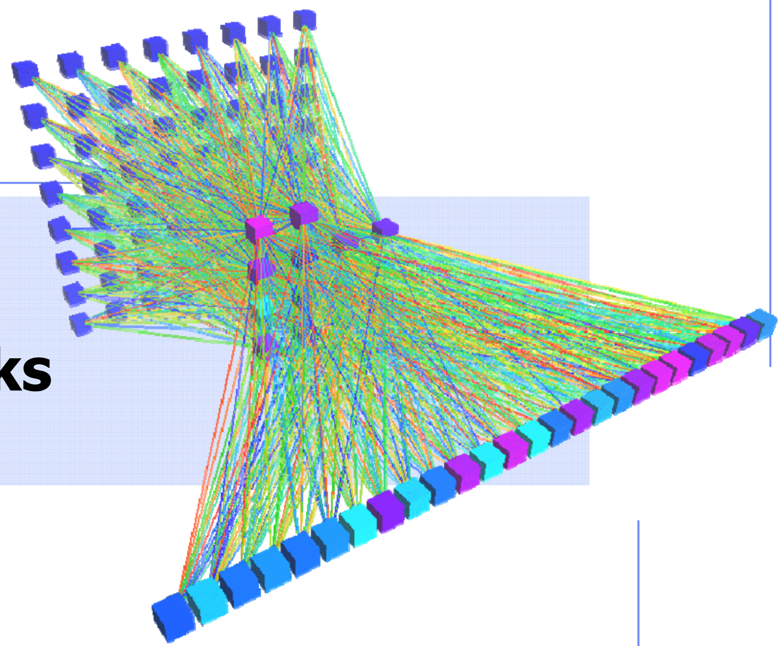
Forecasting with Artificial Neural Networks



EVIC 2005 Tutorial
Santiago de Chile, 15 December 2005

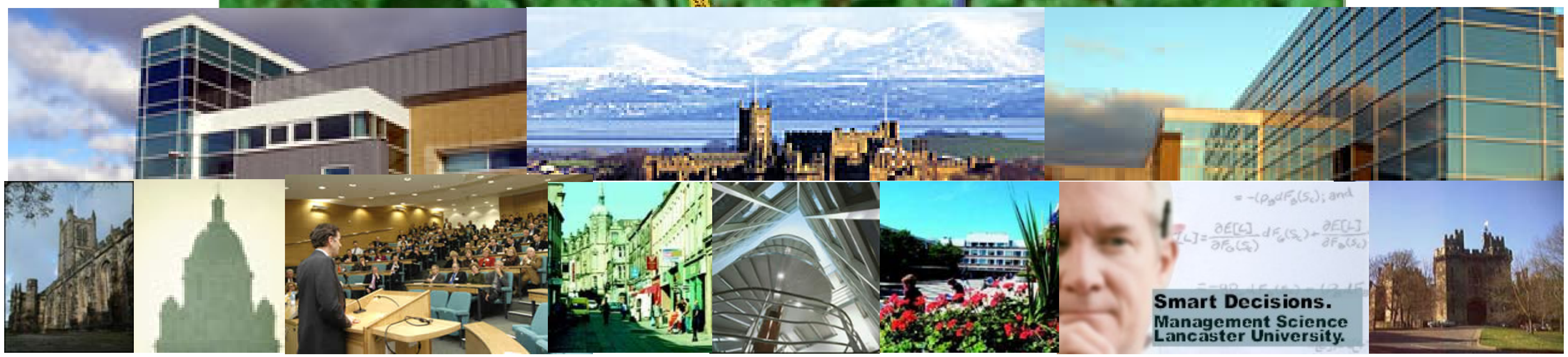
→ slides on www.neural-forecasting.com

Sven F. Crone
Centre for Forecasting
Department of Management Science
Lancaster University Management School
email: s.crone@neural-foreasting.com

Universidad de Chile
Ingenieria Industrial
Gestion de Empresas

Lancaster University

Lancaster University Management School?



What you can expect from this session …

- Simple back propagation algorithm [Rumelhart et al. 1982]

$$E_p = C(t_{pj}, o_{pj}) \quad o_{pj} = f_j(net_{pj}) \qquad \Delta_p w_{ji} \propto -\frac{\partial C(t_{pj}, o_{pj})}{\partial w_{ji}}$$

$$\frac{\partial C(t_{pj}, o_{pj})}{\partial w_{ji}} = \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}}$$

$$\delta_{pj} = -\frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pj}}$$

$$\delta_{pj} = -\frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pj}} = \frac{\partial C(t_{pj}, o_{pj})}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}}$$

$$\frac{\partial o_{pt}}{\partial net_{pj}} = f_j'(net_{pj})$$

$$\delta_{pj} = \frac{\partial C(t_{pj}, o_{pj})}{\partial o_{pj}} f_j'(net_{pj})$$

$$\sum_k \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}} = \sum_k \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pk}} \frac{\partial \sum_i w_{ki} o_{pi}}{\partial o_{pj}}$$

$$= \sum_k \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pk}} w_{kj} = -\sum_k \delta_{pj} w_{kj}$$

$$\delta_{pj} = f_j'(net_{pj}) \sum_k \delta_{pj} w_{kj}$$

$$\delta_{pj} = \begin{cases} \dfrac{\partial C(t_{pj}, o_{pj})}{\partial o_{pj}} f_j'(net_{pj}) & \text{if unit } j \text{ is in the output layer} \\ f_j'(net_{pj}) \sum_k \delta_{pk} w_{pjk} & \text{if unit } j \text{ is in a hidden layer} \end{cases}$$

→ „How to …“ on Neural Network Forecasting **with limited maths!**

→ **CD-Start-Up Kit for** Neural Net Forecasting

- → 20+ software simulators
- → datasets
- → literature & faq

→ slides, data & additional info on www.neural-forecasting.com

Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

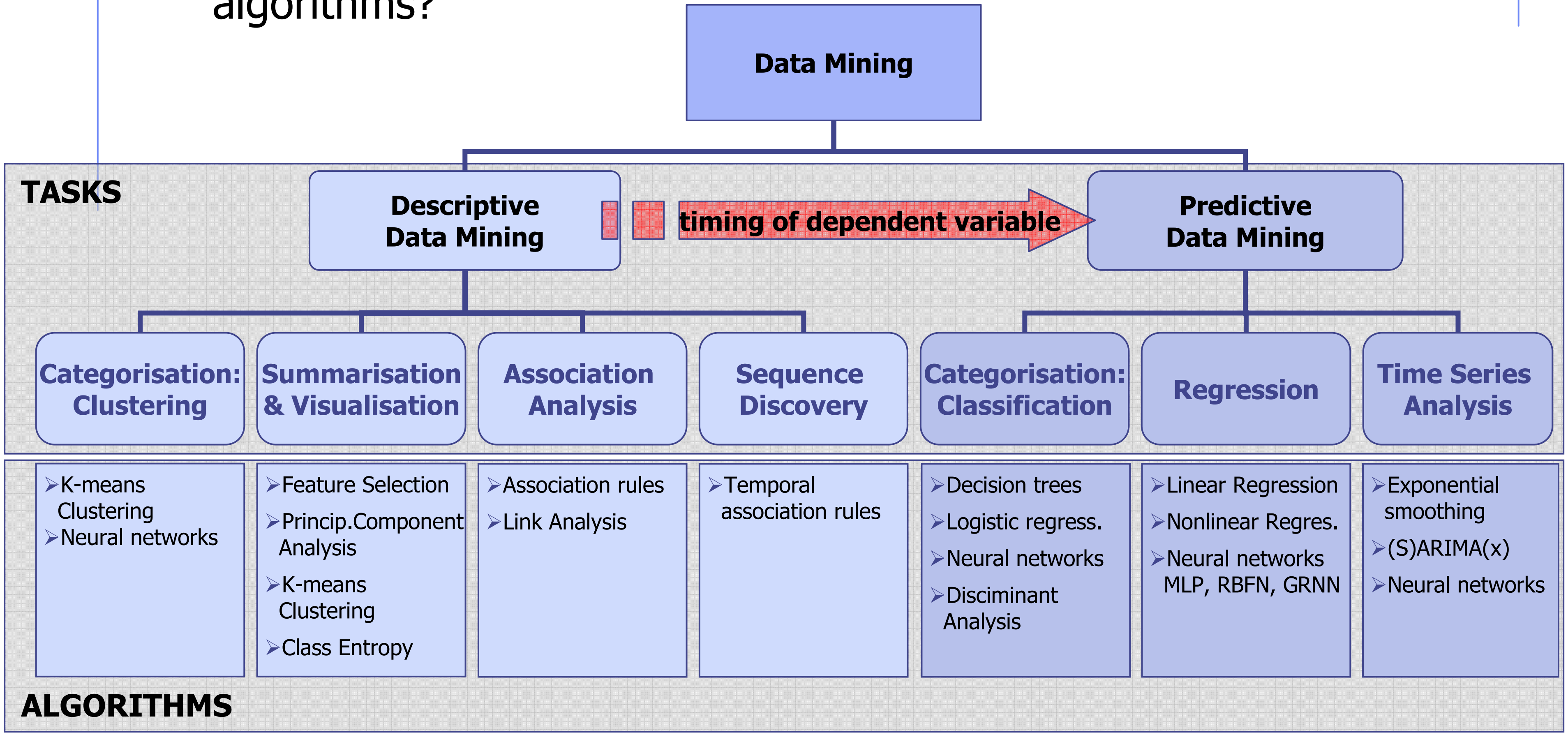
Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
   1. Forecasting as predictive Regression
   2. Time series prediction vs. causal prediction
   3. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Forecasting or Prediction?

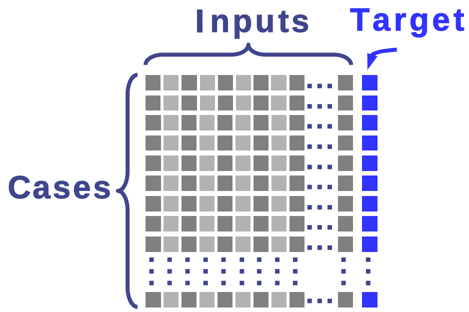
- Data Mining: „ Application of data analysis algorithms & discovery algorithms that extract patterns out of the data" → algorithms?



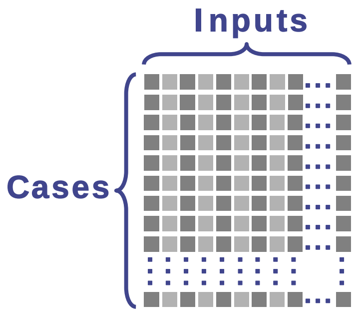
Forecasting or Classification

Independent / dependent	Metric scale	Ordinal scale	Nominal scale
Metric sale	▪ Regression ▪ Time Series Analysis	DOWNSCALE	▪ Analysis of Variance
Ordinal scale	DOWNSCALE	DOWNSCALE	DOWNSCALE
Nominal scale	▪ Classification	DOWNSCALE	▪ Contingency Analysis
NONE	▪ Principal Component Analysis		▪ Clustering

Supervised learning



Unsupervised learning



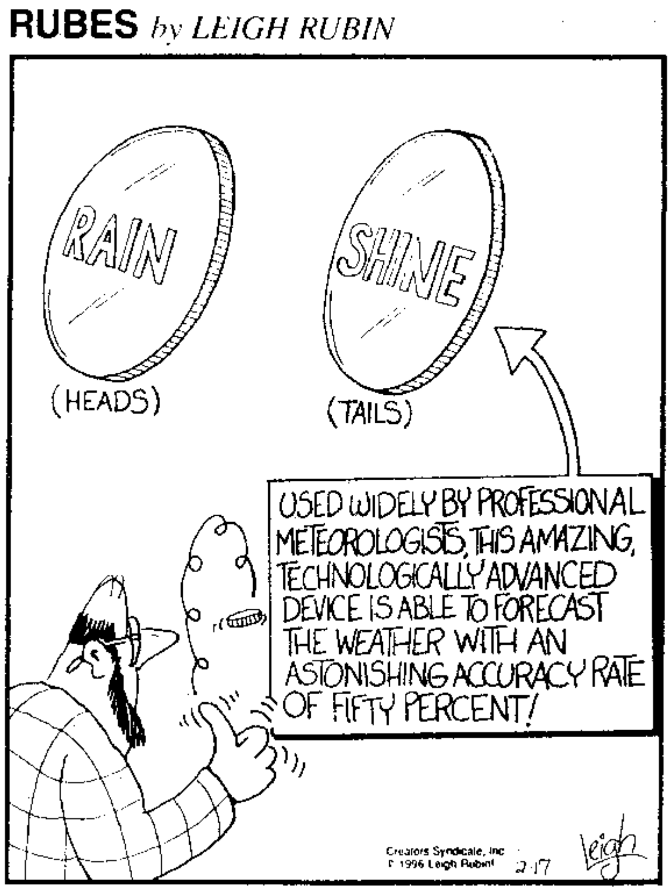
- Simplification from Regression ("How much") → Classification ("Will event occur")

→ FORECASTING = PREDICTIVE modelling (dependent variable is in future)

→ FORECASTING = REGRESSION modelling (dependent variable is of metric scale)

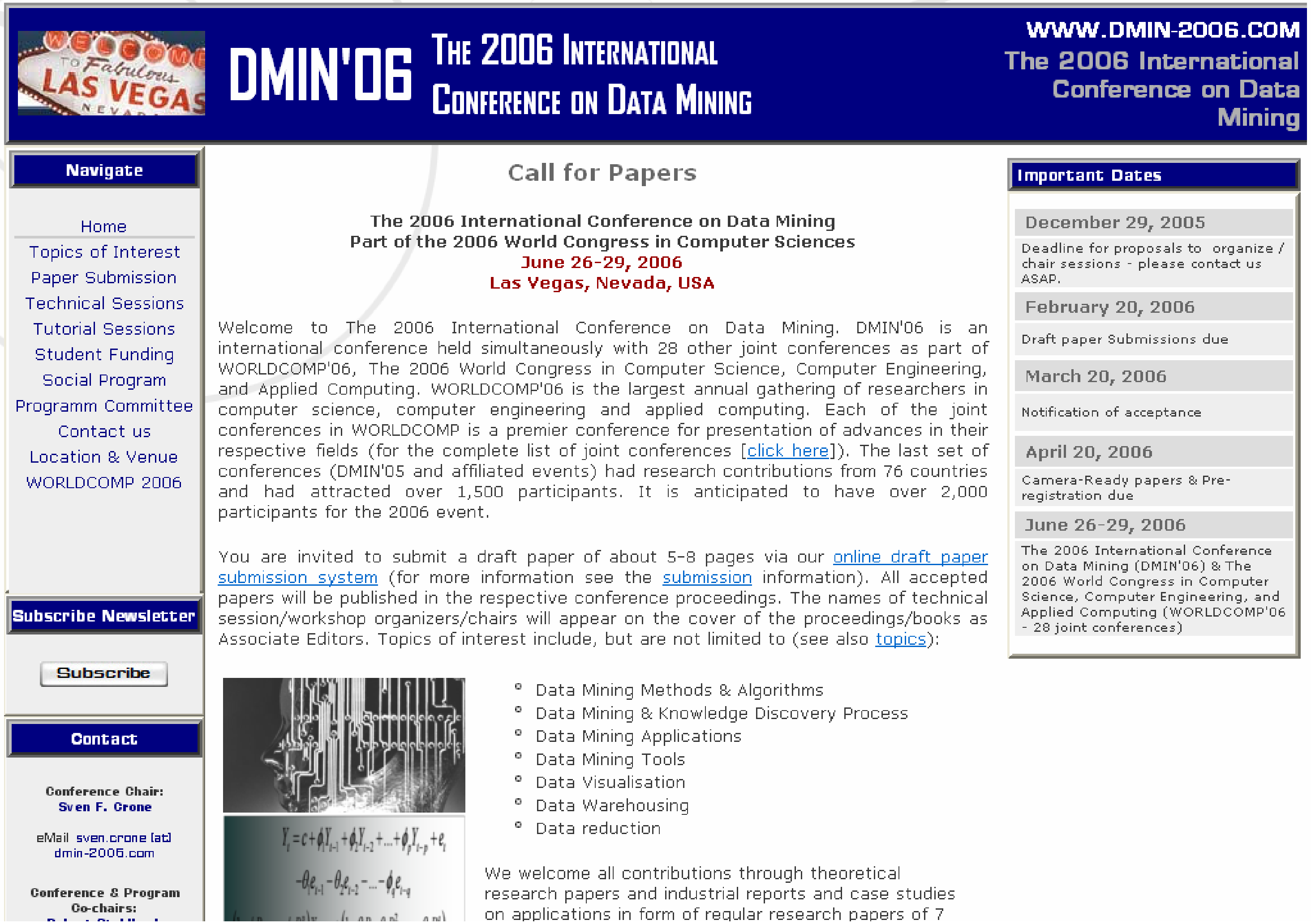
Forecasting or Classification?

- What the experts say ...



International Conference on Data Mining

- You are welcome to contribute … www.dmin-2006.com !!!



DMIN'06 THE 2006 INTERNATIONAL CONFERENCE ON DATA MINING

WWW.DMIN-2006.COM
The 2006 International Conference on Data Mining

Navigate

Home
Topics of Interest
Paper Submission
Technical Sessions
Tutorial Sessions
Student Funding
Social Program
Programm Committee
Contact us
Location & Venue
WORLDCOMP 2006

Subscribe Newsletter

Subscribe

Contact

Conference Chair:
Sven F. Crone

eMail sven.crone [at] dmin-2006.com

Conference & Program Co-chairs:

Call for Papers

The 2006 International Conference on Data Mining
Part of the 2006 World Congress in Computer Sciences
June 26-29, 2006
Las Vegas, Nevada, USA

Welcome to The 2006 International Conference on Data Mining. DMIN'06 is an international conference held simultaneously with 28 other joint conferences as part of WORLDCOMP'06, The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing. WORLDCOMP'06 is the largest annual gathering of researchers in computer science, computer engineering and applied computing. Each of the joint conferences in WORLDCOMP is a premier conference for presentation of advances in their respective fields (for the complete list of joint conferences [click here]). The last set of conferences (DMIN'05 and affiliated events) had research contributions from 76 countries and had attracted over 1,500 participants. It is anticipated to have over 2,000 participants for the 2006 event.

You are invited to submit a draft paper of about 5-8 pages via our online draft paper submission system (for more information see the submission information). All accepted papers will be published in the respective conference proceedings. The names of technical session/workshop organizers/chairs will appear on the cover of the proceedings/books as Associate Editors. Topics of interest include, but are not limited to (see also topics):

- Data Mining Methods & Algorithms
- Data Mining & Knowledge Discovery Process
- Data Mining Applications
- Data Mining Tools
- Data Visualisation
- Data Warehousing
- Data reduction

We welcome all contributions through theoretical research papers and industrial reports and case studies on applications in form of regular research papers of 7

Important Dates

December 29, 2005
Deadline for proposals to organize / chair sessions - please contact us ASAP.

February 20, 2006
Draft paper Submissions due

March 20, 2006
Notification of acceptance

April 20, 2006
Camera-Ready papers & Pre-registration due

June 26-29, 2006
The 2006 International Conference on Data Mining (DMIN'06) & The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'06 - 28 joint conferences)

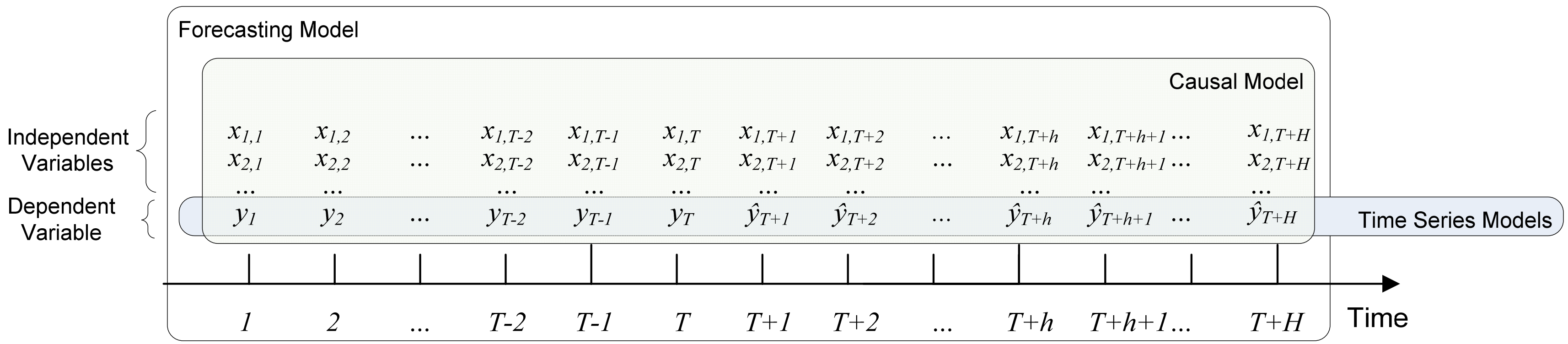
Agenda

Forecasting with Artificial Neural Networks

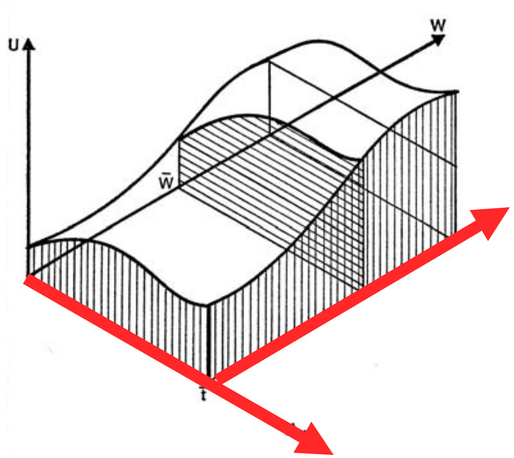
1. Forecasting?
   1. Forecasting as predictive Regression
   2. Time series prediction vs. causal prediction
   3. SARIMA-Modelling
   4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Forecasting Models

- Time series analysis vs. causal modelling



- Time series prediction (Univariate)
  - Assumes that data generating process that creates patterns can be explained only from previous observations of dependent variable
- Causal prediction (Multivariate)
  - Data generating process can be explained by interaction of causal (cause-and-effect) independent variables



Classification of Forecasting Methods

- **Forecasting Methods**
  - **Objective Forecasting Methods**
    - **Time Series Methods**
      - Averages
      - Moving Averages
      - Naive Methods
      - Exponential Smoothing
        - Simple ES
        - Linear ES
        - Seasonal ES
        - Dampened Trend ES
      - Simple Regression
      - Autoregression ARIMA
      - Neural Networks
    - **Causal Methods**
      - Linear Regression
      - Multiple Regression
      - Dynamic Regression
      - Vector Autoregression
      - Intervention model
      - Neural Networks
  - **Subjective Forecasting Methods**
    - Sales Force Composite
    - Analogies
    - Delphi
    - PERT
    - Survey techniques
  - **„Prophecy“ educated guessing…**

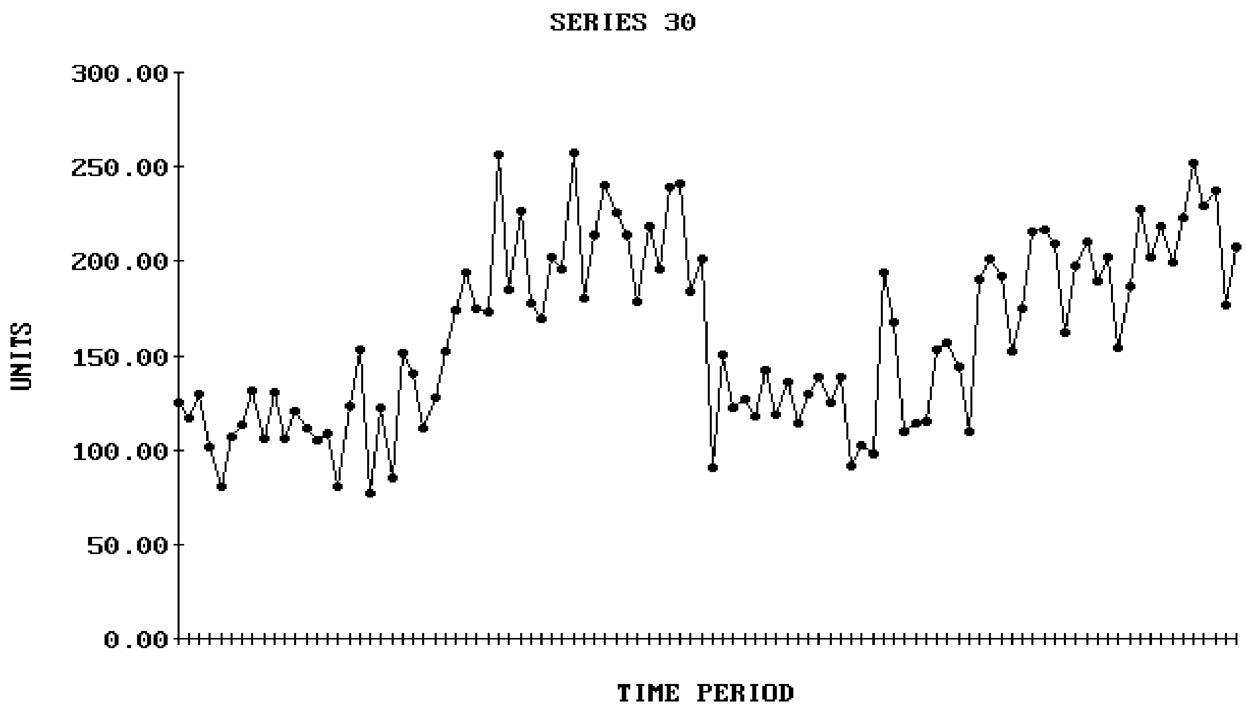
Demand Planning Practice
Objektive Methods + Subjektive correction

Neural Networks ARE
- time series methods
- causal methods

& CAN be used as
- Averages & ES
- Regression …

Time Series Definition

- Definition
  - Time Series is a series of timely ordered, comparable observations y_t recorded in equidistant time intervals
- Notation
  - Y_t represents the t th period observation, t=1,2 … n



Concept of Time Series

- An observed measurement is made up of a
  - **systematic part** and a
  - **random part**

- Approach
  - Unfortunately we cannot observe either of these !!!
  - Forecasting methods try to isolate the systematic part
  - Forecasts are based on the systematic part
  - The random part determines the distribution shape

- Assumption
  - Data observed over time is comparable
    - The time periods are of identical lengths (check!)
    - The units they are measured in change (check!)
    - The definitions of what is being measured remain unchanged (check!)
    - They are correctly measured (check!)
  - data errors arise from sampling, from bias in the instruments or the responses, from transcription.

Objective Forecasting Methods – Time Series

Methods of Time Series Analysis / Forecasting

- Class of objective Methods
- based on analysis of past observations of dependent variable alone

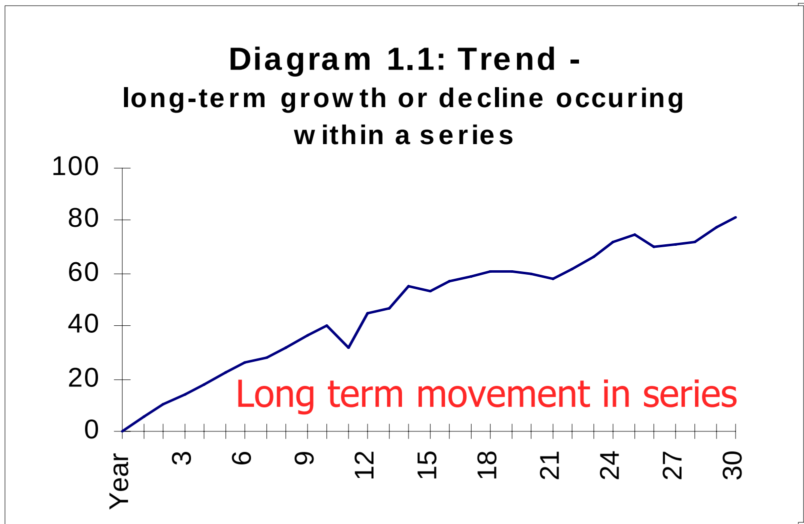
- Assumption
  - there exists a cause-effect relationship, that keeps repeating itself with the yearly calendar
  - Cause-effect relationship may be treated as a BLACK BOX
  - TIME-STABILITY-HYPOTHESIS ASSUMES NO CHANGE:
    → Causal relationship remains intact indefinitely into the future!
  - the time series can be explained & predicted solely from previous observations of the series

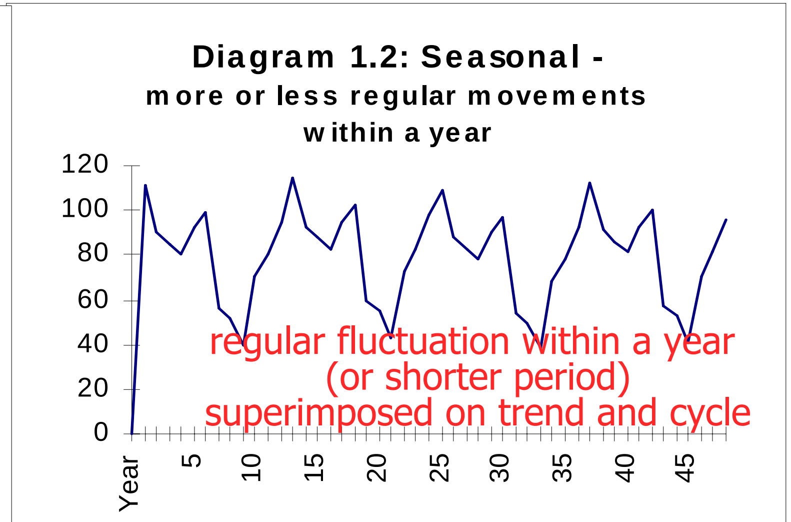
→ Time Series Methods consider only past patterns of same variable

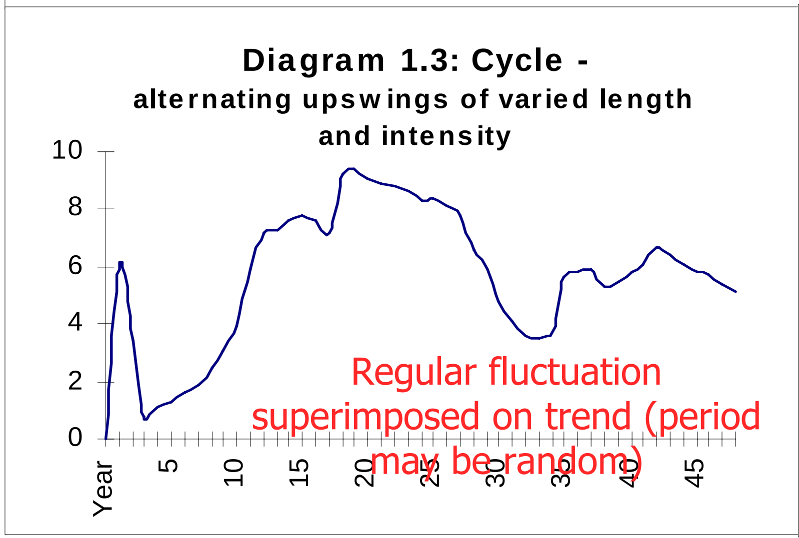
→ Future events (no occurrence in past) are explicitly NOT considered!

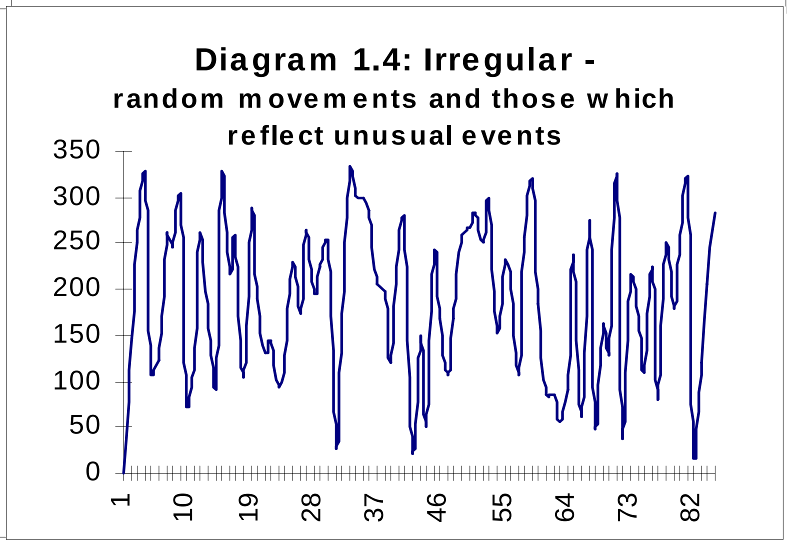
→ external EVENTS relevant to the forecast must be corected MANUALLY

Simple Time Series Patterns





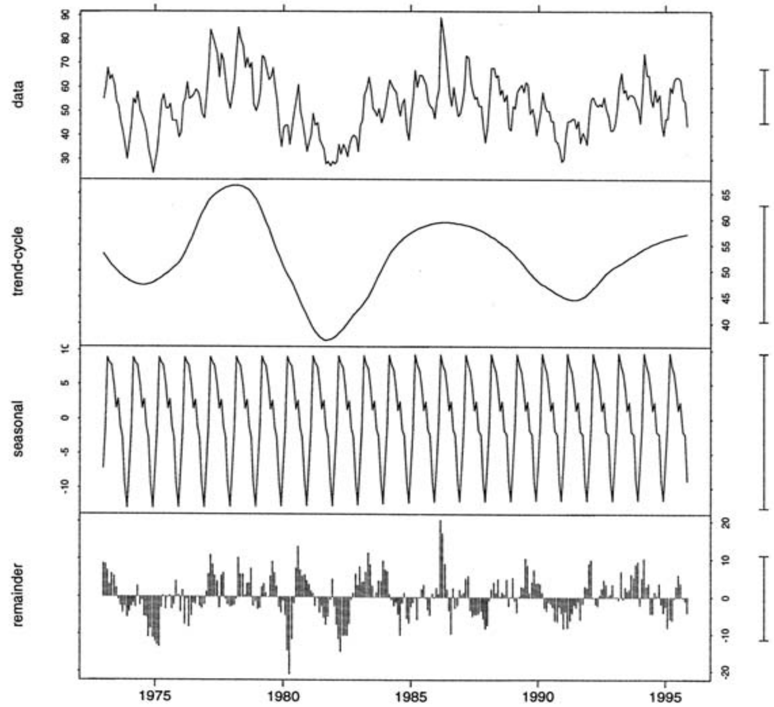




Regular Components of Time Series

A Time Series consists of superimposed components / patterns:

- Signal
  - level 'L'
  - trend 'T'
  - seasonality 'S'
- Noise
  - irregular,error 'e'



Y = L + S + T + E

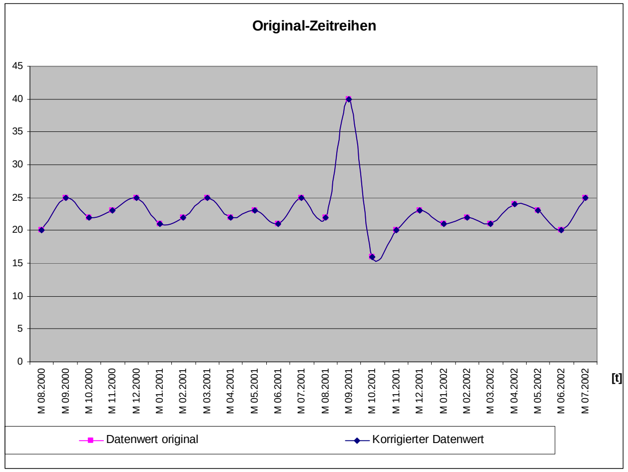
Sales = LEVEL + SEASONALITY + TREND + RANDOM ERROR

Y = L * S * T * E

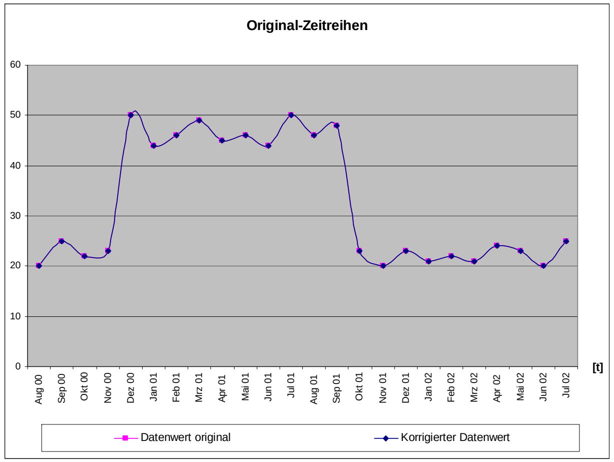
Irregular Components of Time Series

Structural changes in systematic data

- PULSE
  - one time occurrence
  - on top of systematic stationary / trended / seasonal development
- LEVEL SHIFT
  - one time / multiple time shifts
  - on top of systematic stationary / trended / seasonal etc. development
- STRUCTURAL BREAKS
  - Trend changes (slope, direction)
  - Seasonal pattern changes & shifts



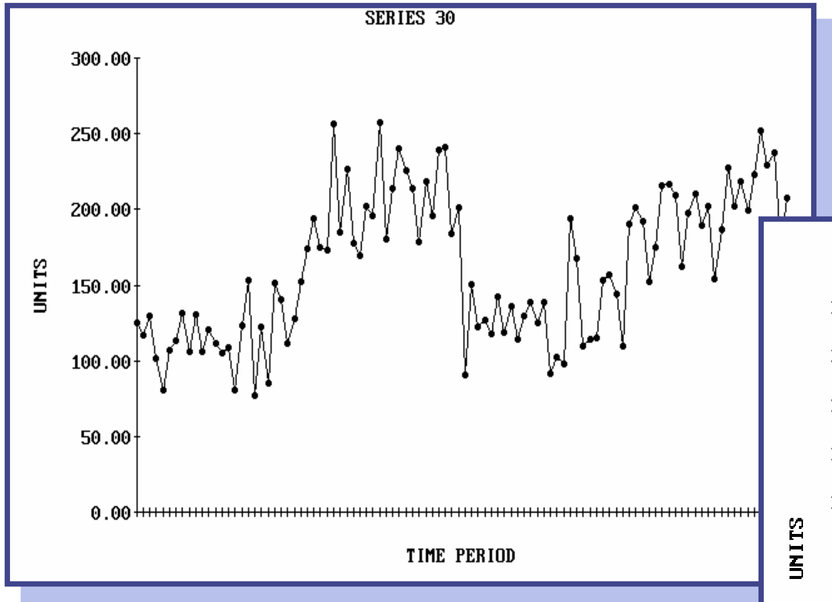
STATIONARY time series with PULSE

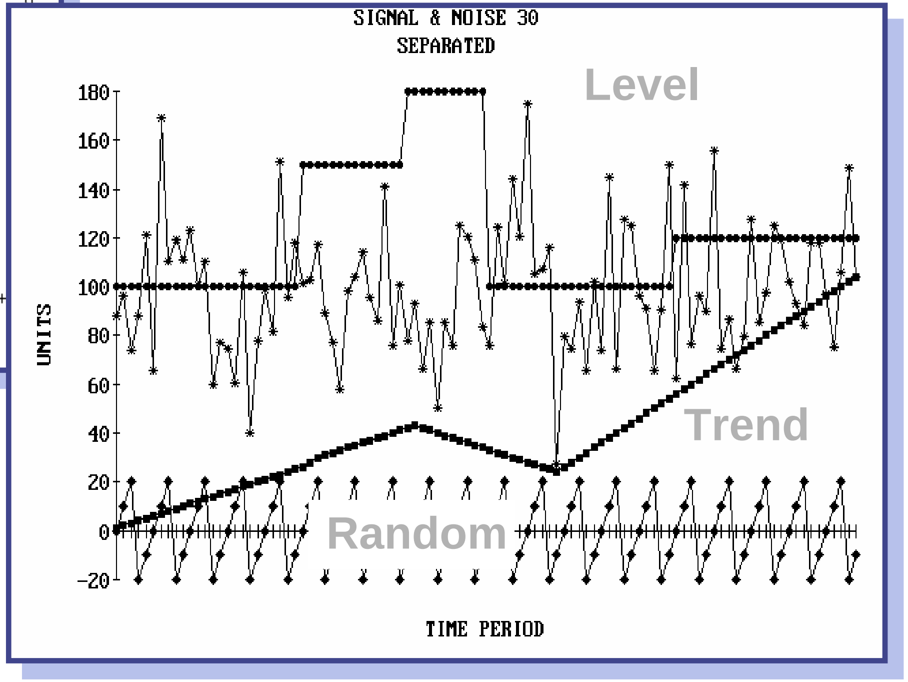


STATIONARY time series with level shift

Components of Time Series

- Time Series



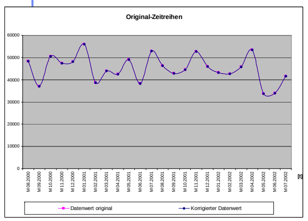


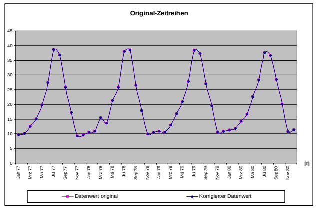
- Time Series → decomposed into Components

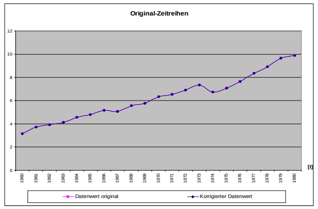
Time Series Patterns

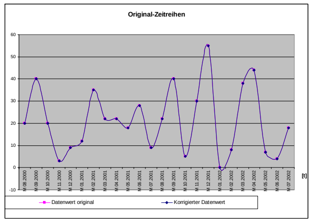
Time Series Pattern

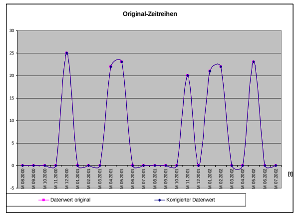
- REGULAR Time Series Patterns
  - STATIONARY) Time Series
  - SEASONAL Time Series
  - TRENDED Time Series
- IRREGULAR Time Series Patterns
  - FLUCTUATING Time Series
  - INTERMITTANT Time Series











$Y_t = f(E_t)$

time series is influenced by level & random fluctuations

$Y_t = f(S_t, E_t)$

time series is influenced by level, season and random fluctuations

$Y_t = f(T_t, E_t)$

time series is influenced by trend from level and random fluctuations

time series fluctuates very strongly around level (mean deviation > ca. 50% around mean)

Number of periods with zero sales is high (ca. 30%-40%)

Combination of individual Components

$$Y_t = f(S_t, T_t, E_t)$$

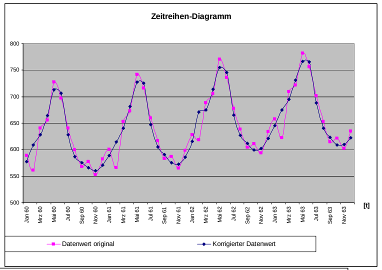
+ PULSES!

