African Journal of Agricultural Research Vol. 6(26), pp. 5775-5784, 12 November, 2011
Available online at http://www.academicjournals.org/AJAR
DOI: 10.5897/AJAR11.1892
ISSN 1991-637X ©2011 Academic Journals

Full Length Research Paper

Groundwater modeling using hybrid of artificial neural network with genetic algorithm

Amir Jalalkamali* and Navid Jalalkamali

Department of Water Engineering, Faculty of Engineering, Islamic Azad University, Kerman branch, Kerman, Iran.

Accepted 29 August, 2011

Accurate estimates of groundwater level have a valuable effect in improving decision support systems of groundwater resources exploitation. The present study investigates the ability of a hybrid model of artificial neural network (ANN) and genetic algorithm (GA) in forecasting groundwater level in an individual well (target well). A standard feed forward networks (FFN) and recurrent neural networks (RNN) are utilized for performing the prediction task. Moreover, GA is used in order to determine the optimal structure of ANN (that is, number of neurons for each hidden layer). Air temperature, rainfall depth and groundwater levels in neighboring wells in Kerman plain (Kerman, Iran) were used as input data of the hybrid model. This study indicates that the ANN-GA model can be used successfully to forecast groundwater levels of individual wells. In addition, a comparative study of both hybrid models indicates that the feed forward networks performed better than the recurrent neural networks.

Key words: Artificial neural network, feed forward networks, recurrent neural networks, genetic algorithm, groundwater level.

INTRODUCTION

The estimation accuracy of the groundwater levels has a valuable effect in efficiency of a supportive deciding system and exploitation of available water resources. Kerman is one of the provinces of Iran which is located in a dry area with 150 mm precipitation on average annually. Because of the development in cities, industries, agriculture and drought in these two recent decades, groundwater level has decreased at this area (around 1 to 3 m annually). In such situation, the simulated models of groundwater level can be used as an instrument for management of withdrawing water from these limited sources.

The conceptual and physically based models require a large quantity of good quality data, sophisticated programs for calibration using rigorous optimization techniques and a detailed understanding of the underlying physical process. When data is not sufficient, empirical models are a good alternative method, and can provide useful results without a costly calibration time (Krishna et al., 2008). In recent years, artificial neural networks being capable of analyzing long series and large-scale data, and it become increasingly popular in hydrology and water resources among researchers and practicing engineers.

ANNs are proven to be effective in modeling virtually any nonlinear function to an arbitrary degree of accuracy. Nasseri et al. (2008) developed a feed-forward neural network coupled with GA to simulate the rainfall field. The technique implemented to forecast rainfall for a number of times using hyetograph of recording rain gauges. Results showed that when FFN coupled with GA, the model performed better compared to similar work of using ANN alone.ANN applications in hydrology vary, from real-time to event based modeling. They have been used for groundwater modeling, level estimation (Coulibaly et al., 2001; Nourani et al., 2008).

A comprehensive review of the applications of ANNs in hydrology can be found in the ASCE Task Committee report (ASCE, 2000a, b). A few applications of the ANN approach in groundwater related problems can be found in the literature (Coppola et al., 2005; Lallahema et al., 2005). Groundwater levels have been forecasted in an individual well by monitoring continuously over a period of time using ANN (Daliakopoulos et al., 2005). In another study a developed ANN model used to forecast groundwater levels in an urban coastal aquifer (Krishna

*Corresponding author. E-mail: ajalalkamali@yahoo.com.

et al., 2008). Groundwater levels have also been forecasted by taking into account the nearby wells and climatic parameters by developing a single model pertaining to each individual well by developing well at various lead periods using artificial neural networks (Nayak et al., 2006). In another study an ANN model used to forecast groundwater changes in an aquifer (Tsanis et al., 2008). In this study, a new method was proposed to improve the forecasting of monthly groundwater level by using genetic algorithm to optimize the structure of multi-layer feed-forward network (FFN) and recurrent neural networks (RNN). Genetic algorithm (GA) is used to search for optimal structure of ANNs for forecasting groundwater level. In other words, a novel, ANNs model based on genetic algorithm was developed to build relationship between time series information nearby wells, climatic parameters data and groundwater level fluctuations in individual well.

MATERIALS AND METHODS

Artificial neural networks

Artificial neural networks estimation approach has received tremendous attentions in the last few decades. An interesting property of ANNs is that they often work well even when the training data sets contain noises and measurement errors (Hammerstrom, 1993). Moreover, they have the capability of representing complex behaviors of nonlinear systems (Maier and Dandy, 2000). The advantage of the ANN is that with no prior knowledge of the actual physical process and, hence, the exact relationship between sets of input and output data, if acknowledged to exist, the network can be trained to learn such a relationship. The ability to train and learn the output from a given input makes ANN capable of describing large scale arbitrarily complex non-linear problems. A neural network is characterized by its architecture that represents the pattern of connection between nodes, its method of determining the connection weights, and the activation function (Fausett, 1994). A typical ANN consists of a number of nodes that are organized according to a particular arrangement.

