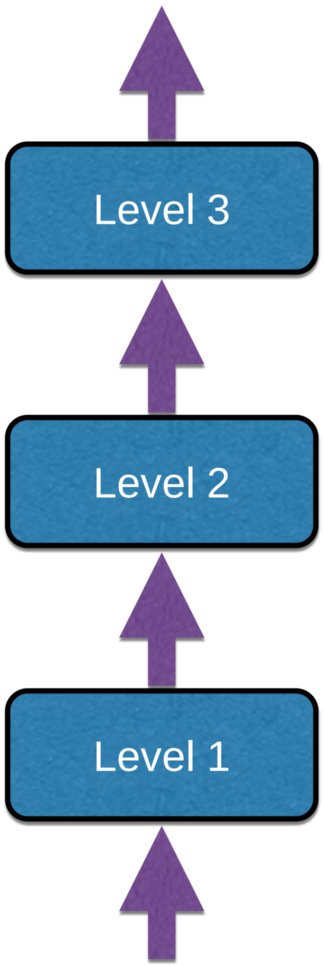
Lecture 10.3
Introduction to deep learning (CNN)

Idar Dyrdal

Deep Learning

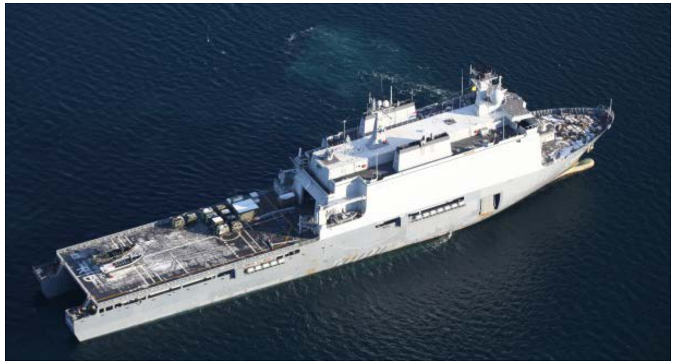
- Computational models composed of multiple processing layers (non-linear transformations)
- Used to learn representations of data with multiple levels of abstraction:
    - Learning a hierarchy of feature extractors
    - Each level in the hierarchy extracts features from the output of the previous layer (pixels —> classes)
- Deep learning has dramatically improved state-of-the-art in:
    - Speech and character recognition
    - Visual object detection and recognition
- Convolutional neural nets for processing of images, video, speech and signals (time series) in general
- Recurrent neural nets for processing of sequential data (speech, text).

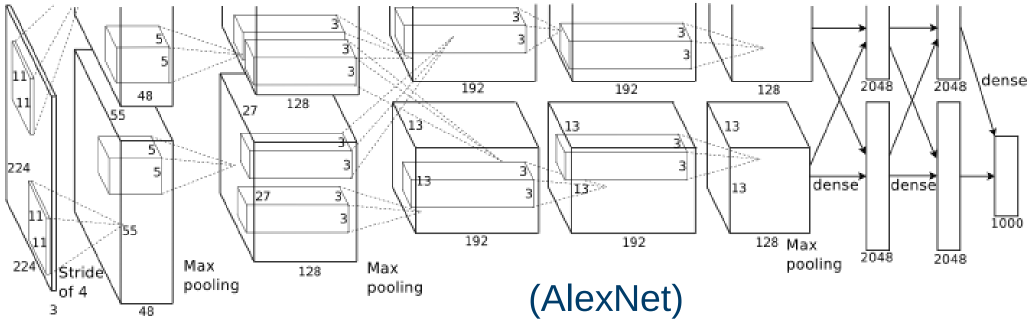
Labels



Raw data (images, video, signals)

Deep Learning for Object Recognition





(AlexNet)

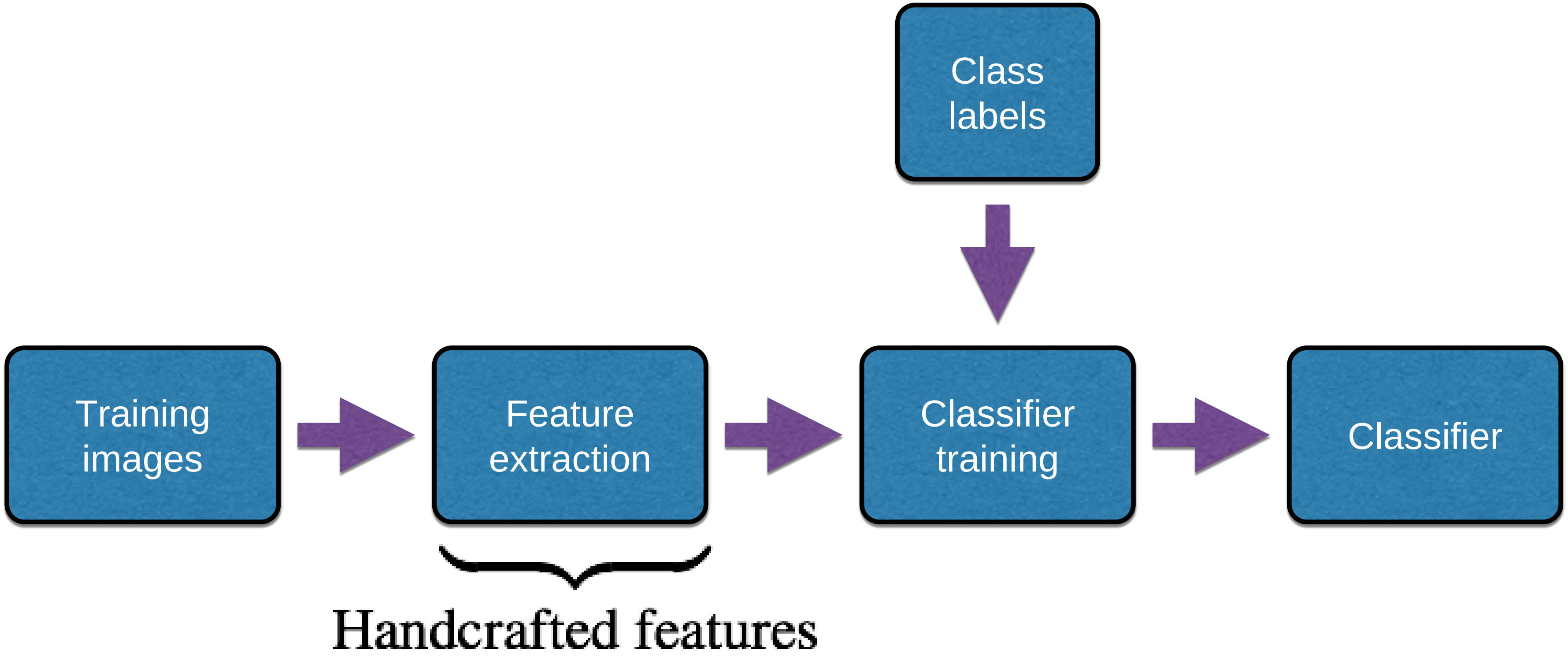
«Ship»

