



Artificial Neural Networks

Andreas Rauber, Alexander Schatten



- „Natural Computation“, Bionik
- The Brain Analogy, Some Inspirations
- Adaptive Systems
- Resources
- Neurons: Brain to Artificial
- Perceptron
- Learning
- Network Types
- Typical Applications

„Natural Computation“

- Technical Solutions *inspired* by natural solutions
- Precise modelling of natural process not necessary, not even desired!
- Bionik
 - Mechanical Engineering
 - Chemical Technology
- Natural Computation
 - Artificial Neural Networks
 - Genetic/Evolutionary Algorithms
 - Cellular Automata
 - ...

The Brain

- Exact work of brain still unknown
- 100 billion neurons
- Each neuron is can connect with up to 200.000 other neurons, typical is 1000 to 10.000
- Functionality of single neuron is *very* complex

Inspired...

- Animals, humans are able to
 - adapt to changes in environment
 - learn
 - solve problems
 - ...
- Artificial Neural Network *could* be used for similar problem solving in technological problems
- Idea: Complexity arises not from complex setup but from interaction of relatively simple units

Adaptive Systems

- Adaptive systems provide an alternative to traditional system design: instead of creating a mechanism to perform function F , we create a mechanism that will learn or adapt itself in order to perform F .
- An adaptive system generally has two distinct components:
 - training (or feedback)
 - operation.
- The advantages of such systems are often critical: flexibility and power.

