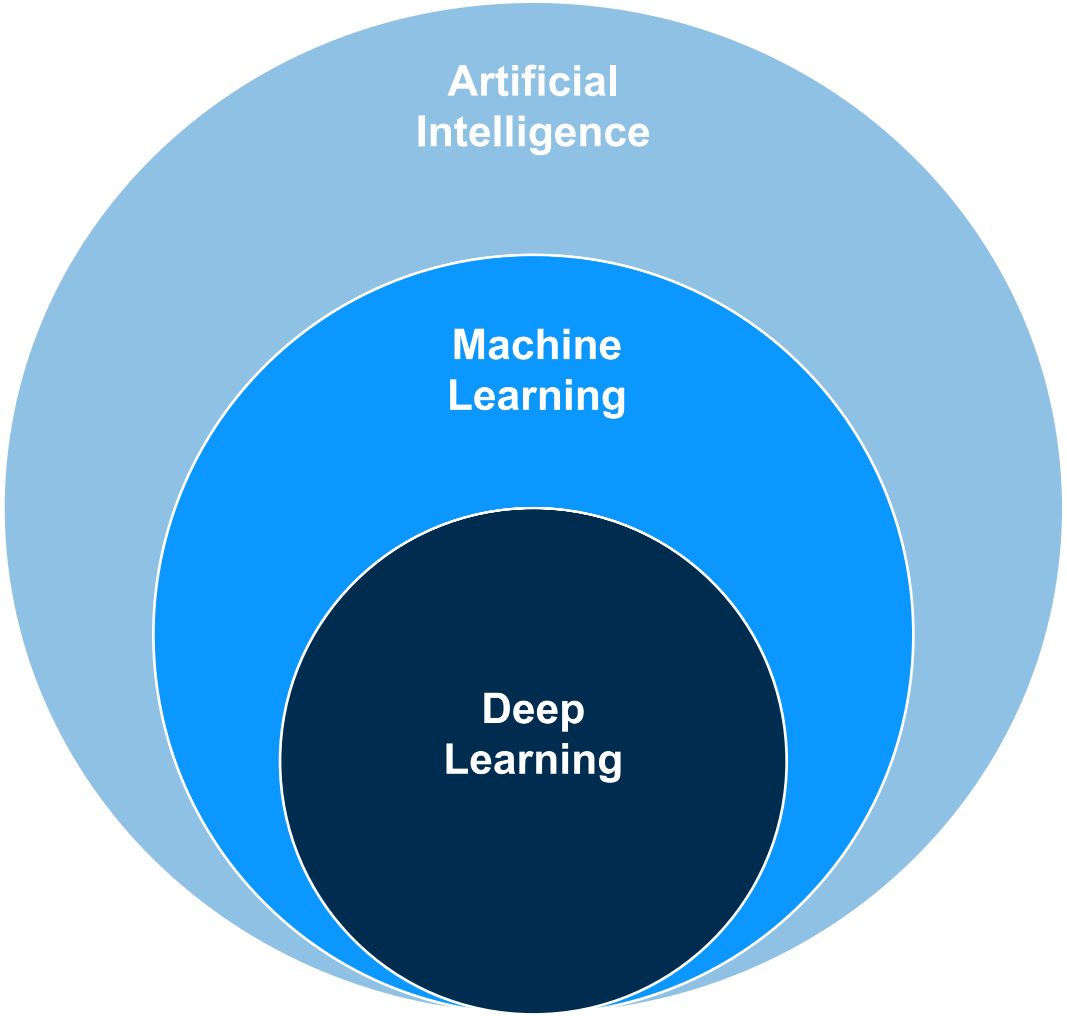
Introduction to Deep Learning

Idar Dyrdal, Sigmund Rolfsjord

Artificial Intelligence (AI)



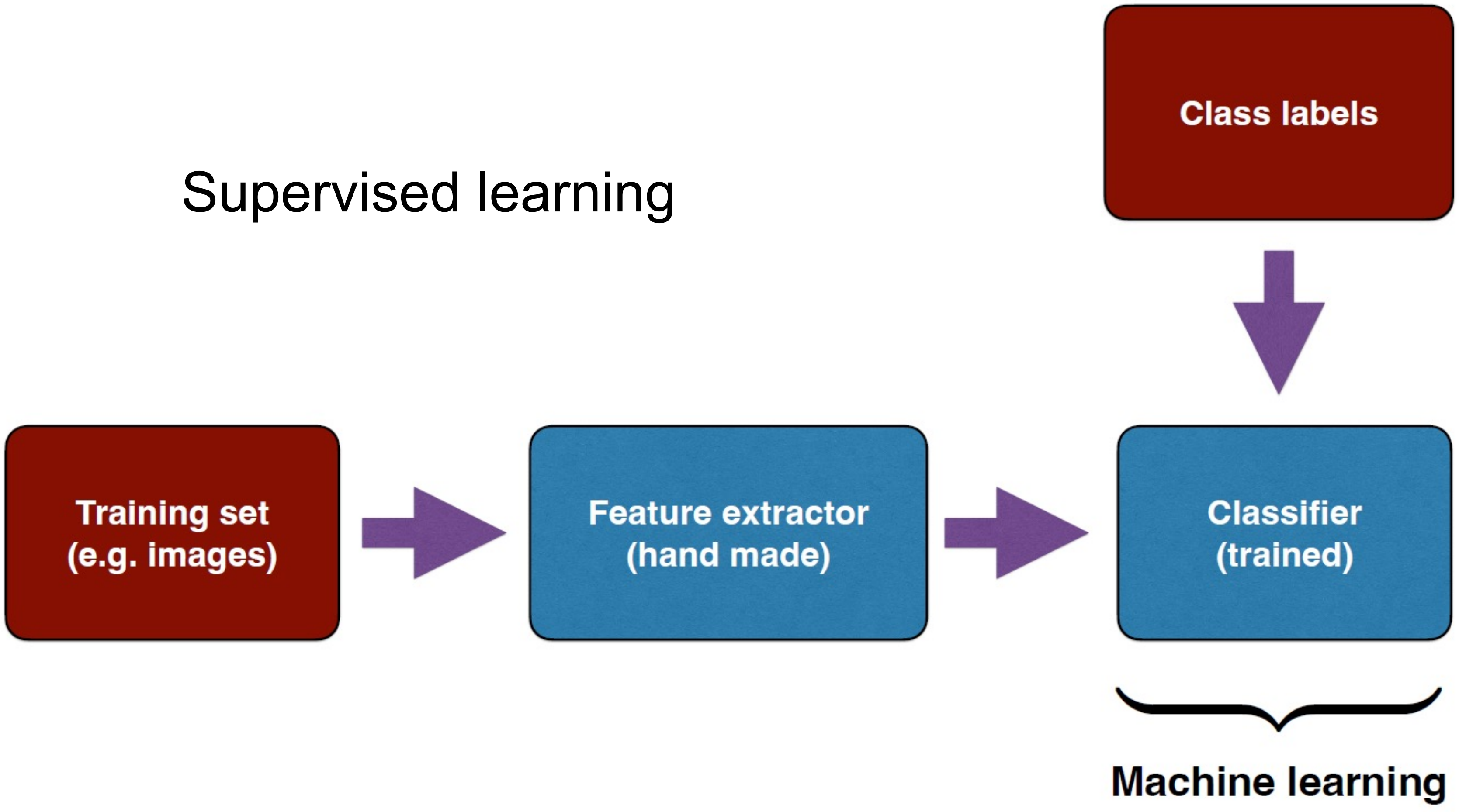
Ability to mimic human intelligence

Ability to mimic human intelligence by learning from experience

Ability to mimic human intelligence by «end-to-end» learning of large neural networks

