POLITÉCNICA

BIT
Biomedical Image Technology

Biomedical Imaging Technology (BIT) www.die.upm.es/im
Departamento de Ingeniería Electrónica, ETSIT-UPM

From Machine Learning to Deep Learning

David Bermejo Peláez

María Jesús Ledesma Carbayo

Temas Avanzados en Señales e Imágenes Médicas

Máster en Ingeniería Biomédica

Introduction: What is Machine Learning?

- Machine learning studies computer algorithms for learning to perform a particular task.
- We are interested in learning to complete a task, or to make accurate predictions.
- The learning that is being done is always based on some example observations or data or past experience.

- So in general, machine learning is about learning to do better in the future based on what was experienced in the past.

- In other words, the goal is to design learning algorithms that do the learning automatically without human intervention or assistance.

- Examples of use:
  - Computer-aided detection/diagnosis
  - Medical image reconstruction
  - Multi-modality fusion for diagnosis, image analysis and image guided interventions
  - etc

Introduction: What is Machine Learning?

- **Supervised learning (predictive task)**
  - Training data includes desired outputs.
  - The algorithm generates a function that maps inputs to desired outputs.
  - E.g. Classification problem
- **Unsupervised learning (descriptive task)**
  - Training data does not include desired outputs.
  - The algorithm models a set of inputs.
  - E.g. Clustering
- **Semi-supervised learning**
  - Training data includes a few desired outputs.
  - Combines both labeled and unlabeled examples to generate an appropriate function or classifier.

Classification

- Suppose we are given a training set of N observations:

  $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n), x_i \in \mathbb{R}^d, y_i \in \{-1,1\}$

  A classification problem aims at estimating $f(x)$ from this training set such that:

  $f(x_i) = y_i$

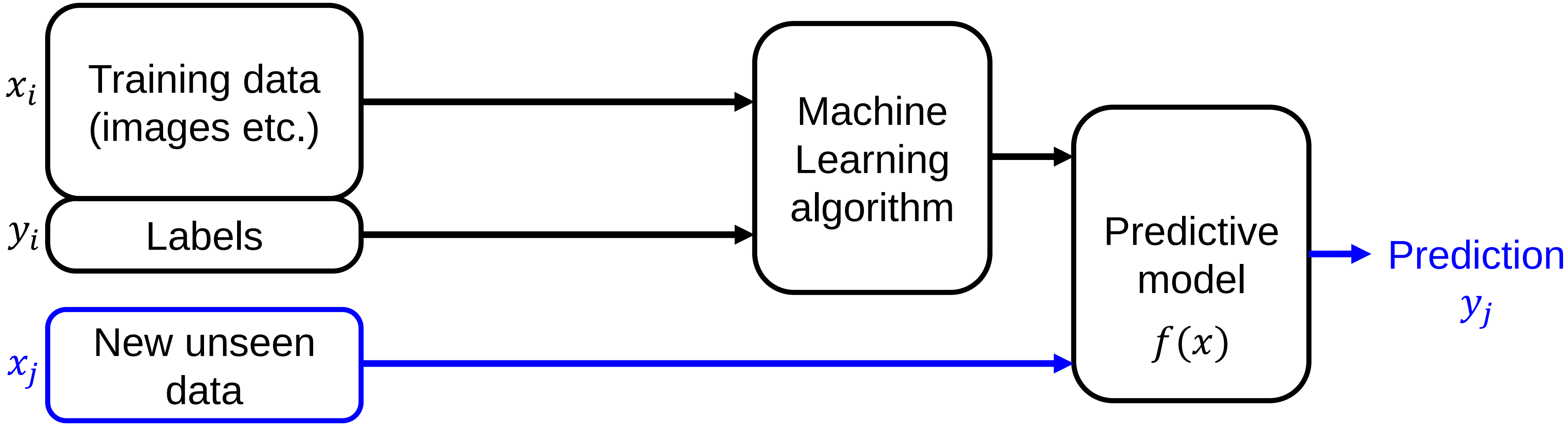


Diagram of a typical classification problem

Classification

- Suppose we are given a training set of N observations:

  $(x_1, \dots, x_n)$ and $(y_1, \dots, y_n), x_i \in \mathbb{R}^d, y_i \in \{-1,1\}$

  Classification problem aims at estimating $f(x)$ from this training set such that:

  $f(x_i) = y_i$

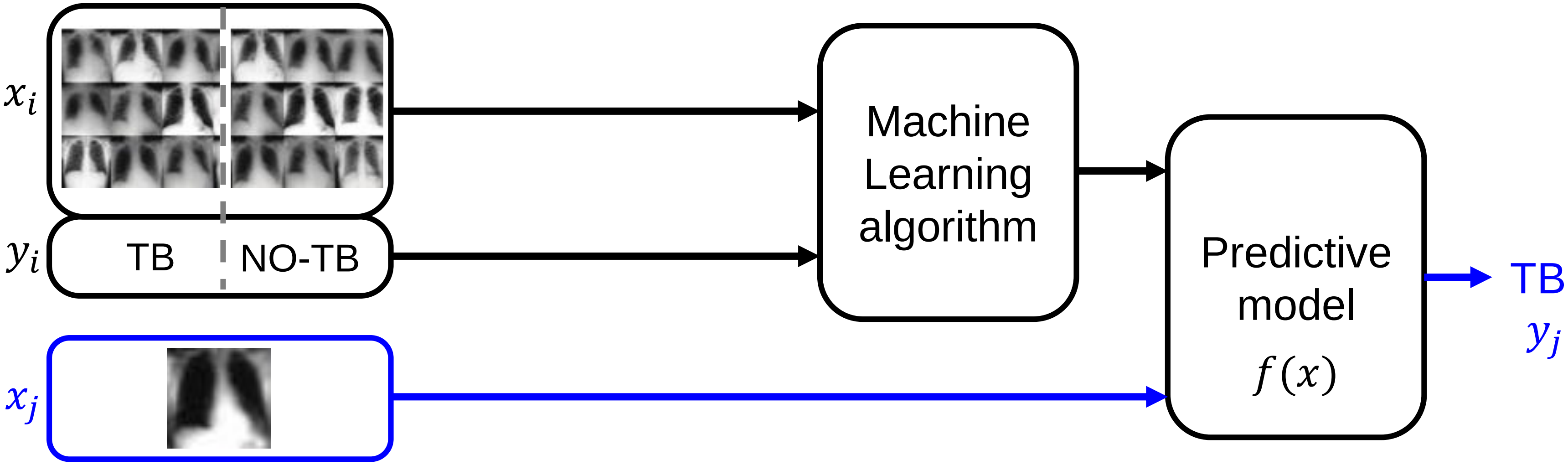
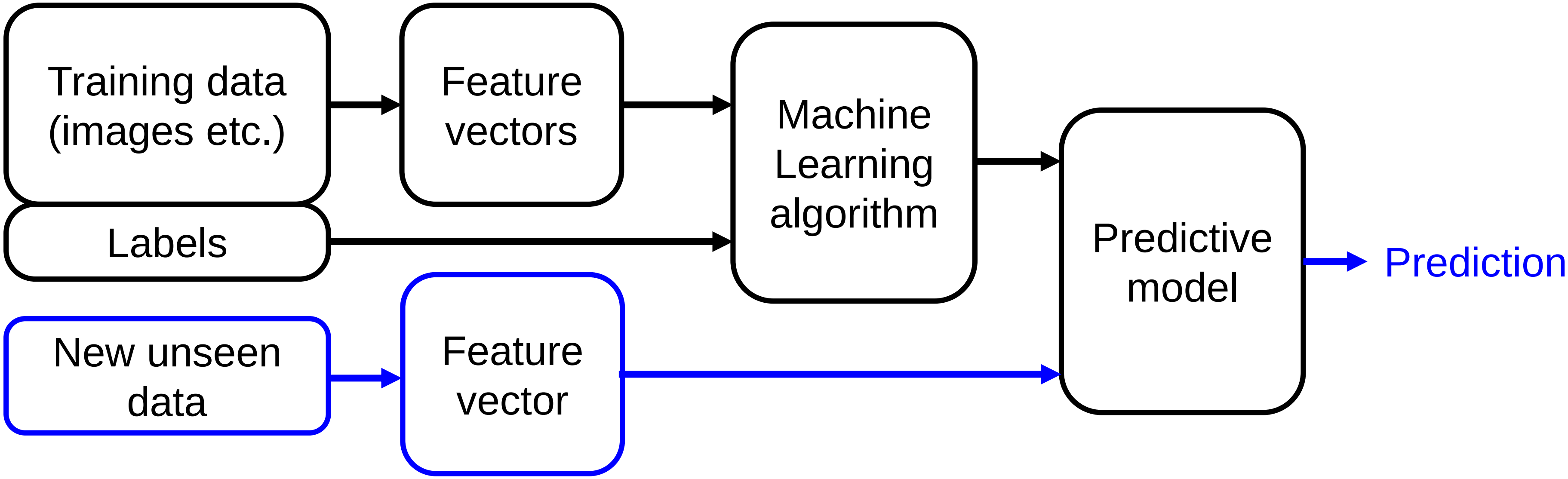


Diagram of a typical classification problem

Feature Extraction

- In practice, the function $f(x)$ that maps input x to output y is a high-complex non-lineal function very hard to be learnt.
- The first step to use this algorithms is to define and extract some features or descriptors that represent as faithfully as possible the given inputs in order to reduce the dimensionality.

- In this way, the algorithm will learn a more simple function $f(x)$ that maps the features to output values.
- The correct definition of this features is usually complex, costly and no trivial task.