+ LEVEL SHIFTS!

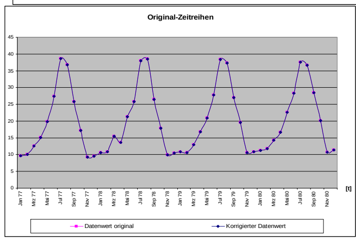
+ STRUCTURAL BREAKS!

Components of complex Time Series

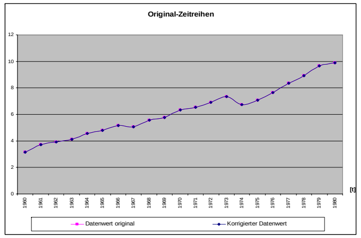
Sales or observation of time series at point t ⇒ Y_t



consists of a combination of = $f(\quad)$

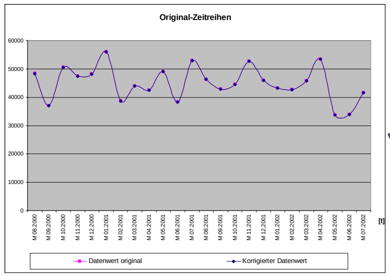


Base Level + Seasonal Component ⇒ S_t ,



Trend-Component ⇒ T_t ,

Irregular or random Error-Component ⇒ E_t



Different possibilities to combine components

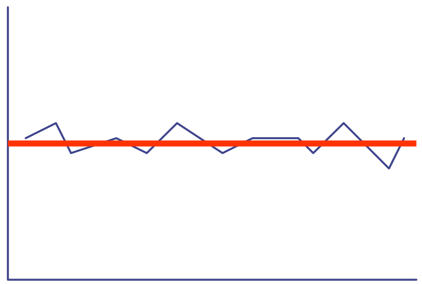
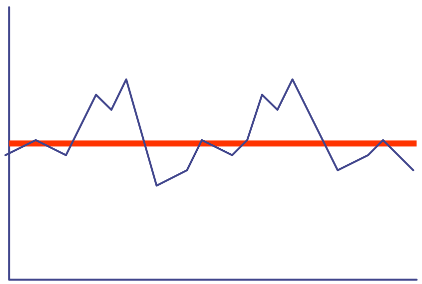
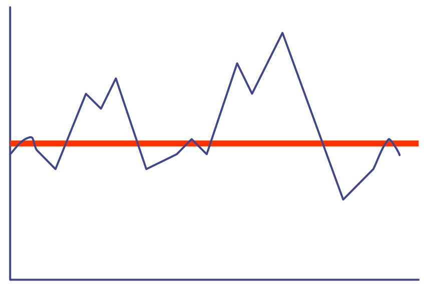
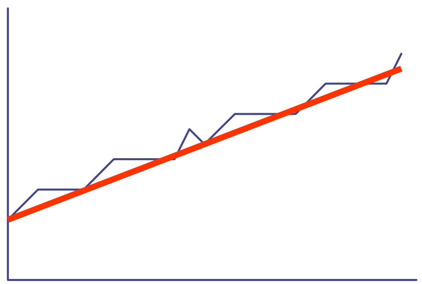
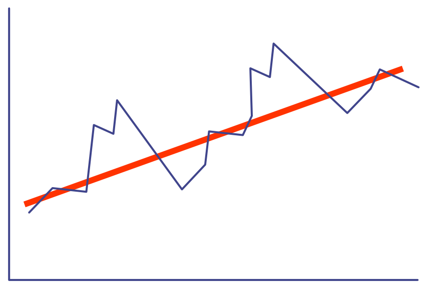
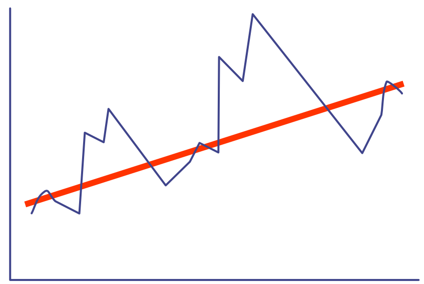
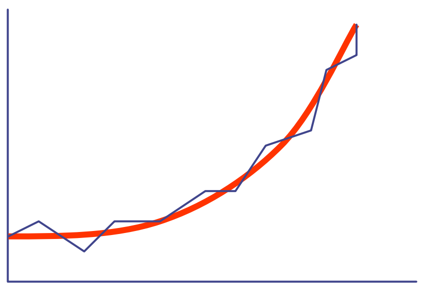
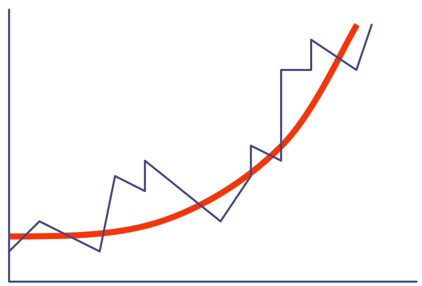
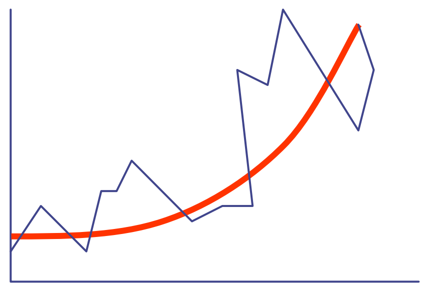
Additive Model

$$Y_t = L + S_t + T_t + E_t$$

Multiplicative Model

$$Y_t = L * S_t * T_t * E_t$$

Classification of Time Series Patterns

	No Seasonal Effect	Additive Seasonal Effect	Multiplicative Seasonal Effect
No Trend Effect			
Additive Trend Effect			
Multiplicative Trend Effect			

[Pegels 1969 / Gardner]

Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
    1. Forecasting as predictive Regression
    2. Time series prediction vs. causal prediction
    3. SARIMA-Modelling
        1. SARIMA – Differencing
        2. SARIMA – Autoregressive Terms
        3. SARIMA – Moving Average Terms
        4. SARIMA – Seasonal Terms
    4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Introduction to ARIMA Modelling

- Seasonal Autoregressive Integrated Moving Average Processes: SARIMA
  - popularised by George Box & Gwilym Jenkins in 1970s (names often used synonymously)
  - models are widely studied
  - Put together theoretical underpinning required to understand & use ARIMA
  - Defined general notation for dealing with ARIMA models

→ **claim that most time series can be parsimoniously represented by the ARIMA class of models**

- **ARIMA (p, d, q)-Models** attempt to describe the systematic pattern of a time series by **3 parameters**
  - **p**: Number of autoregressive terms (AR-terms) in a time series
  - **d**: Number of differences to achieve stationarity of a time series
  - **q**: Number of moving average terms (MA-terms) in a time series

$$\Phi_p(B)(1-B)^d Z_t = \delta + \Theta_q(B)e_t$$

The Box-Jenkins Methodology for ARIMA models

Model Identification

Data Preparation
- Transform time series for stationarity
- Difference time series for stationarity

Model selection
- Examine ACF & PACF
- Identify potential Models (p,q)(sq,sp) auto

Model Estimation & Testing

Model Estimation
- Estimate parameters in potential models
- Select best model using suitable criterion

Model Diagnostics / Testing
- Check ACF / PACF of residuals → white noise
- Run portmanteau test of residuals

Re-identify

Model Application

Model Application
- Use selected model to forecast

ARIMA-Modelling

- ARIMA(p,d,q)-Models
  - ARIMA - Autoregressive Terms AR(p), with p=order of the autoregressive part
  - ARIMA - Order of Integration, d=degree of first differencing/integration involved
  - ARIMA - Moving Average Terms MA(q), with q=order of the moving average of error
  - SARIMA$_t$ (p,d,q)(P,D,Q) with S the (P,D,Q)-process for the seasonal lags

- Objective
  - Identify the appropriate ARIMA model for the time series
  - Identify AR-term
  - Identify I-term
  - Identify MA-term

- Identification through
  - Autocorrelation Function
  - Partial Autocorrelation Function

ARIMA-Models: Identification of *d*-term

- Parameter *d* determines order of integration

- ARIMA models assume stationarity of the time series
  - Stationarity in the mean
  - Stationarity of the variance (homoscedasticity)

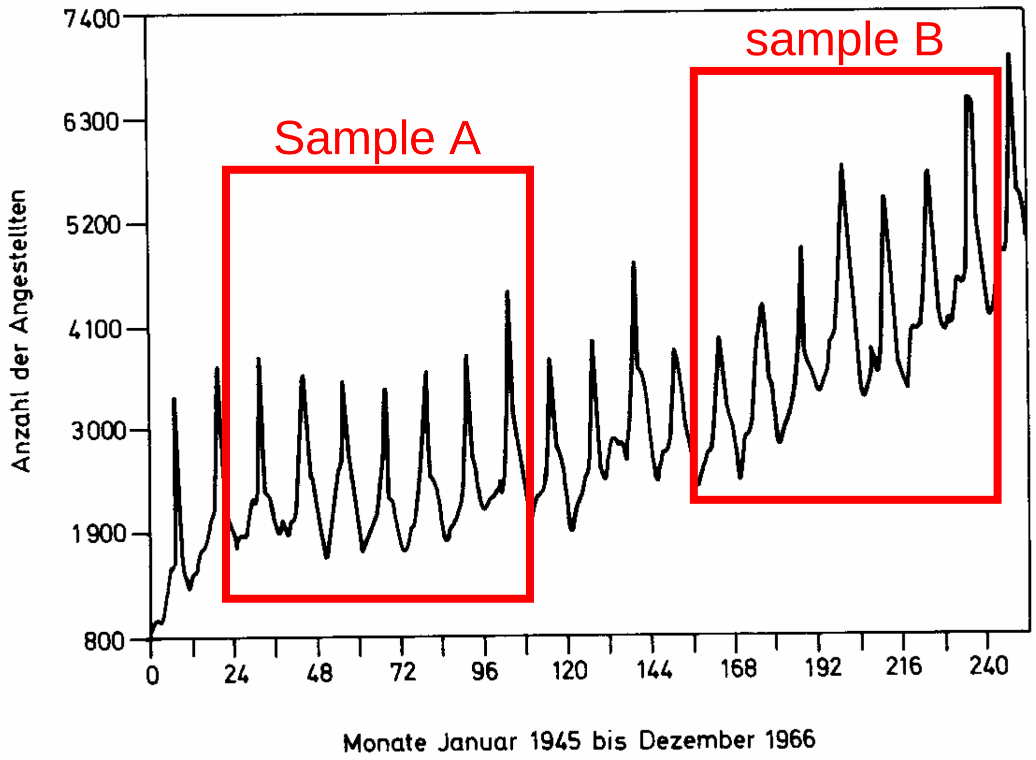
- Recap:
  - Let the mean of the time series at *t* be $\mu_t = E(Y_t)$
  - and $\lambda_{t,t-\tau} = \operatorname{cov}(Y_t, Y_{t-\tau})$

$$\lambda_{t,t} = \operatorname{var}(Y_t)$$

- Definition
  - A time series is stationary if its mean level μ_t is constant for all t and its variance and covariances $\lambda_{t-\tau}$ are constant for all t
  - In other words:
    - all properties of the distribution (mean, varicance, skewness, kurtosis etc.) of a random sample of the time series are independent of the absolute time *t* of drawing the sample → identity of mean & variance across time

ARIMA-Models: Stationarity and parameter *d*

- Is the time series stationary

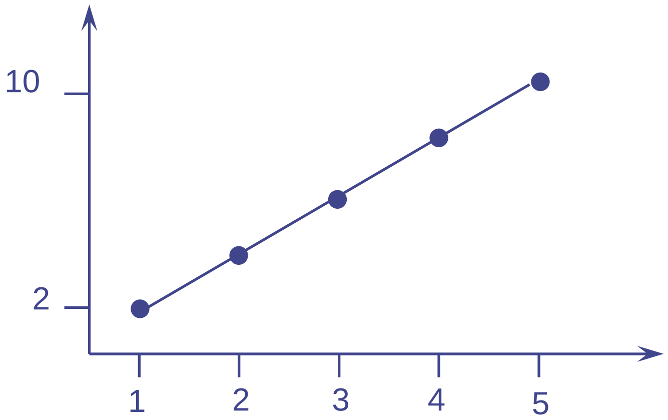


Stationarity:
μ(A)= μ(B)
var(A)=var(B)
etc.

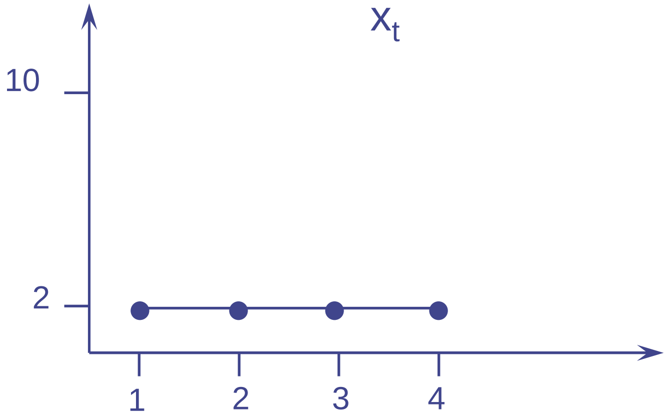
this time series:
μ(B)> μ(A) → trend
→ instationary time series

ARIMA-Modells: Differencing for Stationariry

- Differencing time series



x_t



$x_t - x_{t-1}$

- E.g. : time series $Y_t = \{2, 4, 6, 8, 10\}$.
- time series exhibits linear trend
- 1st order differencing between observation Y_t and predecessor Y_{t-1} derives a transformed time series:

4-2=2
6-4=2
8-6=2
10-8=2

→ The new time series $\Delta Y_t = \{2,2,2,2\}$ is stationary through 1st differencing

→ d=1 → ARIMA (0,1,0) model

→ 2nd order differences: d=2

ARIMA-Modells: Differencing for Stationariry

- **Integration**
  - **Differencing**

$$Z_t = Y_t - Y_{t-1}$$

  - **Transforms: Logarithms etc.**
  - **...**
  - **Where Zt is a transform of the variable of interest Yt chosen to make Z_t-Z_{t-1}-(Z_{t-1}-Z_{t-2})-… stationary**

- Tests for stationarity:
  - Dickey-Fuller Test
  - Serial Correlation Test
  - Runs Test

Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
    1. Forecasting as predictive Regression
    2. Time series prediction vs. causal prediction
    3. SARIMA-Modelling
        1. SARIMA – Differencing
        2. SARIMA – Autoregressive Terms
        3. SARIMA – Moving Average Terms
        4. SARIMA – Seasonal Terms
    4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

ARIMA-Models – Autoregressive Terms

- Description of Autocorrelation structure → auto regressive (AR) term
  - If a dependency exists between lagged observations Y_t and Y_{t-1} we can describe the realisation of Y_{t-1}

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + e_t$$

Observation in time t

Weight of the AR relationship

Observation in t-1 (independent)

random component („white noise")

  - Equations include only lagged realisations of the forecast variable
  - ARIMA(p,0,0) model = AR(p)-model

- Problems
  - Independence of residuals often violated (heteroscedasticity)
  - Determining number of past values problematic

- Tests for Autoregression: Portmanteau-tests
  - Box-Pierce test
  - Ljung-Box test

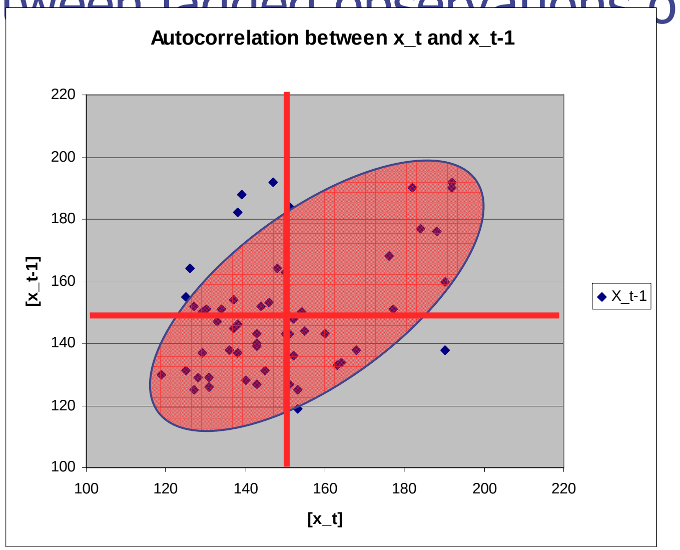
ARIMA-Modells: Parameter *p* of Autocorrelation

- stationary time series can be analysed for autocorrelation-structure
- The autocorrelation coefficient for lag k

$$\rho_k = \frac{\sum_{t=k+1}^{n} \left(Y_t - \bar{Y}\right)\left(Y_{t-k} - \bar{Y}\right)}{\sum_{t=1}^{n} \left(Y_t - \bar{Y}\right)}$$

denotes the correlation between lagged observations of distance k

- Graphical interpretation …
  - Uncorrelated data has low autocorrelations
  - Uncorrelated data shows no correlation patern
  - …



ARIMA-Modells: Parameter *p*

- E.g. time series Y_t 7, 8, 7, 6, 5, 4, 5, 6, 4.

lag 1:

7, 8
8, 7
7, 6
6, 5
5, 4
4, 5
5, 6
6, 4

r_1=.62

lag 2:

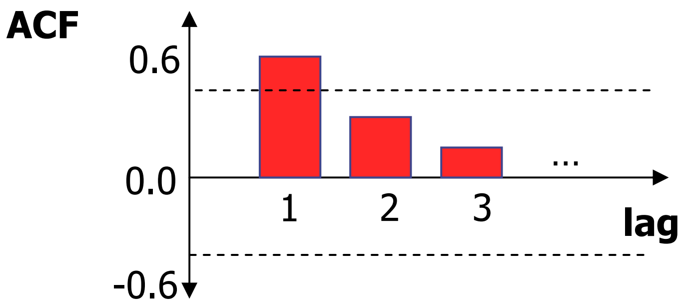
7, 7
8, 6
7, 5
6, 4
5, 5
4, 6
5, 4

r_2=.32

lag 3:

7, 6
8, 5
7, 4
6, 5
5, 6
4, 5

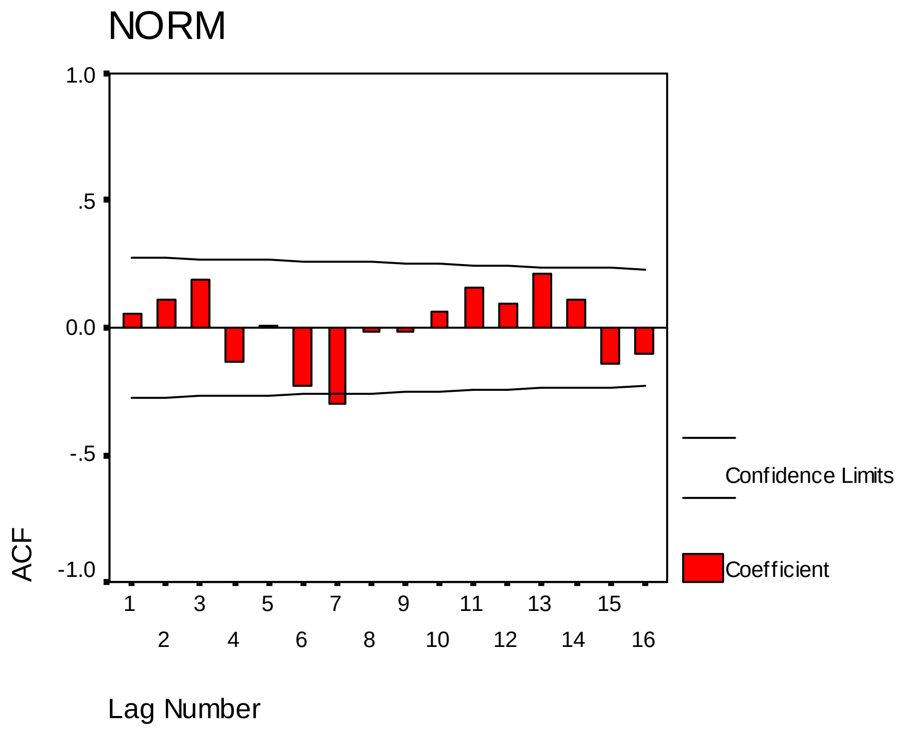
r_3=.15

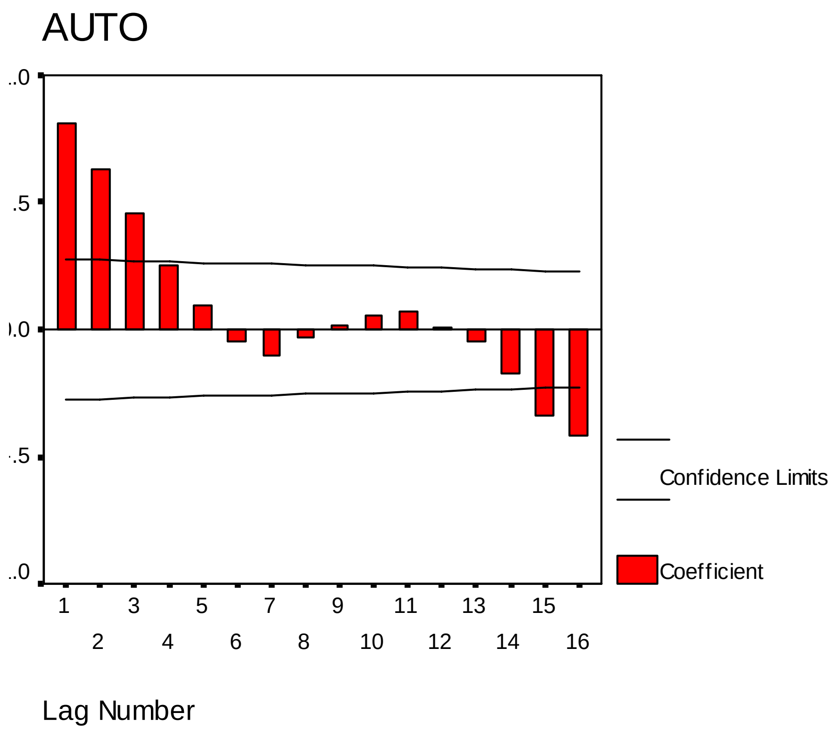


→ **Autocorrelations r_t gathered at lags 1, 2, … make up the autocorrelation function (ACF)**

ARIMA-Models – Autoregressive Terms

- Identification of AR-terms …?





- Random independent observations
- An AR(1) process?

ARIMA-Modells: Partial Autocorrelations

- Partical Autocorrelations are used to measure the degree of association between Y_t and Y_{t-k} when the effects of other time lags $1,2,3,\ldots,k-1$ are removed
  - Significant AC between Y_t and Y_{t-1}
    → significant AC between Y_{t-1} and Y_{t-2}
    → induces correlation between Y_t and Y_{t-2} ! (1st AC = PAC!)

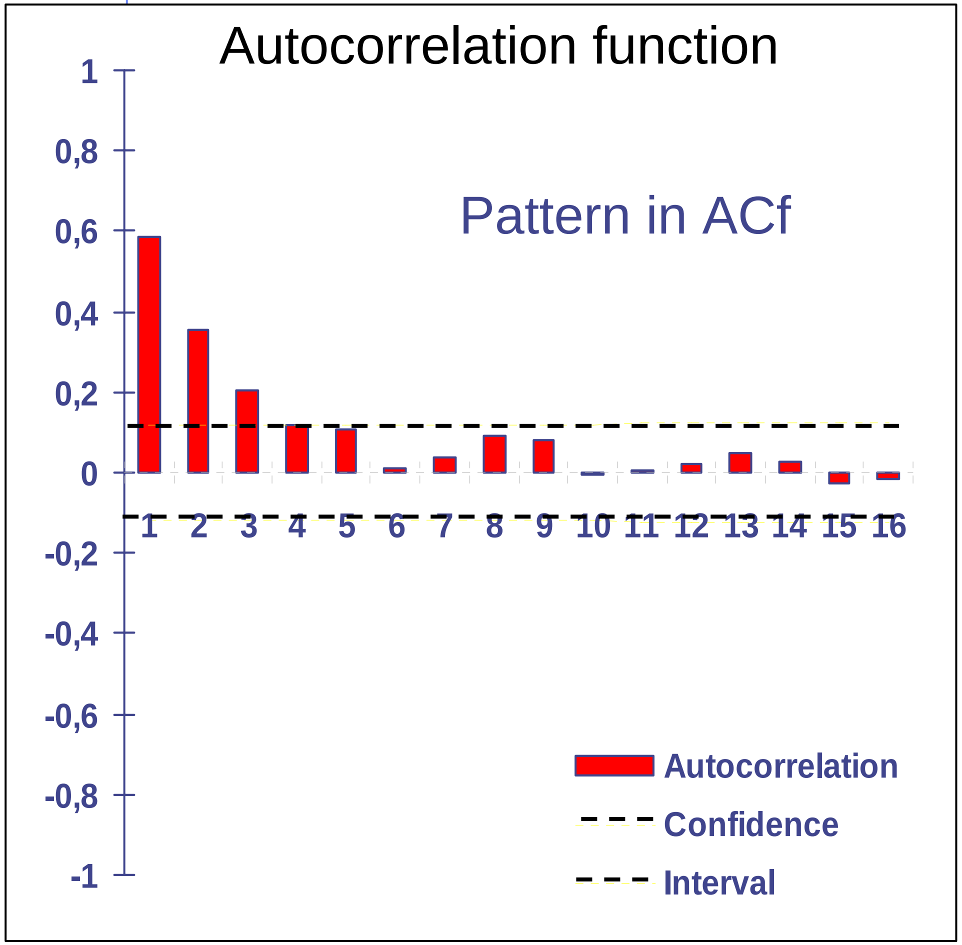
- When fitting an AR(p) model to the time series, the last coefficient p of Y_{t-p} measures the excess correlation at lag p which is not accounted for by an AR(p-1) model. π_p is called the pth order partial autocorrelation, i.e.

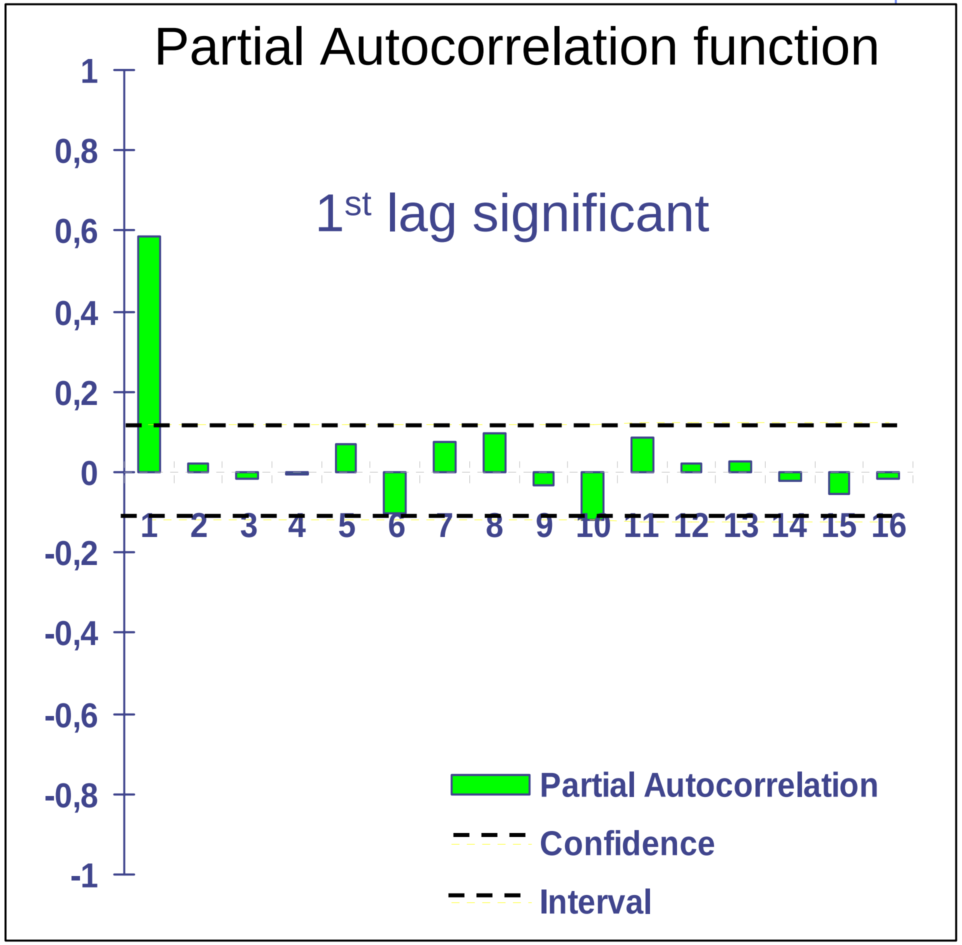
$$\pi_p = \operatorname{corr}\left(Y_t, Y_{t-p} \mid Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p+1}\right)$$

- Partial Autocorrelation coefficient measures true correlation at Y_{t-p}

$$Y_t = \varphi_0 + \varphi_{p1} Y_{t-1} + \varphi_{p2} Y_{t-2} + \ldots + \pi_p Y_{t-p} + \nu_t$$

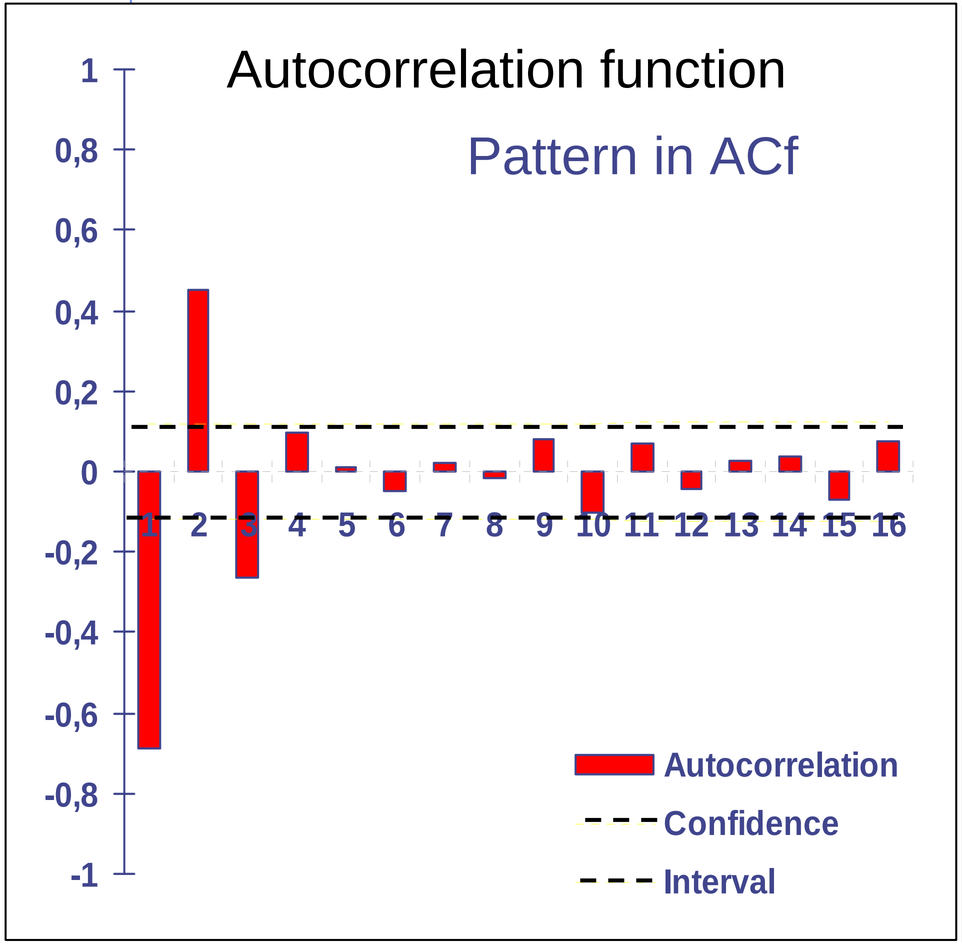
ARIMA Modelling – AR-Model patterns

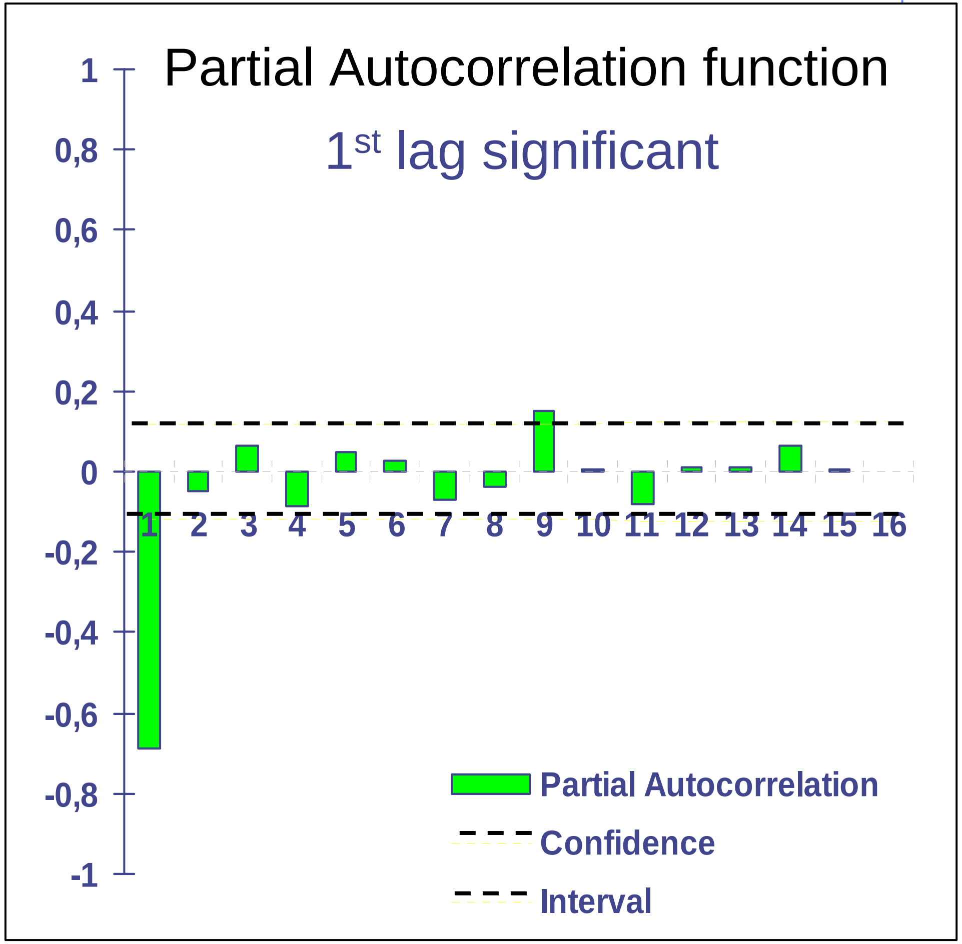




- AR(1) model: $Y_t = c + \phi_1 Y_{t-1} + e_t$ =ARIMA (1,0,0)

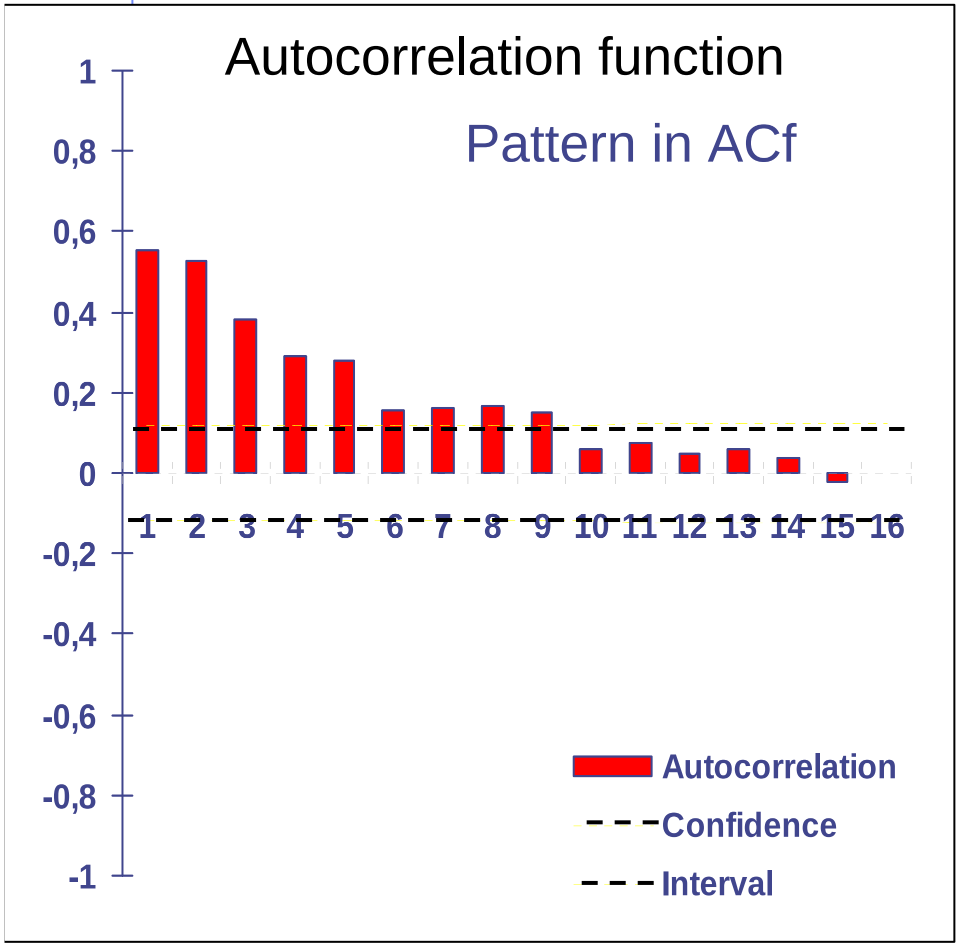
ARIMA Modelling – AR-Model patterns

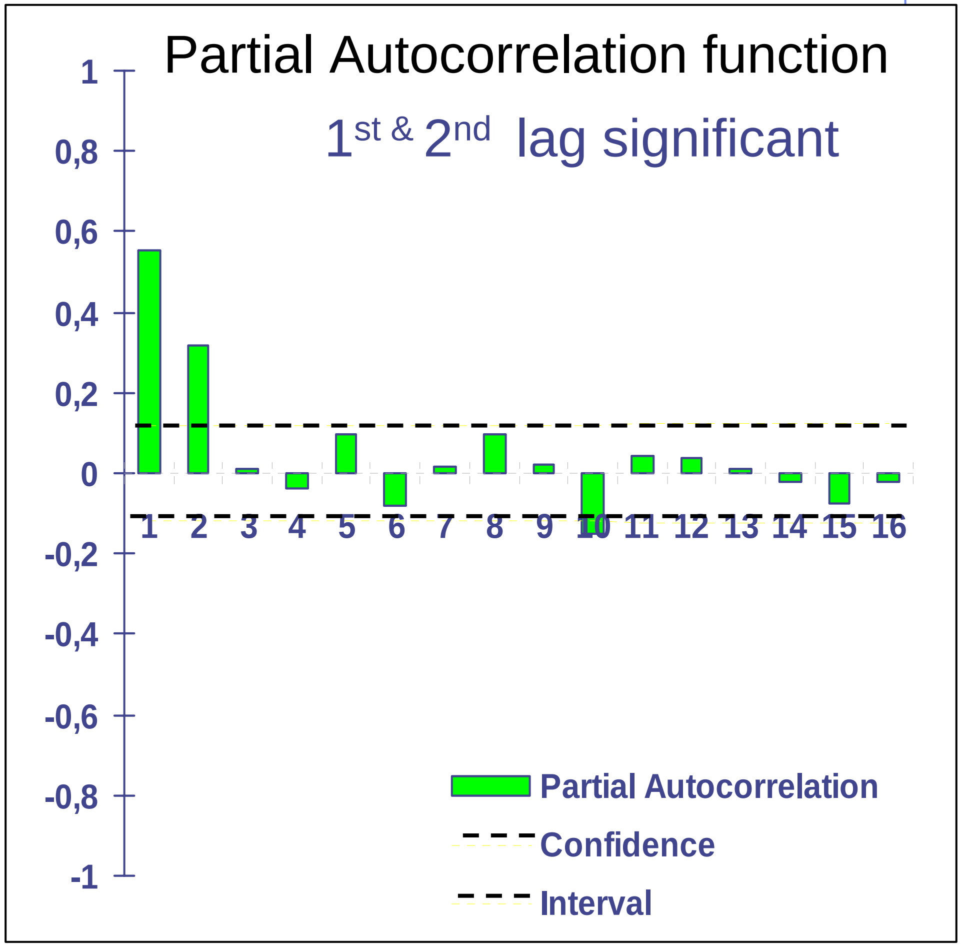




- AR(1) model: $Y_t = c + \phi_1 Y_{t-1} + e_t$ =ARIMA (1,0,0)

