CARPE DIEM

CENTRE FOR WATER RESOURCES RESEARCH

DELIVERABLE 10.1

OPTIMAL USE OF RADAR, NWP AND RAINGAUGE DATA IN PRECIPITATION FORECASTS FOR IMPROVING FLOOD FORECASTS IN URBAN AND RURAL CATCHMENTS

DRAFT

by

Micheal Bruen, Benoit Parmentier and Jianqing Yang

December 2004

Department of Civil Engineering

University College Dublin

Table of Contents

List of Tables

List of Figures

Chapter 1

methodology for optimal combination of raingauge, radar and NWP precipitation information for flood forecasting

1.1 Objectives

Theobjective was to develop a robust methodology for the incorporating the information from rain gauges and radar rainfall estimates with NWP forecasts in order to improve flood predictions. The methodology was tested in both rural and urban catchments. Partner 9 has collected a data set of (i) raingauge precipitation measurements (4 gauges) (ii) radar precipitation estimates (1km spatial resolution, 15 minute intervals) and (iii) HIRLAM forcasts (resolutions 2.5^o, 1.5^o, 0.5^o, with lead times from 6 to 48 hours, depending on resolution) and (iv) discharge measurements. Partner 9 investigated general methodologies for using the three different sources of precipitation information to provide flood forecasts for the Dargle. The major choice of methodology is whether

1. to combine the information in its 2-dimension precipitation form, or

2. to combine in the form of 1-dimensional discharge forecasts.

As Partner 2 has, in earlier work, focussed on the optimal combination of 2-dimensional precipitation fields for the Italian data, Partner 9 investigated combinations of the 1-dimensional discharge forecasts. This is a new approach and the authors have not seen any similar approach published in the literature to date.

1.2 methodology

The methodology can be summarised as follows:

1. Develop a precipitation forecast time-series from the measured precipitation data. At each time-step a forecast of future precipitation is made, based on the pattern of rainfall in the immediate past.

2. Develop a precipitation forecast times-series from the observed radar data. This is essentially what is usually understood as QPF and a large variety of methods are available

3. Produce, from the HIRLAM simulations, precipitation forecast series for all the available lead times (up to a maximum of 48 hours)

4. Each precipitation forecast series is then run through an hydrological model (SMAR in this case, described in Deliverable 9.3) to produce (step by step) a separate forecasts discharge time-series for each of the appropriate lag-times. Note that SMAR can have different parameters for each of the types of input series (gauge, radar, HIRLAM)

5. For each individual lead time, the appropriate discharge forecasts are inputs to an artificial neural network model which is trained (calibrated) to produce the optimal fit to the measured data, and the resulting output is taken as the overall optimal forecast series, Figure 1.1.

Different variations of this method are possible and were tested as part of this project, e.g. different numbers of hidden neurons in the ANN model, different lead times, different HIRLAM spatial resolutions. Although here, all calculations were done on the basis of hourly time series, the forecasts were limited (by the operational methodology of the HIRLAM model) to 3 or 6 hour time steps.

The advantages of this approach are

- the optimal combinations are done with 1 dimension time-series, not time-sequences of 2 dimensional images.
- the hydrological model can be tuned (calibrated) separately for each input source.

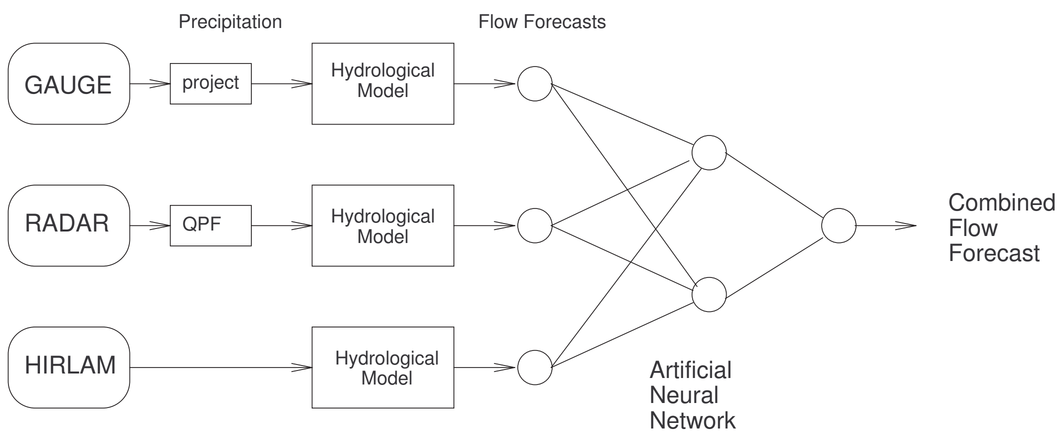

Figure 1.1: Optimal combination of precipitation information (in discharge space) for forecasting accumulations.

