"Science Stays True Here"
Advances in Ecological and Environmental Research (ISSN 2517-9454, USA), 217-237 | Science Signpost Publishing

Comparison of Levenberg Marquardt and Conjugate Gradient Descent Optimization Methods for Simulation of Streamflow Using Artificial Neural Network

Thabo Michael Bafitlhile, Zhijia Li, Qiaoling Li

College of Hydrology and Water Resources, Hohai University, Nanjing 210098, P. R. China.

Received: August 08, 2018 / Accepted: September 12, 2018 / Published: November 25, 2018

Abstract: Prediction of absolute extreme flood peak discharge is a crucial research topic for hydrologists because it is essential in developing the best management practices, addressing water-related issues like flood warning, mitigation schemes planning, management and operation of water resources development projects, etc. The primary purpose of this study was to develop Artificial Neural Network Model (ANN) that can accurately predict Changhua streamflow using hourly data for flood events that occurred between 07/04/1998 to 16/04/2010. Zhejiang province is one of the areas in eastern of China that is prone to severe weather, including heavy rain, thunderstorms, and hail. Since 2011 Zhejiang province has continuously been hit by torrential rain which has left many deaths, loss of property and direct economic loss. Therefore, since Qingshandian reservoir function as a power generator and as flood control system, prediction of the downstream flow of Changhua River is vital for improving the management of the reservoir. Rainfall data from seven stations were used as inputs to the ANN model, and streamflow data were used as the desired outputs of the ANN model. ANN is one of the artificial intelligence method attempting to copy the human brain functioning. It acquires knowledge through a learning process that involves the shifting of connection weight and changing bias parameters to determine the optimal network. Levenberg Marquardt Algorithm (LMA) and Conjugate Gradient Descent (CGD) optimization methods were used to train ANN. The performance of the two algorithms was measured using Residual Standard Error (RSE), R squared, Nash–Sutcliffe Efficiency (NSE) and Pearson's Product Method (PPM). The overall results show that CGD method is the best method for simulation of Changhua streamflow as compared to LMA.

Key words: Artificial neural network, Conjugate Gradient Descent, Levenberg Marquardt, Streamflow simulation

Corresponding author: Thabo Michael Bafitlhile, College of Hydrology and Water Resources, Hohai University, Nanjing 210098, P. R. China.

1. Introduction

Waterway streams are the collective results of climatological and geographical factors which interact within the drainage basin. This integration indicates that river flows are the most directly appropriate components in the hydrological cycle. River flow prediction and forecasting are crucial research topics for hydrologists as are extensively relevant in addressing water-related issues. These include flood warning and mitigation schemes, planning, assessment, management and operation of water resources development projects, irrigation systems, the design of water-related structures and also help in climate change strategies. Also, real time river flow data are crucial for assessing and developing hydropower potential and enhancing both the ecological health of watercourses and wetlands.

Pramanik & Panda (2009)) used LMA, Gradient Descent, and CGD algorithm with ANN and used an adaptive neuro-fuzzy inference system (ANFIS), ANFIS slightly performed better than all the algorithms used with ANN. The work by Pramanik & Panda (2009) shows that ANNs can be able to handle complex hydrological processes as reservoir release, tributary discharge and barrage outflow daily data were used for river flow prediction, and best results were yielded out of ANN model. ANNs have demonstrated its ability due to withstanding performance it has displayed when relating and predicting a runoff ordinate to the pattern of antecedent rainfall depths (Minns & Hall 1996). Performance of ANN models and wavelet ANN hybrid models were compared for 1, 3, 5, 7 days ahead forecast of daily stream flows. The models were also found to be efficient in the elimination of the lags observed in the forecasting of daily flows using ANN models (Santos & Silva 2014). The results obtained here, although limited to a single application, demonstrate and quantify the benefits of wavelet transform used in the forecasting of daily streamflows using ANNs.

Chau, et al (2005) discussed that ANN coupled with Genetic Algorithm (ANN-GA) and ANFIS models produced accurate flood predictions of the channel reach between Luo-Shan and Han-Kou station of the Yangtze River as both models were able to avoid the overfitting problem. Amongst them, the ANFIS model is the best regarding simulation performance and appears better suited to the flood forecasting environment. Also, Al-aboodi et al (2017) compared ANFIS model against ANN, and Auto Regressive Integrated Moving Average (ARIMA) and the author also found that ANFIS displayed a significant improvement in performance when comparing it with ARIMA and slightly better than the ANN model. Four methods were used for streamflow prediction ANN, ANFIC, Multi Linear Regression (MLR) and Hermite Polynomial Projection Pursuit Regression (PPR). Social Spider Optimization (SSO) algorithm and Least Square (LS) Method were used to optimize the Hermite-PPR parameters. ANN was trained with LMA. The results indicated a promising role of the new H-Hermite PPR model with SSO and LS algorithm in annual maximum flood peak discharge forecasting; the results show a higher performance of H-PPR model over ANFIS, ANN, and MLR models in Yangtze River.

Study Area and Data

The Fenshui River is located in the northwest of Zhejiang Province, a tributary of Qingtang River, originated on the Yunshan ling Mountain of Jixi County, Anhui Province. The total area of Changhua river basin is 3442km2, mainstream length of 1624km, at an elevation of 965m. It is a subtropical monsoon climate with abundant rainfall and significant rainfall variation with an annual rainfall of 1638.2mm. During the spring season from March to early April, the southeasterly wind prevailed on the ground surface, and the rainfall generally increases and the amount of precipitation gradually increase. During the period from May to July, the frontal surface often stagnates or swings over the watershed, resulting in continuous rainfall with considerable rainfall intensity and long rainy seasons. During the summer months of July and September, the weather is hot with prevailing southerly thunderstorm and typhoon rainfalls. From October to November, the weather is sunny mainly, from December to February. Winter season the prevalence of the northerly winds on the ground, cloudy weather, temperatures are low, rain and snow weather.

There are twenty small reservoirs in the Fenshui River Basin with a total storage capacity of 62 million m3 and an entire catchment area of about 200km2. Most of them are mainly for irrigation and hydropower with no flood control capacity. Within the Fenshui River basin, there are two large reservoirs among them one is located in Changhua River upstream of Fenshui River, with a catchment area above 1429km2. The project has such functions as power generation and flood control.

LMA and CGD methods were separately once used before for river flow prediction using daily streamflow data. LMA was used for its ability to converge very fast as compared to other optimization algorithms, has been proven to be one of the best methods to solve regression problem. CGD method too provides more accurate results but it is slow to converge, yet it performs faster than backpropagation. Since we are working with a time series or a regression problem, the two methods are suitable to be used to develop a multilayer perceptron (MLP) neural network model to simulate streamflow. This research endeavored to use the two approaches to train the MLP neural network using hourly rainfall and streamflow data, on the different neural network architectures of a different number of neurons in the hidden layer.

