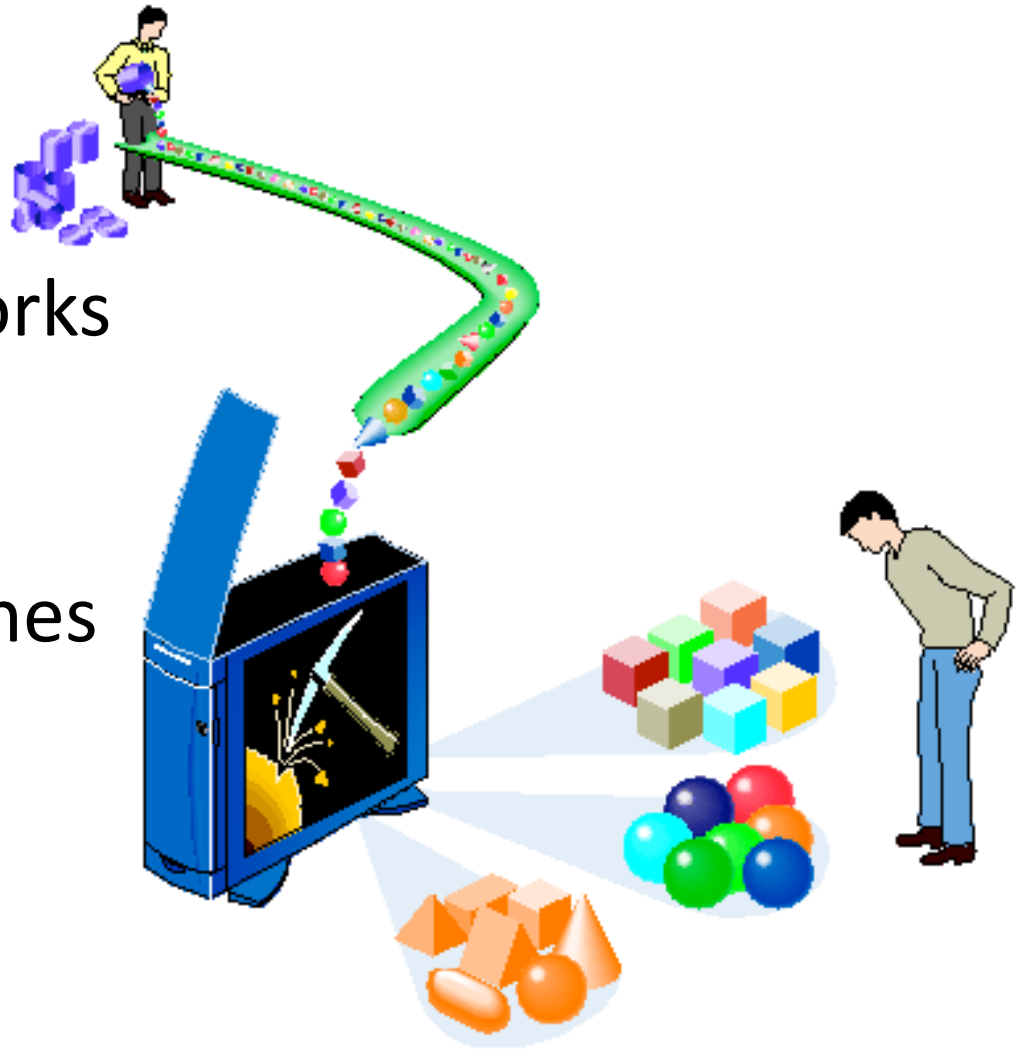
Data Mining
Supervised Methods

Ciro Donalek
donalek@astro.caltech.edu

Summary

- Supervised Methods
- Artificial Neural Networks
  - Multilayer Perceptron
- Support Vector Machines
- Softwares



Supervised Methods

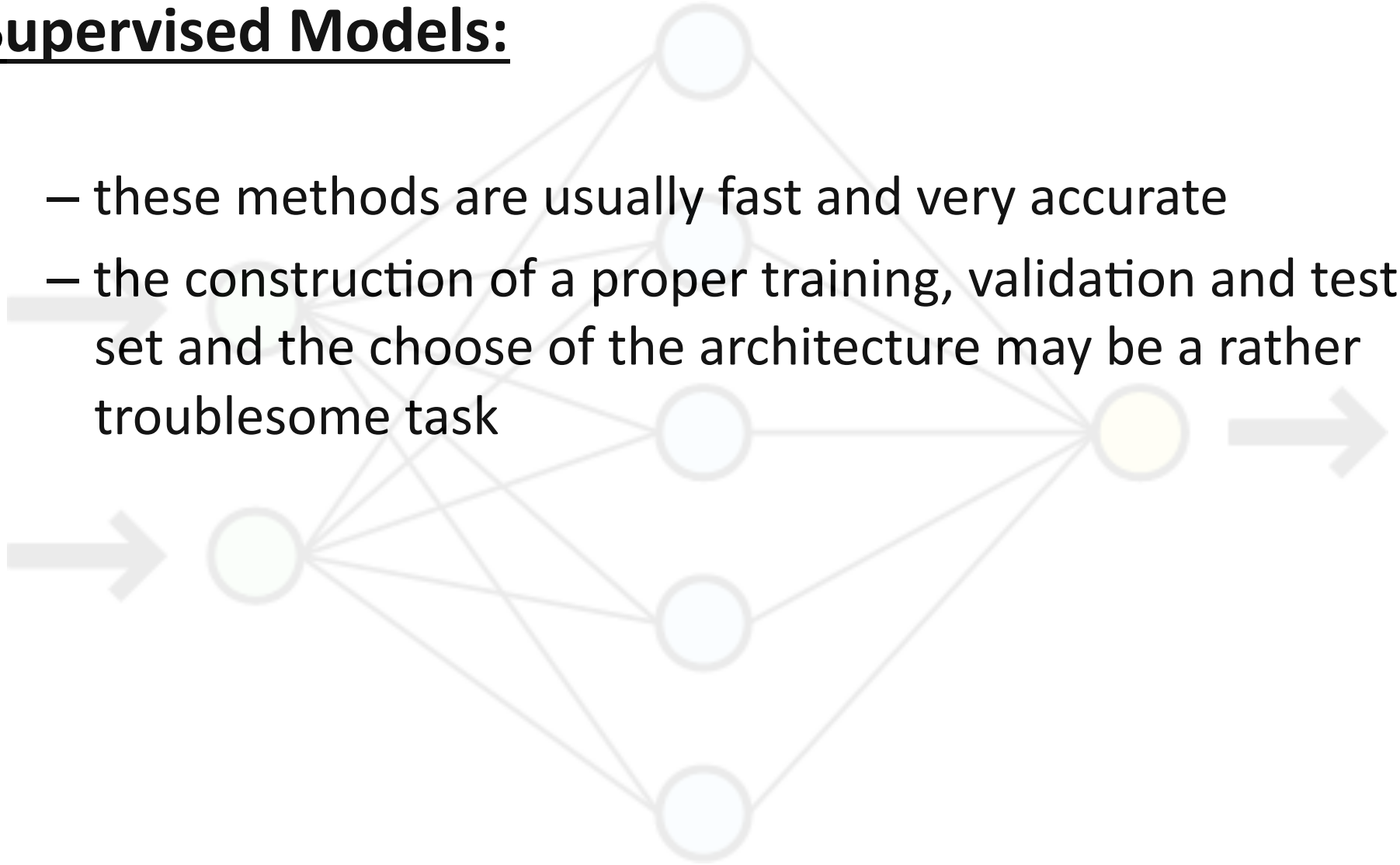
Supervised Models:

- training data includes both the input and the desired results;
- for some examples the correct results (target values) are known and are given in input to the model during the learning process;
- the network has to be able to generalize, ie, to give the correct results when new data are given in input without knowing a priori the target.

