Plant Archives Vol. 21, Supplement 1, 2021 pp. 2253-2260 e-ISSN:2581-6063 (online), ISSN:0972-5210

Plant Archives

Journal homepage: http://www.plantarchives.org

doi link : https://doi.org/10.51470/PLANTARCHIVES.2021.v21.S1.371

RUNOFF ESTIMATION IN URBAN CATCHMENT USING ARTIFICIAL NEURAL NETWORK MODELS

Mahsa Bakhshaei[1]-Hassan Ahmadi*[2]-Baharak Motamedvaziri[3]- Payam Najafi[4]

[1]Department Forest, Range and Watershed management, Science and Research Branch, Islamic Azad University, Tehran, Iran; Email: Mahsabakhshaei@gmail.com

[2]Department Forest, Range and Watershed management, Science and Research Branch, Islamic Azad University, Tehran, Iran; Email: hassanahmadiiau@gmail,

[3]Department Forest, Range and Watershed management, Science and Research Branch, Islamic Azad University, Tehran, Iran; Email: b-bmvaziri@gmail.

[4]Department of Agriculture, Faculty of Agriculture, Kkhorasgan Branch, Isfahan, Iran; Email: payam.najafi@gmail.com

ABSTRACT

Many types of physical models have been developed for runoff estimation with successful results. However, accurate estimation of runoff remains a challenging problem owing to the lack of field data and the complexity of its hydrological process. In this paper, a machine learning method for runoff estimation is presented as an alternative approach to the physical model. Various types of input variables and artificial neural network (ANN) architectures were examined in this study. Results showed that a two-layer network with the tansig activation function and the Levenberg–Marquardt learning algorithm performed the best. For this architecture, the most effective input vector consists of a catchment perimeter, canal length, slope, runoff coefficient, and rainfall intensity. However, results of multivariate analysis of variance indicated the significant interaction effect of input data and the ANN architecture. Thus, to create a suitable ANN model for runoff estimation, a systematic determination of the input vector is necessary.

Keywords: Urban catchment, runoff estimation, artificial neural network, machine learning

Introduction

Storm water and runoff management are common issues in most urban catchments (Whitford, Ennos *et al.*, 2001; Zhang, Xie *et al.*, 2012; Kumar, Arya *et al.*, 2013). The hydrological process in urban catchments is complicated (Freni, Mannina *et al.*, 2009) and involves a complex network of impervious and vegetative surfaces, canals, sewerages, pipelines, etc. (Whitford, Ennos *et al.*, 2001; Zhang, Xie *et al.*, 2012; Kumar, Arya *et al.* 2013). Most urban catchments are ineffective for hydrometric measuring instruments.

The fundamental part of all storm-water runoff management models is the accurate estimation of surface runoff (Chen and Adams, 2007). Runoff forecasting is essential for planning, designing, and operation of water resource projects (Reddy, Babu *et al.*). During the past few decades, runoff estimation has greatly benefitted from conceptual modeling, which retains some of the physical laws in its mathematical formulations (Elshorbagy, Simonovic *et al.*, 2000). However, these models rely on a large amount of input data (Elshorbagy, Simonovic *et al.*, 2000). Therefore, producing output from them is costly (Elshorbagy, Simonovic *et al.*, 2000), and a high uncertainty exists in the results (Freni, Mannina *et al.*, 2009).

In cases of limited data and process complexity, using machine learning techniques is a suitable approach (Chae, Horesh *et al.*, 2016). The artificial neural network (ANN) is a subgroup of machine learning that has received significant attention in the context of estimation problems (Khayatian and Sarto, 2016). Over the past few decades, ANN models have become very widely used in the fields of hydrology, water resources, and watershed management (Chavoshi, Sulaiman *et al.*, 2013, Orimi, Farid *et al.*, 2015).

Elshorbagy *et al.* (2000), for example, studied the applicability and usefulness of ANN models in runoff prediction (Elshorbagy, Simonovic *et al.*, 2000). By developing various ANN-based models in the Red River Valley, Canada and comparing them with traditional techniques, they concluded that ANN-based models yield better results and have a better prediction ability. Similarly, Ahmad and Simonovic (2001) used a feed-forward ANN with a back-propagation algorithm for predicting the peak flow, timing, and shape of a runoff hydrograph of the Red River in Manitoba, Canada (Ahmad and Simonovic, 2001).

To analyze the performance of ANN models for forecasting short-term daily flow, Pulido-Calvo and Portela (2007) applied a feed-forward neural network in large Portuguese watersheds (Pulido-Calvo and Portela 2007). They claimed ANN models can predict watershed flow using insufficient data. Reddy *et al.* (2008) modeled the rainfall-runoff process using empirical models and compared it with ANNs (Reddy, Babu *et al.*). They used the data on the Godavari Basin of India and explored the ANN performance improvement by combining it with empirical methods. Lee *et al*. (2010) built two types of ANN models for prediction of regional runoff utilization and compared their reliabilities (Lee, Lin *et al*. 2010). A network with a radial base function using the Gaussian function showed better stability than a neural network model using back-propagation.

