Lake-Level Prediction Leveraging Deep Neural Network

Jinfeng Wen[1(✉)], Peng-Fei Han[1], Zhangbing Zhou[1,2], and Xu-Sheng Wang[1]

[1] China University of Geosciences (Beijing), Beijing 100083, China
wenjinfeng127@163.com

[2] Computer Science Department, TELECOM SudParis, 91011 Evry, France

Abstract. Accurate estimation of water level dynamics in lakes at daily or hourly time-scales is important for the ecosystem and formulation of water resources policies. In this study, lake level dynamics of Sumu Barun Jaran are simulated and predicted at hourly time scale using Deep Learning (DL) model. Two mature machine learning methods, namely Multiple Linear Regression (MLR) and Artificial Neural Network (ANN), are also adopted for the comparison purpose. The result shows that the DL model preforms the best on three criteria, following by the three-layered Back-Propagation ANN model and MLR model.

Keywords: Lake level · Sumu Barun Jaran · Badain Jaran Desert
Deep learning · Artificial neural network

1 Introduction

Lakes, as the most abundant water resources that carries on land, are an important factor for impacting the human life and earth ecosystem. An accurate estimation of lake level dynamics in an efficient way is essential for effective assessments on water resources and environment in lakes. Generally, this estimation is a complicated mathematical problem. In the traditional methods of hydrology, physics-based numerical model and conceptual hydrological model are often used to simulate water level fluctuations [1]. However, this approach requires a variety of parameters with clear physical meaning, terrain data and boundary conditions, a lack of which may lead to poor model performance or increase the model uncertainty.

In recent years, the autoregressive integrated moving average (ARIMA) model and multiple linear regression model are usually used to define the trend or stochastic processes of variables, but neither of them consider the non-stationarity and non-linear characteristics of the data structure [2]. Artificial neural networks (ANNs) techniques have been applied to solve non-stationary and non-linear problems in time series analyses for the modeling of water level fluctuations [3]. However, because of the shallow number of layers of ANNs, the learning power of this model can hardly be applicable especially when spatial-temporal data are sensed with multiple features. Recently, deep learning is

L. Wang et al. (Eds.): QShine 2017, LNICST 234, pp. 23–32, 2018.
https://doi.org/10.1007/978-3-319-78078-8_3

increasingly popular and accepted by researchers, and it can solve the tasks associated with artificial intelligence and achieve excellent results with efficient operations [4]. However, little research effort has been developed to solve the hydrological problem using the deep learning method at this moment.

In this study, we attempt to use different machine learning methods in estimating the lake level dynamics, for lakes in the Badain Jaran Desert, China. Note that the Deep learning (DL) model is constructed and applied to simulate and predict the lake level dynamics at hourly time scale for the first time, and the model performance is evaluated. Two mature machine learning methods, namely Multiple Linear Regression (MLR) and Artificial Neural Network (ANN), are also adopted for the comparison purpose. The result shows that these three models are appropriate for simulating and predicting lake level dynamics at hourly time scale, and the DL model preforms the best.

The rest of this paper is organized as follows. In Sect. 2, MLR, ANN and DL structures are introduced. In Sect. 3, the MLR, ANN with three-layered BP and DL models are applied and evaluated. Finally, the conclusion is being made in Sect. 4.

2 Prediction Models

Considering the variety of meteorological data, MLR, ANN and DL models are adopted to simulate and predict the lake level dynamics.

2.1 Multiple Linear Regression (MLR)

Multiple linear regression analysis is a multivariate statistical technique aimed to predict the lake level as a dependent variable, Y, by using a set of p predictor variables (x_1, x_2, ..., x_p), as presented by Table 1.

Table 1. The partial meteorological and lake level data.

