Title

Improved streamflow forecasting using self-organizing radial basis function artificial neural networks

Permalink

https://escholarship.org/uc/item/2t73t8mc

Journal

Journal of Hydrology, 295(1-4)

ISSN

0022-1694

Authors

Moradkhani, Hamid
Hsu, Kuo-lin
Gupta, Hoshin V
et al.

Publication Date

2004-08-01

DOI

10.1016/j.jhydrol.2004.03.027

Peer reviewed

ELSEVIER

Improved streamflow forecasting using self-organizing radial basis function artificial neural networks

Hamid Moradkhani[*,1], Kuo-lin Hsu, Hoshin V. Gupta, Soroosh Sorooshian

Department of Hydrology and Water Resources, University of Arizona, Tucson, AZ 85721, USA

Abstract

Streamflow forecasting has always been a challenging task for water resources engineers and managers and a major component of water resources system control. In this study, we explore the applicability of a Self Organizing Radial Basis (SORB) function to one-step ahead forecasting of daily streamflow. SORB uses a Gaussian Radial Basis Function architecture in conjunction with the Self-Organizing Feature Map (SOFM) used in data classification. SORB outperforms the two other ANN algorithms, the well known Multi-layer Feedforward Network (MFN) and Self-Organizing Linear Output map (SOLO) neural network for simulation of daily streamflow in the semi-arid Salt River basin. The applicability of the linear regression model was also investigated and concluded that the regression model is not reliable for this study. To generalize the model and derive a robust parameter set, cross-validation is applied and its outcome is compared with the split sample test. Cross-validation justifies the validity of the nonlinear relationship set up between input and output data.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Streamflow forecasting; Neural network; Radial basis function; Self-organizing feature map; Cross-validation

1. Introduction

Rainfall-runoff (or more generally speaking, precipitation-runoff) modeling is a major focus of hydrological modeling. In particular, streamflow forecasting is of significant importance for planning and operational purposes. A large variety of models have been proposed with the hope of getting more accurate and reliable forecast. As McCuen (1997) pointed out, due to the complex nature of hydrological processes, there is no integrated theory of hydrology. Numerous assumptions and approximations are made to reduce the complexity of models.

There is a highly nonlinear and complex relationship between precipitation and runoff due to temporal and spatial variability of watershed characteristics, heterogeneity in precipitation, as well as numerous factors involved in generating runoff. Among the components involved in transforming precipitation to runoff, the dominant ones are often evaporation, infiltration, interception, soil moisture, overland flow, land use, and geomorphology of watersheds. Conceptual hydrologic models as the abstraction, representation, and ordering of the hydrologic phenomena are being typically used for solving nonlinear problems (Burrough and McDonnell, 1998). In contrast to

---

* Corresponding author. Tel.: +1-949-824-8821; fax: +1-949-884-8831.

E-mail address: moradkha@uci.edu (H. Moradkhani).

[1] Now at University of California Irvine, Civil & Environmental Engineering, E/4130 Engineering gateway, Irvine, CA 92697-2175, USA.

physically based models that employ differential equations of continuity and energy, conceptual models are built upon a base of knowledge of physical, chemical, and biological processes that act on the input to produce the output (US Army Corps of Engineers, 2000). An alternative modeling approach for streamflow forecasting is the empirical model, built upon the observations of the input and output. An example of the latter modeling approach is multivariate regression analysis, used by many researchers for annual flow forecasting (Wong, 1979; Kothyari and Garde, 1991; Swamee et al., 1995). The major concern in empirical models is the data rather than the physical process, i.e. the model learns from data and predicts the future. Empirical Artificial Neural Networks (ANNs) have been applied to solve a variety of nonlinear problems during the latest decade. The establishment of ANNs can be traced back nearly a century (Anderson and Rosenfeld, 1988). ANNs are a class of computational tools that operate approximately analogously to the biological processes of a brain. A more comprehensive definition is given by Haykin (1994) as a massively parallel distributed processor that has a natural tendency for strong experimental knowledge. Neural networks learn from experience and then perform 'recognition without definition' (Kosko, 1992).

A comprehensive review of the applications of ANNs in hydrology was presented by the ASCE task committee on the application of ANNs in hydrology (2000a,b). In the two-part series, the authors investigated the role of ANNs in various fields of hydrology, their robustness, merits, limitation, and in particular, potential research paths. Hsu et al. (1995) introduced a procedure, entitled linear least squares simplex, for identifying the structure and parameters of MFN models and demonstrated the potential of such models for simulating the nonlinear hydrologic behavior of watersheds. The structural components of MFN models have been explained in detail by Hsu et al. (1995). Sezin and Johnson (1999) employed ANN to forecast daily runoff as a function of daily precipitation, temperature, and snowmelt for a watershed in Maryland. They compared the model with a statistical regression technique and a simple conceptual model and concluded the superior performance of ANN models. Thirumalaiah and Deo (1998) emphasized a number of advantages of a neural network in river

stage forecasting. The back propagation NN was applied by Sezin and Markus (2000) in three basins with different climate and physiographic characteristics to model watershed runoff processes and was compared to a conceptual water balance (Wetbal) model. They also used the ANN to model daily rainfall-runoff processes and compared them with the Sacramento Soil Moisture Accounting (SAC-SMA) model. They showed that the performance of the ANN in modeling the precipitation-runoff process for various time scales, topography, and climate patterns was encouraging. The application of the ANNs in daily streamflow forecasting up to 5 days ahead was investigated by Birikundavyi et al. (2002). ANN provided the superior performance when compared with both deterministic and stochastic models. Chang and Chen (2003) presented a hybrid ANN including the fuzzy clustering scheme along with Radial Basis Functions for water stage forecasting in an estuary under high flood effects. They showed that ANN could be a powerful tool for solving such a poorly defined and complex problem.

The above-mentioned capabilities of ANN models suggest the usefulness of empirical models that avoid the complexity of conceptual models, while being well-suited in practice. In this study, a combination of two ANN architectures is considered into which one classifies the input data and, using the characteristics of classified inputs, namely center and standard deviation, the inputs are transferred through the radial basis functions to forecast one-day ahead streamflow. The classification is done using an unsupervised training method, called Self Organizing Feature Map (SOFM) (Kohonen, 1989). This scheme was inspired by Self Organizing Linear Output map (SOLO) proposed by Hsu et al. (2002) and also the classification procedures by Govindraju and Zhang (2000). SOLO classifies the input information using a SOFM and then maps the inputs into the outputs using multivariate linear regression. The scope of this paper is organized as follows. The SORB model structure is described in Sections 2 and 3 followed by the model application and training strategy in Sections 4 and 5. To stabilize the model structure, and to achieve robust parameter estimates, the model complexity issue is addressed in Section 6 where cross-validation technique is employed as a solution to the potential problems of split sample validation. An application to

the Salt River as a sub regional watershed of the lower Colorado River basin in the United States is shown in Section 4.

In this paper we highlight the potential of a hybrid NN model for streamflow forecasting where comparison with the well-established architectures can justify the merit of the algorithm. This research is two-fold, (1) we report a more efficient and effective NN structure by combining two NN models in the streamflow forecasting, and discuss some technical aspects of the algorithm, namely clustering and tuning the spread parameter of the Gaussian functions. Although the possibility of employing SOFM as the clustering method has been reported in literature, the usage of it in such a combination has not been elaborated in hydrologic applications. (2) We derive a robust parameter set when short data sets are available; this is achieved by cross validation technique. The cross-validation enables model generalization while minimizing the sensitivity of the model to the split sample. Detailed discussion on cross-validation is given at Section 6.