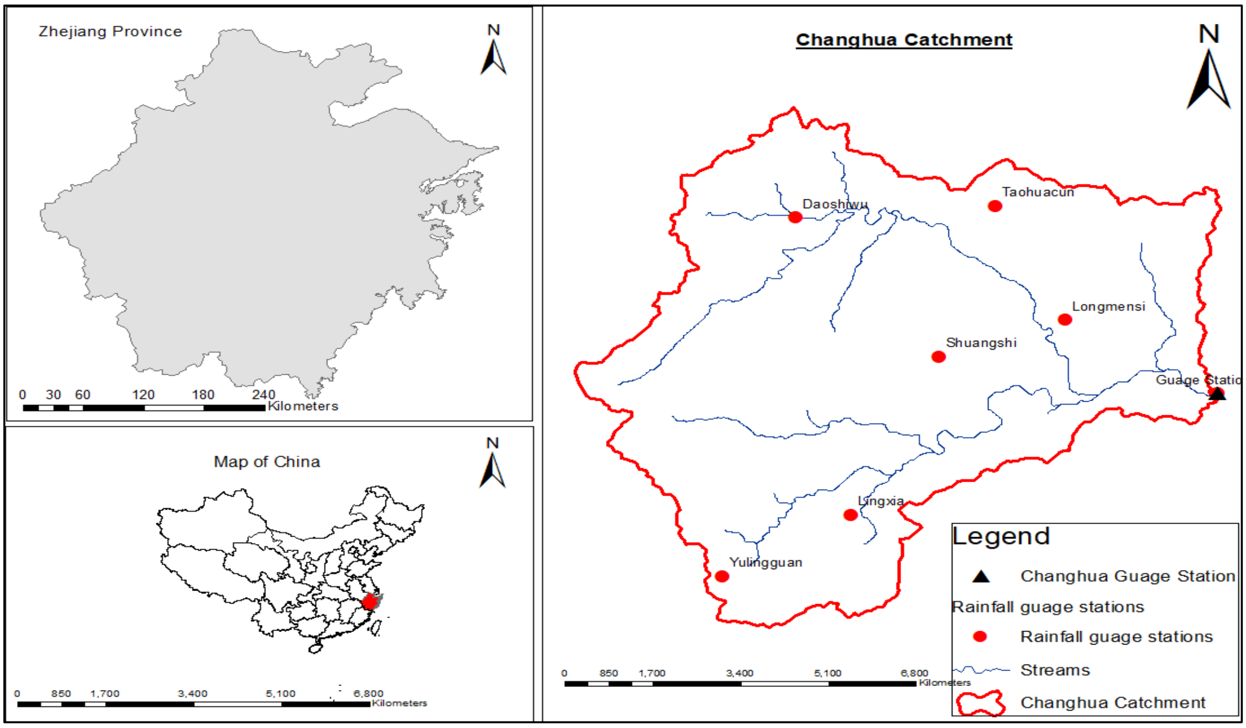


Fig 1 Changhua river basin

Zhejiang province is one of the areas in eastern of China that is prone to severe weather, including heavy rain, thunderstorms, and hail. Since 2011 Zhejiang province has continuously been hit by torrential rain which has left many deaths, loss of property and lastly direct economic loss. Therefore, since Qingshandian reservoir function as a power generator and as flood control system, prediction of the downstream flow of Changhua River is vital for improving the management of the reservoir.

River flow modeling can be complicated because of the physical process involved in the generation of river flow. Many types of research have been carried out on streamflow analysis, but this is a complex area of study because of its non-linearity, its high degree of spatial and temporal variability from different factors such as catchment, storm, geomorphologic and climate characteristics. This complexity is encountered when using distributed physically based models to estimate runoff (Finger, et al 2015). The estimation of rainfall-runoff is a big challenge because of many stochastic processes involved. Consequently, the streamflow prediction turns out to be a challenging process because is directly related to runoff. These include estimation of initial soil moisture content, infiltration rate, evapotranspiration, vegetation intercept, etc. Hence these affects the accuracy of the model because some factors end up being assumed because of limited data.

The paper proposes the development of ANN model for simulation of Changhua streamflow. The main objective was achieved through the following objectives; Data preprocessing, data transformation by normalizing it. Development of ANN, using LMA and CGD to fit the ANN model to data and then determine which method best fit the model. The two algorithms receive a set of inputs along with the corresponding target, the algorithm then learns by comparing its actual output with the desired output to find errors. It then modifies the model accordingly by jogging the connection weight to improve the network performance to reduce errors and also to avoid overfitting. Measures of performance; RSE, R squared, NSE and PPM were used to determine the goodness of fit of the model.

Since, a significant number of physical models have been developed and still not performing well also most of them are not universal. Analysts have now fixed their interest on the use of data-driven models (DDM) which endeavor to create a connection among input and output variables encompassed in a physical process without acknowledging the underlying physical processes (Al-aboodi et al. 2017).

This method alludes to DDM. DDM methods incorporate; artificial intelligence (AI), data mining (DM), knowledge discovery in databases (KDD), computer intelligence (CI), machine learning (ML), intelligent data analysis (IDA) and soft computing (SC). These areas coincide on their application field and can be acknowledged as a methodology centering on the utilization of ML like ANN and Support Vector Machines (SVM), etc (Mishra 2013).

Rainfall-Runoff models that can be successfully calibrated using relatively less and small data are desirable for any basin in general. Therefore more researches are being carried out on hydrological forecasting using ANN (Sajikumar & Thandaveswara 1999). The idea of ANNs is motivated by the living neural networks that depend entirely on the central nervous system, the brain as the leading role in the control of the most bodily function.

Hourly rainfall and streamflow data for Changhua catchment from seven rainfall stations (Changhua, Longmengsi, Shuangshi, Daoshiwu, Lingxia, Yulingguan) and Changhua discharge site along Changhua River, located in Zhejiang province, China were used. A total of 24 storm events that occurred from 7/4/1998 to 14/4/2010 were selected. Table1 shows the duration and dates of these storms. Hourly rainfall data were used as input, and the streamflow data was the desired output of the model. The supervised fully connected feedforward neural network was considered in this study. Only one hidden layer MLP's was used.

Table 1: Excerpt of selected storm events

Flood No	Start time	End time	Duration
1	4/7/98 0:00	4/11/98 0:00	1/4/00 0:00
2	5/13/98 0:00	5/18/98 21:00	1/5/00 21:00
3	7/22/98 21:00	7/24/98 18:00	1/1/00 21:00
21	6/26/08 12:00	6/29/08 10:00	1/2/00 22:00
22	7/26/09 10:00	8/4/09 2:00	1/8/00 16:00
23	2/25/10 8:00	3/10/10 0:00	1/12/00 16:00
24	4/14/10 0:00	4/16/10 19:00	1/2/00 19:00

Artificial Neural Network

ANN is one of the artificial intelligence method attempting to copy the human brain functioning. It acquires knowledge through a learning process that involves the shifting of connection weight and changing bias parameters to determine the optimal network(Wu & Chau 2011). ANN are DDM capable of forecasting because of their flexibility and nonlinearity (Kourentzes, et al 2014). They have demonstrated to be the general approximators (Kourentzes et al 2014), being able to fit any data. ANN has been empirically shown to be able to forecast both linear (Zhang 2001) and nonlinear (Zhang, et al 2001) time series of different forms. Their attractive properties have led to the rise of several types of ANN and application in the literature.

