CNN BASICS

-Omisa Jinsi

What have we done so far?

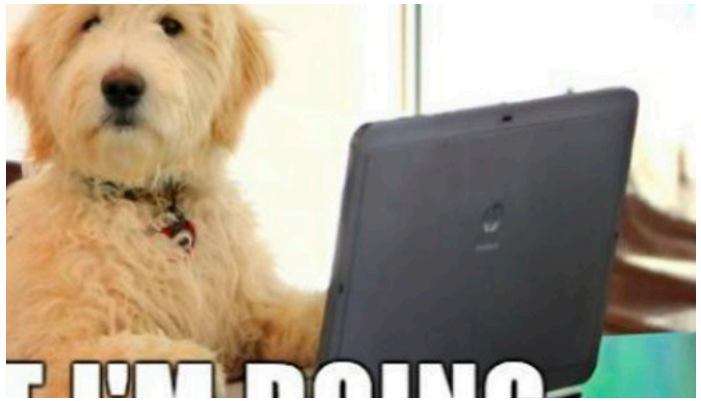
- A perceptron construct
- Design a simple MLP structure
- Think how patterns can be detected
- Backpropagation
- Learn how weights and biases are updated

TOPICS

- 1. Motivation – pitfalls of simple MLP?
- 2. Scanning MLPs
- 3. What is CNN?
- 4. What is filter, channel, stride, and the process of convolution?
- 5. Forward function of CNN, how does the filter convolve? Output formula
- 6. Downsampling techniques: Pooling – Max, Min, Average
- 7. Introduction to Backpropagation

Motivation:

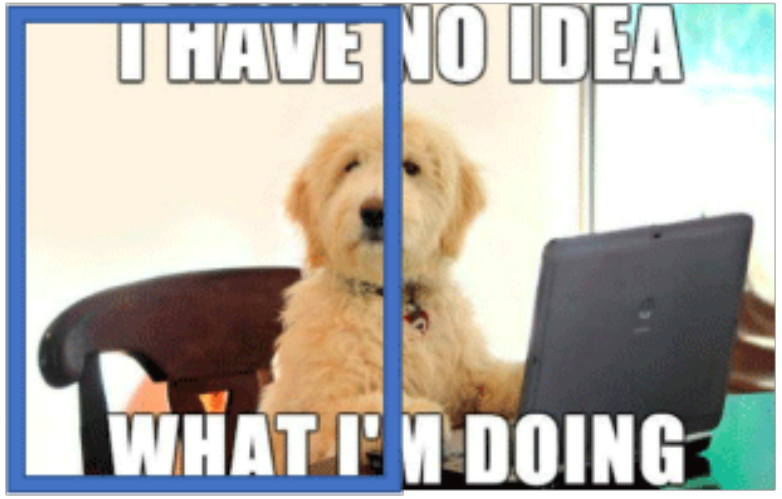
- Your network learns to detect a pattern at a certain region/subspace
- Suppose at the 5th second of an utterance you hear the word – "HarryPotter". What happens if the word appears elsewhere?
- Same thing with memes, we want to detect the dog regardless of where it is placed.



Basically the location of pattern is not important, but detecting the pattern itself is important! – **SHIFT INVARIANCE**

Scanning MLPs:

- How do you detect patterns in a shift invariant format?
- Build a perceptron, scan the input area. Wait SCAN? – Yes!
- 1-D scan for sound; vs 2-D scan for images
- Scan across the entire time if it is an audio clip or utterance
- Or along length and breadth of an image