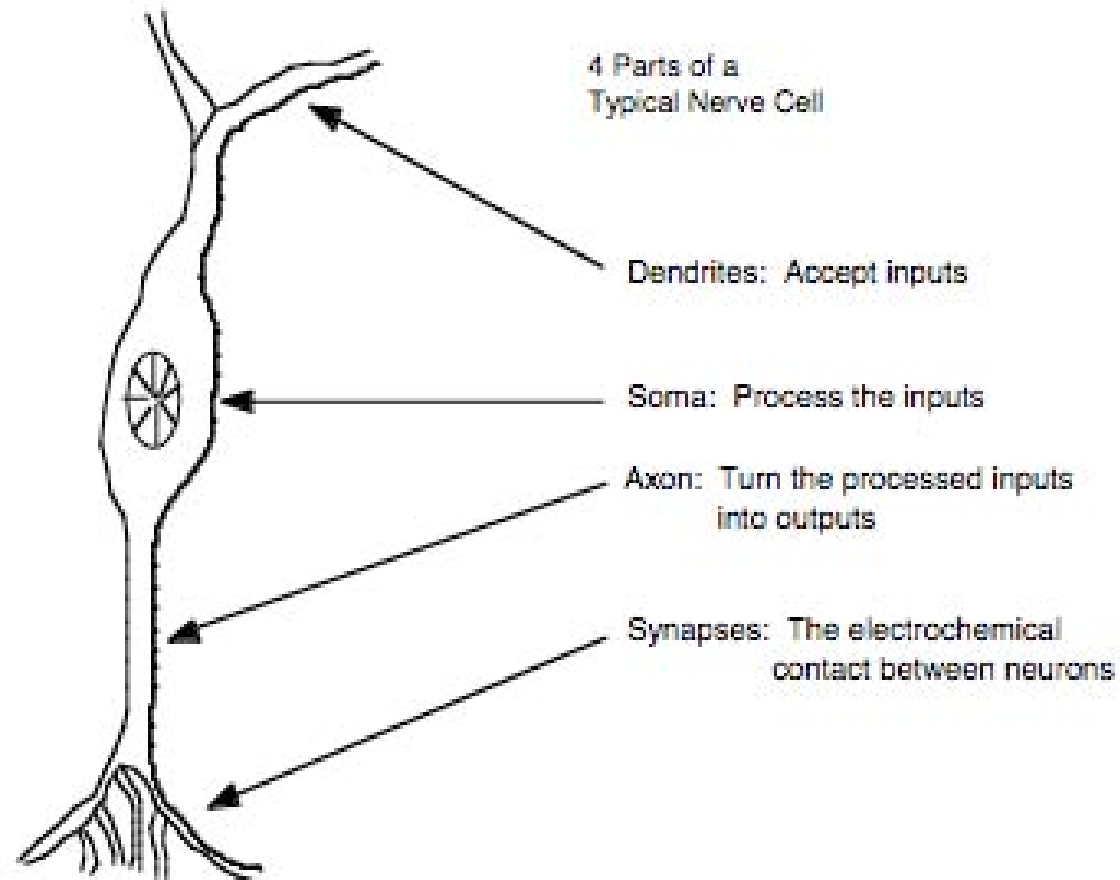
Generalisation

- ANN must not be „overtrained“
- Generalisation desired!
- Test training quality!
 - E.g. using cross validation