Millions of images

Millions of parameters

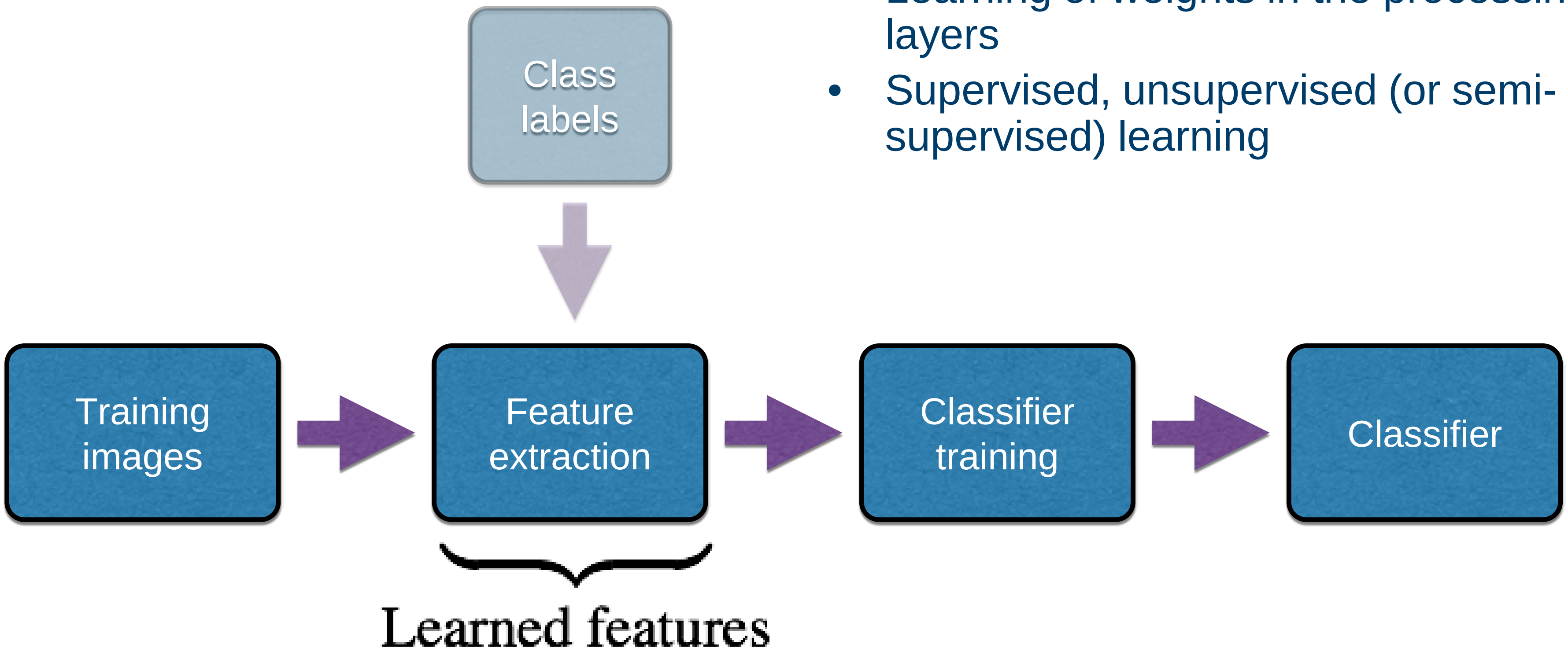
Thousands of classes

Traditional supervised learning

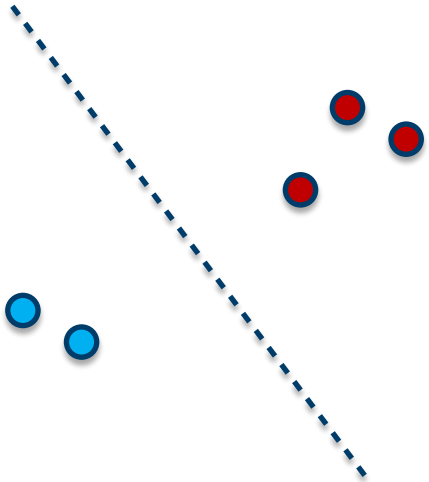


Deep learning

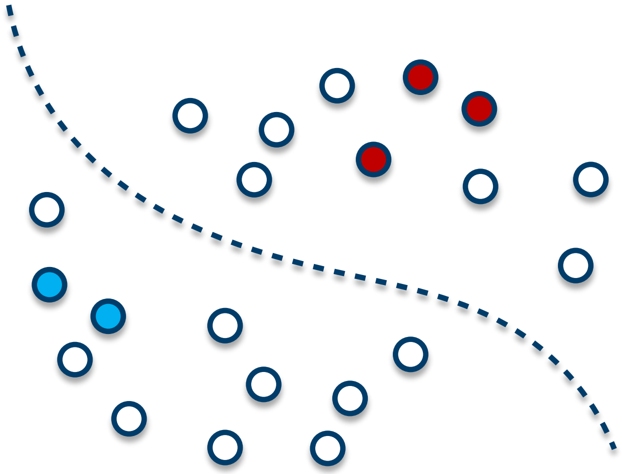
- Learning of weights in the processing layers
- Supervised, unsupervised (or semi-supervised) learning



Semi-supervised learning



Labeled samples and (trained) linear decision boundary



Labeled and unlabeled samples and non-linear decision boundary

Artificial Neural Network (ANN)

Used in Machine Learning and Pattern Recognition:

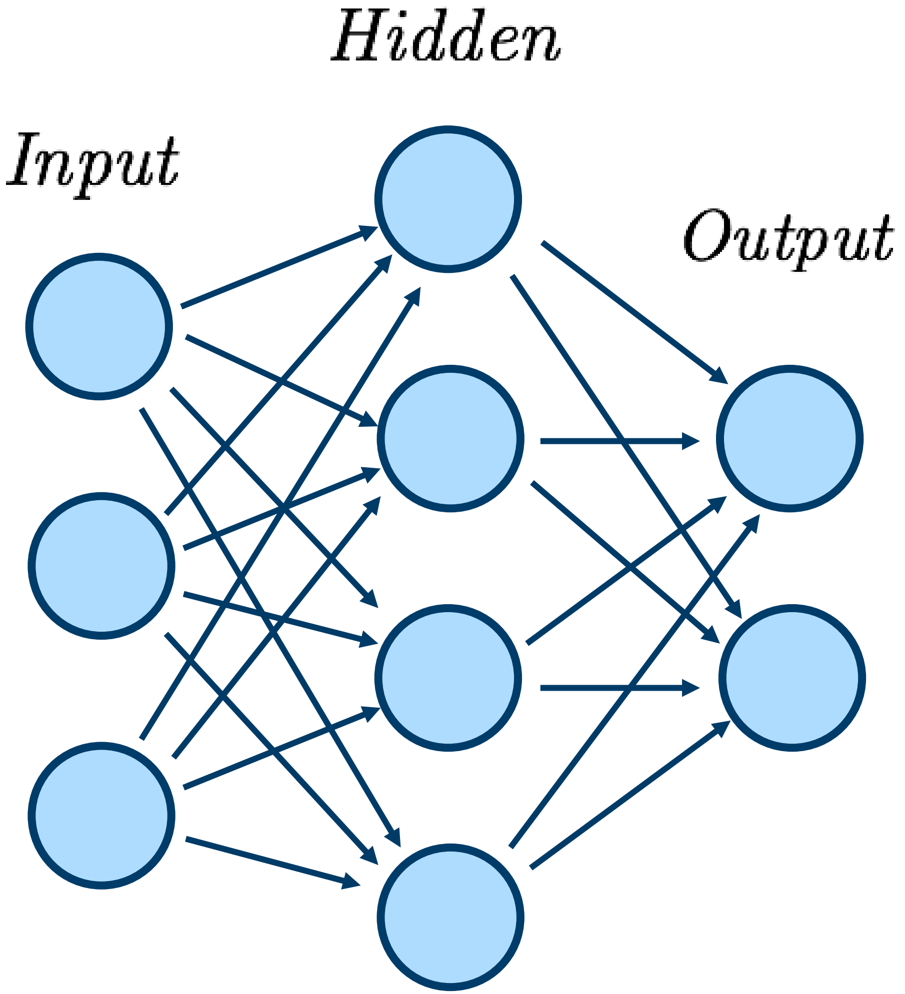
- Regression
- Classification
- Clustering
- …

