Convolution学习

1、普通卷积（Convolution）

对于普通的卷积，假设其输入通道数假定为，卷积核大小为，输出通道数为。那么其总共的参数量为。由此可以看出，使用卷积（也称PW卷积，Pointwise Convolution）时，只需调整的数目，即可通过改变通道数来实现参数的减少或增加。

为什么卷积核常为3：5x5的卷积层的正则等效于2个3 x 3的卷积层的叠加。而这样的设计不仅可以大幅度的减少参数，其本身带有正则性质的 convolution map 能够更容易学一个 generlisable, expressive feature space。

2、组卷积（Group Convolution）

对于组卷积，同样的假设其输入通道数假定为，卷积核大小为，输出通道数为，分组数目为。那么首先将输入的分为份，然后对每一份分别进行卷积，即对份输入通道为的数据进行卷积，如果总的输出通道数为，那么每一份的输出通道数就为，最后的输出将每一份都拼接起来。因此其总共的参数量为，因此只有原来参数量的。

3、DW卷积（Depthwise Convolution）

DW卷积即为组卷积中分组数目等于输入通道数目，即，此时参数量变为。

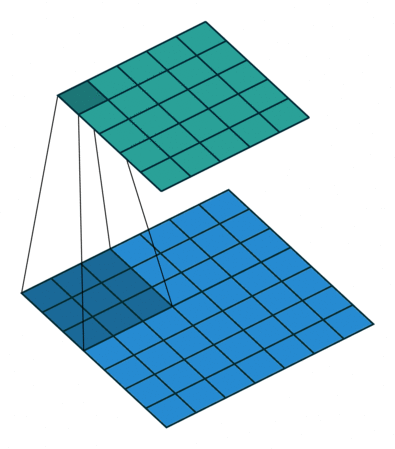
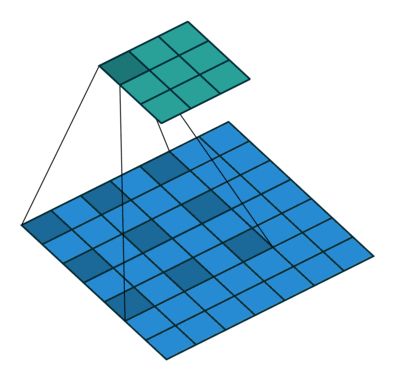
4、可分离卷积（Separable Convolution）

Separable Convolution的核心思想是将一个完整的卷积运算分解为两步进行，分别为Depthwise Convolution与Pointwise Convolution。即使用DW卷积只需很少的参数量即可完成卷积操作，卷积结束后使用卷积来对通道数目进行改变，通常为增加通道数目，以此DW卷积的输出通道数可以更少，因此参数量也更少。同时由于DW卷积时，并未考虑不同通道上 相同空间位置上的信息。而卷积就可以进行利用。避免了DW卷积的缺点。

5、膨胀卷积（Dilated convolution）或空洞卷积（Atrous Convolution）

由于普通卷积时，其感受野与卷积核有关，但卷积核大了之后，所带来的参数量也急速增多，而如果一直堆叠卷积核较小的卷积层时，神经网络层数过高仍然会有退化现象。而当普通卷积的步距增大时，虽然扩大了感受野，但卷积后的特征图大小会减少，因此提出了膨胀卷积/空洞卷积。

膨胀卷积不仅可以增加感受野，同时也能保持原来的输入特征图的W、H。

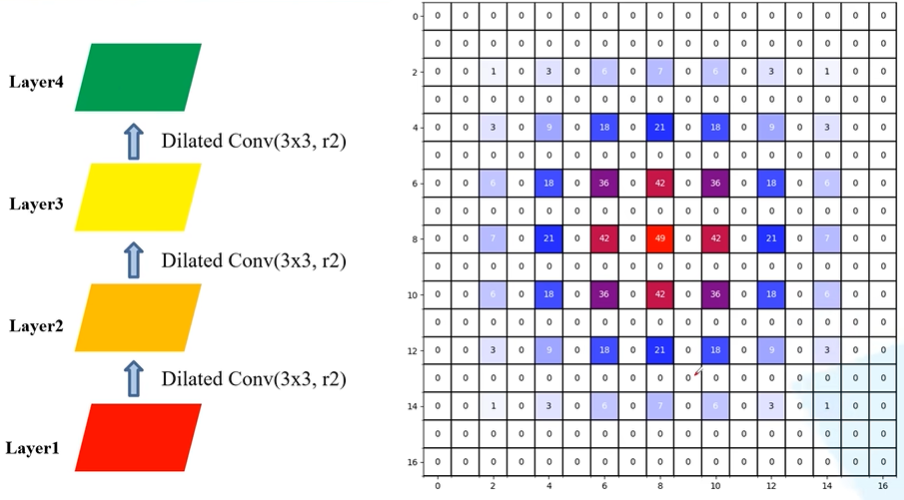
左图为普通卷积，右图为膨胀卷积

膨胀卷积所采用的卷积操作为在原特征图中，特征图上卷积的相邻两个点之间存在间隔。即在普通卷积时注入空洞，以此来增加感受野。相比原来的普通卷积，膨胀卷积多了一个超参数称之为dilation rate指的是卷积核的间隔数量(e.g. 普通卷积是 dilatation rate=1，即间隔为1，上图中膨胀卷积间隔为2)。

