# Hydrologic forecasting using artificial neural networks: a Bayesian sequential Monte Carlo approach

Kuo-Lin Hsu

## ABSTRACT

Sequential Monte Carlo (SMC) methods are known to be very effective for the state and parameter estimation of nonlinear and non-Gaussian systems. In this study, SMC is applied to the parameter estimation of an artificial neural network (ANN) model for streamflow prediction of a watershed. Through SMC simulation, the probability distribution of model parameters and streamflow estimation is calculated. The results also showed the SMC approach is capable of providing reliable streamflow prediction under limited available observations.

**Key words** | artificial neural network, parameter estimation, rainfall–runoff modeling, sequential Monte Carlo, streamflow forecasting

**Kuo-Lin Hsu**
Center for Hydrometeorology and Remote Sensing,
Department of Civil and Environmental Engineering,
University of California Irvine,
Irvine CA 92697-2175,
USA
Tel.: +1 949 824 8826
Fax: +1 949 824 8831
E-mail: *kuolinh@uci.edu*

## INTRODUCTION

Artificial neural network (ANN) models are known for their capability of modeling nonlinear systems through direct learning from data. Many ANN models have been widely applied in hydrologic time series analysis, such as the prediction of catchment runoff, water level and precipitation (Hsu *et al.* 1995, 1999, 2002; Govindaraju & Rao 2000; Maier & Dandy 2000, 2001; Coulibaly *et al.* 2001; Abrahart & See 2002; Hong *et al.* 2004; Moradkhani *et al.* 2004; Coulibaly & Baldwin 2005; Dawson *et al.* 2006; Parasuraman *et al.* 2006*a*). For general discussion of ANN model training and applications in hydrologic systems, readers may refer to several review articles such as Govindaraju & Rao (2000), Maier & Dandy (2000) and Dawson & Wilby (2001).

A number of ANN structures have been explored in the literature. Among those, the most commonly used one is the multilayer perceptron (MLP). In order to train a neural network to perform some tasks, the error back-propagation (BP) training method is the most widely used method. The BP method calculates the error derivative of the weights (or parameters) and then adjusts the weights in the direction to reduce the error. Alternative training methods, such as Newton's methods, conjugate gradient (CG), Levenberg–Marquardet (LM) algorithm, genetic algorithms (GA) or linear least-squares simplex (LLSSIM), were also proposed (Hsu *et al.* 1995; Gupta *et al.* 1997; Coulibaly *et al.* 2000; Maier & Dandy 2000). It is reported that the LM method is more effective than the conventional gradient decent techniques such as BP and CG (Maier & Dandy 2000; Coulibaly *et al.* 2005). The above training algorithms intend to find the optimal ANN model parameters by minimizing the error function of the training dataset. The goal of training a network is not only to fit to the training data perfectly, but also to be capable of generalizing model behavior to the new dataset. Over-fitting the data may happen when training data is insufficient or the model is over-parameterized in the training stage. To prevent over-fitting the training data, many approaches were reported, including (1) using sufficient data for network training, (2) implementing cross-validation strategy, (3) applying early stopping criteria, (4) adding on a regularizing term in the objective function and (5) combining the predictions of several models (e.g. bagging and boosting algorithms). In addition to treating ANN over-fitting, the ensemble estimation of multiple models provides a range of estimates which can be used to quantify the

doi: 10.2166/hydro.2010.044

distribution of forecasts (Schapire 1990; Bishop 1995; Breiman 1996; Haykin 1999; Shu & Burn 2004; Parasuraman *et al.* 2006*b*).

The training algorithms mentioned above intend to provide either the maximum likelihood estimation or the minimum least-squares error solution (a special case of maximum likelihood estimation). However, the uncertainty of the model estimation is not much explored. The development of Bayesian Monte Carlo simulation methods provides an option to calculate the uncertainty of model behavior through the effective sampling and evolution of the sampled distribution (Mackay 1992; Bishop 1995; Neal 1996). Several advantages of using Bayesian Monte Carlo methods for model parameter identification were identified. For example, these approaches can provide a way to integrate both observations and the prior information of parameters in the computational framework through the Bayes' rule. Further, the uncertainty bound of parameters and model estimation can be calculated from the simulation samples. With suitable assigning of the likelihood function, there is no requirement to split the data into "training" and "validation" sets for function generalization (Mackay 1992; Bishop 1995; Neal 1996; Khan & Coulibaly 2006).

Several studies using the Bayesian method in training ANNs for hydrologic applications have been reported in the literature. Khan & Coulibaly (2006) applied the Bayesian learning approach to train an ANN model for streamflow forecasting of a catchment in Northern Quebec. The simulation from a Bayesian neural network (BNN) was compared to the simulation from a standard ANN model and a conceptual rainfall–runoff model (HBV-96). The results show that BNN not only outperformed the others but also is capable of providing the uncertainty range of predictions. The above approach assumes that the probability distribution of ANN parameters is invariant in time and can be estimated by a batch of data based on Bayes' rule. Khan & Coulibaly (2006) consider that parameters of ANN follow a Gaussian probability distribution. Mean and covariance matrix are calculated from samples. Kingston *et al.* (2005) applied ANNs for the prediction of salinity of river flow with models trained using a Markov chain Monte Carlo (MCMC) simulation method (adaptive Metropolis and Gibbs sampling). Using MCMC simulation, Kingston *et al.* found that the probability distribution of ANN weights can be non-Gaussian and multimodal. The simulation using MCMC provides additional insights into the probability distribution of model parameters. One limitation of the MCMC approach is that the computational cost through the selecting or discarding of samples to find the probability distribution of parameters can be high.