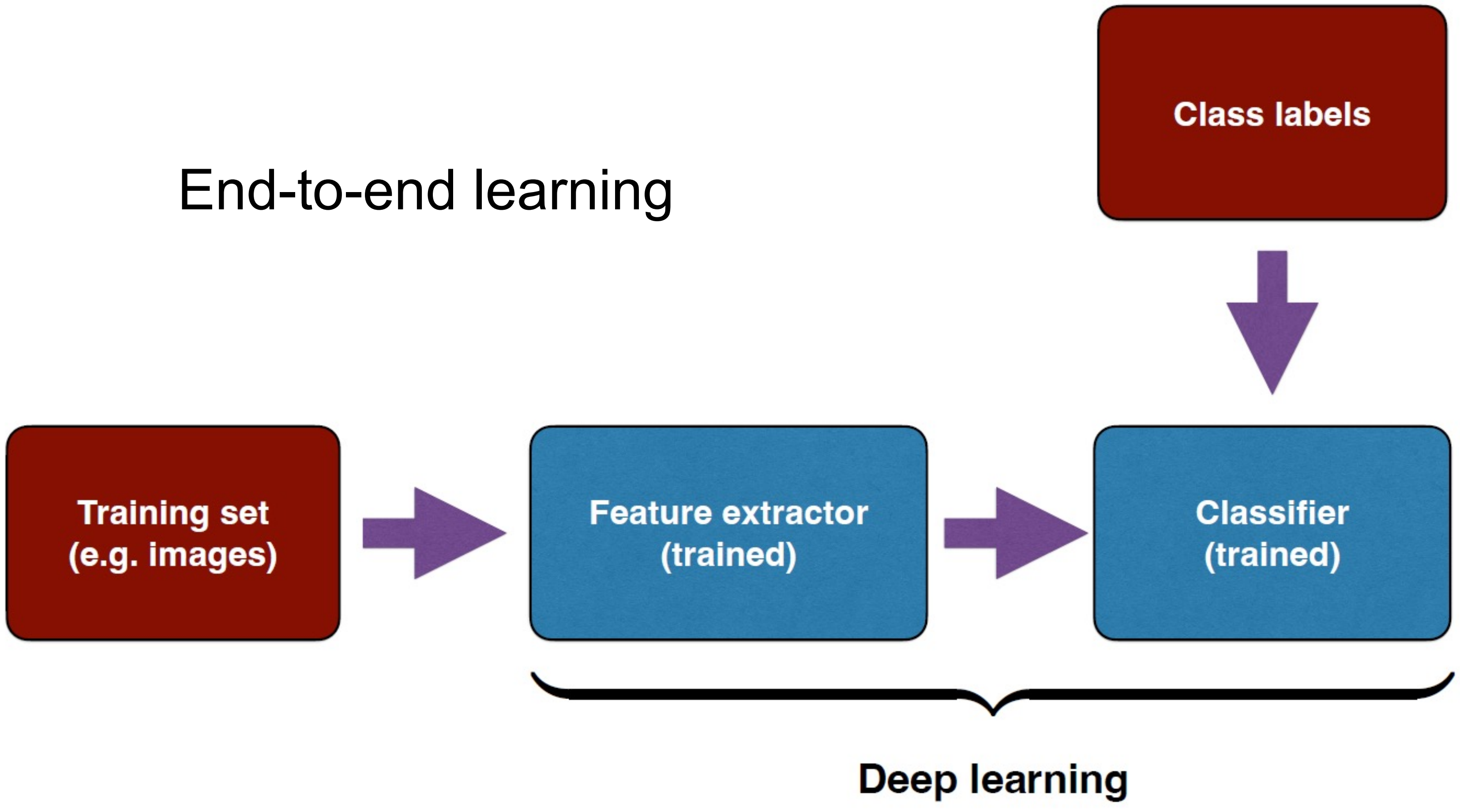
Classic Machine Learning

Supervised learning



Deep learning

End-to-end learning



Applications of Deep Learning

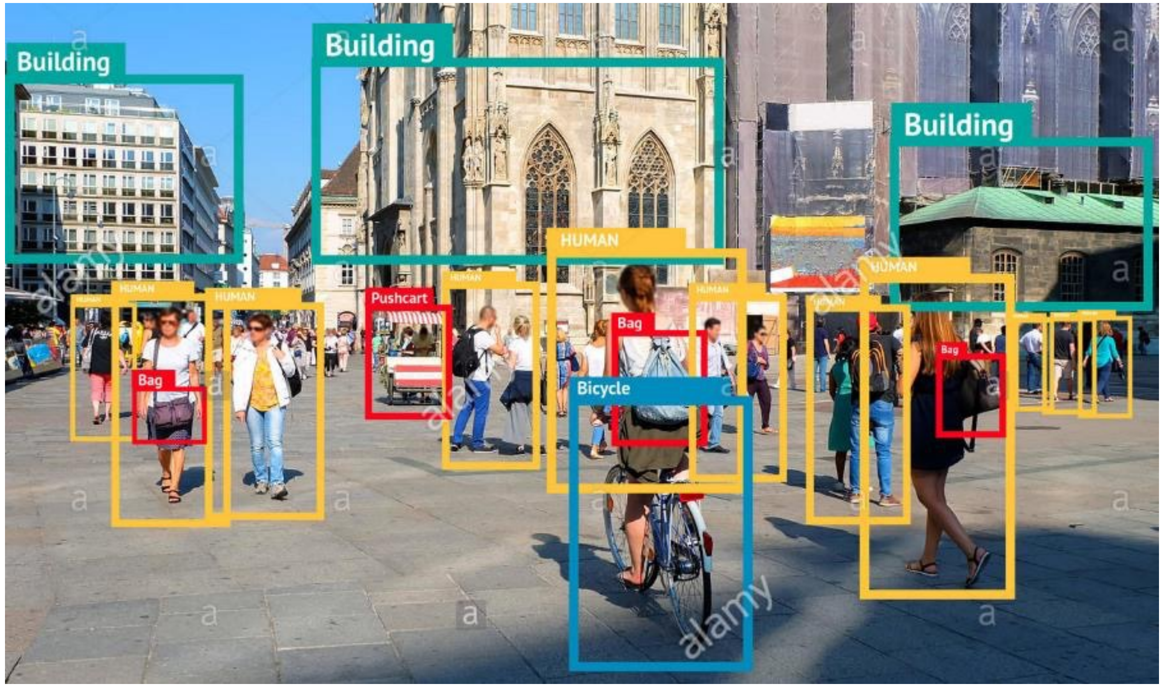


Foto: Alarmy

Object Recognition (Classification)

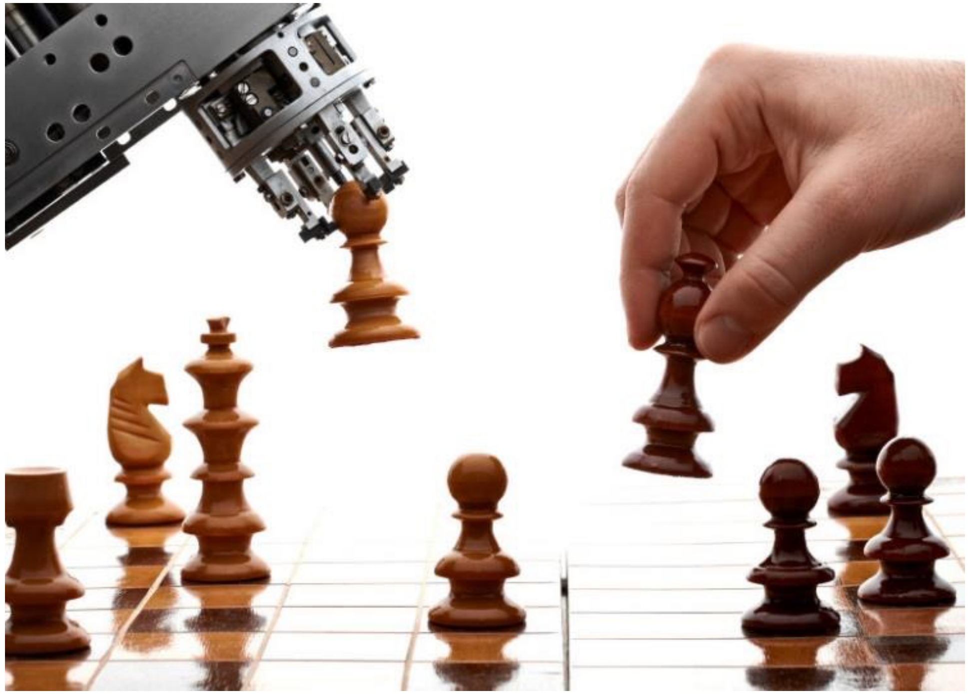
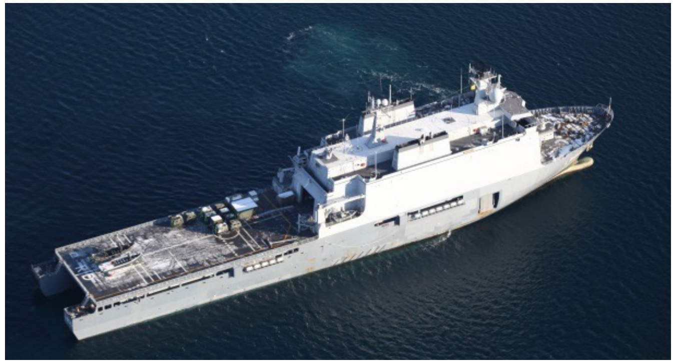
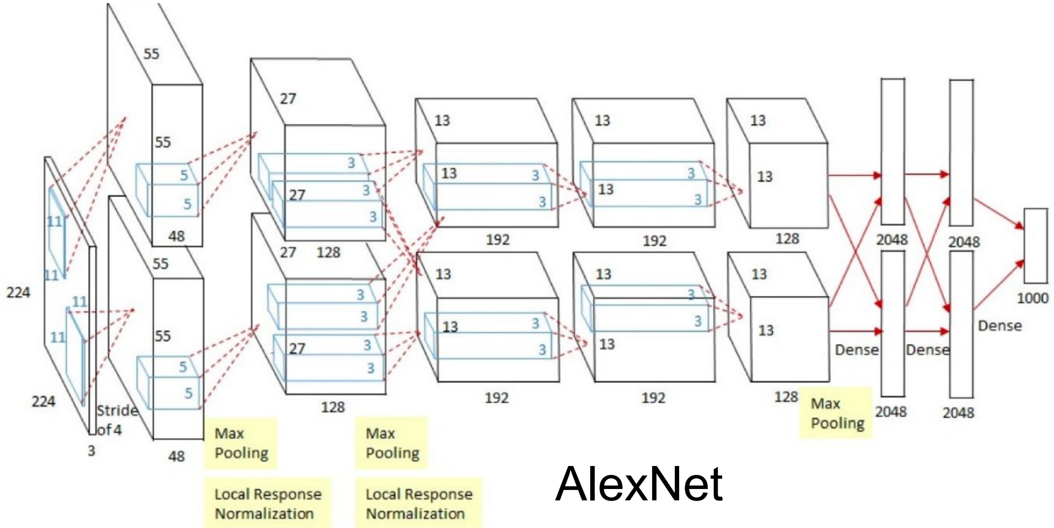


