united nations
educational, scientific
and cultural
organization

international atomic
energy agency

the
**abdus salam**
international centre for theoretical physics

SMR: 1170/6

# *WORKSHOP ON "MODELLING REAL SYSTEMS: A HANDS-ON FIRST ENCOUNTER WITH INDUSTRIAL MATHEMATICS*

( 27 September - 22 October 1999)

---

# *I. Heart Risk Analysis*
# *II. Neural Networks*

*. Feedforward Networks*
*. Recurrent Networks*
*. RBF - Networks*

presented by:

**Dieter PRATZEL-WOLTERS**
Universität Kaiserslautern
Fachbereich Mathematik
Erwin-Schrödinger-Straße
D-67663 Kaiserslautern
Germany

---

# I. Heart Risk Analysis

# II. Neural Networks

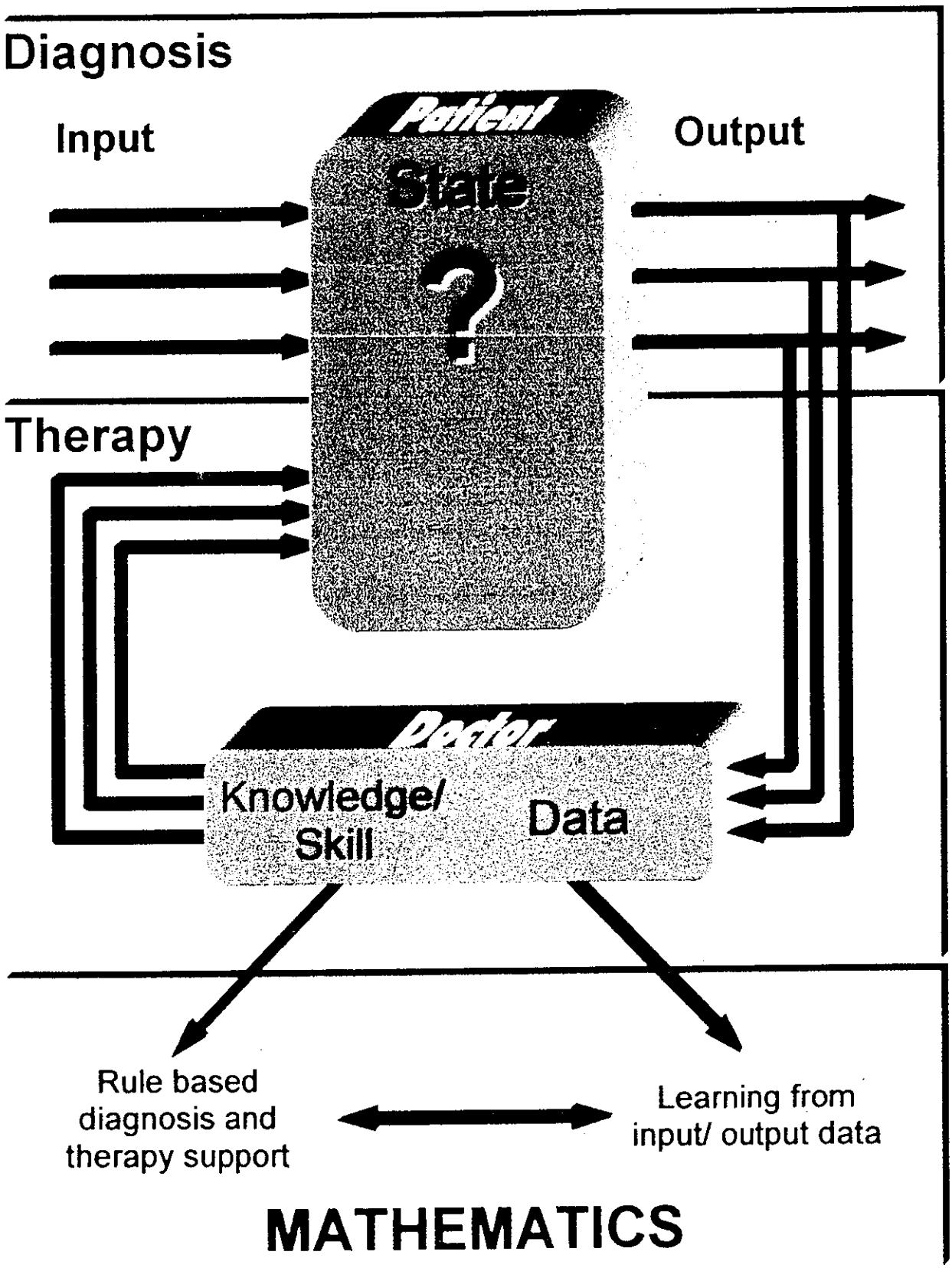
- Feedforward Networks
- Recurrent Networks
- RBF - Networks

D. Prätzel-Wolters
ITWM - KL

"An abstract view
on a date
with your doctor"

"Mixed Feelings"

# Systemtheoretic model for diagnosis and therapy



# Sudden cardiological death

## Computer-Aided Analysis of Longterm ECG's

Cooperation with:

alphaCardio

medizinisch / pharmazeutische Risikoanalysen

Foundation Rhine-land-Palatinate for Innovation

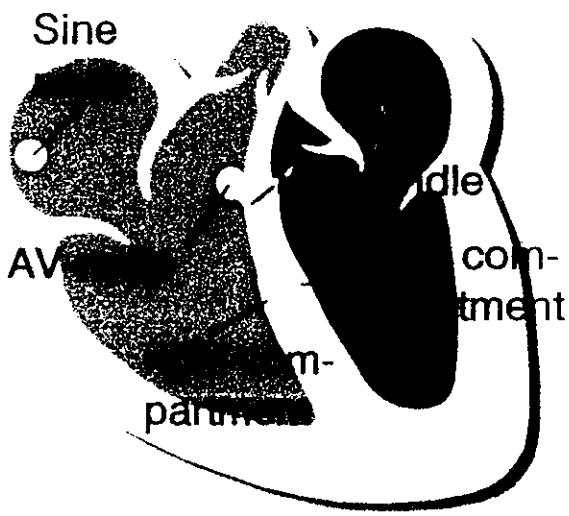
**12 % of all fatalities by sudden cardiological death**

## Aims

- Determination and verification of a reliable risk parameter based on longterm ECG's
- Development of parameters, which allow the evaluation of medicaments for cardiac rhythm control
- Therapie check up

INSTITUT FÜR TECHNO-UND WIRTSCHAFTS

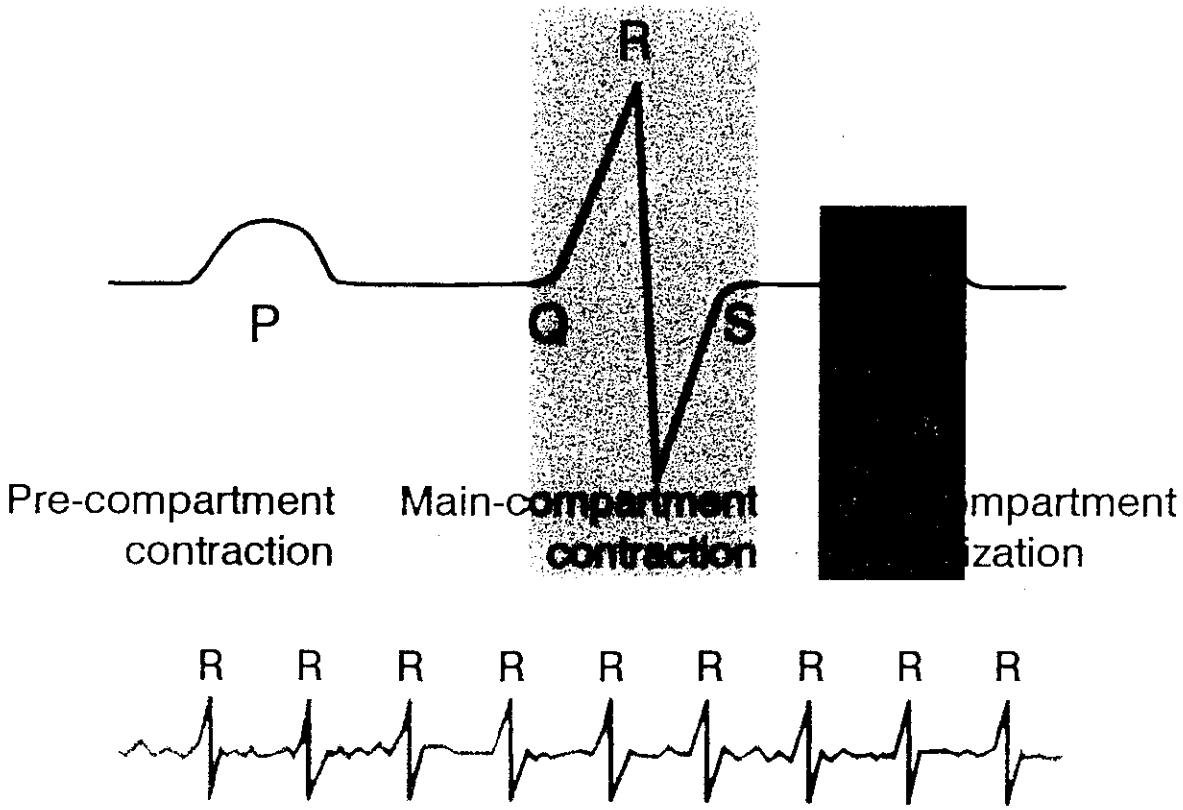
# Wiring diagram of the heart




## Heart beat

Muskelkontraktion, gesteuert durch elektrische Potential-veränderung

# PQRST-complex as basic part of ECG




INSTITUT FÜR TECHNO-UND WIRTSCHAFTS

# Complex control mechanisms

## Supraventricular rhythms

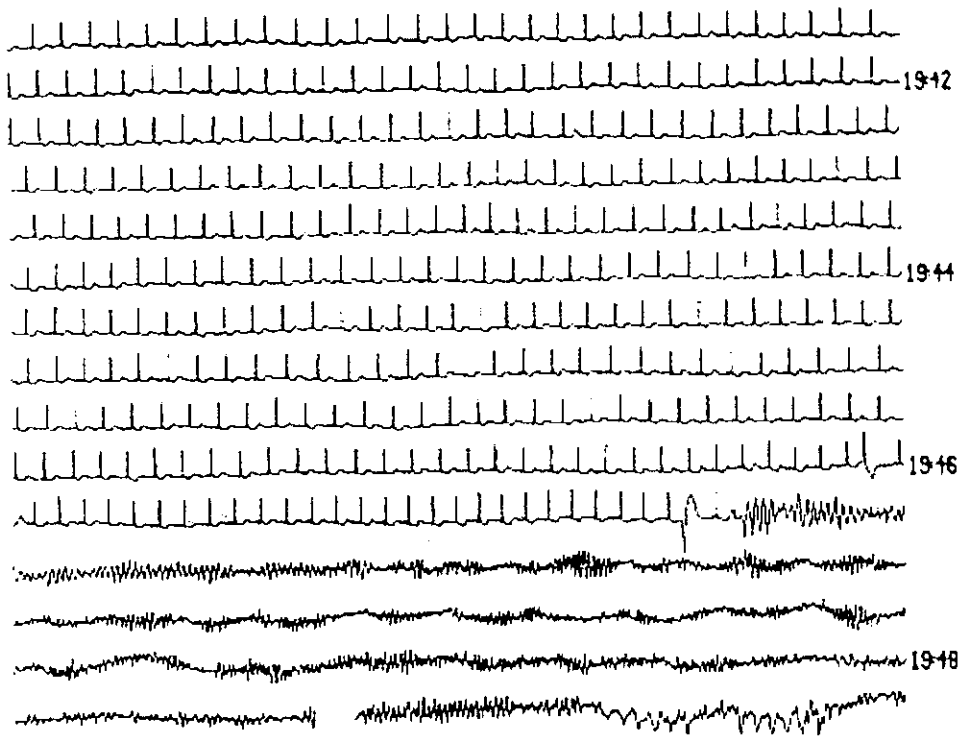
Domin

- Sine rhythm (normal)
- Pre-compartment rhythm
- AV-junctional rhythm

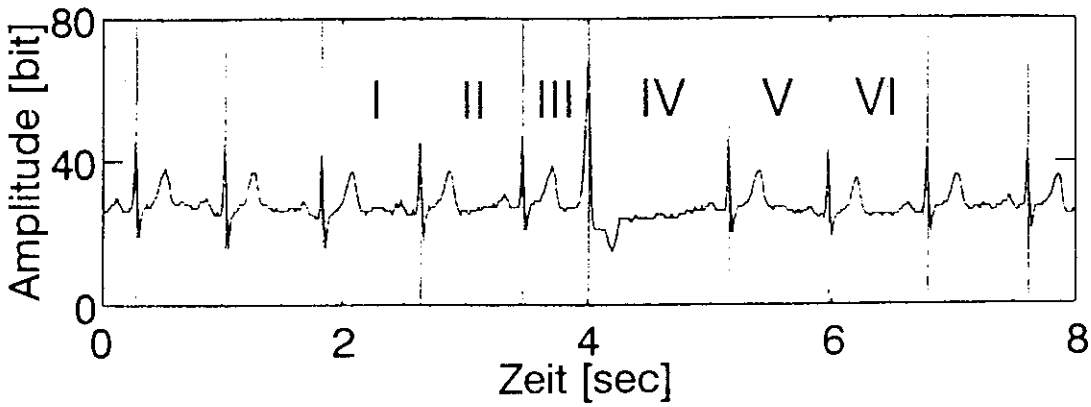
## Ventricular rhythms (broadened QRS-Complex)

