Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis

Kuo-lin Hsu, Hoshin V. Gupta, Xiaogang Gao, Soroosh Sorooshian, and Bisher Imam

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

Received 23 July 2001; revised 3 April 2002; accepted 3 April 2002; published 19 December 2002.

[1] Artificial neural networks (ANNs) can be useful in the prediction of hydrologic variables, such as streamflow, particularly when the underlying processes have complex nonlinear interrelationships. However, conventional ANN structures suffer from network training issues that significantly limit their widespread application. This paper presents a multivariate ANN procedure entitled self-organizing linear output map (SOLO), whose structure has been designed for rapid, precise, and inexpensive estimation of network structure/parameters and system outputs. More important, SOLO provides features that facilitate insight into the underlying processes, thereby extending its usefulness beyond forecast applications as a tool for scientific investigations. These characteristics are demonstrated using a classic rainfall-runoff forecasting problem. Various aspects of model performance are evaluated in comparison with other commonly used modeling approaches, including multilayer feedforward ANNs, linear time series modeling, and conceptual rainfall-runoff modeling. *INDEX TERMS:* 1821 Hydrology: Floods; 1860 Hydrology: Runoff and streamflow; 1869 Hydrology: Stochastic processes; *KEYWORDS:* artificial neural network, self-organizing feature map, principal component analysis, rainfall-runoff modeling, overfitting, SOLO

Citation: Hsu, K., H. V. Gupta, X. Gao, S. Sorooshian, and B. Imam, Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis, *Water Resour. Res.*, *38*(12), 1302, doi:10.1029/2001WR000795, 2002.

1. Introduction

[2] Artificial neural network (ANN) methods have found increasing utility in a variety of hydrological applications [*Maier and Dandy*, 2000; *ASCE Task Committee on the Application of Artificial Neural Networks in Hydrology*, 2000]. Among the most widely used network structures are the Multilayer Feedforward Network (MFN), the recurrent neural network (RNN), and the radial basis function (RBF) network. In previous work [*Hsu et al.*, 1995, 1997a, 1997b, 1999; *Sorooshian et al.*, 2000], the applicability of ANN methods for hydrologic applications such as streamflow forecasting and estimation of spatial precipitation fields was investigated. Relevant to this paper, *Hsu et al.* [1995] showed that a 3-layer MFN (Figure 1a) provides excellent one step ahead predictions of streamflow, including both flood peaks and recessions. *Hsu et al.* [1997a] explored the RNN extension of the MFN structure (Figure 1b) which adds time-delayed feedback loops to simulate the "storage capacity" of a dynamical hydrologic system.

[3] However, the existence of multiple local optima and extensive regions of parameter insensitivity complicates the identification and training of MFN and RFN networks, which significantly limits their widespread application [*Gupta et al.*, 1997]. Therefore it is important to resolve the network identification problem while maintaining high standards of network performance and accuracy. This paper presents a multivariate ANN procedure, entitled self-organizing linear output map (SOLO), whose structure has been designed for rapid, precise, and inexpensive estimation of network structure/parameters and system outputs, as well as estimates of their uncertainty. More important, SOLO provides additional insight into the underlying input-output processes, thereby extending its usefulness beyond forecast applications. The scope of this paper is organized as follows. The architecture of the SOLO model and an illustrative application to a streamflow prediction problem are described in sections 2 and 3, respectively. In addition, the performance of SOLO is evaluated in comparison with multilayer feedforward ANNs, a linear time series model, and a conceptual rainfall-runoff model. Section 4 discusses how the analysis of intermediate products generated by the network can facilitate insight into the underlying structure of the input-output process. Technical issues including identification of network size, stability of the parameter estimates, principal component analysis, the relationship to linear input-output modeling, and "overfitting" are discussed in section 5. Issues of model prediction uncertainty are presented in section 6.

2. SOLO Model

[4] The architecture of a SOLO network is listed in Figure 2. This network consists of three layers. The input layer is comprised of n_0 neural units (one for each variable of the input vector), each connected to all units of the classification and mapping layers. The classification and mapping layers consist of $n_1 \times n_1$ matrixes (Figure 2): one to classify the input information using a self-organizing feature map (SOFM) [*Kohonen*, 1989] and the other to map the inputs into the outputs using multivariate linear regression. The SOFM matrix functions as a "switchboard" to turn "on" or "off" the units of the regression matrix: i.e., the SOFM matrix classifies each input vector and deter-

0043-1397/02/2001WR000795

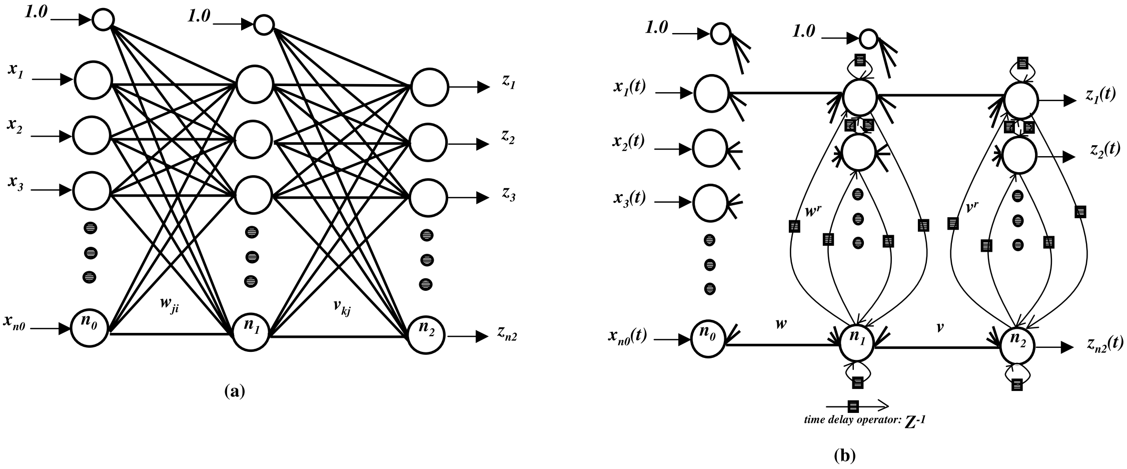

Figure 1. (a) A three-layer feedforward neural network and (b) a three-layer recurrent neural network.

mines to which unit in the regression matrix it must be routed for output prediction. The values of the network parameters (i.e., connection weights linking the units) are determined by the process of calibration (i.e., training).

[5] The mathematical description of SOLO is as follows. Let w_{ji} represent the connection strength (weight, parameter) linking the ith input variable ($i = 1, \ldots n_0$) to the jth SOFM unit ($j \in n_1 \times n_1$), and let v_{ji} represent the connection strength linking the same ith input variable to the jth regression unit. Compute the Euclidian "distance" d_j between the input vector, $\boldsymbol{x} = \{x_i, i = 1, \ldots n_0\}$, and the jth SOFM unit as follows:

$$d_j = \left[\sum_{i=1}^{n_0} \left(x_i - w_{ji}\right)^2\right]^{0.5} \qquad (1)$$

Compute this distance to each SOFM unit and select the unit, c, which has the smallest distance, i.e., $d_c = min(d_j)$, for

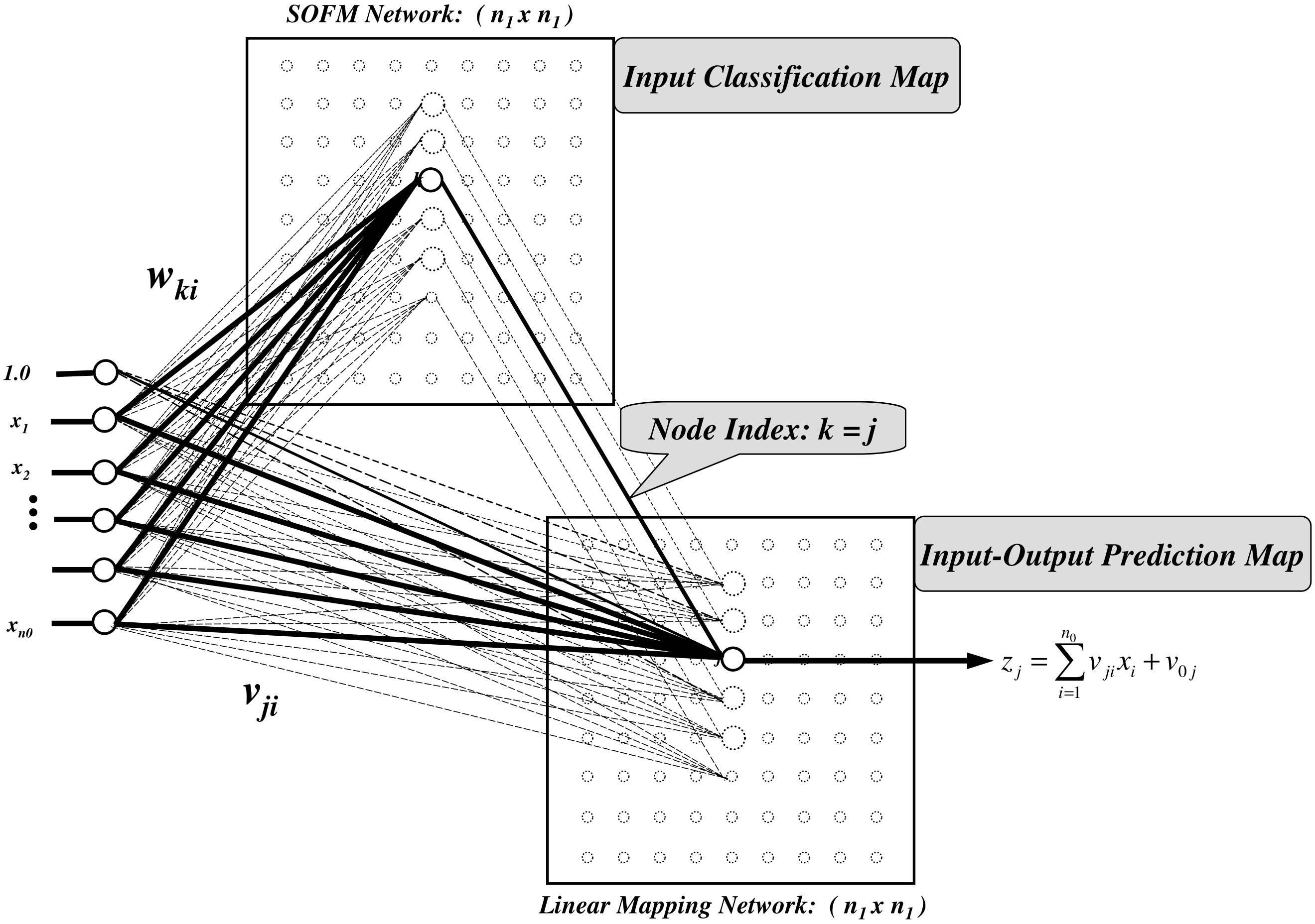
SOFM Network: (n1 x n1)
Input Classification Map
w_ki
1.0
x1
x2
xn0
Node Index: k = j
Input-Output Prediction Map
$z_j = \sum_{i=1}^{n_0} v_{ji} x_i + v_{0j}$
v_ji
Linear Mapping Network: (n1 x n1)

Figure 2. The architecture of a SOLO network.

all j. This identifies the input cluster to which the input vector x_i belongs. All of the corresponding outputs of the units of the SOFM matrix will be "0", except for the c^{th} unit, for which the output will be "1". This output is used to determine which unit of the regression matrix is to be used in the computation of the output value z from the input vector (x)

$$\begin{aligned} z &= \sum_{i=1}^{n_0} v_{ji} x_i + v_{j0} \qquad && \text{if } j = c \\ &= \phi && \text{otherwise} \end{aligned} \tag{2}$$

where ϕ means that the outputs other than the unit $j = c$ are not selected or estimated. In this way, each unit of the regression matrix is represented by a linear input-output regression function that is associated with a single input cluster. The input-output function mapping is therefore accomplished by a set of $n_1 \times n_1$ piecewise linear regression functions that covers the entire input domain.