Chiang *et al*. (2004) studied the stability and effectiveness of two ANN types: static feed-forward and dynamic feed-forward (Chiang, Chang *et al*. 2004). They applied various ANN architectures to the Lan-Yang River, Taiwan and showed that both static and dynamic neural networks yielded reasonable results. However, the static-feed forward type showed better performance than the dynamic feed-forward type if the data were sufficient. In the case of insufficient training data, the dynamic feed-forward ANN demonstrated significantly better performance. Meanwhile, Chavoshi *et al*. (2013) applied ANN for flood estimation in the southern strip of the Caspian Sea watershed (Chavoshi, Sulaiman *et al*., 2013). They compared their results with a multiple regression model and showed the ANN model to be a powerful tool for resolving the hydrological problem complexity. Among the different types of ANN architectures, multilayer feed-forward back propagation with the Levenberg–Marquardt resulted in the best performance.

A broad review of literature in water resource management and hydrology indicates the following points. (1) Several studies were conducted to investigate the applicability of ANNs to forecast runoff in different watersheds and to compare them with traditional physical models. Most of these studies showed the acceptable performance of ANN models, particularly at watersheds with insufficient data. (2) In addition, exploring the ANN architecture with the best performance has been the focus of researchers. Accordingly, various ANN structures were designed and tested through changing neural network components, including several neurons and layers, transform functions, learning methods, and network types. Although a feed-forward perceptron network was recommended by many researchers, there is no consensus on network structure. (3) Few works have focused on studying the effect of the input vector on ANN model performance for runoff estimation. (4) Moreover, few studies have focused on the application of ANNs in urban watersheds. Particularly, owing to the complexity of the hydrological process in urban catchments and the lack of field data (Bertrand-Krajewski 2007), this research area requires more attention.

The aim of this study is thus to determine the ANN architectures that result in the most accurate performance for urban catchment estimation. To this end, a total of 24 ANN models were proposed and tested. The performances of the proposed models were systematically compared. In addition, this study served to explore the interaction effect of input vectors on ANN architecture.

Artificial neural network

An ANN is an information-processing system that shares certain performance characteristics with biological neural networks (Fausett 1994). An ANN consists of a large number of interconnected computational nodes, called neurons, working together (Sethi, Kumar *et al*. 2010). Generally, a neural network consists of three layers: input, middle (hidden), and output layers, which are fully connected. The input layer represents entries; the output layer represents the corresponding values. In the middle layers, there exist several artificial neurons comprised of the activation function (weights and biases to calculate output values), as well as the transfer function for propagating values to subsequent layers. An important characteristic of the ANN is its ability to learn. Learning is the process by which a neural system acquires the ability to carry out certain tasks by adjusting its internal parameters according to some learning scheme (Karayiannis and Venetsanopoulos 2013).

A neural network is characterized by its architecture, which represents the pattern of connections among neurons, its method of determining the connection weights, and the activation function (Fausett, 1994). A typical ANN is the multilayer perceptron (MLP). In MLP, the direction of information flow is feed-forward (where the information flows from the input nodes to output nodes). The learning process is supervised with the back-propagation algorithm. Many studies have shown the ability of MLP to solve complex and diverse problems (Haykin, Haykin *et al*., 2009).

In addition to the configuration of layers and the training algorithm, the number of neurons in the middle layer is significant to ANN performance. An ANN with too few neurons in the middle layer is not capable of making an accurate output, while an ANN with too many neurons in the middle layer is over-fitted and has poor predictive performance (Chae, Horesh *et al*., 2016). To determine the number of hidden layers and neurons, either trial-and-error or intelligent methods can be used (Najafi-Marghmaleki, Khosravi-Nikou *et al*. 2016).

Study area

The area selected for this study is located in the southwest of Isfahan, Iran, encompassing 69 km^2. It is located in a low rainfall zone, with the average annual precipitation of 127.2 mm over the past two decades. To the north and northeast lies the Zayanderood River. To the west, it is surrounded by a residential district. To the east and southwest is an area of elevated terrain. It is located between 51°39´and 51◦43´ E longitude and 32◦35´ to 32◦38´ N latitude (Fig. 1). The study area is characterized by a diverse topography with an overall slope of 2.5%. The land slope in the northern direction is steep toward the Zayanderood River; the slope in the western direction is moderate. Runoff canals flowing through urban areas lead to the Zayanderood River.