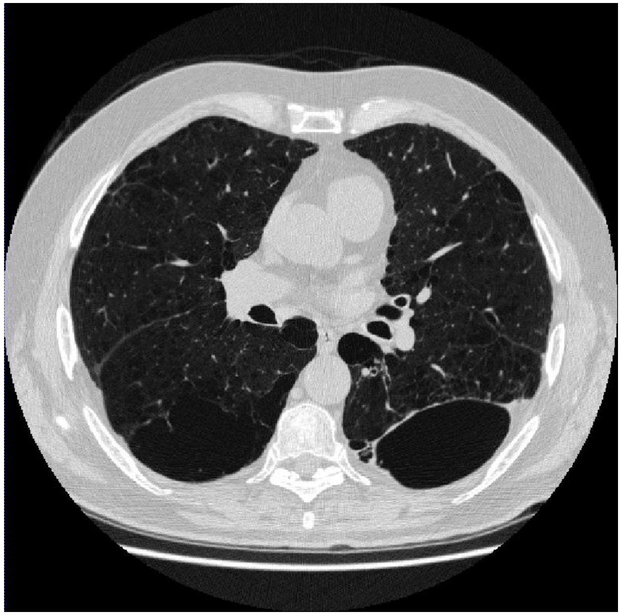


Feature Extraction

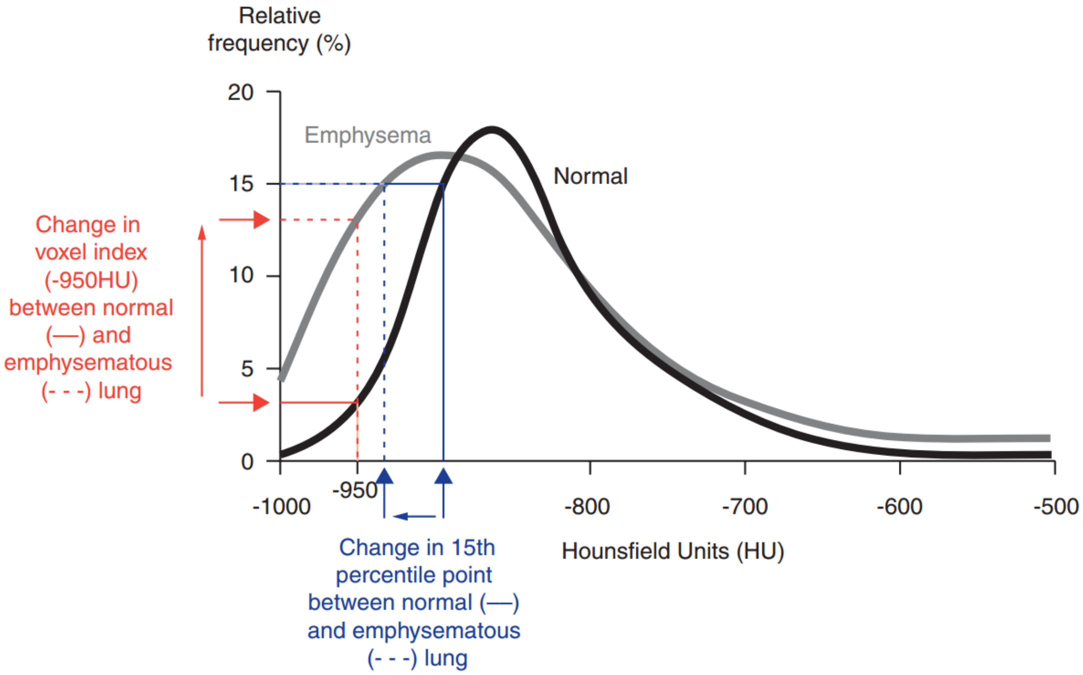
- **Histogram based features** (Mean, Deviation, Variance, Kurtosis, Skewness, Percentiles)
- **Texture features** (Co-occurrence matrices, Local Binary Pattern)
- **SIFT** (Scale-invariant feature transform)
- **SURF** (Speeded Up Robust Features)
- **HoG** (Histogram of Oriented Gradients)

Feature Extraction

- **Histogram based features:** example and motivation

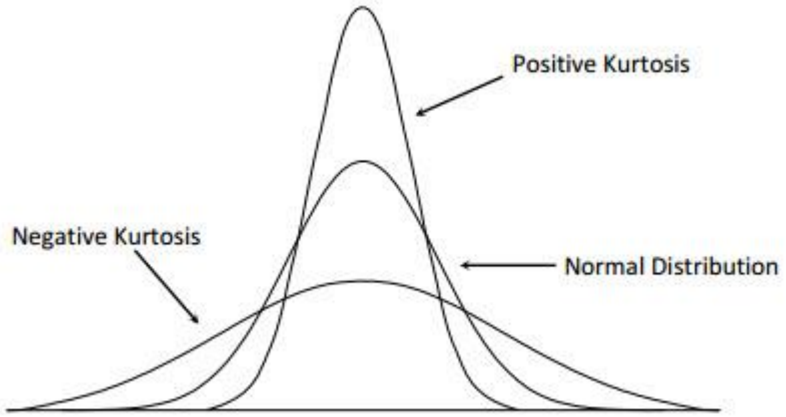


Emphysematous lung

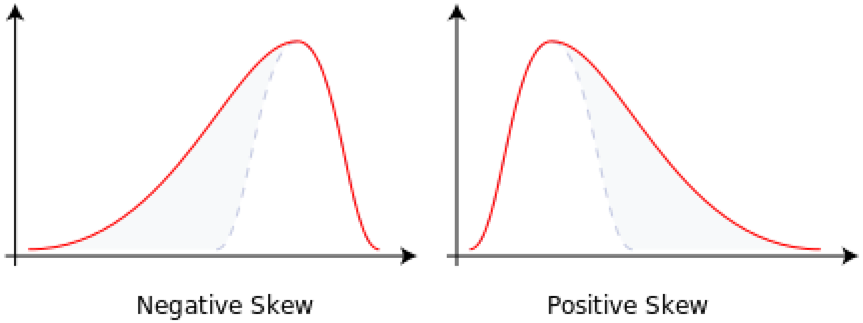


Feature Extraction

- **Histogram based features:**
  1) Mean
  2) Deviation
  3) Variance
  4) Kurtosis
  5) Skewness
  6) Percentiles



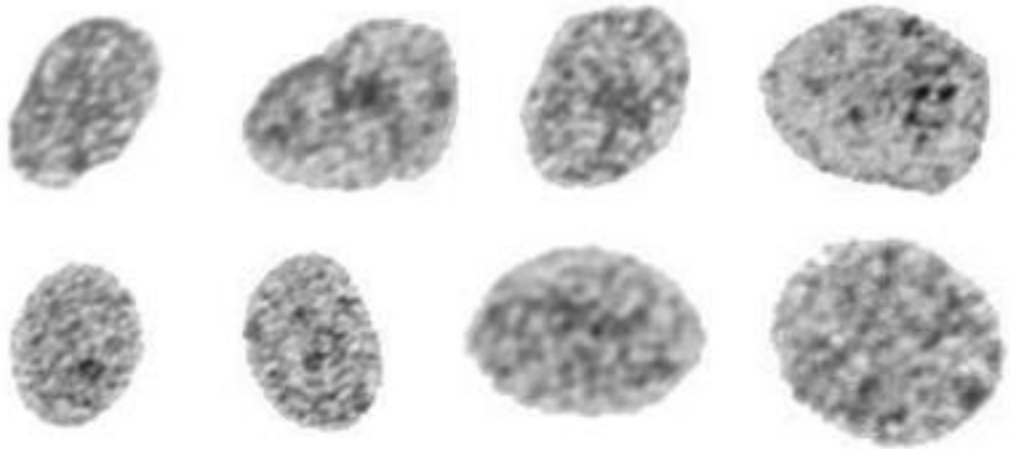
Kurtosis



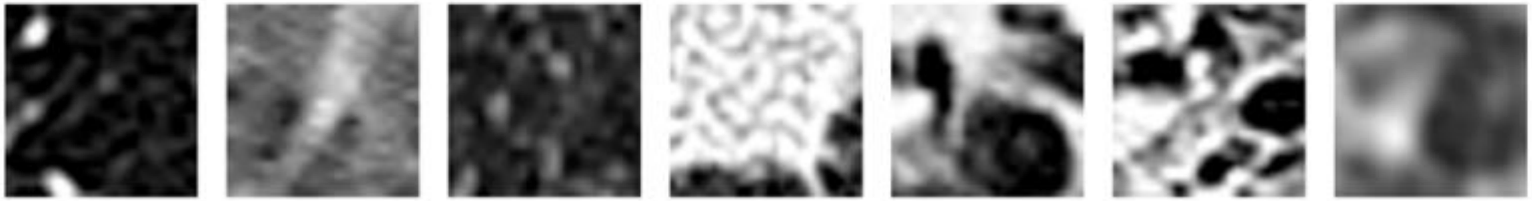
Skewness

Feature Extraction

- **Texture features:** example and motivation



Cells from a tumor with good and poor prognosis.



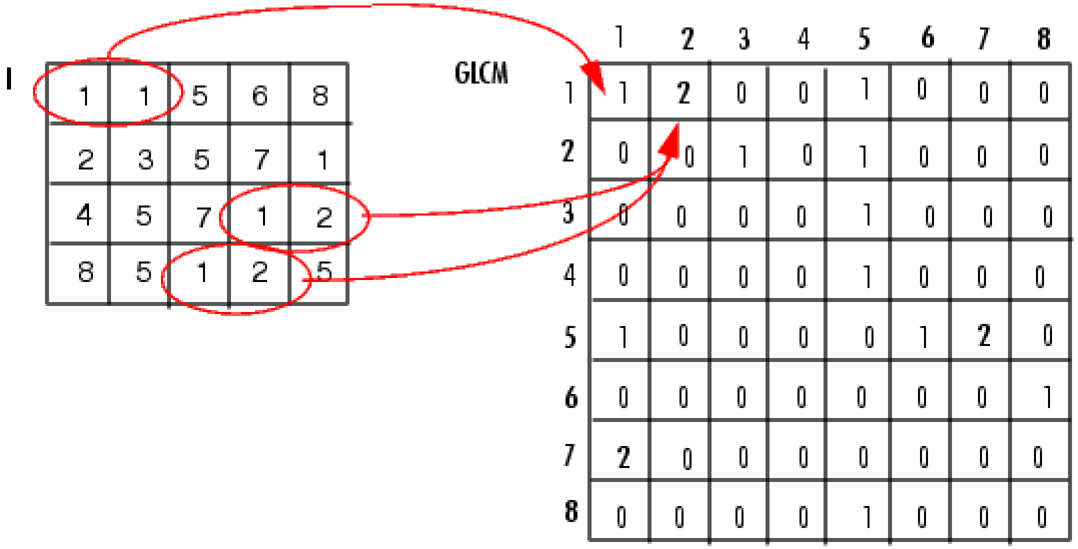
Interstitial Lung Disease (ILD) patterns: healthy, GGO, micronodules, consolidation, reticulation, honeycombing, combination of GGO and reticulation.

