

卷积

Conv2d的计算方式

考虑一个2d卷积核 f : `nn.Conv2d(n, m, k)`，不考虑stride、padding、batch以及bias（仅有weight），一次2d卷积计算是：

$$f(h_i, x, y) = \sum_{j=0}^n \left\{ \sum_{s=-l}^l \sum_{t=-l}^l I(c_j, x+s, y+t) K(h_i, c_j, s+l, t+l) \right\} \quad (1)$$

其中 h_i 表示输出的第 i 通道， $l = \lfloor k/2 \rfloor$ ， $f(x, y, h_i)$ 也就表示了输出像素位置为 (x, y) 位于通道 h_i 的值。 $I(c, x, y)$ 表示通道为 c ，位置为 (x, y) 的input。注意kernel（也就是卷积层的weight）是4维的：

维度1	维度2	维度3	维度4
输出维度（本例中为m）	输入维度（本例中为n）	kernel size：本例中为k	kernel size：本例中为k

那么二维卷积就可以理解为（无序列表每一级表示一个维度）：

- 对于输出张量的每一个通道
 - 可以认为，每一个通道的输出是由一个大小为 (n, k, k) 的kernel来计算的
 - 对于一个固定**输出**通道的值
 - 每个**输入**通道内的值先进行相乘（与kernel）-->累加
 - 再将不同**输入**通道的累加值累加

Conv3d

注意，`nn.Conv3d`需要5维输入，以双目匹配经典的3d convolution cost volume regularization为例：输入的cost volume是： (N, C, S, H, W) 。其中， C 是通道数， S 是采样的disparity数，保存从 $0 \sim \max \text{ allowed disparity} - 1$ 采样的所有信息。3d卷积的每一个卷积核将会综合考虑除了图像 (x, y) 坐标以外的辅助维度信息（当然，在点云3d卷积中， S 代表的这一维度不是辅助维度，地位与最后两个维度是相同的）。

故首先，从输入数据上，2d Conv与3d Conv就已经不同了。3d Conv要么是地位同等的三个维度数据（如3d点），要么是在2d数据上有附加辅助维度信息（如disparity）。2d需要**四维**，3d需要**五维**。

四维张量是否可以3d卷积？需要看3d卷积计算的过程。首先已知，3d卷积层的weight是5维：

- （输出通道数，输入通道数，kernel_size_1，kernel_size_2，kernel_size_3）

那么当输出通道数 = 输入通道数 = batch size时，如果将输入的4维tensor进行如下操作：

```

1 a = torch.normal(0, 1, (N, C, H, W))      # 创建一个随机的(N, C, H, W)大小tensor作为输入
2 a = a.unsqueeze(0)                        # 变为(1, N, C, H, W) 五维
3 b = a.unsqueeze(2)                        # 变为(N, C, 1, H, W) 五维
4 conv = nn.Conv3d(n, m, 3, 1, 1)          # 以3*3*3 kernel为例
5 c = conv(a)
6 d = conv(b)

```

是否可以认为此 `conv` 相当于一个二维卷积？个人认为：

- 对于a而言，答案是否定的
- 对于b而言，答案是肯定的

假设kernel size在三个维度上均一致：

$$f(h_i, z, x, y) = \sum_{j=0}^n \left\{ \sum_{r=-l}^l \sum_{s=-l}^l \sum_{t=-l}^l I(c_j, z+r, x+s, y+t) K(h_i, c_j, z+r, s+l, t+l) \right\} \quad (2)$$

对于a而言，此处干了一件非常不好的事，就是把batch中不同的输入信息给混在一起了（由于把batch维度当成channel维度）。而对于b而言，实际上由于b在 (N, C, S, H, W) 中的S维度上只有1，按照k=3时的same padding（也即padding=1），当进行zero pad时，相当于kernel (N, M, K_1, K_2, K_3) 的 K_1 维度只使用了索引为1的值（另外两个维度都与padding的0相乘，彻底没用了），那么这个情况下，3d卷积将会退化为一个2d卷积。但是这样做没有任何意义，相信论文里应该也不会傻到直接使第3维大小为1。