12th International Conference on Hydroscience & Engineering
Hydro-Science & Engineering for Environmental Resilience
November 6-10, 2016, Tainan, Taiwan.

Similarity Based Neuro-Fuzzy System for Rainfall-Runoff Modeling in an Urban Tropical Catchment

Amin Talei[1], Lloyd Hock Chye Chua[2], Chai Quek[3]
1. Senior Lecturer, School of Engineering, Monash University Malaysia
Selangor, Malaysia
2. Associate Professor, School of Engineering, Deakin University
Melbourne, Victoria, Australia
3. Associate Professor, School of Computer Engineering, Nanyang Technological University
Singapore

ABSTRACT

Neuro-Fuzzy Systems (NFS) are computational intelligence tools that take advantage of low-level learning ability of neural networks and the high-level reasoning ability of fuzzy systems. Approximate Analogical Reasoning Schema (AARS) is one of the approaches which employs a similarity measure to decide on rule firing for a specific observation and is believed to be capable of adding more flexibility to NFS structure. In this study, an NFS model is developed based on AARS for event-based rainfall-runoff modeling in Sungai Kayu Ara, an urban catchment in Selangor State of Malaysia. The model performance is then compared with ANFIS and ARX models.

KEY WORDS: Rainfall-runoff modeling; Neuro-fuzzy system; Similarity Measure; ANFIS

INTRODUCTION

Modeling of rainfall-runoff process is one of the most important tasks in hydrology due to its vast applications in different hydrologic problems such as flood forecasting, design of spillways and waterways, water quality modeling, and water resources management. To-date several approaches have been introduced to model the rainfall–runoff relationship. These methods can be categorized into two main groups: physically-based models and system theoretic models. Physically-based models are designed to approximate the general internal sub-processes and physical mechanisms which govern the hydrologic cycle. They usually incorporate simplified forms of physical laws and are generally non-linear, time-varying, and deterministic, with parameters that are representative of watershed characteristics. Although physically-based models help us in understanding the physics of hydrological processes, they require sophisticated mathematical tools, and usually require significant user expertise. On the other hand, system theoretic models apply a different approach to identify a direct mapping between rainfall and runoff, without the need for a detailed consideration of the physical processes. Neuro-Fuzzy Systems (NFS) which combine the reasoning ability of fuzzy inference systems (FIS) with the learning ability of ANN through the incorporation of if-then rules have recently been used in many of hydrologic modeling. Based on the method chosen for determining the output, a FIS can be categorized as either linguistic also known as Mamdani-type NFS (Mamdani and Assilian, 1975) or precise which also known as Takagi-Sugeno-type NFS (Takagi and Sugeno, 1985) models. The NFS implemented has exclusively been the Takagi-Sugeno-type NFS while the Mamdani-type NFS has rarely if ever, been employed in rainfall-runoff modelling. The Mamdani-type NFS is advantageous as the consequent part of the rules is expressed in linguistic terms, which can help to provide a clearer description of the process as opposed to the Takagi-Sugeno-type NFS which specifies the rule consequents only as quantitative or crisp values. Neuro Fuzzy Syetems can also be categorized based on the reasoning schema adopted. Approximate reasoning is the process of concluding a possible imprecise conclusion from a group of imprecise premises (Zadeh, 1973). Approximate Analogical Reasoning Schema (AARS) is one of the available reasoning schema in literature (Turksen and Zhong, 1990) which is constructed based on similarity measures and to the best of our knowledge has not been used for hydrological modeling yet. AARS employs a similarity measure or SM (distinguishing between the degree of similarity or dissimilarity) to determine whether a rule should be fired for a specific observation in the pattern matching phase. The same similarity measure is used to construct a modification function (ModF) to modify the right side of the rule in the consequent deduction phase. The objective of this study is to investigate the capabilities of a Mamdani-type NFS using the Approximate Analogical Reasoning Schema (AARS) in rainfall-runoff modeling and compare its performance with the commonly used Takagi-Sugeno-type NFS, Adaptive Network-based Fuzzy Inference System (ANFIS). In this paper, the implementation of the AARS within a NFS model, known as Pseudo Outer Product Fuzzy Neural Network (POPFNN-AARS) developed by Quek and Zhou (1999) will be applied to model the rainfall-runoff process for an urban tropical catchment to assess the applicability of similarity-based NFS for rainfall-runoff modeling. For convenience, POPFNN-AARS will be abbreviated as POPFNN in this study.