The optimal combination of inputs required a flexible, potentially nonlinear, black-box type modelling approach and an artificial neural network approach was chosen. After an initial review of the options, the neural network approach was selected. A review of neural network model construction, fitting and performance with special reference to flow forecasting was conducted, to establish overall guidelines for the modelling effort. This is described in the following chapter.

The results show that the ANN combination of the discharge forecasts is better (lower Mean Square Error) than any of the individual forecasts, Figure 2. The Mean Square Error is static up to 12 hours and increases (worsens) for longer lead times.

Figure 2 Forecast accuracy vs lead time for multi-source inputs (with 2.5o HIRLAM) and 3 hidden neurons

Chapter 2

Artificial Neural Networks

2.1 Introduction

Since the introduction of the concept of artificial neurons in 1943 (McCulloch and Pitts), Artificial Neural Networks (ANN) have been used to address a wide range of problems in business, industry, and science. However, the application of ANN in hydrology and water resources engineering is relatively new as most of the relevant research has been published within the last decade. Various aspects of hydrological modelling using ANN have been developed in recent years include rainfall forecasting (French et al, 1992; Luk et al, 2000), rainfall-runoff modelling (Hsu et al., 1995; Shamseldin, 1997; Campolo et al., 1999; Lange, 1999; Zealand et al., 1999; Imrie et al., 2000; Chang and Chen, 2001; Dibike and Solomatine, 2001; Sivakumar et al., 2002; Chiang et al., 2004; Xiong et al., 2004), reservoir inflow forecasting (Coulibaly et al, 2000), urban storm drainage and urban environment (Loke et al, 1997, Brezonik and Stadelmann, 2002; Ren et al., 2003); groundwater flow and water quality parameter prediction (Maier and Dandy, 1996; Morshed and Kaluarachchi, 1998; Gong et al., 1999). ANN models have also been applied to solve problems concerning missing data and data forecasting in hydrology (Tanaka, 1996).

In the water area, the most frequent applicant of ANNs has been rainfall-runoff modelling and forecasting. Maier and Dandy (2000) had an extensive review of 43 papers published in international journals, in which ANN has been used for the prediction or forecasting of water resources variables. More than half concerned flow and about one third of the paper addressed rainfall. A typical example is given by Imrie et al (2000) who developed a methodology for ANN modelling which ensured model generality in terms of overall performance and the ability to predict extreme values. The model was used to predict river flow and quality downstream of industrial effluent discharges.

Recently, the applications of ANN have been further extended in combination with fuzzy logic theory or the principles of chaos theory. Examples include Chang and Chen (2003) who applied the ANN to forecast water-stage in an estuary under high flood and tidal effects. The fuzzy min-max clustering technique is introduced for choosing best patterns for cluster representation in an efficient and automatic way in training the neural network. Tayfur et al. (2003) compared the performance of a fuzzy model with that of artificial neural networks in predicting the mean sediment loads from experimental runs using rainfall intensity and soil surface slope data as input. Using ANN model, Elshorbagy et al. (2002) estimated the missing consecutive daily stream flow data, where the analysis of chaos was made to configure the ANN model structure.

2.2 Why use ANNs

There are a number of reasons why ANNs are increasingly used in these applications. Its attractiveness lies in its ability to capture the non-linear relationships in data that are common in many complex real problems. Since all models seek to simplify the complexity of the real world by selectively concentrating on the fundamental aspects of the system at the expense of incidental detail (Singh, 1995), the simplifications involved in solving the governing equations in conventional models may limit the model performance. Moreover, the use of time series methods may be complicated by non-linearity and non-stationarity in the data (Imrie et al., 2000) which limits the successful application of many of the conventional models. ANN addresses such problems in a flexible and easily implemented manner.

Zhang et al (2001) presented an experimental evaluation of neural networks for non-linear time-series forecasting with three main effective factors, i.e., input nodes, hidden nodes and training sample size. They concluded that neural networks are valuable tools for modelling and forecasting non-linear time series and can improve on traditional linear methods.

Shamseldin (1997) compared the non-linear ANN model with some traditional models (both linear and non-linear) for rainfall-runoff modelling. The results obtained using the neural network in different external input scenarios suggest that although there are variable results, the neural network generally provided more accurate discharge forecasts than those traditional models, because ANN is nonlinear and more flexible.