[6] The training of the network connection weights w_{ji} and v_{ji} is achieved as follows. First, the weights, w_{ji}, of the SOFM matrix are trained using an iterative nonsupervised (self-organizing) procedure. In this procedure, the weights, w_{ji}, in the SOFM matrix are initialized to randomly selected values and are then adjusted so that the nodes which are in the neighborhood Λ_c of the node (c), determined to be closest to the current input vector, are moved towards the input vector using the iterative adjustment rule:

$$\begin{aligned} w_{ji}(m) &= w_{ji}(m-1) + \eta(m)\left[x_j - w_{ji}(m-1)\right] \quad && \text{if } j \in \Lambda_c(m) \\ w_{ji}(m) &= w_{ji}(m-1) && \text{otherwise} \end{aligned} \tag{3}$$

Here, m is the training iteration, $\Lambda_c(m)$ defines the size of a neighborhood around the winner unit c, and $\eta(m)$ is the learning rate (step size) at the iteration m. This procedure is applied several times to the entire data set of input vectors. As the training proceeds, the sizes of both $\Lambda_c(m)$ and $\eta(m)$ are progressively reduced, and the SOFM stabilizes to a form that approximates the distribution of the data in the input space.

[7] Next, the weights v_{ji} of the linear regression matrix are determined. Due to the self-organizing (clustering) procedure outlined above, each node of the regression matrix has become associated with a distinct region of the input space. Therefore the input-output data now associated with node j are used in the determination of the nodal regression parameters (v_{ji}) by solving the linear set of equation

$$Z = X\theta + \varepsilon \tag{4}$$

where Z is a $p \times 1$ vector with p output training data (runoff observations), X is a $p \times n_0$ matrix with p sets (rows) of training input vectors $(x_i)^T$, $i = 1, \ldots n_0$, θ is a $n_0 \times 1$ vector of regression parameters (weights) for unit j, $\theta = [v_{j1}, v_{j2}, \ldots, v_{jn0}]^T$, and ε is a p $\times$ 1 vector of estimation errors with zero mean and variance σ_e^2. In general, because the number of equations (size, p, of the data set) exceeds the number of unknown regression parameters (n_0), the "optimal" (unbiased) estimates of θ can be determined by minimizing the root mean square error of the output residuals, using the equation:

$$\hat{\theta} = \left(X^T X\right)^{-1} X^T Z \tag{5}$$

However, the solution of equation (5) is typically complicated by the presence of significant correlation among the input variables (x_i), causing the matrix (X) to be colinear and the inverse matrix $(X^T X)^{-1}$ to be singular. This problem is avoided by applying a principal component transformation (C) (also called empirical orthogonal function) [*Jolliffe*, 1986; *Tatsuoka and Lohnes*, 1988; *Peixoto and Oort*, 1992] to the matrix X to obtain a matrix Y having independent (orthogonal) column vectors:

$$Y = XC \tag{6}$$

Here Y is the $p \times n_0$ matrix of principal components, and C is the $p \times n_0$ transformation matrix with eigenvectors derived from the covariance matrix of X: $E(X^T X)$ and $C^T C = CC^T = I$ (Appendix A). Substituting equation (6) into equation (4) yields

$$Z = YC^T\theta + \varepsilon = Y\beta + \varepsilon \tag{7}$$

where $\beta = C^T\theta$ can be determined without difficulty, using the relevant analog to equation (5), because the variables in Y are independent (orthogonal). Selecting and including only the largest principal components of Y, those that explain the majority of the variance, can avoid the instability in estimates of the regression parameters. In SOLO, the number (m) of the largest principal components used in the regression is determined to ensure that the ratio $V = \sum_{i=1}^{m} \lambda_i / \sum_{j=1}^{p} \lambda_j .100\% > 95\%$. This reduces the dimension of the inverse matrices to be solved during regression, thereby simplifying and speeding up the network training.

3. Use of SOLO for Streamflow Prediction

[8] Applied ANNs in streamflow prediction are discussed in many recent publications. *Smith and Eli* [1995] trained MFNs to predict the peak discharge and peak time from synthetic data and hypothetical watersheds. In the study by *Minns and Hall* [1996], runoff was generated from a simple nonlinear model using synthetic storm events. The results show that the performance improved with more hidden layers in the network. *Mason et al.* [1996] applied RBF networks in runoff prediction and showed that RBFs are more efficient than slow back propagation learning strategy in the model calibration. *Tokar and Johnson* [1999] explored the network performance to the training data set and found that the model trained from both dry and wet records had a better prediction accuracy. *Ahmad and Simonovic* [2001] used MFNs to predict the amount and timing of peak flow, base flow, timing of rising and falling limbs, and shape of the runoff hydrograph. For more other relevant ANN applications, readers may refer to the summary papers from *Maier and Dandy* [2000] and the *ASCE Task Committee on the Application of Artificial Neural Networks in Hydrology* [2000].

[9] In this case study, the SOLO network was applied to the problem of streamflow prediction, and its performance

was compared with that of four commonly used hydrological modeling approaches. The test data consisted of 36 years (1 October 1948 to 30 September 1983) of daily rainfall and streamflow data for the Leaf River basin (1949 km^2) near Collins, Mississippi. With respect to the data record used in the model calibration, in the previous study, *Yapo et al.* [1996] suggested that approximately eight years of data are required to obtain calibrations that are relatively insensitive to the data period selected. To avoid models calibrated from an insufficient data record, the first 11 years of data were selected for model development and calibration, and the remaining 25 years were used for performance evaluation. The comparison is based only on the ability of each model to provide accurate one-day-ahead predictions of streamflow. Note, however, that each model has different input data requirements, as mentioned in the associated subsections.

3.1. ARX Model

[10] The ARX (n_1, n_2) time series model (autoregressive with exogenous inputs) is a lumped linear regression structure that has been used extensively for the prediction of streamflow using observed rainfall and runoff sequences [*O'Connell and Clark*, 1981; *Wood*, 1980]:

$$q(t+1) = \sum_{i=0}^{n_1} a_i q(t-i) + \sum_{j=0}^{n_2} b_j\, r(t-j) + \varepsilon(t+1) \tag{8}$$

where a_i and b_j are parameters, and $q(t)$ and $r(t)$ are the observed streamflow and rainfall sequences, respectively. The time unit t is one day, and $\varepsilon(t + 1)$ is the error of streamflow estimation. Three previous time steps of rainfall and streamflow observations are used as the inputs to the model (i.e., $n_1 = n_2 = 2$). For a fair comparison of various model performances, the same six input variables in the ARX, MFN, and SOLO models were also selected. The parameters of the ARX model (a_i, b_j) are determined by minimizing the root mean square error (RMSE) of the streamflow residuals computed over the training data set:

$$\min F(a_i, b_j) = \min\left[\sum_t (q(t) - \hat{q}(t))^2\right] \tag{9}$$

3.2. Multilayer Feedforward Network (MFN)

[11] The MFN model [see *Hsu et al.*, 1995] is widely used to model nonlinear processes because the simple three-layer network MFN (n_0, n_1, n_2) shown in Figure 1a has been mathematically proved [*Hornik et al.*, 1990; *Gallant and White*, 1992] to be capable of mapping any kind of continuous nonlinear function (n_0, n_1, and n_2 represent the number of units in the input, hidden, and output layers, respectively). For a comparison of different models, the same six input variables used in the ARX model are used as network inputs:

$$\begin{aligned}\boldsymbol{x} &= [x_1, x_2, x_3, x_4, x_5, x_6]^T \\ &= [r(t), r(t-1), r(t-2), q(t), q(t-1), q(t-2)]^T\end{aligned} \tag{10}$$

The output layer has a single unit representing the runoff prediction, and the hidden layer uses three units, based on tests exploring the use of different numbers of hidden nodes, with performance evaluated over both calibration and evaluation data [*Hsu et al.*, 1995]. The optimal network structure for one-day-ahead prediction of streamflow $\hat{q}\,(t+1)$ on the Leaf River basin was determined to be MFN(6, 3, 1), written as

$$y_j = f\left(\sum_{k=1}^{n_0=6} w_{jk} x_k + w_{j0}\right) \tag{11}$$

$$\hat{q}(t+1) = f\left(\sum_{k=1}^{n_1=3} v_k y_k + v_0\right) \tag{12}$$

$$f(.) = \frac{1}{1 + \exp(.)} \tag{13}$$

where $f(.)$ is the neural transfer function, y_k is the output of unit k in the hidden layer, w_{jk} is the connection weight between the unit k in the input layer and unit j in the hidden layer, v_{jk} is the connection weight between the unit j in the output layer and unit k in the hidden layer, w_{j0} and v_0 are bias weights. Estimates for the parameters (w_{jk} and v_k) were determined by minimizing the RMSE using the linear least squares simplex (LLSSIM) algorithm [*Hsu et al.*, 1995].

3.3. Recurrent Neural Network (RNN)

[12] The RNN model (Figure 1b) is an extension of the MFN network to include internal temporal storage/memory processes. Its application to the problem of streamflow prediction has been discussed by *Hsu et al.* [1997a]. The RNN architecture requires that only the rainfall data at the current time step be used as external input to the network (i.e., $(x_i) = r(t)$, $i = 1$). The network architecture for streamflow prediction on the Leaf River basin is RNN(1, 4, 1), written as

$$y_j(t) = f\left(w_{j1} r(t) + \sum_{k=1}^{n_1} w^r_{jk} y_k(t-1) + w_{j0}\right) \tag{14}$$

where $f(.)$ is the neural transfer function (equation 13), $y_j(t)$ is the output of the hidden unit j at time t, w_{j1} is the connection weight from the input unit to the hidden layer unit j, w^r_{jk} is the time-delayed recurrent connection weights from the hidden unit k to the hidden unit j, and w_{j0} is the bias weight. Model output is calculated by

$$\hat{q}(t+1) = f\left(\sum_{j=1}^{n_1} v_j y_j(t) + v^r \hat{q}(t) + v_0\right) \tag{15}$$

where v_j is the connection weight from hidden unit j to the output unit, v^r is the time-delayed output recurrent connection weight, and v_0 is the bias weight. As with the MFN, the network connection weights, $\{\boldsymbol{w}, \boldsymbol{w^r}, \boldsymbol{v}, \boldsymbol{v^r}\}$, were determined by minimizing the RMSE using the LLSSIM algorithm.

3.4. SAC-SMA Model

[13] The Sacramento Soil Moisture Accounting (SAC-SMA) model is a conceptual multistorage streamflow simulation model, developed and maintained by the U.S. National Weather Service [*Burnash et al.*, 1973; *Burnash*, 1995]. The inputs to the model include the mean basin precipitation at

Table 1. Comparison of Models Using 11-Year Calibration Data and 25-Year Evaluation Data

	Calibration					Evaluation			
Statistics	NSE	RMSE	CORR	BIAS	Time, hours	NSE	RMSE	CORR	BIAS
ARX	0.917	17.24	0.961	0.105	0.08	0.894	20.53	0.949	−0.036
MFN	0.951	13.57	0.976	0.369	46	0.923	17.91	0.962	0.588
RNN	0.904	19.31	0.954	−0.682	41	0.842	25.81	0.924	−0.593
SOLO	0.959	12.36	0.980	0.074	1.5	0.929	17.01	0.965	0.036
SAC-SMA	0.911	17.80	0.959	2.29	8	0.915	19.43	0.960	5.444

the current time step, $r(t)$, and the mean basin evapotranspiration. A description of the SAC-SMA model is given by *Sorooshian et al.* [1993]. The parameters of the SAC-SMA model were calibrated using the shuffled complex evolution (SCE-UA) algorithm, developed at the University of Arizona [*Duan et al.*, 1992]. Discussions related to conceptual rainfall-runoff model identification and use are given by *Sorooshian et al.* [1993], *Sorooshian and Gupta* [1995], *Gupta et al.* [1998], and *Boyle et al.* [2000, 2001].

