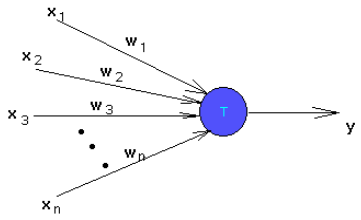
Multiple Layer Perceptron

Dima Novikov
Roman Yampolskiy

Perceptron

review

The Perceptron



The Perceptron

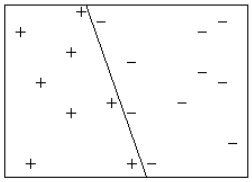
- The *Perceptron* is a kind of a single-layer artificial network with only one neuron
- *The Percepton is a network in which the neuron unit calculates the linear combination of its real-valued or boolean inputs and passes it through a threshold activation function:*

  ***o = Threshold*(S *i=0d wi xi*)**

The Perceptron

- where *xi* are the components of the input **x***e* = (*xe1*, *xe2*,..., *xed*) from the set {(**x***e*, *ye*)} *e* = 1N
- *Threshold* is the activation function defined as follows: *Threshold* (*s*) = 1 if *s* > 0 and – 1 otherwise

The Perceptron is strictly equivalent to a linear discriminant, and it is often used as a device that decides whether an input pattern belongs to one of two classes.

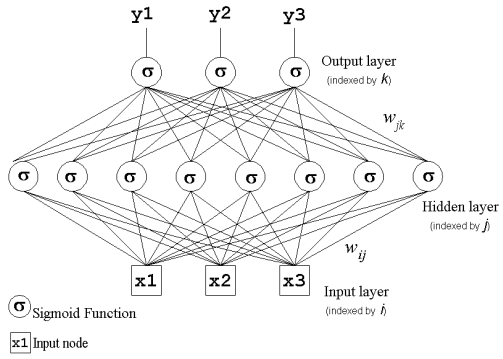


- *Networks of such threshold units can represent a rich variety of functions while single units alone can not*. For example, every boolean function can be presented by some network of interconnected units.
- The Perceptron can represent most of the primitive boolean functions: AND, OR, NAND and NOR but can not represent XOR.

Multilayer Perceptron

- The *multilayer perceptron* (MLP) is a hierarchical structure of several perceptrons, and overcomes the shortcomings of these single-layer networks.
- The *multilayer perceptron* is an artificial neural network that learns nonlinear function mappings. The multilayer perceptron is capable of learning a rich variety of nonlineardecision surfaces.

- The multilayer *network structure*, or *architecture*, or *topology*, consists of an input layer, one or more hidden layers, and one output layer. The input nodes pass values to the first hidden layer, its nodes to the second and so on till producing outputs.
- The training algorithm for multilayer networks requires differentiable, continuous nonlinear activation functions. Such a function is the *sigmoid*.



MLP network and RBF network

- Both networks can function as a **Universal Approximator**

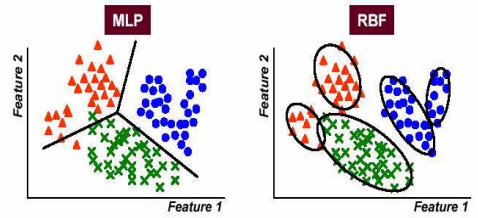
  - Either model can approximate any functional continues mapping with arbitrary accuracy

MLP network VS. RBF network

- Performs a global and distributed approximation of the target function (many hidden units contribute to the output for a given input)
- Great for problems with many input variables
- Feature space is partitioned with hyper-plane

- Performs a local approximation (only a few hidden units are active for a given input)
- Prefers low dimensionality input space
- Feature space is partitioned with hyper-sphere

MLP network VS. RBF network



MLP network VS. RBF network

- Superior generalization properties in regions of feature space outside of the local neighborhoods defined by the training set (may result in unreasonable generalizations)
- Requires fewer parameters to approximate a non-linear function with the same accuracy
- Inferior generalization ability
- Requires additional parameters

MLP network VS. RBF network

- All parameters are trained simultaneously
- Multiple hidden layers(with complex connectivity)
- Parameters in the hidden and output layers are trained separately(using a fast and efficient hybrid algorithm)
- Only one hidden layer(with full connectivity)

MLP network VS. RBF network

- Hidden neurons compute the inner product between an input vector and their weight vector
- All layers are typically non-linear (except than doing non-linear regression, which uses a linear output layer)
- Computes the Euclidian distance between an input vector and the radial basis centers
- Hidden layer is non-linear, but output layer is linear

MLP VS. RBF Conclusions:

- MLPs are better at generalizing
- MLPs require fewer hidden nodes
- If data is hard to obtain – use MLP!

Demo Applets:

- http://diwww.epfl.ch/mantra/tutorial/english/perceptron/html/
  http://neuron.eng.wayne.edu/bpBallBalancing/ball5.html
  http://members.aol.com/Trane64/java/JRec.html
  http://titan.glo.be/mark.casselman/ja/world.htm

Slides are based on the following resources:

- Ricardo Gutierrez-Osuna, "*MLPs, RBFs and Statistical PR*"
  http://faculty.cs.tamu.edu/rgutier/courses/cpsc689_f02/l20.pdf
- Amir Hussain, "*Biologically Inspired Computing*"
  www.cs.stir.ac.uk/~ahu/31YB/lecture11.pdf
- Olle Gällmo, ***Course on Artificial Neural Networks***
  user.it.uu.se/~crwth/ann_course/VT02/L9.pdf