

1 背景

卷积神经网络和普通神经网络的区别：

相同点	不同点
都使用反向传播算法(BP)来进行训练	网络结构不同，卷积神经网络具有局部连接、参数共享的特点

局部连接：相对于普通神经网络的全连接而言，是指这一层的某个节点只有上一层的部分节点相连。

参数共享：是指这一层中多个节点的连接共享一组参数。

一般神经网络	卷积神经网络
全连接：连接个数 $n*m$ 个	局部连接：连接个数 $i*m$ 个
参数不共享：参数个数 $n*m+m$	参数共享：参数个数 $i+1$

2 卷积神经网络组成

2.1 卷积层

卷积层是使用一系列卷积核与多通道输入数据做卷积的线性计算层。其目的是为了利用输入数据（如图像）中的特征的局域性和位置无关性来降低整个模型的参数量。假设卷积层的输入数据为三张二维的特征图（），输出数据是二张二维的特征图（），输入通道和输出通道分别为3和2。该卷积层的参数由矩阵和表示，其中，代表输出数据的各个通道，代表输入数据的各个通道。卷积层的整体计算过程为：当时，卷积核与二维数据分别执行卷积计算，然后求和，再加上偏置项，得到第一个输出通道对应的输出。同理，得到第2个。

卷积计算的结果与卷积核大小（假设卷积核长度和宽度相同）、步长和补零这三个参数密切相关。以卷积核与的卷积为例，具体的计算方法如下：

- 将原始输入数据在x和y方向上补0。比如，则原始输入数据的四周多了一圈值为0的点。
- 在补零后的二维数据上，从左上角开始，大小为的卷积核分别在x和y方向上以间隔s进行滑动，完成对二维数据的扫描。在卷积核滑动到的每个位置上，以该位置为中心的面积为的输入数据与该卷积核中的数据在该对应位置上一一相乘后求和，得到对应位置的输出值。

2.2 激活层

激活层是使用激活函数对输入数据做处理的一种非线性层。卷积操作本质上是对输入数据进行线性变换，为了实现神经网络的非线性建模能力，激活层使用非线性函数如sigmoid、tanh、ReLU等。

2.3 池化层

池化层是用于缩小数据规模的一种非线性计算层。

- 池化类型：平均或者最大化
- 池化核大小：
- 池化核的滑动间隔：

2.4 全连接层

全连接层是对输入数据直接做线性变换的线性计算层。这个层不是必须具有的，比如图像分割任务FCN就没有全连接层。

2.5 Dropout层

Dropout层是一种正则化层。全连接层的参数量非常庞大，很可能发生过拟合，需要正则化的操作。

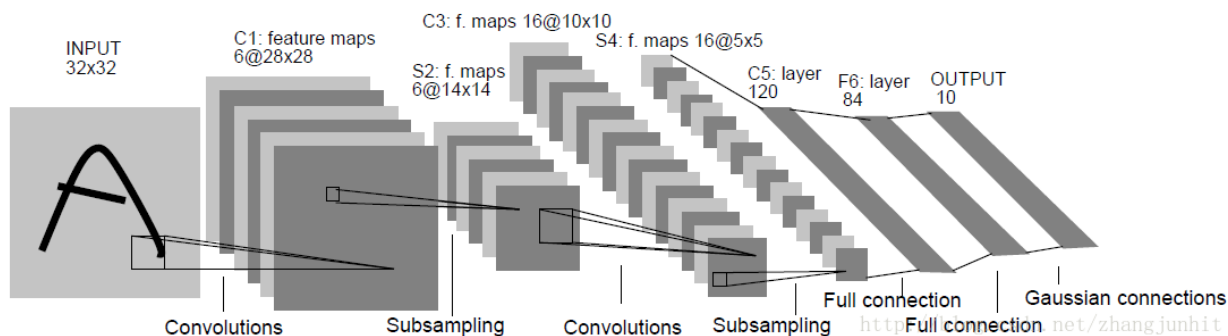
做法：在每次迭代计算的时候，将神经元以一定的概率值暂时随机丢弃，不参与训练。对不丢弃的数据做BP训练。然后再从原网络随机删除某些神经元，重复上述步骤。有利于提升泛化能力。

