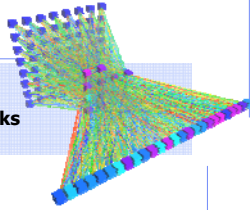


Forecasting with Artificial Neural Networks



EVIC 2005 Tutorial
Santiago de Chile, 15 December 2005

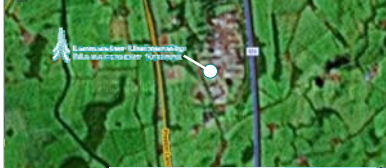
→ slides on www.neural-forecasting.com



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



Lancaster University Management School?



What you can expect from this session ...

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

~~$$E_p = C(U_p, \sigma_p) \quad \sigma_p = f_j(\text{net}_p) \quad \Delta_j \propto \frac{\partial C(U_p, \sigma_p)}{\partial w_j}$$

$$\frac{\partial C(U_p, \sigma_p)}{\partial w_j} = \frac{\partial C(U_p, \sigma_p)}{\partial \text{net}_p} \frac{\partial \text{net}_p}{\partial w_j}$$

$$\delta_p = \frac{\partial C(U_p, \sigma_p)}{\partial \text{net}_p}$$

$$\frac{\partial \text{net}_p}{\partial w_j} = f_j'(\text{net}_p)$$

$$\delta_p = \frac{\partial C(U_p, \sigma_p)}{\partial w_j} \frac{1}{f_j'(\text{net}_p)}$$

$$\frac{\partial C(U_p, \sigma_p)}{\partial w_j} = \sum_{\mu} \frac{\partial C(U_p, \sigma_p)}{\partial \text{net}_\mu} \frac{\partial \text{net}_\mu}{\partial w_j}$$

$$= \sum_{\mu} \frac{\partial C(U_p, \sigma_p)}{\partial \text{net}_\mu} w_{\mu j} = - \sum_{\mu} \delta_\mu w_{\mu j}$$

$$\delta_j = f_j'(\text{net}_j) \sum_{\mu} \delta_\mu w_{\mu j}$$~~

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

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

→ CD-Start-Up Kit for Neural Net Forecasting

- 20+ software simulators
- datasets
- literature & faq

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?

Data Mining

TASKS	
Descriptive Data Mining	Predictive Data Mining
Categorisation: Clustering -> K-means Clustering -> Neural networks	Classification -> Decision trees -> Logistic regress. -> Neural networks -> Discriminant Analysis
Summarisation & Visualisation -> Feature Selection -> Princip. Component Analysis -> K-means Clustering -> Class Entropy	Regression -> Linear Regression -> Nonlinear Regres. -> Neural networks -> MLP, RBFN, GDM
Association Analysis -> Association rules -> Link Analysis	Time Series Analysis -> Exponential smoothing -> (SARIMA(x)) -> Neural networks
Sequence Discovery -> Temporal association rules	

ALGORITHMS

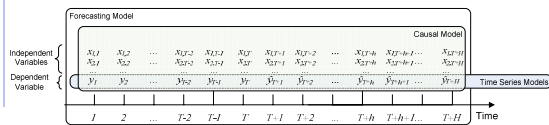
Agenda

Forecasting with Artificial Neural Networks

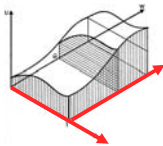
1. Forecasting?
 1. Forecasting as predictive Regression
 2. Time series prediction vs. causal prediction
 3. SARIMA-Modelling
 4. Why NN for Forecasting?
2. Neural Networks?
3. Forecasting with Neural Networks ...
4. How to write a good Neural Network forecasting paper!

Forecasting Models

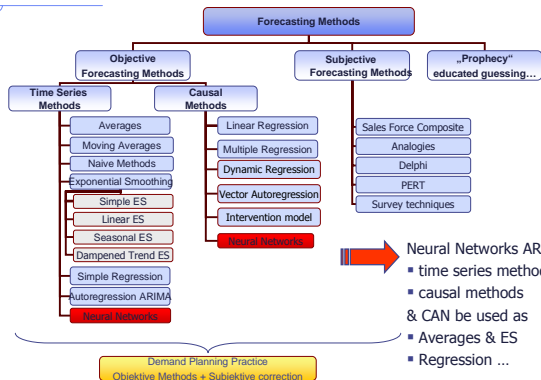
- Time series analysis vs. causal modelling



- 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

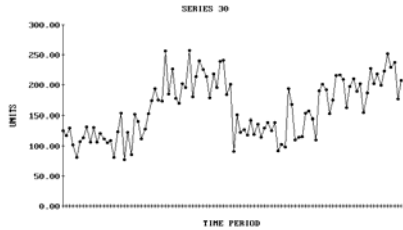


Classification of Forecasting Methods



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, $t=1,2 \dots n$



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 corrected MANUALLY

Components of Time Series

Time Series

• Time Series → decomposed into Components

Time Series Patterns

Time Series Pattern

REGULAR Time Series Patterns

IRREGULAR Time Series Patterns

STATIONARY Time Series

SEASONAL Time Series

TRENDED Time Series

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%)

+ PULSES!
+ LEVEL SHIFTS!
+ STRUCTURAL BREAKS!

Combination of individual Components
 $Y_t = f(S_t, T_t, E_t)$

Components of complex Time Series

Sales or observation of time series at point t $\Rightarrow Y_t$

consists of a combination of $f(\)$

Base Level + Seasonal Component $\Rightarrow S_t$

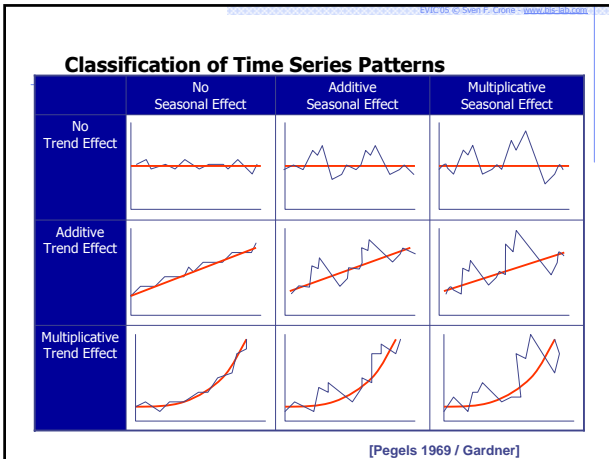
Trend-Component $\Rightarrow T_t$

Irregular or random Error-Component $\Rightarrow E_t$

Different possibilities to combine components

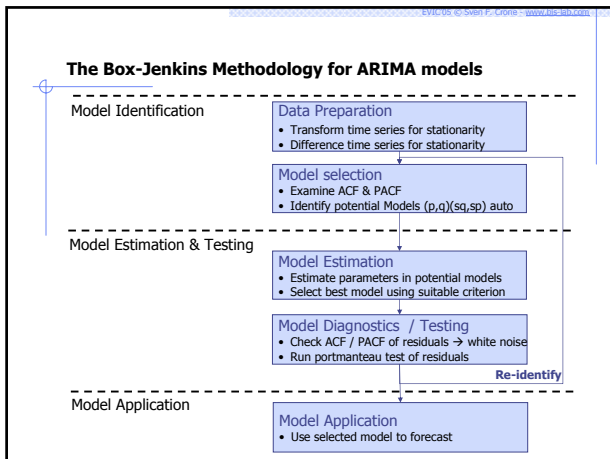
Additive Model
 $Y_t = L + S_t + T_t + E_t$