2. RBF Neural Networks (RBFNs)

RBF neural networks (RBFNs) are a class of feedforward neural networks that are used for classification problems, function approximation, noisy interpolation, and regularization (Kégl et al., 2000). They have increasingly attracted interest for engineering applications due to their advantages over traditional multilayer perceptrons, namely faster convergence, smaller extrapolation errors, and higher reliability (Girosi, and Pogio, 1990). The neural networks suitable for the particular application here belong to the multilayer feed forward type that has the ability to approximate any continuous function; in this case by using radially symmetric basis functions such as the Gaussian function. The RBF technique provides good generalization ability with a minimum number of nodes to avoid unnecessarily lengthy calculations, in comparison with multilayer perceptron networks. The origin of the radial basis function approach can be traced to the work of Powell (1987), which showed that RBFs are highly promising for multivariable interpolation given irregularly positioned data points. To formulate the problem, consider a mapping

function f that maps an n-dimensional input or data space R^n to a 1D output or target space R, as follows:

$$f(\vec{x}_i) = y_i \ \forall i = 1, 2, ..., P, \tag{1}$$

Where each of the P known data points comprises an input vector $\vec{x}_i$ and a corresponding desired output y_i. Powell (1987) introduced a set of n basis functions, $\varphi_i(\|\vec{x} - \vec{x}_i\|) \ \forall i = 1, 2, ..., n$, which are continuous non-linear functions, where the ith RBF φ_i depends on the distance, (typically measured using an Euclidean norm), between any data point $\vec{x}$ and the ith known data point $\vec{x}_i$. Hence, the mapping function can be approximated as a linear combination of the RBFs φ_i with the unknown weights w_i :

$$f(\vec{x}) = \sum_{i=1}^{p} w_i . \varphi_i(\|\vec{x} - \vec{x}_i\|) \tag{2}$$

By inserting the interpolation function (2) in the mapping function (1), a set of linear equations result:

$$y_j = \sum_{i=1}^{p} w_i . \varphi_i(\|\vec{x}_j - \vec{x}_i\|) \ \forall j = 1, 2, \cdots, n, \tag{3}$$

In matrix notation, the above formulation can be written as:

$$\phi \vec{w} = \vec{y} \tag{4}$$

or:

$$\begin{bmatrix} \varphi_1(\|\vec{x}_1 - \vec{x}_1\|) & \cdots & \varphi_n(\|\vec{x}_1 - \vec{x}_n\|) \\ \vdots & \ddots & \vdots \\ \varphi_1(\|\vec{x}_p - \vec{x}_1\|) & \cdots & \varphi_n(\|\vec{x}_p - \vec{x}_n\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix} \tag{5}$$

By inversion of the matrix ϕ at (4), assuming that ϕ^{-1} exists and is nonsingular (Govindraju and Zhang, 2000), the weights for exact interpolation are found to be:

$$\vec{w} = \phi^{-1} \vec{y} \tag{6}$$

This procedure provides an exact interpolation function, which passes through all of the data points. There are several undesirable features of such a mapping, as pointed out by Govindraju and Zhang (2000), including incapability of the network to generalize the mapping at the forecasting stage and also overtraining problem due to the enormous

number of mappings and fitting of the data noise. To deal with the above-mentioned problems, a number of modifications have been suggested (Moody and Darken, 1989; Govindraju and Zhang, 2000):

1. The number of RBFs could be less than the number of data points.
2. Center of the RBFs are not restricted to the data points and they can be found through training.
3. Bias parameters are added in the linear sum of output layers to make the estimation unbiased.

Girosi and Poggio (1990) showed that RBFNs have the best approximation property, which does not hold for multi-layer perceptrons type of neural networks.

Fig. 1 shows the configuration of an RBF network with n_0 input, n hidden layer nodes, and one output layer node for general transformation of P points in input space to one point in output space. Unlike a general type of a MFN network, the connections between the input and hidden layer are not weighted. To describe the network mathematically, the Gaussian functions (RBFs) are used as transfer functions at the hidden nodes:

$$\varphi_j = \exp\left(-\frac{\|X^i - \mu_j\|^2}{2\sigma_j^2}\right) \tag{7}$$

where;

$X^i = n_0$ dimensional input vectors, $i = 1, 2 ... P$;
$\mu_j =$ mean (center), $j = 1, 2 ... n$;
$\sigma_j =$ standard deviation (spread), $j = 1, 2 ... n$;
$\varphi_j =$ basis function value
$n =$ number of hidden nodes

The linear mapping from hidden layer to output layer is given by:

$$Q_i = \sum_{k=1}^{n} \varphi_k w_k + w_0 \quad i = 1, 2, ..., P \tag{8}$$

where;

$Q_i =$ output values; in this study Streamflow on the next day corresponding to X^i input vector;
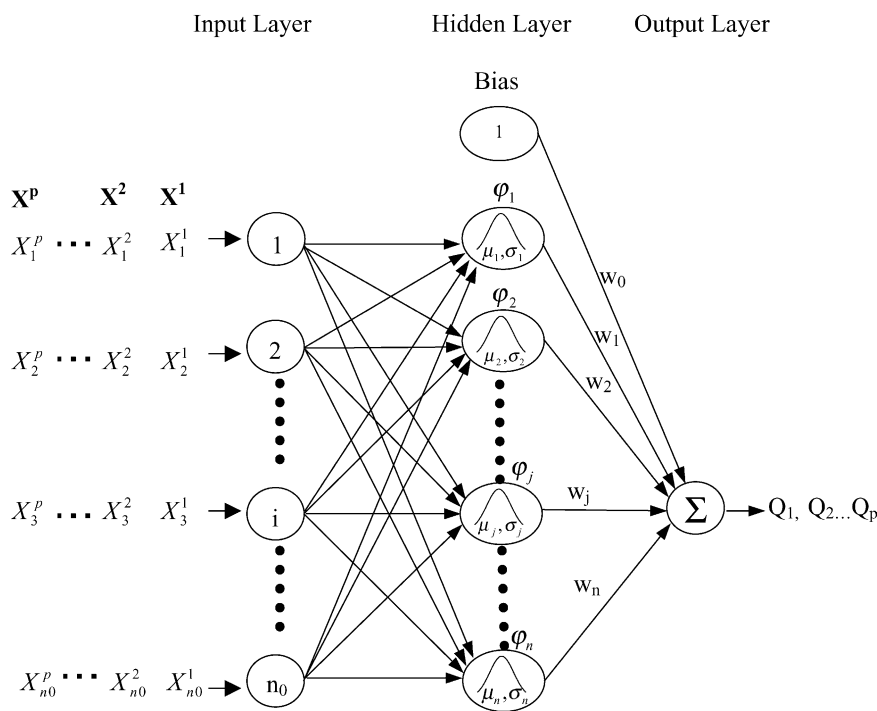$w_k =$ connection weights;
$w_0 =$ bias term.

Fig. 1. Configuration of RBFN.

Note that the Gaussian basis functions in (7) are not normalized to a probability distribution function, such as a normalizing factor of $1/\sigma\sqrt{2\pi}$ in a 1D normal distribution. The use of Gaussian basis functions requires estimation of the values for parameters μ and σ. Therefore, training of the network needs to be performed in two stages:

1. Calibration of parameters μ and σ; and
2. Calibration of connection weights, $\vec{W}$.

A distinct advantage of RBFNs over MFNs is the possibility of selecting appropriate parameters for the transfer functions at the hidden nodes, by estimation in advance without having to accomplish a full nonlinear optimization of the network. Several procedures to obtain these parameters have been reviewed by Bishop (1995); Govindraju and Zhang (2000); Chang and Chen (2003). These include the Random Selection of Centers (subsets of data points), Supervised Selection of Centers, Orthogonal Least Squares, Gaussian mixture models, and Clustering algorithms. Applications of the above methods can be seen in Jayawardena et al. (1998); Achela et al. (1998); Chen et al. (1991), and Moody and Darken (1989). In this study we employ an unsupervised procedure, the Self-Organizing Feature Map (SOFM), to extract the Gaussian function (RBF) parameters.

3. Self-organizing feature map (SOFM)

SOFMs, originally proposed by Kohonen (1989), are characteristically used for density estimation or for projecting patterns from high-dimensional to low-dimensional spaces, (most commonly 2D). SOFM is an unsupervised classification, used to cluster the data set based on statistics only, without any user-defined classes. It is a type of neural network designed to approximate the distribution of target patterns with a small number of weight vectors. They have the capability to adjust the weight vectors of adjacent units in the competitive layer to a similar vector by competitive learning and to approximate the distribution of the target patterns using total weight vectors acquired as the result. A competitive layer of neurons, arranged in a lattice, is connected to all the inputs via adjustable weights. The input-hidden layer therefore identifies similar patterns and groups them into clusters. Fig. 2 displays the SOFM network architecture. The major difference between SOFM and classical
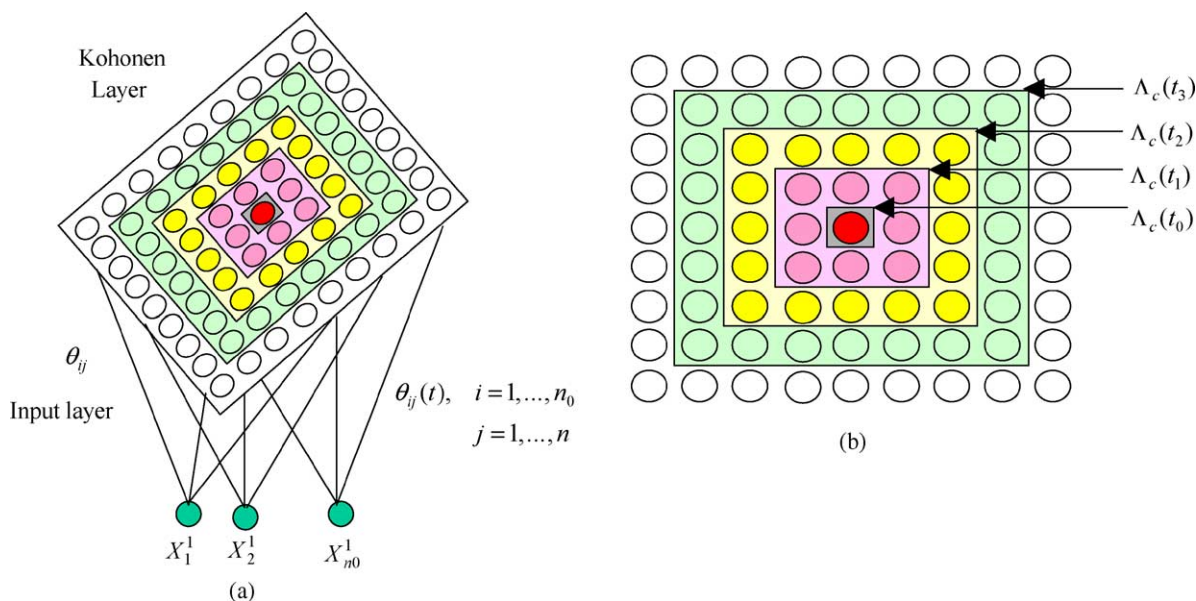

