TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

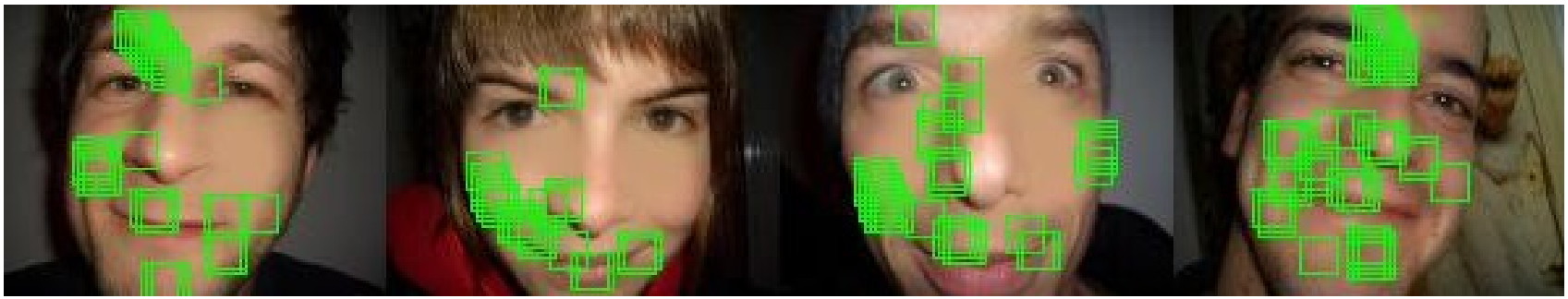
5.2 Convolutional Neural Networks

Machine Learning 1: Foundations

Marius Kloft *(TUK)*

Example: Image Processing

- For many types of images it makes sense to process portions of the image (**patches**)
  - This is because content could be in any position
- Typically one wants to apply filters to the patches to detect "interesting" properties of the images



source: `http://pages.cs.wisc.edu/~andrzeje`

Convolution Filter

Many image filters are based on **convolutions**...

