A survey on rainfall forecasting using artificial neural network

Qi Liu*

Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET),
Nanjing University of Information Science and Technology,
Nanjing, 210044, China
and
School of Computing,
Edinburgh Napier University,
Edinburgh, Scotland, UK
Email: q.liu@napier.ac.uk
*Corresponding author

Yanyun Zou

School of Computer and Software,
Nanjing University of Information Science and Technology,
Nanjing, China
Email: so_cloudy@163.com

Xiaodong Liu

School of Computing,
Edinburgh Napier University,
10 Colinton Road, Edinburgh EH10 5DT, UK
Email: x.liu@napier.ac.uk

Nigel Linge

The University of Salford,
Salford, Greater Manchester, M5 4WT, UK
Email: n.linge@salford.ac.uk

Abstract: Rainfall has a great impact on agriculture and people's daily travel, so accurate prediction of precipitation is well worth studying for researchers. Traditional methods like numerical weather prediction (NWP) models or statistical models can't provide satisfied effect of rainfall forecasting because of nonlinear and dynamic characteristics of precipitation. However, artificial neural network (ANN) has an ability to obtain complicated nonlinear relationship between variables, which is suitable to predict precipitation. This paper mainly introduces background knowledge of ANN and several algorithms using neural network applied to precipitation prediction in recent years. It is proved that neural network can greatly improve the accuracy and efficiency of prediction.

Keywords: rainfall; prediction; precipitation forecasting; artificial neural network; ANN; training algorithms; nonlinear relationship; embedded systems.

Reference to this paper should be made as follows: Liu, Q., Zou, Y., Liu, X. and Linge, N. (2019) 'A survey on rainfall forecasting using artificial neural network', *Int. J. Embedded Systems*, Vol. 11, No. 2, pp.240–249.

Biographical notes: Qi Liu received his BSc in Computer Science and Technology from the Zhuzhou Institute of Technology, China in 2003, and his MSc and PhD in Data Telecommunications and Networks from the University of Salford, UK in 2006 and 2010. His research interests include context awareness, data communication in MANET and WSN, and smart grid. His recent research work focuses on intelligent agriculture and meteorological observation systems based on WSN.

Yanyun Zou received her Bachelor's in Computer Science and Technology from the Nanjing University of Information, Science and Technology in 2016, and she is currently pursuing a Master's in Computer Science and Technology at the Nanjing University of Information Science and Technology. His research interests include the algorithms of artificial neural network.

Xiaodong Liu is a Reader and the Director of Centre for Information and Software Systems in School of Computing at Edinburgh Napier University. His research interests include context-aware adaptive services, service evolution, mobile clouds, pervasive computing, software reuse, and green software engineering. He is a member of IEEE Computer Society and British Computer Society.

Nigel Linge received his BSc in Electronics from the University of Salford, UK in 1983, and his PhD in Computer Networks from the University of Salford, UK, in 1987. He was promoted to Professor of Telecommunications at the University of Salford, UK in 1997. His research interests include location-based and context aware information systems, protocols, mobile systems and applications of networking technology in areas such as energy and building monitoring.

1 Introduction

According to the annual report in 2016, Jiangsu province's annual average precipitation is 1,528.5 mm, which reached the highest peak in the past 65 years. Too much or too little rainfall may cause severe meteorological disasters, such as floods or droughts, which greatly affect people's daily lives. As a result, it is very important to forecast precipitation accurately. Traditional methods to predict rainfall are divided into two main parts: numerical weather prediction (NWP) model and statistical model. But these two methods have great limitations in the case of the dynamic change and nonlinear shift of rainfall (Mohini et al., 2015). For this case, artificial neural network (ANN) has been proposed to apply to precipitation forecasting.

ANN has lots of strengths to predict rainfall. First, ANN is a kind of data-driven model, so there is no need to set restrictions when modelling. Second, because of the learning ability of ANN, it can accumulate a large amount of experiences so as to predict patterns which did not exist before. Third, neurons in ANN work in parallel processing mechanism, so they are able to process big data efficiently. Last but not the least; complicated nonlinear relationship between variables can be extracted using ANN.