Likewise, Sajikumar and Thandaveswara (1999) demonstrated ANN as a monthly non-linear rainfall-runoff model being the most efficient of the black-box models tested for the calibration period. Bodri and Cermak (2000) used ANN to represent the non-linear rainfall process in extreme precipitation forecasting and provided a good fit with actual data.

In fact, the application of ANN is not restricted to cases of non-linearity. Zhang (2001) examined the capability of neural networks for linear time series analysis and forecasting. Results showed that neural networks were well able to model such time series in a variety of situations. ANN can therefore be used in an even broader range of situations, particularly for those in which it is not clear in advance whether a linear or non-linear relationship is appropriate.

Another advantage of ANN is that it does not necessarily require a priori knowledge of the underlying process. As a data-driven model, the relationships of the input-output pattern are self-determined by the network itself during the process of "training". Knowledge of the detailed intrinsic structure of the natural physical process being modelled is not essential for using ANNs. This enables ANN to approximate any continuous function to any desired accuracy in theory if sufficient training is performed. The advantages of ANN against conventional models are also discussed in details by many authors, including, French et al. (1992).

However, no single best forecasting method will probably be appropriate for all situations (Bowerman and O'Connell, 1993). In some cases, the neural network approach may fail to generate satisfactory quantitative results. For example, ANN was found unable to properly predict extreme flows in Campolo et al. (1999). Since ANN is data dependent, if the range of data used for training is limited, then poor validation performance might be obtained when the validation data exceeds the range of the training set. Other limitations such as problems with over-fitting, difficulties in determining the number of hidden neurons, lack of systematic means of selecting the ANN internal structures, and difficulties with local optima, are discussed elsewhere (Morshed and Kaluarachchi, 1998; Zhang et al., 1998; Imrie et al., 2000). Furthermore, the relationship between network performance and the size of hidden layer is not well understood (El-Din and Smith, 2002).

Efforts have been made in ANN applications to address these problems. Falhman and Lebiere (1990) developed a Cascade Correlation (CC) learning procedure to avoid the conventional trial and error procedure in determining the hidden neurons. The CC procedure trains the network by embedding standard learning algorithms (e.g. Back-Propagation) into a number of hidden neuron units to find the most suitable network. Fernando and Jayawardeana (1998) used the radial basis function (RBF) type of ANN to forecast runoff. This has a symmetric basis function, which is represented by a centre highest value U and a spread s indicating the radial distance from the RBF centre. To train the network, they used the orthogonal least-square algorithms (Chen and Cowan, 1991) into which the number of hidden neurons was incorporated and added to the parameters to be trained. The training was carried out in two steps, one for the hidden layer and one for the output layer. An initial guess for the parameter s was required and the tolerance was an important parameter in balancing the accuracy of the network. Similarly, Lin and Chen (2004) used the Gaussian RBF neural network to model the rainfall runoff process, where the fully supervised learning algorithm was used to automatically construct the number of hidden layer neurons.

There are many different types of ANN models in practice. Multi-layer feed-forward neural networks are perhaps the favourite and perform well in most ANN applications. Maier and Dandy (2000) reported that more than 95% of the ANN-related papers they reviewed in the water resources area used feed-forward networks. In forecasting time series, the feed-forward network can be viewed as a general non-linear auto-regressive model. The linear auto-regressive (AR) models are special cases of ANN without hidden nodes (Zhang et al., 2001). Hill et al. (1994) surveyed the literature on regression-based forecasting, time series forecasting and decision making. They summarised that, overall, artificial neural networks are comparable to their statistical counterparts. Hsu et al. (1995) and Hwarng (2001) compared ANN with the time series models of Box and Jenkins (1976), and demonstrated that neural networks generally performed consistently well for time series corresponding to an ARMA(p, q) structure.

Despite the fact that the ANN has already been shown to have powerful pattern classification and recognition capabilities, and performed well in time series modelling and forecasting with respect to conventional models, the potential applications of ANN in the field of hydrology and water resources are not exhausted. Most of those applications are limited to the continuous time series simulation and forecasting. From a purely application's perspective, one remaining question is the suitability of ANN for predicting isolated storm events. Although there is little doubt of its usefulness in predicting the rainfall-runoff process, the capability of an ANN to model discontinuous time series data, such as individual storm event with short time steps (typical of urban catchments), has rarely been discussed.

2.3 The multi-layer feed-forward neural networks

Artificial neural networks are mathematical analogues of human cognition or neural biology (Friedman and Kandel, 1999). It is an information-processing system of massively parallel-distributed processors or nodes connected by weighted links, and operates similarly to biological neural networks. The procedure used to apply ANNs is called learning, through which the synaptic weights of the network are modified (Haykin, 1994).

