Hindawi Publishing Corporation
Journal of Computational Environmental Sciences
Volume 2014, Article ID 290127, 6 pages
http://dx.doi.org/10.1155/2014/290127

Research Article

Comparison of Back Propagation Neural Network and Genetic Algorithm Neural Network for Stream Flow Prediction

C. Chandre Gowda and S. G. Mayya

Department of Applied Mechanics and Hydraulics, NITK Surathkal, Mangalore 575025, India

Correspondence should be addressed to C. Chandre Gowda; chandregowdac@gmail.com

Received 10 July 2014; Revised 7 August 2014; Accepted 11 August 2014; Published 28 August 2014

Academic Editor: Alberto Campisano

Comparison of stream flow prediction models has been presented. Stream flow prediction model was developed using typical back propagation neural network (BPNN) and genetic algorithm coupled with neural network (GANN). The study uses daily data from Nethravathi River basin (Karnataka, India). The study demonstrates the prediction ability of GANN. The statistical tests show that GANN model performs much better when compared to BPNN model.

1. Introduction

Stream flow prediction for a river has been one of the most explored areas of research during recent days. Predicting the flow may facilitate its monitoring. Prediction of stream flows with good probability and reliability is of great concern. Precise prediction of stream flow gives a clear picture of the available water resources. It may also facilitate improved planning and optimum utilization of water. Many factors influence stream flow such as catchment characteristics and geographical and meteorological factors. Stream flow models may show high nonlinearity. From the second half of the last century, different methods such as physical, empirical, and numerical methods and other hybrid black box models have been practised for stream flow prediction. Main drawbacks observed in using physical model are the requirement of a more accurate and large data set which is tedious to acquire. The black box models may have an advantage at this context as they require minimum data and may provide satisfactory results. Neural network (NN), genetic algorithm, and fuzzy and hybrid algorithms are some of the methods which have received lots of attention among all modelling techniques during recent decades.

The potential of NN had already been demonstrated in the context of river flow [1, 2] and dissolution kinetics [3] emphasizing the prediction ability of NN models. NN models were capable of reconstructing rainfall runoff relationships [4]. NN has proven an alternative to conventional rainfall runoff models and its strength in adaptive learning was shown for flow forecasting in the study [5]. Probabilistic forecasting accuracy was achieved using NN [6]. Modelling potential of NN was compared to a physical model and it was proven that NN has good prediction capability [7]. Good prediction accuracy and flexibility of NN were demonstrated in the studies [8, 9]. The ability of neuroemulation to imitate the behaviour of real cases and capture nonlinearity has made it a suitable method for modelling. Back propagation learning algorithm using gradient (steepest descent) based approach is widely used in the neural network training. The training of NN is done by minimizing the error function (Mean Square Error or Root Mean Square Error) between the predicted and the observed value. However, a back propagation learning algorithm with gradient based approach in neural network training has numerous drawbacks such as the fact that performance depends on initial weights and that the likelihood of solution reaching global optimum is not assured. In order to overcome these limitations, it is essential to develop an efficient method to optimize the NN. Genetic algorithm (GA) has been successfully employed in overcoming the limitations of back propagation learning

algorithm in recent investigations [10, 11]. Thus, this study shows the implementation of GA into a neural network for stream flow prediction.

In the present study, two models, back propagation neural network (BPNN) and genetic algorithm neural network (GANN), are developed and compared in predicting stream flow in natural rivers. BPNN was trained using the steepest descent method to optimize connecting weights for fixed network parameters. BPNN architecture parameters (number of neurons in hidden layer, bias, momentum, and learning rate) are obtained by trial and error. In GANN model, genetic algorithm was used to optimize both neural network (NN) parameters and connecting weights, which has not been attempted in the previous studies.

