

Rainfall data calculation using Artificial Neural Networks and adaptive neuro-fuzzy inference systems

L. Mpallas, C. Tzimopoulos & C. Evangelidis

*Department of Rural and Surveying Engineering,
Aristotle University of Thessaloniki, Greece*

Abstract

Missing rainfall data is a frequent problem in every hydrological or water management study. Conventional methods are usually based on statistics and the missing rainfall data are estimated using neighbouring stations. The same procedure is implemented here using two models based on artificial intelligence techniques. A fuzzy logic and an Artificial Neural Network model are used to predict rainfall heights of two meteorological stations. A number of model parameters were modified during the calibration, in order to achieve the best performance. The comparison between observed and estimated values showed that both models perform satisfactorily in the prediction of rainfall data.

Keywords: fuzzy logic, rainfall, neural networks, hydrology.

1 Artificial intelligence

Artificial intelligence includes a number of methods that mimic human reasoning. These methods can be modelled and applied by a computer. The most common ones are Artificial Neural Networks (ANN), Genetic Algorithms (GA) and fuzzy logic theory.

Artificial intelligence is an alternative and also effective method in studying and modelling hydrological systems. They capture their ambiguous and inaccurate nature, even when non-linearities exist. These models are extremely effective when the knowledge of the system is insufficient or when the underlying procedures are not fully understood.



Over the last decades, models based on ANN are applied into hydrology and water management problems. They have been used in river flow prediction [1–3], in reservoir management [4], in lake fluctuation [5] etc.

Fuzzy logic model are also widely applied in groundwater movement [6], in hydrologic cycle description [7], in reservoir management [8–11], in prediction [12–15] etc.

2 Fuzzy inference systems

In Fuzzy Logic, elements are possible to belong partially to a set. Fuzzy sets are a generalization of the classical sets since they don't have clearly defined boundaries. If X is a collection of objects denoted generally by x , then a fuzzy set \tilde{A} in X is a set of ordered pairs:

$$\tilde{A} = \left\{ \left(x, \mu_{\tilde{A}}(x) \right) \mid x \in X \right\} \quad (1)$$

where $\mu_{\tilde{A}}(x)$ is called the membership function or grade of membership of x in \tilde{A} . The degree to which an element belongs to a fuzzy set is shown by the membership function, which takes values from the $[0, 1]$ interval.

A fuzzy rule is composed of explanatory variables (fuzzy premises) connected together with operators and an action entailed, which gives the output (fuzzy response). These operators associate the input variables and their combination defines the result. The difference between the fuzzy rules and the rules corresponding to common logic is that the first allow partial and simultaneous applicability. Partial applicability means that the membership functions of the input values belong to the $[0, 1]$ interval.

A collection of fuzzy IF-THEN rules constitutes a fuzzy rule-based system or a fuzzy inference system. First the input variables, which are usually crisp, are being transformed into fuzzy ones, by applying a membership function. This procedure denotes the degree to which an input value belongs to a fuzzy set. The most common membership function is the triangular one.

The fuzzy premises are connected using operators (AND, OR and rarely XOR). AND is the operator which is most commonly used, corresponding to the intersection of the classical sets. There are many ways to interpret the AND operator, defined by a great variety of t-norms, as they have been proposed by many researchers [16]. The most common ones are the minimum method and the algebraic product. The implementation of a t-norm results the Degree of Fulfilment (DOF) of a rule, which also takes values from the $[0, 1]$ interval.

The DOF shows the applicability of each rule for every vector of premises. Since the fuzzy premises are being fulfilled to some degree, the fuzzy consequences should be fulfilled to the same degree. The fuzzy responses of each rule are aggregated and finally a combined fuzzy response is given.

In all cases, for practical reasons, we need to replace the resulting fuzzy set with a crisp number, usually in order to take a control action. This procedure is called defuzzification.

2.1 ANFIS (Adaptive Neuro Fuzzy Inference System)

ANFIS is a technique for constructing fuzzy inference systems and was proposed by Jang [17]. It works like neural networks and it is able to learn from a data set the membership function parameters in order to capture the system behaviour. These parameters are being adjusted during the training process in order to obtain the IF-THEN rules. The data set consists of input and output values. A gradient vector facilitates the computation of these parameters. Once the gradient vector is obtained, any of the several available optimization routines could be applied in order to adjust the parameters and reduce the model error.

