Project no. 505428 (GOCE)

AquaTerra

Integrated Modelling of the river-sediment-soil-groundwater system; advanced tools for the management of catchment areas and river basins in the context of global change

Integrated Project

Thematic Priority: Sustainable development, global change and ecosystems

Deliverable No.: **T3.1**
Title: **Progress report on trend analysis methods, tools and data preparation (in particular artificial neural networks (ANN))**

Due date of deliverable: November 2005
Actual submission date: November 2005

Start date of project: 01 June 2004 **Duration: 60 months**

Organisation name *and contact* of lead contractor *and other contributing partners* for this deliverable:
University of Tübingen, Center for Applied Geoscience
Claudius M. Bürger, claudius.buerger@uni-tuebingen.de
Olaf Kolditz, kolditz@uni-tuebingen.de

Revision: M. Gerzabek / J. Barth

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

SUMMARY

Based on the focus of the TREND sub-program on the time scale, a comprehensive literature research regarding the state-of-the-art of trend analysis methods with respect to time series data was carried out. Apart from advanced methods like autoregressive integrated moving average models or singular spectrum analysis a particular weight was given to the application of artificial neural networks (ANN). Among the various architectures of ANN the so-called support vector machines (a recent development) are discussed in greater detail.
For most of the covered time series analysis methods internet sources for free software tools are provided.
In contrast to the deliberately theoretical chapters of the literature research a report is provided on practical issues regarding the development process of the data base for the Gallego catchment and modelling strategies in the last chapter.

MILESTONES REACHED (from DOW p. 80 –85)

T3.1: Research of state-of-the-art in artificial neural network (ANN) modelling, genetic algorithms
Through the cooperation with Hayley Fowler from HYDRO daily precipitation, temperature and potential evapotranspiration data could be obtained for the Ebro basin and Gallego river in particular. Through Alicia Navarro Ortega from CSIC (BASIN) the CHE ("Confederacion Hidrologica del Ebro") chemical database was made accessible for TREND 3. Further co-operation exists with the COMPUTE partners, UT in particular. In the Donube basin TREND 3 will also be active through a co-operation with TREND 1, specifically with Georg Lair in the workgroup of Prof. Gerzabek at BOKU, Vienna. Nevertheless, additional data is required especially regarding hydrogeological issues in the Ebro basin. In that respect, TREND 3 is also active outside of the AquaTerra project (University of Zaragoza, Ebro Agua Group of TU Darmstadt, CHE).

1 Glossary

Artificial neural network a collection of learning machines, which are based on principles known from information processing in the human brain

Generalisation the capability of a learning machine to find and represent an (unknown) underlying functional relationship within a (complex) data set (consisting of input and corresponding output data)

Genetic algorithm heuristic, typically binary-coded optimisation algorithms that build upon the principles of biological evolution (heritage, mutation, selection from a population of individuals)

Learning machine a computer program established by machine learning, a scientific discipline concerned with developing methods for software to learn from experience or extract knowledge from examples in a database

Regression a mathematical method used to predict a dependent variable by a combination of a number of independent variables. The combination assumes a certain function class with associated parameters which are typically estimated using least-squares methods

Seasonality fluctuation of a system variable with a constant period (and usually a constant amplitude)

Statistical learning theory a form of computational learning theory, which attempts to explain the learning process from a statistical point of view.

Support vector machine a learning machine that uses the principles of statistical learning theory to find a functional as simple as possible to reach a generalisation as good as possible for the description of a given data set.

Time series a collection of data of the same type ordered by the point in time of their measurement.

Trend a deterministic, non-periodic function describing the change of a system variable with time or along a space coordinate.

Validation procedure to find a particular model and parameter set of a certain type of learning machine that generalises best for a given finite data set

1 Introduction

The overall objective of the AquaTerra sub-programme TREND is concerned with the long term developments of the different compartments of the river-sediment-soil-groundwater system (RSSG). Whereas sub-programmes COMPUTE and HYDRO mainly focus on spatial scales, TREND addresses the time scale and focuses on the detection and modelling of dynamic behaviour of (spatial) phenomena belonging to the RSSG (see DOW from 22/07/2004). In this context, trend analysis may be defined as the analysis of changes in a given item/variable or sets of items/variables of information regarding the RSSG over a period of time. Such items could be hydrological data, e.g. as stream flow at several gauging stations within a river basin or physico-chemical data, as sorption properties of soils, which may change according to a superimposed climate or land use change. While a river system would be considered as relatively rapidly responding to climate changes, a rather gradual change is likely to be attributed to the soil properties. Hence, there is also the notion of different (time) scales of changes to be explored in TREND.

Quite natural, trend analysis as defined above has to deal with time series data. These typically consist of time resolved observations reflecting a system's response to some (usually unknown or not quantified) external driving force(s). A time series can be defined as any kind of (timely) ordered data series :

$$X = \{x_1, x_2, x_3, x_4, ...x_t\}; \text{ with } t > 0$$

An analysis of such time series is commonly carried out with the purpose of i) its description, ii) its modelling and iii) a prognosis of the future evolution of the time series [Schlittgen & Streitberg 1984]. A central idea of i) is a regression (linear or non-linear) analysis in a least squares sense. A typical application involves the time series decomposition into a trend (long-term development of series), a cyclicity (referring to regular, for instance, seasonal fluctuations of known period) and an irregular residual component. These forms of regression models, however, require a number of prerequisites (e.g. independency of the individual components) which may be called unrealistic when dealing with natural time series [e.g. Schlittgen & Streitberg 1984]. Therefore, time series modelling (ii) usually comprises the formulation and fitting of stochastic models. The prognosis (iii), i.e. the forecasting of

future values of the time series from its historic data record, then essentially requires the validity of the fitted (stochastic) model.

When adopting the stochastic viewpoint, the observed time series is understood as a sequence of realisations of a stochastic process, i.e. a sequence of - generally dependent - random variables. In this context the term ‘trend’ is mathematically defined as being the non-random function:

$$\mu(t) = E[X(t)],$$

where *E*[] is the expected value operator.

An important part of trend analysis are test statistics for trend detection. Profound treatment of this issue regarding time series of environmental quality data can be found e.g. in Hirsch et al. [1982], Hirsch and Slack [1984], Hirsch et al. [1991], as well as El-Shaarawi [1993], and Libiseller & Grimvall [2002] for more recent developments. Current applications in hydrology can be found in e.g. Bonaccorso et al. [2005], Kallache et al. [2005]. However, within this report we intend to focus on wider sense regression and modelling techniques and refer to the above cited literature for trend detection tasks.

As time series are observed for a variety of different variables many different scientific disciplines have been concerned with their study, e.g. economics, meteorology, sociology. The methods applied to time series are therefore manifold and this report can only cover a selection of them comprising least-squares regression and Fourier analysis as standard methods, as well as autoregressive integrated moving average models (ARIMA), singular spectrum analysis (SSA) and variants on the more advanced side. Furthermore, a main focus of this report (as mentioned in the title) are artificial neural networks (ANN) containing a whole spectrum of (time series) data analysis methods. In particular, the types of feed-forward and recurrent neural networks, as well as the more recently developed kernel support vector machines will be covered. For virtually all of these methods we provide internet sources with more or less freely available software tools.

An additional chapter is devoted to the status of the current data base of the Gallego catchment in the Ebro basin, the data preparation process and the modelling concepts to be used in TREND 3.

2 Prerequisites

As mentioned above, trend analysis in the connection with future trend prognosis is concerned with finding the right model function to best explain the observed historic time series data. However, a general form of stationarity of the series generating system and its driving forces is crucial for such an analysis [Kantz & Schreiber 1997]. E.g. the parameters of the system have to remain constant throughout the observation period or have to undergo some (known) well-behaved changes rather than abrupt ones caused e.g. by human action or “catastrophic” events. For instance, a new oil spill or the completion of a treatment plant will change the trend in downgradient concentration levels [Hirsch et al. 1991], however, this pattern change can not be inferred from the historic data observed so far. If additional information about such events is abundant and can be related to the time series behaviour, i.e. there is a discernable “before” and “after” within the series, analyses should focus on each phase individually [Hirsch et al. 1991].

Furthermore, an additional number of practical problems has to be faced: Time series analysis requires the data record to be complete[1] and sampled at constant time intervals. However, frequently the number of available data sets from different observation locations (gauging stations) goes hand in hand with multiple starting dates, ending dates, and data gaps in groups of records. Some practical approaches to these problems are mentioned in chapter 4. Nevertheless, multiple gauging station data should be concurrent for an (multivariate) analysis [Hirsch et al. 1991]. Additionally, there exists the difficulty to select a time period long enough to be meaningful and not to exclude too many shorter records. Hirsch et al. [1991] give some practical advice on how to deal with this issue in the context of trend detection tasks (i.e., 1. divide study period into thirds, 2. determine the coverage in each period (e.g. if the record is generally monthly, count the months for which there are data), 3. if any of the thirds has less than 20% of the total coverage then the record should not be included in the analysis).

[1] referring to the commonly used time series analysis methods. However, see e.g. Bardossy [1998] or Schoellhamer [2001] for specialisations regarding missing data.

3 Time series analysis methods

Time series analysis methods can either be applied to the time domain (e.g. regression, ARIMA models) or to the frequency domain (e.g. Fourier analysis and extensions). Most of the methods are introduced by assuming a single, i.e. univariate time series. However, the generalisation of the methods to several, i.e. multivariate time series usually holds. In the following we begin with two standard analysis methods, which offer important concepts. However, the presentation of these methods is intentionally brief, as our main focus lies on the more advanced methods of the later chapters.

3.1 Least squares regression

Least squares regression requires the fitting of a model curve to the time series data. In this deterministic approach, one has to decide beforehand, which function class (e.g. linear, polynomial, logarithmic) $f_\alpha(t)$ to apply (where $\alpha \in \Lambda$ and Λ represents a set of possible values for the function class' free parameters). In the univariate or one-dimensional case (simple curve-fitting), a decision for the function class can be made by visual inspection of a time-value plot (see Figure 3-1). The function parameters α are found by minimising some form of squared residuals, the typical difference or error measure being the mean squared error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(x(t_i) - f_\alpha\left(t_i\right)\right)^2 ,$$

where $x(t_i)$ is the observed time series value at time t_i and $f_\alpha(t_i)$ is the computed function value for time t_i for a fixed α. According to e.g. Gottman [1981], Streitberg & Schlittgen [1984] a least square fit is also a canonical method to de-trend time series for further analysis (e.g. for the use of ARMA models, see section 3.3). Future predictions could also be made using the formula of the fitted curve.

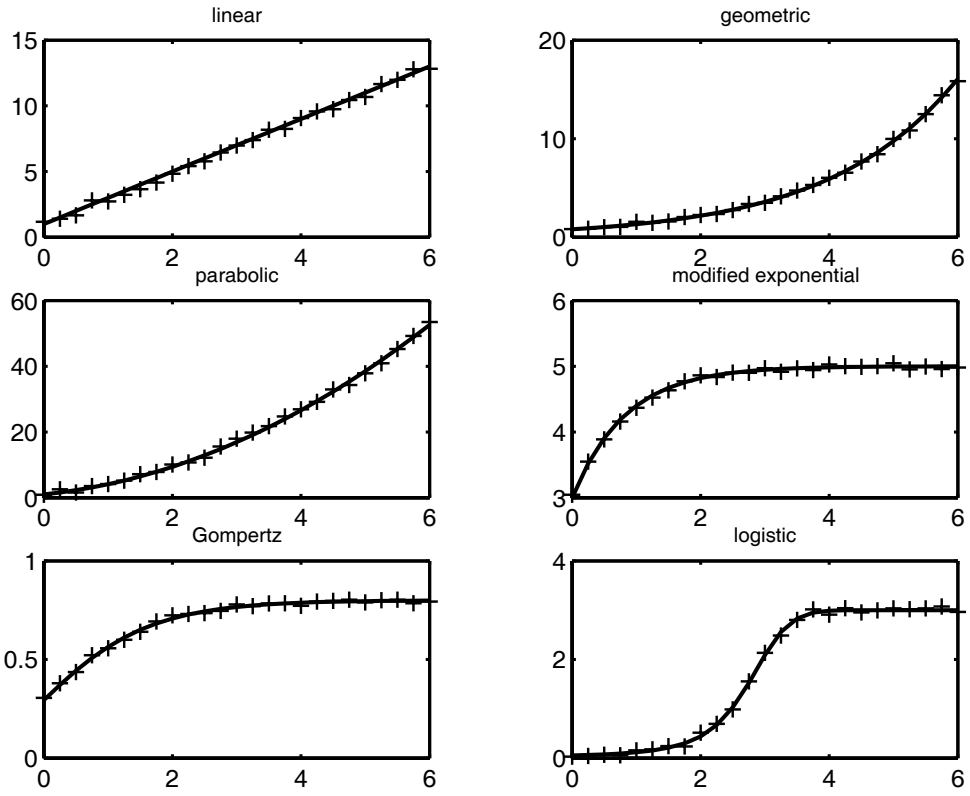

Figure 3-1: Graphical representation of fitted growth curves (+:points of synthetic data set) showing different forms of an upward trend in a time-value plot.

Table 3-1 shows a selection of model curves which reflect different shapes of growth curves i.e. an upward trend (the respective shapes are illustrated in Figure 3-1). A freely available toolbox for MATLAB containing commonly used regression functions can be downloaded from http://www.sci.usq.edu.au/staff/dunn/glmlab/glmlab.html.

It is easy to see, however, that in more complex, higher dimensional cases one has the problem to infer suitable regression functions. Furthermore, the underlying function in the time series data should not be masked or disturbed by noise and/or (pseudo-) periodicities. However, the more advanced methods covered in the later chapters will offer techniques to alleviate such shortcomings. Especially, ANN offer a flexible way to implement a variety of function classes, such that the decision for a particular function class does not have to made a-priori to the model building process.

Table 3-1: List of one-dimensional functions describing an (upward) trend (growth curves)[2]

Name	Formula	Description
linear	$y = ax + b$	straight line; direct proportionality
geometric	$y = a \cdot \mathrm{e}^{(bx)}, b > 0$	b > 1, y $\to \infty$ b = 1, y = a b < 1, y $\to$ 0
parabolic	$y = a + bx + cx^2$	a: y intercept; b: slope at y intercept c: controls curve shape
modified exponential	$y = a + bc^x, 0 < c < 1$	a + b: y intercept b < 0: convex upward b > 0: concave upward a: lower or upper bound
Gompertz	$y = a + c \cdot e^{-eb(x-m)}$	a: lower asymptote; c: upper asymptote; m: time of maximum growth; b: growth rate,
logistic	$y = a + \frac{c}{\left(1 + t \cdot e^{-b(x-m)}\right)^{\frac{1}{t}}}$	a: lower asymptote; c: upper asymptote; m: time of maximum growth; b: growth rate, t: near which asymptote maximum growth occurs

3.2 Fourier Analysis

A standard method for analysing time series with respect to periodicity and seasonality is a Fourier analysis. Here, the (discretely sampled) time series is transformed from the time domain into the frequency or spectral domain, where perfectly sinusoidal oscillations correspond to a single frequency peak. Using the Euler representation of complex numbers $e^{i\phi} = \cos(\phi) + i\sin(\phi)$ a convenient form of a Fourier transform of frequency *n* is (after Gottman [1981]):

$$f_n = \sum_{k=1}^{N-1} e^{-2\pi i k \cdot f} \cdot x_k$$

From this Fourier-transform the periodogram can be computed by

$$I(f) = \frac{1}{2\pi T} \left| \sum_{k=1}^{N-1} e^{-2\pi i k \cdot f} \cdot x_k \right|^2$$

---

[2] compilation of growth curves found at:
http://people.uleth.ca/~maclachlan/Trend_Projection_Models.ppt
http://www.bioss.ac.uk/smart/unix/mgrow/slides/

From the periodogram one can easily detect the key frequencies of seasonalities and remove them from the time series by building the inverse Fourier transform without the corresponding frequency peaks.

