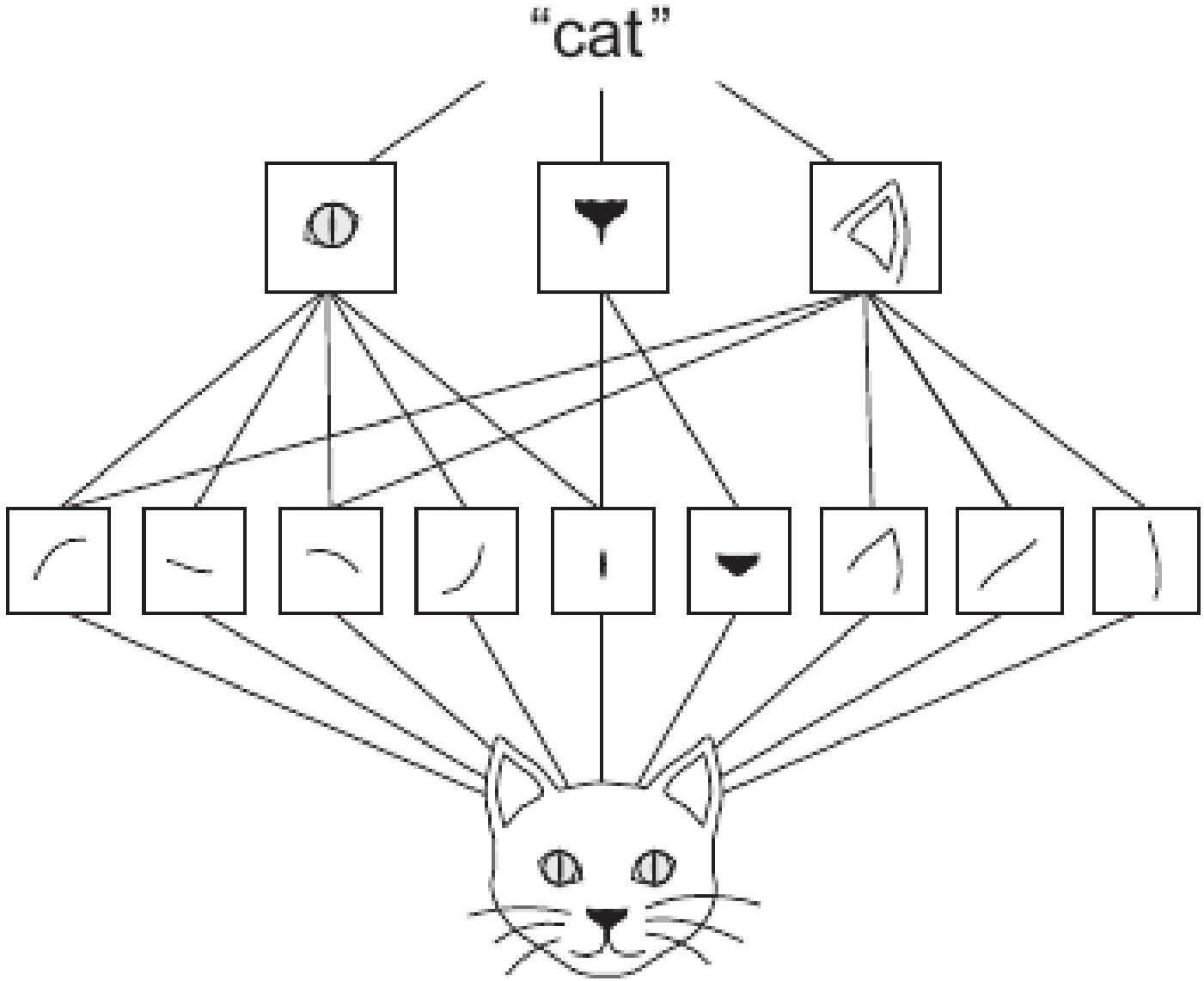
Convolutional neural network

CNN - Hierachies of patterns



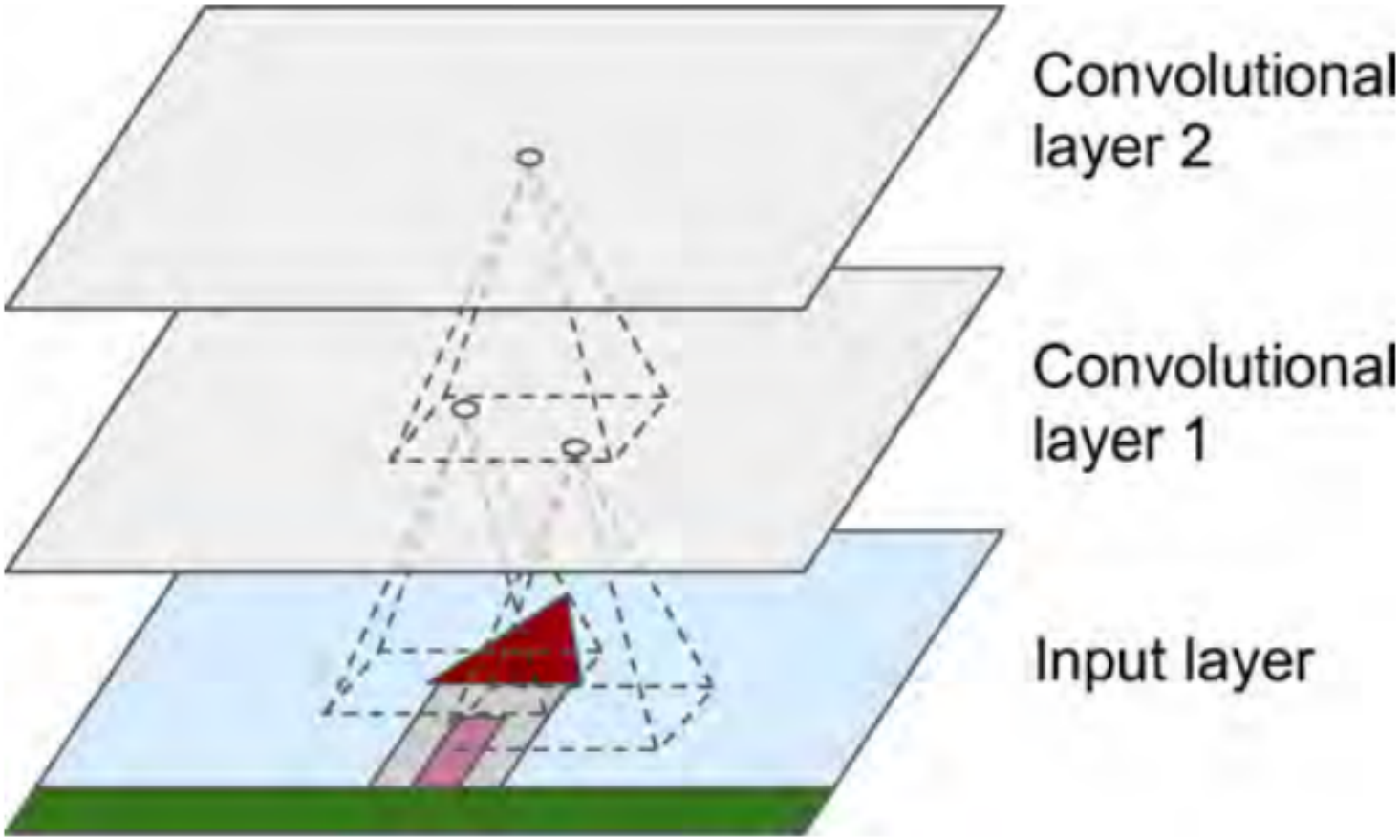
1

[1] Chollet. Deep Learning with python

CNN

- Receptive fields
  - First convolutional layer: neurons are not connected to every single pixel in the input image, but only pixels in thier receptive fields
  - Second convolutional layer: each neuron is connected only to neurons located within a small rectangle
- the networks concentrates on
  - low-level features in the first hidden layer, then
  - assemble them into higher-level features in the next hidden layer,
  - and so on.