Feature Extraction

- **Texture features:**

  1) Co-occurrence Matrices (CM)

  Capture the spatial dependence of gray-level values within an image.
  The GLCM functions characterize the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix.



I

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

GLCM

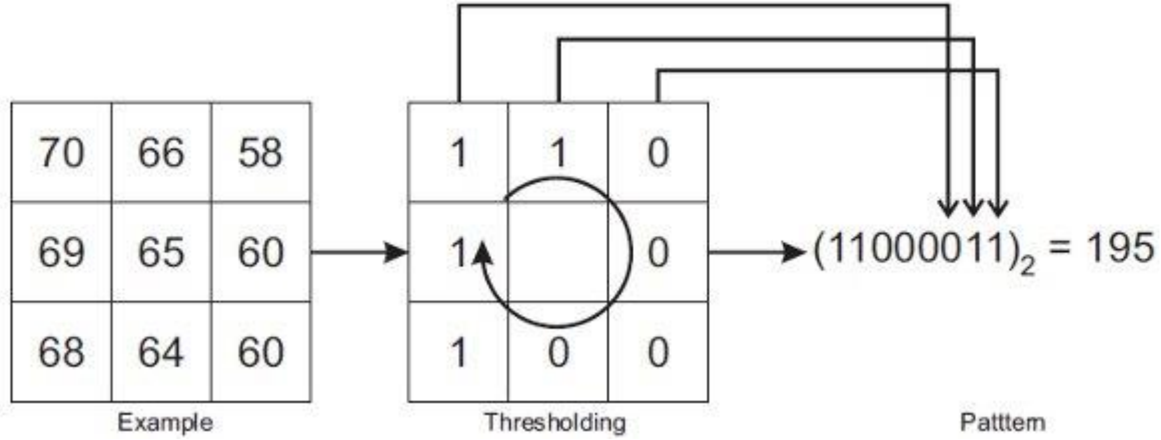
	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Feature Extraction

- **Texture features:**

  2) Local Binary Pattern

  LBP looks at points surrounding a central point and tests whether the surrounding points are greater than (1) or less than (0) the central point. This gives a binary number.



70	66	58
69	65	60
68	64	60

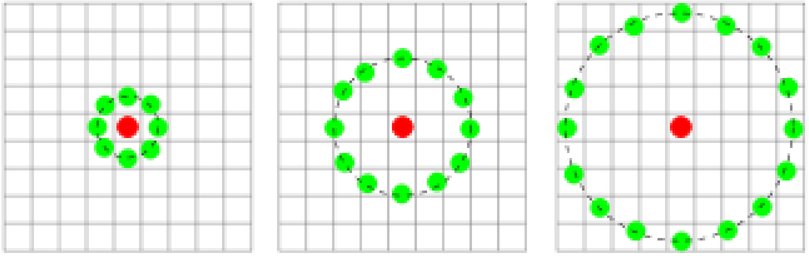
Example

1	1	0
1		0
1	0	0

Thresholding

$(11000011)_2 = 195$

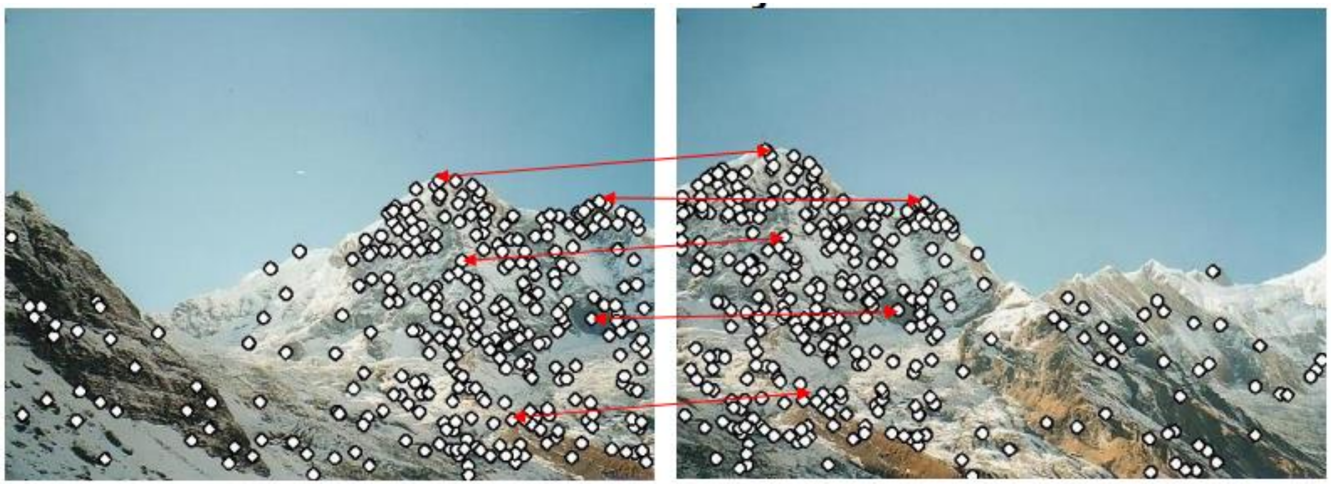
Patttern

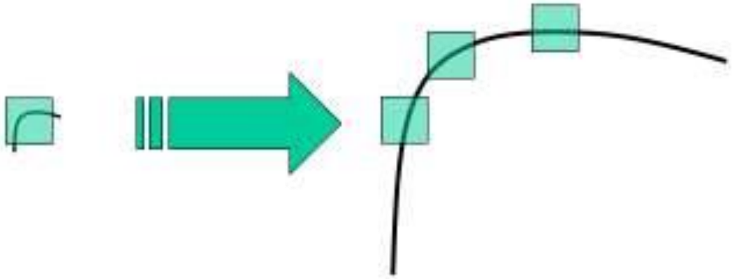


Three neighborhood examples used to define a texture and calculate a local binary pattern (LBP)

Feature Extraction

- **SIFT** (Scale-invariant feature transform): example and motivation





We need a space-invariance feature extractor.

Feature Extraction

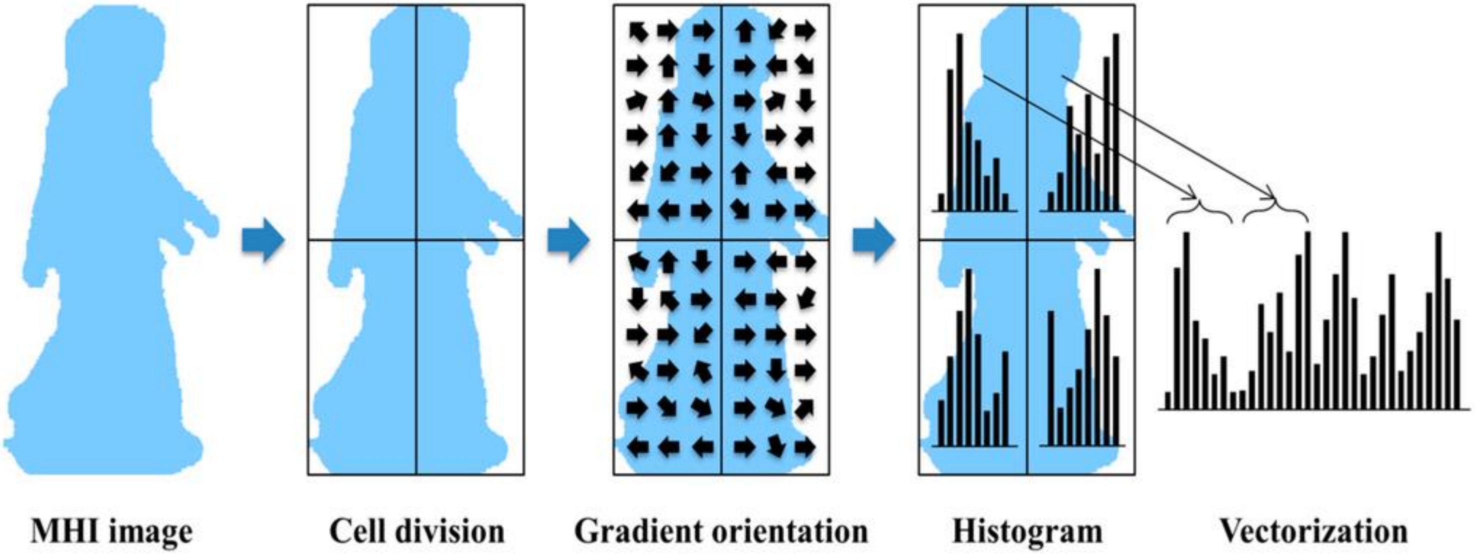
- **SIFT** (Scale-invariant feature transform):
  - Avoid the scale-invariance problem.
  - Useful for track images, detect and identify objects…

- **SURF** (Speeded up robust features):
  - SURF was first presented by Herbert Bay, et al., in 2006.
  - Partly inspired by the scale-invariant feature transform (SIFT) descriptor.
  - The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT.

Feature Extraction

– **HoG** (Histogram of Oriented Gradients):

- Local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions.
- The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled.
- The descriptor is the concatenation of these histograms.