However, there are clear limits of this approach when applied to natural time series, e.g. Jenkin & Watts [1968]: “The basic reason why Fourier analysis breaks down when applied to time series is that it is based on the assumption of fixed amplitudes, frequencies and phases. Time series, on the other hand, are characterized by random changes of amplitudes, frequencies and phases.” Furthermore, the trend should be removed prior to the analysis and Fourier analysis scatters the energy of non-sinusoidal signals into its higher harmonics (e.g. Hsieh [2004]), such that a signal reconstruction becomes complicated, i.e. the number of frequency peaks needed to capture the characteristic features of the time series is not significantly lower than the number of data describing it in the time domain.

Nevertheless, a Fourier analysis may still serve as tool pointing the direction of further analyses. According to Gottman [1981], Schlittgen & Streitberg [1984], and Kruel [1992] some characteristic features of the periodogram may be indicative of the underlying system dynamics (compilation by M. Strickert [2003]):

- An overall noisy structure of frequency intensities with a lack of pronounced peaks indicates a signal that itself is rather noisy.
- Irregularly shaped graphs with an intensity distribution over a wide range of frequencies may belong to noisy observations of a deterministic chaotic source.
- Straight lines on a log-log plot of the periodogram may be indicative of self-similarity pointing towards a fractal property of the source.
- Isolated frequency peaks unequally spaced over the frequency range may belong to quasi-periodic data showing driving forces with incommensurable periodic oscillations superimposed
- Conversely, equally distant frequency peaks appear for an observed oscillation, whose basic form is the result of the sum of harmonics, but with a shape very different from a sine or cosine.

Figure 3-2 shows the results of a Fourier analysis of the temperature data time series (upper graph) from the Gallego catchment in the Ebro basin recorded between 1st of January, 1962 to 31st of December, 1980 at station 9481. The series’ mean was

removed prior to the analysis and the periodogram computed (third diagram from top).

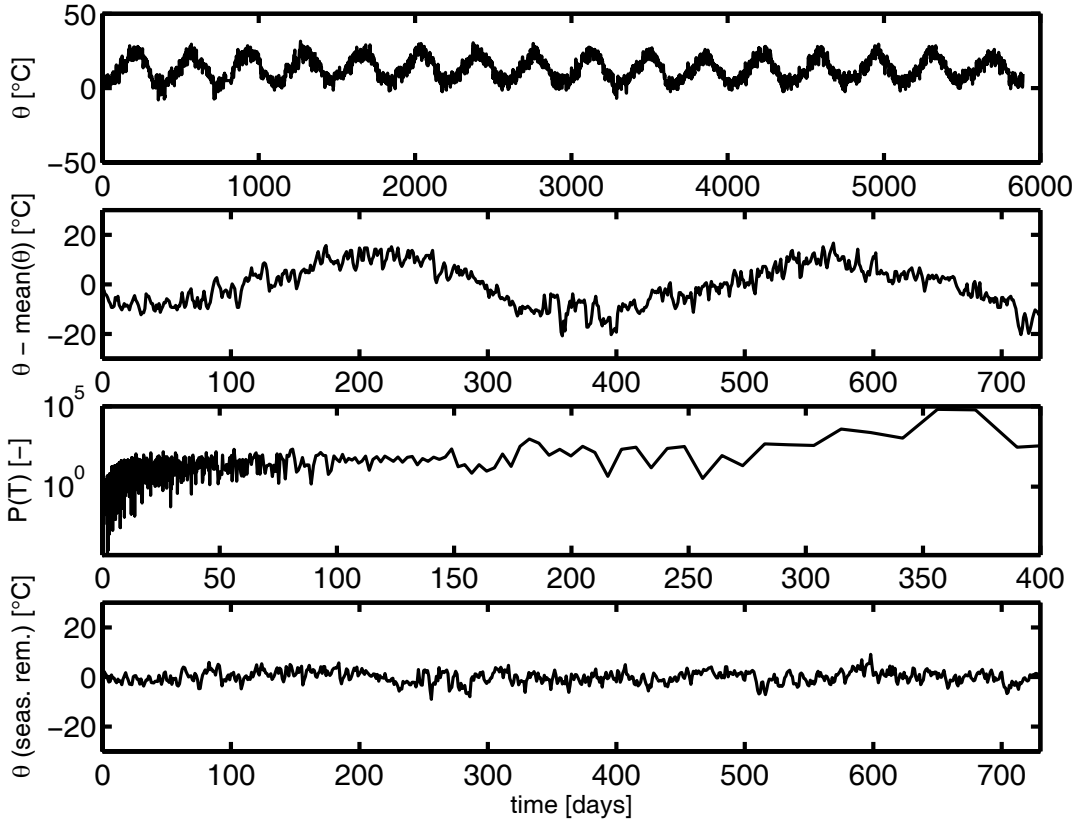

Figure 3-2: Mean daily temperature for gauge 9481 (1962-1990) in the Gallego catchment (top diagram), magnification of temperature series with mean removed (second diagram from top), periodogram of temperature series with mean removed (third diagram from top), series reconstruction with annual seasonality removed (bottom diagram).

This periodogram is not characterized by a few prominent peaks, but by a broad range of medium peaks increasing in (peak) frequency towards zero. The most prominent peak has a relatively broad plateau around the most obvious period of 365 days. Filtering this broad peak and reconstruction of the original series via the inverse Fourier transform leads to the bottom diagram on Figure 3-2, where the remains of the temperature signal appear to be mere high frequency noise. In this case Fourier analysis was helpful for removing the (known) seasonality from the signal and, in turn, the most prominent feature of the signal is easily discovered in the periodogram. However, a signal reconstruction becomes difficult as most of the variability apart from the 365 period is interspersed in various peaks of the higher harmonics.

A further extension of Fourier analysis that proved to be useful in analysing localised variations of power within a time series is the so-called wavelet analysis. For a practical introduction the interested reader is pointed to Torrence & Compo [1998].

3.3 Autoregressive integrated moving average models (ARIMA)

Due to the limitations of a Fourier analysis pointed out above, different modelling approaches were sought for natural time series. Two (independent) discoveries regarding white noise processes in the 1920s spawned the idea to understand a natural time series data record as a realisation of a stochastic process. Slutzky, a Russian mathematician, found that he could transform a white noise data sequence into nearly any pattern by applying a moving average with constant weights. Yule, an American mathematician found that he could transform any time series into noise by subtracting out its own dependence on the past [Gottman 1981]. The operations are now well known as being a linear filter and its corresponding reverse linear filter and are termed a moving average (MA) and an autoregressive (AR) process.

ARIMA models, sometimes referred to as Box-Jenkins models (after Box & Jenkins [1976]), are hybrid models, which combine the MA and the AR processes after differencing consecutive time series data (the I results from the subsequent integration, which becomes necessary for series re-construction). The AR part builds upon a (short-term) memory of the system and explains the next value of a time series by a window of *p* time steps in the past up to an additive Gaussian white noise or error term ε_t , i.e. the random variable ε_t has a time invariant zero mean and a fixed variance σ^2 ,e.g. Strickert [2003]:

$$AR(p): x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t$$

The MA part uses Gaussian white noise as model input to generate a model dynamic. The noise is transformed by a weighted (weights θ_i) sum of a (moving) window of size *q*:

$$MA(q): x_t = \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

The combination of an AR(*p*) and an MA(*q*) process results in an ARMA(*p*,*q*) process with a higher potential for modelling strong system dynamics relative to the individual subprocesses and usually more parsimonious models, i.e. the ARMA model will have less parameters than an AR or MA model with a comparable goodness-of-fit [Streitberg & Schlittgen, 1984].

$$ARMA(p,q): x_t = \varepsilon_t + \sum_{i=1}^{p} \phi_i x_{i-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}$$

However, the major prerequisite for ARMA type models is the (statistical) stationarity of the time series, i.e. the constancy of all of its moments in time. Since such stationarity is hard to verify in practice, it is usually relaxed to a “weak” or second-order stationarity, which refers to the time-constancy of the first two moments, mean and variance (From here on we will refer to the second-order sense wherever using the term ‘stationarity’). Likewise, stationarity implies the absence of trends within the time series, meaning that the trend has to be removed beforehand.

In order to cope with the presence of trends, a differencing of the order *d* (*d* $\in \mathbb{N}$), i.e. values of lag *d* are subtracted from each other, is applied to the time series in ARIMA models. The resulting ARIMA(*p*,*d*,*q*) model equation is usually given using the lag operator *L*, which is specified as (with *t* > 0)

$$\begin{aligned} L^i x_t &= x_{t-i} \\ L^{-i} x_t &= x_{t+i} \end{aligned};$$

$$ARIMA(p,d,q): \left(1 - \sum_{i=1}^{p} \phi_i L^i\right)(1-L)^d x_t = \left(1 + \sum_{j=1}^{q} \theta_j L^j\right)\varepsilon_t$$

With an ARIMA model one is capable of generating non-stationary time series with the presence of trends [Streitberg & Schlittgen 1984]. However, a crucial point is the dimension of differencing, since differencing applied unjustified (overdifferencing) may alter the time series in a more substantial way, than only remove the trend [e.g. Gottman 1981]. Additional problems of AR(I)MA models stem from the use of Gaussian white noise terms, which in the case of forecasting are unknown and have to be averaged over (Strickert [2003] and Kantz & Schreiber [1997] respectively) and are symmetrically distributed, which might not be the case for many natural systems

(Strickert [2003] and Tong [1990] respectively). Nevertheless, the model identification and estimation process is well established and can be found in various time series textbooks, e.g. Box & Jenkins [1976], Gottman [1981], Streitberg & Schlittgen [1984]. The required introduction of some additional concepts (e.g. (partial) autocorrelation functions), however, was considered beyond the scope of this report and we refer to the cited literature for that purpose. A toolbox for MATLAB comprising (adaptive) AR(I)MA type models can be downloaded from the website http://www.dpmi.tu-graz.ac.at/~schloegl/matlab/tsa/download.html.

ARIMA models are widely used for time series modelling and a number of applications can be found in hydrology, one of the first supposedly being annual stream flow modelling by Carlson et al. [1970]. According to Bardossy [1998] annual and monthly precipitation amounts can be modelled by ARMA processes. However, due to their intermittent property modelling daily and hourly precipitation series remains difficult. Abrahart & See [1998] and more recently e.g. Brath et al. [2002] have used AR(I)MA models for river flow and real time flood forecasting as a basis for comparison with neural network approaches.

A logical extension of AR(I)MA type models is to assume the dependence of x_t on past values and the error terms ε_t to be non-linear resulting in non-linear moving average (NMA), non-linear autoregressive (NAR), or non-linear autoregressive moving average (NARMA) models. Concerning the special focus of this report on ANN methods, it is interesting, that it was shown in the works of Connor et al. [1994] that feed-forward ANN are a special case of NAR models and that recurrent ANN are a special case of NARMA models. Chon & Cohen [1997] found an equivalence of ARMA and NARMA models with feed-forward ANN with polynomial transfer functions. Hence, despite the mentioned equivalences ANN are extremely valuable as they provide flexible means of implementing non-linear (AR)MA models [Maier & Dandy 2000]. For instance, an extension from univariate to multivariate cases is relatively easily performed, such that it might be preferable for the practitioner to use ANN approaches [Maier & Dandy 2000].

Other extensions of ARMA models comprise (generalized) autoregressive conditional heteroscedasticity ((G)ARCH) models e.g. Engle [1982], Bollerslev [1986], Engle [2001]. (G)ARCH models assume the variance of the current error term to be a

function of the variances of the previous time period's error terms. ARCH relates the error variance to the square of a previous period's error. If an ARMA model is assumed for the error variance, the model is a GARCH model. (G)ARCH type models have been widely used to model economic and financial time series for so-called volatility analysis. Their application in hydrology appears to be less frequent. For recent publications see e.g. Elek & Markus [2004], Wang et al. [2005].

If the time series are believed to be due to long memory processes then fractional ARIMA (FARIMA, sometimes called ARFIMA) modelling is more appropriate. (Seasonal) FARIMA models with long term memory were more recently applied to river flow data by Montanari et al. [1997, 2000] and Ooms & Franses [2001]. A hybrid model linking GARCH and FARIMA models for daily river flow simulation was given by Elek & Markus [2004].

3.4 Singular Spectrum Analysis

Singular spectrum analysis (SSA) stems from the 1980s interest in non-linear dynamics and chaos (e.g. Elsner & Tsonis [1996], Ghil et al. [2002]). It aims at extracting information from short and noisy time series in order to qualitatively analyse the dynamics of the series generating non-linear system and is an extension of principal component analysis (PCA) (e.g. Broomhead & King [1986], Vautard & Ghil [1989]). Unlike Fourier analysis it is not restricted to sinusoidal waves with constant amplitudes and may be used to extract trends and seasonality components. An application sought after in trend modelling could be the enhancement of the signal to noise ratio after signal reconstruction from its singular spectrum [Ghil et al. 2002] which is outlined hereafter:

In order to perform an SSA, the original time series of length *N* $\{X(t); t = 1,..., N\}$ needs to be decomposed into so-called *M*-dimensional (*M*<*N*) augmented vectors $\vec{X}(t) = [X(t), X(t+1),..., X(t+M-1)]$. These augmented vectors are indexed by $t = 1,..., N'; N' = N - M + 1$. The choice of *M* depends on a trade-off between the quantity of information extracted versus the degree of statistical confidence in that information, i.e. on one hand an *M* as large as possible, on the other hand as many repetitions of the features of interest as possible (implying a large *N*/*M* ratio) [Ghil et al. 2002]. *M* practically determines the largest periodicities that can be extracted. For

the *N*/*M* ratio Ghil et al. [2002] regard *N*/*M* = 11 as fairly save. For the decomposition procedure the approach of Broomhead & King [1986] is followed here[3], who construct the so-called trajectory matrix **D** which has the *N'* augmented vectors as its rows. Subsequently, they perform a singular value decomposition (SVD) of

$$\mathbf{C}_X = \frac{1}{N'}\mathbf{D}'\mathbf{D} \ ,$$

where $\mathbf{C}_X$ is the $M \times M$ covariance matrix of the (augmented) time series. Through the SVD the square roots of the eigenvalues λ_k of $\mathbf{C}_X$ – the so-called singular values (hence, the term 'singular spectrum analysis') – are obtained, whose actual values likewise determine the importance of their according eigenvectors ρ_k. The eigenvectors are analogous to the so-called empirical orthogonal functions (EOFs) which are commonly used in the meteorology literature (e.g. Preisendorfer [1988]). The projection of the time series onto each eigenvector or EOF yields the principal components (PCs) A_k or modes of the SSA:

$$A_k(t) = \sum_{j=1}^{M} X(t+j-1)\rho_k(j)$$

By visual inspection of the slope break in a diagram of singular values versus their rank k and/or the leading PCs (for an example see Figure 3-4 and Figure 3-5) a set Ξ of selected PCs may be formed to reconstruct parts of the signal by combination of the latter (after Ghil & Vautard [1991], Vautard et al. [1992]):

$$R_\Xi(t) = M_t \sum_{k\in\Xi} \sum_{j=L_t}^{U_t} A_k(t-j+1)\rho_k(j)$$

with

[3] see e.g. Allen & Smith [1996] for a discussion of alternative ways

$$(M_t, L_t, U_t) =$$
$$\begin{Bmatrix} \left(\frac{1}{t}, 1, t\right), & 1 \le t \le M-1 \\ \left(\frac{1}{M}, 1, M\right), & M \le t \le N' \\ \left(\frac{1}{N-t+1}, t-N+M, M\right), & N'+1 \le t \le N \end{Bmatrix}$$

For illustration, a synthetic example time series is analysed and reconstructed in the following. Figure 3-3 shows a time series based on the following equation for the discretely sampled time interval [0.01, 38] with a time step Δt of 0.01 days (roughly every 15 Minutes):

$$x(t) = t \cdot \cos(0.8t)\sin(5t) + \log_2(t)\cos(8t) + \log_2(t) + 0.5 \cdot t \cdot z(t)$$

where $z(t)$ is a realisation of a standard normal distribution ($z(t) \sim N(0,1)$) for each time step t. The time series represents sinusoidal waves with modulated amplitudes and an additive logarithmic trend. Moreover, its signal is contaminated with time dependent Gaussian noise. However, none of these features except the increasing amplitudes are easily determined from looking at the series (see Figure 3-3).