Supervised Methods

Supervised Models:

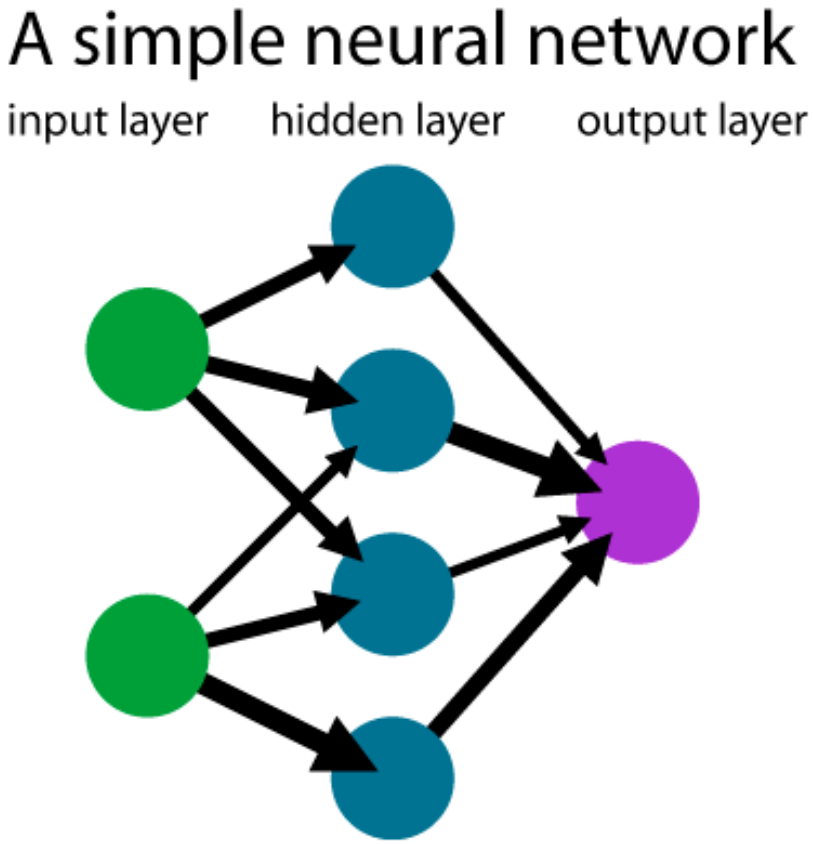
- these methods are usually fast and very accurate
- the construction of a proper training, validation and test set and the choose of the architecture may be a rather troublesome task



Artificial Neural Networks

An Artificial Neural Network is an information processing paradigm that is inspired by the way biological nervous systems process information:

"a large number of highly interconnected simple processing elements (neurons) working together to solve specific problems"

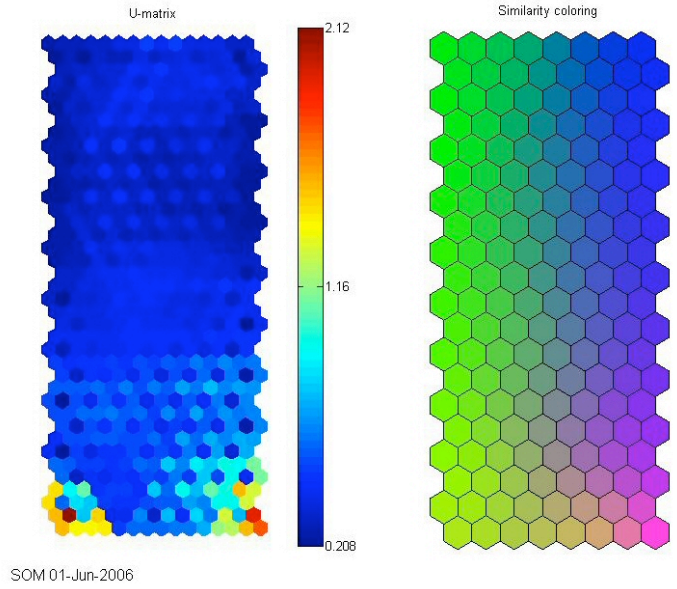


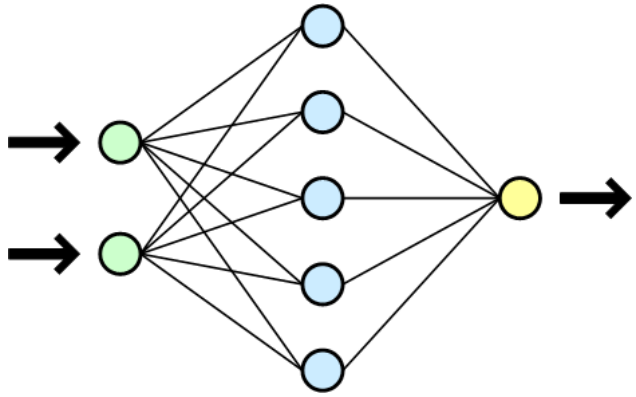
Neural Networks

A Neural Network is usually structured into an input layer of neurons, one or more hidden layers and one output layer.

Neurons belonging to adjacent layers are usually fully connected.

The values of the functions associated with the connections are called “weights“.





Simplified model of a NN

Neural Networks

Feed forward: Single Layer Perceptron, MLP, ADALINE (Adaptive Linear Neuron), RBF

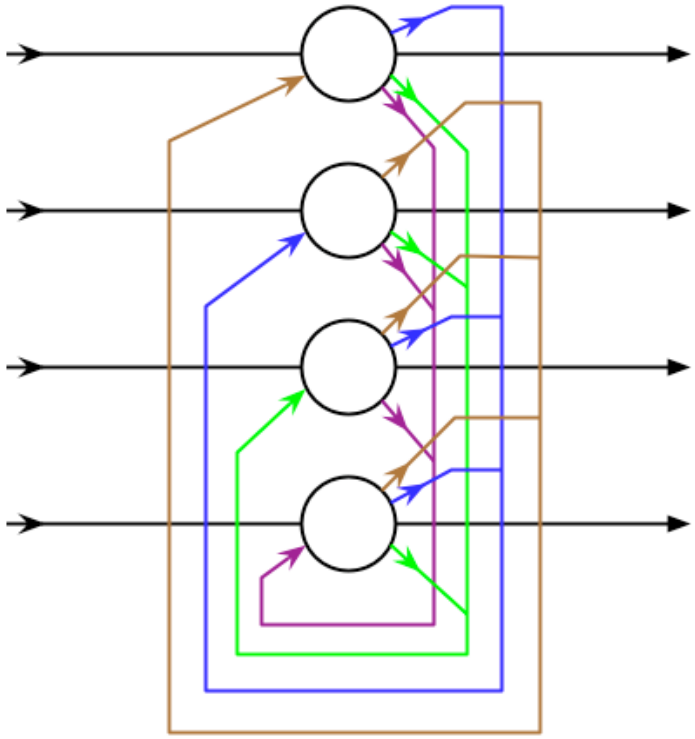
Self-Organized: SOM

Recurrent: Simple Recurrent Network, Hopfield Network.

Stochastic: Boltzmann machines.

Modular: Committee of Machines, ASNN (Associative Neural Networks), Ensembles.

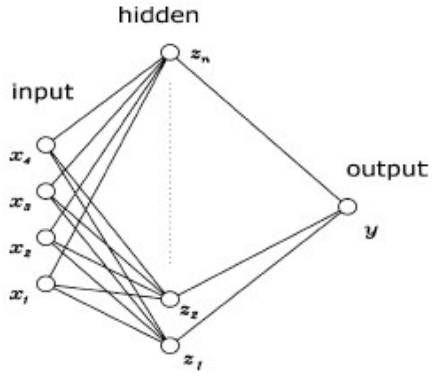
Others: Instantaneously Trained, Spiking (SNN), Dynamic, Cascades, NeuroFuzzy, PPS, GTM.



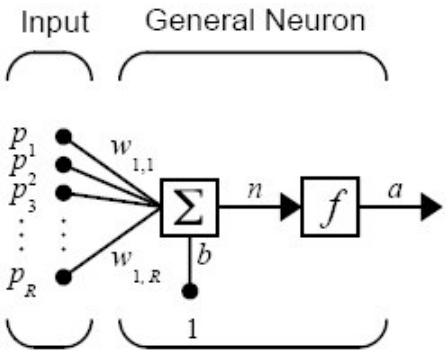
Multilayer Perceptron

The MLP is one of the most used supervised model: it consists of multiple layers of computational units, usually interconnected in a feed-forward way.