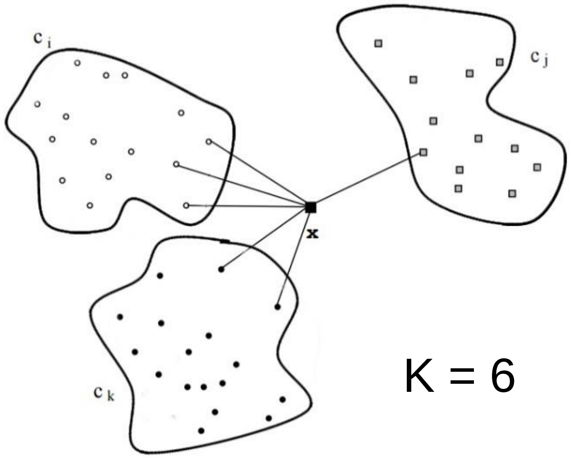
Classifiers

- **Classification algorithms:**
  - K-NN
  - SVM
  - MLP
  - Decision Trees
  - Random Forest
  - Naive Bayes
  - …

Classifiers — K Nearest Neighbors (K-NN)

- Classification is computed from a simple **majority vote** of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

- Algorithm:
  - For each test point x to be classified, find the *K* nearest samples in the training data.
  - Classify the point x according to the majority vote of their class labels

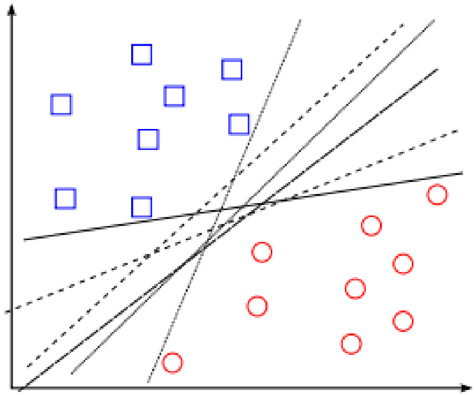


K = 6

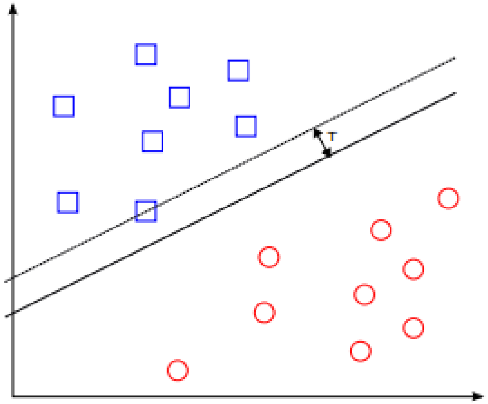
- Parametric classifier:
  - N, number of neighbors. N = 1,2,3, …
  - D, distance function, D = Euclidean, Minkowski, …
  - W, weight function used in prediction, W = uniform, distance, …

Classifiers — Support Vector Machines (SVM)

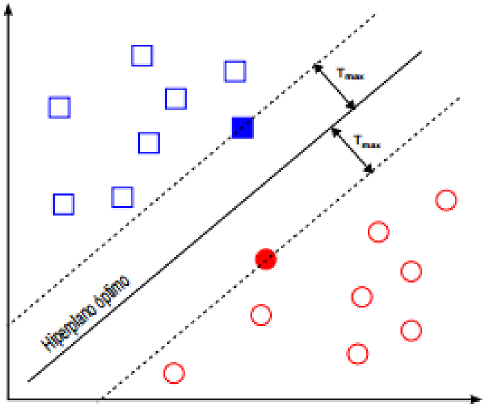
- Assuming two classes are linearly separable, we can find infinite hyperplanes that splits the classes.
- We define the **margin** of a linear classifier as the distance between the hyperplane and the nearest sample on any class.
- The optimal hyperplane will be the one with the maximum margin.



Infinite possible planes that separate the classes



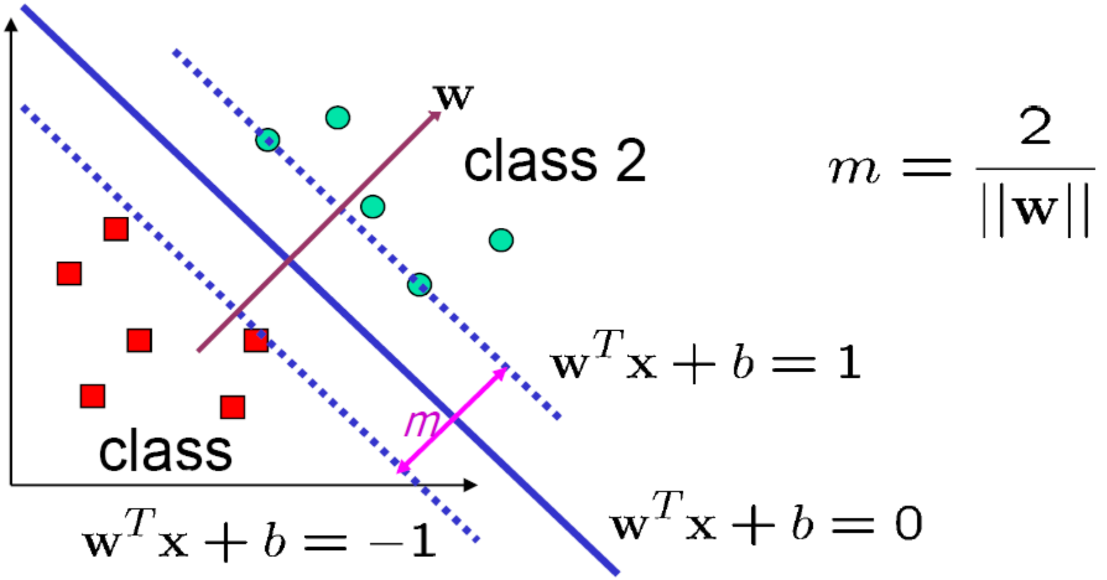
Non-optimal hyperplane



Optimal hyperplane

Classifiers — Support Vector Machines (SVM)

- Assuming two classes are linearly separable, we can find infinite hyperplanes that splits the classes.
- We define the **margin** of a linear classifier as the distance between the hyperplane and the nearest sample on any class.
- The optimal hyperplane will be the one with the maximum margin.



$\mathbf{w}$
class 2
$m = \frac{2}{||\mathbf{w}||}$
$\mathbf{w}^T\mathbf{x} + b = 1$
m
class
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

Classifiers — Support Vector Machines (SVM)

- Assuming two classes are linearly separable, we can find infinite hyperplanes that splits the classes.
- We define the **margin** of a linear classifier as the distance between the hyperplane and the nearest sample on any class.
- The optimal hyperplane will be the one with the maximum margin.

- We can find the optimal hyperplane by the optimizacion problem:

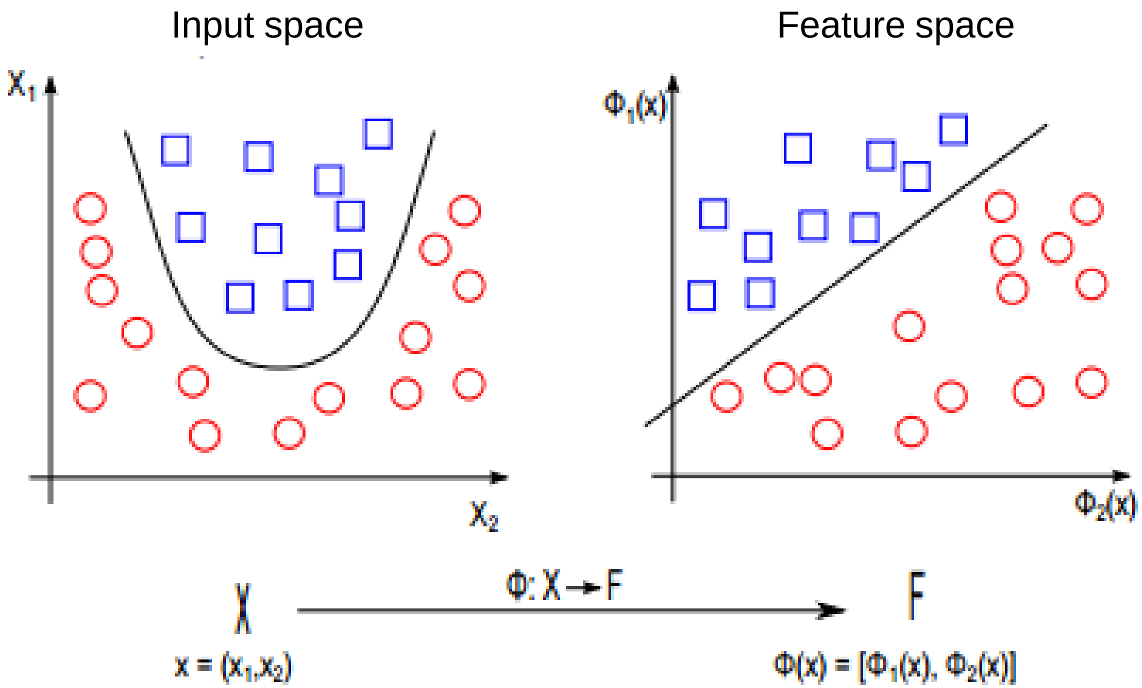
$$\text{minimize } \frac{1}{2}||w||^2$$

$$\text{subject to } y_i(w^T x_i + b) \geq 1 \quad \forall i$$

Classifiers — Support Vector Machines (SVM)

– If the classification problem is not linearly separable, we transform the data x_i into a new higher dimension space through a function ϕ:

  - Input space: original space of x
  - Feature space: new $\phi(x)$ space, now linearly separable



Input space

Feature space

Classifiers — Support Vector Machines (SVM)

- If the classification problem is not linearly separable, we transform the data x_i into a new higher dimension space through a function ϕ:
  - Input space: original space of x
  - Feature space: new $\phi(x)$ space, now linearly separable
- Calculate the transformed space $\phi(x)$ is a complex and computationally expensive task.
- Instead, we only have to specified the kernel function:
  - Linear:

$$K(x_i, x_j) = {x_i}^T x_j$$

  - Polynomial of power *p*:

$$K(x_i, x_j) = \left(1 + {x_i}^T x_j\right)^p$$

  - Gaussian (radial-basis function network):