Applications:

- Speech recognition
- Recognition of handwritten text
- Image classification
- …

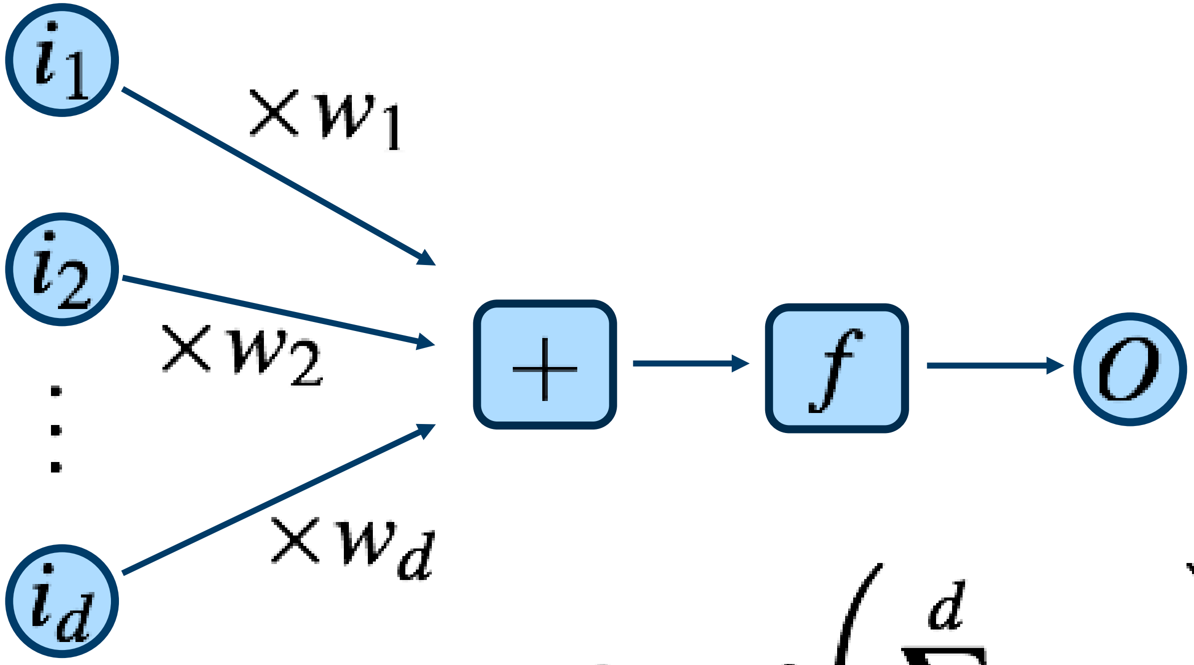
Network types:

- Feed-forward neural networks
- Recurrent neural networks (RNN)
- …

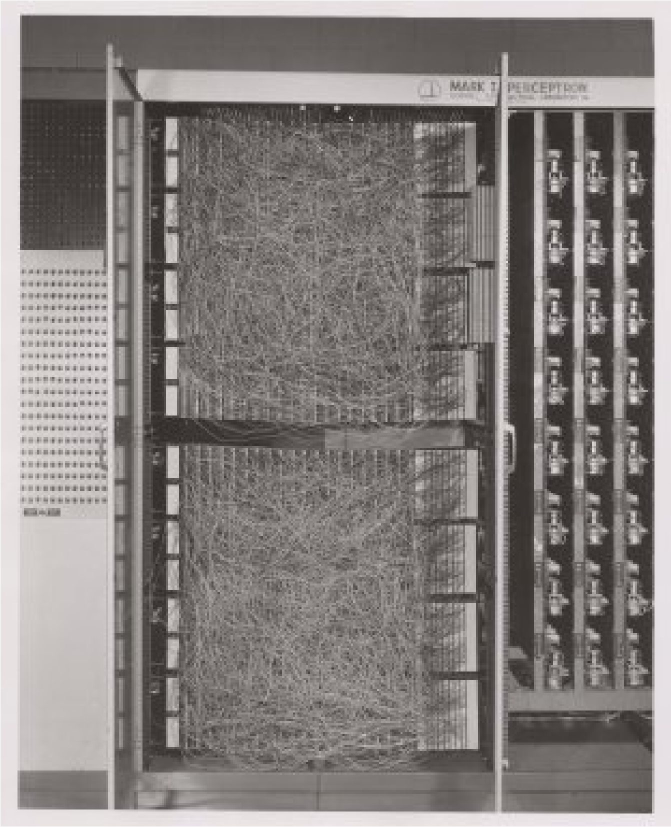


Feed-forward ANN (non-linear classifier)

Mark 1 Perceptron (Rosenblatt, 1957-59)



$$O = f\left(\sum_{k=1}^{d} i_k w_k\right)$$



Cornell Aeronautical Laboratory

Activation functions

- Sigmoid (logistic function):

$$f(x) = \frac{1}{1+e^{-x}}$$

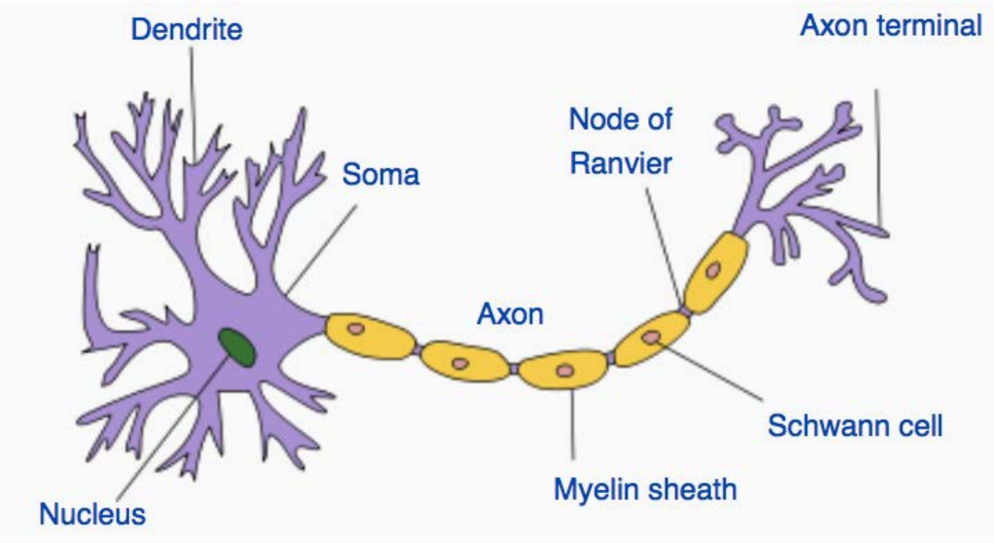
- Hyperbolic tangent:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- Rectified linear unit (ReLU):