APPROXIMATE ANALOGICAL REASONING SCHEMA (AARS)

Given an observed fact A' and a simple fuzzy rule "if A then B"; the basic procedure in AARS is to modify the consequence B of the fuzzy rule according to the similarity between the observed value or A' and the antecedent A, which are compared against a threshold. Based on this comparison, a rule can be fired and the conclusion B' can be deduced using a modification function which is constructed based on the same similarity measure Quek and Zhou (1999). As can be seen from Figure 1, a rule will be fired to estimate the true proposition of the consequence (B) when the observed variable (A') is similar enough (SM larger than a threshold) to a true proposition (A) of the antecedent. Then by using a modification function, the conclusion (B') will be deduced from the true proposition B.

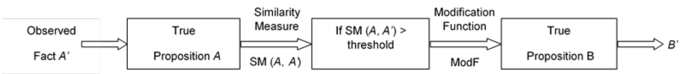

Fig 1. Schematic representation of AARS reasoning schema

Distance measures (DM) have traditionally been employed in the development of similarity measures. In these models, the similarity or dissimilarity between different points in space can be represented by a distance measure (Zwick, Carlstein, and Budescu, 1987; Turksen and Zhong, 1990). Thus, a similarity measure can be expressed as:

$$SM = \frac{1}{1 + DM} \quad (1)$$

In AARS, a fuzzy rule "if A then B" is to be fired by using a ModF which modifies the consequent B of the rule based on the SM between the observed value A' and the antecedent A. The ModF is thus adjusted in such a way that the whole system can function as closely as possible to the real situation, based on the SM. Turksen and Zhong (1988) introduced two types of ModF; namely: (1) Expansion form which is equivalent to the linguistic term of 'more or less'; and (2) Reduction form.

POPFNN

POPFNN is a five-layer network that incorporates the connectionist structure of the neural network to implement a fuzzy rule-based system and the AARS fuzzy inference model. The five layers are defined as: Layer (1) - Input layer, Layer (2) - Condition layer, Layer (3) - Rule-Base layer, Layer (4) - Consequence layer, and Layer (5) - Output layer. A detailed description for each layer is available in Quek and Zhou (1999). The learning process in POPFNN adopts batch or offline learning and consists of three phases; namely: (1) Self-organization for initialization of parameters; (2) POP learning for rule identification, and (3) Supervised learning for the fine-tuning of parameters.

Initialization

Membership functions of the input and output-label nodes of POP-FNN are important since the fuzzy information is stored in the membership functions. The correct determination of the centroids and widths of these membership functions is necessary if information in the training data is to be properly captured and stored. The Kohonen's feature-map algorithm (Kohonen, 1989) is adopted in POP-FNN to identify the initial centroids and widths of membership functions of the input and output variables. At this stage, these centroids and widths are initial estimates; in the third phase of learning, the centroids and widths will be fine-tuned by a supervised learning process.

