Introduction and overview

Johan Suykens

KU Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium
Email: johan.suykens@esat.kuleuven.be
http://www.esat.kuleuven.be/stadius

Lecture 1

Overview of the lectures

1. Introduction and overview
2. Multilayer feedforward networks and backpropagation
3. Training of feedforward neural networks
4. Generalization
5. Bayesian learning of neural networks
6. Recurrent neural networks
7. Unsupervised learning
8. Nonlinear modelling and control
9. Support vector machines
10. Deep learning

(Material available in Toledo)

Overview of the exercise sessions

1. Supervised learning and generalization

2. Bayesian learning of neural networks

3. Recurrent neural networks

4. Unsupervised learning: PCA and SOM

5. Two project works

(Matlab computer exercise sessions; info available in Toledo)

Reporting

1. Supervised learning and generalization (max 2 pages)

2. Bayesian learning of neural networks (max 2 pages)

3. Recurrent neural networks (max 2 pages)

4. Unsupervised learning: PCA and SOM (max 2 pages)

5. Two project works (max 4 pages per project)

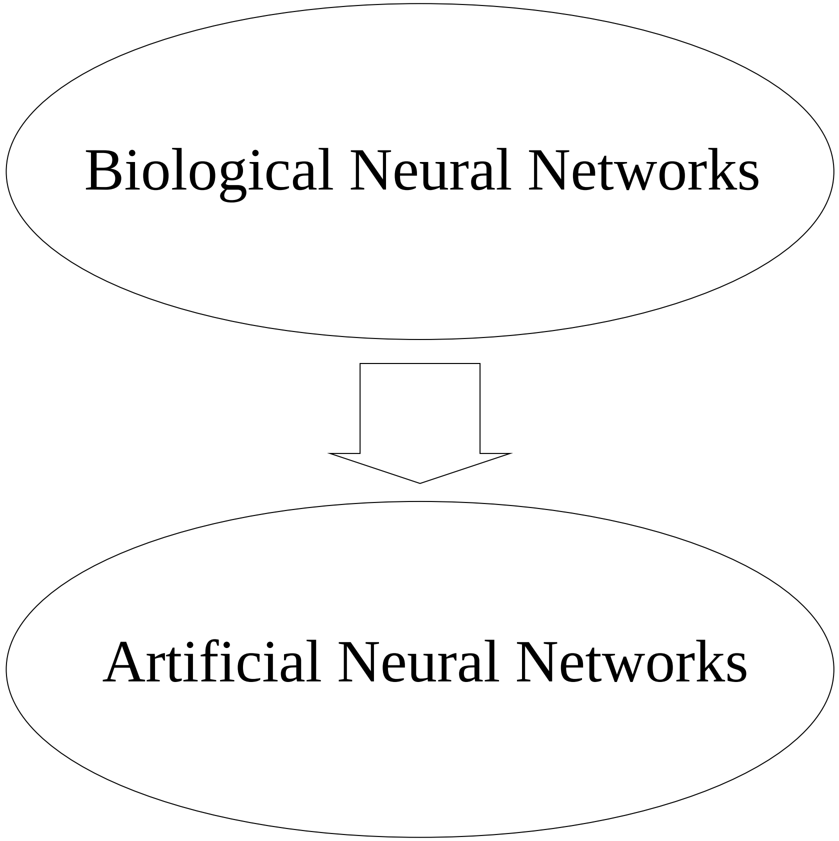
Max number of pages of of the total report:
4 x 2 pages + 2 x 4 pages = 16 pages (including figures)

Exam

- Discussion on the report (score: 12/20) ($4 \times 1.5 + 2 \times 3 = 12$)
- one question from list lectures 1-5 (score: 4/20)
- one question from list lectures 6-10 (score: 4/20)

The exam is open book

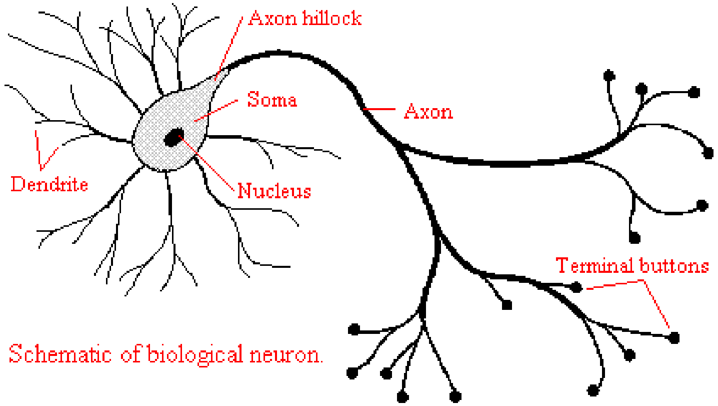
Biological $\neq$ Artificial Neural Networks



(very strong mathematical abstractions!)