Foto: Augmentyka

Computer Chess

Deep Learning for Object Recognition





«Ship»

AlexNet

Millions of images

Millions of parameters

Thousands of classes

Deep Learning for Semantic Segmentation (SegNet)

http://mi.eng.cam.ac.uk/projects/segnet/

Artificial Neural Network (ANN)

Used in Machine Learning:

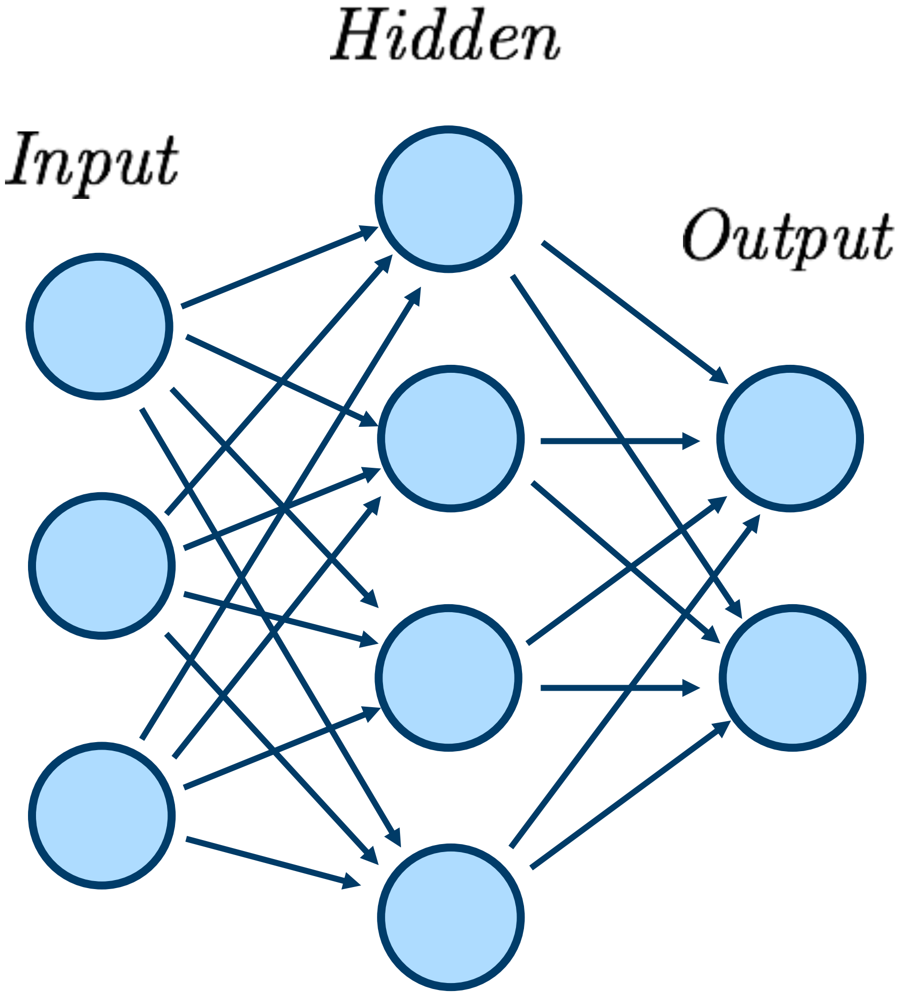
- Regression
- Classification
- Clustering
- …

Applications:

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

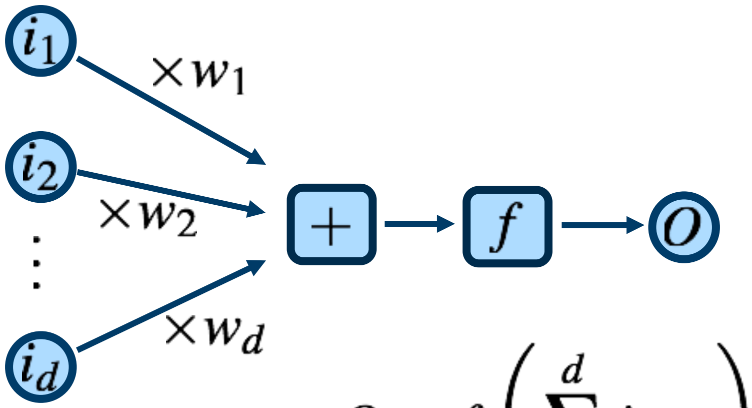
Network types:

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



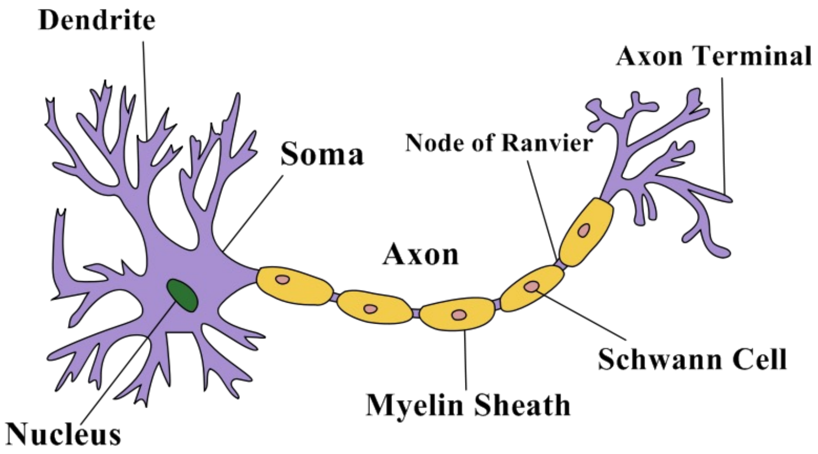
Feed-forward ANN (non-linear classifier)

Neural Networks



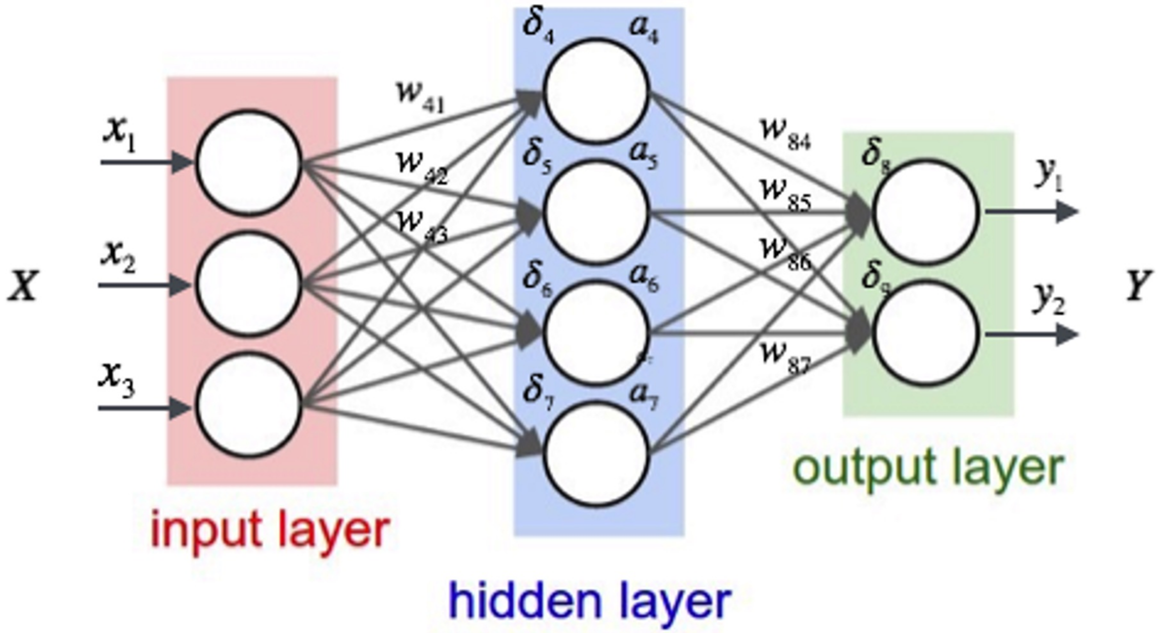
$$O = f\left(\sum_{k=1}^{d} i_k w_k\right)$$

Perceptron (Rosenblatt, 1957-59)



(Credit: Quasar Jarosz, English Wikipedia)

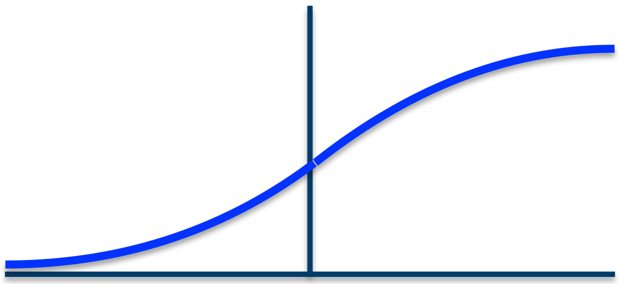
Classic neural networks were inspired by biological nerve cells (neurons)