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



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



ARIMA-Modelling

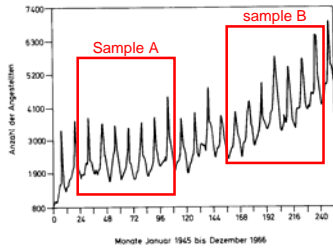
- ARIMA(p,d,q)-Models
 - ARIMA - Autoregressive Terms AR(p), with p=order of the autoregressive part
 - ARIMA - Order of Integration, d=degree of first differencing/integration involved
 - ARIMA - Moving Average Terms MA(q), with q=order of the moving average of error
 - SARIMA_s (p,d,q)(P,D,Q) with 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 $\lambda_{t,t-\tau} = \text{cov}(Y_t, Y_{t-\tau})$
 - $\lambda_{t,t} = \text{var}(Y_t)$
- Definition
 - A time series is *stationary* if its mean level μ_t is constant for all *t* and its variance and covariances $\lambda_{t,t-\tau}$ are constant for all *t*
 - In other words:
 - all properties of the distribution (mean, variance, 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

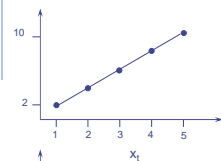


Stationarity:
 $\mu(A) = \mu(B)$
 $\text{var}(A) = \text{var}(B)$
 etc.

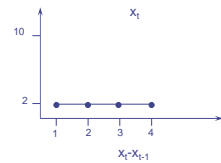
this time series:
 $\mu(B) > \mu(A) \rightarrow$ trend
 \rightarrow instationary time series

ARIMA-Modells: Differencing for Stationary

- Differencing time series



- 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:
 - $4-2=2$
 - $6-4=2$
 - $8-6=2$
 - $10-8=2$



- \rightarrow The new time series $\Delta Y_t = \{2, 2, 2, 2\}$ is stationary through 1st differencing
- $\rightarrow d=1 \rightarrow$ ARIMA (0,1,0) model
- \rightarrow 2nd order differences: $d=2$

ARIMA-Modells: Differencing for Stationary

- Integration**

- Differencing

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

- Transforms: Logarithms etc.

...

- Where Z_t is a transform of the variable of interest Y_t chosen to make $Z_t, Z_{t-1}, (Z_{t-1}, Z_{t+2}), \dots$ 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}

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

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

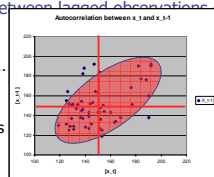
ARIMA-Modells: Parameter p of Autocorrelation

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

$$\rho_k = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

denotes the correlation between lagged observations of distance k

- Graphical interpretation ...
 - Uncorrelated data has low autocorrelations
 - Uncorrelated data shows no correlation pattern
 - ...



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

$r_3 = .15$

ACF

→ Autocorrelations r_k gathered at lags 1, 2, ... make up the autocorrelation function (ACF)

ARIMA-Modells – Autoregressive Terms

Identification of AR-terms ...?

NORM

AUTO

- Random independent observations

- An AR(1) process?

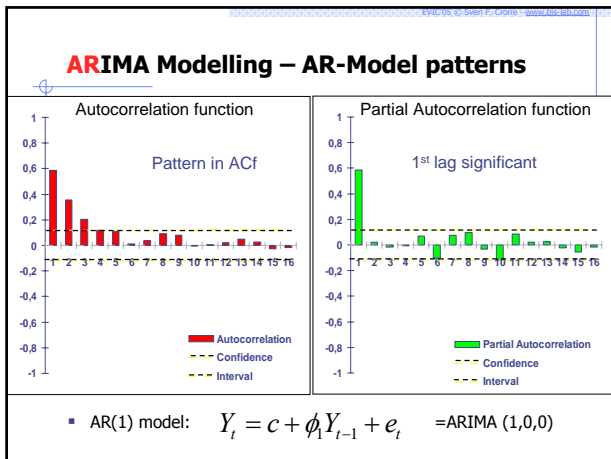
ARIMA-Modells: Partial Autocorrelations

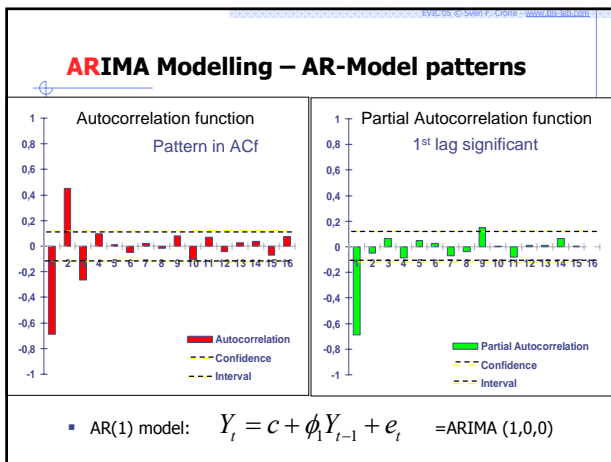
Partial Autocorrelations are used to measure the degree of association between Y_t and Y_{t+k} when the effects of other time lags $1, 2, 3, \dots, k-1$ are removed

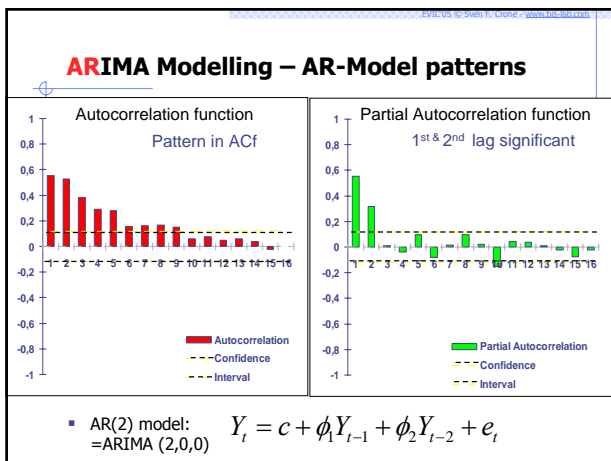
- Significant AC between Y_t and Y_{t-1}
 - significant AC between Y_{t-1} and Y_{t-2}
 - induces correlation between Y_t and Y_{t-2} ! (1st AC = PAC!)
- When fitting an AR(p) model to the time series, the last coefficient p of Y_{t-p} measures the excess correlation at lag p which is not accounted for by an AR(p-1) model. π_p is called the p th order partial autocorrelation, i.e.

$$\pi_p = \text{corr}(Y_t, Y_{t-p} | Y_{t-1}, Y_{t-2}, \dots, Y_{t-p+1})$$
- Partial Autocorrelation coefficient measures true correlation at Y_{t-p}

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \pi_p Y_{t-p} + V_t$$







ARIMA Modelling – Moving Average Prozesse

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

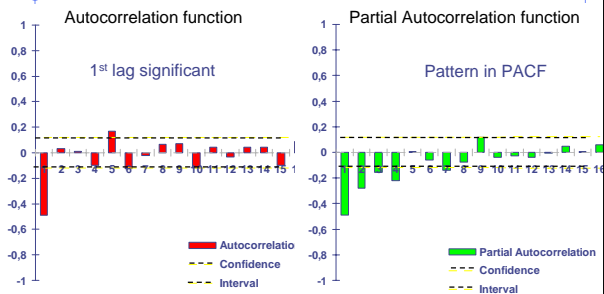
$$Y_t = c + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \phi_q e_{t-q}$$