Time	TA($°C$)	TS_{10}($°C$)	TS_{20}($°C$)	RH(%)	WD(Deg)
2012 09-13 00:00	11.722	20.793	22.38	50.115	357.786
2012 09-13 02:00	10.314	20.602	22.479	60.253	3.004
2012 09-13 04:00	9.545	20.379	21.748	59.077	17.044
2012 09-13 06:00	9.373	20.544	22.157	61.716	342.339
Time	AP(hPa)	$R_{\mathbf{n}}$(W/m^2)	WS(m/s)	P(mm)	H(m)
2012 09-13 00:00	888.409	−114.248	0.336	0	1179.086
2012 09-13 02:00	887.602	−114.823	0.579	0	1179.083
2012 09-13 04:00	886.987	−112.38	0.41	0	1179.082
2012 09-13 06:00	886.987	−114.248	0.616	0	1179.08

The main objectives of MLR are explanation and prediction. After the above two stages, the relationship between the lake level and the predictor variables is represented by the following equation:

$$y_i = b_0 + b_1 \times x_{1i} + b_2 \times x_{2i} + ... + b_p \times x_{pi} + e_i \tag{1}$$

where y_i is the predicted lake level values at time i, x_{1i} to x_{pi} are p predictor variables influencing lake level at time i, b_0 is the constant obtained from data training procedures, b_1 to b_p are the coefficients relating the p predictor variables to the variables of interest, and e_i is a random error term at time i.

The least squares criterion is applied to estimate the Eq. 1. The relationship between the nine predictor variables and the lake level as shown in Table 1 is expressed as follows:

$$\begin{aligned} H = b_0 + b_1 \times TA + b_2 \times TS_{10} + b_3 \times TS_{20} + b_4 \times RH + b_5 \times WD \\ + b_6 \times AP + b_7 \times R_n + b_8 \times WS + b_9 \times P \end{aligned} \tag{2}$$

2.2 Artificial Neural Networks (ANN)

Artificial neural network is a simple and efficient neural network and has been widely used in data fitting, prediction and classification. In this study, no matter how complicated the relationship between the data, ANN models can obtain satisfactory lake level values.

Neural Network (NN). Figure 1 illustrates a neural network with three layers. In each layer the circle with solid line represents a neuron that is a simple computational unit and has an input and output, denoted as z and a, respectively. Suppose that the number of layers of a neural network is L. The $z_k^{(l)}$ and $a_k^{(l)}$ denote the input and output of the kth neuron on the lth layer, respectively.

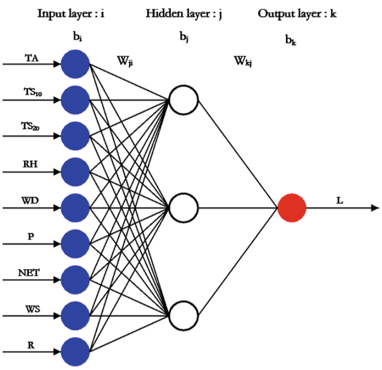


Fig. 1. Artificial neural network

The relationship between an input and output of neurons is usually described by an activation function $g(.)$ as follows:

$$a = g(z) \tag{3}$$

The sigmoid activation function:

$$g(z) = \frac{1}{1 + exp^{-z}} \tag{4}$$

or the hyperbolic tangent activation function:

$$g(z) = \frac{exp^{z} - exp^{-z}}{exp^{z} + exp^{-z}} \tag{5}$$

can be specified.

For the output layer about lake level, a linear transfer activation function:

$$g(z) = z \tag{6}$$

is adopted, which can avoid and correct the gradient disappearance problem.

A neuron receives signals from every neuron on the previous layer as the following:

$$z_k^{(l+1)} = \sum_{i=1}^{n_l} w_{ki}^{(l)} a_i^{(l)} + b_k^{(l)} \tag{7}$$

where $l \in \{1, \mathrm{L}-1\}$, and $w_{ki}^{(l)}$ describe the relationship between the kth and the ith neurons on the $(l+1)$th and lth layers,respectively; $b_k^{(l)}$ is the bias associated with the kth neuron on the $(l+1)$th layer, and n_l is the number of neurons on the lth layer.