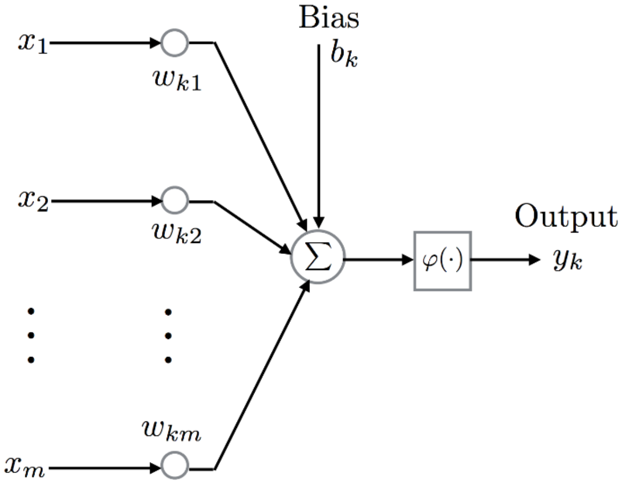
$$K(x_i, x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)$$

  - Sigmoid:

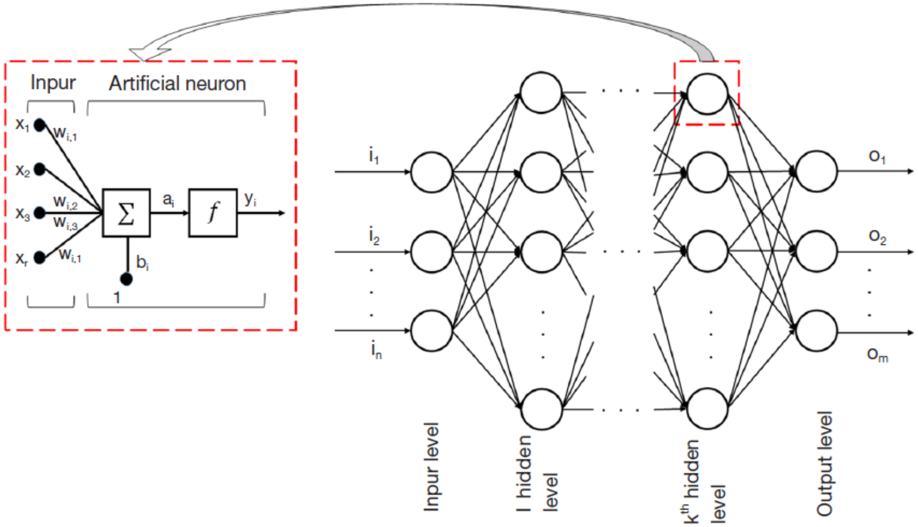
$$K(x_i, x_j) = \tanh(\beta_0 {x_i}^T x_j + \beta_1)$$

Classifiers — MultiLayer Perceptron (MLP)

- An artificial neuron can be modeled as: $y = f(xw + b)$.
  - y output
  - x input vector
  - w synaptic weight vector
  - f nonlinear activation function.



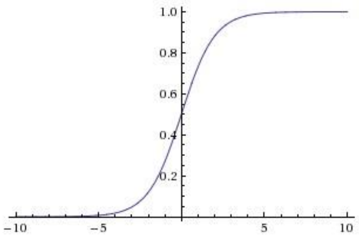
- MLP is an artificial neural network model which consists in multiple layers of nodes (neurons) with a nonlinear activation function in a directed graph, with each layer fully connected to the next one.



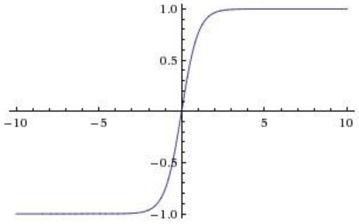
Classifiers — MultiLayer Perceptron (MLP)

– Commonly used activation functions are:

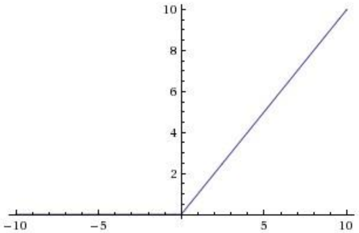
- Sigmoid $\sigma(x) = \dfrac{1}{(1 + e^{-x})}$
- Tanh $\tanh(x) = 2\sigma(2x) - 1$
- ReLU $\mathrm{ReLU}(x) = \max(0, x)$



Sigmoid



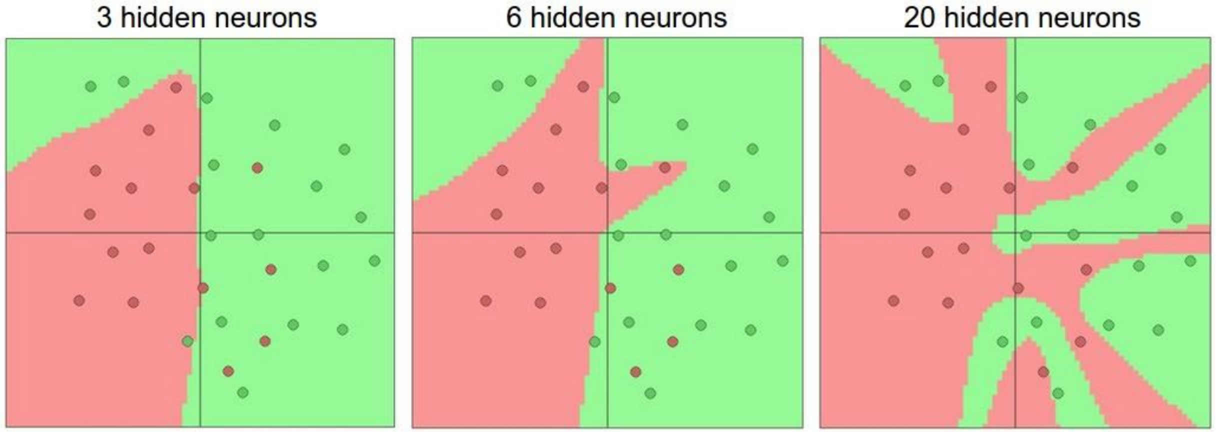
Tanh



ReLU

– The weights W of the MLP are learned by minimizing the error between the real targets,y, and the estimated output $\hat{y} = F(x, W)$.

– As we increase the size and number of layers in a MLP, the capacity of the network increases.


3 hidden neurons

6 hidden neurons

20 hidden neurons

Performance measures & Validation Method (I)

– How predictive is the model we learned?

– Natural performance measure for classification problems:
  - **Error** rate: proportion of error made over the wole set of instances.
  - **Accuracy**: proportion of correctly classified instances over whole set of instances. $Accuracy = 1 - error\ rate$

→ Problem: assumes relatively uniform class distribution

→ Solution: confusion matrix

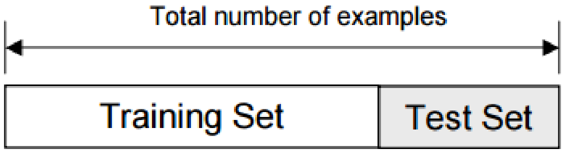
– Confusion matrix (contingency table):
  - TP: Number of True Positives
  - FP: Number of False Positives
  - TN: Number of True Negatives
  - FN: Number of False Negatives

		predicted	
		negative	positive
Actual class	negative	TN	FP
	positive	FN	TP

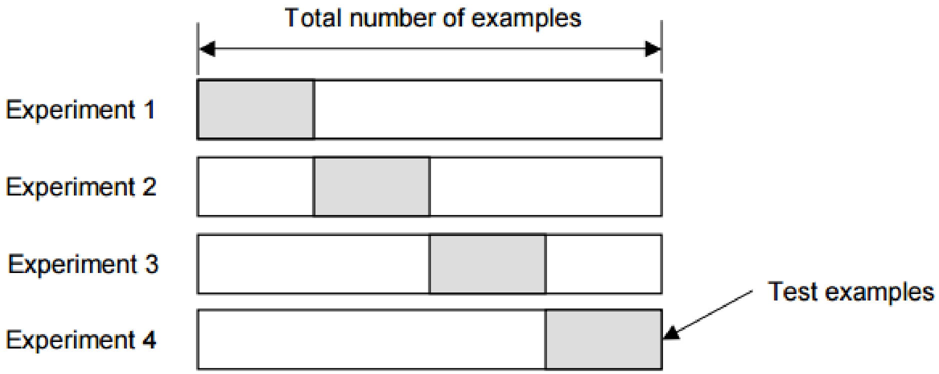
Performance measures & Validation Method (II)

- Validation techniques are motivated by two problems: model selection (select the optimal parameters of the model) and model performance estimation (once we have chosen a model, we measure the performance).

- In real applications we only have access to a finite set of samples. One approach is to Split the training data into subsets: training set and validation set.



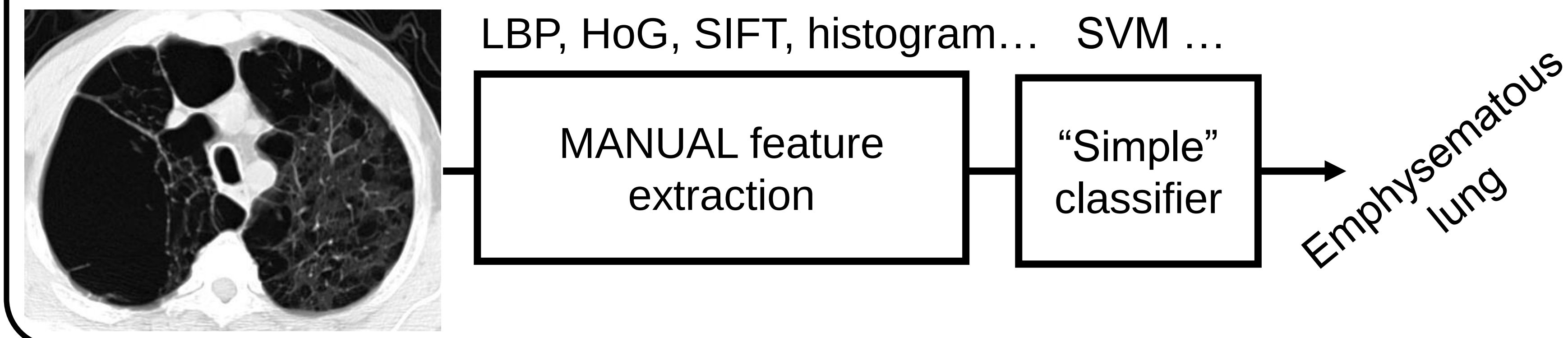
- **K-Fold Cross Validation** consists on create a K-fold partition of the the dataset.
  - For each of K experiments, use K-1 folds for training and the remaining one for testir