There are several basic elements of neural network models (Friedman and Kandel, 1999, p. 255):

1. A set of simple mathematical processing elements are called neurons.

2. Signals are transmitted between neurons along connection links.

3. Each connection link is characterised by a weight, which multiplies the transmitted signal.

4. Each neuron applies an activation function on its net input (which is the sum of the weighted input signals) to obtain its output signal.

An artificial neural network is regarded as a mathematical technique or mathematical model. In systems terminology, a neural network may be considered as a data processing technique that maps, or relates some type of input information to output data (Azoff, 1994, p. 2). Artificial neural networks are usually classified as non-linear multi input-output black-box models.

2.4 Model structure

The structures and principles of the ANN are discussed in detail elsewhere (See, Haykin, 1994). Therefore, only a brief description on ANN model is presented in this chapter, with the emphasis mainly on modelling processes.

The neural network consists of a number of neurons. These neurons are also called cells, nodes or units. Each neuron is connected to other neurons by direct links with their associated weights. Each neuron receives different input signals while giving a single output. Each neuron can also send the same signal to different neurons at the same time. The input signals can be either from outside or within the network. The weights represent information related to the given problems to the neural networks. All the input signals to a neuron are multiplied by their corresponding weights and then added together. The result is then transformed through its activation function giving an output signal which goes to other neurons (Haykin, 1994, p. 8). The functionality of a neuron is shown in Figure 5.1.

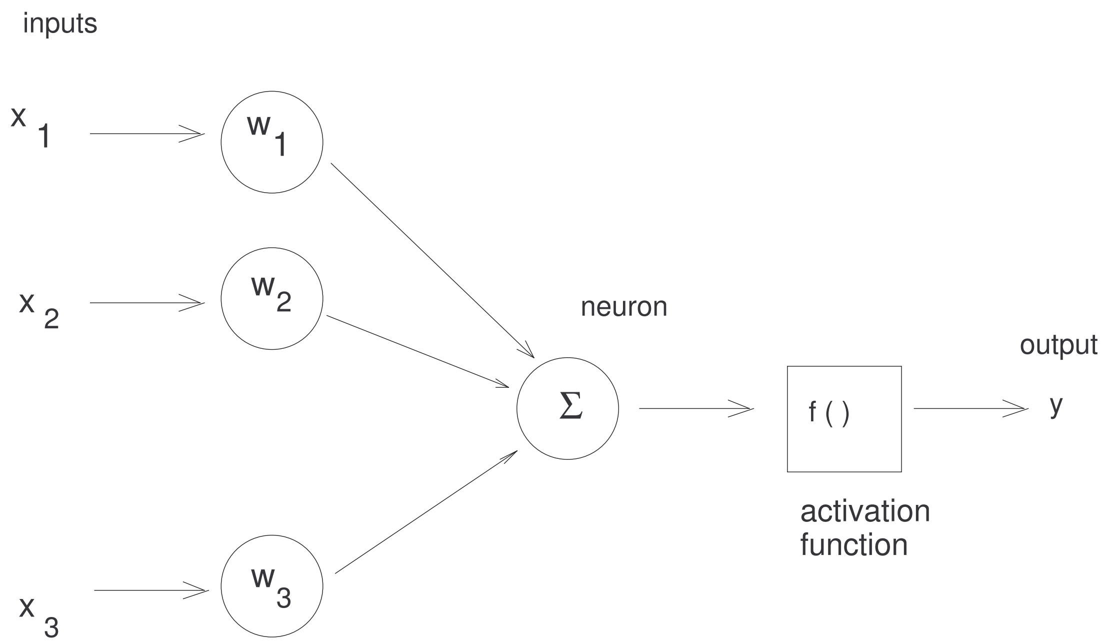

Figure 2.1: Functionality of a single neuron (after, Haykin, 1994)

Figure 5.1: Functionality of a single neuron (Haykin, 1994)

The neurons are organised in the form of layers producing different types of neural networks. The size of each layer is defined by the number of neurons used in it. The specific number of input, hidden, and the output neurons used depends on the particular problem (French et al., 1992). In general, neural networks can be classified into four different classes. These are, the single-layer feed-forward networks, multi-layer feed-forward networks, recurrent networks and lattice structures (Haykin, 1994, pp. 18-22).

Multi-layer feed-forward neural networks are the most widely used network structures. A multi-layered feed-forward network contains one input layer, one output layer and one or more hidden layers in between. In feed-forward networks, signals pass through the network only in the forward direction, i.e., start from input layer to hidden layer (or layers) and then to the output layer. Since all external information is passed directly through the input neuron without any transformation, the output from the input layer is thus identical to the external inputs. In some literature (e.g., Morshed and Kaluarachchi, 1998; Imrie et al., 2000), therefore, the input layer is not considered as a neuron layer, only the layers that perform the calculation are accounted for.