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

for $q = 1, -1 < \theta_1 < 1$

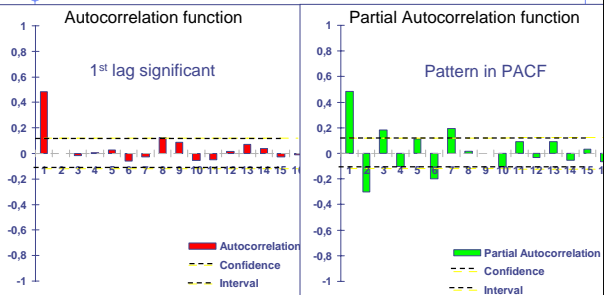
$q = 2, -1 < \theta_2 < 1 \wedge \theta_2 + \theta_1 < 1 \wedge \theta_2 - \theta_1 < 1$

ARIMA Modelling – MA-Model patterns

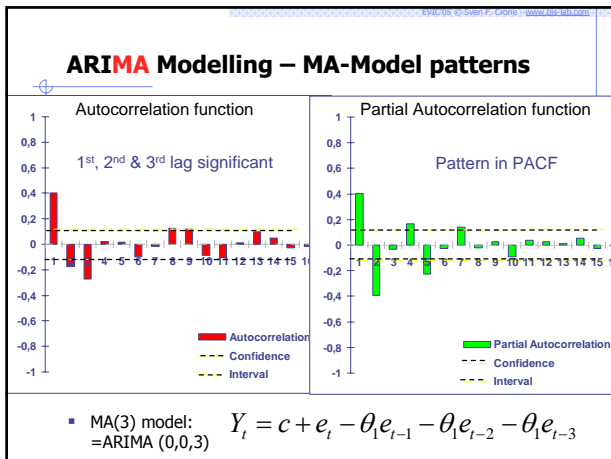


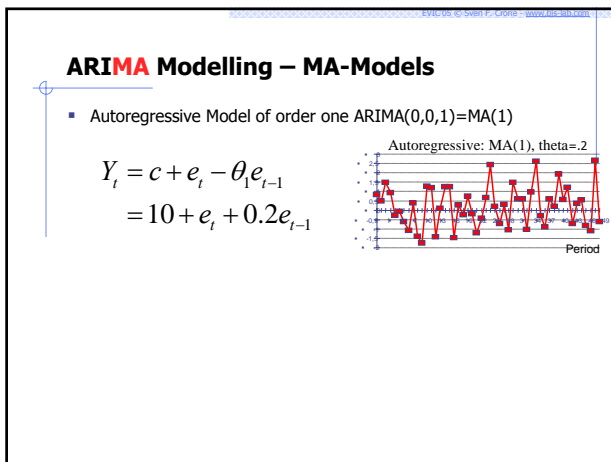
- MA(1) model: $Y_t = c + e_t - \theta_1 e_{t-1}$ =ARIMA (0,0,1)

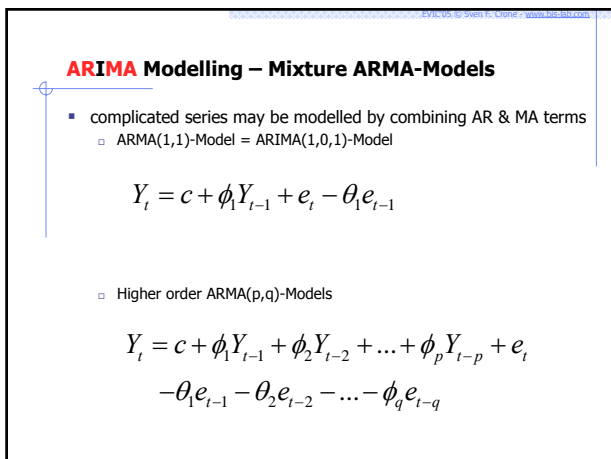
ARIMA Modelling – MA-Model patterns

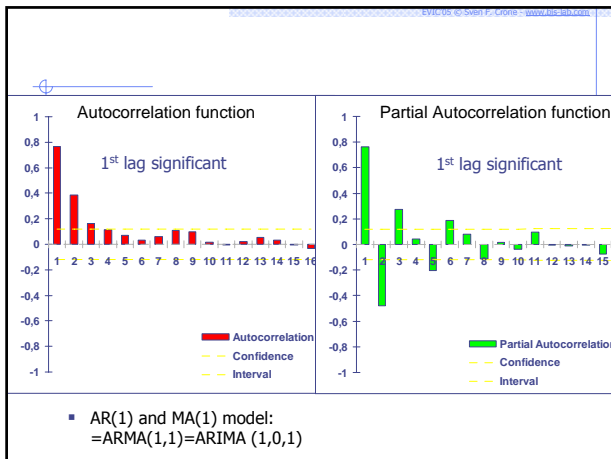


- MA(1) model: $Y_t = c + e_t - \theta_1 e_{t-1}$ =ARIMA (0,0,1)







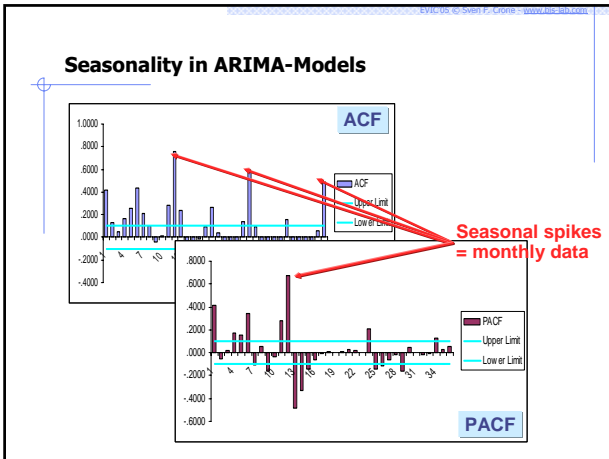


- ### Agenda
- #### Forecasting with Artificial Neural Networks
- Forecasting?
 - Forecasting as predictive Regression
 - Time series prediction vs. causal prediction
 - SARIMA-Modelling
 - SARIMA – Differencing
 - SARIMA – Autoregressive Terms
 - SARIMA – Moving Average Terms
 - SARIMA – Seasonal Terms
 - SARIMAX – Seasonal ARIMA with Interventions
 - Why NN for Forecasting?
 - Neural Networks?
 - Forecasting with Neural Networks ...
 - 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.
 - $t-12$ & $t-13$ for yearly
 - $t-4$ & $t-5$ 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

- Extension of Notation of Backshift Operator

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

- Seasonal difference followed by a first difference: $(I - B)(I - B^s) Y_t$

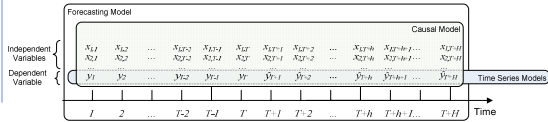
$$(1 - \phi_1 B)(1 - \Phi_1 B^4)(1 - B)(1 - B^4) Y_t = c + (1 - \theta_1 B)(1 - \Theta_1 B^4) e_t$$

Non-seasonal AR(1) Seasonal AR(1) Non-seasonal difference Seasonal difference Non-seasonal MA(1) Seasonal MA(1)

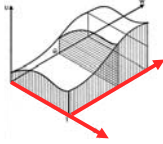
- ### Agenda
- Forecasting with Artificial Neural Networks
- Forecasting?
 - Forecasting as predictive Regression
 - Time series prediction vs. causal prediction
 - SARIMA-Modelling
 - SARIMA – Differencing
 - SARIMA – Autoregressive Terms
 - SARIMA – Moving Average Terms
 - SARIMA – Seasonal Terms
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