Each neuron in one layer has directed connections to all the neurons of the subsequent layer.



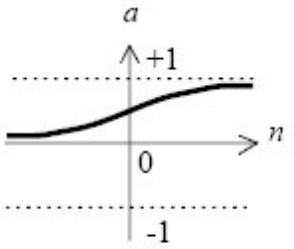
The architecture of a two layer MLP.



$a = f(\mathbf{W}\mathbf{p} + b)$

Where

R = number of elements in input vector

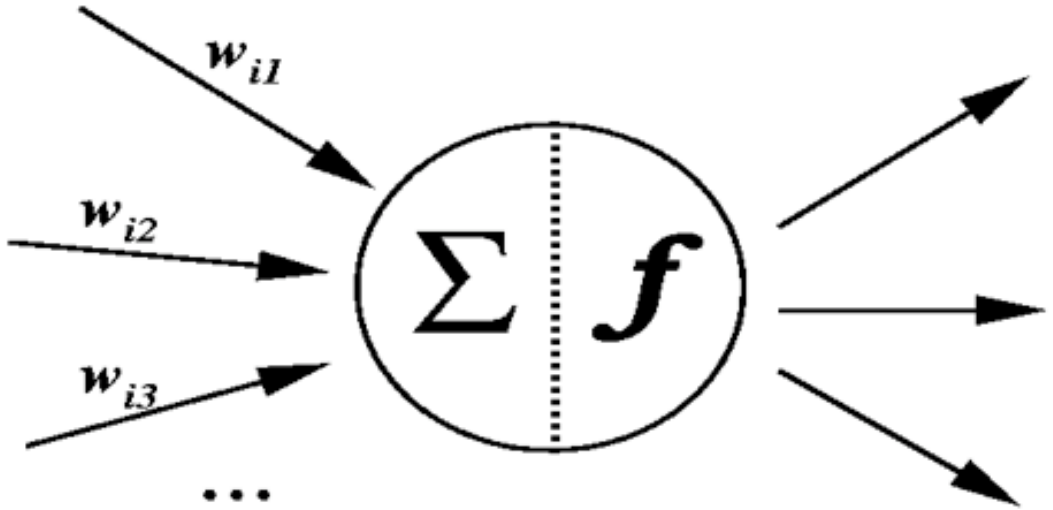


$a = logsig(n)$

A Simple Artificial Neuron

The basic computational element is often called a node or unit. It receives input from some other units, or from an external source. Each input has an associated weight w, which can be modified so as to model synaptic learning. The unit computes some function f of the weighted sum of its inputs:

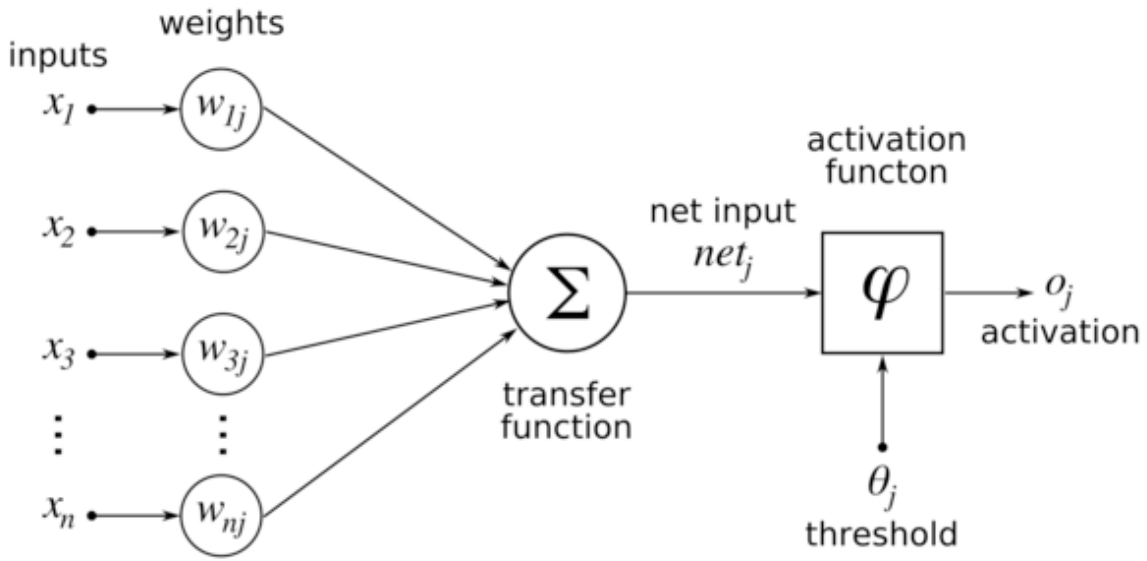
$$y_i = f(\sum_j w_{ij}\, y_j)$$



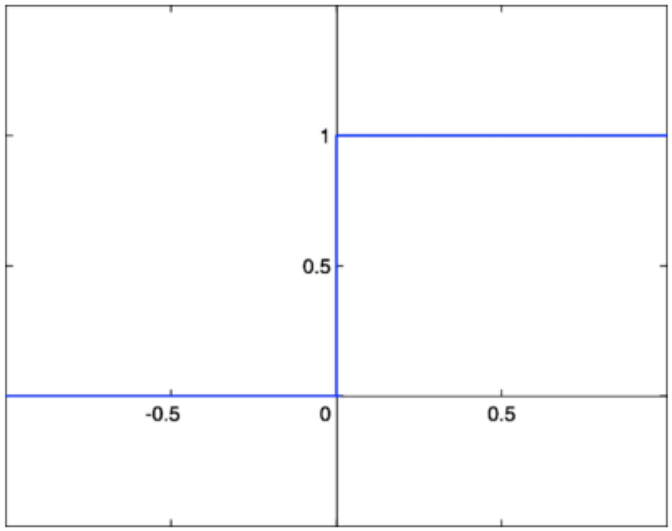
$$y_i = f(net_i)$$

Activation Functions

- Activation Functions
  - scalar to scalar function;
  - used by most units to transform their inputs;
  - needed to introduce non-linearity into the network
  - linear, logistic, tanh, softmax...



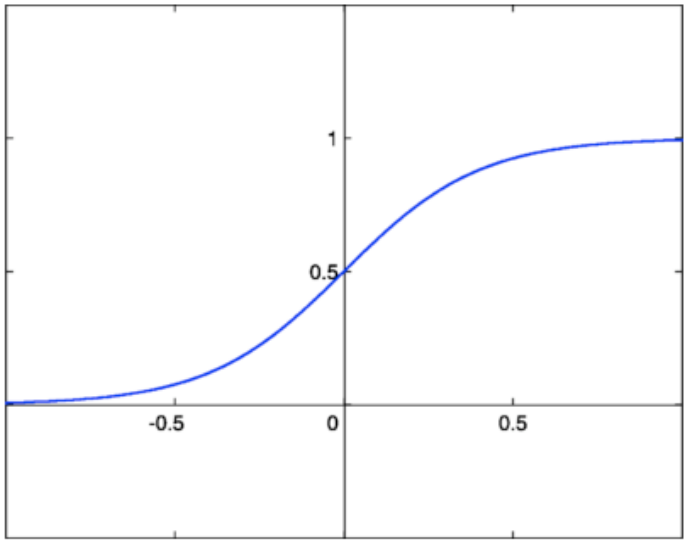
Activation Functions - 2



Step function

The output is a certain value A1, if the input sum is above a certain threshold and A0 if the input sum is below a certain threshold.

When we want to classify an input pattern into one of two groups, we can use a binary classifier with a step activation function.



Sigmoid function

Has the property of being similar to the step function, but with the addition of a region of uncertainty.