ARIMA Modelling – AR-Model patterns

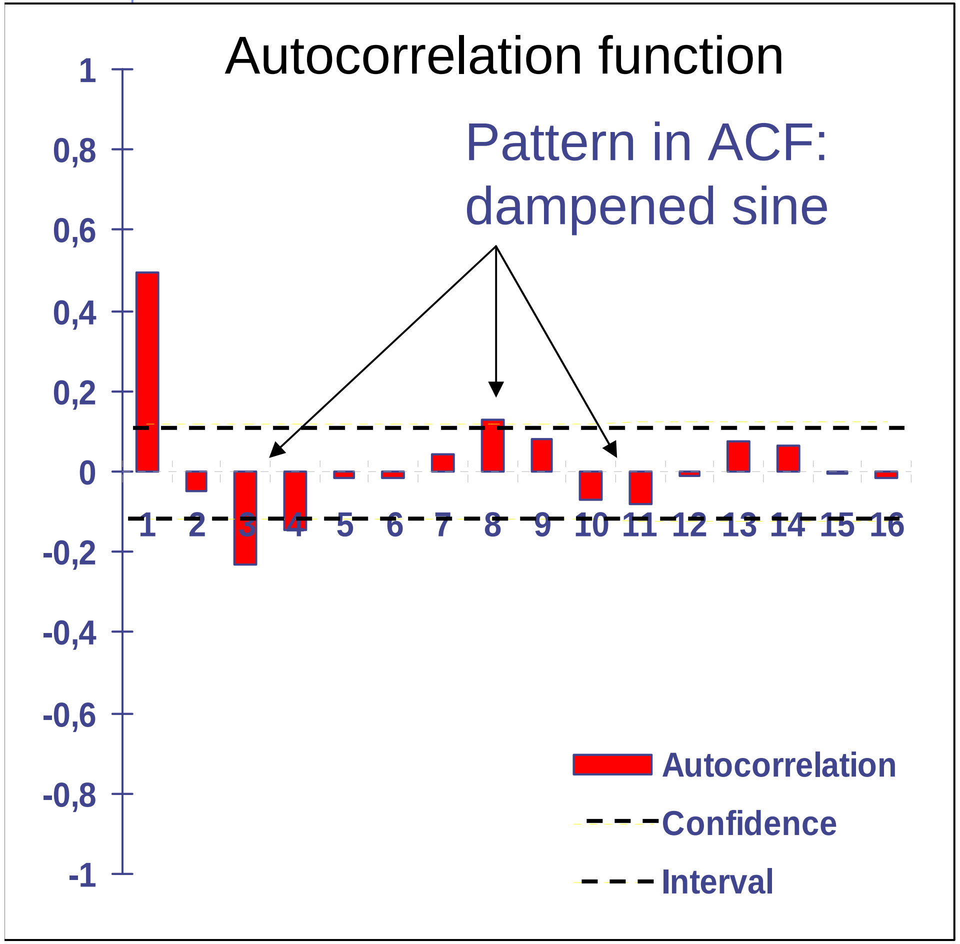


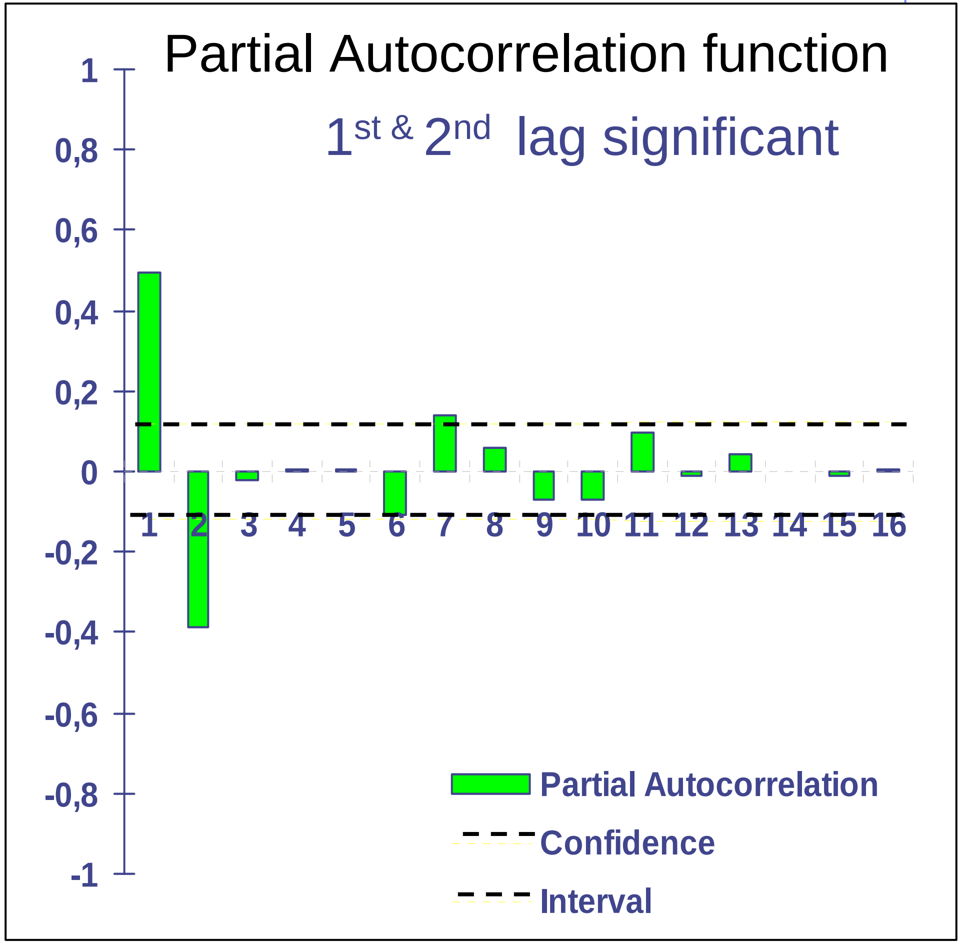


- AR(2) model: =ARIMA (2,0,0)

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$$

ARIMA Modelling – AR-Model patterns



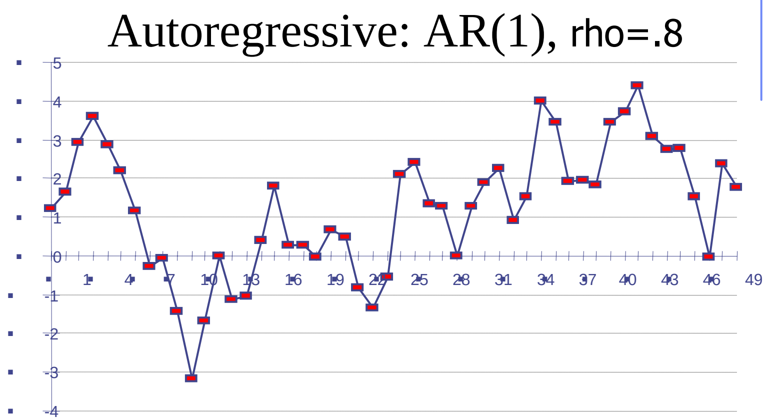


- AR(2) model: =ARIMA (2,0,0) $$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$$

ARIMA Modelling – AR-Models

- Autoregressive Model of order one ARIMA(1,0,0), AR(1)

$$Y_t = c + \phi_1 Y_{t-1} + e_t$$
$$= 1.1 + 0.8 Y_{t-1} + e_t$$



- Higher order AR models

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + e_t$$

$$\text{for } p = 1, -1 < \phi_1 < 1$$
$$p = 2, -1 < \phi_2 < 1 \wedge \phi_2 + \phi_1 < 1 \wedge \phi_2 - \phi_1 < 1$$

Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
    1. Forecasting as predictive Regression
    2. Time series prediction vs. causal prediction
    3. SARIMA-Modelling
        1. SARIMA – Differencing
        2. SARIMA – Autoregressive Terms
        3. SARIMA – Moving Average Terms
        4. SARIMA – Seasonal Terms
    4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

ARIMA Modelling – Moving Average Processe

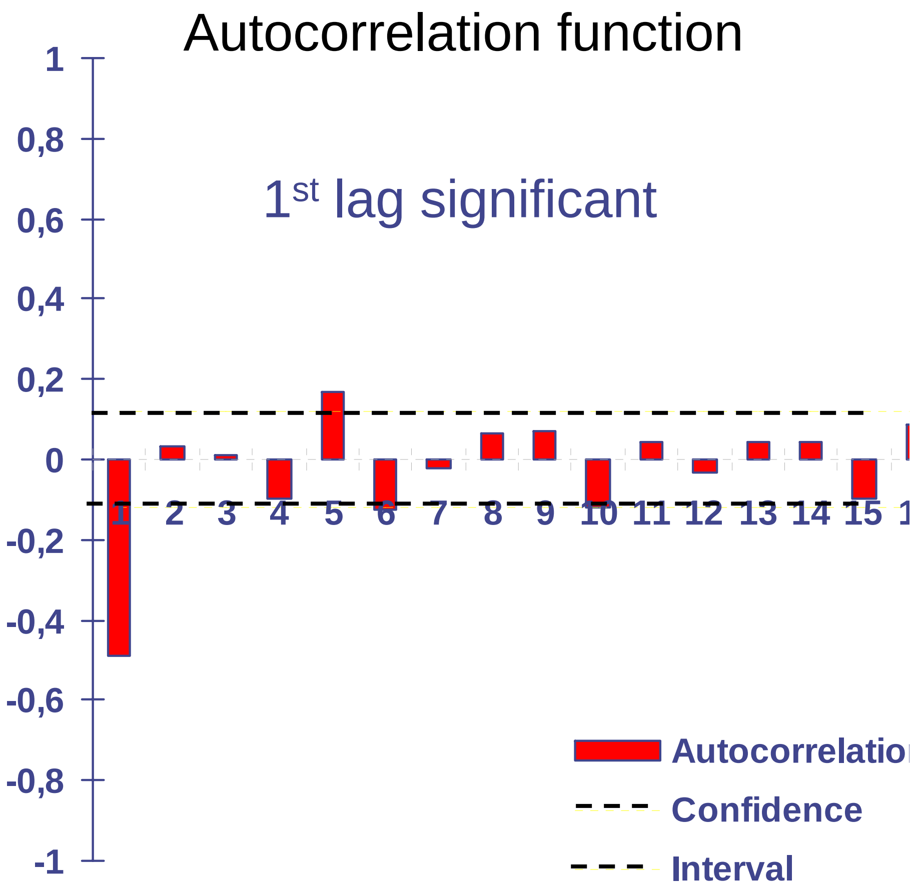
- Description of Moving Average structure
  - AR-Models may not approximate data generator underlying the observations perfectly → residuals e_t, e_{t-1}, e_{t-2}, …, e_{t-q}
  - Observation Y_t may depend on realisation of previous errors e
  - Regress against past errors as explanatory variables

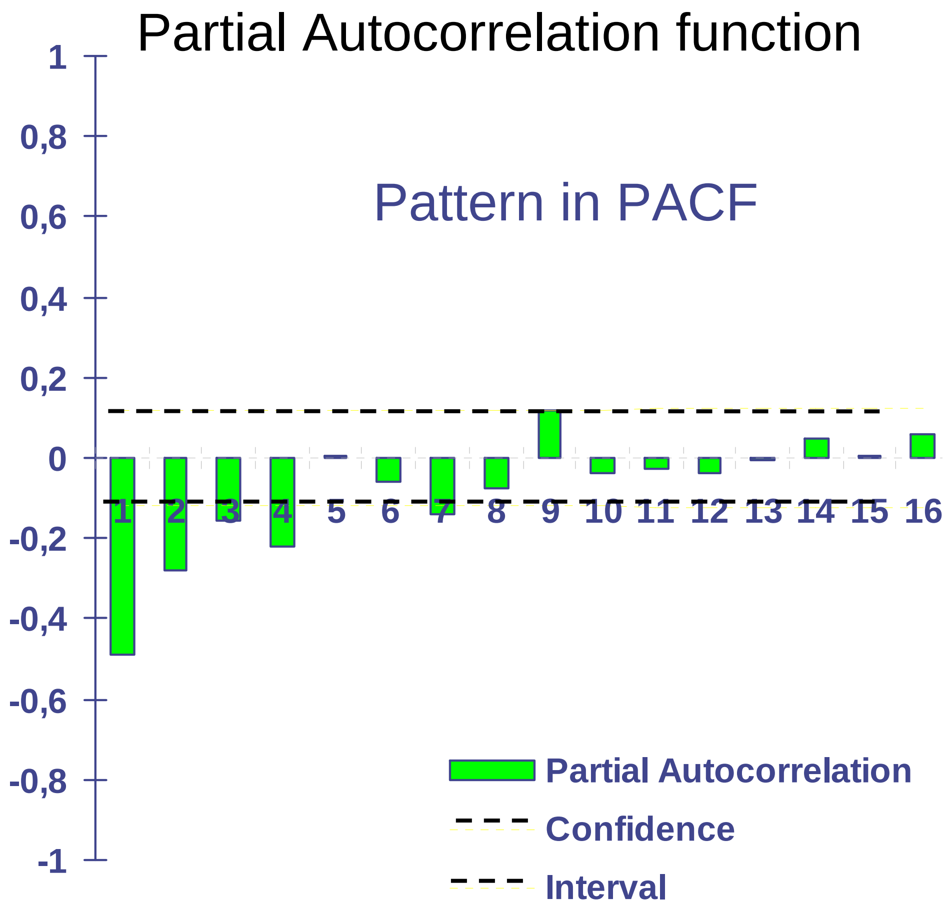
$$Y_t = c + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \ldots - \phi_q e_{t-q}$$

  - ARIMA(0,0,q)-model = MA(q)-model

$$\text{for } q = 1, -1 < \theta_1 < 1$$
$$q = 2, -1 < \theta_2 < 1 \wedge \theta_2 + \theta_1 < 1 \wedge \theta_2 - \theta_1 < 1$$

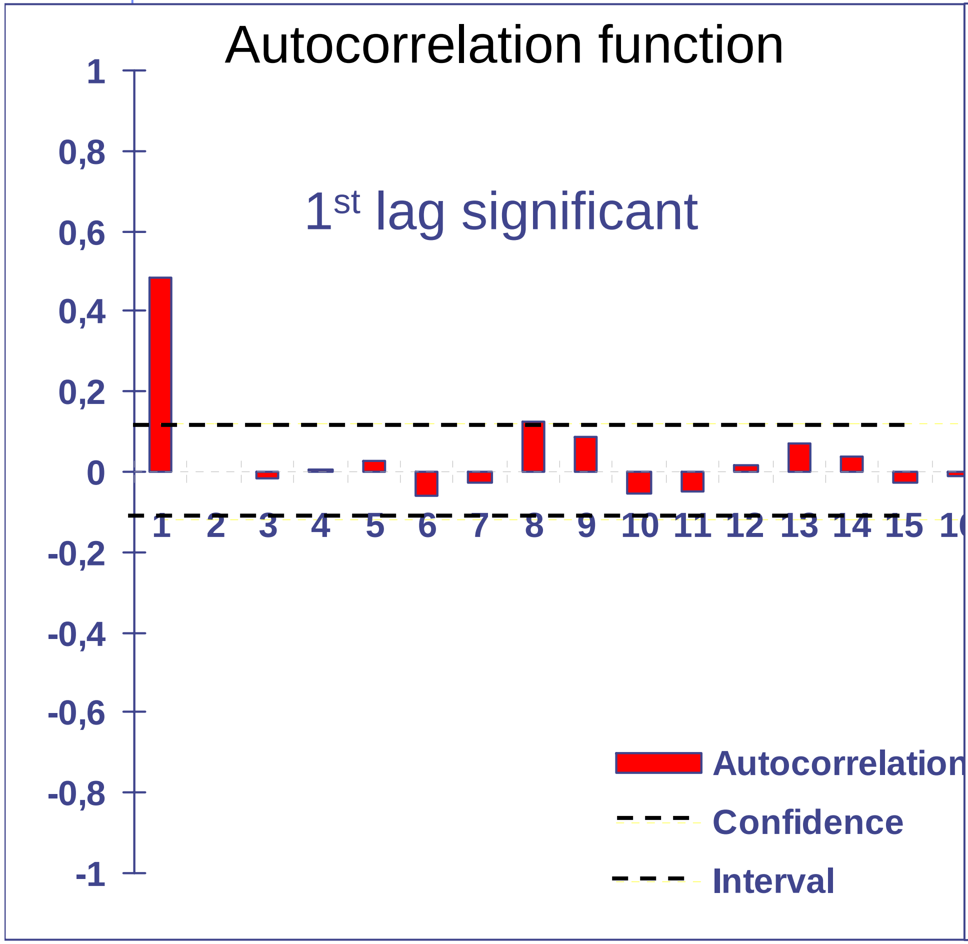
ARIMA Modelling – MA-Model patterns

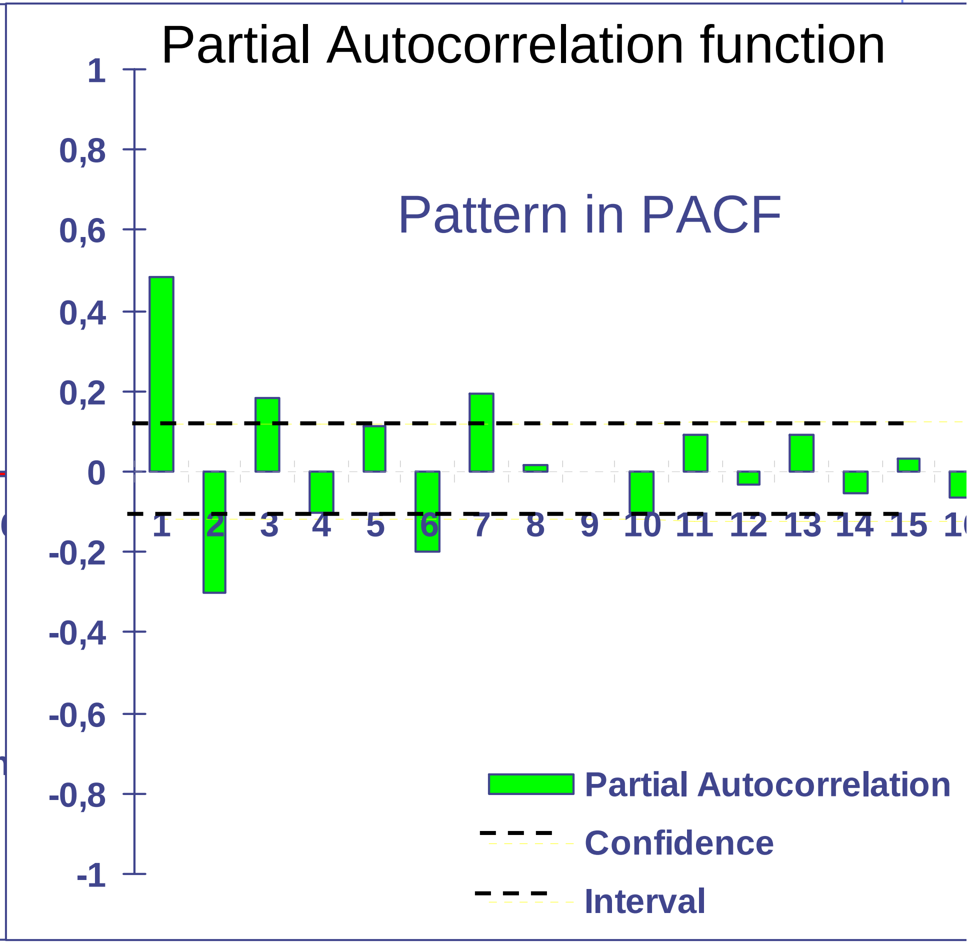




- MA(1) model: $Y_t = c + e_t - \theta_1 e_{t-1}$ =ARIMA (0,0,1)

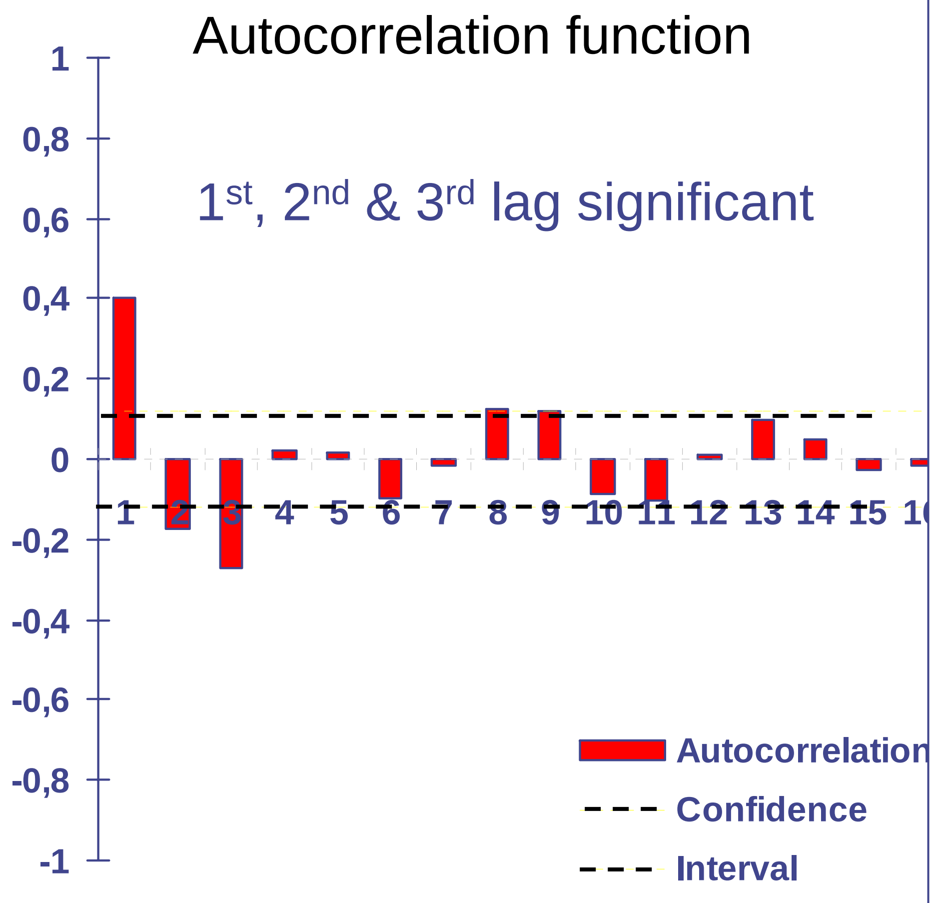
ARIMA Modelling – MA-Model patterns

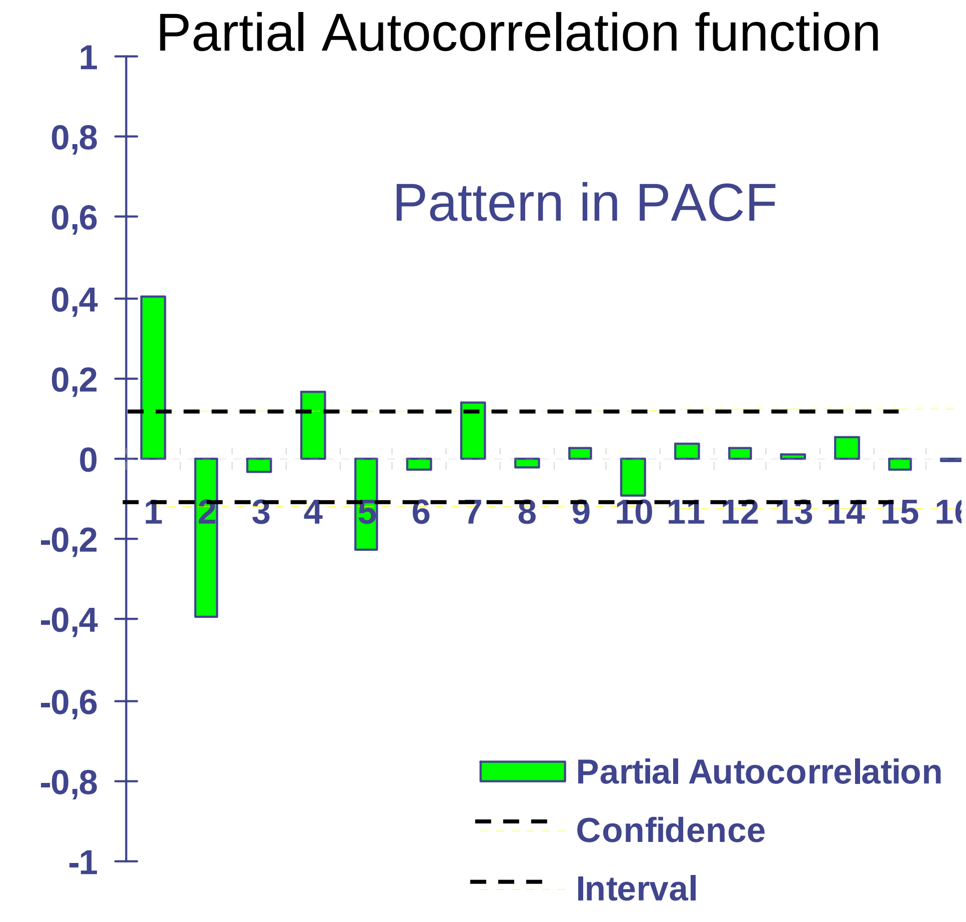




- MA(1) model: $Y_t = c + e_t - \theta_1 e_{t-1}$ =ARIMA (0,0,1)

ARIMA Modelling – MA-Model patterns





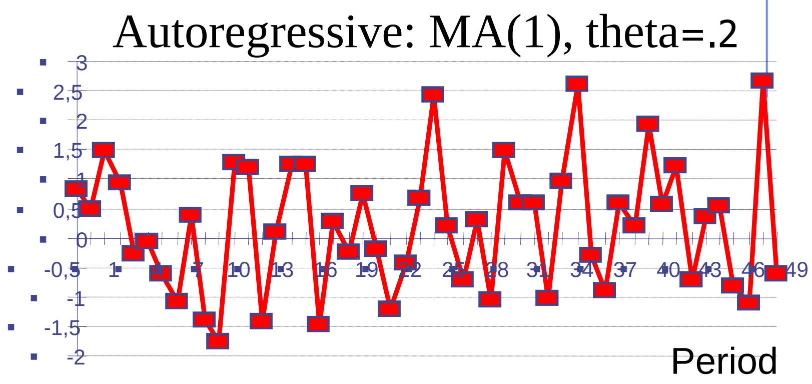
- MA(3) model: =ARIMA (0,0,3)

$$Y_t = c + e_t - \theta_1 e_{t-1} - \theta_1 e_{t-2} - \theta_1 e_{t-3}$$

ARIMA Modelling – MA-Models

- Autoregressive Model of order one ARIMA(0,0,1)=MA(1)

$$Y_t = c + e_t - \theta_1 e_{t-1}$$
$$= 10 + e_t + 0.2 e_{t-1}$$



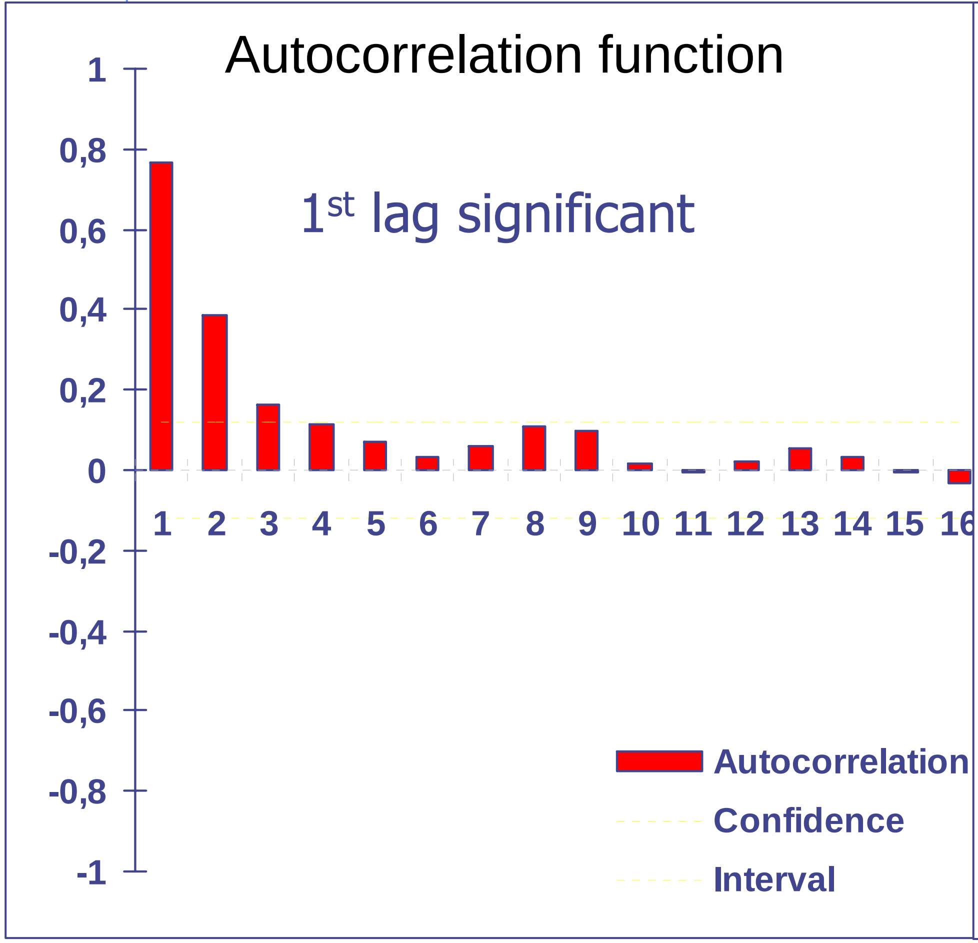
ARIMA Modelling – Mixture ARMA-Models

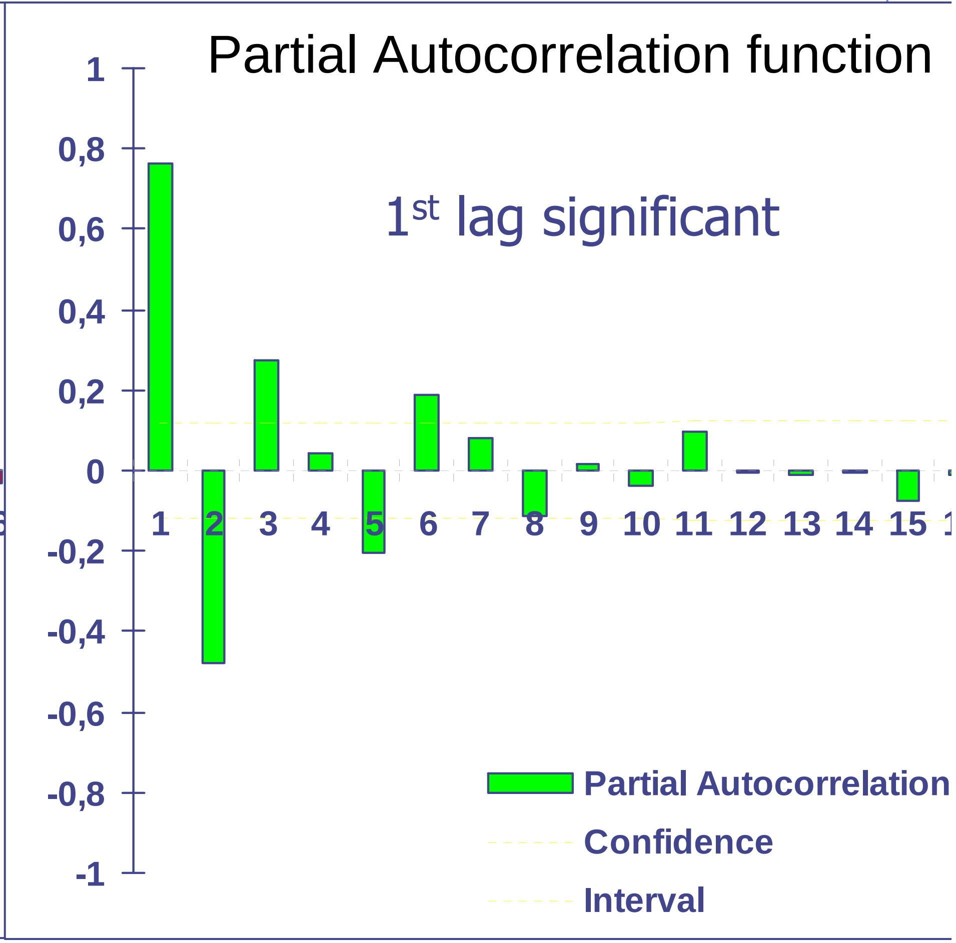
- complicated series may be modelled by combining AR & MA terms
  - ARMA(1,1)-Model = ARIMA(1,0,1)-Model

$$Y_t = c + \phi_1 Y_{t-1} + e_t - \theta_1 e_{t-1}$$

  - Higher order ARMA(p,q)-Models

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + e_t$$
$$-\theta_1 e_{t-1} - \theta_2 e_{t-2} - \ldots - \phi_q e_{t-q}$$





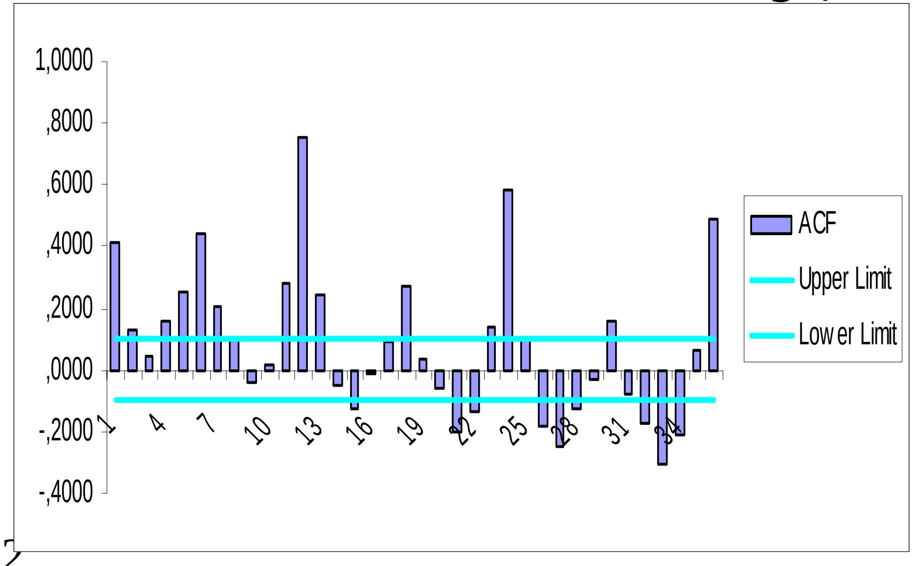
- AR(1) and MA(1) model:
  =ARMA(1,1)=ARIMA (1,0,1)

Agenda

1. Forecasting?
    1. Forecasting as predictive Regression
    2. Time series prediction vs. causal prediction
    3. SARIMA-Modelling
        1. SARIMA – Differencing
        2. SARIMA – Autoregressive Terms
        3. SARIMA – Moving Average Terms
        4. SARIMA – Seasonal Terms
        5. SARIMAX – Seasonal ARIMA with Interventions
    4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Seasonality in ARIMA-Models

- Identifying seasonal data:Spikes in ACF / PACF at seasonal lags, e.g
  - $t\text{-}12$ & $t\text{-}13$ for yearly
  - $t\text{-}4$ & $t\text{-}5$ for quarterly

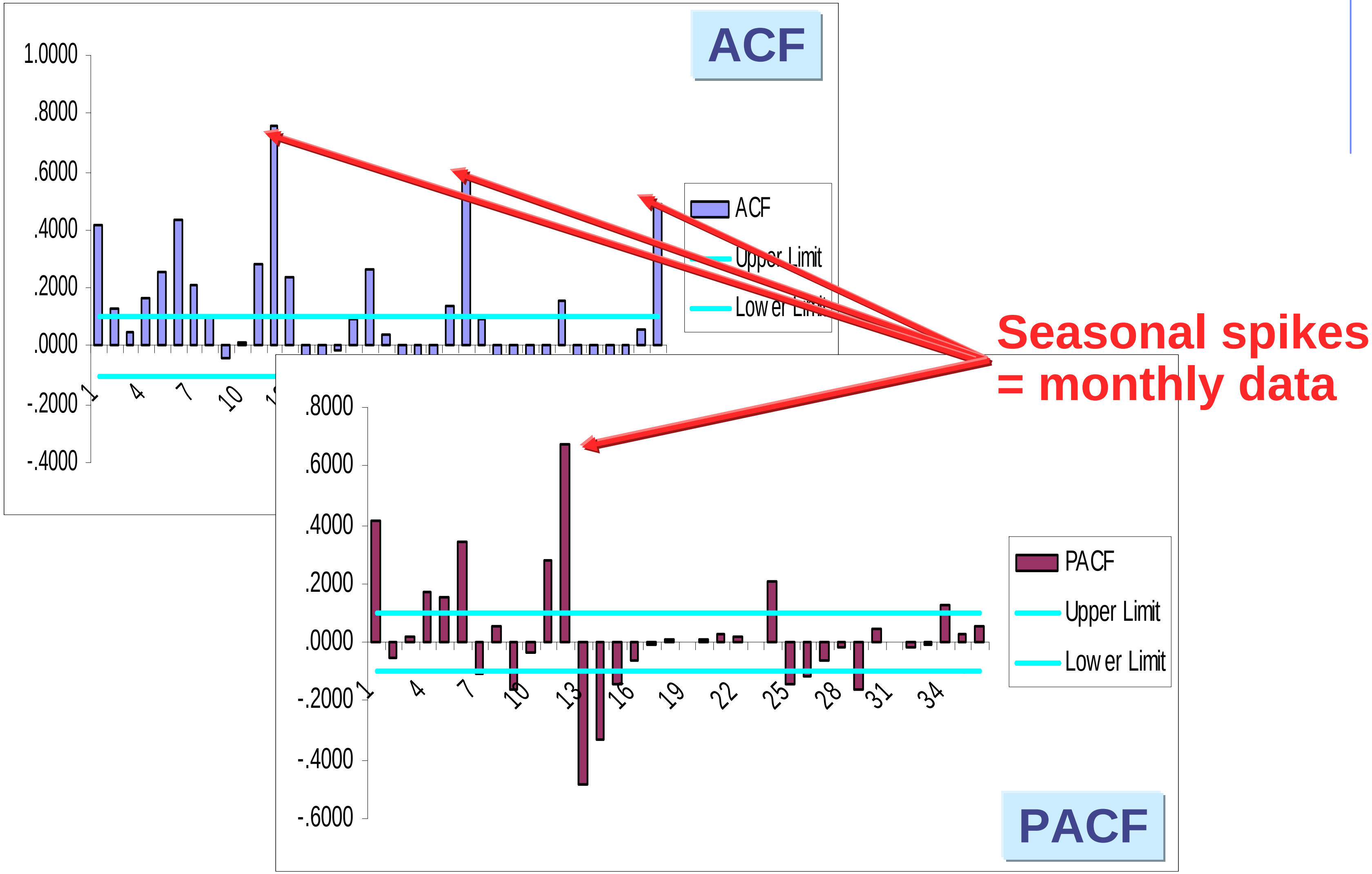


- Differences
  - Simple: $\Delta Y_t=(1-B)Y_t$
  - Seasonal: $\Delta^s Y_t=(1-B^s)Y_t$
    with s= seasonality, eg. 4, 12