Biological neurons

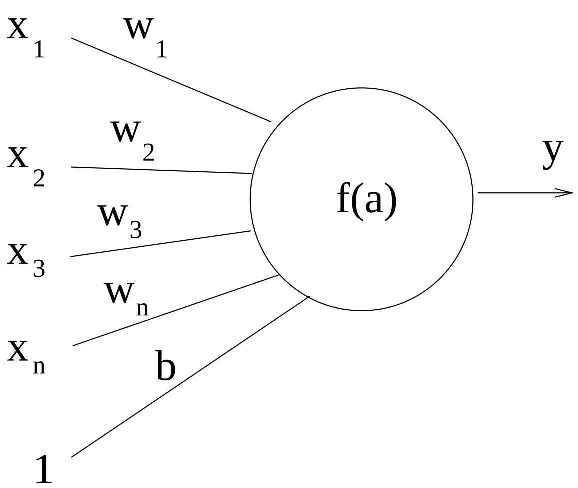
One estimates that the human brain contains over 10^{11} neurons and 10^{14} synapses in the human nervous system. The neuron's switching time is much slower than for transistor computer elements, but the connectivity is higher than in today's supercomputers.



A neuron has three main parts:
- neuron cell body
- branching extensions called dendrites for receiving input
- an axon that carries the neuron's output to the dendrites of other neurons

McCulloch-Pitts model of a neuron

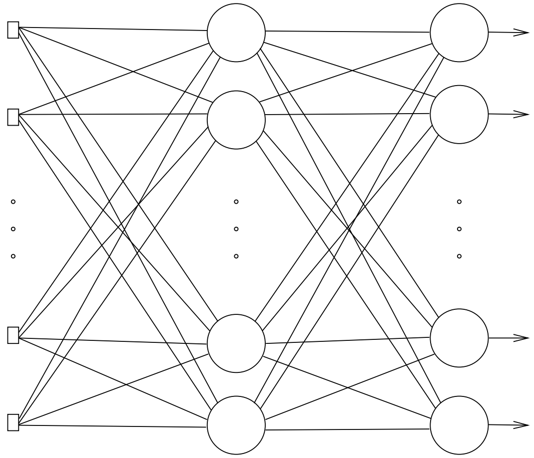
Incoming signals x_i
Interconnection weights w_i
Bias term (threshold value) b
Activation a
Nonlinearity $f(\cdot)$
Output y



$$a = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$$

$$y = f(a)$$

Multilayer perceptron (MLP)



Input $x \in \mathbb{R}^m$, output $y \in \mathbb{R}^l$, hidden layer: n_h hidden neurons
Interconnection weight matrices: $W \in \mathbb{R}^{l \times n_h}$, $V \in \mathbb{R}^{n_h \times m}$
Bias vector (thresholds of hidden neurons): $\beta \in \mathbb{R}^{n_h}$

Matrix-vector notation

$$y = W\, \sigma(V x + \beta)$$

Elementwise

$$y_i = \sum_{r=1}^{n_h} w_{ir}\, \sigma(\sum_{j=1}^{m} v_{rj} x_j + \beta_r),\ i = 1, ..., l$$

Learning = adapting weights from examples

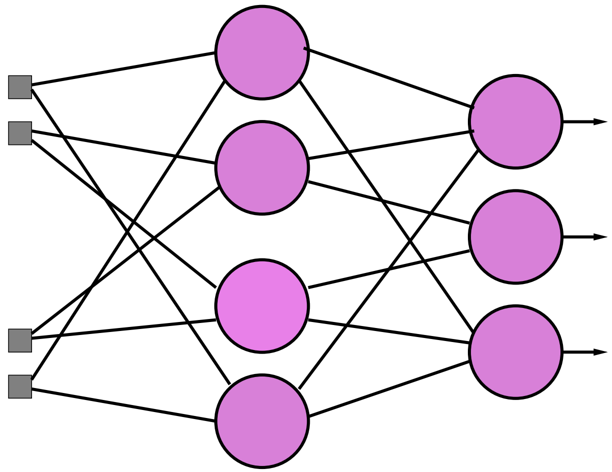
- weights adapted during learning or training

- learning rule adaptation of the weights according to the examples

- a neural network learns from examples e.g. children classify animals from living examples and photographs

- neural networks obtain their information during the learning process and store the information in the weights

- But, a neural network can learn something unexpected

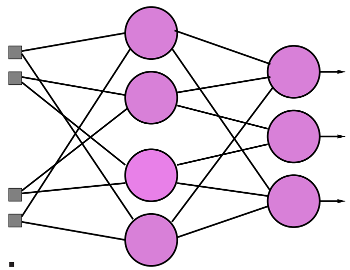


Digital computer vs neural network

Digital computer:

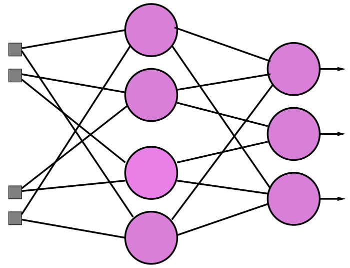
- **working principle:** symbols 1 and 0;
program von Neumann principle;
mathematical logic and Boolean algebra;
programs with software algorithms
- **parallellisation is difficult:** sequential processing of data
- **useless without software**
- **rigid:** modifying one bit could cause a disaster

Neural network:



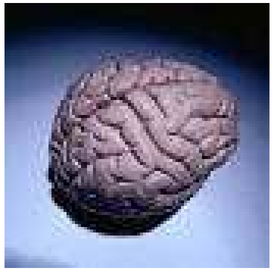
- **working principle:** learn a nonlinear map
from given patterns; mathematics of nonlinear
functions or dynamical systems; need for design methodologies
- **parallellisation is easy:** parallel by definition (like the brain),
training needed
- **choice of learning rule and examples crucial**
- **robust against inaccuracies** in data, defect neurons and error-correcting
capability (collective behavior in the brain)