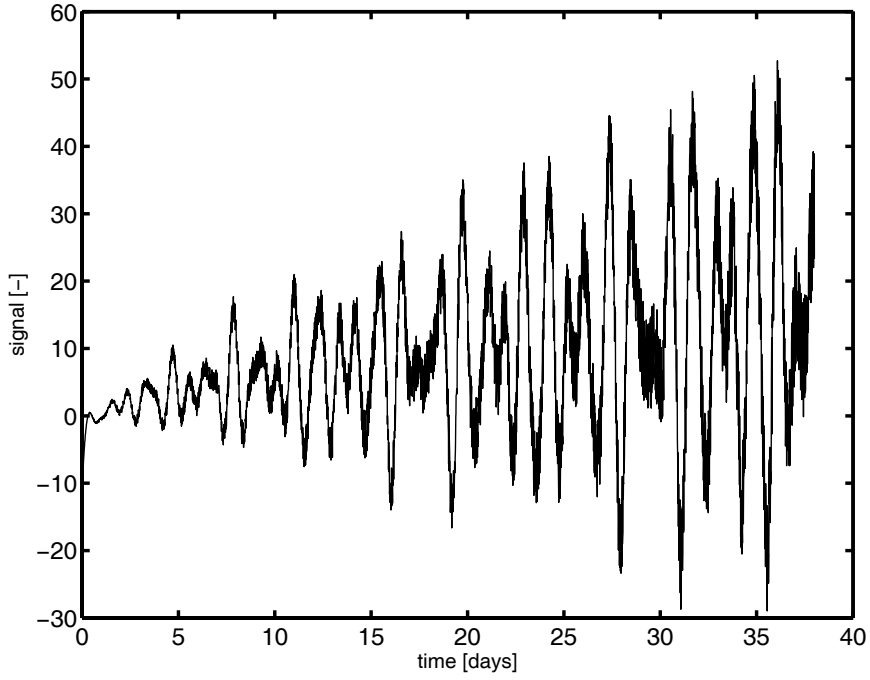

Figure 3-3: Synthetically generated time series with non-linear trend, frequency modulated amplitude and time-dependent white noise.

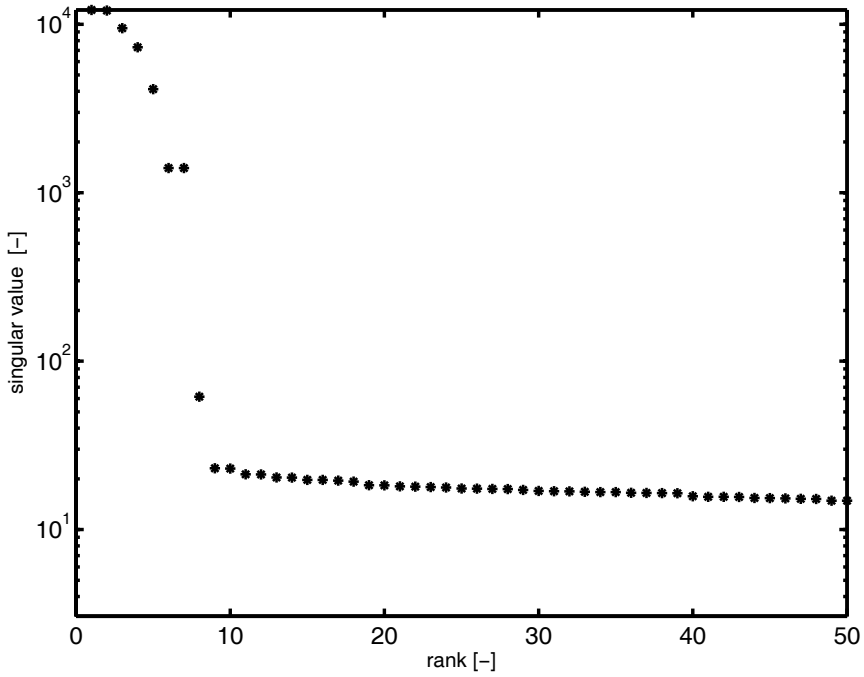

Figure 3-4: The first 50 singular values for the example time series

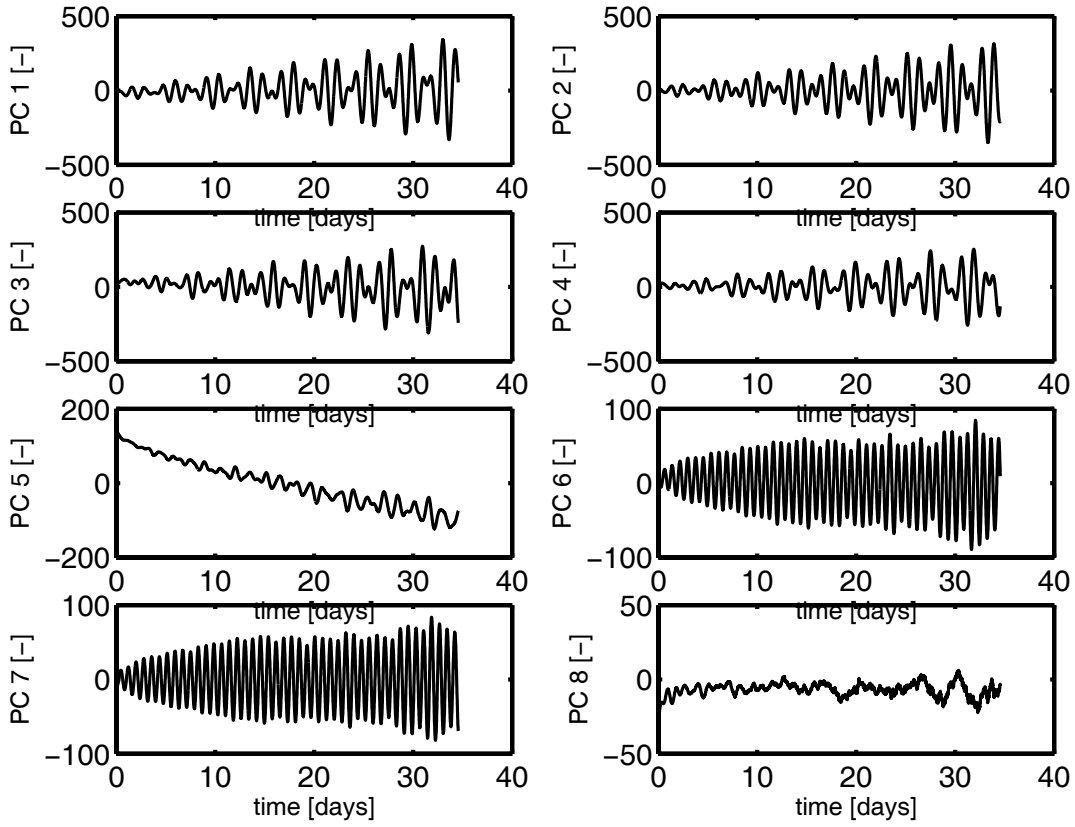

Figure 3-5: The first eight PCs for the example time series.

Figure 3-4 shows the singular value spectrum for the first 50 ranks, where the slope break at rank = 9 is easily determined. Therefore, the first eight PCs were selected (shown in Figure 3-5) for a closer examination and for use in signal reconstruction. As can be seen on Figure 3-5 PCs 1 to 4 and PCs 6-7 seem to contain the periodic components with increasing amplitudes, whereas PC 5 can be associated with a trend. PC 8 appears to capture the feature of time dependent noise, however in absence of a clear trend. In Figure 3-6 only PC 5 was used for signal reconstruction and yields – apart from the deviations on both ends of the data window – the (known) logarithmic trend.

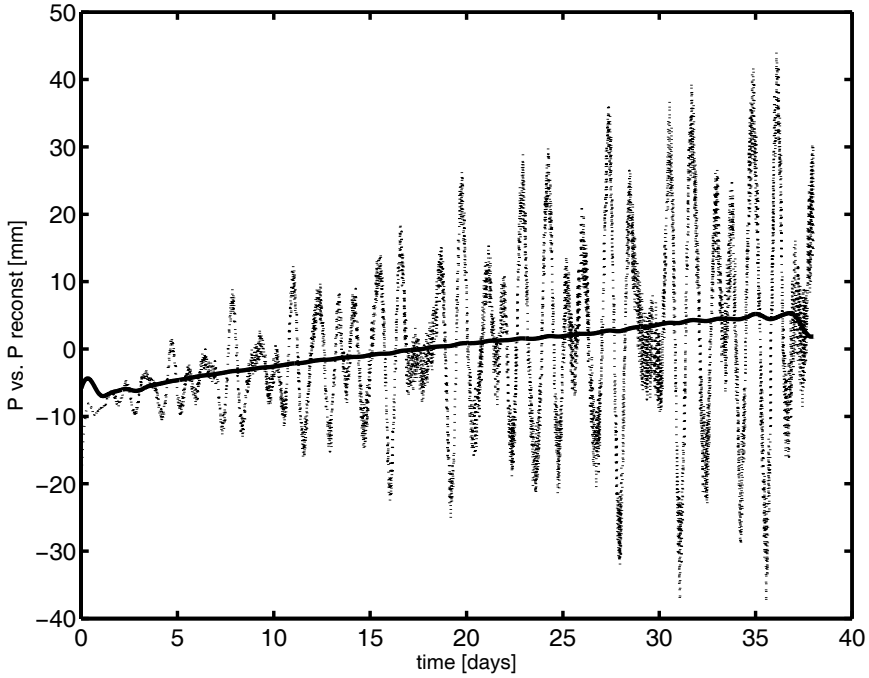

Figure 3-6: Reconstructed Trend from PC 5 (solid line). The original series is shown as a stippled line.

A close-up view of the full signal reconstruction from the first eight modes is shown in Figure 3-7. Obviously, the complex signal pattern can be very well modelled by a combination of these modes and the signal to noise ratio is clearly enhanced. Due to the smaller data coverage at both ends of the data window, the reconstruction slightly deviates from the signal there (not shown).

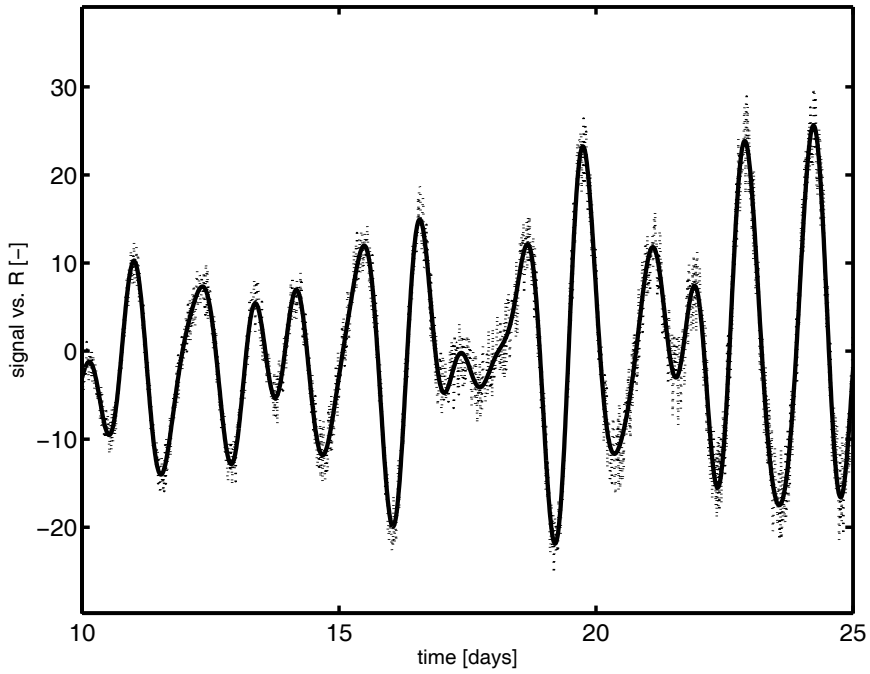

Figure 3-7: The signal (stippled) and its reconstruction (R, solid line) from the first eight SSA modes for the time intervall [10, 25].

In terms of making a forecast for the analysed time series, the idea is, that the reconstructed main components of the series constitute narrowband time series themselves. Therefore a these can be modelled and predicted fairly robust by fitting e.g. a low order AR(*p*) model. The original time series is then forecasted by the combination of the individual AR(*p*) model forecasts of the reconstructed main components [Keppenne & Ghil 1992; Vautard et al. 1992].

According to Ghil et al. [2002] the signal noise separation by inspecting the slope break of a singular value versus rank *k* works well, if the noise – potentially present in the natural time series – is white. However, if the noise should be red i.e. corresponds to an AR(1) process or is otherwise correlated between time steps difficulties arise [Vautard & Ghil 1989]. As a remedy the SSA method was extended to Monte Carlo SSA by Allen [1992] and Allen & Smith [1994, 1996], as well as an optimal filtering approach Allen & Smith [1997]. Further extensions involve multi-scale SSA in combination with wavelet analysis [Yiou et al. 2000] and the use of SSA for time series with missing data (SSAM) [Schoellhamer 2001]. SSA methods were also extended to multivariate analysis, where the term multichannel SSA (M-SSA) is used. The latter is closely related to extended EOF (EEOF) analysis used predominately in the meteorological literature [see Ghil et al. 2002]. A UNIX/Linux

based toolkit for SSA comprising at graphical user interface and plotting routines can be downloaded at http://www.atmos.ucla.edu/tcd/ssa/index.html.

More recently the limitations of (linear) SSA led Hsieh [2004] to the development of non-linear SSA (NLSSA) by use of ANN. According to Hsieh [2004] there are two limitations of conventional linear SSA, 1) the PCs from SSA are linearly uncorrelated (by definition). However, non-linear correlations may exist, which can be detected by NLSSA. 2) Although SSA modes are not restricted to sinusoidal oscillations (like Fourier analysis) the linear modes remain insufficient when modelling non-sinusoidal waves as – similar to Fourier analysis – energy is scattered into many SSA modes, which may be represented by a few non-linear modes. However, the advantages of NLSSA are highly data set dependent, i.e. NLSSA will only be effective, if the underlying physics are non-linear, the data record is long enough to support non-linear relationships and if the signal to noise ratio is sufficiently high. The original NLSSA–tools (comprising MATLAB code and manual) of William Hsieh can be downloaded at http://www.ocgy.ubc.ca/~william/.