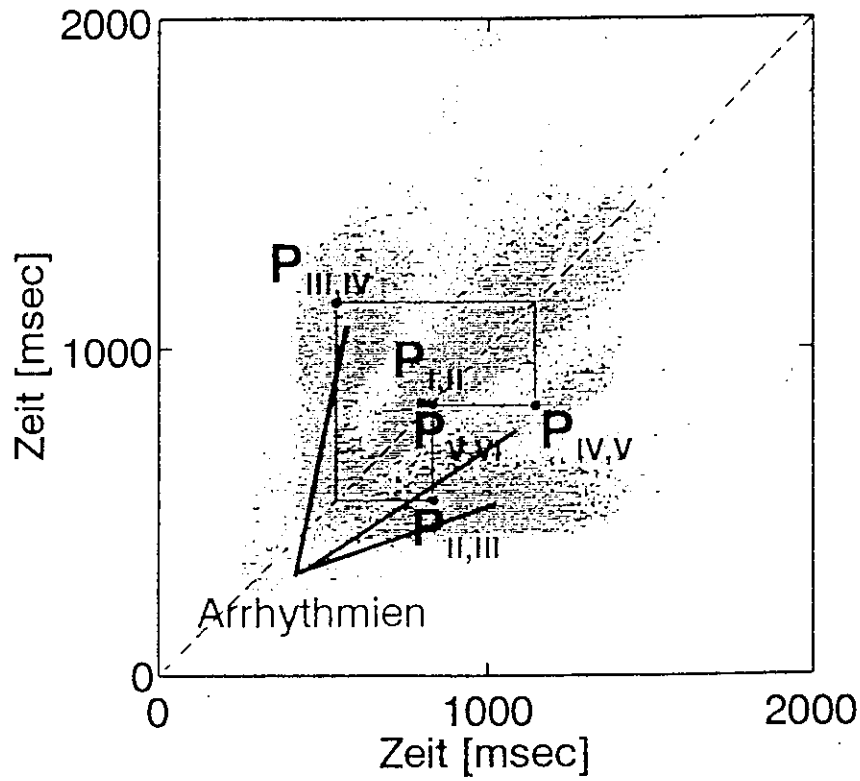
## Long-term ECG

(appr. 100.000 beats)

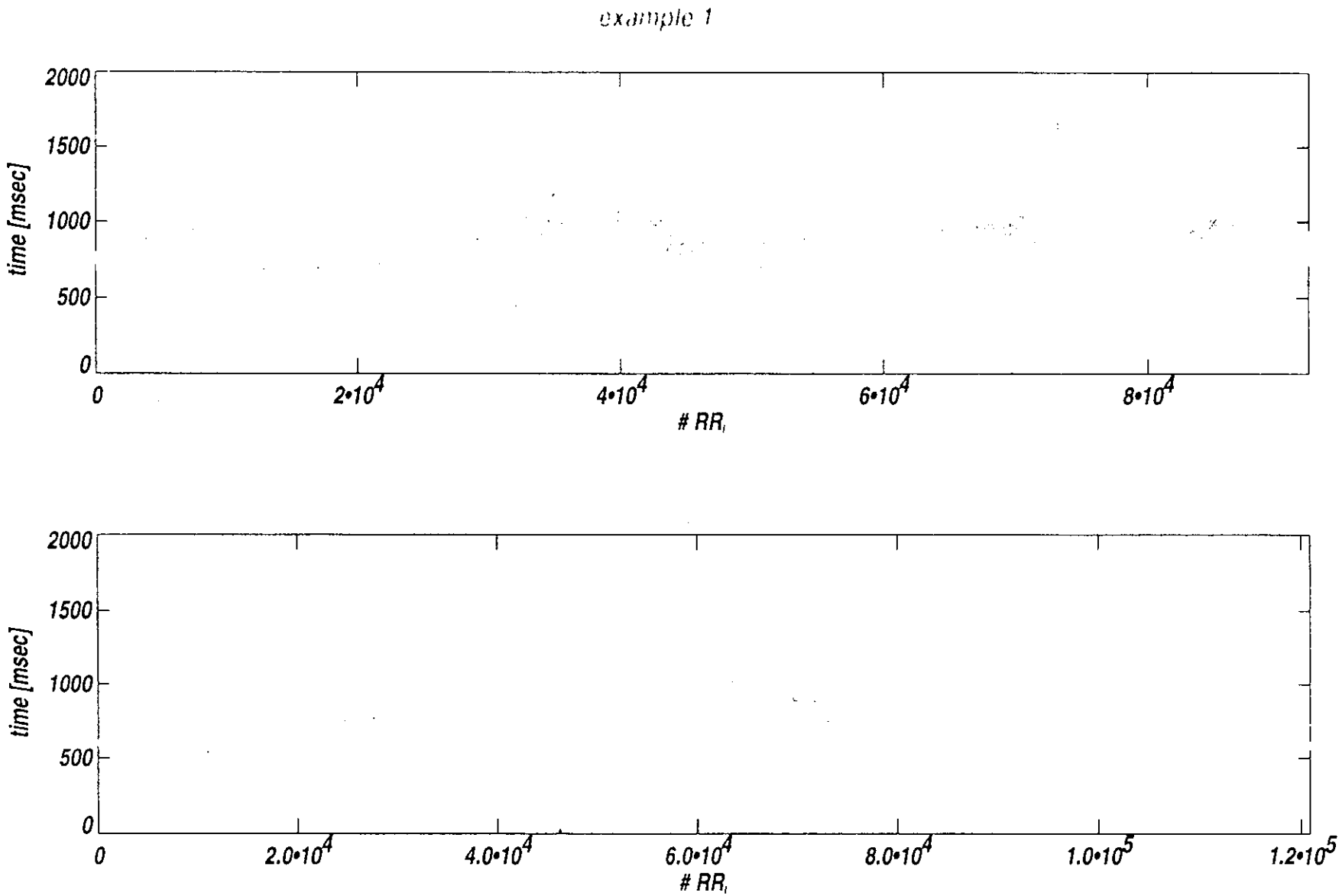


# Die Zustandsraumdarstellung der RR-Intervalle

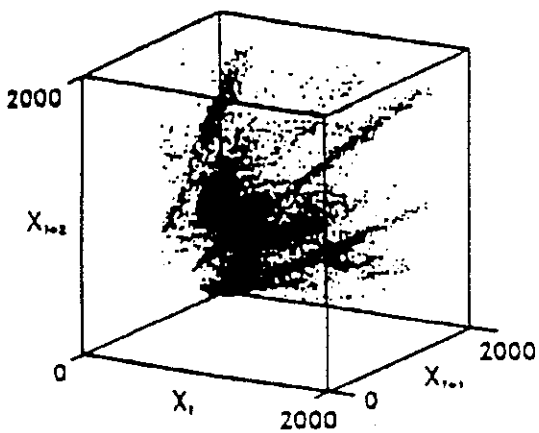


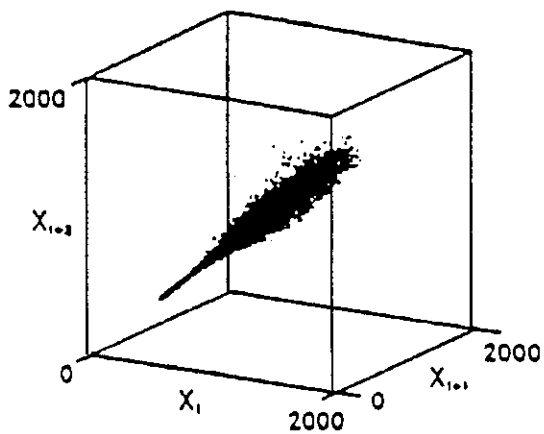
DATA: 400 ECG's, classified - 0 = dead, -1 = alive



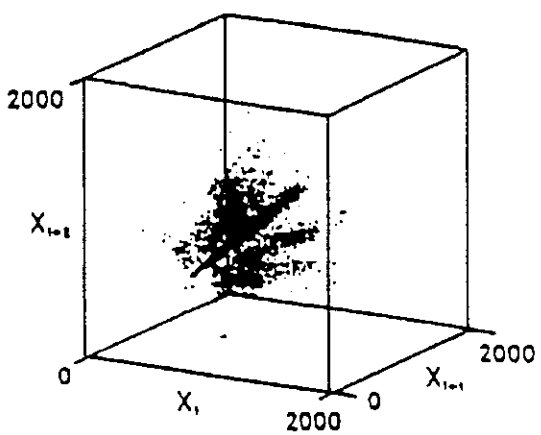
# ERKRANKUNGSGRAD




Long time survivor
(8 years, post infarct)




Healthy patient




Sudden cardiac death

# TASKS

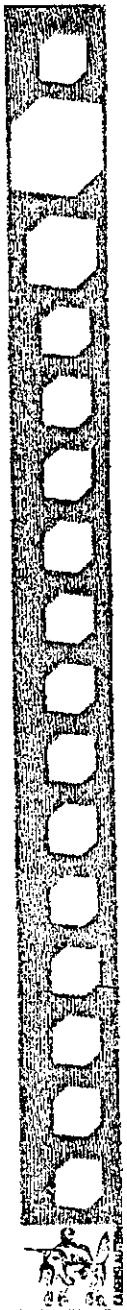
- Clustering of the 400 time series
- "Minimal" Parametrization

  - Statistical (1-d) analysis
  - Generation of higher dimensional representations
  - Data
    - cleaning
    - transformation
    - reduction
    - clustering
  - Parameter generation

# Neural Networks

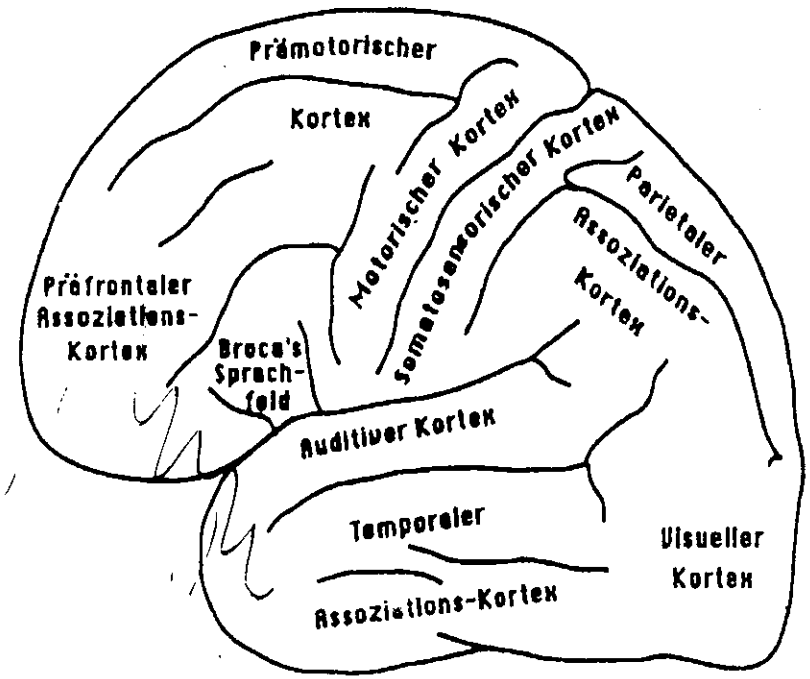
- Introduction
- Feedforward Networks
- Recurrent Networks
- Self-organizing Networks

Prof. Dr. D. Prätzel-Wolters



INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# Neurocomputer Brain



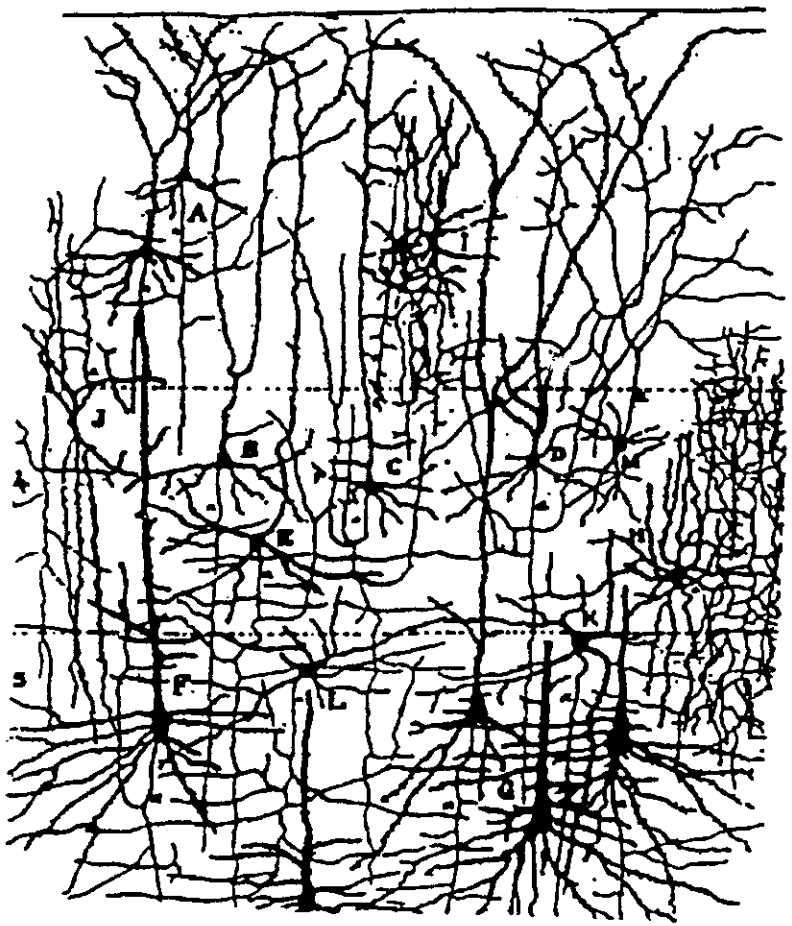

## Neocortex Human

- Cerebral cortex: place of in the intelligence
- folded layer: 0.2 qm
- Thickness of layer: 2-3 mm
- 1 Quadratmillim.: 100 000 N.
- strongly connected

INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# Sensory Fields

# Neurocomputer Brain

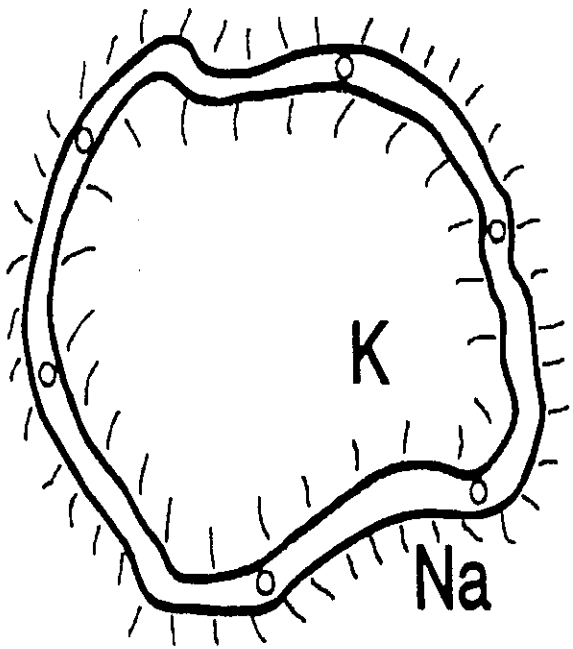


## Neocortex Cat

- Pyramide cells (A-G)
  - Portion 60 %
  - Axon to cortex surface
  - excitatory synapse
- Star cells (H-M)
  - Portion 40 %
  - Axon only in neighborhood
  - inhibitory synapse
- Density of neurons real: x factor 100

INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# Electrochemical Processes



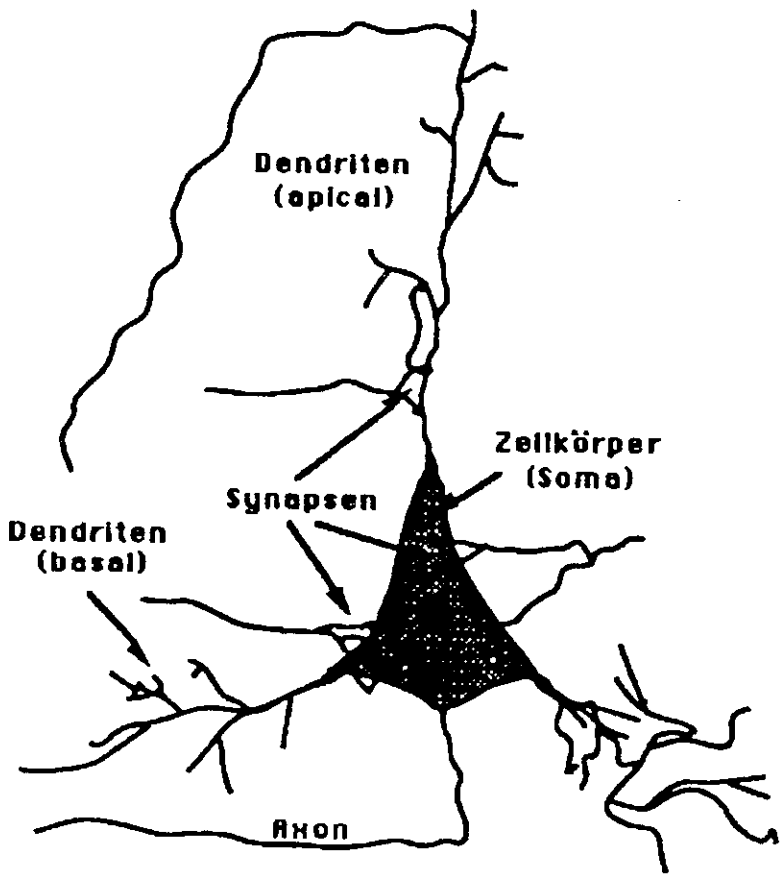

## Neuron potential

- State of rest
  - Surplus of K-ions in cell body
  - Surplus of Na-ions outside
  - rest potential -70 mV
- Kalium-balance potential -75 mV
- Natrium-balance potential +55 mV
- Controlable sluices of ions: o



INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

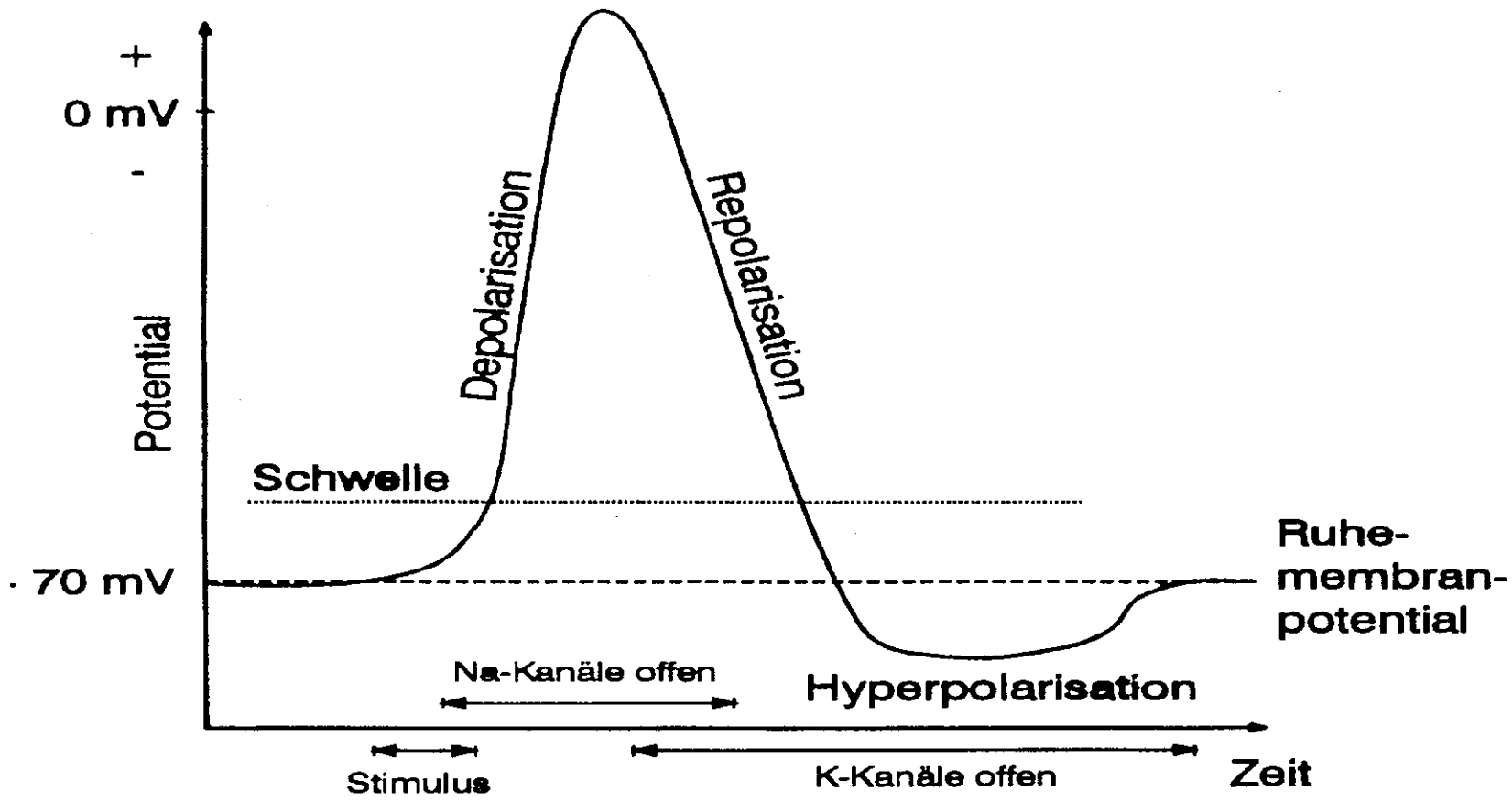
# Neuron Processing Unit




## Neuron

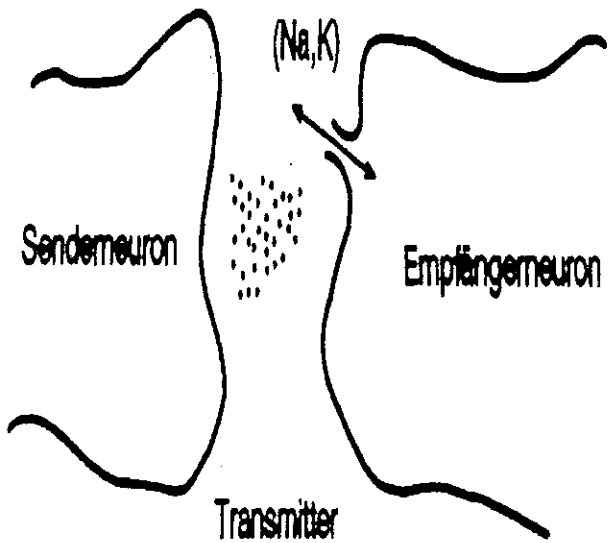
- Dendrites **(Input)**
  - Summation of the signals of the environment
  - transmission of the electrical potential
- Cell body **(Processing)**
  - Diameter: 5-100 Mycrometer
  - Spike: exceed threshold
- Axon **(Output)**
  - Lenght until 1 Meter
  - Conductor for Spike

# Spiking




Aufsummierung synaptischer Anregung

INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK




## Synapses

- excitatory postsynaptic potential (EPSP), positive stress moving
- inhibitory postsynaptic potential (IPSP), negative stress moving
- Transmitter substance depends not only on power of stimulus
- Frequent use: Sensitivity or adaption(Training!)



FhM
INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

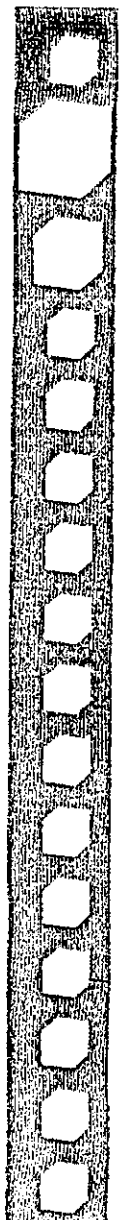
# Summary: Neural Networks

- Neural Network: Neurons have an activation state and build a network.
- Neurons receive external signals (Inputs) (e.g. sense organs)

and send signals out of the network (Outputs) (e.g. to a muscle).

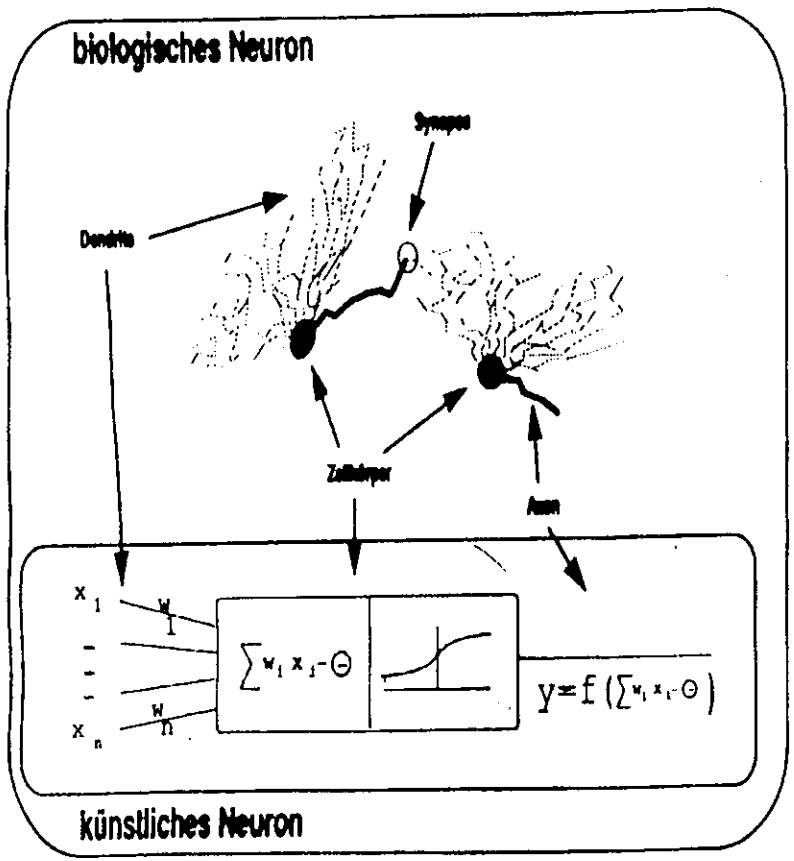
- The activation state of the neurons and the structure and the intensity of the network change with time.
- Each neuron can do only a simple job, but all neurons together can do complex processes.
- The essential data processing is in the connections and not in a neuron.





INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

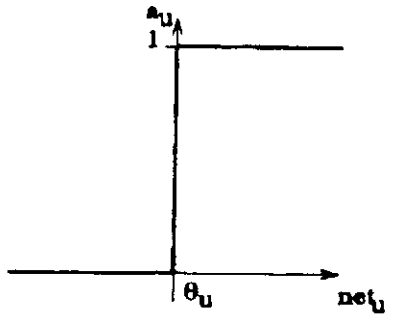
# Mathematical Model of a Neuron




## Simplifications

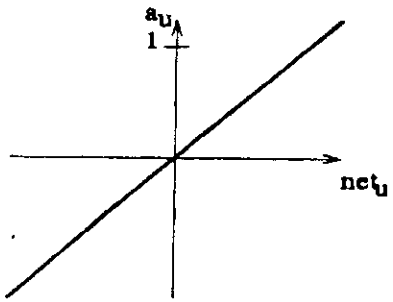
- Cell body, axon and dendrites are represented through a real number x, the activation state
- Synapses between two neurons are described by a real number w
- For the Heavyside-function as activition function f(), we obtain the Mc Culloch-Pitts-neuron (1943)

# Examples of Transfer Functions



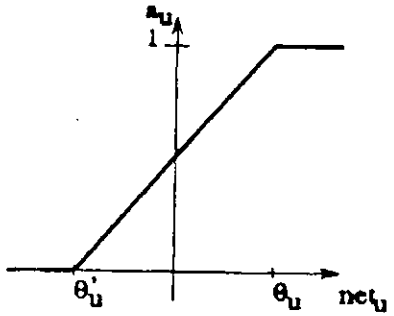

Lineare Schwellenwertfunktion

$$a_u = \begin{cases} 1 & \text{falls } net_u > \theta_u \\ 0 & \text{sonst.} \end{cases}$$



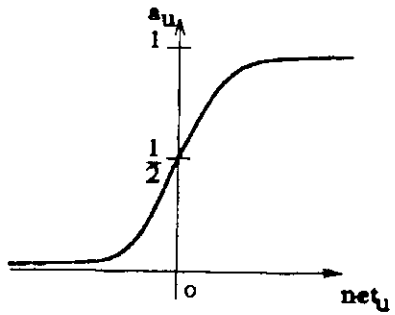

Lineare Funktion

$$a_u = c_u \cdot \text{net}_u$$

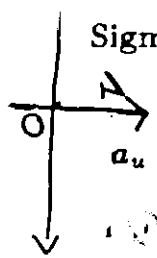



Semilineare Funktion

$$a_u = \begin{cases} 1 & \text{falls } net_u > \theta_u \\ 0 & \text{falls } net_u < \theta'_u \\ \dfrac{\text{net}_u - \theta'_u}{\theta_u - \theta'_u} & \text{sonst.} \end{cases}$$




Sigmoide Funktion



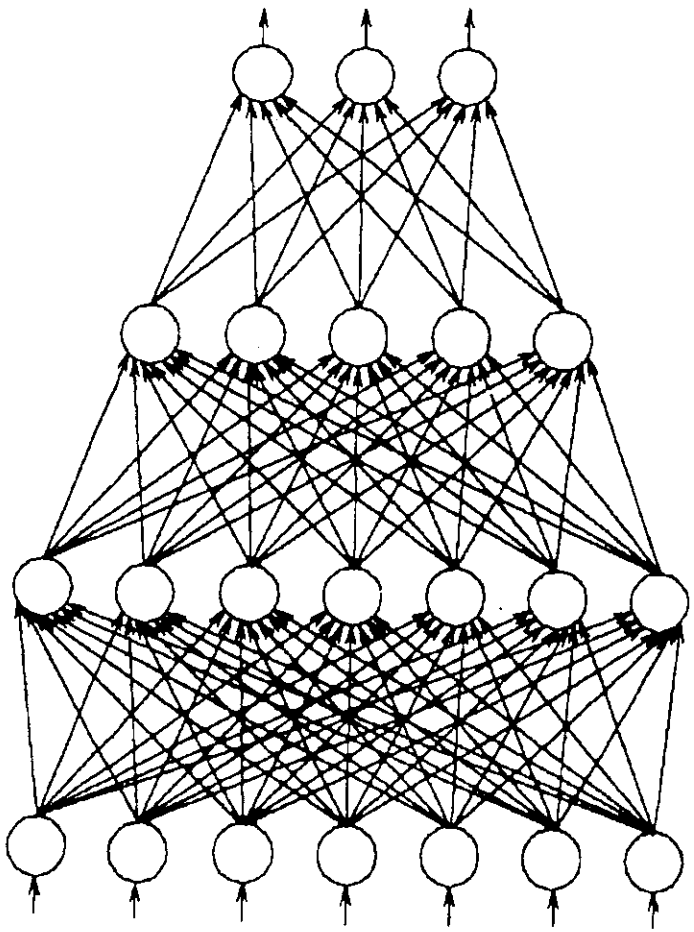

$$a_u = \frac{1}{1 + e^{-\text{net}_u}}$$



FhM

INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Feedforward Networks



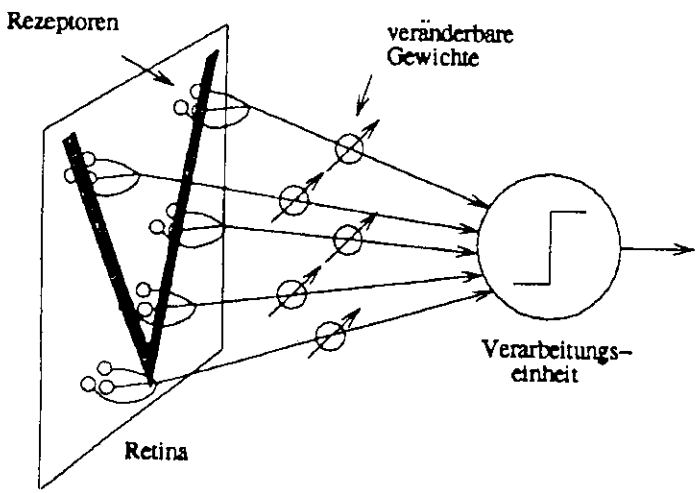
- Bottom up architecture
  - Output Layer
  - Hidden Layers
  - Input Layer
- Aimed information flow from input layer to output layer



FhM

INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# The Perceptron




- Definition (F. Rosenblatt;1958)
  - Input layer without processing
  - No hidden layer
  - 1 output unit with linear threshold function
- Application in pattern classification problems
- Solve linear separable classification problems
- No universal machine (Minsky and Papert 1969)

# A Perceptron Learning Algorithm

## Adaption step

- Cluster pair ( x, t)
- Learn rate $\sigma$
- Weight vector w
- Threshold value $\theta$

$$\Delta \mathbf{w} = \begin{cases} \mathbf{0} & \text{falls } a = t \\ +\sigma \cdot \mathbf{x} & \text{falls } a = 0 \text{ und } t = 1 \\ -\sigma \cdot \mathbf{x} & \text{falls } a = 1 \text{ und } t = 0 \end{cases}$$

$$\Delta \theta = \begin{cases} 0 & \text{falls } a = t \\ -\sigma & \text{falls } a = 0 \text{ und } t = 1 \\ +\sigma & \text{falls } a = 1 \text{ und } t = 0 \end{cases}$$

## Learning Algorithm

- Initialize weight vector and threshold value arbitrary and choose learn rate $\sigma > 0$
- Propagate all given pattern (x ,t) through the perceptron until the learning problem is solved

## Convergence of the learning algorithm

- Is the learning problem linearly separable, so you will find a weigth vector and a threshold value in a finite number of steps.



FhM
INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

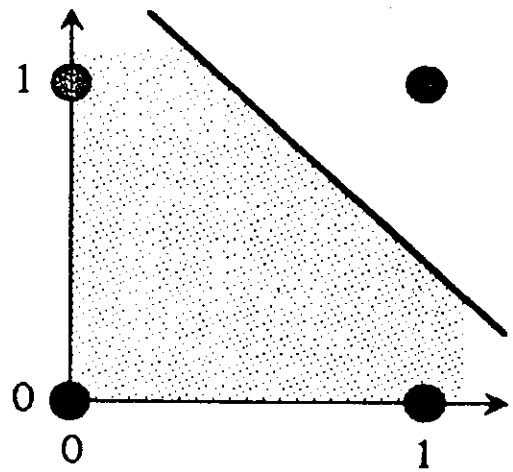
# Linear Separability

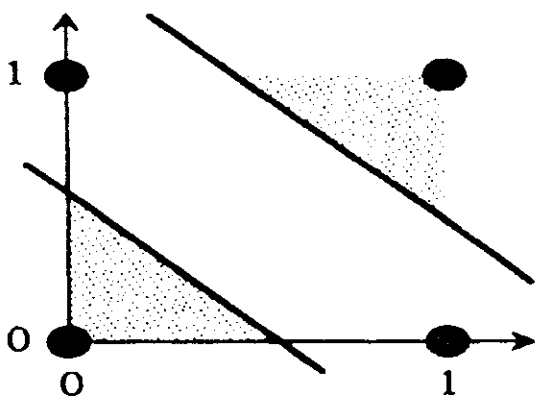
**Definition:** 2 sets X and Y are separable, if there is a values $\Theta \in \Re$ and n real values $w_1, \cdots, w_n$ so that for all points $(x_1, \cdots, x_n) \in X$ :

$$\sum_{i=1}^{n} w_i \, x_i \rangle \, \Theta$$

and for all points $(y_1, \cdots, y_n) \in Y$ :

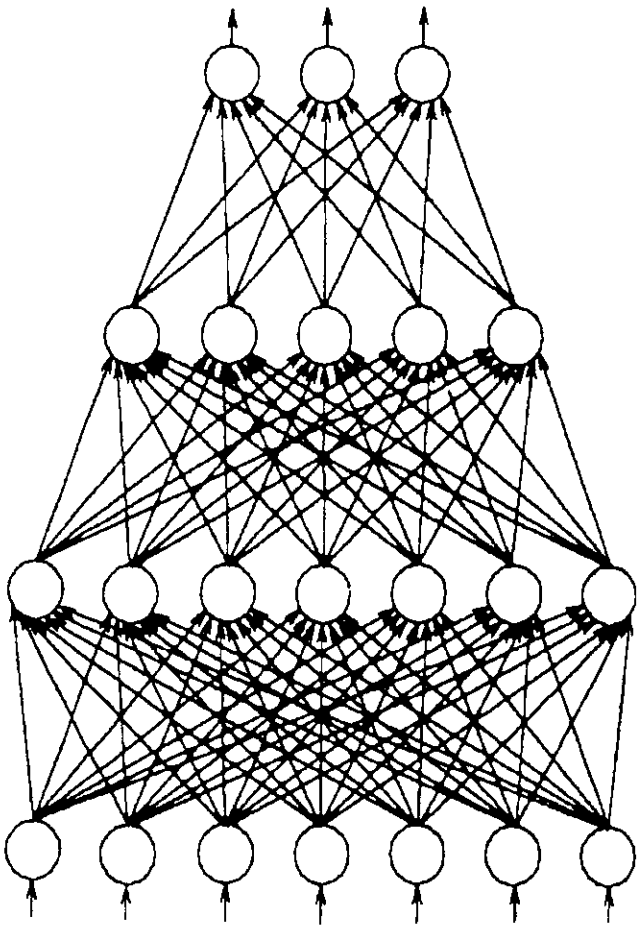
$$\sum_{i=1}^{n} w_i \, y_i \langle \, \Theta$$



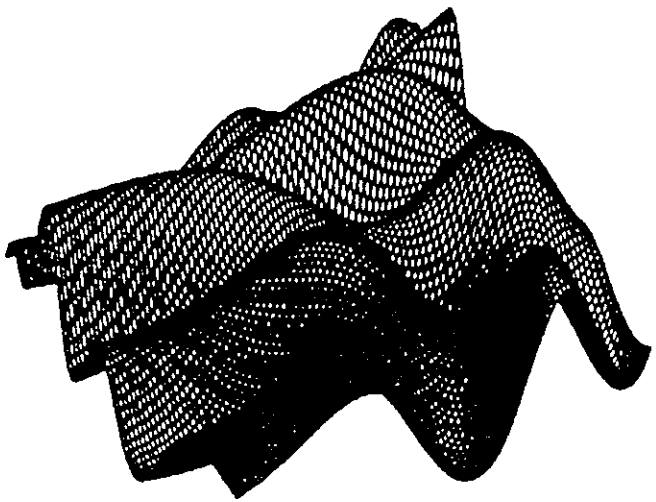

- Weight vector and threshold value define a hyperplane
- Logical functions AND and OR are linearly seperable
- XOR is **not** linearly seperable

# Multilayer Perceptron (MLP)



- Structure of an MLP
  - input layer without processing
  - One or more layers with nonlinear transfer functions
  - One output layer
- Solves non-seperable classification problems too
- Until 1986 no learnung algorithm, then backpropagation algorithm (Rummelhart et al.)
- MLPs are general approximation tools (Hornik et al.)

