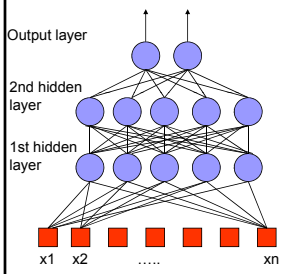
Various Neural Networks

Neural Networks

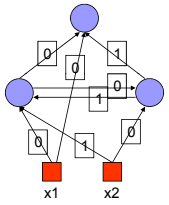
- A mathematical model to solve engineering problems
  - Group of connected neurons to realize compositions of non linear functions
- Tasks
  - Classification
  - Discrimination
  - Estimation
- 2 types of networks
  - Feed forward Neural Networks
  - Recurrent Neural Networks

Feed Forward Neural Networks



- The information is propagated from the inputs to the outputs
- Computations of functions from n input variables by compositions of algebraic functions
- Time has no role (NO cycle between outputs and inputs)

Recurrent Neural Networks



- Can have arbitrary topologies
- Can model systems with internal states (dynamic ones)
- Delays are associated to a specific weight
- Training is more difficult
- Performance may be problematic
  - Stable Outputs may be more difficult to evaluate
  - Unexpected behavior (oscillation, chaos, …)

Properties of Neural Networks

- Supervised networks are universal approximators networks)
- Theorem : Any limited function can be approximated by a neural network with a finite number of hidden neurons to an arbitrary precision

Supervised learning

- The desired response of the neural network in function of particular inputs is well known.
- A “Professor” may provide examples and teach the neural network how to fulfill a certain task

Unsupervised learning

- Idea : group typical input data in function of resemblance criteria un-known a priori
- Data clustering
- No need of a professor
  - The network finds itself the correlations between the data
  - Examples of such networks :
    - Kohonen feature maps

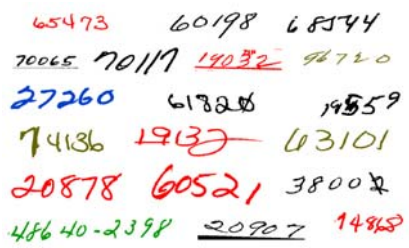
Classification (Discrimination)

- Class objects in defined categories
- Rough decision OR
- Estimation of the probability for a certain object to belong to a specific class

Example : Data mining

- Applications : Economy, speech and patterns recognition, sociology, etc.

Example



Examples of handwritten postal codes
drawn from a database available from the US Postal service

What needed to create NN ?

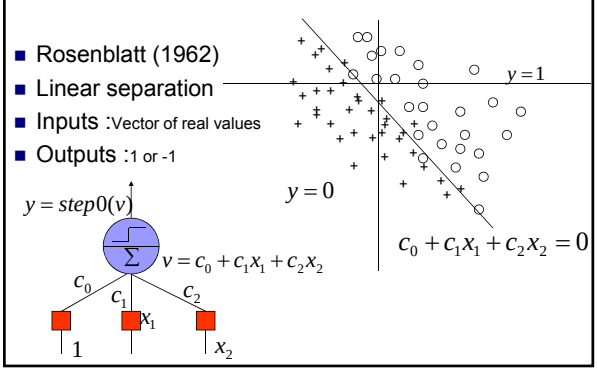
- Determination of relevant inputs
- Collection of data for the learning and testing phases of the neural network
- Finding the optimum number of hidden nodes
- Learning the parameters
- Evaluate the performances of the network
- If performances are not satisfactory then review all the precedent points

Popular neural architectures

- Perceptron
- Multi-Layer Perceptron (MLP)
- Radial Basis Function Network (RBFN)
- Time Delay Neural Network (TDNN)
- Other architectures

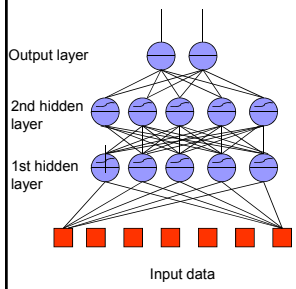
Perceptron

- Rosenblatt (1962)
- Linear separation
- Inputs :Vector of real values
- Outputs :1 or -1



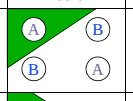
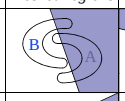
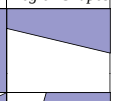
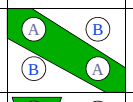
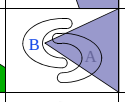
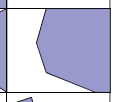
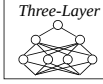
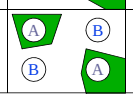
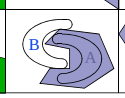
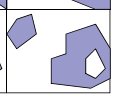
- The perceptron algorithm converges if examples are linearly separable