Feed forward neural network models

One way of characterizing ANNs is based on the direction of information flowing and processing, as feed-forward (where the information flows through the nodes from the input to the output side) and recurrent (where the information flows through the nodes in both directions). Among these combinations, the multi-layer feed-forward networks, also known as multi-layer perceptron (MLPs), trained with a back-propagation learning algorithm have been found to provide the best performance with regard to input-output function approximation, such as forecasting applications. A typical MLP with one hidden layer is shown in Figure 1; (a). The first Layer connects with the input variables and is called the input layer. The last layer connects to the output variables and is called the output layer. The layer between the input and output layers, is called the hidden layer (there may be more than one hidden layer in an MLP).The processing elements in each layer are called nodes or units. Each node is connected to the nodes of neighboring layers. The parameters associated with each of these connections are called weights. The architecture of a typical node (in the hidden or output layer) is also shown in Figure 1; (b) Each node *j* receives incoming signals from every node *i* in the previous layer. Associated with each incoming signal x_i is a weight w_{ji}. The effective incoming signal s_j to node *j* is the weighted sum of all the incoming signals, is passed through the effective incoming signal, s_j non-linear activation function (sometimes called a Transfer function or threshold function) to produce the outgoing signal y_j of the node.

$$f(s_j) = \frac{1}{1+\exp^{-s_j}}$$

The most commonly used function in an MLP trained with back-propagation algorithm is the sigmoid function. The sigmoid function most often used for ANNs is the logistic function (Sivakumar et al., 2002):

$$S_j = \sum_{i=0}^{n} w_{ji} x_i \tag{2}$$

Recurrent neural network models

The recurrent neural network (RNN) is another multi-layer architecture that has been used for a variety of applications including control systems and forecasting of dynamic processes. In this section RNN structure is briefly discussed.

The RNN architecture, a variation of general feed-forward back-propagation (FFBP) architecture, is used to capture dynamic and highly nonlinear systems by including a feedback mechanism in the architecture. The general RNN architecture uses specialized hidden nodes to introduce feedback to the network. In such a network, the output of these specialized nodes is provided as input to others. Once such feedback connections are allowed, the network topology becomes more connected since any node can be connected to any other node, including to itself. These self-connected or self-recurrent feedback nodes form the "context" layer of a network and are tagged on to the network structure along with the usual non-feedback nodes. The "context" layer is used to retain information between training iterations and serve as memory of the system by retaining the state of the network before the next set of data is processed. Each time a pattern is presented, each context node computes its activation just as in a feed forward network. However, its output is now able to reflect the state of the network before the pattern is seen. When subsequent patterns are presented, the hidden and output units' states will be a function of everything the network has seen so far. Thus, at each time period, activation propagates only forward through one layer of connections. Once some level of activation is present in the network, it will continue to flow through all the remaining hidden layers, even in the absence of any new input whatsoever. However, this added feedback mechanism (memory function) requires additional network connections, a large amount of storage and computation, and a larger training set in order for the RNN to work well. This invariably leads to difficult network training and slow convergence (Atya and Parlos, 2000).

Training methodology in RNN

The method of RNN training is similar to that of feed forward network models. The training algorithm is explained with the help of a simple example. A small network which has two input neurons, one hidden layer having three neurons and one output neuron is shown in Figure 2. In addition, a neuron taking input from the output layer and connected to the hidden layer is added as shown. This neuron is the additional neuron in RNN.

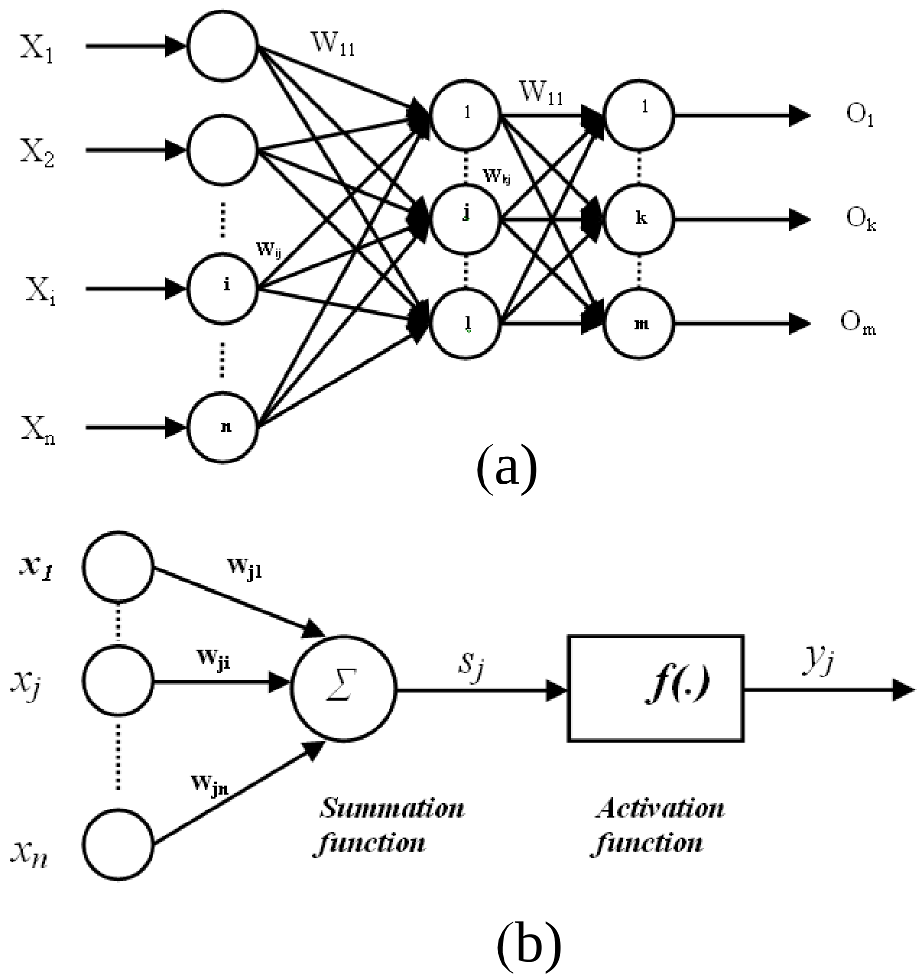

Figure 1. (a) Multi layer forward ANN (b) a simple neuron.