Sequential Monte Carlo (SMC) approaches further extend the paradigm from the batch fixed-data-length calibration to sequential data assimilation. The SMC approaches are flexible for taking into account the new information (data) which is continuously available in time. Many system identification problems involved in describing the model states or parameters are stochastic in nature through the statistical inference of the likelihood function or the posterior distribution. Under the process of a model being linear and Gaussian-distributed, the Kalman filter is one very popular approach to update model states and estimation sequentially. When a process model is nonlinear and non-Gaussian, it is not easy to obtain a reliable statistical property of estimates from the high-dimension and multimodal probability distribution. Monte Carlo sampling approaches can be very useful for such a situation, but require much higher computation cost. Recently, with the increasing computational capacity of modern computers, SMC methods are becoming popular for the simulation of nonlinear and non-Gaussian systems. The basic process of the SMC methods starts with drawing a large amount of samples and allowing them to sequentially evolve in time using Bayes' rule. Studies have shown that the SMCs are effective in tracking the hidden states and time-varying parameters of complex dynamic systems (de Freitas *et al.* 2000; Doucet *et al.* 2000; Arulampalam *et al.* 2002; Doucet & Tadic 2003; Moradkhani *et al.* 2005).

In this paper, the SMC method was applied to the parameter estimation of Artificial Neural Network (ANN) models for streamflow forecasting. Through SMC simulation, the distributions of model parameters as well as the hydrologic responses were calculated. The uncertainty model estimates at 75% and 95% confidence intervals were calculated. The scope of this paper is organized as follows: the next two sections describe ANN models and parameter estimation through Bayesian statistical inference. In the fourth section, the sequential Monte Carlo sampling strategy for the parameter identification of nonlinear and non-Gaussian models is presented. The fifth section provides a case study of using the ANN model and SMC simulation in streamflow forecasting. Finally, the conclusions of this study are provided in the last section.

## ARTIFICIAL NEURAL NETWORKS

The MLPs are widely used to model nonlinear processes because of their capability of mapping complex continuous nonlinear functions (Hornik *et al.* 1990; Gallant & White 1992). As shown in Figure 1, a three-layer perceptron, MLP($n_0$, $n_1$, 1), is used in this study, where $n_0$, $n_1$ and $n_2 = 1$ represent the process variables in the input, hidden and output layers, respectively. The relationship of input ($x_i$), hidden ($y_i$) and output ($z_k$) variables can be described below:

$$y_j = f\left(\sum_{i=1}^{n_0} \phi_{ji} x_i + \phi_{j0}\right) \tag{1}$$

$$\hat{z} = g\left(\sum_{j=1}^{n_1} \varphi_j y_j + \varphi_0\right) \tag{2}$$

where $\theta = \{\vec{\phi}, \vec{\varphi}\}$ are network parameters, and $f$ and $g$ are transfer functions. A sigmoid function is often used as the transfer function, which can be presented as: $f(a) = g(a) = 1/[1 + \exp(-a)]$.

Other types of transfer functions, such as hyperbolic tangent and Gaussian functions, are also frequently used in other applications. More discussions about the transfer functions can be found in Haykin (1999) and Schalkoff (1997).

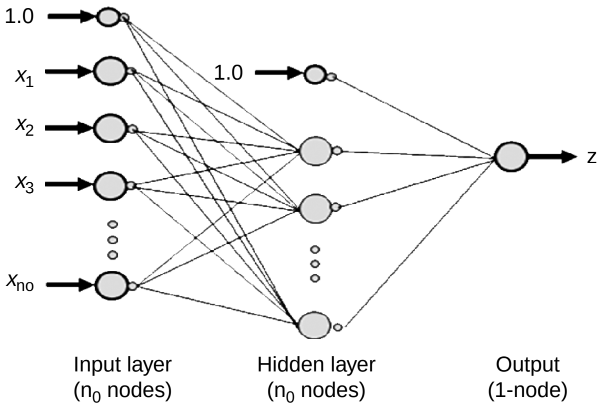

**Figure 1** | A three-layer feedforward neural network.

## BAYESIAN PARAMETRIC INFERENCE

Bayesian methods have a number of features making them very useful in many practical applications. For example, one direct advantage from Bayes' rule is that it allows us to incorporate the prior knowledge with the observed data in a coherent way. Further, Bayesian implementation offers a recursive formulation, enabling the evolution of the prior probability distribution of parameters from the newly available observations. Let's assume that the measurement is available up to time step $k$; the posterior distribution of model parameters given observations $Z = Z_t = \{z_1,z_2, \dots ,z_k\}$ can be calculated by Bayes' rules:

$$p(\theta|Z) = \frac{p(\theta, Z)}{p(Z)} = \frac{p(Z|\theta)p(\theta)}{p(Z)} \tag{3}$$

where $p(\theta)$ is the prior probability of parameters, $p(Z|\theta)$ is the likelihood function and $c = p(Z) = \int_\theta p(Z|\theta)p(\theta)\mathrm{d}\theta$ is a normalizing constant. Equation (3) can be presented below:

$$p(\theta|Z) = \frac{1}{c}p(Z|\theta)p(\theta) \propto p(Z|\theta)p(\theta) \tag{4}$$

To accommodate the continuing available data, Bayes' rule can be presented in sequential format. At time $t$, the new observation $z_t$ is available. Bayes' rule is presented as

$$p(\theta|Z_t) = p(\theta|z_t, Z_{t-1}) \propto p(z_t|\theta, Z_{t-1})p(\theta|Z_{t-1}) \tag{5}$$

where $p(z_t|\theta, Z_{t-1})$ is the likelihood function of parameters and $p(\theta|Z_{t-1})$ is the prior distribution of parameters provided by observation up to time step $t - 1$. When an initial probability of parameters $p(\theta_0)$ is assigned, given a set of observations, $z_{1k} = \{z_1,z_2,\dots z_k\}$, the parameters,$\theta_k$, can be calculated from SMC simulation, which is discussed in the next section.