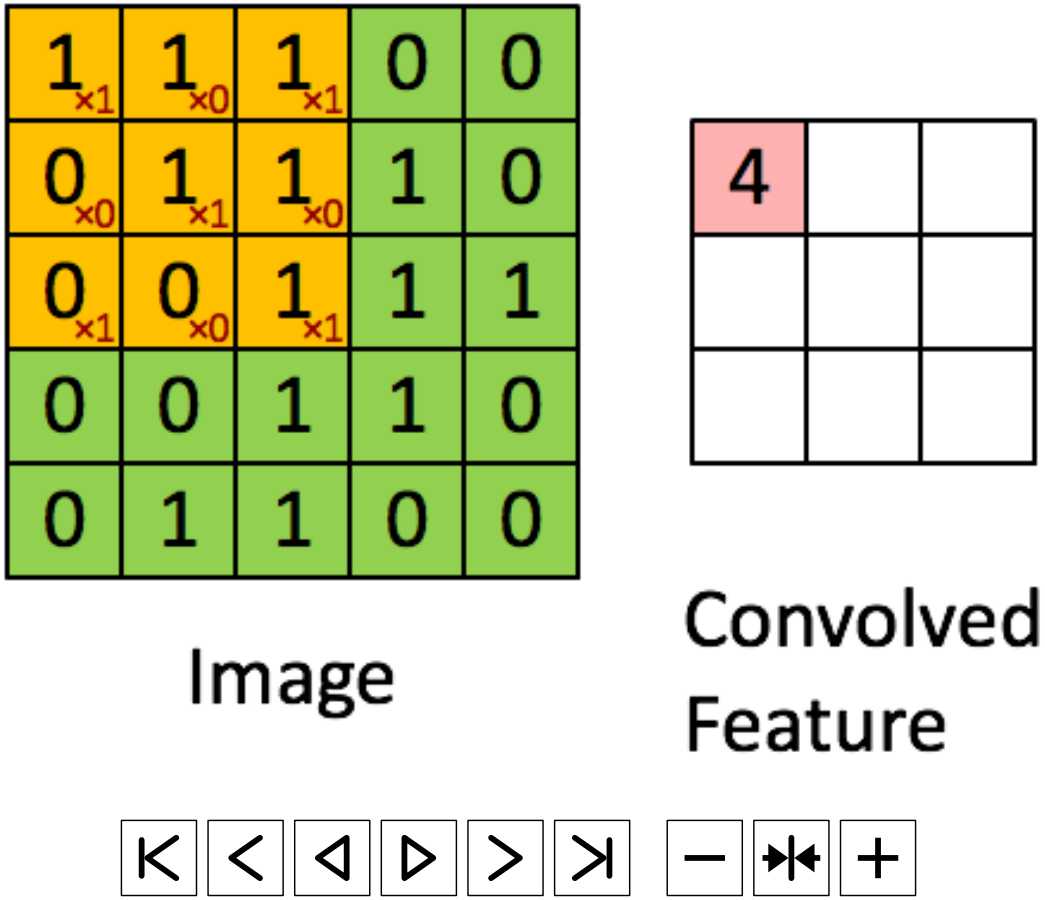


image source: http://stats.stackexchange.com/

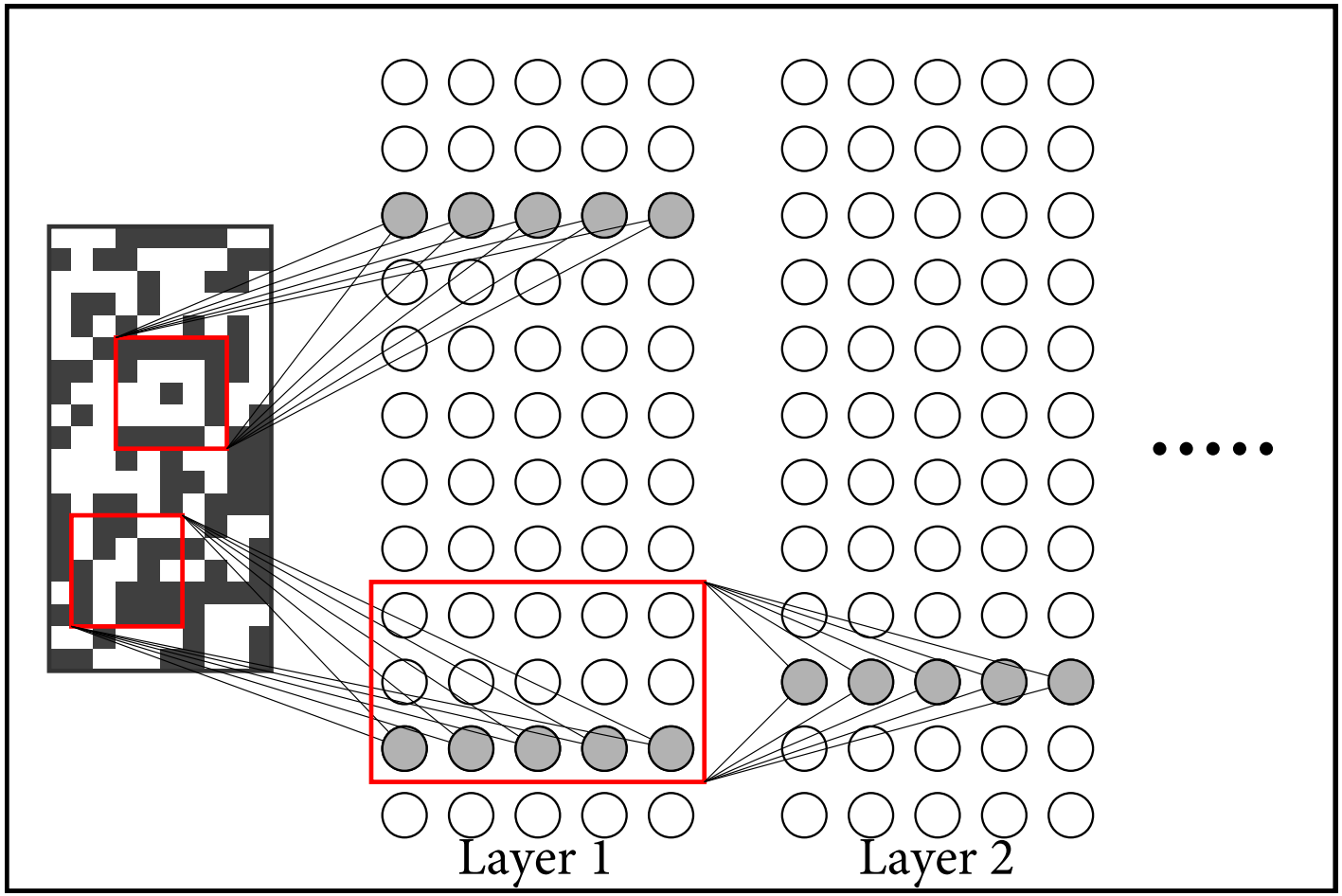
Examples of Convolution Filters

original blur Kuwahara

source: `http://stats.stackexchange.com/`

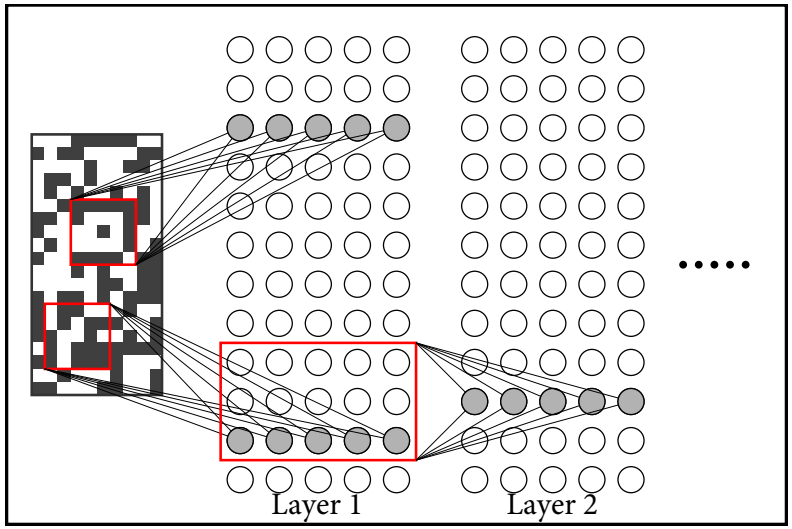
Can we learn such filters?

Convolutional Neural Networks (CNNs)



- Each layer consists of multiple **vertical planes** (**five** planes in the example shown here)

