络层数限制** 我们知道网络层数越多其表达能力越强，但是通过梯度下降方法训练深度全连接神经网络很困难，因为全连接神经网络的梯度很难传递超过3层。因此，我们不可能得到一个很深的全连接神经网络，也就限制了它的能力。

那么，卷积神经网络又是怎样解决这个问题的呢？主要有三个思路：

- **局部连接** 这个是最容易想到的，每个神经元不再和上一层的所有神经元相连，而只和一小部分神经元相连。这样就减少了很多参数。
- **权值共享** 一组连接可以共享同一个权重，而不是每个连接有一个不同的权重，这样又减少了很多参数。
- **下采样** 可以使用Pooling来减少每层的样本数，进一步减少参数数量，同时还可以提升模型的鲁棒性。

对于图像识别任务来说，卷积神经网络通过尽可能保留重要的参数，去掉大量不重要的参数，来达到更好的学习效果。

卷积神经网络

卷积层，pooling层，全连接层

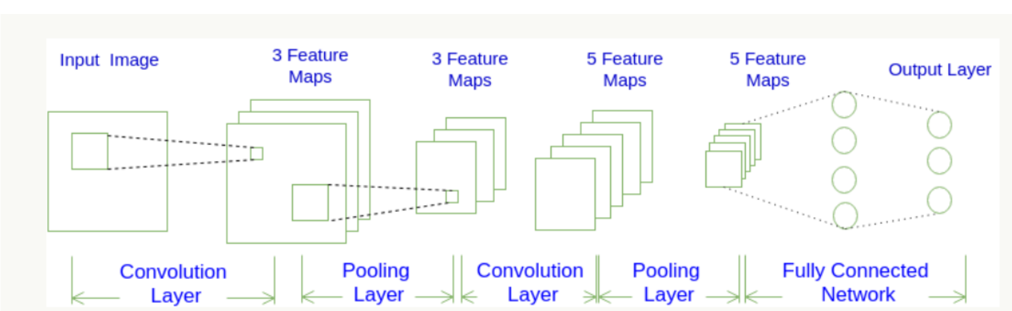
Input → (conv *N → pool?)×M → FC×k

N个卷积层叠加

可能叠加一个pooling层

重复M次

最后叠K个全连接层



Input → conv → pool → conv → pool → FC → FC

⇓

Input → [conv*1 → pool] *2 → FC*2

N=1 M=2 K=2

三维层结构

全连接网络为一维排列

卷积为三维 → 长宽高

图片中看出

1个卷积层 → 3个Filter (得3个Feature Map)

↓

3个通道

Pooling层在第一个卷积层后又对三个Feature Map采样 (取了3个小的)

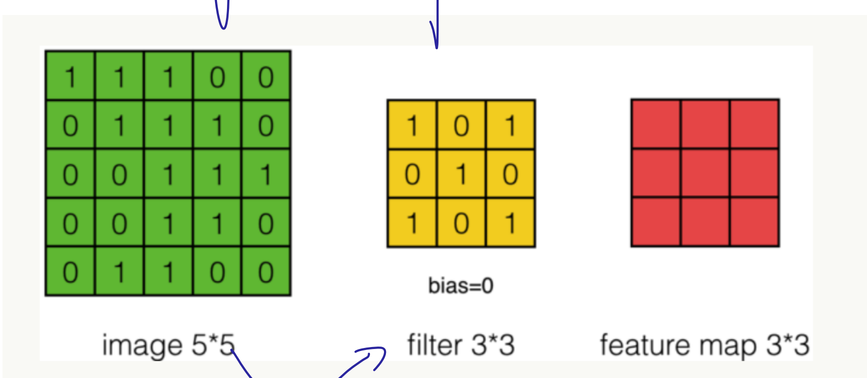
第2个卷积层 → 有5个Filter

每个Filter都将前3个采样层连一起形成的Feature Map.

5个Feature Map

第2个pooling变成5个变小的Feature Map

卷积层输出值计算



卷积

想这样，用 x_{ij} 表示第i行j列元素，

对filter每个权重编号

而用 $W_{m,n}$ 表示第m行n列权重，用 W_b 表示filter的偏置项

用 $a_{i,j}$ 表示 Feature Map 元素

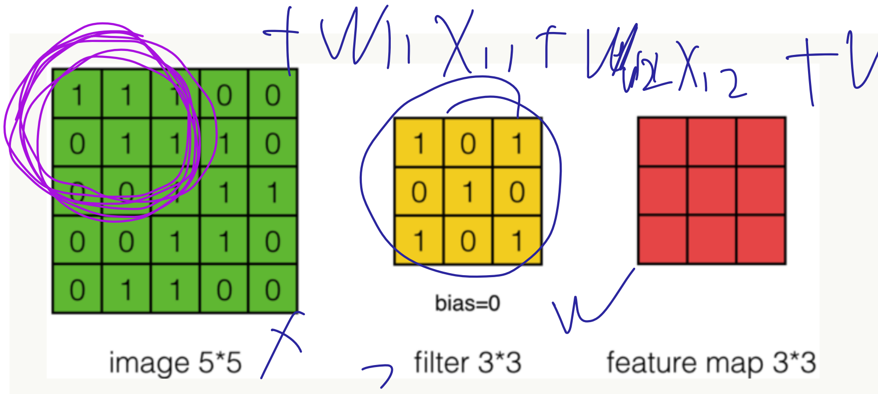
算卷积

$$a_{i,j} = f\Big(\underbrace{\sum_{m=0}^{2}\sum_{n=0}^{2} W_{m,n}\, x_{i+m,\,j+n} + W_b}_{(wx+b)}\Big)$$

Feature Map 左上角 $a_{0,0}$ 的计算结果为

$$a_{0,0} = f\Big(\sum_{m=0}^{2}\sum_{n=0}^{2} W_{m,n}\, x_{m+0,\,n+0} + W_b\Big)$$

$$= \mathrm{relu}(w_{0,0}x_{0,0} + w_{0,1}x_{01} + w_{02}x_{02} + w_{10}x_{10} + w_{11}x_{11} + w_{12}x_{12} + w_{20}x_{20} + w_{21}x_{21})$$



$= relu(1+0+0+0+1+0+0+0+1+0+1+0)$

$= relu(4) = 4$ 结$(0,0)=4$

算 $a_{0,1}$ 这值

$f\left(\sum_{m=0}^{2}\sum_{n=0}^{2} w_{m,n} x_{m+0,n+1} + w_b\right)$

$= relu(w_{0,0}x_{01} + w_{01}x_{02} + w_{02}x_{03} + w_{10}x_{11} + w_{11}x_{12} + w_{12}x_{13} + w_{20}x_{21} + w_{21}x_{22} + w_{22}x_{23} + w_b)$

$= relu(1\times1 + 0\times1 + 1\times0 + 0\times1 + 1\times1 + 0\times1 + 1\times0 + 0\times1 + 1\times1 + w_b)$

$= relu(1+0+0+0+1+0+0+0+1+0)$

$= relu(3) = 3$ 结$(0,1) = 3$

依此类推 Feature Map

4	3	4
2	4	3
2	3	4

算出来

当步幅为2时 feature Map 变为 2X2 的

那么如何计算是2X2

$$W_2 = (\underbrace{W_1}_{\text{输入宽度}} - \underbrace{F}_{\text{Filter宽度}} + 2p) / \underbrace{S}_{\text{步幅}} + 1$$

$$= (5-3+0)/2+1$$

$$= 2$$