The hidden layer is possibly the most important part in neural network architecture. It is in fact the flexibility in choosing the numbers of hidden layers and hidden neurons that provides the flexibility of the ANN model. Likewise, the non-linearity of the model also comes from the hidden neurons with their non-linear activation functions, which play a very important role in detecting the features of the input-output pattern in the training data. There is no theoretical formula to specify the optimal number of hidden neurons. In practice, this is mostly determined by trail and error.

It has been shown that a single hidden layer network can approximate any complex function with arbitrary accuracy if sufficient hidden neurons are provided (Hornik et al., 1989). Furthermore, many researchers demonstrate that ANNs with one hidden layer are usually sufficient for most cases including forecasting problems (Gautam et al., 2000). However, in some particular problems, more than one hidden layer may be desirable in order to avoid the possible large number of hidden neurons required in a single hidden layer. In the present study, the most popular multi-layer feedforward neural network with one hidden layer is used. It is believed that a single hidden layer is sufficient to tackle the modelling problems in this study. Figure 5.2 illustrated an example of a three-layer feed-forward neural network, of the type used here.

2.5 Activation function

The neural transfer function is also called the activation function. Theoretically, any mathematical function that is differentiable may be used. Generally, the same

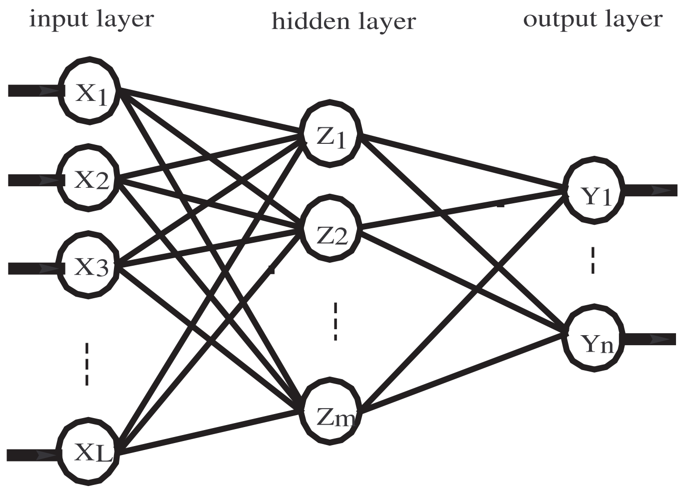

Figure 2.2: Three-layer feed-forward neural networks

transfer function is used for all neurons in a layer. Several forms of transfer functions popular in use are linear, logistic (sigmoid) and hyperbolic tangent. Of these, the logistic function is possibly the most commonly used. It is defined as (Haykin, 1994, p.138),

$$f(s) = \frac{1}{1 + exp(-s)} \tag{2.1}$$

where, s is the sum of the weighted (w_i) input signals (x_i) to a neuron defined by,

$$s = w_0 + \sum w_i x_i \tag{2.2}$$

The logistic function has an S shape and a limited output range [0, 1]. Therefore, the measured output data must be scaled to lie within this range. However, if the net internal activity level of a neuron is computed to have a magnitude close to the limits of the range, then, the output of the neuron will be very close to 0 or 1. In such a situation, the neuron is in "incorrect saturation" (Haykin, 1994, pp. 156-157). It is common practice to scale the training data to lie within a smaller range, say [0.1, 0.9], to avoid saturation. The formula used to scale data can be a simple mathematical function. Shamseldin (1997) showed a simple formula for the re-scaling range [0.1, 0.85].

Some authors suggest that all the input data should also be scaled to the range of the activation function. However, if the objective function used to train the network minimises the sum of square error of the differences of the measured and the final model outputs, then the input data does not have to be scaled. The scaling of output alone is sufficient. Moreover, data scaling can be avoided if the linear transfer function is used in the output layer. Nevertheless, scaling to uniform ranges is still recommended (Maier and Dandy, 2000) and it could provide a quicker training if all the trained data sets are within the same range.

2.6 Training (Optimisation)

The procedure used to calibrate the ANN model is known as "training" or "learning" and it seeks the optimal weights.

The number of parameters to be estimated (trained) depends on the type and structure of the ANN. In the case of the three-layer feed-forward neural network, the number of parameters to be trained is

$$p = (n_x + 1) * n_h + (n_h + 1) * n_y \tag{2.3}$$

(5.4)

where, p represent the total number of parameters (weights) required by the network, and n_x, n_h, n_y are the number of neurons in the input, hidden and output layers, respectively.