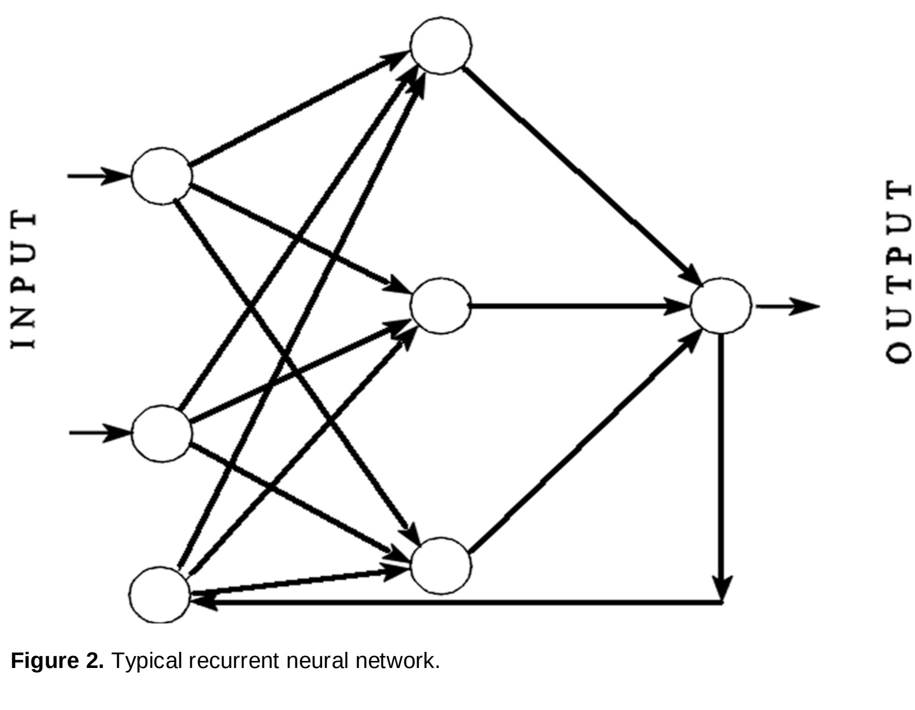

Figure 2. Typical recurrent neural network.

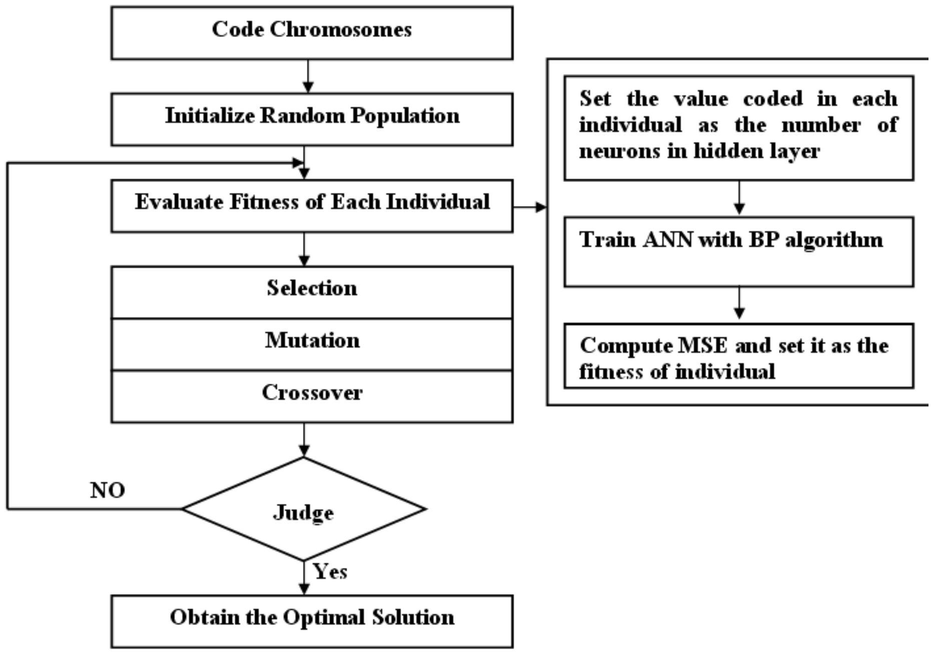

Figure 3. The flow chart of ANN-GA model.

Genetic algorithm

GA optimizes using a search process that emulates natural evolution. On the other hand GA is a global heuristic, stochastic, optimization technique based on evolution theory and genetic principles developed by (Holland, 1975). Goldberg and Michalewicz (1992) discussed the mechanism and robustness of GA in solving nonlinear optimization problems (Goldberg, 1989; Michalewicz, 1992). The algorithm begins with a randomly generated population which is consisting of chromosomes, and applies three kinds of genetic operators: The selection, crossover and mutation operators to find the optimal solutions. The selection operator chooses chromosomes from the current population based on fitness value of the individuals. The crossover operator combines the features of two parent chromosomes to form two similar offspring by swapping corresponding segments of the parents (Goldberg, 1989). The mutation operator creates new chromosomes by randomly changing the genes of existing chromosomes. GA can explore the entire design space by the genetic manipulations; it does not easily fall into a certain local minima or maxima.

As this occurs, the GA converges to increasingly better solutions. Improvements in fitness, however, diminish as the population diversity decreases and the population converges toward a good solution. Stopping criteria such as “100 generations without improvement” and minimum population diversity are often used to terminate the algorithm when improvements are sufficiently small and infrequent. These concepts are well described in (Davis, 1991; Goldberg, 1989). Therefore, GA is an aggressive search technique that quickly converges to find the optimal solution in a large solution domain.