Suppose a rule system consisting of two fuzzy premises x and y and a fuzzy response. The rule system is constituted by two fuzzy “IF-THEN” rules:

Rule 1: If x is A_1 and y is B_1 Then $f_1 = p_1x + q_1y + r_1$

Rule 2: If x is A_2 and y is B_2 Then $f_2 = p_2x + q_2y + r_2$

Layer 1: Every node has a membership function:

$$O_{i,1} = \mu_{A_i}(x) \quad (2)$$

where x is a crisp input, A_i a linguistic variable and μ_{A_i} the membership function. If a bell-shaped function is applied then:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i} \right)^2 \right] b_i} \quad (3)$$

where $\{a_i, b_i, c_i\}$ are function parameters that change the function shape.

Layer 2: In the rule nodes the AND/OR operator is applied to get one output representing the results of the antecedent for a fuzzy rule, which is, firing strength. It denotes the degrees by which the antecedent part of the rule is satisfied and it indicates the shape of the output function for that rule. So the outputs of the second layer, called by firing strengths $O_{2,i}$, are the products of the corresponding degrees obtaining from layer 1, named as w in below:

$$O_{i,2} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2 \quad (4)$$

Layer 3: The main target is to compute the ratio of each rule's firing strength to the sum of all rules' firing strength. The firing strength in average nodes (layer 3) is normalized.

$$O_{i,3} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (5)$$

Layer 4: In the consequent nodes (layer 4), the contribution of each i th rule toward the total output or the model output and/or the function defined as below is calculated.

$$O_{i,4} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \quad (6)$$

where $\{p_i, q_i, r_i\}$ the parameter set in the consequent function.

Layer 5: Includes the defuzzification procedure.

3 Artificial Neural Networks

A neural network is an interconnected assembly of simple processing elements, *units* or *nodes*, whose functionality is based on the actual neurons. The processing ability of the network is stored in the connection strengths, or weights, obtained by a process of adaptation from a set of training patterns.

Human neurons communicate via electrical signals that are short-lived impulses in the voltage of the cell wall or membrane. The interneuron connections are mediated by electrochemical junctions called *synapses*, which are located on branches of the cell referred to as *dendrites*. Each neuron typically receives many thousands of connections from other neurons and is therefore constantly receiving a multitude of incoming signals, which eventually reach the cell body. They are summed together and if the resulting signal exceeds some threshold, then the neuron will "fire" or generate a voltage impulse in response. This is transmitted to other neurons via a branching fibre known as the axon.

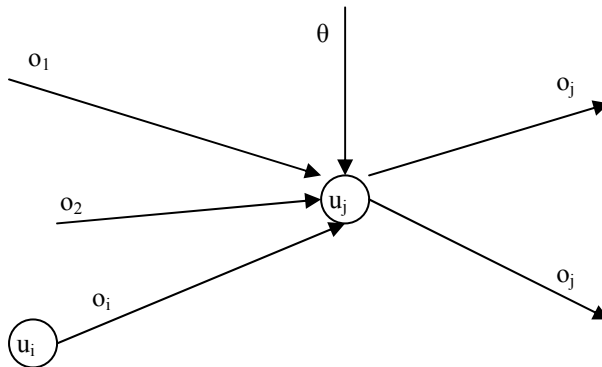


Figure 1: Artificial Neural Network (general form).

Synapses are modelled by a single number or weight so that each input is multiplied by a weight, before being sent to the equivalent of the cell body. Every unit receives signals from more than one units and the signal is sent to another one. The weighted signals are summed together by simple arithmetic addition to supply node activation. The activation is compared with a threshold. If the activation exceeds the threshold, the unit produces an output (conventionally "1"), otherwise it outputs zero. The activation or not of the node depends on a function f_j , usually a sigmoid one. The sigmoid function is given by the following equation:

$$f_j = \frac{1}{1 + e^{(-v_j)}} \quad (7)$$

where v a parameter that defines the exact shape.

The whole system works in parallel. Each neurons output is equal to the summed weighted input signals with a constant added. Each neurons output is an input to the next neurons until the output layer [18].