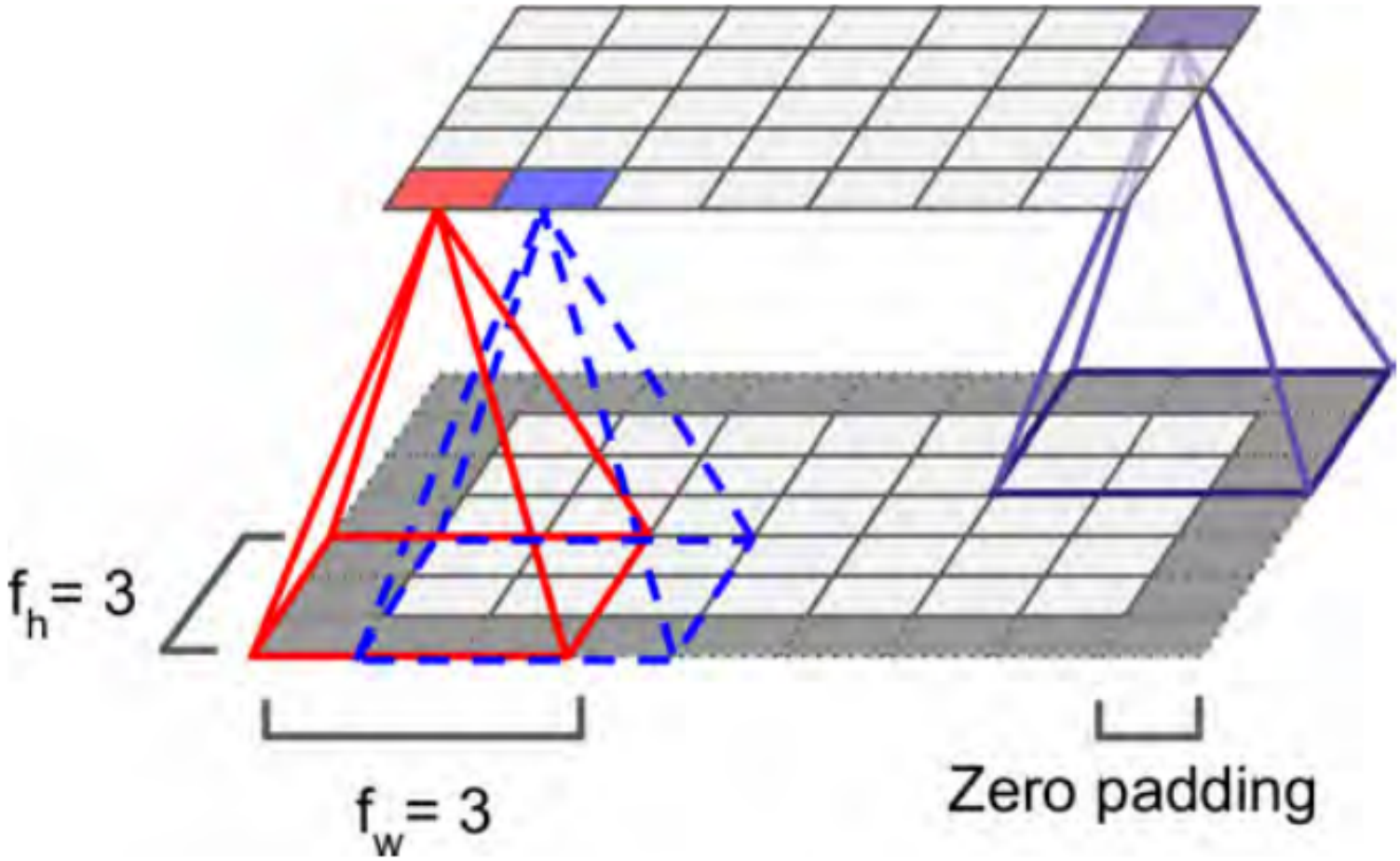
CNN layers with rectangular local receptive fields