3.5. SOLO Model

[14] As described above, the SOLO model uses piecewise linear regression functions to predict the streamflow. To remain consistent with the ARX, MFN, and RNN models, the input vector ($\boldsymbol{x}$) used here consisted of the three most recent precipitation and streamflow observations (see equation 10). The SOFM and regression matrixes were selected to consist of 15 × 15 nodes each. A discussion regarding the selection of the size of the SOFM and regression matrices is given later.

3.6. Results

[15] Summary statistics of the one-day-ahead streamflow prediction performance of the five models (ARX, MFN, RNN, SOLO, and SAC-SMA) are presented in Table 1. It shows that the SOLO model consistently provides the best values for the NSE (Nash-Sutcliffe coefficient of efficiency), RMSE, CORR (correlation), and BIAS (bias) statistics for both the 11-year calibration and 25-year evaluation periods. The MFN model provides the next best performance, having similar statistics to SOLO. The remaining three models (RNN, ARX, and SAC-SMA) provide significantly worse calibration and evaluation performance; in particular, the SAC-SMA model has noticeably larger residual bias (BIAS = 5.44).

[16] The CPU time required for model calibration (on a SUN SPARC workstation) is a gross indication of the relative efficiencies of the different modeling procedures. As indicated in Table 1, the ARX model is the most efficient procedure, requiring less than five minutes to calibrate the model using 11 years of data. SOLO is next, requiring 1.5 hours, most of which were spent on the SOFM classification step. However, the SAC-SMA model required eight hours, and the MFN and RNN models each required over 40 hours of CPU time. While this is a somewhat crude basis for comparison, the results strongly support the conclusion that the SOLO network can provide a superior approximation of the complex nonlinear rainfall-runoff relationship in an efficient manner.

[17] A more detailed comparison of model performance is given in Figures 3 and 4. The daily RMSE values (cms) plotted against the volume of annual streamflow (cms) for each model for each year are shown in Figure 3. The solid squares represent calibration years, while the solid circles represent the evaluation years. For each model, it is clear that the error variance increases with "wetness" of the year (i.e., the annual RMSE increases with annual streamflow) in a somewhat linear fashion. An arbitrary line has been added to each graph from (0,0) to (70,40) to provide a basis for simple visual comparison. Notice that the SOLO model provides relatively smaller RMSEs over the full range of annual flows. The SAC-SMA model performs well on high-flow (wetter) years but not as well on low-flow (drier) years. The performance of the ARX model is similar to the SAC-SMA for low-flow years but is poorer for high-flow years. The RNN model provides the worst performance in high-flow years.

[18] Plots of the hydrographs comparing one-day-ahead model predictions with streamflow observations are shown in Figure 4 for the wettest year (1980) of the evaluation period. For this year, the RMSEs are (in order of decreasing performance) SOLO = 25.39 cms, SAC-SMA = 30.44 cms, MFN = 32.31 cms, ARX = 34.36 cms, and RNN = 47.45 cms. The SOLO and MFN models show the closest tracking of the flows on all portions of the hydrograph. The SAC-SMA shows some difficulty in tracking the recessions. The ARX model shows undesirable (high-frequency) "spikes" during transitions between high- and low-flow periods and a significant tendency to underestimate the recessions.

4. Insights Into the Network Structure

[19] The SOFM layer of SOLO partitions the input space into a number of regions, each represented by a number of nodes, such that the nodal connection weights (w_{ji}) represent the cluster mean values for the associated subset of the data. A linear regression equation is then fit between the inputs and outputs for each region so that the result is an efficient, and arbitrarily accurate, piecewise linear approximation of the entire input-output domain.

[20] An interesting by-product of this approach is that an analysis of the properties of the SOFM layer can facilitate insight into the underlying structure of the input-output process. To illustrate this, we present an example in which the SOFM layer was constructed using only a 2 × 2 grid of nodes. The model was trained using the same 11-year Leaf River basin daily rainfall-runoff time series described in section 3. A 100-day portion of the observed rainfall hyetograph and observed and simulated streamflow hydrographs for water year 1961 is shown in Figure 5. As a result of the classification step, each of the four SOFM nodes is activated by a different characteristic pattern (let us call it a "mode") of input (rainfall) behavior. Hence, a different

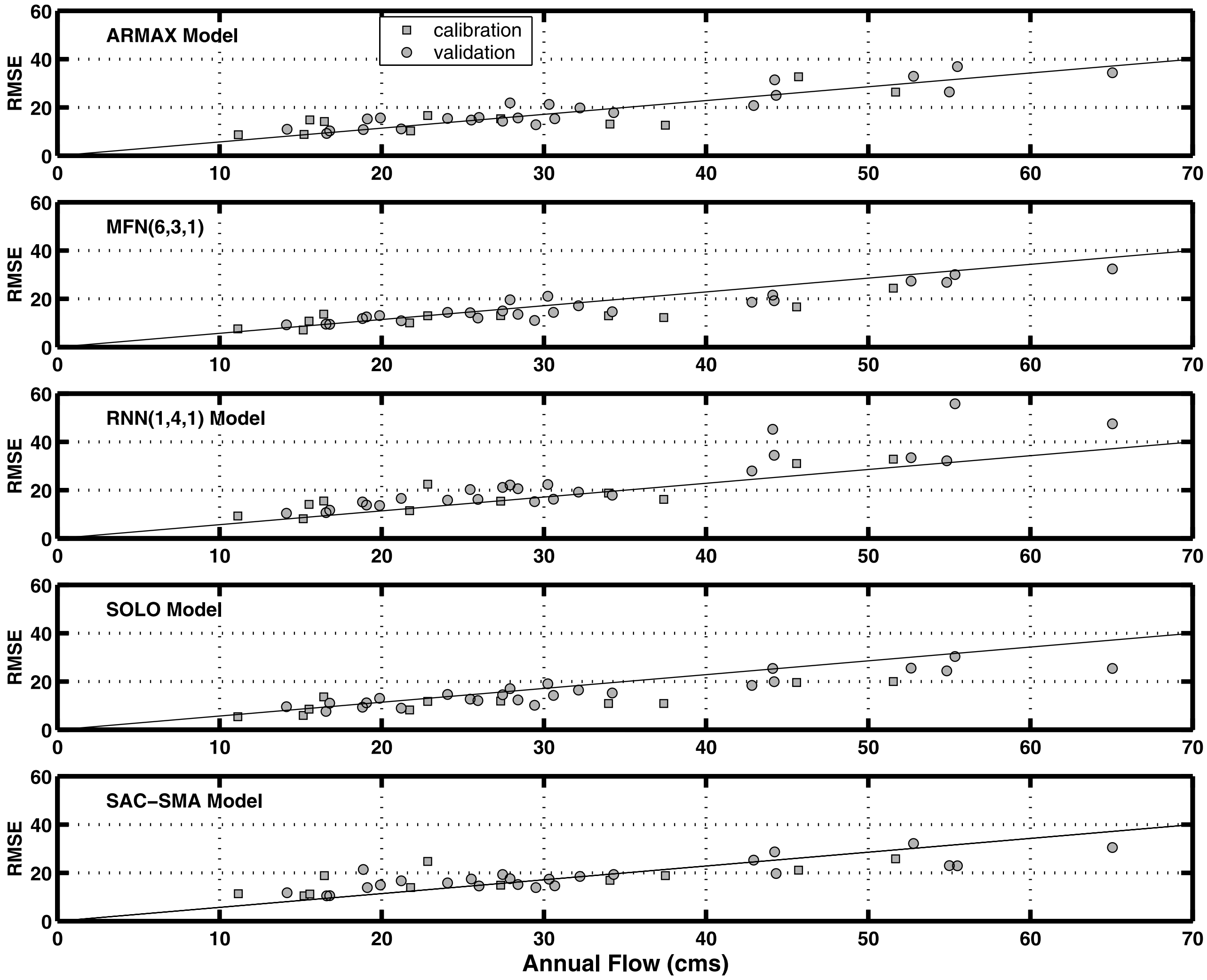

Figure 3. The annual RMSE (cms) with respect to the annual streamflow (cms) of the testing models.

linear regression equation (one of four) is utilized to provide one-step-ahead predictions of different portions of the streamflow hydrograph. In Figure 5, the streamflow predictions, $q(t + 1)$, associated with each of these "modes" of input-output behavior are indicated by different symbols (squares, triangles, diamonds, and stars). Note that the automatic classification algorithm has identified four distinct modes of behavior: base flow recessions (indicated by squares), rising limbs (indicated by diamonds), peaks and quick recessions (indicated by triangles), and early portions of the rising limb having temporary reductions in rainfall intensity (indicated by stars). It is interesting to note that the first three of these are consistent with how a hydrologist might visually partition the hydrograph [e.g., *Boyle et al.*, 2000, 2001], while the fourth represents a subtlety of behavior that might not normally draw the attention.

[21] Detailed results for the 15 × 15 node SOFM classification utilized in the model comparison study presented earlier are shown in Figure 6. Note that each SOFM node has six weights, corresponding to each of the six input variables ($j = 1, \ldots, 6$). To illustrate the distribution of weights across the SOFM matrix, an icon was constructed consisting of three vertical bars (representing the three rainfall inputs $\{r(t - 2), r(t - 1), r(t)\}$ and three line-connected squares (representing the three streamflow inputs $\{q(t - 2), q(t - 1), q(t)\}$. The size of the vertical bar is proportional to the strength of the rainfall input contribution, and the relative vertical position of the squares represents the strength of streamflow contribution. Figure 6 visually illustrates the distribution of rainfall-runoff modes identified by the classification algorithm. For comparison, an interpolated contour plot of the distribution of average streamflow prediction $q(t + 1)$ associated with the layout of SOFM nodes is given in Figure 7a. Analyses of the input-output relationships constructed (learned) by the SOLO procedure are presented in Figures 6 and 7a. For discussion, the five specific SOFM regions outlined loosely in Figure 6 by ellipses and connected by arrows have been identified. The association between these five classification regions and the rainfall-runoff process is also clearly illustrated in Figure 8. The regions are (1) base flow region (region I), (2) increasing rainfall region (region II), (3) peaking hydrograph region (region III), (4) quick recession region (region IV), and (5) slow recession region (region V).