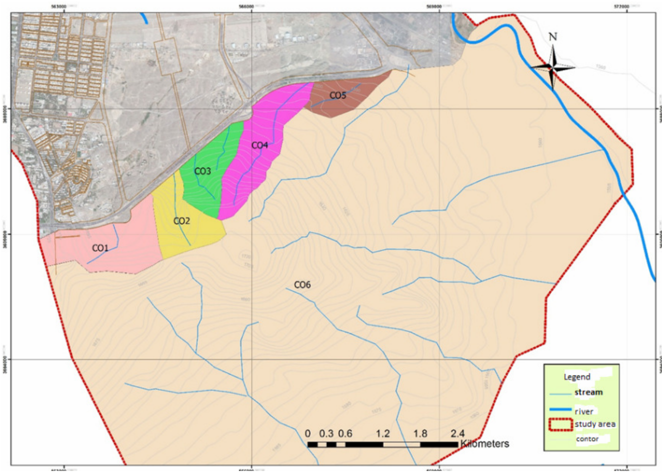
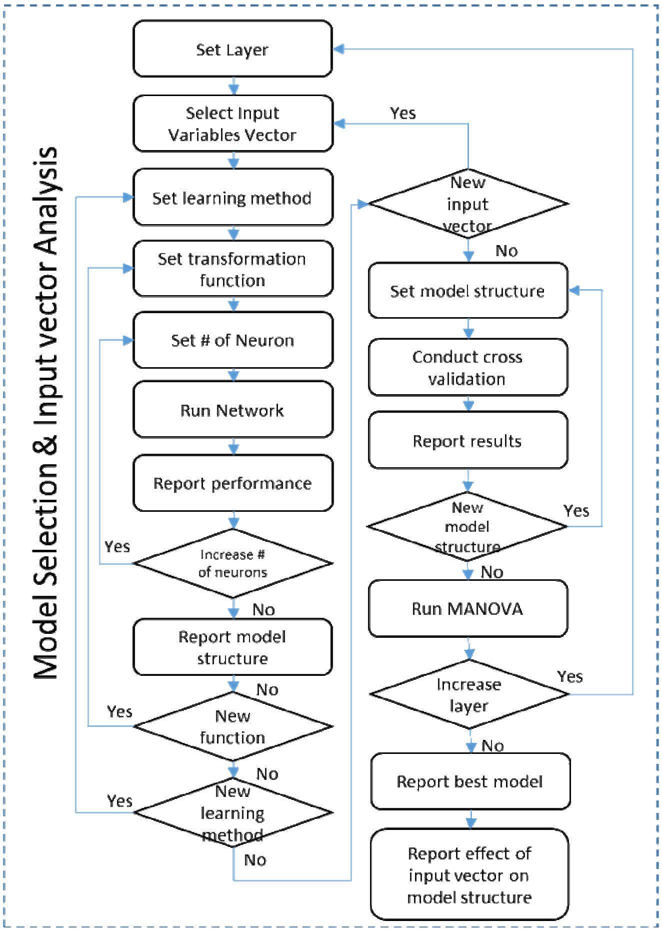


Fig. 1 : Area study.

The study area was divided into two parts: urban and suburban watersheds. The suburban catchment consisted of six sub-watersheds (CO-1 through CO-6); the urban catchment included 35 sub-watersheds (CI-1 through CI-35). Since the runoff in CO-6 flowed out of the study area, this sub-watershed was omitted. For each sub-watershed in the urban catchment, the physiographic parameters (area, perimeter, canal length, slope) and time of concentrations were calculated. The runoff coefficient for different land was obtained from American Society of Civil Engineering (ASCE). The rainfall-runoff data from 2000 to 2016 were used for model development.

Methodology

The methodology adopted in this study consisted of two phases. Phase 1 was dedicated to model selection and input vector analysis. In this phase, through changing network components, including the number of neurons, transform functions, learning methods, and hidden layers, various ANN models were developed and evaluated. The interaction effect of the input vector on the ANN structure was analyzed by using multivariate analysis of variance (MANOVA) techniques. The data set for MANOVA was generated by a cross-validation procedure. The second phase involved the applicability of ANN models for runoff estimation. For this purpose, the ANN model outputs were compared with the SWMM results. By implementing MANOVA, the significant differences between these models were studied. A detailed description of the methodology is illustrated in Fig. 2.



ANN architecture selection

An MLP artificial neural network with a back propagation algorithm was used to estimate the runoff in the urban watersheds. MLP is a prominent ANN architecture that is used in many water resource and hydrological applications (Braddock, Kremmer *et al.* 1998, WANG, Traore *et al.* 2008).

In many nonlinear problems, use of a single hidden layer is sufficient (Funahashi 1989, Hornik, Stinchcombe *et al.* 1989, Sreekanth, Sreedevi *et al.* 2011). Furthermore, studies have shown that using more than two hidden layers may not produce considerable improvement (Patuwo, Hu *et al.* 1993). In this study, we examined both a one-layer and two-layer network. To determine the number of neurons in the hidden layers, we applied the following rules. (1) The number of neurons in the first layer should not be exceeded by three times the number of input variables. (2) The number of neurons in the second hidden layer should be limited to two times the number of neurons in the first layer.

