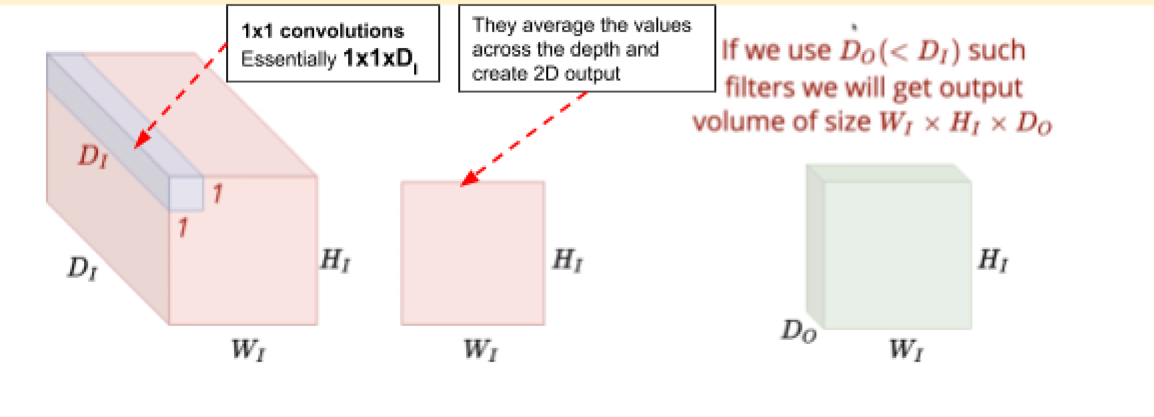
PadhAI: CNN Architectures

One Fourth Labs

1x1 Convolutions

What is a 1x1 convolution used for?

1. We've mostly worked with 3x3 convolutions so far, i.e. a grid containing 3 rows, 3 columns with 9 cells in total.
2. The result of a single 3x3 operation is the weighted average of all the points in the grid, applied to our selected pixel.
3. Now, let us look at a 1x1 convolution

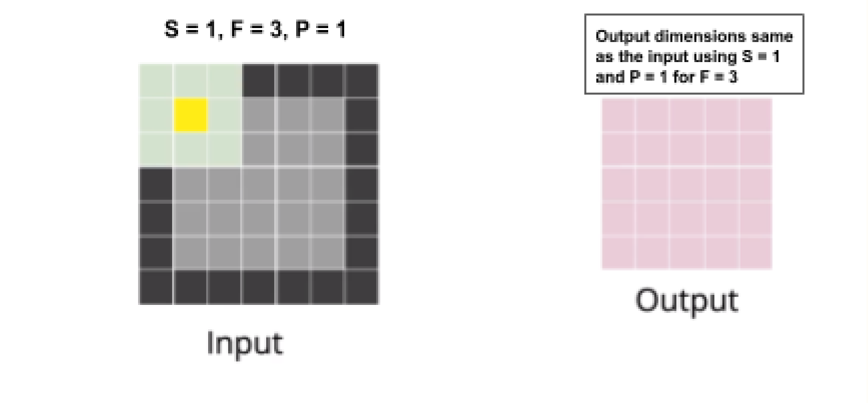


1x1 convolutions Essentially 1x1xD_I

They average the values across the depth and create 2D output

If we use $D_O (< D_I)$ such filters we will get output volume of size $W_I \times H_I \times D_O$

4. A 1x1 kernel takes a neighborhood of 1 row and 1 column, which is essentially the pixel itself. Since the kernel is of size 1x1xD_I, it computes the weighted average of all the pixels across the input depth D_I
5. Here, the Input is 3D, the filter is 3D but the operation is 2D, as we are only moving horizontally and vertically. The 3D volume is compressed to a 2D area.
6. If we were to use D_O number of filters, where ($D_O < D_I$), we will get an output of $W_I \times H_I \times D_O$. Each of the 1x1 kernels will give one 2D output and D_O such kernels gives us an output volume of the dimensions $W_I \times H_I \times D_O$.
7. If D_O is much smaller than D_I, we effectively shrink the input volume while still effectively retaining the depth information (Due to averaging across depth).
8. Now, this output behaves as an input to the next layer, resulting in a much smaller number of computations due to smaller depth.
9. In a nutshell, 1x1 filters are used to compress input volumes across their depth to get a smaller output volume of same Width and Height.
10. Another operation we need to look at is Max Pooling. We usually perform Max-pooling with a Stride=2, resulting in halving the input dimensions. However, we can also perform it with a stride of 1. With S = 1 and appropriate padding, we can preserve input dimensions.



Output dimensions same as the input using S = 1 and P = 1 for F = 3