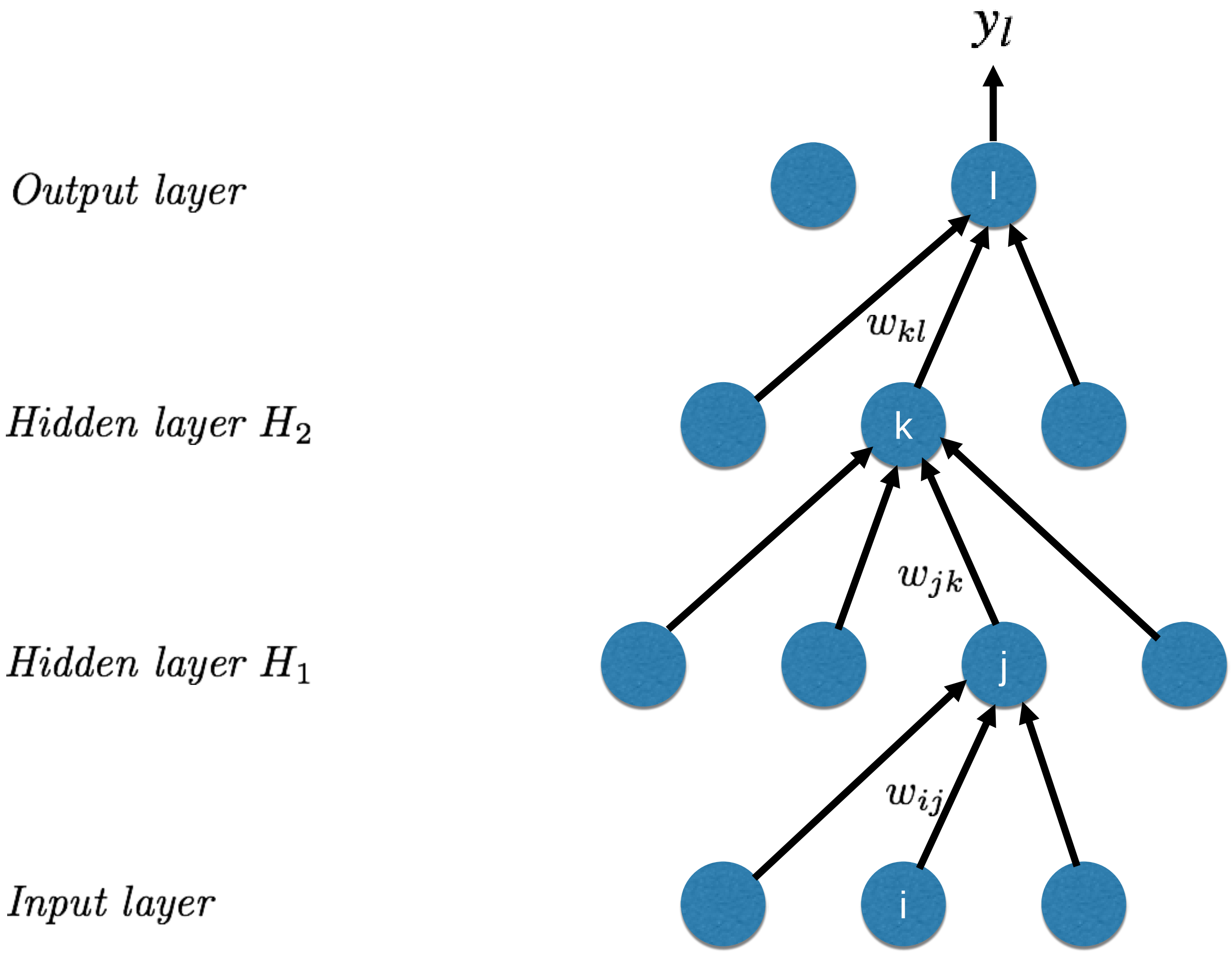
$$f(x) = \max(x, 0)$$

Biological neuron:



(Quasar Jarosz, English Wikipedia)