In our ANN model, the output of a neuron used to explain the lake level is the same with its input:

$$z_k^{(1)} = a_k^{(1)} (k \in \{1, n_1\}) \tag{8}$$

The output of the last layer can be denoted as $a^{(L)}$:

$$a^{(L)} = (a_1^{(L)}, \ldots, a_{n_L}^{(L)}) \tag{9}$$

Suppose $x = (x_1, \ldots, x_{n_1})^T$ is a representation of lake level predictor records in this paper, if the x is an input into a neural network:

$$z^{(1)} = x \tag{10}$$

a lake level output $a^{(L)}$ can be computed by this network (Eqs. 3 and 7). Therefore, a neural network implements a non-linear mapping $h_{w,b}(.)$ from an input $x = (x_1, \ldots, x_{n_1})^T$ to an output $a^{(L)}$:

$$a^{(L)} = h_{w,b}(x) \tag{11}$$

where

$$b = b^{(l)} \tag{12}$$

is the set of biases, and

$$W = \{W^{(l)}\}(l \in \{1, L\}) \tag{13}$$

is the set of the weights of a NN in Eq. 7, where $b^l = \{b_j^{(l)}, 1 \leq j \leq n_l\}$ and $W^{(l)} = W_{ji}^{(l)}$.

A Back-Propagation Algorithm for Obtaining a NN. Suppose that

$$S = \{(x, y)\} \tag{14}$$

is a training set for a NN, where $x = (x_1, \ldots, x_{n_1})^T$ is the representation of a water level record, and y is the expected lake water level output with respect to x.

In a NN, some parameters W and b can be set through minimizing an objective function, namely, J, as presented by Eq. 15:

$$\mathbf{J}(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum\nolimits_{x \in S} (\frac{1}{2} \|h_{W,b}(x) - y\|^2) + \frac{\lambda}{2} \sum\nolimits_{l=1}^{L-1} \sum\nolimits_{i=1}^{n_l} \sum\nolimits_{j=1}^{n_{l+1}} (w_{ji}^{(l)})^2 \tag{15}$$

where N is the number of samples in a training set S, and $\lambda \geq 0$ is a preset parameter. λ is commonly referred to as a weight decay parameter.

To obtain our NN from a training set, we initialize each parameter $w_{ji}^{(l)}$ and $b_i^{(l)}$ to a small random value near zero. Subsequently, two parameters W and b are iteratively optimized using a gradient descent method based on the objective function J in Eq. 15. This learning scheme is referred to as Back-Propagation (BP) algorithm.

2.3 Deep Learning (DL)

Deep learning is synonymous with deep neural networks (DNNs). In recent years, the DL model is adopted for solving the regression problems in several research areas. The combination of DL and lake data makes the problem better handled. In this paper, we abstract the more useful features by creating a multi-level and multi-neuron neural network, called the fully connected deep neural network, which automatically learns more appropriate weights and thresholds based on the structure of lake level data. The DNN basic structure is shown in Fig. 2.

This article trains the DL model mainly through the following two processes:

- First, our DL network uses the top-down supervised learning, which can be seen as a feature leaning process. Due to the constraints of the model capacity and the sparseness constraints, our DL model can learn the structure of the data itself. If the $n-1$ layer is obtained (Eq. 7), the output of this layer is the input as the nth layer (Eq. 8), and the nth layer is trained (Eq. 7) similarly.

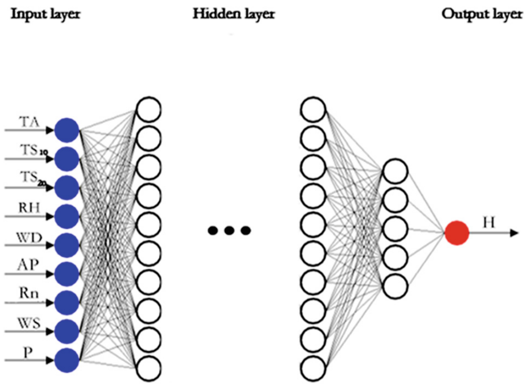


Fig. 2. Deep learning model.