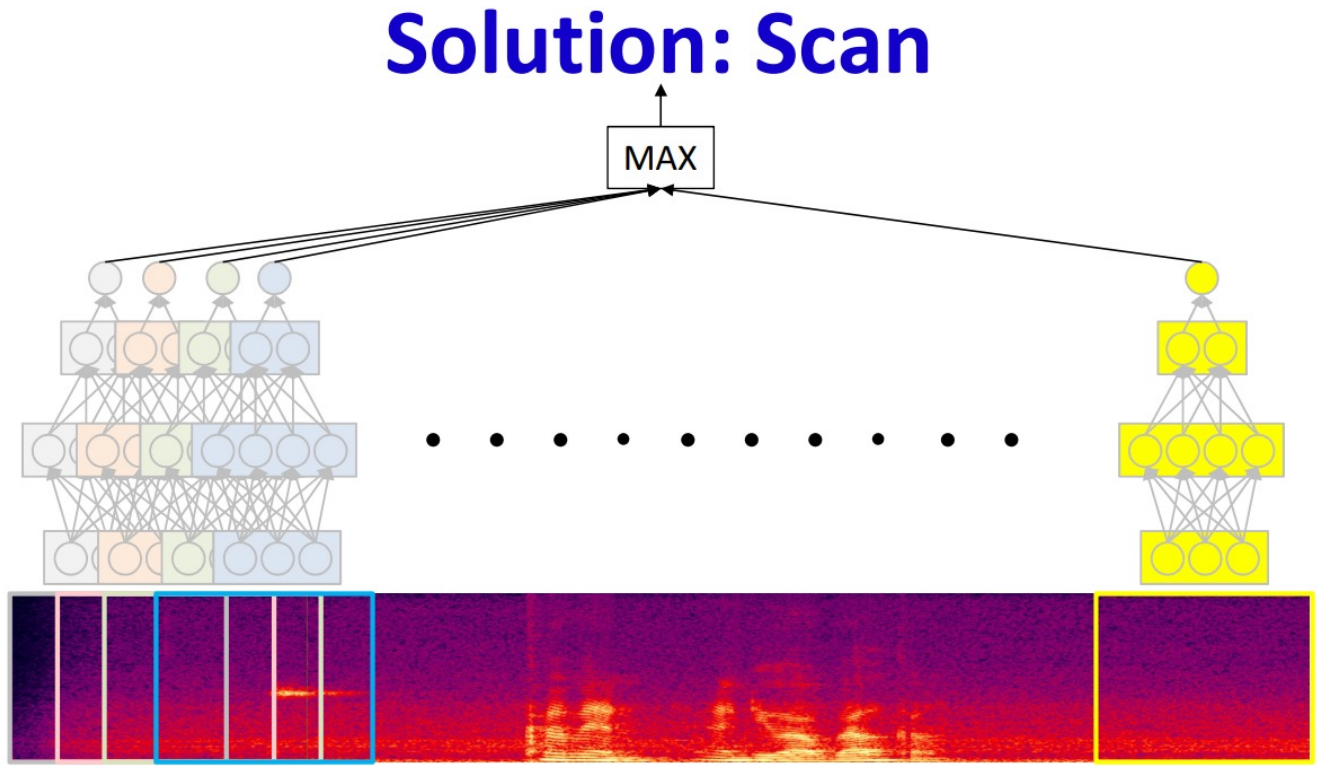
Special thanks: Mansi A. for helping me @12am last night with her free PS to refine this gif

What is CNN? Scanning MLPs

Think of every "window" of the input being scanned by a single MLP and to detect a pattern.

