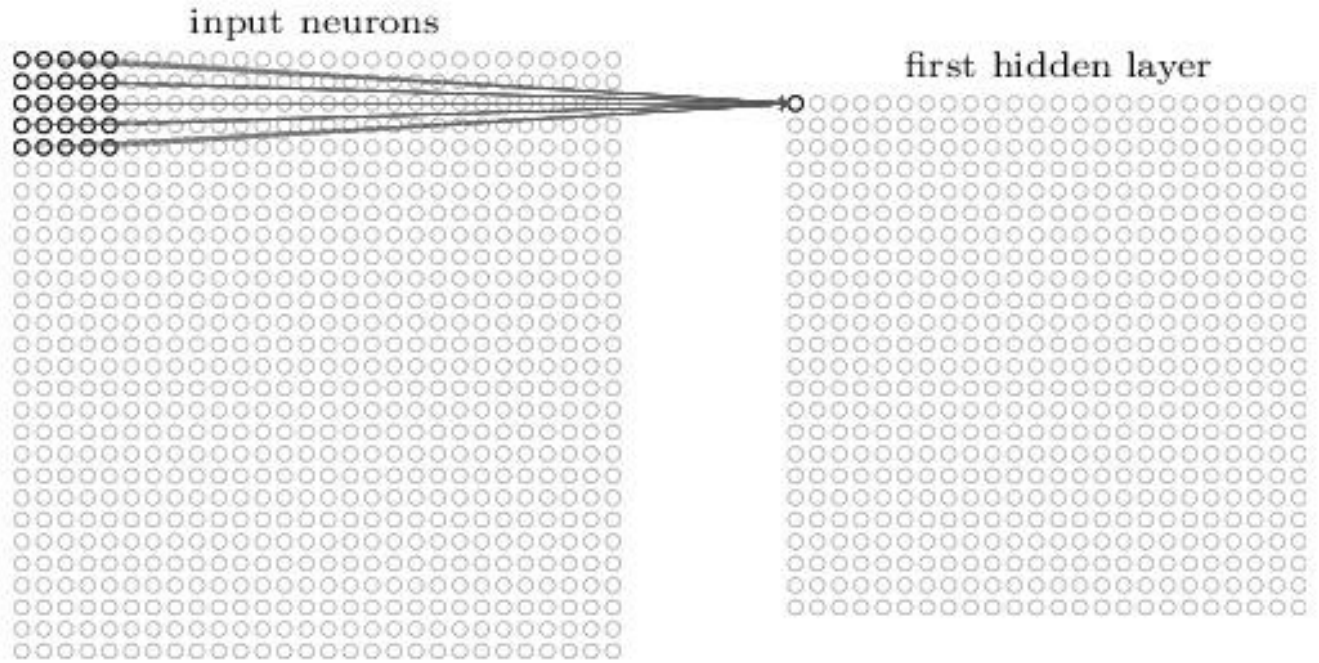
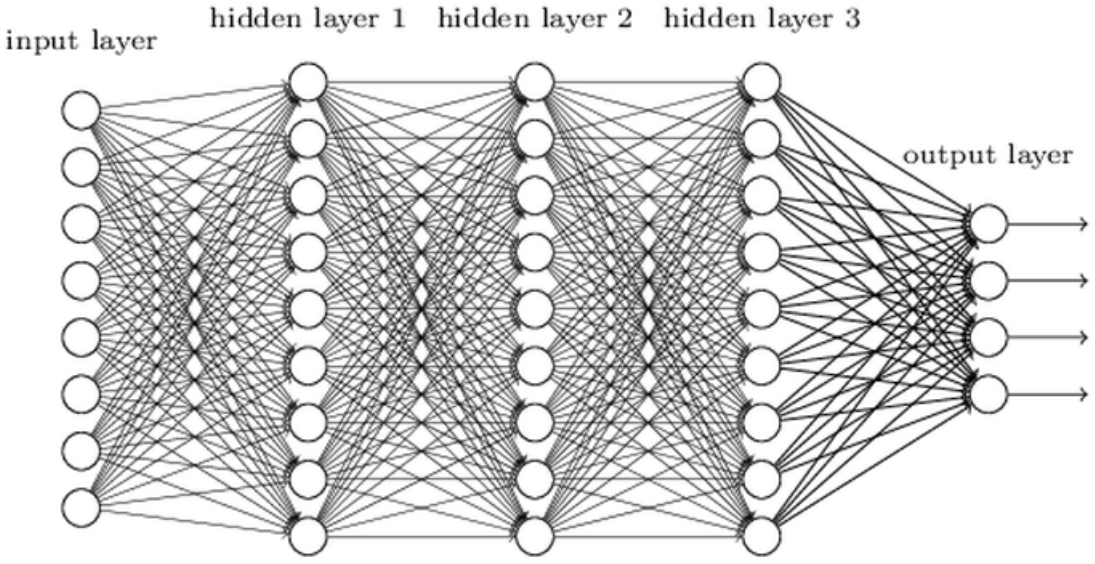