As early as a few decades ago, researchers have used ANN techniques for precipitation prediction. After so many years of development, a lot of progress has been made in this field. Young and Liu (2014) used a physical model called HEC-HMS and an ANN to predict hourly rainfall-flow. Then result showed that the hybrid model was more excellent than a single ANN model. Abhishek et al. (2012) combined back-propagation algorithm (BPA) with multilayer neural network to predict average precipitation, and compared to CBP and LRN, BPA worked better. Chai et al. (2015) proposed an EMD-LSSVM (empirical mode decomposition least squares support vector machine) model to analyse the 300 index of CSI. It concluded that EMD-LSSVM model with grid search method is a promising option. Abbot and Marohasy (2015) came up with an independent ANN model, which achieved a more effective result for medium-term rainfall forecasts for Queensland. Ahmed et al. (2015) proposed a multi-layer perceptron neural network to downscale rainfall in an arid region. Also, for the prediction of time series data, it turned out that FLANN got lower AAPE than the other two models (Santosh et al., 2013). Gyanesh et al. (2013) found that BPN model had capacity to learn features of monsoon rainfall data time series. Rajkumar et al. (2015) combined FNN as well as HPSO technique to reduce the size and improve training speed of network. Mislan et al. (2015) employed BPNN architecture to predict precipitation, which showed a good performance. Namitha et al. (2015) put forward an idea about ANN implemented on Map-reduce framework to forecast short-term precipitation. The result showed that implement this solution on Hadoop improves its speed and scale. Dubey (2015) utilised three different training algorithms to create ANN. It suggested that feed-forward distributed time delay algorithm was te most accurate algorithm. The hybrid model, which combined MLP and CAPSO algorithm, provided a good ability with improved accuracy (Zahra et al., 2015). There are also many optimisation algorithms to improve the performance in this field recently. In order to improve the processing speed, Digalwar et al. (2017) proposed a real-time scheduling algorithm, which contains periodic tasks and mixed tasks of aperiodic tasks. This method not only completes periodic tasks on time, but also provides minimum response time for aperiodic tasks. Zhang et al. (2018) used kernel PCA network based on two-layer convolutional network in the data pre-processing stage, which is an optimisation algorithm to extract abstract features. In order to optimise operation efficiency and space utilisation ratio, an optimised kernel-based extreme learning machine algorithm was brought up (Liu et al., 2016). Xue et al. (2017) came up with a self-adaptive ABC algorithm based on the global best candidate (SABCGB) for the issue of global optimisation. Tian and Chen (2017) developed a learning method to achieve the relevance between the heterogeneous databases in human age estimation, so the general feature space can be provided. Gupta and Omkar (2017) calculate prediction error by using the prediction made by GARCH mode. Chen et al. (2016) in order to solve the stereo matching problem, a model is used to output a reconstruction as the final result of the input of the captured images, and the reconstruction was accurate and dense.

This paper is arranged as follows. The concept of ANN and popular training algorithms are elaborated in Section 2. Section 3 focuses on several excellent applications using ANN for precipitation forecasting. Moreover, analysis and comparison of the applications in Section 3 is explained in Section 4.

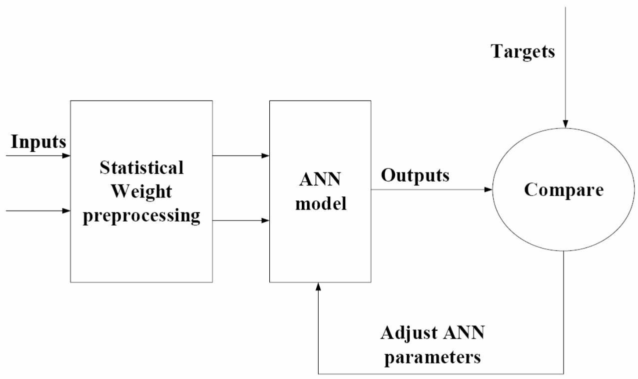
2 Basic knowledge

In this section, the conception of ANN is illustrated. And, training algorithms in ANN are explained in this part as well.

2.1 The concept of ANN

The structure of neural network is inspired by human brain. Just as neurons in human brain, there are many neurons in neural network to process information received. Neural network can be regarded as a parallel distributed processor, which contains many processing units. It can learn rules from experiential knowledge so as to play a valuable role (Deepak et al., 2013). Of course, the human brain is much more complex than the neural network. The ANN generally does not have a ring structure. The electrical signals of the human brain neurons are not only strong or weak, but also have time and urgency. It is like the Morse code, and there is no such complex signal mode in ANN. The fundamental unit of ANN is an artificial neuron, which can accept inputs, then process them and export relative outputs finally. Basic working process of ANN is shown in Figure 1. As shown in this figure, after inputs are pre-processed, they are sent to an ANN. After that, the outputs of this model are compared with targets, and a value which represents the error of this comparison goes back to adjust ANN's parameters. After a large amount of iterations, the error reaches an optimal value, and the structure of the ANN is adjusted to a suitable one.

Figure 1 Basic working process of ANN



Just like the biological neural networks, ANN has similar structure and functions, which are represented by a mathematical model. Every ANN is composed of an artificial neuron, which is a simple mathematical function with three basic rules: multiplication, summation and activation. For example, in the first step, each input value is multiplied with individual weight because the inputs are weighted. In the next step, the sum function works: it sums all weighted inputs and bias. In the last step, the result of the sum is going through a transfer function, which can be seen as an activation function.

It seems that the working principle of artificial neurons is very simple, nothing special. But when the neuron is integrated into an ANN, its potential is activated. So an ANN is able to play a powerful role through the self-learning ability of neurons.

In order to fully reflect the powerful function of ANNs, people usually do not randomly connect neurons. In the past, researchers have proposed several predefined topologies that can help people solve problems more efficiently and simply. But different problems should be solved in different ways. After determining the type of problems, we are supposed to take a suitable topology, and then take some measurements to adjust it. Usually the object to adjust is the structure and its parameters. As the biological neural network can learn from the inputs of the environment to learn the relative behaviours and responses, an ANN can follow this point. It can learn all the time through a large amount of inputs, so as to the best condition after the training.

Researchers put forward a variety of learning rules and algorithms to meet the needs of different network models. The effective learning algorithm makes the neural network can construct the objective representation of the objective world through the adjustment of the connection weight, and form the information processing method with characteristic information. The information storage and processing are reflected in the connection of the network. According to the different learning environment, neural network learning can be divided into supervised learning and unsupervised learning. In the supervision of learning, the training sample data is added to the network input, while the corresponding expected outputs and network outputs compared to get an error signal, in order to control the weights of the connection strength adjustment, after repeated training to a convergence determined weights. When the sample situation changes, the study can modify the weights to adapt to new environment. There are many kinds of neural networks using supervised learning, such as a back-propagation network, a perceptron and so on. As for unsupervised learning, it does not give a standard sample in advance, the network directly into the environment, so the learning phase is as the same time as the working phase. At this time, the change of learning law follows the evolution equation of connection weight. The simplest example of unsupervised learning is the Hebb learning rule.