– Then, the top-down and non-supervised learning is used in our DL network. The entire multi-layer model parameters are further fine-tuned (Eq. 15) based on the parameters obtained in the first step. Importantly, initial parameters of DL are not the same as the setting way of NN. In our DL model, the initial value corresponds to the global optimum by learning the structure of the input data, so that better results can be achieved.

2.4 Model Evaluation Criteria

The accuracy of the approximated three models' results is evaluated using the average relative error ($ARER$), the mean squared error ($RMSE$) and the coefficient of determination (R^2), and they are specified as follows:

- $ARER$: It reflects the overall forecast level of the data.

$$ARER = \overline{RER_i} \tag{16}$$

  where RER_i is the radio between the absolute error of the index and the true value.
- $RMSE$: It is used to quantify the simulation results.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(e_i - t_i)^2}{n}} \tag{17}$$

- R^2: It indicates that there is a variation between the predicted value and the true value.

$$R^2 = (\frac{\sum_{i=1}^{n}(e_i - \overline{e_i})(t_i - \overline{t_i})}{\sqrt{(e_i - \overline{e_i})^2(t_i - \overline{t_i})^2}})^2 \tag{18}$$

  where e_i and t_i represent the model output and measured actual lake level value. $\overline{RER_i}$, $\overline{e_i}$ and $\overline{t_i}$ represent their average values, respectively, and n denotes the number of observations.

3 Study Area and Prediction Models Evaluation

3.1 Studying Area and Data Collection

As a typical arid region, the Badain Jaran Desert (BJD) is famous in the world for the presence of a number of lakes among the mega dunes. Due to the huge gap between precipitation and evaporation, the formation mechanism and evaluation trend of lakes become the important scientific problem concerned by researchers. Over the past decades, this issue has never reached a consensus, though a large number of scientists have conducted the scientific inquiry and survey. Therefore, to understand the formation mechanism of lakes, lake level fluctuations should be studied clearly. As shown in Fig. 3, the BJD (39°20′–41°30′N, 100°01′–103°10′E) is located in the western Alxa Plateau in Inner Mongolia, China. It is the second largest desert in China and covers an area of $4.9 \times 10^4\,\text{km}^2$ [5]. Consistent with the desert terrain, the overall flow of groundwater is from south to north and from east to west with hydraulic gradient between 0.8% and 7.9% [6]. About 100 lakes lie in the hinterland in BJD, but the lake area is generally less than $0.2\,\text{km}^2$. A few of them are larger than $1\,\text{km}^2$. The lakes in BJD are most salty with TDS between 1 g/L and 400 g/L. The lake studied in this presentation is the second largest salt lake, Sumu Barun Jaran, with an area of $1.24\,\text{km}^2$ and maximum depth more than 11 m [7].

In order to monitor the meteorological factors in Sumu Baran Jaran, a wooden bridge with steel structure was built in the lake to install an automatic weather station (Fig. 3c) since 2012, where the precipitation (P) is monitored by a self-recording pluviometer. The automatic weather station is used for measuring air temperature (TA), air relative humidity (RH), wind direction (WD), atmospheric pressure (AP), net radiation (R_n) and wind speed (WS). Water

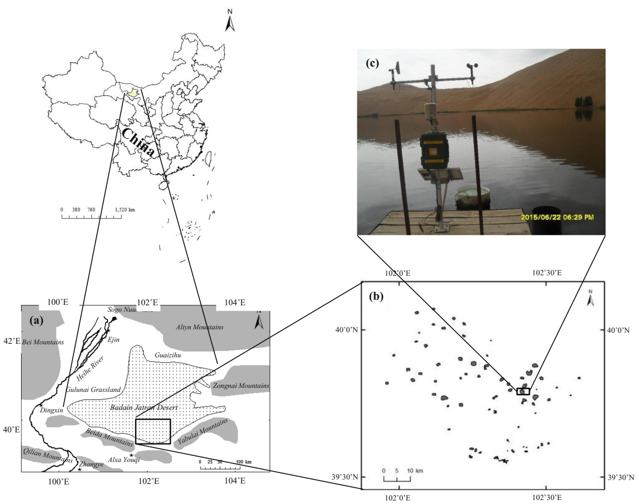


Fig. 3. Location of the study area (a), lakes (b) and weather station (c).