Rule Identification

POFNN uses a simple one-pass algorithm known as the Pseudo Outer-Product (POP) learning rule (Zhou and Quek, 1996). Consider a simple example of having two input variables x_1 and x_2 and an output variable y_1 where $X^T = [x_1, x_2] \in D_1$, $Y^T = [y_1] \in D_2$ and suppose that after the self-organization phase, each linguistic variable has a label set of {small, medium, large} that is abbreviated as {S, M, L}. The POPFNN learning algorithm considers all the possible rules (see Fig. 4) and each of these rules is fully connected to the output-label nodes in the consequence layer. Once the membership functions have been determined, the set of training data ($X^{(r)}$, $Y^{(r)}$) is fed into both the input and output layers simultaneously. When $X^{(r)}$ is presented at the input layer, the membership values of each input-label node is derived using the membership functions that have earlier been determined during the initialization phase. Subsequently, the firing strength of the kth rule node in the rule-base layer is depending on the type of modification function used. Similarly, $Y^{(r)}$ is fed into the output layer to calculate the membership values of the output-label nodes. For instance, if the training data satisfies the rule "if x_1 is large and x_2 is small then y_1 is medium" then the derived membership functions will have large values at the corresponding input-label nodes, namely the input-label node "L" for linguistic variable "x_1", the input-label node "S" for the linguistic variable "x_2", and the output-label node "M" for the defuzzification node "y_1" (see Fig. 2). On the other hand, the membership values of the other input-label nodes in the condition layer as well as the output-label nodes in the consequence layer will be relatively small. Therefore, the firing strength of the rule which has the label "L" for the input variable "x_1" and the label "S" for the input variable "x_2" as its condition will be larger than other rules. If the weights of the links which connect the rule nodes and the output-label nodes are raised by the product of their firing strength and membership function values, then the improvement in the strength of the link between the rule node having conditions "x_1 is L" and "x_2 is S" and consequence "y_1 is M" will be greater than other links. This is shown as a solid line in Fig. 2. Consequently, after training the fuzzy rules embedded in training data will have larger weights compared to other rules. After the POP learning process, the weights represent the strengths of the fuzzy rules having the corresponding output-label nodes as their consequence. Amongst the links between a rule node and all the output-label nodes of a defuzzification node, the link with the largest weight is chosen and others deleted. When the weights of the links between a rule node and the output-label nodes are very small this indicates that there is little or no relation to the output variable represented by that particular defuzzification node. Hence, among all the links between a rule node and all the output-label nodes, the link with the largest weight is chosen while others can be deleted without affecting the outputs.

Supervised Learning

The well-known back-propagation algorithm (Werbos, 1988) is employed for this phase where the error signals are calculated and then are fed back to the system to adjust the centroid and width of membership functions for both output-label and input-label nodes.

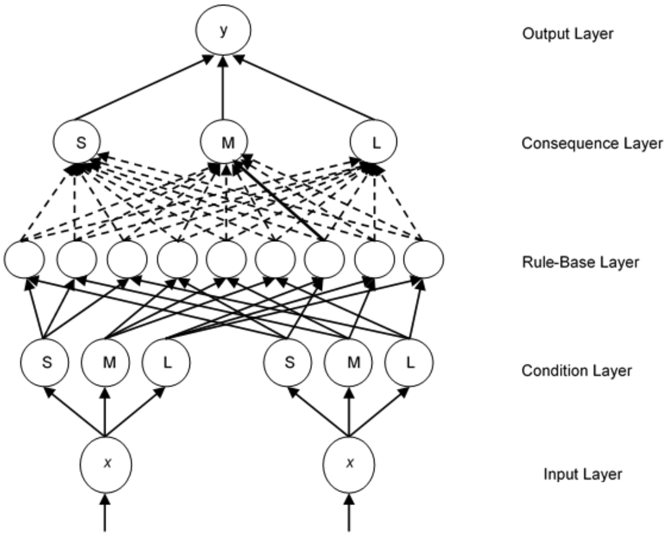

Fig. 2. Sample POP-FNN with two inputs and one output linguistic variable

STUDY SITE AND DATA USED

Sungai Kayu Ara river basin is located in southeast of Kuala Lumpur, Malaysia, and covers an area of 23.22 km² as shown in Fig. 3. The main river of this basin originates from the reserved highland area of Penchala and Segambut. Sungai Kayu Ara river basin lies in equatorial zone. In this study, 24 major 10-minutes interval rainfall-runoff events between March-1996 to July-2004 are considered from which the first 18 (chronological order) were used for training while the remaining 6 events were used as testing data.