2.2 Back propagation algorithm

BPA is a traditional training algorithm of ANN.

2.2.1 The concept of BPA

BP algorithm is one of the most commonly used training algorithms in neural networks (Rumelhart et al., 1986). BPA consists of two channels: a forward road and a backward road. In the forward road, vectors are transmitted into the neurons in input layer, then output real responses through layers at last. In the backward road, the main purpose is adjusting weights of synapses to make actual responses close to expected responses. Specifically, there is an error because of the gap between actual responses and desired responses. Then the error signal is delivered back along the opposite direction with synapses.

2.2.2 Optimisation methods of BPA

Five popular optimisation methods which applied to BPA are briefly introduced below.

1 Gradient descent

Gradient descent method is the easiest training algorithm. It only needs the information of gradient vectors, so it belongs to first order algorithm. It defines that

$$f(w_i) = f_i, \nabla f(w_i) = g \tag{1}$$

w_0 is an original point, w_i is moving to next point w_{i+1} along with the direction of $d_i = g_i$, repeatedly iterates like this until termination condition is satisfied. The recurrence formula of gradient descent is shown as follow

$$w_{i+1} = w_i - d_i \bullet \varphi_i, i = 0, 1, ... \tag{2}$$

φ_i denotes learning rate. This parameter can be set as immutable, but also can be used to update the calculation along the direction of the training.

2 Levenberg-Marquardt algorithm

Levenberg-Marquardt algorithm is also called least square method of attenuation. It does not need to calculate Hessian matrix, but gradient vectors and Jacob matrix are needed in this algorithm. Assuming that the loss function f is the sum of squared errors

$$f = \sum e_i^2, i = 1, ..., m \tag{3}$$

m represents number of training samples. Jacob matrix of the loss function is consists of partial derivatives of error term, which is

$$J_{i,j} f(w) = \frac{de_i}{dw_j} (i = 1, ..., m \text{ and } j = 1, ..., n) \tag{4}$$

m and n denotes the amount of samples in training set and the amount of parameters of neural network individually. The scale of Jacob matrix is $m \cdot n$. The gradient vector of the loss function is

$$\nabla f = 2J^T \bullet e \tag{5}$$

e is a vector consists of error terms. In the end, this formula can be used to estimate the Hessian matrix

$$HF \approx 2J^{T^{J+\lambda L}} \tag{6}$$

λ is an attenuation factor, to ensure that the Hessian matrix is positive, I is the unit matrix. The parameters updating formula of the algorithm is as follows

$$w_{i+1} = w_i - \left(J_i^T \bullet J_i + \lambda_j I\right)^{-1} \bullet \left(2J_i^T \bullet e_i\right), i = 0, 1, ... \tag{7}$$

If λ is equal to zero, this algorithm is comparable with Newton method. If λ is set to be very large, this algorithm is just like gradient descent method with a small learning rate.

3 Newton's method

The goal of this algorithm is to use the second order partial derivative of the loss function to find a better learning direction.

It can define $f(w_i) = f_i, \nabla f(w_i)$ and $Hf(w_i) = H_i$, use the Taylor expansion to estimate the function f at the value w_0

$$f = f_0 + g_0 \cdot (w - w_0) + 0.5 \cdot (w - w_0)^2 \cdot H_0 \tag{8}$$

H_0 is the Heisen matrix value of the function f at w_0. At the minimum of $f(w)$, we get the second formula at $g = 0$

$$g = g_0 + H_0 \cdot (w - w_0) = 0 \tag{9}$$

Therefore, the parameters are initialised to w_0, and the iterative formula of the Newton algorithm is

$$w_{i+1} = w_i - H_{i-1} \cdot g_i, i = 0, 1, ... \tag{10}$$

$H_{i-1} \cdot g_i$ is called Newton. It is worth noting that if the Hysen matrix is a non-definite matrix, then the parameter may move in the direction of the maximum value, rather than the minimum direction. So the loss of the function value does not guarantee that each iteration is decreased. To avoid this issue, we usually modify the equation of the Newton algorithm:

$$w_{i+1} = w_i - (H_{i-1} \cdot g_i) \cdot \eta, i = 0, 1, ... \tag{11}$$

Learning rate η can be set as an immutable value or dynamically adjusted. The vector $d = H_{i-1} \cdot g_i$ has the name of Newton training direction.

The efficiency of this method to train the neural network model is proved to be better than the gradient descent method. Since the conjugate gradient method is not necessary to calculate the Hysen matrix, we also recommend that when the neural network model is large.

4 Conjugate gradient

The algorithm is expected to accelerate the convergence rate of gradient descent, while avoiding the use of the Cypriot matrix for evaluation, storage and inversion to obtain the necessary optimisation information.

In this algorithm, because the seek is performed with the conjugate direction, it is usually better to converge more quickly than the gradient descent direction. The training direction of the conjugate gradient method is conjugated with the Cypriot matrix.

d can be used to express the training direction vector, and then start from the original parameter vector w_0 and the original training direction vector $d_0 = -g_0$. The algorithm builds the training direction sequence as follows:

$$d_{i+1} = g_{i+1} + d_i \cdot Y_i,\ i = 0, 1, ... \quad (12)$$

In the above equation, γ is called a conjugate parameter, and there are some methods to compute this parameter. The two most common methods are derived from Fletcher, Reeves and Polak, Ribiere.