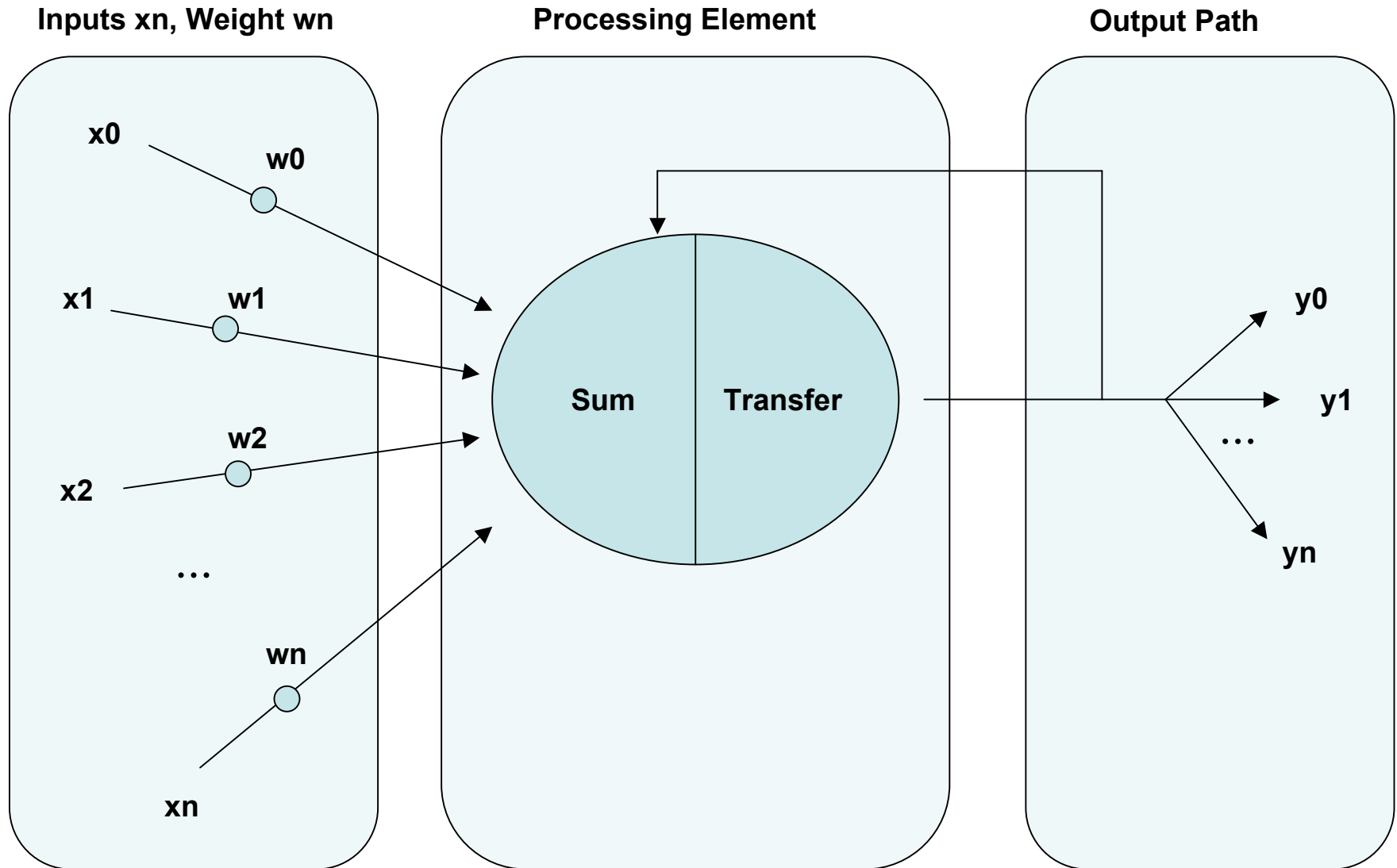
Resources

- <http://www.dbourget.com>
 - ANN for Philosophers
- <http://www.dacs.dtic.mil/>
 - DACS State-of-the-Art Report
- <http://neuralnetworks.ai-depot.com/Tutorials.html>
 - Examples, Applets...

Simple Neuron



Artificial Neuron 1



Artificial Neurons 2

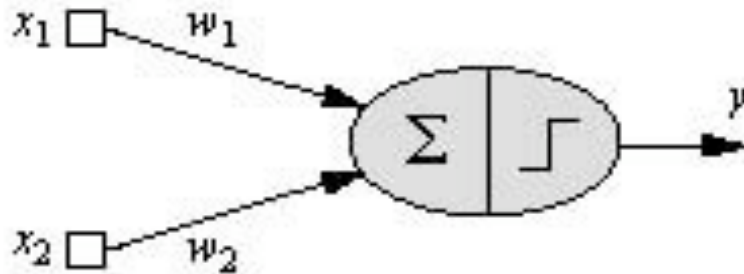


Fig1: an artificial neuron

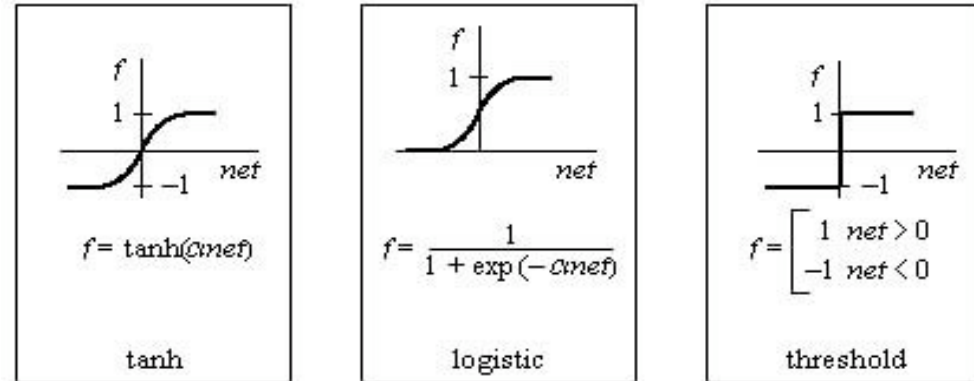


Fig2: common (non-linear) activation functions

- An artificial neuron has three components:
 - Input connections (x_1, x_2, \dots) with weights (w_1, w_2, \dots) (the equivalent of dendrites)
 - An output (y) (the equivalent of an axon eventually linked with other neurons' "dendrites").
 - A computing unit (the equivalent of a living neuron's soma). The computing unit constantly sums the weighted inputs to produce a *net* out of which the output is calculated using a non-linear activation function (f) :

$$y = f\left(\sum w_i x_i\right)$$

The Perceptron

- The (multi-layer) perceptron builds a network
- A perceptron embodies a function $X \rightarrow Y$
- Usually, the output layer has less neurons (less dimensions) than the input layer so the perceptron acts as a classifier.