[22] Region I is located in the central area of the SOFM matrix. The behavior is characterized by no-rain and low-

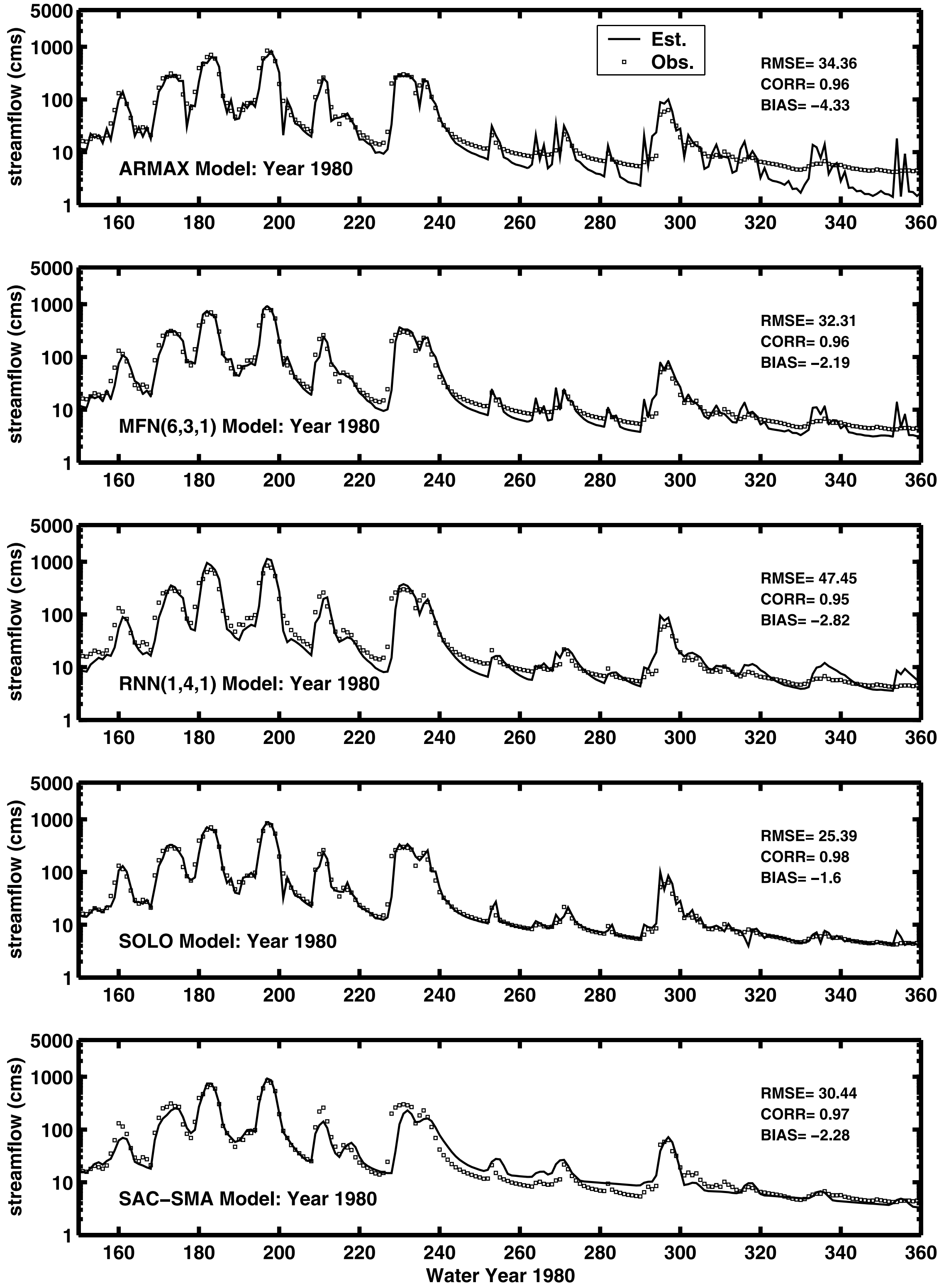

Figure 4. The daily flow time series of testing models over highest validation flow year (1980).

level, almost unchanging streamflows during a 3-day period (Figure 6). The corresponding streamflow prediction is associated with a region of very small values, as shown in Figure 7a.

[23] Region II is located in the top right corner area of the SOFM matrix. Rainfall is steadily increasing during the 3-day period, but streamflows have only just begun to respond (Figure 6). The model predicts high streamflow levels during the next period, as given in Figure 7a. Thus this region identifies the initial stages of a rain storm and the associated rising limb of the hydrograph.

[24] Region III is located in the bottom right and middle portion of the SOMF matrix. The rainfall has peaked, but the streamflow continues to increase (Figure 6). A region of

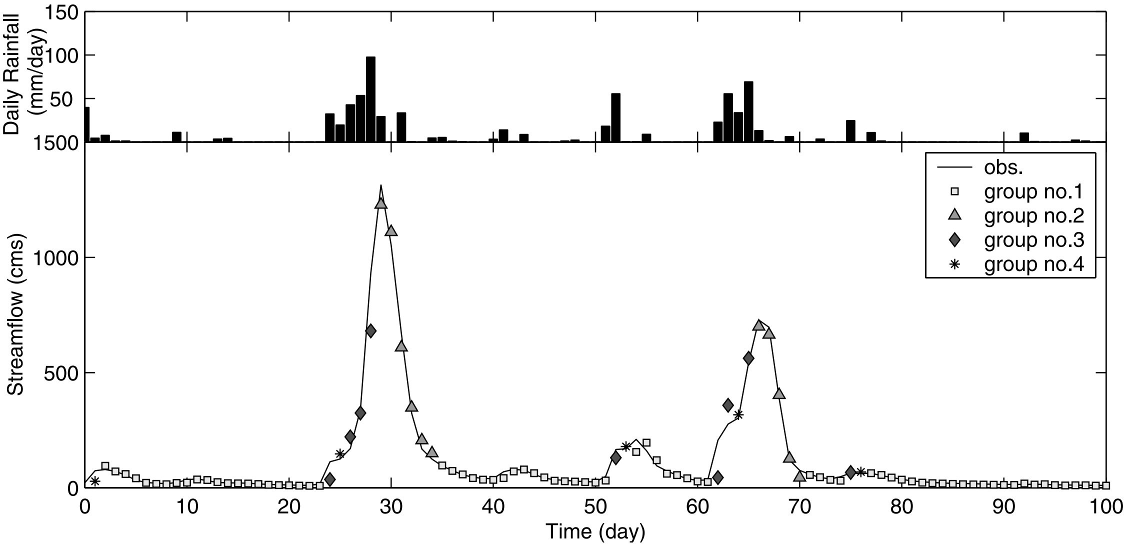

Figure 5. Daily rainfall and observed and simulated streamflow from 2 × 2 units installed in the SOFM layer.

moderate to high streamflow predictions is displayed in Figure 7a. This region is therefore associated with prediction of the peak levels of the hydrograph.

[25] Region IV is located in the bottom left corner of the SOFM matrix. Rainfall intensities have reduced considerably, and the hydrograph has begun to recede. Streamflows, however, are still reasonably high. This region is associated with the early (quick) streamflow recession and determines the rate at which the streamflow will diminish.

[26] Region V is located in the left-middle portion of the SOFM matrix. There has been no rainfall during the past three days, and streamflow has continued to recede. The model predicts a progressively diminishing streamflow value. The remaining portion of the SOFM matrix, not

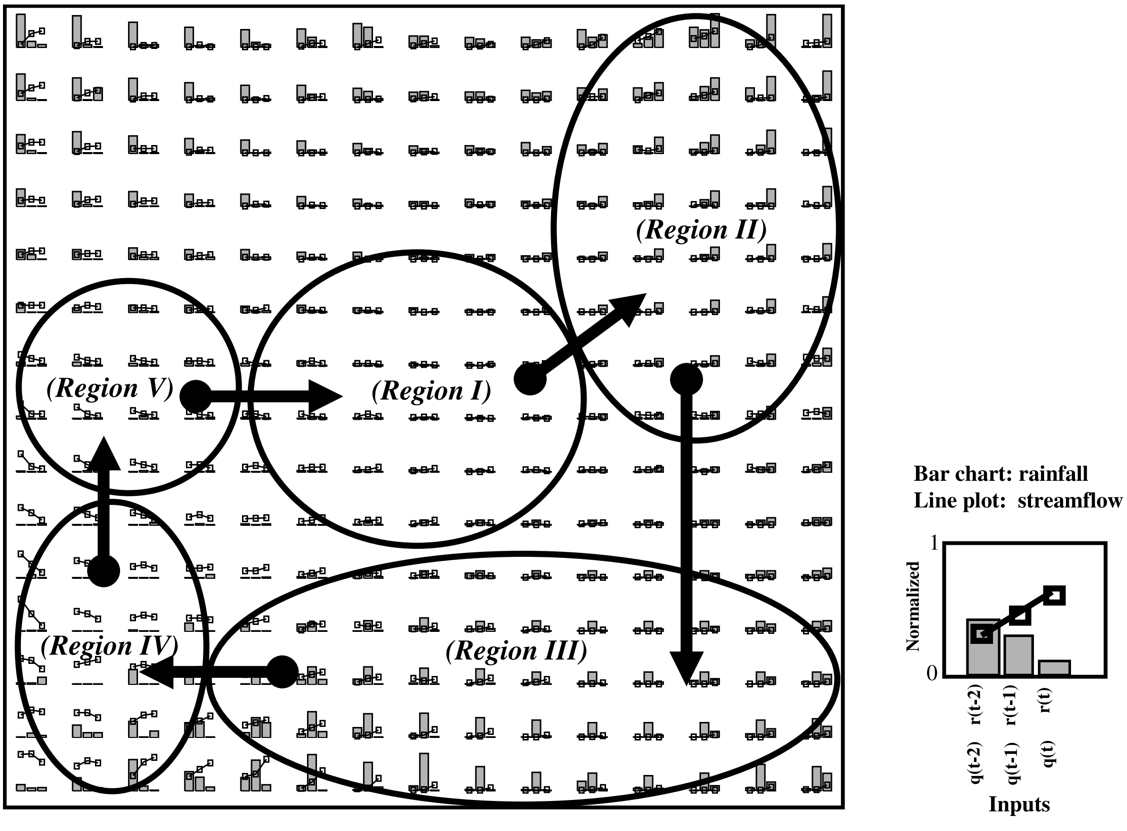

Figure 6. Classified characteristics in 15 × 15 SOFM units demonstrated by six normalized network connection weights.

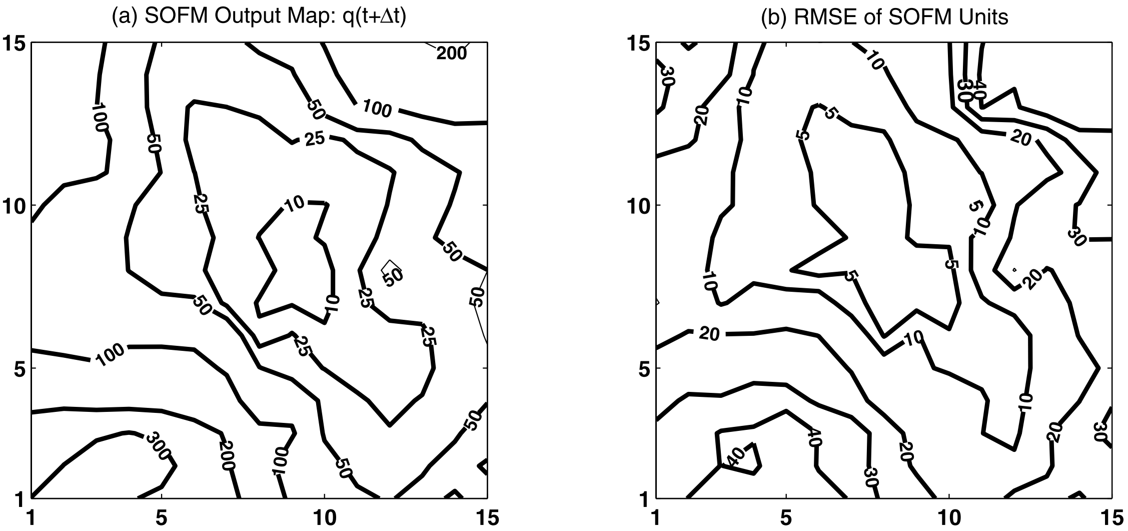

Figure 7. Contour maps of (a) averaged estimated streamflow and (b) estimated RMSEs over the 15 × 15 SOFM units.

discussed above, is clearly associated with events with small durations of rainfall leading to moderate streamflow responses.

