

# Transposed convolution

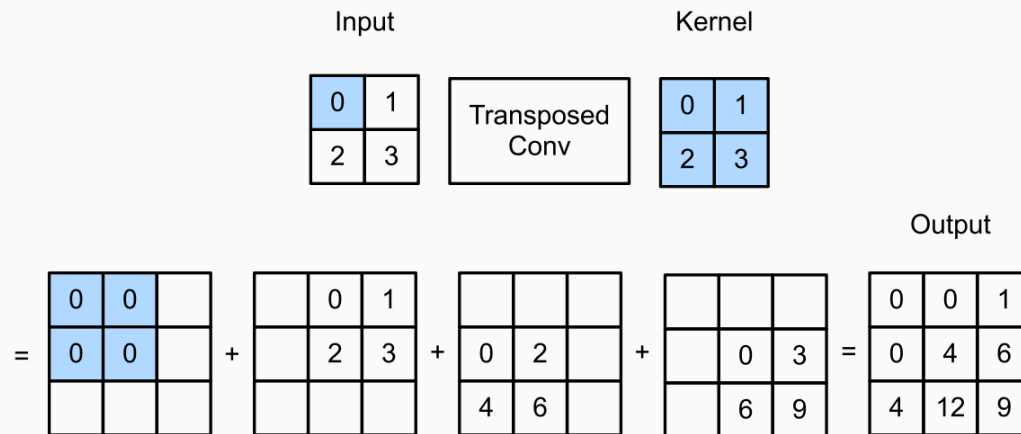


Fig. 13.10.1 Transposed convolution with a  $2 \times 2$  kernel. The shaded portions are a portion of an intermediate tensor as well as the input and kernel tensor elements used for the computation.

We can implement this basic transposed convolution operation `trans_conv` for a input matrix `X` and a kernel matrix `K`.

MXNET

PYTORCH

```
def trans_conv(X, K):  
    h, w = K.shape  
    Y = np.zeros((X.shape[0] + h - 1, X.shape[1] + w - 1))  
    for i in range(X.shape[0]):  
        for j in range(X.shape[1]):  
            Y[i:i + h, j:j + w] += X[i, j] * K  
    return Y
```

# Drawback of Transposed Convolutions

<https://distill.pub/2016/deconv-checkerboard/>

Explains why