ANN-GA model scheme

In this research, a multi-layered feed-forward neural network (FFN) and recurrent neural network (RNN) with a back propagation algorithm are adopted. Although the back propagation algorithm is successful, it has some disadvantages. The algorithm is not guaranteed to find global minimum of error space and the convergence tends to be extremely slow. In addition, the selection of the learning factor and inertial factor affects the convergence of the BP neural network which is usually determined by experience. In present research, the number of neurons in the hidden layer is determined using the genetic algorithm. The number of hidden layers and the number of nodes in each layer depends on the complexity of the patterns and the nature of the problem to be solved. The use of a single hidden layer is sufficient to approximate to any continuous function as closely as requested (Funahashi, 1989; Hornik et al., 1990) and studies also showed that having more than two layers may not result in significant performance improvements (Patuwo et al., 1993) Thus, in our study, a two-layer ANN is utilized (Figure 1). The number of neurons in the input and output layers are given by the number of input and output variables of network. The number of neurons in hidden layer is obtained by GA. In this study, an ANN with one hidden layer is employed. The number of neurons in this layer is determined by GA. The optimization process flow chart of the ANN-GA model is shown in Figure 3. The sigmoid function was used in each node of the hidden layer and output layer as the transfer function.

Number of neurons in hidden layer is the only information that is coded in a chromosome in GA. After that, the GA is run and in its fitness assignment past, an ANN which the number of its hidden layer neuron is determined by coded chromosome is trained via ANN. Then the MSE of this trained ANN is set as the fitness values. The GA will generate many of individual values which they will be set to MSE. This process is depicted in Figure 3.

Simulation setup

Study area and data set

The area which studied in this research is the Kerman plain aquifer which is a part of Kerman province located in the south-eastern of Iran as shown in Figure 4. In this plain, no permanent river exists; therefore, the supply of water demands in agriculture, industry,

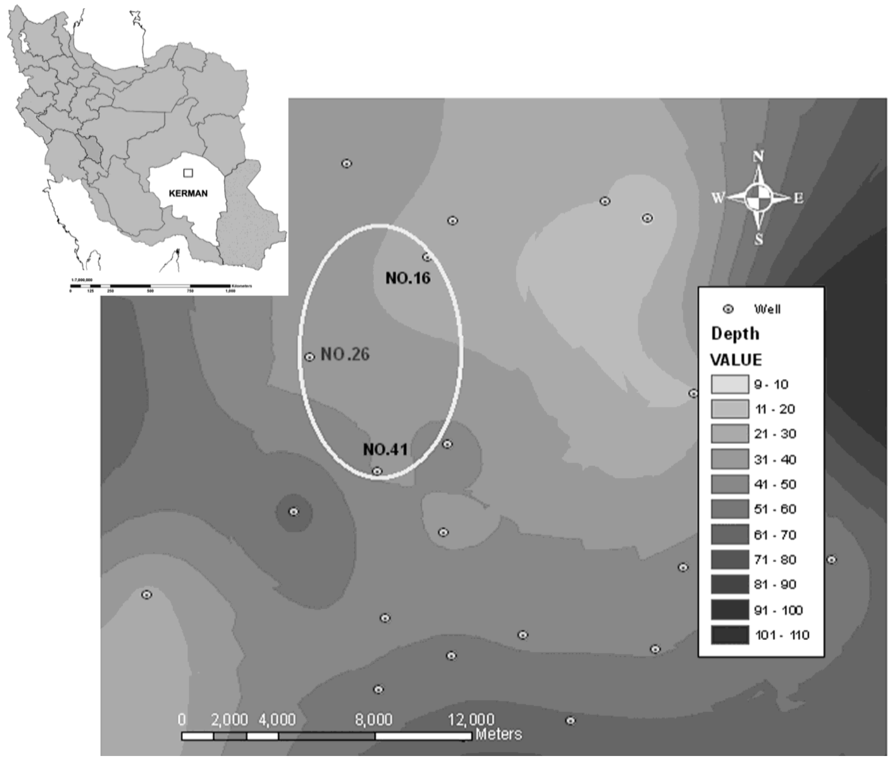

Figure 4. The location of wells in Kerman plain.

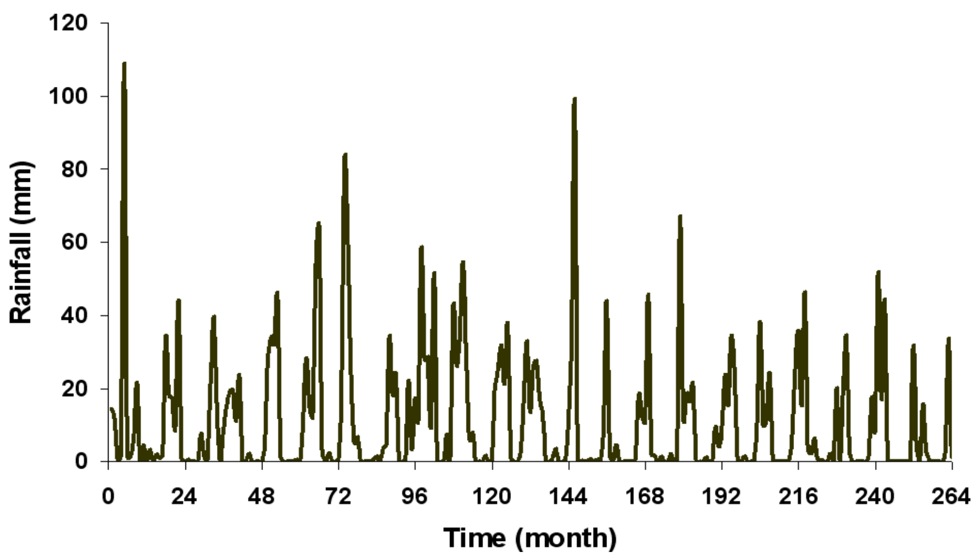

Figure 5. Time series plot for the rainfall versus month.

domestic and municipal sectors in 3200 km^2 area around this plain highly depends on groundwater. In the past two decades, frequent hydrologic droughts besides the increasing number of pumping wells have caused a decline rate of 1 to 3 m annually. As a consequent the groundwater quality has decreased as well. The long-term annual precipitation for the area has noticeably decreased from 150 to 100 (mm/year) during the 20 past years (1988 to 2009).