- Data may require seasonal differencing to remove seasonality
  - To identify model, specify seasonal parameters: (P,D,Q)
    - the seasonal autoregressive parameters P
    - seasonal difference D and
    - seasonal moving average Q

→ Seasonal ARIMA (p,d,q)(P,D,Q)-model

Seasonality in ARIMA-Models



Seasonality in ARIMA-Models

- Extension of Notation of Backshift Operator

$$\Delta^s Y_t = Y_t - Y_{t-s} = Y_t - B^s Y_t = (1-B^s) Y_t$$

- Seasonal difference followed by a first difference: $(1-B)\,(1-B^s)\,Y_t$

$$(1-\phi_1 B)(1-\Phi_1 B^4)(1-B)(1-B^4)Y_t = c + (1-\theta_1 B)(1-\Theta_1 B^4)e_t$$

Non-seasonal AR(1)

Seasonal AR(1)

Non-seasonal difference

Seasonal difference

Non-seasonal MA(1)

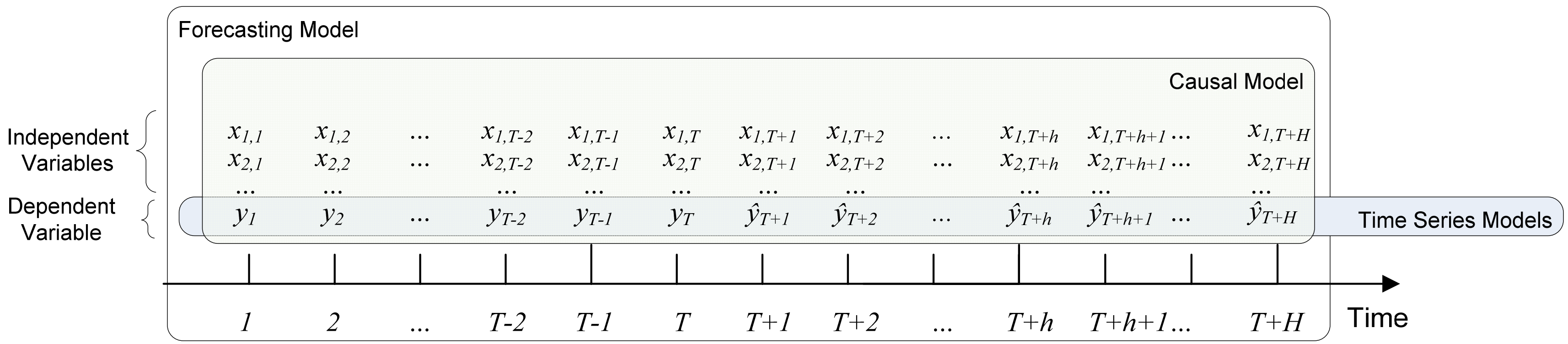
Seasonal MA(1)

Agenda

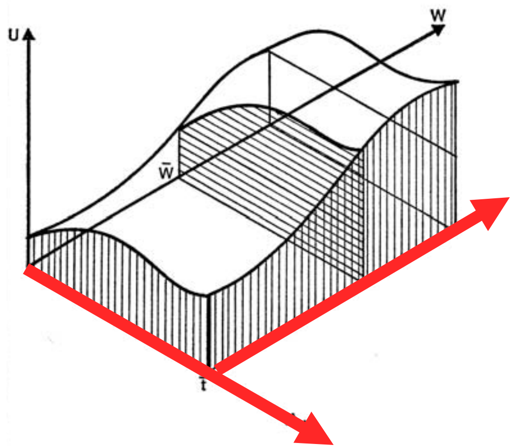
1. Forecasting?
    1. Forecasting as predictive Regression
    2. Time series prediction vs. causal prediction
    3. SARIMA-Modelling
        1. SARIMA – Differencing
        2. SARIMA – Autoregressive Terms
        3. SARIMA – Moving Average Terms
        4. SARIMA – Seasonal Terms
        5. SARIMAX – Seasonal ARIMA with Interventions
    4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Forecasting Models

- Time series analysis vs. causal modelling

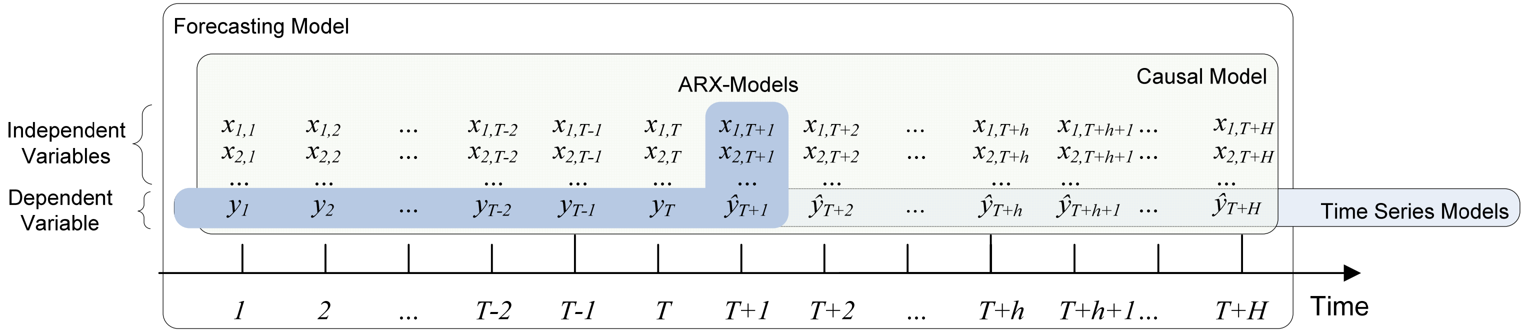


  - Time series prediction (Univariate)
    - Assumes that data generating process that creates patterns can be explained only from previous observations of dependent variable
  - Causal prediction (Multivariate)
    - Data generating process can be explained by interaction of causal (cause-and-effect) independent variables

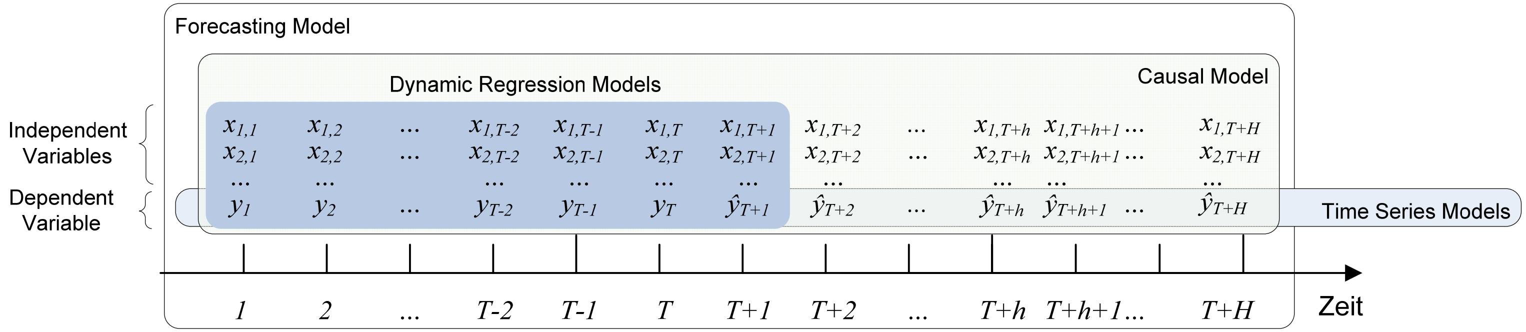


Causal Prediction

- ARX(p)-Models



- General Dynamic Regression Models



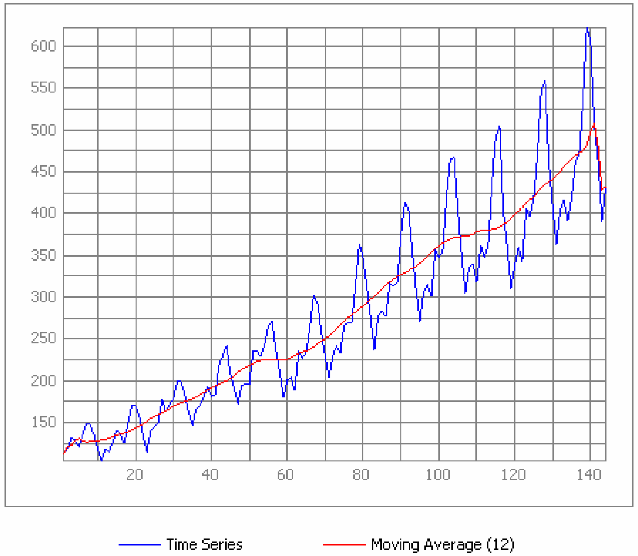
Agenda

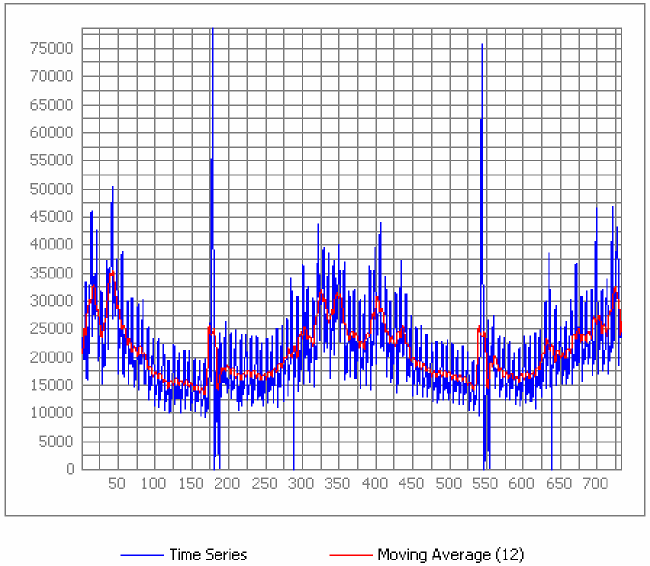
Forecasting with Artificial Neural Networks

1. Forecasting?
   1. Forecasting as predictive Regression
   2. Time series prediction vs. causal prediction
   3. SARIMA-Modelling
   4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Why forecast with NN?

- Pattern or noise?





→Airline Passenger data

→Seasonal, trended

→Real "model" disagreed: multiplicative seasonality or additive seasonality with level shifts?

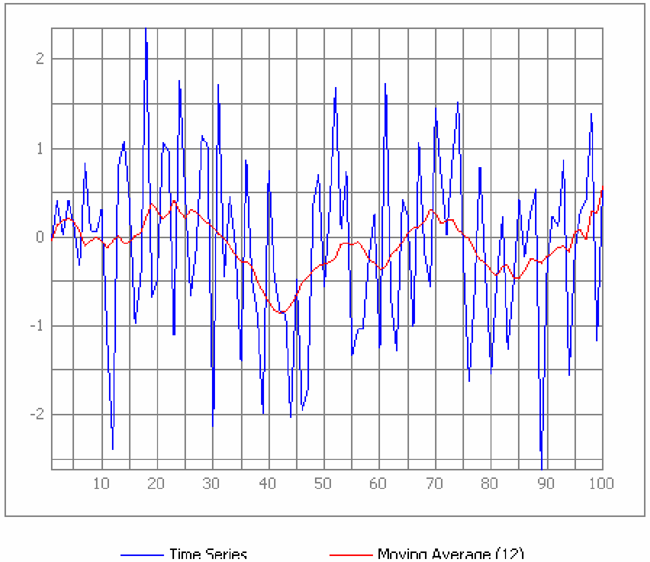
→ Fresh products supermarket Sales

→ Seasonal, events, heteroscedastic noise

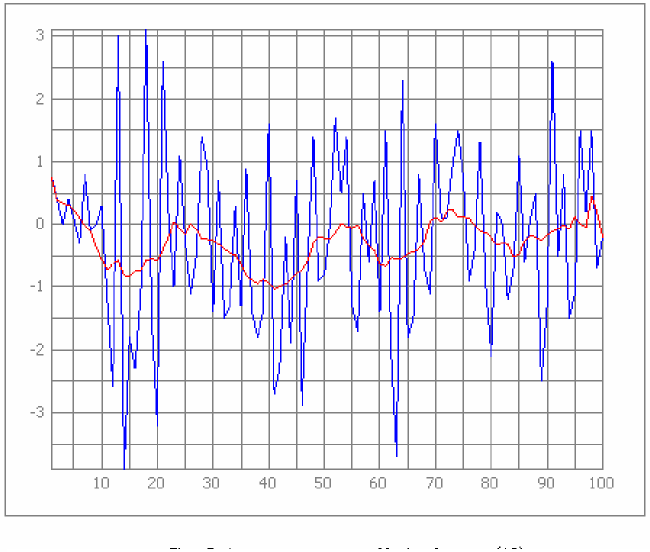
→ Real "model" unknown

Why forecast with NN?

- Pattern or noise?



→ Random Noise iid (normally distributed: mean 0; std.dev. 1)

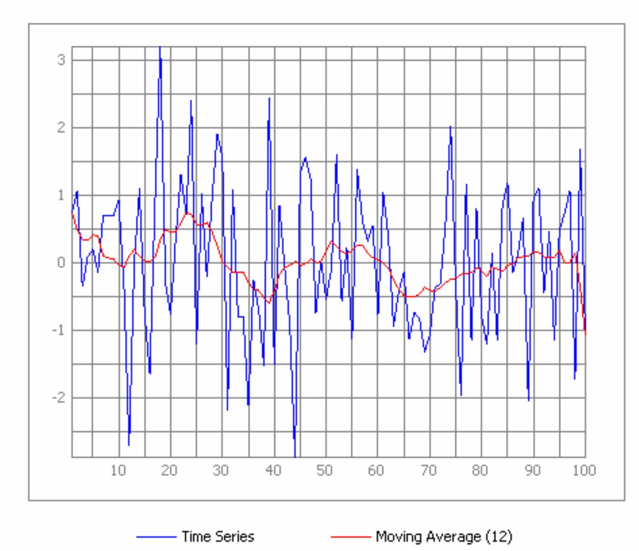


→ BL(p,q) Bilinear Autoregressive Model

$$y_t = 0.7 y_{t-1} \varepsilon_{t-2} + \varepsilon_t$$

Why forecast with NN?

- Pattern or noise?



→ **TAR(p) Threshold Autoregressive model**

$$y_t = 0.9 y_{t-1} + \varepsilon_t \quad \text{for } |y_{t-1}| \le 1$$
$$\quad = -0.3 y_{t-1} - \varepsilon_t \quad \text{for } |y_{t-1}| > 1$$

→ **Random walk**

$$y_t = y_{t-1} + \varepsilon_t$$

Motivation for NN in Forecasting – Nonlinearity!

→ True data generating process in unknown & hard to identify
→ Many interdependencies in business are nonlinear

- NN can approximate any LINEAR and NONLINEAR function to any desired degree of accuracy
  - Can learn linear time series patterns
  - Can learn nonlinear time series patterns

  → Can extrapolate linear & nonlinear patterns = generalisation!
- NN are nonparametric
  - Don't assume particular noise process, i.e. gaussian
- NN model (learn) linear and nonlinear process directly from data
  - Approximate underlying data generating process

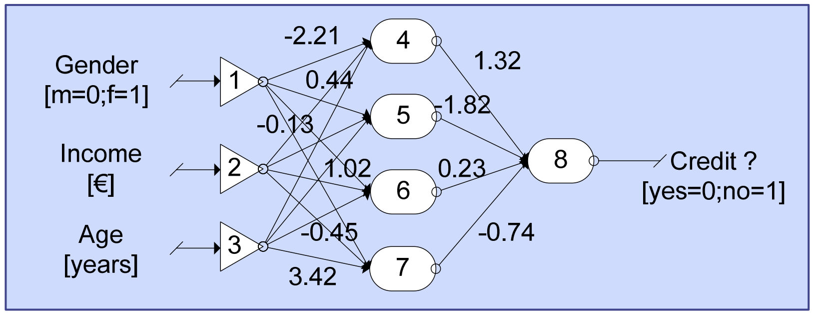
→ **NN are flexible forecasting paradigm**

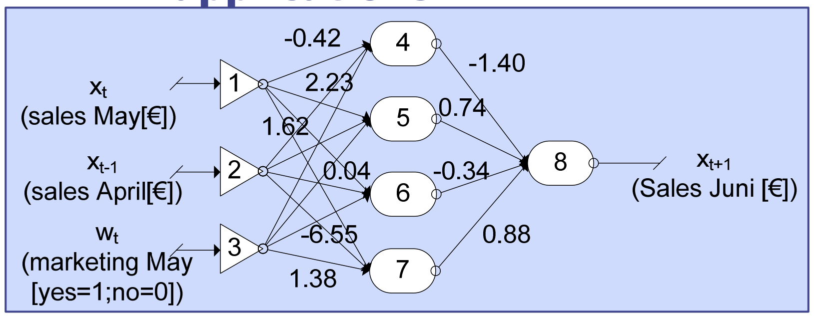
Motivation for NN in Forecasting - Modelling Flexibility

→ Unknown data processes require building of many candidate models!

- Flexibility on Input Variables → flexible coding
  - binary scale [0;1]; [-1,1]
  - nominal / ordinal scale (0,1,2,…,10 → binary coded [0001,0010,…]
  - metric scale (0.235; 7.35; 12440.0; …)
- Flexibility on Output Variables
  - binary → prediction of single class membership
  - nominal / ordinal → prediction of multiple class memberships
  - metric → regression (point predictions) OR probability of class membership!
- Number of Input Variables
  - …
- Number of Output Variables
  - …

→ One SINGLE network architecture → MANY applications



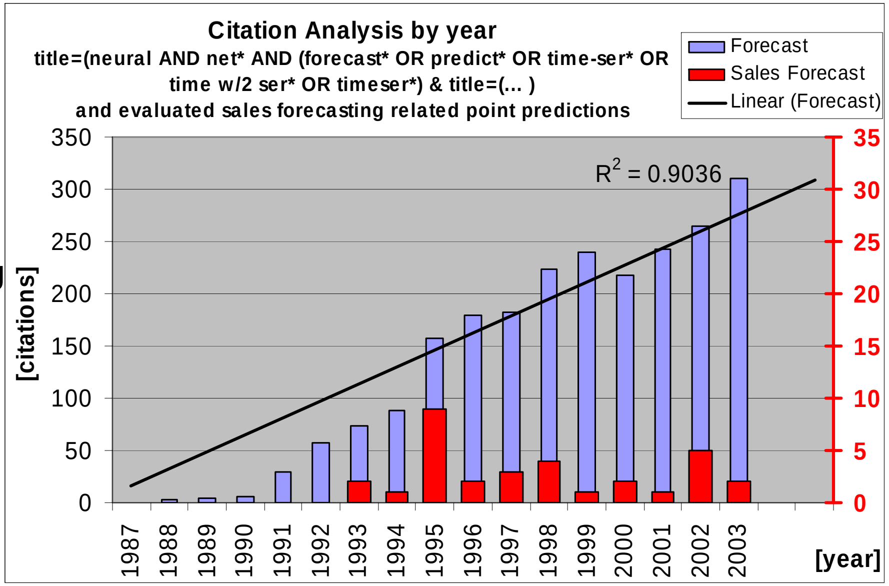


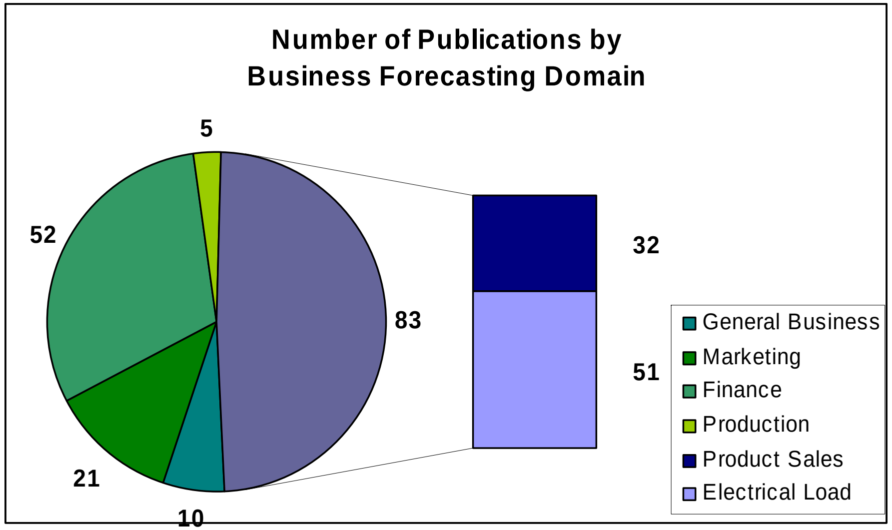
Applications of Neural Nets in diverse Research Fields
→ 2500+ journal publications on NN & Forecasting alone!

- **Neurophysiology**
  → simulate & explain brain
- **Informatics**
  → eMail & url filtering
  → VirusScan (Symmantec Norton Antivirus)
  → Speech Recognition & Optical Character Recog
- **Engineering**
  → control applications in plants
  → automatic target recognition (DARPA)
  → explosive detection at airports
  → Mineral Identification (NASA Mars Explorer)
  → starting & landing of Jumbo Jets (NASA)
- **Meteorology / weather**
  → Rainfall prediction
  → ElNino Effects
- **Corporate Business**
  → credit card fraud detection
  → simulate forecasting methods

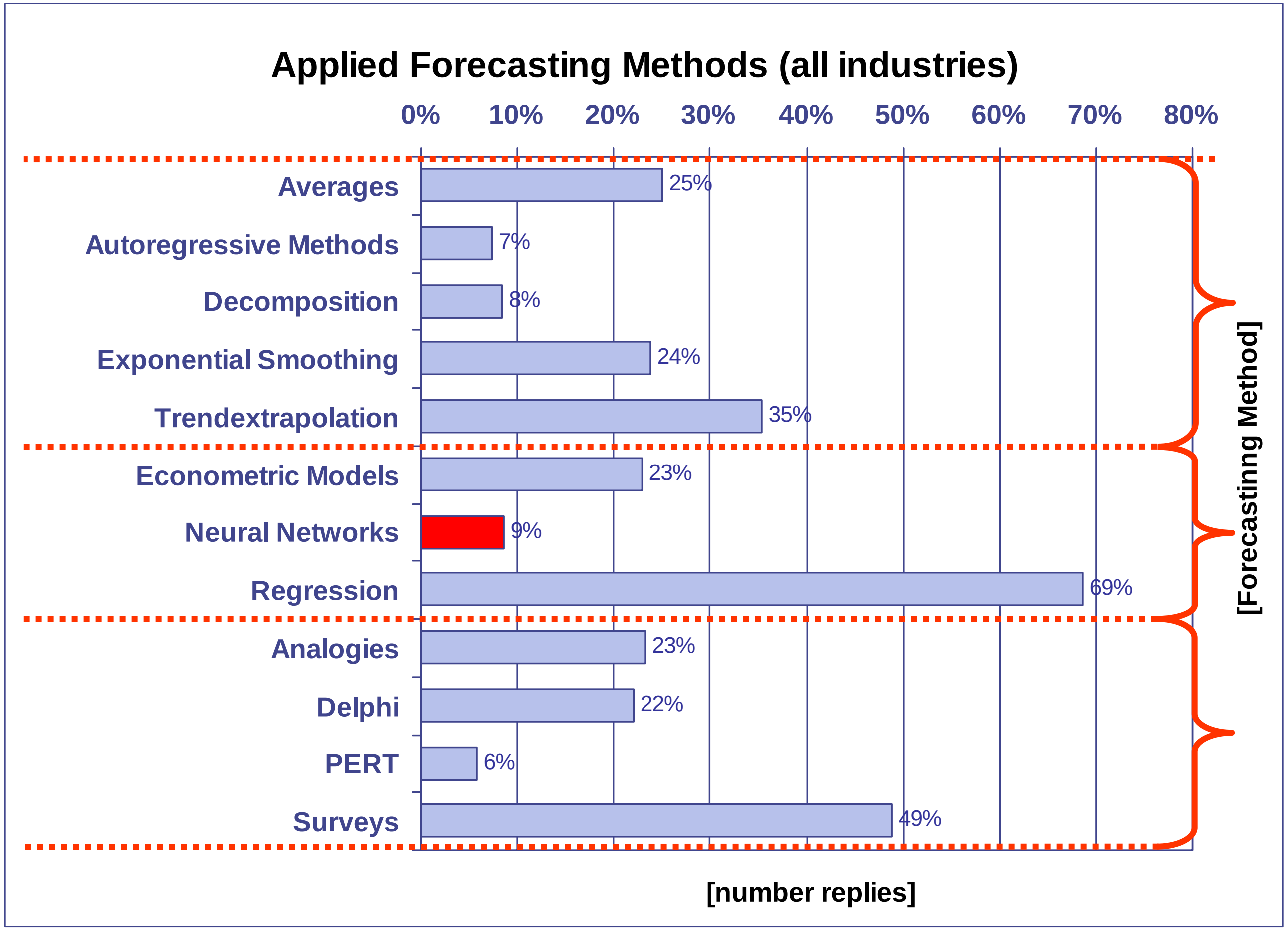
- **Business Forecasting Domains**
  - Electrical Load / Demand
  - Financial Forecasting
    - Currency / Exchange rate
    - stock forecasting etc.
  - Sales forecasting

→ not all NN recommendations are useful for your DOMAIN!





IBF Benchmark– Forecasting Methods used



Time Series methods (objective) → 61%

Causal Methods (objective) → 23%

Judgemental Methods (subjective) → 2x%

→ Survey 5 IBF conferences in 2001
- 240 forecasters, 13 industries

→NN are applied in coroporate Demand Planning / S&OP processes!

[Warning: limited sample size]

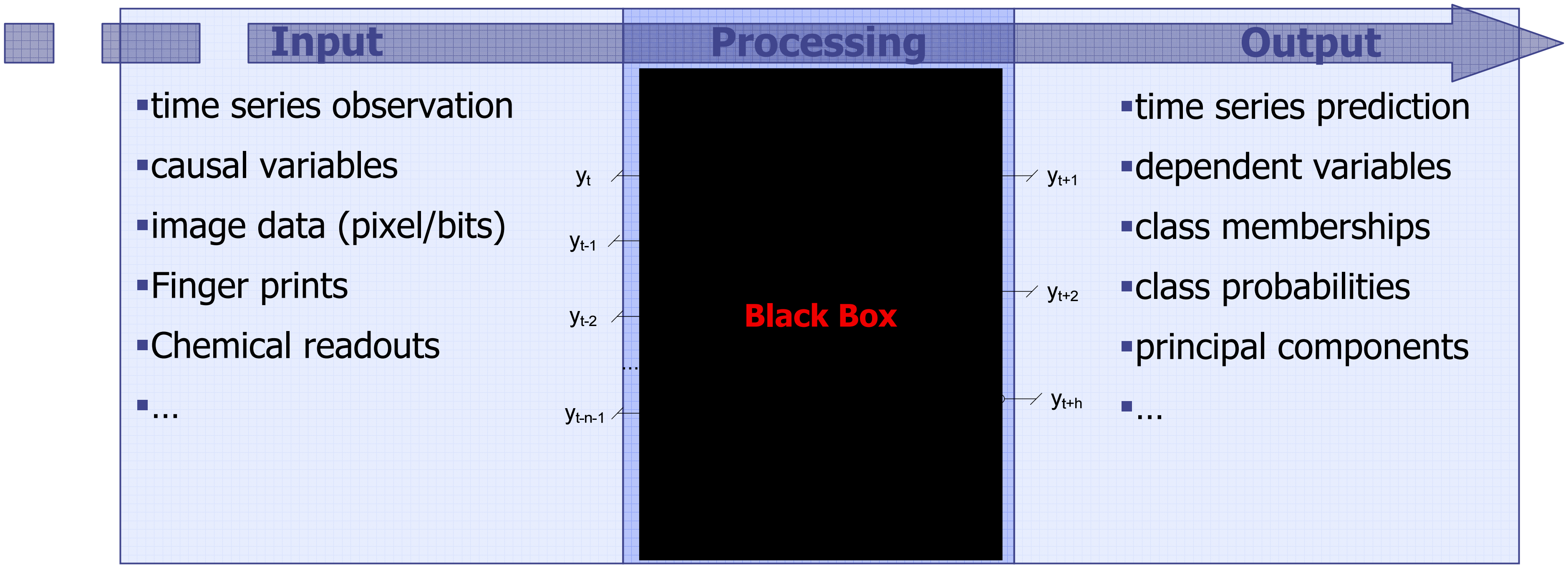
Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
    1. What are NN? Definition & Online Preview …
    2. Motivation & brief history of Neural Networks
    3. From biological to artificial Neural Network Structures
    4. Network Training
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

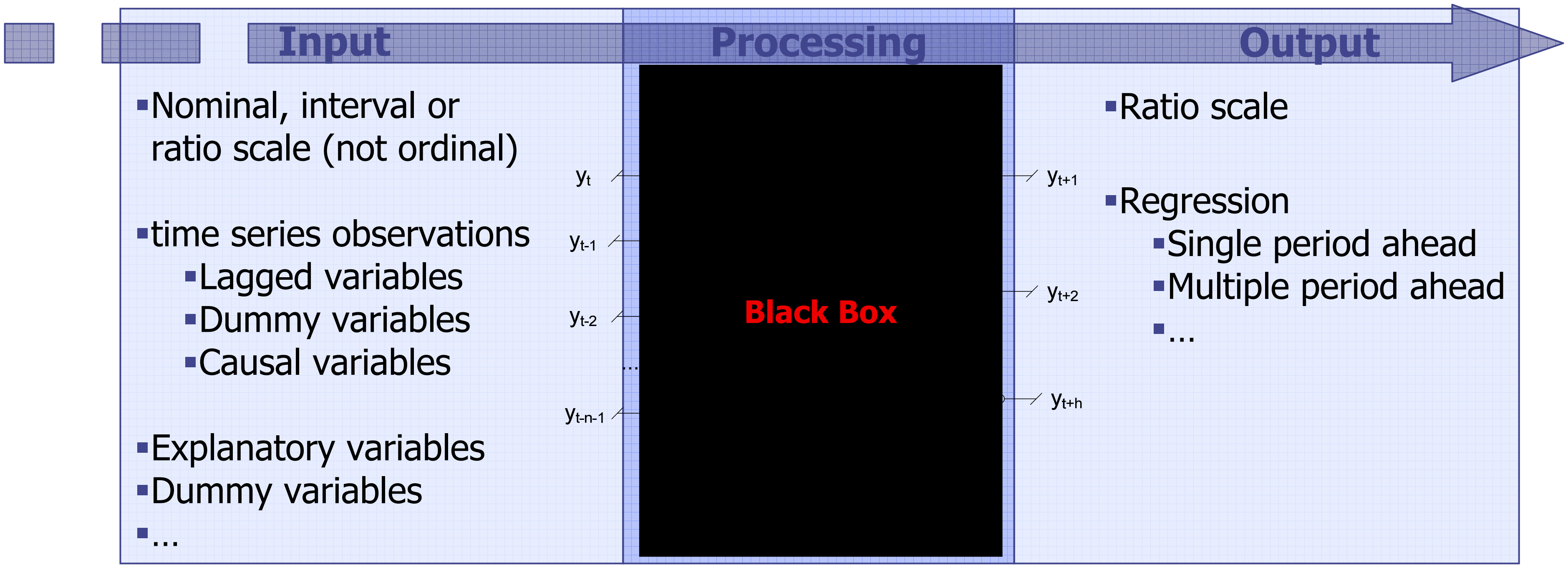
What are Artificial Neural Networks?

- Artificial Neural Networks (NN)
  - „a machine that is designed to *model* the way in which the brain performs a particular task …; the network is … implemented … or .. simulated in software on a digital computer." [Haykin98]
  - class of statistical methods for information processing consisting of large number of simple processing units (neurons), which exchange information of their activation via directed connections. [Zell97]



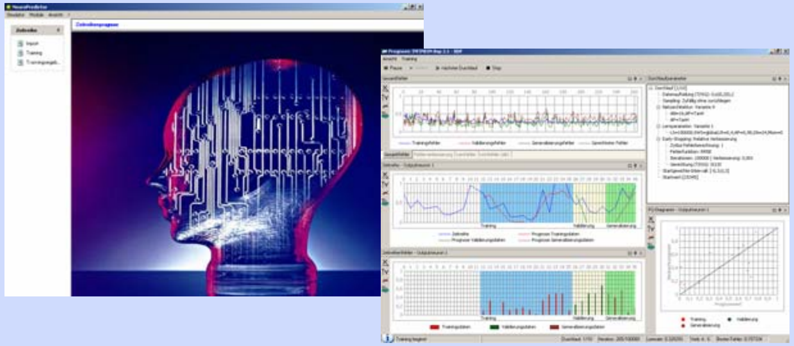
What are Neural Networks in Forecasting?

- Artificial Neural Networks (NN) → a flexible forecasting paradigm
  - A class of statistical methods for time-series and causal forecasting
  - Highly flexible processing → arbitrary input to output relationships
  - Properties → non-linear, nonparametric (assumed), error robust (not outlier!)
  - Data driven modelling → "learning" directly from data

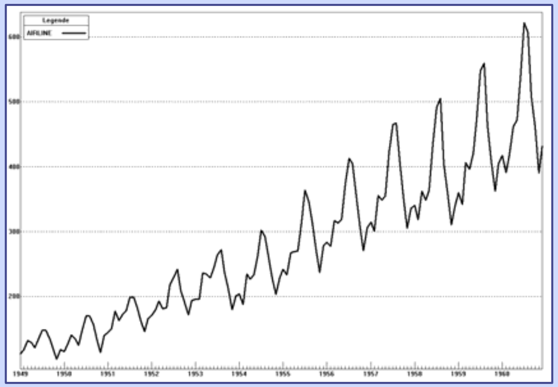


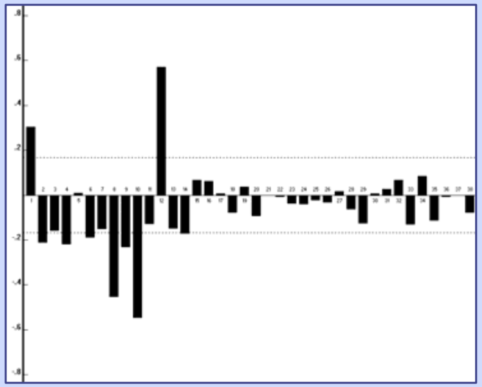
DEMO: Preview of Neural Network Forecasting

- Simulation of NN for Business Forecasting



- Airline Passenger Data Experiment
  - 3 layered NN: (12-8-1) 12 Input units - 8 hidden units – 1 output unit
  - 12 input lags t, t-1, …, t-11 (past 12 observations) → time series prediction
  - t+1 forecast → single step ahead forecast



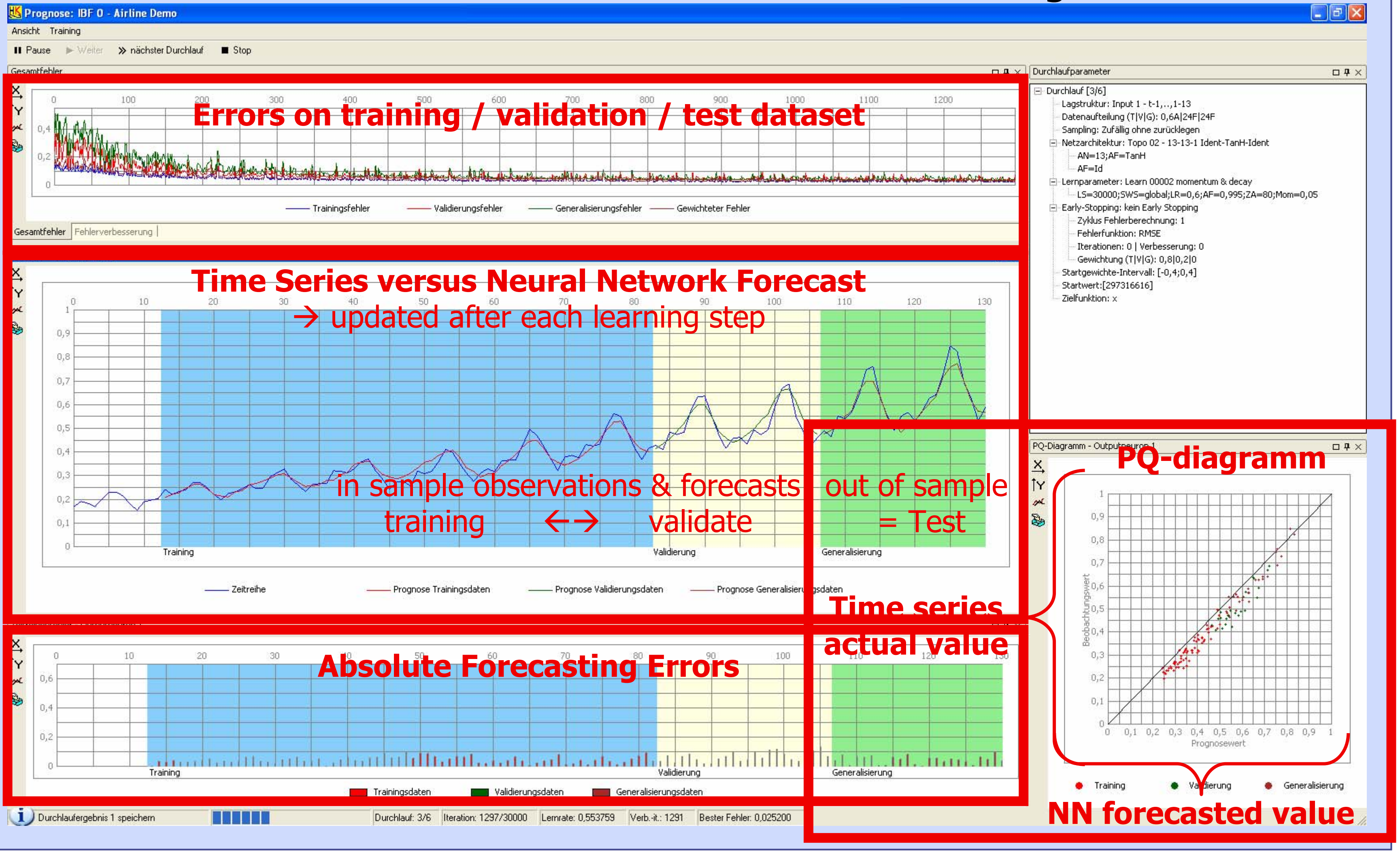


→ Benchmark Time Series
[Brown / Box&Jenkins]

- **132 observations**
- **13 periods of monthly data**

Demonstration: Preview of Neural Network Forecasting

- NeuraLab Predict! → „look inside neural forecasting"



Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
    1. What are NN? Definition & Online Preview …
    2. Motivation & brief history of Neural Networks
    3. From biological to artificial Neural Network Structures
    4. Network Training
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Motivation for using NN ... BIOLOGY!

- Human & other nervous systems (animals, insects → e.g. bats)
  - Ability of various complex functions: perception, motor control, pattern recognition, classification, prediction etc.
  - Speed: e.g. detect & recognize changed face in crowd=100-200ms
  - Efficiency etc.