The parameters are updated and optimised by the following expression. The usual learning rate η can be obtained using the univariate function optimisation method.

$$w_{i+1} = w_i + d_i \cdot \eta_i,\ i = 0, 1, ... \quad (13)$$

The conjugate gradient method has been shown to be much more effective in the neural network than the gradient descent method. And because the conjugate gradient method does not require the use of Cypriot matrix, so in large-scale neural network, it can still be a very good performance.

5 Quasi-Newton method

Since the Newton method needs to calculate the Hessian matrix and the inverse matrix, more computational resources are needed, so a variant algorithm, called Cauchy-Newton method, can be used to compensate for the large computational complexity. This method does not directly compute the Hysen matrix and its inverse matrix, but only the first order partial derivative of the loss function is used to estimate the inverse matrix of the Hessian matrix at each iteration.

The Hessian matrix is consists of the second derivative of the loss function. The core idea of this method is to estimate the inverse matrix of the Hessian matrix with matrix G, which requires only the first derivative of the loss function. The updating equation of the Cauchy-Newton method can be written as:

$$w_{i+1} = w_i - (G_i \cdot g_i) \cdot \eta_i,\ i = 0, 1, ... \quad (14)$$

Learning rate η can be set to be immutable or dynamically adjusted. Hessian matrix inverse matrix estimation G has many different types.

In many cases, this is the default choice algorithm: it improves speed of the other methods, without the need to accurately calculate the Hysen matrix and its inverse matrix.

In recent years, some effective algorithms like GA, PSO and wavelet analysis are used with ANN to predict rainfall. Next section is going to explain these techniques in detail.

3 Applications of neural networks for rainfall prediction

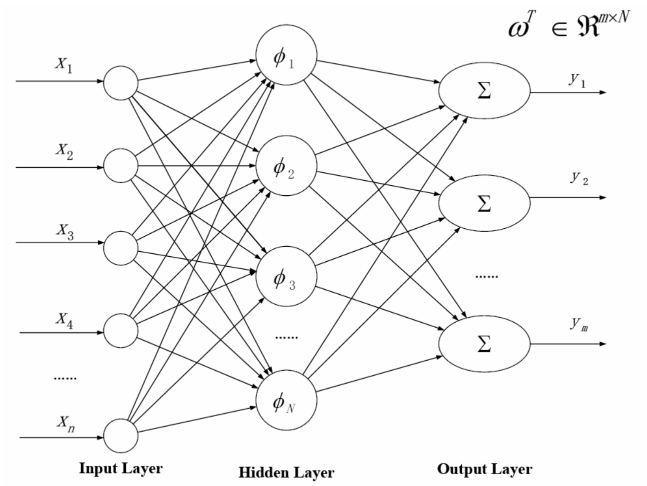
3.1 RBF-NN with GAPSO algorithm

Wu et al. (2015) brought up a hybrid optimisation method, namely HPSOGA, which combined particle swarm optimisation (PSO) with genetic algorithm (GA), to construct a radial basis function neural network (RBF-NN) automatically. The object was to predict Liuzhou's monthly precipitation from 2005 to 2011.

3.1.1 Radial basis function neural network

Radial basis function (RBF) networks were introduced by Aydin and Kisi (2014) RBF-NN has feed-forward structure, which is nonlinear and hierarchic. It regards neural network as a fitting curve in high dimensional space. The architecture contains three layers: an input layer, a hidden layer and an output layer. It is shown in Figure 2.

Figure 2 Architecture of RBF-NN



The output can be obtained using the following formula

$$y_t = \sum_{i=1}^{N} \omega_{ti}\varphi_i(x, c_i) = \sum_{i=1}^{N} \omega_{ti}\varphi_i\left(\|x, c_i\|_2\right),\ t = 1, 2, ..., m \quad (15)$$

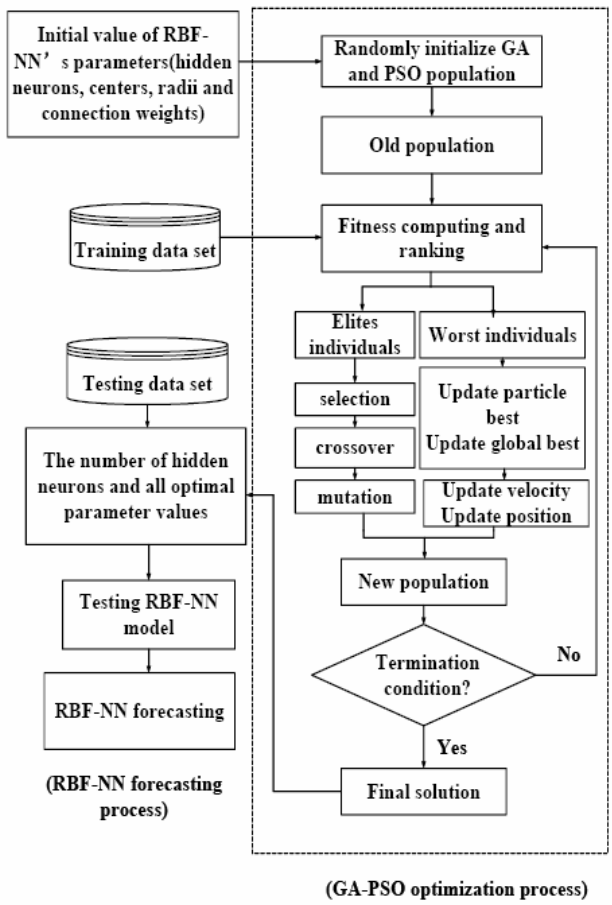
where $x \in R^{n\times 1}$denotes an input vector, $\varphi_k(\cdot)$ means a radial function from $R^{n\times 1}$, ω_{ti} are equal to weights of the links from the number of hidden neuron i to the amount of output neuron t in output layer. N represents the amount of neurons in the hidden layer.