The data acquired from the area consists of rainfall depth, temperature and depth of the wells time series measured at Kerman airport station (latitude: 30^o, 16' N, longitude: 56^o, 54' E). The data set was provided by Iranian Ministry of Energy (IMOE).

The time series used in this research are summarized for a 22 year period (1988 to 2009). Figure 5 presents, the monthly precipitation at meteorological Kerman airport station. In this region most of annual rainfall occurred during the winter season. Because

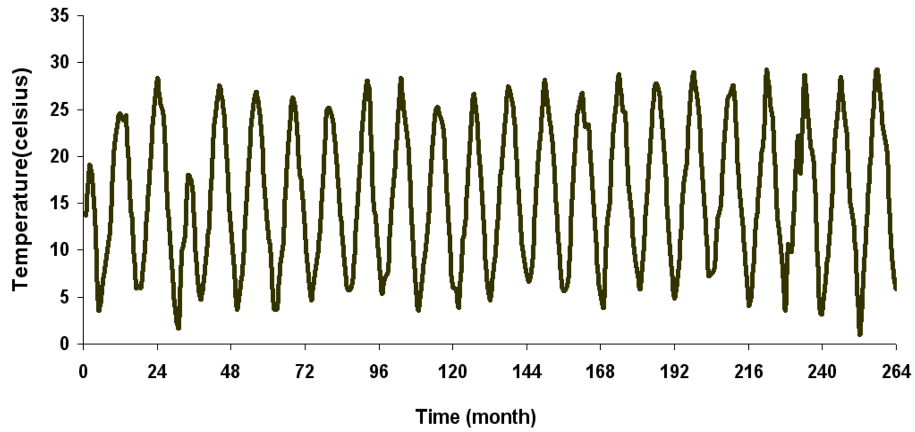

Figure 6. Time series plot for the temperature versus month.

Table 1. The monthly statistical parameters of data.

Data set	Unit	X_{mean}	S_x	C_{sx}	X_{max}	X_{min}
$H_{NO.26}$	m	-33.36	1.74	0.63	-28.82	-35.93
$H_{NO.16}$	m	-37.71	4.3	-0.39	-30.35	-45.49
$H_{NO.41}$	m	-34.57	3.72	0.43	-26.11	-40.53
R	mm	11.26	17.39	2.3	109.1	0
T	c^o	15.82	7.68	0.03	29.25	1.05

of relatively high temperature of this province, temperature plays an important role in the water budget. Figure 6 shows the monthly temperature for the period mentioned.

The data sample consisted of 22 years (1988 to 2009) of monthly records of air temperature (*T*), rainfall (*R*) and water levels in target well ($H_{NO.26}$) and neighboring wells ($H_{NO.16}$ and $H_{NO.41}$). The first 19 years (1988 to 2006) data were used to train the models and the remaining data for testing. The monthly statistics of each time series are given in Table 1. In the table the X_{mean}, S_x, C_{sx}, X_{max} and X_{min} respectively denote the mean, standard deviation, skewness coefficient, maximum and minimum of observations.

Parameter setup

Population size and generation numbers are set to 100. The tournament selection is used as selection method in GA, two point crossover and an uniform mutation are consider for reproduction Crossover rate and mutation probability are set to 0.7 and 0.01 respectively. Learning rate in BP algorithm is set to 0.02 and 50 epochs are considered for training the ANN.

RESULTS AND DISCUSSION

Input names are temperature (T), rainfall (R), water fluctuation in the nearest lateral piezometers ($H_{No.16}$, $H_{No.41}$), and the target well (H_{No26}). These inputs are selected according to previous successful researches in this field (Atya and Parlos, 2000; Davis, 1991).

Various combinations of inputs, which are input values in different times, are experimented for feeding into the ANN. Inputs are values of aforementioned variables in times *t*, *t-1*, *t-2* and the goal is to forecast the water level value in time (*t+1*). Input combinations, which are rational according to previous researches, are shown in Table 2.

These combinations are titled cases (1 to 4) concisely. As it can be seen from Table 1 the results of water level prediction is highly dependent on input combination. Considering the training and testing results it can be inferred that the water level variable is not very dependent to climatic variables such as temperature and precipitation (Cases 1 and 2 in Table 1). Also the time series of observed water levels could not manage to improve the results (Case 4) while the model accuracy shows sensitivity to water levels of neighboring wells (Case 3).

The best number of neurons in the hidden layer of both FFN and RNN models derived by GA algorithm are presented in Table 2. Ratio of the number of hidden layer neurons to those of input parameters in FNN models range from 1.6 to 6 and as for RNN models the ratio ranges from 1.8 to 5.7. Hence both model structures need similar optimum number of neurons.

Besides, as the number of input variables increases the mentioned ratio decreases drastically while the number of hidden layer neurons does not show a significant change. In other words, based on the present study, it seems that the optimum number of hidden layer neurons is independent of the number of input variables.

In order for the performance evaluation of the models,

Table 2. Details of ANN-GA model architecture.