对膨胀卷积做适当的padding，可以使其输入输出的特征图大小不变，其输出特征图大小的计算公式如下：

在实际使用膨胀卷积时，常取，当时，可求得。

膨胀卷积有一个潜在问题The gridding effect问题。该问题为当使用同样的膨胀系数（dilation rate）时，其后面层的特征图虽然感受野变大了，但并未使用该感受野内全部的元素，而是只使用了一般分。如下图膨胀系数全为2时，Layer4层某个点的感受野虽然为13×13，但并未完全使用该感受野内的全部元素，只是使用了部分元素。



理想的情况下，连续使用膨胀卷积时，应该将感受野内的所有元素均使用上，这时提出了一个叫Hybrid Dilated Convolution (HDC)的设计方法。假设有N个卷积核均为的膨胀卷积层，其膨胀系数分别为，首先需满足，其次令，如果按照如下公式计算得到的，那么可以满足使用全部元素的要求。

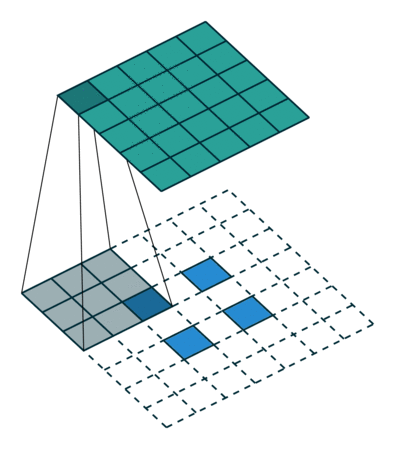
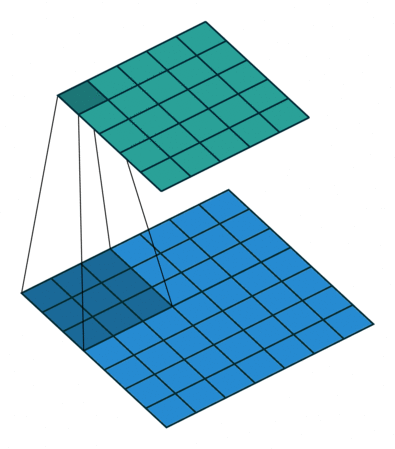
同时HDC设计方法还建议将膨胀系数设置为锯齿形状。如[1,2,3,1,2,3]这样。且膨胀系数的公约数不能大于1。

在实际中，组卷积（包括DW卷积）、膨胀卷积都不好优化，即使组卷积运算量少，但速度甚至可能不如普通卷积。

6、转置卷积（transposed convolution）（fractionally-strided convolutions）（deconvolution）

转置卷积并非卷积的逆，其只是在形状上是互逆的，但在数值上是没有关系的。其目的是为了扩大特征图的大小，充当一个上采用的作用。转置卷积也是卷积的一种。转置卷积的由来是由扩展（扩展分为两种，一种为对input扩展，一种对kernel扩展，两者均可以将卷积变为矩阵乘积）后的卷积核与普通卷积后的结果进行转置相乘得来。即首先使用普通卷积，得到结果，然后对每一处卷积的卷积核都进行扩展，拉平，排列成一个矩阵，对该矩阵进行转置再与普通卷积的结果进行矩阵乘积即可。转置卷积运算步骤如下：

1. 在输入特征图的元素之间填充行、列
2. 在输入特征图四周填充行、列
3. 将卷积核参数上下、左右翻转
4. 做正常的卷积运算。



左图为普通卷积，右图为转置卷积

在上面右图中，其，因此在元素之间填充1行、列。其，因此在特征图四周填充2行、列。输出特征图大小的计算公式如下：

但由于转置卷积在生成任务中如果卷积核大小为3，步长为2，会有非常明显的棋盘效应，因此更推荐使用最临近插值或双线性插值后（此时已满足所需的特征图大小）再接一个卷积来取代转置卷积。

棋盘效应是由于转置卷积的“不均匀重叠”（Uneven overlap）的结果。使图像中某个部位的颜色比其他部位更深。其具体原因为：在转置卷积时，如果卷积核（Kernel）大小不能被步长（Stride）整除时，转置卷积输出的结果就会不均匀重叠。