卷积神经网络(CNN)简介

这篇仍然是《神经网络与深度学习总结系列》中后面的章节。这里提前发布，日后会按照正常顺序给出文章连接列表。
识别手写数字的网络结构是这样的：



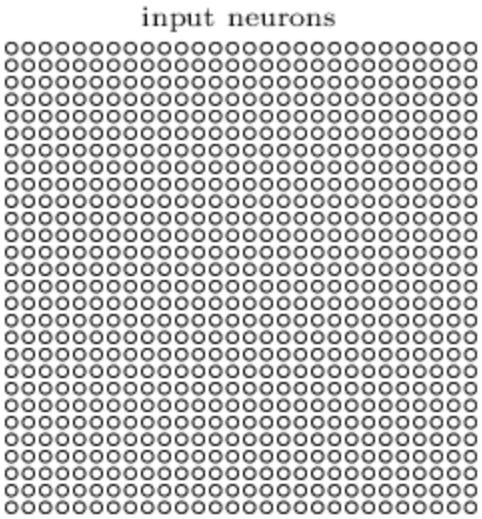
每一层网络都和相邻层全部连接。但是这样并没有考虑到图片中像素的空间分布，不管两个像素离的很近还是非常远都一视同仁，显然这样做是不合理的。

所以卷积神经网络就出现了，它考虑到了输入值的（像素的）空间分布，（再加上一些人工设定的特性 例如共享权重等）使得它非常容易训练。也就可以做出更深层的网络结构，拥有更好的识别效果。

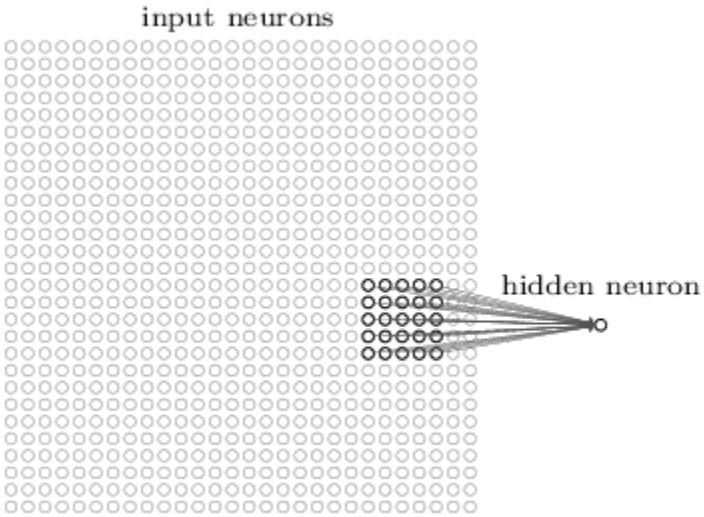
卷积神经网络主要概念有：local receptive fields, shared weights, and pooling。下面一个个解释。

【local receptive fields】

输入是 28x28的像素值：



对于第一个隐藏层，每一个隐藏层神经元 与 输入层中 5x5的神经元有连接。



这个5x5的区域就叫做 感受视野(local receptive field)，表示一个隐藏层神经元在输入层的感受区域。这5x5=25个连接对应有25个权重参数w，还有一个全局共用的基值b。