Fig. 2. (a) A 2D Self-Organizing Feature Map (SOFM), (b) Size of training neighborhood and training iteration.

pattern recognition techniques is that SOFM provides a graphical organization of pattern relationships and close estimates of the underlying probability density function. Haykin (1994) summarized the unsupervised training of connection weights in SOFM, as follows:

1. Initialize randomly the weight vectors for each SOFM connection weight:

$$\theta_{ij}(0), \ i = 1,...,n_0 \ j = 1,...,n \tag{9}$$

2. Compute the winner unit at iteration t based on minimum distance, typically Euclidian distance, of sample x from the input vectors. In other words the competitive layer unit, which satisfies the following equation becomes the winner unit:

$$d = \min \|X^P - \Theta\| = \min \left(\sum_{j=1}^{n0} (x_i(p) - \theta_{ij})^2 \right)^{0.5} \tag{10}$$

3. Adjust the connection weights vectors of all neurons:

$$\theta_{ij}(t) = \theta_{ij}(t-1) + \eta(t) \Big[x_i(p) - \theta_{ij}(t) \Big] \ \text{if} \ i \in \Lambda_c(t) \tag{11}$$

$$\theta_{ij}(t) = \theta_{ij}(t-1)$$

Otherwise

$$\eta(t) = \eta_0 \left(1 - \frac{t}{T} \right) \tag{12}$$

where t is the current iteration of learning, T is the total number of learning iterations, $\eta(t)$ is the learning rate, and $\Lambda_c(t)$ defines the size of a neighborhood around the winner unit c. The value of $\eta(t)$ decreases from an initial value η_0 as learning progresses, finally approaching 0.0. Larger values are given in the initial setting for $\eta(t)$ and $\Lambda_c(t)$ which are reduced gradually while iteration t is increased, $\eta_0 = 0.2-0.5$ and $\Lambda_c(0) = n/2$ (Hsu et al., 1999; 2000).

The connection weights obtained by SOFM are representative points in the input space; in other words, they can be regarded as centers, μ, of the Gaussian functions. Spread parameters, σ, can be computed indirectly for each cluster based on the density of the points that surrounds the centers by calculating the distance of all input points from the cluster centers and finding the points belonging to each cluster by minimizing the distance. By computing the standard deviation of the points in each cluster, the initial values of the spread parameters σ can be estimated. Optimization of the spread parameter will be discussed more in a later section.

4. Model application

The SORB was used to develop a one-step ahead daily flow forecast model for the Salt River, a sub watershed of the lower Colorado River basin. The Salt River has special characteristics in the southwestern United States, including dense forests to the east and the dry desert valley of Phoenix to the west. The basin is located in central Arizona, and covers an area of approximately 10,000 km². Two wet seasons govern precipitation throughout the basin. In the winter (January through March), frontal storms from the Pacific Ocean dominate the landscape. The heat of the hot summer days and the moisture coming from the Gulf of Mexico control the other wet season from July through September. These widespread storms distribute precipitation, often in the form of snow in the higher elevations. The Salt River flows into the Roosevelt reservoir system (Fig. 3); therefore, timely and accurate forecasts of daily river flows result in significant operational benefits. Precipitation, stream-flow, and temperature data were available for the period of 1989–1998. The precipitation-monitoring platform utilized in this study is the precipitation gauge network. Because the model under consideration is lumped, the watershed is regarded as one unit; thus the variables and parameters represent average values for the entire watershed. Accordingly, the mean-areal precipitation was computed by the Thiessen method and considered as one of the input variables at the Roosevelt reservoir. The daily average temperature and the streamflow were also used as the other input variables in the model.

An important aspect of ANN modeling is to establish a meaningful relationship between the input vector and output variable. The autocorrelation of streamflow and the cross-correlation of precipitation-streamflow and temperature-streamflow were performed for two combined seasons, winter-spring and summer-fall, to explore the time dependence among the variables. Due to the complex nature of
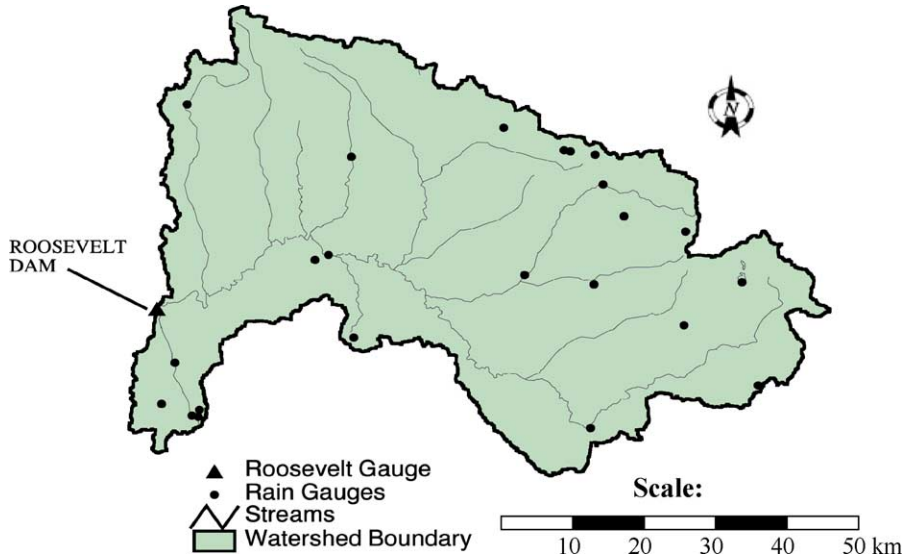
Fig. 3. Salt River Watershed.

the precipitation-runoff relationship result from the combined effects of rainfall and snow, we included terms to account for both the short-term and long-term effects of precipitation and temperature on streamflow. Therefore, a qualitative assessment of the correlation analysis encourages to establish the relation in Eq. (13).

$$Q_t = f(Q_{t-1}, Q_{t-2}, Q_{t-3}, p_{t-1}, p_{t-2}, \bar{p}_{t_{5-14}}, \bar{p}_{t_{30-39}}, \bar{T}_{t_{1-5}})$$
(13)

where: $Q_{t-1}, Q_{t-2}, Q_{t-3}$: streamflow at one, two, and three days ago, respectively;

p_{t-1}, p_{t-2} : precipitation at one and two days ago, respectively;

$\bar{p}_{t_{5-14}}$: average precipitation in the period of $5-14$ days in the past, similarly $\bar{p}_{t_{30-39}}$ is the average precipitation in the period of $30-39$ days in the past;

$\bar{T}_{t_{1-5}}$: average temperature in the period of $1–5$ days in the past.