2

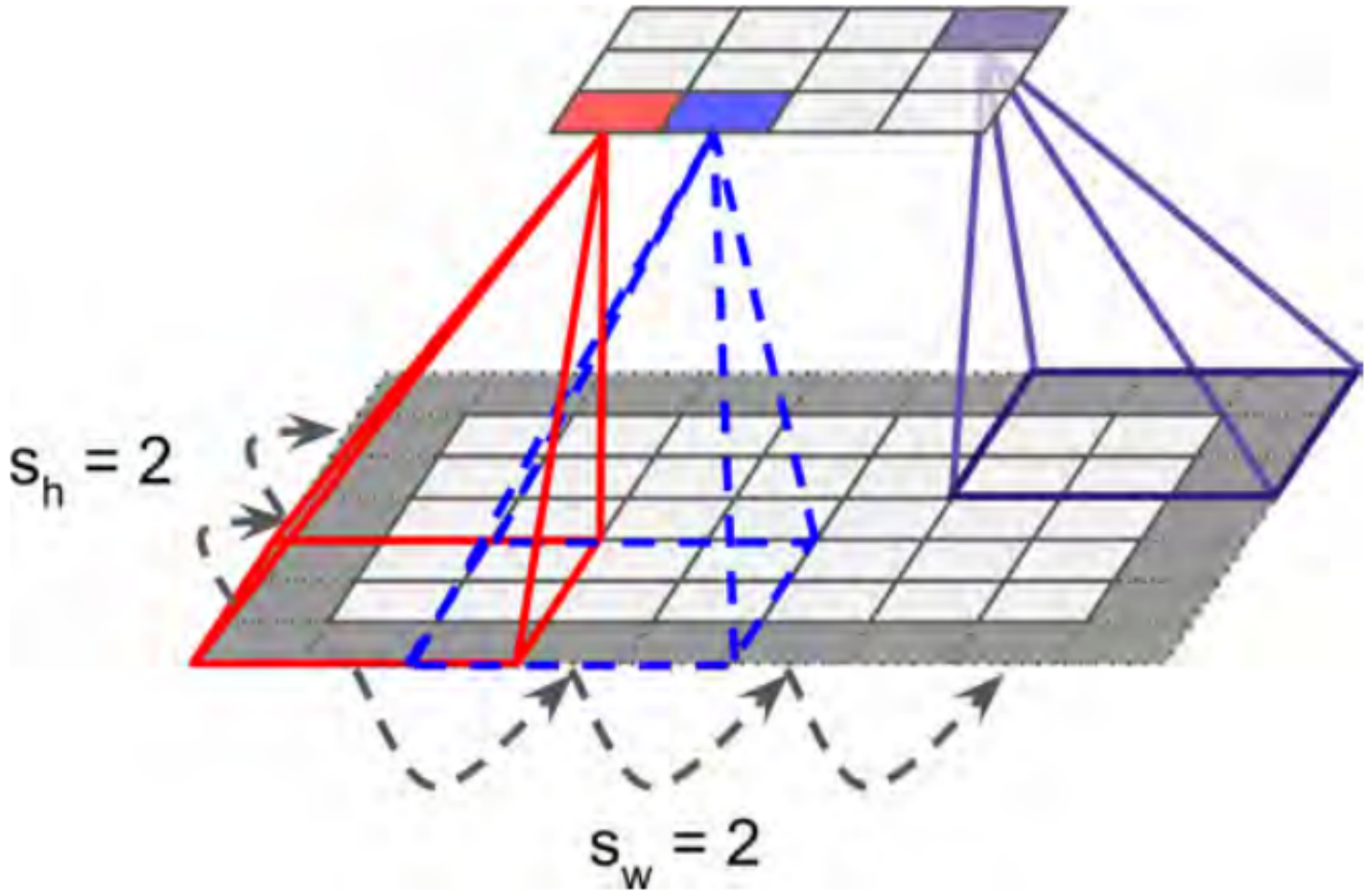
[2] Pictures taken from Geron. Hands-on Machine Learning

Connections between layers (with zero padding)



- neuron i, j - connected to neurons in previous layer in rows i to $i + f_h - 1$, columns j to $j + f_w - 1$.
- the layer has the same height and width as the previous layer (with zero padding)

