



张萌

—

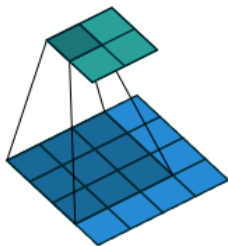
539 人赞同了该回答

一句话解释：逆卷积相对于卷积在神经网络结构的正向和反向传播中做相反的运算。

逆卷积(Deconvolution)比较容易引起误会，转置卷积(Transposed Convolution)是一个更为合适的叫法。

举个栗子：

4x4的输入，卷积Kernel为3x3，没有Padding / Stride，则输出为2x2。



输入矩阵可展开为16维向量，记作 \mathbf{x}

输出矩阵可展开为4维向量，记作 \mathbf{y}

卷积运算可表示为 $\mathbf{y} = \mathbf{C}\mathbf{x}$

不难想象 \mathbf{C} 其实就是如下的稀疏阵：

$$\begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix}$$

平时神经网络中的正向传播就是转换成了如上矩阵运算。

那么当反向传播时又会如何呢？首先我们已经从更深层的网络中得到的 $\frac{\partial \text{Loss}}{\partial \mathbf{y}}$ 。

$$\frac{\partial \text{Loss}}{\partial \mathbf{x}_j} = \sum_i \frac{\partial \text{Loss}}{\partial \mathbf{y}_i} \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_j} = \sum_i \frac{\partial \text{Loss}}{\partial \mathbf{y}_i} C_{i,j} = \frac{\partial \text{Loss}}{\partial \mathbf{y}} \cdot \mathbf{C}_{*,j} = \mathbf{C}_{*,j}^T \frac{\partial \text{Loss}}{\partial \mathbf{y}}$$

Undefined control sequence \hdotsfor

回想第一句话，你猜的没错，所谓逆卷积其实就是正向时左乘 \mathbf{C}^T ，而反向时左乘 $(\mathbf{C}^T)^T$ ，即 \mathbf{C} 的运算。