DDM allows ANN great flexibility in modeling time series data because of their robust approximation and self-adaptive capabilities (Toth & Brath 2007). Direct optimization through conventional minimization of error is not possible under the multilayer architecture of ANN, and the back-propagation learning algorithm has been recommended to solve this problem when used with MLP. The selection of a suitable architecture for an ANN is determined by the problem to be solved and the type of algorithm to be applied (Minns & Hall, 1996). Vigilance is crucial when increasing the size of hidden layers because it results in overfitting problem, the adaptive structure of MLP is necessary for real problems (Gaurang, et al 2011). However, when applied more generally to systems identification, the purpose is to train an ANN to provide a correct output response to the measured runoff from a catchment (Minns & Hall, 1996). A multilayer, feedforward, perceptron type is one of the most suitable kinds of ANN for learning the stimulus-response relationship for a given set of measured data (Minns & Hall 2004; Minns & Hall 1996).

The architecture of ANN is in such a way that each input neuron in the hidden layer is joined to every single element of the input vector through the weight matrix. The outputs of the hidden layer are the input to the next layer and are connected to each other by a weight vector (Tyagi & Panigrahi, 2017).

$$y_t = \alpha_0 + \sum_{h=1}^{H} \alpha_H \, g\left(\beta_{0j} + \sum_{j=1}^{P} \beta_{ij} y_{t-1}\right) + \varepsilon_t$$

Where α_h and β_{ij} are the model parameters (connection weights); p is the number input nodes depending on the number of variables used as inputs and H is the number of hidden nodes in the hidden layer, $y_{(t)}$ is predicted flow, $y_{(t,-1)}$ is observed flow, ε_t is error componentlg. Logistic and Hyperbolic functions are normally used as transfer functions in the hidden layer to estimate nonlinear time series and can also be used in the output layer depending on the nature of the problem.

Logistic Function; $F(x) = \frac{1}{1+e^{-x}}$ ranges between (0 and 1).

Hyperbolic function $F(x) = \frac{1}{1+e^{-x}}$ ranges between (-1 and 1).

In supervised learning, inputs and targets are both provided. The network then processes the inputs and compares the results of the model against the target (Al-Abadi 2016; Khan et al. 2016). The error is computed then propagated back to each weight in the network, causing the system to adjust the weights which control the network. The process will go through many iterations as the weight are continually improving to match the desired output result. The set of data which enables training is 'training data,' the data is processed many times as the network is trying to find the right model to match the target. A heuristic function was used to estimate the best number of neurons in the hidden layer. Network training parameters like the hidden and output layer activation functions and output error functions were considered and included for architecture selection as they are less demanding, i.e., have a smaller number of iteration and smaller error limits. Test error fitness criterion was used to determine the best network, the smaller the error on the test set the better the network. This parameter is calculated as inversed mean value absolute network error on the test set. The neurons in the hidden layer were adjusted from two to ten for both LMA and CGD method to find the best architecture of the ANN model. The logistic function is used for all nodes in the hidden layers and output layer as the activation function. The data was normalized and divided into three datasets "training data sets, validation data-sets and test datasets," in a ratio of 68%, 16%, 16% consecutively. Two training optimization methods were used for training a neural network; LMA and CGD. The network stops immediately when validation error starts to rise to avoid overtraining or generalization loss consistently.

To find the best architecture of the ANN model, the neurons in the hidden layer were adjusted from two to ten for both LMA and CGD method. The logistic function is used for all nodes in the hidden layer and output layer as the activation function. The data was normalized and divided into three datasets ('training data sets, validation data-sets and test datasets'), in a ratio of 68%, 16%, 16% consecutively. Two training optimization methods were used for training a neural network; LMA and CGD. The network stops immediately when validation error starts to rise to avoid overtraining or generalization loss consistently.

Levenberg Marquardt Optimization

LMA is known as the damped least squares (DLS) (Majumder 2010). The algorithm was first published in 1944 by Kenneth Levenberg, and it was later rediscovered in 1963 by Donald Marquardt. LMA is used for solving curve fitting problems, is an iterative technique that locates the minimum of a multivariable function that is expressed as the sum of squares of nonlinear real-valued functions (Levenberg 1944; Marquardt 1963). Is now a standard optimization that is broadly used in most of the disciplines. LM algorithm varies the parameter updates between the gradient descent update and the guass newton update; it combines the advantages of both methods (Hariharan, et al 2015).

LM algorithm starts with an initial guess of x_o. x is adjusted by δ only for downhill steps.

In each iteration step, the parameter vector β is replaced by a new estimate of $\beta + \delta$. To determine δ, the function $f(x_i, \beta + \delta)$ is approximated by its linearization.

$f(x_i, \beta + \delta) \approx f(x_i, \beta) + J_i\delta$, where $J_i = \frac{\partial f(x_i,\beta)}{\partial \beta}$, is gradient (row vector) of f with respect to β

The first order approximation of $f(x_i, \beta + \delta)$ gives

$$S(\beta + \delta) \approx \sum_{i=1}^{m} (y_i - f(x_i, \beta) - J_i\delta)^2$$

Taking the derivative of $S(\beta + \delta)$ with respect to δ in setting the result to zero gives

$$(J^T J)\delta = J^T(y - f(\beta)$$

Levenberg contribution is to replace this equation with a damped version

$$[J^T J + \lambda I]\delta = J^T(y - \hat{y})$$

$$[J^T J + \lambda I]\delta = J^T r$$

J = Jacobian matrix of derivatives of the residuals with respect to the parameters

λ = damping parameter (adaptive balance between the two steps)

r = residual vector

The (non-negative) damping parameter is adjusted in every iteration, where small values of the algorithmic parameter λ result in Gauss-Newton update, and large values of λ result in a gradient descent update (Gavin 2013). The parameter λ is initialized to be large so that first updates are small steps in the steepest descent direction (Mehdizadeh et al 2016). If any iteration happens to lead to a poor approximation, then λ is increased.

Therefore, for large values of λ, the step will be taken approximately in the direction of the gradient (Kaufman & Itskovich 2017). Otherwise, as the solution improves, λ is decreased, the LM method approaches the Gauss-Newton method, and the solution typically accelerates to the local minimum.

Levenberg method experiences the drawback whenever the value of damping factor λ is enormous, hence $\boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I}$ is not used at all (Hooshyaripor, et al 2015). Marquardt's contribution to the method is that to achieve a greater movement along the directions where the gradient is smaller, each component of the gradient should be scaled according to the curvature (Wright & Nocedal 1999). This shuns slow convergence in the direction of small gradient (Marquardt, 1963). Therefore, Marquardt substituted the identity matrix $\boldsymbol{I}$ with the diagonal matrix consisting of the diagonal elements of $\boldsymbol{J}^T\boldsymbol{J}$, resulting in the Levenberg–Marquardt algorithm.

$$[J^TJ + \lambda diag(J^TJ)]\delta = J^T(y - f(\beta\ )$$

Conjugate Gradient Descent Method (CGD)

The CGD method solves systems of linear equations, also used to solve system where A (matrix) is not symmetric, not positive-definite, and still not square (Shewchuk 1994). CGD is an advanced method for training multi-layer neural network. It is based on the linear search usage in the line of an optimal network weight change. Weight correction is executed once per iteration. In most cases, this method performs faster than back-propagation and provide more accurate forecasting result (Majumder et al 2010).