## SEQUENTIAL MONTE CARLO SIMULATION

A general nonlinear stochastic dynamic system can be represented by two equations: the state equation describes the evolution of state variables in time, while the measurement equation represents how the output variables are generated from model state variables. In this study, the state transition equation is used to describe the evolution of ANN model parameters in time, while the measurement equation describes the nonlinear relation between the input and output variables, that is:

$$\text{State equation:} \quad \theta_k = f(\theta_{k-1}, u_{k-1}) = \theta_{k-1} + u_{k-1} \tag{6}$$

$$\text{Measurement equation:} \quad z_k = h(\theta_k, x_k, v_k) \tag{7}$$

where $z_k \in R^1$ denotes the output measurement at time step $k$; $x_k \in R^d$ are the input measurements; $\theta_k = \{\bar{\phi}, \vec{\varphi}\} \in R^m$ are the parameter sets; $u_{k-1}$ and $v_k$ are the process and measurement noise. The measurement equation $h(\cdot)$ is a nonlinear mapping function approximated by $MLP(n_0, n_1, 1)$. The process states (parameters) are considered as "time-evolving." That is, at each time step, the artificial evolution provides a mechanism for new parameters, which are reweighted at the posterior step (Gordon *et al.* 1993; Liu & West 2001).

In this study, an SMC sampling method is used to calculate the posterior distribution of the model parameters (de Freitas *et al.* 2000; Doucet *et al.* 2000; Arulampalam *et al.* 2002; Doucet & Tadic 2003; Moradkhani *et al.* 2005). Through effective sampling strategy, the posterior probability distribution of parameters and model estimates can be estimated recursively using Bayes' rule. Assume that the posterior distribution of model parameters at time $t = k$ is presented by $N$ samples:

$$p(\theta_k|Z_k) = \frac{1}{N}\sum_{i=1}^{N}\delta(\theta_k - \theta_k^i) \tag{8}$$

where $\delta(\cdot)$ is the Dirac delta function located at parameter samples or particles $\left(\theta_k^i\right)$. The computation of an expectation is

$$E[g(\theta_k)] = \int g(\theta_k)p(\theta_k|Z_k)\mathrm{d}\theta_k = \frac{1}{N}\sum_{i=1}^{N}g\left(\theta_k^i\right) \tag{9}$$

Because the true distribution of $p(\theta|Z_k)$ is not available, it can be approximated by drawing random samples from a distribution, $\pi(\theta_k)$, also known as an "importance function". The above expectation can be approximated by

$$E[g(\theta_k)] = \frac{1}{N}\sum_{i=1}^{N}w_k^j g(\theta_k^i) \tag{10}$$

where $w_k^i \propto \frac{p(\theta_k|Z_k)}{\pi(\theta_k|Z_k)}$ and $\sum_{i=1}^{N}w_k^i = 1$ The updated weight can be obtained according to

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|\theta_k^i)p(\theta_k^i|\theta_{k-1}^i)}{\pi(\theta_k|Z_k)} \tag{11}$$

The quality of approximation is relevant to the selection of the important function $\pi(\theta_k)$. One popular importance function used is given the transitional prior:$p(\theta_k^i|\theta_{k-1}^i)$ (Arulampalam *et al.* 2002), and therefore the updated weights can be represented as

$$w_k^i \propto w_{k-1}^i p(z_k|\theta_k^i) \tag{12}$$

The sequential important sampling (SIS) algorithm can be used to take samples from the importance function and then compute the weights of $w_t^i$ according to Equation (12). However, there is a problem with SIS. As the iteration in time continued, the sample weights degenerated rapidly (i.e. only a few samples that are assigned with high weights). This requires an effective resampling strategy to replace those samples with very small weights by those samples with larger weights.

## Resampling

The SIS filter with resampling option is usually referred as SIR (Sampling Importance Resampling) filter. When the SIS filter is running into significant degeneracy, resampling is needed to redistribute the existing samples. Resampling mainly removes those samples with low importance weights