# The Backpropagation Algorithm

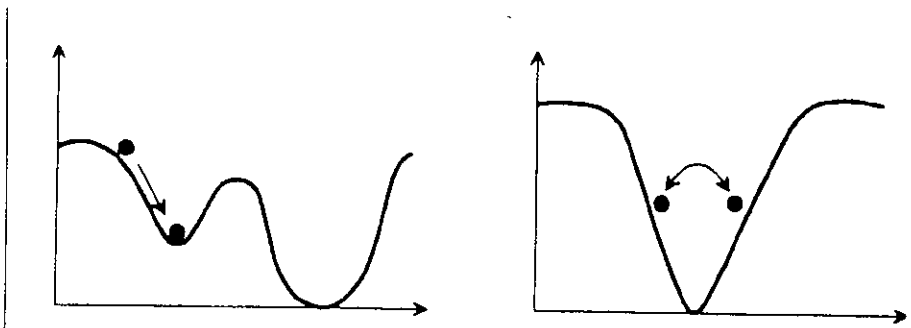


■The backpropagation algorithm gives an approximation of the gradient of the energy function:

$$E(w) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} (y_i^j - \mathrm{O}^j(x_i))^2$$

■Random initialization of the weights, **no symmetry !**

■Problem: **Local minima** of the energy function



■Modifications of the backpropagation-A.:

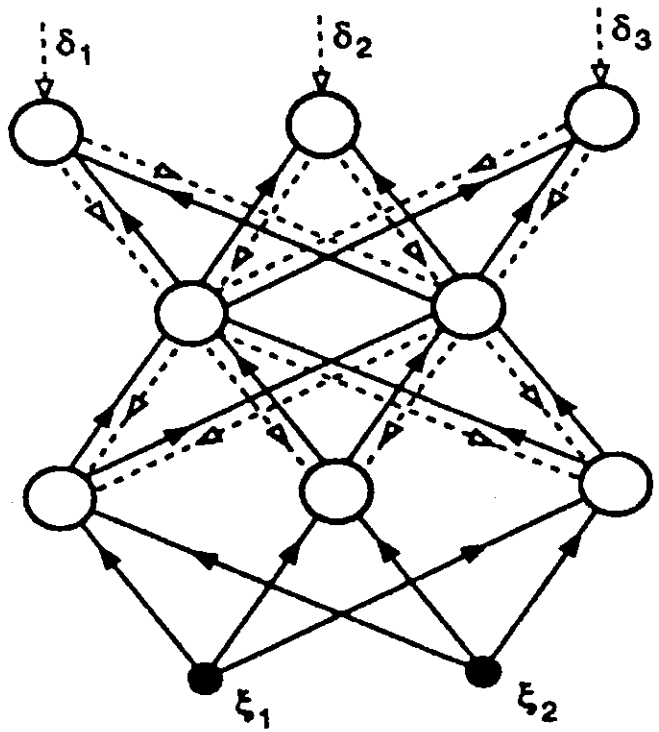
- ■Momentum terms (inertia)
- ■Modification of the energy functions
- ■Updates of the weights after one complete training epoch



FhM
INSTITUT FÜR TECHNO- UND WIRTSCHAFTSMATHEMATIK

# The Backpropagation Algorithm

- The errors are backpropagated from output- to input-layer and weights updated per layer
- the backpropagation algorithm is a generalization of the delta rule for multi layers networks
- Assumption: Transfer functions must be differentiable




$$f(x) = \frac{1}{1+e^{-\beta x}}, \quad \beta > 0$$

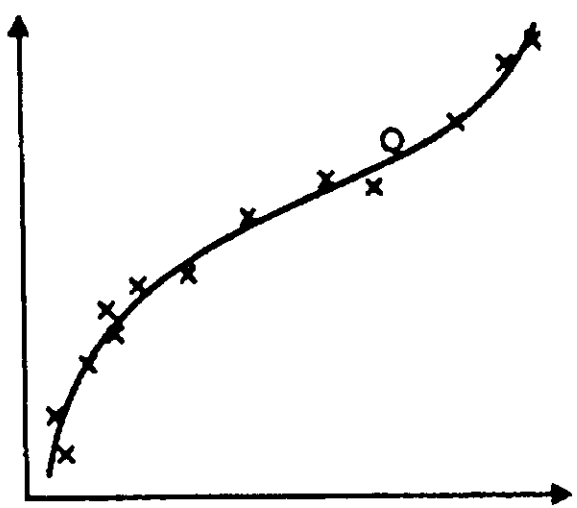
$$f(x) = \tanh(x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$$

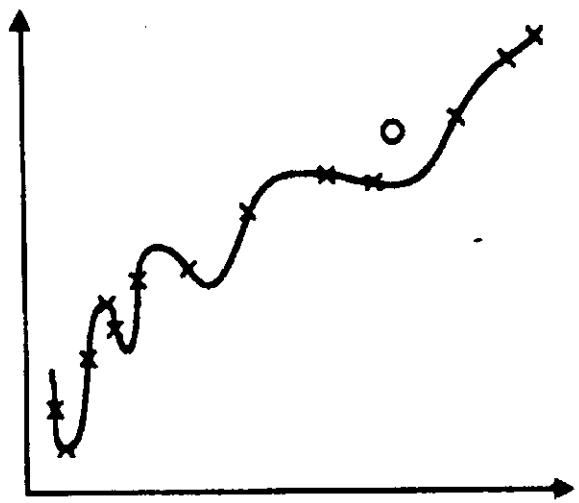


INSTITUT FÜR TECHNO- UND WIRTSCHAFTSMATHEMATIK

# Problem: Determination of the dimension of the network

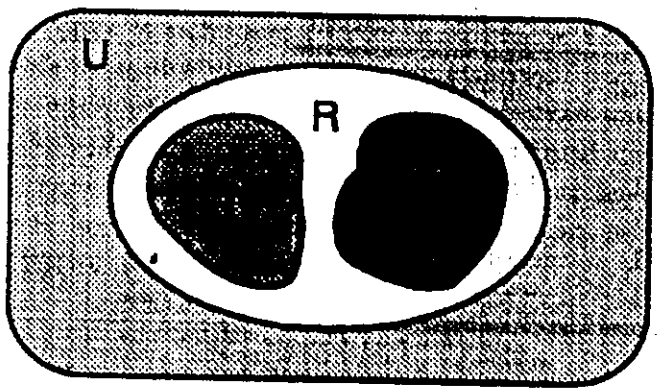
- Theoretical one hidden layer is enough (Hornik et al.)
- Overfitting (interpolation) if the network dimension is too large, bad generalization
- Ansatz: Networkgeneration

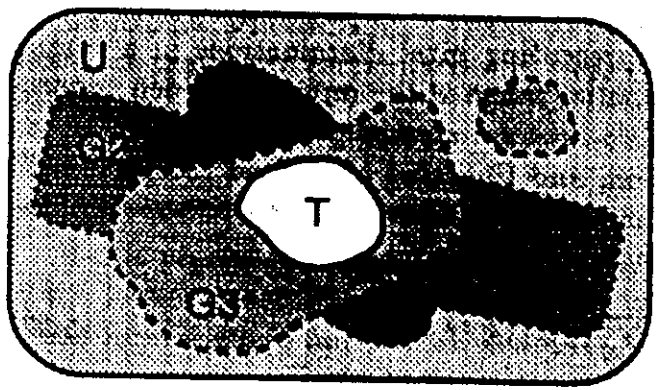




# Generalization Problem

- Training- and test set distribution
  - T: training set
  - X: test set
- Select a representive distribution
- Performance of T measures the training success, performance on X measures the generalization capability
- Stopping criteria for learning algorithm should be the rise of the generalization error, training error is mostly monotonically decreasing






INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# Recurrent networks

Distinctions

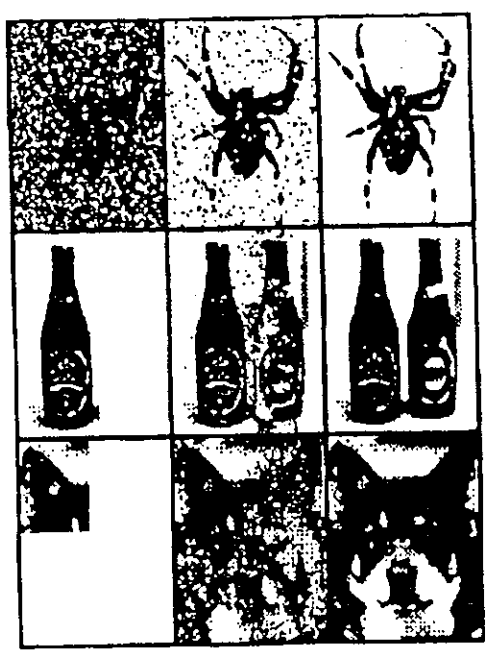
- discrete and continuous time
- Value of a neuron: binary, continous
- Structure of the network
- Neurons: deterministic or stochastic

$$x(t+1) = \sigma(Wx(t)+v)+u(t)$$

$$\frac{d}{dt}x(t) = -x(t)+\sigma(Wx(t)+v)+u(t)$$

$$\frac{d}{dt}x(t) = -x(t)+W\sigma(x(t))+v+u(t)$$

- Autoassociative memory: image, pattern and character recognition
- Learning of time dependent signals
- combinatorical optimization problems

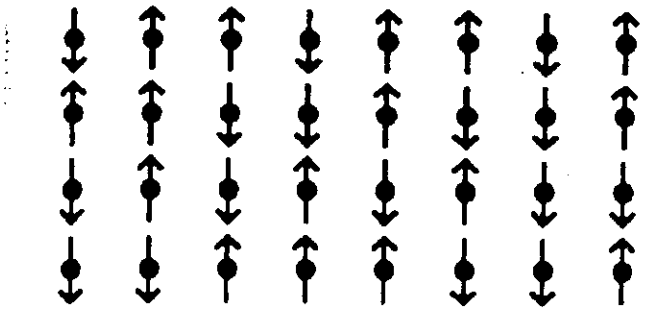




INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Hopfield Networks

- Physical motivation: Ising model for spins



- Autonomous discret time system

$$x(t+1) = \operatorname{sgn}(Wx(t)+\eta) \quad x(t) \in \{+1,-1\}^n, W \in R^{n,n}, \eta \in R^n$$

- Inputs about initial state x(0)
- Solutions: sequences on hypercube
- Goal: Coding of $\xi_1, \ldots, \xi_M$ as a fixpoint for the difference equation

$$\xi_k = \operatorname{sgn}(W\xi_k + \eta)$$

- Autoassociative memory problem: Coding of data, so that x(t) converges for every initial state x(0) against the nearest pattern $\xi_k$
- Hamming-distance:

$$d(x,y) = \frac{1}{2}\sum_{i=1}^{n} |x_i - y_i|$$



INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Recurrent networks

## Associative memory

Heteroassociation

- Maps x to y
- Finite sets of pairs (x, y)
- Robustness


Autoassociation

- Maps x to x
- Finite sets of vectors x
- Reconstruction of noise data


Pattern recognation

- Assign to a class



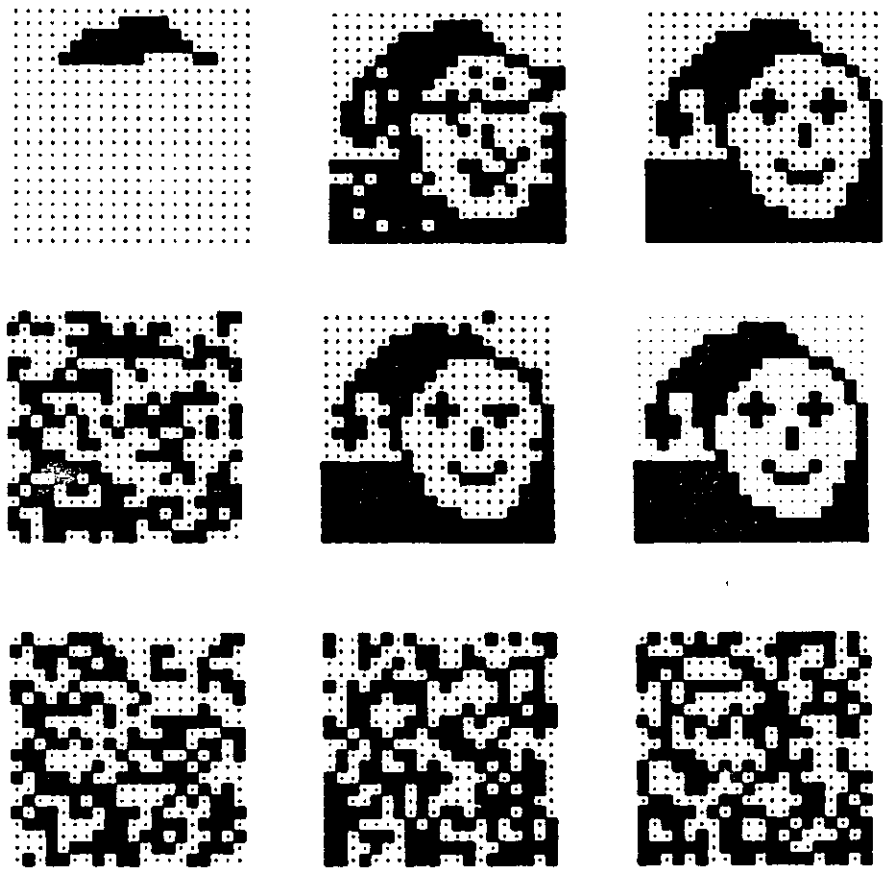

INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Hopfield Networks

## Convergence to a Fixpoint

- Black and white pictures in±1 coding
- Pattern coded in weight matrix
- Presentation of an incomplete or grainy image x(0)
- Stepwise approximation to a pattern, if similarity is given
- Konvergence to an undesired fixpoint is possible

- Desirable: big attractor region and big storage capacity
- Inconsistent goals!





INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# Recurrent Neural Networks
## Convergence

Alternative interpretation of the dynamic (short term memory)

- Until now: synchronic update, neurons work parallel $x(t+1) = \mathrm{sgn}(Wx(t)+\eta)$
- Asynchronic update: neurons work sequential

$$x_1(t+1) = \mathrm{sgn}(\sum_{j=1}^{n} w_{1j}x_j(t)+\eta_1)$$

$$x_2(t+1) = \mathrm{sgn}(w_{21}x_j(t+1)+\sum_{j=2}^{n} w_{2j}x_j(t)+\eta_j)$$

...