Case	Input combination	Number of inputs	No. of hidden neurons for FFN-GA model	No. of hidden neurons for RNN-GA model
1	$H_{NO.26}(t)$, $H_{NO.26}(t-1)$, $H_{NO.16}(t)$, $H_{NO.16}(t-1)$, $H_{NO.41}(t)$,$H_{NO.41}(t-1)$, R(t), R(t-1), T(t), T(t-1)	10	16	18
2	$H_{NO.26}(t)$, $H_{NO.26}(t-1)$, R(t), R(t-1), T(t), T(t-1)	6	15	13
3	$H_{NO.26}(t)$, $H_{NO.26}(t-1)$, $H_{NO.16}(t)$, $H_{NO.16}(t-1)$, $H_{NO.41}(t)$, $H_{NO.41}(t-1)$	6	19	18
4	$H_{NO.26}(t)$, $H_{NO.26}(t-1)$, $H_{NO.26}(t-2)$	3	18	17

Table 3. Error analysis of level forecasting in test and train period.

SET		Training			Testing		
MODEL	Case	RMSE	MAPE	R^2	RMSE	MAPE	R^2
FFN-GA	1	0.067	0.143	0.998	0.116	0.256	0.65
FFN-GA	2	0.135	0.320	0.995	0.104	0.248	0.73
FFN-GA	3	0.109	0.208	0.995	0.05	0.106	0.95
FFN-GA	4	0.045	0.103	0.999	0.113	0.244	0.75
RNN-GA	1	0.186	0.459	0.996	0.125	0.272	0.63
RNN-GA	2	0.108	0.215	0.995	0.114	0.232	0.69
RNN-GA	3	0.115	0.215	0.995	0.073	0.159	0.89
RNN-GA	4	0.036	0.103	0.999	0.099	0.208	0.80

three different types of standard statistical fitness criteria were considered, namely correlation coefficients (R^2), root mean square error (RMSE), and mean absolute percentage error (MAPE). The three performance evaluation criteria are based on the following equations:

$$RMSE=\sqrt{\frac{\sum_{i=1}^{n}(H_i^{o}-H_i^{p})^2}{n}} \quad (3)$$

$$MAPE=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{H_i^{p}-H_i^{o}}{H_i^{o}}\right|\times 100 \quad (4)$$

$$R^2=\left(\frac{\sum_{i=1}^{n}(H_i^{o}-\overline{H}_i^{o})(H_i^{p}-\overline{H}_i^{p})}{\sqrt{\left[\sum_{i=1}^{n}(H_i^{o}-\overline{H}_i^{o})^2\right]\left[\sum_{i=1}^{n}(H_i^{p}-\overline{H}_i^{p})^2\right]}}\right)^2 \quad (5)$$

Where, H_i^O is the observed groundwater level at present time and H_i^P is the forecasted groundwater level. RMSE = 0, MAPE = 0 and R^2 = 1 shows a perfect fit. Table 3 represents the resulted fitness statistics of testing and training steps for the considered models. Performance measures apparently show that the third row of Table 3 is the best case.

The fitness criteria of FFN and RNN models are shown in Table 3. The fourth case of input combination of FFN-GA models in the training phase portrays the best fitness. In this modeling structure, case 1 stands in the second rank of fitness. The Cases 4 and 1 are models with the most and least numbers of input variables respectively, which reveals an interesting finding: in the training phase, the model with the least number of input variables (3 input variables) and so with the largest value of neurons/inputs ratio = 6 presents the best performance. However testing phase shows different results. In this phase the best FFN-GA model is Case 3 with RMSE, MAPE and R2 value equal to 0.05, 0.106 and 0.95 respectively.

An interesting point to be mentioned is that except for R^2, other fitness criteria improved over testing phase compared to the training ones. It must be noted that the number of neurons/inputs ratio in this case is 3.2 which offers a logical number of model parameters (including weights and biases) in neural network modeling practices.

Based on results presented in Table 3, Case 3 apparently shows the best performance among other input combinations, that is, using more input variables could not necessarily guarantee higher model performance. The situation is similar in RNN-GA models. In these models, as well, case 4 shows the best performance in