Activation functions

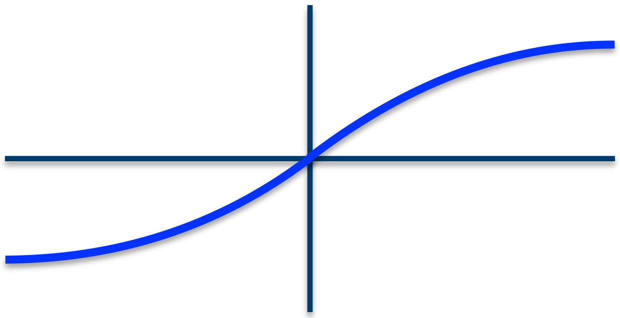
- Sigmoid (logistic function):

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



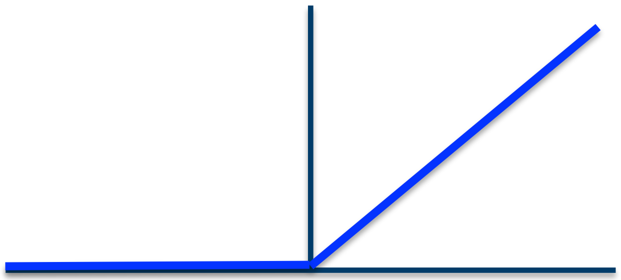
- Hyperbolic tangent:

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



- Rectified linear unit (ReLU):

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



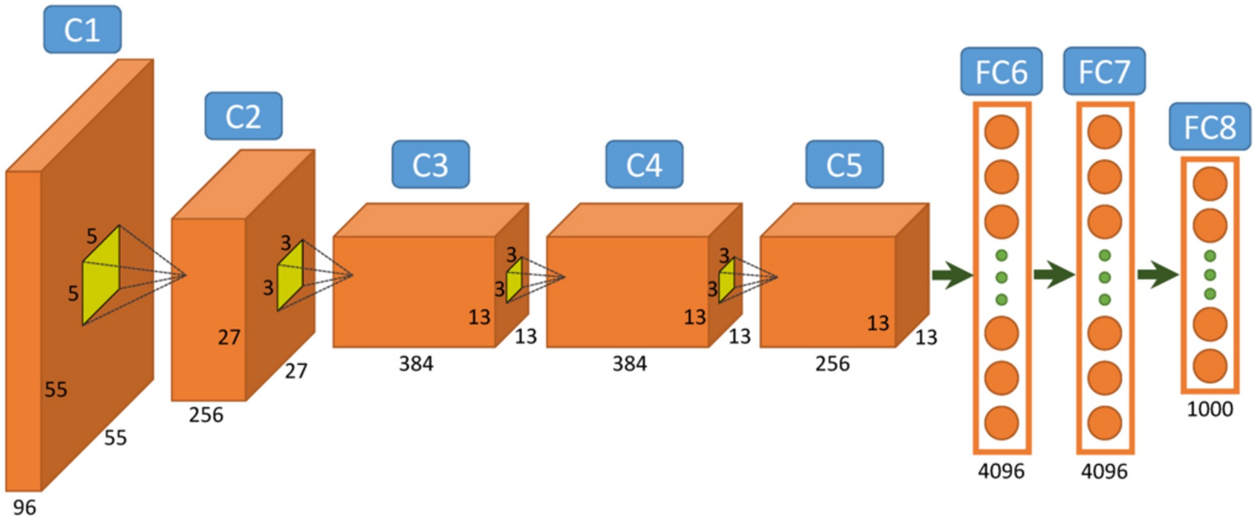
Deep learning

- Main goal is to approximate a function
- The approximation is improved by using many layers
- **Simplest form:** Matrix multiplications and non-linear functions
- **Deep learning:** Matrix multiplications often replaced by **convolutions**

$$f = Wx$$

$$f = W_2 \max(0, W_1 x)$$

$$f = W_3 \max(0, W_2 \max(0, W_1 x))$$



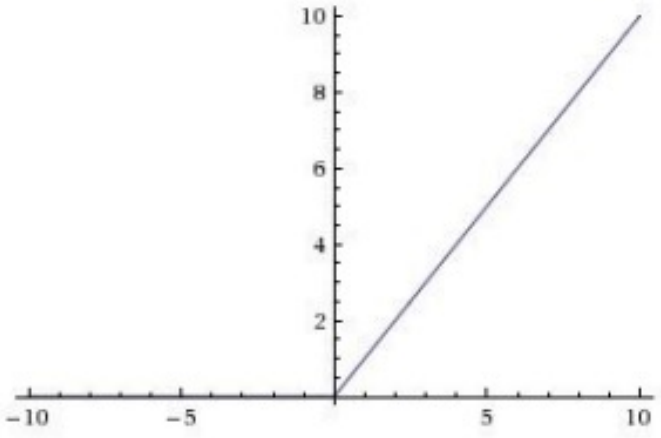
Deep learning

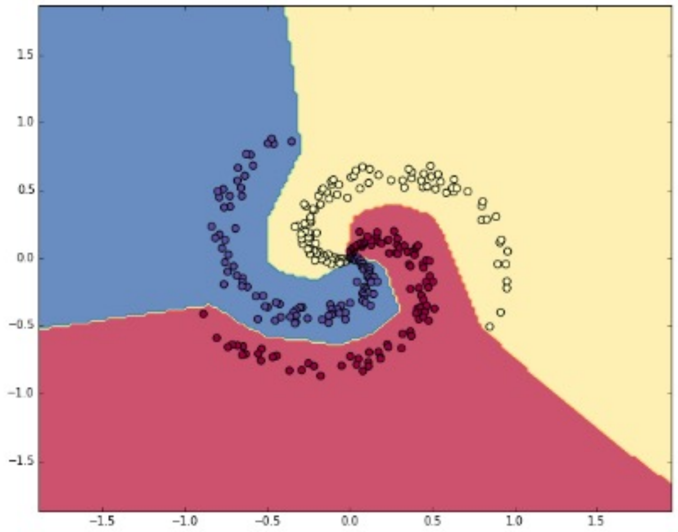
- Non-linear functions are important
- The ReLu function max(0, x) is commonly used, providing good gradients and efficient training og networks

