Convolutional Neural Networks – 02506 Spring 2020

Anders Bjorholm Dahl & Vedrana Andersen Dahl
DTU Compute

Advanced Image Analysis

April 2020

Learning objectives

- Understand the principles of convolutional neural networks
- Perform image classification using a convolutional neural network library
- Understand the different parameters used in the convolutional neural network
- Understand the principles of learning parameters of convolutional neural networks

Some images from: www.cs.toronto.edu/~rgrosse/csc321/lec11.pdf

Convolutional neural networks

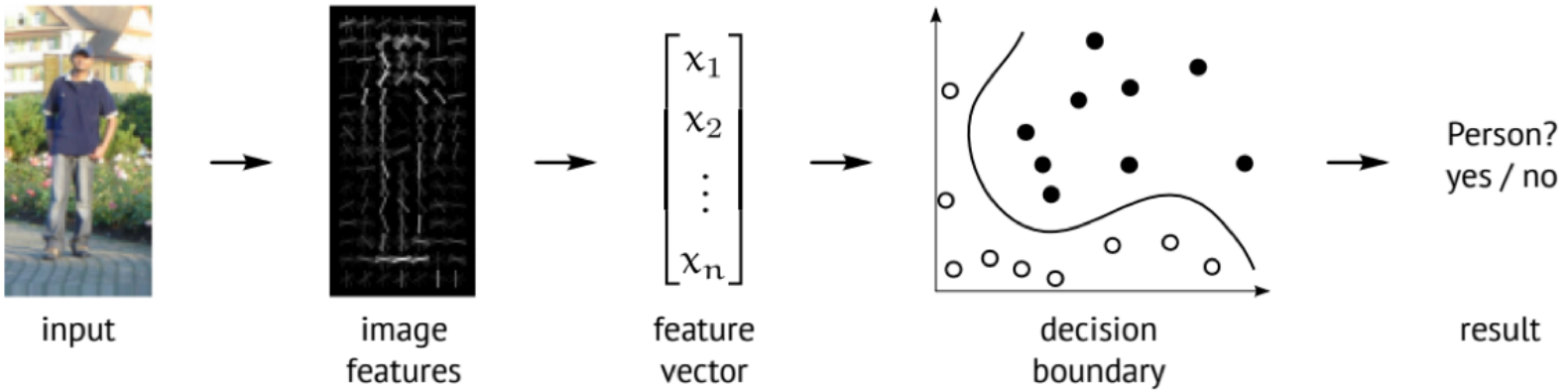
- You now know deep neural networks with fully connected layers
- What is a convolutional neural network? How is it different?
- What makes them especially suited for images?
- What are the parameters in convolutional neural networks?
- How are the parameters learned?

Suggested reading material

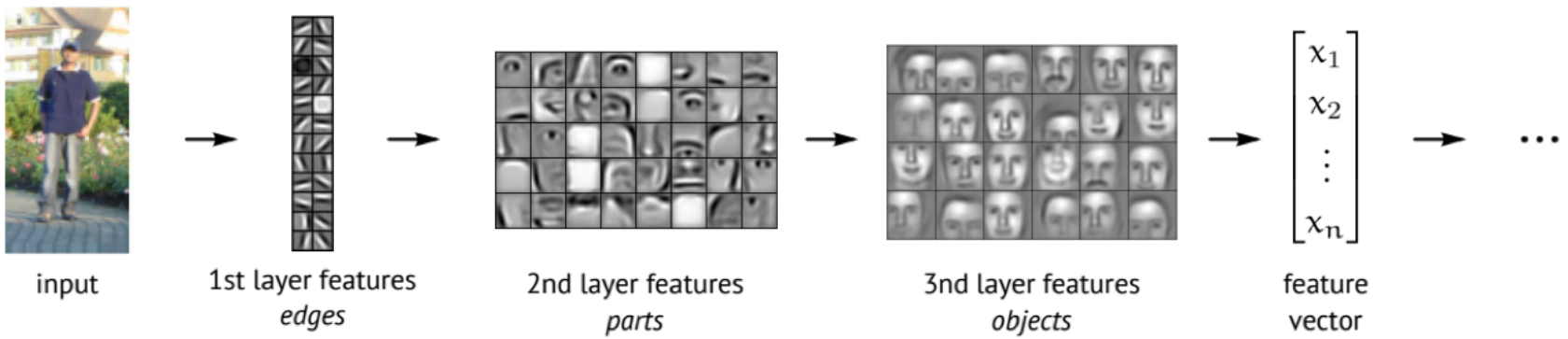
- Goodfellow, Bengio, and Courville. Deep Learning, Chapter 9.