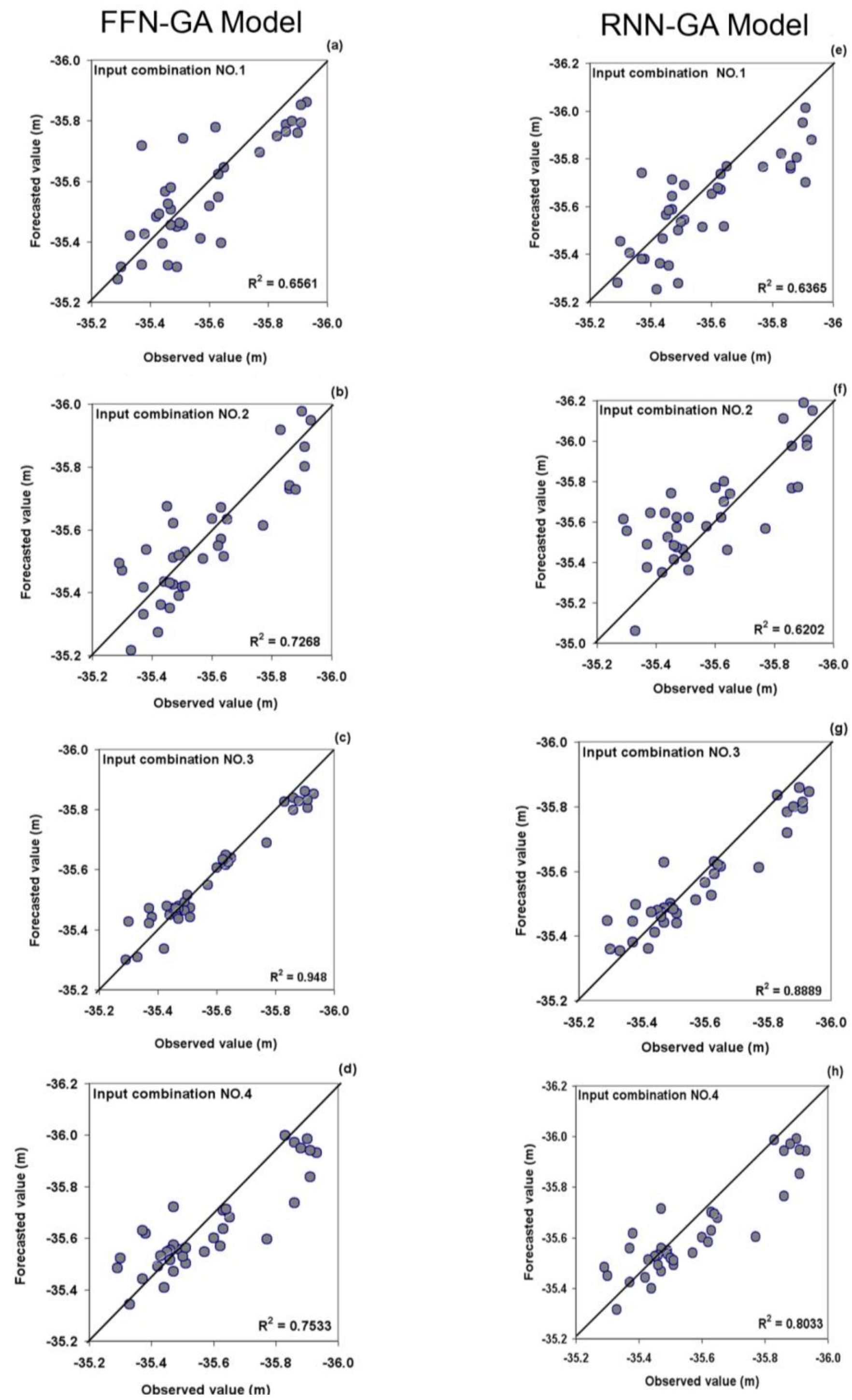

Figure 7. Assessing the ability of FFN-GA model compared to the RNN-GA model.

the training phase while it is ranked second in testing phase. Case 3 maintaining RMSE, MAPE and R2 values equal to 0.073, 0.159 and 0.89 respectively shows the best performance in the testing phase. Furthermore, as in FFN-GA models, performance criteria of Case 3 improved over testing phase except for R^2 coefficient (36.5% improvement for RMSE and 26% for MAPE). Comparison of two FFN-GA and RNN-GA hybrid models in case 3 reveals the privilege of FFN-GA model in both training and testing phases, however, the determination coefficient of R2 is larger than 0.89 in both models.

Assessing the ability of FFN-GA model compared to the RNN-GA model is shown in Figure 7 (a-h) in form of scatter plot Figure 7, (c), which depicts observed values

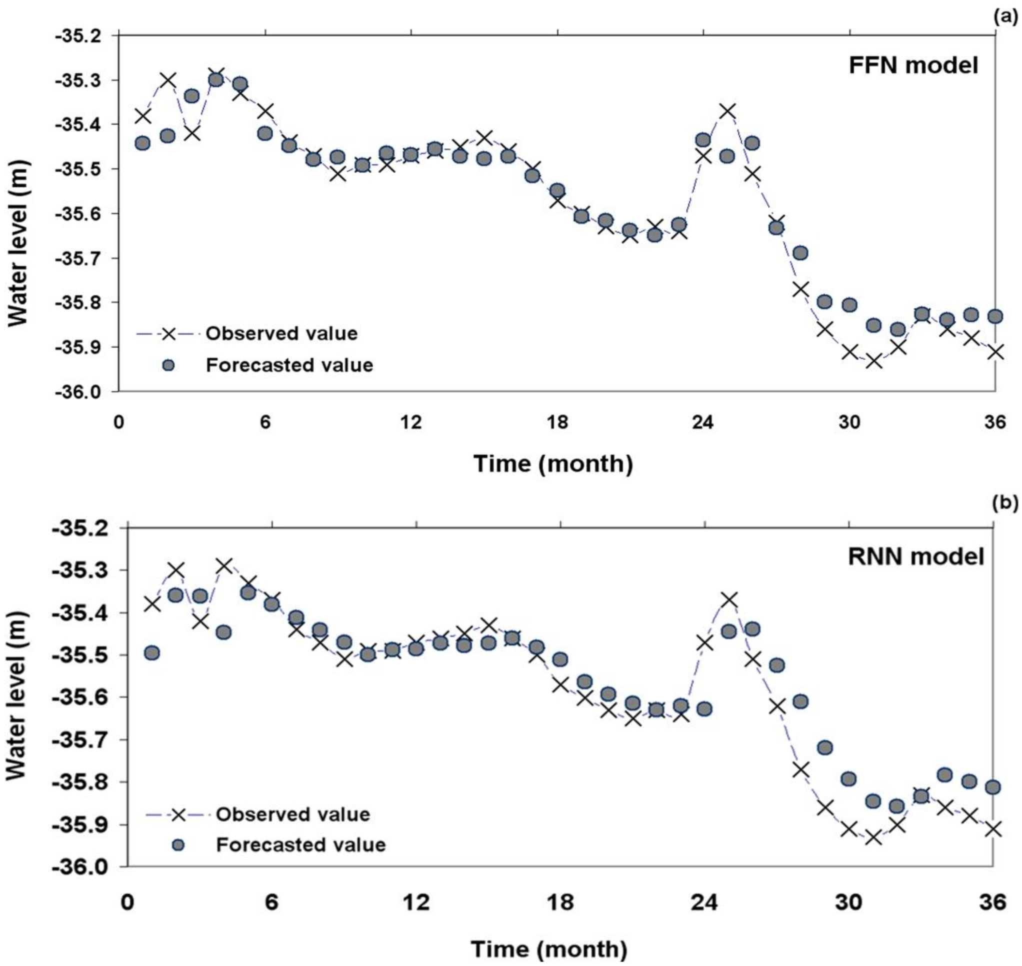

Figure 8. Comparison of FFN-GA, RNN-GA and observed monthly groundwater level at well No.26.

versus the estimated values for Case 3, shows that it is the best fitted model in terms of obtaining a tradeoff between accuracy and the structure of FFN-GA model. It can visually be verified that observed values and estimated values have a better correspondence according to Figure 7, (c) rather than Cases 1, 2, and 4. Therefore case 3 is more appropriate from numerical and graphical points of view.