CGD also extends to a method for minimizing quadratic functions, which can subsequently be generalized to reduce arbitrary functions. In the CGD method, the line is not searched, but a plane is searched. A plane is formulated from a random linear combination of two vectors. For minimizing quadratic functions, the plane search requires only the solution of a two by two sets of linear equation for α and β (Claerbout 2009).
Solving convex optimization problems using CGD

$$f_{(x)} = \frac{1}{2}x^2 + \frac{1}{3}y^2 + \frac{1}{5}xy$$

Gradient Descent Method will try to find the minimum by computing the gradient of *(f)* at the initial guess. Let $\{d_1 \ldots d_n\}$ be n A-Orthogonal vectors, let χ^* be the minimum of $f_{(\chi)}$.

$$x^* = \sum_{i=1}^{n} \alpha_i A d_i \ \Rightarrow b = Ax$$

Multiply by one of the vectors d_k

$$d_k^T b = \sum_{i=1}^{n} a_i d_k^T A d^i$$

d_i, d_k are orthogonal

$$d_k^T = \alpha_k d_k^T A d_k$$

$$\alpha_k = \frac{d_k^T b}{d_k^T A d_k}$$

Once we have α we can calculate a minimum of f by taking the α linear combination of n Q-orthogonal vectors

Generating n Q-orthogonal vectors, given some initial guess (x_1)

$$d_1 = -\Delta f_{(x_1)} = b - Ax_1 d_1$$

$$\alpha_1 = \frac{-\Delta f_{(x_1)}^T d_1}{d_k^T A d_1}$$

$$x_2 = x_1 + \alpha_1 d_1$$

The gradient of the new guess must first be calculated to calculate the second orthogonal vector

$$g_2 = \Delta f_{x_{(2)}}$$

$$\beta_1 = \frac{g_2^T A d_1}{d_1^T}$$

$$d_2 = -g_2 + \beta_1 d_1$$

In general n-Q, orthogonal vectors can be generated, and the minimum solution can be found as follows given the initial guess,

$$d_1 = -\Delta f_{(x_1)} = b - Ax_1$$

Then calculate the value of α to update a guess

$$\alpha_k = \frac{-g_k^T d_k}{d_k^T A d_k} \Rightarrow x_{k+1} = x_k + \alpha_k d_k$$

Then β is calculated to update the value of the orthogonal vector

$$\beta_k = \frac{-g_{k+1}^T A d_k}{d_k^T A d_k} \Rightarrow d_{k+1} = -g_{k+1} + \beta_k d_k$$

To achieve the value of x close to optimal solution the whole process has to iterate.

2. Results and Discussion

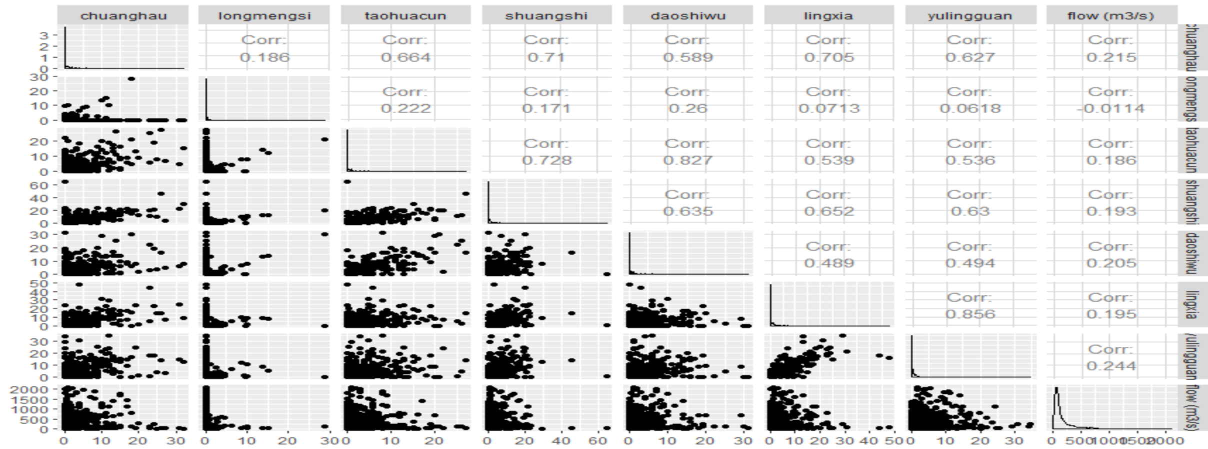


Fig 2 Correlation of rainfall stations with streamflow

The relationship between the inputs dataset (rainfall data) and also the relationship between the input data set and target dataset (discharge). There is a positive relationship between the rainfall data and the discharge except for one station ‘Longmengsi’ which has a negative relationship with streamflow because of less rainfall that is being received in the station as shown in Figure 2, Yulingunn has the highest correlation of 0.244 as compared to other rainfall station.

Table 2: Architecture search

Architecture	no of Weights	Fitness	Train Error	Validation Error	Test Error	R-Squared
[7-1-1]	10	0.0141	5.792	72.203	70.479	0.946
[7-10-1]	91	0.03	37.122	48.849	32.616	0.969
[7-6-1]	**55**	**0.067**	**23.068**	**41.444**	**14.831**	**0.99**
[7-4-1]	37	0.027	37.666	51.355	36.234	0.967
[7-8-1]	73	0.029	35.864	44.456	33.823	0.976
[7-7-1]	64	0.0187	46.248	55.834	53.425	0.964
[7-5-1]	46	0.0238	38.166	44.024	42.002	0.972

The search parameters used to determine the best ANN architecture using Heuristic search method include Inverse Test Error fitness criteria, the single hidden layer was used, and neurons in the hidden layer range from 2 to 20 with a search accuracy of 1. From table 2, seven network architecture were confirmed and [7-6-1] architecture was selected for training hidden layer activation function: Sigmoid curve and output parameters including error function: Sum of Squares and activation function: Linear. Nine ANN models were trained on different architectures based on table 2; figure 3 shows how two methods of ANN model performed under different architectures by raising the number of hidden neurons from two to ten, the coefficient of determination (R2) was used as a measure of performance. LMA did well under 7-6-1 architecture, and CGD performed well under 7-8-1. That is six neurons in the hidden layer for LMA and eight neurons in the hidden layer for CGD method.

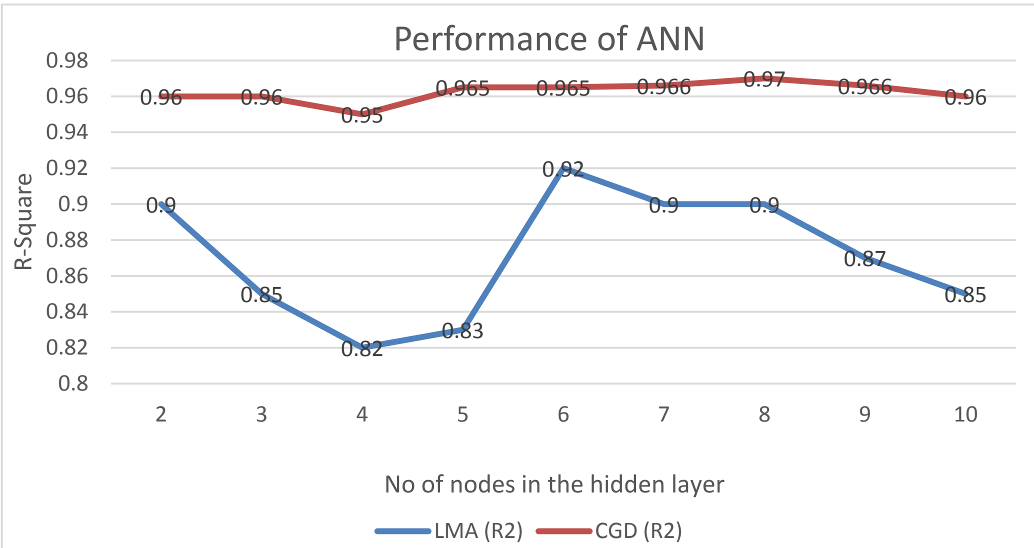


Fig 3 Performance of ANN models against a different number of neurons in the hidden layer.