当 local receptive field 沿着 整个输入照片向右（向下）滑动时，每一个 local receptive field 在第一个隐藏层都对应一个隐藏神经元。

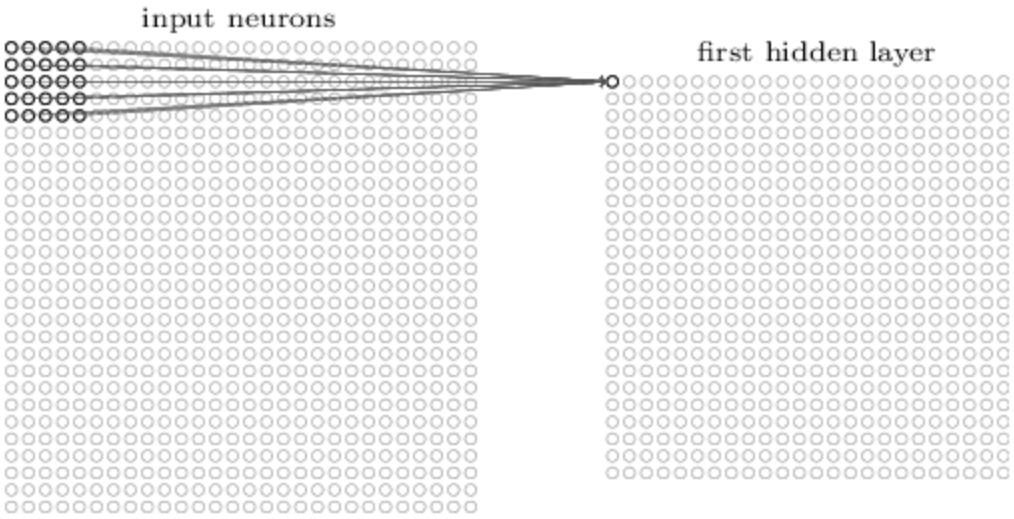


图1： 第一个 local receptive field

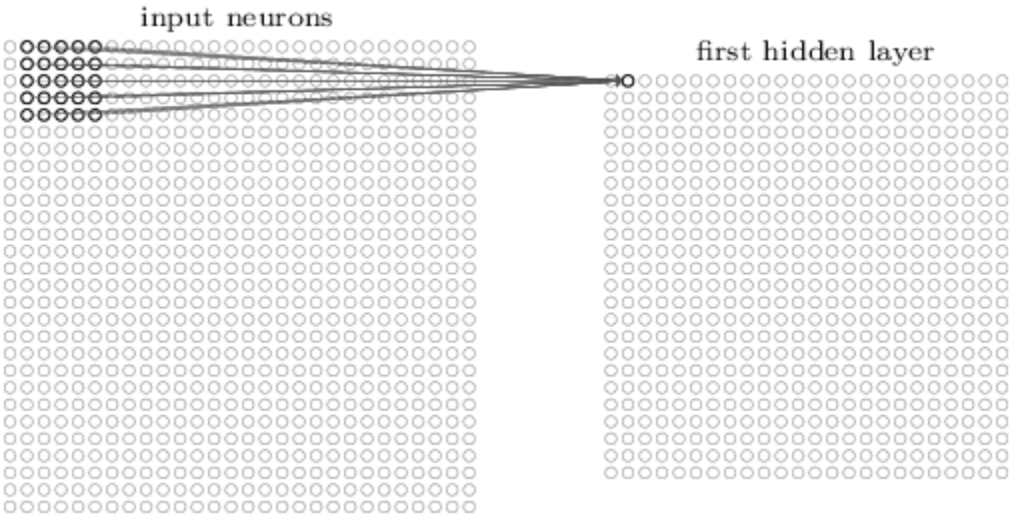


图2：第二个 local receptive field （向右滑动一个像素）

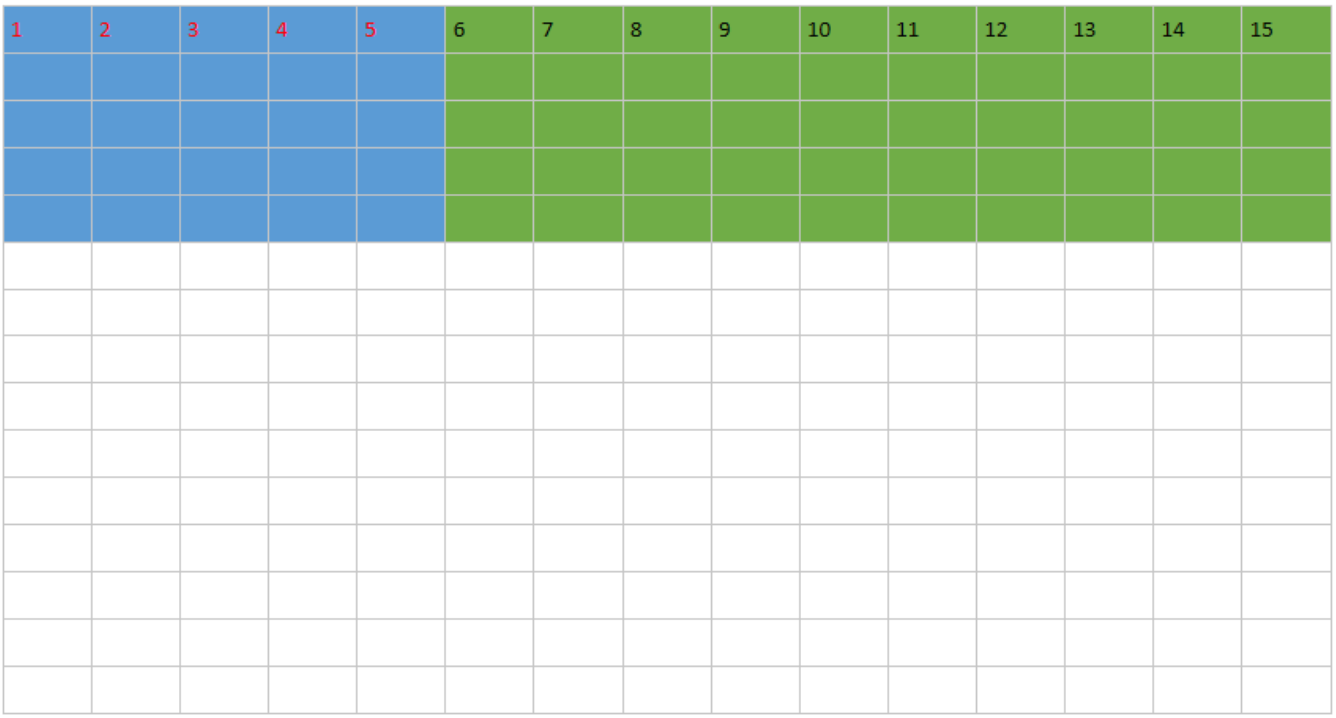