[27] Another by-product of the classification scheme is the ability to analyze the level of prediction uncertainty associated with each SOFM node, and by extension, the uncertainty level associated with prediction of different portions of the hydrograph. An interpolated contour plot of the evaluation period root mean square error (RMSE) over the SOFM matrix is presented in Figure 7b. Consistent with the findings presented earlier (Figure 5), a comparison of Figures 7a and 7b indicates that the RMSE is directly related to the level of streamflow; for region I, the RMSEs are around 5 cm/d, increasing to 10–30 cms/d for regions II and III, and 30–40 cm/d for flood peaks (streamflows exceeding 100 cms/d). This information is used later (see section 6) to provide estimates of 95% confidence intervals on the predictions.

[28] Prediction accuracy provided by the SOLO model will depend on correct selection of the physical input variables containing information relevant to the generation of streamflow (output variable). Further, the training data set must contain sufficient variability that is representative of the underlying input-output process. While the skill and experience of the researcher will ultimately determine which input variables are selected for model development, the kinds of tools and analyses presented above can significantly facilitate the input selection and model construction process.

5. Technical Issues Related to Model Performance

[29] Some technical issues related to model performance are given in this section. These include the selection of the size of the SOLO network (i.e., the number of nodes in the SOFM matrix), the identification of stable estimates for the network weights via windowing and principal component analysis, and comments regarding "overfitting" to the data.

5.1. Selecting the Size of the SOFM Matrix

[30] In the example presented above, the selection of SOFM matrix sizes (2 × 2) and (15 × 15) was essentially

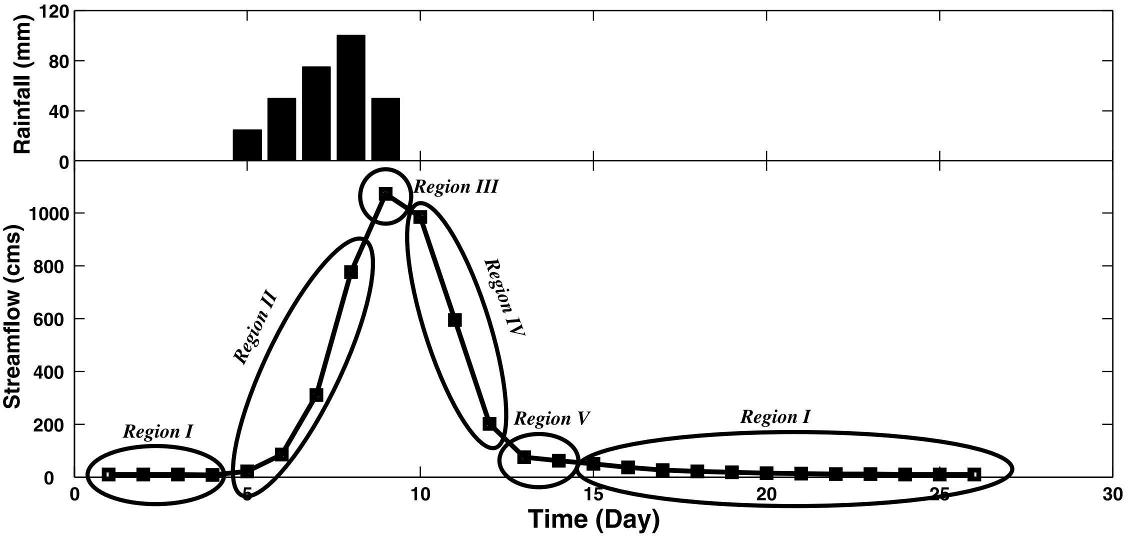

Figure 8. A relationship between five classification regions and the rainfall-runoff process.

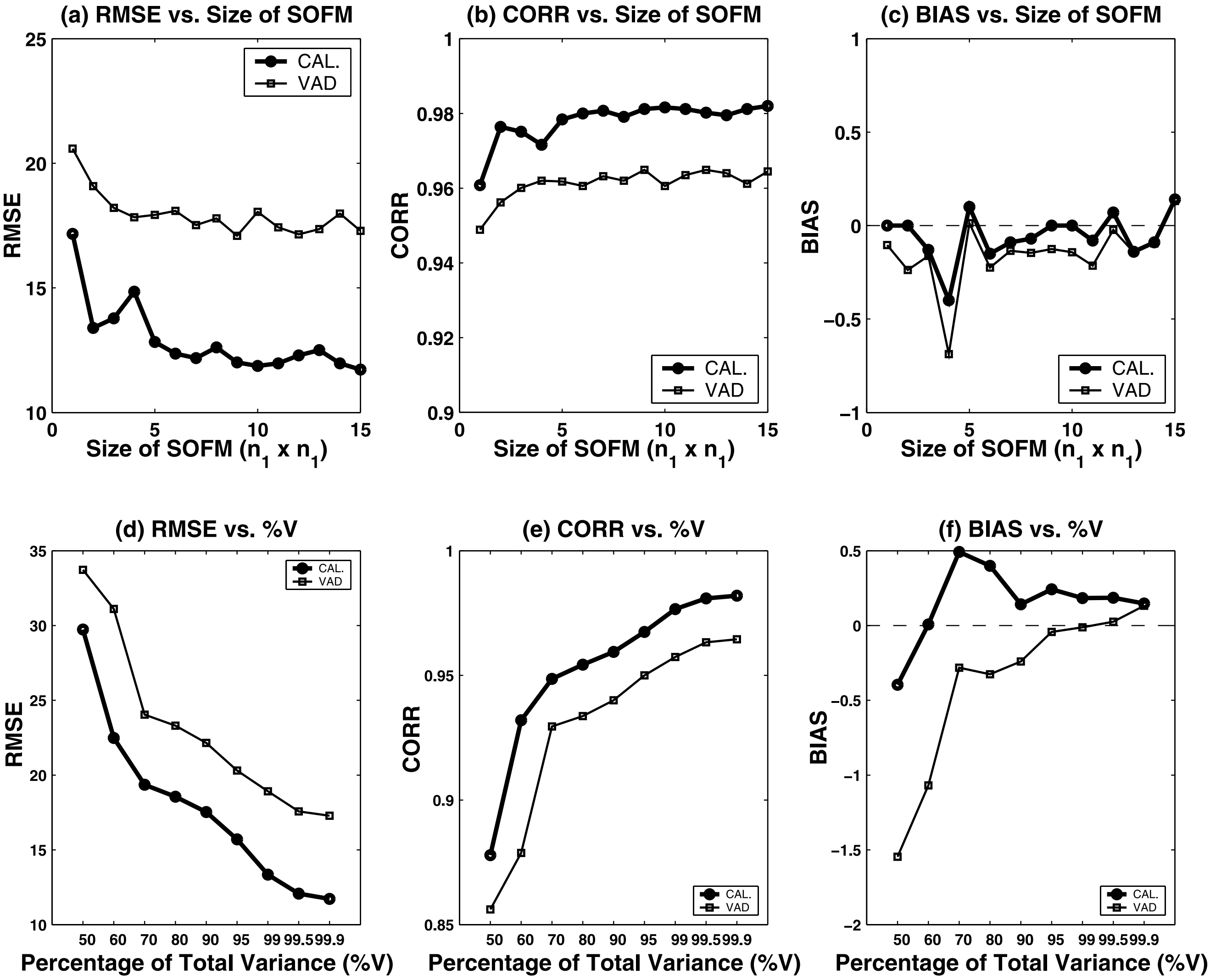

Figure 9. Test of model performance from different size of SOFM units: (a) root mean square error, (b) correlation coefficient, and (c) bias estimates. Model performance of the SOLO model with respect to the percentage of the total data variation (explained by the principal components): (d) root mean square error, (e) correlation coefficient, and (f) bias estimates.

arbitrary and for illustrative/exploratory purposes only. To determine parsimonious/optimal size for the Leaf River basin model, a series of experiments using progressively larger SOFM matrix sizes was conducted. The RMSE, correlation (CORR), and bias (BIAS) statistics, respectively, evaluated over the streamflow prediction residuals as a function of network matrix size are displayed in Figures 9a–9c. The results indicate only marginal improvements for networks of size exceeding 5 × 5. The statistics indicate a small tendency towards negative bias (underestimation) over both the calibration and evaluation periods. High correlation (0.96) between observed and one-day-ahead predicted streamflows over the 25-year evaluation period is obtained.

5.2. Windowing to Ensure Stability of the Model Parameters (Network Weights)

[31] The stability of the parameters estimated for the regression function associated with each SOFM node depends to a large extent on the number of data points available. For the Leaf River basin rainfall-runoff example presented above, 11 years of daily data correspond to a total of 4017 sample data points. The distribution of training data across the 15 × 15 SOFM matrix after the input classification step has been completed is shown in Table 2. Notice that the distribution is highly uneven, with most of the data points belonging to the central region (region I) associated with base flow recession and very small numbers of data points in regions associated with high streamflow values. For many of these nodes, the small number of available data points is insufficient to ensure stable estimates of the regression parameters (in this example, each node has seven regression parameters: one corresponding to each of the six input variables and one to allow for a bias adjustment term).

[32] The solution implemented in SOLO is to extend a large enough window around each node to include data from the surrounding nodes in the fitting of the nodal regression equation. This amounts to a strategy of "borrowing" data from neighboring nodes having somewhat "similar" input-output characteristics. The method involves the selection of a minimum sample size threshold (herein selected empirically to be five times the number of param-

Table 2. Number of Data Classified in Each SOFM Unit: 15 × 15 SOFM Units

SOFM Unit Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	4	3	1	2	4	2	2	0	3	1	3	2	1	6
2	5	1	4	8	4	1	5	5	4	2	2	1	2	2	3
3	1	6	2	5	6	5	4	4	2	2	5	2	3	3	9
4	2	1	8	11	6	29	12	9	4	2	3	3	2	3	7
5	1	5	2	3	2	3	55	10	3	7	3	7	4	18	4
6	3	4	2	4	2	5	12	76	29	11	64	24	28	0	2
7	2	0	2	7	4	3	11	200	195	101	2	4	5	4	2
8	1	6	3	17	44	39	9	727	1422	19	17	7	5	3	2
9	3	4	4	12	16	19	7	4	1	21	4	2	0	2	1
10	2	4	2	5	17	4	4	3	7	77	5	28	11	10	1
11	3	2	3	2	3	4	1	2	1	13	13	32	7	2	3
12	3	1	6	3	1	2	3	5	3	7	32	3	0	3	4
13	5	1	1	1	1	1	1	2	6	5	13	4	2	3	1
14	3	1	2	3	1	3	3	4	2	2	5	4	1	2	2
15	5	0	4	0	3	2	3	3	5	6	4	1	2	2	3

eters) and an extension of the window associated with that node to be just large enough to include enough data to satisfy that threshold. The size of the neighborhood Ω is defined as a matrix of ($2n + 1 \times 2n + 1$), $n = 0, 1\ldots$ centered at the calculation unit. The neighborhood sizes determined for the Leaf River basin example are given in Table 3, and the resulting number of training data points thereby associated with each unit are presented in Table 4. Notice that the nodes with large numbers of data do not need to borrow data from the neighbors, while the neighborhood size for some of the nodes is as large as 3.