3.5 Artificial neural networks

3.5.1 Introduction

The idea of artificial neural networks (ANN or more simple: neural networks, NN) dates back to McCulloch & Pitts [1943] who first proposed a mathematical model of a biological neuron. A motivation concerning ANN research back then was (and to some extent still is) the desire to understand the human brain and model its functioning. However, nowadays a variety of different network types exist, which do not necessary aim at human brain resemblance. Rather ANN may be viewed – though inspired by biological analogues – as a collection of mathematical tools for complex data mining tasks like pattern recognition, cluster analysis and classification, as well as non-linear regression and prediction. As opposed to physically-based models, ANN are data-driven models, which require important features (classification) or input-output relationships (regression) to be defined within the available data set.

Common elements of ANN comprise the following [after ASCE 2000a]:

1. Information processing occurs at many single elements called neurons, also referred to as units, cells, or nodes
2. Signals are passed between neurons through connection links
3. Each connection link has an associated weight that represents its connection strength
4. Each neuron typically applies a non-linear transformation called an activation function to its net input to determine its output signal.

The mathematical operation usually carried out by an individual neuron k to compute its output y_k is a weighted (weights w_{ik}) summation of the input signals x_i (i= 1,…,n) from all of its n connection links and the subsequent application of the activation function $\varphi(.)$ (see upper part of Figure 3-8):

$$y_k = \varphi\left(x_{0k} + \sum_i w_{ik} \cdot x_i \right),$$

where x_{0k} is an optional constant input.

Furthermore, the ANN is characterized by its architecture that represents the pattern of connection between neurons, its method of determining the connection weights, and the activation function [Fausett 1994]. Individual neurons are arranged in layers, which are themselves ordered hierarchically. The first layer (though not counted as a layer by some authors) is the input layer, where an input data vector enters the network and is distributed componentwise to the next layer. The next layer(s) could be an optional number (including zero) of so-called hidden layers and/or an output layer, where the calculated network output is presented (see lower part of Figure 3-8 for a three layer illustration). The number of neurons in the input and the output layer is determined by the respective number of input and output variables. The determination of the number of hidden layers and hidden layer neurons is part of the model building process and has to be answered problem-specific. Due to the usually complete connectedness between layers of neurons a massively parallel-distributed information processing is achieved [Haykin 1999].

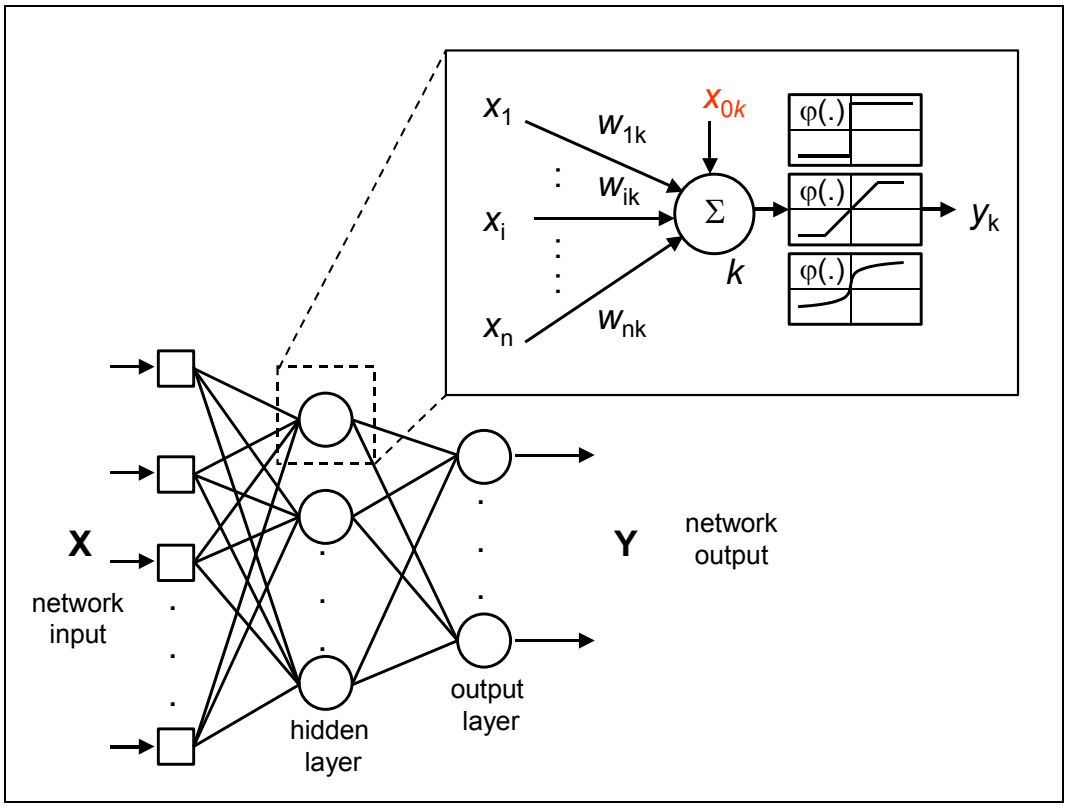

Figure 3-8: Schematic description of a three layer feed-forward ANN and of the elements of its (mathematical) neurons, i.e. the weighted sum of neuron inputs ($\Sigma w_i \times x_i$) and the subsequent - typically bounded, non-linear activation function φ(.). The variable x_{0k} is an optional constant neuron input.

A freely available, portable Java version of the famous SNNS (Stuttgart Neural Network Simulator) model system comprising numerous ANN types can be downloaded at http://www-ra.informatik.uni-tuebingen.de/downloads/JavaNNS/. The more commonly used architectures (e.g. feed-forward, self-organising maps, see below) are also implemented in the MATLAB neural-network toolbox.

3.5.2 ANN learning

There are several types of ANN learning with supervised and unsupervised learning probably being the most important. In general, an ANN learns or is trained by adjusting the connection weights parameters, that link the neurons [e.g. Dawson & Wilby 2001]. For classification and regression tasks supervised learning is used [ASCE 2000a], where the available data set consists of corresponding input and output values representing a characteristic pattern or underlying functional behaviour. This data set is the so-called training set. The adaptation of the weights is carried out by an optimisation algorithm that tries to minimise a difference or error measure between the ANN output based on the training set input values and their corresponding training set output value(s). The most renowned procedure in that respect is the so-called back-propagation algorithm (Werbos [1974], Rumelhart et al. [1986]), which is an analogue of steepest gradient descent optimisation [ASCE 2000a]. However, any optimisation method could be used for supervised learning (e.g. see Maier & Dandy [1999] for a comparison of five classic alternative optimisation methods in a salinity forecasting task). A recently favoured, particular class of optimisation algorithms are the so-called evolutionary algorithms (EA) (e.g. Pohlheim [1999]), which are based on the biological principles of heritage, mutation and selection. Typically, a (starting) population of feasible individuals (e.g. several candidate vectors of weight values) is generated randomly and a number of (good) individuals is selected to produce the next generation population. Through the selection mechanism the population evolves in time (here referring to the number generations of populations) towards the optimum (e.g. the population contains the weight vector producing the minimum error on the training set). By far the most widely applied EA are genetic algorithms (GA) (e.g. Goldberg [1989]). A helpful overall GA design methodology was developed by Reed et al. [2003]. Further recent applications concerning ANN are e.g. Jain & Srinivasulu [2004] and Rivas et al. [2004.]

In application to a time series, the regression task could be stated as follows: Either the input values consist of a vector of precedent time series values and the output value(s) consist of the current or predicted time steps,

$$(x_t,...,x_{t+s}) = ANN(x_{t-u},...,x_{t-u'}),$$
$$\text{with} \quad u,u' \in \mathbb{N} \setminus \{0\}; u' \le u; u-u' < t; s \in \mathbb{N}$$

(where *ANN*() denotes whatever function the ANN represents) or several input time series predict one or more output time series:

$$\left(y^1_t,...,y^1_{t+s(1)},y^2_t,...,y^2_{t+s(2)},...,y^j_t,...,y^j_{t+s(j)}\right) = ANN\left(x^1_{t-u(1)},...,x^1_{t-u'(1)},x^2_{t-u(2)},...,x^2_{t-u'(2)},...,x^i_{t-u(i)},...,x^i_{t-u'(i)}\right),$$
$$\text{with} \quad i,j,u(i),u'(i) \in \mathbb{N} \setminus \{0\}; u'(i) \le u(i); u(i)-u'(i) < t; s(j) \in \mathbb{N}$$

In unsupervised learning only an input data set is used without a corresponding output set. The specialised types of ANN used in unsupervised learning are employed for (nonparametric) density estimation of the underlying distribution of a data set or for projecting patterns from a high-dimensional to a low-dimensional space [ASCE 2000a]. However, despite the usefulness of unsupervised learning ANN within the process of building ANN-based hydrological catchment or basement models we will only dedicate a short section (3.5.5) to this interesting topic, as this report is primarily concerned with regression tasks.

3.5.3 ANN architectures

3.5.3.1 Feed-forward ANN

One apparent classification scheme of ANN is based on the flow of information through the network [ASCE 2000a]. In the so-called feed-forward type information is passed on from the input layer through the hidden layers to the output layer with no back-looping to preceding layers or exchange among neurons within one layer. Probably the most widely applied feed-forward networks and the classic ANN are multi-layer perceptrons (MLPs), which have at least one hidden layer of neurons (see Figure 3-8) and use a non-linear, sigmoidal-shaped activation function like the logistic φ_l or hyperbolic tangent function φ_{tanh}:

$$\varphi_l = \frac{1}{1+e^{-x}}$$

$$\varphi_{\tanh} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

MLPs are known to be universal function approximators [Hornik et al. 1989], i.e. for a large enough number of hidden neurons a three layer MLP can approximate any mapping from a set of input vectors to the corresponding set of output vectors[4]. However, no statement is made by this regarding the optimality of the approximation with respect to network size, time needed to fit or train the ANN model or the validity of the approximated function between the fitted data points [Haykin 1999] (see section 3.5.4 for this purpose).

Another feed-forward ANN is the radial basis function network (RBF) which uses Gaussian basis functions φ_G in the hidden layer (typically with only a single hidden layer) and a linear activation function in the output layer [Dawson & Wilby 2001]:

$$\varphi_G = \exp\left(-\frac{x^2}{2\sigma^2}\right),$$

[4] The available data set is usually (linearly) mapped into the interval [0,1] or [-1,1] for that purpose (e.g. Maier & Dandy [2000])

where σ^2 is the sphere of influence or width of the basis function. Apart from the connection weight parameters, also the sphere of influence parameter needs to be established in the RBF model building process (usually a clustering algorithm that acts upon the training set is employed for that purpose). Hence, the basic difference to MLPs is the handling of non-linearities – MLPs use a fixed activation function, whereas an RBF bases its non-linearities on the data in the training set [ASCE 2000a]. An advantage of RBF is that the linear activation functions in the output layer allow the weight computation by direct matrix inversion, such that it is faster to train than a comparable MLP [Dawson & Wilby 2001].

As mentioned above, feed-forward ANN have been widely applied by the hydrology community and spawned a number of review papers, which provide information and insight on further modelling issues. We therefore would like to point the interested reader to ASCE [2000a, b], Maier & Dandy [2000] and Dawson & Wilby [2001] for details on input variable selection, training set selection, optimisation algorithms, further feed-forward ANN variants, and references.

3.5.3.2 Recurrent ANN

An obvious extension of feed-forward ANN is to allow connection links that loop back to the input layer, such that they provide for a feedback. According to ASCE [2000a] there are three ways to introduce “memory” into static ANN (in increasing order of complexity and capability):

1. tapped delay line models: Past inputs are explicitly available for the ANN to determine its response at a given point in time (e.g. Haykin [1999]).
2. context or partial recurrent models: past outputs (of hidden or output neurons) are fed into the ANN along with the current inputs (e.g. Elman [1990])
3. fully recurrent models: complete feedback and interconnection between all nodes (e.g. Pineda [1987, 1989], Williams & Zipser [1989])

In the following we will use the term ‘recurrent’ in order to refer to models of type 3. or 2. Apart from being able to model NARMA processes (see Connor et al. [1994],

Fortin et al. [1997] and section 3.3), recurrent ANN provide the advantage of dynamic modelling, which is relevant in hydrology in the context of real-time forecasting. More specifically, a data stream may be fed to recurrent ANN online allowing for forecasts based on the latest available information. Another advantage of recurrent ANN is that they are supposed to perform better in the presence of noise (see Gencay & Liu [1997]). However, according to Maier & Dandy [2000] it were feed-forward ANN, that have almost exclusively been used for prediction and forecasting of water resources variables. This might be due to the fact, that fitting recurrent ANN models is considerably more time consuming and does involve more complex algorithms than fitting static feed-forward ANN. Furthermore, Lin et al. [1996] found that recurrent ANN due have difficulties when long-term dependencies become important, i.e. when inputs at high lags should have a significant impact on the ANN output. Along these lines Hochreiter & Schmidhuber [1997] state, that recurrent ANN“ …do not provide clear practical advantages over …feedforward nets with limited time windows”. Nevertheless, Chiang et al. [2004] more recently compared recurrent ANN with MLPs in a rainfall-runoff modelling study for the Lan-Yang river in Taiwan and found that the performances of both ANN types differed according to the length of the data record used for fitting. In the case of adequate record lengths, the MLP performance was slightly better in terms of the output errors on the test data, whereas the recurrent ANN better captured the peak flows. However, for the short data record cases, recurrent ANN performed significantly better than the feed-forward ANN. An explanation may be offered by the popular No-Free-Lunch theorems (e.g. Magdon-Ismail [2000]) essentially stating that for each task on which a particular learning algorithm performs extraordinarily well, there exist as many tasks where it fails.

3.5.3.3 Support vector machines

Support vector machines (SVM) [Vapnik, 1995, 1998] are a relatively new, highly principled generalisation of feed-forward ANN, which incorporate a constraint of the SVM model capacity[5] based on the available data into the learning task. The underlying principles are rooted in the so-called statistical learning theory (SLT) which was developed by Vapnik and Chervonenkis (VC) in 1974 [Dibike et al. 2001].

In the following an introduction to the SVM learning algorithm for regression tasks is given according to Dibike et al. [2001]. Let the observed variable be a real value scalar y and $f(\mathbf{x},\alpha)$ with $\alpha \in \Lambda$ (with Λ defined as in 3.1) be a set of real functions containing the regression function $f(\mathbf{x},\alpha_0)$. If the problem were to approximate the finite data set X (size n) with a linear function $f(\mathbf{x},\alpha) = (\mathbf{w} \bullet \mathbf{x}) + b$ (where $\mathbf{w}$ (weights) and b are adjustable free parameters and $\bullet$ denotes a scalar product), the optimal regression function is given by minimizing the empirical risk:

$$R_{emp}(\mathbf{w},b) = \frac{1}{n}\sum_{i=1}^{n} \left| y_i - f(\mathbf{x}_i,\alpha) \right|_\varepsilon$$

using the so-called ε-insensitive loss function (Vapnik [1995]) defined as:

$$\left| y_i - f(\mathbf{x}_i,\alpha) \right|_\varepsilon = 0, \qquad \text{if} \quad \left| y_i - f(\mathbf{x}_i,\alpha) \right| \le \varepsilon$$
$$\left| y_i - f(\mathbf{x}_i,\alpha) \right|_\varepsilon = \left| y_i - f(\mathbf{x}_i,\alpha) \right| \qquad \text{otherwise}$$

The ε-insensitive loss function allows for the incorporation of a pre-defined, un-penalised deviation of the SVM outputs from the training data observations (see Figure 3-9 for an illustration).