$$f = Wx$$

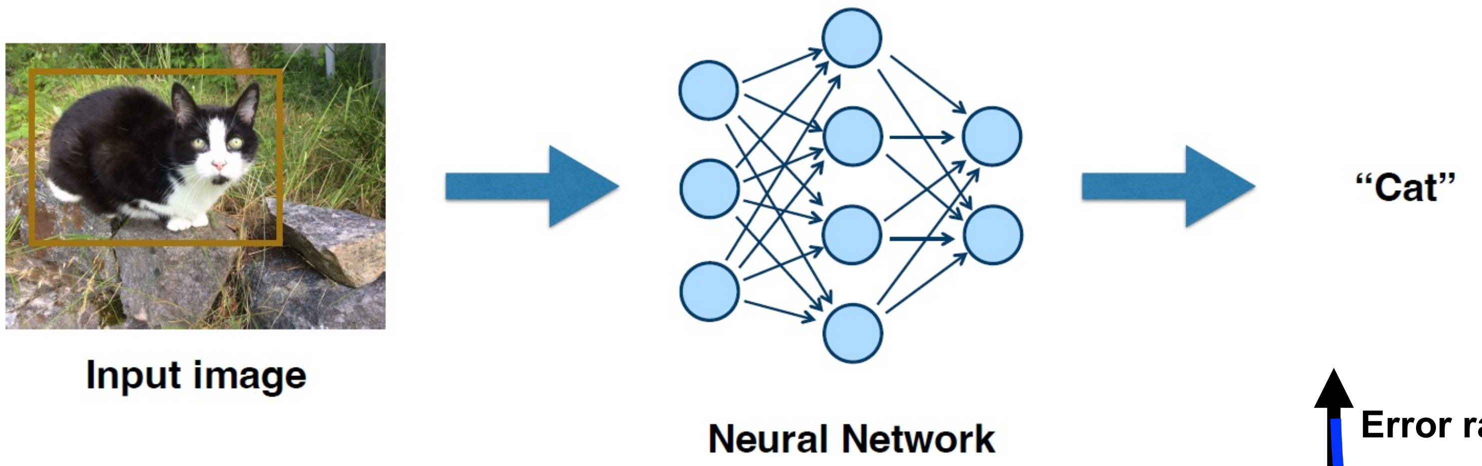
$$f = W_2 \max(0, W_1 x)$$

$$f = W_3 \max(0, W_2 \max(0, W_1 x))$$



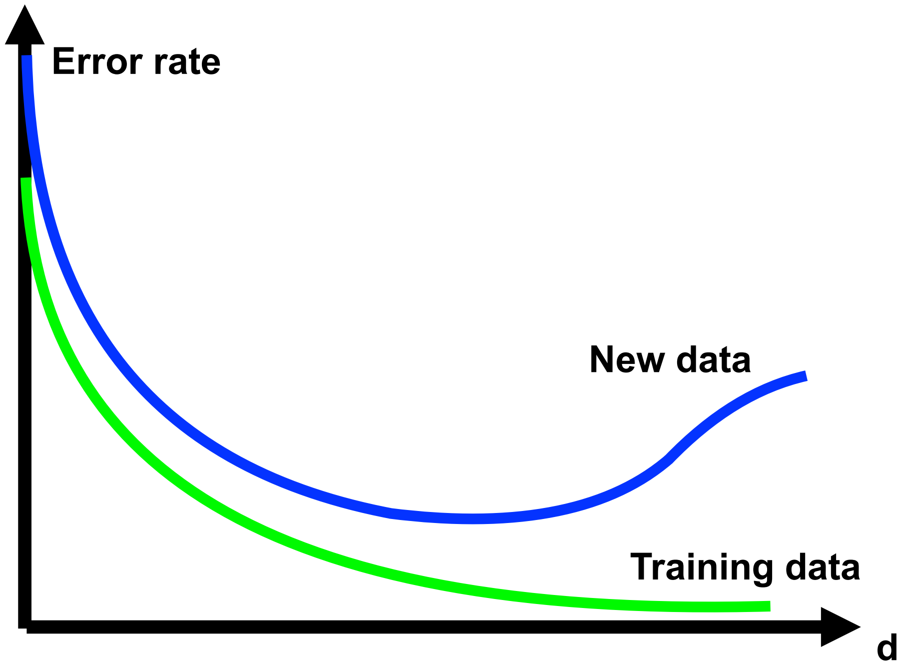


Object Classification with Neural Networks

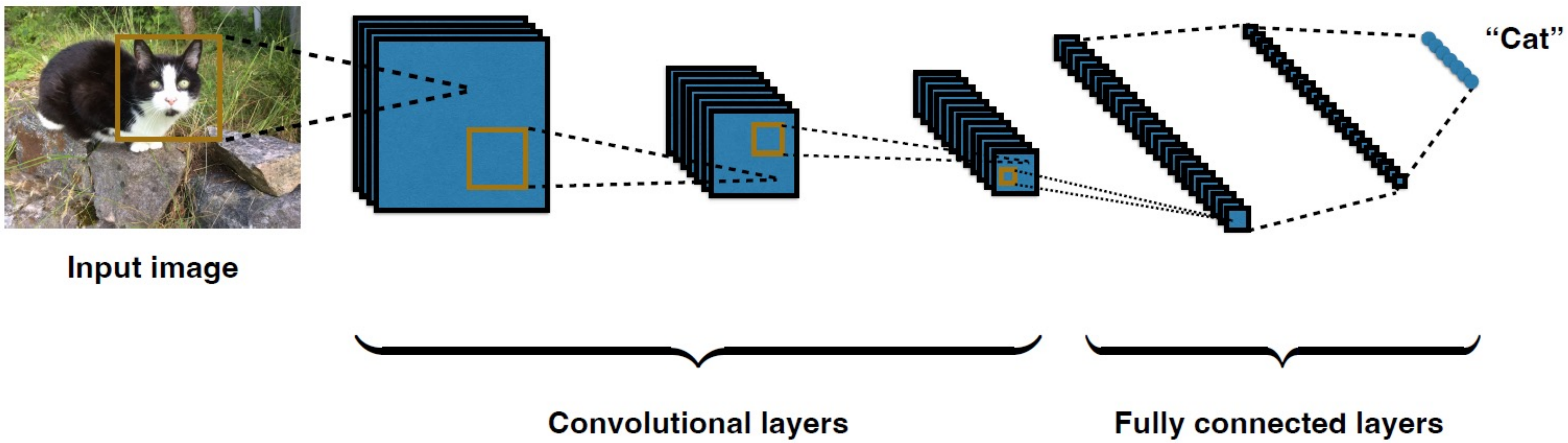


Problems:

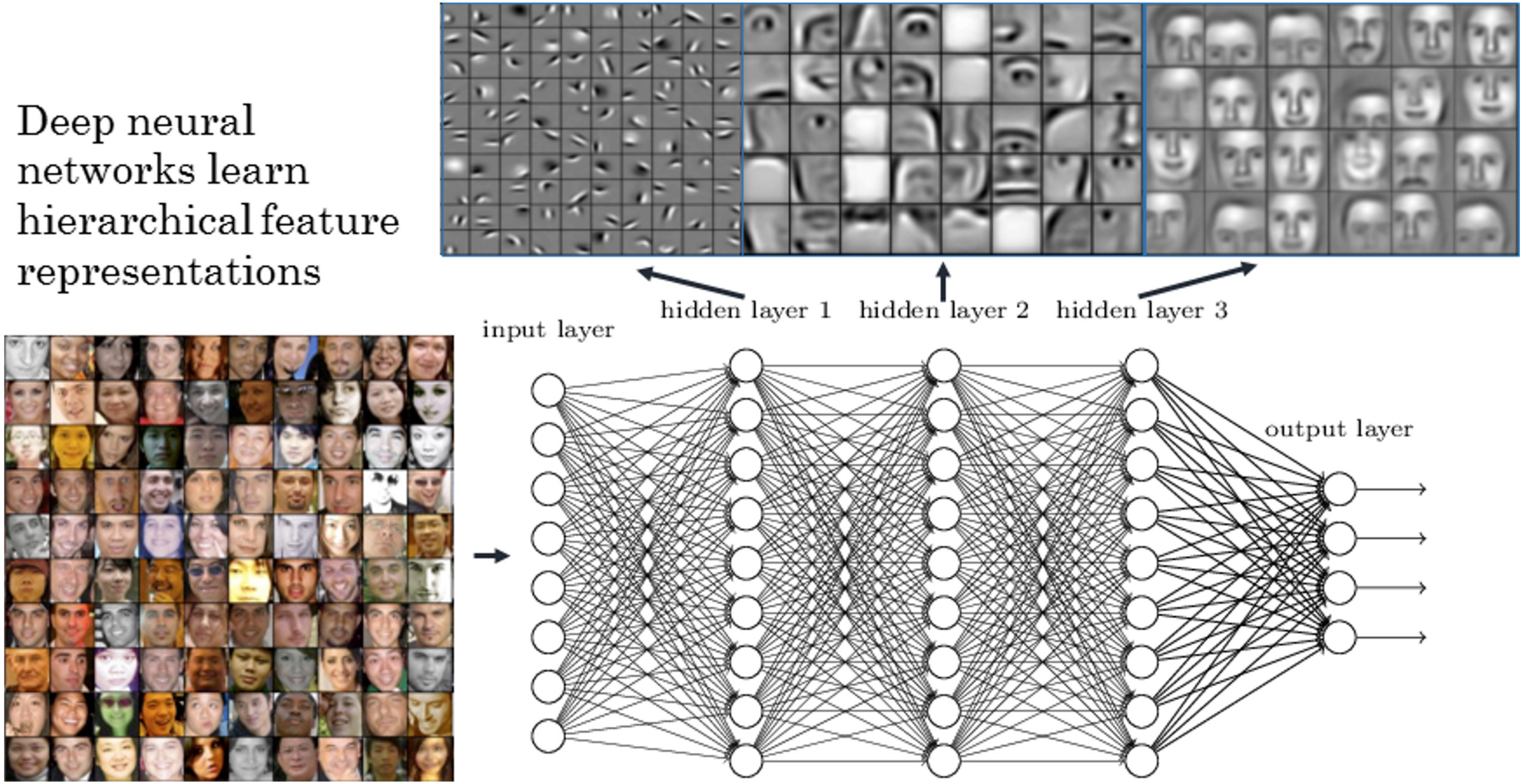
- Curse of dimensionality (too many input samples)
- Too many connections (weights) in a fully connected network
- Input pattern different if object has moved



Convolutional Neural Network



Learning of feature representations



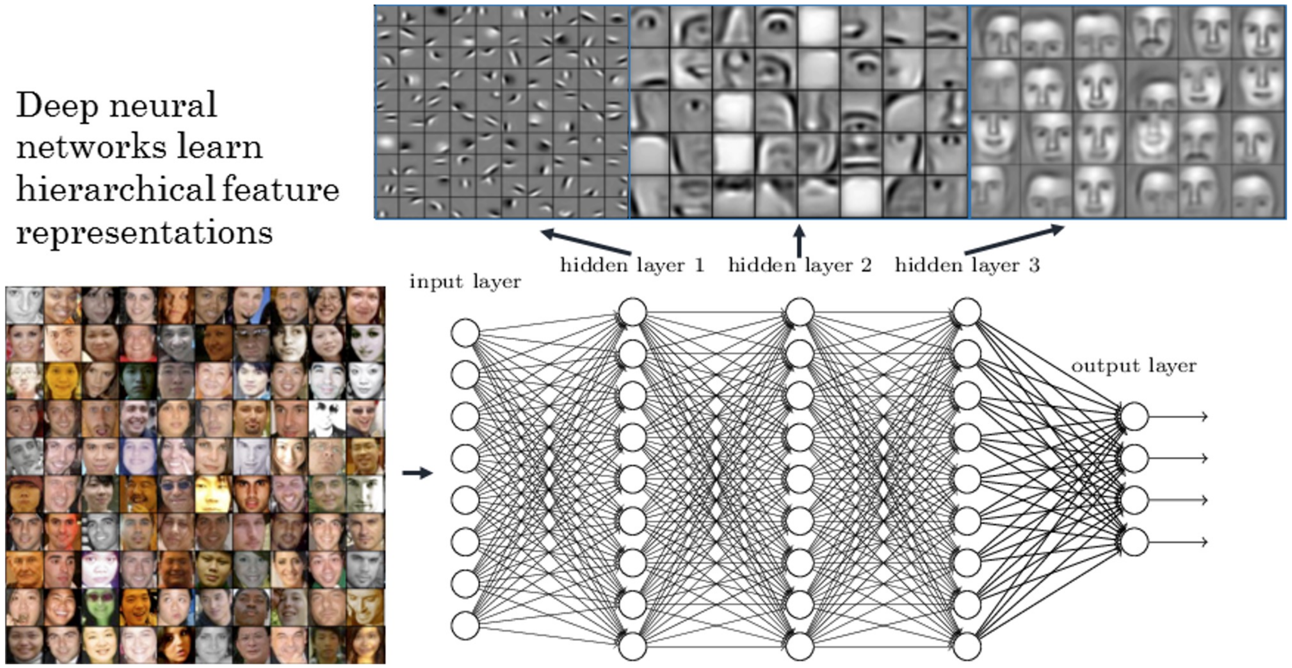
How is deep learning possible?