Sigmoid functions in this respect are very similar to the input-output relationships of biological neurons.

$$\sigma(t) = \frac{1}{1 + e^{-\beta t}}$$

Error Functions

- Error Functions
  - most methods for supervised learning require a measure of the discrepancy between the netwrok output values and the target;
  - sum of the squared errors (SSE), cross entropy (CE), etc.

$$E_p = \frac{1}{2}\sum_j \left(t_j^p - y_j^p\right)^2$$

Using a Multilayer Perceptron with a softmax activation function and cross-entropy error, the network outputs can be interpreted as the conditional probabilities $p(C_1|\mathbf{x})$ and $p(C_2|\mathbf{x})$ where $\mathbf{x}$ is the input vector, C_1 the first class, C_2 the second class.

MLP – Supervised Learning

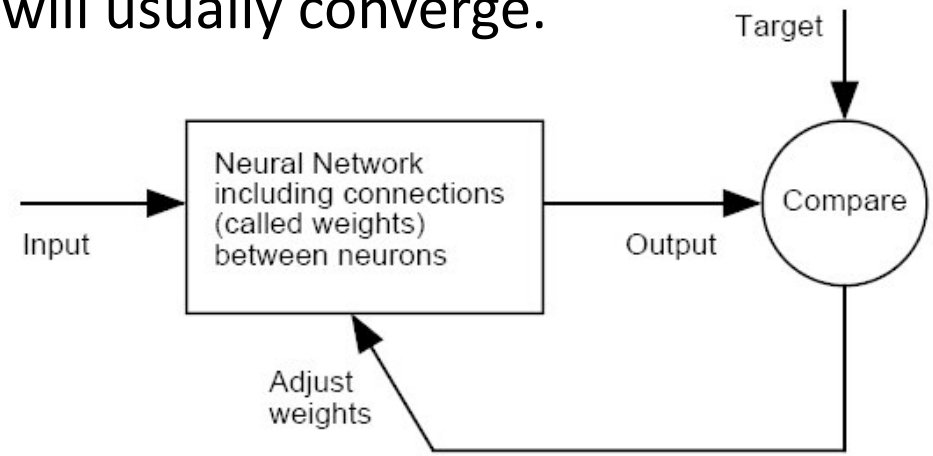
Supervised neural networks are adjusted, or trained, so that a particular input leads to a specific target output.

It requires that for a subset of data in the input space there must be an a accurate knowledge of the desired property (e.g. the real class).

Learning Process

- Back Propagation
  - the output values are compared with the target to compute the value of some predefined error-function;
  - the error is then fed back through the network;
  - using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function.

After repeating this process for a sufficiently large number of training cycles, the network will usually converge.



Generalization

- Generalization refers to the neural network ability to produce reasonable outputs for inputs not encountered during the training

In other words: NO PANIC when "never seen before" data are given in input!

Datasets

- Training set: a set of example used for learning, where the target value is known.
- Validation set: a set of examples used to tune the architecture of a classifier and estimate the error.
- Test set: a set of examples used only to assess the performance of a classifier.

The test set is never used during the training process so that the error on the test set provides an unbiased estimate of the generalization error.

Data Selection

- "Garbage in, garbage out": training, validation and test data must be representative of the underlying model;
- All eventualities must be covered
- Unbalanced data sets.
  - Since a network minimizes an overall error, the proportion of types of data in the set is critical;
  - inclusion of a loss matrix (Bishop, 1995);
  - often, the best approach is to ensure even representation of different cases, then to interpret the network's decisions accordingly.

Hidden Units

The best number of hidden units depends on:

- numbers of inputs and outputs
- number of training cases
- the amount of noise in the targets
- the complexity of the function to be learned
- the activation function

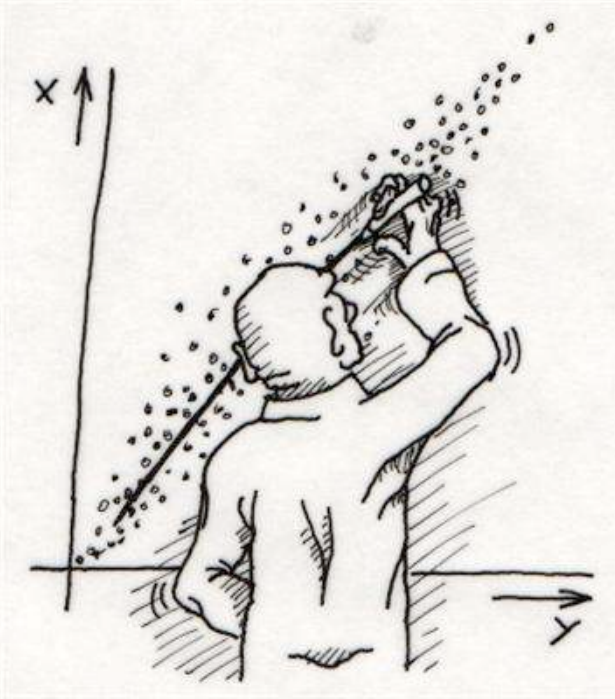
Too few hidden units => high training and generalization error, due to underfitting and high statistical bias.

Too many hidden units => low training error but high generalization error, due to overfitting and high variance.

MLP – Data Exploration

Regression

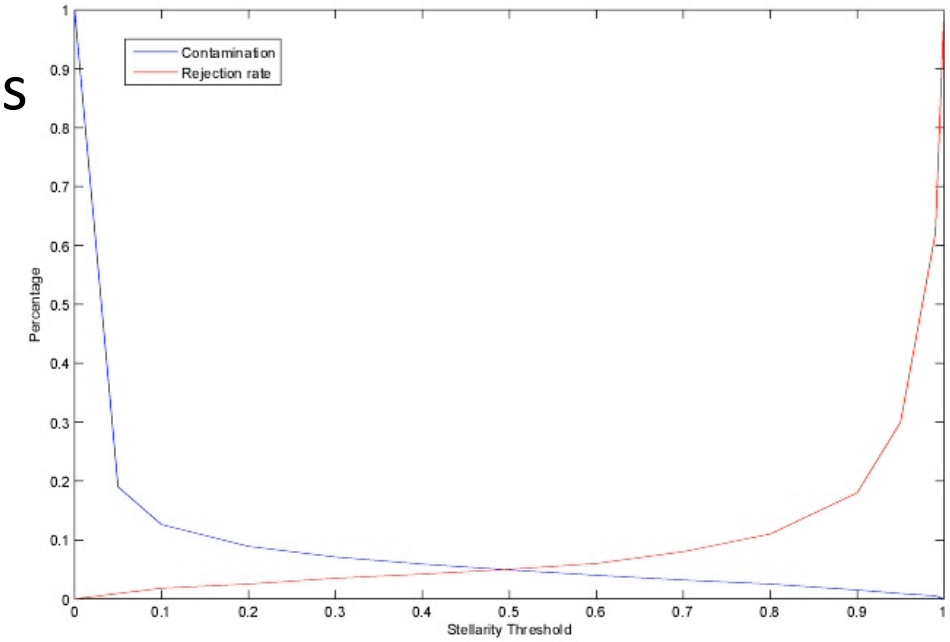
- Data table statistical correlation
  - mapping without any prior assumption on the functional form of the data distribution;
  - machine learning algorithms well suited for this.
- Curve fitting
  - find a well defined and known function underlying the data.