---

[5] Please note, that there is a distinction between model complexity and capacity. Complexity usually refers to the number of free parameters of a model function, whereas the capacity is measured by the so-called VC dimension. However, a detailed treatment of the VC dimension is beyond the scope of this report (see e.g. Vapnik [1995, 1998])

Now the objective is to find a function $f(\mathbf{x},\alpha)$ that has at most a deviation of ε- from the observed y_i and is as flat as possible. Mathematically speaking the functional Φ has to be minimised:

$$\Phi\left(\mathbf{w},\xi^*,\xi\right)=\tfrac{1}{2}\left\|\mathbf{w}\right\|^2+C\left(\sum\xi_i^*+\sum\xi_i\right) \qquad (*)$$

where C is a pre-specified value defining the cost of constraint violation and ξ^*, $\xi \geq 0$ are slack variables, which represent upper and lower constraints on the outputs of the SVM as follows:

$$y_i-\left(\mathbf{w}\square\mathbf{x}_i+b\right)\leq\varepsilon+\xi_i, \qquad i=1,...,n$$
$$\left(\mathbf{w}\square\mathbf{x}_i+b\right)-y_i\leq\varepsilon+\xi_i^*, \qquad i=1,...,n$$

From there the Lagrange function is constructed out of the objective function and the inequality constraints by introducing a dual set of variables:

$$\mathbf{L}=\tfrac{1}{2}\left\|\mathbf{w}\right\|^2+C\left(\sum_{i=1}^{n}\left(\xi_i^*+\xi_i\right)\right)-\sum_{i=1}^{n}\alpha_i\left[\varepsilon+\xi_i-y_i+\left(\mathbf{w}\square\mathbf{x}_i\right)+b\right]$$
$$-\sum_{i=1}^{n}\alpha_i^*\left[\varepsilon+\xi_i+y_i-\left(\mathbf{w}\square\mathbf{x}_i\right)-b\right]-\sum_{i=1}^{n}\left(\eta_i\xi_i+\eta_i^*\xi_i^*\right)$$

From the saddle point condition of constraint optimisation theory it follows, that the partial derivatives of **L** with respect to the primary variables (**w**, b, ξ_i^*, ξ_i) have to vanish for optimality. Substituting the results of this derivation into (*) results in the so-called dual optimisation problem:

$$W(\alpha^*,\alpha)=-\varepsilon\sum_{i=1}^{n}\left(\alpha_i^*+\alpha_i\right)+\sum_{i=1}^{n}y_i\left(\alpha_i^*-\alpha_i\right)-\tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_i^*-\alpha_i\right)\left(\alpha_i^*-\alpha_i\right)\left(\mathbf{x}_i\square\mathbf{x}_j\right) \qquad (**)$$

which has to be maximised subject to the constraints

$$\sum\alpha_i^*=\sum\alpha_i; \qquad 0\leq\alpha_i^*\leq C; \qquad 0\leq\alpha_i\leq C$$

for i = 1,…,n.

Once the coefficients α^*_i and α_i are obtained from (**) by a quadratic optimisation routine, the corresponding support vectors (coining the name SVM) are obtained as

the only data points $\mathbf{x}_i$ with non-zero Lagrangian multipliers α^*_i and α_i. The according optimal linear regression function is then:

$$f(\mathbf{x}) = \sum_{\text{support vectors}} \left(\alpha_i^* - \alpha_i\right)\left(\mathbf{x}_i \square \mathbf{x}\right) + b_0$$

where

$$b_0 = -\tfrac{1}{2}\mathbf{w}_0 \cdot \left[\mathbf{x}_r + \mathbf{x}_s\right]$$
$$\mathbf{w}_0 = \sum_{\text{support vectors}} \left(\alpha_i^* - \alpha_i\right)\mathbf{x}_i$$

with $\mathbf{x}_r$ and $\mathbf{x}_s$ being any support vectors (in practice, one often averages over all support vectors found [Asefa et al. 2004]).

Please note, that the final optimal regression function is unique in the sense that the (convex) optimisation problem is guaranteed to converge to a unique global optimum. It follows, that the number of support vectors and, hence, the capacity of the SVM is also a property of the uniquely defined optimum.

Nevertheless, the regression tasks of interest in hydrology are usually highly non-linear. The remedy used in case of SVM is the transformation of the input data by a – generally non-linear – mapping Γ: $X \rightarrow F$. Γ is applied to the input space X in order to 'flatten out' the underlying functional relationship in a higher dimensional feature space F, such that it can be represented by a (optimal) linear regression function in F. Intuitively, this does not seem to be an advantage, as the mapping Γ first needs to be found and a higher dimensionality usually comes at the (exponentially) higher cost of memory space and computation time. This is where the so called 'kernel-trick' comes into play. In its most general form stated by Schölkopf et al. [1998] *any* algorithm depending exclusively on scalar products in F can be expressed by an admissible kernel[6] $\mathbf{K}(\mathbf{x},\mathbf{x}')$ in X, meaning that

[6] 'admissible kernel', mathematically speaking means that $\mathbf{K}(\mathbf{x},\mathbf{x}')$ has to be the continuous kernel of a positive integral operator on a Hilbert space $L_2(C)$ on a compact set $C \subset \Re^n$ (see e.g. Schölkopf & Smola [2002] chapter 2 for a more profound treatment of the subject).

$$\mathbf{K}(\mathbf{x},\mathbf{x}') = \Gamma(\mathbf{x})\Gamma(\mathbf{x}')$$

The implication of this result is, that Γ neither has to be known, nor has the scalar product to be evaluated in the higher dimensional space *F*. Hence, for non-linear regression, the parameters $\mathbf{w}_0$ and b_0 building the 'linear' regression function $f(\mathbf{x}) = (\mathbf{w}_0 \bullet \mathbf{x}) + b_0$ are found by

$$b_0 = -\tfrac{1}{2} \sum_{\text{support vectors}} \left(\alpha_i^* - \alpha_i\right)\left[\mathbf{K}\left(\mathbf{x}_r, \mathbf{x}_i\right) + \mathbf{K}\left(\mathbf{x}_s, \mathbf{x}_i\right)\right]$$

$$\mathbf{w}_0 \mathbf{x} = \sum_{\text{support vectors}} \left(\alpha_i^* - \alpha_i\right)\mathbf{K}\left(\mathbf{x}_i, \mathbf{x}\right)$$

Proven admissible kernels are given in Table 3-2. A conceptual illustration of the kernel-trick is given in Figure 3-9.

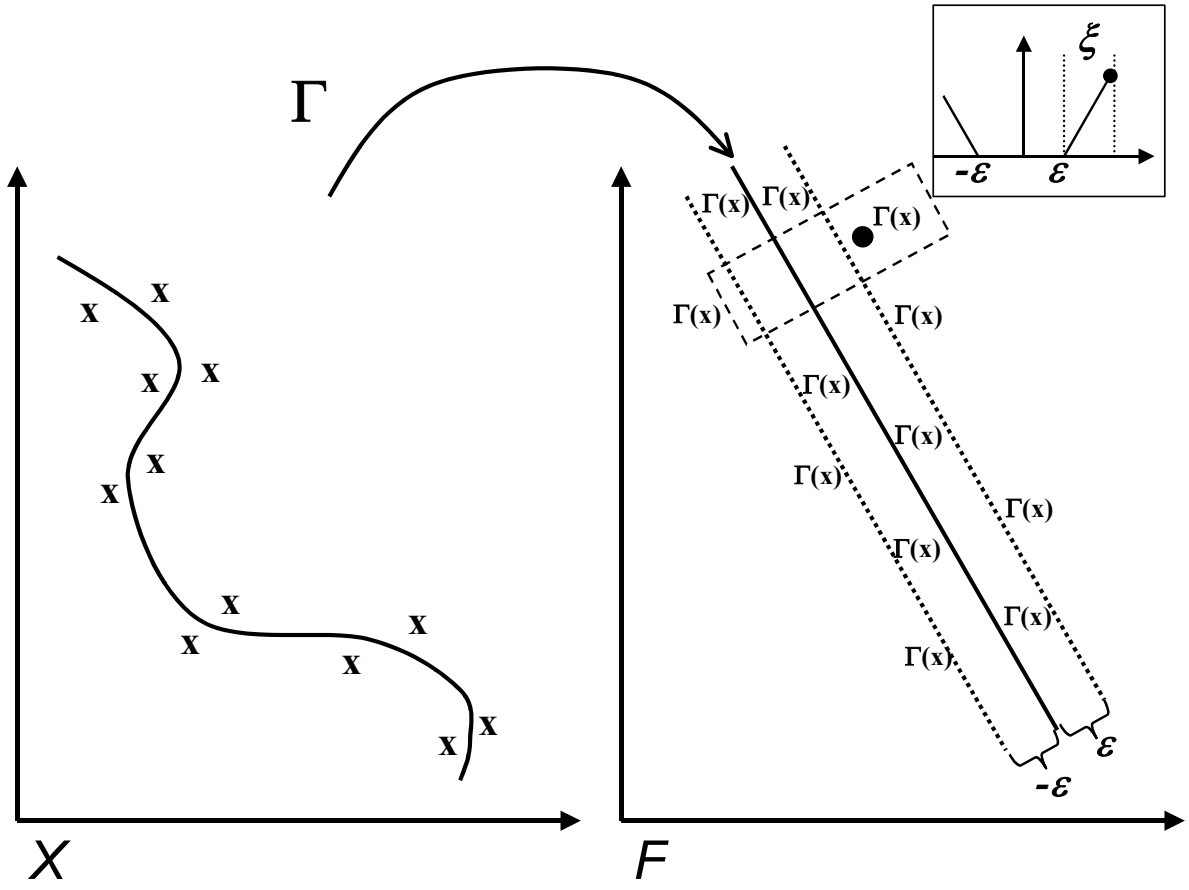

Figure 3-9: Conceptual illustration of the kernel-trick and the ε-insensitive loss function for an arbitrary data point (black circle) (after Schölkopf & Smola [2002]).

Hence, it is the choice of a particular kernel and the values of *C* and ε, where the user may influence the SVM building process. The lower the value of *C*, the more

weight is given to the regulariser ($\|\mathbf{w}\|$). Suggestions, on how to determine a suitable value for *C* can be found in e.g. Cristianini & Shawe-Taylor [2000] or Saunders et al. [1998]. Interestingly, the determination of ε can also be incorporated into the optimisation process, by a slight modification of the objective function (*). In this so-called ν-SVM regression [Schölkopf et al. 2000]), the user specifies an upper bound $0 \le \nu \le 1$ on the fraction of points allowed to lie outside the ε-tube and the value of ε is computed automatically minimising:

$$\Phi\left(\mathbf{w}, \xi^*, \xi, \varepsilon\right) = \tfrac{1}{2}\|\mathbf{w}\|^2 + C\left(\nu\varepsilon + \tfrac{1}{n}\sum_{i=1}^{n}\left(\xi_i^* + \xi_i\right)\right)$$

An additional advantage of ν-SVM regression is that ν is more physically interpretable in various applications [Schölkopf et al. 2000] and that the asymptotically optimal value for ν can be found, when a suitable noise model (i.e. an estimate of the distribution function of the noise) for the problem exists [Smola 1998]. A complete object-oriented environment for machine learning in MATLAB (The Spider) comprising SVM among a large quantity of alternatives can be downloaded from http://www.learning-with-kernels.org/.

Table 3-2: Admissible kernels found in Schölkopf & Smola [2002] and Dibike et al. [2001].

Formula	Kernel **K**(**x**,**x**')[7]
$\left[\left(\mathbf{x}\cdot\mathbf{x}'\right)+1\right]^d$	simple polynomial kernel of order *d* (user defined)
$\exp\left(-\lambda\left\|\mathbf{x}-\mathbf{x}'\right\|^2\right)$	radial basis function kernel (λ user defined)
$\tanh\left(b\left(\mathbf{x}\cdot\mathbf{x}'\right)-c\right)$	neural network kernel (*b* and *c* user defined
$\frac{1}{\sqrt{\left\|\mathbf{x}-\mathbf{x}'\right\|^2+c^2}}$	inverse multiquadric kernel (*c* user defined)
$-\sqrt{\left\|\mathbf{x}-\mathbf{x}'\right\|^2+c^2}$	multiquadric kernel (*c* user defined)
$\left\|\mathbf{x}-\mathbf{x}'\right\|^{2n}\ln\left\|\mathbf{x}-\mathbf{x}'\right\|$	thin plate spline kernel (*n* user defined)

SVM have excelled in a number of pattern recognition and regression tasks from other fields [e.g. Schölkopf et al. 1999], yet, they have only recently been discovered for application in hydrology. Liong & Sivapragasam [2000] issued SVM a superior

[7] Please, note that some of these kernels are not admissible for all possible values of the user defined parameters (see Schölkopf & Smola [2002]).

performance over conventional feed-forward ANN in flood stage forecasting. Dibike et al. [2001] used SVM for rainfall-runoff regression finding the same superiority. More recently, Asefa et al. [2004] employed an SVM to estimate spatially redundant monitoring wells and applied the methodology to the design of a long-term ground water head monitoring network in the Water Resources Inventory Area 1, Whatcom County, Washington State, USA. In Asefa et al. [2005a] it is shown that the SVM could also aid in designing a contaminant concentration monitoring network. A recent application to basin scale water management comprising ANN is found in Khalil et al. [2005a]. Moreover, Khalil et al. [2005b] have used - among other learning machines - SVM and an extension called relevance vector machine (RVM) [Tipping 2000, 2001] to replace complex process-based numerical models for nitrate concentration forecasting in ground water at specified receptors. The authors report strong predictive capabilities of the established learning machine models and state savings concerning modelling efforts and improved performance relative to process-based models. Most recently, Asefa et al. [2005b] also applied the SVM approach for multi-time scale stream flow predictions using a local scale climatological model that required far less input than a physically-based model.

3.5.4 ANN model validation

As mentioned above ANN are data driven models, that perform a classification or regression task without knowledge of the underlying process physics. In the following we will consider the ANN to be employed for a regression task and a MLP architecture is assumed for the sake of simplicity (However, the validation methods are not specific to MLP, rather not even to ANN in general).

An advantage of MLPs is their high potential to approximate any underlying functional relationship that might be present within a data set. However, in order to establish a well generalising MLP, i.e. to find the optimal function approximating this functional relationship, rather than just the data points, some general model building principles have to be followed. The problem arises from the fact, that the available finite amount of data could be fit by various ANN models differing in size and topology, i.e. the number of hidden neurons, the number of hidden layers and the connectivity among them. On the one hand, it is a practical advantage of MLPs, that model complexity is easily varied by adding (hidden) neurons providing (additional)

function classes to fit the data (see e.g. Maier & Dandy [2000]). On the other hand the ANN complexity or the number of function classes implementable by the MLP has to be restricted in accordance with the available data set. A simple illustration example of this generalisation problem is given in Figure 3-10.