One type of network is shown in Fig. 2 where the nodes are arranged in a layered structure. Usually a neuron is connected only to the neurons of the next layer. When there are multiple layers then the intermediate ones are called hidden. This feed forward structure is only one of several available and is typically used to place an input pattern into one of several classes according to the resulting pattern of outputs [19].

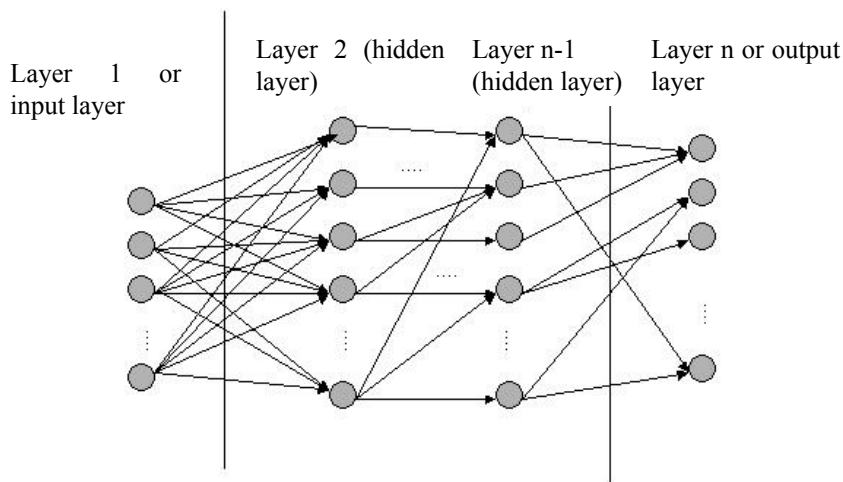


Figure 2: Typical Artificial Neural Network.

3.1 Training

In real neurons the synaptic strengths may, under certain circumstances, be modified so that the behaviour of each neuron can change or adapt to its particular stimulus input. In artificial neurons the equivalent of this is the modification of the weight values. In terms of processing information, there are no computer programs running here. The network knowledge is stored in its weights, which evolve by a process of adaptation to a stimulus from a set of pattern examples. In one training paradigm called supervised learning, an input pattern is presented to the net and its response then compared with a target output. In terms of our previous letter recognition example, an "A", say, may be input and the network output compared with the classification code for A. The difference between the two patterns of output then determines how the weights are altered. Each particular recipe for change constitutes a learning rule, details of which form a substantial part of subsequent chapters. When the required weight updates have been made another pattern is presented, the output compared with the target, and new changes made. This sequence of events is repeated iteratively many times until the network's behaviour converges so that

its response to each pattern is close to the corresponding target. The process as a whole, including any ordering of pattern presentation, criteria for terminating the process, etc., constitutes the training algorithm.

3.2 Levenberg-Marquardt algorithm

Levenberg-Marquardt algorithm is the combination between Gauss - Newton method (for optimization problems) and the Gradient Descent method. It was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix is:

$$H = J^T J \quad (8)$$

and the gradient can be computed as

$$g = J^T e \quad (9)$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard back propagation technique that is much less complex than computing the Hessian matrix. The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (10)$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near a minimum error, so the aim is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function [20].

4 Application

Rainfall itself is an extremely time and location varying phenomenon. These data are usually obtained from meteorological stations, satellites or radar. The most common problem of meteorological stations is the fact that the data is usually incomplete. This is due to station malfunction, observer's negligence, etc. In these cases the estimation of missing values is implemented based on neighbouring stations. The aim of this paper is to predict the missing rainfall data of two stations, using fuzzy logic and Artificial Neural Network theory. The fuzzy inference model is trained using ANFIS method and the ANN is trained using Levenberg-Marquardt algorithm.

4.1 Meteorological stations

In this project there were four stations under examination, all located in Strymonas hydrological basin, in northern Greece. Serres and Achladochori



meteorological stations are used as inputs for both fuzzy logic and ANN model. N. Zichni and Nigrita meteorological stations are used as outputs. Two runs were finally implemented for both models. In the first case the available rainfall data heights of Serres and Achladochori station were used as inputs and the corresponding data of N. Zichni were used as outputs. In the second case input data were kept the same, while Nigrita rainfall data were used as outputs. The difference between these two cases is the fact that the N. Zichni station data are more linearly correlated to the data of the input stations, than Nigrita station. The aim was to examine the influence of correlation between input and output data to model performance. The meteorological stations are:

Table 1: Meteorological stations.