3.1.2 *Hybrid of PSO and GA for RBF-NN design*

To decide the parameters of RBF, i.e., the values for centre, radii and weights, PSO and GA are used in RBF-NN.

GA is a computational model, which comes from Darwin's theory of biological evolution and biological evolution process of natural selection. PSO is motivated by social behaviours among animals. In this algorithm, a group (called swarm) is created which contains random search solutions. So each solution is regarded as a particle. Moreover, every particle has random velocity flying through multi-dimension to find global minimum. This algorithm has ability to remember information of good solutions in all particles, which is different with GA. The hybrid algorithm is displayed in Figure 3.

Figure 3 RBF-HPSOGA hybrid algorithm



Details of Figure 3 are displayed as follows:

1. Population initialisation: parameters of GA, PSO and RBF-NN are set in the suitable constraint range. And a population whose size is 4N is produced by random. The length L of each chromosome equals to GþH, where G is the size of binary code of the amount of hidden nodes and H is the size of actual valued code. The centre, radius, and weights are initially randomly set between –10 and 10, which are evenly distributed in this range.
2. Ranking: rank the 4N population by the rule of fitness values.
3. The top 2N individuals are separated from others according to the fitness values in each generation. And the top 2N individuals are also called elites, then they are sent to GA operators. After selecting the tops, we perform basic crossover and mutation operations to adjust the control code. That is to say, according to the mutation operation, the hidden nodes are deleted or added, and the corresponding control code is encoded by 0 or 1. The crossover and mutation operators of actual value are as follows. Crossover operation with probability p_c:

$$\begin{cases} X_i^{t+1} = c_i \cdot X_i^t + (1 - c_i) \cdot X_{i+1}^t \\ X_i^{t+1} = (1 - c_i) \cdot X_i^t + c_i \cdot X_{i+1}^t \end{cases} \tag{16}$$

where X_i^t, X_{i+1}^t □ are a pair of individuals before crossing, X_i^{t+1}, X_{i+1}^{t+1} □□□ are individuals after crossing, and c_i is the random number in between [0, 1]. Mutation operation with probability p_m:

$$X_i^{t+1} = X_i^t + c_i \tag{17}$$

where c_i is the random number in the interval $[u_{\min} - \delta_1 - x_i^t, u_{\max} + \delta_1 + x_i^t]$. This will make sure that the mutated individual is still in the range of searching.
4. PSO method: after 4N individuals are divided into two parts, 2N best individuals are applied to PSO operators, which are regarded as the particle velocities, while the rest of all individuals are used as the particle positions applied to PSO operators.
5. A new population is produced.
6. Termination condition: if the new population with updated fitness value is not able to meet ending requirement, go back to step 2, or it comes to the final result.
7. Observation: validation accuracy curve is observed all the time in order to prevent over training. Once it shows the best accuracy, it will end the training processure.
8. Recall: according to the termination condition of training procedure in the former step, the amount of hidden neurons and the parameters of RBF neural network are recalled.
9. Retrain and build: RBF neural network is going to trained on a larger amount of testing dataset, on the basis of the number of hidden neurons and parameters of RBF-NN which are recalled in the former step.

3.1.3 *Experimental result*

Data sources are from 24 stations of Liuzhou meteorology administration rain gauge networks, which are monthly rainfall data from 1949 to 2011.

Training set is total of 480 samples, and validation set is total of 180 samples, and testing set is total of 84 samples.

Three models were compared in that experiment, which were single RBF-NN, RBF-NN with pure GA and RBF-NN with HPSOGA. After tested 80 samples, the result was shown in Table 1.

Table 1 Performance of three models

Model	*AARE*	*RMSE*	*CC*
RBF-NN	1.01	170.46	0.71
RBF-GA	0.88	111.95	0.85
RBF-HPSOGA	0.61	67.73	0.93

In Table 1, evaluation indexes, AARE, RMSE and CC are used to assess performance of monthly rainfall. Obviously, RBF-HPSOGA is outstanding among these three models. So the result proves that HPSOGA method is helpful to build an efficient architecture of radial basis neural network.

3.2 Wavelet neural network

An attempt had been brought up to produce an effective way for rainfall forecasting using a hybrid technique with ANN (Ramana et al., 2013). In this hybrid model, input signal was processed by using wavelet analysis to forecast monthly precipitation.

3.2.1 Wavelet analysis

Wavelet (wavelet) this term, as its name implies, 'wavelet' is a small waveform. The so-called 'small' means that it has a decay; and called 'wave' is its volatility. It is widely applied in signal processing, which has attracted much attention in many fields. Compared with the Fourier transform, the wavelet transform is more suitable in extracting information from signal. By scaling and translation operation, the function or signal can be refined and analysed in many scales, which can solve many difficult problems that cannot be solved by Fourier transform. Wavelet has been validated to be a powerful weapon for the analysis and synthesis of data from long-term memory processes by decomposing time series data (Ramana et al., 2013).