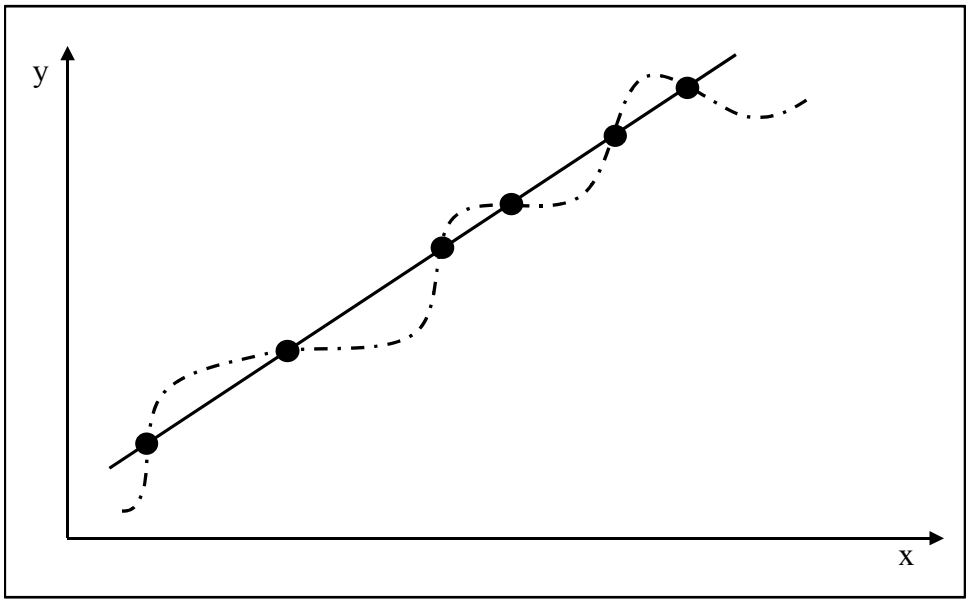

Figure 3-10: Illustration of the problem of good generalisation. Both curves are regression functions that perfectly fit the data points (filled circles). However, the complex function (dot-dashed) does not generalise well, as it fits the data points, but does not capture the underlying linear functional relationship.

Furthermore, the data could be contaminated with noise, such that an overly complex MLP fits the noise (overfitting) rather than the masked underlying functional relationship (e.g. Hsieh [2004]). In the following we cover some techniques to establish a sound compromise between optimal data fitting and model complexity. For this purpose we will also introduce some notation concerning learning tasks [Shakhnarovich et al. 2001], which should prove to be helpful.

A data set X containing N input-output data vector pair elements $<\mathbf{x}_i,\mathbf{y}_i>=\chi_i \in X$ (also denoted by $X^{(N)}$ to incorporate the set size) is assumed to be the result of the generation of the χ_i identically and independently distributed in a d-dimensional data space D, according to an unknown probability distribution Z. Furthermore, we use the shorthand convention that $L(A,B)$ denotes some sort of error function L (e.g. the

MSE) on a set *A* of a learning algorithm trained on set *B*. The true error *Err* (or generalisation error) for a fixed training set $X^{(n)}$ can then be written as:

$$Err\left(X^{(n)}, Z\right) = \int_D L\left(\mathbf{x}, X^{(n)}\right) Z\left(\mathbf{x}\right) d\mathbf{x}$$

As *Z* is unknown, *Err* is unknown. However, it is the purpose of validation methods to find good estimates for *Err*, which in turn are used to evaluate different MLP topologies. The simplest estimate would be to compute the error on the same set that an MLP was trained on, e.g. *X*. The resulting so-called training error (also called empirical or resubstitution error) could be arbitrarily small for a large number of hidden neurons. Nevertheless, it would give an overly optimistic estimator for *Err* (Shakhnarovich et al. [2001]). The training error is also said to be downward biased. In the following we cover some improved validation methods.

3.5.4.1 Split-sample validation

In split-sample validation (or "hold out" validation) a division of *X* into a training set $X^{(n)}$ and validation set $X^{(m)}_{\text{val}}$ is performed (with $X^{(n)} \cap X^{(m)}_{\text{val}} = \varnothing$, where $\varnothing$ denotes the empty set). Based on the training data set $X^{(n)}$ and for a given MLP topology this MLP model's connection weight parameters are established by minimising some error measure of the MLP predicted output $f_\alpha(\mathbf{x}_i)=\mathbf{P}_i$, (with $\alpha \in \Lambda$, with Λ representing a set of possible weight values) versus the output according to the training data $\mathbf{y}_i$. For instance, the training error for the MLP based on the *MSE* error measure is then calculated as follows (where *SE* denotes the squared error):

$$SE\left(X^{(n)}, X^{(n)}\right) = \sum_{i=1}^{n} (\mathbf{P}_i - \mathbf{y}_i)^t (\mathbf{P}_i - \mathbf{y}_i)$$

$$MSE\left(X^{(n)}, X^{(n)}\right) = \frac{1}{n} SE\left(X^{(n)}, X^{(n)}\right)$$

As the minimisation of the training error proceeds, the validation error, here $\text{MSE}(X^{(m)}_{\text{val}}, X^{(n)})$, of the MLP is monitored. Once the validation error starts indicating a trend reversal (from decreasing to increasing) this is taken as a sign of the beginning of overfitting and the optimisation is stopped (e.g. Maier & Dandy [2000], Dawson & Wilby [2001]). If the training is carried out for an number of MLPs with

different topologies the sum of validation and training error after stopping the minimisation can be used to find the model which best generalises best.

It is easy to see that this procedure is demanding regarding the amount of available data in the first place. Moreover, it is commonly agreed, that in order to minimise the generalisation error all important features of the underlying functional relationship have to be represented within the training data set [Maier & Dandy 2000]. ANN performance on validation data outside the range of the training data is expected to be poor [Minns & Hall 1996]. Accordingly, split-sample validation needs a large amount of data and suffers from training data reduction and/or from the problem of the selection of an optimal training set (see ASCE [2000b]) to obtain optimal generalisation.

3.5.4.2 Cross-validation

The recognition of the data demand and the optimal training set problem in machine learning has led to a shift to alternative techniques to estimate *Err* of a particular learning algorithm on a given data set.

A well-known remedy is the so-called leave-one-out cross-validation. More recently, a generalisation of leave-one-out was theoretically explored in depth by Blum et al. [1999], the so-called *k*-fold cross-validation. Here the data set X is split into k mutually exclusive subsets S_j with $j = 1,\ldots,k$. For each S_j one MLP out of k MLPs with the same fixed topology is trained on all subsets except for S_j. The resulting true error estimate is then computed as the average error over the k trained MLPs:

$$Err_{k-fold} = \frac{1}{k}\sum_{j=1}^{k} L(S_j, X \setminus S_j)$$

where $X \setminus S_j$ denotes the difference set of X and S_j. For $k = n$ $Err_{k\text{-fold}}$ produces the leave-one-out error estimate of *Err*. The advantage of *k*-fold cross-validation is, that the MLP is trained on all data, so that no optimal training set has to be selected. Blum et al. [1999] have shown that *k*-fold cross-validation provides for a better generalisation error estimate than testing on a validation data set comprising $1/k$–th of the available data (for $k > 2$). The same authors also have proven that the *k*-fold estimate is strictly more accurate than the leave-one-out estimate. A favourable MLP topology can be selected by comparing their individual $Err_{k\text{-fold}}$. For the best

generalising topology the optimal weight values are then found by (an additional) training on the whole data set *X*.

Obviously, the mentioned advantages come at the cost of an increased amount of training, i.e. computation time, as *k* MLP weight parameter sets have to be found for a single fixed topology. Although the $Err_{k\text{-}fold}$ estimate is regarded to be almost unbiased [Efron & Tibshirani 1993], it may exhibit a high variance, especially if the available data set size becomes too small or $k = n$, i.e. leave-one-out cross-validation is performed (Kohavi [1995], Efron [1983]).

3.5.4.3 Bootstrapping

Another less data intensive, but even more computation time consuming approach is bootstrapping, a resampling technique that avoids the potentially high variance of the cross-validation error estimates in the case of small data sets. The basic idea of bootstrapping is to resample the original data set with replacement and to use the resampled sets to estimate the generalisation error *Err*. However, bootstrap error estimates usually come at the cost of a higher bias [Efron & Tibshirani 1993]. An advanced bootstrap estimator (the so-called .632+ estimator) was developed by Efron & Tibshirani [1997] which is supposed to have almost no bias with a low variance:

1. Starting with a loop over the original data set *X* successively each individual data vector pair χ_i with $i = 1,\ldots N$ is removed from *X* to form N data sets $X\backslash\chi_i$ of *N*-1 data vector pairs.

2. From each of the *N* data sets $X\backslash\chi_i$ there are *B* samples of size *N*-1 drawn randomly with replacement to form the resampled data sets $X\backslash\chi_i{}^*{}_b$, with $b = 1,\ldots,B$. The asteriks * hereby denotes the resampled data sets.

3. Assume a fixed MLP topology for the remaining steps. Then $N \times B$ MLP models are trained on the $N \times B$ data sets $X\backslash\chi_i{}^*{}_b$ (one MLP for one data set) and the so-called leave-one-out bootstrap $Err^{(1)}{}_{BS}$ is computed:

$$Err_{BS}^{(1)} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{B}\sum_{b=1}^{B} L(\chi_i, X \backslash \chi_i{}^*{}_b)$$

4. The .632 estimator for the ANN model is then given as a hybrid estimator between the leave-one-out bootstrap estimator, which is regarded to have an upward bias [Shakhnarovich et al. 2001], and the downward biased training error on the original data set *X* (requiring one additional training of an MLP).

$$Err_{.632} = 0.632 Err_{BS}^{(1)} + \left(1 - 0.632\right) L(X, X)$$

(The fraction 0.632 is based on the heuristic, that a bootstrap sample of size N is expected to be supported by approximately 0.632N original data points for large N.)

5. The .632+ estimator is obtained by incorporating an overfitting estimate and adjusting the 0.632 value accordingly. For that purpose Efron & Tibshirani [1997] define the "no-information" error rate γ and estimate it by averaging the error of the MLP trained on *X* on a test set comprised of all possible permutations of the data vector pairs in *X* (i.e. statistical independence between inputs and outputs):

$$\gamma = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} L(\left\langle \mathbf{x}_i, \mathbf{y}_j \right\rangle, X)$$
$$\left\langle \mathbf{x}_i, \mathbf{y}_i \right\rangle = \chi_i \in X$$

6. The resulting relative overfitting rate *R* is derived as

$$0 \le R = \frac{Err_{BS}^{(1)} - L(X, X)}{\gamma - L(X, X)} \le 1,$$

which yields *R* = 0 for the not overfitting MLP topology and a maximum of overfitting where *R* = 1.

7. Finally, the .632+ estimator is defined as:

$$Err_{.632+} = \omega Err_{BS}^{(1)} + \left(1 - \omega\right) L(X, X)$$
$$\omega = \frac{0.632}{1 - 0.368R}$$

A special treatment of cross-validation and bootstrapping regarding time series data is given in Hjorth [1994]. More recently, Lendasse et al. [2003] found that the .632(+) estimators were superior to cross-validation in a number of time series modelling experiments with RBF. Further extensions of the .632+ bootstrap can be found in [Shakhnarovich et al. 2001].

As a final statement it should be noted that if the prediction or forecasting error of the MLP model should be estimated, an exclusive set of test data has to be retained, as e.g. in case of split-sample validation, the training and validation data have already been used to decide upon the model topology and the stopping of the training process (see Dawson & Wilby [2001], Hsieh [2004]).

3.5.5 Further ANN architectures

As pointed out in section 3.5.2 ANN performing unsupervised learning are somewhat different from the ones presented so far. In a brief section the so-called self-organised maps (SOM, Kohonen [1989, 1990]), sometimes called Kohonen-maps or feature maps are introduced. In the SOM architecture all neurons are arranged in a two-dimensional (sometimes three-dimensional) lattice (see Figure 3-11 for an illustration). Each neuron has an associated weight vector of the same dimension as the input data vector. The neurons compete with each other, in the sense that during the training phase a similarity or distance measure between the current input vector and each neuron's associated weight vector is computed. The most similar or closest neuron is the winning neuron and its weight vector is updated by a pre-defined learning rate to become even more similar to the input vector. Moreover, the neighbouring neurons of the winning neuron are also updated by a fraction of the learning rate, which depends on the lattice distance of the neighbouring neurons to the winning neuron. By performing this competition over a large number of input vectors, it can be shown, that the weight vectors in the neuron lattice form clusters resembling the main features of the data set and their neighbourhood relationships or topological order. In the two-dimensional case a so-called feature map evolves.

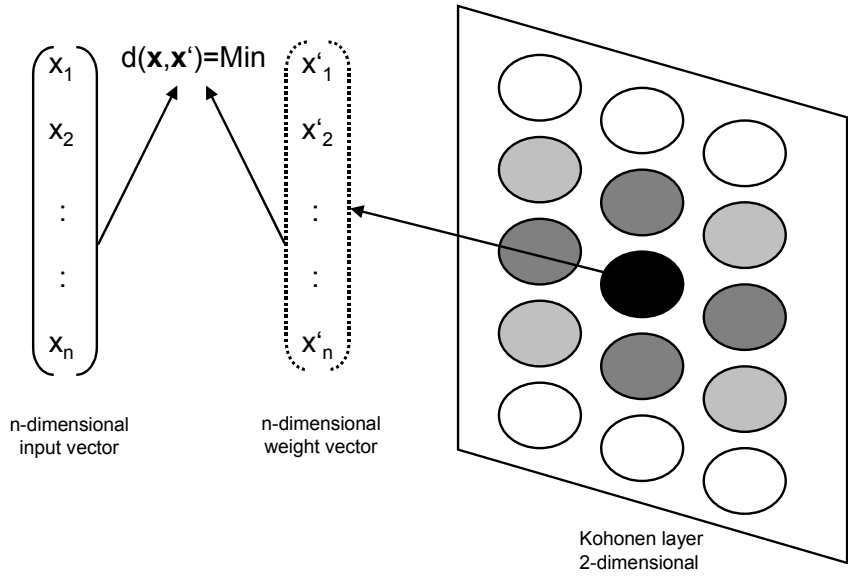

Figure 3-11: Illustration of a SOM. The winning neuron (black circle) is determined as the one with the minimum distance measure d(x,x') between an input vector x and the associated weight vectors of the Kohonen layer neurons. During training the winning neuron's weight vector, as well as those of its neighbouring neurons (grey circles) are updated to become more similar to the input vector.

An interesting property of this feature map in comparison to the well known *k*-means clustering is, that the number of features or clusters does not have to be predefined, e.g. Dawson & Wilby [2001]. In the context of hydrological applications SOM serve as a valuable tool for input variable selection [see Kocjancic & Zupan [2000], Bowden et al. 2005a, 2005b].

Further interesting architectures in the context of real-time or dynamic classification tasks could be adaptive resonance theory (ART) [Grossberg 1976] networks, which allow for the building of new data classes in an online manner. This means, that ART can adapt to changing data inputs rapidly by generating new classes whenever the currently available classes do not fit a new input sample. Popular generalisations of ART involve e.g. fuzzy ARTMAP a synthesis of elements from neural networks, expert systems, and fuzzy logic [e.g. Carpenter & Grossberg 1996]. An interesting application of fuzzy ARTMAP for the classification of data samples with missing values is given by e.g. Lim et al. [2005].