The linear activation function and logistic sigmoid function are the most widely used functions in the output layer and hidden layer, respectively (Sivakumar, Jayawardena *et al.* 2002). A study by Yonaba *et al.* (2010) showed that the tangent sigmoid is the most pertinent transfer function for stream-flow forecasting (Yonaba, Anctil *et al.* 2010). They found that a nonlinear transfer function in the output layer failed to improve performance value. To obtain the best ANN architecture, both the logistic sigmoid function and tangent sigmoid are considered in this study.

Learning method selection

Various ANN learning algorithms exist, such as the scaled conjugate gradient (SCG), Levenberg–Marquardt (LM), and resilient back-propagation (Ruck, Rogers *et al.* 1990). Based on performance statistics for back-propagation algorithms, the LM is the best (Affandi and Watanabe 2008). In this research, we used both LM and the Bayesian regularization (BR) algorithm in the training procedure.

Input vector selection

In contrast to statistical methods, ANNs are categorized into various data-driven approaches (Chakraborty, Mehrotra *et al.* 1992). Therefore, selecting a set of appropriate input vectors is a critical step in the process of ANN model development (Zealand, Burn *et al.* 1999, Dogan, Demirpence *et al.* 2008). The input vector must be uncorrelated, free of noise, and have a significant relationship with the output vector (Bowden, Dandy *et al.* 2005). Data-driven approaches can usually determine the critical input vector; nonetheless, this approach is not efficient (Bowden, Dandy *et al.* 2005). By increasing the number of variables, computational complexity, learning process difficulty, low accuracy, and poor performance will result (Back and Trappenberg 1999, Maier and Dandy 2000, Bowden, Dandy *et al.* 2005).

Despite the importance of input vector determination on ANN performance, Maier and Dandy (2000) claimed that, in most water-resource ANN applications, minimal attention is given to the task of selecting appropriate model input (Maier and Dandy 2000). In this study, we employed a combination of input determination methods, including the "prior knowledge" method (Bowden, Dandy *et al.* 2005) and "saliency analysis" method (Abrahart, See *et al.* 2001) to select the appropriate input vector.

According to these approaches, two vectors of hydrological variables are defined. With vector 1, the input variables consist of the catchment area, concentration time, rainfall intensity, and runoff coefficient. As in the hydrological watershed, the variable time of concentration can be estimated by experimental approaches. Thus, with the second input vector, the concentration time was substituted by affecting the variables consisting of the catchment perimeter, channel length, slope, runoff coefficient, and rainfall intensity. Accordingly, the relationships among these variables were explored and the urban runoff value was estimated.

Data preparation

Since the acceptable data range for the sigmoid activation function is mostly in the range of −1 to 1, the normalization must be performed to place input data in the range of −1 to 1 before applying the data to ANN. For normalization, the following equations are used:

$$\mathrm{X_N} = 2 \times \left(\frac{\mathrm{X - Min\,X}}{\mathrm{Max\,X - Min\,X}}\right) - 1 \qquad (1)$$

where x is the original data for each input variable, Min X and Max X are respectively the minimum and maximum value of X, and X_N is the normalized value. For operation of an ANN, it is usually required to divide the dataset into three subsets for the purpose of training, validation, and testing. Training handles the weight values of the network. During the training phase, approximately 75% of the whole dataset is frequently fed to the network until the acceptable weight values are determined. The purpose of validation is to ensure the proper training and to avoid over-fitting or over-training. A total of 12.5% of the dataset was chosen for validation. For the final evaluation of the ANN performance, the remaining 12.5% of the dataset was used.

Evaluation criteria

To assess ANN performance during training, validation, and testing, two evaluation measures were applied. A mean squared error (MSE) is one of the most commonly used performance measures in hydrological modeling (Elshorbagy, Simonovic *et al*., 2000). The other index used to evaluate the correlation between observed and predicted runoff was the coefficient of determination, R^2. The formulas for MSE and R^2 are as follows:

$$\mathrm{R}^2 = 1 - \frac{\sum e_i^2}{\sum y_i^2} \qquad (2)$$

$$MSE = \frac{1}{N}\sum_{i=0}^{N} e_i{}^2 \qquad (3)$$

$$e_i = \mathrm{Y_i} - \hat{Y}_i \qquad (4)$$

where $\mathrm{Y_i}$ denotes the observed (actual) value of runoff, $\hat{Y}_i$ is the estimated value, and $N =$ the number of observations.

Results and Discussion

To determine the appropriate ANN configuration for obtaining satisfactory results, various ANN models with two input vectors were investigated. Each model was developed by using different network model parameters, such as learning algorithms (LM, BR), activation functions (logsig, radbas, tansig), numbers of hidden layers (one and two), and four to nine neurons in the hidden layers. These models were trained 84 times and the best performances were documented.