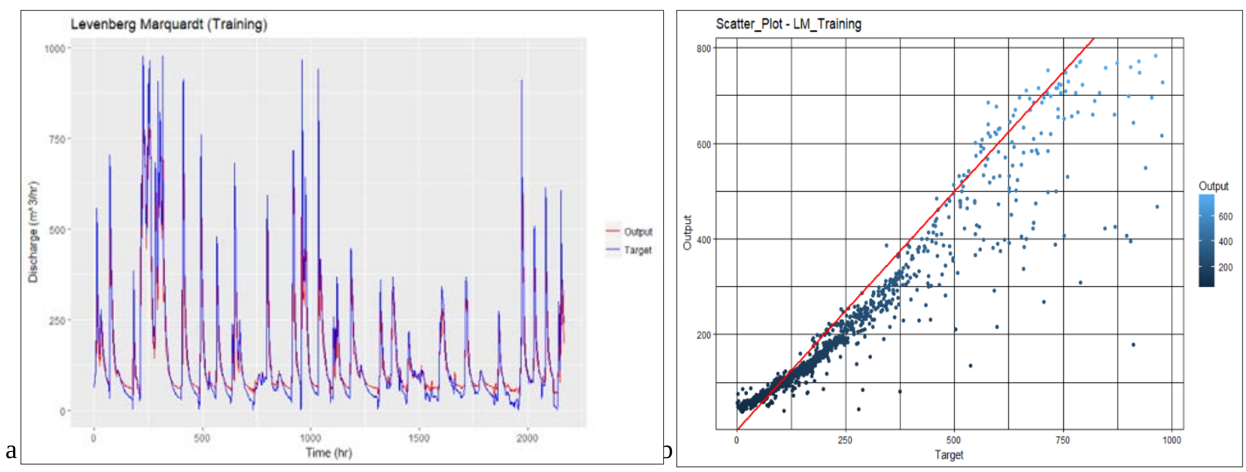


Fig 4 Results produced by ANN model using LMA over training data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.

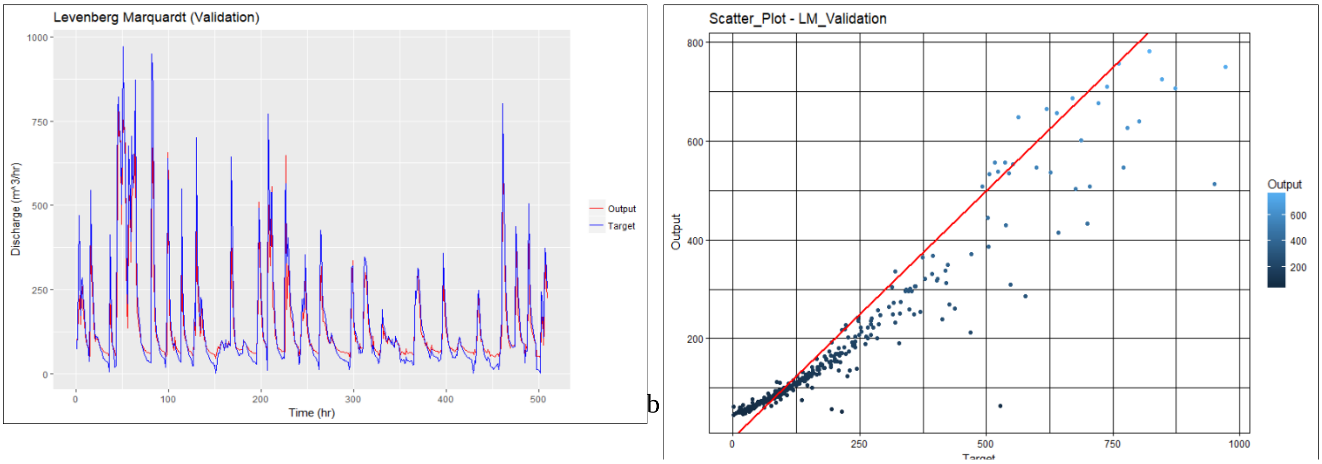
a  b

Fig 5 Results produced by ANN model using LMA over validation data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.

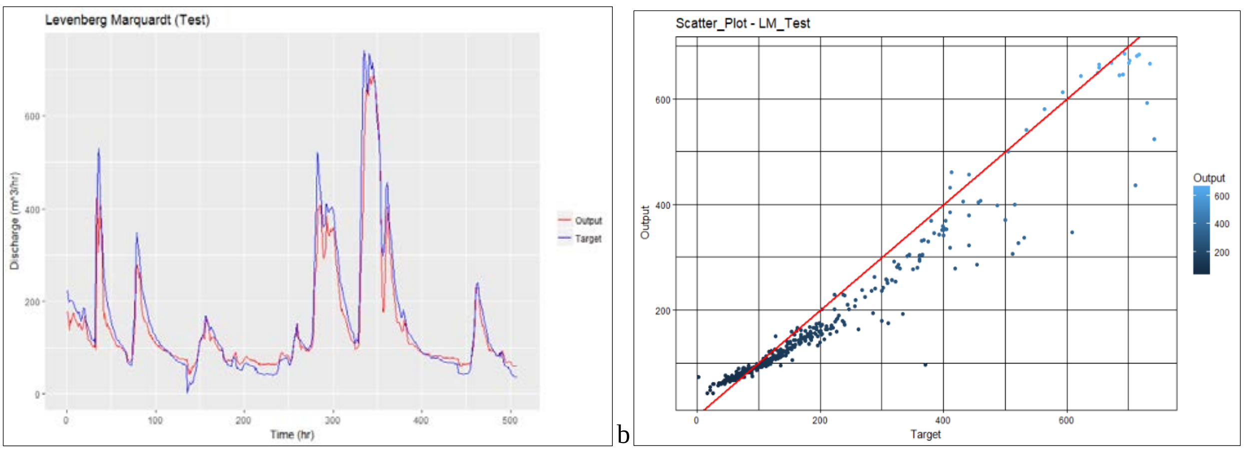
a  b

Fig 6 Results produced by ANN model using LMA over testing data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.