K = 4

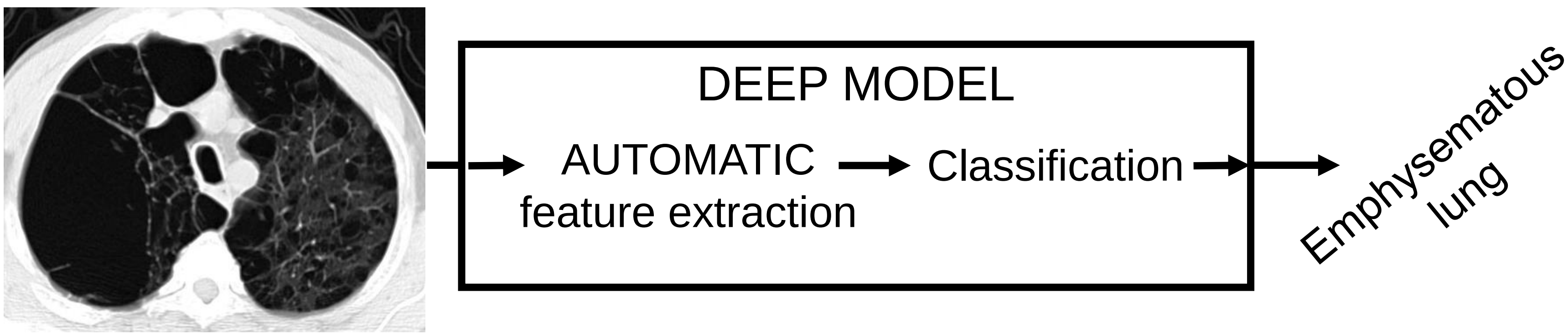
Introduction: What is Deep Learning?

TRADITIONAL MACHINE LEARNING: **manual** feature extraction



LBP, HoG, SIFT, histogram… SVM …

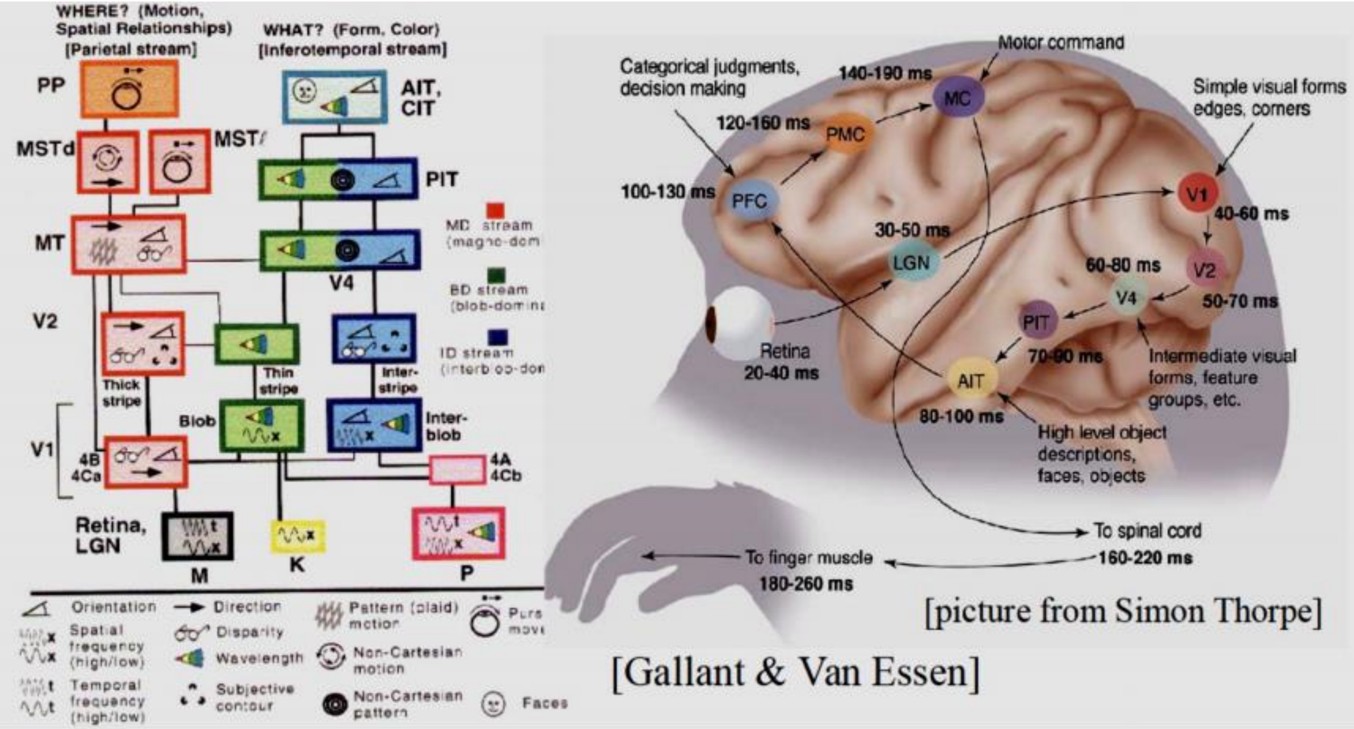
DEEP LEARNING: **automatic** feature extration.



Deep Learning allows us to model the function $f(x)$ that maps **directly** the original input (raw data) to output.

Deep Learning – Brain inspired

- Most current learning algorithms are shallow architectures (SVM, KNN, PCA…), but the mammal **brain is organized in a deep architecture** and the ventral pathway in the **visual cortex has multiple stages** (= hierarchical).



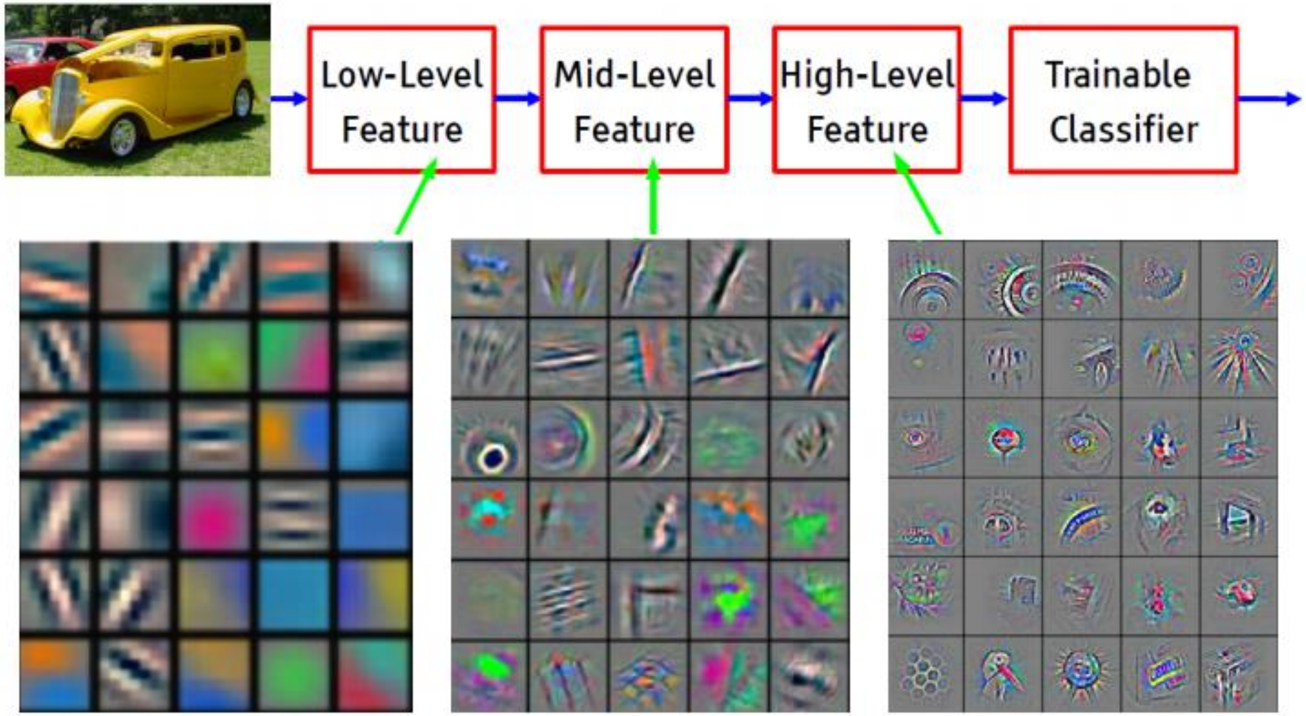
[picture from Simon Thorpe]

[Gallant & Van Essen]

- Deep architectures are composed of **multiple levels** of non-linear operations.
- Deep learning methods aim at:
  - Learning feature hierarchies
  - Where features from higher levels of the hierarchy are formed by lower level features.

Deep Learning – Trainable features hierarchy

- Deep learning assumes it is possible to “learn” a hierarchy of descriptors (features) with increasing abstraction, i.e., layers are trainable feature transforms.
- In image recognition: Pixel → Edge → Texton → Motif → Part → Object.
- Also applies to text analysis (character → meaning), speech recognition (voice → word) etc.