Results for input vector 1

Table 1 illustrates the values of statistical indicators for a total of six ANN models with input vector 1 during training and testing periods. As mentioned earlier, input vector 1 consists of variables, including catchment area, concentration time, rainfall intensity, and runoff coefficient. The differences between the models related to the number of neurons, the activation function form, and training method. The results from the model performances indicated that the single-layer network with five neurons—when the activation function was radbas, and the training algorithm was LM—performed the best. This network resulted in an R^2 of 0.853 for the testing dataset; an MSE of 0.96 m^6 for the testing dataset, and 0.6 m^6 for the training dataset, respectively.

To investigate the influence of the hidden layer on network performance, other combinations of ANN models with input vector 1 were developed. In these models the number of layers was increased by two, and different network parameters, including the number of neurons, activation function forms, and training algorithms were examined. For input vector 1, the results from the model performance (Table 2) indicated that, when the number of hidden layers increased by two, a network consisting of five and eight neurons with logsig and tansig activation functions, respectively, performed the best. In this combination, the best training algorithm was LM. This network resulted in an R^2 of 0.957 for the testing dataset, an MSE of 0.53 m^6 for the testing dataset, and 0.43 m^6 for the training dataset, respectively.

With input vector 1, a comparison of the statistical indicators displayed better performance for the network with two hidden layers. This model returned an MSE of 2.41 m^6, while the network with a single layer returned an MSE of 4.96 m^6. Moreover, in terms of the coefficient of determination, the network with two hidden layers demonstrated better performance. It was observed that the network with a single hidden layer returned 0.432, while the network with two hidden layers returned 0.704.

Results for input vector 2

Table 3 illustrates the values of statistical indicators for a total of six ANN models with input vector 2 during training and testing periods. As mentioned earlier, input vector 2 consisted of the variables of the catchment perimeter, channel length, slope, runoff coefficient, and rainfall intensity. Results of the model performance indicated that a single-layer network with seven neurons—when the activation function was logsig and the training algorithm was LM—performed the best. This network resulted in an R^2 of 0.886 for the testing dataset, an MSE of 0.69 m^6 for the testing dataset, and 0.11 m^6 for the training dataset, respectively.

To investigate the influence of the hidden layer on network performance, other combinations of ANN models with input vector 2 were developed. In these models, the number of layers was increased by two, and different network parameters, including the number of neurons, activation function forms, and training algorithms were examined. For input vector 2, the model performance results (Table 4) indicated that, when the number of hidden layers increased by two, the performances of the first three ANN architectures were very similar. However, among the six

ANN models, as outlined in Table 4, the network consisting of eight and nine neurons with tansign activation functions in both layers performed the best. In this architecture, the best training algorithm was LM. This network resulted in an R^2 of 0.987 for the testing dataset, an MSE of 0.05 m^6 for the testing dataset, and 0.002 m^6 for the training dataset, respectively.

As outlined in Table 5, a comparison of the proposed network performances indicates the following. (1) Input vector 2 provides better performance for runoff estimation of urban watersheds. (2) Increasing the number of hidden layers is often helpful for improving the runoff estimation in an urban catchment. (3) Two hidden layers with eight and nine neurons, respectively, and the tansig activation function in both layers, displays the best performance. The Mean Square Error (MSE), Sum Square error (SSE), and R^2 observed for this network architecture are 0.05 m^6, 0.314 m^6, and 0.987.

Table 1 : Performances of different ANN models with a one-layer network and input vector 1

Activation function	**No. of Neurons**	**Training Method**	**Validation**	**Training**	**Testing**		
			MSE	MSE	MSE	R^2	SSE
Logsig	6	LM	1.21	0.9	1.63	0.537	255.9
Radbas	5	LM	1.32	0.6	0.96	0.853	150.7
Tansig	7	LM	1.16	1.1	1.65	0.668	259.1
Logsig	7	BR	4.48	3.95	7.53	0.257	1182
Radbas	7	BR	1.77	1.66	1.76	0.255	276.3
Tansig	4	BR	4.29	3.52	16.23	0.025	2548

Table 2 : Performances of different ANN models with a two-layer network and input vector 1

Activation function Layer 1	**Activation function Layer 2**	**No. of Neurons Layer 1**	**No. of Neurons Layer2**	**Training Method**	**Validation**	**Training**	**Testing**		
					MSE	MSE	MSE	R^2	SSE
tansig	logsig	5	8	LM	0.56	0.43	0.53	0.957	83.21
tansig	radbas	7	8	LM	0.63	0.41	1.01	0.806	158.6
tansig	tansig	7	10	LM	0.29	0.28	0.71	0.918	111.5
tansig	logsig	7	9	BR	2.47	1.92	8.81	0.552	1383
tansig	radbas	7	10	BR	1.34	1.18	1.86	0.442	292
tansig	tansig	6	6	BR	2.03	1.58	1.59	0.547	249.6