5. Training and testing

Years 1990 and 1991 were used as test data set to evaluate the performance of the model in a moderate climate condition. The remaining data were used for training (calibration). Training of the network consists of two parts: finding the parameters of the RBFs using the SOFM clustering algorithm, and optimizing the

connection weights between the hidden and output layers. Training of the SOFM was illustrated in Section 3. Fig. 4 displays the clustering of the input space in a 2D problem using SOFM. Circles with
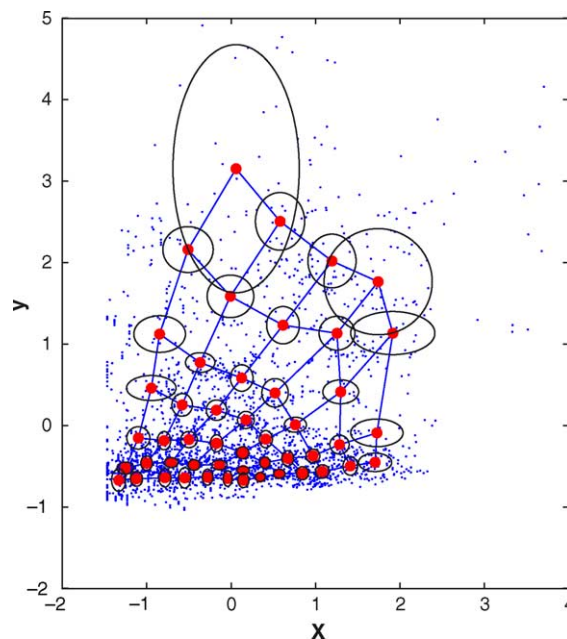

Fig. 4. Clustering of 2D input data using SOFM. Solid dots display the cluster centers surrounded by circles with radius of average standard deviation.

the radius of average standard deviation of the points belonging to each cluster have been drawn around each cluster center. It can be seen that the accuracy of the simulated streamflow changes with the spread parameter (aforementioned standard deviation) of the RBFs. In fact, SOFM determines the location of representatives in the input space (cluster centers), which are used as the parameters μ of the RBFs. The standard deviation calculated above should be regarded as an initial guess and is to be tuned in the calibration phase. To do this, the multiplier parameter, β, is considered at RBFs as follows:

$$\varphi_j = \exp\left(-\frac{\|X^i - \mu_j\|^2}{2\beta\sigma_j^2}\right), \tag{14}$$

The best value of β is estimated such that it minimizes root mean square error (RMSE) of the training as stated below:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{N} (q_t^{\text{sim}} - q_t^{\text{obs}})^2}{N}} \tag{15}$$

where q_t^{sim} = simulated flow (daily), q_t^{obs} = observed flow (daily), and N = total number of daily streamflow values.

The function of parameter β is to shrink or expand the extent of Gaussian function, which accordingly alters the contribution of the hidden nodes in forecasting stage. It was found that training tend to result in values of $\beta < 1$. This results in more clusters contributing in the regression part of the network.

To calibrate the nodal regression parameters (connection weights) at (8), the least square method is employed. If (8) is written in a matrix form, we have:

$$Q = \Phi W \tag{16}$$

In general, if the inverse of Φ exists, the parameter vector W can be found by $W = \Phi^{-1}Q$, and the error associated with this estimation would be equal to zero. Owing to the indeterminacy of the problem at hand, due to the number of equations (the number of input vectors) exceeding the number of unknown parameters (m), matrix Φ cannot be inverted. In this case, it is possible to calculate a so-called pseudoinverse solution:

$$\hat{W} = (\Phi^T\Phi)^{-1}\Phi^T Q \tag{17}$$

The errors associated with this estimation would be the minimized total squared error in the following:

$$\text{Min}\,(\varepsilon_Q^T\varepsilon_Q) = (Q - \Phi\hat{W})^T(Q - \Phi\hat{W}) \tag{18}$$

Sometimes, due to the presence of correlation among input variables, φ_i, the matrix Φ may become colinear, causing $(\Phi^T\Phi)^{-1}$ to be singular (Hsu et al., 2002). Ill-conditioning was also reported by Mason et al. (1996) while they used a large model having too many centers. To avoid this problem, orthogonal transformation can be applied to the matrix Φ to obtain a matrix with independent components (Haykin, 1994; Hsu et al., 2002).

Figs. 5 and 6 display the comparison between the performance of the SORB model with the MFN, SOLO and LINREG in training and testing, respectively. Lower RMSE in SORB; specially in testing period, comparing to other models demonstrates the superior capability of SORB for forecasting purposes in Salt River basin. A more detailed evaluation of model performance is given in Figs. 7 and 8. The RMSE values (m³/s) plotted against the volume of seasonal (3-month) streamflow for each model in both training and testing periods are shown in Fig. 7. As seen, RMSE increases rather linearly with the magnitude of flow for all of the models and to a better extent in the SORB model, especially in the low flows.

Plots of correlation coefficients between observed and estimated streamflow with respect to the magnitude of streamflow are displayed in Fig. 8. Because less than 1% of the observed streamflow is greater than 400 (m³/s), a more realistic estimate of the correlation between observed and estimated streamflow could be obtained for those values which are less than 400 (m³/s). Correlation was calculated among those observed and estimated values, which are less than or equal to a certain magnitude. For instance, the correlation of observation and estimation in training for those streamflows that are less than or equal to 200 (m³/s) is 0.9. Therefore, the correlation values shown in Figs. 6 and 7 have been influenced by a few high flows, which is not a fair evaluation of model in terms of this performance measure. This also happens in the calculation of RMSE and, in order

to have a rational evaluation of the model performance for this specific data set, one might prefer to exclude those few high flows from the performance measure computation.

To avoid the complexity of the ANN model, one may consider the applicability of linear regression model. This was investigated and seen that performance measures were showing poor results from which RMSE of training was 69.4 noticeably larger than the ANN models'. By checking the regression model over the testing period, RMSE of 66.2 was obtained comparable and similar to other models. This can be explained as inconsistency of model behavior by having poor performance

measure in training, but similar behavior with ANN models. The best performance among the models was obtained in SORB model with RMSE of 46.6 and 50.3 in training and testing, respectively.

6. Best parameter set derivation while having a short data set

If an ANN properly learns the essential features of the data, and can adapt itself with the new information it receives and correspondingly respond better, then, the ANN is said to achieve good generalization.
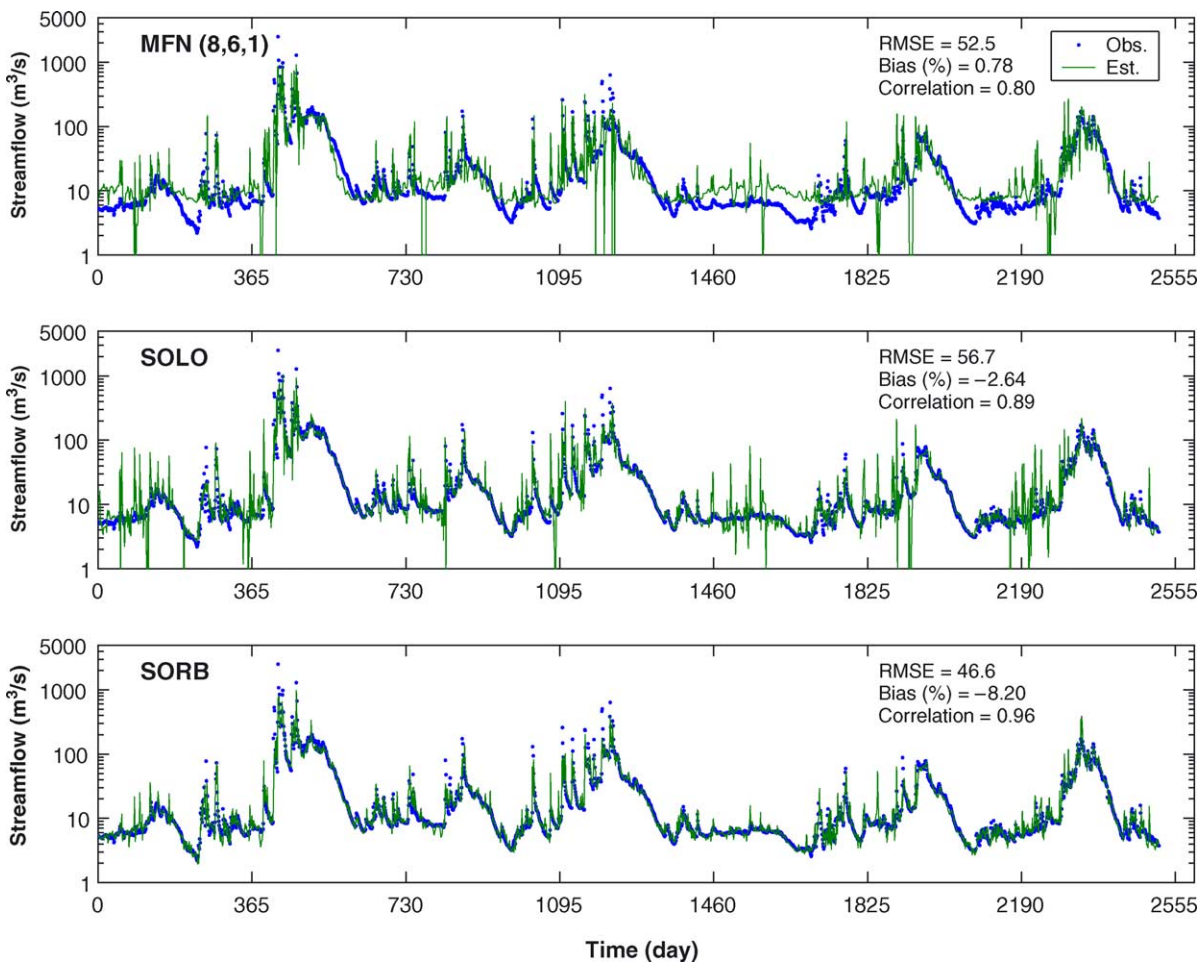