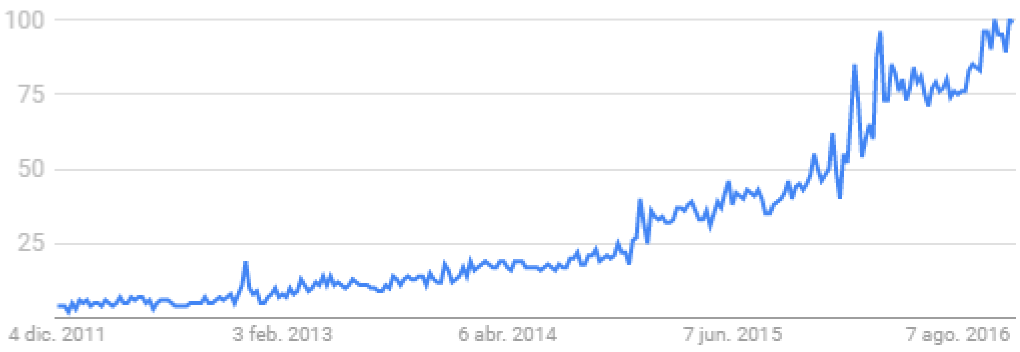


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Deep Learning – Types

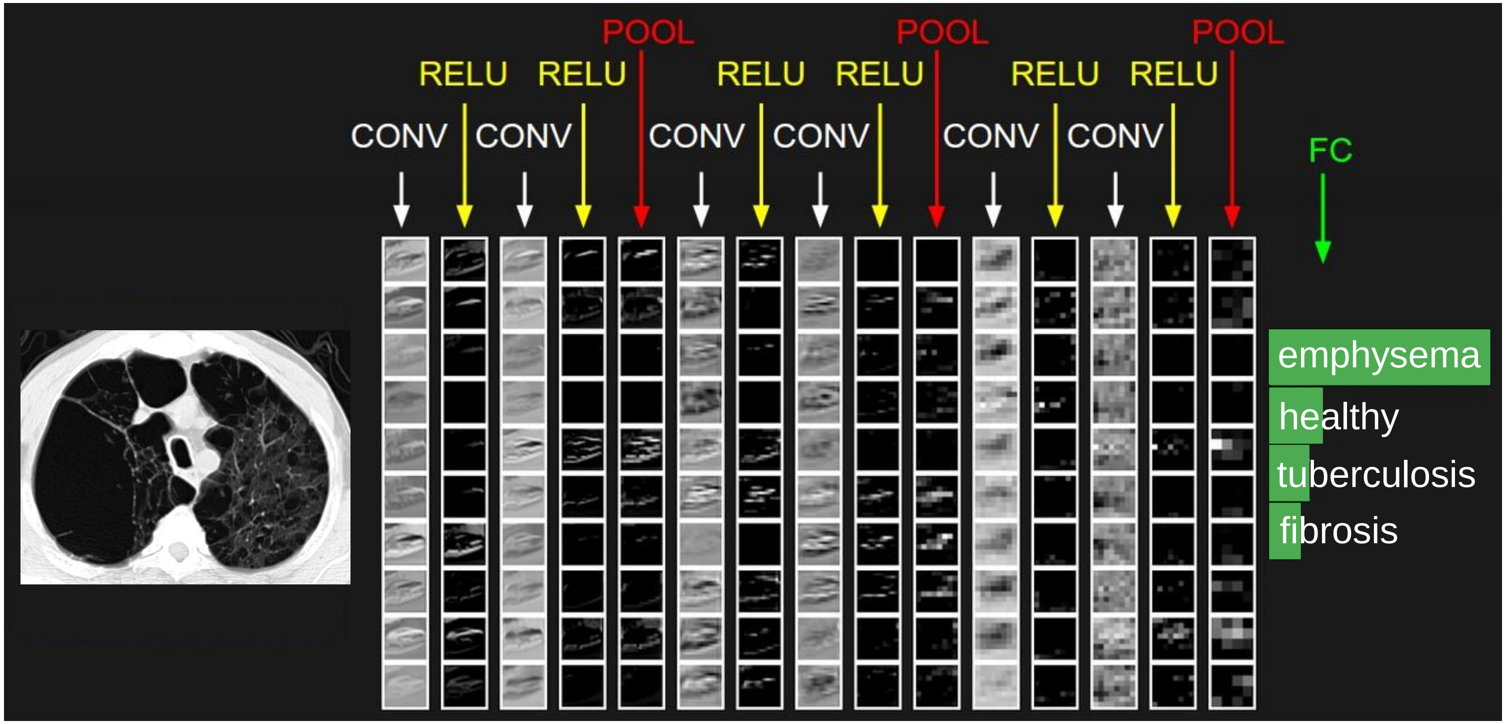
- **Supervised learning**
  - Deep Convolutional Neural Networks
- **Unsupervised learning**
  - Stacked Auto Encoders (several Auto Encoders stacked)
  - Deep Belief Networks (several Restricted Boltzmann Machines stacked)

Google Trends: deep learning



Convolutional Neural Networks (CNN)

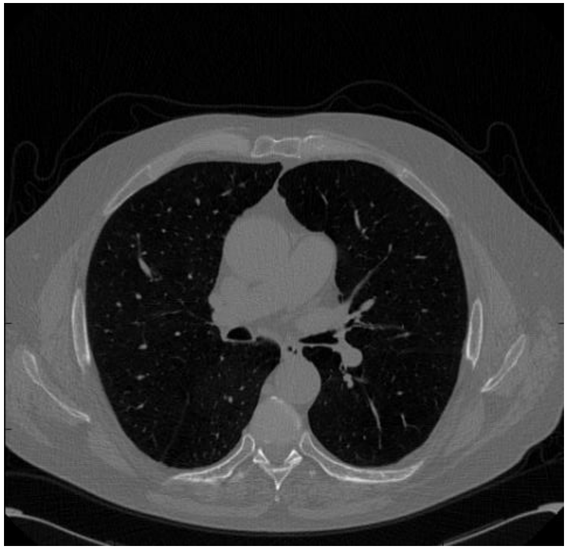
- Convolutional nets were inspired by the visual system’s structure.
- CNN are composed by three principal blocks:
  - Convolutional layers
  - Pooling layers
  - Fully connected layers



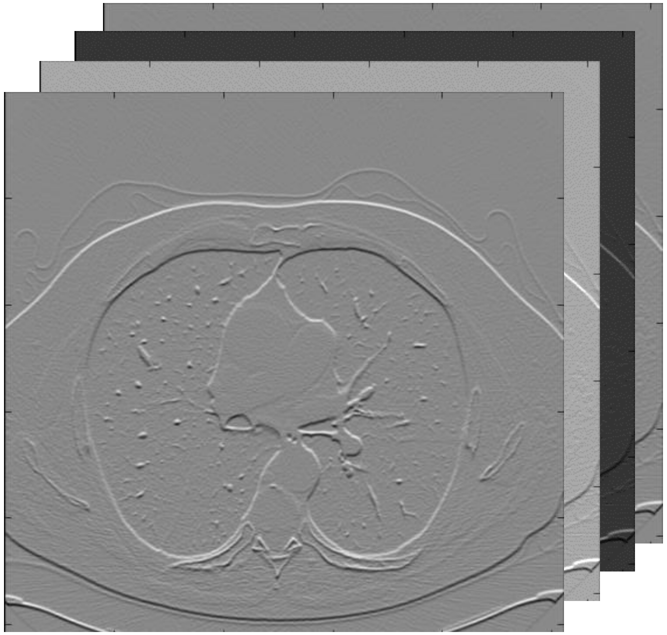
Adapted from CS231n Convolutional Neural Networks for Visual Recognition, Stanford

Convolutional Neural Networks (CNN) – Conv Layer

- **Convolutional Layer** learns a bank of N filters (K), that produces N feature maps from the input image.



Input: image

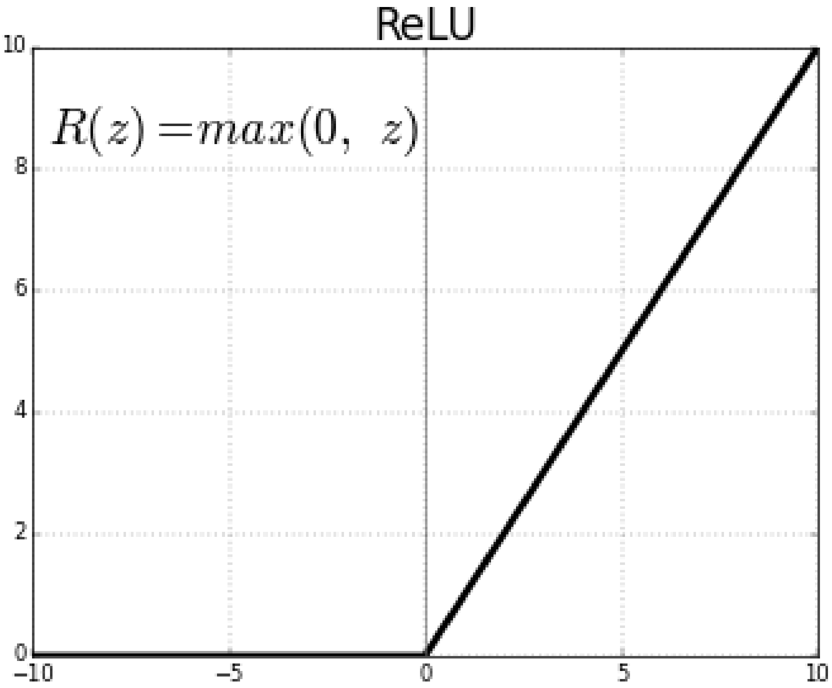


N feature maps which result from the convolution with N filters

- In this layer each neuron is only connected with a subset of neurons of the previous layer (receptive filed = kernel size).
- Weight sharing increases learning efficiency by greatly reducing the number of free parameters being learnt.