→ brains are the most efficient & complex computer known to date

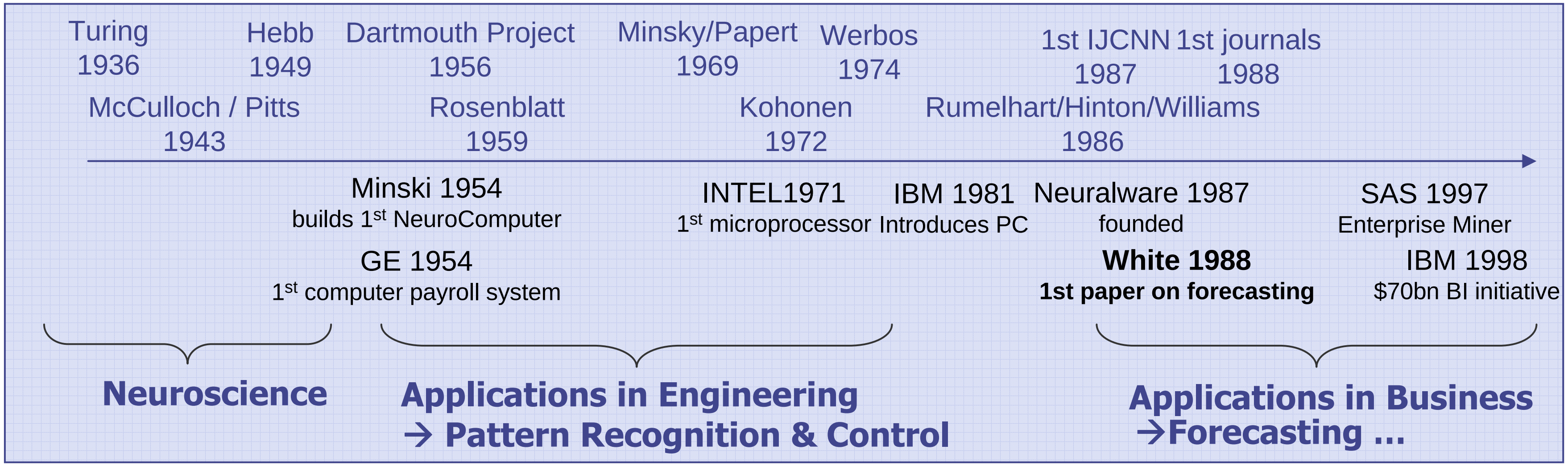
	Human Brain	Computer (PCs)
Processing Speed	10^{-3}ms (0.25 MHz)	10^{-9}ms (2500 MHz PC)
Neurons/Transistors	10 billion & 10^3 billion conn.	50 million (PC chip)
Weight	1500 grams	kilograms to tons!
Energy consumption	10^{-16} Joule	10^{-6} Joule
Computation: Vision	100 steps	billions of steps

→ Comparison: Human = 10.000.000.000 → ant 20.000 neurons

Brief History of Neural Networks

- History
  - Developed in interdisciplinary Research (McCulloch/Pitts1943)
  - Motivation from Functions of natural Neural Networks
    - neurobiological motivation
    - application-oriented motivation

[Smith & Gupta, 2000]



- Research field of Soft-Computing & Artificial Intelligence
  - Neuroscience, Mathematics, Physics, Statistics, Information Science, Engineering, Business Management
  - different VOCABULARY: statistics versus neurophysiology !!!

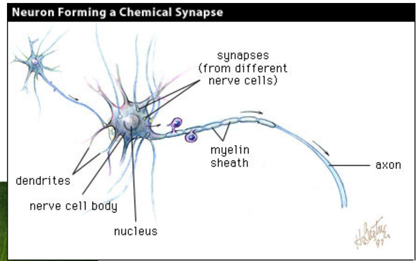
Agenda

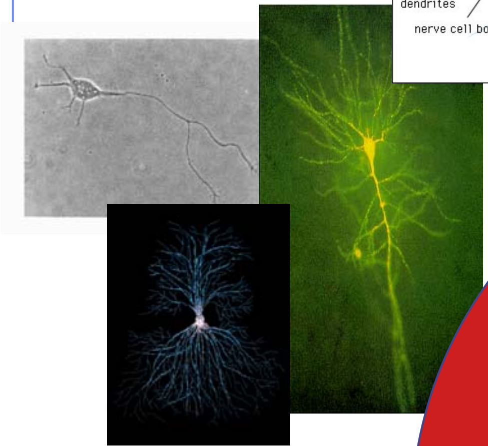
Forecasting with Artificial Neural Networks

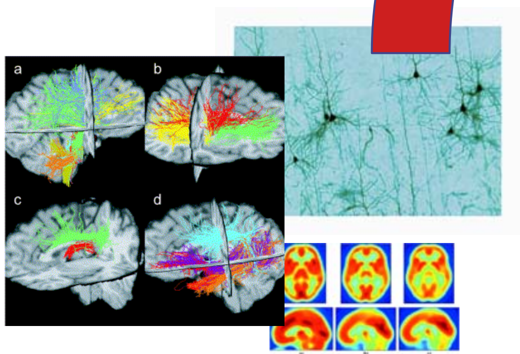
1. Forecasting?
2. Neural Networks?
   1. What are NN? Definition & Online Preview …
   2. Motivation & brief history of Neural Networks
   3. From biological to artificial Neural Network Structures
   4. Network Training
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

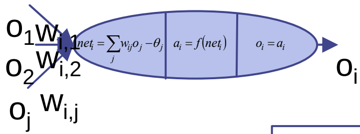
Motivation & Implementation of Neural Networks

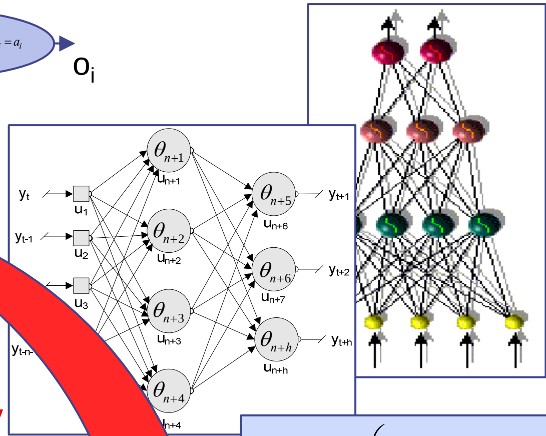
- From biological neural networks … to artificial neural networks









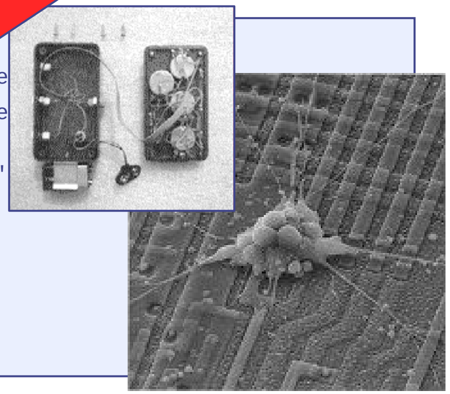


Mathematics as abstract representations of reality

→ use in software simulators, hardware, engineering etc.

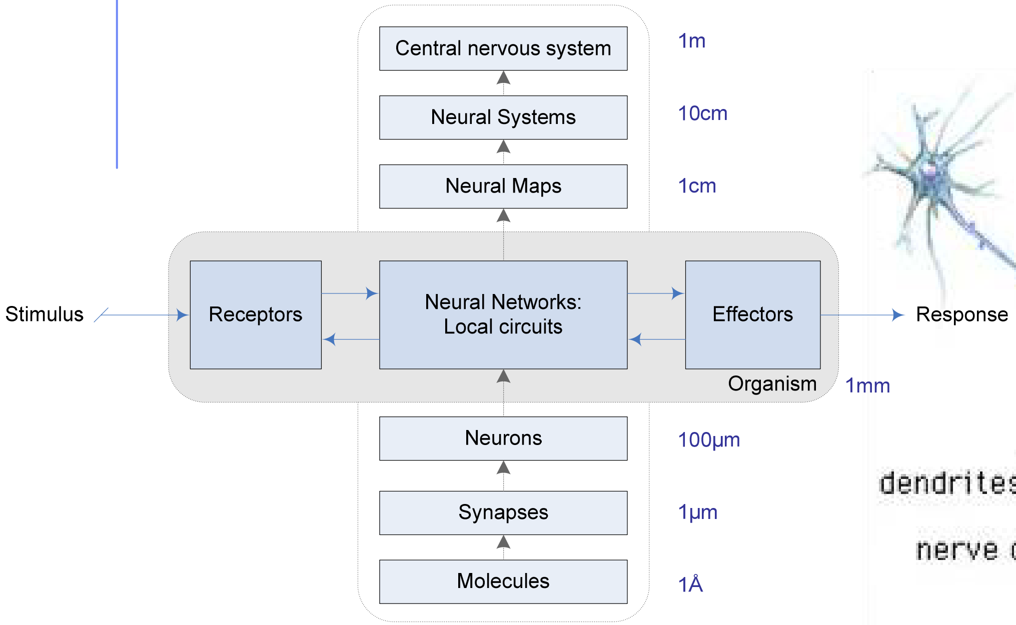
$$o_i = \tanh\left(\sum_j w_{ji} o_j - \theta_i\right)$$

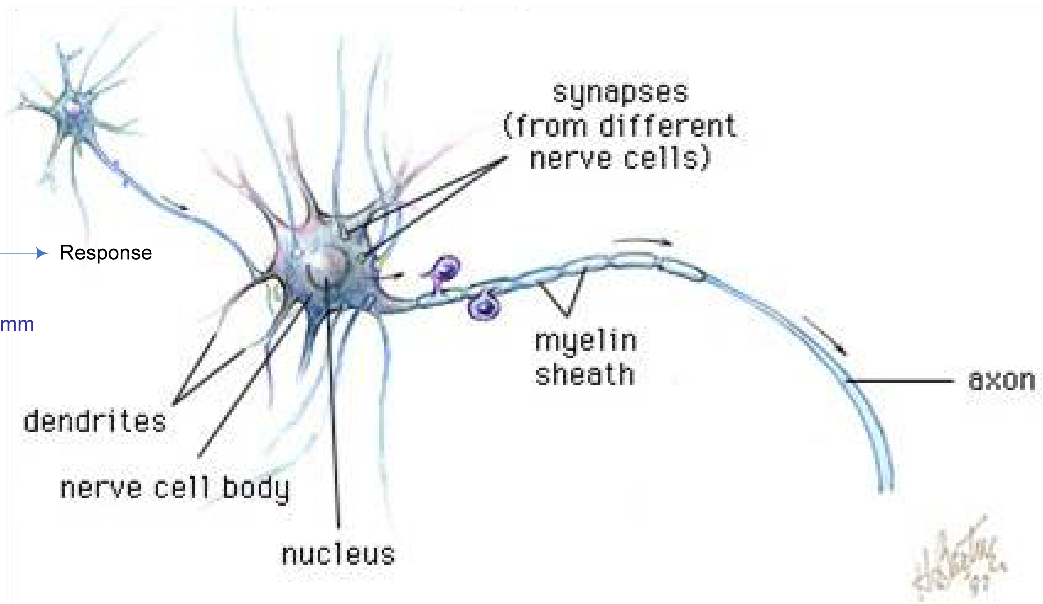
```
neural_net = eval(net_name);
[num_rows, ins] = size(neural_ne
[outs,num_cols] = size(neural_ne
neural_net.numLayers-1});
if (strcmp(neural_net.adaptFcn,'
net_type = 'RBF';
else net_type = 'MLP';
end

fid = fopen(path,'w');
```



Information Processing in biological Neurons

- **Modelling of biological functions in Neurons**
  - 10-100 Billion Neurons with 10000 connections in Brain
  - Input (sensory), Processing (internal) & Output (motoric) Neurons

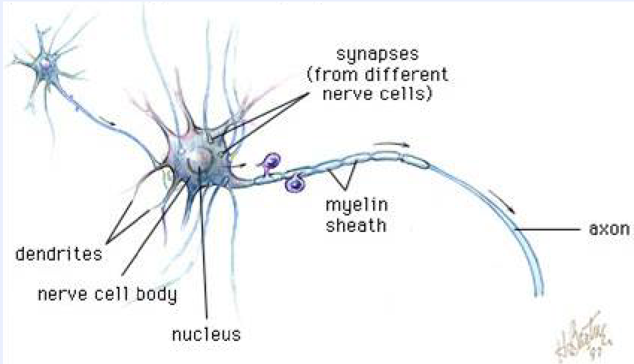




  - **CONCEPT of Information Processing in Neurons …**

Alternative notations – Information processing in neurons / nodes

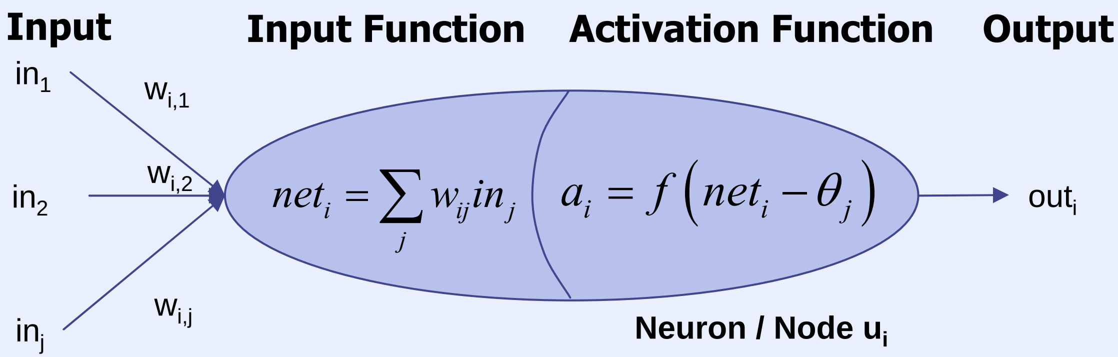
Biological Representation



Graphical Notation

β_i => weights w_i

β_0 => bias θ



Mathematical Representation

$$y_i = \begin{cases} 1 & \text{if} \quad \sum_j w_{ji} x_j - \theta_i \geq 0 \\ 0 & \text{if} \quad \sum_j w_{ji} x_j - \theta_i < 0 \end{cases}$$

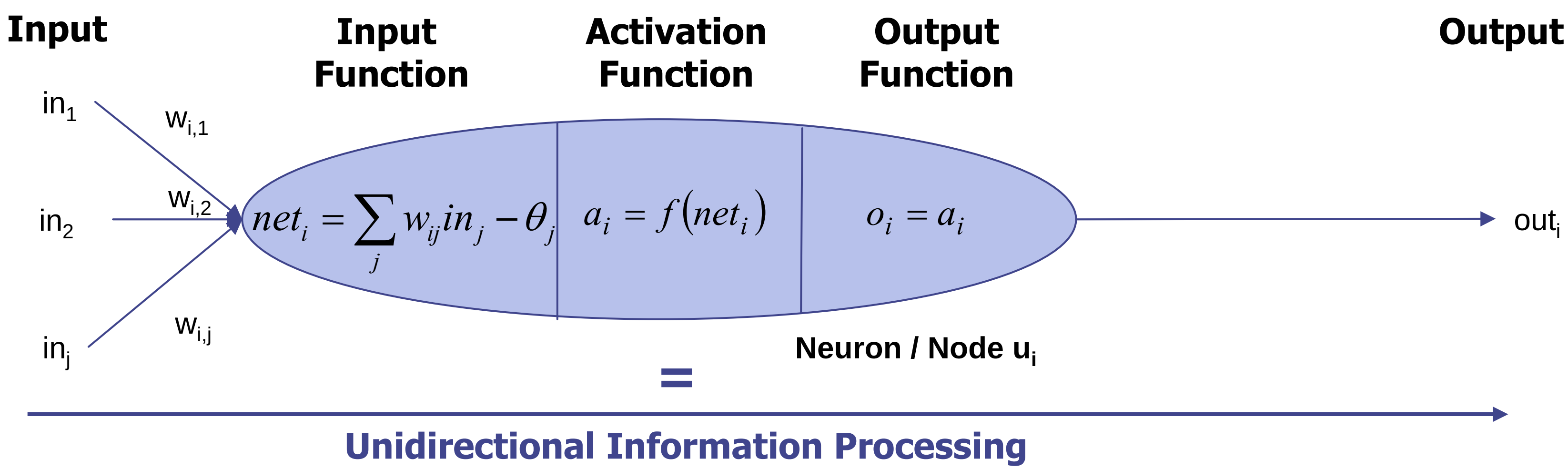
alternative:

$$y_i = \tanh\left(\sum_j w_{ji} x_j - \theta_i\right)$$

...

Information Processing in artificial Nodes

- **CONCEPT of Information Processing in Neurons**
  - Input Function (Summation of previous signals)
  - Activation Function (nonlinear)
    - binary step function {0;1}
    - sigmoid function: logistic, hyperbolic tangent etc.
  - Output Function (linear / Identity, SoftMax ...)



$$out_i = \begin{cases} 1 & \text{if} \quad \sum_j w_{ji} o_j - \theta_i \geq 0 \\ 0 & \text{if} \quad \sum_j w_{ji} o_j - \theta_i < 0 \end{cases}$$

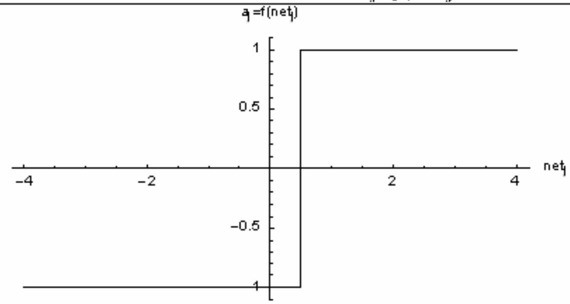
Input Functions

Input Function	Formula
Sum	$net_j = \sum_i o_i w_{ij}$

Binary Activation Functions

- Binary activation calculated from input

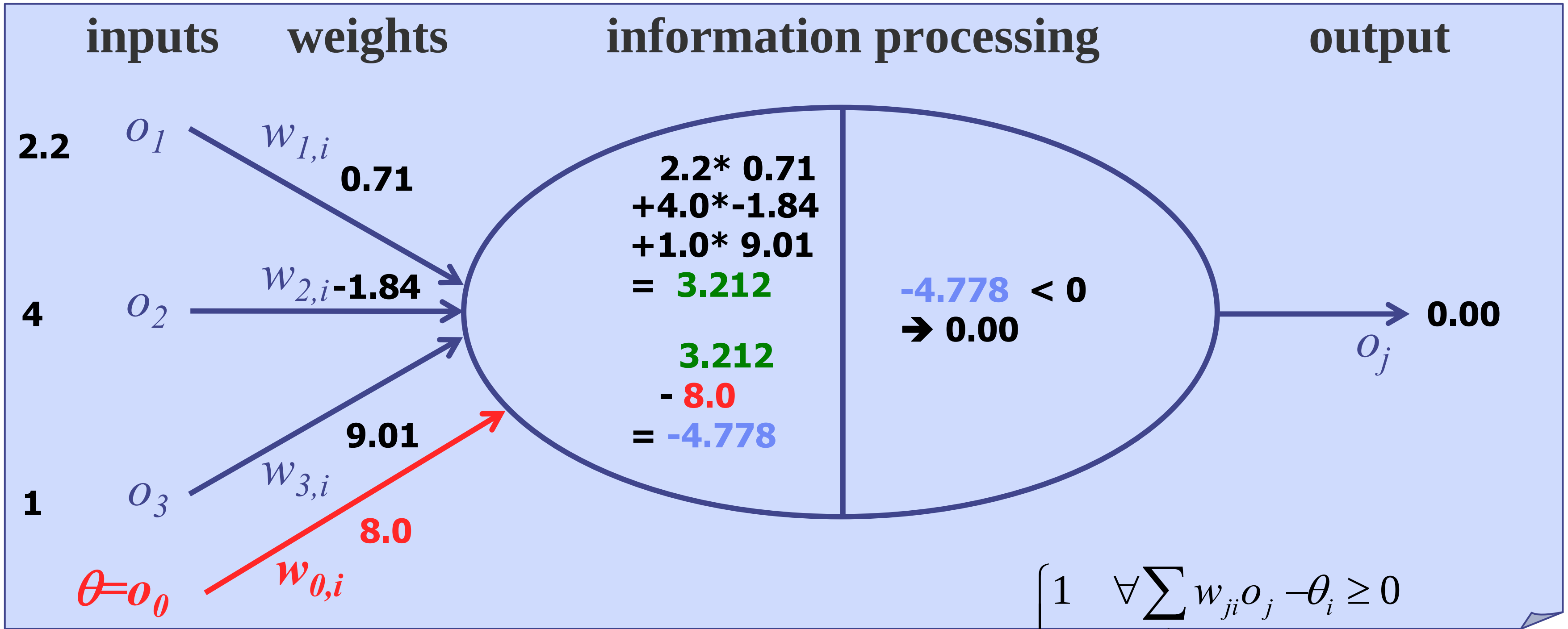
$$a_j = f_{act}\left(net_j, \theta_j\right) \qquad \text{e.g. } a_j = f_{act}\left(net_j - \theta_j\right)$$

Activation Function $a_j=f(net_j)$	Activation State a_j
 Binary Stepfunction / Threshold Function	Binary $o_j = \begin{cases} 1 & \forall\, net_j \geq \theta_j \\ 0 & \forall\, net_j < \theta_j \end{cases}$ $a_j = \mathrm{sgn}\left(net_j\right)$

Information Processing: Node Threshold logic

Node Function → BINARY THRESHOLD LOGIC

1. weight individual input by connection strength
2. sum weighted inputs
3. add bias term
4. calculate output of node through BINARY transfer function → RERUN with next input



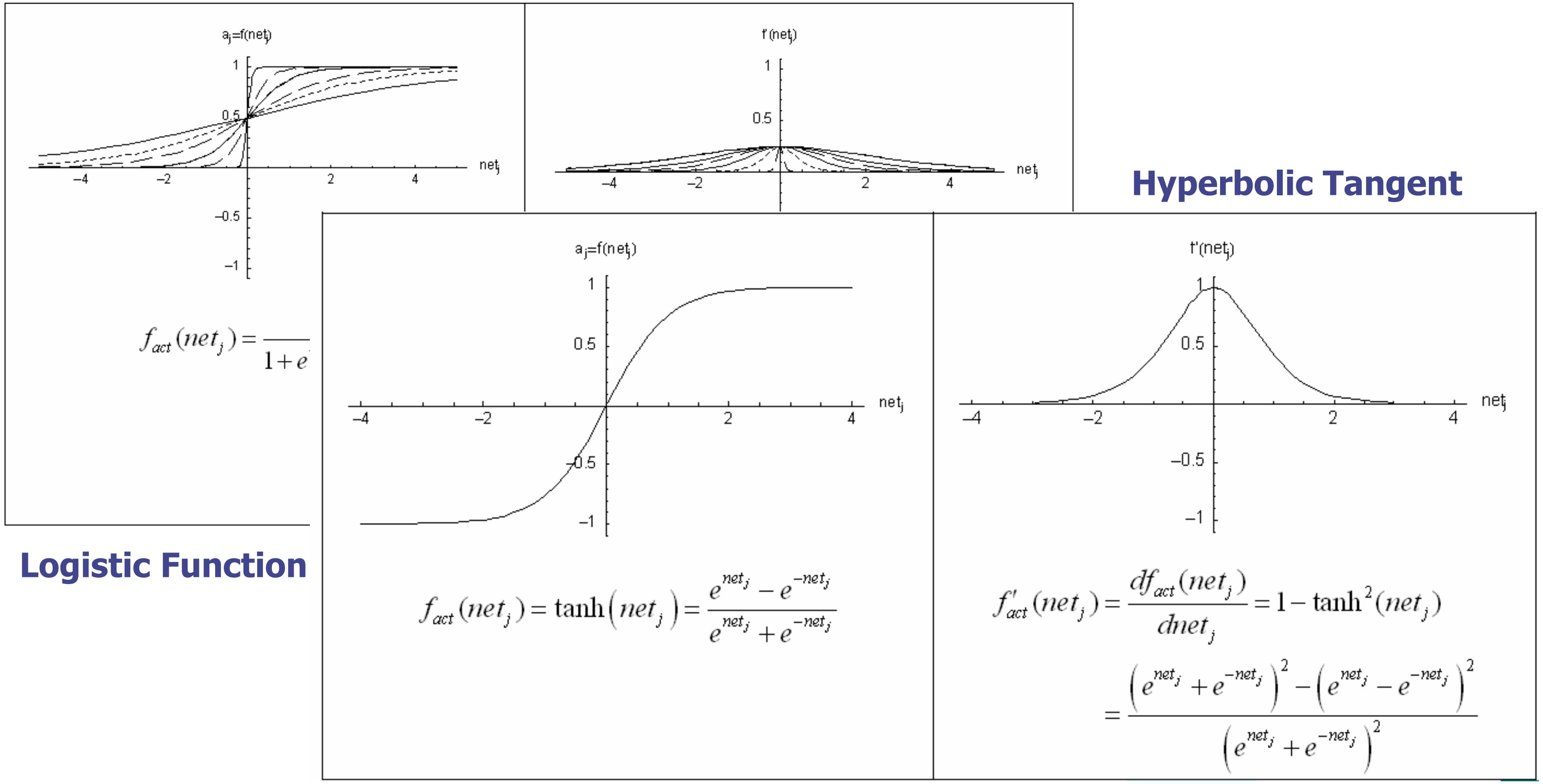
$$o_i = \begin{cases} 1 & \forall \sum_j w_{ji} o_j - \theta_i \geq 0 \\ 0 & \sum_j w_{ji} o_j - \theta_i < 0 \end{cases}$$

$$\Rightarrow \quad net_i = \sum_j w_{ij} o_j - \theta_j \quad \Rightarrow \quad a_i = f(net_i) \quad \Rightarrow$$

Continuous Activation Functions

- Activation calculated from input

$$a_j = f_{act}\left(net_j, \theta_j\right) \qquad \text{e.g.} \quad a_j = f_{act}\left(net_j - \theta_j\right)$$



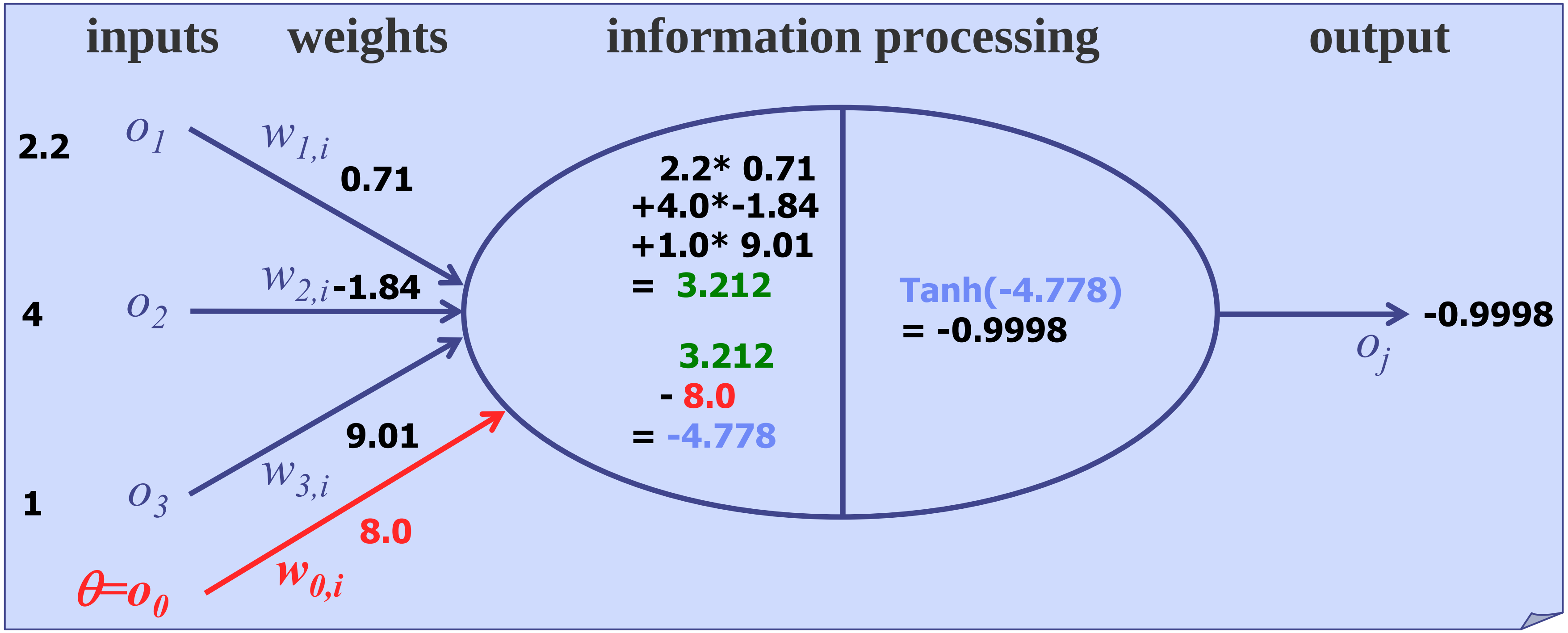
Hyperbolic Tangent

Logistic Function

Information Processing: Node Threshold logic

Node Function → Sigmoid THRESHOLD LOGIC of TanH activation function

1. weight individual input by connection strength
2. sum weighted inputs
3. add bias term
4. calculate output of node through BINARY transfer function → RERUN with next input



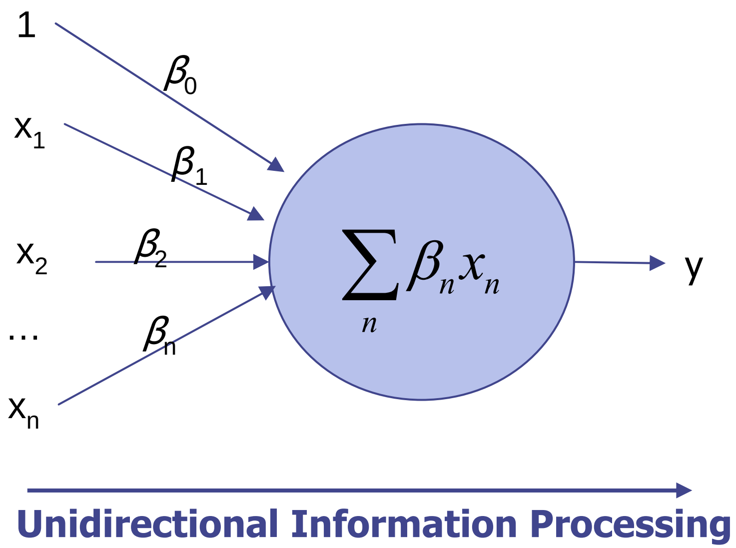
$$net_i = \sum_j w_{ij} o_j - \theta_j \Rightarrow a_i = f(net_i) \Rightarrow o_i = \tanh\left(\sum_j w_{ji} o_j - \theta_i\right)$$

A new Notation … GRAPHICS!

- Single Linear Regression … as an equation:

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$$

- Single Linear Regression … as a directed graph:



Why Graphical Notation?

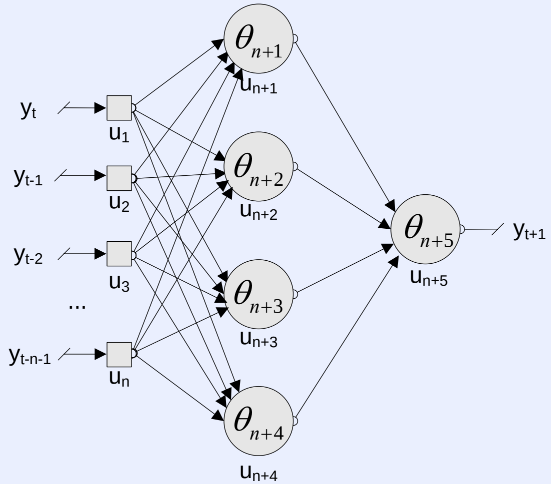
- Simple neural network equation without recurrent feedbacks:

$$y_k = \tanh\left(\sum_k w_{kj} \tanh\left(\sum_i w_{ki} \tanh\left(\sum_j w_{ji} x_j - \theta_j\right) - \theta_i\right) - \theta_k\right) \Rightarrow Min!$$

  - with … $\beta_i => w_i$, $\beta_0 => \theta$

$$\tanh\left(\left(\sum_{i=1}^{N} x_i \, w_{ij}\right) - \theta_j\right) = \frac{\left(e^{\left(\sum_{i=1}^{N} x_i w_{ij}\right) - \theta_j} - e^{-\left(\sum_{i=1}^{N} x_i w_{ij}\right) - \theta_j}\right)^2}{\left(e^{\left(\sum_{i=1}^{N} x_i w_{ij}\right) - \theta_j} + e^{-\left(\sum_{i=1}^{N} x_i w_{ij}\right) - \theta_j}\right)^2}$$

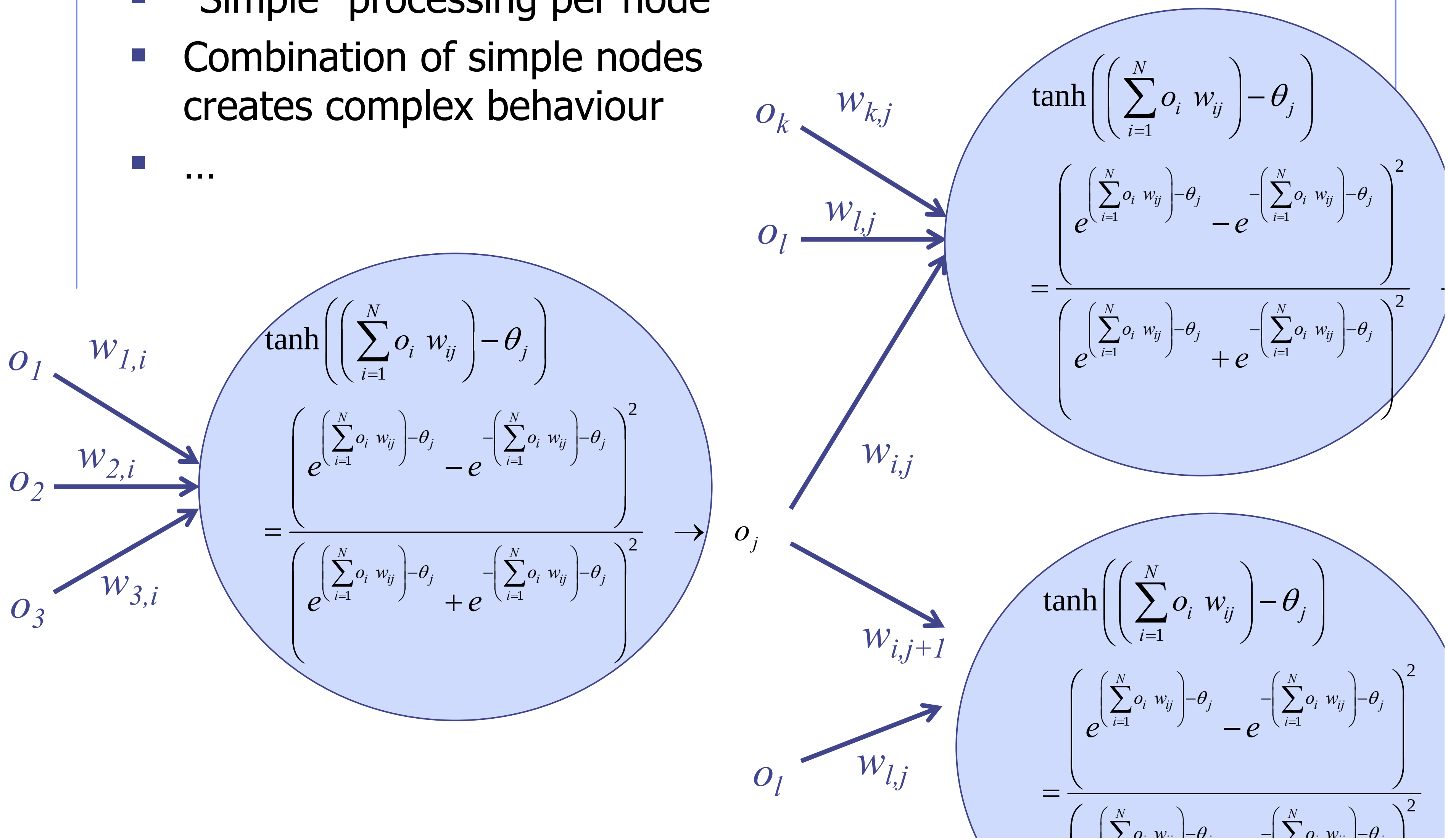
- Also:



→ Simplification for complex models!

Combination of Nodes

- "Simple" processing per node
- Combination of simple nodes creates complex behaviour
- …

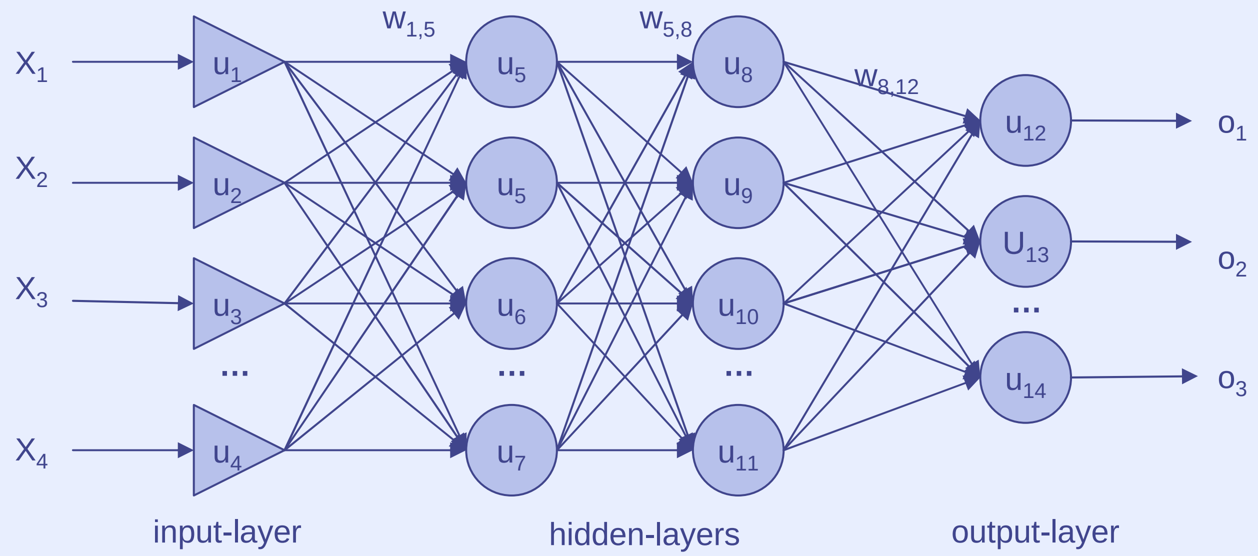


Architecture of Multilayer Perceptrons

- **Architecture of a Multilayer Perceptron**

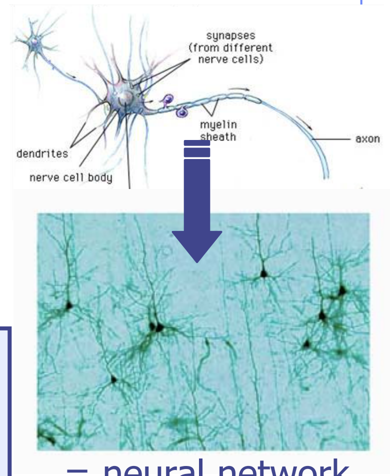
→ **Classic form of feed forward neural network!**

  - **Neurons u_n (units / nodes) ordered in Layers**
  - **unidirectional connections with trainable weights $w_{n,n}$**
  - **Vector of input signals x_i (input)**
  - **Vector of output signals o_j (output)**



$$o_k = \tanh\left(\sum_k w_{kj} \tanh\left(\sum_i w_{ki} \tanh\left(\sum_j w_{ji} o_j - \theta_j\right) - \theta_i\right) - \theta_k\right) \Rightarrow Min!$$

Combination of neurons



= neural network

Dictionary for Neural Network Terminology

- Due to its neuro-biological origins, NN use specific terminology

Neural Networks	Statistics
Input Nodes	Independent / lagged Variables
Output Node(s)	Dependent variable(s)
Training	Parameterization
Weights	Parameters
…	…

→ don't be confused: ASK!

Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
    1. What are NN? Definition & Online Preview …
    2. Motivation & brief history of Neural Networks
    3. From biological to artificial Neural Network Structures
    4. Network Training
3. Forecasting with Neural Networks …
4. How to write a good Neural Network forecasting paper!