2.6.1 Training methods

The most commonly used training algorithms are (i) the back-propagation (BP) and (ii) the genetic algorithm (GA). BP is by far the most popular optimisation technique for training the feed-forward neural networks (c.f., Lachtermacher and Fuller, 1995). BP is based on the method of steepest descent in which an error function (objective function) is minimised by changing the weights in proportion to the objective function calculated. The error being the difference between the model output and the actual (measured) values. The connected weights are adjusted by moving a small step in the direction of the negative gradient of the error function at each iteration. A fixed step size is used as the learning rate to control the training process. Maier and Dandy (2000) found that the majority of published water resources applications used the standard back-propagation (BP) algorithm to optimise the connection weights.

Despite its successful applications, as a local optimisation technique, BP also suffers from the limitation of its inability to escape from local optima and slow convergence (Lin and Chen, 2004). Attempts have been made to improve on the standard BP (c.f., Shoemaker, et al., 1991; Ooryen and Nienhuis, 1992).

The Genetic Algorithm is a global search procedure that addresses the problem of local oscillation of the BP. It uses a probabilistic transition from one estimate to the next trial rather than deterministic rules, and searches among a population of points instead of a single point. Sexton and Dorsey (2000) compared BP with GA for ANN training and found that GA regularly outperforms the commonly used BP algorithm as an alternative ANN training technique. However, Morshed and Kaluarachchi (1998) demonstrate that GA performs less robustly than BP for solving groundwater problems, alternatively GA may be developed as a complementary search procedure to BP.

In addition to the BP and GA methods, another alternative for ANN training is the conjugate gradient (CG) algorithm (Press et al., 1986). The CG method is a widely used numerical technique for solving various optimization problems (Ham and Kostanic, 2001). The method of steepest decent searches in each iteration in the direction of the steepest decent of the function at the current location. The conjugate gradient method is a more complicated second order scheme which provides an improvement on the steepest descent search direction that is constructed to be conjugate to the old gradient and, to all previous directions traversed as far as possible. That is, the CG algorithm searches a system of conjugate directions on the error surface and updates the weights along all these directions. The conjugate gradient method has been shown to provide fast and efficient training for neural networks (Moller, 1993;

Shamseldin, 1997). Chiang et al. (2004) provide a systematic comparison of two basic types of neural network, static and dynamic-feedback network to demonstrate the efficiency and practicability of the neural networks for one-hour ahead stream flow forecasting in Taiwan. Two back-propagation (BP) learning optimization algorithms, the standard BP and conjugate gradient (CG) method, are used for the static network. In a comparison of searching algorithms for a static network, the results show that the CG method is superior to the standard BP method in terms of the efficiency and effectiveness of the constructed network performance.

Over-fitting is more likely to occur in neural network models than in other statistical models due to the typical large parameter set to be estimated (Zhang et al., 2001). Since ANN learns underlying processes from the data, the input information can therefore have a significant impact on the training process, which eventually affect the model performance. Furthermore, though there is no need for a priori knowledge of the input-out pattern, in order to capture the generality of their relationship, it is important for training data sets to represent the physically based dynamical range of the forecasts.

It is also worth mentioning that training of neural networks is sensitive to the initial weights. Improper initial weights may lead to either saturation or local optima. It is therefore important to select the initial weights for the success of network training. The saturation phenomenon could be avoided by choosing the suitable initial values of the weights in training. A common practice is to use the random values of the range [-0.5, 0.5] as the initial weights (Menahem and Abraham, 1999).

Chapter 3

Model Construction

Following this review, a simple feed-forward neural network model with sigmoid activation function and a single layer of hidden neurons was chosen. Calibration was with a second order, conjugate gradient optimisation algorithm, and based on minimising a mean square residual criterion.

A FORTRAN code was developed to

1. input the gauge, radar and HIRLAM precipitation inputs

2. to implement the a simple nowcasting procedure with the radar information

3. apply the SMAR model to all of the appropriate precipitation input series, with different parameter sets for the gauge and radar inputs. The same parameters as used with the gauge input was used with the model processing the HIRLAM precipitation.

4. estimate the weights of the neural network optimal combination model.

5. apply this model to the measured data to generate the final forecasted discharges for model assessment.

The HIRLAM output comes in files with a special compressed format(GRIB). before implementing this model routine were written to extract the required precipitation information from the GRIB files. This was more easily done on a UNIX system because of the available libraries.

This approach was tested and assessed by applying it to a rural and an urban catchment as described in Deliverable 10.3.

Chapter 4

References

Azoff, E.M., 1994. Neural network time series forecasting of financial markets. Published by John Wiley & Sons Ltd., England.