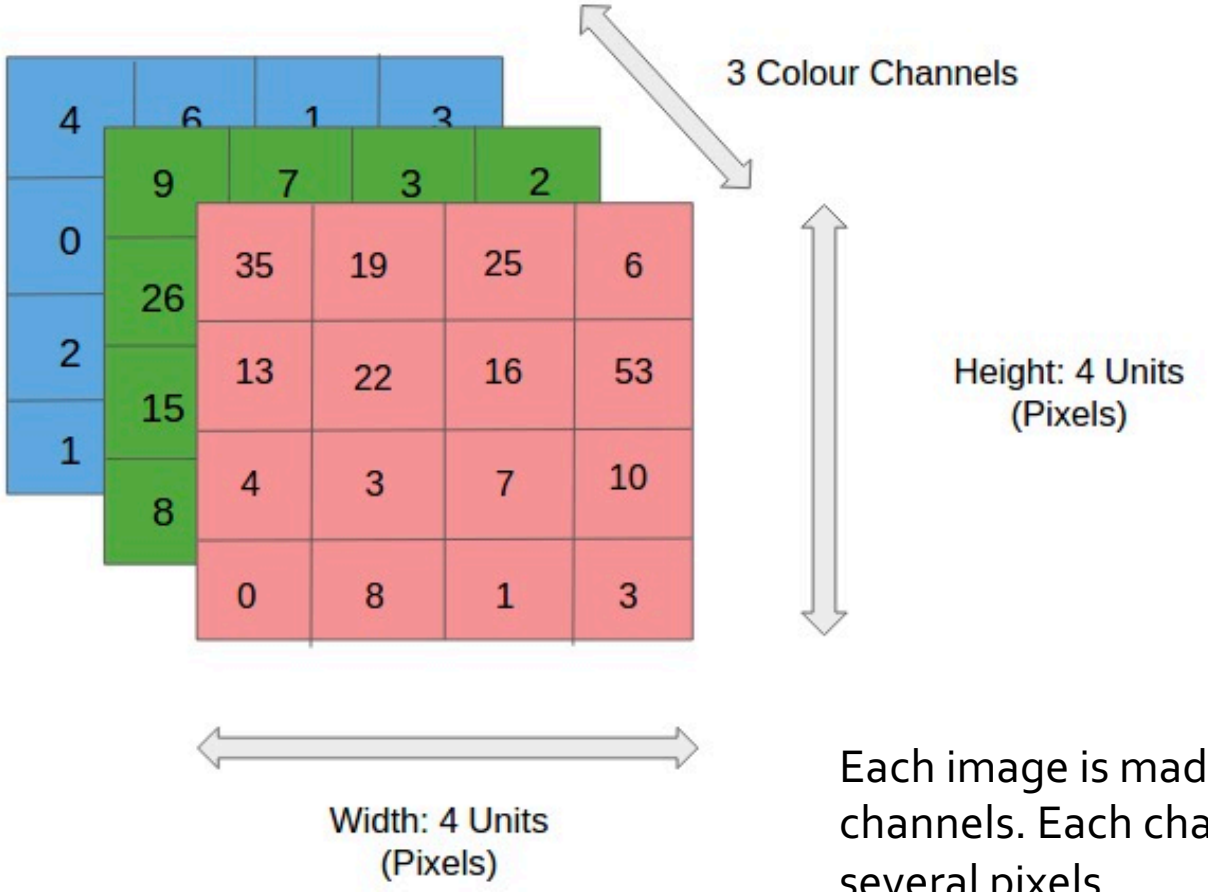
Ex: we want to detect which window of the input has the audio "Harry Potter"?

Now each MLP has weights that will create an output in the forward function, and since our pattern will have specific input values (like frequencies), we want to do a max() over all MLPs and see which one was able to detect it.



Courtesy: Dr. Bhiksha's slides

Images – what are these?



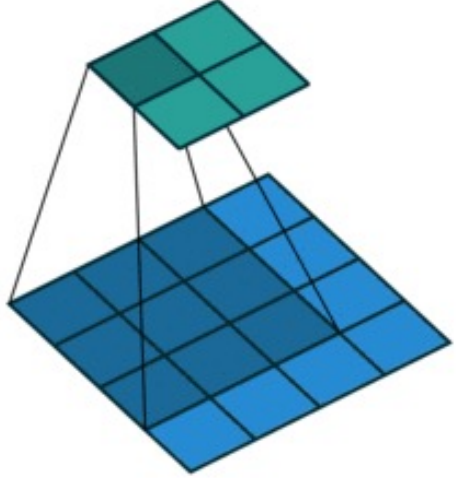
Each image is made up of a set of channels. Each channel comprises of several pixels

3 for a colored image, 2 for B&W.

The number of channels you encounter could even increase!

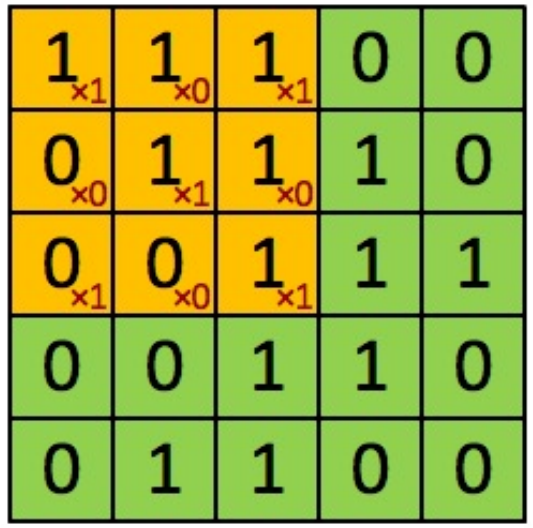
Image Source: Medium blogpost: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way