Motivation

- Feature engineering – e.g. SIFT

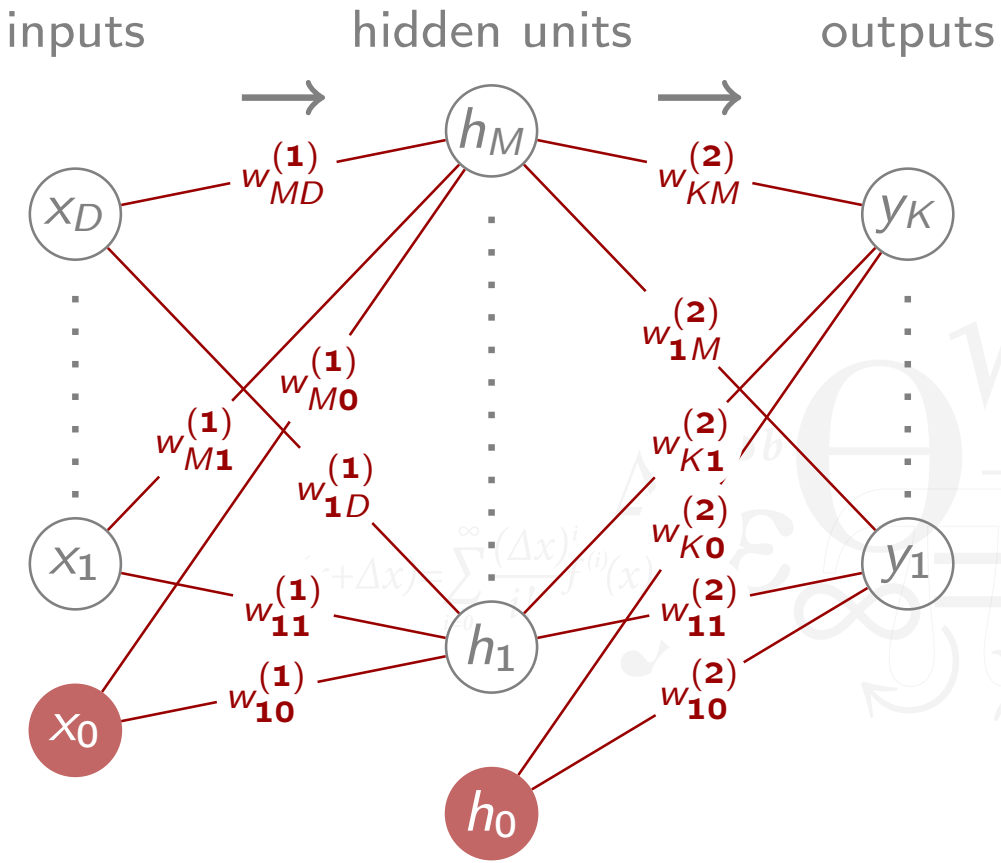


- Deep convolutional neural network



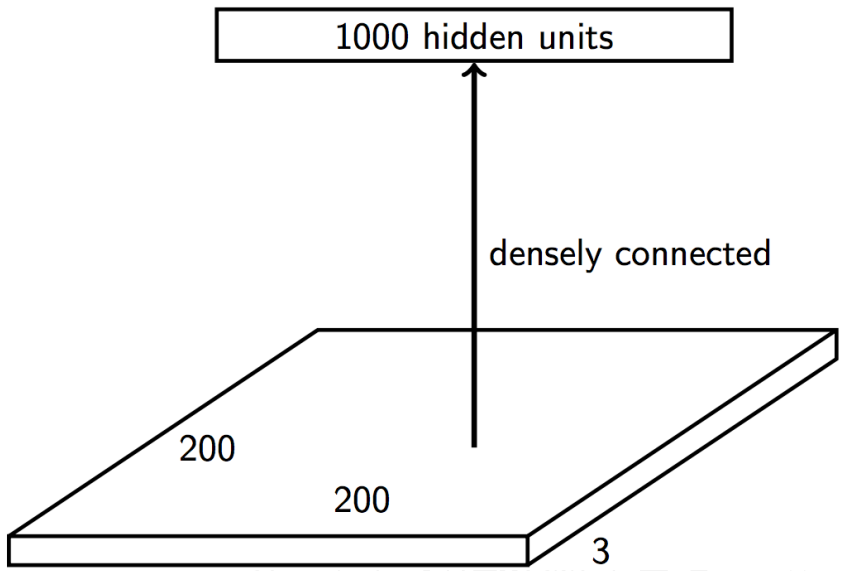
Learning deep neural networks

- Fully connected neural networks



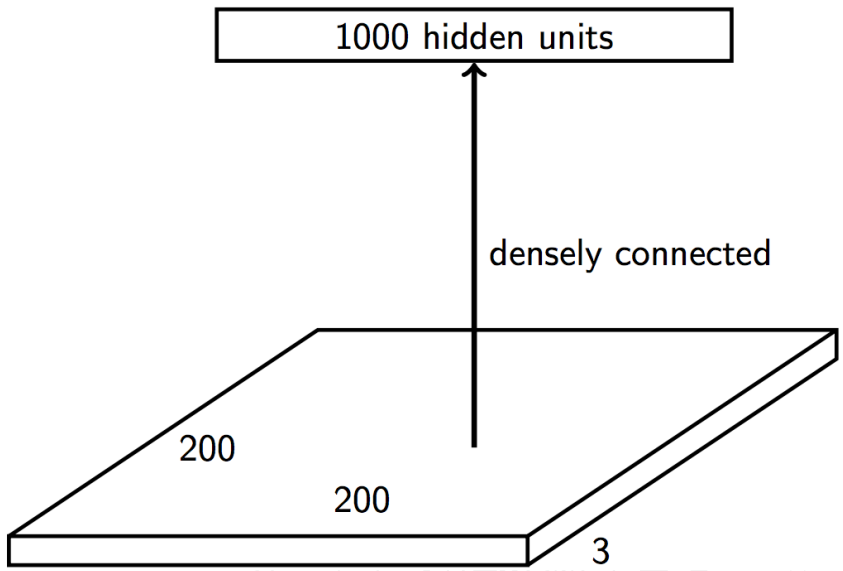
Images

- What happens when we get to larger images?



Images

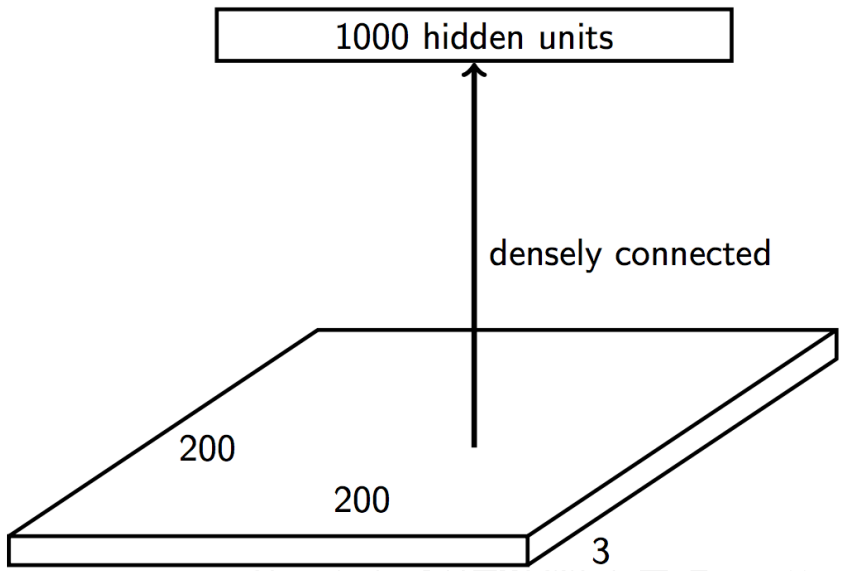