Forecasting Models

- Time series analysis vs. causal modelling

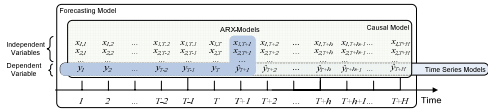


- Time series prediction (Univariate)
 - Assumes that data generating process that creates patterns can be explained only from previous observations of dependent variable
- Causal prediction (Multivariate)
 - Data generating process can be explained by interaction of causal (cause-and-effect) independent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

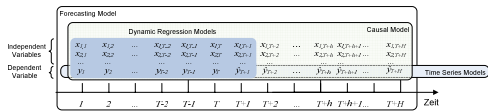


Causal Prediction

- ARX(p)-Models



- General Dynamic Regression Models



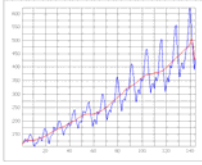
Agenda

Forecasting with Artificial Neural Networks

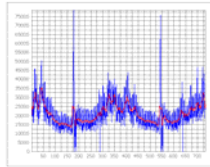
- Forecasting?
 - Forecasting as predictive Regression
 - Time series prediction vs. causal prediction
 - SARIMA-Modelling
 - Why NN for Forecasting?
- Neural Networks?
- Forecasting with Neural Networks ...
- How to write a good Neural Network forecasting paper!

Why forecast with NN?

- Pattern or noise?



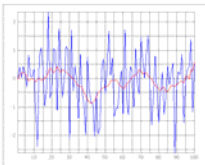
→ Airline Passenger data
→ Seasonal, trended
→ Real "model" disagreed:
multiplicative seasonality
or additive seasonality
with level shifts?



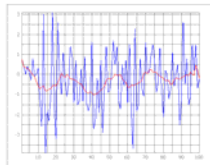
→ Fresh products
supermarket Sales
→ Seasonal, events,
heteroscedastic noise
→ Real "model" unknown

Why forecast with NN?

- Pattern or noise?



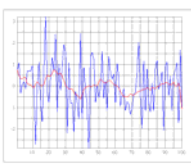
→ Random Noise iid
(normally distributed:
mean 0; std.dev. 1)



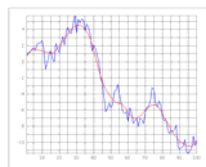
→ BL(p,q) Bilinear
Autoregressive Model
 $y_t = 0.7y_{t-1}\varepsilon_{t-2} + \varepsilon_t$

Why forecast with NN?

- Pattern or noise?



→ TAR(p) Threshold
Autoregressive model
 $y_t = 0.9y_{t-1} + \varepsilon_t$ for $|y_{t-1}| \leq 1$
 $= -0.3y_{t-1} - \varepsilon_t$ for $|y_{t-1}| > 1$



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

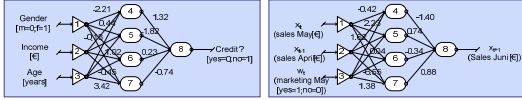
Motivation for NN in Forecasting – Nonlinearity!

- 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
 - ...

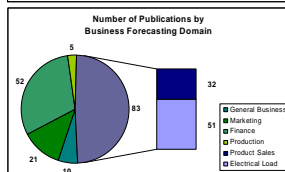
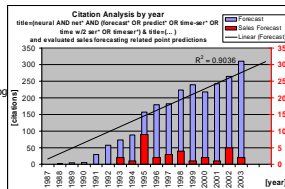
→ One SINGLE network architecture → MANY applications

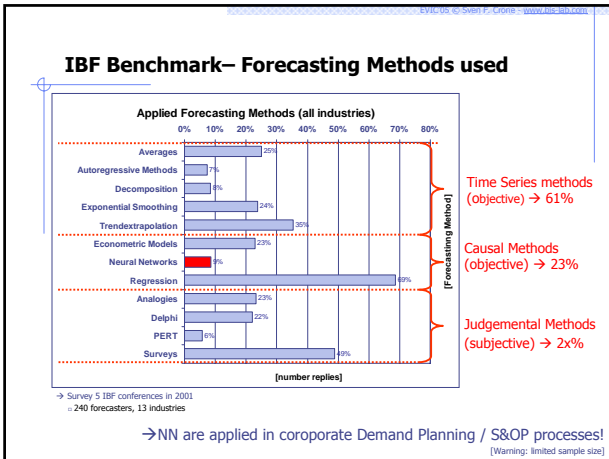


Applications of Neural Nets in diverse Research Fields

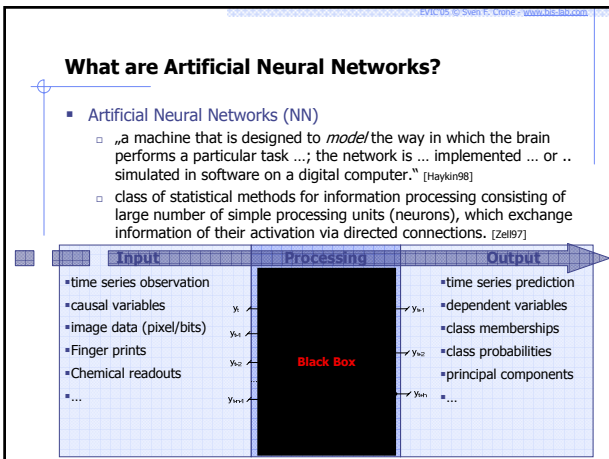
→ 2500+ journal publications on NN & Forecasting alone!

- Neurophysiology
 - simulate & explain brain
 - Informatics
 - eMail & url filtering
 - VirusScan (Symantec Norton Antivirus)
 - Speech Recognition & Optical Character Recognition
 - 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!



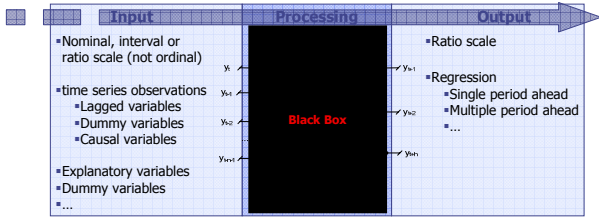


- ### Agenda
- #### Forecasting with Artificial Neural Networks
1. Forecasting?
 2. Neural Networks?
 1. What are NN? Definition & Online Preview ...
 2. Motivation & brief history of Neural Networks
 3. From biological to artificial Neural Network Structures
 4. Network Training
 3. Forecasting with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



What are 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



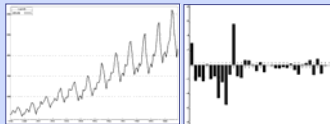
DEMO: Preview of Neural Network Forecasting

- 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 $t, t-1, \dots, t-11$ (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“

The screenshot shows the NeuraLab Predict! software interface with several key components highlighted in red boxes:

- Errors on training / validation / test dataset:** A line graph showing the performance of the model on different datasets.
- Time Series versus Neural Network Forecast:** A plot comparing actual observations (blue line) with the neural network forecast (green line). It includes labels for "in sample observations & forecasts training" and "out of sample = Test".
- Absolute Forecasting Errors:** A plot showing the absolute errors of the forecasts.
- PQ-diagramm:** A plot showing the relationship between the time series actual value and the NN forecasted value.
- Time series actual value:** A plot showing the actual values of the time series.
- NN forecasted value:** A plot showing the values forecasted by the neural network.

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!