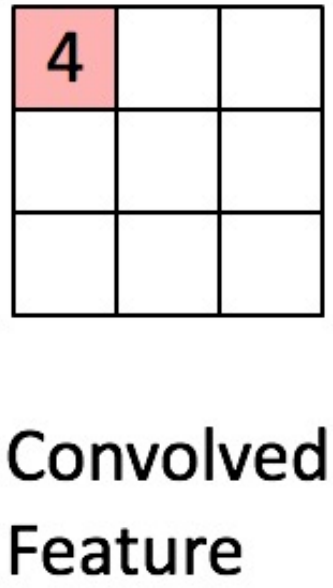
Basic Definition:



- **Filter/Kernel:** Basically a collection of weights that can be applied to an image so that the area where a pattern is located gets more emphasis. How? (Wait for next slide)
- **Stride:** The amount by which the filter/kernel moves. If stride is 1, the filter will move by 1 pixel dimension.
- **Input Channel:** The number of channels in the input
- **Output channels:** Number of filters applied
- **Padding:** What if there is a mismatch between kernel size and allowable dimension of input (width and height), the kernel cannot "fall-off". We need to accommodate the kernel and hence we pad the input.
- **Output size:** $[(\frac{Width\ or\ Height - Kernel\ size + 2 * Padding}{Stride})] + 1.$

Convolution in the forward function:





Source: TDS blog post - A comprehensive guide to convolutional neural network the eli5way

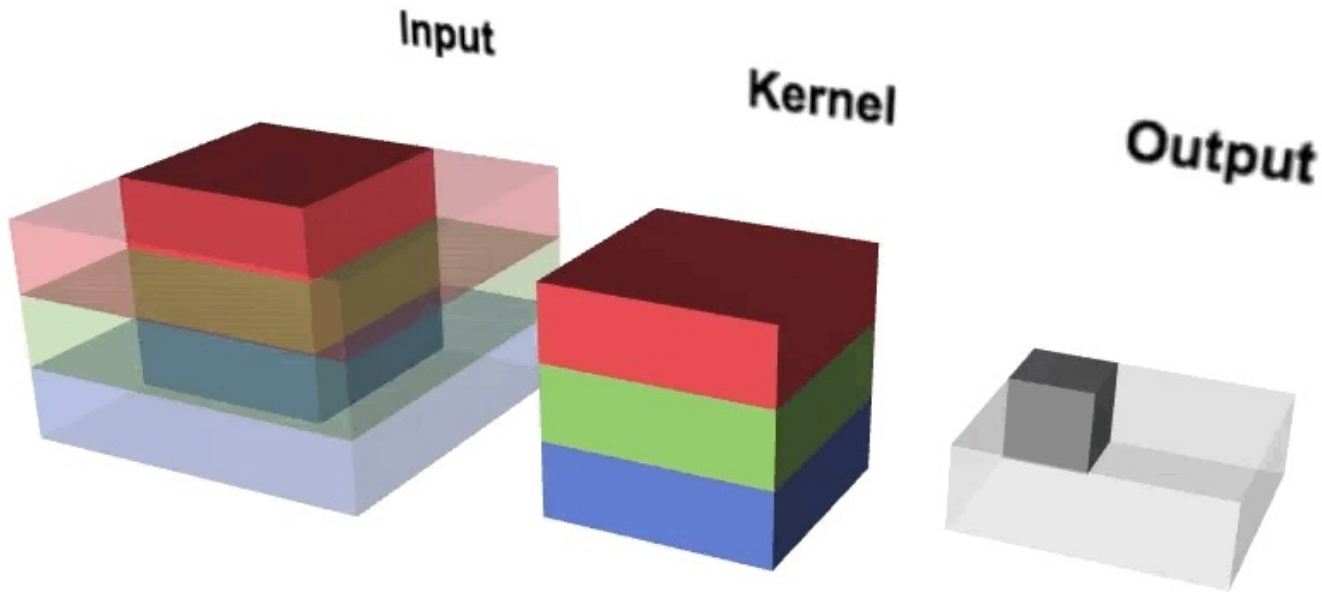
What is happening here? As we apply the filter of $\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}$