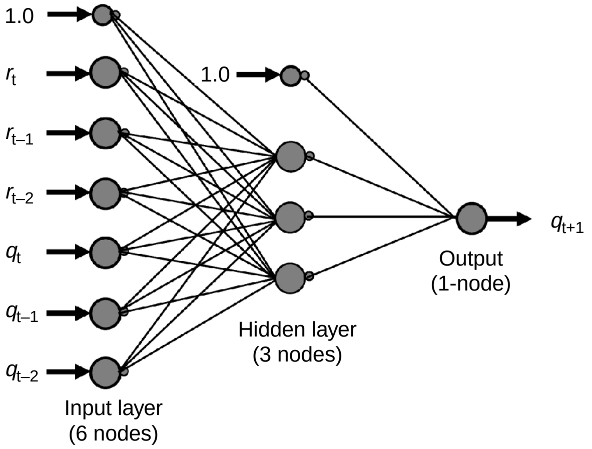

**Figure 2** | Configuration of a three-layer perceptron MLP(6,3,1) for streamflow forecasting.

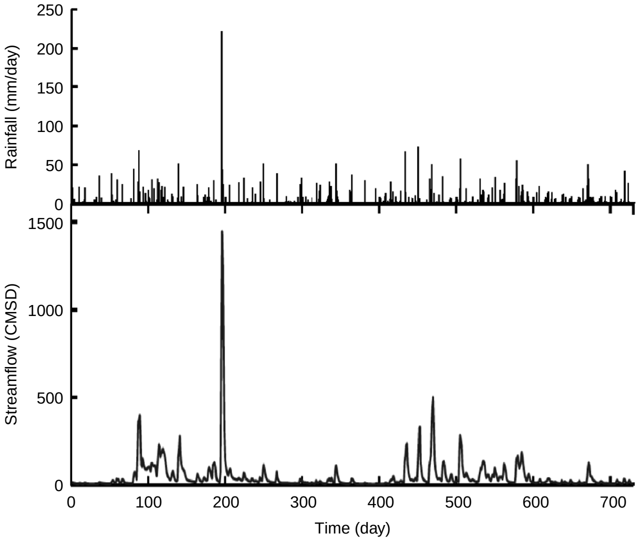

**Figure 3** | Daily rainfall and streamflow time series used in the model simulation.

(low probability) and then assigns more samples to those with high important weights (high probability). The new samples ($w_k^{i*}$) are with uniform weights ($1/N$), that is to assign a number of repeating samples which are proportional to the important weights $w_k^i$ (Arulampalam *et al.* 2002). An easy way to do resampling is to calculate the number of samples to be repeated by $n^i = N{\cdot}w_k^i$ (Kong *et al.* 1994). In order to know when to give resampling, a criterion is provided to evaluate the degeneracy of the SIS filter (Doucet *et al.* 2000; Arulampalam *et al.* 2002). A measure of degeneracy of the algorithm is dependent on the effective sample size $N_{\text{eff}}$, which is defined as

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N}\left(w_k^i\right)^2} \tag{13}$$

A small $N_{\text{eff}}$ indicates severe degeneracy. Resampling should be performed when the effective sample size $N_{\text{eff}}$ is below a fixed heuristic threshold. Adopted from Arulampalam *et al.* (2002), the resampling algorithm and an iteration of SIR filter is listed in the appendix.

## CASE STUDY: WATERSHED STREAMFLOW FORECASTING

The process of watershed runoff generated from rainfall is identified as very complex and is influenced by many factors, including the temporal and spatial distribution of rainfall, the topographic and soil characteristics of the watershed, and the mechanism by which water enters into long-term groundwater storage. Based on many previous studies, ANN models can be used as an alternative to the physically based hydrologic models in streamflow prediction (Hsu *et al.* 1995, 2002; Govindaraju & Rao 2000; Maier & Dandy 2000; Coulibaly *et al.* 2001; Abrahart & See 2002; Moradkhani *et al.* 2004; Dawson *et al.* 2006). In this study, the SMC is use in the ANN model parameter estimation and streamflow forecasts.

Based on a previous study (Hsu *et al.* 2002), an ANN model with a structure of MLP(6,3,1) is suitable for one-day-ahead streamflow prediction of this watershed. As shown in Figure 2, the variables connecting to the inputs of MLP(6,3,1) are the time-lagged daily rainfall $r(t-\tau)$ and streamflow $q(t-\tau)$, that is: $\vec{x} = [x_1, x_2, \ldots, x_6] = [r_t, r_{t-1},$

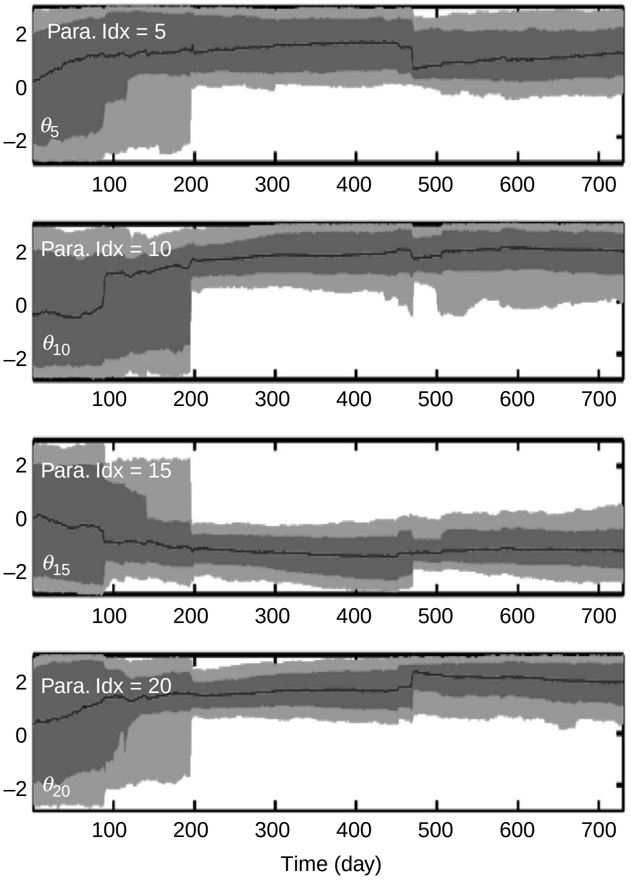

**Figure 4** | The mean (solid line) and uncertainty bounds at 75% (dark gray) and 95% (light gray) confidence interval of model parameters at each computation time step. (only four parameters [$\theta(5)_k$, $\theta(10)_k$, $\theta(15)_k$, $\theta(20)_k$] are displayed).

$r_{t-2}$, $q_{t,}$ $q_{t-1}$, $q_{t-2}$], while the output variable, $z = [q_{t+1}]$, is the one-day-ahead streamflow prediction. In total, the MLP(6,3,1) contains 25 parameters and they are presented as $\theta_k = [\vec{v}_k, \vec{w}_k] = [\theta(1)_k, \theta(2)_k, \ldots, \theta(25)_k]$. Two years of rainfall–runoff data from the Leaf River Basin, Mississippi, were used in the experiment (see Figure 3).

## SMC simulation

In the simulation, the parameters of MLP(6,3,1) were confined in the range of [−3, 3]. At the beginning of the simulation ($k = 0$), because the prior information of parameters is not known, uniform distributions are assigned to the parameters. The total sample size ($N$) is set to 500 for the SMC simulation. At each simulation time step, the transient distribution of ANN model parameters and streamflow estimates are presented by the distribution of all samples.