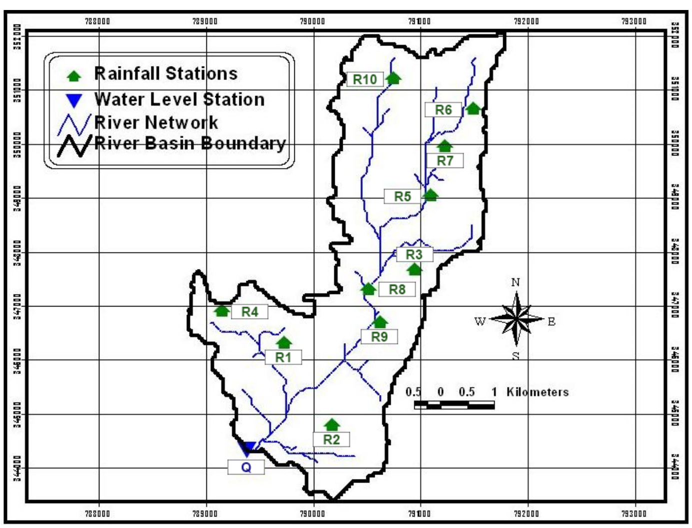

Fig. 3. Schematic map of Sungai Kayu Ara catchment, Selangor, Malaysia.

Data standardization were implemented before model training and testing. Standardization concentrates the dispersed data to a defined interval. All input and output data were standardized over an interval between 0.1 and 0.9 using a standardization method proposed by Rajurkar, Kothyari, and Chaube (2002).

INPUT SELECTION AND MODEL DEVELOPMENT

Inputs for all models of this study consisted of antecedent rainfall from 10 different rainfall stations and antecedent discharge from the outlet station. The rainfall-runoff time series of the 18 training events was used for input selection process. Two criteria were applied in the selection of inputs for both POPFNN and ANFIS models used in this study. Firstly, the inputs used should be highly correlated with the output and secondly, the inputs used should possess low mutual information (Talei and Chua. 2012). The input selection process showed that for POPFNN model the input combination of R1(t-2), R3(t-4), R5(t-2), and Q(t-1) gives the best performance in simulating runoff at present time Q(t); however, for ANFIS the input selection process resulted in R1(t-8), R3(t-3), R5(t-6), Q(t-1). It was concluded that the input selection method has chosen rainfall inputs from the same rainfall stations (stations 1, 3, and 5) for both models. Therefore, the afore-mentioned input combinations were considered to be used in testing phase.

Using 2 Gaussian membership functions for each input variable and the output gave the best performance during the training process in POPFNN. Other parameters of the model were chosen based on trial and error during training phase. In ANFIS, employing 2 triangular membership functions for each input variable gave the best results in training. Moreover, a sensitivity analysis was also conducted for both models to identify the proper number of epochs to avoid over-fitting. Epoch number of 60 was resulted and adopted for both POPFNN and ANFIS models.

RESULTS AND DISCUSSION

POPFNN model results in testing phase were compared with the ones were obtained by ANFIS model. For further comparison an ARX model was also calibrated with the same training data set and was used for predicting Q(t) in testing data set. The average coefficient of efficiency (CE), r^2, RMSE, MAE, and relative peak error (RPE) values obtained by the 3 models for the 6 testing events of this study are compared in Table 1. As can be seen, POPFNN performed comparable to ANFIS it outperformed ARX model significantly in terms of all statistics. In peak estimation, POPFNN model was able to produce the best result as it gave the lowest RPE value compared to the other two models.

Table 1. Comparison of average CE, r^2, MAE, RMSE, and RPE values of 6 testing events obtained by POPFNN, ANFIS, and ARX models

Models	CE	r^2	RMSE (m^3/s)	MAE (m^3/s)	RPE
POPFNN	0.85	0.86	5.18	3.18	0.09
ANFIS	0.87	0.88	4.96	2.49	0.11
ARX	0.57	0.59	7.92	5.86	0.27

A qualitative assessment of the predicted hydrographs is demonstrated in Fig. 4 to compare the observed and simulated Q(t) hydrographs by POPFNN and ANFIS models for Event 6 which is the biggest event in terms of peak flow. As can be seen, POPFNN was successful to predict the peak quite accurately (RPE = 0.06) while ANFIS overestimated the peak (RPE = 0.20). It is worth mentioning that both models had some delay in estimating the major peak of this event.