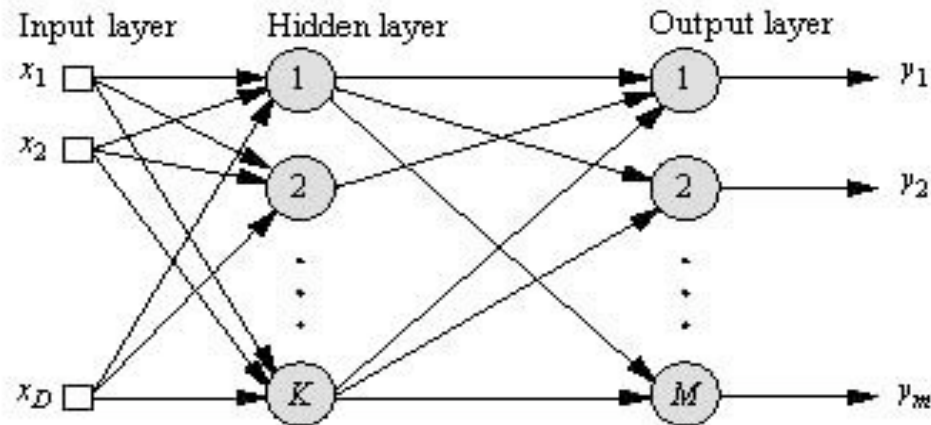
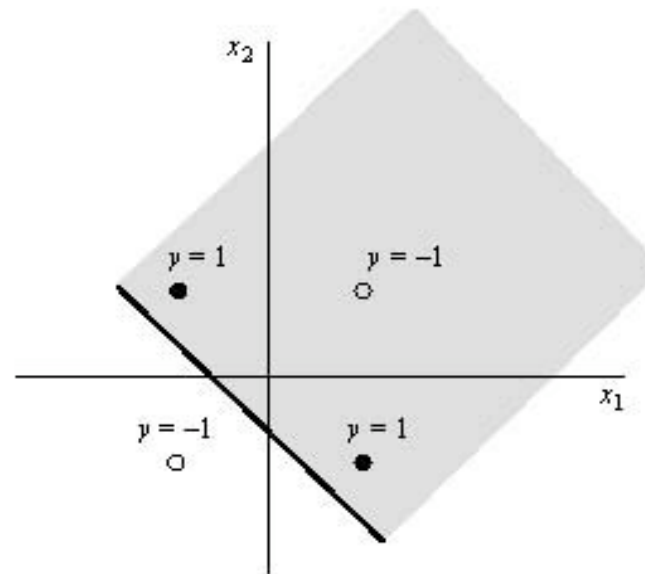


Fig3: a multi-layer perceptron

Limitations of perceptrons

- Perceptrons without hidden layers can only recognize linearly-separable classes.
- Perceptrons with sufficient layers and connections can implement any function whatsoever.

x_1	x_2	y
-1	-1	-1
1	-1	1
-1	1	1
1	1	-1



Learning

- Supervised
 - Input & Output Provided
- *Unsupervised*
 - No output provided
 - Not well understood so far

Network Types

- Feedforward, Back Propagation
- Delta Bar Delta
- Radial Base Functions
- Hopfield Networks
- Probabilistic Networks
- Self-Organising Map
- ...



Typical Applications

- Character Recognition
- Modeling (Functions)
- Signal Processing
- Financial Applications
- Machine Control

Disadvantages of ANN

- In some cases: difficult parameter setup
- Learning Time
- ANN is a „blackbox tool“
 - If applied properly, one receives a working system, but no idea about the underlying model
 - Deriving „real world models/meaning“ from ANN parameters is hardly possible
- ANNs cannot extrapolate! Errors at boundaries are increasing dramatically (e.g. at RBF type)



Demonstration

- Demonstration of
 - Neuron/Perceptron functionality
 - Different network models
 - typical ANN Application