MLP – Data Exploration

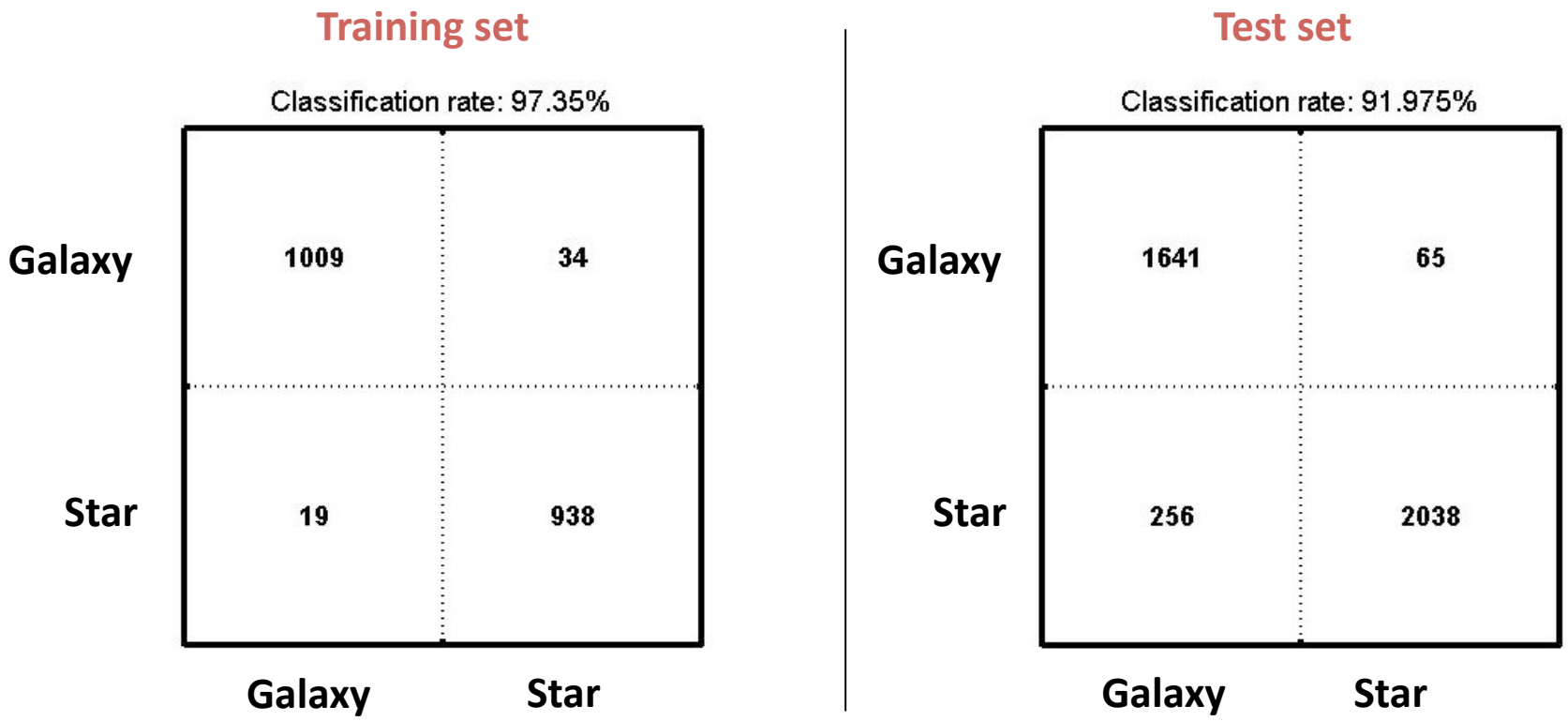
Classification

- Crispy classification
  - given an input, the classifier returns its label.
- Probabilistic classification
  - given an input, the classifier returns its probabilities to belong to each class
  - useful when some mistakes can be more costly than others
  - winner take all rule



Results: Confusion Matrix

In the confusion matrix the network prediction Y are compared with the target T: the rows represent the true classes and the columns the predicted classes.



Performances

The performances of the classifiers are rated based on the following three criteria.

Supposing we have 2 classes A and B:

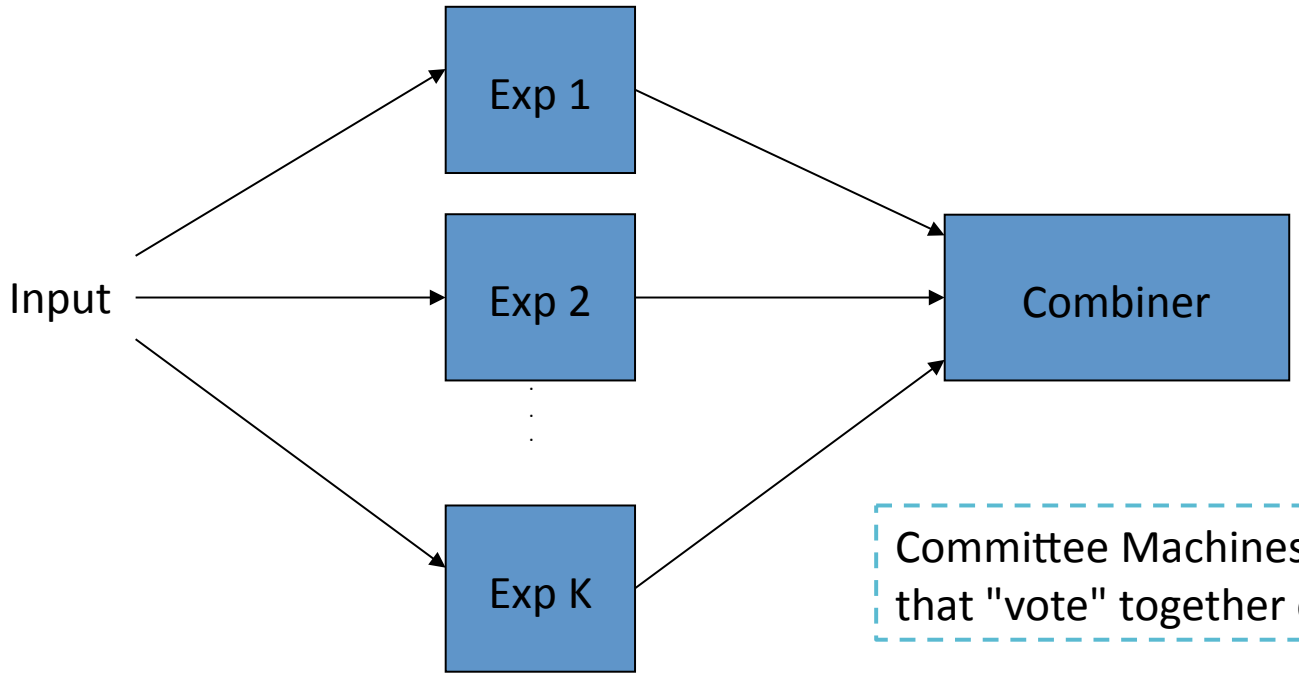
✓ **completeness**: the percentage of objects of class A correctly classified as such;

✓ **contamination**: the percentage of objects of class A incorrectly classified as objects belonging to the class B;

✓ **classification rate**: the overall percentage of objects correctly classified.

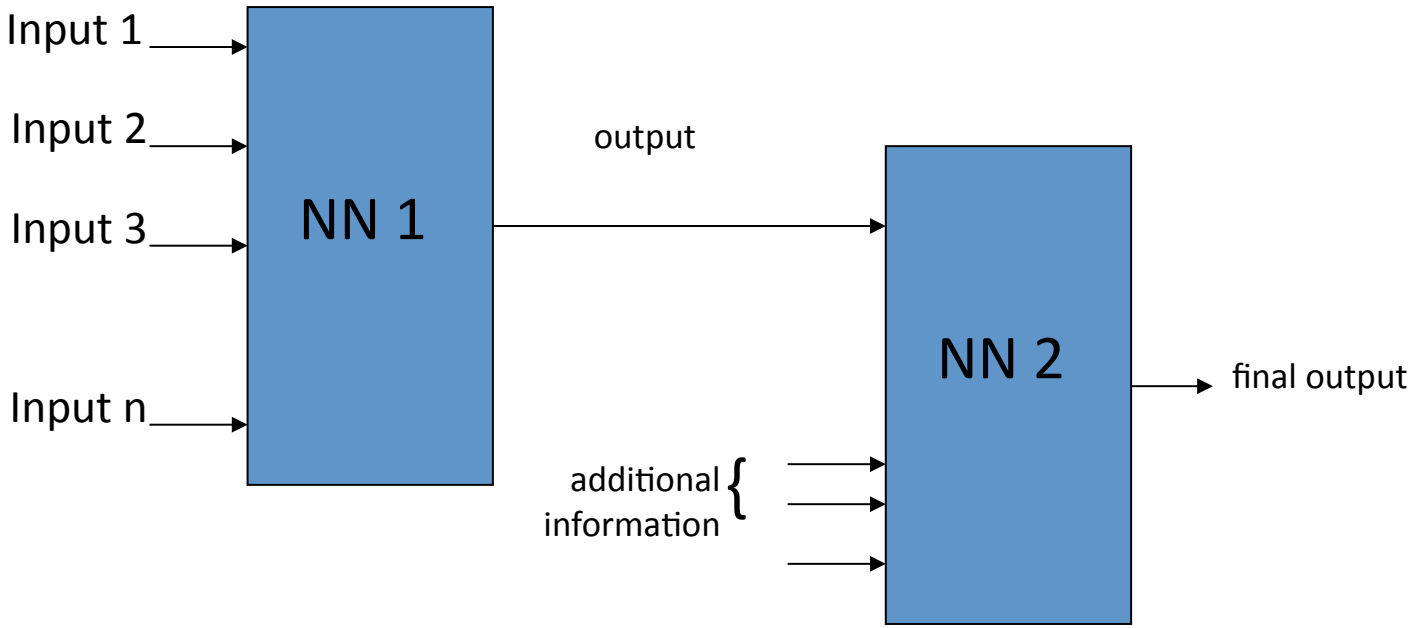
Combining Models

It is often found that improved performance can be obtained by combining models together in some way, instead of using a single model in isolation. In this way, individual classifiers may be optimized or trained differently.