- No feature extraction for compression of raw data (i.e. the input image)
- Many more parameters than training examples!
- How can we reduce the need for training examples?
- How can overfitting be prevented?

Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [29]	224×224	11.2 M	1.94 B	74.8	92.2
NASNet-A (5 @ 1538)	**299×299**	**10.9 M**	**2.35 B**	**78.6**	**94.2**
Inception V3 [59]	299×299	23.8 M	5.72 B	78.0	93.9
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [57]	299×299	55.8 M	13.2 B	80.4	95.3
NASNet-A (7 @ 1920)	**299×299**	**22.6 M**	**4.93 B**	**80.8**	**95.3**
ResNeXt-101 (64 x 4d) [67]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [68]	331×331	92 M	34.7 B	81.3	95.8
DPN-131 [8]	320×320	79.5 M	32.0 B	81.5	95.8
SENet [25]	**320×320**	**145.8 M**	**42.3 B**	**82.7**	**96.2**
NASNet-A (6 @ 4032)	**331×331**	**88.9 M**	**23.8 B**	**82.7**	**96.2**

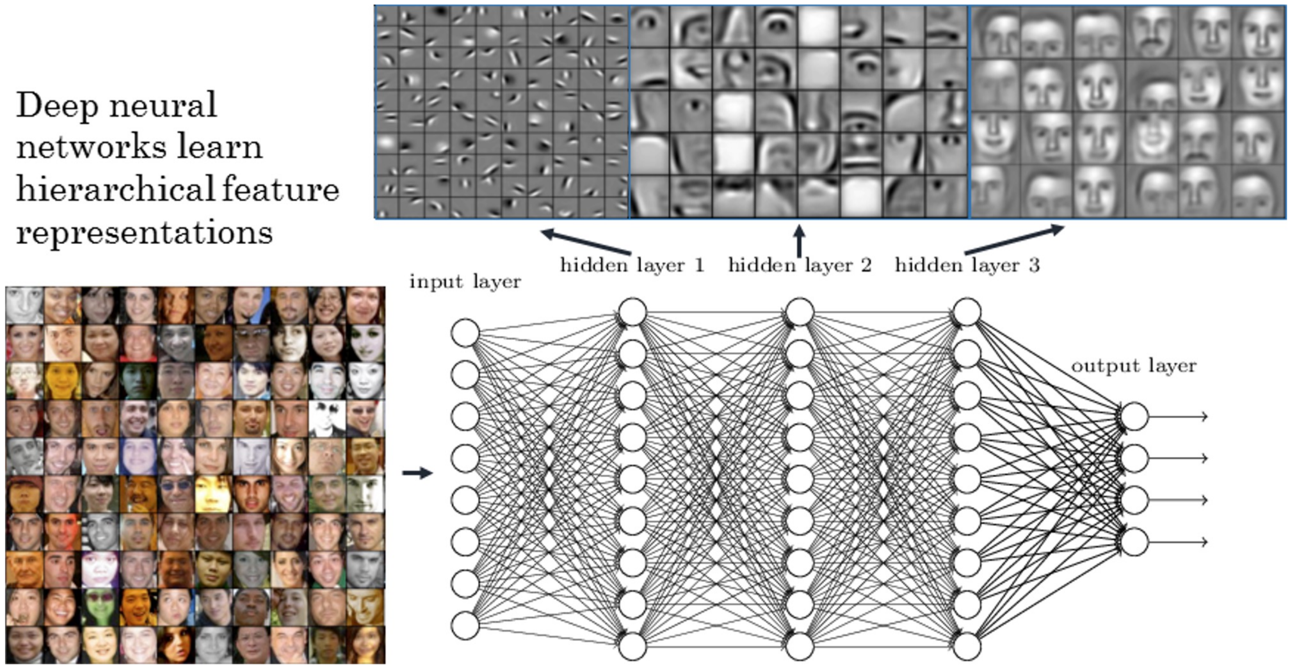
How is deep learning possible?

- Depth off the network (many layers)
- Use of hierarchical representations
- Large training set



How is deep learning possible?

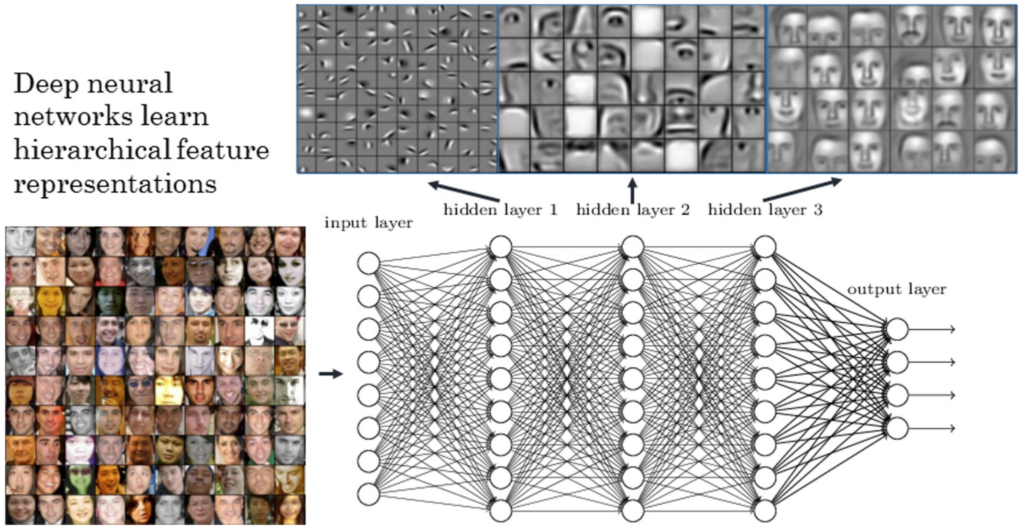
- Sharing of representations for several classes (blobs, edges etc.)
- Reusing training data for multiple classes
- Blobs and edges can be combined to higher order features (e.g. eyes, nose etc.) in next level
- No need for training data spanning all combinations

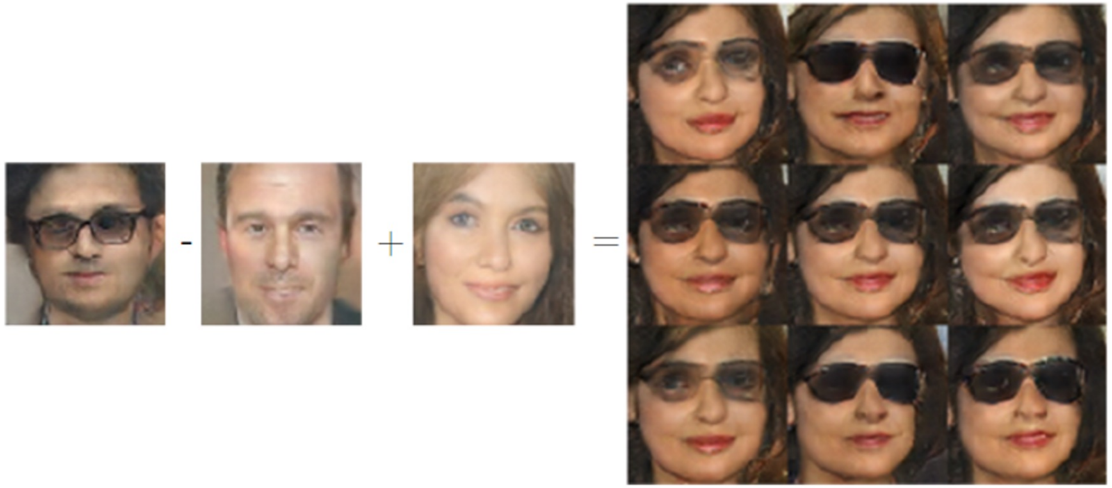


How is deep learning possible?