Also the groundwater level predictions of each model in the test period for the best input combination are represented in Figure 8 (a-b) in the form of hydrographs. It is obviously seen from the hydrographs and scatter plots that the FFN-GA (Figure 8 (a)) predicts are closer to the corresponding observed values than the RNN-GA model. Performance measures given in Table 3 confirm the results shown in Figure 8.

Based on Figure 8, both models show less accuracy in capturing the peak and nadir points of water level although as it is expected the FFN-GA hybrid model shows closer performance to the observed data. In contrast, the models predict the average water levels with high accuracy while they encounter difficulties in predicting in case of water level fluctuations.

Conclusion

Neural networks have proven to be an extremely useful method for empirical modeling of hydrological variables. The present study utilized artificial neural networks in corporation with genetic algorithm aiming at forecasting groundwater level. Number of neurons in the hidden layer is derived using the genetic algorithm for four input combinations separately. The study showed that the best input combination for groundwater level forecasting is water level time series data in neighboring wells (input combination Number 3).

Two ANN-GA hybrid models (FFN-GA and RNN-GA) were tested and results indicated an excellent agreement between the forecasted and observed data. However the FFN-GA hybrid model was found to perform better than RNN-GA in forecasting monthly groundwater levels; although the performance of FFN-GA model is categorized as nearly perfect in predicting the middle range of water level, and it experienced little problems in water fluctuation and extreme cases.

Considering the vast range of neurons/inputs ratio, exploiting such hybrid models may lead to optimum and fast results concerning the number of hidden layer neurons. Also it was showed that higher numbers of input variables in neural network modeling could not necessarily guarantee better performance, although this task will decrease the number of neurons/inputs ratio significantly. The most logical ratio in the present study was estimated at 3, which is recommended in similar researches. Another finding of this study is the probable independence of the optimized number of neurons from the number of input variables, which is not in agreement with some recommendations stated in previous researches.

REFERENCES

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000a). Artificial neural networks in hydrology- I: Preliminary concepts. J. Hydrolo. Eng., 5(2): 115-123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000b). Artificial neural networks in hydrology-II: Hydrologic applications. J. Hydrolo. Eng., 5(2): 124-137.

Atya AF, Parlos AG (2000). New results on recurrent network training: Unifying the algorithms and accelerating convergence. IEEE Trans. Neural Netw., 2(3): 697-709.

Coppola E, Anthony JR, Poulton M, Szidarovszky F, Vincent W (2005).A neural network model for predicting aquifer water levels elevations. Groundwater. 43(2): 231-241.

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.

Daliakopoulos IN, Coulibalya P, Tsanis IK (2005). Groundwater level forecasting using artificial neural networks. J. Hydrol., 309: 229-240.

Davis L (1991). Handbook of genetic algorithms. Van Nostrand Reinhold, New York.

Fausett L (1994). Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Englewood Cliffs, NJ: Prentice Hall.

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

Goldberg DE (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts: pp. 412.

Hammerstrom D (1993). Working With Neural Networks. IEEE Spectrum: 46-53.

Holland JH (1975). Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.

Hornik K, Stinchcombe M, White H (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feed forward networks. Neural Netw., 3(5): 551-560.

Krishna B, Satyaji Rao YR, Vijaya T (2008). Modeling groundwater levels in an urban coastal aquifer using artificial neural networks. Hydrol. Process, 22: 1180-1188.

Lallahema S, Maniaa J, Hania A, Najjarb Y (2005). On the use of neural networks to evaluate groundwater levels in fractured media. J. Hydrol., 307: 92-111.

Maier HR, Dandy GC (2000). Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications. Environ. Model. Softw., 15: 101-124.

Goldberg DE, Michalewicz Z (1992). Genetic algorithms + data structures = evolution programs (3rd ed.). Springer-Verlag.

Nasseri M, Asghari K, Abedini MJ (2008). Optimized scenario of Rainfall Forecasting using Genetic Algorithms and Artifitial Neural Networks. Expert Syst. Appl., 35(3): 1415-1421.

Nayak PC, Satyaji Rao YR, Sudheer KP (2006). Groundwater level forecasting in a shallow aquifer using artificial neural network approach. Water Resour. Manage., 20: 77-90.

Nourani V, Mogaddam AA, Nadiri AO (2008). An ANN-based model for spatiotemporal groundwater level forecasting. Hydrol. Process, 22: 5054-5066.

Patuwo E, Hu MY, Hung MS (1993). Two group classification problem using neural networks. Decis. Sci., 24(4): 825-846.

Sivakumar B, Jayawardena AW (2002). Fernando TMKG. River flow forecasting: use of phase space reconstruction and artificial neural networks approaches. J. Hydrol., 265: 225-245.

Tsanis IK, Coulibaly P, Daliakopoulos IN (2008). Improving groundwater level forecasting with a feedforward neural network and linearly regressed projected precipitation. J. Hydroinf., 10(4): 317-330.