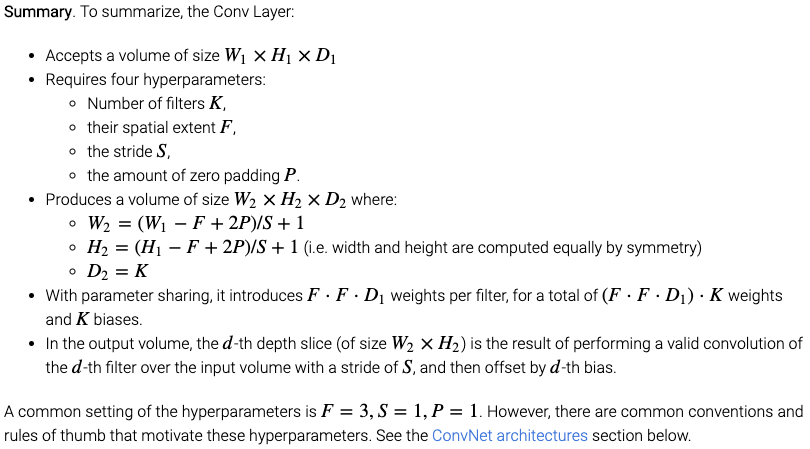
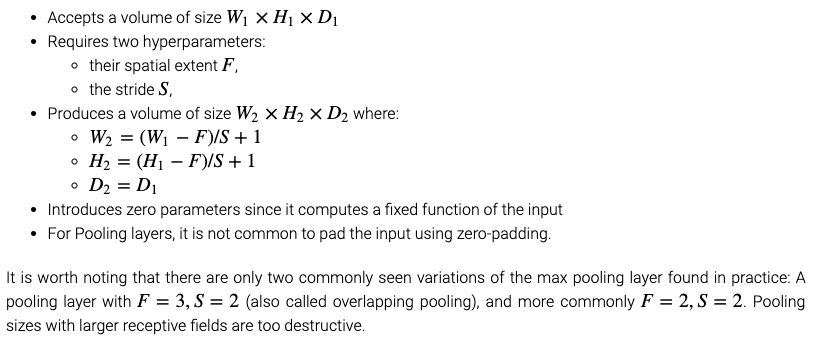
<http://cs231n.github.io/convolutional-networks/>

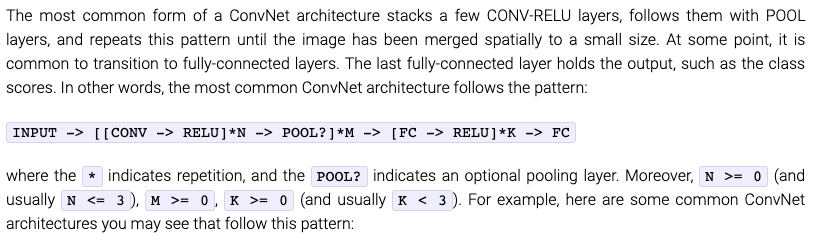
* fc neural network does not scale easy to larger input
* convnets input images in width, height, depth (channels)
* three types of layer: convolutional, pooling, fc
* filters are small but convolve over entire input so output (activation map) stays same
* The connections are local in space (along width and height), but always full along the entire depth of the input volume
* Output depth is a hyper parameter, sets the number of filters. Neurons looking at the same region belong to the same depth column.
* Stride larger than 1 will shrink the output volume spatially
* Zero-padding is used to counteract shrinking the output volume spatially

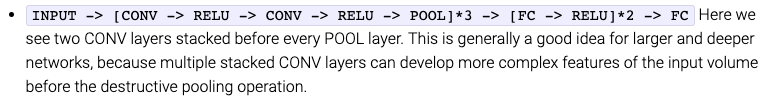


The pooling layer:



ik denk dat we een combinatie van layers moeten bedenken zodat we de input van 784 kunnen schalen naar 24 outputs. Aan de hand van filter groottes en pooling layers.





Intuitively, stacking CONV layers with tiny filters as opposed to having one CONV layer with big filters allows us to express more powerful features of the input, and with fewer parameters. As a practical disadvantage, we might need more memory to hold all the intermediate CONV layer results if we plan to do backpropagation.

Instead of rolling your own architecture for a problem, you should look at whatever architecture currently works best on ImageNet, download a pretrained model and finetune it on your data. You should rarely ever have to train a ConvNet from scratch or design one from scratch.

The input layer (that contains the image) should be divisible by 2 many times. Dus misschien goed idee om onze plaatjes te zero padden?? Aangezien we 28x28 hebben.

The conv layers should be using small filters (e.g. 3x3 or at most 5x5), using a stride of S=1, and crucially, padding the input volume with zeros in such way that the conv layer does not alter the spatial dimensions of the input.

The pool layers are in charge of down sampling the spatial dimensions of the input. The most common setting is to use max-pooling with 2x2 receptive fields (i.e. F=2), and with a stride of 2 (i.e. S=2).

Smaller strides work better in practice. Additionally, as already mentioned stride 1 allows us to leave all spatial down-sampling to the POOL layers, with the CONV layers only transforming the input volume depth-wise.

If the CONV layers were to not zero-pad the inputs and only perform valid convolutions, then the size of the volumes would reduce by a small amount after each CONV, and the information at the borders would be “washed away” too quickly.