Station	Altitude (m)	Available data
Serres	34	1966-today
Achladochori	485	1980-2003
N. Zichni	280	1980-2003
Nigrita	111	1980-2003

For all cases the mean monthly values were used. Part of them was used for training and the rest for validation.

4.2 ANFIS

Fuzzy inference models are based on “If-Then” rules, where “If” refers to the fuzzy premises vector. Fuzzy premises are Serres and Achladochori rainfall heights. The fuzzy response vector consists of N. Zichni and Nigrita rainfall heights and refers to the “Then” part of the rule. The rule system was trained using ANFIS, where the parameters of membership function was adapted using an optimization method. In this project a hybrid optimization system was used, which is included in MATLAB.

The training set consists of mean monthly rainfall heights from years 1980 to 1995 (192 values). Validation set was constructed using the mean monthly rainfall heights from years 1996 to 2003 (96 values).

Fuzzy logic model performance was tested when two parameters were changed. These were the type and the number of membership function utilized in dividing the domain of fuzzy premises. It must be noted that the influence of the overlap between membership functions on the results, was also investigated. We concluded that the overlap does not influence the final result significantly and finally 50% overlap between fuzzy premises was selected.

4.3 Membership function number

The number of the membership functions utilized in order to divide the fuzzy premises domain, is a parameter that affects fuzzy logic model performance significantly. Usually a trial and error method is implemented, where the number of the membership functions with the smallest model error is sought. The model

was tested using from 2 to 5 membership functions for the division of the domain of Serres and Achladochori meteorological stations. It must be noted that high number of membership functions results to a high rule number. When the rule number is high then the model overfits. Finally 2 membership functions were used for model runs.

4.4 Membership function type

Fuzzy logic models are flexible in the selection of membership function type. The aim is to find the type of membership function that results the best model performance and models the system sufficiently. The model was tested using triangular, bell-shaped and Gaussian membership function. Finally the Gaussian membership function was selected although model performance was not significantly better. Gaussian membership function is given by eqn (11):

$$\mu_A(x) = e^{-(x-c)^2/2\sigma^2} \quad (11)$$

where c the point where membership function equals unit.

4.5 ANN model

The performance of the ANN model was examined in respect to the number of hidden layers when implementing the Levenberg-Marquardt algorithm. The choice of layer number was implemented through a trial and error procedure and affected model performance directly. The model reproduced the rainfall heights of N. Zichni and Nigrita stations satisfactory with the use of five hidden layers. The training set consisted of the mean monthly rainfall data from years 1980 to 1995 and validation set the data from 1996 to 2003.

4.6 Model evaluation

Model evaluation is the procedure where the results derived from both models are compared to the observed values. This is carried out for both training and validation set. Model performance is measured using reduced mean square error and coefficient of determination (R^2). Reduced mean square error is:

$$reduced \text{ MSE} = \frac{1}{n} \sum_{i=1}^n \left(\frac{Y_{obs} - Y_{calc}}{Y_{obs}} \right)^2 \quad (12)$$

where Y_{obs} the observed and Y_{calc} the calculated values.

5 Results

Model performance was measured for both training and validation sets using reduced mean square error and coefficient of determination (R^2) between calculated and observed values. The results obtained from model execution are shown in tables 2 and 3 for N. Zichni and Nigrita meteorological stations respectively. It must be noted that a high number of runs was performed for both

Table 2: Model performance for calculating N. Zichni station rainfall heights.

Method	Set	R ²	Reduced MSE
ANFIS	Training	0,7097	0,270025
“	Validation	0,6775	0,311568
ANN	Training	0,6469	0,307301
“	Validation	0,7069	0,297889

Table 3: Model performance for calculating Nigrita station rainfall heights.

Method	Set	R ²	Reduced MSE
ANFIS	Training	0,7352	0,261991
“	Validation	0,761	0,298543
ANN	Training	0,6769	0,302348
“	Validation	0,7798	0,267172

models in order to achieve the best performance. The results shown in tables 2 and 3 are the ones obtained with the parameter selection that performed better. The results obtained from model execution are also shown in fig. 3 and 4 graphically.