- Blocksynchronic update: neurons work in groups

| | asynchron | synchron | Block-synchron |
|---|---|---|---|
| $W = W^T$ | Zyklen möglich | Zyklenlänge höchstens 2 | Zyklen möglich |
| $w_{ii} \geq 0$ | Nur Fixpunkte | Zyklenlänge höchstens 2 | Zyklen möglich |
| $W \geq 0$ auf $\{-1,0,1\}^n$ | Nur Fixpunkte | Nur Fixpunkte | Nur Fixpunkte |

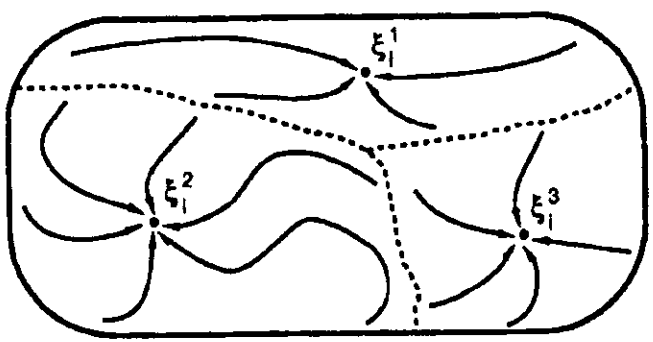
Energyfunction

$$E(x) = -\frac{1}{2}x^T Wx - x^T\eta$$

# Hopfield Networks

## Fixpoint implementation

- Fixpoint implementation: How to contruct weights?
- Freedom of cycles: Convergence for all initial value?
- Capacity: How many clusters can be stored by a given network dimension n?
- Are there unexpected fixpoints?
- How robust is the network?




Freedom of cycles, if

- W is symmetric and positive definite in $\{-1, 0, 1\}^n$

$$W = W^T, \qquad x^T W x \geq 0 \text{ for all } x \in \{-1,0,1\}^n$$

- W is symmetric and diagonal dominant (all vectors in $\{-1, 0, 1\}^n$ are fixpoints)

Symmetry is not enough; 2 cycles are posible



INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Hopfield Networks
## weighted Hebb rule

$$W = \sum_{k=1}^{M} \lambda_k \xi_k \xi_k^T, \qquad \eta = 0$$ give fixpoints $\xi_1, \ldots, \xi_M$, if

$$1 = \sum_{k=1}^{M} \lambda_k,$$

$$\lambda_k \geq \sum_{m \neq k}^{M} \lambda_m \cos(k,m) \quad \text{für alle k, wobei } \cos(k,m) = \frac{1}{n}(\xi^k)^T \xi^m$$

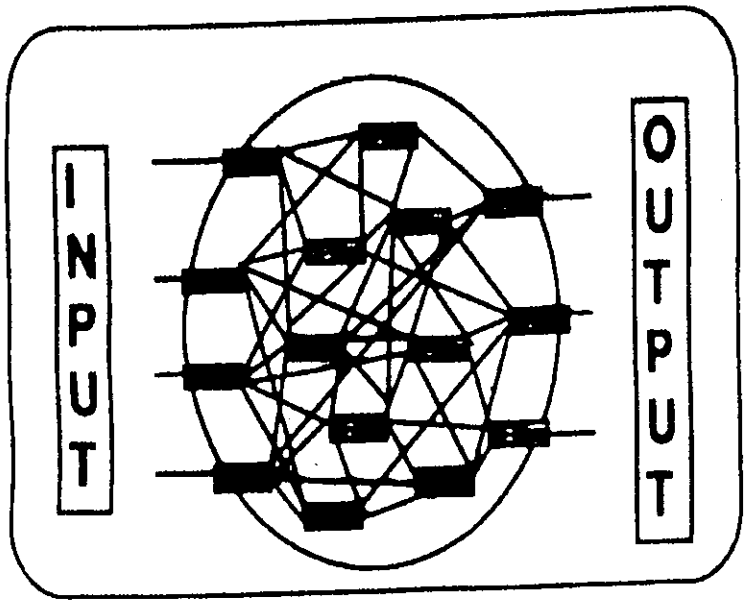
Features

- $W = W^T$, $w_{ii} \geq 0$: no fixpoint
- only n clusters can be stored
- alternative construction: Moore-Penrose inverse, same features

Radius of attractor: maximal distance, if convergence is fulfilled

- Estimation for direct radius of attractor depends on $\xi_1, \ldots, \xi_M$
- Radius of attractor (long term memory) may be difficult to determine

# Recurrent Networks




## Features

- full network structure
- Well-chosen input and output neurons
- Nonlinear transfer function
- Dynamical Network
- Supervised learning



INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# Learning Methods of Recurrent Neural Networks

## real-time recurrent learning

Dynamic and output as backpropagation through time

$$x(t+1) = \sigma(Wx(t) + u(t))$$

Error function

$$E_k(t) = (\xi_k(t) - x_k(t))^2 \qquad E(t) = \left\|\xi(t) - x(t)\right\|^2 \qquad E = \frac{1}{2}\sum_{t=0}^{T} E(t)$$

Gradient

$$\Delta w_{pq}(t) = -\eta \frac{\partial E(t)}{\partial w_{pq}} = \eta \sum_k E_k(t) \frac{\partial x_k(t)}{\partial w_{pq}}$$

$$\frac{\partial x_i(t)}{\partial w_{pq}} = \tanh'(h_i(t-1)[\delta_{ip} x(t-1) + \sum_j w_{ij} \frac{\partial x_j(t)}{\partial w_{pq}}]), \quad h_i(t) = \sum_j w_{ij} x_j(t)$$

Initial state

$$\frac{\partial x_i(0)}{\partial w_{pq}} = 0$$

Alternative: [redacted]



INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Recurrent Networks

## Learning of periodic signals

Model with sparse weighting matrix

- Reference model:

$$\frac{dx^*}{dt}(t) = -x^*(t) + W^* \tanh(x^*(t)) + f(x^*(t))$$

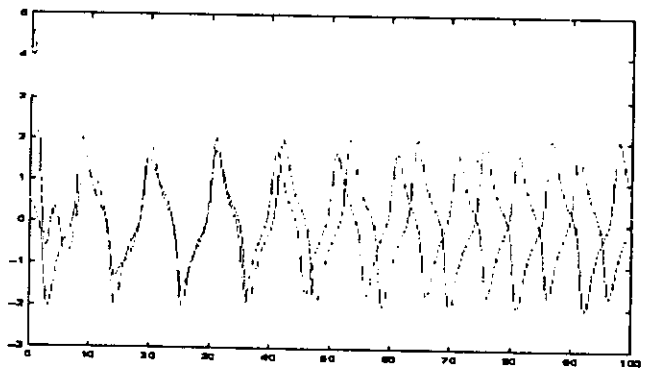
- Learning Network:

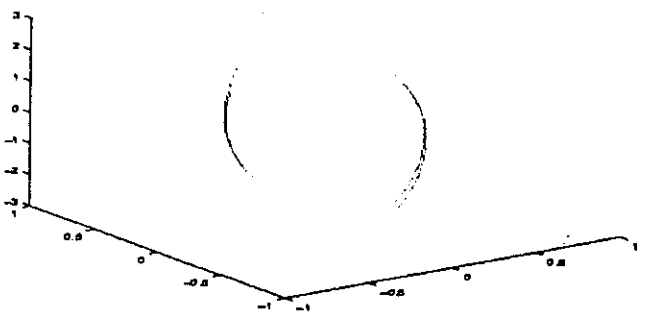
$$\frac{dx}{dt}(t) = -x(t) + W(t) \tanh(x(t)) + f(x^*(t))$$

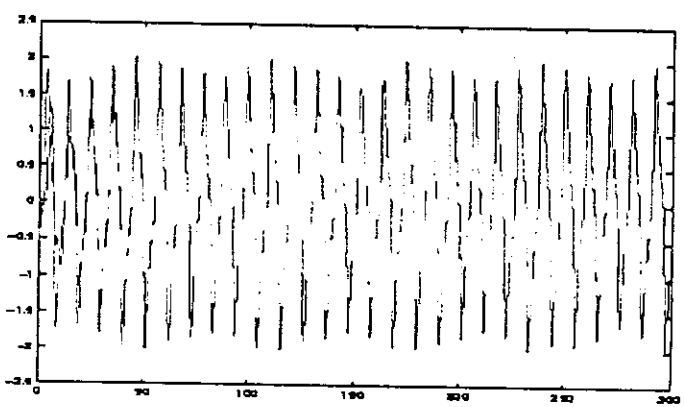
- modell reference approach:

$$\frac{dW}{dt}(t) = -(x(t) - x^*(t)) \tanh(x(t))^T$$

- For some weighting matrices: For some matrices convergence is fulfilled
- Tracking of parameters W* by slow variation









INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Elman-Networks

## Learning of time dependent signals

Net dynamic gives good approximation

$$x(t+1) = V \tanh(W_1 x(t) + W_2 p(t) + b) + a$$

State x(t), input p(t), recurrent connection to input-layer

- Series recognition: classification of signals, e.g.: speech recognition
- Series reproduction: replay of signals, see autoassociation
- Time association: associative memory problem for series

Algorithms

- Presentation of an input or of an input-series
- Backpropagation of the quadratic error, but neglect recurrent connections
- Take approximative gradient for adaption of the weights
- It does not always work

Alternative: Restriction of the system; special algorithms which converge

# Learning Methods of Recurrent Neural Networks

## backpropagation through time

- Only for discrete time

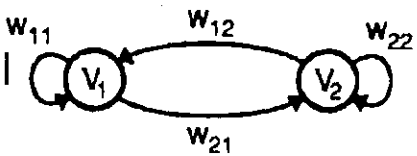
$$x(t+1) = \sigma(Wx(t) + u(t))$$

- Job: Output of a series ξ(t) as response of input ζ(t)
- Idea: Development of the network through time for short time T
- backpropagation for unfolded network

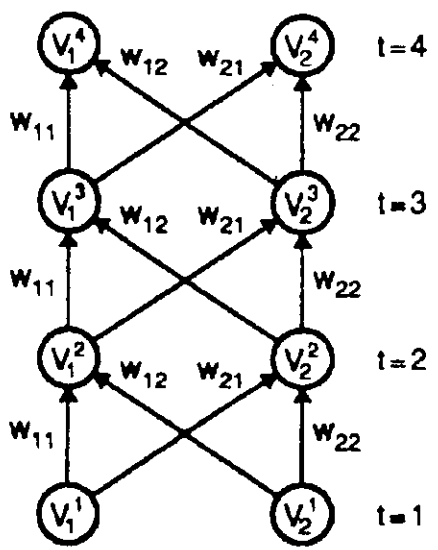
Example:

- Original network (a) with 2 neurons, full recurrent
- Unfolded network (b) for T=4 time steps
- Approximate solution only für large T
- Expensive for large T or high network dimension

(a)




(b)



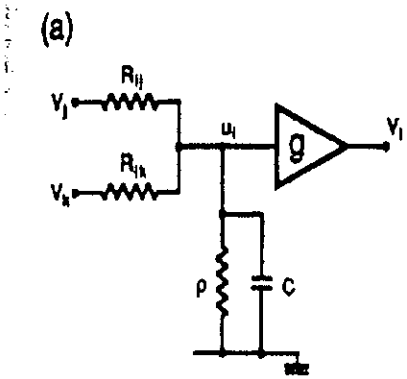
# Hopfield Networks
## continuous time

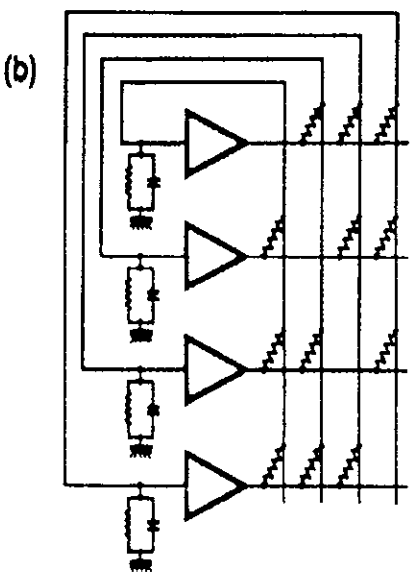
Continuous transfer function, e.g. tanh

$$\frac{d}{dt}x(t) = -\tau x(t) + \sigma(Wx(t) + v)$$

Hardware implementation

- $u_i$ Input voltage, $V_i$ output voltage
- $V_i = g(u_i)$
- Capacity C, resistance $R_{ij}$




$$C\frac{du_i}{dt} + \frac{u_i}{\rho} = \sum_j \frac{1}{R_{ij}}(V_j - u_j)$$

$$\tau_i \frac{du_i}{dt} = -u_i + \sum_j w_{ij} g(u_j)$$

where

$$\tau_i = R_i C \qquad \frac{1}{R_i} = \frac{1}{\rho} + \sum_j R_{ij} \qquad w_{ij} = \frac{R_i}{R_{ij}}$$

Convergence for symmetric W

# Optimization with Hopfield Networks

Combinatorical optimization problems

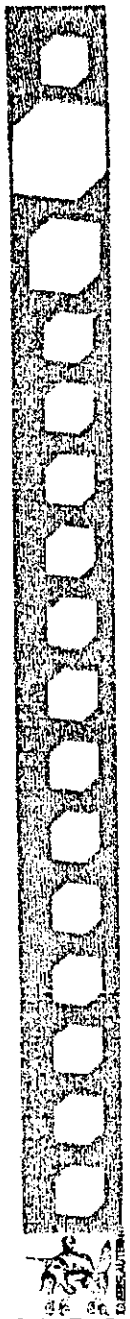
- e. g. travelling salesman problem (TSP), 8-Damen problem, sorting,...
- Solution is often a permutation matrix
- Problem often np-complete, e.g. for large size n (number of towns by TSP) no exact solution

[redacted]

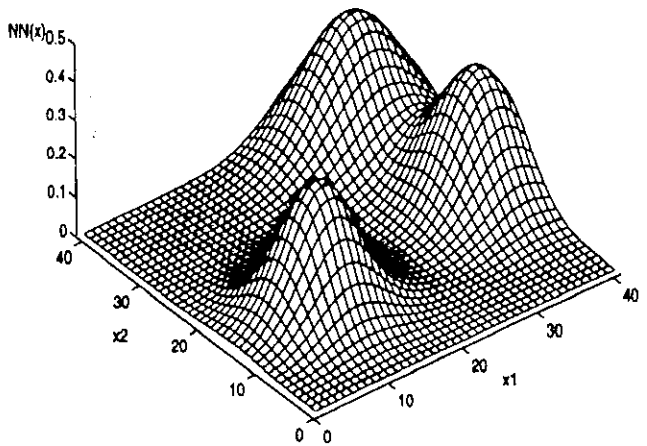
- Appropriate coding of the problem
- Construction of an energy function E, so that minima of E are minima of the cost function for the optimization problem
- Coding of the constraints in the weights
- Simulation of the network
- Decoding of a solution from a stable final state

Construction may be tricky!