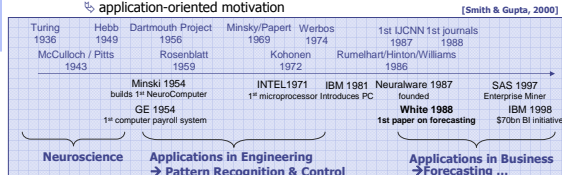
- Human & other nervous systems (animals, insects → e.g. bats)
 - Ability of various complex functions: perception, motor control, pattern recognition, classification, prediction etc.
 - Speed: e.g. detect & recognize changed face in crowd=100-200ms
 - Efficiency etc.
- brains are the most efficient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



- ↳ 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.

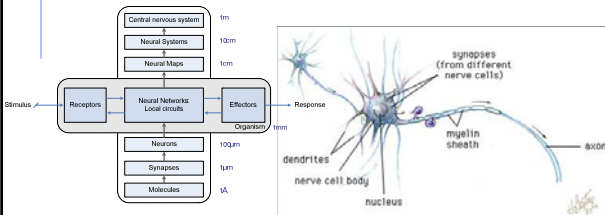
$$o_j = \tanh \left(\sum w_{ij} a_j - \theta_j \right)$$

```

neural_net = eval('net_' + str(neural_net_type))
[init_params, bias] = init(neural_net, num_neurons, num_hidden_neurons)
[output_neurons] = size(neural_net.neural_net_hidden_neurons)
if (strcmp(neural_net_adapter, 'net_type' == 'RBP'))
    neural_net_type = 'RBP';
end
fid = fopen(path, 'w');
  
```

Information Processing in biological Neurons

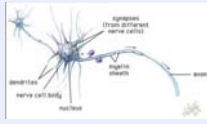
- Modelling of biological functions in Neurons
 - 10-100 Billion Neurons with 10000 connections in Brain
 - Input (sensory), Processing (internal) & Output (motoric) Neurons



- CONCEPT of Information Processing in Neurons ...

Alternative notations – Information processing in neurons / nodes

Biological Representation



Graphical Notation

Input: in_1, in_2, in_j with weights $w_{1,1}, w_{1,2}, w_{1,j}$

Input Function: $net_i = \sum_j w_j in_j$

Activation Function: $a_i = f(net_i - \theta_i)$

Output: out_i

Neuron / Node u_i

$\beta_i \Rightarrow$ weights w_i
 $\beta_0 \Rightarrow$ bias θ

Mathematical Representation

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

alternative: $y_i = \tanh\left(\sum_j w_j x_j - \theta_i\right)$...

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 ...)

Diagram: Input (in_1, in_2, in_j) with weights ($w_{1,1}, w_{1,2}, w_{1,j}$) feeds into the Input Function ($net_i = \sum_j w_j in_j - \theta_i$). This feeds into the Activation Function ($a_i = f(net_i)$), which feeds into the Output Function ($o_i = a_i$), resulting in Output (out). The entire process is labeled as Unidirectional Information Processing.

Equation:

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

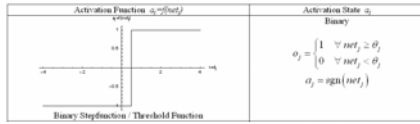
Input Functions

Input Function	Formula
Sum	$net_i = \sum_j o_j w_{ij}$

Binary Activation Functions

- Binary activation calculated from input

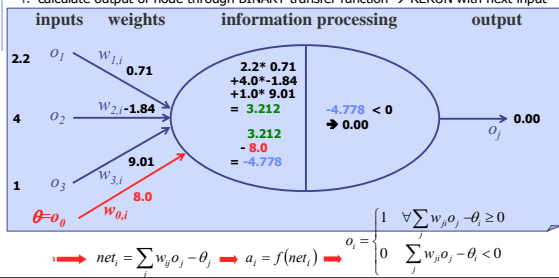
$$a_j = f_{act}(net_j, \theta_j) \quad \text{e.g. } a_j = f_{act}(net_j - \theta_j)$$



Information Processing: Node Threshold logic

Node Function → BINARY THRESHOLD LOGIC

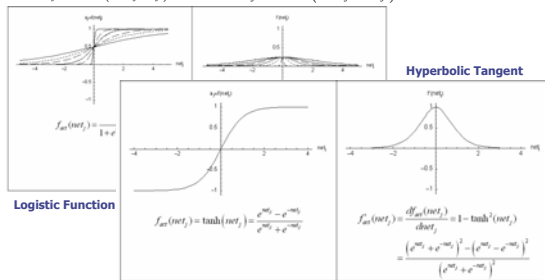
- weight individual input by connection strength
- sum weighted inputs
- add bias term
- calculate output of node through BINARY transfer function → RERUN with next input



Continuous Activation Functions

- Activation calculated from input

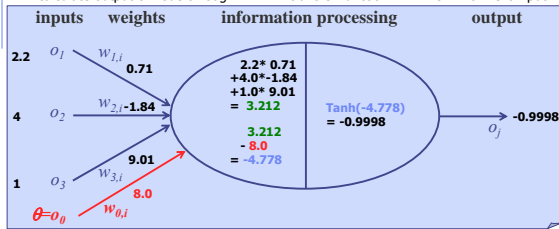
$$a_j = f_{act}(net_j, \theta_j) \quad \text{e.g. } a_j = f_{act}(net_j - \theta_j)$$



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



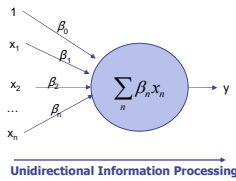
$$\text{net}_i = \sum_j w_{ij} o_j - \theta_i \Rightarrow a_i = f(\text{net}_i) \Rightarrow o_i = \tanh\left(\sum_j w_{ij} o_j - \theta_i\right)$$

A new Notation ... GRAPHICS!

- Single Linear Regression ... as an equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon$$

- Single Linear Regression ... as a directed graph:



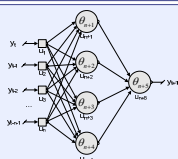
Why Graphical Notation?

- Simple neural network equation without recurrent feedbacks:

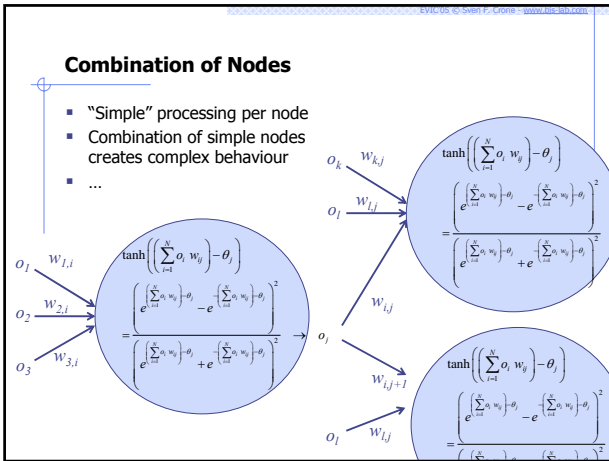
$$y_k = \tanh\left(\sum_k w_{kj} \tanh\left(\sum_j w_{ij} x_j - \theta_j\right) - \theta_k\right) \Rightarrow \text{Min!}$$