Neural networks vs human brain



Neural networks

- **low complexity:** electronic VLSI chips and simulations on computers with significantly less neurons than the brain
- **high processing speed**
- **energetic inefficiency:** computers consume 10^{-6} Joule per operation and per sec



Human brain:

- **high complexity:**
human brain 100.000.000.000 neurons
- **low processing speed:**
reaction time of biologic neural networks: 1 to 2 msec.
- **energetic efficiency:** biologic neural network much better: 10^{-16} Joule per operation and per sec

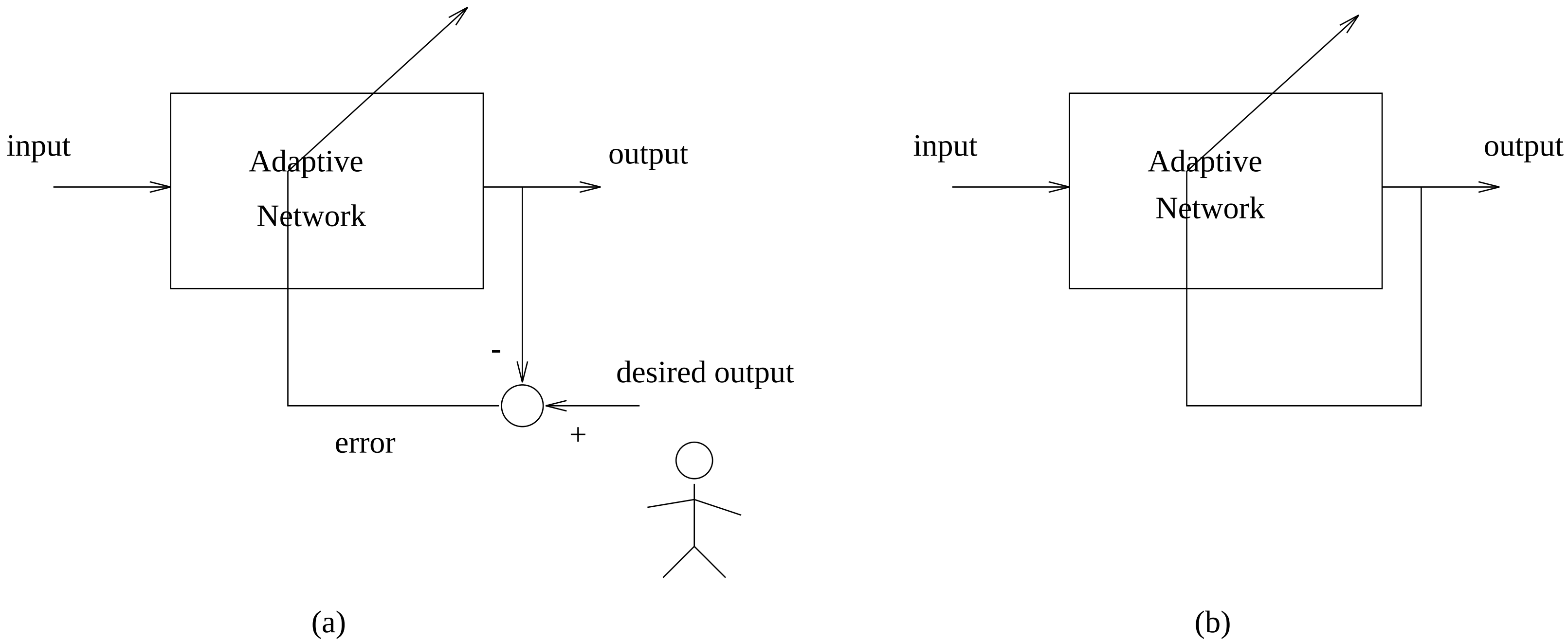
History of Neural Networks

1942 McCulloch and Pitts: mathematical models for neurons
1949 psychologist Hebb first learning rule (memorize by adapting weights)
1958 Rosenblatt: book on perceptrons, a machine capable to classify information by adapting weights
1960 Widrow and Hoff: adaline and LMS learning rule
1969 Minsky and Papert prove limitations of perceptron

13 years of hibernation, but some persistent researchers: Grossberg (US), Amari and Fukushima (Japan), Kohonen (Finland) and Taylor (UK)

1982 Kohonen describes the self-organizing map
1986 Rumelhart rediscovers backpropagation
1987 and later: much research on neural networks, new journals, conferences, applications, products, industrial initiatives, startup companies

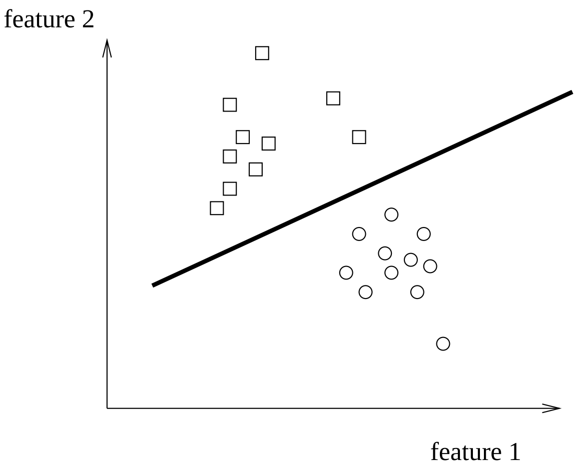
Supervised vs unsupervised learning



Block diagram that illustrates the basic difference between: (a) supervised learning and (b) unsupervised learning. In supervised learning there is a teacher, someone who tells the neural net how certain given inputs have to match certain desired outputs. This information will influence the way in which the synaptic (interconnection) weights are adapted. In unsupervised learning there is no external teacher.