2. Methodology

2.1. Neural Network. Neural network (NN) models are parallel computing networks inspired by animal nervous system. They are adopted more commonly for forecasting and prediction in many fields. A neural network typically consists of input layer (with "n" input neurons), one or many hidden layers (with "m" number of hidden layers and "o" number of hidden neurons), and an output layer (with "p" number of output neurons). The neural network vectors are shown below:

$$\underbrace{\begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{pmatrix}}_{\text{input layer}} \underbrace{\begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1m} \\ h_{21} & h_{22} & \cdots & h_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ h_{o1} & h_{o2} & \cdots & h_{om} \end{pmatrix}}_{\text{hidden layer}} \underbrace{\begin{pmatrix} o_1 \\ o_2 \\ \vdots \\ o_p \end{pmatrix}}_{\text{output layer}}. \quad (1)$$

Each layer will be interconnected with the weights (randomly generated). The information has to be feed-forwarded from each input neuron to all hidden neurons through these weights. Then, information processes use transfer function (linear or sigmoid) at each hidden neuron. Then, all the processed values have to be summed up at each hidden neuron and information to be passed on to the output neuron through connecting weights. Then again, the information is to be processed through transfer function at output neuron to get final value. Bias is considered in order to eliminate or offset the dominant solutions at hidden layer and at output layer. The whole process of feed-forward from input layer to output layer is termed as feed-forward process.

The final observed value at the output layer is compared with the target value. The difference in error between the observed and predicted value is then evaluated. Then, a back propagation process is used to back-propagate errors until the weights are optimized to obtain minimum error between the observed and predicted value. In back propagation, partial derivatives with respect to the connected weights are calculated. Chain rule is used to get the updated weights [12].

The updating continues until the stopping criteria are met (for thousand iterations or minimum difference in error is obtained). Different learning techniques, like steepest descent method, Scaled Conjugate, Levenberg Marquardt, and others, are available. Learning rate accelerates the learning process and momentum pushes the solution towards convergence. Minimization of (MSE, RMSE) error is considered as the objective of the neural network.

In the current study, neural network with an input layer (with 5 input neurons), single hidden layer (with 10 hidden neurons and tan sigmoid as activation function), and output layer (with 1 hidden neuron and tan sigmoid as activation function) was adopted. Bias, learning rate, momentum, activation constant, and number of hidden neurons were considered as the parameters of neural network architecture. The optimal value of network parameters was selected by trial and error. The best combination of the network parameters was used to optimize the network weights. The error between the predicted and the observed value was computed from (2). Steepest descent method was adopted as a learning technique to optimize the weights. The model was termed as back propagation neural network (BPNN) in the study:

$$E = \frac{1}{2}(T_o - O_o)^2, \quad (2)$$

where T_o = predicted output, O_o = observed output, and E = error function.

2.2. Genetic Algorithm. Genetic algorithm (GA) is a heuristic search technique that works on the principle of natural genetics and natural selection [13]. It has been proven that genetic algorithms are able to find the global optimum solution in many research problems. The working procedure of GA usually starts with random strings representing design or decision variables. Later, each string is evaluated (checking objective and constraint conditions) to allocate the fitness value. Then termination condition is verified in the algorithm. In case if termination criterion is not met, then population has to be operated by the crossover, reproduction, and mutation functions. These three functions are used to create a new population. The new population is then evaluated and tested for fitness function. Reproduction duplicates the good strings. Roulette wheel, rank selection, and tournament selection are the three types of reproduction operator (in the study, rank selection has been adopted). Crossover operation creates new strings. Mutation operator takes care of diversity (to avoid the trapping of the good strings) in the population. The iterative operation is continued till the last generation in the population or till the desired solution is obtained.

In the current study, genetic algorithm was used to optimize network parameters (bias, learning rate, momentum, activation constant, and number of hidden neurons) and weights in neural network algorithm. The GA parameters (mutation probability, number of generations, and number of populations) were selected by trial-and-error method. This method was named as genetic algorithm neural network (GANN) in the study.

3. Study Area

Nethravathi River basin is situated in Karnataka, India. It is located between 74° 45′ E and 75° 45′ E longitude and 12° 30′ N and 13° 10′ N latitude on Western Ghats (Figure 1).