The best performance for the ANFIS model was achieved with the use of two membership functions for the division of the domain of the fuzzy premises. This means that the rule system consisted of four fuzzy rules. When the rule number increased, then the model over fitted. This means that model performance was high for the training and poor for the validation set. In this case the model is not suggested for prediction because it loses its ability to generalize. For this reason the purpose was to find the proper rule number in order to achieve similar model performance for the training and the validation sets.

Three different membership function types were tested to divide the domains of Serres and Achladochori rainfall heights. The fuzzy logic model was finally constructed with the use of Gaussian membership functions. However it must be noted that the difference in model performance was not noticeable between this type of membership function and the other two types.

Concerning the ANN model, the effect of the hidden layer number was examined in model performance. Several model run were performed in order to find the number of hidden layers which result to the best model performance. Five hidden layers were finally used. It must be noted that the hidden layer increase resulted in model over fitting, just like the fuzzy logic model. The best result was achieved when model performance was balanced between two sets.

Both fuzzy logic and ANN model reproduced the rainfall time series satisfactory. Model performance was high and both models are suggested for prediction. However they show some weakness in achieving extreme values.

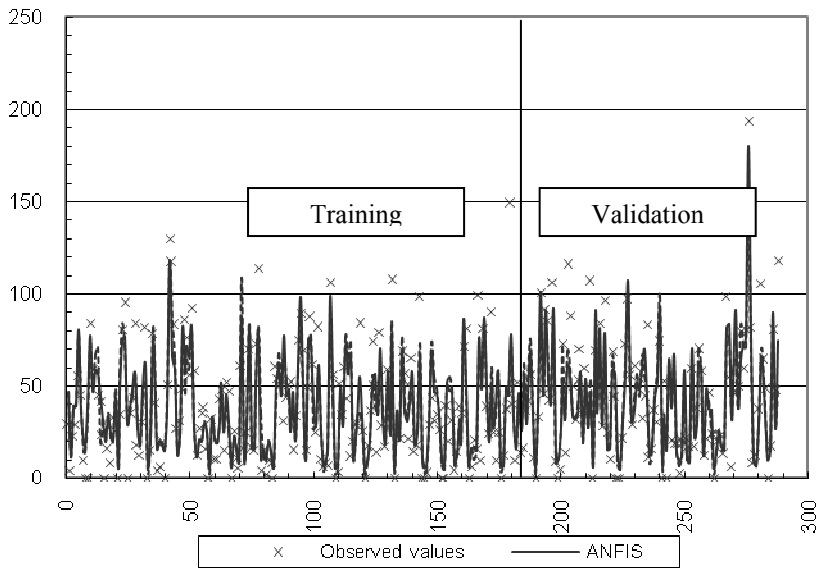


Figure 3: Mean monthly rainfall heights in N. Zichni.

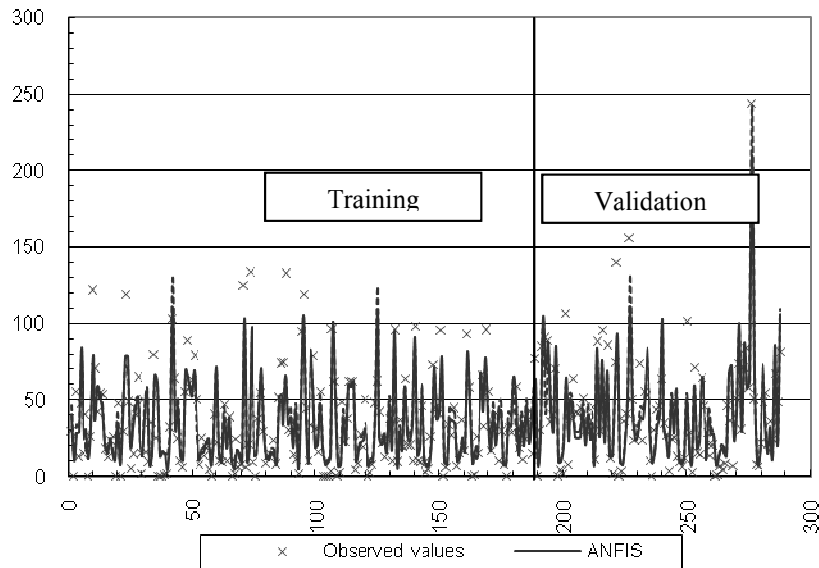


Figure 4: Mean monthly rainfall heights in Nigrita station.