[33] The use of data/node windowing has a very interesting implication. Consider an extreme case where the window size for each node is made so large that it includes the data from all of the other nodes. In this case, the procedure would identify an identical linear input-output regression model for all nodes, so that the effective output of SOLO would be identical to that of the linear time series ARX model. If the window size were then to be gradually reduced, the individual regression models at each node would begin to differ, and the overall input-output relationship would progressively adjust to match the underlying nonlinearity of the system. In the extreme case of no-windowing, the regression model at each node can be quite different (if so indicated by the data). The SOLO procedure therefore implements a sensible identification strategy in which (1) the availability of very small amounts of data would lead to identification of a linear input-output model, (2) the availability of large amounts of data would support the detection of system nonlinearity, and (3) the availability of data that are distributed in a nonuniform fashion over different modes of system behavior would lead to a smoothing (windowing) effect which ensures a balanced trade-off between system stability and system nonlinearity.

5.3. Regression Using Principal Component Analysis

[34] Another strategy used in SOLO to ensure stable estimates for the network weights is the use of principal component analysis. A complication inherent in the selection of relevant input variables is the tendency for informative input variables to be correlated to some degree. In the rainfall-runoff example presented earlier, the three previous rainfall values and three previous streamflow values are necessarily correlated in time. The SOLO procedure utilizes principal component (PC) transformations of the input space associated with each node before construction of the nodal input-output regression equation. This ensures a well-posed regression problem and also reduces the number of regression variables (and hence model parameters). The variation of the performance of the 15 × 15 Leaf River basin SOLO model with respect to the percentage of total data variation (explained by the principal components) detailed in Figures 9d–9f. By selecting the number of PCs to account for 95% of the total variance (V = 95%),

Table 3. Window Size of SOFM Unit for Including Sufficient Number of Data in Regression Analysis

SOFM Unit Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	2	2	2	2	2	2	2	2	2	3	3	2	2	3
2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2
3	2	2	1	1	1	1	1	1	2	2	2	2	2	2	2
4	2	2	1	1	1	1	1	1	1	2	2	2	1	1	1
5	2	2	1	1	1	1	0	1	1	1	1	1	1	1	2
6	2	2	2	2	2	1	1	0	1	1	0	1	1	1	2
7	2	2	1	1	1	1	1	0	0	0	1	1	1	1	2
8	2	2	1	1	0	0	1	0	0	1	1	1	2	2	2
9	2	2	1	1	1	1	1	1	1	1	1	1	1	2	2
10	2	2	1	1	1	1	1	2	1	0	1	1	1	1	2
11	2	2	2	1	1	1	2	2	1	1	1	1	1	1	2
12	2	2	2	2	2	2	2	2	1	1	1	1	1	2	2
13	2	2	2	2	2	2	2	2	1	1	1	1	2	2	2
14	3	2	2	2	2	2	2	2	2	1	1	1	2	2	3
15	3	3	3	3	3	3	2	2	2	2	2	2	2	3	3

Table 4. Total Number of Data Included in Finding Regression Parameters According to the Window Size of Each SOFM Unit Listed In Table 2

SOFM Unit Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	65	43	55	56	56	58	50	45	43	38	77	82	45	37	52
2	40	65	83	111	36	125	110	101	73	59	48	47	63	52	43
3	48	76	46	54	75	72	74	49	151	89	72	86	99	85	69
4	47	78	43	45	70	122	131	103	43	284	219	203	45	53	44
5	41	71	40	40	65	126	55	210	151	126	124	138	89	68	94
6	42	84	142	212	295	97	375	76	632	415	64	141	94	67	89
7	42	85	45	85	125	129	1082	200	195	101	249	156	80	51	80
8	42	87	55	109	44	39	1019	727	1422	1782	177	46	233	141	76
9	41	84	57	120	173	159	816	2184	2281	1573	180	79	68	131	58
10	47	86	38	64	82	75	48	2404	129	77	195	102	94	37	54
11	44	67	105	42	41	39	118	203	118	158	210	131	96	41	50
12	39	53	76	74	77	79	84	165	44	93	122	106	56	123	52
13	40	49	58	51	57	57	65	89	36	75	75	64	148	81	37
14	49	39	45	37	44	45	54	68	114	48	44	36	91	37	81
15	39	45	53	63	61	74	40	48	64	66	62	56	49	91	37

the evaluation period RMSE reaches 21 cms/d, the CORR is 0.94, and the BIAS is −0.02 cms. Notice that the performance degrades sharply for V less than 80%. For the 15 × 15 Leaf River basin SOLO model, the numbers of principal components associated with different selections for V are given in Figure 10. With V chosen to be 99.9%, 196 of the 225 nodes utilize all six of the principal components (PCs), while 29 of the nodes use five PCs. In contrast, when V is reduced to 90%, only five nodes use five PCs, 40 nodes use four PCs, 170 nodes use three PCs, and 10 units use two PCs. To balance model efficiency with performance, the value, V = 95%, is suggested.

5.4. Overfitting

[35] Overfitting is a term used when the model fits the calibration data very well, but the performance degrades significantly over independent evaluation periods. To avoid this phenomenon, it is common to reference both the calibra-

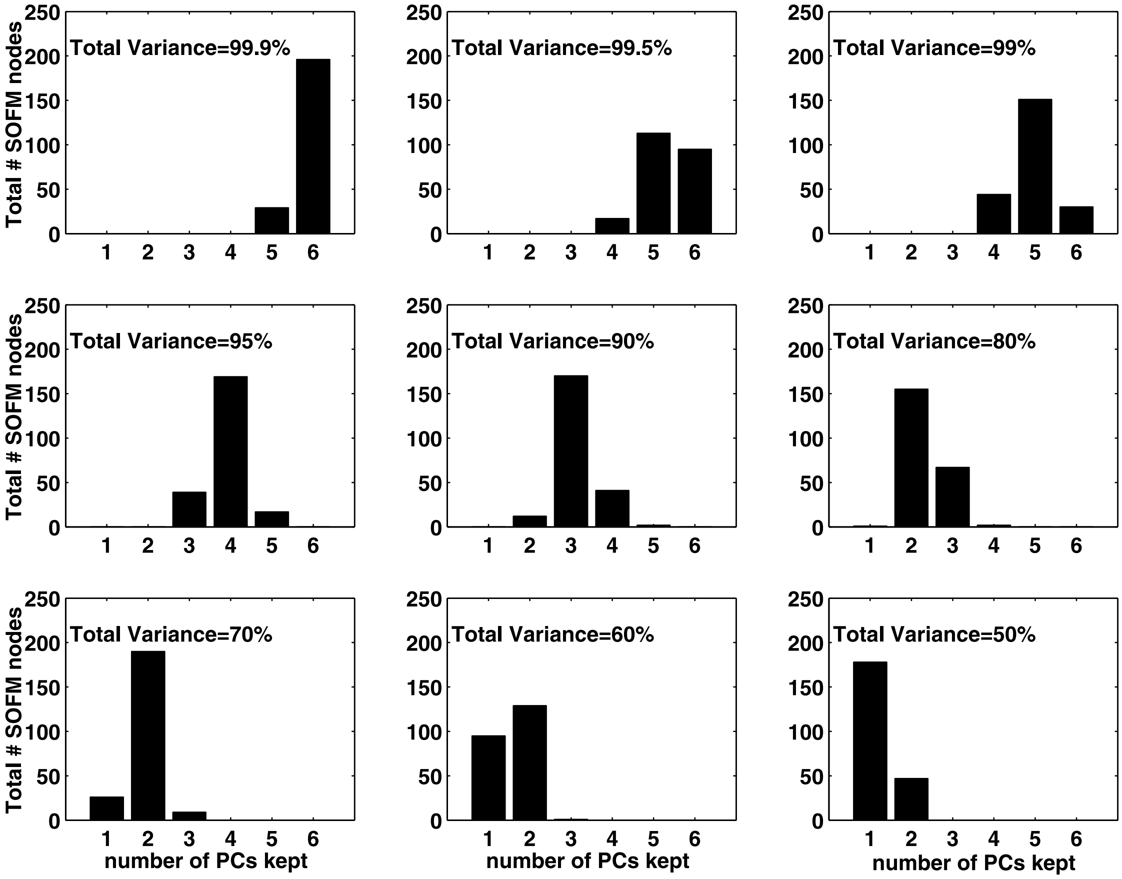

Figure 10. Distribution of the number of principal component variables and the selected total data variance used (V%); the Leaf River SOLO model has six input variables and 15 × 15 SOFM nodes.

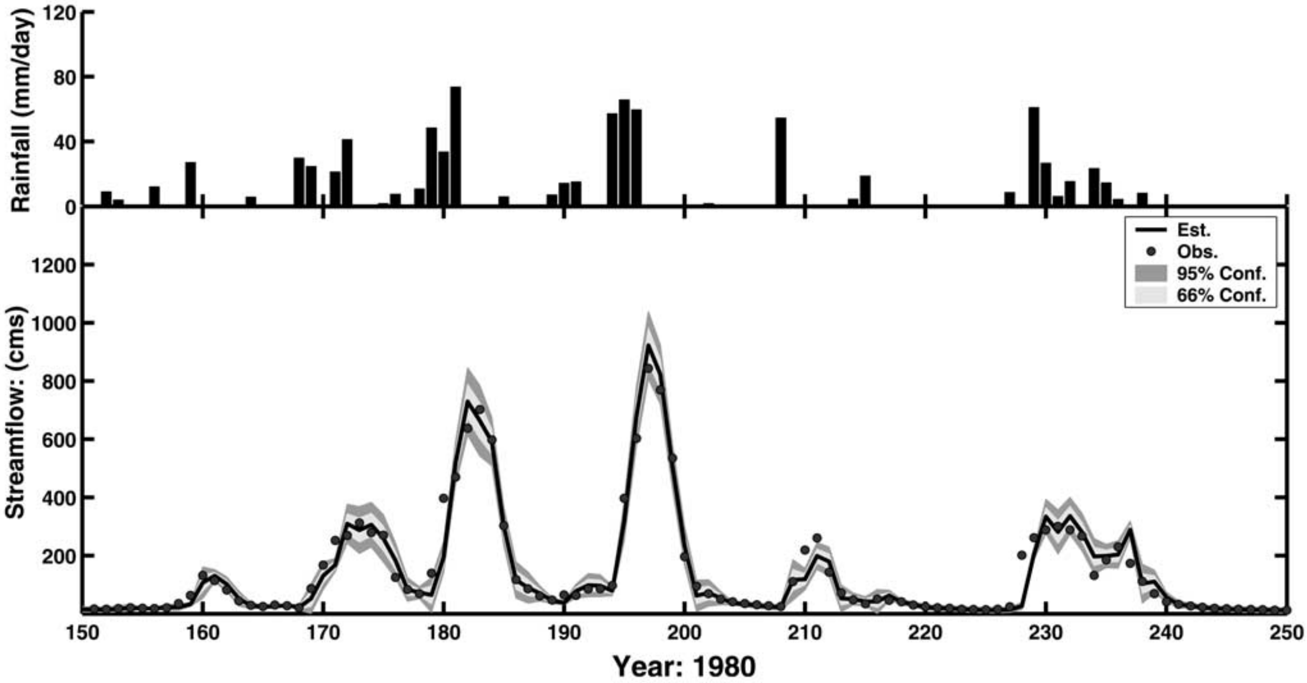

Figure 11. 100 days of observed and predicted streamflow data for water year 1980 (evaluation year); predicted streamflow confidence bounds (66% and 95%) included.

tion and evaluation data during model development and training. An interesting by-product of the procedures described above for selection of network size and for stabilizing the parameter estimates is that these procedures seem to counter the tendency towards "overfitting" of the regression functions. For network sizes smaller than $n_1 = 5$, the network performance tends to oscillate, but for $n_1 > 5$, the performance remains consistent (see Figures 9a–9c). In fact, the performance of the 15 × 15 network is no worse than that of the 5 × 5 network during both calibration and evaluation, suggesting that the tendency towards overfitting has been avoided.