Hebbian Learning

- HEBB introduced idea of learning by adapting weights [0,1]

$$\Delta w_{ij} = \eta o_i a_j$$

- Delta-learning rule of Widrow-Hoff

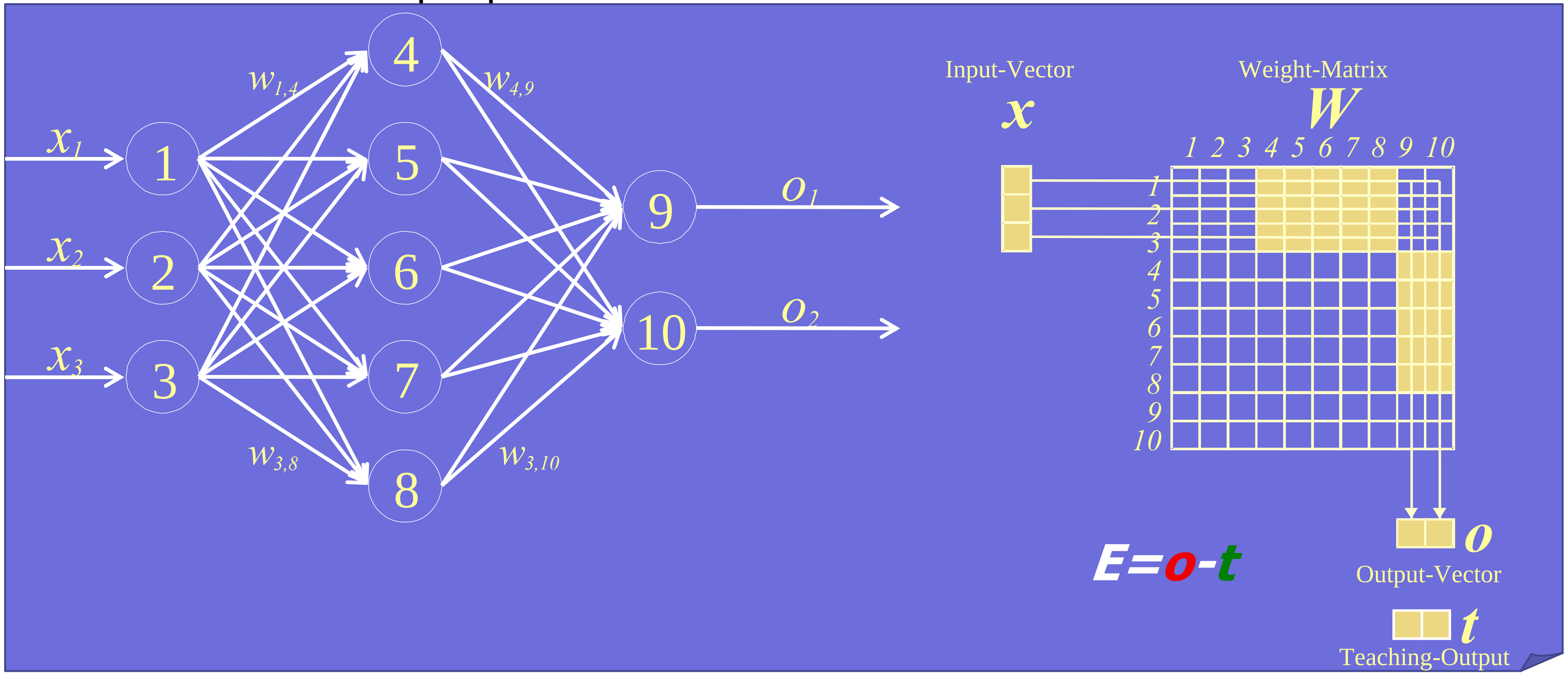
$$\begin{aligned}\Delta w_{ij} &= \eta o_i (t_j - a_j) \\ &= \eta o_i (t_j - o_j) = \eta o_i \delta_j\end{aligned}$$

Neural Network Training with Back-Propagation

Training → LEARNING FROM EXAMPLES

1. Initialize connections with randomized weights (symmetry breaking)
2. Show first Input-Pattern (independent Variables) (demo only for 1 node!)
3. Forward-Propagation of input values unto output layer
4. Calculate error between NN output & actual value (using error / objective function)
5. Backward-Propagation of errors for each weight unto input layer

➲ RERUN with next input pattern…



Neural Network Training

- Simple back propagation algorithm [Rumelhart et al. 1982]

$$E_p = C(t_{pj}, o_{pj}) \quad o_{pj} = f_j(net_{pj}) \qquad \Delta_p w_{ji} \propto -\frac{\partial C(t_{pj}, o_{pj})}{\partial w_{ji}}$$

$$\frac{\partial C(t_{pj}, o_{pj})}{\partial w_{ji}} = \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}}$$

$$\delta_{pj} = -\frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pj}}$$

$$\delta_{pj} = -\frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pj}} = \frac{\partial C(t_{pj}, o_{pj})}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}}$$

$$\frac{\partial o_{pt}}{\partial net_{pj}} = f_j^{'}(net_{pj})$$

$$\delta_{pj} = \frac{\partial C(t_{pj}, o_{pj})}{\partial o_{pj}} f_j^{'}(net_{pj})$$

$$\sum_k \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}} = \sum_k \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pk}} \frac{\partial \sum_i w_{ki} o_{pi}}{\partial o_{pj}}$$

$$= \sum_k \frac{\partial C(t_{pj}, o_{pj})}{\partial net_{pk}} w_{kj} = -\sum_k \delta_{pj} w_{kj}$$

$$\delta_{pj} = f_j^{'}(net_{pj}) \sum_k \delta_{pj} w_{kj}$$

$$\delta_{pj} = \begin{cases} \dfrac{\partial C(t_{pj}, o_{pj})}{\partial o_{pj}} f_j^{'}(net_{pj}) & \text{if unit } j \text{ is in the output layer} \\ f_j^{'}(net_{pj}) \sum_k \delta_{pk} w_{pjk} & \text{if unit } j \text{ is in a hidden layer} \end{cases}$$

$$\Delta w_{ij} = \eta o_i \delta_j$$

$$\text{mit } \delta_j = \begin{cases} f_j'(net_j)(t_j - o_j) & \forall \text{output nodes } j \\ f_j'(net_j) \sum_k (\delta_k w_{jk}) & \forall \text{hidden nodes } j \end{cases}$$

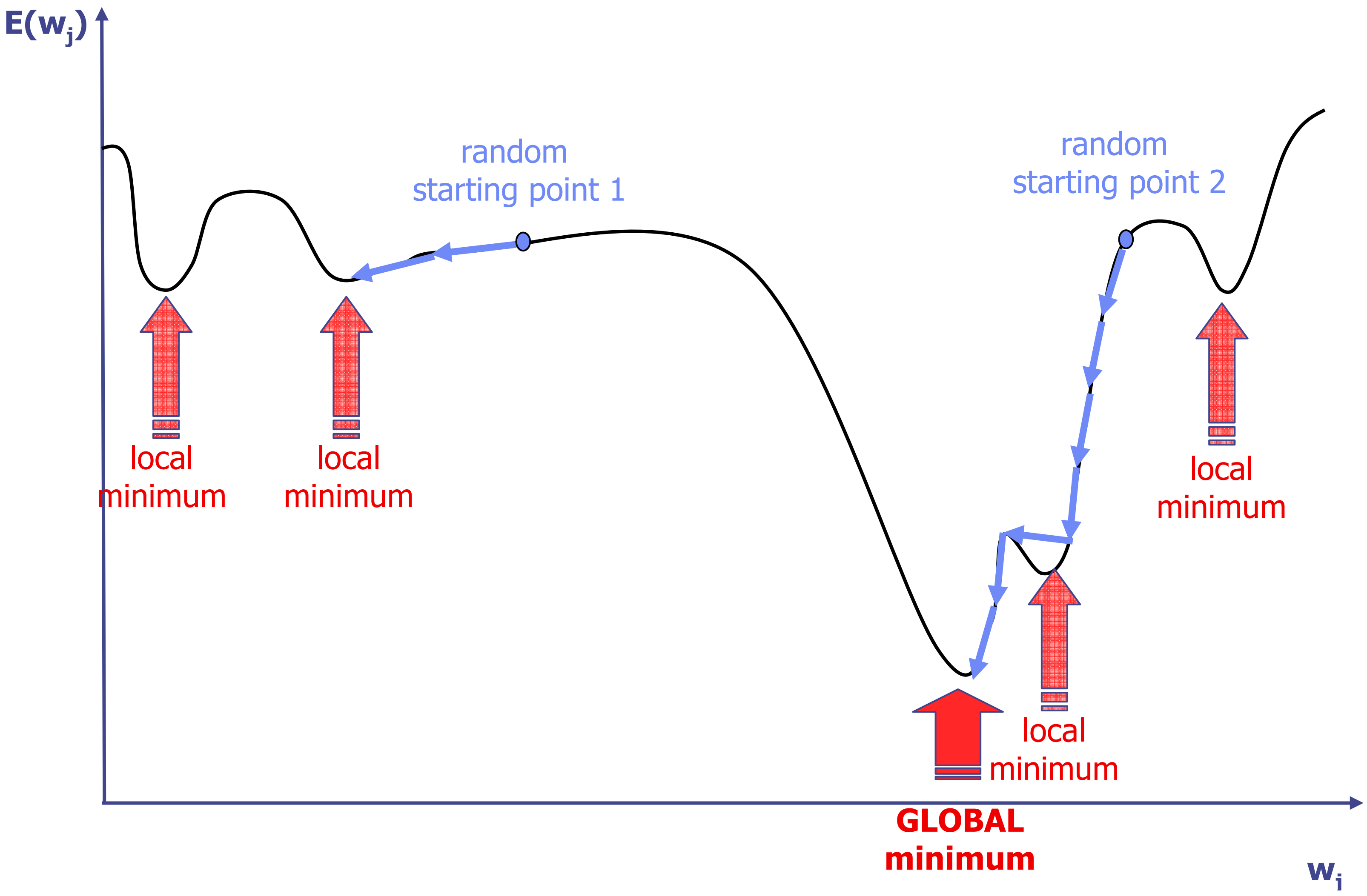
$$\Delta w_{ij} = \eta o_i \delta_j$$

$$\text{mit} \quad f(net_j) = \frac{1}{1 + e^{\sum_i o_i(t) w_{ij}}} \rightarrow f'(net_j) = o_j(1 - o_j)$$

$$\delta_j = \begin{cases} o_j(1 - o_j)(t_j - o_j) & \forall \text{output nodes } j \\ o_j(1 - o_j) \sum_k (\delta_k w_{jk}) & \forall \text{hidden nodes } j \end{cases}$$

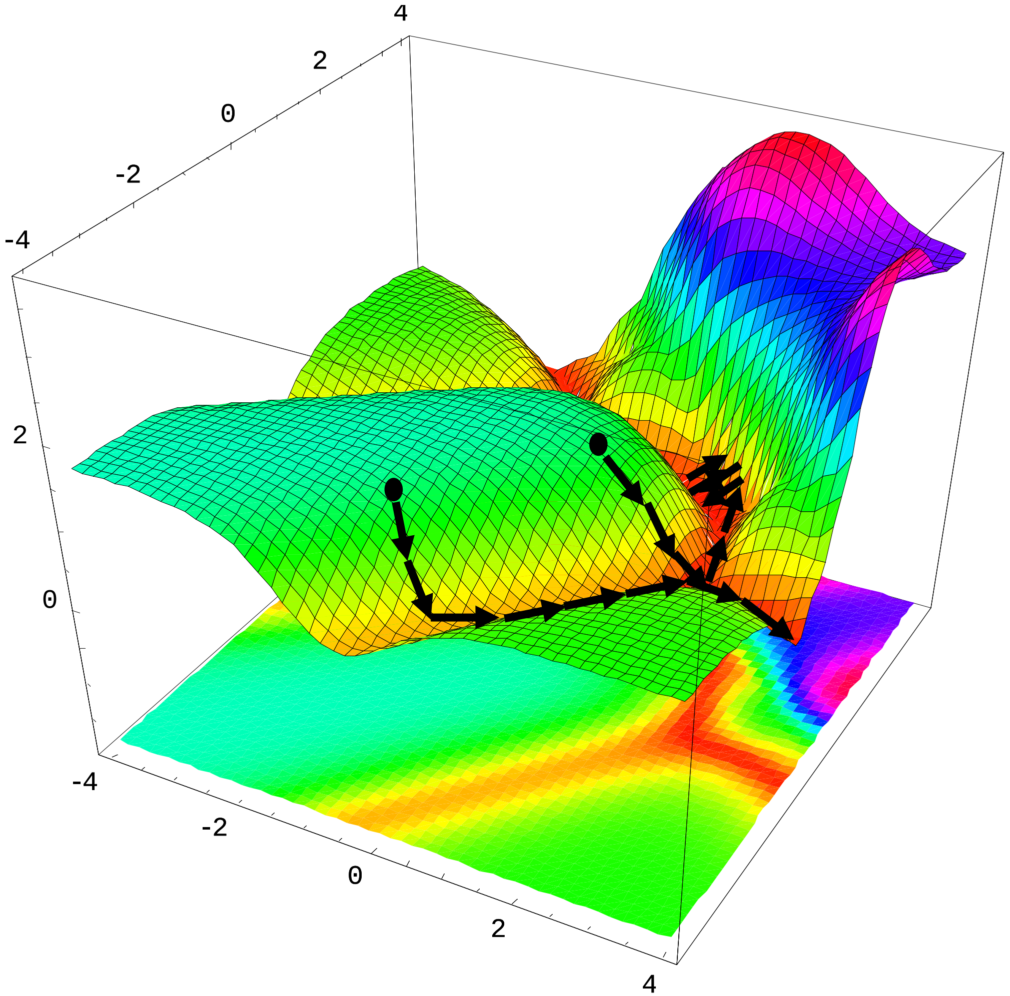
Neural Network Training = Error Minimization

- Minimize Error through changing ONE weight w_j



Error Backpropagation = 3D+ Gradient Decent

- Local search on multi-dimensional error surface

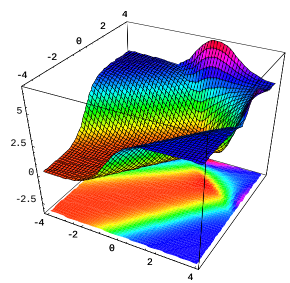


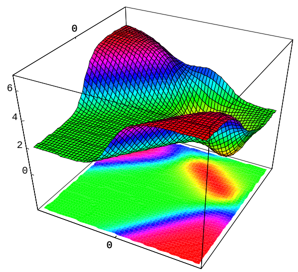
- task of finding the deepest valley in mountains
  - local search
  - stepsize fixed
  - follow steepest decent

→local optimum = any valley

→global optimum = deepest valley with lowest error

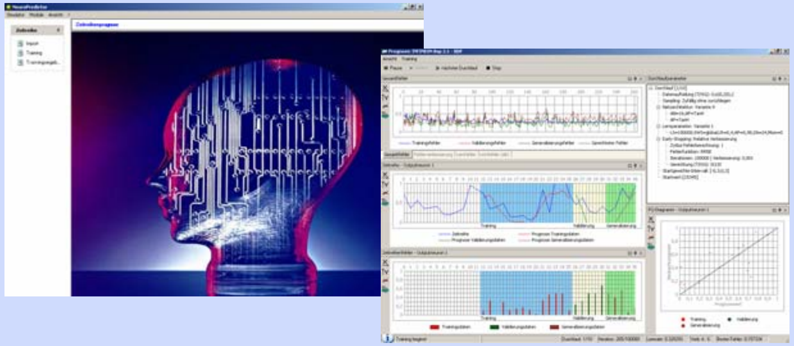
→varies with error surface



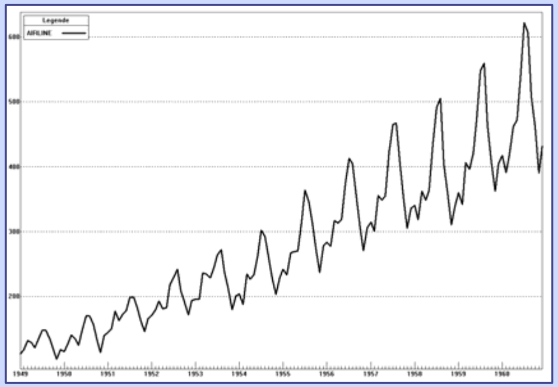


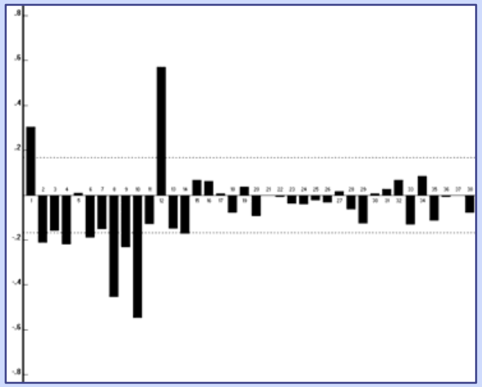
Demo: Neural Network Forecasting revistied!

- Simulation of NN for Business Forecasting



- Airline Passenger Data Experiment
  - 3 layered NN: (12-8-1) 12 Input units - 8 hidden units – 1 output unit
  - 12 input lags t, t-1, …, t-11 (past 12 observations) → time series prediction
  - t+1 forecast → single step ahead forecast





→ **Benchmark Time Series** [Brown / Box&Jenkins]

- **132 observations**
- **13 periods of monthly data**

Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
    1. NN models for Time Series & Dynamic Causal Prediction
    2. NN experiments
    3. Process of NN modelling
4. How to write a good Neural Network forecasting paper!

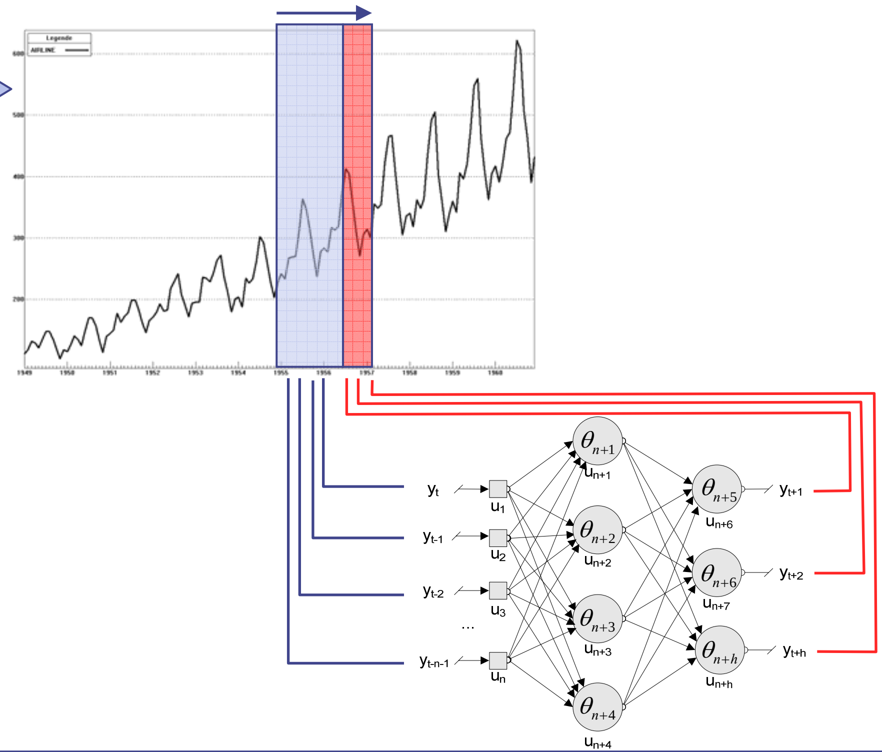
Time Series Prediction with Artificial Neural Networks

- ANN are universal approximators [Hornik/Stichcomb/White92 etc.]
  - Forecasts as application of (nonlinear) function-approximation
  - various architectures for prediction (time-series, causal, combined...)

$$\hat{y}_{t+h} = f(x_t) + \varepsilon_{t+h}$$

y_{t+h} = forecast for $t+h$
$f(-)$ = linear / non-linear function
x_t = vector of observations in t
e_{t+h} = independent error term in $t+h$

- Single neuron / node
  ≈ nonlinear AR(p)
- Feedforward NN (MLP etc.)
  ≈ hierarchy of nonlinear AR(p)
- Recurrent NN (Elman, Jordan)
  ≈ nonlinear ARMA(p,q)
- ...

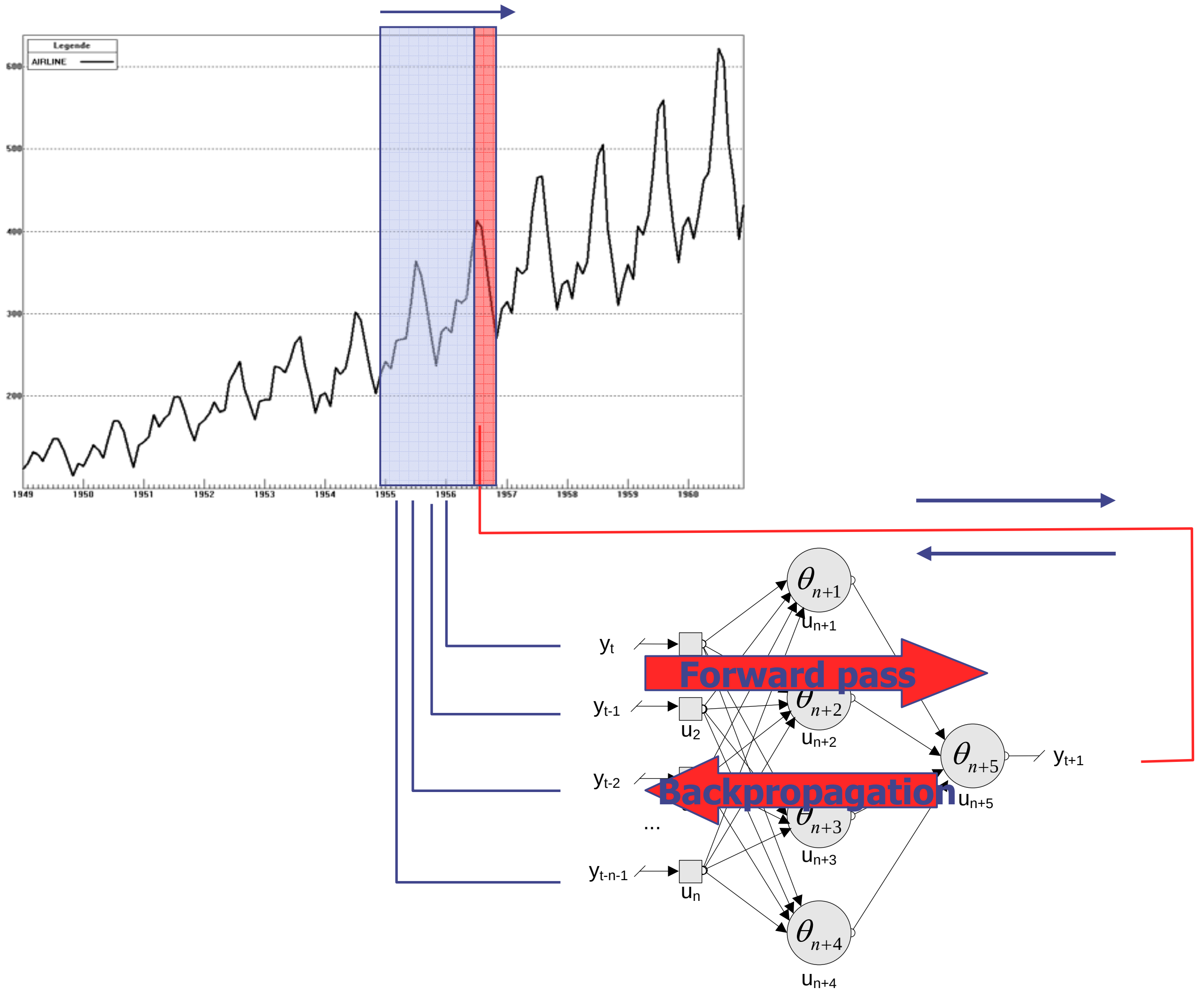


$$\hat{y}_{t+1} = f(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n-1})$$

Non-linear autoregressive AR(p)-model

Neural Network Training on Time Series

- Sliding Window Approach of presenting Data



Input

Present new data pattern to Neural Network

Calculate

Neural Network Output from Input values

Compare

Neural Network Forecast agains <> actual value

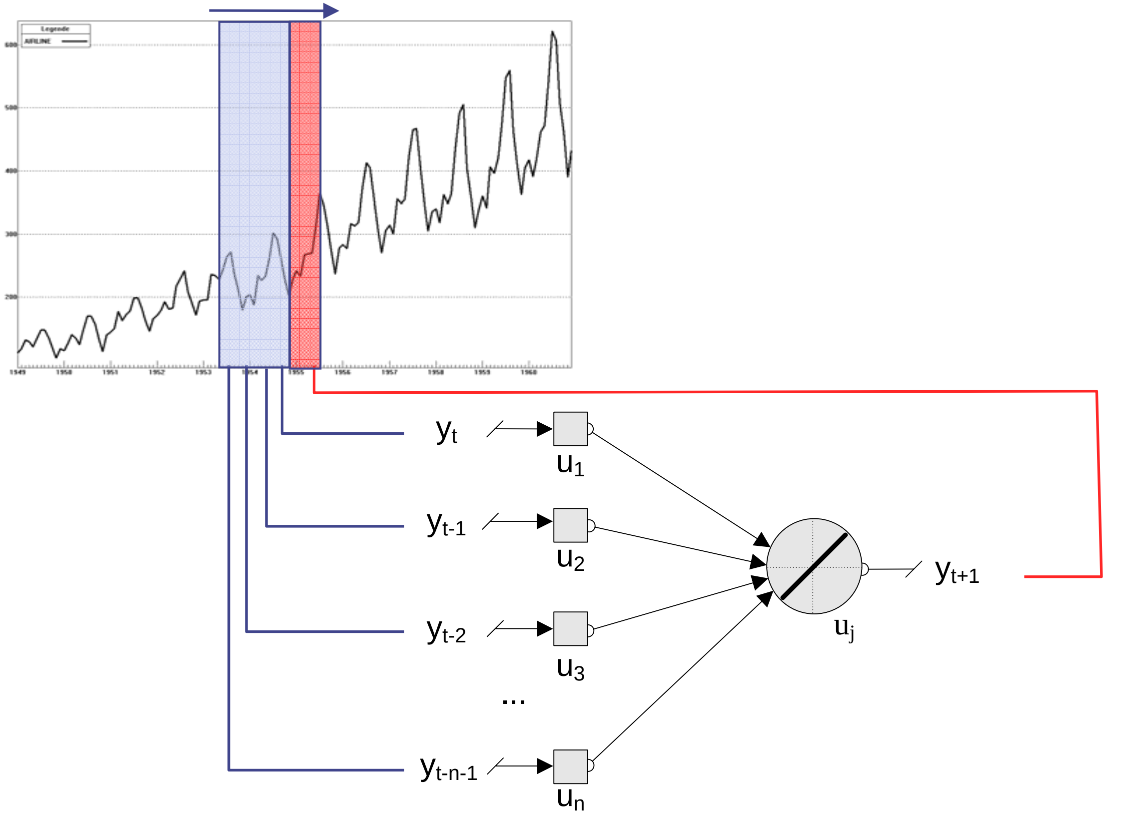
Backpropagation

Change weights to reduce output forecast error

New Data Input

Slide window forward to show next pattern

Neural Network Architectures for Linear Autoregression



$$\hat{y}_{t+1} = f\left(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n-1}\right)$$

$$\hat{y}_{t+1} = y_t w_{tj} + y_{t-1} w_{t-1j} + y_{t-2} w_{t-2j} + \ldots + y_{t-n-1} w_{t-n-1j} - \theta_j$$

linear autoregressive AR(p)-model

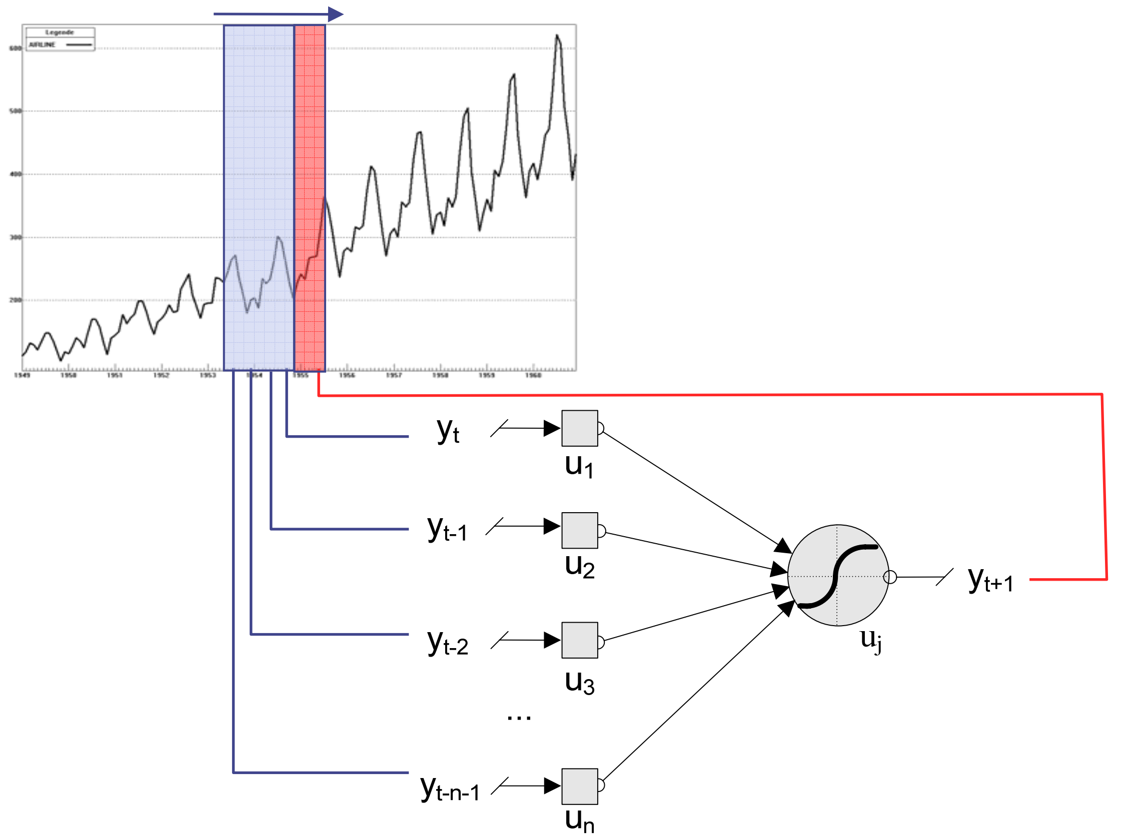
→ Interpretation

- **weights represent autoregressive terms**
- **Same problems / shortcomings as standard AR-models!**

→ Extensions

- **multiple output nodes = simultaneous auto-regression models**
- **Non-linearity through different activation function in output node**

Neural Network Architecture for Nonlinear Autoregression



$$\hat{y}_{t+1} = f\left(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n-1}\right)$$

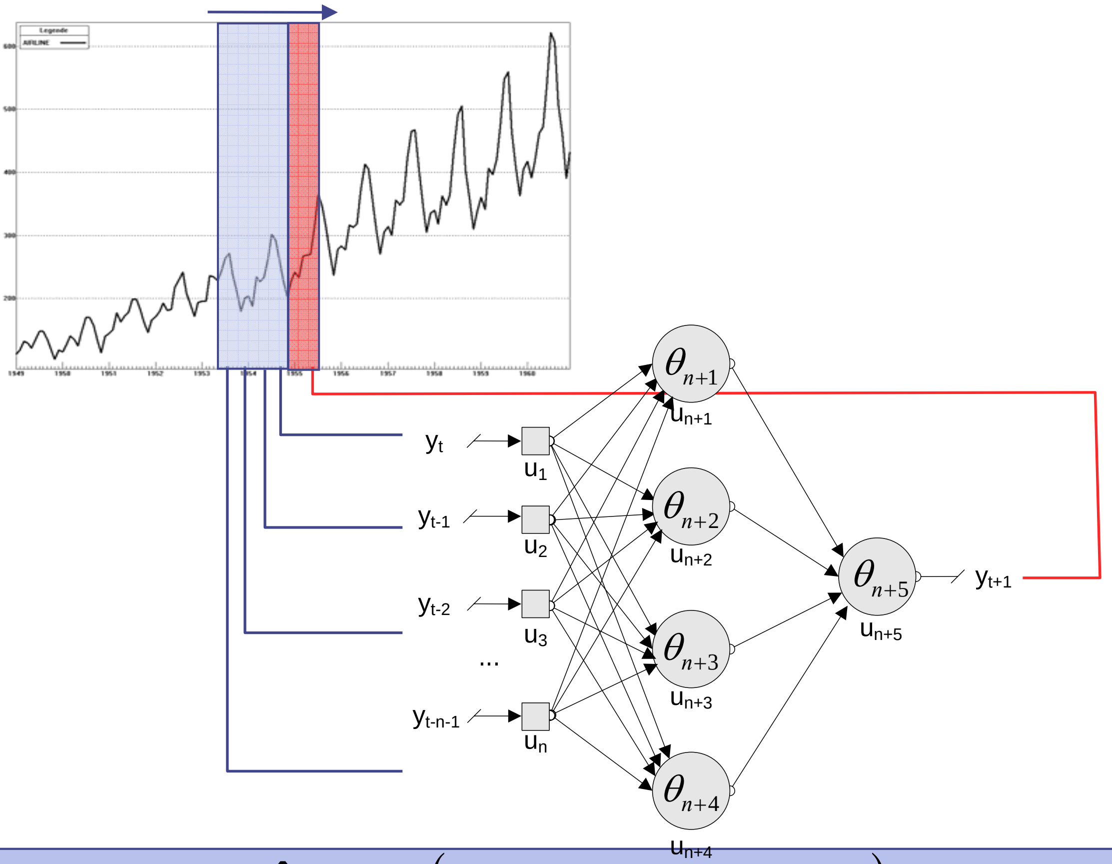
$$\hat{y}_{t+1} = \tanh\left(\sum_{i=t}^{t-n-1} y_i w_{ij} - \theta_j\right)$$

Nonlinear autoregressive AR(p)-model

→ Extensions

- ***additional layers with nonlinear nodes***
- ***linear activation function in output layer***

Neural Network Architectures for Nonlinear Autoregression



$$\hat{y}_{t+1} = f\left(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n-1}\right)$$

$$\hat{y}_{t+1} = \tanh\left(\sum_k w_{kj} \tanh\left(\sum_i w_{ki} \tanh\left(\sum_j w_{ji} y_{t-j} - \theta_j\right) - \theta_i\right) - \theta_k\right)$$

Nonlinear autoregressive AR(p)-model

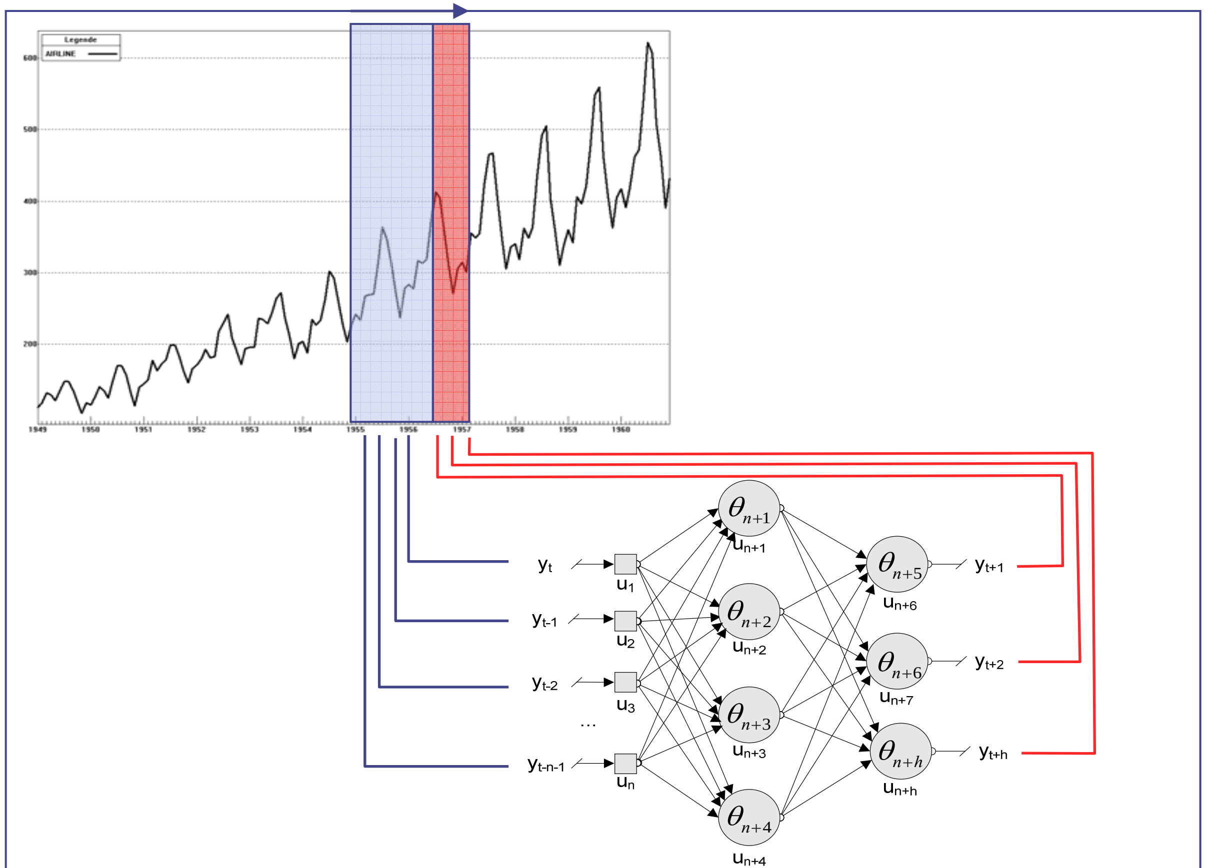
→ Interpretation

- **Autoregressive modeling AR(p)-approach WITHOUT the moving average terms of errors ≠ nonlinear ARIMA**
- **Similar problems / shortcomings as standard AR-models!**

→ Extensions

- **multiple output nodes = simultaneous auto-regression models**

Neural Network Architectures for Multiple Step Ahead Nonlinear Autoregression



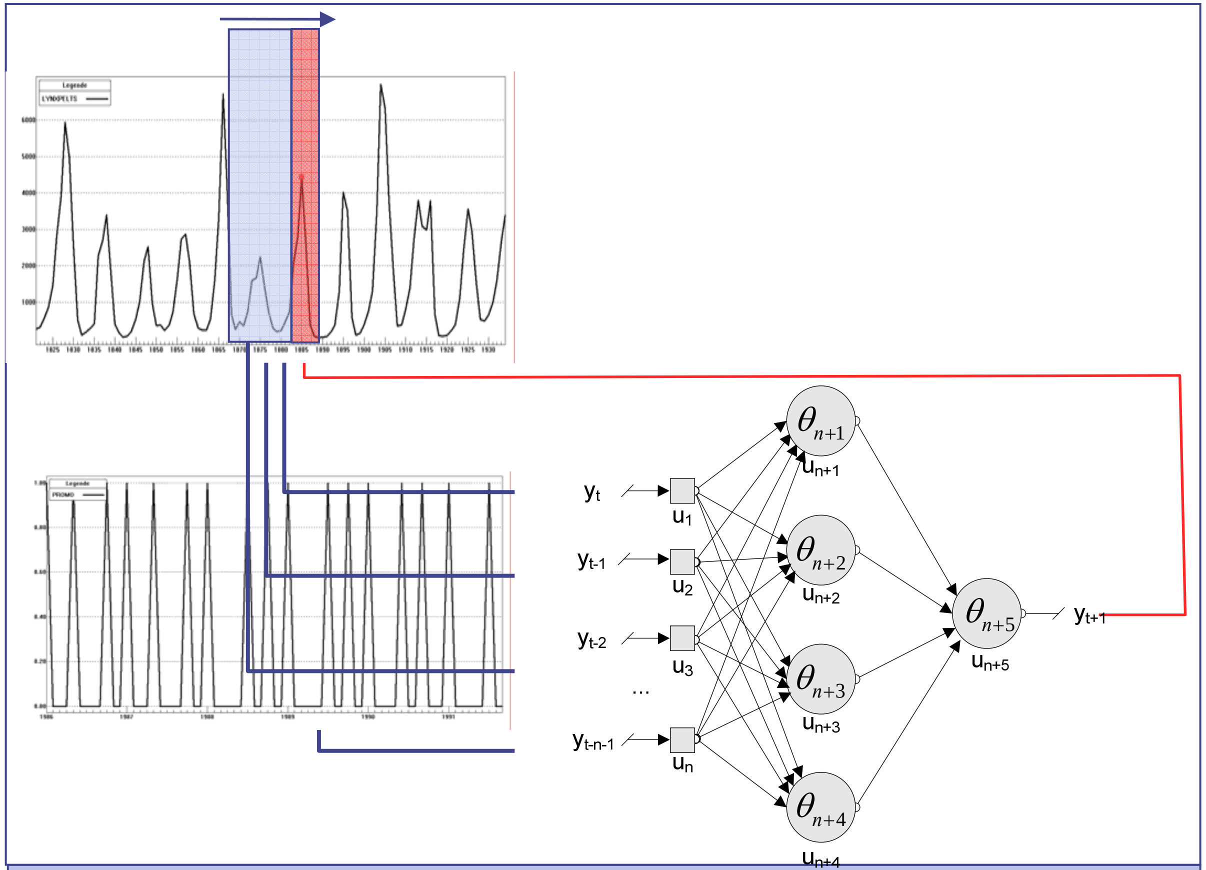
$$\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+n} = f\left(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n-1}\right)$$