Table 3 : Performances of different ANN models with a one-layer network and input vector 2

Activation function	**No. of Neurons**	**Training Method**	**Validation**	**Training**	**Testing**		
			MSE	MSE	MSE	R^2	SSE
logsig	7	LM	0.17	0.11	0.69	0.886	108.33
radbas	7	LM	0.98	0.69	2.27	0.744	356.39
tansig	7	LM	0.34	0.26	1.17	0.803	183.69
logsig	9	BR	1.63	1.35	1.34	0.694	210.38
radbas	9	BR	1.33	0.75	1.78	0.538	279.46
tansig	5	BR	1.87	1.54	2.29	0.438	359.53

Table 4 : Performances of different ANN models with a hidden two-layer network and input vector 2

Activation function Layer 1	**Activation function Layer 2**	**No. of Neurons Layer 1**	**No. of Neurons Layer2**	**Training Method**	**Validation**	**Training**	**Testing**		
					MSE	**MSE**	**MSE**	$\mathbf{R^2}$	**SSE**
tansig	logsig	8	7	LM	0.05	0.002	0.07	0.997	0.314
tansig	radbas	9	9	LM	0.05	0.001	0.09	0.986	0.157
tansig	tansig	8	9	LM	0.013	0.002	0.05	0.987	0.314
tansig	logsig	9	11	BR	1.11	0.98	3.14	0.621	153.86
tansig	radbas	9	12	BR	1.26	1.05	3.94	0.696	164.85
tansig	tansig	9	9	BR	1	0.95	0.92	0.723	149.15

Table 5 : Comparison of the performances of the four best fitted networks

Input Combination	**Hidden layers**	**Training Method**	**Testing**		
			MSE	$\mathbf{R^2}$	**SSE**
Vector 1	1	radbas(5) @LM *	0.96	0.853	150.7
	2	tansig(5) logsig(8) @LM **	0.53	0.957	83.21
Vector 2	1	logsig(7) @LM	0.69	0.886	108.33
	2	tansig(8)-tansig (9) @LM	0.05	0.987	0.314

Input vector interaction effect on ANN architecture

To determine whether the input vector and ANN architecture (e.g., learning algorithm, transfer function) have a significant effect on network performance, a two-way MANOVA was used. An experiment was thus conducted in which input vector 1 and input vector 2 were exposed to a combination of learning methods and transfer functions. The performance data were generated using ten-fold cross validation. The dataset was randomly divided into ten parts. Each part was held out in turn, and the network was trained on the remaining nine-tenths. Then, its performance indexes (MSE and R^2) were calculated on the holdout set. The network was executed a total of ten times on different training sets. Finally the ten performance indexes were averaged to yield a performance estimate. A two-way MANOVA was performed by SPPS for a one-layer and two-layer ANN, respectively. The overall conclusions are outlined below.

- In both networks (one-layer ANN and two-layer ANN), the multivariate effect of the ANN architecture was significant ($P< 0.05$). Thus, the ANN architectures differed with respect to ANN performance indexes.
- In both networks (one-layer ANN and two-layer ANN), the multivariate effect of the input vector was also significant ($P< 0.05$). Therefore, the input vectors differed with respect to ANN performance indexes.
- In both networks (one-layer ANN and two-layer ANN), the F-ratio (26.73) indicated that the interaction effect of the input vector and network architecture was statistically significant at an alpha 0.05. Therefore, the architecture performance was a function of the input vector, and input vector changes engendered significant differences in ANN performance with particular architectures. Accordingly, in an urban catchment in which the hydrological process is complex and data are not sufficient, the runoff estimation requires simultaneous examination and comparison of a diverse range of input vectors and ANN architectures.

Applicability analysis

The proposed ANN model developed in this study was verified and the model performance under different conditions of rainfall and vegetation were evaluated in the area study. The verification of the ANN model was performed by comparing the ANN model results to the observed runoff and SWMM simulation results. To determine whether there were significant differences among the results, a one-way MANOVA was carried out. The area study was composed of streets and highways, apartments (with less than 10% vegetation), houses (with 10% to 15% vegetation), and greenbelts (with 75% vegetation). The rainfall type was classified as rainfall with two-, five-, and ten-year return periods.

For this purpose, an experiment was designed in which nine rainfall-runoff events were divided into three groups according to three measurement models (ANN, SWMM, observed). To investigate the performance of the proposed model in different rainfall situations, we selected the subjects in accordance with three types of rainfall period returns (two, five, and ten years). The model outputs were measured by four response variables, y_1, y_2, y_3, y_4, where y_i is the runoff volume (cm^3/h) pertaining to the four types of catchment vegetation. Table 6 lists the values of the four dependent variables in each of the cells.