6. Model Prediction Uncertainty

[36] An important extension of the piecewise linear methodology used by SOLO is the ability to use the classical regression theory to provide robust estimates of the model output uncertainty. If the principal component data for an SOFM node are represented by y and the model prediction of streamflow by $\hat{z}$, then $\hat{z} = y\hat{\beta}$, where $\hat{\beta}$ is the vector of linear regression parameters [see equation (7)]. Accordingly, the expected value of the estimate $\hat{z}$ is $y\,\hat{\beta}$, and the variance of $\hat{z}$ is $\sigma^2\gamma$ with $\gamma = y^T(Y^TY)^{-1}y$, if the error in z comes from a normal distribution. Based on this, the upper and lower bounds (U_α, L_α) of the model output predictions corresponding to a $100(1 - \alpha)$ confidence range can be derived as

$$\begin{aligned} U_\alpha &= y\hat{\beta} + t_{1-\alpha/2,n-p}\sigma\sqrt{\gamma} \\ L_\alpha &= y\hat{\beta} - t_{1-\alpha/2,n-p}\sigma\sqrt{\gamma} \end{aligned} \tag{16}$$

where $t_{1-\alpha/2,n-p}$ is a t distribution with $n - p$ degrees of freedom, n is the size of data, and p is the rank of Y [*Haan*, 1977].

[37] A 100-day portion of the observed and predicted streamflow hydrographs for water year 1980 (an evaluation year) is given in Figure 11. The plot also shows the 66% and 95% upper and lower confidence bounds associated with the one-step-ahead streamflow predictions. Note that the piecewise linear SOLO model uses a total of 225 linear regression functions, each associated with a slightly different characteristic input-output behavior. The prediction uncertainty, as shown in Figure 11, is relatively small for the low and medium range of streamflows.

[38] It has been suggested in the literature [*Sorooshian and Dracup*, 1980; *Sorooshian et al.*, 1983] that the errors associated with the rainfall-runoff process are not homogenous (i.e., have nonconstant variance). *Sorooshian and Dracup* [1980] commented that large flows tend to have larger error variance compared to smaller flows, partially because of the nonlinear nature of the rating curve used to transform stage measurements to flow volume estimates. The information depicted in Figures 7a, 7b, and 12 support this view. Figures 7a and 7b show that the size of residual RMSE increases with flow value and the distributions of the calibration data residuals (on the range [-20, 20] cms/d) for each of the 225 SOFM nodes, as presented in Figure 12. The variances are small in the central region I (low flows) and higher in the regions associated with precipitation variability and larger flows. The assumption of normally distributed residual error is reasonably good for node location (14,4), as displayed in Figure 13. However, the assumption does not hold up as well for other nodes. Further study of the residual distribution and its relation to the construction of accurate confidence bounds is ongoing and will be reported in due course.

7. Summary and Discussion

[39] In previous work [*Hsu et al.*, 1995, 1997a, 1997b, 1999; *Sorooshian et al.*, 2000], the applicability of ANN methods for hydrologic applications such as streamflow

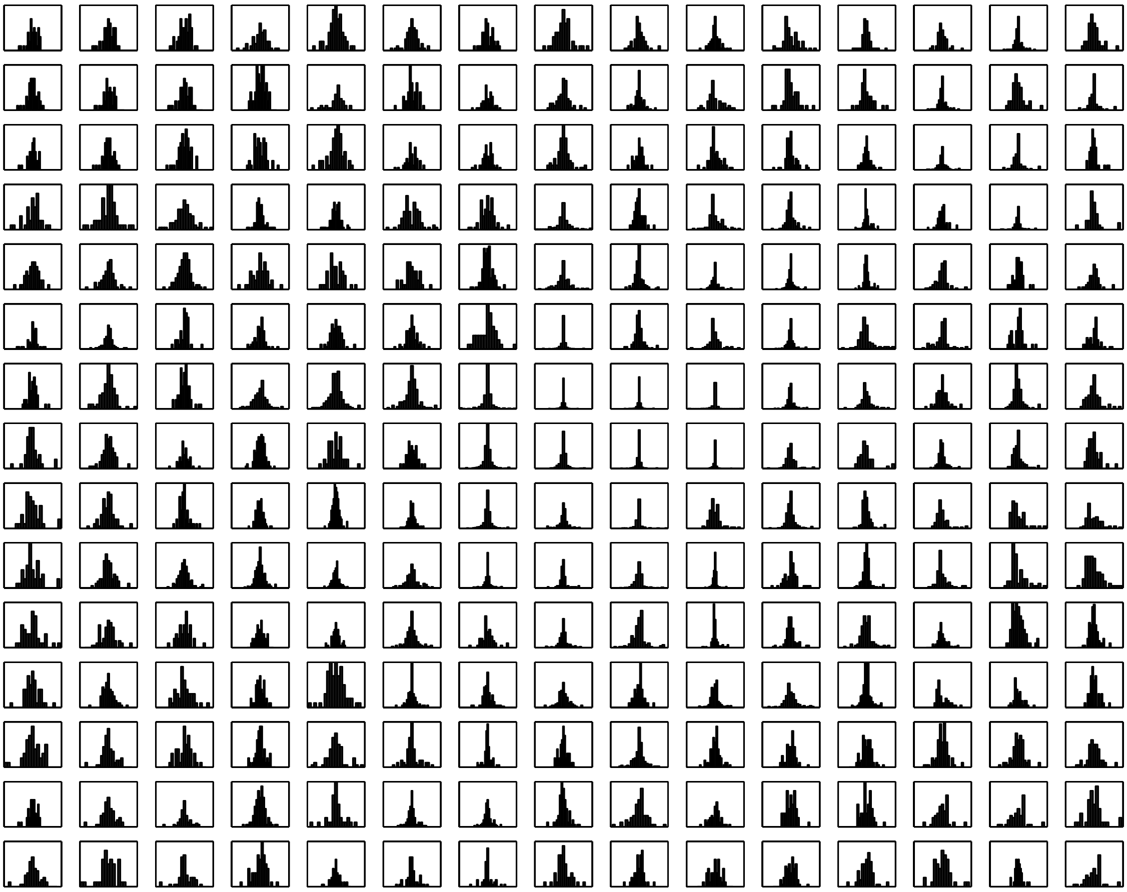

Figure 12. Distribution of calibration data residuals (on the range of [−20 20]) at 15 × 15 SOFM nodes.

forecasting and estimation of spatial precipitation fields was investigated. The work presented here was motivated by network identification difficulties associated with classical ANN approaches that reduce performance accuracy and limit widespread application. A novel artificial neural network structure (SOLO) suitable for a wide variety of hydrologic (and nonhydrologic) applications was presented and illustrated using a case study application to streamflow forecasting. The similarities and differences between SOLO and several commonly used streamflow-forecasting approaches were discussed. The case study illustrates the relative superiority of the SOLO procedure and its ability to provide rapid, precise, and inexpensive estimation of network structure/parameters and system outputs. Equally important to scientists are the characteristics of SOLO that facilitate insight into the underlying physical/functional processes, thereby extending SOLO's usefulness beyond applied forecast applications.

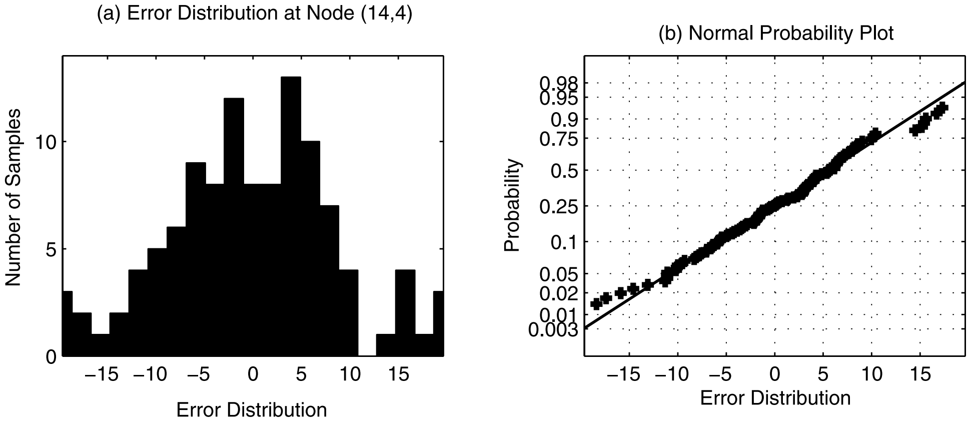

Figure 13. (a) Distribution of error residuals for the SOFM node (14,4) and (b) normal distribution plot of error residuals for the SOFM node (14,4).

[40] The power of the SOLO approach comes from a judicious merging of classical linear regression theory with self-organizing (data clustering) ideas developed by ANN researchers. The result is a function-mapping algorithm based on nodal piecewise linear principal component regression. Of course, the prediction capability of the resulting model is critically dependent on judicious selection of informative input variables. The structure of SOLO facilitates detailed analyses of the explanatory power of the current input selection separately over different portions of the input-output process. The value of this capability will be reported in the future.

[41] Our experiences with the SOLO architecture lead us to suggest that further study of its capabilities and usefulness is warranted. In related work, the use of the SOLO architecture for estimating regional- and global-scale precipitation fields by combining satellite-based remotely sensed data with atmospheric model outputs and surface measurements is being explored. One feature of SOLO which has been explored in that context (and has not been discussed here) is that the piecewise linear mapping can be readily combined with a recursive parameter identification procedure to facilitate (1) progressive model identification from new data as they become available and (2) recursive model updating to track temporal changes in the behavior of the underlying system. Furthermore, extensions of the nodal principal component regression algorithm from ARX to other regression structures such as ARMAX (autoregressive moving average with exogenous inputs) are straightforward to implement. Of particular interest to researchers, however, will be further investigation to determine what information about model system behavior can be revealed (visualized) by creative analyses of the properties of the SOFM matrix.

[42] As always, constructive dialog and collaboration with researchers interested in these methods is invited. The code for the SOLO algorithm can be obtained by request from the first author (hsu@sahra.arizona.edu).

Appendix A

A1. Principal Component Transformation

[43] Consider a standardized data matrix, X, which has p rows (observations) and n_0 columns (variables). Let the covariance matrix of standardized input variables be Σ, where $\Sigma = Cov(X) = E(X^TX)$. The linear transformed orthogonal matrix Y is presented as:

$$Y = XC \tag{A1}$$

where Y is the principal components with element (i, j) of ith observation and jth principal component, and C is a $(n_0 \times n_0)$ matrix with eigenvector elements of the covariance matrix of X: $\Sigma = Cov(X) = E(X^TX)$, and $C^TC = CC^T = I$.