不断的向右（向下）滑动就可以得到第一个隐藏层。

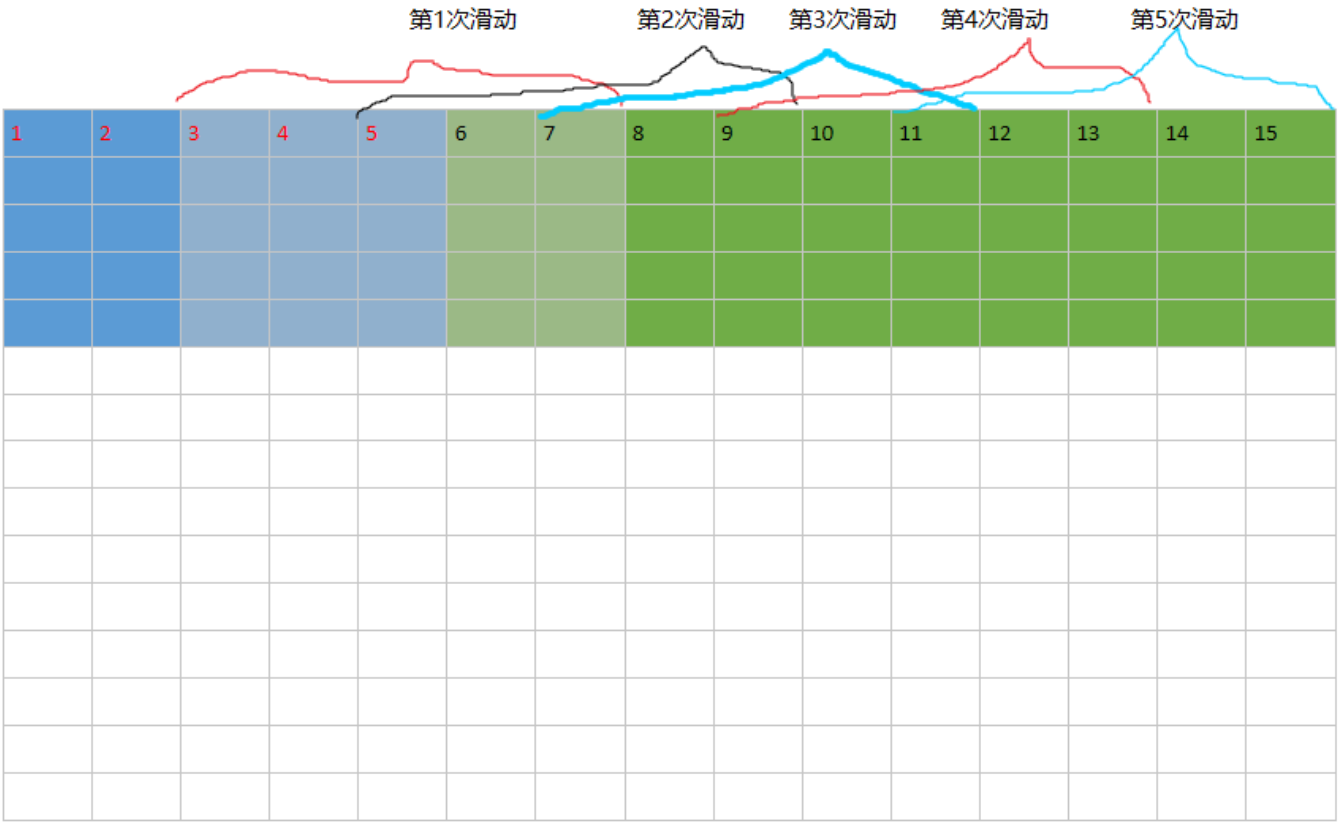
- 28x28 的输入照片 （W=28）
- 5x5的local receptive fields(滑窗，也叫卷积核 ） （K=5)
- 滑动步长（(stride）为1 （S = 1）

可以得到 24x24的隐藏层神经元。

假如现在 W=15，K=5， S=2 对应的隐藏神经元是多少?

蓝色表示 左上角的5x5感受视野，绿色是其向右滑动 轨迹。



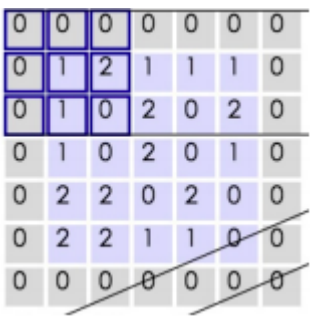


因为 W=15，K=5， S=2

> 一次向右滑动对应 6个隐藏神经元 =>
>
> = 5次滑动 + 1个原始感受视野
>
> = 10个绿色区域/步长2 + 1个原始感受视野
>
> = （W-K）/S + 1

得到隐藏层对应的计算公式： 隐藏层边长 = （W-K）/S + 1 。

（不过有时候为了控制输出的隐藏层空间分布 会在 输入层外围做零填充，假设填充 P个像素， 此时：边长= （W - K + 2P）/S + 1 ，特别的 当S=1时， 设置零填充为 $P = (K-1)/2$ 可以保证 输入层与输出层有相同 的空间分布 ）