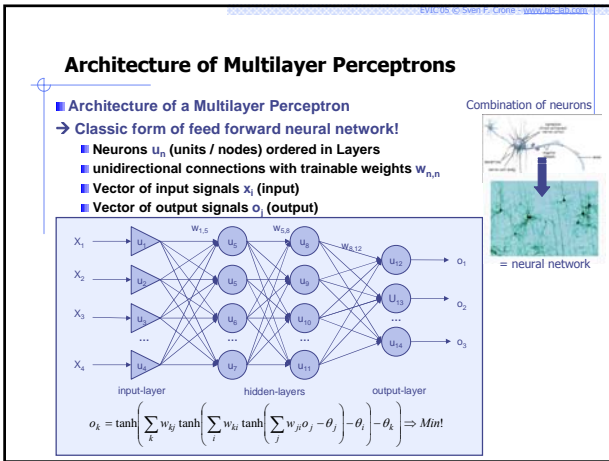
□ with ... $\beta_j \Rightarrow w_{ij}$ $\beta_0 \Rightarrow \theta$ $\tanh\left(\left(\sum_{j=1}^n x_j w_j\right) - \theta_j\right) = \frac{\left(e^{\left(\sum_{j=1}^n x_j w_j\right) - \theta_j} - e^{-\left(\sum_{j=1}^n x_j w_j\right) - \theta_j}\right)^2}{\left(e^{\left(\sum_{j=1}^n x_j w_j\right) - \theta_j} + e^{-\left(\sum_{j=1}^n x_j w_j\right) - \theta_j}\right)^2}$

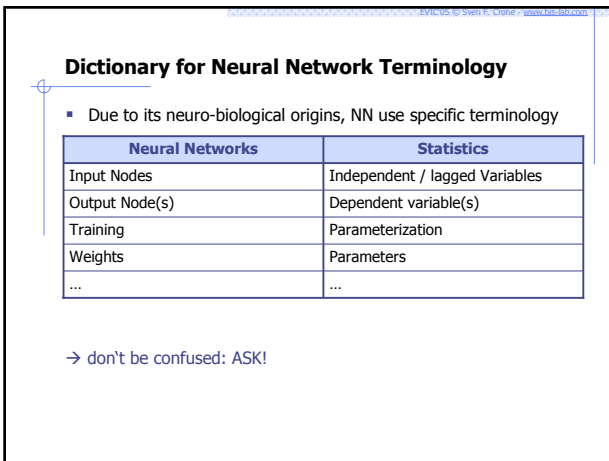
- Also:



→ Simplification for complex models!







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]

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

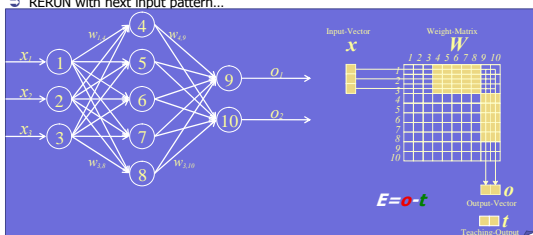
- Delta-learning rule of Widrow-Hoff

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

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]

$$E_{\mu} = C(u_{\mu}, o_{\mu}) \quad o_{\mu} = f_{\mu}(net_{\mu}) \quad \Delta_{\mu} w_{ij} \propto - \frac{\partial C(u_{\mu}, o_{\mu})}{\partial w_{ij}}$$

$$\frac{\partial C(u_{\mu}, o_{\mu})}{\partial w_{ij}} = \frac{\partial C(u_{\mu}, o_{\mu})}{\partial net_{\mu}} \frac{\partial net_{\mu}}{\partial w_{ij}}$$

$$\delta_{\mu} = - \frac{\partial C(u_{\mu}, o_{\mu})}{\partial net_{\mu}}$$

$$\delta_{\mu} = - \frac{\partial C(u_{\mu}, o_{\mu})}{\partial net_{\mu}} = - \frac{\partial C(u_{\mu}, o_{\mu})}{\partial o_{\mu}} \frac{\partial o_{\mu}}{\partial net_{\mu}}$$

$$\frac{\partial o_{\mu}}{\partial net_{\mu}} = f'_{\mu}(net_{\mu})$$

$$\delta_{\mu} = \frac{\partial C(u_{\mu}, o_{\mu})}{\partial o_{\mu}} f'_{\mu}(net_{\mu})$$

$$\sum_{\mu} \frac{\partial C(u_{\mu}, o_{\mu})}{\partial net_{\mu}} \frac{\partial net_{\mu}}{\partial w_{ij}} = \sum_{\mu} \frac{\partial C(u_{\mu}, o_{\mu})}{\partial net_{\mu}} \frac{\partial \sum_k w_{ik} o_{\mu k}}{\partial w_{ij}}$$

$$= \sum_{\mu} \frac{\partial C(u_{\mu}, o_{\mu})}{\partial net_{\mu}} w_{ij} = - \sum_{\mu} \delta_{\mu} w_{ij}$$

$$\delta_{\mu} = f'_{\mu}(net_{\mu}) \sum_k \delta_{\mu k} w_{ik}$$

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

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

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

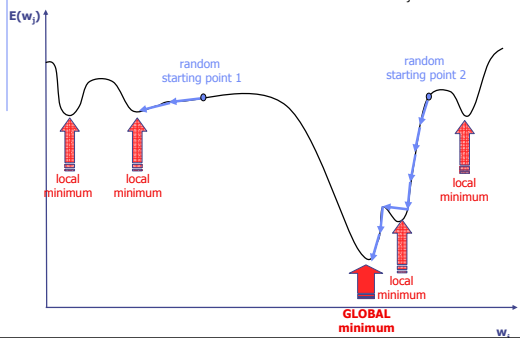
mit $f(net_j) = \frac{1}{1 + e^{-\sum_k w_{jk} o_k}} \rightarrow f'(net_j) = o_j(1 - o_j)$

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

$\delta_{\mu} = \begin{cases} \frac{\partial C(u_{\mu}, o_{\mu})}{\partial o_{\mu}} f'_{\mu}(net_{\mu}) & \text{if unit } j \text{ is in the output layer} \\ f'_{\mu}(net_{\mu}) \sum_k \delta_{\mu k} w_{jk} & \text{if unit } j \text{ is in a hidden layer} \end{cases}$

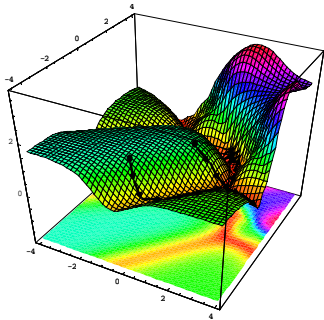
Neural Network Training = Error Minimization

- Minimize Error through changing ONE weight w_j



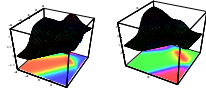
Error Backpropagation = 3D+ Gradient Decent

- 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



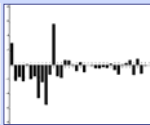
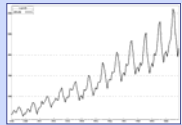
Demo: Neural Network Forecasting revisited!

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 $t, t-1, \dots, t-11$ (past 12 observations) \rightarrow time series prediction
- $t+1$ forecast \rightarrow single step ahead forecast



- \rightarrow 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!