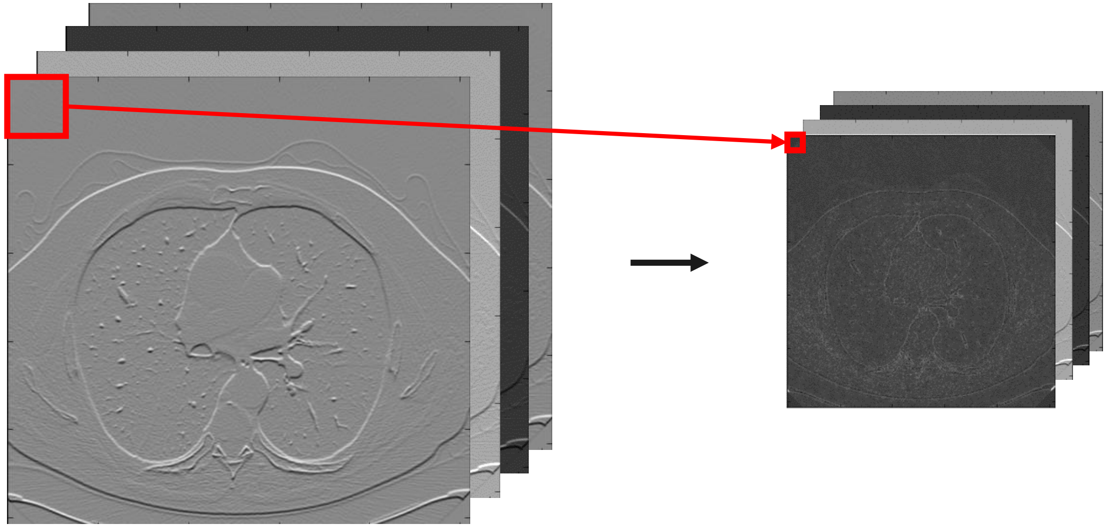
Convolutional Neural Networks (CNN) – Conv Layer

- After computing the feature maps by the Convolutional Layer, usually we apply to the output a nonlinear transformation (as we do in the MLP).
- A commonly used function is ReLU: $f(x) = \max(0, x)$



Convolutional Neural Networks (CNN) – Pool Layer

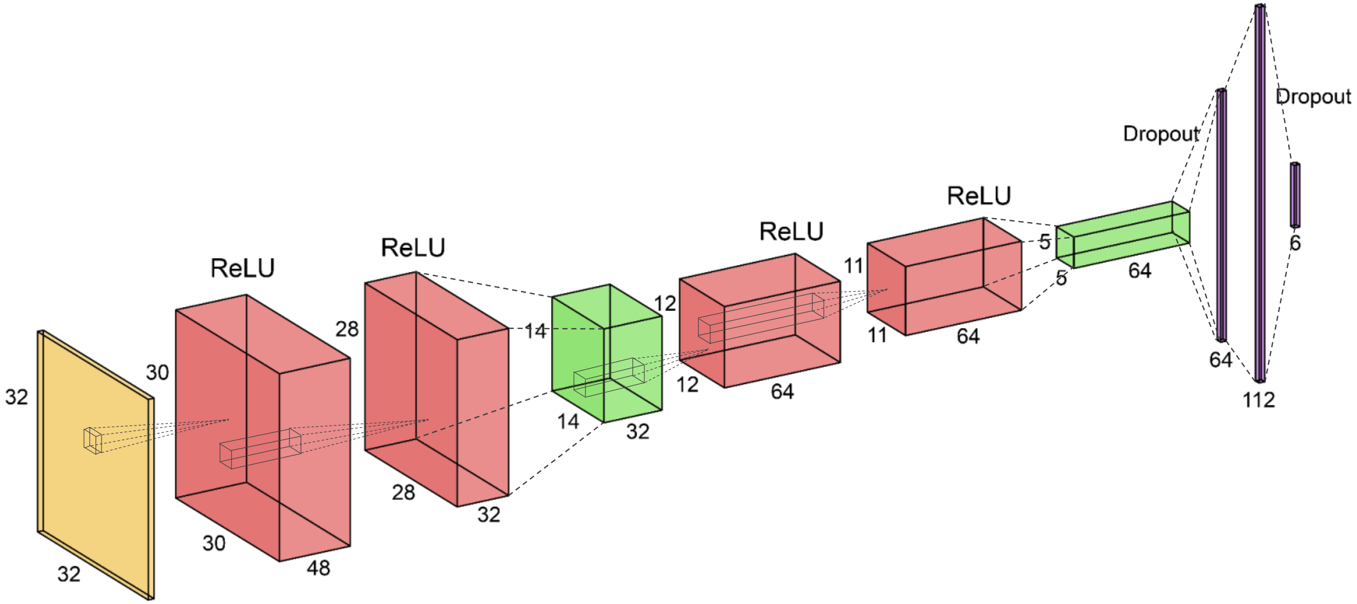
- **Pooling Layer** implements a nonlinear subsampling in order to reduce the feature dimensionality.
- Commonly used pooling function is max-pooling, which consist on taking the maximum value from the values in a receptive field. Max-pooling has been favoured over others due to its better performance characteristics.



- It provides a form of translation invariance.

Convolutional Neural Networks (CNN)

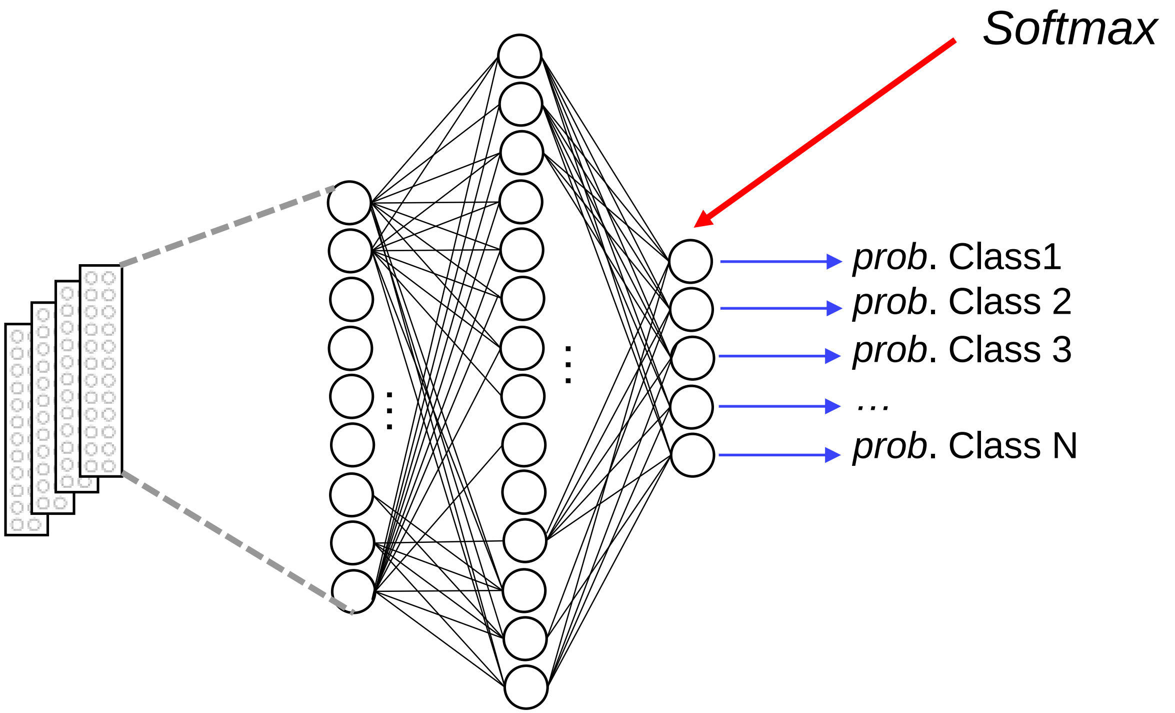
- Convolutional Networks are a sequence of Convolutional layers (plus a nonlinear transformation) and Pooling layers where every layer of a CNN transforms one volume of activations to another through a differentiable function.
- The hierarchy given by stacking this layers leads to a hierarchy in the features, going from local features (edges, etc) to more abstract and complex ones.



Convolutional Neural Networks (CNN) – Fully connected Layer

- Once the CNN has already extracted the optimal features that represent the data, we perform a classification task.
- The classification is usually done by a **MLP**.

$$\begin{cases} z_i^l = W_i^l y^{l-1} + b_i^l \\ y_i^l = \dfrac{e^{z_i^l}}{\sum_j e^{z_j^l}} \end{cases}$$



Softmax

prob. Class1

prob. Class 2

prob. Class 3

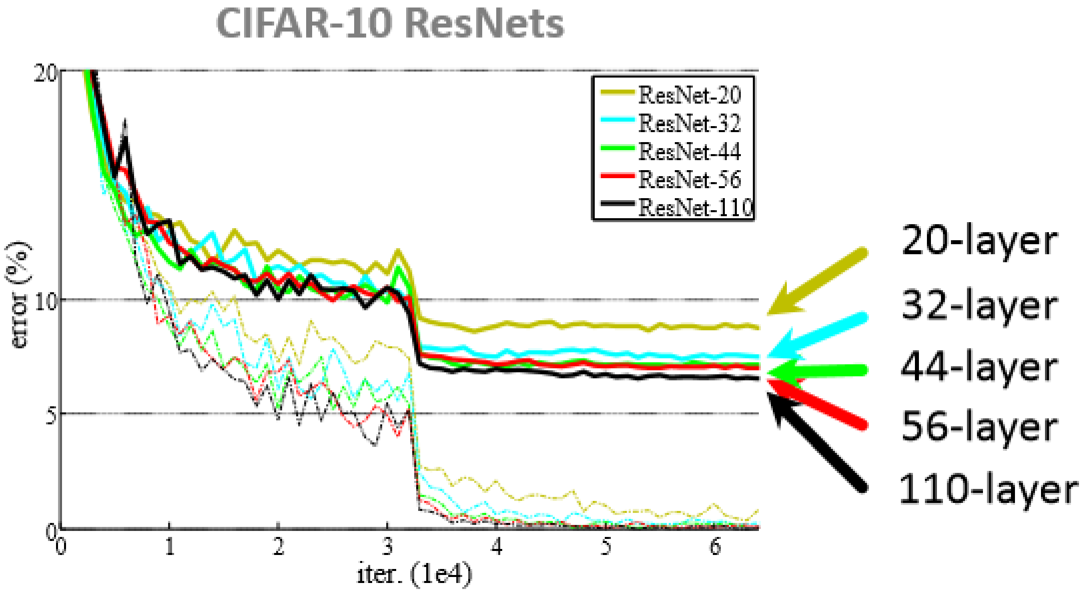
…

prob. Class N

- The **softmax** function converts output scores (arbitrary values) into probabilities (sum=1, values between 0 and 1).

How to choose the correct CNN architecture?

- The optimal topology of the CNN depends on your data and problem to be solved…
- … But we know that shallow CNNs do not have enough discriminative power while too deep architectures are computationally expensive to train and easy to be overfitted.
- In general, the deeper, the better:



The CIFAR-10 dataset consists of 60.000 32x32 colour images in 10 classes.