[44] Because the transformed components are uncorrelated to each other, the covariance matrix of principal components is listed below:

$$\begin{aligned} Cov[Y] &= E\left[Y^TY\right] = E\left[C^TX^TXC\right] = \Lambda \\ &= \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_{n_0} \end{pmatrix} = C^T\Sigma C \end{aligned} \tag{A2}$$

[45] The solution of the PCA provides a set of orthogonal-based eigenvectors, C, with their eigenvalues, λ_i, representing the variance of each component after PCA transformation. The total variance of the data matrix is represented as:

$$\text{trace}(\Sigma) = \text{trace}\left(C\Lambda C^T\right) = \text{trace}(\Lambda) = \sum_{i=1}^{n_0} \lambda_i \tag{A3}$$

[46] This shows that the total variance of the data matrix is identical to the total variance after PCA transformation. The orthogonal PCA coordinates are selected according to the first component having the largest variance of the data matrix, X, whereas the other principal components are ranked from large variance to smaller variance, i.e., $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n0}$. To preserve most of the data variance after transformation, one could select the first few principal components with the coverage of most variances in the original data matrix. The percentage of total variance explained by the first mth component is:

$$V = \sum_{i=1}^{m} \lambda_i \bigg/ \sum_{j=1}^{n_0} \lambda_j .100\% \tag{A4}$$

[47] The higher the selection of the total data variance, V, the better the properties of the data matrix are preserved. A small number of principal components are selected, but still retain most of the data variance in the selected components, if the reduction of variables is considered. If the transformation is to prevent the colinearity of regression variables, the selected component number m in equation (A4) can be set for a higher total variance, such as $V = 95\% \sim 99\%$. This higher total variance is mainly to avoid very small λ_k, $k = m + 1 \sim n_0$, which may cause very high variance in the estimated regression parameters.

A2. Principal Component Regression

[48] A multivariate linear regression model having p observations and n_0 independent variables is given below:

$$Z = X\theta + \varepsilon \tag{A5}$$

where Z is a vector of p observations $(p \times 1)$, X is $p \times n_0$ matrix with element (i, j) of ith observation and jth independent variable, θ is a vector of regression coefficients, $\theta = [v_1, v_{2, \ldots,} v_{n0}]^T$, and ε is a vector of estimation error $(p \times 1)$ with zero mean and variance σ_e^2. Parameters are estimated from minimizing the root mean square error of sample data and are given below

$$\hat{\theta} = \left(X^TX\right)^{-1}X^TZ \tag{A6}$$

where $\hat{\theta}$ is the unbiased estimates of regression parameters. The above equation estimates the unbiased regression parameters that minimized the root mean square error. When the input variables are colinear, the inverse matrix of $(X^TX)^{-1}$ becomes singular, which makes finding regression parameters difficult. To reduce the uncertainty of the regression estimates, principal component transformation of input variables into uncorrelated variables before regression analyses is useful for finding more reliable regression parameters.

[49] Substituting the input variable of the linear regression function, as shown in equation (A5), with the transformed principal component variables from equation (A1), we obtain

$$Z = YC^T\theta + \varepsilon = Y\beta + \varepsilon \tag{A7}$$

where $\beta = C^T\theta$ are the regression parameters of principal components.

[50] Parameters of principal component regression are estimated as

$$\hat{\beta} = \left(Y^TY\right)^{-1}Y^TZ \tag{A8}$$

When multicolinearities exist among original input variables, the regression parameters show high variance to those variables that are colinear to others. The regression parameters of the original variable, θ, are given below

$$\hat{\theta} = C\beta = C\left(Y^TY\right)^{-1}Y^TZ = C\Lambda^{-1}C^TX^TZ = \sum_{k=1}^{n_0}\lambda_k^{-1}e_ke_k^TX^TZ \tag{A9}$$

where Λ is a diagonal matrix with kth largest eigenvalue, λ_k, on kth diagonal element. The e_k is the eigenvector of the principal component with kth largest eigenvalue. Assume that observations are uncorrelated and have a constant variance of σ^2 for each observation z_i. The covariance matrix of $\hat{\theta}$ is given below

$$\begin{aligned} E\left(\hat{\theta}\hat{\theta}^T\right) &= \sigma^2C\left(Y^TY\right)^{-1}Y^TY\left(Y^TY\right)^{-1}C^T = \sigma^2C\left(Y^TY\right)^{-1}C^T \\ &= \sigma^2C\Lambda^{-1}C^T = \sigma^2\sum_{k=1}^{n_0}\lambda_k^{-1}e_ke_k^T \end{aligned} \tag{A10}$$

If multicolinearity appears in the original variables, X, it will reveal that the eigenvalues are very small in the later principal components. The variances of the regression parameters become very large from the value of λ_k^{-1} term in the above equation. To avoid large variance on the regression parameters, those small eigenvalue terms in the calculation are removed. The new regression parameters are then expressed as

$$\tilde{\theta} = \sum_{k=1}^{m}\lambda_k^{-1}e_ke_k^TX^TZ \tag{A11}$$

where λ_k, $k=m+1, m+2,\ldots,n0$, are with very small eigenvalues being removed. The covariance of new regression parameters is reduced, and the covariance matrix of those regression parameters becomes:

$$E\left(\tilde{\theta}\tilde{\theta}^T\right) = \sigma^2\sum_{k=1}^{m}\lambda_k^{-1}e_ke_k^T \tag{A12}$$

Because none of the above eigenvalues are small numbers, the variances of the estimated regression parameters are not that high. We have:

$$\theta' = \hat{\theta} - \tilde{\theta} = \sum_{k=m+1}^{n_0}\lambda_k^{-1}e_ke_k^TX^TZ, \qquad E\left[\hat{\theta}\right] = \theta \tag{A13}$$

If the above term is nonzero, omitting this term would result in a biased estimate. However, the advantage from the reduction of parameter variance is substantial under multicolinear circumstances.

[51] **Acknowledgments.** Partial support for this research was provided by SAHRA (NSF STC for "Sustainability of semi-Arid Hydrology and Riparian Areas") (grant EAR-9876800), NASA-EOS, TRMM, and HyDIS research programs (grants NA56GPO185, NAG5-7716, and NAG-8503), and the National Weather Service (grant NA47WG0408). Corrie Thies provided careful proofing and editing of the manuscript.

References

Ahmad, S., and S. P. Simonovic. Developing runoff hydrograph using artificial neural networks, in *Bridging the Gap: Meeting the World*'s *Water and Environmental Resources Challenges, Proceedings of the World Water and Environmental Resources Congress*, pp. 1–17, Am. Soc. of Civ. Eng., New York, 2001.

ASCE Task Committee on the Application of Artificial Neural Networks in Hydrology, Artificial neural networks in hydrology, II, Hydrologic applications, *J. Hydrol. Eng.*, *5*(2), 124–137, 2000.

Boyle, D. P., H. V. Gupta, and S. Sorooshian, Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods, *Water Resour. Res.*, *36*(12), 3663–3674, 2000.

Boyle, D. P., H. V. Gupta, S. Sorooshian, and M. Smith, Toward improved stream forecasting: The value of semi-distributed modeling, *Water Resour. Res.*, *37*(11), 2749–2759, 2001.

Burnash, R. J. C., The NWS river forecasting system-catchment modeling, *Computer Models of Watershed Hydrology,* edited by V. P. Singh., pp. 311–366, Water Resour. Publ., Highlands Ranch, Colo., 1995.

Burnash, R. J. C., R. L. Ferrell, and R. A. McGuire, A generalized streamflow simulation system, report of the Joint Federal-State River Forecast Center, Dep. of Water Resour., State of Calif., and Natl. Weather Serv., 204 pp., Sacramento, Calif., 1973.

Duan, Q., S. Sorooshian, and V. K. Gupta, Effective and efficient global optimization for conceptual rainfall-runoff models, *Water Resour. Res.*, *28*(4), 1015–1031, 1992.

Gallant, A., and H. White, On learning the derivatives of an unknown mapping with multilayer feedforward networks, *Neural Networks*, *5*, 129–138, 1992.

Gupta, H. V., K. Hsu, and S. Sorooshian. Superior training of artificial neural networks using weight-space partitioning, in *Proceedings of the IEEE 1997 International Conference on Neural Networks, June 9–12, Houston, Texas, 1997*, vol. 3, pp. 1919–1923, IEEE Press, Piscataway, N. J., 1997.

Gupta, H. V., S. Sorooshian, and P. O. Yapo, Toward improved calibration of hydrologic models: Multiple and noncommensurable measure of information, *Water Resour. Res.*, *34*(4), 751–763, 1998.

Haan, C. T., *Statistical Methods in Hydrology*, Iowa State Univ. Press, Ames, 1977.

Hornik, K., M. Stinchcombe, and H. White, Universal approximation of an unknown mapping and its derivates using multi-layer feedforward networks, *Neural Networks*, *3*, 551–560, 1990.

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

Hsu, K., V. K. Gupta, and S. Sorooshian, Application of a recurrent network to rainfall-runoff modeling, in *Proceedings of Aesthetics in the Constructed Environment*, pp. 68–73, Am. Soc. of Civ. Eng., New York, 1997a.

Hsu, K., X. Gao, S. Sorooshian, and H. V. Gupta, Precipitation estimation from remotely sensed information using artificial neural networks, *J. Appl. Meteorol.*, *36*, 1176–1190, 1997b.

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

Jolliffe, I. T., *Principal Component Analysis*, Springer-Verlag, New York, 1986.

Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag, New York, 1989.

Maier, H., and G. Dandy, Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications, *Environ. Model. Software*, *15*(1), 101–124, 2000.

Mason, J. C., R. K. Price, and A. Tem'me, A neural network model of rainfall-runoff using radial basis functions, *J. Hydrol. Res.*, *34*(4), 537–548, 1996.

Minns, A. W., and M. J. Hall, Artificial neural networks as rainfall-runoff models, *Hydrol. Sci. J.*, *41*(3), 399–417, 1996.

O'Connell, P. E., and R. T. Clarke, Adaptive hydrological forecasting—A review, *Hydrol. Sci. Bull.*, *26*(2), 179–205, 1981.

Peixoto, J. P., and A. H. Oort, *Physics of Climate*, Am. Inst. of Phys., New York, 1992.

Smith, J., and R. N. Eli, Neural-network models of rainfall-runoff process, *J. Water Resour. Plann. Manage.*, *121*(6), 499–508, 1995.

Sorooshian, S., and J. A. Dracup, Stochastic parameter estimation procedures for hydrologic rainfall-runoff models: Correlated and heteroscedastic error cases, *Water Resour. Res.*, *16*(2), 430–442, 1980.

Sorooshian, S., and H. V. Gupta, Model calibration, in *Computer Models of Watershed Hydrology*, edited by V. P. Singh, pp. 23–63, Water Resour. Publ., Highlands Ranch, Colo., 1995.

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

Sorooshian, S., Q. Duan, and V. K. Gupta, Calibration of rainfall-runoff models-application of global optimization to the Sacramento soil moisture accounting model, *Water Resour. Res.*, *29*(4), 1185–1194, 1993.

Sorooshian, S., K. Hsu, X. Gao, H. V. Gupta, B. Imam, and D. Braithwaite, Evaluation of PERSIANN system satellite-based estimates of tropical rainfall, *Bull. Am. Meteorol. Soc.*, *81*(9), 2035–2046, 2000.

Tatsuoka, M. M., and P. R. Lohnes, *Multivariate Analysis: Techniques for Educational and Psychological Research*, MacMillan, New York, 1988.

Tokar, A. S., and P. A. Johnson, Rainfall-runoff modeling process, *J. Hydrol. Eng.*, *4*(3), 232–239, 1999.

Wood, E. F., (Ed.), *Workshop on Real Time Forecasting/Control of Water Resources Systems*, Pergamon, New York, 1980.

Yapo, P. O., H. V. Gupta, and S. Sorooshian, Automatic calibration of conceptual rainfall-runoff models: Sensitivity to calibration data, *J. Hydrol.*, *181*, 23–48, 1996.

X. Gao, H. V. Gupta, K. Hsu, B. Imam, and S. Sorooshian, Department of Hydrology and Water Resources, University of Arizona, Tucson, AZ 85721, USA. (hsu@hwr.arizona.edu)