- What happens when we get to larger images?



- Number of pixels: $200 \times 200 \times 3 = 120000$
- Number of weights: $120000 \times 1000 = 120000000$

Images

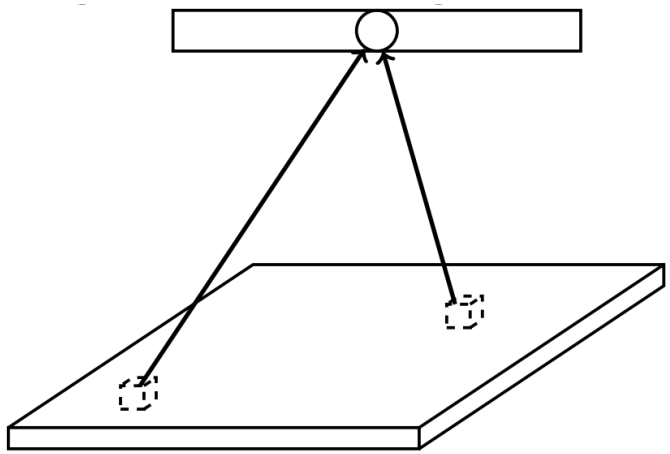
- What happens when we get to larger images?



- Number of pixels: $200 \times 200 \times 3 = 120000$
- Number of weights: $120000 \times 1000 = 120000000$
- How can we avoid this explosion?