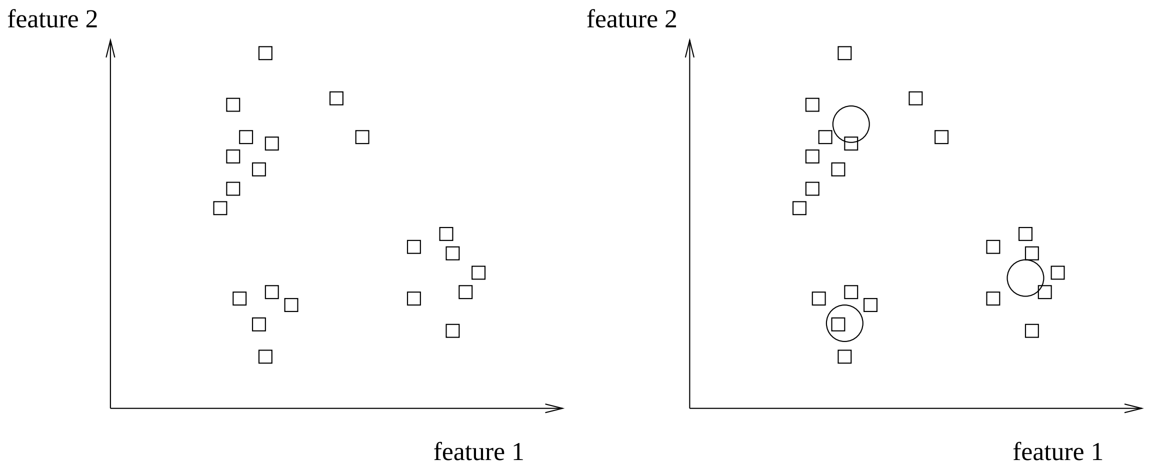
Classification vs clustering

Supervised:



Training data with given labels

Unsupervised:

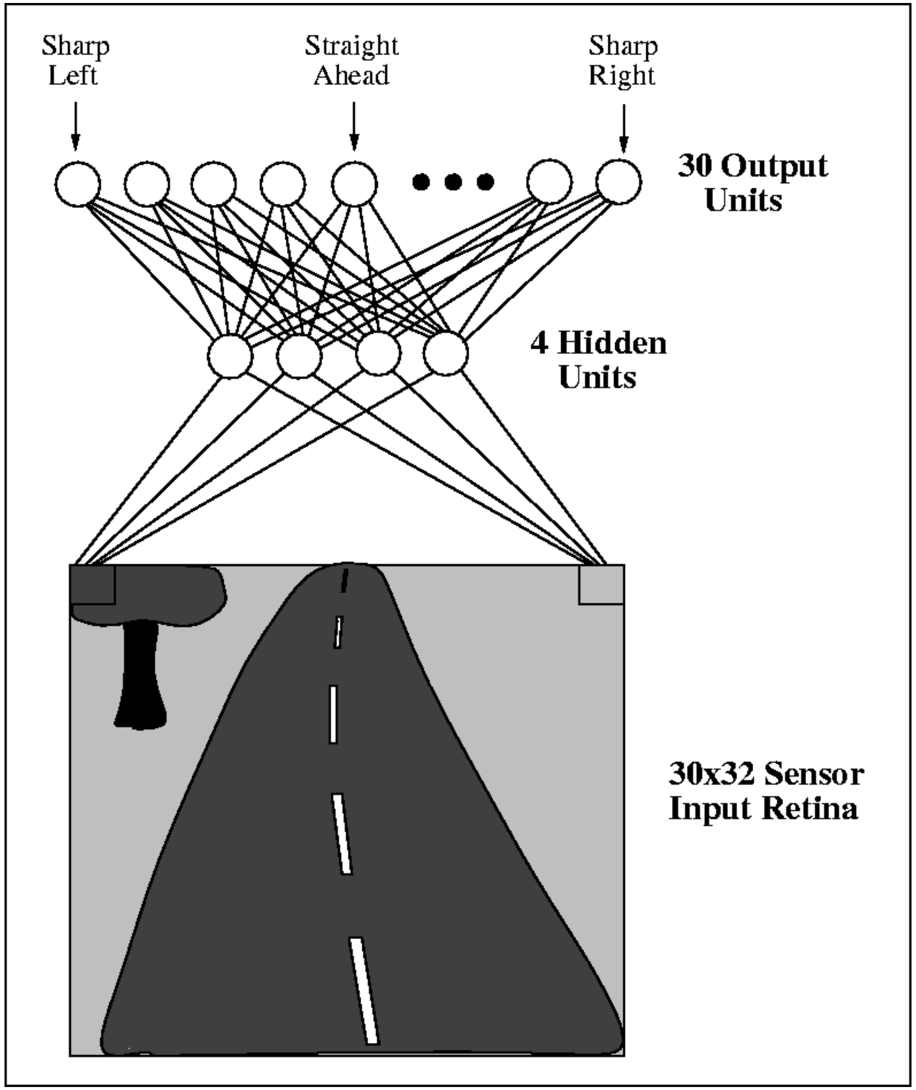


Discovering clusters (no given class labels)

Training neural networks: local minima problem

Existence of many local minima, depending on the choice of initial weights.