# Radial Basis Networks




- Structure of the network:
  - Input layer without processing
  - 1 hidden layer with radial symmetric processing,

    Parameters: C (center) and $\sigma$ (form parameter):

$$f(x) = \exp\left(-\frac{1}{2\sigma^2}\|x - C\|^2\right), \quad \sigma > 0$$

$$f(x) = \frac{1}{\sqrt{\frac{\|x - C\|^2}{\sigma^2} + d^2}}, \quad \sigma, d > 0$$

  - Linear output layer
- RBF-Networks are universal approximation tools (Sandberg&Park,1993)



INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# 3 Lerning Rules for RBF-Networks

## Fixed choice of the centres and form parameters

- Centres are fixed by randomly selected initial clusters
- If m is the number of inner units and d the maximal distance between the centres:

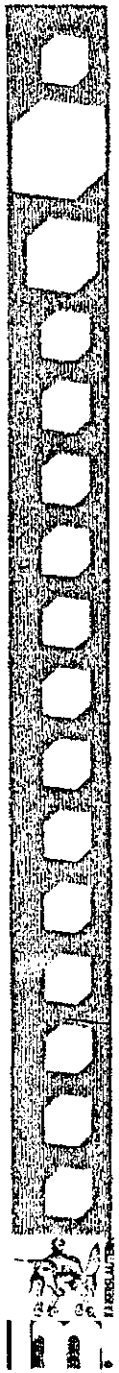
$$\sigma = \frac{d}{\sqrt{2\,m}}$$

- Computation of the weights of the RBF-layer to the output layer with Pseudo-Inverse

## Self-organizing learning of the centres

## Supervised learning of all the network parameters

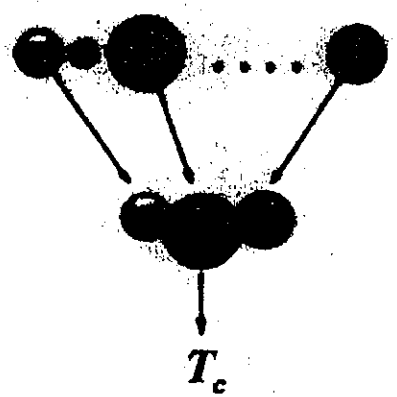




INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# Prediction of Critical Temperature

- Problem: Prediction of the critical temperature for complex organic molecules, consisted of 38 different structure groups




$$T_C : \mathrm{N}^{38} \rightarrow \Re$$

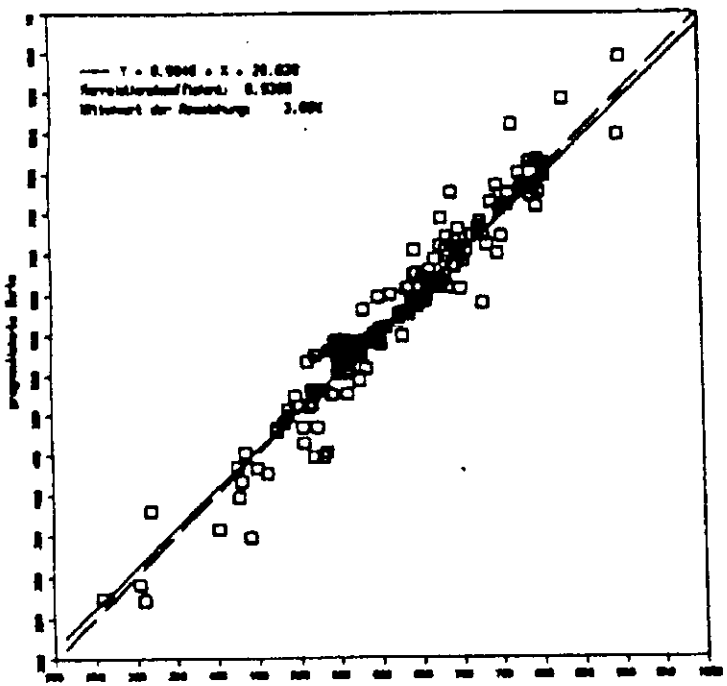
- Databasis has approximately 600 molecules

- There exist estimation methods, for example the Joback-formula:

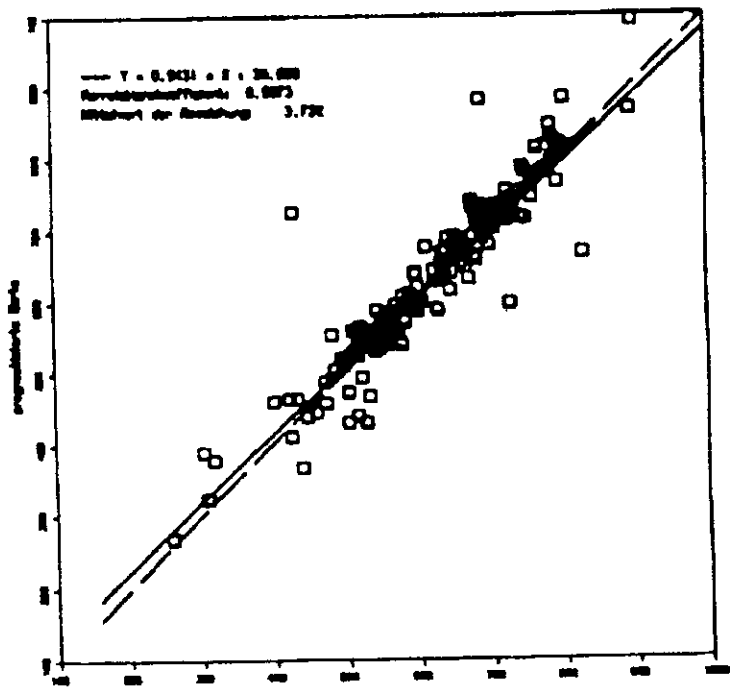
$$T_C = \frac{\mathrm{T_b}}{0.584 + 0.965\,(\sum_i \Delta \mathrm{T_i}) - (\sum_i \Delta \mathrm{T_i})^2}$$

# Prediction of the Critical Temperature

## Test set



Sigmoid–N., Test



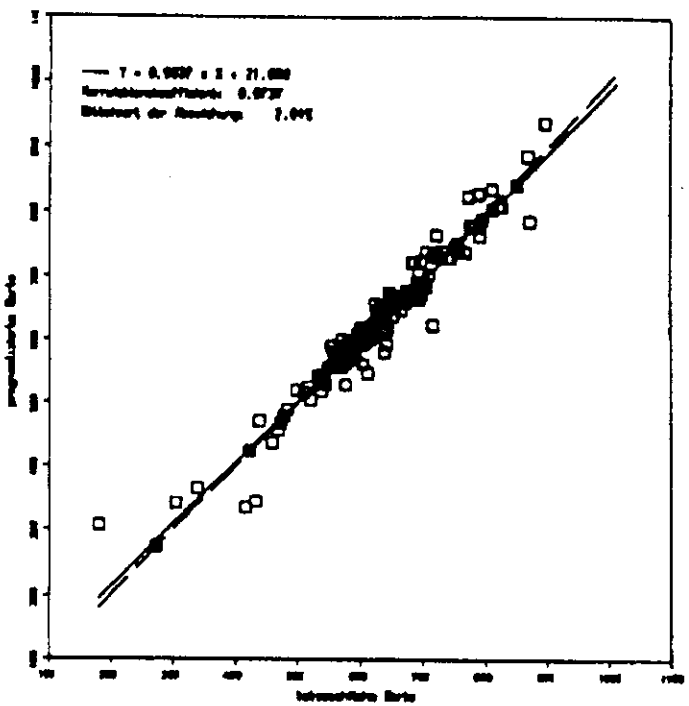
Gauß–N., Test



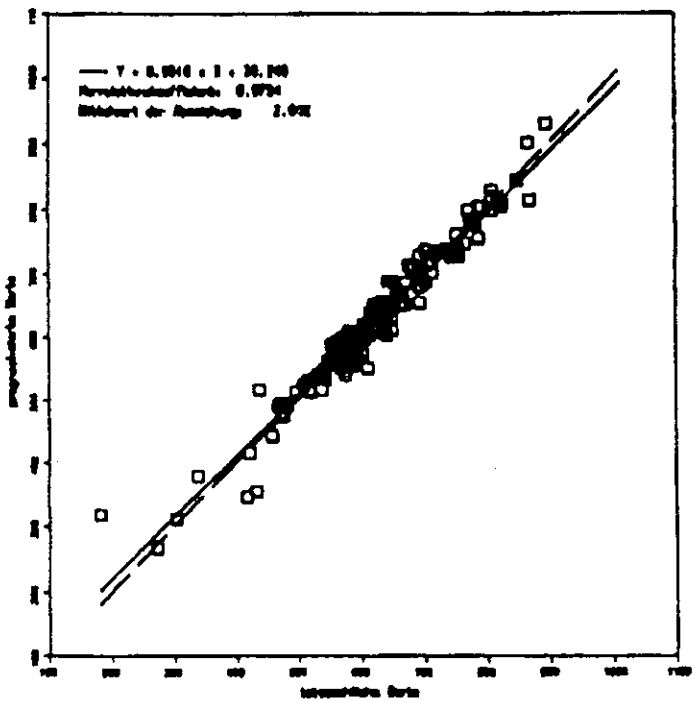
INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Prediction of the critical Temperature

## Training Set



Sigmoid–N., Lern

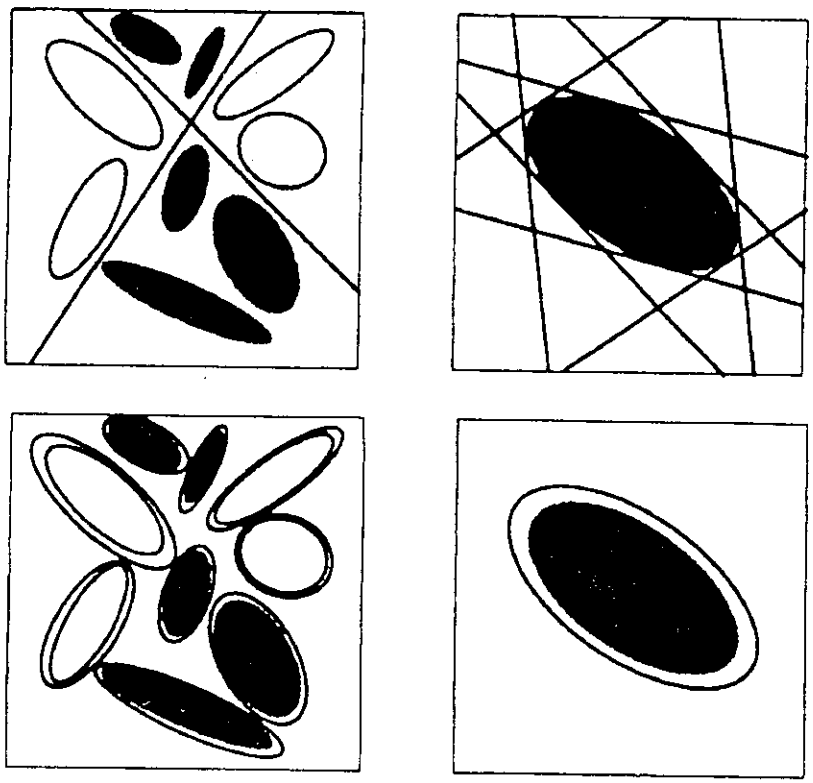


Gauß–N., Lern



INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

# RBF- versus Sigmoid-Networks



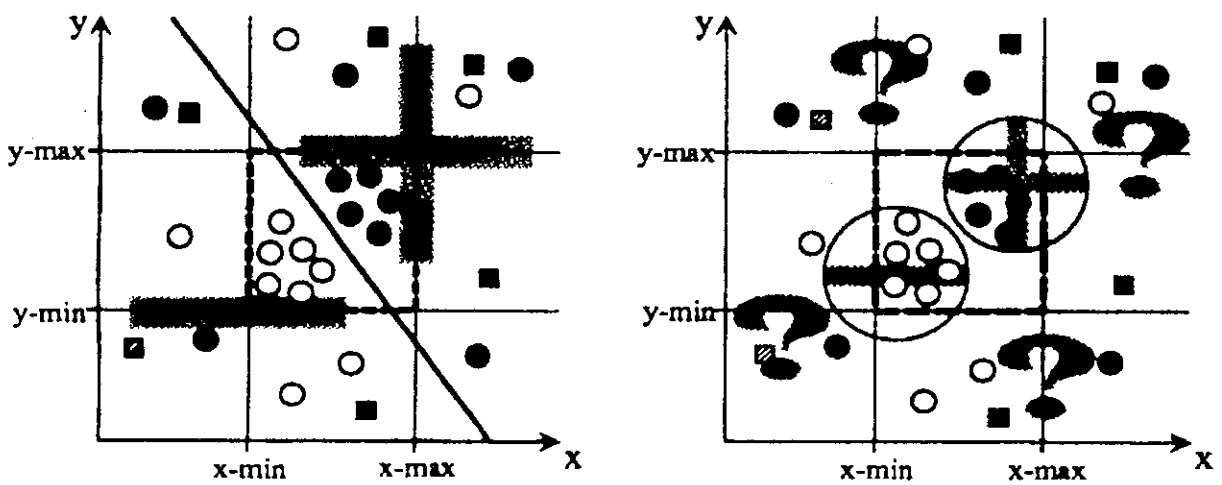
- Training and computation of a RBF-network is expensive compared with MLP
- Left column shows the situation: global classification is good
- Right column shows the situation: correct classification only through many hyper planes, good by a RBF-unit



INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK

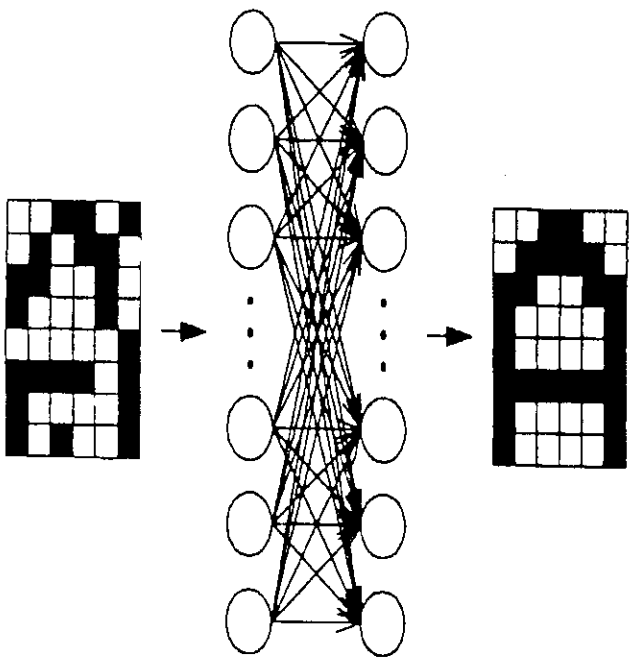
## Local versus global classification

- MLPs have hyper planes through the whole input space, i.e. global classification
- RBF-networks cluster the input space, i.e. only local classification
- Global classification gives problems in non-representative learning problems




□ : Betrachteter Wertebereich
● : Positivklasse
○ : Negativklasse
■ : Bisher unbekannte 3. Klasse

# Linear Neural Networks



- Structure of a linear network
  - Input layer without processing
  - No hidden layer
  - Many linear output units
- Similarity to perceptron
- Application in cluster association and completion
- Add of linear hidden layers gives no benefit



INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# The Delta Rule

## Adaption step

- Cluster pair ( x, y)
- Learning rate $\sigma$
- Weight vector w

$$\Delta w_{ik} = \sigma (y_i - a_i) x_k$$

## Learning algorithm (delta-rule)

- Initialization of weight vector arbitrary, choose learning rate $\sigma > 0$
- Propagate all given clusters (x ,y) per epoch through the network and sum up the changes in the weights
- Do a modification of the weights after one epoch

## Remarks

- A linear neural network with n input units and m output units can store with the delta rule maximal n linear independent clusters without any error
- The delta rule gives an approximate gradient step
- Linear independence implies linear separabiliy, but not reverse



INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Higher Units

- In many cortex areas: unions of neighborhood neurons to higher functional units (micro column)
- Micro columns have neurons of a small little vertical cortex cylinder with little tenth of millimeters of the diameter (E.g.*: special orientation of a edge of a picture*)
- Micro columns of one type are orderd to special fields in a next higher step **(cortex)**.
*(E.g.: Analyse of orientation of edges)*
- Main groups of the cortex:
  - primary and secundary sensor fields
  - Association fields
  - primary and secundary motor fields
- All cortex have a similar structure, 6 layers



INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Control with Neural Networks

System $x(t+1) = f(x(t),u(t)) \quad x \in R^n, u \in R, \quad f(0,0) = 0$

$y(t) = h(x(t)) \quad h(0) = 0$

situations

- Given: f, h available: x
- Unknown: f, h available: x
- Unknown: f, h available: y instead of x

[redacted]

$$y(t) = h(x(t) =: \Psi_1(x(t))$$

$$y(t+1) = h(f(x(t),u(t)) =: \Psi_2(x(t),u(t))$$

...

$$y(t+n-1) = h \circ f^{n-1}(x(t),u(t)) =: \Psi_n(x(t),u(t),u(t+1),...,u(t+n-2))$$

with $Y_n(t) = (y(t),y(t+1),...,y(t+n-1)$, $U_{n-1}(t) = (u(t),u(t+1),...,u(t+n-2))$ $\quad \det \frac{\partial Y_n(t)}{\partial x(t)} \neq 0$

where

$$x(t) = g(Y_n(t),U_{n-1}(t))$$

$$y(t+n) = h(x(t+n)) = h(g(Y_n(t),U_{n-1}(t),u(t+n-1))$$

$$=: F(Y_n(t),U_n(t))$$

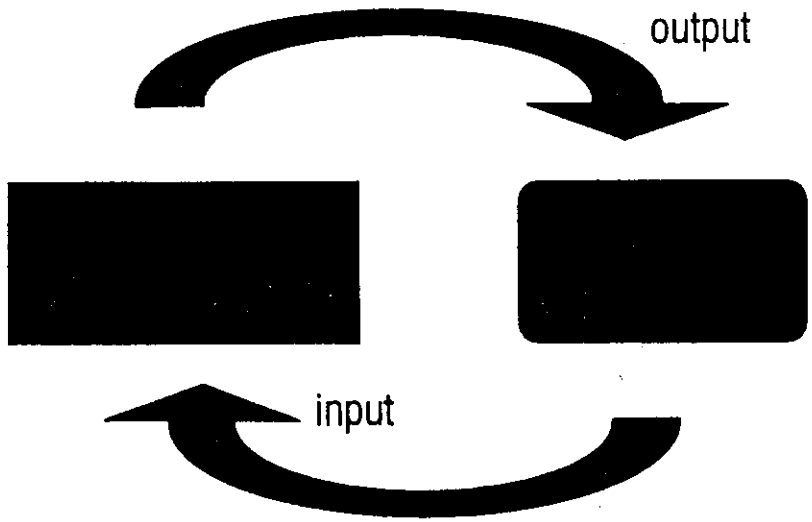


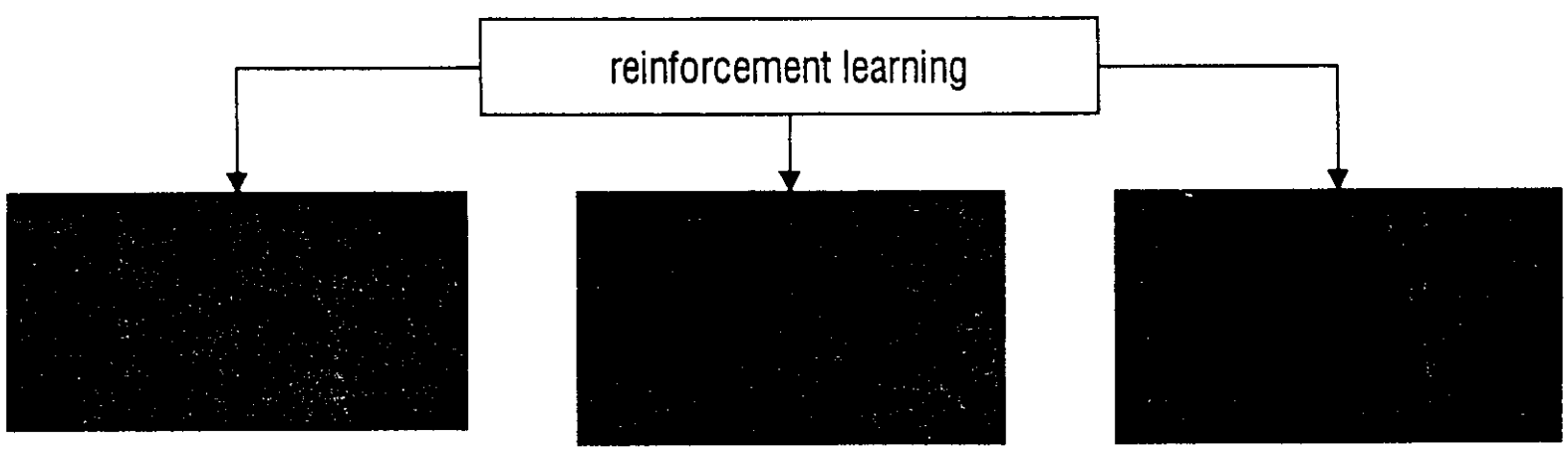
INSTITUT FÜR TECHNO-UND WIRTSCHAFTSMATHEMATIK

# Reinforcement learning

- Supervised learning
- 1 bit information: yes / no, result: good/ bad as gain signal
- Learning from reviewer instead of learning from supervisor
- also for non-recurrent networks
- stochastical neurons:

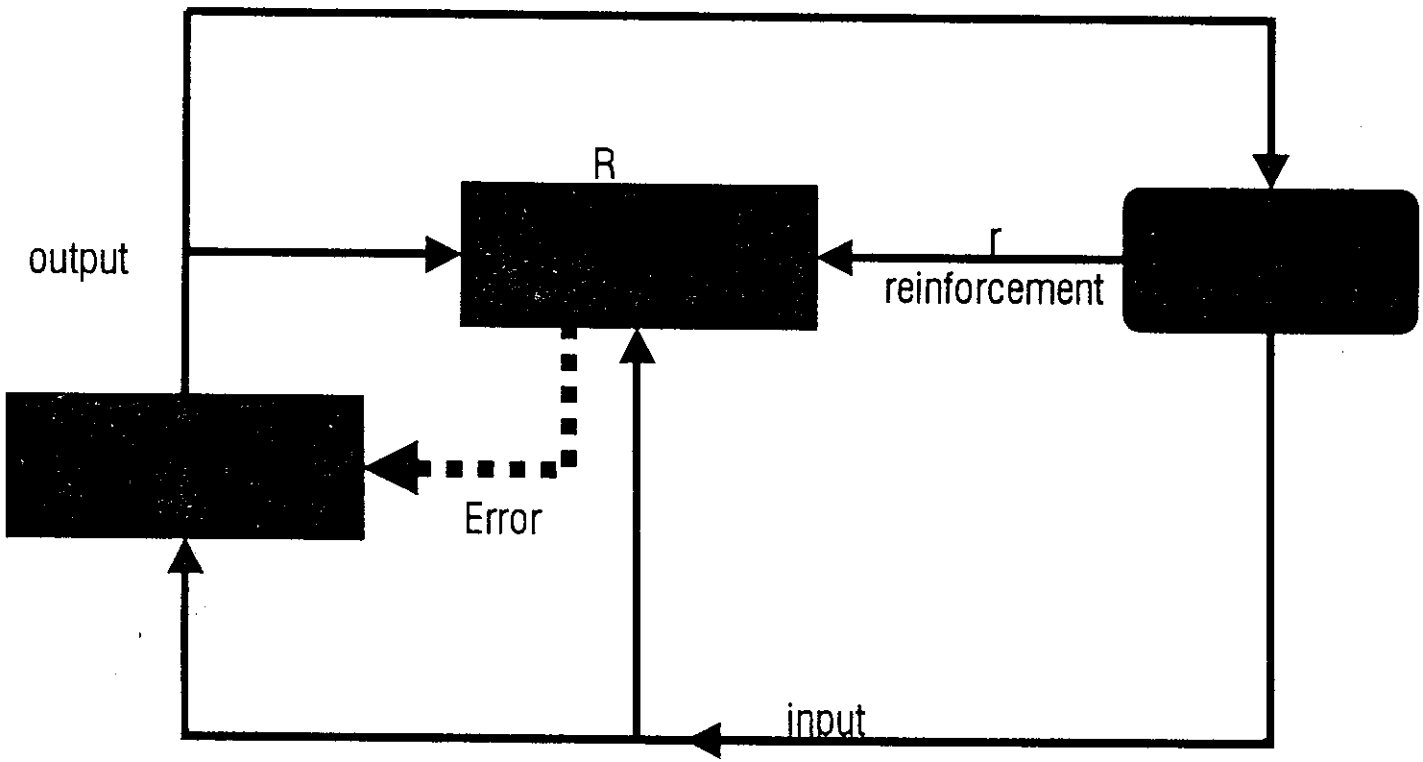
$$\Pr(S_i = \pm 1) = \frac{1}{1 + \exp(\mp \lambda h_i)}$$



# Reinforcement learning

## 2 steps method




- Evaluation network learns map input - reinforcement signal
  Minimization of $(R-r)^2$
- Trained evaluation network for training of the network
  Maximization of R, where 1 = success, 0 = failure
- Minimization of $(R-r)^2$ but useless without maximization of R

INSTITUT FÜR TECHNO-UND
WIRTSCHAFTSMATHEMATIK