Images

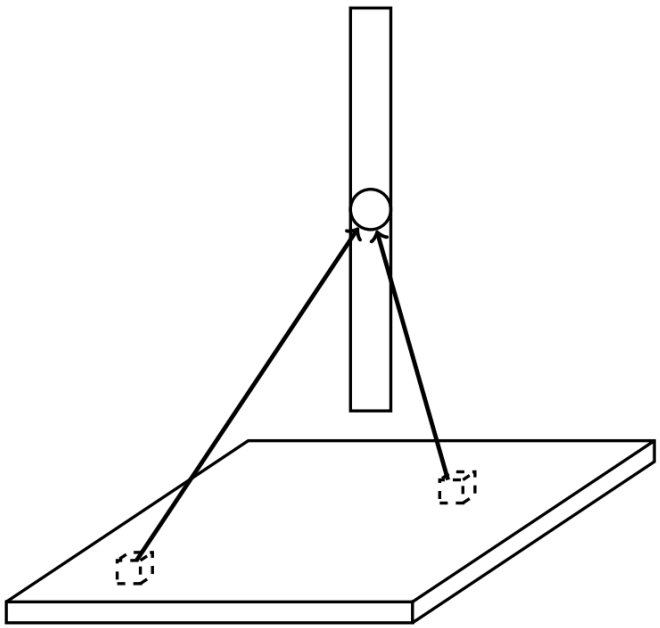
- Each neuron is connected to all pixels
- Far away pixels – not correlated



- Look locally and share weights!
- Use layers to correlate features at longer distances.