4 Data Preparation for the Gallego catchment (Ebro basin)

A common problem in basin and/or catchment modelling is, that many different types of data from various sources have to be incorporated into a single modelling environment. Letting alone the question of data quality from various sources a practical problem is, that data comes in different formats, which are unlikely to fit the format specifications for the intended modelling environment. In a former version of this report we did not expand on that point. However, looking at the growing time effort and facing an explicit request from our reviewers, we feel urged to mention a few details regarding data formats and quality:

The precipitation, temperature, and potential evapotranspiration data obtained for the Gallego catchment from the CHE ('Confederacion Hidrografica del Ebro', kindly provided by Hayley Fowler from HYDRO) are given in ASCII format. The first 15 character column (variably with or without □white spaces) holds a string with the name of the gauging station and the date (format: year year month month) followed by a value for each day of the indicated month. This means, that the record length of each line is date dependent - leap years included. A straightforward data processing is therefore not possible (strings containing white spaces, millenium bug, varying column sizes). However, with a clever choice of a processing/programming language this can be alleviated e.g. by built-in date and string handling functions.

Another problem are missing months or years, which occur sometimes, apparently at random throughout the data files. Now time series data should be concurrent and complete (e.g. Kantz & Schreiber [1997], Hirsch et al. [1991]) for its proper analysis - i.e., with respect to TREND 3 for the purposes of trend determination and/or catchment model development and calibration. Therefore, one has to come up with techniques to supplement missing data or select time intervals, in which enough data series (preferably covering the whole catchment) are available. For the example of precipitation data, the former would e.g. require the statistical analysis of the rainfall series at the gauging station (and/or close stations around it) and the subsequent synthetic generation of the missing time series data - while keeping consistency with the observed statistics. Specialised software for such purposes was developed e.g. by Bardossy [1998], Marani [2003, 2005] and Bardossy & Brommundt [2005].

However, it is not within the scope of TREND to (re-) develop such software, nor is it clear at the moment, whether such software will be suitable/available for use within AquaTerra (a first contact to Prof. Marani (University of Padova) has been made only recently). Therefore, the latter technique was followed and a converter was written in the MATLAB programming language, which - at present state - allows for a (linear) interpolation between the bordering data records, or the filling of the missing daily data records with a specified NO VALUE indicator. Furthermore, the data from the individual files for each gauging station are combined into a large file of concurrent time series columns, such that time intervals with large amounts of missing data can be visually detected. A graphical visualization of the potentially available precipitation gauges and the currently selected ones is given in Figure 4-1 (additionally, potential stream flow gauging stations are provided; source: CHE website: http://oph.chebro.es/):

Figure 4-1: Rainfall gauging stations (labels according to the CHE labelling) within the Gallego catchment. Red ovals mark the selected precipitation stations (1962-1991). Red circles mark generally available stream flow gauging stations (From north to south: Bubal, Anzanigo, Santa Eulalia, Ardisia, Zaragoza). Map produced with the help of David Kuntz.

Through the visual inspection of the processed time series the time span 1962 – 1991 was selected as being an interval with a reasonable number of time series and relatively representative spatial coverage. The selected data set comprises the following 17 stations (see also Figure 4-1):

P9446	P9446E	P9451	P9452	P9460
P9465	P9476	P9477	P9480	P9484
P9485	P9487	P9489	P9492	P9495
P9498	P9499			

Interspersed missing months - encountered somewhat frequently - are linearly interpolated. In order to find more recent data sets additional persons outside of AquaTerra were contacted:

Prof. Jesus Abaurea from the Statistics department of the University of Zaragoza has investigated the trends of rainfall extremes for the Ebro basin based on data series covering the time span 1951-2000 (Abaurrea et al. 2005). He kindly pointed us to the source of his data: Mr. Miguel Garcia Vera from the CHE in Zaragoza. By recommendation from the Ebro Agua Group at the Technical University of Darmstadt (by name Dr. Jens Hartmann, Dr. Oswald Marinoni) the contact to Mr. Miguel Garcia Vera was established. This is a valuable achievement, as the chances are high to obtain more recent meteoric input data, as well as hydrogeological data. Moreover, the need for widespread information is underlined by the response we obtained from Prof. Javier Martinez Gil, an expert regarding the hydrology of the Ebro river from the University of Zaragoza (contact provided by Prof. Jesus Carrera, University of Barcelona). He pointed us to problematic issues regarding the "natural" flow regime of the Gallego river:

"El régimen natural del caudales del río Gállego, como el de la totalidad de los grandes ríos pirenaicos, desde finales de los años 20 está altamente distorsionado por culpa de los embalses, y en especial en el Gállego, tanto por las detracciones para los canales hidroeléctricos como para los canales de riego. Los riegos en el bajo Gállego (aguas bajo de Zuera) tienen más de siete siglos de historia. La gran distorsión empezó con la puesta en servicio de la presa de La Peña, situada en la cuenca media, a finales de los años 20. Luego la puesta en servicio del embalse de La Sotonera, al que llegan las aguas detraidas del Gállego en Ardisa por un canal que tiene una capacidad de transporte de ¡90 m^3/s! es una nueva e importante fuente de distorsión."

In a condensed summary: the natural flow regime of the Gallego river is distorted due to various man-made lakes (water reservoirs) and channels built for irrigation purposes and to supply electric power plants. Supposedly the most important distortion arising at Ardisia, where an artificial channel of a capacity of 90 m^3/s extracts water from the river. Within the meetings with the Ebro Agua group in Darmstadt it was also disclosed that through large scale irrigation channels water might actually be transferred from the Gallego catchment to the Segre catchment in the east. Thus, - unless used as calibration parameters - for a process-based catchment model information is needed regarding these water abstructions. For a data-based modelling with ANN such water abstructions are implicitly taken care of by the measured downstream data series.

Finally, as a result of discussions with COMPUTE partners from the University of Trento regarding the promotion of data transfer within AquaTerra we set up a website providing the raw data, as well as the converted data for use with GeoSys for the Gallego catchment (http://www.uni-tuebingen.de/zag/geohydrology/AT/index.html). The data is managed according to the philosophy implemented in the GeoSys modelling system (see Kolditz & Bauer [2004], Wang & Kolditz [2005]): In short, an object-oriented separation of different types of information is performed. Geometric information, e.g. the location of gauging stations is contained in the GEO object. Information regarding neighbourhood relationships e.g. a network of (point) gauging stations and e.g. their corresponding area of support is stored in the MSH object (MSH, derived from ‘mesh’, contains topological information). Data series can be stored in the FCT object (FCT derived from ‘function’), where the FCT object typically is a time function. Finally, spatio-temporal data may be aggregated to a PCS object (PCS, derived from ‘process’, e.g. precipitation) by combining information from GEO, MSH, and/or FCT.

The modelling strategy to be followed in TREND 3 is two-fold. On one hand a spatially resolved process-based model will be established by use of GeoSys. For a flow-chart visualisation of the process-based modelling approach see Figure 4-2. On the other hand the functional relationship between the time series data from different gauging points can be extracted by ANN. For a possible application example of rainfall-runoff modelling within the Gallego catchment please refer to Figure 4-3.

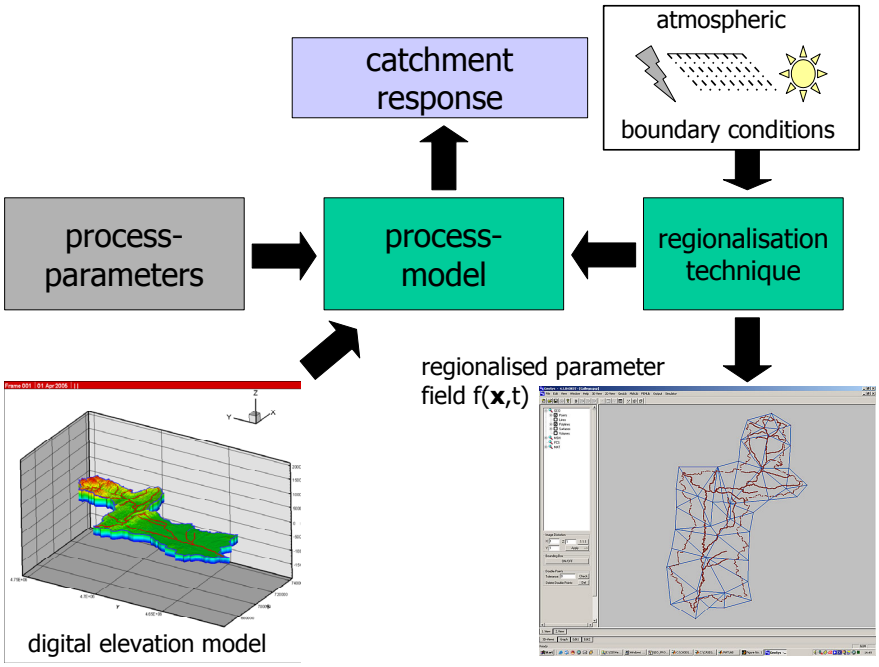

Figure 4-2: Structure of process-based catchment modelling for the Gallego catchment. The regionalisation function could be established by an ANN.

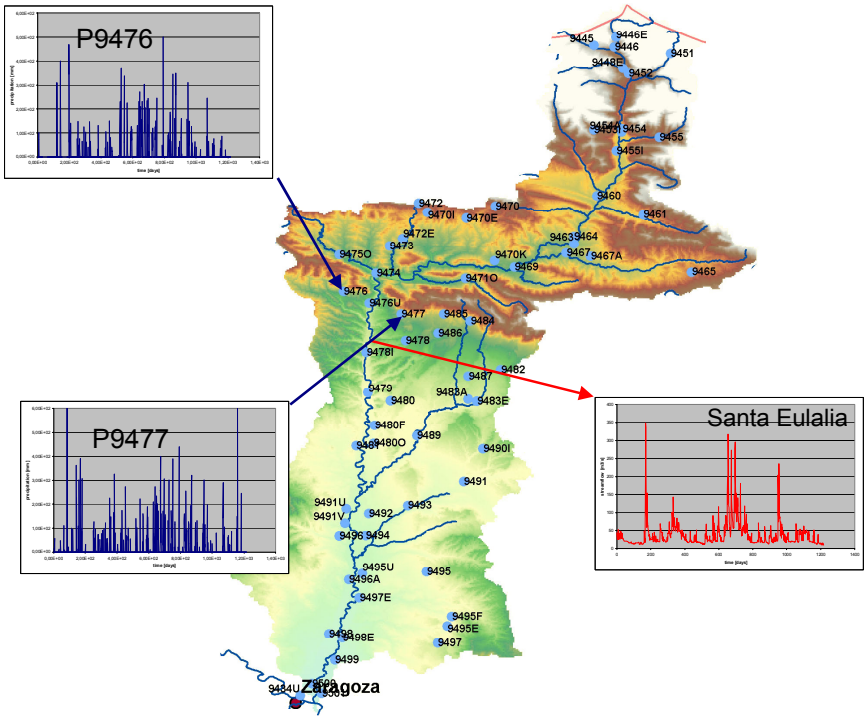

Figure 4-3: A preliminary example for ANN catchment modelling: rain gauge specific input time series (blue) are transformed to river flow predictions (red). Another important input variable is the current river flow stage (e.g. river flow some time before the targeted prediction time; not shown explicitely).

The ANN will be fed with consecutive rainfall measurements and data representing the current system state, e.g. the current stream flow. The output of the ANN model would then be a predicted stream flow value for some time step ahead. Another potential use of ANN could be the regionalisation of the atmospheric boundary conditions for the process-model (see Figure 4-2).

In either case of modelling strategy a catchment response due to a forcing (e.g. climate change) can be predicted. This response could be water yield, river chemistry etc. depending on the objective in mind. However, the appropriateness of the models as well as the strategy will, ultimately, depend on the abundance, type and quality of the catchment data.

5 References

Abaurrea, J., Asín, J., Cebrián, A.C., and A. Centelles (2005), An Analysis of Rainfall Extremes in the Ebro River Basin (1951-2000), using local Indices and an Areal Index, *Geophysical Research Abstracts*, 7, 09985, SRef-ID: 1607-7962/gra/EGU05-A-09985

Abrahart, R.J., and L. See (1998), Neural networks vs. ARMA modelling: constructing benchmark case studies of river flow prediction, In: *Proceedings of the 3rd International Conference on Geocomputation, University of Bristol,17-19 September*, internet-resource: www.geocomputation.org/1998/05/gc_05.htm (23.08.2005).

Allen, M. (1992), *Interactions between the atmosphere and oceans on time scales of weeks to years*, Ph.D. thesis, University of Oxford, Oxford, UK.

Allen, M., and L.A. Smith (1994), Investigating the origins and significance of low-frequency modes of climate variability, *Geophysical Research Letters*, 21, p. 883-886.

Allen, M.R., and L.A. Smith (1996), Monte Carlo SSA: Detecting irregular oscillations in the presence of coloured noise, *Journal of Climate*, 9, p. 3373-3404.

Allen, M.R., and L.A. Smith (1997), Optimal filtering in singular spectrum analysis, *Physics Letters A*, 234, p. 419-428.

ASCE Task Committee on Applications of Artificial Neural Networks in Hydrology (2000a), Artificial Neural Networks in Hydrology. I: Preliminary Concepts, *Journal of Hydrologic Engineering*, ASCE 5(2), p. 115-123.

ASCE Task Committee on Applications of Artificial Neural Networks in Hydrology (2000b), Artificial Neural Networks in Hydrology. II: Hydrologic Applications, *Journal of Hydrologic Engineering*, ASCE 5(2), p. 124-137.

Asefa, T., Kemblowski, M.W., Urroz, G., McKee, M., and A. Khalil (2004), Support vectors-based groundwater head observation networks design, *Water Resources Research*, 40, W11509, doi: 10.1029/2004WR003304.

Asefa, T. Kemblowski, M., Urroz, G., and M. McKee (2005a), Support vector machines (SVMs) for monitoring network design, *Ground Water*, 43(3), doi: 10.1111/j.1745-6584.2005.0050.x.

Asefa, T., Kemblowski, M., McKee, M., and A. Khalil (2005b), Multi-time scale stream flow predictions: The support vector machine approach, *Journal of Hydrology*, in press, doi: 10.1016/j.jhydrol.2005.06.001.

Bardossy, A. (1998), Generating precipitation time series using simulated annealing, *Water Resources Research*, 34(7), p. 1737-1744.

Bardossy and Brommundt (2005): Erzeugung simultan-synthetischer Niederschlagsreihen in hoher zeitlicher und räumlicher Auflösung für Baden-Württemberg.- Zwischenbericht anlässlich des Statusseminars des BWPLUS am 22.02. und 23.02. 2005 im Forschungszentrum Karlsruhe, 10 S., Internet-Quelle: http://bwplus.fzk.de/berichte/ZBer/2005/ZBerbwr24012.pdf

Blum, A., Kalai, A., and J. Langford (1999), Beating the Hold-Out: Bounds for K-fold and Progressive Cross-Validation, paper presented at the *12th Annual Conference on Computational Learning Theory*, ACM, Santa Cruz, California, USA.

Bollerslev, T. (1986), Generalized Autoregressive Conditional Heteroskedasticity, *Journal of Econometrics*, 31, p. 307-327.

Bonaccorso, B., Cancelliere, A., and G. Rossi (2005), Detecting trends of extreme rainfall series in Sicily, *Advances in Geosciences*, 2, p. 7-11.

Bowden, G.J., Maier, H.R., and G.C. Dandy (2005a), Input determination for neural network models in water resources applications. Part 1. background and methodology, *Journal of Hydrology*, 301, p. 75-92, doi: 10.1016/j.jhydrol.2004.06.021.

Bowden, G.J., Maier, H.R., and G.C. Dandy (2005b), Input determination for neural network models in water resources applications. Part 2. Case study: forecasting salinity in a river, *Journal of Hydrology*, 301, p.93-107, doi: 10.1016/j.jhydrol.2004.06.020.