Fuzzy logic model performed better in calculating rainfall heights for N. Zichni and Nigrita meteorological stations in comparison with the ANN model. The first one is much more flexible in modelling because of the greater parameter number that can be modified. The ANN is working much more like a “black box”, because the final result is extracted with a less transparent way.

Both models performed better in approximating Nigrita’s rainfall heights. We believe that is due to the fact input and output data are more correlated than in the case of N. Zichni station.

Acknowledgement

This research was funded by State Scholarship Foundation.

References

- [1] Aquil, M., Kita I. & Yano, A., Nishiyama, S., *Neural Networks for Real Time Catchment Flow Modelling and Prediction*, Water Resources Management. **21**:1781–1796, 2007
- [2] Gopacumar, R., Takara, K. & James, E., *Hydrologic Data Exploration and River Flow Forecasting of a Humid Tropical River Basin Using Artificial Neural Networks*. Water Resources Management **21**, 2007
- [3] Firat, M. & Turan, E., *Monthly river flow forecasting by an adaptive neuro-fuzzy inference system*. Water and environment journal., 2009
- [4] Antar, M., Ellassiouti, I. & Alam M., *Rainfall-runoff modelling using artificial neural networks technique: a Blue Nile catchment case study*. Hydrological Processes **20**, 2006
- [5] Altunkaynak, A., *Forecasting surface water level fluctuations of lake van by artificial neural networks*, Water Resources Management **21**:399–408, 2007
- [6] Bárdossy, A & Disse, M., *Fuzzy Rule-Based Models for Infiltration*. Water Resources **29** (2): 373-382, 1993
- [7] Bárdossy, A., The use of fuzzy rules for the description of elements of the hydrological cycle. Ecological Modeling **85**: 59-65, 1996
- [8] Russell, S & Campbell, P., *Reservoir Operating Rules with Fuzzy Programming*. Journal of Water Resources Planning and Management. **122** (3): 165-170., 1996
- [9] Shrestha, B, Duckstein, L & Stakhiv E., *Fuzzy rule-based modelling of reservoir operation*. Journal of Water Resources Planning and Management **122**(4): 262-269., 1996
- [10] Panigrahi, D & Mujumdar, P., *Reservoir Operation Modelling with Fuzzy Logic*. Water Resources Management. **14**: 89-109, 2000
- [11] Chang, Li-Chiu & Chang, Fi-John, *Intelligent control for modelling of real-time reservoir operation*. Hydrological Processes **15**: 1621-1634, 2001
- [12] Pongracz, R, Bogardi, I & Duckstein L., *Application of fuzzy rule-based modeling technique to regional drought*. Journal of Hydrology. **224**: 100-114, 1999



- [13] Abebe, A J, Solomatine, D P & Venneker R. G. W., *Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events*. Hydrological Sciences Journal **45 (3)**: 425-436, 2000
- [14] Mahabir, C, Hicks, F. E & Fayek R., *Application of fuzzy logic to forecast seasonal runoff*. Hydrological Processes. **17**: 3749-3762, 2003
- [15] Nayak, P, Sudheer, K. & Ramasastri K., *Fuzzy computing based rainfall-runoff model for real time flood forecasting*. Hydrological Processes. **19**: 955-968, 2005
- [16] Zimmermann, H. J., *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers: Boston, 2005
- [17] Jang, J-SR., *ANFIS: Adaptive-Network-based Fuzzy Inference Systems*. IEEE Transactions on System Man and Cybernetics. **23 (3)**: 665-685, 1993
- [18] Martin, JD, Morton, YT. & Zhou Q., *Neural network development for the forecasting of upper atmosphere parameter distributions*. Advances in Space Research. **36: no.12**. 2480-2485, 2005
- [19] Gurney, K., *An introduction to Neural Networks*, CRC Press, Boca Raton, 1997
- [20] Mathworks, *Neural Network Toolbox 6, Users guide*. MATLAB, 2007.