Fig. 5. The daily flow time series of training models over 7 years of data.
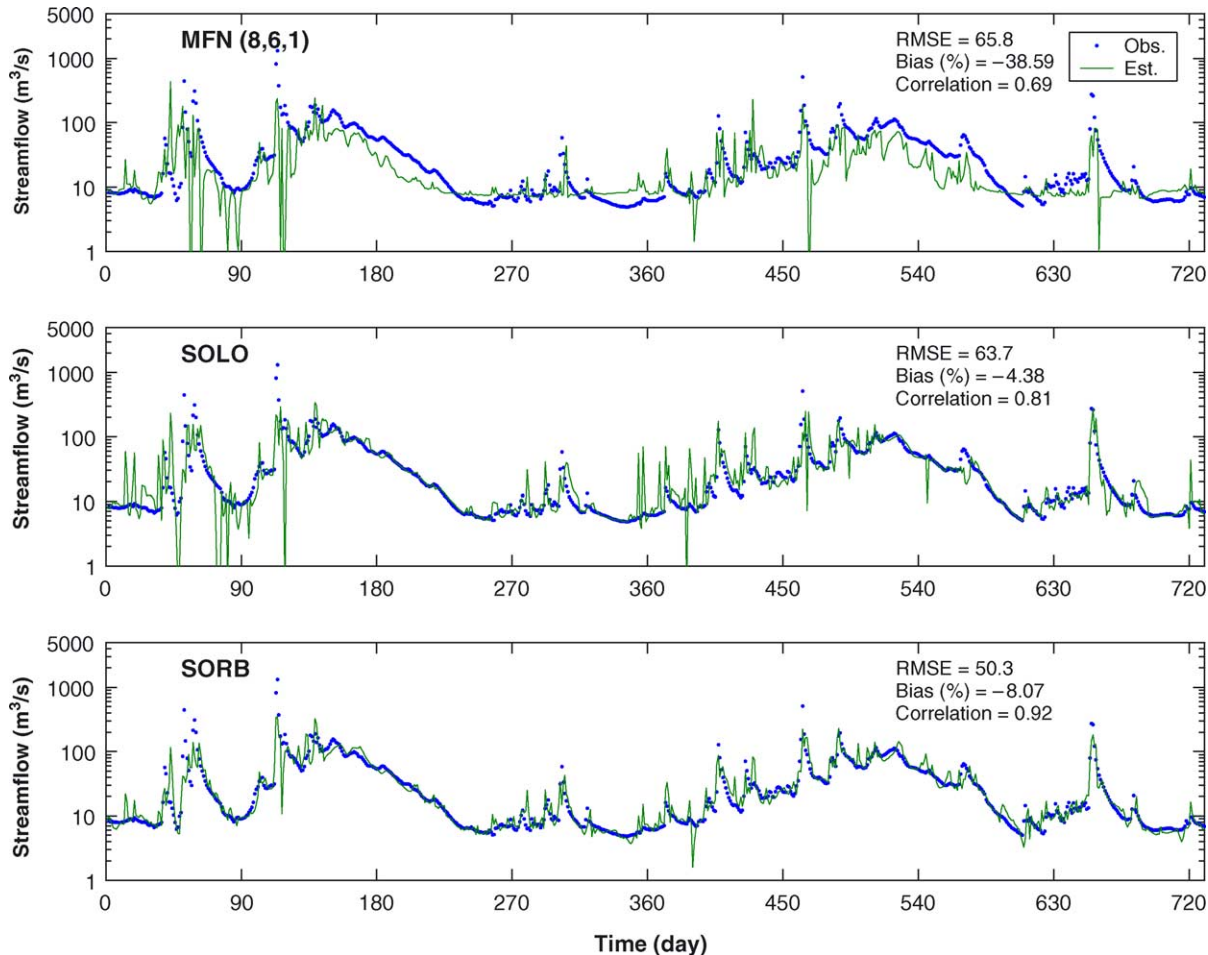
Fig. 6. The daily flow time series of testing models over 2 years of data (1990 and 1991).

In order to achieve the best generalization, which is to have the optimal performance in training and testing, the complexity of the model needs to be optimized. The model complexity can be measured in terms of the number of adaptive parameters, such as the number of hidden nodes, parameters of the transfer function, RBFs, and the training and testing data sets. Depending on the complexity of the network, however, an ANN can suffer from either overfitting or underfitting. Bias-variance trade-off and regularization are two of the techniques that are being utilized to stabilize the structure of the model (Haykin, 1994; Bishop, 1995). Another technique, which addresses the model generalization by calibrating over several data sets, is cross-validation. Cross-validation is a method for

estimating generalization error based on 'resampling' (Plutowski, et al., 1994; Efron and Tibshirani, 1997). The resulting estimates of generalization error are often used for choosing among various models, such as different network architectures. A few attempts have been made in the context of conceptual rainfall-runoff models to investigate the influence of the length of data on the model performance (Sorooshian et al., 1983; Yapo et al., 1996). In these studies, the authors tried to find the minimum length of data required for calibration (still as a split sample) in order to obtain a parameter set that is relatively insensitive to the period selected. The selection of a training data set is even more crucial in the case of ANNs, which are more data-dependent than a conceptual model.
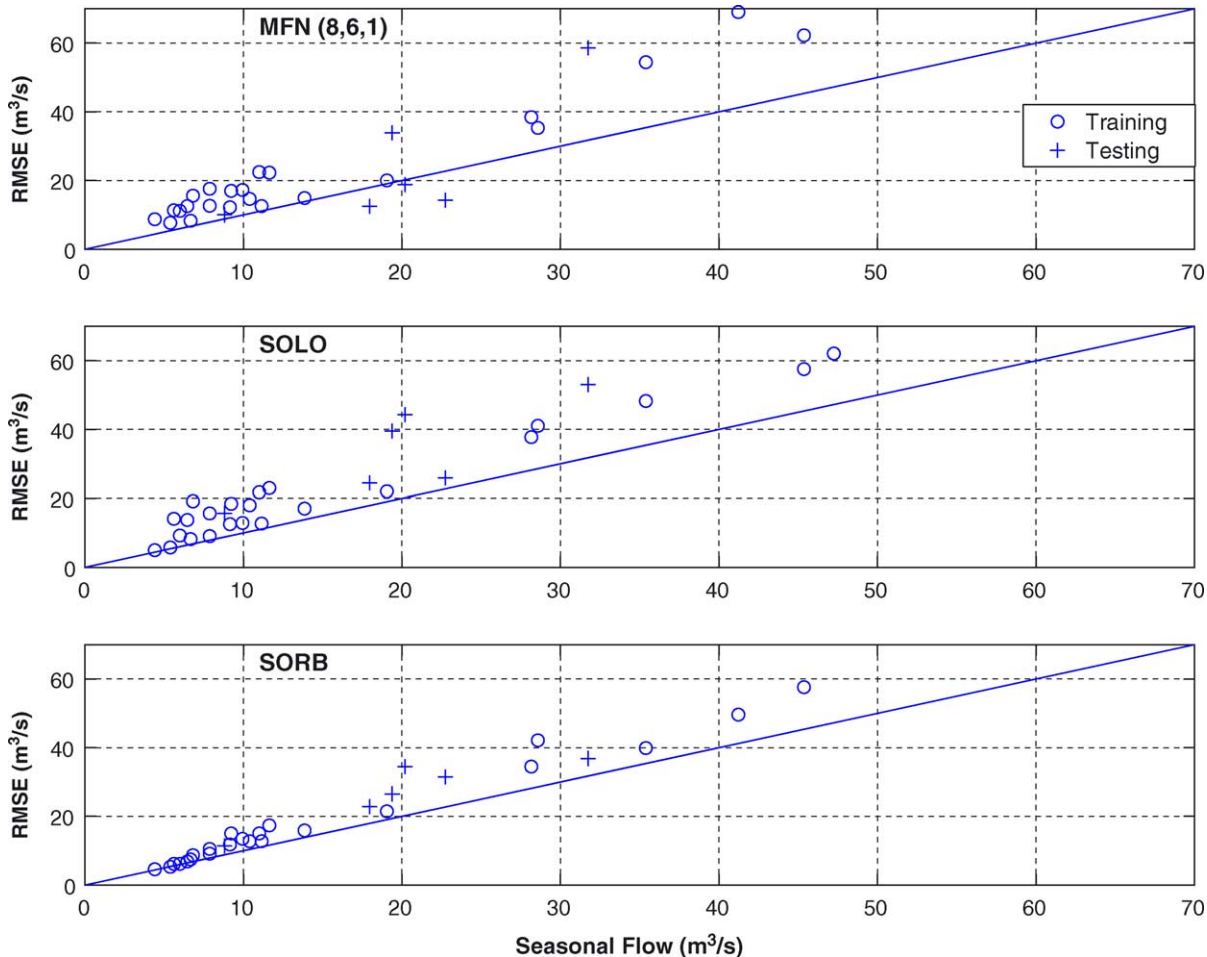
Fig. 7. The seasonal (3 months) RMSE with respect to the seasonal streamflow (m^3/s) for different models.