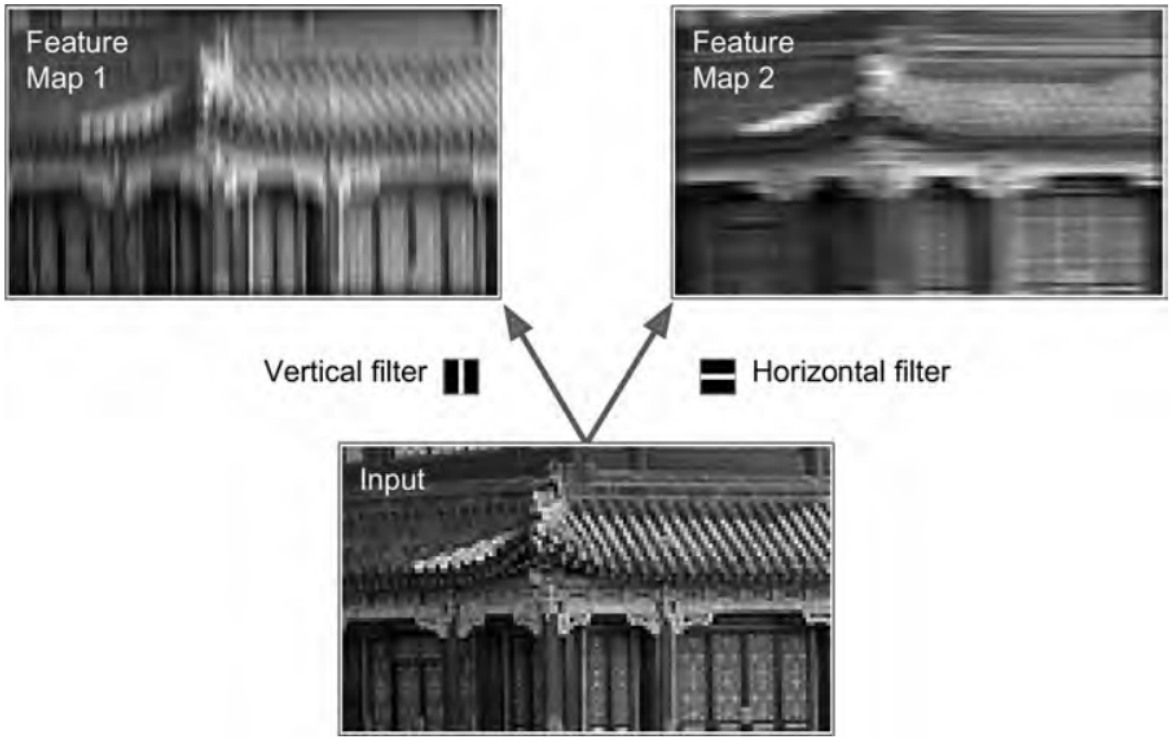
Reduce dimensionality using a stride



Input layer : $5x7$. Filter: $3x3$. Stride: 2.Result: $3x4$

Filters

- a neuron's weight can be seen as a small image the size of the receptive field
- a layer full of neurons using the same filter gives a feature map = highlights the areas in an image that are most similar to the filter