2.6 BN层

随着输入数据的分布随着迭代次数的不同而不同，因此，模型参数需要不断去适应这种随机迭代变换的输入分布，导致模型参数学习很慢。如果能够将每个卷积层输入的分布固定，这种输入分布归一化的方法叫做批归一化，相应的层为**BN**层。

一般而言，CNN在完成卷积或者全连接运算完成后进行**BN**计算，然后再激活。这种方法可以有效提高训练速度。

3 LeNet-5模型



3.1 输入

数据输入 **INPUT** 输入图像的尺寸归一化为 32×32

3.2 C1层

对输入图像进行第一次卷积运算（使用 6 个大小为 5×5 的卷积核），得到 6 个 C1 特征图（6 个大小为 28×28 的 feature maps, $32 - 5 + 1 = 28$ ）。我们再来看看需要多少个参数，卷积核的大小为 5×5 ，总共就有 $6 \times (5 \times 5 + 1) = 156$ 个参数，其中 +1 是表示一个核有一个 bias。对于卷积层 C1，C1 内的每个像素都与输入图像中的个像素和 1 个 bias 有连接，所以总共有 $156 \times 28 \times 28 = 122304$ 个连接（connection）。有 122304 个连接，但是我们只需要学习 156 个参数，主要是通过权值共享实现的。

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

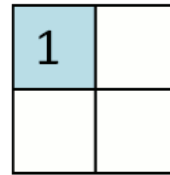
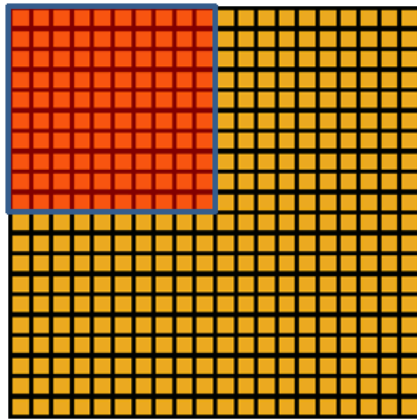
4		

Convolved
Feature

3.3 S2层

第一次卷积之后紧接着就是池化运算，使用核 2×2 进行池化，于是得到了 S2，6 个的 14×14 特征图（ $28 / 2 = 14$ ）。S2 这个 pooling 层是对 C1 中 2×2 的区域内的像素求和乘以一个权值系数再加上一个偏置，然后将这个结果再做一次映射。于是每

个池化核有两个训练参数，所以共有 $2*6=12$ 个训练参数，但是有 $5*14*14*6=5880$ 个连接。



Convolved
feature

Pooled
feature

3.4 C3层

第一次池化之后是第二次卷积，第二次卷积的输出是C3，16个 $10*10$ 的特征图，卷积核大小是 $5*5$ 。我们知道S2有6个 $14*14$ 的特征图，怎么从6个 $14*14$ 特征图得到16个 $10*10$ 特征图了？这里是通过S2的特征图特殊组合计算得到的16个特征图。具体如下：

	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

前6个feature map（对应上图第一个红框的6列）与S2层相连的3个feature map相连接（上图第一个红框），后面6个feature map与S2层相连的4个feature map相连接（上图第二个红框），后面3个feature map与S2层部分不相连的4个feature map相连接，最后一个与S2层的所有feature map相连。卷积核大小依然为 $5*5$ ，所以总共有个

$6*(3*5*5+1)+6*(4*5*5+1)+3*(4*5*5+1)+1*(6*5*5+1)=1516$ 参数。而图像大小为 $10*10$ ，所以共有151600个连接。 纵坐标的数字表示S2层，共6层。横坐标的

16个数字代表C3层。

3.5 S4层

S4是pooling层，窗口大小仍然是**2*2**，共计16个feature map，C3层的16个**10*10**的图分别进行以**2*2**为单位的池化得到16个**5*5**的特征图。这一层有**2*16**共32个训练参数，**5*5*5*16=2000**个连接。连接的方式与S2层类似。