Bodri, L., Cermak, V., 2000. Prediction of extreme precipitation using a neural network: application to summer flood occurrence in Moravia. Advances in Engineering Software 31: 311-321.

Bowerman, B.L., O'Connell, R.T., 1993. Forecasting and Time Series: An Applied Approach (3rd edition). Duxbury Press, Belmont, CA.

Box, G.E.P., Jenkins, G.M., 1970. Time series analysis: Forecasting and control. San Francisco: Holden-Day Inc.

Box, G.E.P., Jenkins, G.M., 1976. Time series analysis - forecasting and control (Revised edition). Holden-Day Inc. San Fransisco.

Brezonik, P.L., Stadelmann, T.H., 2002. Analysis and predictive models of stormwater runoff volumes, loads, and pollutant concentrations from watersheds in the Twin Cities metropolitan area, Minnesota, USA. Water Research 36: 1743-1757.

Campolo, M., Andreussi, P., Soldati, A., 1999. River flood forecasting with a neural network model. Water Resources Research 35(4): 1191-1197.

Chang F.J., Chen, Y.C., 2001. A counterpropagation fuzzy-neural network modeling approach to real time streamflow prediction. Journal of Hydrology 245(1-4): 153-164.

Chang, F.J., Chen, Y.C., 2003. Estuary water-stage forecasting by using radial basis function neural network. Journal of Hydrology 270 (1-2): 158-166.

Chen, S., Cowan, C.F.N., 1991. Grant PM. Orthogonal least square methods and their application to non-linear system identification. IEEE Trans. Neural Networks 2(2): 302-309.

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

Coulibaly, P., Anctil, F., Bobee, B., 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. Journal of Hydrology 230: 244-257.

Elshorbagy, A., Simonovic, S.P., Panu, U.S., 2002. Estimation of missing streamflow data using principles of chaos theory. Journal of Hydrology 255 (1-4): 123-133.

Falhman, S.E., Lebiere, F., 1990. The cascade-Correlation learning architecture. Advances in Neural Information Processing 2.

Fernando, D.A.K., Jayawardeana, A.W., 1998. Runoff forecasting using RBF Networks with OLS Algorithm. Journal of Hydrologic Engineering 3(3): 203-209.

French, M.N., Krajewski, W.F., Cuykendall R.R., 1992. Rainfall forecasting in space and time using a neural network. Journal of Hydrology 137: 1-31.

Friedman, M., Kandel, A., 1999. Introduction to pattern recognition- statistical, structural, neural and fuzzy logic approaches. Published by Imperial College Press, London, pp. 255-310.

Gautam, M.R., Watanabe, K., Saegusa, H., 2000. Runoff analysis in humid forest catchment with artificial neural network. Journal of Hydrology 235: 117-136.

Gong, N., Denoeux, T., Bertrand-Krajewski, J., 1996. Neural networks for solid transport modeling in sewer systems during storm events. Wat. Sci. Tech. 33(9): 85-92.

Ham, F.M., Kostanic, I., 2001. Principles of neurocomputing for science and engineering. McGraw-Hill, New York.

Haykin, S., 1994. Neural networks- a comprehensive foundation. Macmillan College Publishing Company, Inc. New York.

Hiang, Y.M., Chang, L.C., Chang, F.J., 2004. Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modelling. Journal of Hydrology 290: 297-311.

Hill, T., Marquez, L., O'Connor, M., Remus, W., 1994. Artificial neural network models for forecasting and decision making. International Journal of Forecasting 10 (1): 5-15.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2 (5): 359-366.

Hsu, K.L., Gupta, H.V., Sorooshian, S., 1995. Artificial neural network modelling of the rainfall-runoff process. Water Resources Research 31 (10): 2517-2530.

Hsu, M.H., Chen, S.H., Chang, T.J., 2000. Inundation simulation for urban drainage basin with storm sewer system. Journal of Hydrology 234 (1-2): 21-37.

Hwarng, H.B., 2001. Insights into neural network forecasting of time series corresponding to ARMA(p, q) structures. Omega (the international journal of Management Science) 29, 273-289.

Hwarng, H.B., Ang, H.T., 2001. A simple neural network for ARMA(p,q) time series. Omega 29: 319-333.

Imrie, C.E. Durucan, S., Korre, A., 2000. River flow prediction using artificial neural networks: generalization beyond the calibration range. Journal of Hydrology 233: 138-153.

Lachtermacher, G. Fuller, J.D., 1995. Backpropagation in time-series forecasting. Journal of Forecasting 14: 381-393.