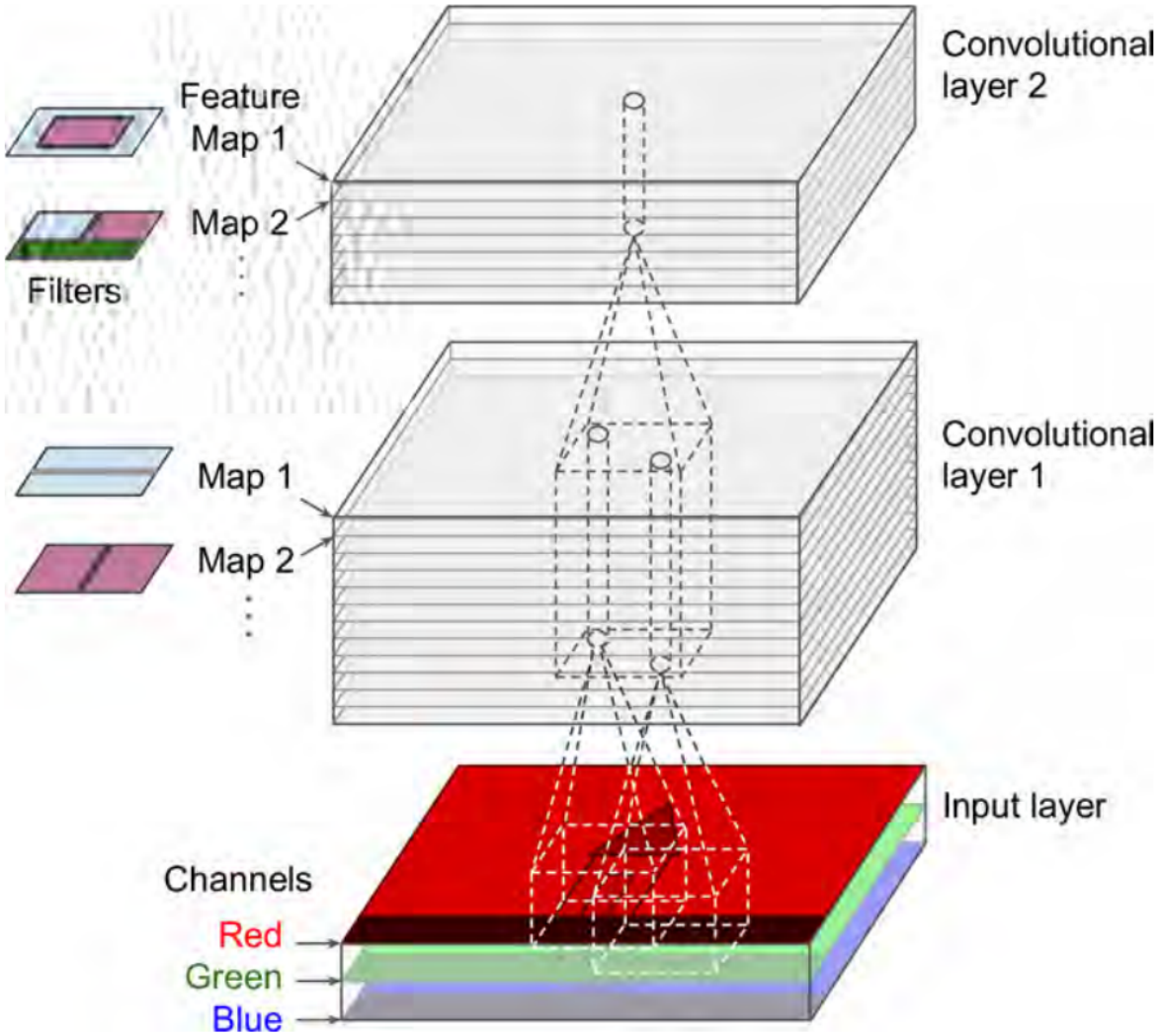
Time for notebook 0

Stacking multiple feature maps

Obs: All the previous representations were simplifications

- a convolutional layer is composed of several feature maps of equal sizes
- within one feature map, all neurons share the same parameters
- different feature maps may have different parameters
- a neuron's receptive field includes a limited number of neurons but across all the feature maps of the previous layer

