**一文了解各种卷积结构原理及优劣**

http://www.dataguru.cn/article-11860-1.html

**摘要**: 卷积神经网络作为深度学习的典型网络，在图像处理和计算机视觉等多个领域都取得了很好的效果。卷积核大小（Kernel Size）：定义了卷积操作的感受野。在二维卷积中，通常设置为3，即卷积核大小为3×3。步幅（Stride） ...

|  |
| --- |
| 卷积[神经网络](http://www.dataguru.cn/article-8976-1.html?union_site=innerlink)作为[深度学习](http://www.dataguru.cn/article-9400-1.html?union_site=innerlink)的典型网络，在图像处理和计算机视觉等多个领域都取得了很好的效果。  Paul-Louis Pröve在Medium上通过这篇文章快速地介绍了不同类型的卷积结构（Convolution）及优势。为了简单起见，本文仅探讨二维卷积结构。  卷积  首先，定义下卷积层的结构参数。  http://attachbak.dataguru.cn/attachments/portal/201707/31/140628fl0nloilg0kxh8lx.jpg  卷积核为3、步幅为1和带有边界扩充的二维卷积结构  卷积核大小（Kernel Size）：定义了卷积操作的感受野。在二维卷积中，通常设置为3，即卷积核大小为3×3。  步幅（Stride）：定义了卷积核遍历图像时的步幅大小。其默认值通常设置为1，也可将步幅设置为2后对图像进行下采样，这种方式与较大池化类似。  边界扩充（Padding）：定义了网络层处理样本边界的方式。当卷积核大于1且不进行边界扩充，输出尺寸将相应缩小；当卷积核以标准方式进行边界扩充，则输出数据的空间尺寸将与输入相等。  输入与输出通道（Channels）：构建卷积层时需定义输入通道I，并由此确定输出通道O。这样，可算出每个网络层的参数量为I×O×K，其中K为卷积核的参数个数。例，某个网络层有64个大小为3×3的卷积核，则对应K值为 3×3 =9。  空洞卷积  空洞卷积（atrous convolutions）又名扩张卷积（dilated convolutions），向卷积层引入了一个称为 “扩张率(dilation rate)”的新参数，该参数定义了卷积核处理数据时各值的间距。  http://attachbak.dataguru.cn/attachments/portal/201707/31/140628hd70lnfsx14dnnas.jpg  卷积核为3、扩张率为2和无边界扩充的二维空洞卷积  一个扩张率为2的3×3卷积核，感受野与5×5的卷积核相同，而且仅需要9个参数。你可以把它想象成一个5×5的卷积核，每隔一行或一列删除一行或一列。  在相同的计算条件下，空洞卷积提供了更大的感受野。空洞卷积经常用在实时图像分割中。当网络层需要较大的感受野，但计算资源有限而无法提高卷积核数量或大小时，可以考虑空洞卷积。  转置卷积  转置卷积（transposed Convolutions）又名反卷积（deconvolution）或是分数步长卷积（fractially straced convolutions）。  反卷积（deconvolutions）这种叫法是不合适的，因为它不符合反卷积的概念。在深度学习中，反卷积确实存在，但是并不常用。实际上，反卷积是卷积操作的逆过程。你可以这么理解这个过程，将某个图像输入到单个卷积层，取卷积层的输出传递到一个黑盒子中，这个黑盒子输出了原始图像。那么可以说，这个黑盒子完成了一个反卷积操作，也就是卷积操作的数学逆过程。  转置卷积与真正的反卷积有点相似，因为两者产生了相同的空间分辨率。然而，这两种卷积对输入数据执行的实际数学运算是不同的。转置卷积层只执行了常规的卷积操作，但是恢复了其空间分辨率。  <http://attachbak.dataguru.cn/attachments/portal/201707/31/140812q6lzlca46ga62lc6.gif>  卷积核为3、步幅为2和无边界扩充的二维卷积结构  举个例子，假如将一张5×5大小的图像输入到卷积层，其中步幅为2，卷积核为3×3，无边界扩充。则卷积层会输出2×2的图像。  若要实现其逆过程，需要相应的数学逆运算，能根据每个输入像素来生成对应的9个值。然后，将步幅设为2，遍历输出图像，这就是反卷积操作。  <http://attachbak.dataguru.cn/attachments/portal/201707/31/140628oxtrh7tyhha47sfa.jpg>  卷积核为3×3、步幅为2和无边界扩充的二维转置卷积  转置卷积和反卷积的共同点在于两者输出都为5×5大小的图像，不过转置卷积执行的仍是常规的卷积操作。为了实现扩充目的，需要对输入以某种方式进行填充。  你可以理解成，至少在数值方面上，转置卷积不能实现卷积操作的逆过程。  转置卷积只是为了重建先前的空间分辨率，执行了卷积操作。这不是卷积的数学逆过程，但是用于编码器-解码器结构中，效果仍然很好。这样，转置卷积可以同时实现图像的粗粒化和卷积操作，而不是通过两个单独过程来完成。  可分离卷积  在可分离卷积（separable convolution）中，可将卷积核操作拆分成多个步骤。卷积操作用y=conv(x, k)来表示，其中输出图像为y，输入图像为x，卷积核为k。接着，假设k可以由下式计算得出：k=k1.dot(k2)。这就实现了一个可分离卷积操作，因为不用k执行二维卷积操作，而是通过k1和k2分别实现两次一维卷积来取得相同效果。  <http://attachbak.dataguru.cn/attachments/portal/201707/31/140628ruz2lustdaqlbkym.jpg>  X、Y方向上的Sobel滤波器  Sobel算子通常被用于图像处理中，这里以它为例。你可以分别乘以矢量[1,0,-1]和[1,2,1]的转置矢量后得到相同的滤波器。完成这个操作，只需要6个参数，而不是二维卷积中的9个参数。  这个例子说明了什么叫做空间可分离卷积，这种方法并不应用在深度学习中，只是用来帮你理解这种结构。  在神经网络中，我们通常会使用深度可分离卷积结构（depthwise separable convolution）。  这种方法在保持通道分离的前提下，接上一个深度卷积结构，即可实现空间卷积。接下来通过一个例子让大家更好地理解。  假设有一个3×3大小的卷积层，其输入通道为16、输出通道为32。具体为，32个3×3大小的卷积核会遍历16个通道中的每个数据，从而产生16×32=512个特征图谱。进而通过叠加每个输入通道对应的特征图谱后融合得到1个特征图谱。最后可得到所需的32个输出通道。  针对这个例子应用深度可分离卷积，用1个3×3大小的卷积核遍历16通道的数据，得到了16个特征图谱。在融合操作之前，接着用32个1×1大小的卷积核遍历这16个特征图谱，进行相加融合。这个过程使用了16×3×3+16×32×1×1=656个参数，远少于上面的16×32×3×3=4608个参数。  这个例子就是深度可分离卷积的具体操作，其中上面的深度乘数（depth multiplier）设为1，这也是目前这类网络层的通用参数。  这么做是为了对空间信息和深度信息进行去耦。从Xception模型的效果可以看出，这种方法是比较有效的。由于能够有效利用参数，因此深度可分离卷积也可以用于移动设备中。 |