3.6 C5层

C5层是一个卷积层。由于S4层的16个图的大小为**5*5**，与卷积核的大小相同，所以卷积后形成的图的大小为**1*1**。这里形成120个卷积结果。每个都与上一层的16个图相连。所以共有**(5*5*16+1)*120=48120**个参数，同样有**48120*1*1**个连接。

3.7 F6层

F6层是全连接层。F6层有84个节点，对应于一个的比特图，-1表示白色，1表示黑色，这样每个符号的比特图的黑白色就对应于一个编码。该层的训练参数和连接数是**(120+1)*84=10164**。ASCII编码图如下：

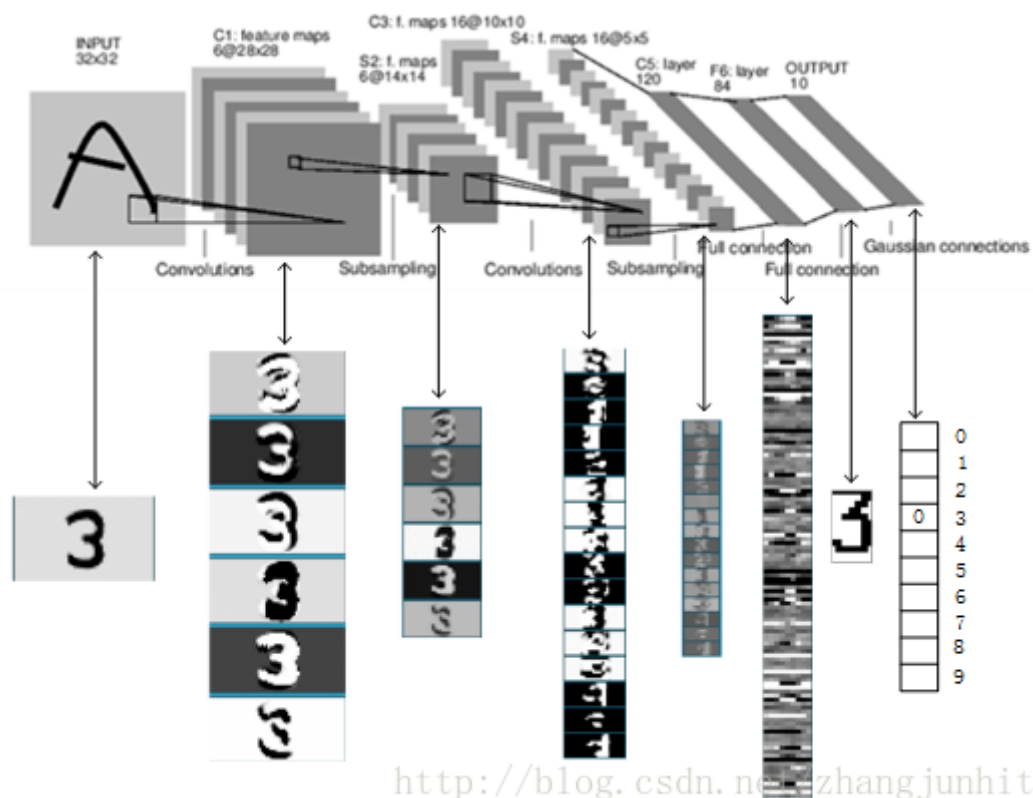


3.8 Output层

Output层也是全连接层，共有10个节点，分别代表数字0到9，且如果节点*i*的值为0，则网络识别的结果是数字*i*。采用的是径向基函数（RBF）的网络连接方式。假设*x*是上一层的输入，*y*是RBF的输出，则RBF输出的计算方式是：

$$y_i = \sum_j (x_j - w_{ij})^2$$

上式的值由*i*的比特图编码确定，*i*从0到9，*j*取值从0到**7*12-1**。RBF输出的值越接近于0，则越接近于*i*，即越接近于*i*的ASCII编码图，表示当前网络输入的识别结果是字符*i*。该层有**84*10=840**个参数和连接。



4 参考

深度学习 CNN卷积神经网络 LeNet-5详解 <https://blog.csdn.net/happyorg/article/details/78274066>

从LeNet-5看卷积神经网络CNNs <https://blog.csdn.net/wangjian1204/article/details/50479749>

《深入理解Tensorflow架构设计与实现原理》