Anctill et al. (2003) investigated the performance of a conceptual rainfall-runoff and ANN model for different data lengths and concluded that longer training sets were more beneficial to the ANN model. In all the above studies, the model was not suffering from limited data and, by having a long data set, researchers were able to investigate the role of the data length in accurate parameterization while still using one realization (split sample) to calibrate the model. In the current study, a fairly short data set was available; therefore, employing a strategy to deduce a more identifiable parameter set seemed necessary. In the following, a technique called S-fold cross-validation is elaborated.

In S-fold cross-validation, the data is partitioned into S subsets of equal size and then the model is trained S times (Fig. 9). The first subset is the training set where the model parameters are found. The second subset is the validation set where the performance of the trained model is monitored and used for selection of the best parameter set; validation set is basically used to cover any tendency during training. The remaining data (testing set) is regarded as an independent data set and it is used to compare different models (Chang and Chen, 2003). Cross-validation is quite different from the split sample method that is commonly used for early stopping in ANNs. In the split sample method (like the training method in Section 5), only a single
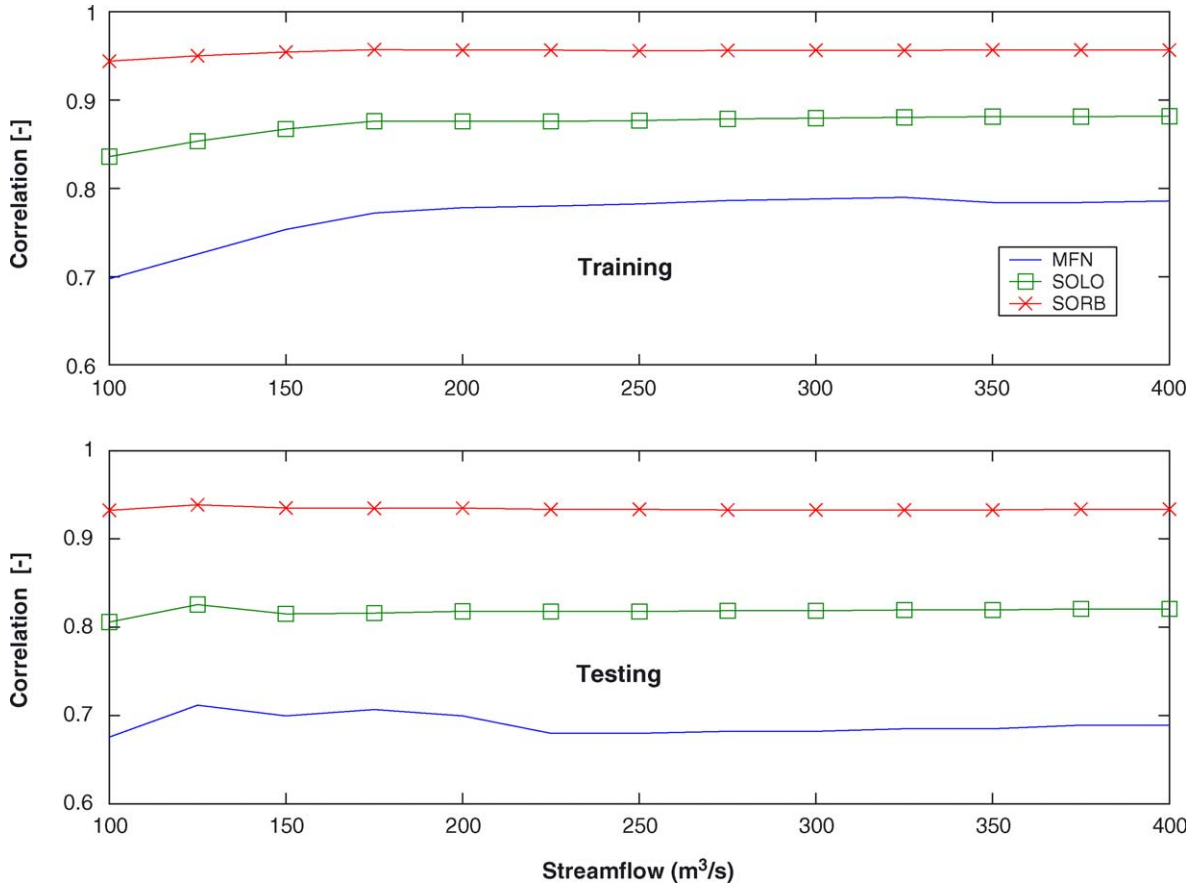
Fig. 8. Correlation of observed and estimated streamflow with respect to the magnitude of streamflow for different models.
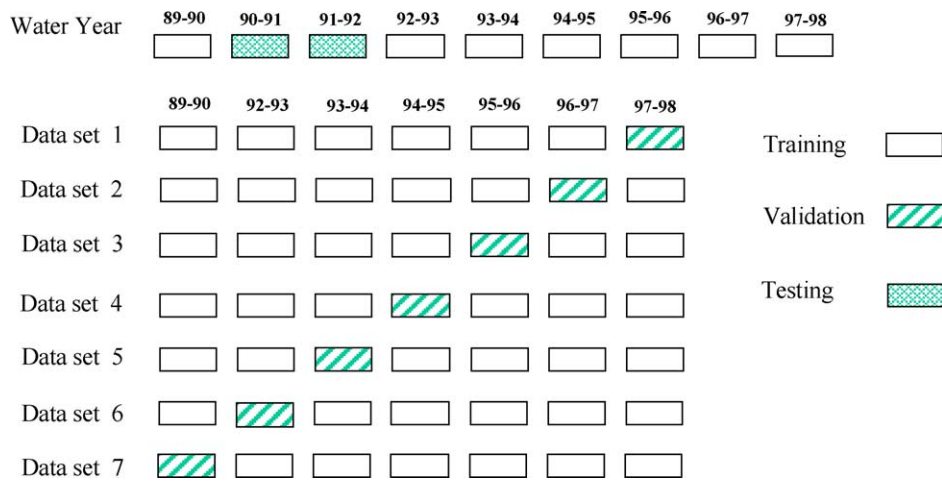

Fig. 9. 7-fold training and validation data sets.

subset (the testing set), instead of S different subsets, is used to evaluate the generalization error. Goutte (1997) demonstrated that S-fold cross-validation produces noticeably better results.

As explained in Section 5, a set of parameters, namely, center and standard deviation of each hidden node $(\vec{\mu}, \vec{\sigma})$, are extracted from SOFM and a second set including the standard deviation multiplier, β, and the connection weights, $\vec{W}$, between hidden and output layers are estimated via training of the RBF. In cross-validation, parameters are obtained by training the model over 7 training data sets and evaluated over the validation data sets. The simple linear transformation in the output layer of the RBF network can be optimized using a traditional linear modeling technique as elaborated in Eqs. (16)–(18); no iteration in optimizing the connection weights is needed. The main concern in the calibration process

therefore, is to find the optimum value for the multiplier β. As a conventional procedure to optimize the parameters via the training process, the RMSE over the training data set is minimized as follows:

$$\text{Min } RMSE_{trn}(\beta, \vec{W}) \tag{19}$$

In typical training, the RMSE generally decreases as a function of the number of iterations as seen in gradient descent algorithms. However, the RMSE with respect to the validation data set may decrease in the beginning and then start to increase when the model begins to overfit. To avoid overfitting, the early stopping method is used. Early stopping has been reported to be superior to regularization methods in many cases (Finnoff, 1993). In this procedure, the best parameter set is selected at an iteration number when the RMSE of the validation set starts to increase. In the current study, the connection weights are
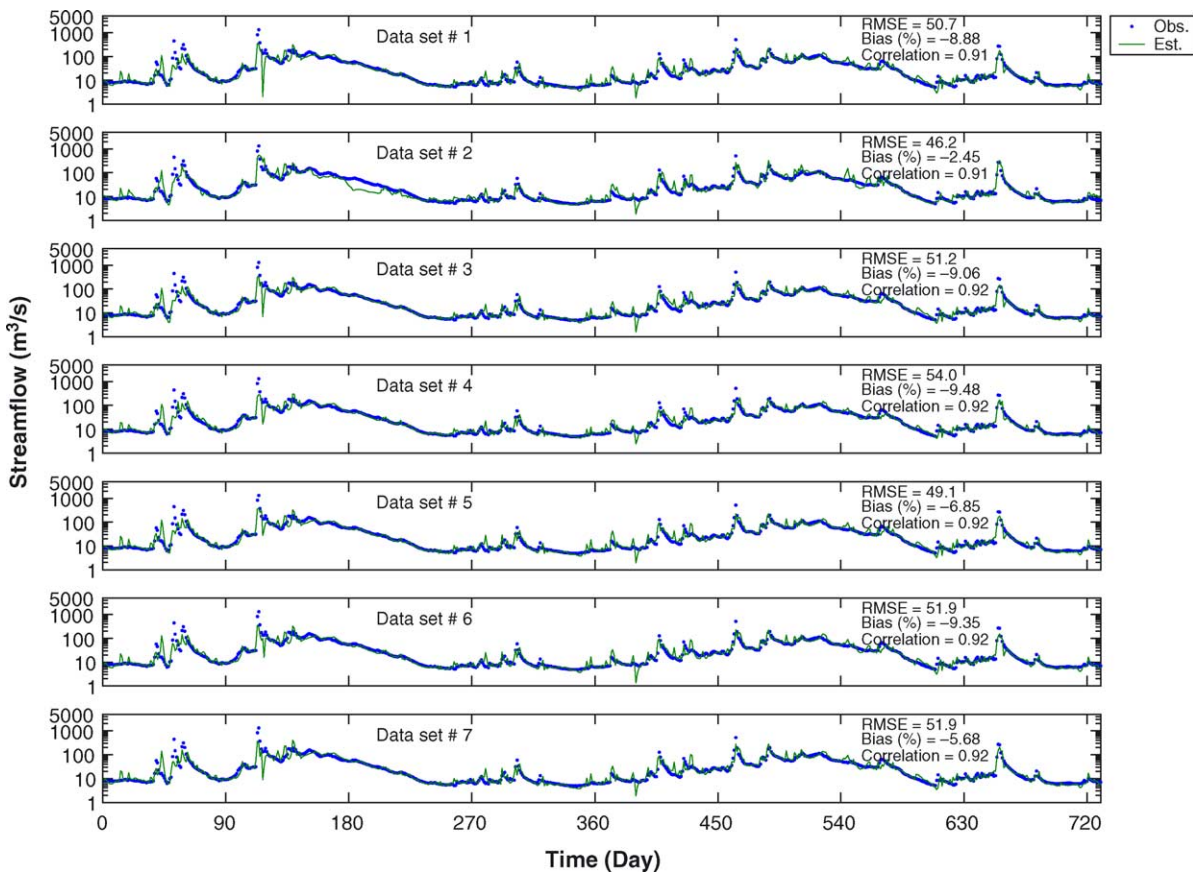

Fig. 10. The simulation of the testing data set (years 1990 and 1991) by SORB model over each cross validation data set.

obtained for different values of β, and RMSE is estimated. By increasing β, the minimum RMSE of

$$\text{Min RMSE} = \frac{t_{\text{trn}}.\text{RMSE}_{\text{trn}}(\beta, \vec{W}|\text{training}) + t_{\text{val}}.\text{RMSE}_{\text{val}}(\beta, \vec{W}|\text{training})}{t_{\text{trn}} + t_{\text{val}}} \tag{20}$$

training and validation sets might occur at different values of β. In such a case, selection of β based on either training or validation may satisfy the optimization criterion for one (training or validation) but deteriorate another one. As a remedy to this, a weighted RMSE can be applied where the RMSE of training and validation periods and their lengths are taken into account. Given the connection weights obtained from training set for each β value, the RMSE of the validation set is also calculated. The following

objective function was used as a compound RMSE to derive the best β and $\vec{W}$:

where t_{trn} = training period; t_{val} = validation period; RMSE_{trn} = root mean square error of the simulation over the training period, RMSE_{val} = root mean square error of the simulation over the validation period.

To perform the cross-validation, clustering was carried out over the whole data set (training and validation period), and the resulting cluster centers were used in training each one of the 7-fold training data sets. The connection weights, $\tilde{W}$, were obtained using (17) for each β. They were then evaluated over
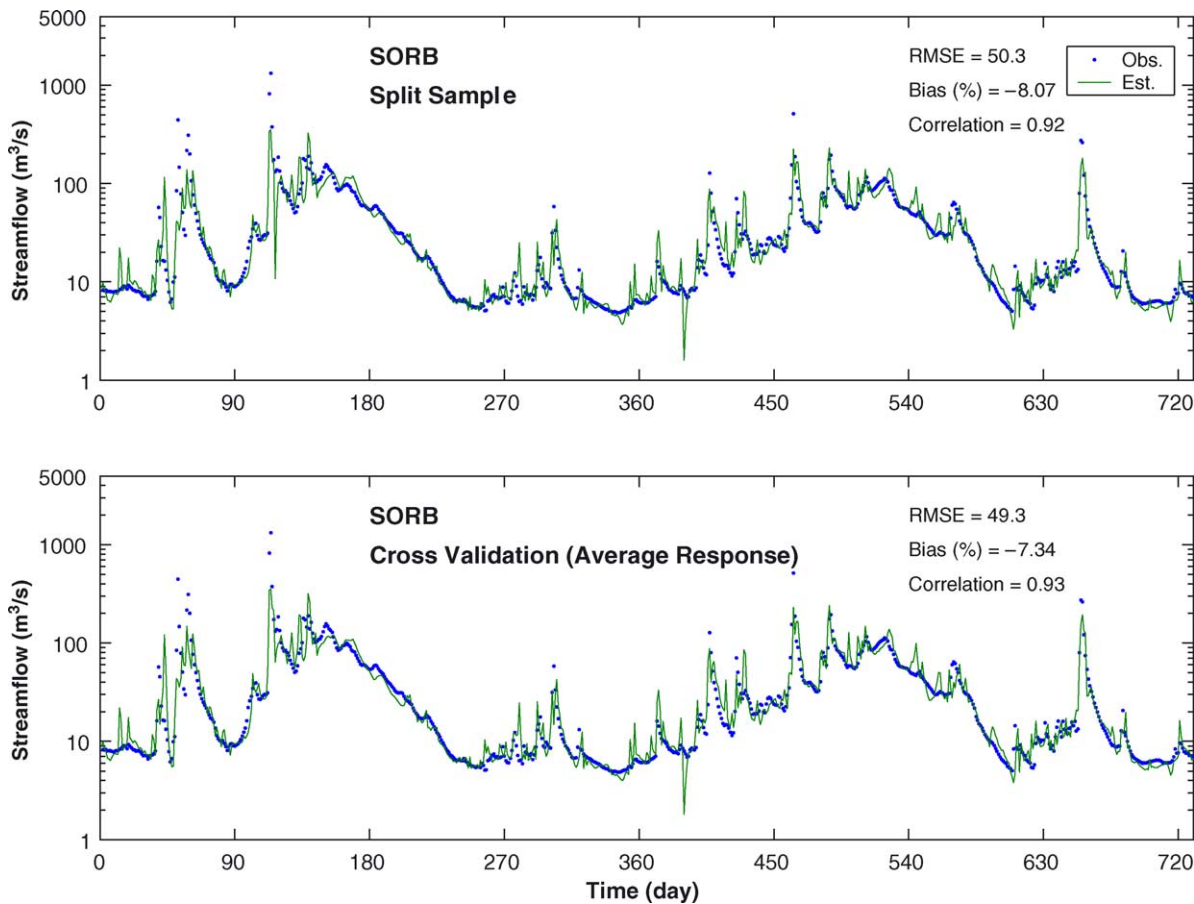

Fig. 11. Comparison of overall testing performance on both split sample and cross-validation data sets.

the validation set, and the final selection of the parameters was based upon (19) for each of the data sets shown in Fig. 9. The parameters obtained from each data set were applied over the testing data set (years 1990 and 1991) to evaluate the generalization capability of the model. Fig. 10 demonstrates the simulation of the testing data set (years 1990 and 1991) by the RBF-SOFM model over each cross-validation data set. Overall comparison of performance in testing the models using the estimated parameters (RBFs parameters and connection weights) derived using different data sets demonstrates similar results. The parameter set associated with the minimum RMSE from specific data set in S-fold cross validation (Fig. 10-b) can be regarded as the representative parameter set or the average response of all cross-validated networks can be used as the representative model for forecasting purposes. The average performance of cross-validations is displayed in Fig. 11 as compared to the performance of model in split sample test. The similarity of results in both split sample and cross-validation justifies the validity of the nonlinear relationship set up between input and output data set (Eq. (13)). This similarity also explains that the choice of split sample data set was an appropriate representative of the whole data set for training and testing purposes which makes the split sample technique comparable to cross-validation.