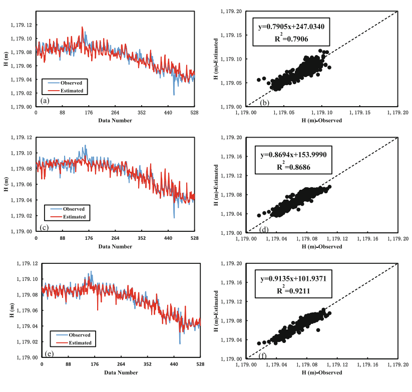
temperature at 10 cm (TS_{10}) and 20 cm (TS_{20}) depths are monitored by two sensors. The monitoring interval is 30 min. Near the bottom of the lake, a CDT-Diver sensor is installed to measure the conductivity, temperature and pressure once every 2 h. After air pressure calibration, lake level (H) can be calculated using the pressure monitored by CDT-Diver sensor. This study collected the meteorological data and lake level data during the period from 00:00 on September 13, 2012 to 10:00 on October 28, 2012 with the interval of 2 h.

3.2 Model Application

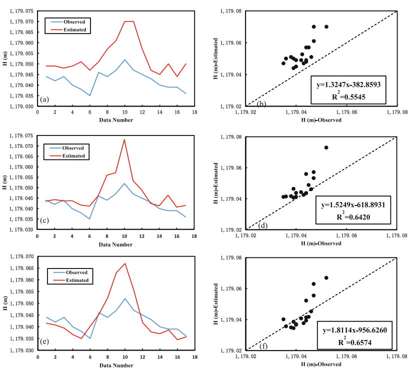
Training Models. In the DL network configuration, we build a six-layer deep neural network (including the input and output layers) as shown in Fig. 2. This network is a fully connected network. In addition, the number of neurons on the hidden layer is 10, 15, 10 and 5, respectively. The activation function of the neurons on the hidden layer is the hyperbolic tangent function. For the activation function of the output layer, rectified linear units function is applied. Importantly, our DL model supports the backward propagation and uses the random gradient descent algorithm for achieving the optimization of the weight.

In addition, the MLR and ANN models are applied to this problem for comparison. We carry out the MLR model training by giving the existing historical meteorological and lake level data, and the trained model is used to predict the future lake level changes. Nine variables serve as predictor variables as shown in Table 1. The ANN model is performed with a three-layer BP neural network as shown in Fig. 1, which is constructed by the LM optimization algorithm (Eq. 15) using MATLAB.

In this study, 528 hourly data during the period from 00:00 on September 13, 2012 to 22:00 on October 26, 2012 are used for the training purpose, and the result has been shown in Fig. 4(a).



(a) Training period.



(b) Testing period.

Fig. 4. Observed and estimated lake level from 00:00 on September 13, 2012 to 22:00 on October 26, 2012 using MLR in (a) and (b), ANN3 in (c) and (d) and DL in (e) and (f).

Testing Models. The trained models are tested by predicting the lake level using 17 h data ranging from 00:00 on October 27, 2012 to 8:00 on October 28, 2012 as shown in Fig. 4(b).

3.3 Model Evaluation

In the training process, these three models can generate the accurate value of the average lake level (1179.07 m). In the testing procedure, on average, the lake level of observation is 1179.04 m, whereas the value calculated by MLR, ANN3, and DL are, 1179.05 m, 1179.05 m, and 1179.04 m, respectively. The average value modeled by DL is equal to the observation. On the whole, the MLR and ANN3 significantly overestimate the lake level for the testing period, while the DL performs the best.