Feed-forward neural network



$$y_l = f(z_l)$$

$$z_l = \sum_{k \in H_2} w_{kl} x_k$$

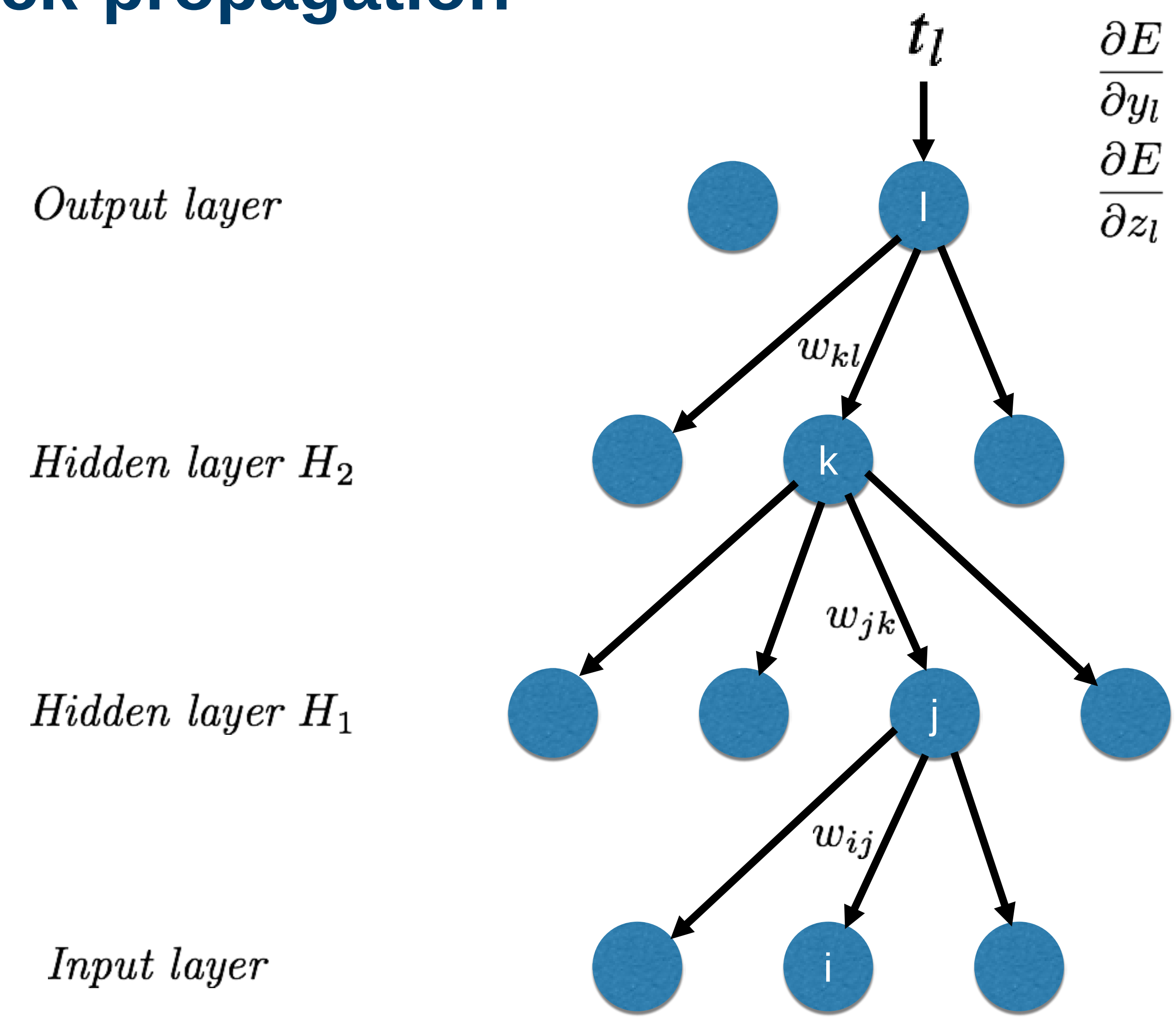
$$y_k = f(z_k)$$

$$z_k = \sum_{j \in H_1} w_{jk} x_j$$

$$y_j = f(z_j)$$

$$z_j = \sum_{i \in Input} w_{ij} x_i$$

Back-propagation



$$E(\vec{w}) = \sum_{k=1}^{n} (t_i - y_i)^2$$

$$\frac{\partial E}{\partial y_l} = y_l - t_l$$

$$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial z_l}$$

$$\frac{\partial E}{\partial y_k} = \sum_{l \in Output} w_{kl} \frac{\partial E}{\partial z_l}$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k}$$

$$\frac{\partial E}{\partial y_j} = \sum_{k \in H_2} w_{jk} \frac{\partial E}{\partial z_k}$$

$$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial z_j}$$

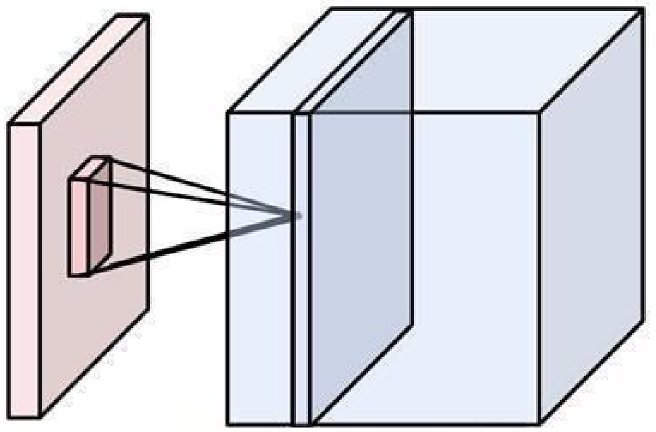
Convolutional Neural Network (CNN)

Used in Signal and Image Analysis:

- Speech Recognition
- Image Recognition
- Video Recognition
- Image Segmentation
- …

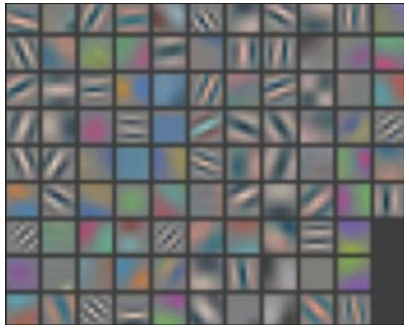
Convolutional neural network:

- Multi-layer feed-forward ANN
- Combinations of *convolutional* and fully connected layers
- Convolutional layers with *local* connectivity
- *Shared* weights across spatial positions
- Local or global pooling layers

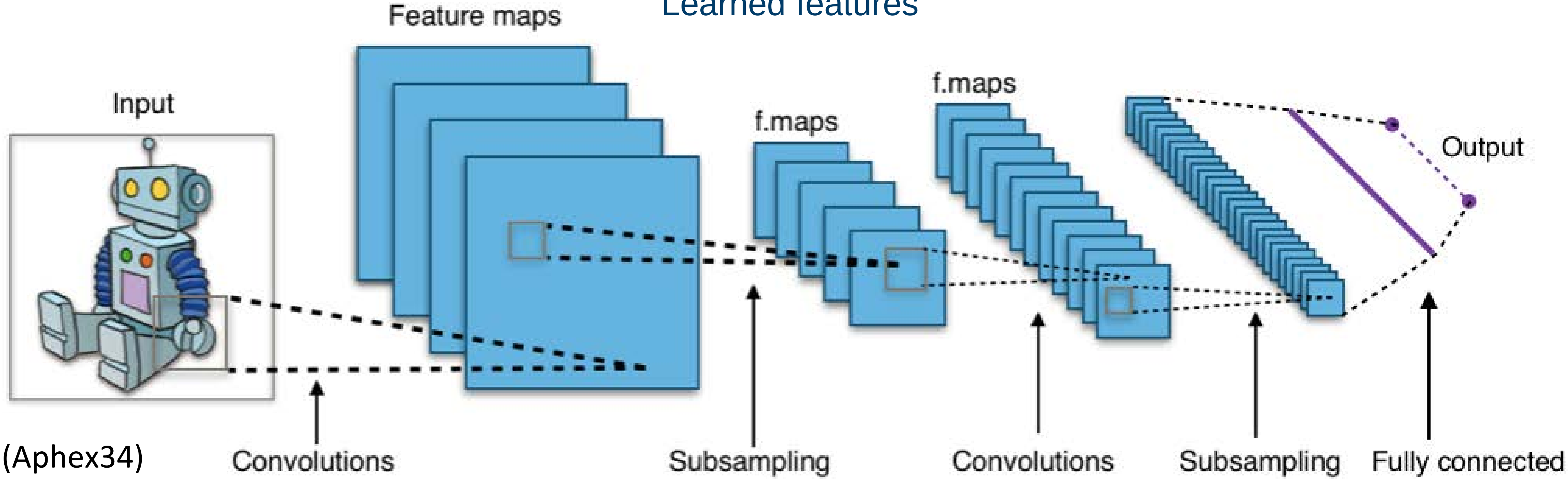


(A. Karpathy)

Typical CNN



Learned features



(Aphex34)