Table 6 : Comparison of ANN model and SWMM results and observed runoff in different types of urban catchments and rainfall return periods

Rainfall Return Period	Mean of dependent variables (cm^3/h)	Model: ANN	Model: SWMM	Observed
2 Year	y_1	7.96	7.7	6.5
	y_2	168	164.2	132
	y_3	90.8	89.76	63.7
	y_4	29.5	30.9	19.15
5 Year	y_1	11.84	11.2	9.2
	y_2	263	239	233
	y_3	144	156.1	99.1
	y_4	42	45	26.05
10 Year	y_1	14.2	14	9.8
	y_2	294.3	299.3	245
	y_3	165.3	167.3	115
	y_4	51.3	56.2	31.5

The one-way MANOVA analysis was performed by SPSS. The results are illustrated in Table 7. As shown in the table, none of the outcome variables is statistically significant at the 0.05 level of alpha. Therefore, we can conclude that no statistically significant difference exists between the value of runoff estimated by the ANN model, the SWMM model, and the one observed in the catchments. As the experiment was performed in various vegetation environments and rainfall periods of return, the MANOVA result suggests the responsiveness and applicability of the ANN model in a real-life scenario.

Table 7 : Multivariate Tests

	Value	F	Hypothesis df.	Error df.	Sig.
Pillai's Trace	1.102	1.228	8.000	8.000	.389
Wilks' Lambda	.055	2.450	8.000	6.000	.146
Hotelling's Trace	14.345	3.586	8.000	4.000	.116
Roy's Largest Root	14.143	14.143	4.000	4.000	.013

Conclusion

In this study, various ANN architectures were examined to explore the best topology for runoff estimation in an urban catchment. The proposed topology comprises these characteristics: two hidden layers, eight neurons in the first layer, nine neurons in the second layer, the same activation function of tansig in both layers, and the LM training algorithm. The result from one-way MANOVA indicated that the proposed architecture can estimate runoff for different types of urban vegetation and rainfall intensities. A comparison of the runoff values generated by the proposed ANN model with those of SWMM showed no statistically significant differences between them.

The results of this research support the application of ANN as a suitable alternative for physical-based models of runoff estimation. Particularly, in urban catchments where data are insufficient and hydrological processes are complex, the application of ANN is suitable. However, the ANN performance in urban catchments is the function of the input vector and network architecture. The results of a two-way MANOVA implied the significant effect of the ANN architecture and the input vector on ANN performance. Moreover, the interaction effect of the ANN architecture and input vector was additionally significant.

These findings demonstrate the importance of input variables in ANN-based modeling of runoff estimation in urban catchments. Accordingly, a methodology is required to explore and select the best variables affecting the input vectors. The methodology developed in this study is based on an existing physical equation of the hydrological process. In future research, it is suggested to apply multivariate statistical techniques, such as exploratory factor analysis and structural equation models. These techniques will contribute to exploration of unobservable constructs and to create an input vector that will foster a more accurate ANN model.

References

Abrahart, R.J.; See, L. and Kneale, P.E. (2001). Investigating the role of saliency analysis with a neural network rainfall-runoff model. Computers & Geosciences 27(8): 921-928.

Affandi, A. and Watanabe, K. (2008). Analysis of groundwater level fluctuation in a plain area using genetic algorithms and an artificial neural network. Lowland technology international: the officisl journal of the International Association of Lowland Technology (IALT)/Institute of Lowland Technology, Saga University 10(2): 76-85.

Ahmad, S. and Simonovic, S.P. (2001). Developing runoff hydrograph using artificial neural networks. Bridging the Gap: Meeting the World's Water and Environmental Resources Challenges.

Back, A.D. and Trappenberg, T.P. (1999). Input variable selection using independent component analysis. Neural Networks, 1999. IJCNN'99. International Joint Conference on, IEEE.

Bertrand-Krajewski, J.-L. (2007). Stormwater pollutant loads modelling: epistemological aspects and case studies on the influence of field data sets on calibration and verification. Water Science and Technology 55(4): 1-17.

Bowden, G.J.; G.C. Dandy and Maier, H.R. (2005). Input determination for neural network models in water resources applications. Part 1-background and methodology. Journal of Hydrology 301(1-4): 75-92.

Braddock, R.D.; Kremmer, M.L. and Sanzogni, L. (1998). Feed-forward artificial neural network model for forecasting rainfall run-off. Environmetrics: The official journal of the International Environmetrics Society, 9(4): 419-432.

Chae, Y.T.; Horesh, R.; Hwang, Y. and Lee, Y.M. (2016). Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. Energy and Buildings 111: 184-194.

Chakraborty, K.; Mehrotra, K.; Mohan, C.K. and Ranka, S. (1992). Forecasting the behavior of multivariate time series using neural networks. Neural networks 5(6): 961-970.