We 'convolve' in our forward function and get a corresponding output value from the area convolved over. That is:

$(1 * 1) + (1 * 0) + (1 * 1) + (0 * 0) + (1 * 1) + (1 * 0) + (0 * 1) + (0 * 0) + (1 * 1) = 4$ which is the first value of the output at (o,o)

Visualizing 2-D convolution with multiple filters in 3-D

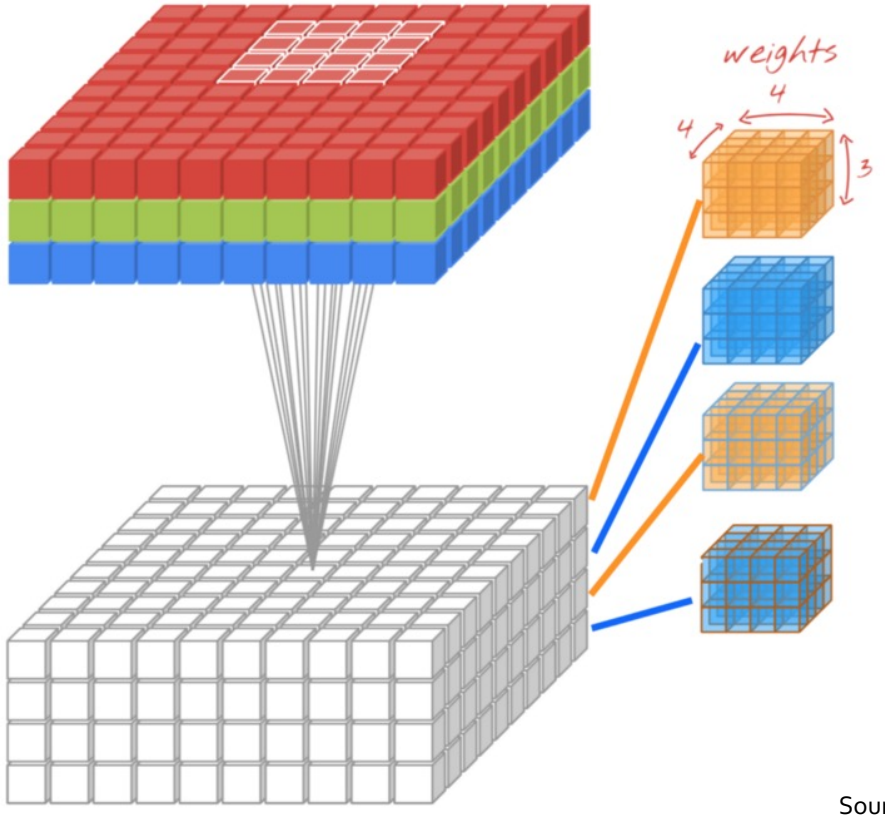
Now visualize the same 2-D convolution process in 3-D.



Here kernel has same number of channels as input! - the red weights (kernel) is applied to the red channel and its pixel values.

Source: EECS251Leture UCBerkeley

Convolution in 3-d with multiple kernels



More filters, more output channels.

