Exercise 1 - Transposed Convolution Output Sizes

What is the size of the output for a input tensor and a transposed convolutional layer if the parameters are given as follows (assume the number of channels is given in the first dimension).

(a) Input tensor size: $3 \times 2 \times 2$

Transposed convolution: 3×3 kernel, stride 1, output channels: 2

(b) Input tensor size: $3 \times 5 \times 5$

Transposed convolutional: 2×2 kernel, stride 2, output channels: 4

(c) Input tensor size: $c_{in} \times h \times w$

Transposed convolutional: 3×3 kernel, stride 2, output channels: 2

(d) Input tensor size: $c_{in} \times h \times w$

Transposed convolutional: $h_k \times w_k$ kernel, stride s, output channels: c_{out}

Exercise 2 - Transposed Convolution Parameter Sizes

What is the number of learnable parameters for each of the following transposed convolution layers defined in PyTorch. Try to calculate those by hand first and use pytorch later to verify your results.

(a) `nn.ConvTranspose2d(in_channels=3, out_channels=2, kernel_size=3, stride=1)`

(b) `nn.ConvTranspose2d(in_channels=3, out_channels=10, kernel_size=3, stride=2)`

(c) `nn.ConvTranspose2d(in_channels=3, out_channels=2, kernel_size=4, stride=5)`

(d) `nn.ConvTranspose2d(in_channels=3, out_channels=4, kernel_size=3, stride=23)`

Exercise 3 - Transposed Convolution by Hand

You are given an input matrix X (consisting of a single channel) and a kernel K as follows:

$$X = \begin{pmatrix} 1 & 0 & 2 \\ 2 & 3 & 0 \\ -1 & 0 & 3 \end{pmatrix}, \quad K = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$$

(a) Compute the transposed convolution by hand assuming stride 2.

(b) Compute the transposed convolution by hand assuming stride 1.

(c) Use PyTorch to verify your answers.

(d) Implement the transposed convolution in PyTorch without using its own implementaiton. You can assume no bias term a square kernel and no separate batch dimension. I.e. your task is to implement the following function

```python
def conv_transpose2d(inp, weight, stride=1):
    # inp - input of shape (C_in, H, W)
    # weight - kernel of shape (C_in, C_out, K, K)
    # stride - stride of the transposed convolution
    # RETURNS
    # output - output of shape (C_out, H_out, W_out)
    #
    # YOUR CODE HERE
    return output
```