1 Discrete wavelet transform

   Discrete wavelet transform (DWT) is useful in numerical analysis and time-frequency analysis. The first DWT is invented by Hungarian mathematicians. DWT is the discrete input and discrete output, but there is not a simple and clear formula to represent the relevance between input and output, only in a hierarchical architecture. The significance of wavelet decomposition lies in the ability to decompose signals at different scales, and the choice of different scales can be determined according to different objectives.

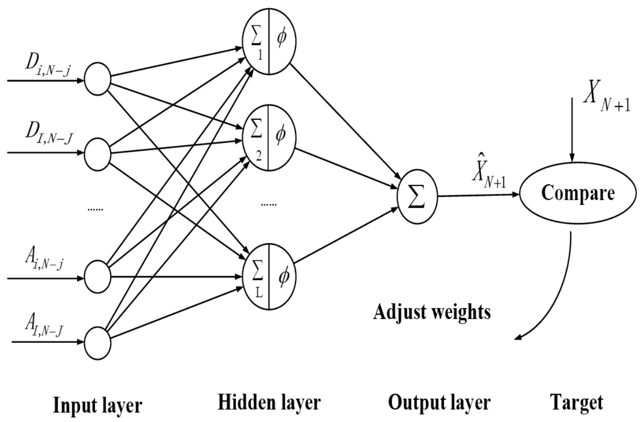
2 Continuous wavelet transform

   Continuous wavelet transform (CWT) is used instead of the window Fourier transform (WFT) to overcome the problem that the resolution cannot change invariably with time and frequency. When the window function is selected, the window shape of the time-frequency window is immutable for the WFT, and it cannot be changed as the signal component analysed is high-frequency information or low-frequency information, and the non-stationary signal is rich of the frequency components, so their ability to analyse non-stationary signals is very limited. Wavelet transform is similar to WFT, that is, the signal is multiplied by wavelet, and wavelet transform is calculated for different time periods of time domain signal. But there are two differences between WFT and wavelet transform: windowing signal does not do Fourier transform; the most essential feature of wavelet transform is to compute the frequency of each component can change the shape of the window.

3.2.2 Method of combining wavelet analysis with ANN

Multilayer perceptron (MLP) neural network architecture is adopted. Figure 4 shows the structure of wavelet-based MLP.

Figure 4 Wavelet-based MLP neural network structure



The decomposed details (D) and approximation (A) are as input signal into network composition as displayed in Figure 4. As shown in Figure 4, i represents a decomposition degree varying from 1 to I, j denotes the quantity of previous values whose range is from 0 to J. And N is the size of time series data.

It is worth reminding that Levenberg-Marquardt algorithm which has been mentioned in section 2 is taken as training algorithm in this wavelet neural network.

3.2.3 Experimental result

Darjeeling is located in the east coast of 88 degrees 15 minutes 47 seconds, latitude 27 degrees 2 minutes and 30 seconds, is a small town in West Bengal, India, the capital of Darjeeling, located in the Himalayas foothills of the West Vallic Mountains, an average elevation of 2,134 meters. Darjeeling is also known as 'King Kong Island'.

To prove the superiority of WNN intuitively, auto-regression (AR) model and ANN model were used to make comparison with WNN model.

Table 2 Performance of calibration and validation of WNN and ANN for rainfall prediction

Model	*Calibration*		
	RMSE	*R*	*COE (%)*
WNN	35.12	0.992	98.48
ANN	123.23	0.902	81.49
AR	226.00	0.659	37.70
Model	*Validation*		
	RMSE	*R*	*COE (%)*
WNN	63.01	0.974	94.78
ANN	163.79	0.807	64.73
AR	221.82	0.642	34.91

Rainfall and temperature data of 74 years from Darjeeling rain gauge station was used to validate the hybrid algorithm. As a result, WNN outperformed with a lower RMSE and a higher COE as shown in Table 2. The comprehensive analysis shows that the performance of the wavelet neural network model is more excellent than that of the ANN, which may be the result of physical factors. Terrain and meteorological factors play a great role in wavelet-based nonlinear dynamic models. It is worth noting that if the hydrological data model is decomposed more accurately, it may be better to simulate the rainfall process.

3.3 FLANN

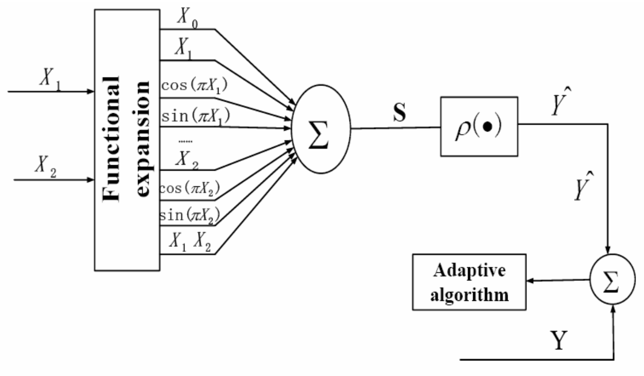
Santosh et al. (2013) compared different ANN models, namely, functional-link artificial neural network (FLANN), MLP and Legendre polynomial equation (LPE), for predicting time series of precipitation in India. Among these models, FLANN showed an optimum ability of rainfall forecasting. So this part is going to discuss FLANN in detail.