Nonlinear autoregressive AR(p)-model

→ Interpretation

- **As single Autoregressive modeling AR(p)**

Neural Network Architectures for Forecasting - Nonlinear Autoregression Intervention Model



$$\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+n} = f\left(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n-1}\right)$$

Nonlinear autoregressive ARX(p)-model

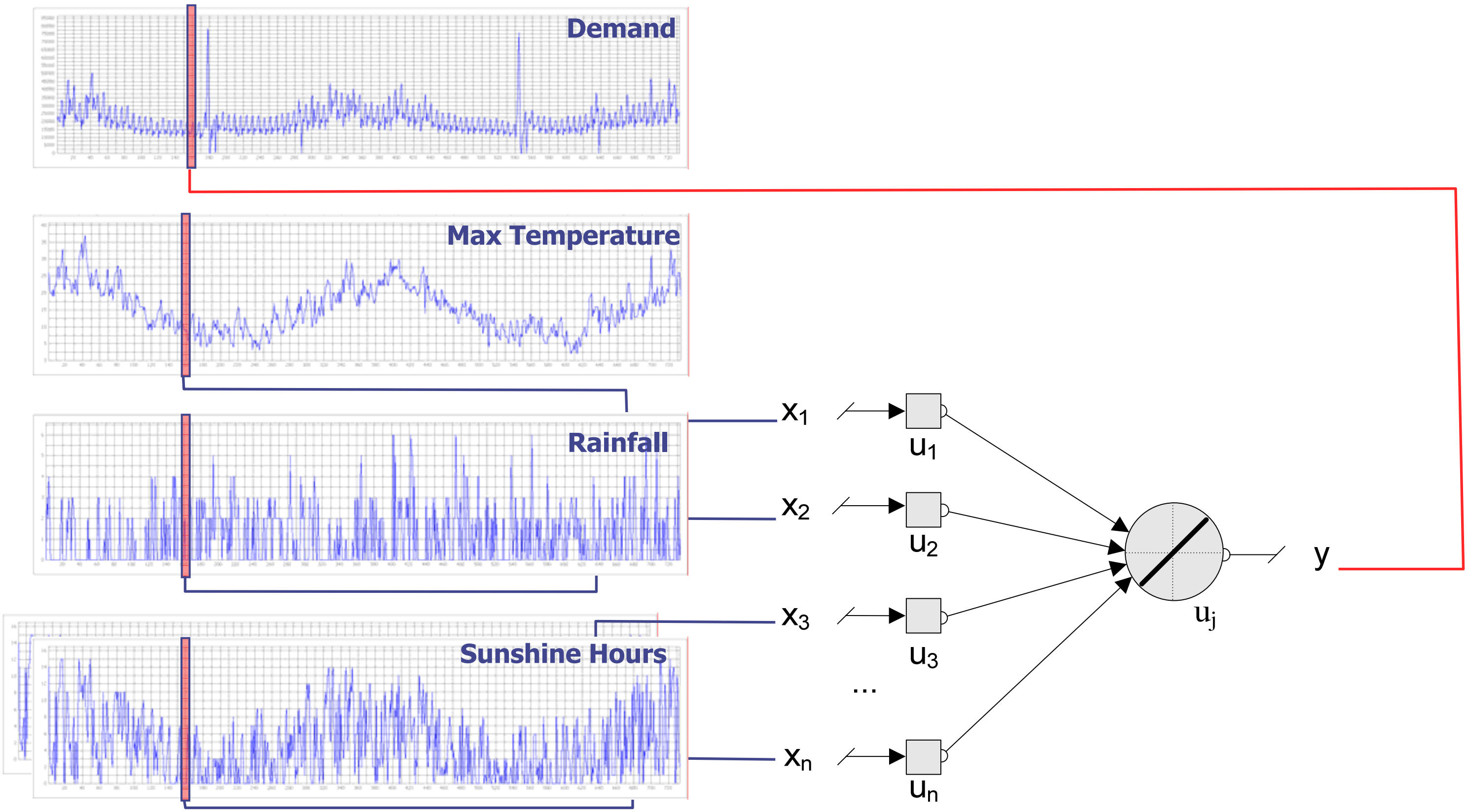
→ Interpretation

- **As single Autoregressive modeling AR(p)**
- **Additional Event term to explain external events**

→ Extensions

- **multiple output nodes = simultaneous multiple regression**

Neural Network Architecture for Linear Regression

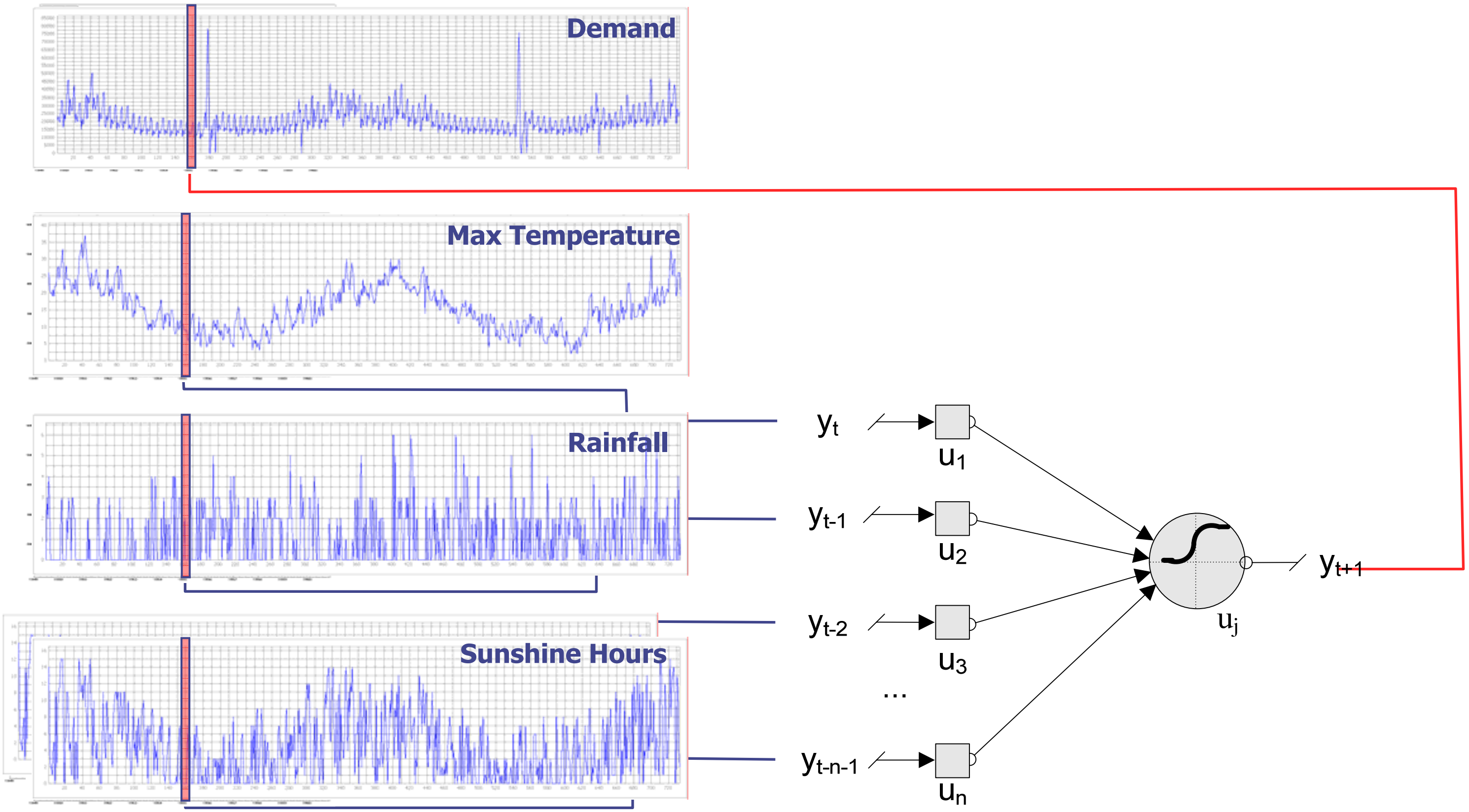


$$\hat{y} = f\left(x_1, x_2, x_3, \ldots, x_n\right)$$

$$\hat{y} = x_1 w_{1j} + x_{2t} w_{2j} + x_{3t} w_{3j} + \ldots + x_n w_{nj} - \theta_j$$

Linear Regression Model

Neural Network Architectures for Non-Linear Regression (≈Logistic Regression)

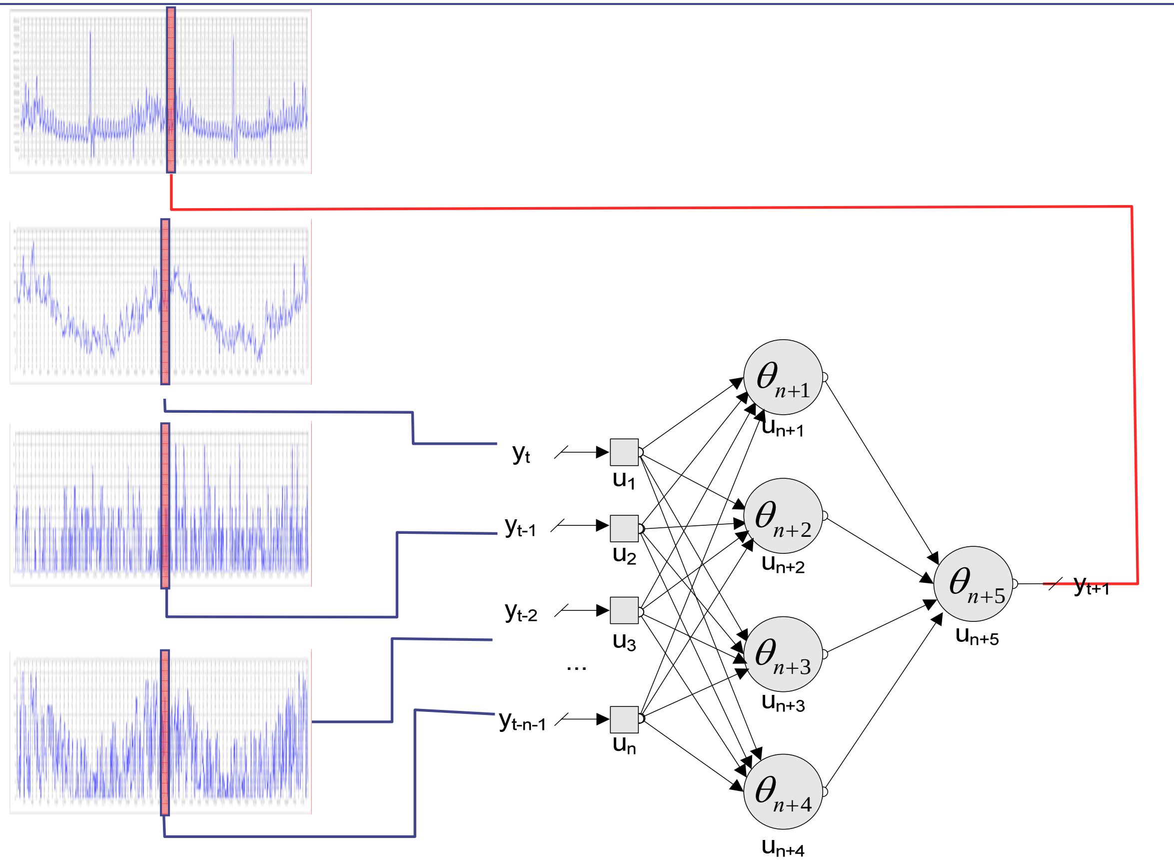


$$\hat{y}_{t+1} = f\left(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n-1}\right)$$

$$\hat{y}_{t+1} = \log\left(\sum_{i=t}^{t-n-1} y_i w_{ij} - \theta_j\right)$$

Nonlinear Multiple (Logistic) Regression Model

Neural Network Architectures for Non-linear Regression



$$\hat{y} = f\left(x_1, x_2, x_3, ..., x_n\right)$$

$$\hat{y} = x_1 w_{1j} + x_{2t} w_{2j} + x_{3t} w_{3j} + ... + x_n w_{nj} - \theta_j$$

Nonlinear Regression Model

→ Interpretation

- **Similar to linear Multiple Regression Modeling**
- **Without nonlinearity in output: weighted expert regime on non-linear regression**
- **With nonlinearity in output layer: ???**

Classification of Forecasting Methods

- Forecasting Methods
  - Objective Forecasting Methods
    - Time Series Methods
      - Averages
      - Moving Averages
      - Naive Methods
      - Exponential Smoothing
        - Simple ES
        - Linear ES
        - Seasonal ES
        - Dampened Trend ES
      - Simple Regression
      - Autoregression ARIMA
      - Neural Networks
    - Causal Methods
      - Linear Regression
      - Multiple Regression
      - Dynamic Regression
      - Vector Autoregression
      - Intervention model
      - Neural Networks
  - Subjective Forecasting Methods
    - Sales Force Composite
    - Analogies
    - Delphi
    - PERT
    - Survey techniques
  - „Prophecy“ educated guessing…

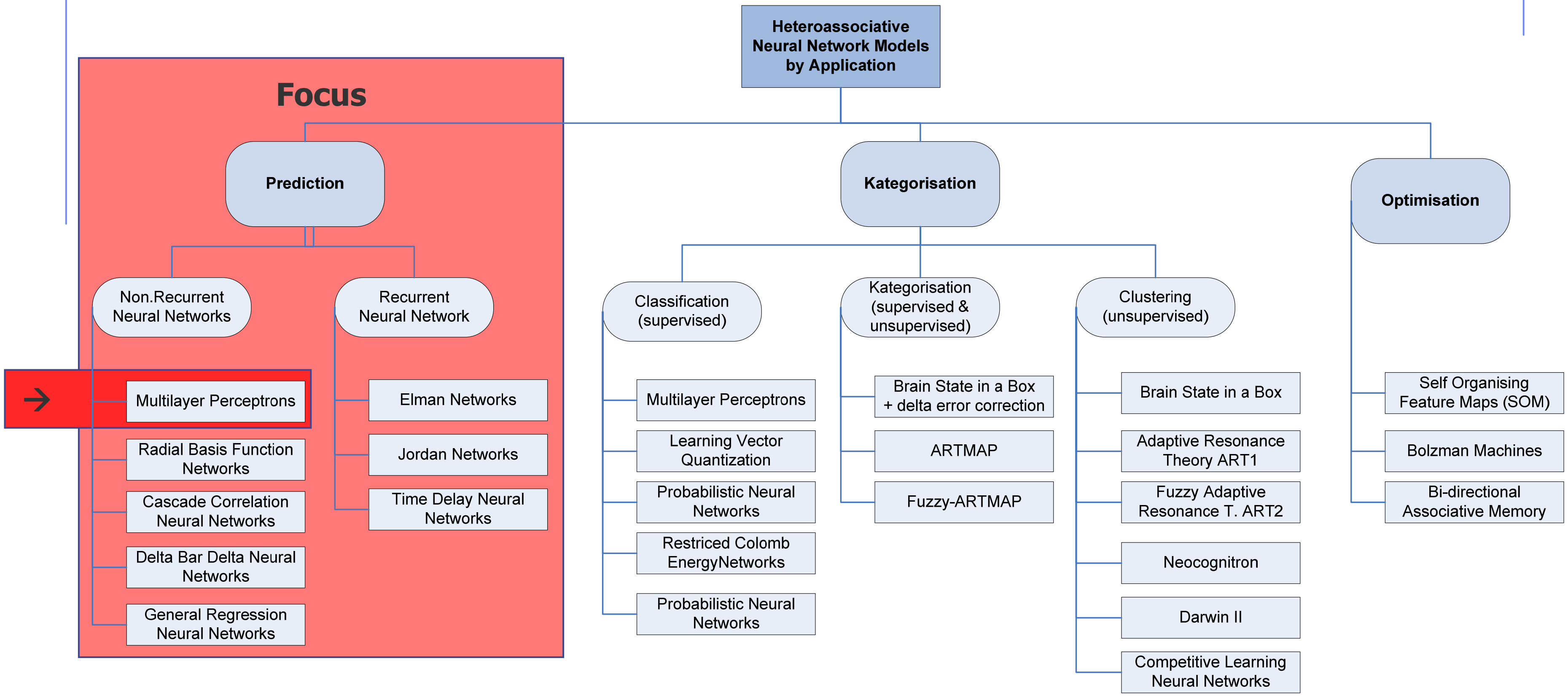
Demand Planning Practice
Objektive Methods + Subjektive correction

Neural Networks ARE
- time series methods
- causal methods

& CAN be used as
- Averages & ES
- Regression …

Different model classes of Neural Networks

- Since 1960s a variety of NN were developed for different tasks
  → Classification ≠ Optimization ≠ Forecasting → Application Specific Models



- Different CLASSES of Neural Networks for Forecasting alone!
  → Focus only on original Multilayer Perceptrons!

Problem!

- MLP most common NN architecture used
- MLPs with sliding window can ONLY capture nonlinear seasonal autoregressive processes nSAR(p,P)

- BUT:
  - Can model MA(q)-process through extended AR(p) window!
  - Can model SARMAX-processes through recurrent NN

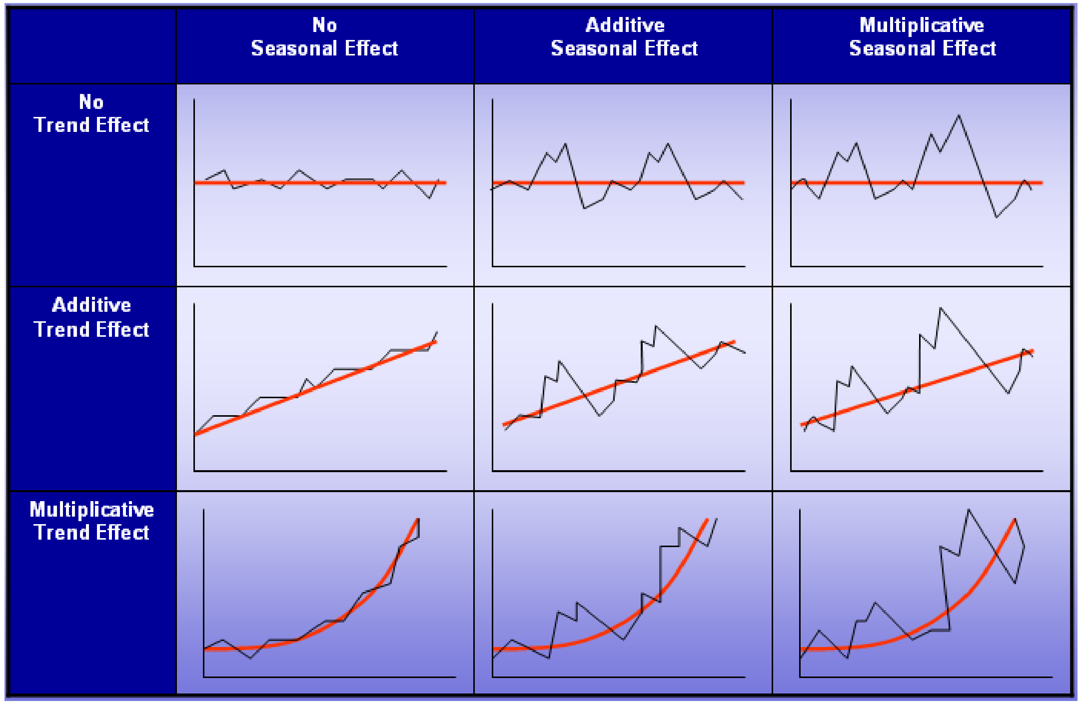
Agenda

Forecasting with Artificial Neural Networks

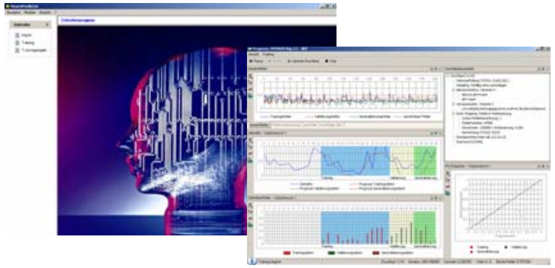
1. Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
    1. NN models for Time Series & Dynamic Causal Prediction
    2. NN experiments
    3. Process of NN modelling
4. How to write a good Neural Network forecasting paper!

Time Series Prediction with Artificial Neural Networks

- Which time series patterns can ANNs learn & extrapolate? [Pegels69/Gardner85]



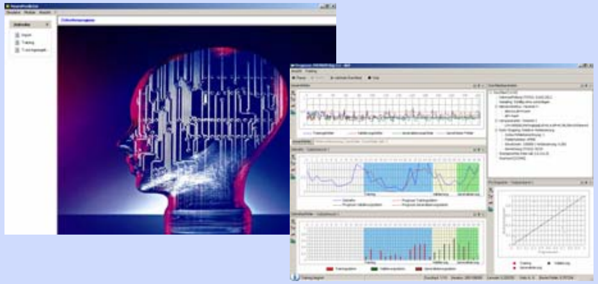
- ... ???



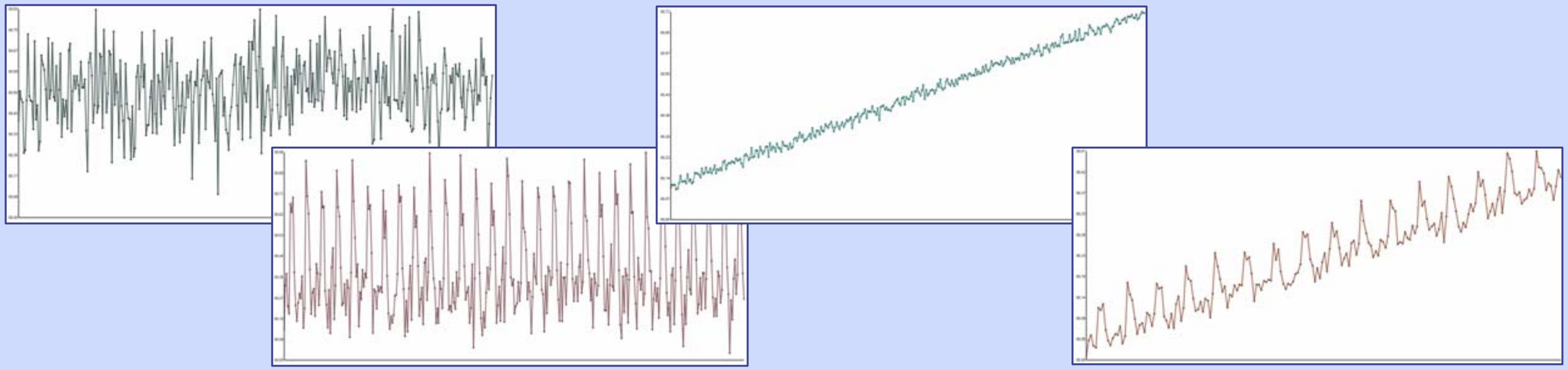
→ **Simulation of Neural Network prediction of Artificial Time Series**

Time Series Demonstration – Artificial Time Series

- Simualtion of NN in Business Forecasting with NeuroPredictor

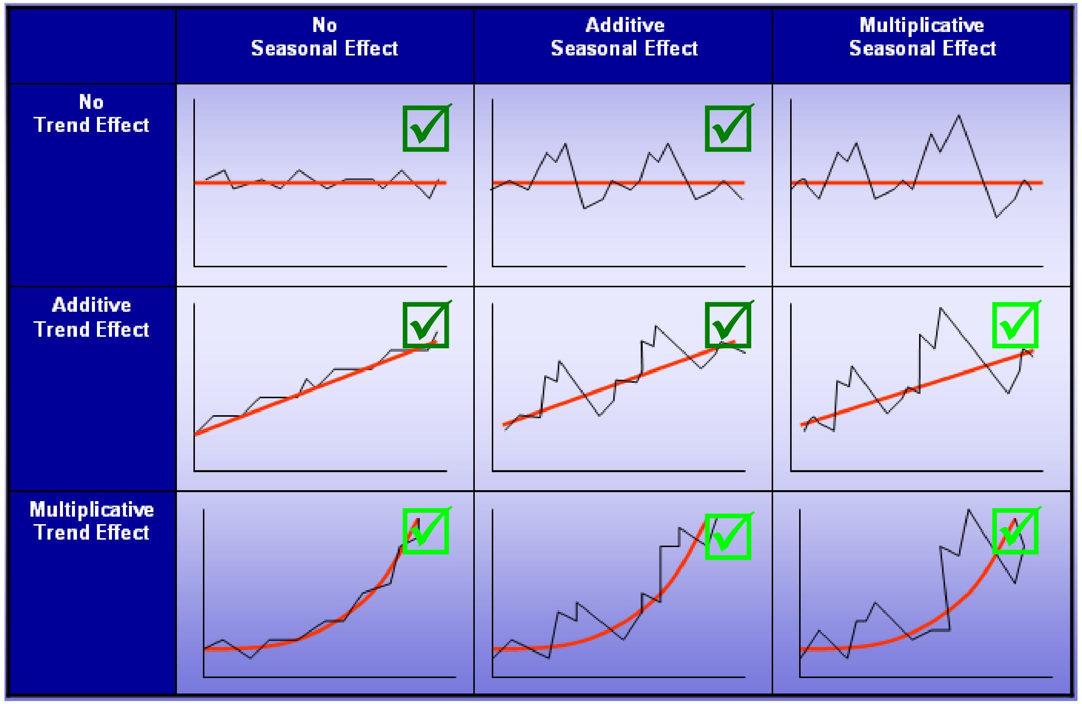


- Experiment: Prediction of Artificial Time Series (Gaussian noise)
  - Stationary Time Series
  - Seasonal Time Series
  - linear Trend Time Series
  - Trend with additive Seasonality Time Series



Time Series Prediction with Artificial Neural Networks

- Which time series patterns can ANNs learn & extrapolate? [Pegels69/Gardner85]

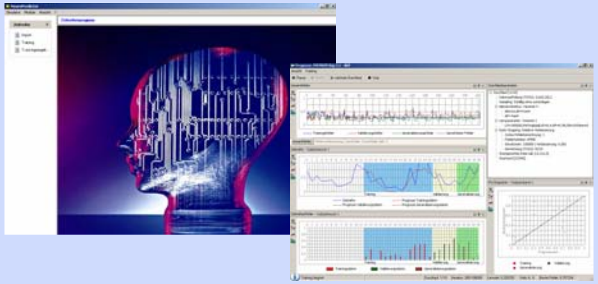


→ **Neural Networks can forecast ALL mayor time series patterns**

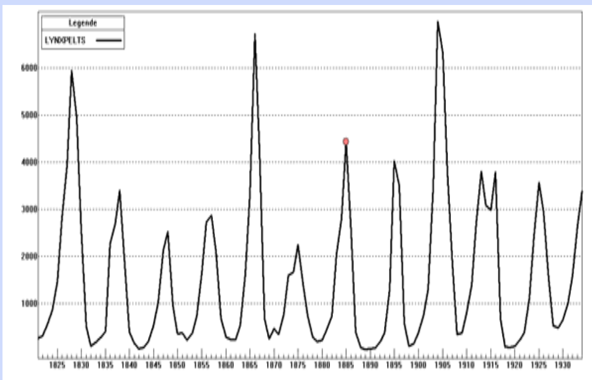
- → NO time series dependent preprocessing / integration necessary
- → NO time series dependent MODEL SELECTION required!!!
- → **SINGLE MODEL APPROACH FEASIBLE!**

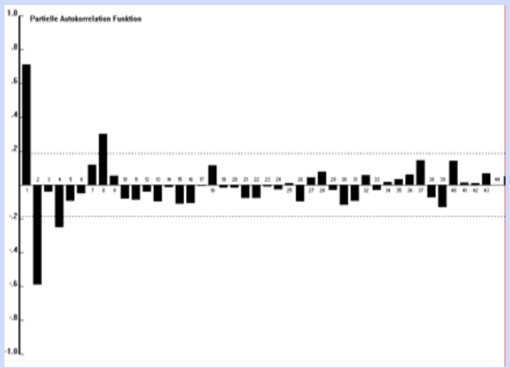
Time Series Demonstration A - Lynx Trappings

- Simulation of NN in Business Forecasting



- Experiment: Lynx Trappings at the McKenzie River
    - 3 layered NN: (12-8-1) 12 Input units - 8 hidden units – 1 output unit
    - Different lag structures: t, t-1, …, t-11 (past 12 observations
    - t+1 forecast → single step ahead forecast



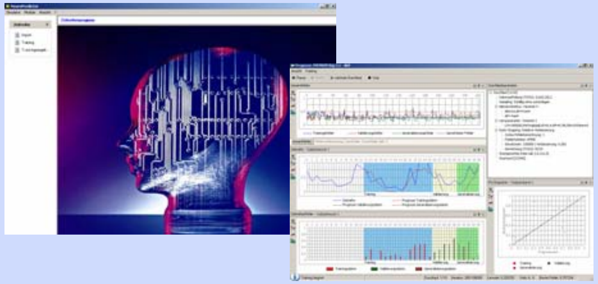


→ **Benchmark Time Series** [Andrews / Hertzberg]

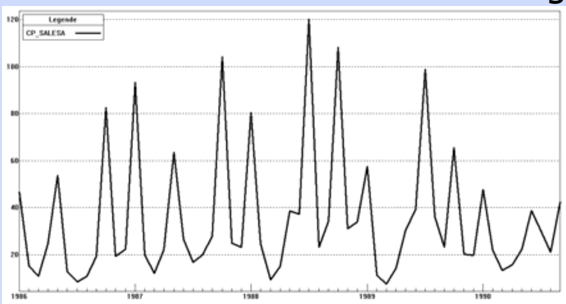
- **114 observations**
- **Periodicity? 8 years?**

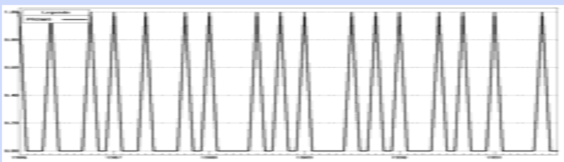
Time Series Demonstration B – Event Model

- Simulation of NN in Business Forecasting



- Experiment: Mouthwash Sales
  - 3 layered NN: (12-8-1) 12 Input units - 8 hidden units – 1 output unit
  - 12 input lags t, t-1, …, t-11 (past 12 observations) → time series prediction
  - t+1 forecast → single step ahead forecast





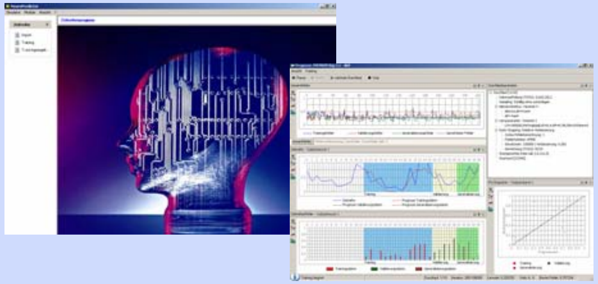
→ Spurious Autocorrelations from Marketing Events

▪Advertisement with small Lift

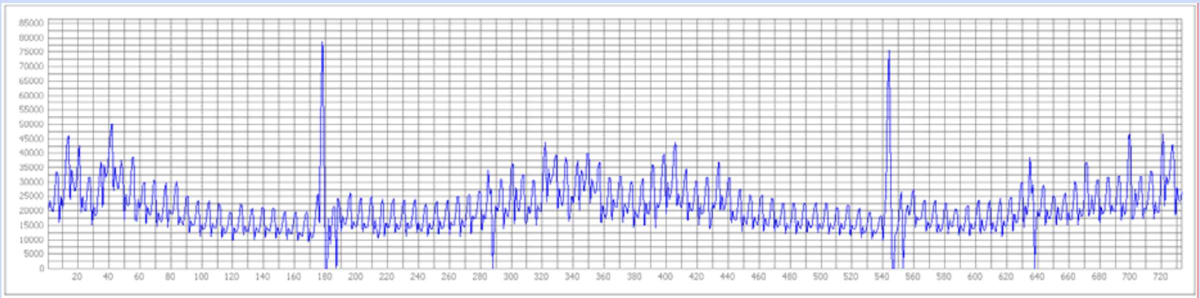
▪Price-reductions with high Lift

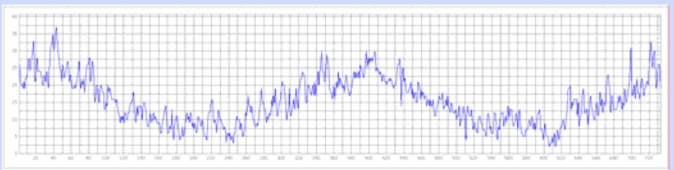
Time Series Demonstration C – Supermarket Sales

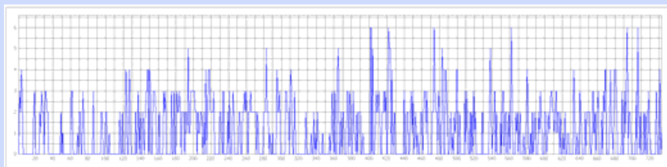
- Simulation of NN in Business Forecasting



- Experiment: Supermarket sales of fresh products with weather
  - 4 layered NN: (7-4-4-1) 7 Input units - 8 hidden units – 1 output unit t+4
  - Different lag structures: t, t-1, …, t-7 (past 12 observations)
  - t+4 forecast → single step ahead forecast







Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
    1. NN models for Time Series & Dynamic Causal Prediction
    2. NN experiments
    3. Process of NN modelling
        1. Preprocessing
        2. Modelling NN Architecture
        3. Training
        4. Evaluation
4. How to write a good Neural Network forecasting paper!

Decisions in Neural Network Modelling

NN Modelling Process

- Data Pre-processing
  - Transformation
  - Scaling
  - Normalizing to [0;1] or [-1;1]
- Modelling of NN architecture
  - Number of INPUT nodes
  - Number of HIDDEN nodes
  - Number of HIDDEN LAYERS
  - Number of OUTPUT nodes
  - Information processing in Nodes (Act. Functions)
  - Interconnection of Nodes
- Training
  - Initializing of weights (how often?)
  - Training method (backprop, higher order …)
  - Training parameters
  - Evaluation of best model (early stopping)
- Application of Neural Network Model
- Evaluation
  - Evaluation criteria & selected dataset

manual
Decisions recquire
Expert-Knowledge

Modeling Degrees of Freedom

- **Variety of Parameters must be pre-determined for ANN Forecasting:**

D= Dataset	[D^{SE} Selection	D^{SA}] Sampling				
P= Preprocessing	[C Correction	N Normalization	S] Scaling			
A= Architecture	[N^{I} no. of input nodes	N^{S} no. of hidden nodes	N^{L} no. of hidden layers	N^{O} no. of output nodes	K connectivity / weight matrix	T] Activation Strategy
U= signal processing	[F^{I} Input function	F^{A} Activation Function	F^{O}] Output Function			
L= learning algorithm	[G choice of Algorithm	$P^{T,L}$ Learning parameters phase & layer	I^{P} initializations procedure	I^{N} number of initializations	B] stopping method & parameters	
O objective Function						

→ **interactions & interdependencies between parameter choices!**

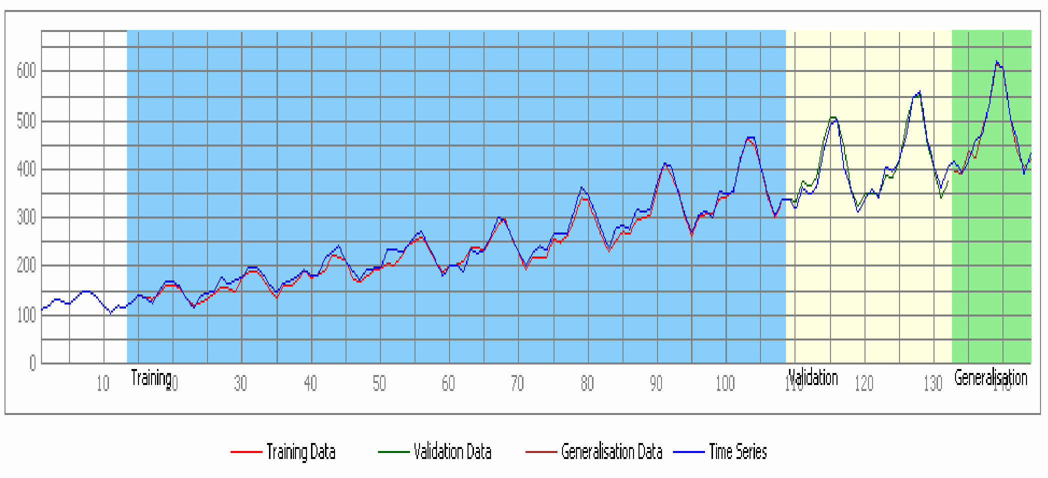
Heuristics to Reduce Design Complexity

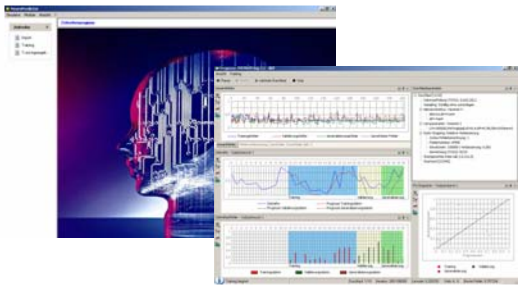
- Number of Hidden nodes in MLPs (in no. of input nodes n)
  - 2n+1 [Lippmann87; Hecht-Nielsen90; Zhang/Pauwo/Hu98]
  - 2n [Wong91]; n [Tang/Fishwick93]; n/2 [Kang91]
  - 0.75n [Bailey90]; 1.5n to 3n [Kasstra/Boyd96] …
- Activation Function and preprocessing
  - logistic in hidden & output [Tang/Fischwick93; Lattermacher/Fuller95; Sharda/Patil92]
  - hyperbolic tangent in hidden & output [Zhang/Hutchinson93; DeGroot/Wurtz91]
  - linear output nodes [Lapedes/Faber87; Weigend89-91; Wong90]
- ... with interdependencies!

→ no research on relative performance of all alternatives
→ no empirical results to support preference of single heuristic
→ ADDITIONAL SELECTION PROBLEM of choosing a HEURISTIC
→ INCREASED COMPLEXITY through interactions of heurístics
→ AVOID selection problem through EXHAUSTIVE ENUMERATION

Tip & Tricks in Data Sampling

- Do's and Don'ts
  - Random order sampling? Yes!
  - Sampling with replacement? depends / try!
  - Data splitting: ESSENTIAL!!!!
    - Training & Validation for identification, parameterisation & selection
    - Testing for ex ante evaluation (ideally multiple ways / origins!)