Multi-Layer Perceptron



- One or more hidden layers

Different non linearly separable problems

Structure	*Types of Decision Regions*	*Exclusive-OR Problem*	*Classes with Meshed regions*	*Most General Region Shapes*
Single-Layer 	*Half Plane Bounded By Hyperplane*			
Two-Layer 	*Convex Open Or Closed Regions*			
Three-Layer 	Abitrary (Complexity Limited by No. of Nodes)			

Radial Basis Functions

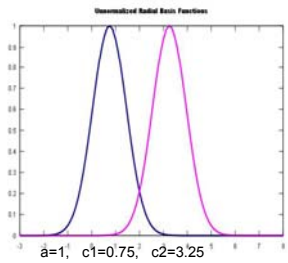
- A **radial basis function (RBF)** is a real-valued function whose value depends only on the distance from some other point *c*, called a *center*, $\varphi(\boldsymbol{x}) = f(||\boldsymbol{x}-\boldsymbol{c}||)$
- Any function φ that satisfies the property $\varphi(\boldsymbol{x}) = f(||\boldsymbol{x}-\boldsymbol{c}||)$ is a radial function.
- The distance is usually the Euclidean distance

$$|| x - c ||^2 = \sum_{i=1}^{N} \left(x_i - c_i\right)^2$$

Radial Basis Functions

- The popular output of radial basis functions is the Gaussian function:

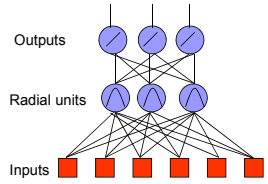
$$\Phi\left(\left\|x - c_j\right\|\right) = \exp\left(-a\left(\frac{\left\|x - c_j\right\|}{\sigma_j}\right)^2\right)$$

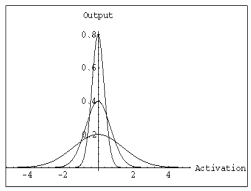


a=1, c1=0.75, c2=3.25

Radial Basis Functions Network (RBFN)

- Features
  - One hidden layer
  - The activation of a hidden unit is determined by a radial basis function





- Generally, the hidden unit function is the Gaussian function
- The output Layer is linear:

$$s(x) = \sum_{j=1}^{K} W_j \Phi\left(\left\| x - c_j \right\|\right)$$

$$\Phi\left(\left\| x - c_j \right\|\right) = \exp\left(-w_j \left(\frac{\left\| x - c_j \right\|}{\sigma_j} \right)^2\right)$$

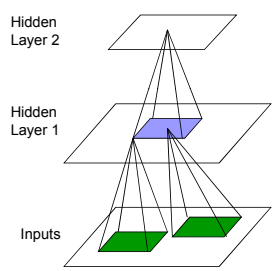
RBFN Learning

- The training is performed by deciding on
  - How many hidden nodes there should be
  - The centers and the sharpness of the Gaussians
- 2 steps
  - In the 1st stage, the input data set is used to determine the parameters of the RBF
  - In the 2nd stage, RBFs are kept fixed while the second layer weights are learned (Simple BP algorithm like for MLPs)

Time Delay Neural Network (TDNN)

- Introduced by Waibel in 1989
- Properties
  - Local, shift invariant feature extraction
  - Notion of receptive fields combining local information into more abstract patterns at a higher level
  - Weight sharing concept (All neurons in a feature share the same weights)
    - All neurons detect the same feature but in different position
- Principal Applications
  - Speech recognition
  - Image analysis

TDNNs (cont'd)



- Objects recognition in an image
- Each hidden unit receive inputs only from a small region of the input space : receptive field
- Shared weights for all receptive fields => translation invariance in the response of the network

- Advantages
  - Reduced number of weights
    - Require fewer examples in the training set
    - Faster learning
  - Invariance under time or space translation
  - Faster execution of the net (in comparison of full connected MLP)

Summary

- Neural networks are utilized as statistical tools
  - Adjust non linear functions to fulfill a task
  - Need of multiple and representative examples but fewer than in other methods
- Neural networks enable to model complex static phenomena (Feed-Forward) as well as dynamic ones (Recurent NN)
- NN are good classifiers BUT
  - Good representations of data have to be formulated
  - Training vectors must be statistically representative of the entire input space
  - Unsupervised techniques can help
- The use of NN needs a good comprehension of the problem