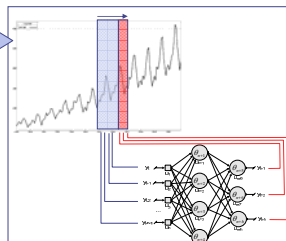
Time Series Prediction with Artificial Neural Networks

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

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

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

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



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

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 against 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

→ **Interpretation**

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

→ **Extensions**

- multiple output nodes = simultaneous autoregression models
- Non-linearity through different activation function in output node

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

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

linear autoregressive AR(p)-model

Neural Network Architecture for Nonlinear Autoregression

→ **Extensions**

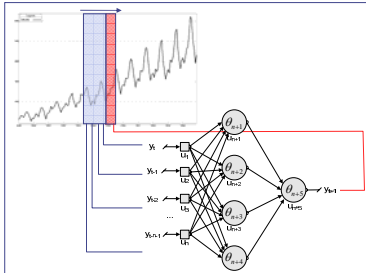
- additional layers with nonlinear nodes
- linear activation function in output layer

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

$$\hat{y}_{t+1} = \tanh\left(\sum_{j=1}^{n-1} y_j w_{tj} - \theta_j\right)$$

Nonlinear autoregressive AR(p)-model

Neural Network Architectures for Nonlinear Autoregression



$$\hat{y}_{t+1} = \tanh \left(\sum_k w_{kj} \tanh \left(\sum_l w_{kl} \tanh \left(\sum_j w_{jl} y_{t-j} - \theta_l \right) - \theta_k \right) \right)$$

Nonlinear autoregressive AR(p)-model

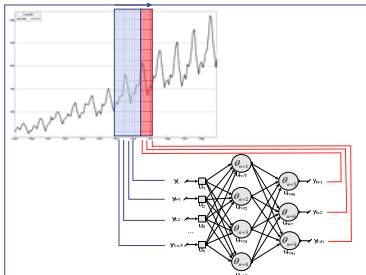
→ Interpretation

- 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 autoregression models

Neural Network Architectures for Multiple Step Ahead Nonlinear Autoregression



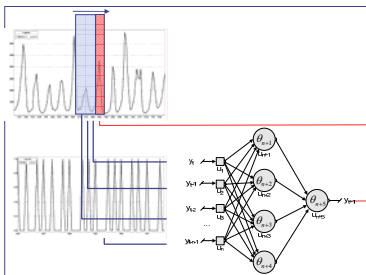
$$\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+n} = f(y_t, y_{t-1}, y_{t-2}, \dots, y_{t-n+1})$$

Nonlinear autoregressive AR(p)-model

→ Interpretation

- As single Autoregressive modeling AR(p)

Neural Network Architectures for Forecasting - Nonlinear Autoregression Intervention Model



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

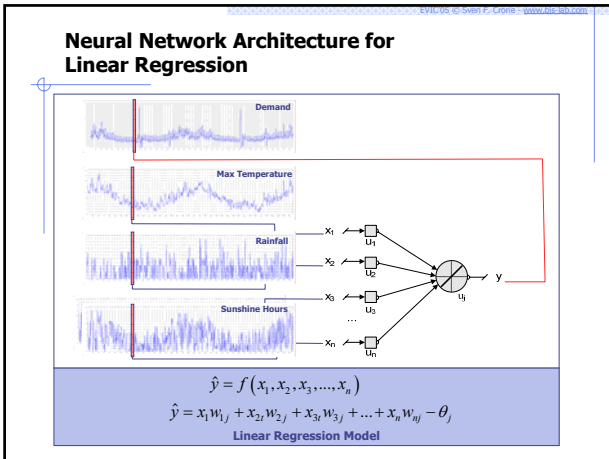
Nonlinear autoregressive ARX(p)-model

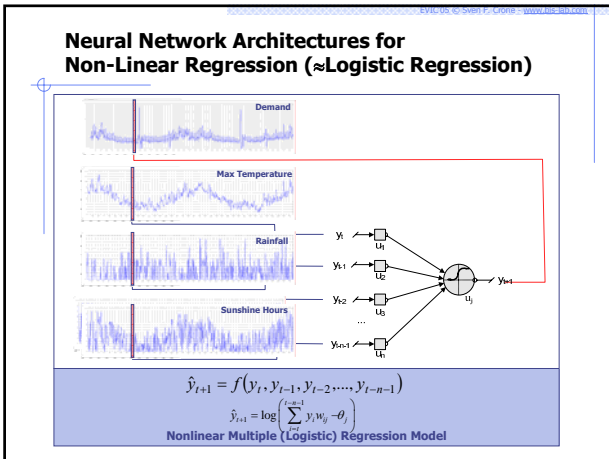
→ Interpretation

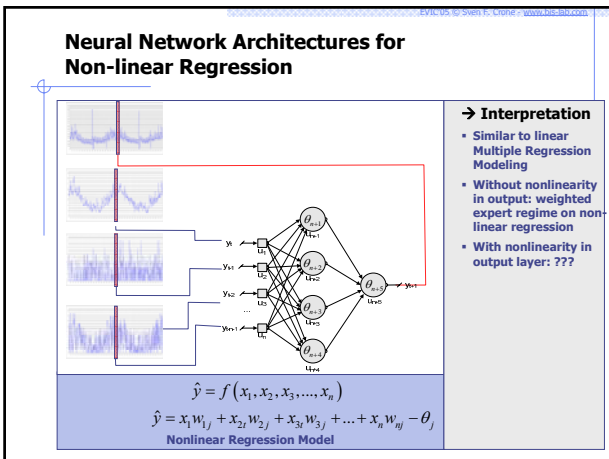
- As single Autoregressive modeling AR(p)
- Additional Event term to explain external events

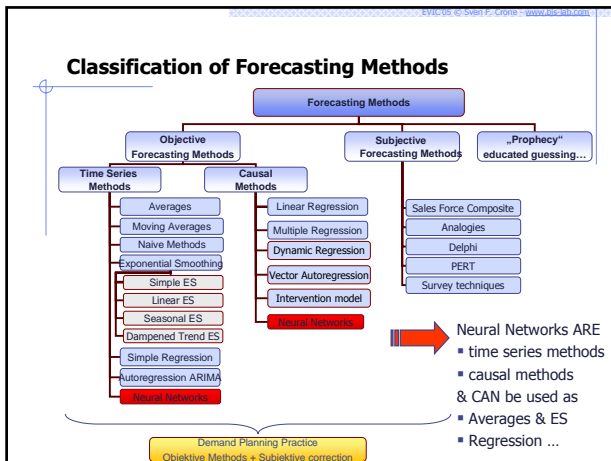
→ Extensions

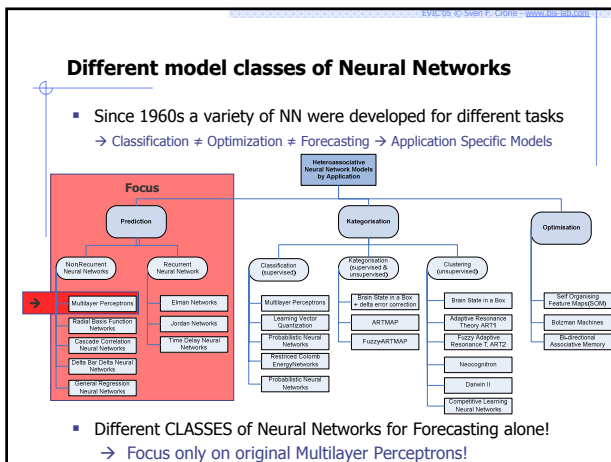
- multiple output nodes = simultaneous multiple regression











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

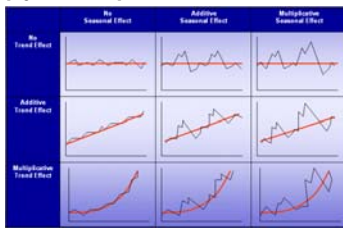
Agenda

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]



- ... ???