Chavoshi, S.; Sulaiman, W.N.A.; Saghafian, B.; Sulaiman, M.N.B. and Manaf, L.A. (2013). Flood prediction in southern strip of Caspian Sea watershed. Water Resources 40(6): 593-605.

Chen, J. and Adams, B.J. (2007). Development of analytical models for estimation of urban stormwater runoff. Journal of Hydrology 336(3-4): 458-469.

Chiang, Y.-M.; Chang, L.-C. and Chang, F.-J. (2004). Comparison of static-feedforward and dynamic-feedback neural networks for rainfall–runoff modeling. Journal of hydrology 290(3-4): 297-311.

Dogan, A.; Demirpence, H. and Cobaner, M. (2008). Prediction of groundwater levels from lake levels and climate data using ANN approach. Water Sa 34(2): 199-208.

Elshorbagy, A.; Simonovic, S. and Panu, U. (2000). Performance evaluation of artificial neural networks for runoff prediction. Journal of Hydrologic Engineering 5(4): 424-427.

Fausett, L.V. (1994). Fundamentals of neural networks: architectures, algorithms, and applications, Prentice-Hall Englewood Cliffs.

Freni, G.; Mannina, G. and Viviani, G. (2009). Urban runoff modelling uncertainty: Comparison among Bayesian and pseudo-Bayesian methods. Environmental modelling & software 24(9): 1100-1111.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. Neural networks 2(3): 183-192.

Haykin, S.S.; Haykin, S.S.; Haykin, S.S. and Haykin, S.S. (2009). Neural networks and learning machines, Pearson Upper Saddle River.

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

Karayiannis, N. and Venetsanopoulos, A.N. (2013). Artificial neural networks: learning algorithms, performance evaluation, and applications, Springer Science & Business Media.

Khayatian, F. and Sarto, L. (2016). Application of neural networks for evaluating energy performance certificates of residential buildings. Energy and Buildings, 125: 45-54.

Kumar, D.S.; Arya, D. and Vojinovic, Z. (2013). Modeling of urban growth dynamics and its impact on surface runoff characteristics. Computers, Environment and Urban Systems 41: 124-135.

Lee, S.; Lin, H. and Yang, T. (2010). Artificial neural network analysis for reliability prediction of regional runoff utilization. Environmental monitoring and assessment 161(1-4): 315-326.

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

Najafi-Marghmaleki, A.; Khosravi-Nikou, M.R. and Barati-Harooni, A. (2016). A new model for prediction of binary mixture of ionic liquids+ water density using artificial neural network. Journal of Molecular Liquids 220: 232-237.

Orimi, M.G.; Farid, A.; Amiri, R. and Imani, K. (2015). Cprecip parameter for checking snow entry for forecasting weekly discharge of the Haraz River flow by artificial neural network. Water resources 42(5): 607-615.

Patuwo, E.; Hu, M.Y. and Hung, M.S. (1993). Two-group classification using neural networks. Decision Sciences 24(4): 825-845.

Pulido-Calvo, I. and Portela, M.M. (2007). Application of neural approaches to one-step daily flow forecasting in Portuguese watersheds. Journal of Hydrology 332(1-2): 1-15.

Reddy, J.M.; Babu, A.S. and Mallikarjuna, C. "Rainfall–Runoff Modeling: Comparison and Combination of Simple Time-Series, Linear Autoregressive and Artificial Neural Network Models."

Ruck, D.W.; Rogers, S.K.; Kabrisky, M.; Oxley, M.E. and Suter, B.W. (1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. IEEE Transactions on Neural Networks 1(4): 296-298.

Sethi, R.R.; Kumar, A.; Sharma, S. and Verma, H. (2010). Prediction of water table depth in a hard rock basin by using artificial neural network. International Journal of Water Resources and Environmental Engineering 4(2): 95-102.

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

Sreekanth, P.; Sreedevi, P.; Ahmed, S. and Geethanjali, N. (2011). Comparison of FFNN and ANFIS models for estimating groundwater level. Environmental Earth Sciences 62(6): 1301-1310.

Wang, Y.; Traore, S. and Kerh, T. (2008). Feed forward backpropagation algorithm for estimating reference evapotranspiration in Burkina Faso. Proceedings of the 12th WSEAS International Conference on Computers. World Scientific and Engineering Academy and Society, Stevens Point, WI.

Whitford, V.; Ennos, A.R. and Handley, J.F. (2001). City form and natural process-indicators for the ecological performance of urban areas and their application to Merseyside, UK. Landscape and urban planning 57(2): 91-103.

Yonaba, H.; Anctil, F. and Fortin, V. (2010). Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. Journal of Hydrologic Engineering 15(4): 275-283.

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

Zhang, B.; Xie, G.; Zhang, C. and Zhang, J. (2012). The economic benefits of rainwater-runoff reduction by urban green spaces: A case study in Beijing, China. Journal of environmental management 100: 65-70.