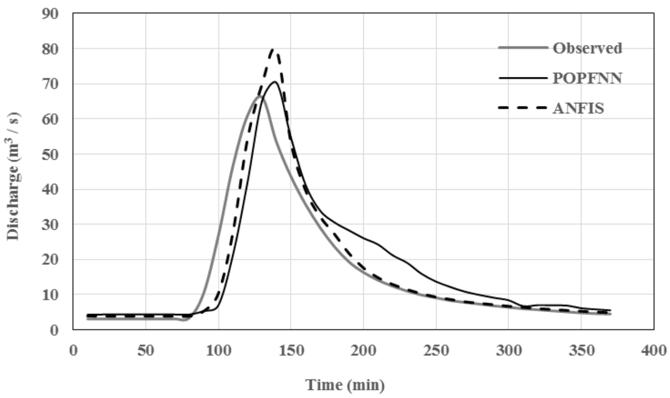

Fig.4. Comparison between the observed and simulated hydrographs by POPFNN and ANFIS models for event 6.

As seen in the results, POPFNN showed comparable performance to ANFIS in terms of different statistics. It is worth mentioning that the rule creation procedure for POPFNN and ANFIS are different. For ANFIS, which is a Takagi-Sugeno-type NFS, the output is a crisp value which is defined by a linear function while in POPFNN, which is a Mamdani-Type NFS, the output is defined by fuzzy sets. Therefore, if n membership functions are specified for each of the input variables, n^q rules where q is the number of input variables will be produced in ANFIS while $n^q \times m$ rules where m is the number of membership functions for each output variable will be produced in POPFNN. In most batch learning models, n has to be selected by trial and error. Although the number of rules can be increased by the specification of greater number of membership functions, this can result in over-fitting leading to deterioration in the model results. On the other hand, decreasing the number of rules by choosing a smaller number of membership functions can also worsen model performance due to an insufficient number of rules required to properly capture the associations between inputs and output variables. POPFNN learning algorithm incorporates a rule pruning mechanism that begins with a large number of $n^q \times m$ rules, with the unused or less important rules removed during the training process. This represents an advantage over the learning mechanism used in ANFIS where the number of rules is determined by the user from the specification of the number of member functions, where an improper choice may lead to redundant rules that may result in a deterioration in model performance. In contrast, POPFNN has the ability to select only the essential rules, via the rule pruning procedure, from all possible rules. Thus, the deteriorating effect of having redundant or an excessive number rules is avoided.

CONCLUSIONS

The following can be concluded from this study:

(i) Pseudo Outer Product Fuzzy Neural Network (POPFNN) was tested against ANFIS and ARX models for a tropical urban catchment. POPFNN was comparable to ANFIS and superior to ARX models in terms of different statistics.

(ii) Rule pruning mechanism used in the POPFNN learning algorithm has a more flexible rule structure with an optimal number of rules compared to ANFIS which has a fixed number of rules. This can be considered as an advantage of POPFNN model over ANFIS model in capturing the association between input and output.

REFERENCES

Kohonen, T., 1989. Self-organization and associative memory. Springer.

Mamdani, E.H., Assilian, S., 1975. Experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 7(1): 1-13.

Quek, C., Zhou, R.W., 1999. POPFNN-AARS(S): A pseudo outer-product based fuzzy neural network. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 29(6): 859-870.

Rajurkar, M.P., Kothyari, U.C., Chaube, U.C., 2002. Artificial neural networks for daily rainfall-runoff modelling. Hydrological Sciences Journal, 47(6): 865-878.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics, 15(1): 116-132.

Talei, A., Chua, L.H.C., 2012. Influence of lag time on event-based rainfall-runoff modeling using the data driven approach. Journal of Hydrology, 438-439(0): 223-233.

Turksen, I.B., Zhong, Z., 1990. An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets. Fuzzy Sets and Systems, 34(3): 323-346.

Werbos, P.J., 1988. Backpropagation: Past and future. IEEE International Conference on Neural Networks. Publ by IEEE, San Diego, CA, USA, pp. 343-353.

Zadeh, L.A., 1973. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man and Cybernetics, SMC-3(1): 28-44.

Zhou, R.W., Quek, C., 1996. POPFNN: A pseudo outer-product based fuzzy neural network. Neural Networks, 9(9): 1569-1581.

Zwick, R., Carlstein, E., Budescu, D.V., 1987. Measures of similarity among fuzzy concepts: A comparative analysis. International Journal of Approximate Reasoning, 1(2): 221-242.