Figure 4 shows the time evolution of four MLP(6,3,1) parameters, [$\theta(5)k$,$\theta(10)k$,$\theta(15)_k$,$\theta(20)_k$] for time steps from $k = 1$ to $k = 730$. The time evolution of parameters during the sequential training period is shown by the mean value (solid line), 75% (dark gray) and 95% (light gray) confidence intervals of the parameters. As shown in Figure 4, the uncertainty bounds (95% confidence intervals) are widely extended to the range near [−3 3] when the simulation was initiated (see the time steps from 0–50 d). As observations are continuously available for the training of parameters, the uncertainty bounds, in terms of 95% and 75% confidence intervals, are gradually reduced. Around the time steps 80–150 d, a progressive changing of parameters and their uncertainty bounds were observed. During this time period, rainfall frequency and intensity increased, and the streamflow response of the watershed is also increased. The overall 95% confidence range of parameters is still widely extended to the range of [−3 3]: however, the 75% confidence intervals of parameters are moderately changing to more narrow ranges (e.g. see $\theta(5)_k$, $\theta(15)_k$ and $\theta(20)_k$). Some parameters (such as $\theta(10)_k$) still remain large uncertainty bounds. The most significant variation of parameters is near the time steps of 190–210 d where the highest rainfall and streamflow were observed. After the time step at 200 d, the probability distributions of model parameters were relatively stabilized, and little variation of parameters were found around time steps 450–500 d.

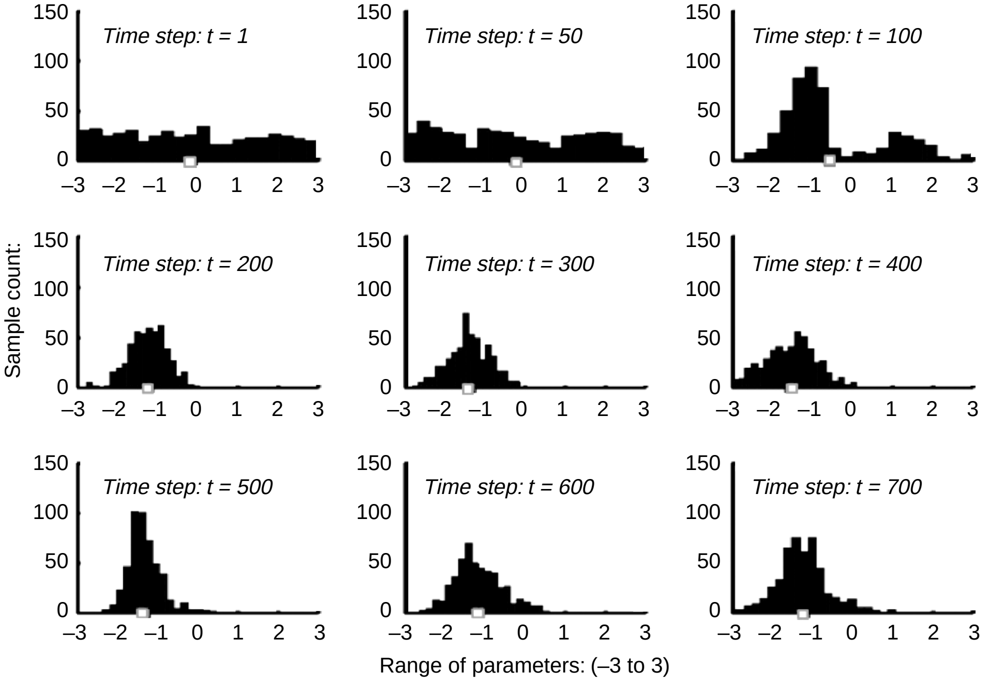

**Figure 5** | Distribution and mean of parameter $\theta(15)_k$ at $t = 1$, 50, 100, 200, 300, 400, 500, 600 and 700.

Figure 5 shows the sample distributions of $\theta(15)_k$ at time steps $t = 1$, 50, 100, 200, 300, 400, 500, 600 and 700. In the plot, the $x$ axis is for parameters in the range of $[-3\ 3]$, while the $y$ axis is the sample count. The probability distribution of the parameters can be estimated by dividing the discrete samples by the total sample count (i.e. $N = 500$ in this case).

As shown in Figure 5, from $t = 1$ to 100, the parameter distribution is evolved from uniform distribution to multimodal distribution (two peaks), with the highest probability located in the negative parameter range and two peaks near $-1$ and 1, respectively. Following $t = \sim 200$ and later, the distribution is stabilized to the unimodal distribution.

Figures 6 and 7 show the 75% and 95% confidence interval of all 25 parameters at simulation time steps: $t = 10$, 50, 100, 200, 300, 400, 500 and 600. They show a significant change of the model uncertainty bound at time steps from $t = 50$ to $t = 200$. After time steps $t = 200$ (see $t = 200$ to $t = 400$), the change of parameters is minor.

Figure 8 shows the streamflow prediction and its uncertainty bound at the second simulation year. The figure shows that the 75% and 95% confidence intervals cover the observations reasonably well. Meanwhile, the uncertainty bound is relatively lower for the low-flow period and larger during the high-flow period. The expected values of flow estimates fit well to the observations (see the solid line for model estimation and the 'o' as observation).

Finally, the scatterplot of the observed and predicted streamflow (expected value) is displayed in Figure 9. Although the 95% uncertainty bounds are relatively higher for the high flows (>200 cmsd), the expected values of model estimates fit well with the observed streamflow. The medium streamflow (see $50 < q_t < 200$ cmsd) shows higher variation from several underestimation points. Model estimation at the low-flow range is excellent. The evaluation statistics of model estimation based on the mean value is also listed in Figure 9. The root mean square error (RMSE), correlation coefficient (CORR) and bias (BIAS) are 15.237 cmsd, 0.971 and $-0.985$ cmsd, respectively.