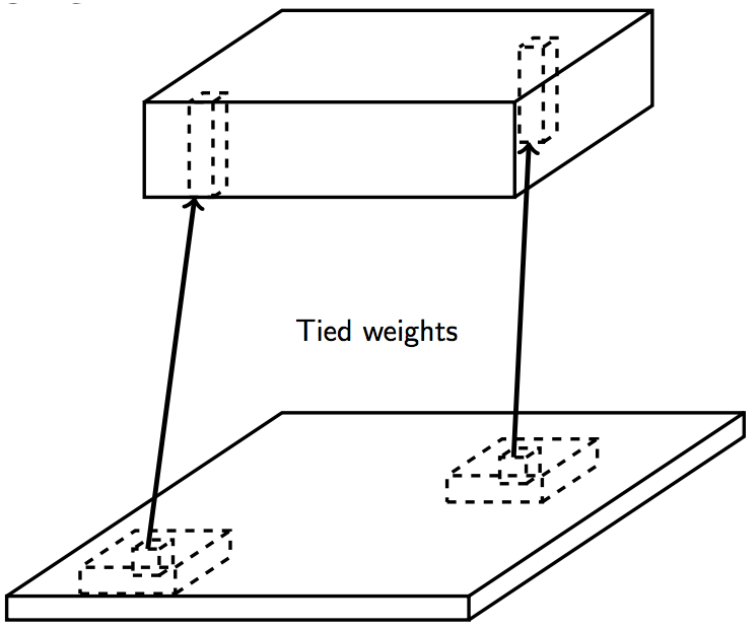
Fully connected vs convolutional

Fully connected



- Each neuron looks in the entire image

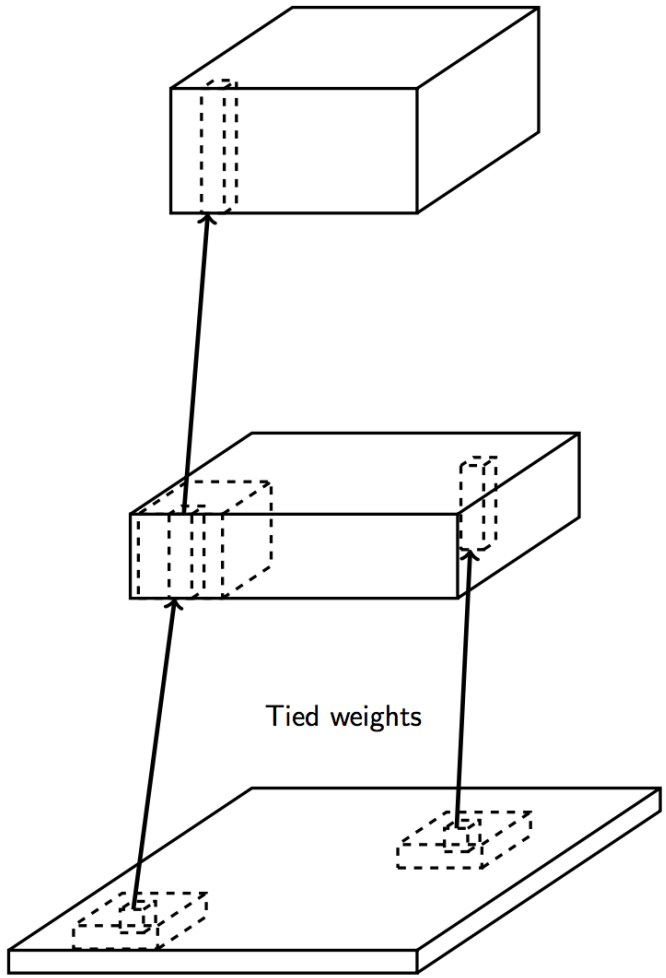
Convolutional neural network



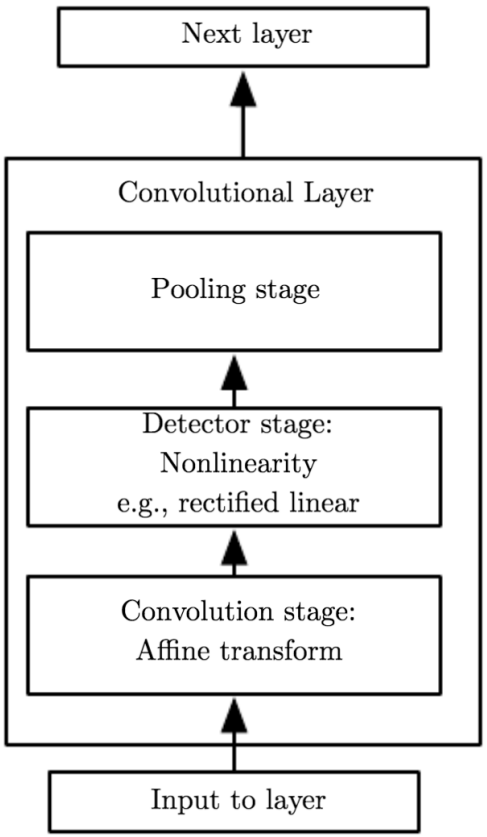
- Each column of hidden units only looks at a patch

Deep convolutional neural network

- Build a network based on convolutions
- Parameters are the convolution weights – shared over the image
- Filters are tensors (a set of small volumes)
- Forward propagation as a series of convolutions
- Layer operations: Convolutions (weighted average), activation, pooling