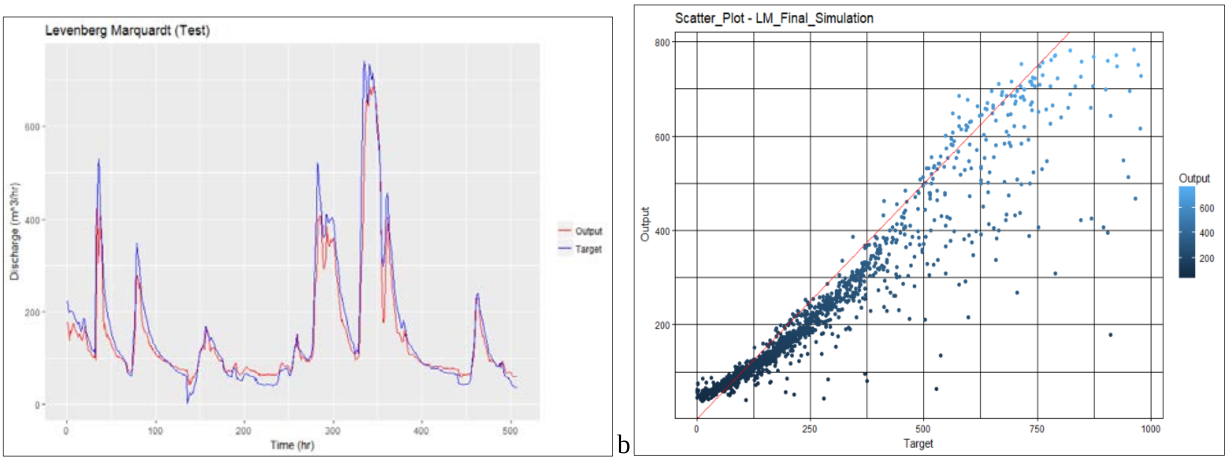


Fig 7 Results produced by ANN model using LMA over the whole data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.

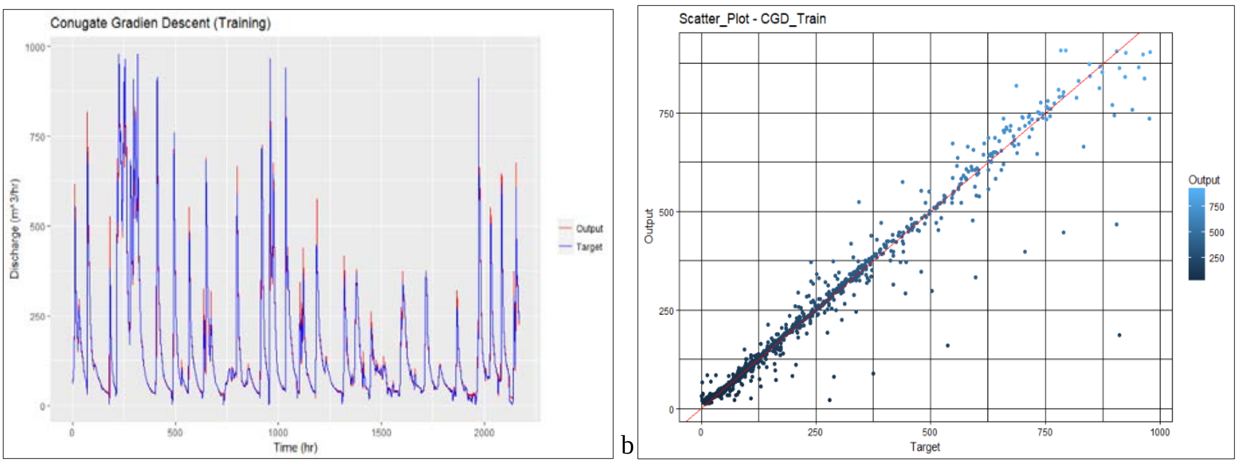
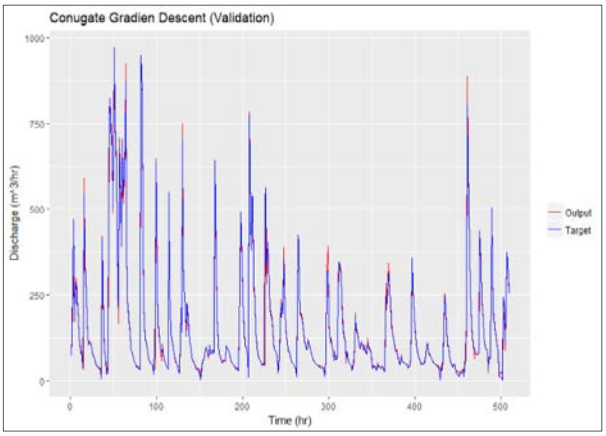
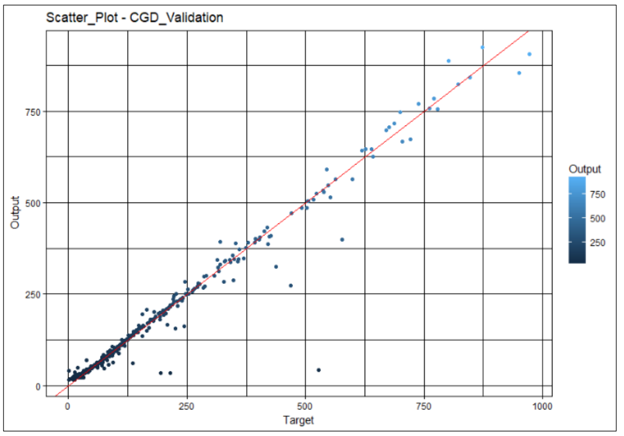


Fig 8 Results produced by ANN model using CGD method over training data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.

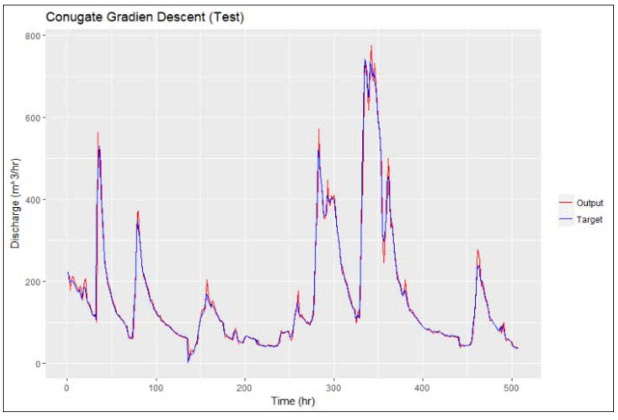


a

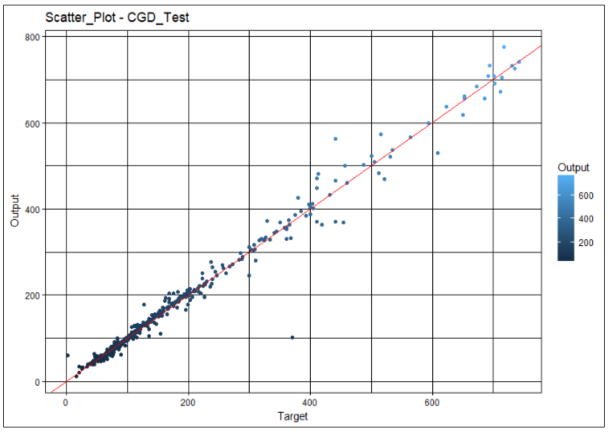


b

Fig 9 Results produced by ANN model using CGD method over validation data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.



a



b

Fig 10 Results produced by ANN model using CGD method over testing data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.