For further comparison of SMC with the convectional batch training approach, the MLP(6,3,1) model was trained from the full two years of data using the LM algorithm and then evaluated in the same SMC testing period. The scatterplot of the validation period using the LM algorithm

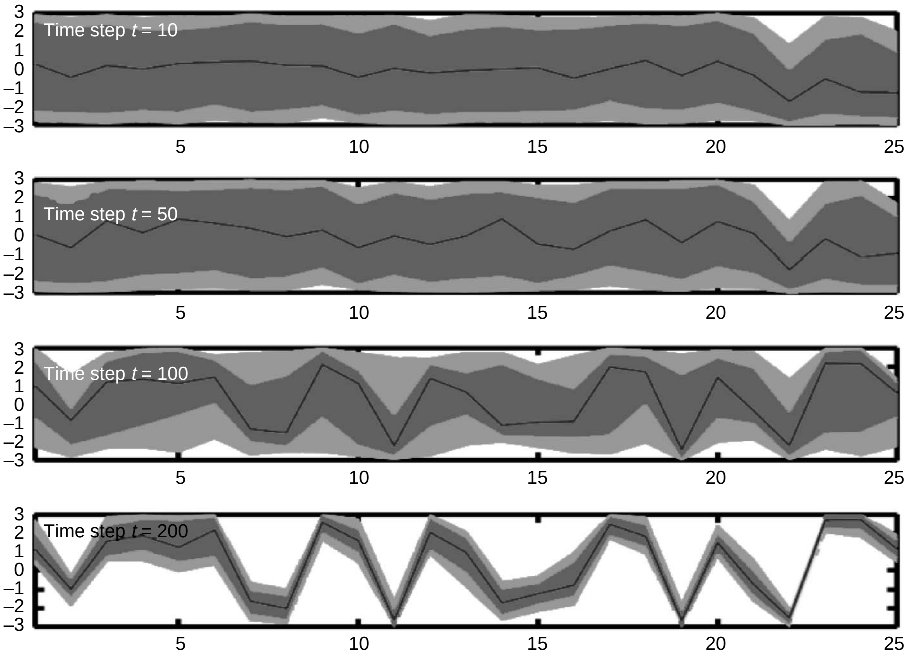

**Figure 6** | The mean (solid line) and 75% (dark gray) and 95% (light gray) uncertainty bound of parameters at $t = 10$, 50, 100, 200, 300 and 400. All 25 MLP(6,3,1) parameters, as indexed 1–25 listed in the $x$ axis, are displayed.

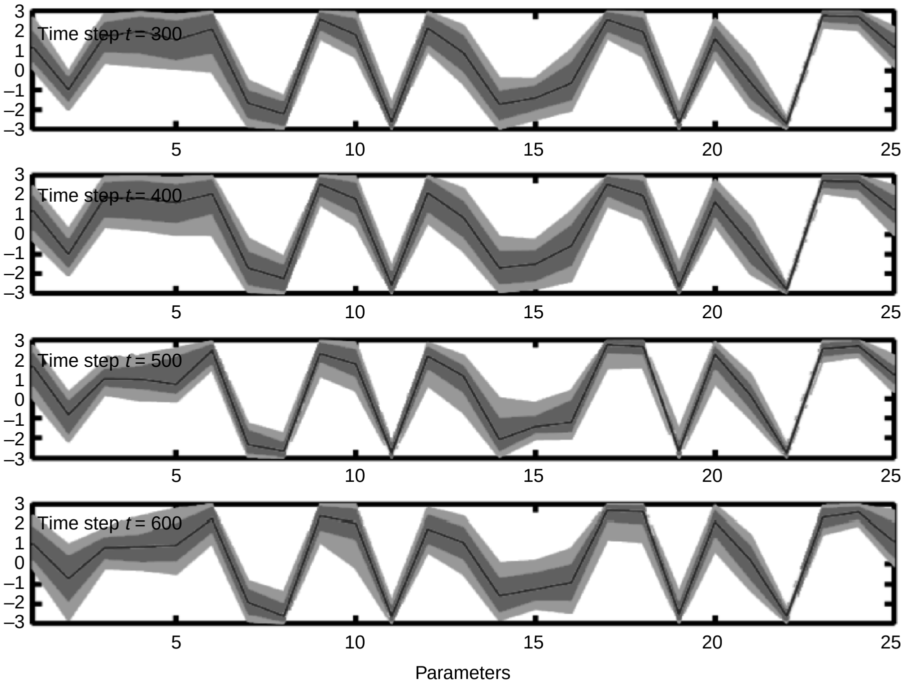

**Figure 7** | Same as Figure 6, but for the time step $t = 300$, 400, 500 and 600.

is listed in Figure 10. The MLP(6,3,1) estimates trained by the LM algorithm fit very well with the observations along the 1:1 line with [RMSE, CORR and BIAS] = [16.81 cmsd, 0.96 and 0.89 cmsd], respectively. Compared with conventional fixed-parameter estimation, the SMC simulation gives better statistics on RMSE and CORR, but gives worse values on BIAS.

While the test results show the SMC simulation performs better than the conventional LM batch simulation on RMSE and CORR, the implementation of these two approaches is relatively different. That is, the observation data in the testing period were fully covered in the LM batch training, but only progressively included in the SMC training.

If only one-year data were used for LM batch training, the evaluation statistics will be much worse than training using all of the available data. In this case, the evaluation statistics on the second-year period based on the LM algorithm are changed to [RMSE, CORR and BIAS] = [22.2 cmsd, 0.94 and 2.65 cmsd].

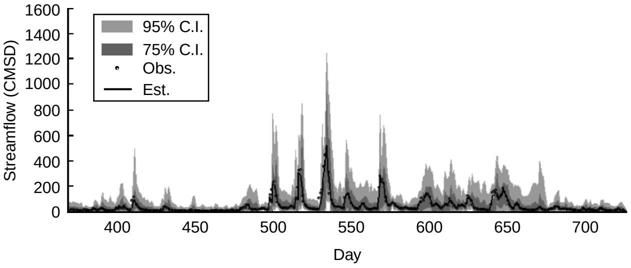

**Figure 8** | The 75% (dark gray) and 95% (light gray) confidence intervals and the expected value (black line) of streamflow prediction of MLP(6,3,1). The observed streamflow is marked by circles ('o').