Box, G.E.P., and F.M. Jenkins (1976), *Time Series Analysis: Forecasting and Control*, second edition., Holden-Day, Oakland, CA, USA.

Brath, A., Monatanari, A., and E. Toth (2002), Neural networks and non-parametric methods for improving real-time flood forecasting through conceptual hydrological models, *Hydrology & Earth System Sciences*, 6(4), p. 627-640.

Broomhead, D.S., and G.P. King (1986), Extracting qualitative dynamics from experimental data, *Physica D*, 20, p. 217-236.

Carlson, R. F., MacCormick, A.J.A., and D. G. Watts (1970), Application of linear models to four annual streamflow series, *Water Resources Research*, 6, p.1070–1078, 1970.

Carpenter, G.A., and S. Grossberg (1995), Adaptive Resonance Theory (ART), In: *The Handbook of Brain Theory and Neural Networks*, edited by M.A. Arbib, MIT Press, Cambridge, MA, USA.

Chiang, Y.-M., Chang, L.-C., and F.-J. Chang (2004), Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling, *Journal of Hydrology*, 290, p. 297-311.

Chon, K.H., and R.L. Cohen (1997), Linear and nonlinear ARMA model parameter estimation using an artificial neural network, *IEEE Transactions on Biomedical Engineering*, 44(3). p. 168-174.

Connor, J.T., Martin, R.D., and L.E. Atlas (1994), Recurrent neural networks and robust time series prediction, *IEEE Transactions on Neural Networks*, 5(2), p. 240-254.

Cristianini, N., and J. Shawe-Taylor (2000), *An Introduction to Support Vector Machines and Other Kernel Based Learning Methods*, 1st ed., Cambridge University Press, Cambridge, UK.

Dawson, C.W., and R.L. Wilby (2001), Hydrological modelling using artificial neural networks, *Progress in Physical Geography*, 25(1), p. 80-108.

Dibike, B.Y., Velickov, S., Solomatine, D., and B.M. Abbot (2001), Model induction with support vector machines: Introduction and applications, *Journal of Computing in Civil Engineering*, 15, p. 180-188.

Efron, B. (1983), Estimating the error rate of a prediction rule: improvements on cross-validation, *Journal of the American Statistical Association*, 78, p. 316-331.

Efron, B., and R.J. Tibshirani (1993), An introduction to the bootstrap, Chapman & Hall/CRC Press, Boca Raton, FL, USA, 456 p.

Efron, B., and R.J. Tibshirani (1997), Improvements on cross-validation: The .632+ bootstrap method, *Journal of the American Statistical Association*, 92, p. 548-560.

El-Shaarawi, A.H. (1993), Environmental monitoring, assessment and prediction of change, *Environmetrics*, 4, p. 381-398.

Elek, P., and L. Markus (2004), A long range dependent model with nonlinear innovations for simulating daily river flows, *Natural Hazards and Earth System Sciences*, 4, p. 277-283.

Elman, J.L. (1990), Finding structure in time, *Cognitive Science*, 14, p. 179-211.

Elsner, J.B., and A.A. Tsonis (1991), Do bidecadal oscillations exist in the global temperature record?, *Nature*, 353, p. 551-553.

Engle, R.F. (1982), Autoregressive Conditional Heteroscedasticity with Estimates of Variance of United Kingdom Inflation, *Econometrica*, 50, p.´987-1008.

Engle, R.F. (2001), GARCH 101: The Use of ARCH/GARCH Models in Applied Econometrics, *Journal of Economic Perspectives*, 15(4), p. 157-168.

Fausett, L. (1994), *Fundamentals of neural networks*, Prentice Hall, Englewood Cliffs, N.Y., USA.

Fortin, V., Ouarda, T.B.M.J., and B. Bobee (1997), Comment on “The use of artificial neural networks for the prediction of water quality parameters” by H.R. Maier and G.C. Dandy, *Water Resources Research*, 33(10), p. 2423-2424.

Gencay, R., and T. Liu (1997), Nonlinear modelling and prediction with feedforward and recurrent networks, *Physica D*, 108, p. 119-134.

Ghil et al. (2002), Advanced spectral methods for climatic time series, *Reviews of Geophysics*, 40(1), 1003, doi:10.1029/2000RG000092.

Ghil, M., and R. Vautard (1991), Interdecadal oscillations and the warming trend in global temperature time series, *Nature*, 350, p. 324-327.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Professional, Reading, MA, USA.

Gottman, J.M. (1981), *Time-series analysis A comprehensive introduction for social scientists*, Cambridge University Press, Cambridge, USA.

Grossberg, S (1976), Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors, *Biological Cybernetics*, 23, p. 121-134.

Haykin, S (1999), *Neural networks – A comprehensive foundation*, Prentice Hall International, Inc., New Jersey, USA, 824 p.

Hirsch, R.M., Slack, J.R., Smith, P.A. (1982), Techniques of trend analysis for monthly water quality data, *Water Resources Research***,** 18, p. 107-121.

Hirsch, R.M. and Slack, J.R. (1984), A nonparametric trend test for seasonal data with serial dependence, *Water Resources Research*, 20, p. 727-732.

Hirsch, R. M., R.B. Alexander and R.A. Smith (1991), Selection of Methods for the Detection and Estimation of Trends in Water Quality, *Water Resources Research*, 27(5), p. 803-813.

Hjorth, J.S.U. (1994), *Computer Intensive Statistical Methods Validation, Model Selection, and Bootstrap*, Chapman & Hall, London, UK.

Hochreiter, S., and J. Schmidhuber (1997), Long short-term memory, *Neural Computation*, 9, p. 1735-1780.

Hornik, K., Stinchcombe, M., and H.White (1989), Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, p. 359-366.

Hsieh, W.W. (2004), Nonlinear multivariate and time series analysis by neural network methods, *Reviews of Geophysics*, 42, RG1003, doi:10.1029/2002RG000112.

Jain, A., and S. Srinivasulu (2004), Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques, *Water Resources Research*, 40, W04302, doi: 10.1029/2003WR002355.

Jenkins, G.M., and D.G. Watts (1968), *Spectral analysis and its applications*, Holden-Day, San Francisco, USA.

Kantz, H., and T. Schreiber (1997), *Nonlinear time series analysis*, Cambridge University Press, Cambridge, UK.

Kallache, M., Rust, H.W., and J. Kropp (2005), Trend assessment: applications for hydrology and climate research, *Nonlinear Processes in Geophysics*, 12, p. 201-210.

Keppenne, C. L., and M. Ghil (1992), Adaptive filtering and prediction of the Southern Oscillation Index, *Journal of Geophysical Research*, 97, p. 20449–20454.

Khalil, A., McKee, M., Kemblowski, M., and T. Asefa (2005a), Basin scale water management and forecasting using artificial neural networks, *Journal of the American Water Resources Association*, 41(1), p. 195-208.

Khalil, A., Almasri, M.N., McKee, M., and J.J. Kaluarachchi (2005b), Applicability of statistical learning algorithms in groundwater quality modeling, *Water Resources Research*, 41, W05010, doi: 10.1029/2004WR003608.

Kocjancic, R., and J. Zupan (2000), Modelling of the river flowrate: the influence of the training set selection, *Chemometrics and Intelligent Laboratory Systems*, 54, p. 21-34.

Kohavi, R. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Mateo, Morgan Kaufmann, p. 1137-1145.

Kohonen, T. (1989), *Self organization and associative memory*, Springer Verlag, New York, USA.

Kohonen, T. (1990), The self organizing map, *Proceedings of the IEEE*, 78, p. 1464-1480.

Kolditz, O., and S. Bauer (2004), A process-oriented approach to computing multi-field problems in porous media, *Journal of Hydroinformatics*, 06.3,p. 225-244.

Kruel, T.-M. (1992), *Zeitreihenanalyse nichtlinearer Systeme – Chaos und Rauschen*, PhD thesis, Department of Chemistry, University of Würzburg, Germany.

Lendasse, A., Wertz, V., and M. Verleysen (2003), Model Selection with Cross-Validation and Bootstraps – Application to Time Series Prediction with RBFN Models, *Proceedings of the Joint International Conference on Artificial Neural Networks and on Neural Information Processing* (ICANN/ICONIP 2003), edited by O. Kaynak et al., Istanbul, Turkey, p. 573-580.

Libiseller C. and Grimvall A. (2002), Performance of Partial Mann Kendall Tests for Trend Detection in the Presence of Covariates, *Environmetrics*, 13, p. 71-84 .

Lim, C.P., Kuan, M.M., and R.F. Harrison (2005), Application of fuzzy ARTMAP and fuzzy *c*-means clustering to pattern classification with incomplete data, *Neural Computation and Applications*, 14, p. 104-113, doi: 10.1007/s00521-004-0445-9.

Lin, T., Horne, B.G., Tino, P., and C.L. Giles (1996), Learning long-term dependencies in NARX recurrent neural networks, *IEEE Transactions on Neural Networks*, 7(6), p. 1329-1338.

Liong, S.Y., and C. Sivapragasam (2000), Flood stage forecasting with SVM, *Journal of the American Water Resources Association*, 38, p. 173-186.

Magdon-Ismail, M. (2000), No Free Lunch for Noise Prediction, *Neural Computation*, 12(3), p. 547-564.

Maier, H.R., and G.C. Dandy (1999), Empirical comparison of various methods for training feed-forward neural networks for salinity forecasting, *Water Resources Research*, 35(8), p. 2591-2596.

Maier, H.R., and G.C. Dandy (2000), Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, *Environmental modelling & Software*, 15, p. 101-124.

Marani, M. (2003), On the correlation structure of point rainfall, *Water Resources Research*, 39(5), doi:10.1029/2002WR001456.

Marani, M. (2005), Non-power-law scale relations in rainfall, *Water Resources Research*, in press.

McCulloch, W.S., and W. Pitts (1943), A logic calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5, p. 115-133.

Minns, A.W., and M.J. Hall (1996), Artificial neural networks as rainfall-runoff models, *Hydrological Sciences Journal*, 41(3), p. 399-417.

Montanari, A., Rosso,R., and M. Taqqu (1997), Fractionally differenced ARIMA models applied to hydrologic time series: Identification, estimation and simulation, *Water Resources Research*, 33(5), p.1033-1044.

Montanari, A., Rosso,R., and M. Taqqu (2000), A seasonal fractional ARIMA model applied to the Nile River monthly flows at Aswan, *Water Resources Research*, 36(5), p. 1249-1259.

Ooms, M., and P.H. Franses (2001), A seasonal periodic long memory model for monthly river flows, *Environmental Modelling & Software*, 16, p. 559-569.

Pineda, F.J. (1987), Generalization of back-propagation to recurrent neural networks, *Physical Review Letters*, 59, p. 2229-2232.

Pineda, F.J. (1989), Recurrent back-propagation and the dynamical approach to adaptive neural computation, *Neural Computation*, 1, p. 161-172.

Pohlheim, H. (1999), *Evolutionäre Algorithmen: Verfahren, Operatoren und Hinweise für die Praxis* (Evolutionary algorithms: Methods, operators and notes for practical applications), Springer-Verlag, Berlin, Germany.

Preisendorfer, R.W. (1988), *Principal Component Analysis in Meteorology and Oceanography*, Elsevier Science, New York, USA.

Reed, P., Minsker, B.S., and D.E. Goldberg (2003), Simplifying multiobjective optimization: An automated design methodology for the nondominated sorted genetic algorithm-II, *Water Resources Research*, 39(7), 1196, doi: 10.1029/2002WR001483.

Rivas, V.M., Merelo, J.J., Castillo, P.A., Arenas, M.G., and J.G. Castellano (2004), Evolving RBF neural networks for time-series forecasting with EvRBF, *Information Science*, 165, p. 207-220.

Rumelhart, D.E., Hinton, G.E., and R.J. Williams (1986), Learning representations of back-propagation errors, *Nature*, 323, p. 533-536.

Saunders, C., Stitson, M.O., Weston, J., Bottou, L., Schölkopf, B., and A. Smola. (1998), Support vector machine reference manual, *Royal Holloway Technical Report CSD-TR-98-03*, Royal Holloway, UK.

Schoellhamer, D.H. (2001), Singular spectrum analysis for time series with missing data, *Geophysical Research Letters*, 28(16), p. 3187-3190.

Schölkopf, B., Smola, A., and K. Müller (1998), Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation*, 10, p. 1299–1319.

Schölkopf, B., Burges, J.C., and A. Smola (1999), *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA, USA.

Schölkopf, B., Smola, A.J., Williamson, R.C., and P.L. Bartlett (2000), New Support Vector Algorithms, *Neural Computation*, 12, p. 1207-1245.

Schölkopf, B., and A. Smola (2002), *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA, USA.

Schlittgen, R. and Streitberg, B.H.J. (1984), *Zeitreihenanalyse*, R. Oldenburg Verlag, Munich, Germany.

Shakhnarovich, G., El-Yaniv, R., and Y. Baram (2001), Smoothed Bootstrap and Statistical Data Cloning for Classifier Evaluation, paper presented at the *14th International Conference on Machine Learning*, Williams Collage, Berkshires, MA, USA.

Smola, A. (1998), *Learning with kernels*, PhD thesis, Department of Computer Science, Technical University of Berlin, Germany.

Strickert, M. (2003), *Time Series and Data Analysis*, Lecture script, Applied System Science, University of Osnabrück, Germany, Internet-resource: http://luna2.informatik.uni-osnabrueck.de/marc/lectures/zra_ss03/scriptparts/ (04.10.2005)

Tipping, M. (2000), The relevance vector machine, In: *Advances in Neural Information Processing Systems 12*, edited by S.A. Solla, T.K. Leen, and K.-R. Müller, MIT Press, Cambridge, MA, USA, p. 652-658.

Tipping, M. (2001), Sparse Bayesian learning and the relevance vector machine, *Journal of Machine Learning Research*, 1, p. 211-244.

Tong, H. (1990), *Non-linear Time Series: A Dynamical System Approach*, Clarendon Press, Oxford, UK.

Torrence, C., and G.P. Compo (1998), A Practical Guide to Wavelet Analysis, *Bulletin of the American Meteorological Society*, 79(1), p.61-78.

Vapnik, C.N. (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag, NY, USA.

Vapnik, C.N. (1998), *Statistical Learning Theory*, Wiley, NY, USA.

Vautard, R., and M. Ghil (1989), Singular spectrum analysis in non-linear dynamics with applications to paleoclimatic time series, *Physica D*, 35, p. 395-424.

Vautard, R., Yiou, P., and M. Ghil (1992), Singular spectrum analysis: A toolkit for short noisy chaotic signals, *Physica D*, 58, p. 95-126.

Wang, W., and O. Kolditz (2005), Object-oriented finite element analysis of thermo-hydro-mechanical (THM) problems in porous media, International Journal for Numerical Methods in Engineering, (submitted).

Wang, W., Van Gelder, P.H.A.J.M., Vrijling, J.K., and J. Ma (2005), Testing and modelling autoregressive conditional heteroskedasticity of streamflow processes, *Nonlinear Processes in Geophysics*, 12, p. 55-66.

Werbos, P.J. (1974), *Beyond regression: New tools for prediction and analysis in behavioural sciences*, PhD thesis, Harvard University, Cambridge, MA, USA.

Williams, R., and D. Zipser (1989), A learning algorithm for continually running fully recurrent neural networks, *Neural Computation*, 1, p. 270-280.

Yiou,P., Sornette, D., and M. Ghil (2000), Data-adaptive wavelets and multi-scale singular-spectrum analysis, *Physica D*, 142, p. 254–290.