Transposed Convolution

定义简单理解：转置卷积和卷积对应线代矩阵逆运算，看下面Demo吧！

本章对我来说超级难理解，尤其是padding和stride部分

Demo

卷积操作及转置卷积的定义对于一个输入大小为3×3的图像，卷积核大小为2×2：

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

$$C = \begin{bmatrix} w_{11} & w_{12} & 0 & 0 & w_{21} & w_{22} & 0 & 0 & 0 \\ 0 & w_{11} & w_{12} & 0 & 0 & w_{21} & w_{22} & 0 & 0 \\ 0 & 0 & w_{11} & w_{12} & 0 & 0 & w_{21} & w_{22} & 0 \\ 0 & 0 & 0 & w_{11} & w_{12} & 0 & 0 & w_{21} & w_{22} \end{bmatrix}$$

每一行向量表示在一个位置的卷积操作，0填充表示卷积核未覆盖到的区域。将输入X展开为列向量：$x = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{21} & x_{22} & x_{23} & x_{31} & x_{32} & x_{33} \end{bmatrix}^T$

则卷积操作可以表示为：

$$y = Cx$$

输出向量y的大小为4×1的列向量，改写为矩阵即为2×2。

转置卷积则是将Cx中的输入输出互换：$y = Cx$

$$x = C^T y$$

C^T表示矩阵转置，此时大小为9×4。

即由4×1的输入y，经过转置卷积，

得到输出大小为9×1的列向量x，此时的x, y数值上已经和原来不同，只是在形状上一致。

Traditional Transpose Convolution Code

```
import torch
from torch import nn
from d2l import torch as d2l
def transpose_conv2d(X, K):
    h, w = K.shape
    # 创建一个新的张量Y，其尺寸为输入X的尺寸加上卷积核K的尺寸减去1。在常规卷积中，输出尺寸通常是输入尺寸减去卷积核尺寸加1
    Y = torch.zeros((X.shape[0] + h - 1, X.shape[1] + w - 1)) # 正常的卷积后尺寸为(X.shape[0] - h + 1, X.shape[1] - w + 1)
    for i in range(X.shape[0]):
        for j in range(X.shape[1]):
            # 对于输入X的每一个元素，我们将其与卷积核K进行元素级别的乘法，然后将结果加到输出张量Y的相应位置上
            Y[i:i + h, j:j + w] += X[i, j] * K # 按元素乘法，加回到自己矩阵
    # 返回转置卷积的结果
    return Y
X = torch.tensor([[0.0, 1.0, 2.0], [3.0, 4.0, 5.0], [6.0, 7.0, 8.0]])
K = torch.tensor([[0.0, 1.0], [2.0, 3.0]])
Y = transpose_conv2d(X, K)
print(Y)
```

Pytorch Transpose Convolution Code

```
X, K = X.reshape(1,1,3,3), K.reshape(1,1,2,2)
tconv = nn.ConvTranspose2d(1,1,kernel_size=2,bias=False)
tconv.weight.data = K
print(tconv(X))

tconv_p1 = nn.ConvTranspose2d(1,1,kernel_size=2,padding=1,bias=False)
tconv_p1.weight.data = K
print(tconv_p1(X))

tconv_s2 = nn.ConvTranspose2d(1,1,kernel_size=2,stride=2,bias=False)
tconv_s2.weight.data = K
print(tconv_s2(X))
# stride此处可以理解成在输入每个元素之间补（stride-1）个0，然后按照stride=1的正常卷积进行计算
```