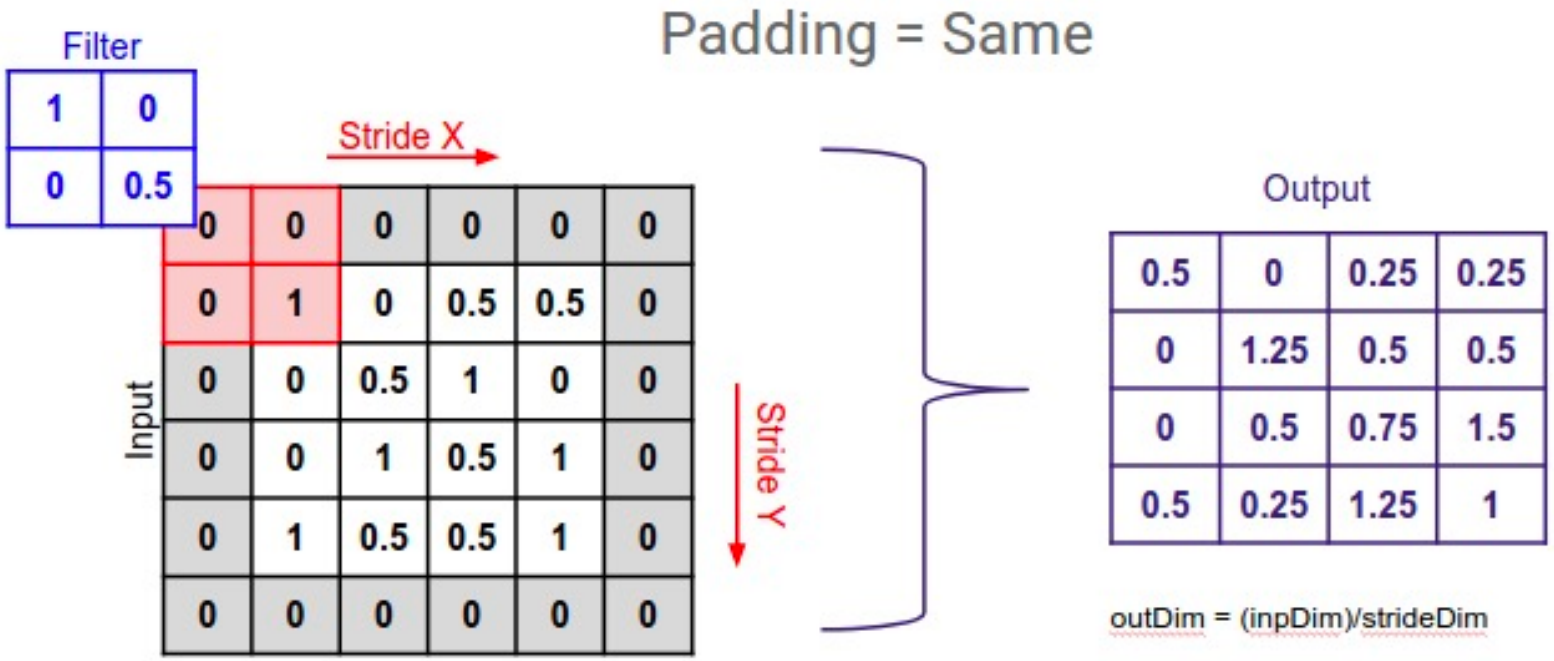
Source: Towards Data Science blogpost on a beginners guide to CNN

Convolution in 3d is just like 2d but you can imagine you are doing the 2d work 4 times, that is because each kernel can be considered a different set of weights which are applied to input channels, and we obtain 4 output channels.
because # of kernels = output channels

Padding:

Why padding?

Because a greater stride could lead to filter "falling off" the input dimension. And stride is our hyper-paramter!



Source: Medium blogpost - What is Padding in Convolutional Neural Network's(CNN's) padding