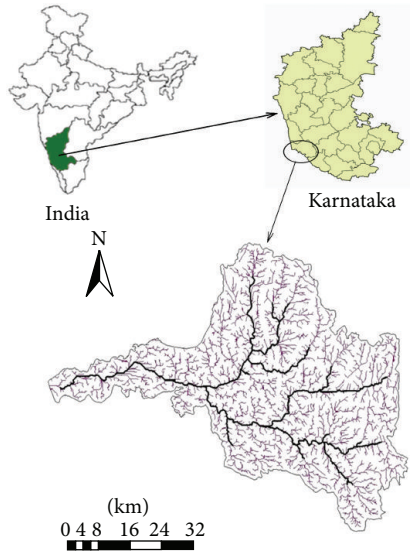

Figure 1: Nethravathi River basin in Karnataka, India.

Catchment stretches around 3184 km^2. The annual rainfall over the area varies between 1500 mm and 4500 mm, receives rainfall mainly during monsoon months (June to September), and continues till November. The daily rainfall data and stream flow data used in the study are obtained from Indian Meteorological Department (IMD) and Central Water Commission (CWC).

Twelve rain gauge stations in the Nethravathi River basin were selected and their corresponding Thiessen weights were found. Since the rainfall in nonmonsoon periods in the river basin is zero, only the monsoon days are considered. Lag time, precipitation P_t, P_{t-1}, and P_{t-2}, and runoff Q_{t-1}, Q_{t-2} were considered as the input for modelling purposes. Evaporation and base flow were not considered in the analysis. The current runoff Q_t was considered as the output model variable, with t being current time period, $t - 1$ being lag of 1 day, and $t - 2$ being lag of 2 days. The inputs were selected by partial autocorrelation analysis, which showed good correlation values up to two days' lag.

The daily rainfall and daily runoff data were used for modelling. 80% of the data was used for training and 20% of the data for testing. The stream flow and rainfall data were normalized in the range from 0.1 to 0.9 from

$$x_s = 0.1 + 0.8\left(\frac{x_i}{x_{\max}}\right), \tag{3}$$

where x_s is normalized value of x_i, x_i is the observed value, and $x_{\max}$ is the maximum value of the data set used.

4. Prediction Model

Daily stream flow modelling was carried out using back propagation neural network (BPNN) and genetic algorithm neural network (GANN). The parameters of BPNN architecture were number of neurons (at hidden layer), learning rate (at hidden layer and output layer), bias, momentum or alpha (at hidden layer and output layer), and activation constant (at hidden layer and output layer), as shown in Table 1. BPNN architecture was selected by trial and error. After selecting the best suitable architecture, network was simulated to update weights. Steepest descent method was adopted to train the network and to optimize the weights in the BPNN model. The parameters selected in GANN model are mutation probability, population size, and number of generations, shown in Table 2. In GANN, the genetic algorithm parameters were also selected by trial and error. In the study, the adopted neural network consists of an input layer (with 5 input neurons), single hidden layer (with 10 hidden neurons), and output layer (with one output neuron), shown in Figure 2.

In genetic algorithm, neural network model (GANN) genetic algorithm was adopted to optimize the weights and neural network parameters. The program was written in C++ language for BPNN and GANN (binary coded genetic algorithm integrated with neural network).

The GANN and BPNN performances were compared. Nash Sutcliffe efficiency (NS), coefficient of determination (R^2), Mean Absolute Percentage error (MAPE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) were used to check the performances of the models.

Table 1: Parameter range of NN architecture.

Parameter	Range		Optimum (from trial and error)
	Minimum	Maximum	
Number of hidden neurons	2.0	25.00	10.00
Learning rate at hidden layer	0.1	0.99	00.500
Learning rate at output layer	0.1	0.99	00.550
Momentum rate	0.1	0.99	00.445
Hidden layer activation constant	1.0	10.00	05.500
Output layer activation constant	1.0	10.00	05.500
Bias	0.000001	0.00001	0.000085

Table 2: Parameter range of GANN.