- It has been found that some neurons (filters) will react to eyes in general, both human eyes and eyes from various animals

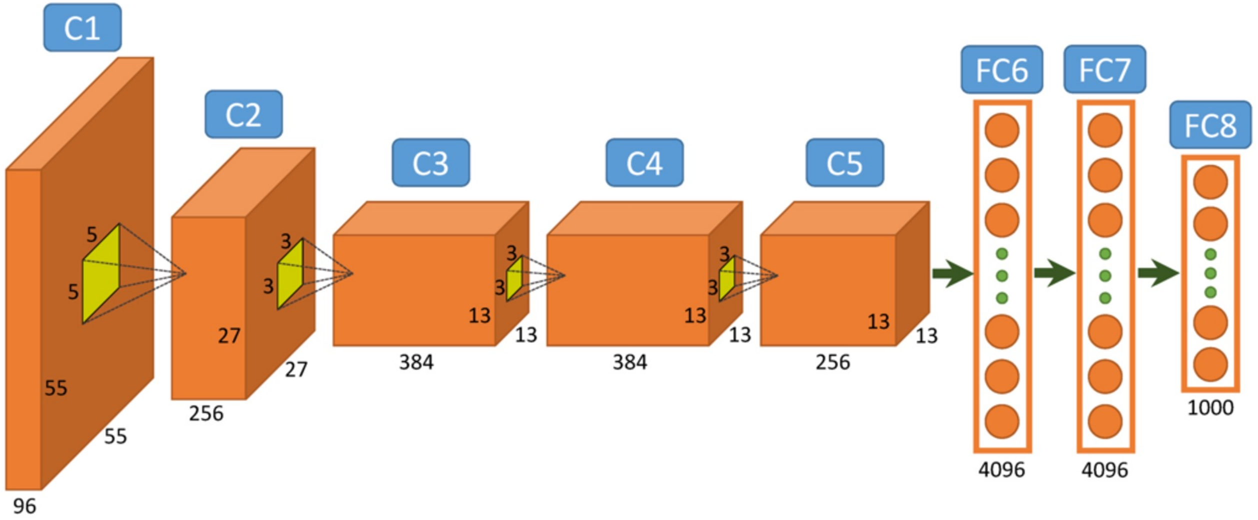
- Studies have shown that deep networks may learn isolated concepts (e.g. spectacles)



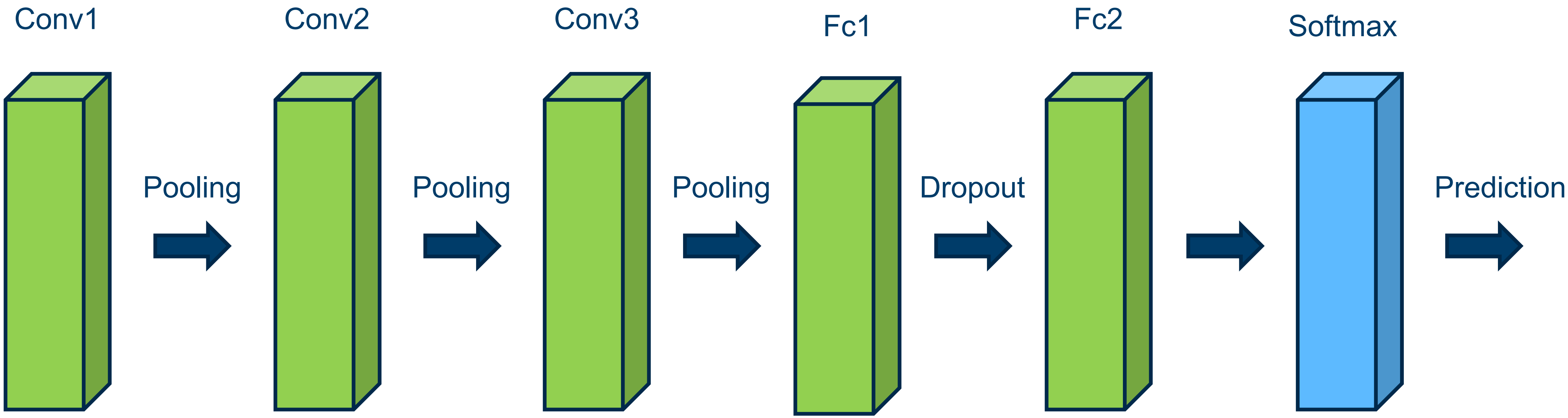


Ways to prevent overfitting

- Regularization and weight decay
- Dataset Augmentation
- Dropout
- Batch normalization



Pooling, dropout and softmax



Training of weights in a neural net (Gradient decent)

- Specify a loss function, i.e a measure of distance between the outcome f(**x**) of the network and the target value y

  L = (f(**x**) - y)² e.g. $f = Wx$

- Compute the gradient using the chain rule

  $$\frac{\partial L}{\partial W} = 2(f(\mathbf{x}) - y)\frac{\partial f}{\partial W}$$

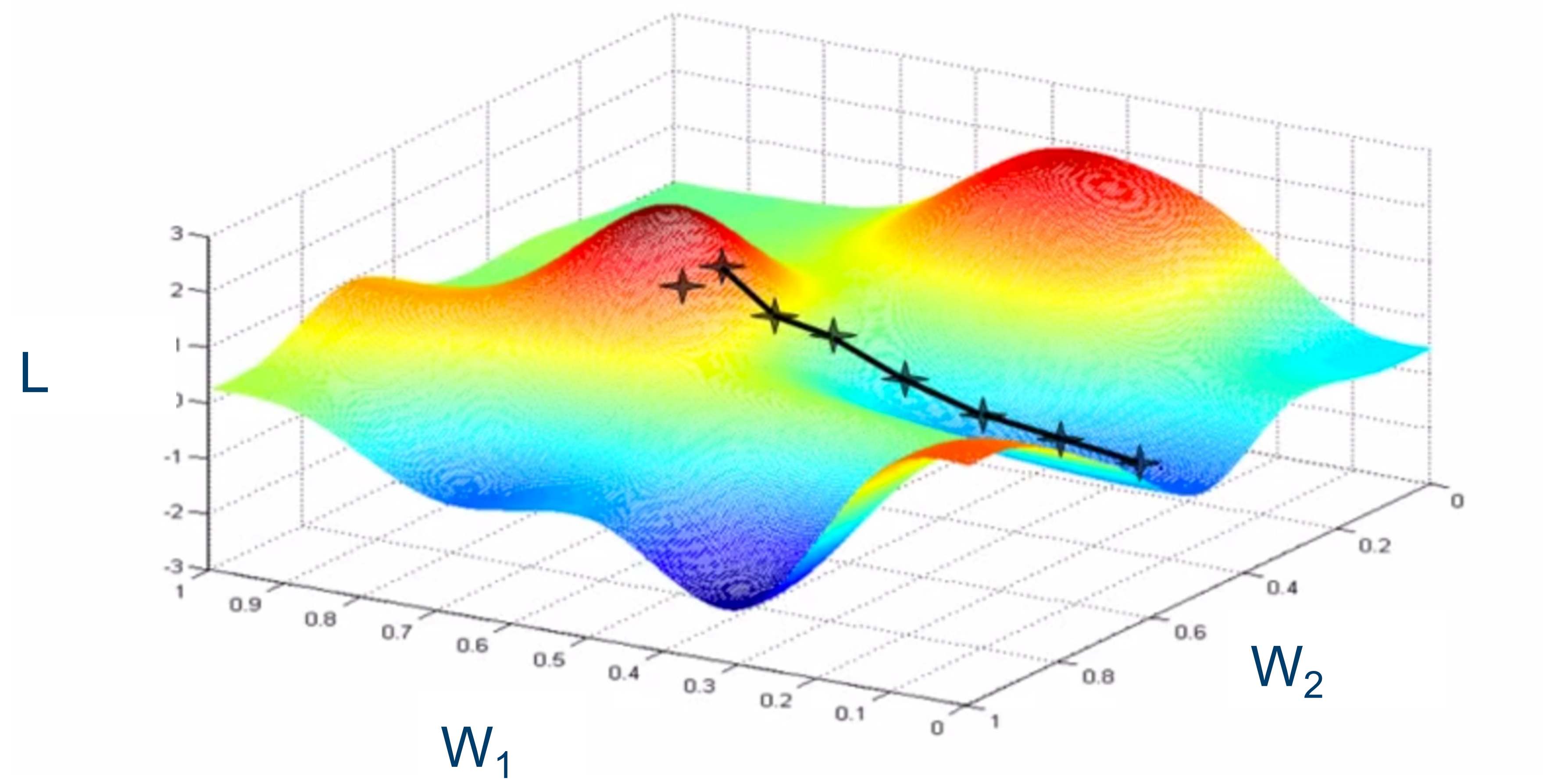
- The gradient is usually computen for a «mini batch» of training samples to reduce noise