7. Summary and conclusions

The primary goal of this paper was to investigate the applicability of hybrid structure of Artificial Neural Networks, SORB, in streamflow forecasting. The architecture employed consists of SOFM as an unsupervised training scheme for data clustering, which correspondingly provides the parameters required for the Gaussian functions in RBF neural network. Spread of the Gaussian functions extracted from SOFM seemed to be tunable, and tuning was done in parallel to training the RBF network.

The secondary goal was to compare the SORB architecture with two other ANN architectures, namely MFN and SOLO and also the linear regression model, LINREG. The relative superiority of SORB in terms of forecasting accuracy is seen in Figs. 5 and 6. Although the comparative ability of different

approaches is generally problem-dependent, this comparison offers some insight and is therefore an addition to other comparison studies.

The selection of a training data set is crucial in the ANN modeling and reliance on just one realization (split sample) of the training set may not yield a parameter set with good generalization capability. Moreover, there exists no authoritative procedure to suggest how to partition the data and confirm that each split sample is a good representation. To achieve better generalization, cross validation was employed. A compound RMSE, rather than the simple early stopping method was used as the objective function in cross validation to satisfy both training and validation.

The interpretation of the cross-validation results over an independent data set (testing set) can be done in two ways: (1) the parameter set associated with the minimum error from specific data set in S-fold cross validation (Fig. 10-b) can be regarded as the representative parameter set or (2) the average response of all cross-validated networks can be used as the representative model for forecasting purposes (Fig. 11). While little improvement has been made on cross-validation, it still offers more reliable parameter set than split sample method as it considers different combinations of data set and removes the bias towards the data selection in split sample. It could also be stated that if the cross-validation result is quite different from split sample's, the split sample does not provide enough information for the model to generalize well or a revision of the model structure might be suggested.

Although cross-validation can be regarded as a procedure to avoid the danger of overfitting, it may be sensitive to the method used for partitioning data and making subsets where the historical information in one or more of the subsets may not be enough to yield an appropriate parameter set and correspondingly become deficient with respect to fitness for purpose. As a remedy to this potential problem, a method of continuous resampling may be explored with large number of samples being selected to provide statistical information and provide more accurate approximations. Such a procedure, however, will be computationally expensive.

Acknowledgements

References

Achela, D., Fernando, K., Jayawardena, A.W., 1998. Runoff forecasting using RBF networks with OLS algorithm. Journal of Hydrologic Engineering 3(3), 203–209.

Anctill, F., Perrin, C., Andereassian, A., 2003. Impact of the length of observed records on the performance of ANN and of conceptual parsimonious rainfall-runoff forecasting models, Environmental Modeling and Software, in press.

Anderson, J.A., Rosenfeld, E., 1988. In: Anderson, J.A., Rosenfeld, E. (Eds.), Neurocomputing: Foundations of Research, MIT Press, Cambridge, MA.

ASCE Task Committee on the application of ANNs in hydrology, 2000a. Committee on the application of anns in hydrology artificial neural networks in hydrology I: preliminary concepts. Journal of Hydrologic Engineering 5(2), 115–123.

ASCE Task Committee on the application of ANNs in hydrology, 2000b. artificial neural networks in hydrology, II: hydrologic application. Journal of Hydrologic Engineering 5(2), 124–137.

Birikundavyi, S., Labib, R., Trung, H.T., Rousselle, J., 2002. Performance of neural networks in daily streamflow forecasting. Journal of Hydrologic Engineering 7(5), 392–398.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition, Clarendon Press, Oxford.

Burrough, P.A., McDonnell, R.A., 1998. Principles of Geographical Information Systems, Oxford University Press.

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

Chen, S., Cowan, C.F.N., Grant, P.M., 1991. Orthogonal least squares learning algorithm for radial basis function networks. IEEE Transactions on Neural Networks 2(2), 302–309.

Efron, B., Tibshirani, R.J., 1997. Improvements on cross-validation: the.632 + bootstrap method. Journal of the American Statistical Association 92, 548–560.

Finnoff, W., 1993. Improving model selection by nonconvergent method. Neural Networks 6, 771–783.

Girosi, F., Pogio, T., 1990. Networks and the best approximation property. Biological Cybernetics 63, 169–176.

Goutte, C., 1997. Note on fee lunches and cross-validation. Neural Computation 9, 1211–1215.

Govindraju, R.S., Zhang, 2000. In: Govindraju, R.S., Ramachandra, Rao (Eds.), Artificial neural networks in hydrology, Water Science and Technology Library, Vol. 36., p. 36.

Haykin, S., 1994. Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, Inc..

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

Hsu, K., Gupta, H.V., Gao, X., Sorooshian, S., 1999. Estimation of physical variables from multichannel remotely sensed imagery using a neural network: application to rainfall estimation. Water Resources Research 35(5), 1605–1618.

Hsu, K.L., Gupta, H.V., Gao, X., Sorooshian, S., 2000. In: Govindraju, R.S., Rao Ramachandra (Eds.), Rainfall estimation from satellite imagery, artificial neural networks in hydrology, Water Science and Technology, vol. 36.

Hsu, K.L., Gupta, H.V., Gao, X., Sorooshian, S., Imam, B., 2002. SOLO-An Artificial neural network suitable for hydrologic modeling and analysis. Water Resources Research 38(12), 1–38.see also page 17.

Jayawardena, A.W., Achela, D., Fernando, K., 1998. Use of radial basis type artificial neural networks for runoff simulation. Computer-Aided Civil and Infrastructure Engineering 13, 91–99.

Kégl, B., Krzyzak, A., Niemann, H., 2000. Radial basis function networks and complexity regularization in function learning and classification, International Conference in Function Learning and Classification.

Kohonen, T., 1989. Self-Organizing and Associative Memory.

Kosko, B., 1992. Neural Networks and Fuzzy Systems, A dynamical systems approach to machine intelligence, Prentice-Hall.

Kothyari, U.C., Garde, R.J., 1991. Annual runoff estimation for catchments in India. Journal of Water Resources Planning and Management, ASCE 1171, 1–10.

Mason, J.C., Price, R.K., Temme, A., 1996. A neural network model of rainfall-runoff using radial basis functions. Journal of Hydraulic Research 34 No. 4.

McCuen, R.H., 1997. Hydrologic Analysis and Design, 2nd Ed., Prentice-Hall.

Moody, J., Darken, C.J., 1989. Fast learning in networks of locally tuned processing units. Neural Computation 1(2), 281–294.

Plutowski, M., Sakata, S., White, H., 1994. Cross-validation estimates IMSE. In: Cowan, J.D., Tesauro, G., Alspector, J. (Eds.), Advances in Neural Information Processing Systems 6, 6. Morgan Kaufman, San Mateo, CA, pp. 391–398.

Powell, M.J.D., 1987. Radial Basis Function for Multivariable Interpolation: A review. In: Mason, J.C., Cox, M.G. (Eds.), Algorithms for Approximation, Oxford, Clarendon Press, pp. 143–167.

Sezin, T.A., Johnson, P.A., 1999. Rainfall-Runoff modeling using artificial neural networks. Journal of Hydrologic Engineering 4(3), 232–239.

Sezin, T.A., Masrkus, M., 2000. Precipitation-Runoff modeling using artificial neural networks and conceptual models. Journal of Hydrologic Engineering 5(2), 156–161.

Sorooshian, S., Gupta, V.K., Fulton, J., 1983. Evaluation of maximum likelihood parameter estimation techniques for conceptual rainfall-runoff models: influence of calibration data variability and length on model credibility. Water Resources Research 19(1), 251–259.

Swamee, P.K., Ojha, C.S., Abbas, A., 1995. Mean annual flood estimation. Journal of Water Resources Planning and Management, ASCE 121(6), 403–407.

Thirumalaiah, K., Deo, M.C., 1998. River stage forecasting using artificial neural networks. Journal of Hydrologic Engineering 3(1), 26–32.

US Army Corps of Engineers, (2000). Hydrologic modeling system HEC-HMS.

Wong, S.T., 1979. A Dimensionally homogeneous and statistically optimal model for predicting the mean annual flood. Journal of Hydrology, Amsterdam 42, 269–279.

Yapo, P.O., Gupta, H.V., Sorooshian, S., 1996. Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data. Journal of Hydrology. 181, 23–48.