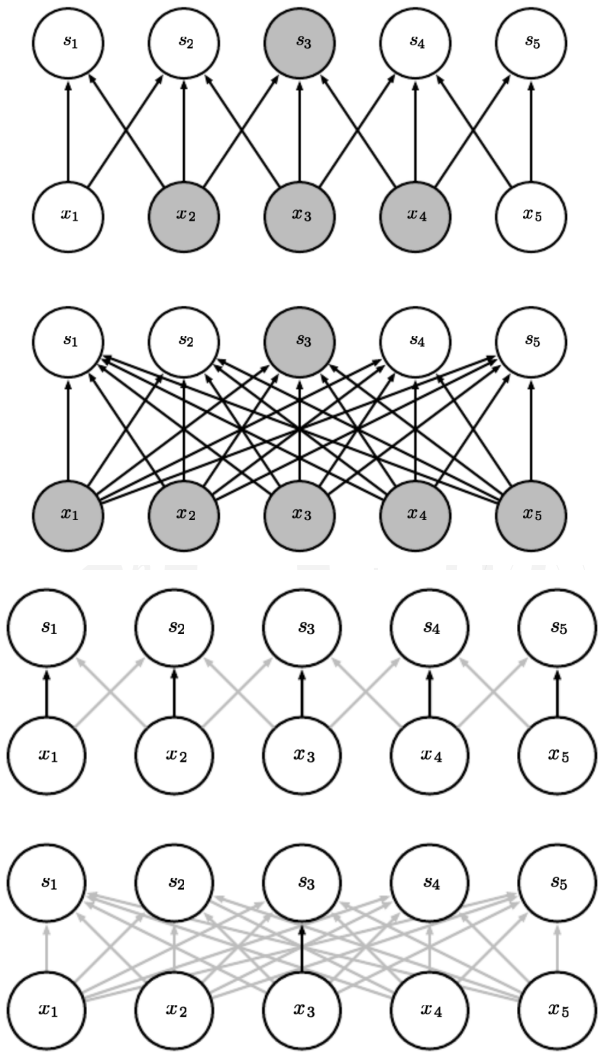
Forward propagation



- Steps in each layer:
  - Convolution
  - Nonlinear map
  - Pooling

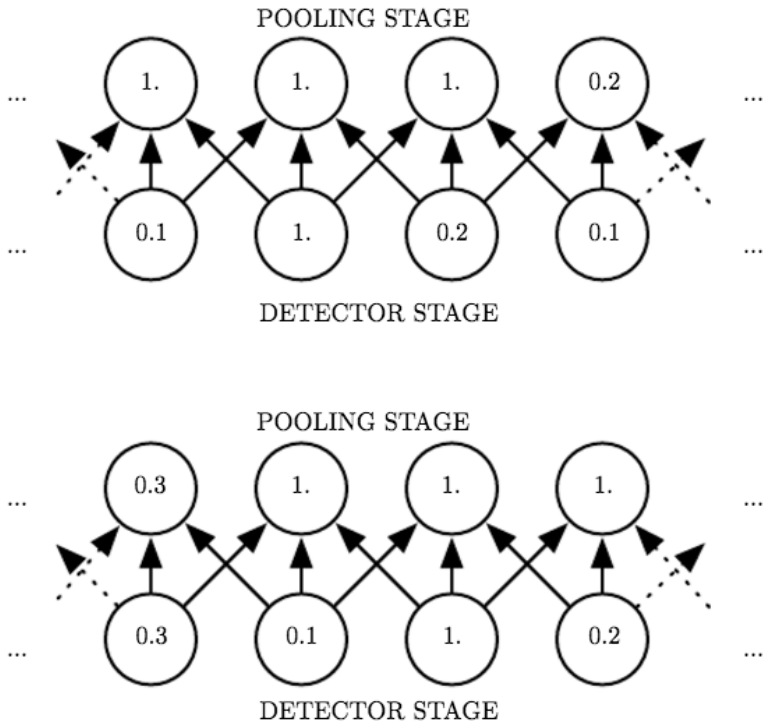
Convolution

- Weighted average
- Sparse interactions
- Shared weights
- Typically one bias per filter
- Filters as tensors (a set of small volumes)
- Zero padding to avoid shrinkage of image



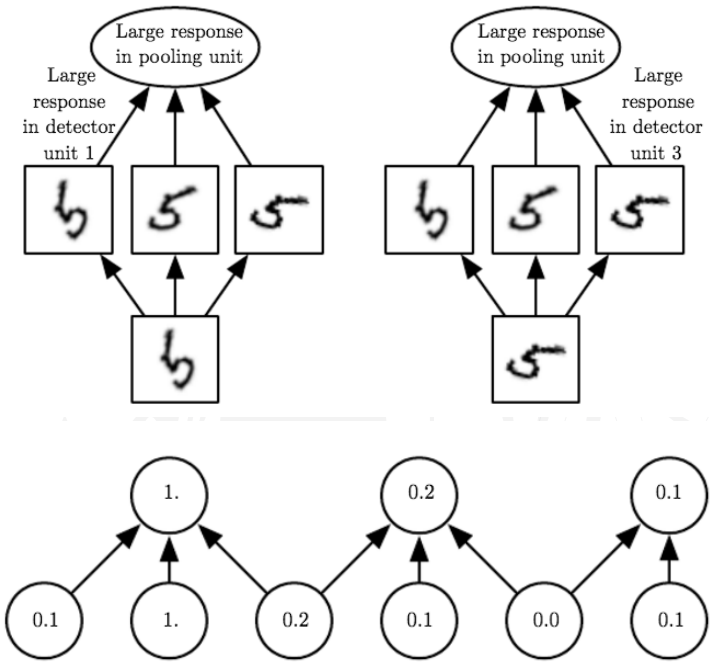
Pooling

- Max pooling
- Average, weighted average, etc. are also possible
- Introduces invariance – translation, different features
- Possibility for down-sampling