零填充示意图

【Shared weights and biases】

上文提到 每一个隐藏层神经元 对应 5x5=25个权重参数w和一个基值参数b，实际上 我们规定 每一个隐藏层神经元的这些25个权值w 和 b都共享。 也就是说隐藏层神经元共享权值。

大大减少参数个数：

如此一来 上图的 隐藏层 只有 5x5+1 = 26 个参数 ， 而如果时全部连接则需要 28x28xN (N代表隐藏神经元个数) 将远远多于26个参数，共享权值 就大大减小了参数个数 。

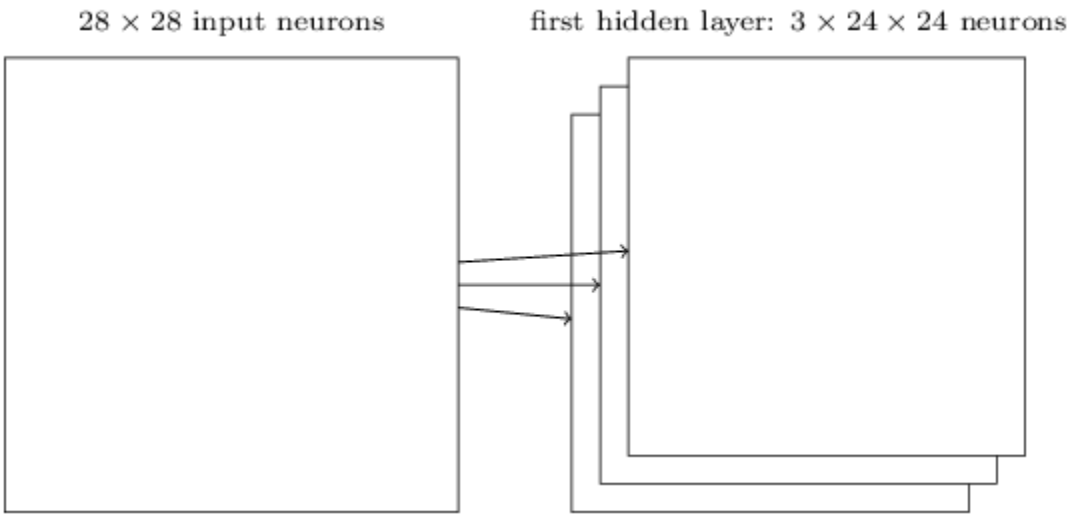
> 用数学化的语言描述一下（大致了解即可）对于第j，k个隐藏神经元 对应的值为
>
> $$\sigma\left(b+\sum_{l=0}^{4}\sum_{m=0}^{4}w_{l,m}a_{j+l,k+m}\right)$$

平移不变性：

这种做法有着很直观的意义： 上图第一个隐藏层的神经元 都在 检测 同一个特征（或模式）是否出现。（相同的权值自然就是在检测同一个特征）。例如 每一个神经元都在 检测是否出现 “垂直边” ，无论这个垂直边在图片中的那个位置上都可以被卷积核扫描到。 这就是 卷积神经网络的 平移不变特性 （ translation invariance）。

因此我们 通常把输入层 到 隐藏层的映射称为 特征映射(a feature map) ， 卷积核共享的权值 w 叫做 shared weights.，b叫做 shared bias。 shared weights和shared bias 定义了一个核或者过滤器(a kernel or filter)

实际上我们在应用中会有很多特征图，分别检测不同 的特征：



上图有3个特征图，每个特征图 对应 5x5的 shared weights 和1个shared bias.。

为什么叫做卷积神经网络?