7、可变形卷积（Deformable Convolution）

对于普通的卷积，其卷积公式可以写为：

其中。

可变形卷积网络 v1（DCNv1）

可变形卷积通过在传统的卷积操作中引入额外的偏移参数，使得卷积核能够根据输入特征的几何形变进行自适应调整，其卷积公式可以写为：

为了生成这些偏移量，可变形卷积使用了一个额外的卷积层。这个层的输出尺寸与输入特征图相同，但通道数为2N（N为卷积核大小），分别对应x和y方向上的偏移量。这些偏移量随后被用来调整卷积核的采样点位置。其工作原理如下：

1. **偏移学习：**在输入特征图上应用额外的卷积层来生成偏移量。这些偏移量表示每个采样点应从其原始位置偏移多少。
2. **偏移应用：**按照思想是利用学到的偏移量调整卷积核的采样点位置，并进行特征提取，但是为了便于计算，是将学习到的偏移反向应用于特征图中，即通过重采样的方法将特征图采样为偏移后的特征图。

由于偏移量可能导致卷积核采样点落在非整数像素位置，因此需要使用插值算法来计算这些位置的像素值。常用的插值方法包括双线性插值，它能够提供足够的精度来估计非整数位置的像素值。梯度是通过双线性插值来进行反向传播的。

可变形卷积网络 v2 (DCNv2)

DCNv2 引入了一种调制机制，使网络不仅可以学习偏移量，还可以学习调制标量。这些标量调整输入特征在不同空间位置的幅度，为网络提供了额外的自由度，使其可以选择性地关注相关特征，同时忽略不相关的特征。该机制有助于网络更好地使其感受野适应输入数据的相关部分，从而提高特征提取过程的精度和鲁棒性。

其中的调制标量也是通过在相同的输入特征图x上应用一个单独的卷积层得到的。即现在可变形卷积使用了一个额外的卷积层。这个层的输出尺寸与输入特征图相同，但通道数为3N（N为卷积核大小），分别对应x和y方向上的偏移量。剩下的N个通道送到sigmoid层以获得调制标量。

可变形卷积网络 v3（DCNv3）

DCNv3 的主要改进包括：

1. 用dw卷积（深度可分离卷积）生成和，提升模型的效率。
2. 将空间聚集过程分成G组，每个组都有单独的采样偏移和调制规模 。DCNv3生成的大小是(N,H,W,group\*kernel\_size\*kernel\_size), offset的大小是(N,H,W,2\*group\*kernel\_size\*kernel\_size)，每个组都有不同的采样偏移和调制规模。而DCNv2生成的大小是(N,H,W,kernel\_size\*kernel\_size), offset的大小是(N,H,W,2\*kernel\_size\*kernel\_size)。
3. 将基于element-wise的sigmoid归一化改为基于样本点的softmax归一化。这是因为DCNv2中的由sigmoid函数进行element-wise的归一化。因此，每个调制标量都在[0,1]范围内，所有采样点的之和在0到k之间变化，这就导致在使用大规模参数和数据进行训练时，DCNv2层的梯度不稳定。

可变形卷积网络 v4 (DCNv4)

DCNv4的主要改进包括：

1. **去除softmax归一化**。这种简化不仅加快了计算速度，而且还降低了网络的复杂性，而不会牺牲准确性。因为对于每个位置拥有独立采样窗口的操作来说，softmax的0到1范围限制了表达能力。
2. **优化的内存访问模式**：DCNv3中存在大量内存访问冗余，通过优化内存访问模式，DCNv4 实现了更好的计算效率。

感受野的计算

对于感受野，其最开始的卷积层的感受野即为当前层的卷积核的大小。对于空洞卷积感受野的计算，当其采用HDC设计方法后，可以将原卷积核大小扩展为等效的卷积核大小，其扩展公式如下：

对于经过多层卷积之后的感受野，其计算公式如下：

其中为当前层的感受野，为上一层的感受野，表示之前所有层的步长的乘积(不包括本层)，公式如下:

通常上述公式求取的感受野通常很大,而实际的有效感受野(Effective Receptive Field)往往小于理论感受野，因为输入层中边缘点的使用次数明显比中间点要少，因此做出的贡献不同，所以经过多层的卷积堆叠后,输入层对于特征图点做出的贡献分布呈高斯分布形状。

卷积层的FLOPS计算

不妨设卷积核大小为，特征图通道维为，卷积核数量为，卷积核图像大小为，，则输出的一个像素点的FLOPS为：

这个公式分成了两部分，第一项是乘法运算数，第二项是加法运算数，因为个数相加，要加次。如果考虑偏置bias的话，则会再多一次加法运算，即：

所有的卷积核的FLOPS为：

如果是深度可分离卷积（一部分是分通道卷积，另一部分是1\*1卷积），则其FLOPS为：