The results of these three criteria for each of the models in the training and testing procedure are presented in Table 2. In the training process, the DL model obtains the highest value of R^2 (0.9211) and the smallest values of AREA (3.19E-06) and RMSE (0.005007). In the testing procedure, these models are acceptable according to the values of the ARER, RMSE and R^2. The DL model performs the best with the highest value of R^2 (0.6574) and smallest values of AREA (4.06E-06) and RMSE (0.001238).

Table 2. Models results

Model	ARER		RMSE		R^2	
	Training	Test	Training	Test	Training	Test
MLR	5.26E-06	8.59E-06	0.007983	0.011190	0.7906	0.5545
ANN3	4.27E-06	4.17E-06	0.006315	0.001316	0.8686	0.6420
DL	3.19E-06	4.06E-06	0.005007	0.001238	0.9211	0.6574

3.4 Discussion

Although the DL model established in this study performed better than the MLR and ANN3 models, the advantage is not very obvious because of a relatively small amount of data used in this study. The experimental settings should be considered for constructing the DL model, since these settings should affect the capabilities of the model to some extent in the training and testing processes, which should be set properly according to our requirements.

– *How to design a network structure.* The optimal number of hidden layers and neurons in each layer should be gotten according to the experience or comparative experimental results.

– *Selection of the activation function.* In this study, the linear activation function (Eq. 6) is selected to set the threshold as zero, which should significantly improve the convergence speed of the random gradient descent algorithm.
– *Selection of the optimization algorithms.* The stochastic gradient descent algorithm is chosen in this paper. In the large sample size circumstances, the samples only needs to be partially trained and this strategy can get a loss value within the acceptable range of the model.
– *How to set the rate at which the optimization algorithm moves in the search space.* If the learning rate is too large, it is possible to cross the optimal value. Otherwise, if the learning rate is too small, the efficiency of optimization may be too low, and the algorithm can not converge for a long time.

To summarize, a relatively simple framework of deep learning has been constructed in this study.

4 Conclusion

In this paper, we attempt to compare the DL model with the MLR and ANN3 models in simulating the hourly lake level dynamics in the Sumu Barun Jaran lake, Badain Jaran Desert, China. The performance of these models is evaluated with criteria including the average relative error, the mean squared error, and the coefficient of determination. Results indicate that the DL model performed the best on all of these criteria, and it has the potential to simulate and predict water level dynamics in rivers and groundwater systems. As the number of training samples increases, the DL model behaves better than other machine learning models with more efficient operation.

References

1. Gong, Y., Wang, X., Hu, B.X., Zhou, Y., Hao, C., Wan, L.: Groundwater contributions in water-salt balances of the lakes in the Badain Jaran Desert, China. J. Arid Land **8**(5), 694–706 (2016)
2. Nagesh, K.D., Rajib, M.: Bayesian dynamic modelling for nonstationary hydroclimatic time series forecasting along with uncertainty quantification. Hydrol. Process. **22**(17), 3488–3499 (2008)
3. Wei, C.-C.: Comparing lazy and eager learning models for water level forecasting in river-reservoir basins of inundation regions. Env. Modell. Softw. **63**, 137–155 (2015)
4. Li, X.-R., Pan, R.-Y., Duan, F.-Q.: Parameterizing stellar spectra using deep neural networks. Res. Astron. Astrophys. **17**(4), 036 (2017)
5. Wang, T.: Formation and evolution of Badain Jirin Sandy Desert, China. China J. Desert Res. **10**(1), 29–40 (1990)
6. Jing, Z., Wang, X.S., Jia, F.C., Guo-Min, L.I., Dong, Y.H.: New insights into the flow directions of groundwater in Western Alxa, Inner Mongolia. Geoscience **35**(3), 774–782 (2015)
7. Chen, T., Wang, X., Hu, X., Lu, H., Gong, Y.: Clines in salt lakes in the Badain Jaran Desert and their significances in indicating fresh groundwater discharge. J. Lake Sci. **27**(1), 183–189 (2015)