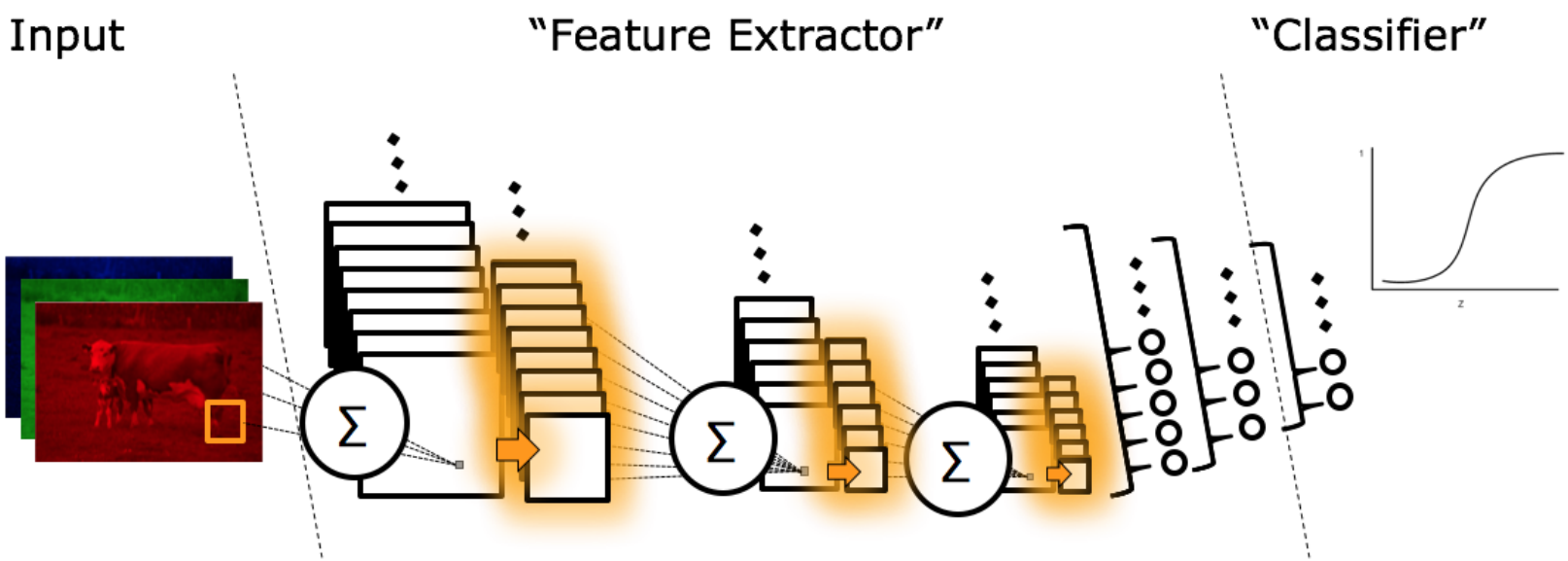
Pooling

- Max pooling
- Average, weighted average, etc. are also possible
- Introduces invariance – translation, different features
- Possibility for down-sampling



Network architecture

- Large number of layers
  - GoogLeNet – 22 layers
  - VGGnet – 19 layers
  - 2015 Microsoft – 152 layers



Backpropagation

- MLP backpropagation

$$\frac{\partial L}{\partial z_i^l} = \delta_j^l$$

$$\delta_j^l = a'(z_i^l) \sum_k w_{ki}^{l+1} \delta_k^{l+1}$$

$$\frac{\partial L}{\partial w_{ij}^l} = \delta_i^l h_i^{l-1}$$

- CNN backpropagation

$$\frac{\partial L}{\partial z_i^l} = \delta_i^l$$

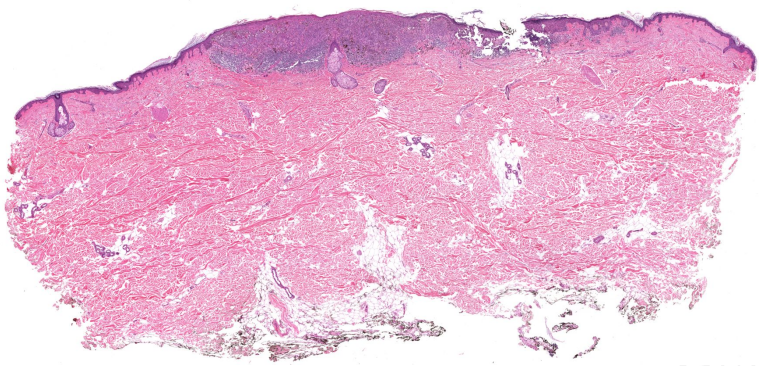
$$\delta_i^l = a'(z_i^l) \sum_k w_{ki}^{l+1} \delta_k^{l+1}$$

$$\frac{\partial L}{\partial w_{ij}^l} = \sum_{m,n} \delta_m^l h_n^{l-1}$$

- Can be implemented using convolutions

Problem

- Whole-slide imaging[1]