3.3.1 Functional-link artificial neural network

Patra et al. (2008) came up with FLANN and it is an ANN structure with only one layer emerging in recent years. It is able to form discretional and complicated decision regions through producing nonlinear decision boundaries. This kind of ANN model owns a notable feature that it removes hidden layers, which greatly reduces the complexity of calculation compared to MLP. In the structure of FLANN, there is a functional expansion block, which uses a functional model. The model includes a subset of orthogonal sin and cos basis functions. The FLANN structure is displayed in Figure 5. For example, suppose a two dimensional input pattern, $X = (x_1, x_2)^T$, the improved pattern is adjusted to $X = [x_1, \cos(\pi x_1), \sin(\pi x_1), \ldots, x_2, \cos(\pi x_2), \sin(\pi x_2), \ldots, x_1x_2]^T$.

Wherein, BPA was as a training algorithm in this literature, which turned to be very simple because there are no hidden layers.

Figure 5 FLANN structure



3.3.2 Rainfall forecasting using FLANN

Here is the procedure of rainfall estimation using FLANN.

Step 1 Begin

Step 2 Three parameters are input, Ø, θ, Y_{t-1}, and initial values are 0.5486, 0.9585, 0.03216 individually.

Step 3 Initialise weights ω_i, $i = 1, 2, \ldots$ where i represents the amount of functional elements.

Step 4 A functional block is produced

$$X_i = [1, x_1, \sin(\pi_{x1}), \cos(\pi_{x1}), \sin(\pi_{x2}), \cos(\pi_{x2}), \ldots] \tag{18}$$

Step 5 The output is calculated as

$$O_i = \sum_{i=1}^{N} \omega_i * X_i \tag{19}$$

Step 6 The output error is calculated as

$$e_i = d_i - O_i \tag{20}$$

d_i denotes the desired output, while is the predicted output of this system.

Step 7 Weights are updated as follow

$$\omega_i(k+1) = \omega_i(k) + \alpha e_i(k) X_i(k) \tag{21}$$

k means the time index error $\leq \varepsilon(0.01)$ and α is the momentum parameter.

Step 8 If error $\leq \varepsilon(0.01)$ then go to next step. Otherwise, back to Step 3.

Step 9 The procedure of training is completed. Then testing can be implemented.

Step 10 End.

3.3.3 Experimental result

To validate the method which proposed by the paper, the researcher constructed a simulation environment using MATLAB. Data sources were obtained from India meteorological department (IMD).

The researcher compared rainfall forecasting performances by proposed three models with actual data. AAPE was adopted as an evaluation index. From the experimental outcome, it was found that FLANN owned a lower AAPE than MLP and LPE which is displayed in Table 3.

Table 3 Absolute error analysis of MLP, FLANN and LPE over real rainfall data

	MLP	*FLANN*	*LPE*
Absolute value of average % of error	14.92 db	4.94 db	11.04 db

Through the comparison of different ANN models, it was found that FLANN gives a better forecasting result with less AAPE for forecasting time series data. So FLANN was proved to be an excellent neural network for precipitation prediction.

4 Conclusions

Because of the ability of distributed storage and nonlinear data processing, ANN is an optical option to predict rainfall. By combining effective algorithms with ANN, accuracy and efficiency of precipitation forecasting is able to increase greatly.

In this paper, many methods using ANN of previous studies, as well as background knowledge and techniques of ANN, are introduced. And as major contributions in this paper, three methods are elaborated in detail. First, GAPSO algorithm is used to construct the architecture of RBF-NN by determining parameters of RBF (centre, radii) and weights. By comparing with pure RBF-NN and RBF-GA, RBF-NN with GAPSO is verified to be more accurate for precipitation prediction. But the disadvantage is the computing speed because of the complexity. Second, WNN is also an excellent way to forecast. This method mainly focuses on input signal processing. By contrasting WNN with ANN and AR models, WNN provides a better experimental result than the other two models. However, the redundancy of the wavelet transform is very large. Finally, a single layer ANN, namely, FLANN, is illustrated in this paper. With a faster computing speed, it shows a more outstanding capability than MLP and LPE. But it may not able to solve complicated problems because of the simple network structure. We can see that these three methods improve the ANN from different aspects. And they can achieve satisfactory results when applying to precipitation forecasting.

This paper introduces three methods to forecast rainfall in detail, though each of them performs excellent, there are still some parts that need to be improving as mentioned above. Considering meteorological data are typical time series data, in future work, a better algorithm that focuses on the forecasting time series data problem needs to be brought up.

Acknowledgements

This work is supported by Marie Curie Fellowship (701697-CAR-MSCA-IF-EF-ST), the Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), the NSFC (61300238 and 61672295), the 2014 Project of six personnel in Jiangsu Province under Grant No. 2014-WLW-013, and the PAPD fund.

References

Abbot, J. and Marohasy, J. (2015) 'Using artificial intelligence to forecast monthly rainfall under present and future climates for the Bowen Basin, Queensland, Australia', *International Journal of Sustainable Development and Planning*, Vol. 10, No. 1, pp.66–75.

Abhishek K., Kumar, A., Rajeev, R. and Kumar, S. (2012) 'A rainfall prediction model using artificial neural network', *2012 IEEE Control and System Graduate Research Colloquium (ICSGRC 2012)*, pp.82–87.

Ahmed, K., Shahid, S., Haroon, S.B. and Wang, X.J. (2015) 'Multilayer perceptron neural network for downscaling rainfall in arid region: a case study of Baluchistan, Pakistan', *Journal of Earth System Science*, Vol. 124, No. 6, pp.1–17.