Parameter	Range		Optimum (from trial and error)
	Minimum	Maximum	
Mutation probability	0.00001515	0.0001515	0.0000424
Number of population	50	300	190
Number of Generations	100	500	210

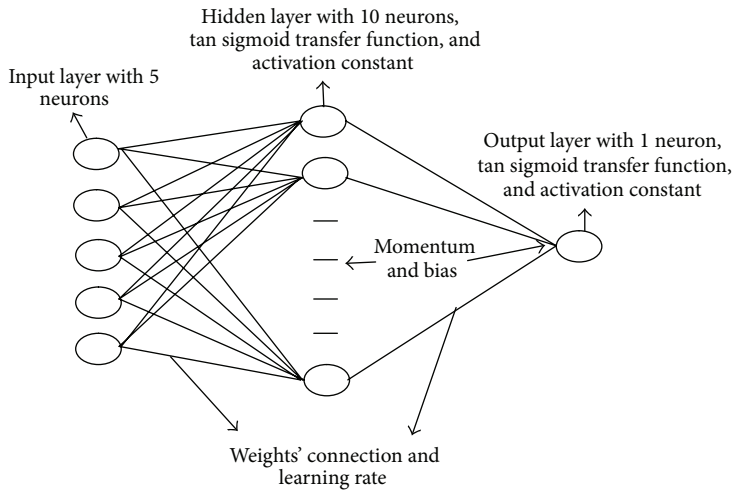

Figure 2: Neural network architecture.

Table 3: Efficiency for test cases of the models developed.

	GANN	BPNN
Nash Sutcliffe (NS)	0.847	0.815
Coefficient of determination (R^2)	0.901	0.881
Mean Absolute Error (MAE) m^3/s	167	182
Root Mean Square Error (RMSE) m^3/s	220	242
Mean Absolute Percentage Error (MAPE) %	15.68	17.29

5. Results and Discussion

One day ahead stream flow prediction model was developed using BPNN and GANN. Rainfall lag time and stream flow lag time (i.e., one-day lag and two-day lag) were used as input to predict one day ahead stream flow. BPNN and GANN models were tested using NS, R^2, MAE, RMSE, and MAPE, shown in Table 3.

Scatter plots in Figures 3 and 4 show the comparison between the model predicted and the model observed flow values for BPNN and GANN models, respectively. In particular, results in Figure 3 show the BPNN model to overestimate the predicted values of flow with respect to the observed values when stream flow range is less than 1300 m^3/s. Oppositely, when stream flow range is higher than 1300 m^3/s, the BPNN model underestimates the observed flow values. Very similar results are observed in Figure 4 for simulations concerning the GANN model.

The time series plot of BPNN and GANN is plotted in Figures 5 and 6. It was observed that both of the models have not captured extreme values properly. However, GANN follows the trend of the observed flow and has captured more extremities when compared to BPNN (Figures 5 and 6).

Table 3 shows that MAE, MAPE, and RMSE of the GANN model were much lower compared to those of the BPNN model. MAPE of GANN was nearly 2% lower than

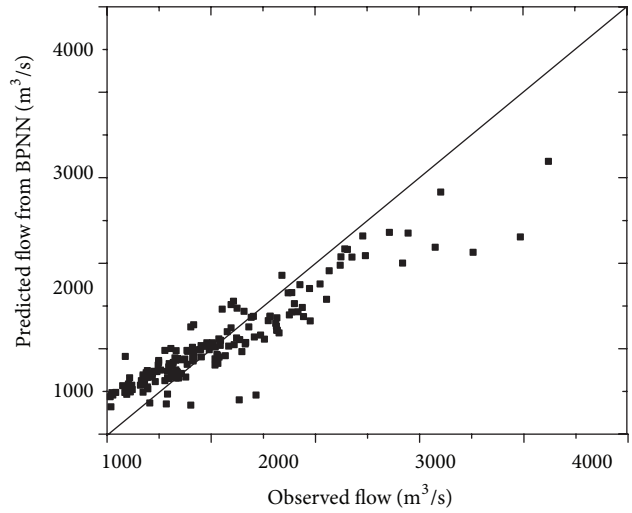

Figure 3: Scatter plot of the observed stream flow against the BPNN model predicted stream flow.