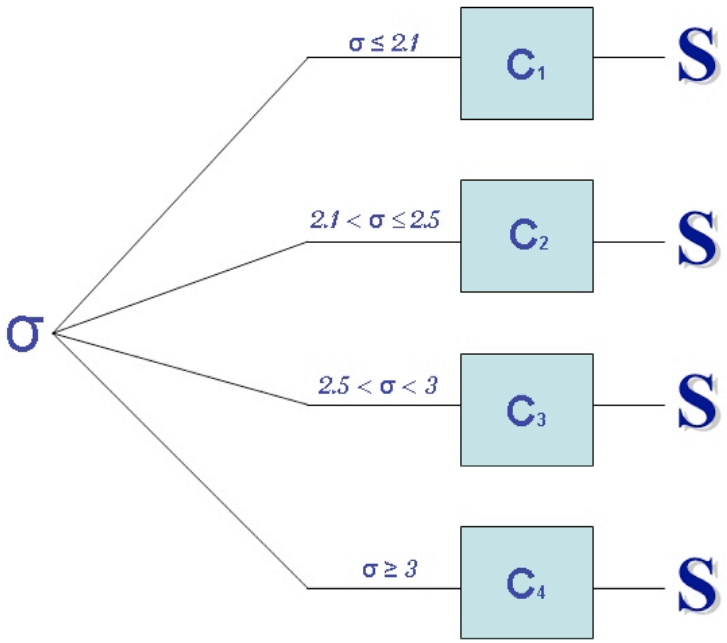
Committee Machines: combination of experts that "vote" together on a given example.

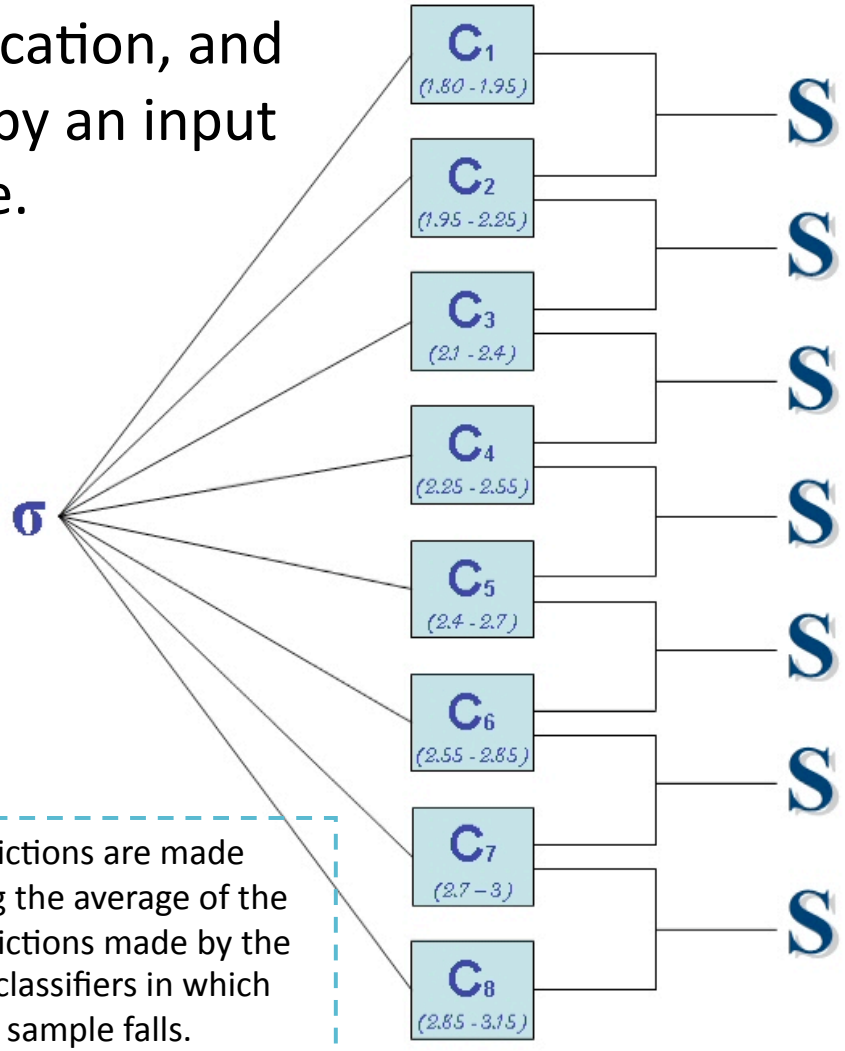
Combining Models



A priori Knowledge

An alternative of model combinations is to select one of the models to make the classification, and let the choice of the model be driven by an input parameter or by an a priori knowledge.





Predictions are made using the average of the predictions made by the two classifiers in which each sample falls.

Support Vector Machines

Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression.

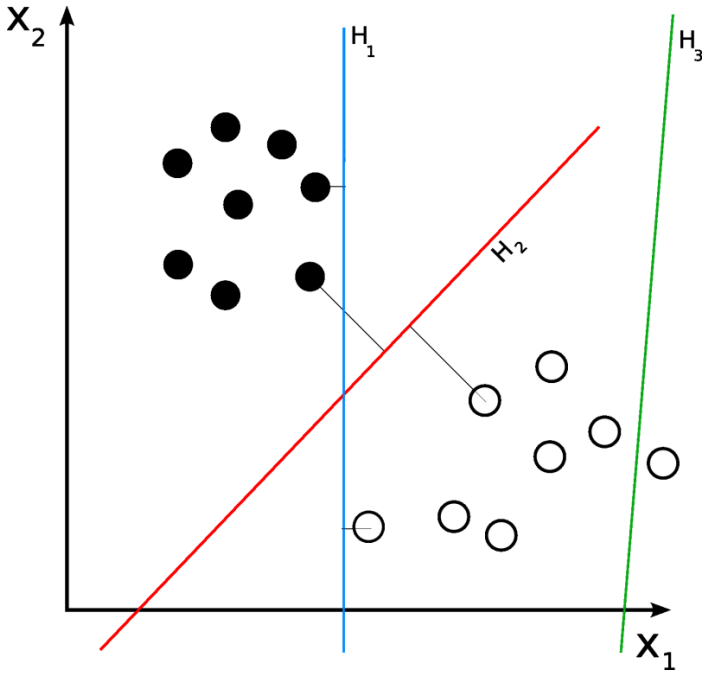
In a short period of time, SVM found numerous applications in a lot of scientific fields like physics, biology, chemistry:

- drug design (discriminating between ligands and nonligands, inhibitors and noninhibitors, etc.),
- quantitative structure-activity relationships (QSAR, where SVM regression is used to predict various physical, chemical, or biological properties),
- chemometrics (optimization of chromatographic separation or compound concentration prediction from spectral data as examples),
- sensors (for qualitative and quantitative prediction from sensor data),
- chemical engineering (fault detection and modeling of industrial processes),
- text mining (automatic recognition of scientific information)

SVM - Hyperplanes

SVM models were originally defined for the classification of linearly separable classes of objects.