Lange, N.T., 1999. New mathematical approaches in hydrological modeling- an application of artificial neural networks. Phys. Chem. Earth (B) 24 (1-2): 31-35.

Lin, G.F, Chen, L.H., 2004. A non-linear rainfall-runoff model using radial basis function network. Journal of Hydrology 289 (1-4): 1-8.

Lin, G.F., Chen, L.H., 2004. A non-linear rainfall-runoff model using radial basis functional network. Journal of Hydrology 289: 1-8.

Loke, E., Warnaars, E.A., Jacobsen, P., Nelen, F., Almeida, M., 1997. Artificial neural networks as a tool in urban storm drainage. Wat. Sci. Tech. 36(8-9): 101-109.

Luk, K.C., Ball, J.E., Sharma, A., 2000. A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. J. Hydrology 227: 56-65.

Maier, H.R., Dandy, G.C., 1996. The use of artificial neural networks for the prediction of water quality parameter. Water Resources Research 32 (4): 1013-1022.

Maier, H.R., Dandy, G.C., 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. Environmental Modeling & Software 15 (1): 101-124.

McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas imminent in nervous activity. Bulletin and Mathematical Biophysics 5: 115-133.

Menahem, F., Abraham, k., 1999. Introduction to pattern recognition- statistical, structural, neural and fuzzy logic approaches. Imperial College Press, London, pp. 255-310.

Moller, M., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6: 525-533.

Morshed J., Kaluarachchi, J.J., 1998. Application of artificial neural network and genetic algorithm in flow and transport simulations. Advances in Water Resources 22(2): 145-158.

Ooryen, A., Nienhuis, B., 1992. Improving the convergence of the back propagation algorithm. Neural Networks 5: 465-471.

Ren, W.W., Zhong, Y., Meligrana, J., Anderson, B., Watt, W.E., Chen, J.K., Leung, Urbanisation, land use, and water quality in Shanghai 1947-1996. Environment International 29: 649- 659.

Sajikumar, N., Thandaveswara, B.S., 1999. A non-linear rainfall-runoff model using an artificial neural network. J. Hydrology 216: 32-55.

Sexton, R.S., Dorsey, R.E., 2000. Reliable classification using neural networks: a genetic algorithm and backpropagation comparison. Decision support systems 30 (1): 11-22.

Shamseldin, A., 1997. Application of a neural network technique to rainfall-runoff modeling. Journal of Hydrology 199: 272-294.

Shamseldin, A.Y. and O'Connor, K.M., 2001. A non-linear neural network technique for updating of river flow forecasts. Hydrology and Earth System Sciences 5 (4): 577-597.

Shaw, E.M., 1994. Hydrology in Practice (Third edition). Chapman & Hall, 2-6 Boundary Row, London, SE1 8HN.

Singh, V.P., 1995. Watershed modeling. In: Computer models of watershed hydrology. Edited by Singh, V.P. Water resources publications, USA.

Sivakumar, B., Jayawardena, A.W., Fernando, T.M.K.G., 2002. River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches. Journal of Hydrology 265 (1-4): 225-245.

Tanaka, M., 1996. Identification of nonlinear systems with missing data using stochastic neural network, decision and control. Proceedings of the 35th IEEE Conference-Journal 1: 933-934.

Tayfur, G., Ozdemir, S., Singh, V.P., 2003. Fuzzy logic algorithm for runoff-induced sediment transport from bare soil surfaces. Advances in Water Resources 26 (12): 1249-1256.

Zaghloul, N.A., Kiefa, M.A. Abu, 2001. Neural network solution of inverse parameters used in the sensitivity-calibration aanalyses of the SWMM model simulations. Advances in Engineering Software 32 (7): 587-595.

Zealand, C.M., Burn, D.H., Simonovic. S.P., 1999. Short term streamflow forecasting using artificial neural networks. Journal of Hydrology 214 (1-4): 32-48.

Zeiler, M., 1999. Modelling our world- the ESRI guide to geodatabase design. Environmental Systems Research Institute, Inc (ESRI). California.

Zhang, G., Patuwo, B.E., Hu, M.Y., 1998. Forecasting with artificial neural networks: the state of the art. International Journal of Forecasting 14: 35-62.

Zhang, G.P, 2001. An investigation of neural networks for linear time-series forecasting. Computers & Operations Research 28: 1183-1202.

Zhang, G.P, Patuwo, B.E., Hu, M.Y., 2001. A simulation study of artificial neural networks for nonlinear time-series forecasting. Computers & Operations Research 28: 381-396.

Zhang, W., Cundy, T.W. 1989. Modeling of Two-Dimensional Overland Flow. Water Resources Research 25 (9): 2019-2035.