因为下面的公式里的包含卷积操作

$$\sigma\left(b+\sum_{l=0}^{4}\sum_{m=0}^{4}w_{l,m}a_{j+l,k+m}\right)$$

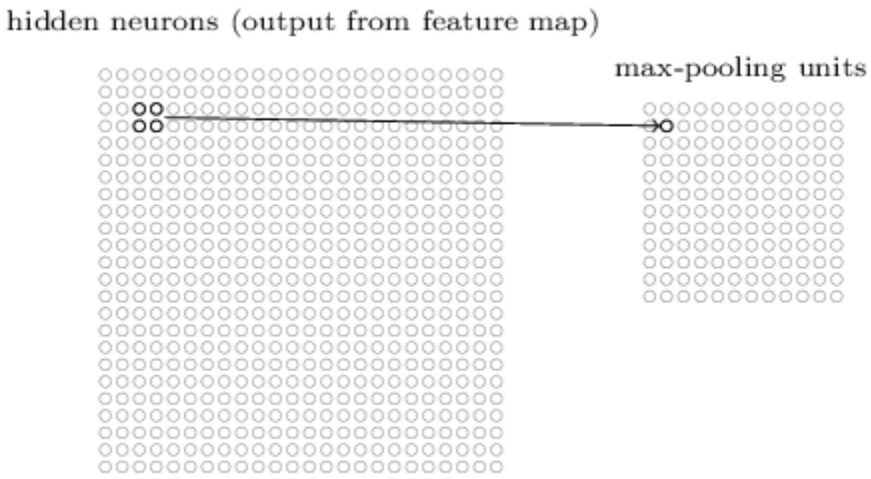
用卷积符号可以把公式写成：

$$a^1=\sigma(b+w*a^0)$$

$*$ 是卷积操作，a^1 表示特征映射的输出响应，a^0 表示特征映射的输入响应。

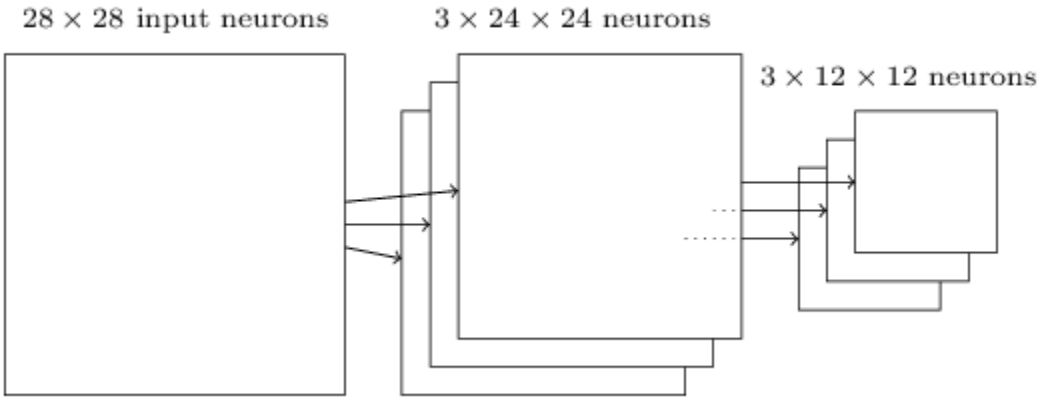
【Pooling layers】

池化层一般接在卷积层后面，用于简化卷积的输出结果。 其实就是把卷积层的输出结果进行压缩：

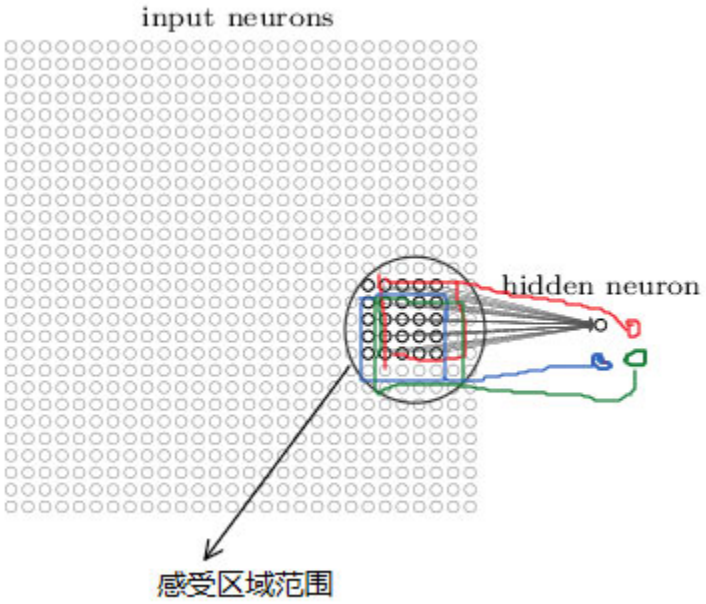


例如 上图就是使用 max-pooling 把四个神经元压缩为一个神经元：取最大的像素值，其它丢弃。（当然也有其它的 pooling 例如 L2 pooling 是把 2x2的区域值 平方，求和，开根号）。 上图 24x24 的特征图 经过 2x2的max-pooling 之后变成了 12 x 12的特征图。