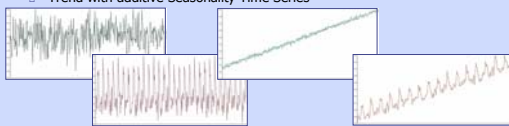
→ Simulation of
Neural Network prediction of
Artificial Time Series

Time Series Demonstration – Artificial Time Series

- Simulation of NN in Business Forecasting with NeuroPredictor

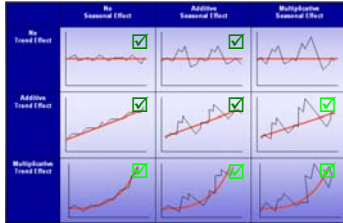


- Experiment: Prediction of Artificial Time Series (Gaussian noise)
 - Stationary Time Series
 - Seasonal Time Series
 - linear Trend Time Series
 - Trend with additive Seasonality Time Series



Time Series Prediction with Artificial Neural Networks

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



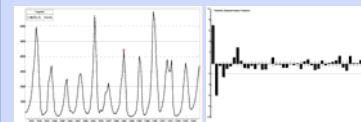
- Neural Networks can forecast ALL mayor time series patterns
 - NO time series dependent preprocessing / integration necessary
 - NO time series dependent MODEL SELECTION required!!!
 - **SINGLE MODEL APPROACH FEASIBLE!**

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: $t, t-1, \dots, t-11$ (past 12 observations)
 - $t+1$ forecast → single step ahead forecast



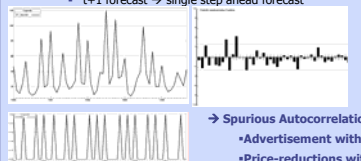
- Benchmark Time Series [Andrews / Hertzberg]
 - 114 observations
 - Periodicity? 8 years?

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 $t, t-1, \dots, t-11$ (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

- Forecasting?
- Neural Networks?
- Forecasting with Neural Networks ...
 - NN models for Time Series & Dynamic Causal Prediction
 - NN experiments
- Process of NN modelling
 - Preprocessing
 - Modelling NN Architecture
 - Training
 - Evaluation
- 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 require Expert-Knowledge

Modeling Degrees of Freedom

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

D= Dataset	[DSE Selection]	[DSA Sampling]						
P= Preprocessing	[C Corrector]	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!

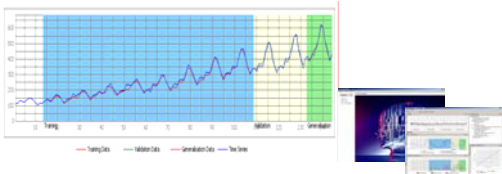
Heuristics to Reduce Design Complexity

- Number of Hidden nodes in MLPs (in no. of input nodes n)
 - 2n+1 [Lippmann87; Hecht-Nielsen90; Zhang/Pauwo/Hu98]
 - 2n [Wong91]; $\frac{n}{2}$ [Kang91]
 - 0.75n [Bailey90]; 1.5n to 3n [Kasra/Boyd96] ...
- Activation Function and preprocessing
 - logistic in hidden & output [Tang/Fischwick93; Lattermacher/Fuller95; Sharda/Pati92]
 - 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 heuristics
 → 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

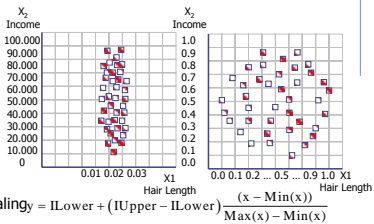
- Data Transformation
 - Verification, correction & editing (data entry errors etc.)
 - Coding of Variables
 - Scaling of Variables
 - Selection of independent Variables (PCA)
 - Outlier removal
 - Missing Value imputation
- Data Coding
 - Binary coding of external events → binary coding
 - n and n-1 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 $y = I_{Lower} + (I_{Upper} - I_{Lower}) \frac{(x - \text{Min}(x))}{\text{Max}(x) - \text{Min}(x)}$
- Interval features, e.g. „turnover“ [28.12 ; 70; 32; 25.05 ; 10.17 ...]

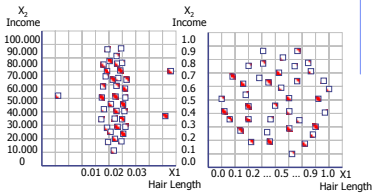
Linear Interval scaling to target interval, e.g. [-1;1]

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

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

Data Preprocessing – Variable Scaling

Scaling of variables



- Standardisation / Normalisation

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

- 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!

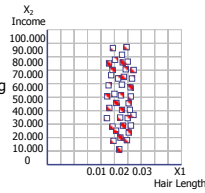
Data Preprocessing – Data Encoding

- Downscaling & Coding of variables
 - metric variables → create bins/buckets of ordinal variables (=BINNING)
 - Create buckets of equally spaced intervals
 - 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

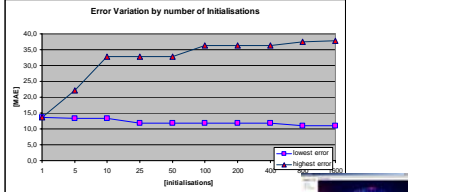
Forecasting with Artificial Neural Networks

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 – careful 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

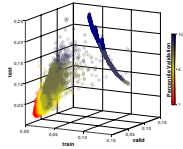
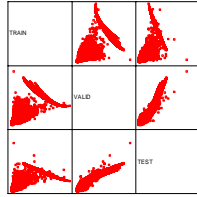
Forecasting with Artificial Neural Networks

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

Experimental Results

Experiments ranked by validation error

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

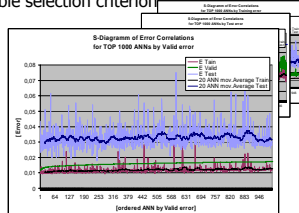
Problem: Validation Error Correlations

Correlations between dataset errors

Data included	Correlation between datasets		
	Train - Validate	Validate - Test	Train - Test
14400 ANNs	0,7786**	0,9750**	0,7886**
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, Armstrong 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

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

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

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

- 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:

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

Selecting Forecasting Error Measures

- $MAPE$ & MSE are subject to upward bias by single bad forecast
- Alternative measures may be 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

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

- Median Squared Error

$$MdSE_f = \text{Med}(e_{f,t}^2)$$

Evaluation of Forecasting Methods

- The *Base Line* model in a forecasting competition is the Naive 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 through our method (outperforms naive) if $U < 1$

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

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!

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

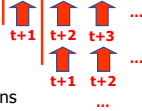
↑ ↑ ↑ ...
t+1 t+2 t+3
SIMULATED = EX POST Forecasts

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



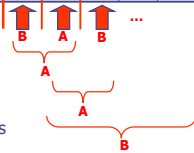
- Evaluate only RELEVANT horizons
 - omit t+2 if irrelevant for planning!

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



- 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
 - Matlab 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





EVR-05 to Shell F. Glore - www.bis-lab.com

Neural Networks Software - Times Series friendly!

 Alyuda Inc.		
 Ward Systems	AT Trilogy: NeuroShell Predictor, NeuroShell Classifier, GeneHunter NeuroShell 2, NeuroShell Trader, Pro, DayTrader	
Altrasoft Inc.	Predictor Predictor PRO	
 Promised Land	Braincell	
Neural Planner Inc.	Easy NN Easy NN Plus	
 NeuroDimension	NeuroSolutions Cos consultant NeuroSolutions for Excel NeuroSolutions for Matlab Trading Solutions	

EVR-05 to Shell F. Glore - www.bis-lab.com

Neural networks Software – General Applications

 Neuralware Inc	Neural Works Professional II Plus	
 SPSS	SPSS Clementine Data Mining Suite	
 SAS	SAS Enterprise Miner	
...	...	

EVR-05 to Shell F. Glore - www.bis-lab.com

Further Information

- **Literature & websites**
 - NN Forecasting website www.neural-forecasting.com or www.bis-lab.com
 - Google web-resources, SAS NN newsgroup FAQ <http://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: IJF & 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 available:
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