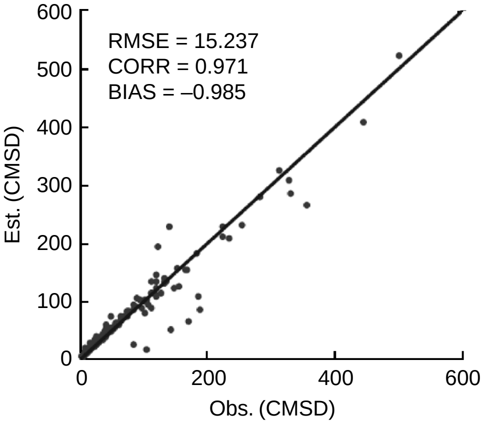

**Figure 9** | The scattterplot of observed and expected values of MLP(6,3,1) estimation.

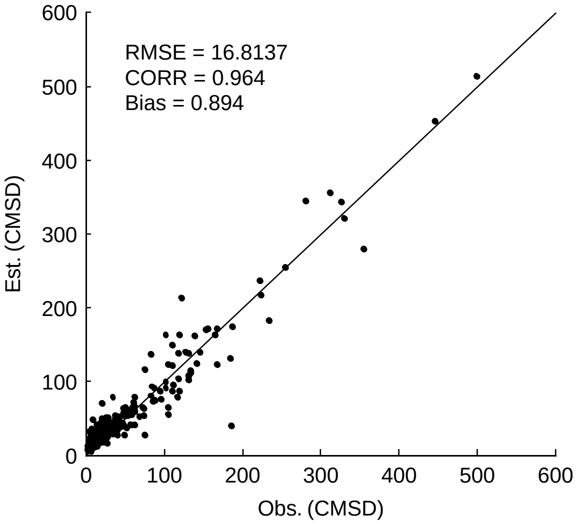

**Figure 10** | Simulation of streamflow estimation based on MLP(6,3,1) trained by LM algorithm.

## CONCLUSIONS

In the literature, SMC has been applied to many practical applications for tracking the state variables and parameters of time-varying systems. In this study, an SMC approach based on SIR is used to train an ANN model for streamflow forecasting. Although the computational cost of SMC approaches can be relatively higher than those convectional batch approaches, the SMC approach not only gives better streamflow forecasting but also provides the transitional probability distribution of parameters and states, which are very useful for quantifying the uncertainty of the parameters and forecasts. In this study, an arbitrary sample size ($N = 500$) and limited data (2 years) were used in the evaluation. Studies in the selection of effective sample size and uncertainty analysis of model behavior through SMC simulation are ongoing and will be reported in the near future.

## ACKNOWLEDGEMENTS

Partial financial support of this study was made available through research grants from the NASA NEWS (grant no. NNX06AF93G), NOAA CPPA (grant no. NA04OAR431 0086) and NSF SAHRA (grant no. EAR 9876800) programs. The author would also like to thank Diane Hohnbaum for her editing of this manuscript.

## REFERENCES

Abrahart, R. J. & See, L. 2002 Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments. *Hydrol. Earth Syst. Sci.* **6** (4), 655–670.

Arulampalam, M. S., Maskell, S., Gordon, N. & Clapp, T. 2002 A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **50** (2), 174–188.

Bishop, C. 1995 *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford.

Breiman, L. 1996 Bagging predictors. *Machine Learning* **24** (2), 123–140.

Coulibaly, P., Anctil, F., Aravena, R. & Bobee, B. 2001 Artificial neural network modeling of water table depth fluctuations. *Water Resour. Res.* **37** (4), 885–896.

Coulibaly, P., Anctil, F. & Bobee, B. 2000 Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *J. Hydrol.* **230** (3–4), 244–257.

Coulibaly, P. & Baldwin, C. K. 2005 Nonstationary hydrological time series forecasting using nonlinear dynamic methods. *J. Hydrol.* **307** (1–4), 164–174.

Coulibaly, P., Hache, M., Fortin, V. & Bobee, B. 2005 Improving daily reservoir inflow forecasts with model combination. *J. Hydrol. Eng.* **10** (2), 91–99.

Dawson, C. W., Abrahart, R. J., Shamseldin, A. Y. & Wilby, R. L. 2006 Flood estimation at ungauged sites using artificial neural networks. *J. Hydrol.* **319** (1–4), 391–409.

Dawson, C. W. & Wilby, R. L. 2001 Hydrological modelling using artificial neural networks. *Prog. Phys. Geogr.* **25** (1), 80–108.

de Freitas, J. F. G., Niranjan, M., Gee, A. H. & Doucet, A. 2000 Sequential Monte Carlo methods to train neural network models. *Neural Comput.* **12** (4), 955–993.

Doucet, A., Godsill, S. & Andrieu, C. 2000 On sequential Monte Carlo sampling methods for Bayesian filtering. *Statist. Comput.* **10** (3), 197–208.

Doucet, A. & Tadic, V. B. 2003 Parameter estimation in general state-space models using particle methods. *Ann. Inst. Statist. Math.* **55** (2), 409–422.

Gallant, A. R. & White, H. 1992 On learning the derivatives of an unknown mapping with multilayer feedforward networks. *Neural Networks* **5** (1), 129–138.

Gordon, N. J., Salmond, D. J. & Smith, A. F. M. 1993 Novel approach to nonlinear non-gaussian bayesian state estimation. *IEE Proc. F Radar Signal Process.* **140** (2), 107–113.

Govindaraju, R. S. & Rao, A. R. 2000 *Artificial Neural Networks in Hydrology*. Kluwer, Dordrecht.