a b

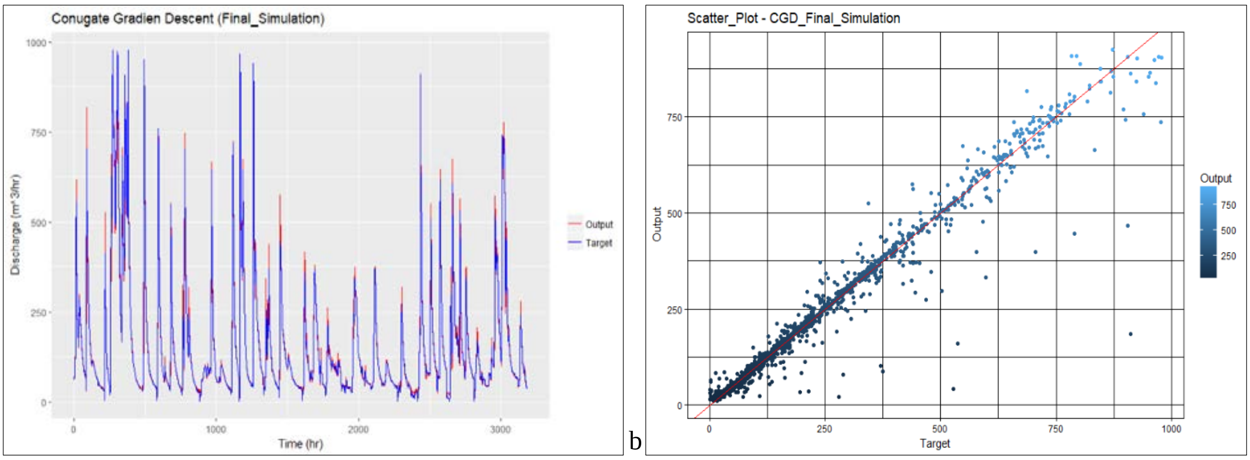


Fig 11 Results produced by ANN model using CGD method over the whole data (a) Time Series simulation plot showing observed and simulated streamflow (b) Scatter plot of observed against simulated streamflow.

Table 3: Goodness of fit results

Measure of Performance	LM Training	CGD Training	LM Validation	CGD Validation	LM Test	CGD Test	LM_Final Simulation	CGD_Final Simulation
RSE	49.04	31.91	48.14	30.53	35.28	18.99	47.05	30
R^2	0.9198	0.966	0.919	0.967	0.942	0.989	0.922	0.97
Nash–Sutcliffe Efficiency	0.888	0.97	0.887	0.97	0.918	0.988	0.89	0.97
Pearson's Product Method	0.96	0.98	0.96	0.98	0.97	0.99	0.96	0.98

Figure (5a, 6a, 7a) present the details of the observed and simulated streamflow for training, validation and testing respectively using Levenberg Marquardt algorithm on MLP ANN model. From the results, the peak discharge was underestimated as shown in table 3 that residual standard for LMA error is very high which is the difference between the predicted values and desired values. Moreover, the coefficient of determination is not satisfactory considering that it measures the goodness of fit of the model. Figure (5b, 6b, 7b) illustrate that the predicted data are poorly fitted to the regression line by ANN model using LMA. LMA poorly performed on training data, and there was a bit of improvement on the testing data because of the validation process as it indirectly affects the weight configuration and also makes sure that the model does not overfit by monitoring the error that is within some range. Initially, the validation error reduces with training error but when the validation error immediately starts to increase the training stops to avoid overfitting. When the validation error is less mostly, the test result will be better. For all that, LMA has not performed very well especially when compared to ANN model using CDG optimization method.

Figure (9a, 10a, 11a) show the observed and predicted stream flow for training, validation and testing respectively using Conjugate Gradient Descent optimization method. Is clear that CGD performed very well even though, it underestimated peak discharge on training data but on validation data and testing data we can plainly see that CGD method has well predicted the actual output. Comparing RSE of CGD with of LMA there is a significant improvement paying attention the test results of both methods that LMA and CGD scored 35.28 and 18.99 respectively. Also, regarding coefficient of determination, figure (9b, 10b, 11b) plainly show how well the predicted flow fit the desired output. From table 3; coefficient of determination ranges from 96.6% to 98.9%, comparing with R-Square for LMA this confirms that CGD indeed outperformed LMA. However, is important to note that a high value of the coefficient of determination does not necessarily indicate that the model has accurately predicted the actual values, but that the fit is highly correlated with data. Therefore, the use of Nash Sutcliffe Efficiency to compare LMA and CGD is crucial since NSE, is used to determine how well the model can predict the actual values. From table1 regarding test result is evident that CGD has simulated observed values much better than LMA, since, CGD is 98.8% an LMA is 91.8%. According to all measures of performance is distinct that they support each other as they have found CGD to have performed far much better than LMA.

LMA using MLP ANN happen to predict actual values when using data recorded daily accurately or more because (Remesan, et al 2009) used daily, and monthly data for forecasting flood using LMA and the authors found LMA to have performed very well. It could be due to the presence of correlation on the data since hourly data mostly show no correlation amongst it. Guimarães (2014) also used daily data and found that LMA performed very well whereas Chaipimonplin & Vangpaisal (2014) compared LM algorithm and Bayesian Regularization (BR) for flood forecasting and they found that BR performed better than LM algorithm. The other reason why LMA did not perform well could be because of the negative correlation of one other input with the stream flow, from figure 2 we can also see that after the sixth neuron, LMA's performance drop because the increase in some parameters in the network affects its performance. But its advantage against CGD method is that is very fast to converge.

3. Conclusion

For Changhua river basin, using data from seven rainfall stations as input to the model and streamflow (discharge) as the desired output of the model. ANN model of 7-6-1 using CGD method is recommended over LM algorithm because it has proved to work well with ANN and has scored high in all method used for measuring the performance of models especially R squared and Nash–Sutcliffe Efficiency. Literature has shown that LM algorithm is a suitable optimization method for flood prediction, but it usually performs well in small-medium size networks, but for large networks, it has a problem of generalization. Further research can be done to determine which rainfall data can be relevant to be used as input data considering their correlation and overall relationship with the streamflow to improve the performance of the network also regarding the transfer function and weight function. Furthermore, the variability of the MLP ANN models performance due to the difference in the number of neurons in the hidden layer need always to be highly considered. However, there is no specific answer on how to determine these components since there are many thumb of rules to use. Most of those rules are not applicable to most conditions as they do not contemplate the training set size, the complexity of the dataset to be learned, etc. It is believed that the best number of hidden neurons depends on various aspects of neural network and the training process.

Acknowledgements

This research was supported by the Research Funds for the Central Universities 2018B11014 and National Key R&D Program of China(Grant No. 2016YFC0402705), and by Natural Science Foundation of China (51679061, 41130639).

References

[1]. Al-Abadi, A. M. (2016). Modeling of stage–discharge relationship for Gharraf River, southern Iraq using backpropagation artificial neural networks, M5 decision trees, and Takagi–Sugeno inference system technique: a comparative study. Applied Water Science, 6(4), 407–420. https://doi.org/10.1007/s13201-014-0258-7.

[2]. Al-aboodi, A. H., Dakheel, A. A., & Ibrahim, H. T. (2017). Comparison of Data-Driven Modelling Techniques for Predicting River Flow in an Arid Region, 12(11), 2647–2655.