Convolutional neural net

Input image

credit: S. Lazebnik

Feature map

Normalisation

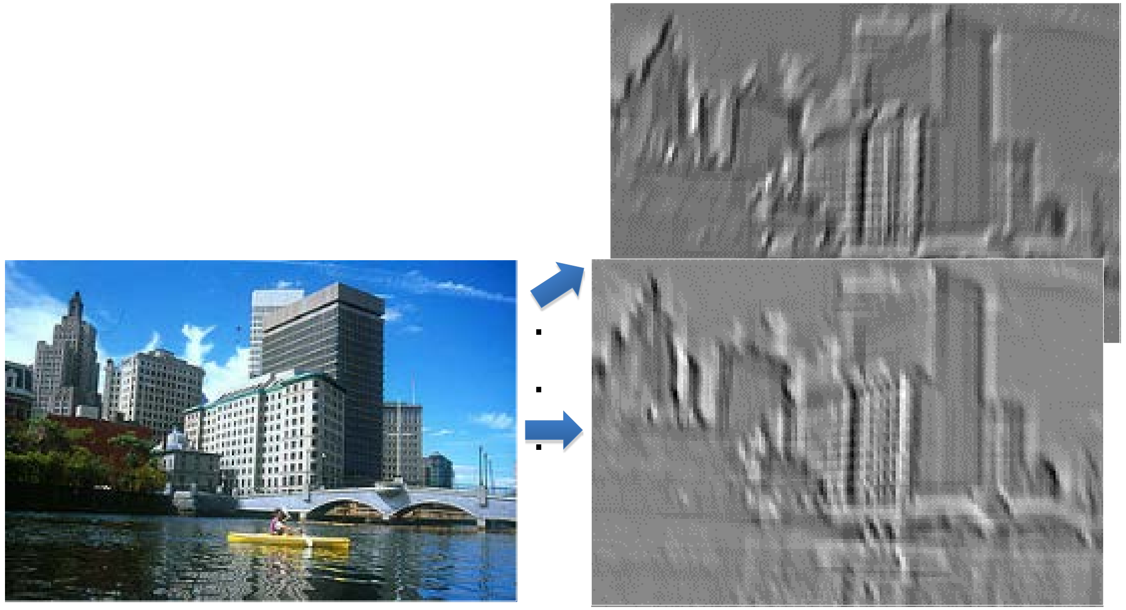
Spatial pooling

Non-linearity

Convolution (learned)