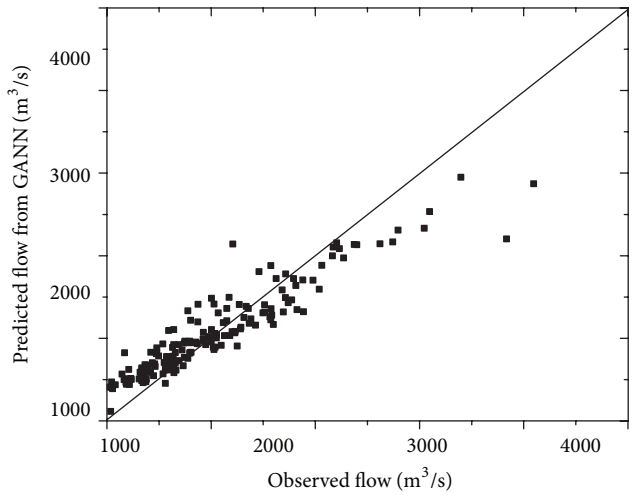

Figure 4: Scatter plot of the observed stream flow against the GANN model predicted stream flow.

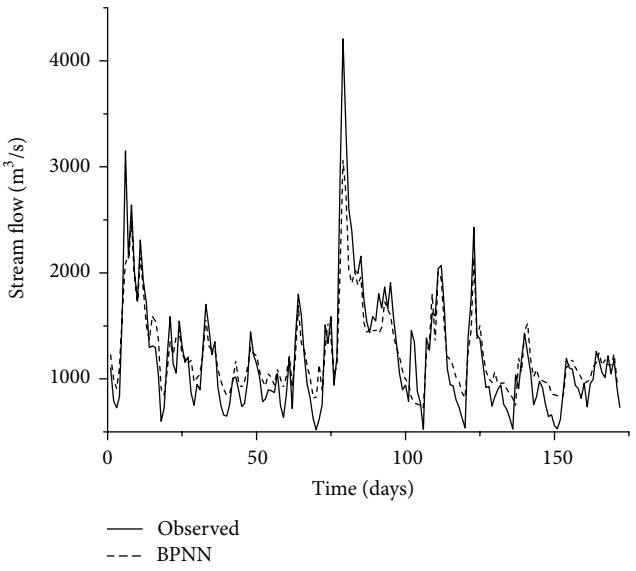

Figure 5: Time series plot of the observed stream flow and the BPNN model predicted stream flow.

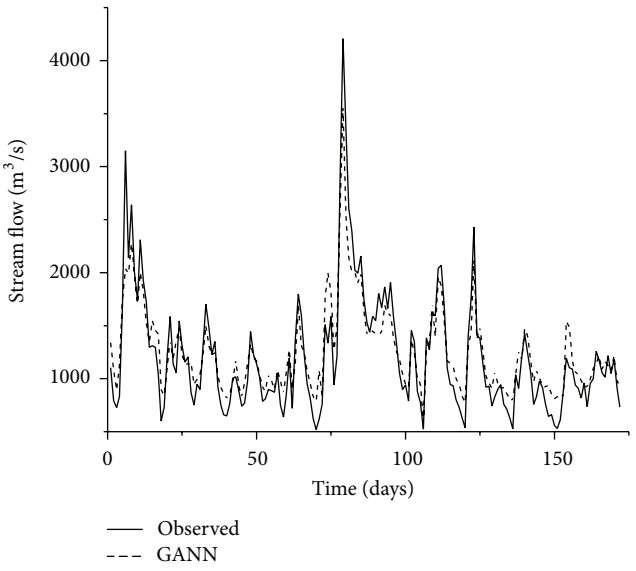

Figure 6: Time series plot of the observed stream flow and the GANN model predicted stream flow.

