卷积神经网络

本周工作重点：主要初学习CNN

看了一位大神的博客，他是这么总结的：

第一点，在学习Deep learning和CNN之前，总以为它们是很了不得的知识，总以为它们能解决很多问题，学习了之后，才知道它们不过与其他[机器学习](http://lib.csdn.net/base/machinelearning" \o "机器学习知识库" \t "http://blog.csdn.net/stdcoutzyx/article/details/41596663/_blank)算法如svm等相似，仍然可以把它当做一个分类器，仍然可以像使用一个黑盒子那样使用它。

第二点，Deep Learning强大的地方就是可以利用网络中间某一层的输出当做是数据的另一种表达，从而可以将其认为是经过网络学习到的特征。基于该特征，可以进行进一步的相似度比较等。

第三点，Deep Learning算法能够有效的关键其实是大规模的数据，这一点原因在于每个DL都有众多的参数，少量数据无法将参数训练充分。

在图像处理中，由于图像的表示往往是用图像向量做的，若隐藏层的数目和输入一样，那么用神经网络就会出现要处理的参数过多，那么就会大大拖慢运算速度。所以这时就要用卷积神经网络来加快速度。

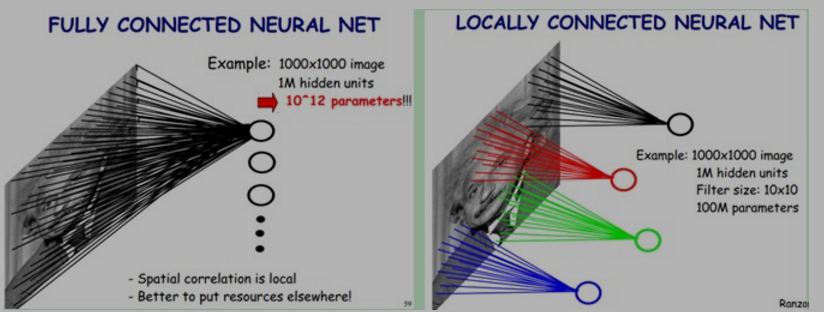
卷积神经网络：

主要有三个基本概念：局部感受野（local receptive fields），共享权重（shared weights），池化亦成为混合（pooling）。

局部感受野：

基于认为图像的空间联系是局部的像素联系比较密切，而距离较远的像素联系较弱。因此神经元没必要进行全局的感知，只需对局部进行感知即可，然后在更高层将局部的信息进行综合就形成了全局的信息。

全局感知和局部感知如下图：



图一：左图是全局感知，右图是局部感知

共享权重：

即使进行了局部感知，但是参数还是太多，那么就要上用上参数共享这个好东西了。

基于图像的一部分特征和和其他部分是一样的，这样就意味着一部分学习的特征也能够应用在另一部分。例如，从大图像中随机选出8\*8的样本，并且从这个小块样本中学习到了一些特征，这时我们可以把从这个 8x8 样本中学习到的特征作为探测器（卷积核），应用到这个图像的任意地方中去。特别是，我们可以用从 8x8 样本中所学习到的特征跟原本的大尺寸图像作卷积，从而对这个大尺寸图像上的任一位置获得一个不同特征的激活值。

卷积如下图；

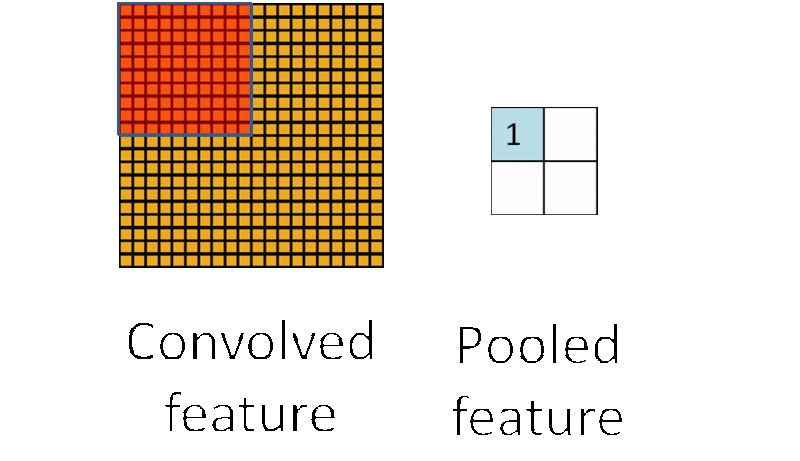


共享权重能够用不同的权重对局部图像进行卷积，形成不同的特征层。

池化（混合）：

在通过卷积获取了特征后，下一步要做的就是分类。但由于巨大的计算量，我们采取池化的方法。首先，我们做卷积是由于图像具有“静态性”的属性，这就意味着在一个图像区域有效的特征值极有可能在另一个区域也适用。因此，为了描述一个大的图像，那么我们可以对不同位置的特征值进行聚合统计，例如平均值、最大值。

池化：



还有L2池化：选取区域中激活值平方和的根作为结果。

卷积具体操作：

<https://www.cnblogs.com/qggg/p/6832342.html>

池化具体操作：

<https://blog.csdn.net/mao_xiao_feng/article/details/53453926>

PS:PS：参数最后一维都是通道数，同一个通道的矩阵一起计算，不通通道分开计算，最终合并在同一行里。例如，RGB，先池化R，再G，再B，在把这三个值放在同一行不同列里

关于卷积核大小和池化大小对输入和输出的影响：原输入n\*n

1. 卷积核（kernel size）a\*a，步长（strides）b\*b，填充（padding）p

输出层大小：（n-a+2\*p）/b+1

PS:在Tensorf函数中，可能padding=’SAME’时，输出和输入是一样大小的

1. 池化窗口大小（kernel size）a\*a，步长（strides）b\*b，填充（padding）p

输出层大小：（n-a+2\*p）/b+1