Gupta, H. V., Hsu, K. L. & Sorooshian, S. 1997 Superior training of artificial neural networks using weight-space partitioning. In: *Int. Conf. on Neural Networks, Houston, TX*, pp. 1919–1923.

Haykin, S. S. 1999 *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Englewood Cliffs, NJ.

Hong, Y., Hsu, K. L., Sorooshian, S. & Gao, X. G. 2004 Precipitation estimation from remotely sensed imagery using an artificial neural network cloud classification system. *J. Appl. Meteorol.* **43** (12), 1834–1852.

Hornik, K., Stinchcombe, M. & White, H. 1990 Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks* **3** (5), 551–560.

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

Hsu, K. L., Gupta, H. V., Gao, X. G., Sorooshian, S. & Imam, B. 2002 Self-organizing linear output map (SOLO): an artificial neural network suitable for hydrologic modeling and analysis. *Water Resour. Res.* **38** (12), 1–17.

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

Khan, M. S. & Coulibaly, P. 2006 Bayesian neural network for rainfall-runoff modeling. *Water Resour. Res.* **42** (7), W07409 doi:10.1029/2005WR003971.

Kingston, G. B., Lambert, M. F. & Maier, H. R. 2005 Bayesian training of artificial neural networks used for water resources modeling. *Water Resour Res.* **41** (12), W12409 doi:10.1029/2005WR004152.

Kong, A., Liu, J. S. & Wong, W. H. 1994 Sequential imputations and Bayesian missing data problems. *J. Am. Statist. Assoc.* **89** (425), 278–288.

Liu, J. & West, M. 2001 Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo Methods in Practice* (ed. A. Doucet, N. d. Freitas & N. Gordon), pp. 197–224. Springer-Verlag, New York.

Mackay, D. J. C. 1992 A practical bayesian framework for backpropagation networks. *Neural Comput.* **4** (3), 448–472.

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

Maier, H. R. & Dandy, G. C. 2001 Neural network based modelling of environmental variables: a systematic approach. *Math. Comput. Modell.* **33** (6–7), 669–682.

Moradkhani, H., Hsu, K., Gupta, H. V. & Sorooshian, S. 2004 Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. *J. Hydrol.* **295** (1–4), 246–262.

Moradkhani, H., Hsu, K. L., Gupta, H. & Sorooshian, S. 2005 Uncertainty assessment of hydrologic model states and parameters: sequential data assimilation using the particle filter. *Water Resour. Res.* **41** (5), 1–17.

Neal, R. M. 1996 *Bayesian Learning for Neural Networks*. Springer, New York.

Parasuraman, K., Elshorbagy, A. & Carey, S. K. 2006*a* Spiking modular neural networks: a neural network modeling approach for hydrological processes. *Water Resour. Res.* **42** (5), W05412 doi:10.1029/2005WR00431.

Parasuraman, K., Elshorbagy, A. & Si, B. C. 2006*b* Estimating saturated hydraulic conductivity in spatially variable fields using neural network ensembles. *Soil Sci. Soc. Am. J.* **70** (6), 1851–1859.

Schalkoff, R. J. 1997 *Artificial Neural Networks*. McGraw-Hill, New York.

Schapire, R. E. 1990 The strength of weak learnability. *Machine Learning* **5** (2), 197–227.

Shu, C. & Burn, D. H. 2004 Artificial neural network ensembles and their application in pooled flood frequency analysis. *Water Resour. Res.* **40** (9), W09301 doi:10.1029/2003WR002816.

First received 7 May 2008; accepted in revised form 20 August 2009. Available online 2 April 2010

## APPENDIX

Adopted from Arulampalam *et al.* (2002), the systemic resample algorithm and an iteration of SIR filter is provided in Algorithms #1 and #2 below:

**Algoritham 1** | Resampling algorithm

```
$\left[\left\{\theta_k^i, w_k^i, i^j\right\}_{i=1}^N\right] = RESAMPLE\left[\left\{\theta_k^i, w_k^i\right\}_{I=1}^J\right]$
•  Initialized the CDF: $c_1 = 0$
•  FOR $i = 2: N$
   -  Construct CDF: $c_i = c_{i-1} + w_k^i$
•  END FOR
•  Start at the bottom of the CDF: $i = 1$
•  Drawing a starting point from uniform distribution: $u_1 \sim U[0, N^{-1}]$
•  FOR $j = 1: N$
   -  Move along the CDF: $u_j = u_1 + N^{-1}(j - 1)$
   -  WHILE $u_j > c_i$
         $i = i + 1$
   -  END WHILE
   -  Assign sample: $\theta_k^{j*} = \theta_k^i$
   -  Assign weight: $w_k^{j*} = N^{-1}$
   -  Assign parent: $i^j = i$
•  END FOR
```

CDF = cumulative distribution function.

**Algorithm 2** | SIR filter

```
$\left[\left\{\theta_k^i, w_k^i\right\}_{i=1}^N\right] = SIR\left[\left\{\theta_{k-1}^i, w_{k-1}^i\right\}_{i=1}^N, z_k\right]$
•  FOR $i = 1:N$
   -  Draw $\theta_k^i \sim p\left(\theta_t^i | \theta_{t-1}^i\right)$
   -  Calculate $w_k^i = p\left(z_k | \theta_k^i\right)$
•  END FOR
•  Calculate total weight:$s = SUM\left[\left\{w_k^i\right\}_{i=1}^N\right]$
•  FOR $i = 1:N$
   -  Normalized $w_k^i = s^{-i} w_k^i$
•  END FOR
•  Resample using Algorithm 2 below:
   -  $\left[\left\{\theta_k^i, w_k^i, -\right\}_{i=1}^N\right] = RESAMPLE\left[\left\{\theta_k^i, w_k^i\right\}_{i=1}^j\right]$
```