BPNN values. MAE of GANN model was nearly 10% lower than MAE of BPNN model. Both BPNN and GANN show good Nash Sutcliffe efficiency but GANN shows a better coefficient of determination than BPNN (Table 3). RMSE of BPNN model was 10% higher than RMSE of GANN model. Due to the effective random search and flexible problem solving method of GANN, it was able to predict better than BPNN. It was observed that GANN model has outperformed BPNN model by showing good efficiency during testing. The limitations of BPNN must have contributed for its lower performance when compared with GANN.

6. Summary and Conclusion

Two NN based models, namely, BPNN and GANN, were developed for the prediction of daily stream flows. The performances of the models were evaluated using statistical analysis. From their analysis, GANN model's predicted values were found to be very close to the observed values in comparison to BPNN model. This indicates that GANN shows greater potential to capture the existing nonlinearity in stream flows. The improved performance of GANN might be due to heuristic search for the optimal solution at many distinct locations simultaneously. Thus, the GA has a greater probability to reach the global minima. Conversely, back propagation algorithm training on steepest descent approach having fallen behind GANN model might be due to the trapping of good solutions in local optima, when the error surface is multimodal. Therefore, GANN model is considered to be more useful for hydrological forecasting and water resource management.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors sincerely thank head of the Department of Applied Mechanics and Hydraulics (NITK, Surathkal) and all faculty and research scholars who helped to improve the quality of this paper. The first author wishes to thank Manjunath for suggestions and discussions.

References

[1] N. Karunanithi, W. J. Grenney, D. Whitley, and K. Bovee, "Neural networks for river flow prediction," *Journal of Computing in Civil Engineering*, vol. 8, no. 2, pp. 201–220, 1994.

[2] M. Jajarmizadeh, S. Harun, and M. Salarpour, "An assessment of a proposed hybrid neural network for daily flow prediction in arid climate," *Modelling and Simulation in Engineering*, vol. 2014, Article ID 635018, 10 pages, 2014.

[3] H. Elçiçek, E. Akdoğan, and S. Karagöz, "The use of artificial neural network for prediction of dissolution kinetics," *The Scientific World Journal*, vol. 2014, Article ID 194874, 9 pages, 2014.

[4] A. W. Minns and M. J. Hall, "Artificial neural networks as rainfall-runoff models," *Hydrological Sciences Journal*, vol. 41, no. 3, pp. 399–417, 1996.

[5] C. W. Dawson and R. Wilby, "An artificial neural network approach to rainfall-runoff modelling," *Hydrological Sciences Journal*, vol. 43, no. 1, pp. 47–66, 1998.

[6] D.-I. Jeong and Y.-O. Kim, "Rainfall-runoff models using artificial neural networks for ensemble streamflow prediction," *Hydrological Processes*, vol. 19, no. 19, pp. 3819–3835, 2005.

[7] M. A. Antar, I. Elassiouti, and M. N. Allam, "Rainfall-runoff modelling using artificial neural networks technique: a Blue Nile catchment case study," *Hydrological Processes*, vol. 20, no. 5, pp. 1201–1216, 2006.

[8] M. P. Rajurkar, U. C. Kothyari, and U. C. Chaube, "Artificial neural networks for daily rainfall—runoff modelling," *Hydrological Sciences Journal*, vol. 47, no. 6, pp. 865–878, 2002.

[9] A. S. Tokar and P. A. Johnson, "Rainfall-runoff modeling using artificial neural networks," *Journal of Hydrologic Engineering*, vol. 4, no. 3, pp. 232–239, 1999.

[10] A. R. Senthil Kumar, K. P. Sudheer, S. K. Jain, and P. K. Agarwal, "Rainfall-runoff modelling using artificial neural networks: comparison of network types," *Hydrological Processes*, vol. 19, no. 6, pp. 1277–1291, 2005.

[11] A. Sedki, D. Ouazar, and E. El Mazoudi, "Evolving neural network using real coded genetic algorithm for daily rainfall-runoff forecasting," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4523–4527, 2009.

[12] D. K. Pratihar, *Soft Computing*, Narosa Publishing House, New Delhi, India, 2nd edition, 2008.

[13] D. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, 1989.