[3]. Chaipimonplin, T., & Vangpaisal, T. (2014). Comparison of the Efficiency of Input Determination Techniques with LM and BR Algorithms in ANN for Flood Forecasting, Mun Basin, Thailand. International Journal of Computer and Electrical Engineering, 6(2), 90–94. https://doi.org/10.5963/JWRHE0402002.

[4]. Chau, K. W., Wu, C. L., & Li, Y. S. (2005). Comparison of Several Flood Forecasting Models in Yangtze River. Journal of Hydrologic Engineering, 10(6), 485-491. https://doi.org/10.1061/(ASCE)1084-0699(2005)10:6(485).

[5]. Claerbout, J. F. (2009). Conjugate Gradient for Beginners. Retrieved from http://sep.stanford.edu/data/media/public/oldreports/sep44/44_14.pdf.

[6]. Finger, D., Vis, M., Huss, M., & Seibert, J. (2015). The value of multiple data set calibration versus model complexity for improving the performance of hydrological models in mountain catchments. Water Resources Research, 51(4), 1939–1958. https://doi.org/10.1002/2014WR015712.

[7]. Gaurang Panchal, Amit Ganatra, Y P Kosta, & Devyani Panchal. (2011). Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers. International Journal of Computer Theory and Engineering, 3(2), 332–337. Retrieved from http://search.proquest.com/openview/0940eedf2aa6acad82c310062696507b/1?pq-origsite=gscholar&cbl=2027404.

[8]. Gavin, H. P. (2013). The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering, Duke University, 1–17. https://doi.org/10.1080/10426914.2014.941480.

[9]. Guimarães Santos, C. A., & Silva, G. B. L. da. (2014). Prévision des débits journaliers utilisant des modèles hybrides de transformées en ondelettes et de réseaux de neurones artificiels. Hydrological Sciences Journal, 59(2), 312–324. https://doi.org/10.1080/02626667.2013.800944.

[10]. Hariharan, K., Nguyen, N. T., Chakraborti, N., Barlat, F., & Lee, M. G. (2015). Determination of anisotropic yield coefficients by a data-driven multiobjective evolutionary and genetic algorithm. Materials and Manufacturing Processes, 30(4), 403–413. https://doi.org/10.1080/10426914.2014.941480.

[11]. Hooshyaripor, F., Tahershamsi, A., & Behzadian, K. (2015). Estimation of peak outflow in dam failure using neural network approach under uncertainty analysis. Water Resources, 42(5), 721–734. https://doi.org/10.1134/S0097807815050085.

[12]. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8.

[13]. Kaufman, A. A., & Itskovich, G. (2017). Basic principles of induction logging: electromagnetic methods in borehole geophysics. Elsevier.

[14]. Kelly, C. T. (1999). Iterative Methods for Optimization. SIAM.

[15]. Khan, M. Y. A., Hasan, F., Panwar, S., & Chakrapani, G. J. (2016). Neural network model for discharge and water-level prediction for Ramganga River catchment of Ganga Basin, India. Hydrological Sciences Journal, 61(11), 2084–2095. https://doi.org/10.1080/02626667.2015.1083650.

[16]. Kourentzes, N., Barrow, D. K., & Crone, S. F. (2014). Neural network ensemble operators for time series forecasting. Expert Systems with Applications, 41(9), 4235–4244. https://doi.org/10.1016/j.eswa.2013.12.011.

[17]. Levenberg, K. (1944). A Method for the Solution of Certain Non – Linear Problems in Least Squares. Quarterly of Applied Mathematics, 2, 164–168. https://doi.org/citeulike-article-id:10796881.

[18]. Majumder, M., Roy, P., & Mazumdar, A. (2010). A generalized overview of artificial neural network and genetic algorithm. In Impact of Climate Change on Natural Resource Management (pp. 393–415). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-90-481-3581-3_22.

[19]. Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics, 11(2), 431–441. https://doi.org/10.1137/0111030.

[20]. Mehdizadeh, S., Behmanesh, J., & Khalili, K. (2016). Comparison of artificial intelligence methods and empirical equations to estimate daily solar radiation. Journal of Atmospheric and Solar-Terrestrial Physics, 146, 215–227. https://doi.org/10.1016/j.jastp.2016.06.006.

[21]. Minns, A. W., & Hall, M. J. (2004). 20: Artificial Neural Network Concepts in Hydrology. Encyclopedia of Hydrological Sciences, 1–13. https://doi.org/10.1002/0470848944.

[22]. MINNS, A. W., & HALL, M. J. (1996). Artificial neural networks as rainfall-runoff models. Hydrological Sciences Journal, 41(3), 399–417. https://doi.org/10.1080/02626669609491511.

[23]. Mishra, P. (2013). Development of Rainfall-Runoff Model Using Soft Computing Techniques. Retrieved from http://apsurewa.ac.in/downloads/Corrected_Synopsis_Pradeep_Mishra latest.docx.

[24]. Pacelli, V., Bevilacqua, V., & Azzollini, M. (2011). An Artificial Neural Network Model to Forecast Exchange Rates. Journal of Intelligent Learning Systems and Applications, 03(02), 57–69. https://doi.org/10.4236/jilsa.2011.32008.

[25]. PRAMANIK, N., & PANDA, R. K. (2009). Application of neural network and adaptive neuro-fuzzy inference systems for river flow prediction. Hydrological Sciences Journal, 54(2), 247–260. https://doi.org/10.1623/hysj.54.2.247.

[26]. Remesan, R., Shamim, M. A., Han, D., & Mathew, J. (2009). Runoff prediction using an integrated hybrid modelling scheme. Journal of Hydrology, 372(1–4), 48–60. https://doi.org/10.1016/j.jhydrol.2009.03.034.

[27]. Sajikumar, N., & Thandaveswara, B. S. (1999). A non-linear rainfall – runoff model using an artificial neural network, 216, 32–55.

[28]. Shewchuk, J. R. (1994). An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Science, 49(CS-94-125), 64. https://doi.org/10.1.1.110.418.

[29]. Toth, E., & Brath, A. (2007). Multistep ahead streamflow forecasting: Role of calibration data in conceptual and neural network modeling. Water Resources Research, 43(11). https://doi.org/10.1029/2006WR005383.

[30]. Tyagi, S., & Panigrahi, S. K. (2017). An SVM—ANN Hybrid Classifier for Diagnosis of Gear Fault. Applied Artificial Intelligence, 31(3), 209–231. https://doi.org/10.1080/08839514.2017.1315502.

[31]. Wright, J., & Nocedal, J. (1999). Numerical optimization. Springer. https://doi.org/10.1007/BF01068601.

[32]. Wu, C. L., & Chau, K. W. (2011). Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis. Journal of Hydrology, 399(3–4), 394–409. https://doi.org/10.1016/j.jhydrol.2011.01.017.

[33]. Zhang, G. P. (2001). An investigation of neural networks for linear time-series forecasting. Computers and Operations Research, 28(12), 1183–1202. https://doi.org/10.1016/S0305-0548(00)00033-2.

[34]. Zhang, G. P., Patuwo, B. E., & Hu, M. Y. (2001). A simulation study of artificial neural networks for nonlinear time-series forecasting. Computers & Operations Research, 28(4), 381–396. https://doi.org/10.1016/S0305-0548(99)00123-9.