Convolutional Neural Networks (CNNs)

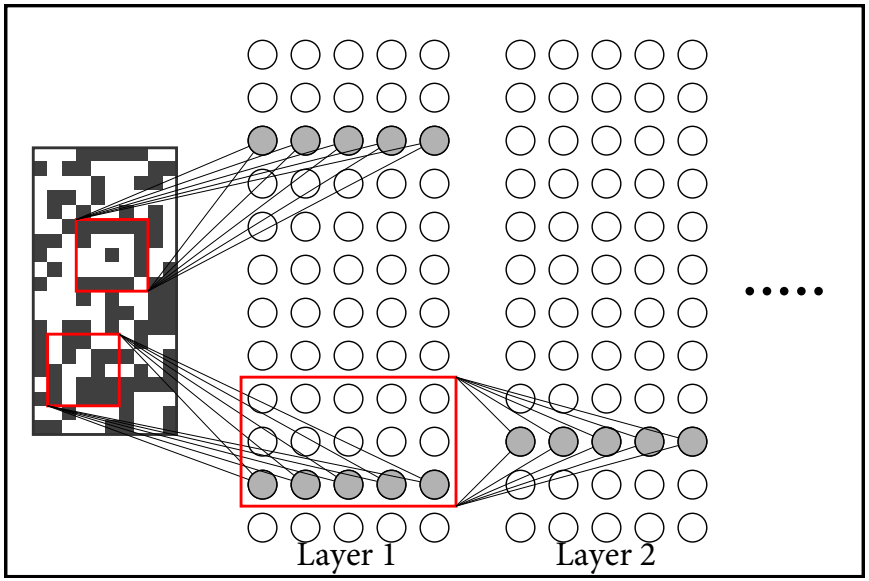


Each vertical plane acts as a convolution filter as follows:

- We cover the image with overlapping patches
- Each patch corresponds to five neurons on the same horizontal level
- Every pixel in a patch is connected with any of the five neurons in the corresponding level
  - Each of these neurons thus defines a filter on the pixels in the patch