Convolution layers with multiple feature maps and image with three channels



cont. Feature maps

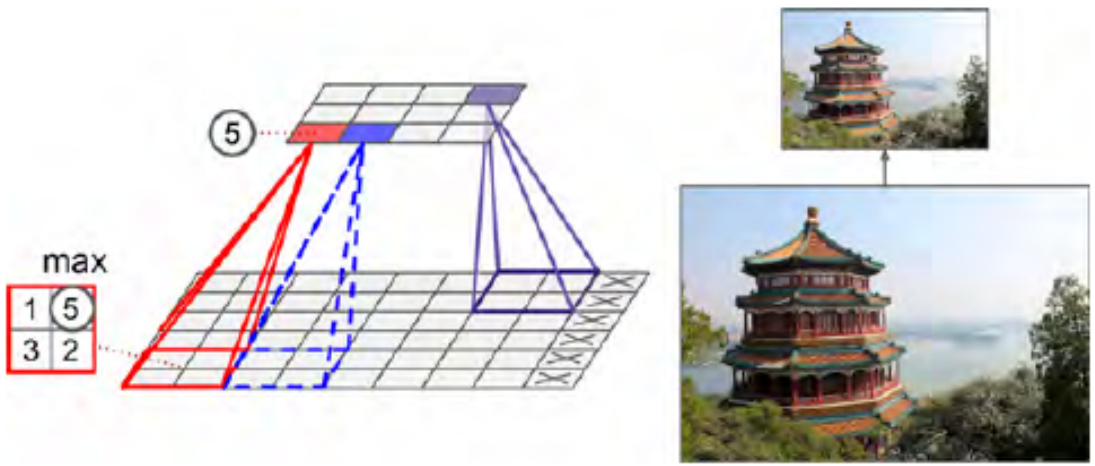
A neuron located

- in row i, column j of the feature map k is connected
- to the outputs of the neurons in the previous layer located in rows ixs_w to $ixs_w + f_w - 1$, columns jxs_h to $jxs_h + f_h - 1$ across all feature maps

All neurons located in teh same row and column but in different feature maps are connected to the outputs of the exact same neurons in the previous layer

Pooling layer: max, mean

- Subsample the input image - reduce number of parameteres - reduce overfitting
- Required info: size, stride, padding type
- Has no weights
- Worls on every input channel independently: output depth is the same as input depth



max pooling, 2x2, stride 2, no padding

Time for notebook

- Fashion mnist with only dense layers (fully connected)
- Fashion mnist with convolutional layers
  - CNN
  - number of parameters (model.summary)
  - improvement over fully connected

Why not simply use fully connected layers

- #parameters for dense layer: input image 100x100. First dense layer 1000 neurons (Obs: it restricts the amount of information transmitted to the next layer) $\rightarrow 10^7$ parameters for only one layer!!!
- CNN: the learnt patterns are translation invariants: a certain pattern can be recognized anywhere
- CNN can learn hierarchies of patterns

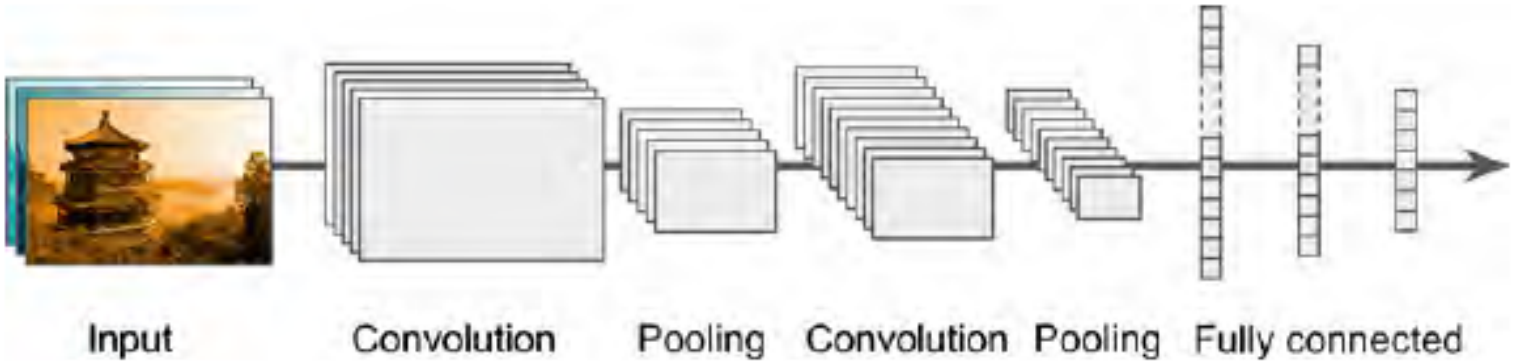
Frame Title

A. Geron: A common mistake is to use convolution kernels that are too large. You can often get the same effect as a $9x9$ kernel by stacking two $3x3$ kernels on top of each other, for a lot less compute.

Time for notebook

nice visualisation

Typical CNN architecture



Pooling effect on FMNIST

Transfer knowledge (ImageNet) - cifar10

Data augmentation - dogs/cats

Covid radiology 2 1