For any particular set of two-class objects, an SVM finds the unique hyperplane having the maximum margin.



H3 (green) doesn't separate the 2 classes.

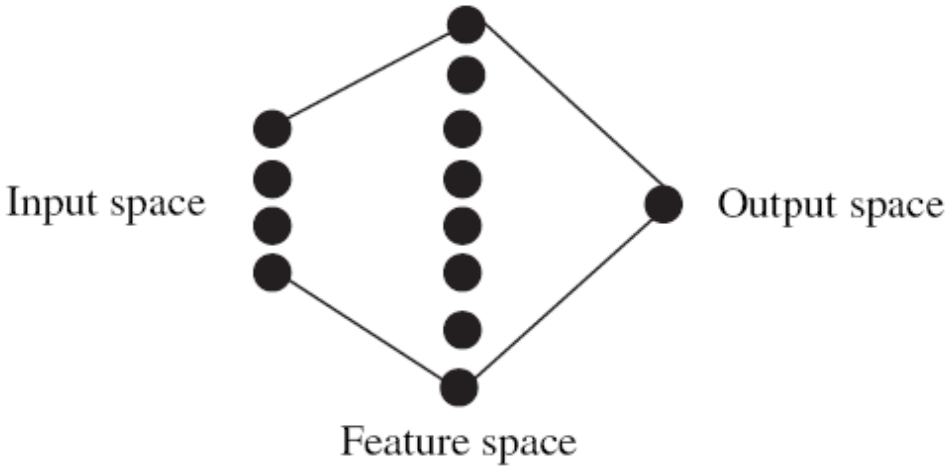
H1 (blue) does, with a small margin.

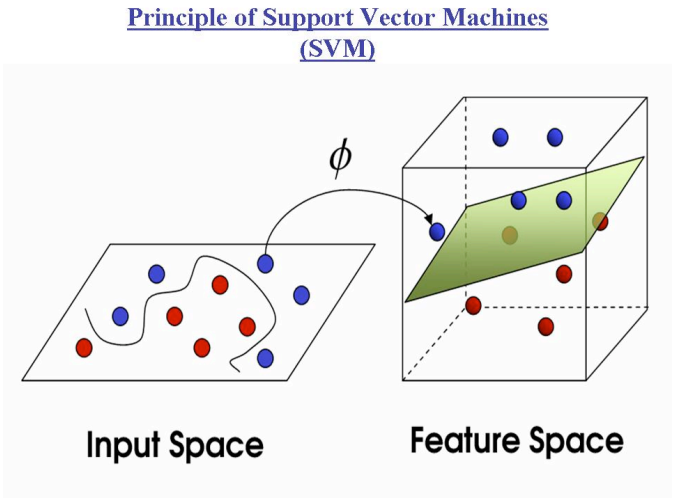
H2 (red) does with the maximum margin.

SVM - Classification

SVM can be used to separate classes that cannot be separated with a linear classifier.

Training vectors are mapped into an higher dimensional feature space using nonlinear functions φ. The feature space is a high-dimensional space in which the two classes can be separated with a linear classifier.





SVM - Classification

The mapping into a more dimensional space is done using the so called Kernel Functions.

- linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$.
- polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i{}^T \mathbf{x}_j + r)^d$, $\gamma > 0$.
- radial basis function (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$, $\gamma > 0$.
- sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i{}^T \mathbf{x}_j + r)$.

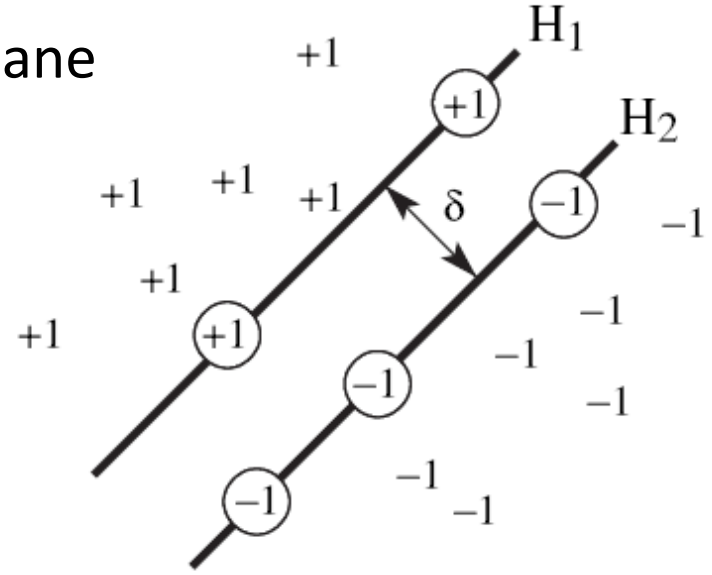
SVM – Support Vectors

A special characteristic of SVM is that the solution to a classification problem is represented by the support vectors that determine the maximum margin hyperplane.

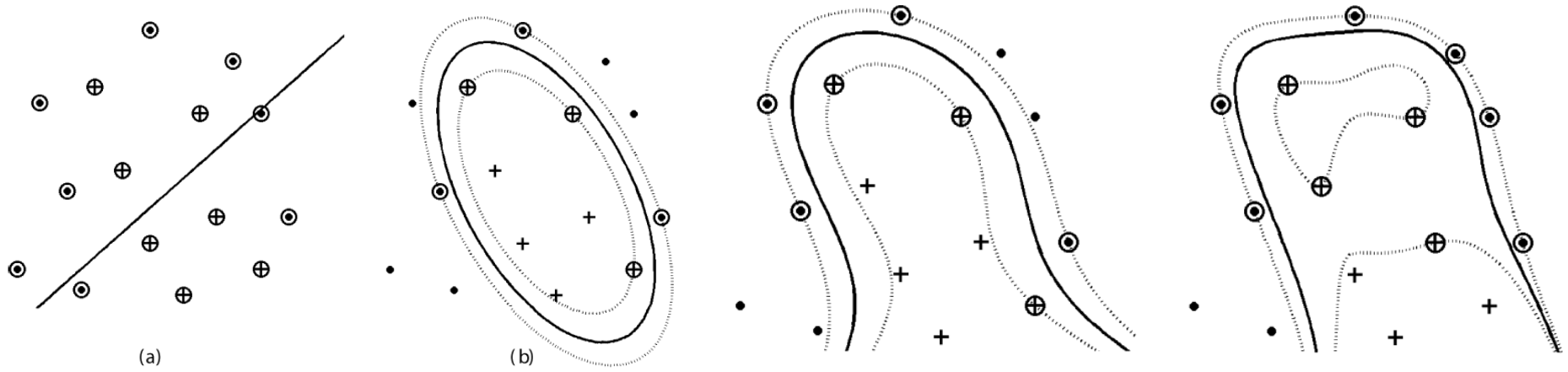
These objects, represented inside circles in Figure, are called support vectors.

The hyperplane H1 defines the border with class +1 objects, whereas the hyperplane H2 defines the border with class 1 objects.

Two objects from class +1 define the hyperplane H1, and three objects from class -1 define the hyperplane H2.