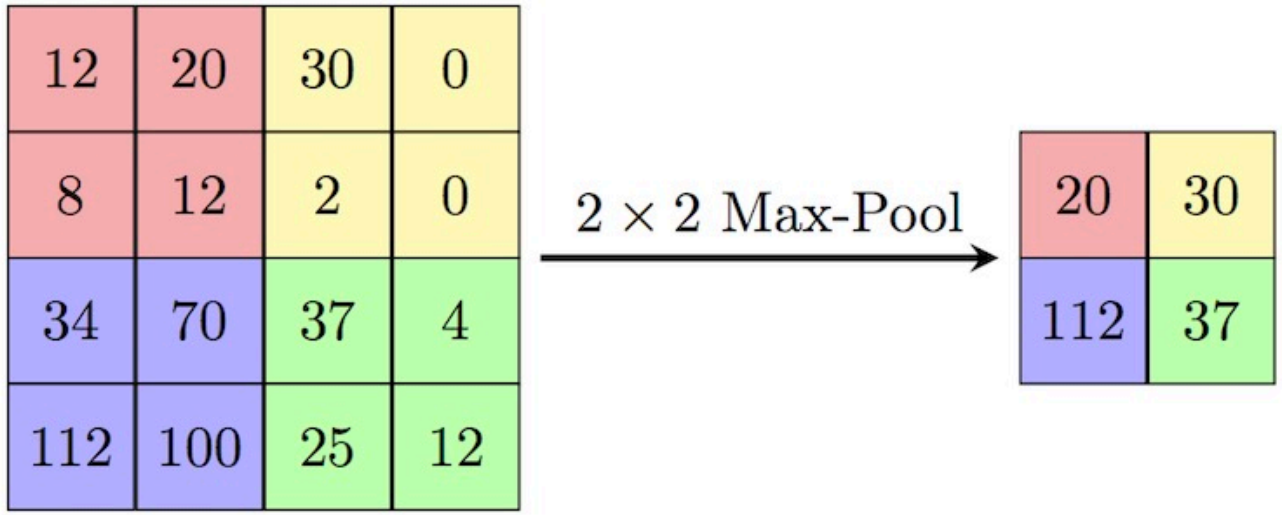
Pooling

Why pooling?: It reduces the spatial size of the output by reducing the number of parameters

Pooling layer operates on each feature map independently.

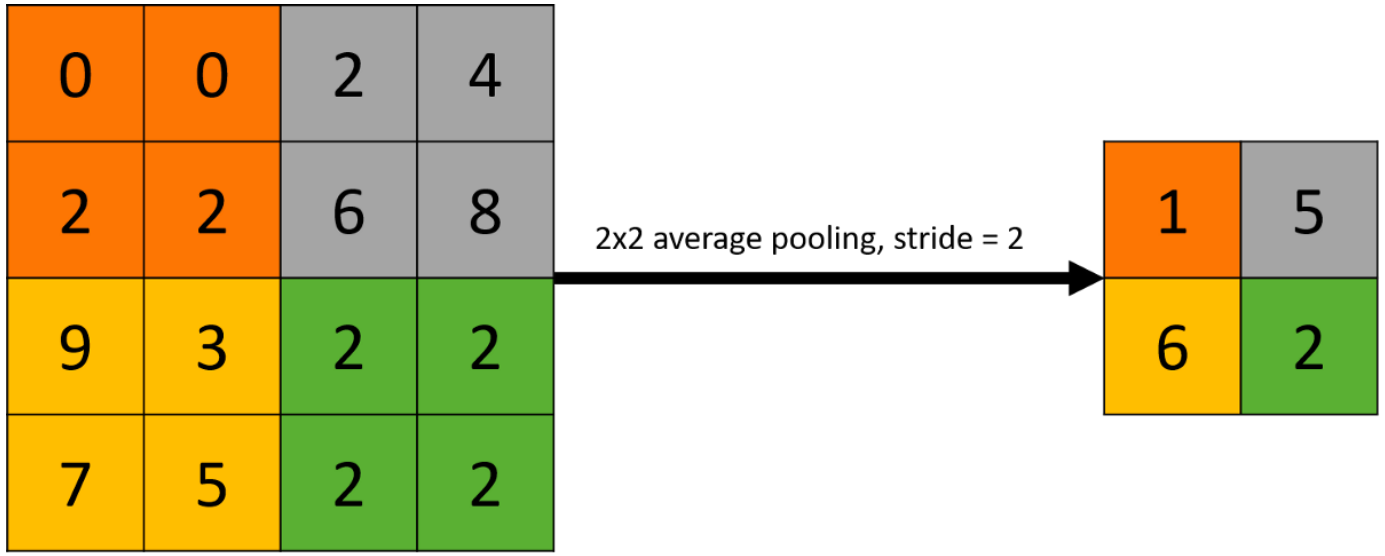
Three main types of pooling:

1. Max pooling
2. Min Pooling
3. Average Pooling



Source: https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

Pooling



Source: Avg pooling: Kaggle - https://www.kaggle.com/questions-and-answers/59502

Basically calculate the average of the output values over stride of 2

Hence:

Output 1 = $\frac{0+0+2+2}{4} = 1$

Output 2 = $\frac{2+4+6+8}{4} = 5$