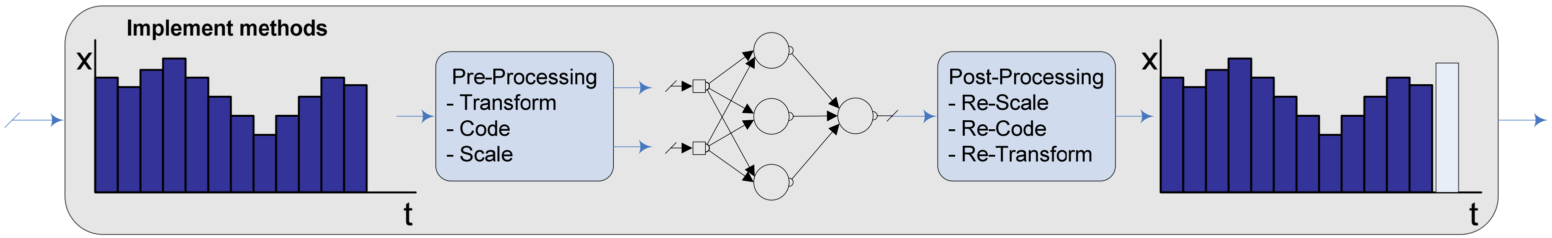


→ Simulation Experiments

Data Preprocessing

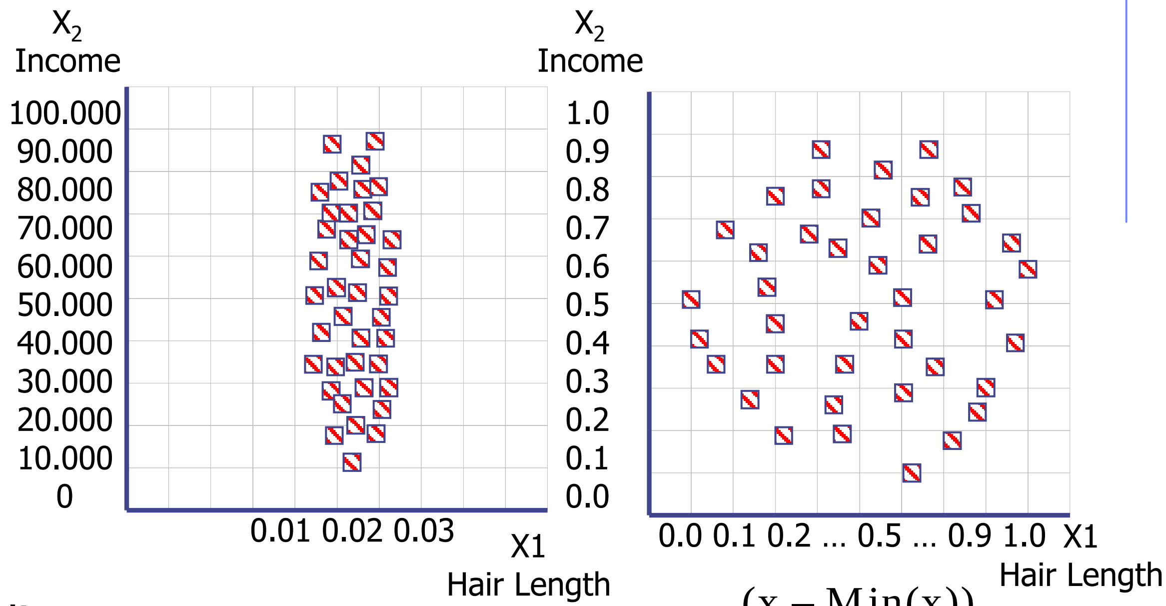
- Data Transformation
  - Verification, correction & editing (data entry errors etc.)
  - Coding of Variables
  - Scaling of Variables
  - Selection of independent Variables (PCA)
  - Outlier removal
  - Missing Value imputation
- Data Coding
  - Binary coding of external events → binary coding
  - n and n-1 coding have no significant impact, n-coding appears to be more robust (despite issues of multicollinearity)

→ Modification of Data to enhance accuracy & speed



Data Preprocessing – Variable Scaling

- Scaling of variables



  - Linear interval scaling $y = ILower + \left(IUpper - ILower\right)\dfrac{(x - Min(x))}{Max(x) - Min(x)}$

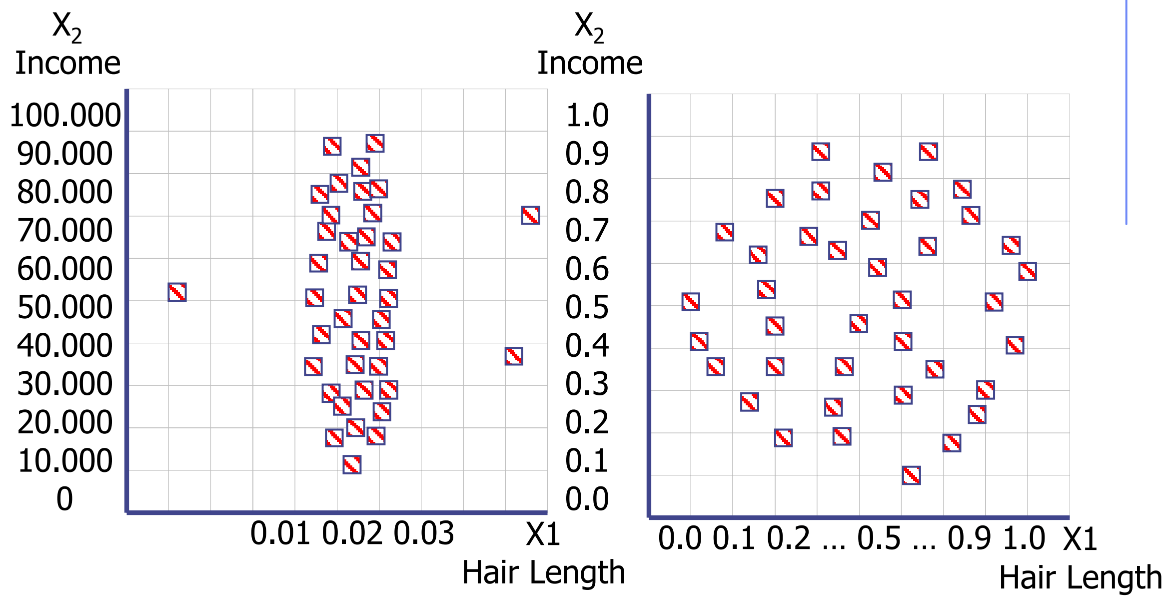
  - Intervall features, e.g. „turnover" [28.12 ; 70; 32; 25.05 ; 10.17 …]
    Linear Intervall scaling to taget intervall, e.g. [-1;1]

$$\text{eg. } x = 72 \quad Max(x) = 119.95 \quad Min(x) = 0 \quad \text{Target [-1;1]}$$

$$y = -1 + (1-(-1))\frac{(72-0)}{119.95-0} = -1 + \frac{144}{119.95} = 0.2005$$

Data Preprocessing – Variable Scaling

- Scaling of variables



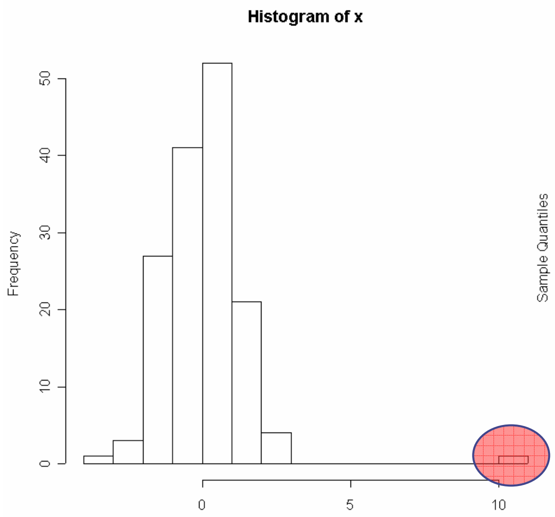
  - Standardisation / Normalisation

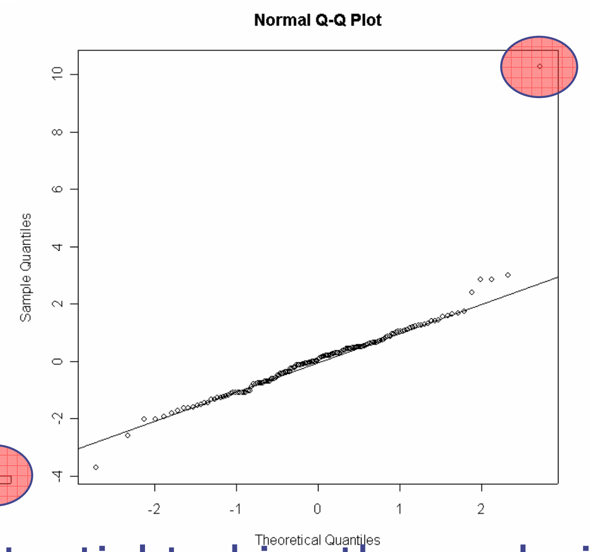
$$y = \frac{x - \eta}{\sigma}$$

- Attention: Interaction of interval with activation Function
  - Logistic [0;1]
  - TanH [-1;1]

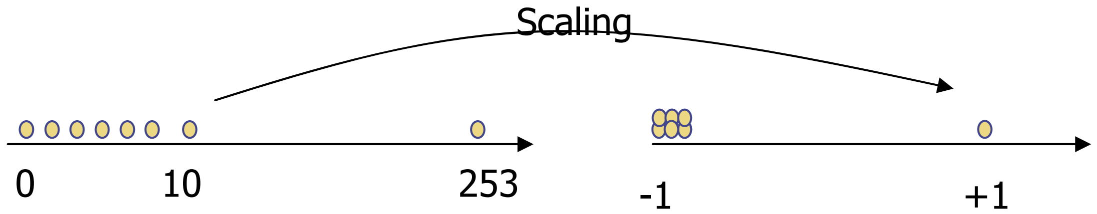
Data Preprocessing – Outliers

- Outliers
  - extreme values
  - Coding errors
  - Data errors





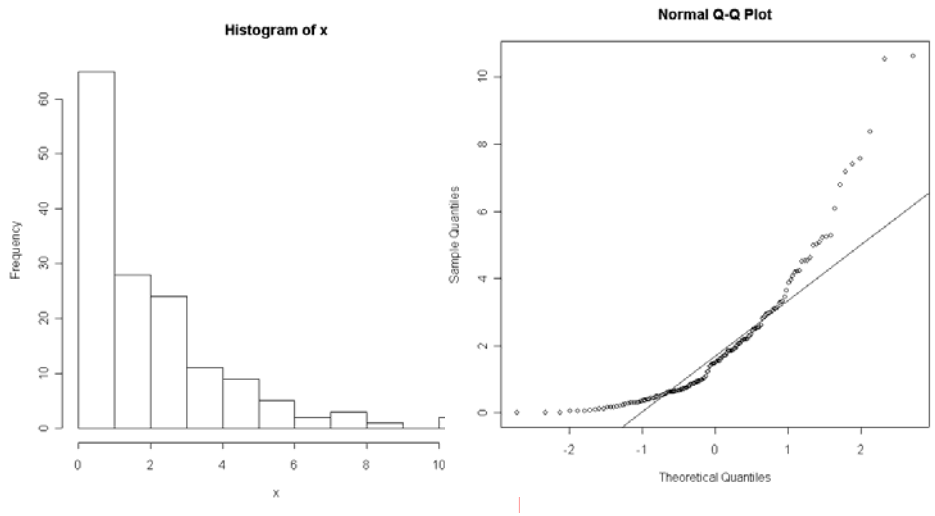
- Outlier impact on scaled variables → potential to bias the analysis
  - Impact on linear interval scaling (no normalisation / standardisation)



- Actions
  - → Eliminate outliers (delete records)
  - → replace / impute values as missing values
  - → Binning of variable = rescaling
  - → Normalisation of variables = scaling

Data Preprocessing – Skewed Distributions

- Asymmetry of observations
- ...



→ Transform data
  - Transformation of data (functional transformation of values)
  - Linearization or Normalisation

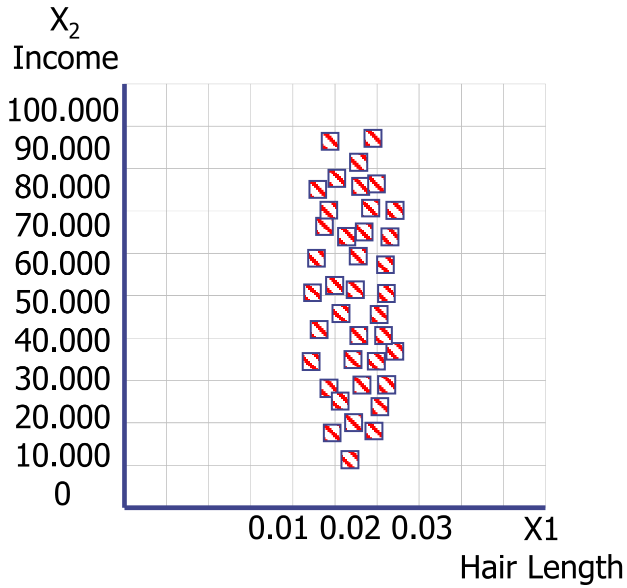
→ Rescale (DOWNSCALE) data to allow better analysis by
  - Binning of data (grouping of data into groups) → ordinal scale!

Data Preprocessing – Data Encoding

- Downscaling & Coding of variables
  - metric variables → create bins/buckets of ordinal variables (=BINNING)
    - Create buckets of equaly spaced intervalls
    - Create bins if Quantile with equal frequencies
  - ordinal variable of *n* values
    → rescale to *n* or *n*-1 nominal binary variables
  - nominal Variable of *n* values, e.g. {Business, Sports & Fun, Woman}
    → Rescale to *n* or *n-1* binary variables
      - 0 = Business Press
      - 1 = Sports & Fun
      - 2 = Woman
    - Recode as 1 of N Coding → 3 new bit-variables
      - 1 0 0 → Business Press
      - 0 1 0 → Sports & Fun
      - 0 0 1 → Woman
    - Recode 1 of N-1 Coding → 2 new bit-variables
      - 1 0 → Business Press
      - 0 1 → Sports & Fun
      - 0 0 → Woman

Data Preprocessing – Impute Missing Values

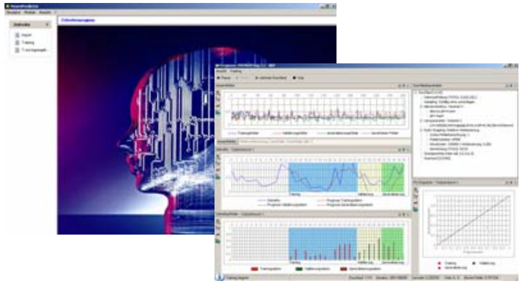
- Missing Values
  - missing feature value for instance
  - some methods interpret " " as 0!
  - Others create special class for missing
  - ...



- Solutions
  - Missing value of interval scale → mean, median, etc.
  - Missing value of nominal scale → most prominent value in feature set

Tip & Tricks in Data Pre-Processing

- Do’s and Don'ts
  - De-Seasonalisation? NO! (maybe … you can try!)
  - De-Trending / Integration? NO / depends / preprocessing!
  - Normalisation? Not necessarily → correct outliers!
  - Scaling Intervals [0;1] or [-1;1]? Both OK!
  - Apply headroom in Scaling? YES!
  - Interaction between scaling & preprocessing? limited
  - …



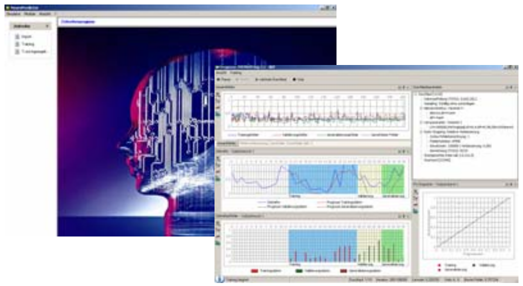
→ Simulation Experiments

Outlier correction in Neural Network Forecasts?

- Outlier correction? YES!

- Neural networks are often characterized as
  - Fault tolerant and robust
  - Showing graceful degradation regarding errors

  → Fault tolerance = outlier resistance in time series prediction?



→ Simulation Experiments

- Number of OUTPUT nodes
  - Given by problem domain!
- Number of HIDDEN LAYERS
  - 1 or 2 … depends on Information Processing in nodes
  - Also depends on nonlinearity & continuity of time series
- Number of HIDDEN nodes
  - Trial & error … sorry!

- Information processing in Nodes (Act. Functions)
  - Sig-Id
  - Sig-Sig (Bounded & additional nonlinear layer)
  - TanH-Id
  - TanH-TanH (Bounded & additional nonlinear layer)

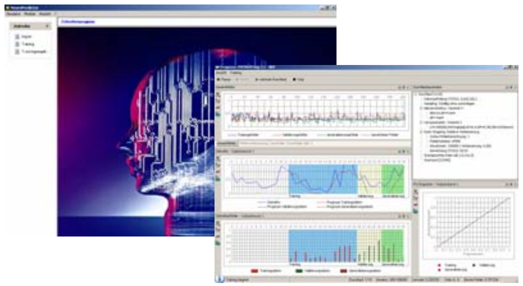
- Interconnection of Nodes
  - ???

Tip & Tricks in Architecture Modelling

- Do’s and Don'ts
  - Number of input nodes? DEPENDS! → use linear ACF/PACF to start!
  - Number of hidden nodes? DEPENDS! → evaluate each time (few)
  - Number of output nodes? DEPENDS on application!

  - fully or sparsely connected networks? ???
  - shortcut connections? ???

  - activation functions → logistic or hyperbolic tangent? TanH !!!
  - activation function in the output layer? TanH or Identity!
  - ...



→ **Simulation Experiments**

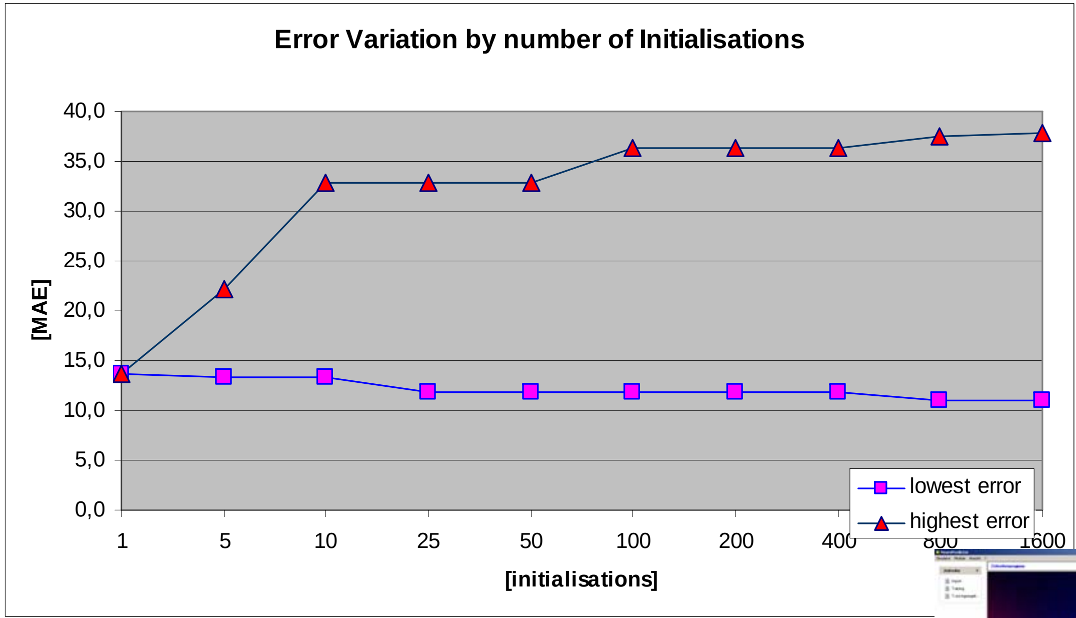
Agenda

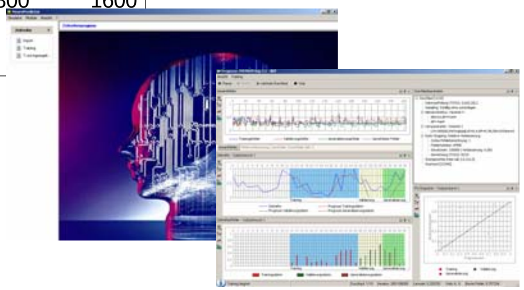
Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
    1. NN models for Time Series & Dynamic Causal Prediction
    2. NN experiments
    3. Process of NN modelling
        1. Preprocessing
        2. Modelling NN Architecture
        3. Training
        4. Evaluation & Selection
4. How to write a good Neural Network forecasting paper!

Tip & Tricks in Network Training

- Do's and Don'ts
  - Initialisations? A MUST! Minimum 5-10 times!!!

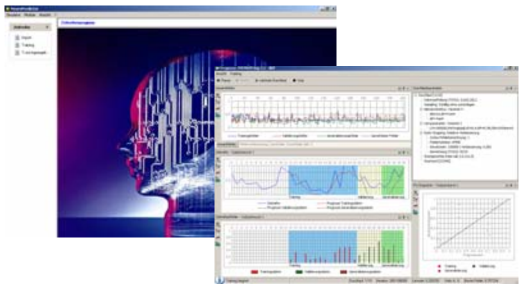




→ **Simulation Experiments**

Tip & Tricks in Network Training & Selection

- Do's and Don'ts
  - Initialisations? A MUST! Minimum 5-10 times!!!
  - Selection of Training Algorithm? Backprop OK, DBD OK …
    … not higher order methods!
  - Parameterisation of Training Algorithm? DEPENDS on dataset!
  - Use of early stopping? YES – carefull with stopping criteria!
  - …
  - Suitable Backpropagation training parameters (to start with)
    - Learning rate 0.5 (always <1!)
    - Momentum 0.4
    - Decrease learning rate by 99%
  - Early stopping on composite error of Training & Validation



→ **Simulation Experiments**

Agenda

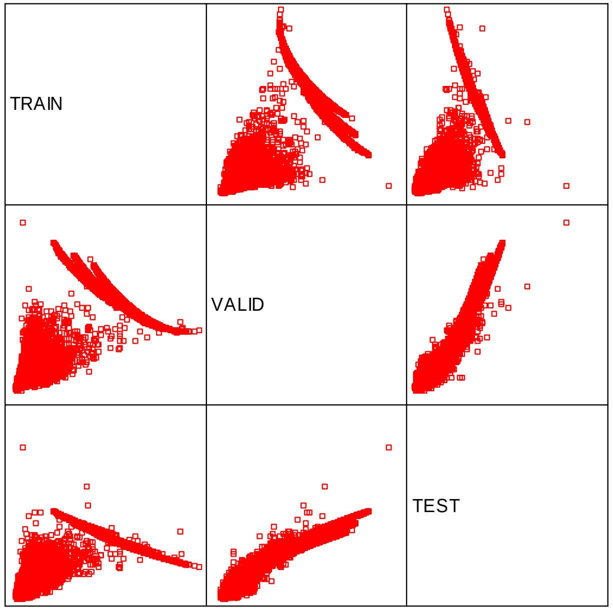
Forecasting with Artificial Neural Networks

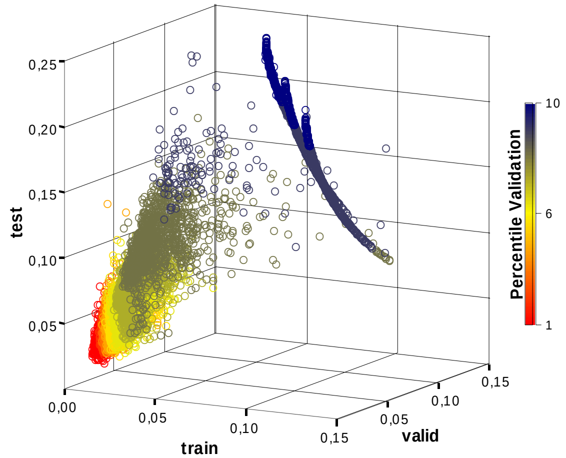
1. Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
    1. NN models for Time Series & Dynamic Causal Prediction
    2. NN experiments
    3. Process of NN modelling
        1. Preprocessing
        2. Modelling NN Architecture
        3. Training
        4. Evaluation & Selection
4. How to write a good Neural Network forecasting paper!

Experimental Results

- Experiments ranked by validation error

Rank by valid-error	Data Set Errors			
	Training	Validation	Test	ANN ID
overall lowest	0,009207	0,011455	0,017760	
overall highest	0,155513	0,146016	0,398628	
1st	0,010850	0,011455	0,043413	39 (3579)
2nd	0,009732	0,012093	0,023367	10 (5873)
…	…	…	…	…
25th	0,009632	0,013650	0,025886	8 (919)
…	…	…	…	…
14400th	0,014504	0,146016	0,398628	33 (12226)





→ significant positive correlations
- training & validation set
- validation & test set
- training & test set

□ inconsistent errors by selection criteria
- low validation error → high test error
- higher validation error → lower test error

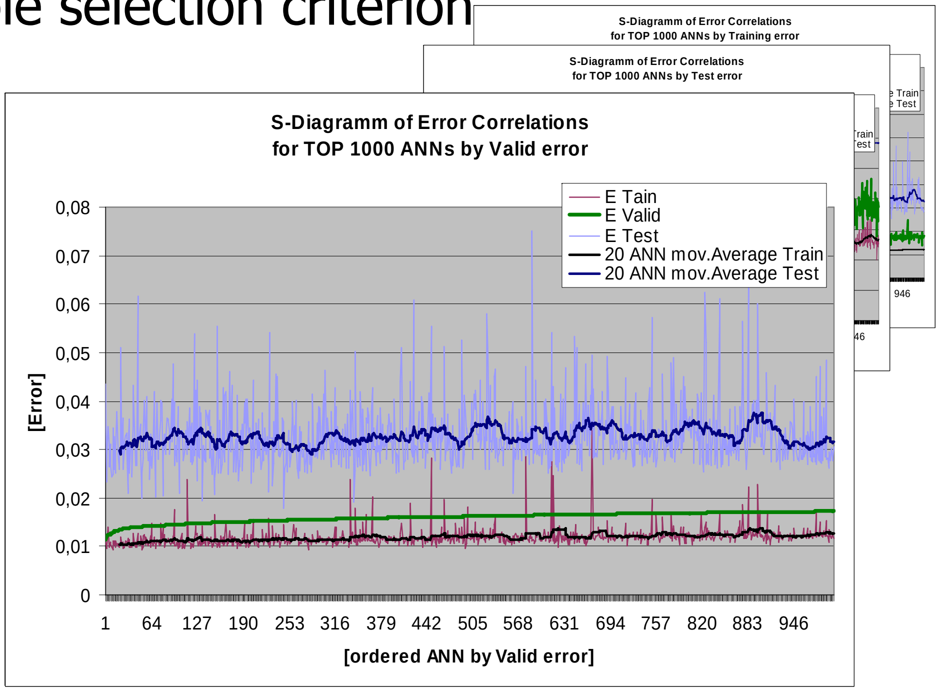
Problem: Validation Error Correlations

- Correlations between dataset errors

	Correlation between datasets		
Data included	**Train - Validate**	**Validate - Test**	**Train - Test**
14400 ANNs	0,7786**	0,9750**	0,7686**
top 1000 ANNs	0,2652**	0,0917**	0,4204**
top 100 ANNs	0,2067**	0,1276**	0,4004**

→ validation error is questionable selection criterion

- decreasing correlation
- high variance on test error
- same results ordered by training & test error



- ***Desirable properties of an Error Measure:***
  - summarizes the cost consequences of the errors
  - Robust to outliers
  - Unaffected by units of measurement
  - Stable if only a few data points are used

Fildes,IJF, 92, Armstrong and Collopy, IJF, 92;
Hendry and Clements, Armstong and Fildes, JOF, 93,94

Model Evaluation through Error Measures

- forecasting *k* periods ahead we can assess the forecast quality using a holdout sample

- Individual forecast error
  - e_{t+k}= Actual - Forecast

$$e_t = y_t - F_t$$

- Mean error (ME)
  - Add individual forecast errors
  - As positive errors cancel out negative errors, the ME should be approximately zero for an unbiased series of forecast

$$ME_t = \frac{1}{n}\sum_{k=1}^{n} Y_{t+k} - F_{t+k}$$

- Mean squared error (MSE)
  - Square the individual forecast errors
  - Sum the squared errors and divide by n

$$MSE_t = \frac{1}{n}\sum_{k=1}^{n}\left(Y_{t+k} - F_{t+k}\right)^2$$

Model Evaluation through Error Measures

→avoid cancellation of positive v negative errors: absolute errors

- Mean absolute error (MAE)
  - Take absolute values of forecast errors
  - Sum absolute values and divide by n

$$MAE = \frac{1}{n}\sum_{k=1}^{n}\left|Y_{t+k} - F_{t+k}\right|$$

- Mean absolute percent error (MAPE)
  - Take absolute values of percent errors
  - Sum percent errors and divide by n

$$MAPE = \frac{1}{n}\sum_{k=1}^{n}\left|\frac{Y_{t+k} - F_{t+k}}{Y_{t+k}}\right|$$

→This summarises the forecast error over different lead-times

→May need to keep *k* fixed depending on the decision to be made based on the forecast:

$$MAE(k) = \frac{1}{(n-k+1)}\sum_{t=T}^{T+n-k}\left|Y_{t+k} - F_t(k)\right| \qquad MAPE(k) = \frac{1}{(n-k+1)}\sum_{t=T}^{T+n-k}\left|\frac{Y_{t+k} - F_t(k)}{Y_{t+k}}\right|$$

Selecting Forecasting Error Measures

- *MAPE* & *MSE* are subject to upward bias by single bad forecast
- Alternative measures may are based on median instead of mean

- Median Absolute Percentage Error
  - median = middle value of a set of errors *sorted in ascending order*
  - If the *sorted* data set has an even number of elements, the median is the average of the two middle values

$$MdAPE_f = \mathrm{Med}\left(\left|\frac{e_{f,t}}{y_t}\right| \times 100\right)$$

- Median Squared Error

$$MdSE_f = \mathrm{Med}\left(e_{f,t}^2\right)$$

Evaluation of Forecasting Methods

- The *Base Line* model in a forecasting competition is the Naïve 1a **No Change** model → use as a benchmark

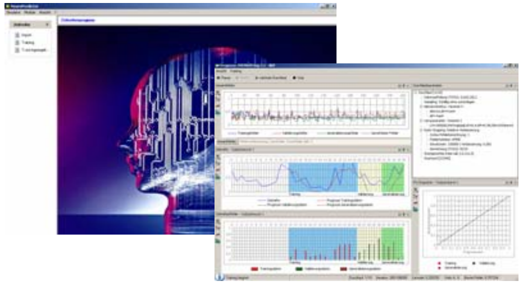
$$\hat{y}_{t+f|t} = y_t$$

- Theil's *U* statistic allows us to determine whether our forecasts outperform this base line, with increased accuracy trough our method (outperforms naïve) if *U* < 1

$$U = \sqrt{\frac{\sum \left(\frac{\left(\hat{y}_{t+f|t} - y_{t+f}\right)}{y_t} \right)^2}{\sum \left(\frac{\left(y_t - y_{t+f}\right)}{y_t} \right)^2}}$$

Tip & Tricks in Network Selection

- Do's and Don'ts
  - Selection of Model with lowest Validation error? NOT VALID!
  - Model & forecasting competition? Always multiple origin etc.!
  - ...



→ **Simulation Experiments**

Agenda

Forecasting with Artificial Neural Networks

1. Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks …
    1. NN models for Time Series & Dynamic Causal Prediction
    2. NN experiments
    3. Process of NN modelling
4. How to write a good Neural Network forecasting paper!

How to evaluate NN performance

Valid Experiments

- Evaluate using ex ante accuracy (HOLD-OUT data)
  - Use training & validation set for training & model selection
  - NEVER!!! Use test data except for final evaluation of accuracy
- Evaluate across multiple time series
- Evaluate against benchmark methods (NAÏVE + domain!)
- Evaluate using multiple & robust error measures (not MSE!)
- Evaluate using multiple out-of-samples (time series origins)

→ Evaluate as Empirical Forecasting Competition!

Reliable Results

- Document all parameter choices
- Document all relevant modelling decisions in process

→ Rigorous documentation to allow re-simulation through others!

Evaluation through Forecasting Competition

- Forecasting Competition
  - Split up time series data → 2 sets PLUS multiple ORIGINS!
  - Select forecasting model
  - select best parameters for IN-SAMPLE DATA
  - Forecast next values for DIFFERENT HORIZONS t+1, t+3, t+18?
  - Evaluate error on hold out OUT-OF-SAMPLE DATA
  - choose model with lowest AVERAGE error OUT-OF-SAMPLE DATA

- Results → M3-competition
  - simple methods outperform complex ones
  - exponential smoothing OK
    → neural networks not necessary
  - forecasting VALUE depends on
    VALUE of INVENTORY DECISION

Evaluation of Forecasting Methods

- HOLD-OUT DATA → out of sample errors count!

... today | Future ...

... 2003 "today" | presumed | Future ...

Method	Jan	Feb	Mar	Apr	Mai	Jun	Jul	Aug	Sum	Sum
Baseline Sales	90	100	110	?	?	?	?	?		
Method A	90	90	90	90	90	90	90	90		
Method B	110	100	120	100	110	100	110	100		
absolute error AE(A)	0	10	20	?	?	?	?	?	30	?
absolute error AE(B)	20	0	10	?	?	?	?	?	10	?

t+1 t+2 t+3 ...

t+1 t+2 t+3 ...

SIMULATED = EX POST Forecasts

Evaluation of Forecasting Methods

- Different Forecasting horizons, emulate rolling forecast …

… 2003 “today” presumed Future …

Method	Jan	Feb	Mar	Apr	Mai	Jun	Jul	Aug	Sum	Sum
Baseline Sales	90	100	110	100	90	100	110	100		
Method A	90	90	90	90	90	90	90	90		
Method B	110	100	120	100	110	100	110	100		
absolute error AE(A)	0	10	20	10	0	10	20	10	30	50
absolute error AE(B)	20	0	10	0	20	0	0	0	30	20

t+1 t+2 t+3 …

t+1 t+2 …

…

- Evaluate only RELEVANT horizons
  - omit t+2 if irrelevant for planning!

Evaluation of Forecasting Methods

- Single vs. Multiple origin evaluation

… 2003 "today" | presumed Future …

Method	Jan	Feb	Mar	Apr	Mai	Jun	Jul	Aug	Sum	Sum
Baseline Sales	90	100	110	100	90	100	110	100		
Method A	90	90	90	90	90	90	90	90		
Method B	110	100	120	100	110	100	110	100		
absolute error AE(A)	0	10	20	10	0	10	20	10	30	50
absolute error AE(B)	20	0	10	0	20	0	0	0	30	20

B A B …

A

A

B

- Problem of sampling Variability!
  - Evaluate on multiple origins
  - Calculate t+1 error
  - Calculate average of t+1 error

→ GENERALIZE about forecast errors

Software Simulators for Neural Networks

Commercial Software by Price

- High End
  - Neural Works Professional
  - SPSS Clementine
  - SAS Enterprise Miner
- Midprice
  - Alyuda NeuroSolutions
  - NeuroShell Predictor
  - NeuroSolutions
  - NeuralPower
  - PredictorPro
- Research
  - Mathlab Library
  - R-package
  - NeuroLab
- ...

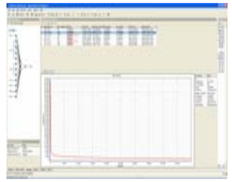
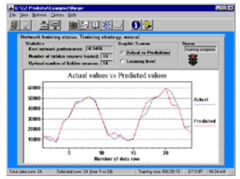
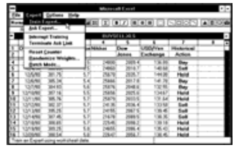
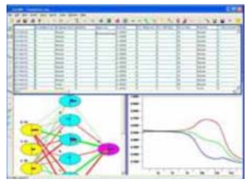
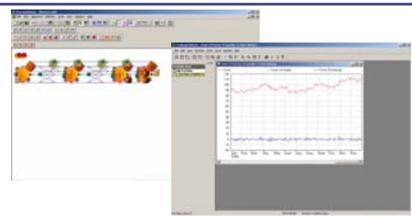
Public Domain Software

- Research oriented
  - SNNS
  - JNNS JavaSNNS
  - JOONE
  - ...

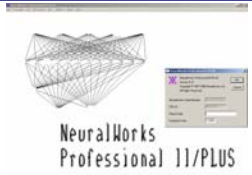
→ **FREE** CD-ROM for evaluation
- Data from Experiments
  - M3-competition
  - airline-data
  - lynx-data
  - beer-data
- Software Simulators

→ **Consider Tashman/Hoover Tables on forecasting Software for more details**

Neural Networks Software - Times Series friendly!

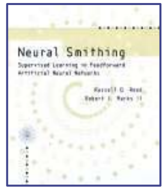
Alyuda Inc. ALYUDA research company	ALYUDA NEUROSIGNAL XL ALYUDA FORECASTER XL ALYUDA NEUROINTELLIGENCE ALYUDA FORECASTER Tradecision	
Ward Systems  "Let your systems learn the wisdom of age and experience"	AITrilogy: NeuroShell Predictor, NeuroShell Classifier, GeneHunter NeuroShell 2, NeuroShell Trader, Pro,DayTrader	
Attrasoft Inc.	Predictor Predictor PRO	
Promised Land PROMISED LAND TECHNOLOGIES, INC.	Braincell	
Neural Planner Inc.	Easy NN Easy NN Plus	
NeuroDimension	NeuroSolutions Cosunsultant Neurosolutions for Excel NeuroSolutions for Mathlab Trading Solutions	

Neural networks Software – General Applications

Neuralware Inc	Neural Works Professional II Plus	
SPSS	SPSS Clementine DataMining Suite	
SAS	SAS Enterprise Miner	
…	…	

Further Information

- **Literature & websites**
  - NN Forecasting website www.neural-forecasting.com or www.bis-lab.com
  - Google web-resources, SAS NN newsgroup FAQ ftp://ftp.sas.com/pub/neural/FAQ.html
  - BUY A BOOK!!! Only one? Get: Reeds & Marks 'Neural Smithing'



- **Journals**
  - Forecasting … rather than technical Neural Networks literature!
    - JBF – Journal of Business Forecasting
    - IJF – International Journal of Forecasting
    - JoF – Journal of Forecasting

- **Contact to Practitioners & Researchers**
  - Associations
    - IEEE NNS – IEEE Neural Network Society
    - INNS & ENNS – International & European Neural Network Society
  - Conferences
    - Neural Nets: IJCNN, ICANN & ICONIP by associations (search google …)
    - Forecasting: IBF & ISF conferences!
  - Newsgroups news.comp.ai.nn
  - Call Experts you know … me ;-)

Agenda

Business Forecasting with Artificial Neural Networks

1. Process of NN Modelling
2. Tips & Tricks for Improving Neural Networks based forecasts

a. Copper Price Forecasting

b. Questions & Answers and Discussion

   a. Advantages & Disadvantages of Neural Networks
   b. Discussion

Advantages … versus Disadvantages!

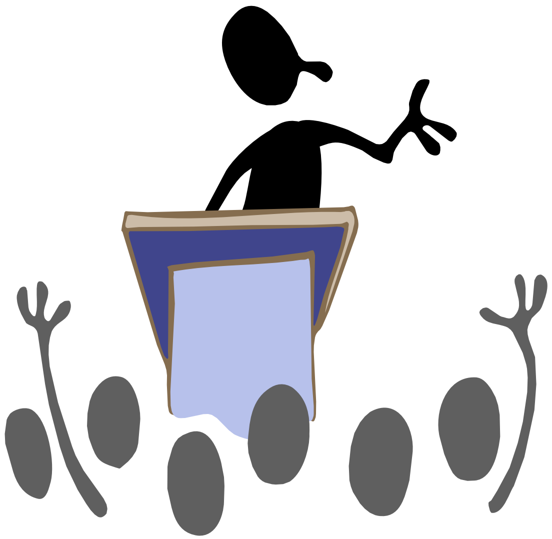
Advantages

- ANN can forecast any time series pattern (t+1!)
  - without preprocessing
  - no model selection needed!
- ANN offer many degrees of freedom in modeling
  - Freedom in forecasting with one single model
  - Complete Model Repository
    - linear models
    - nonlinear models
    - Autoregression models
    - single & multiple regres.
    - Multiple step ahead
    - …

Disadvantages

- ANN can forecast any time series pattern (t+1!)
  - without preprocessing
  - no model selection needed!
- ANN offer many degrees of freedom in modeling
  - Experience essential!
  - Research not consistent
- explanation & interpretation of ANN weights IMPOSSIBLE (nonlinear combination!)
  - impact of events not directly deductible

Questions, Answers & Comments?



Sven F. Crone
crone@bis-lab.de

SLIDES & PAPERS availble:
www.bis-lab.de

www.lums.lancs.ac.uk

Summary Day I

- ANN can forecast any time series pattern (t+1!)
  - without preprocessing
  - no model selection needed!
- ANN offer many degrees of freedom in modeling
  - Experience essential!
  - Research not consistent

What we can offer you:

- NN research projects with complimentary support!
- Support through MBA master thesis in mutual projects

Contact Information

Sven F. Crone
Research Associate

Lancaster University Management School
Department of Management Science, Room C54
Lancaster LA1 4YX
United Kingdom

Tel +44 (0)1524 593867
Tel +44 (0)1524 593982 direct
Tel +44 (0)7840 068119 mobile
Fax +44 (0)1524 844885

Internet www.lums.lancs.ac.uk
eMail s.crone@lancaster.ac.uk