对于3个特征图 先卷积后池化：



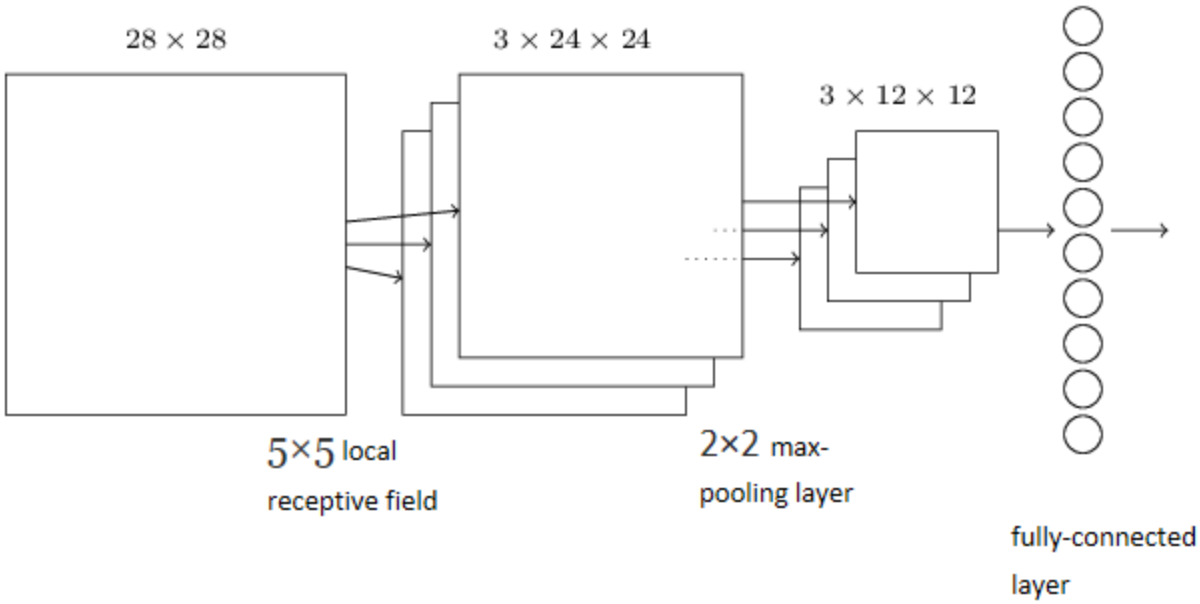
我们可以把 max-pooling看作去检测某个特征是否在输入图像的某个感受区域内出现过。 2x2的隐藏层神经元对应输出图像中的感受局域是 下图 黑色圆圈所圈部分：



我们的直觉是这样的： 一旦一个特征被检测到，它相对于其它特征的位置是精确的还是模糊的 差别并不大。 但是我们获得了很大的好处： 池化后特征数量大大减少了，之后的网络层需要的参数也就减少了。

【综合在在一起 】

MNIST image识别 架构图：



接下来就可以用随机梯度下降和反向传播进行训练。不过由于不再是全链接，反向传播算法需要做一些修改。

总的来说 卷积神经网络 大大减少了网络参数，具备平移不变特性，多个卷积层连接在一起就是 一个个抽象度 逐次增加的 特征图。具体来讲假如要识别一只猫，最底层的检测初级特征（例如 垂直边，斜边等） 然后下一层会基于前一层的基础特征检测 抽象一点的特征 （例如 是否有 圆圈等），接着检测是否有 鼻子，眼睛等， 最后一层会检测是否是一个猫。

参考：

Neural networks and deep learning