**Transposed Convolutional Neural Networks — How to Increase the Resolution of Your Image**

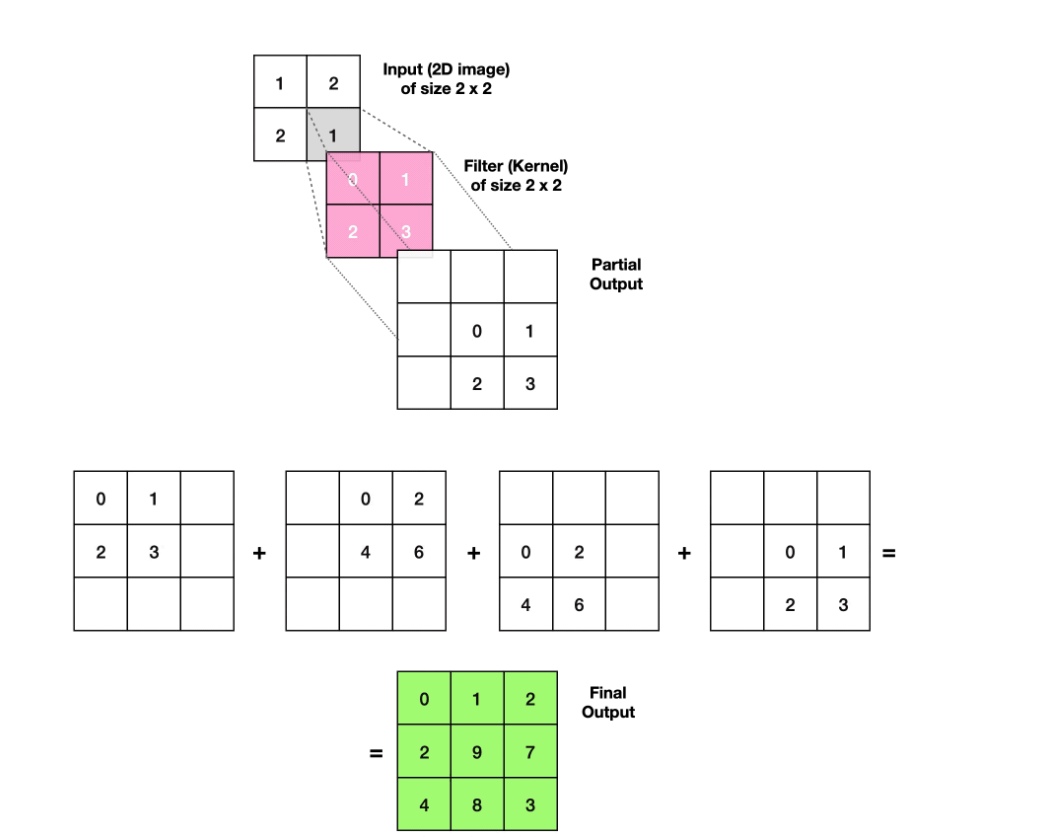
*Note, in some literature, Transposed Convolutions are also referred to as Deconvolutions or Franctionally Strided Convolutions.*

The goal of a Transposed Convolution is to do the opposite of a regular Convolution, i.e., **to upsample the input feature map to a desired larger size output feature map**.

To achieve this, Transposed Convolution goes through an iterative process of multiplying entries in the input feature map by the filter and adding them up together. Note that we also move along by the specified number of places (**stride**) within each step.

The below gif illustrates how the Transposed Convolution works. The example moves from a 2x2 input to a 3x3 output via a 2x2 filter using a stride of 1.

* Take the first input entry and multiply it by the filter matrix. Temporarily store the result.
* Then, take the second input entry and multiply it by the filter matrix. Temporarily store the result. Continue this process for the rest of the input matrix.
* Finally, sum up all the partial outputs to get the final result.

****

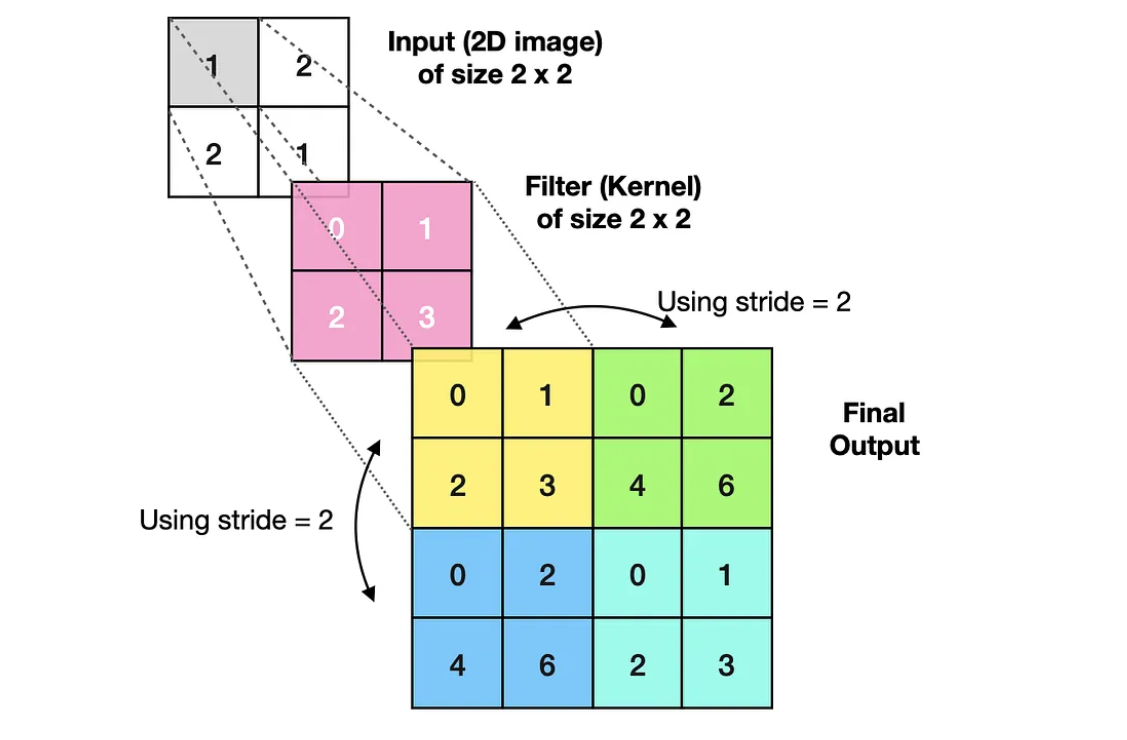
**For example the center cell has the output of 3 +4 +2 = 9.**

**We create multiple features maps 3 \* 3 coming from a 2d 2\*2 image (length + height).**

It is worth nothing that we are essentially generating additional data during the transposed convolution operation as we are upsampling the feature map from a smaller to a larger size.

However, this operation is not exactly the reverse of a Convolution. It is because some information is always lost during a Convolution, meaning that we can never precisely recreate the same data by applying a Transposed Convolution.

Lastly, we can experiment with a filter size or stride to achieve the desired size of an output feature map. E.g., we could increase the stride from 1 to 2 to avoid section overlaps and produce a 4x4 output (see image below):

****

## **What are Transposed Convolutions used for?**

Transposed Convolutions are crucial for **Semantic Segmentation** and data generation in the **Generative Adversarial Networks (GANs)**. One of the more straightforward examples would be a Neural Network trained to increase image resolution. We will build one such network right now.