Applications


Sharp
Left
Straight
Ahead
Sharp
Right
30 Output
Units
4 Hidden
Units
30x32 Sensor
Input Retina

Autonomous vehicle control with a neural network (ALVINN)

- **goal:** keep the vehicle without driver on the road
- **car equipped** with videorecorder with 30 x 32 pixels and a laserlocalizer that measures the distance between the car and the environment
- **multilayer perceptron** with one hidden layer

- **steering direction of the car:** middle neuron highest = straight forward; most right neuron highest = maximal turn right and analogously for left
- **learning phase:** recording 1200 combinations of scenes, light and distortions with human driver as a teacher to the neural network

- **quality of driving** up to 90 km/h comparable to the best navigation systems
- **major advantage of neural networks:** fast development time.

[Pomerleau, 1991]

Google's Self-Driving Car

youtube.com

Self-Driving Car Test

Neural Network driving RC car

Speed Sign Recognition by Convolutional Neural Networks

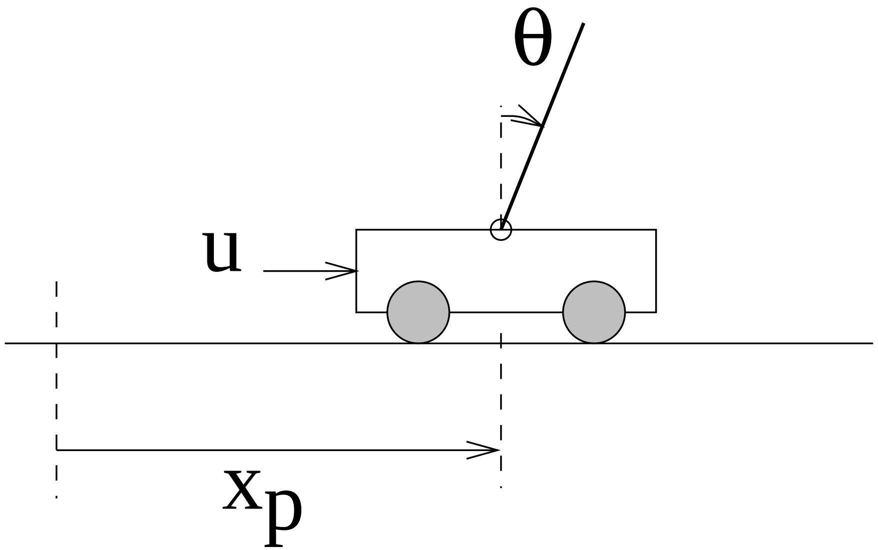
Inverted Pendulum with Neuro Fuzzy Controller

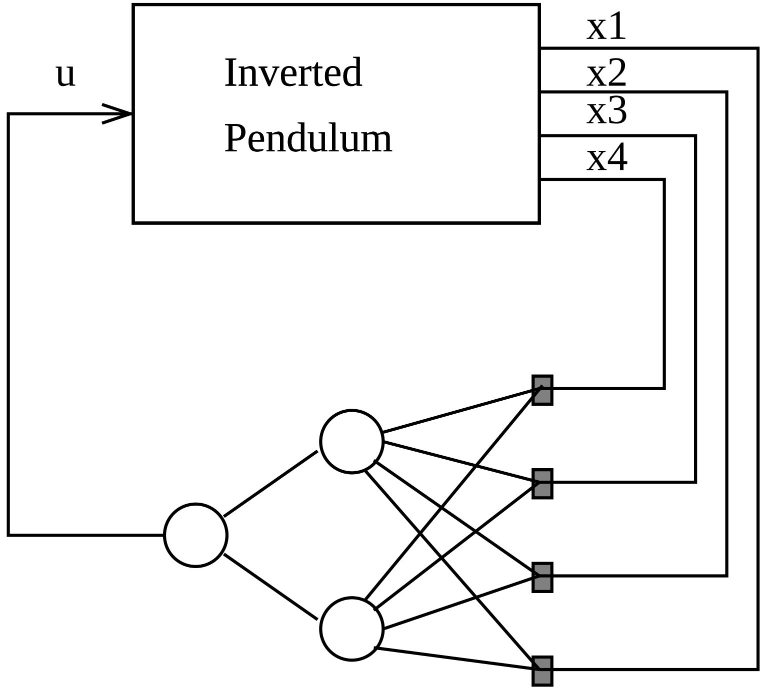
Learning to Swing-Up and Balance from Scratch