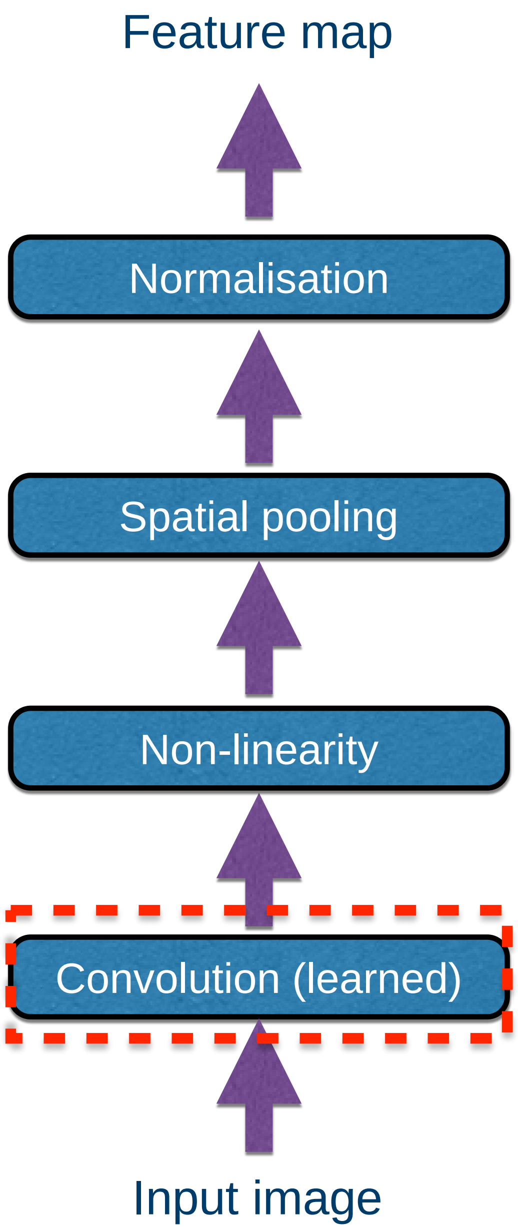
Input image

Convolutional neural net



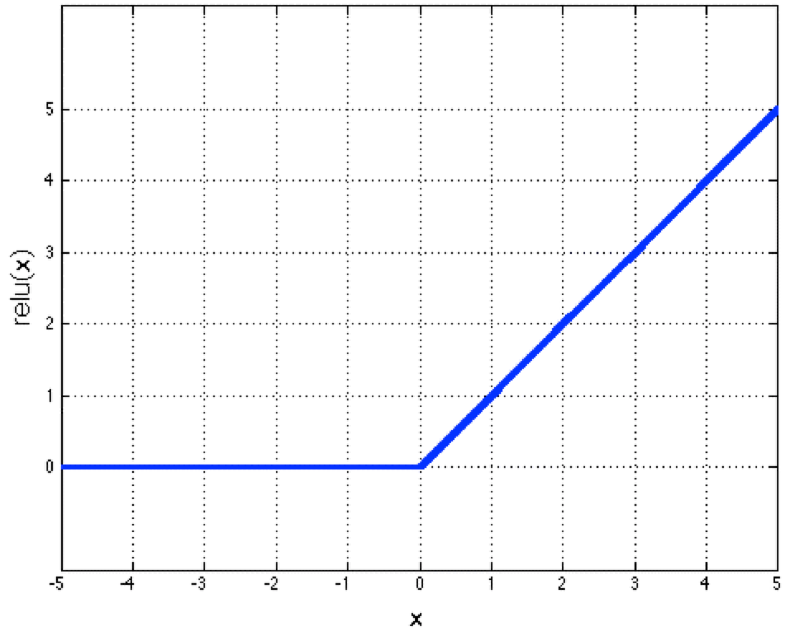
Input　　Feature Map

credit: S. Lazebnik

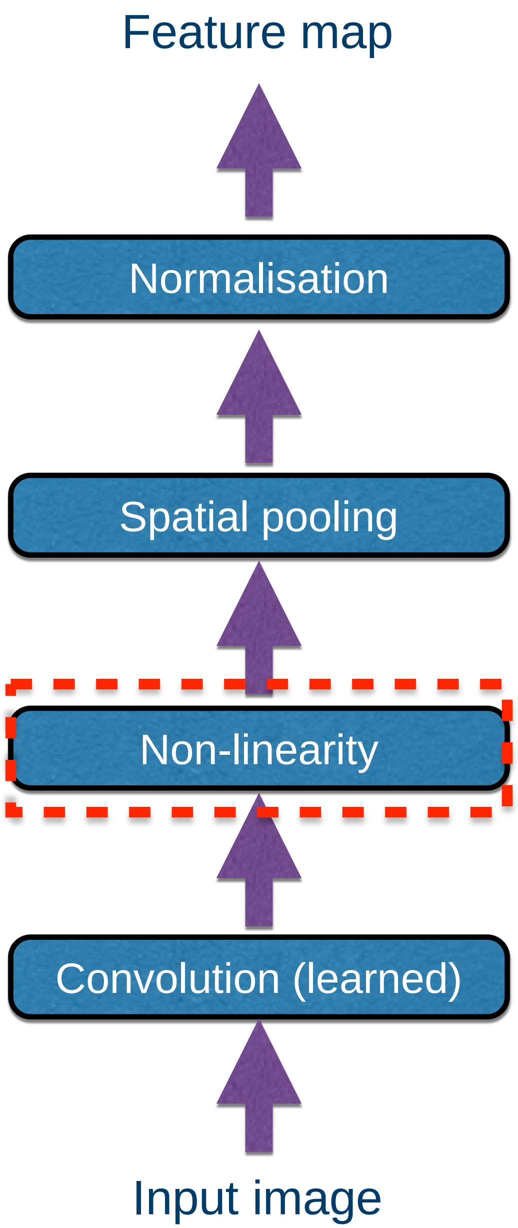


Convolutional neural net

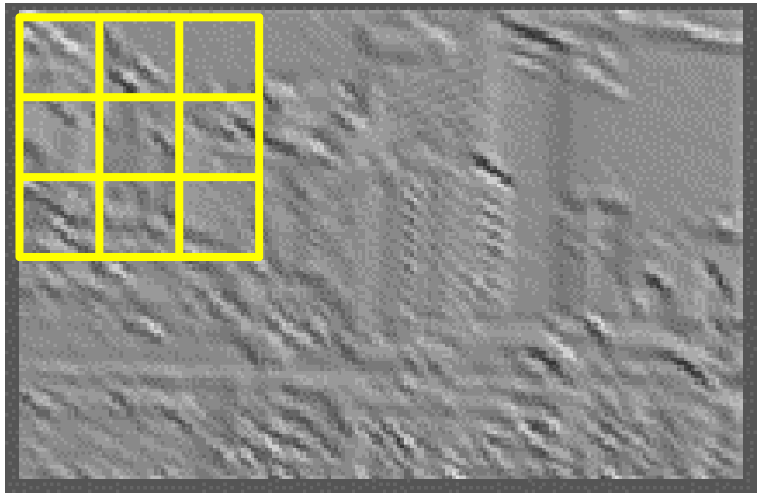
Rectified Linear Unit (ReLU)