Aydin, K. and Kisi, O. (2014) 'Damage detection in Timoshenko beam structures by multilayer perceptron and radial basis function networks', *Neural Computing and Applications*, Vol. 24, No. 3, pp.583–597.

Chai, J., Du, J., Lai, K.K. and Yan, P.L. (2015) 'A hybrid least square support vector machine model with parameters optimization for stock forecasting', *Mathematical Problems in Engineering*, pp.1–7.

Chen, Y., Hao, C., Wu, W. and Wu, E. (2016) 'Robust dense reconstruction by range merging based on confidence estimation', *Science China Information Sciences*, Vol. 59, No. 9, pp.1–11.

Deepak, N.R., Amitav, M. and Pranati, M. (2013) 'A survey on rainfall prediction using artificial neural network', *International Journal of Computer Applications*, Vol. 2, No. 16, pp.32–40.

Digalwar, M., Gahukar, P., Raveendran, B.K. and Mohan, S. (2017) 'Energy efficient real-time scheduling algorithm for mixed task set on multi-core processors', *Int. J. of Embedded Systems*, Vol. 9, No. 6, pp.523–534

Dubey, D.A. (2015) 'Artificial neural network models for rainfall prediction in pondicherry', *International Journal of Computer Applications*, Vol. 120, No. 3, pp.30–35.

Gupta, B.B. and Omkar, B.P. (2017) 'GARCH and ANN-based DDoS detection and filtering in cloud computing environment', *Int. J. of Embedded Systems*, Vol. 9, No. 5, pp.391–400.

Gyanesh, S., Sanjeev, K., Manoj, K.K. and Pulak, G. (2013) 'BPN model for long-range forecast of monsoon rainfall over a very small geographical region and its verification for 2012', *Short Communication*, Vol. 30, No. 2, pp.155–170.

Liu, Q., Cai, W., Shen, J., Fu, Z., Liu, X. and Linge, N. (2016) 'A speculative approach to spatial-temporal efficiency with multi-objective optimization in a heterogeneous cloud environment', *Security and Communication Networks*, Vol. 9, No. 17, pp.4002–4012.

Mislan, H., Sumaryono, H.S. and Aipassa, M. (2015) 'Rainfall monthly prediction based on artificial neural network: a case study in Tenggarong station, East Kalimantan-Indonesia', *International Conference on Computer Science and Computational Intelligence (ICCSCI)*, Vol. 59, pp.142–151.

Mohini, P., Vipul, K. and Harshadkumar, B. (2015) 'Rainfall forecasting using neural network: a survey', *2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)*, IMS Engineering College, Ghaziabad, India, pp.706–707.

Namitha, K., Jayapriya, A. and Kumar, G.S. (2015) 'Rainfall prediction using artificial neural network on MapReduce framework', *International Symposium on Women in Computing and Informatics (ISWCI)*, pp.492–495.

Patra, J.C., Chin, W.C., Meher, P.K. and Chakraborty, G. (2008) 'Legendre-FLANN-based nonlinear channel equalization in wireless communication system', *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp.1826–1831.

Rajkumar, R., Albert James, A. and Chandrakala, D. (2015) 'Weather forecasting using fuzzy neural network (FNN) and hierarchy particle swarm optimization algorithm (HPSO)', *Indian Journal of Science and Technology*, Vol. 8, No. 12, pp.1–8.

Ramana Venkata, R., Krishna, B., Kumar, R.S. and Pandey, G.N. (2013) 'Monthly rainfall prediction using wavelet neural network analysis', *Water Resour Manage*, Vol. 27, No. 10, pp.3697–3711.

Rumelhart, D., Hinton, G. and Williams, R. (1986) 'Learning representations by back propagating errors', *Nature*, Vol. 323, No. 9, pp.533–536.

Santosh, N.K., Debi, T.P., Simanta, N.K. and Subhasis, M. (2013) 'Prediction of rainfall in India using artificial neural network (ANN) models', *I.J. Intelligent Systems and Applications*, Vol. 5, No. 12, pp.1–22.

Tian, Q. and Chen, S. (2017) 'Cross-heterogeneous-database age estimation through correlation representation learning', *Neurocomputing*, Vol. 238, No. C, pp.286–295.

Wu, J., Long, J. and Liu, M. (2015) 'Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm', *Neurocomputing*, Vol. 148, No. 2, pp.136–142.

Xue, Y., Jiang, J., Zhao, B. and Ma, T. (2017) 'A self-adaptive articial bee colony algorithm based on global best for global optimization', *Soft Computing*, Vol. 22, No. 9, pp.1–18.

Young, C.C. and Liu, W.C. (2014) 'Prediction and modelling of rainfall-runoff during typhoon events using a physically-based and artificial neural network hybrid model', *Hydrological Sciences Journal*, Vol. 60, No. 12, pp.1–36.

Zahra, B., Morteza, F., Siti, S.M., Masoumeh, Z. and Zulkifli, Y. (2015) 'A new rainfall forecasting model using the CAPSO algorithm and an artificial neural network', *Neural Comput. & Applic.*, Vol. 27, No. 8, pp.1–15.

Zhang, J., Huang, Q., Wu, H. and Liu, Y. (2018) 'Effective traffic signs recognition via kernel PCA network', *Int. J. of Embedded Systems*, Vol. 10, No. 2, pp.120–125.