Neural network training application, fruit recognition

Cellular Neural Networks And The Matrix

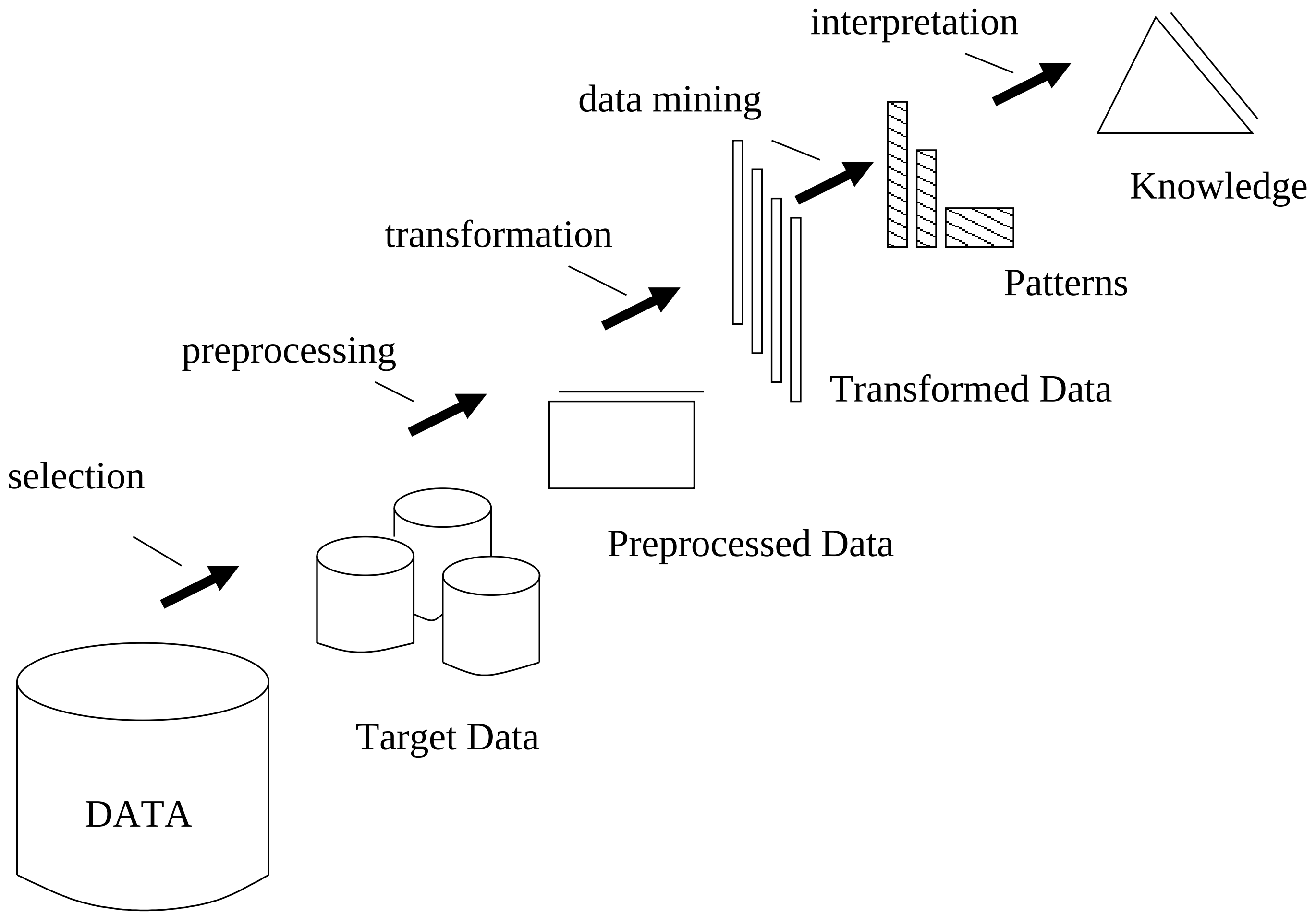
Neural control



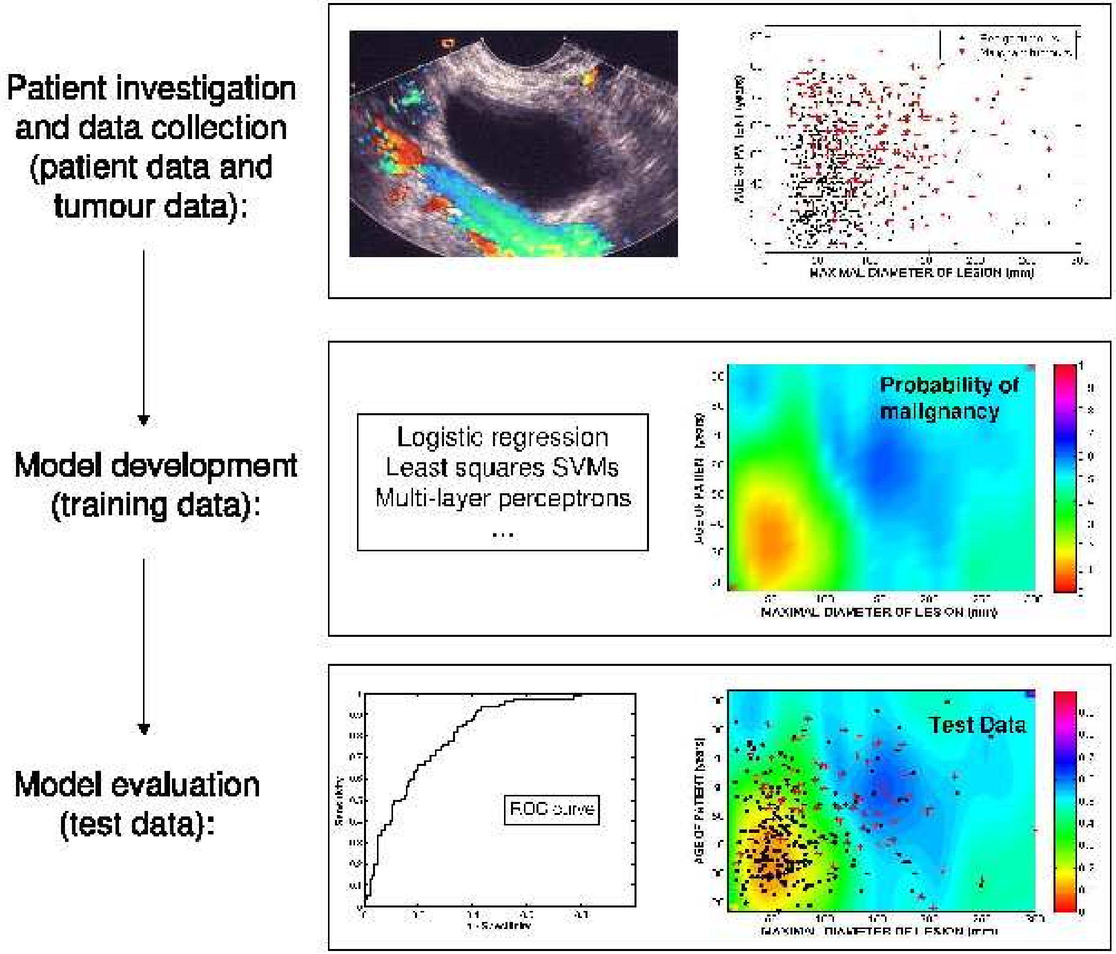


Swinging up an inverted pendulum using a multilayer perceptron [Suykens et al., 1994]