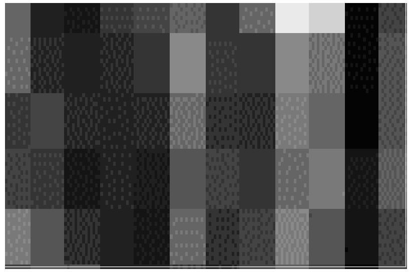
credit: S. Lazebnik



Convolutional neural net



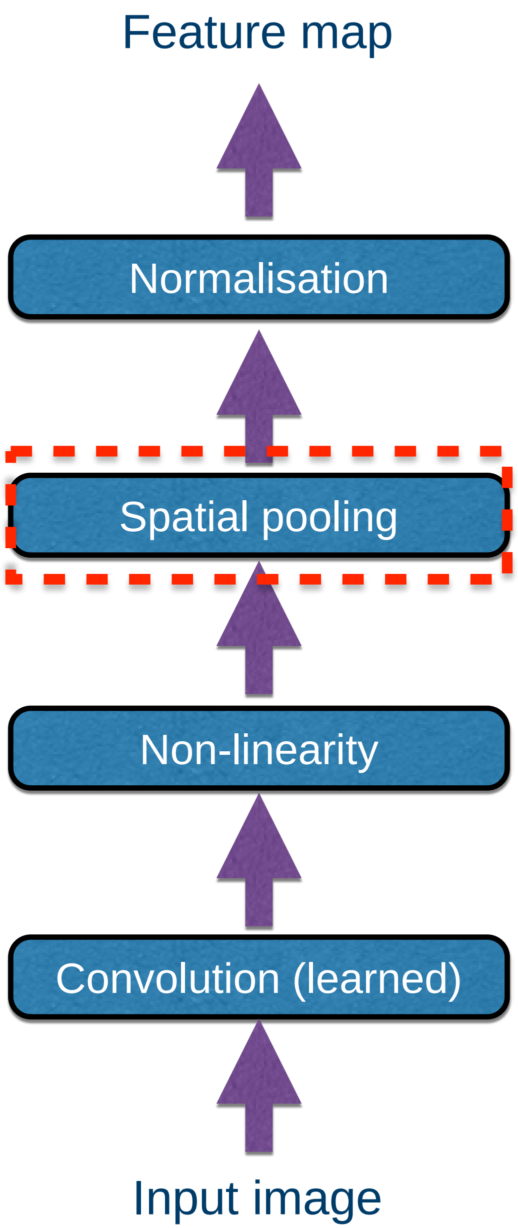
Max pooling



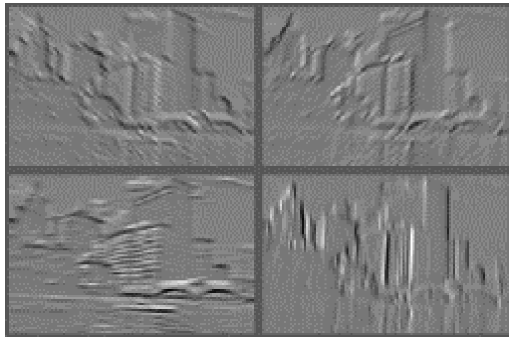
Max-pooling: a non-linear down-sampling

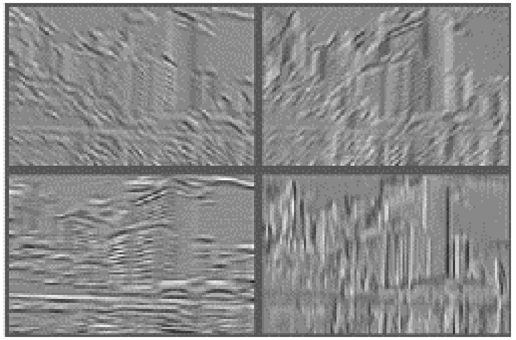
Provide *translation invariance*

credit: S. Lazebnik

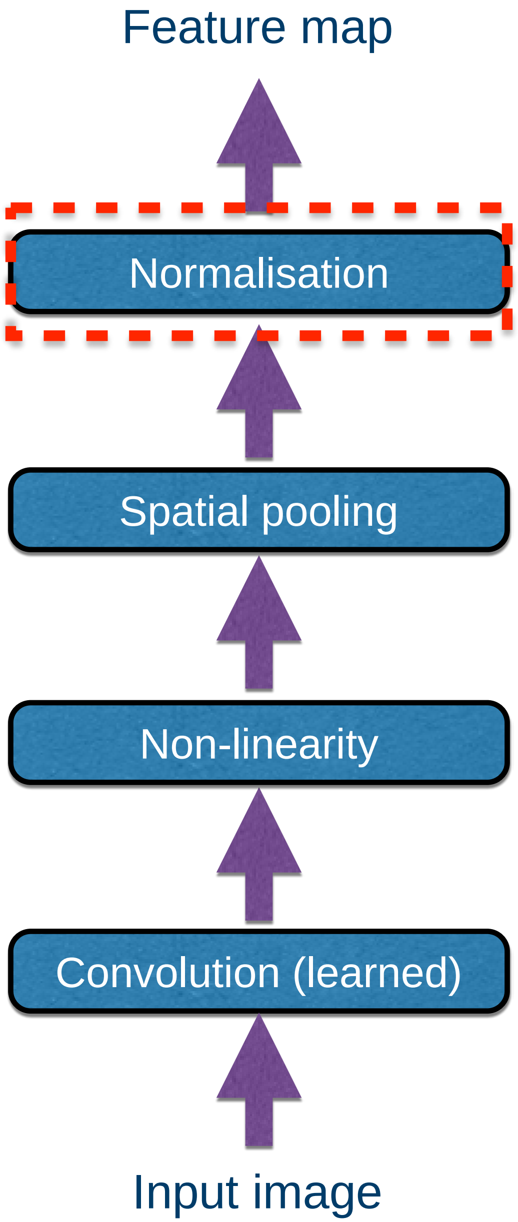


Convolutional neural net

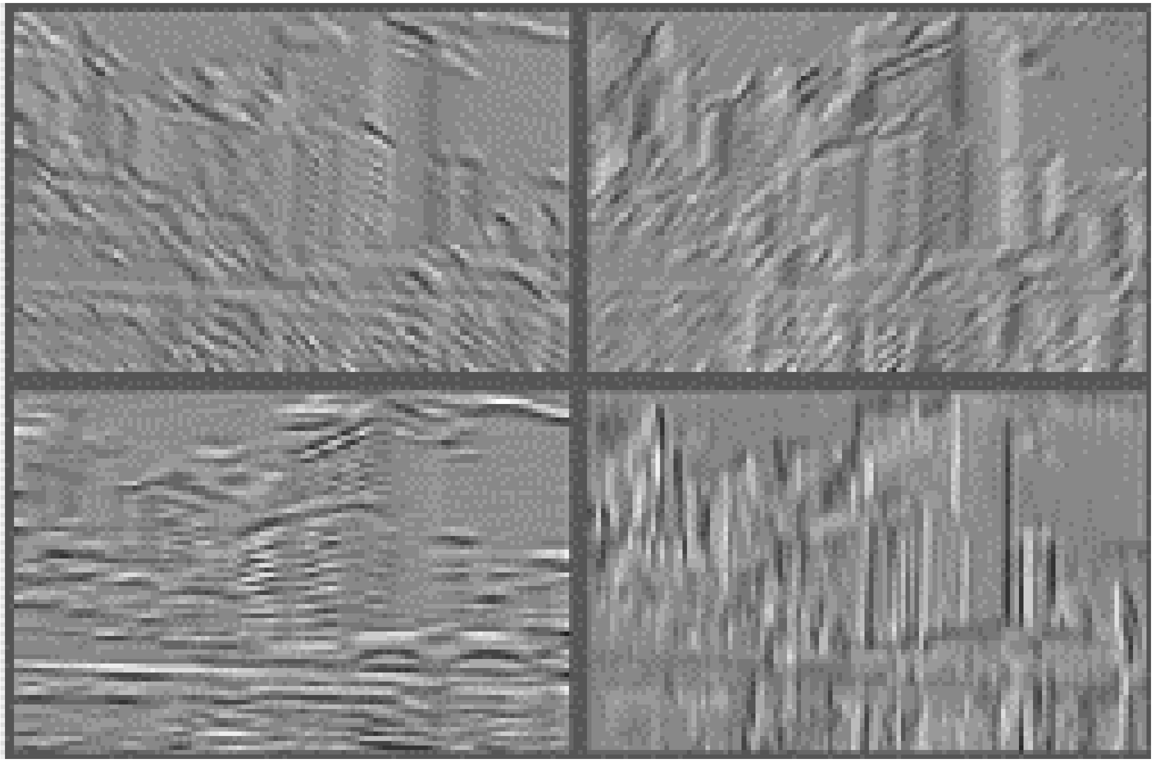

Feature Maps


Feature Maps
After Contrast
Normalization

credit: S. Lazebnik

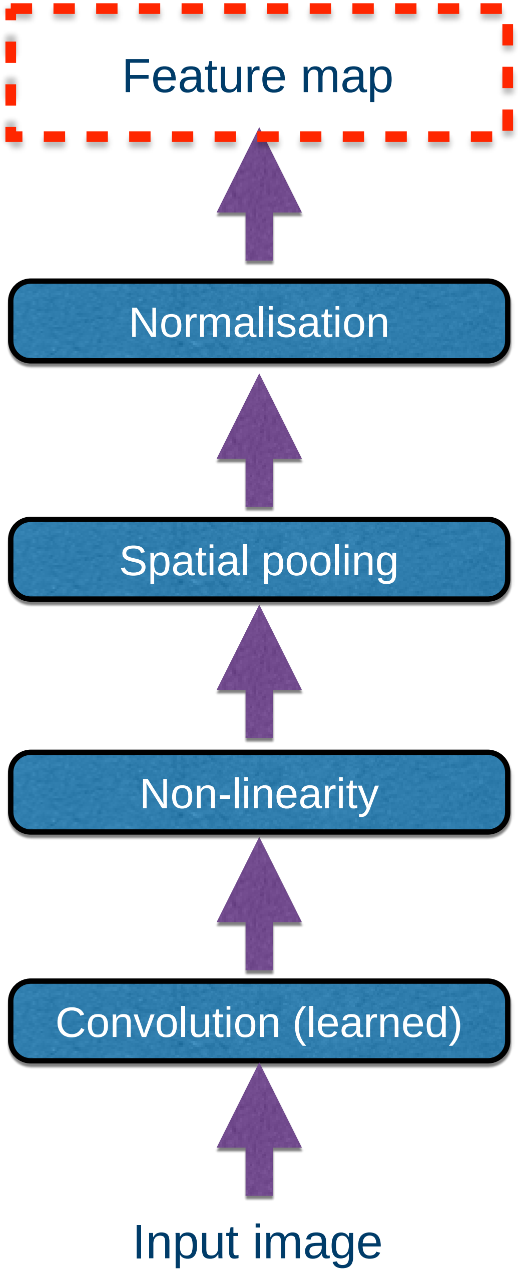


Convolutional neural net



Feature maps after contrast normalization

credit: S. Lazebnik



Example - Caffe Demos

The Caffe neural network library makes implementing state-of-the-art computer vision systems easy.

Classification

Click for a Quick Example

Maximally accurate | Maximally specific

cat	1.34462
domestic cat	1.32269
feline	1.26249
domestic animal	0.67113
carnivore	0.62083

CNN took 0.103 seconds.

Caffe Demos

The Caffe neural network library makes implementing state-of-the-art computer vision systems easy.

Classification

Click for a Quick Example

Maximally accurate | Maximally specific

macaw	3.83737
parrot	3.13682
bird	1.40822
lorikeet	0.21526
lory	0.21210

CNN took 0.067 seconds.

Caffe Demos

The Caffe neural network library makes implementing state-of-the-art computer vision systems easy.

Classification

Click for a Quick Example

Maximally accurate | Maximally specific

bridge	0.72819
structure	0.71525
geological formation	0.60429
suspension bridge	0.52708
pier	0.36455

CNN took 0.169 seconds.

Summary

Topics covered:

- Deep learning
- Artificial neural networks
- Convolutional neural networks.

Further reading:

- Szeliski, chapter 14
- Yann LeCun ,Yoshua Bengio & Geoffrey Hinton, “Deep learning”, Nature, Vol 521, 28. May 2015.

Software:

- Caffe
- MatConvNet (Matlab)
- …