- Update the weights (steepest descent)

  $$W \leftarrow W - \alpha \nabla W$$

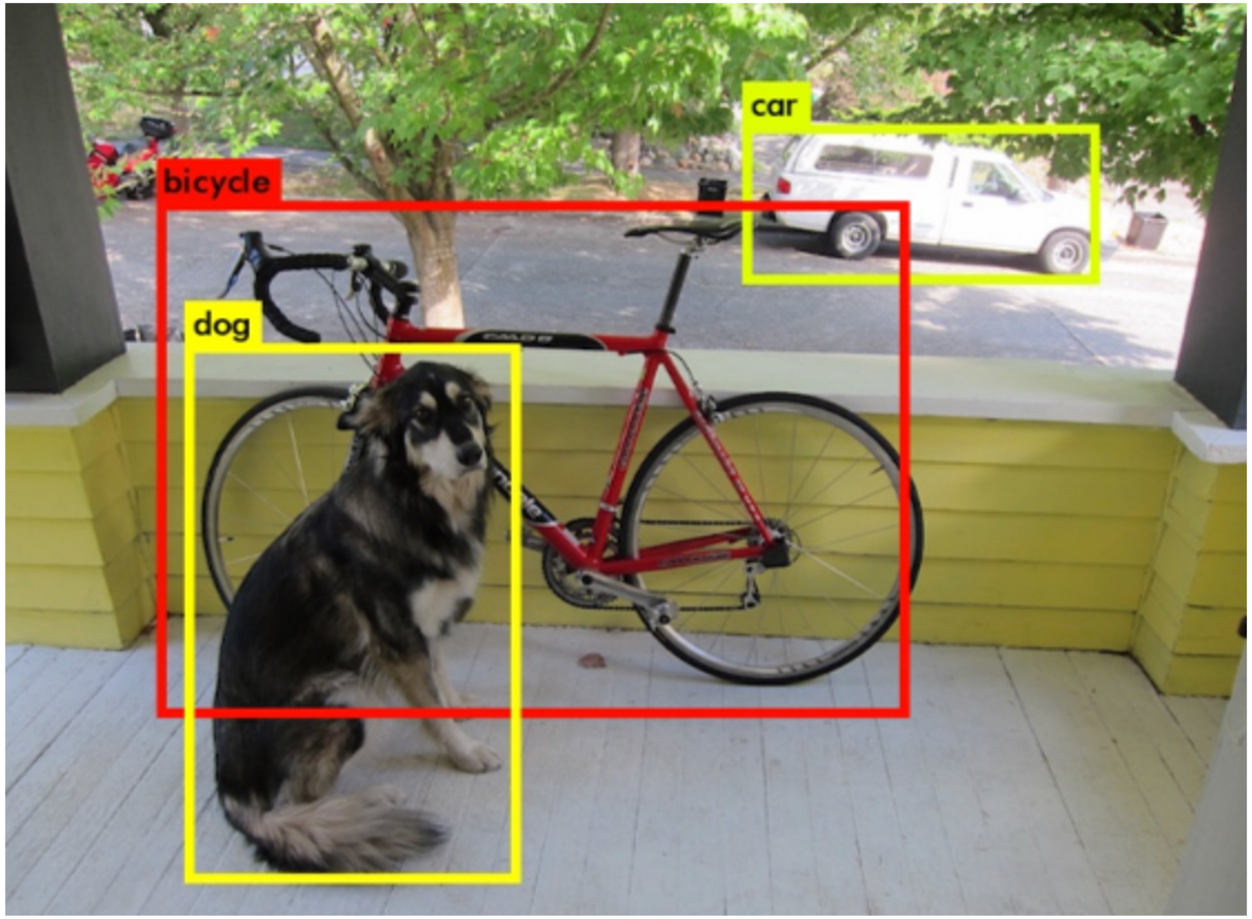
- «Momentum» can be introduced in the update to avoid stalling when reaching a «flat spot» in search space

Gradient decent



Object detection with deep learning

Find and classify objects separatly (very common application)

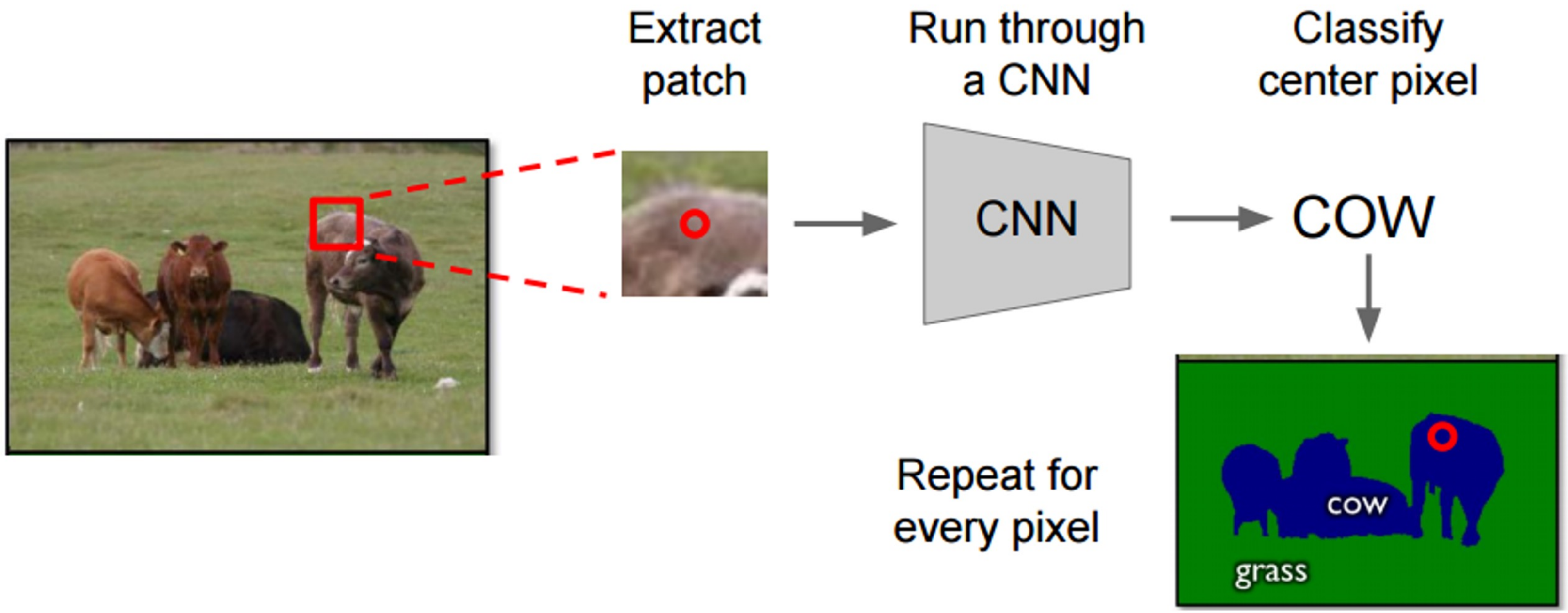


Segmentation

Associate single pixels with objects

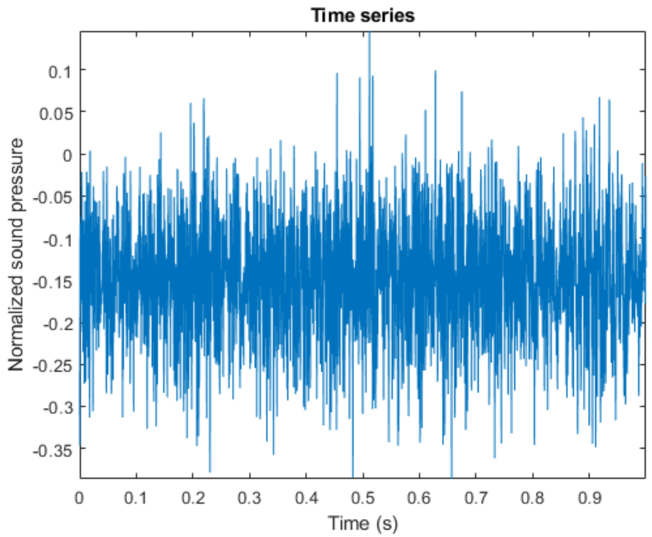
MSCOCO is a standard benchmark

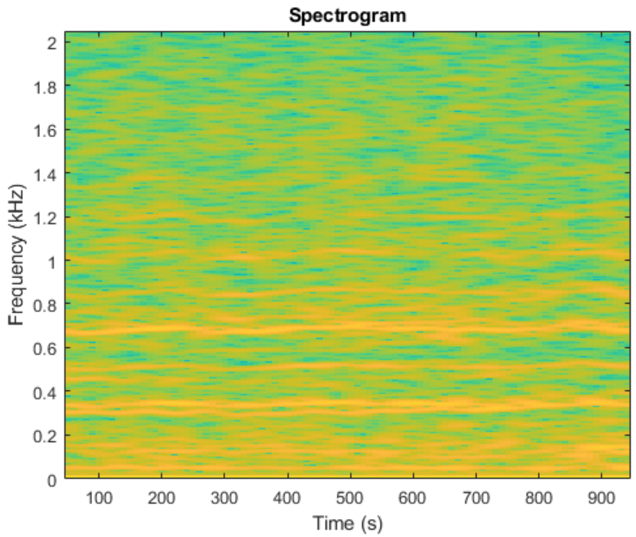
Segmentation



Pixel-by-pixel classification

Audio Classification with Deep Learning





Input to deep neural network

Summary

Deep learning:
- Perceptrons and neural networks
- Deep networks
- Backpropagation
- Examples of applications

Recommended reading:
- Szeliski 5.3