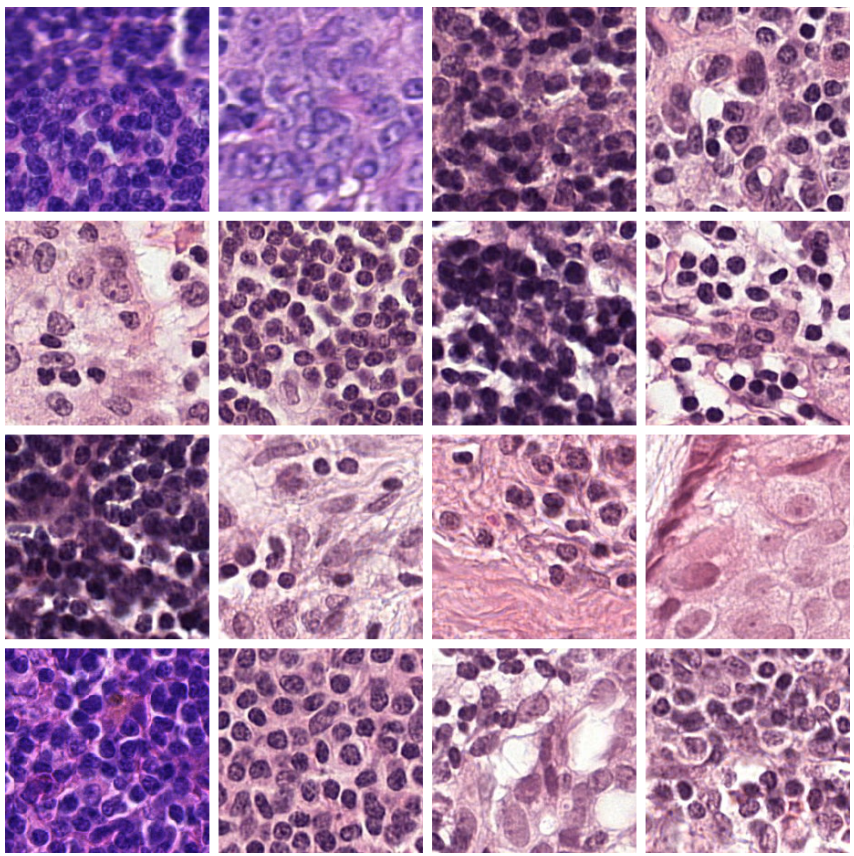


[1] Obtained from: http://wwwx.cs.unc.edu/~mn/?q=content/image-analysis-histopathology-melanoma-classification,http://www.sysmex.co.nz/the-9th-asia-pacific-congress-recap/

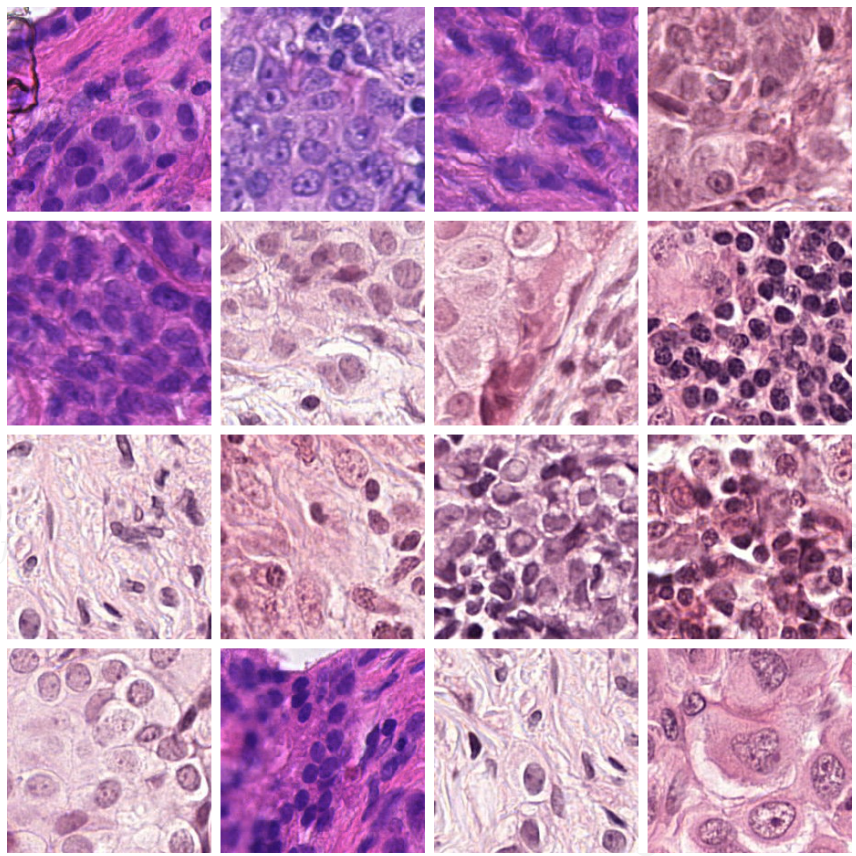
Data

- CEMELYON17 competition[2]

Healthy tissue



Cancer Tissue



[2] Obtained from: https://camelyon17.grand-challenge.org/

Exercise: Pre-trained network for classification

- Now sparse connections through convolution
- Now use others implementation – MatConvNet library (`http://www.vlfeat.org/matconvnet/`) or Keras (`https://keras.io/`)
- Problem – image classification using k-nearest neighbor
- Investigate an already trained network
- Use a simple classifier on the output features

02514 Deep learning for Computer Vision

- Recognition – classification
- Image generation
- Segmentation
- Runs in 3 weeks period in June

MNIST competition – remember the prize!

- MNIST classification
- Competition – not insignificant eatable prize for winning team (probably some post easter theme!)