Recursion: higher layers convolute output of earlier layers

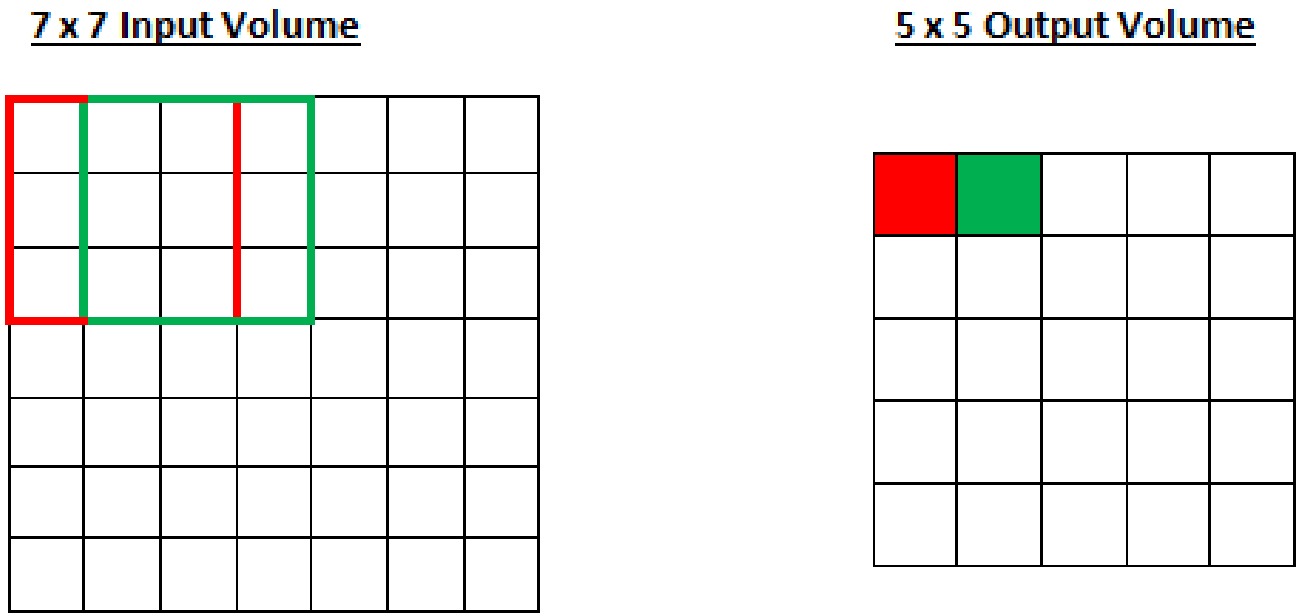
Weight Sharing



- In CNNs, neurons of the same plane are forced to share the same weights w_{ij} (**weight sharing**)
- Hence, the very same five filters are applied to every patch, no matter where in the image the patch is
  - $\Rightarrow$ **invariance against translation of objects in image** & drastic reduction in number of parameters

Stride

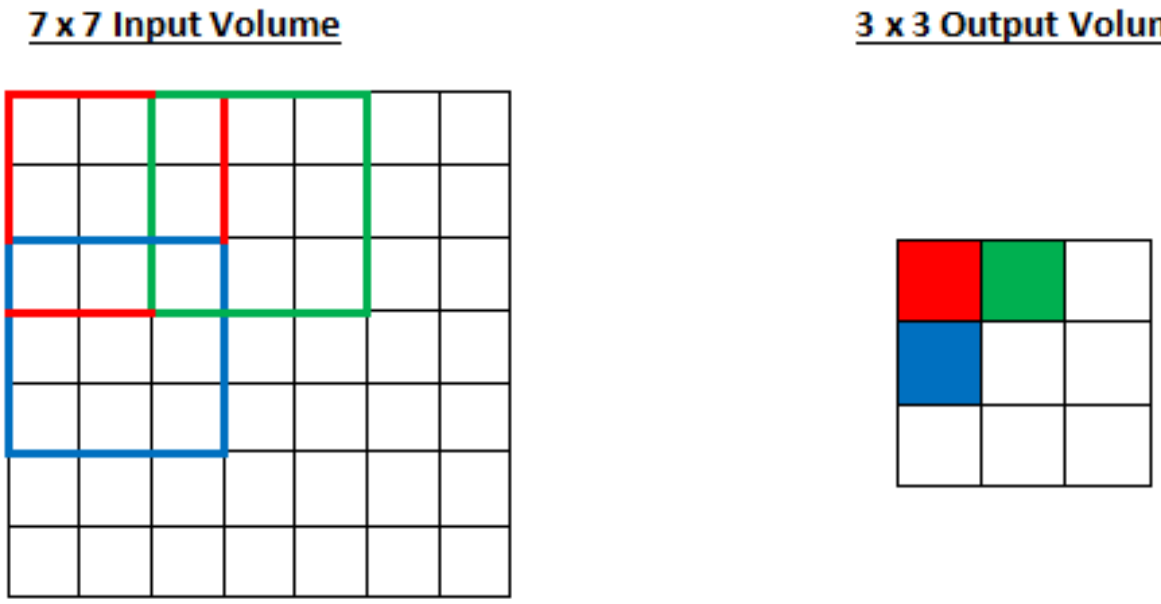
The **stride** determines how far apart the patches are:



Stride = 1

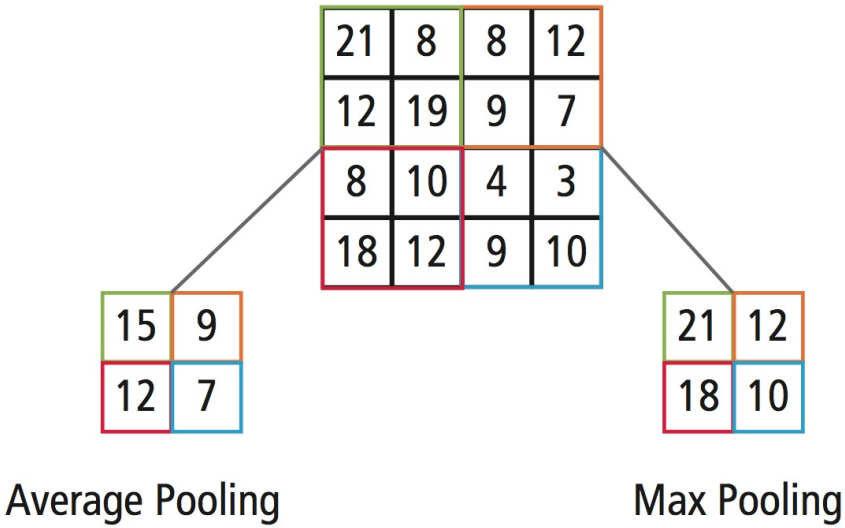
Stride

The **stride** determines how far apart the patches are:



Stride = 2

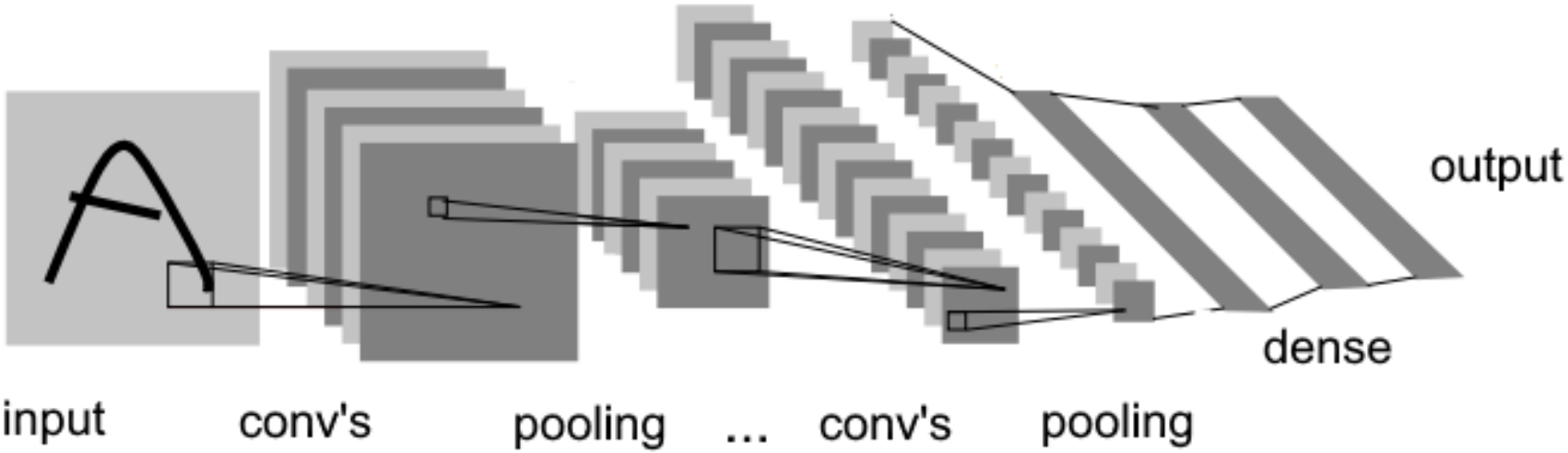
Pooling Layers



- In CNNs, convolution layers are alternated with **pooling layers**
  - these aggregate values in a patch
  - e.g., **max pooling** (computing in each patch the maximum entry)

  $\Rightarrow$ adds robustness against noise and changes in relative positions

CNN – Summary



Resulting architecture:

- input layer
- alternating convolution and pooling layers
- typically 2-3 dense layers (to capture non-linearities)
- output layer

Conclusion

Convolutional neural networks (CNN)

- instead of handcrafting image features
- let the learning machine (logistic regression) figure out a good representation
- wrap the learning machine around a CNN and learn a prediction model and an image representation at the same time

Next week: deep learning and how to train ANNs.

References I

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.