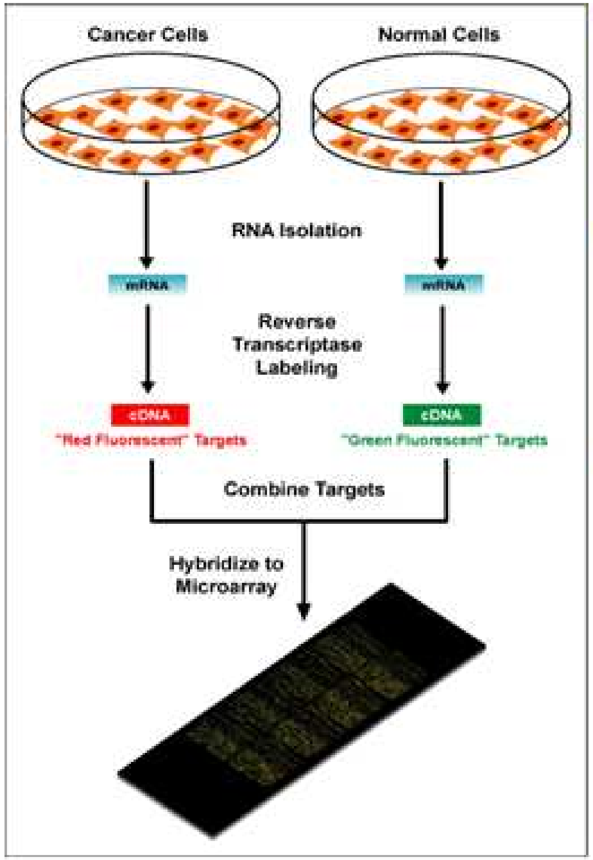
Datamining applications

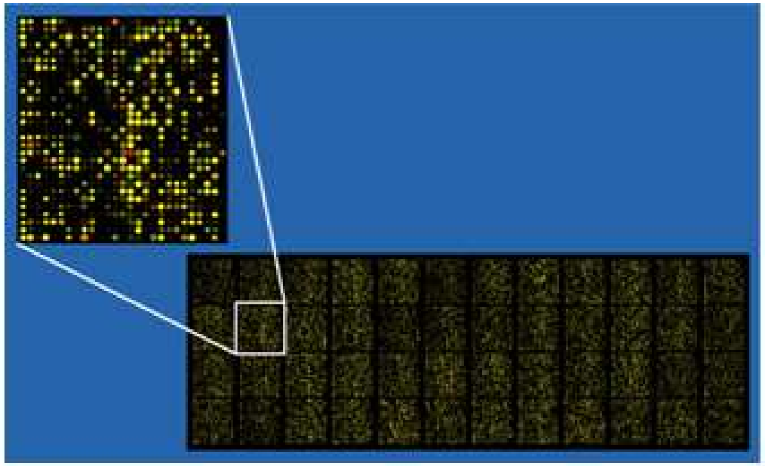


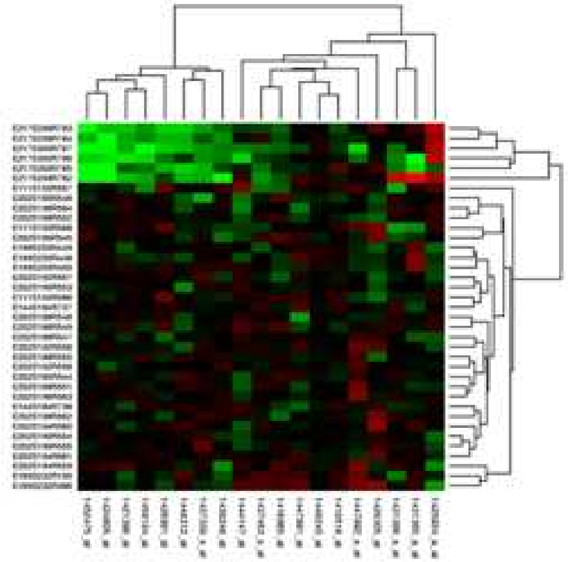
Ovarian cancer data analysis



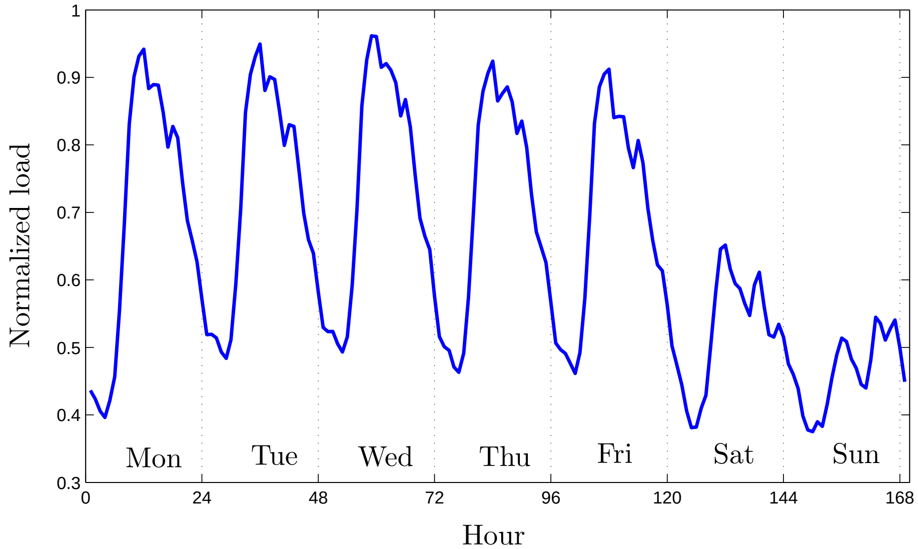
Microarray data analysis

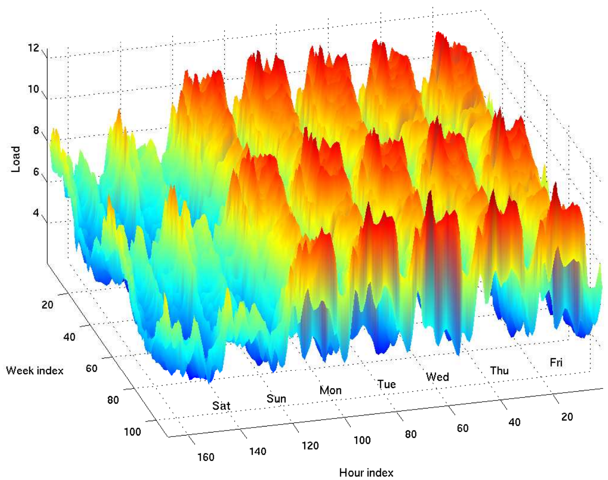






Power load forecasting





Examples of successful neural networks applications (1)

- Prediction of Yarn Properties in Chemical Process Technology
- Current Prediction for Shipping Guidance
- Recognition of Exploitable Oil and Gas Wells
- Modelling Market Dynamics in Food-, Durables- and Financial Markets
- Prediction of Newspaper Sales Production Planning
- Qualification of Shock-Tuning for Automobiles
- Diagnosis of Spot Welds
- Automatic Handwriting Recognition
- Automatic Sorting of Pot Plants
- Fraud detection in credit card transactions
- Drinking Water Supply Management
- On-line Quality Modelling in Polymer Production
- Neural OCR Processing of Employment Demands
- Neural OCR Personnel Information Processing
- Neural OCR Processing of Sales Orders
- Neural OCR Processing of Social Security Forms

Examples of successful neural networks applications (2)

- Predicting Sales of Articles in Supermarket
- Automatic Quality Control System for Tile-making Works
- Quality Assurance by "listening"
- Optimizing Facilities for Polymerization
- Quality Assurance and Increased Efficiency in Medical Projects
- Classification of Defects in Pipelines
- Computer Assisted Prediction of Lymphnode-Metastasis in Gastric Cancer
- Facilities for Material-Specific Sorting and Selection
- Optimized Dryer-Regulation
- Evaluating the Reaction State of Penicillin-Fermenters
- Substitution of Analysers in Distillation Columns
- Short-Term Load Forecast for Power Utility
- Monitoring of Water Dam
- Access Control Using Automated Face Recognition
- Control of Tempering Furnaces
- Helicopter Flight Data Analysis

Successful neural networks applications at KU Leuven

- modelling and prediction of gas and electricity consumption in Belgium
- diagnosis of corrosion and support of metal selection
- modelling and control of chemical processes
- modelling and control of fermentation processes
- temperature compensation of machines
- control of robots
- control of chaotic systems
- Dutch speech recognition
- design of analog neural chips for image processing
- diagnosis of ovarian cancer
- fraud detection/customer profiling