SVM - Example

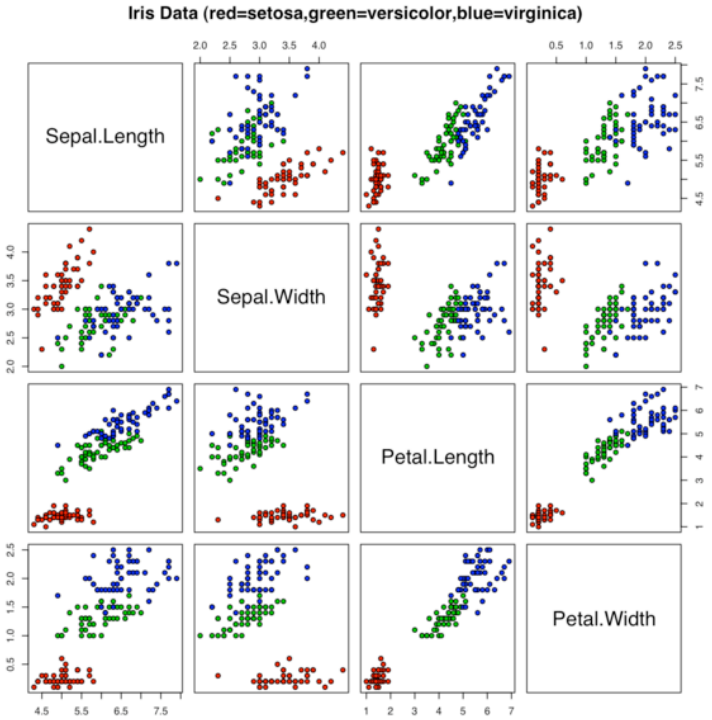


Linear Poly=2 Poly=3 Poly=10

- The linear kernel doesn't work
- The polynomials discriminate perfectly among the two class
  - avoid overfitting

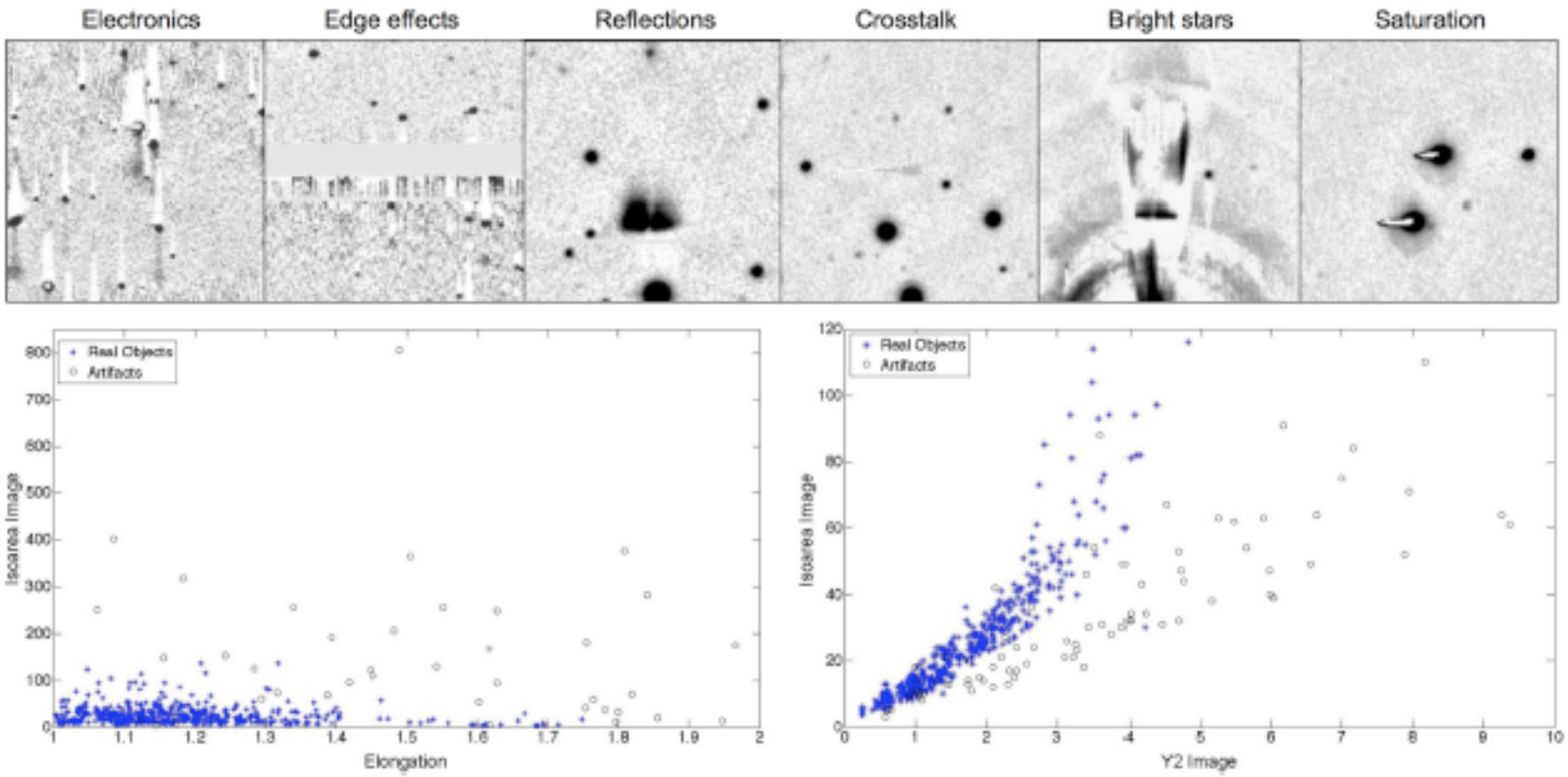
Datasets

- IRIS (Bi)
  - consists of 3 classes, 50 instances each and 4 numerical attributes (sepal and petal lengths and widths in cm);
  - each class refers to a type of Iris plant (Setosa, Versicolor, Verginica);
  - the first class is linearly separable from others while that latter are not linearly separable;



Datasets

- PQ Artifacts (Ay)
  - 2 main classes and 4 numeric attributes;
  - classes are: true objects, artifacts



Softwares

- FANN: Fast Artificial Neural Networks

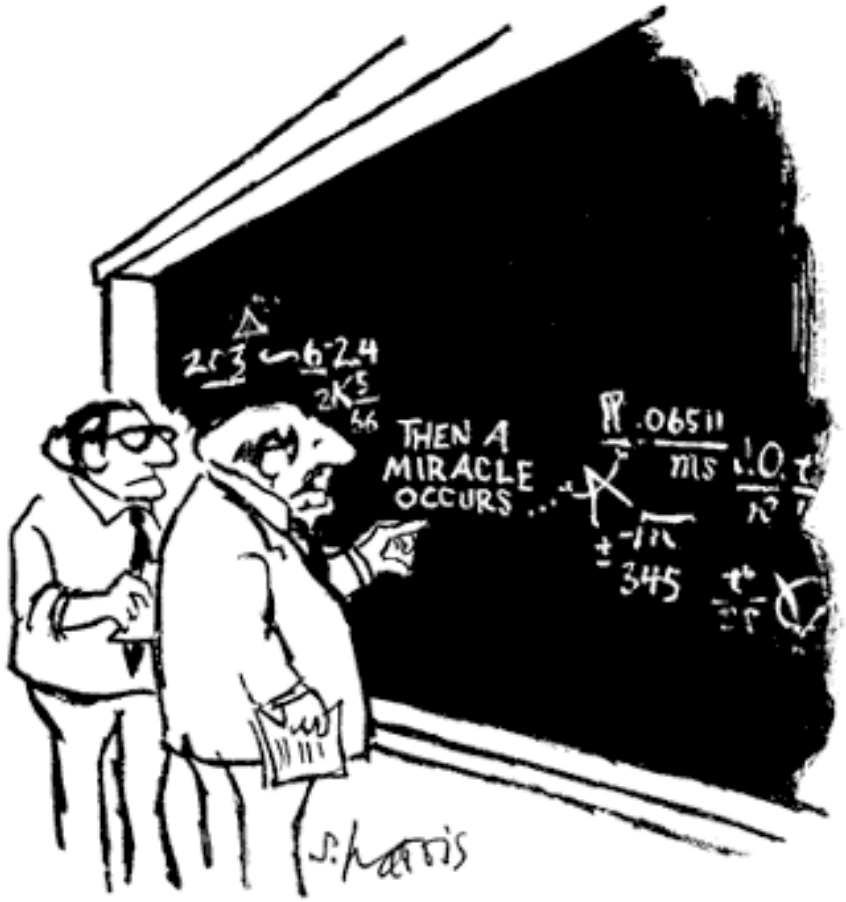
  http://leenissen.dk/fann/
- Netlab (Matlab toolbox)

  http://www.ncrg.aston.ac.uk/netlab/index.php
- LIBSVM

  http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- VONEURAL/DAME

  http://voneural.na.infn.it/

Send your comments...



"I think you should be more explicit here in step two."