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 & Technology, Nanjing, 210044
Email: qi.liu@nuist.edu.cn
*Corresponding author

Yanyun Zou

School of Computer & Software, Nanjing University of Information Science & Technology, China
E-mail: so_cloudy@163.com

Xiaodong Liu

School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh EH10 5DT, UK

Nigel Linge

The University of Salford, Salford, Greater Manchester, M5 4WT, 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) is able to obtain complex 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; neural networks; ANN; training algorithms.

Reference to this paper should be made as follows: Qi Liu, Yanyun Zou, Xiaodong Liu and Nigel Linge (2017) 'A survey on rainfall forecasting using artificial neural network ', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. X, No. Y4, pp.000–000.

Biographical notes: Qi Liu (M'11) received his BSc degree in Computer Science and Technology from 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 degree in Computer Science and Technology from Nanjing University of Information, Science and Technology in 2016, and she is currently pursuing a master's degree 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 & 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 degree 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 a report (Report, 2017), in 2016, Jiangsu province's annual average precipitation is 1528.5mm, 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 P. et al., 2015). To solve this problem, 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 have been made in this field. Young et al. (2014) used a physically based model HEC-HMS and an artificial neural network to predict hourly rainfall-runoff. Then result showed that the hybrid model performed better than a single ANN model. Abhishek et al. (2012) combined Back-Propagation Algorithm (BPA) with multilayer neural network to predict average rainfall over Udupi district of Karnataka, and compared to CBP and LRN, BPA worked better. Chai et al. (2015) proposes an EMD-LSSVM (empirical mode decomposition least squares support vector machine) model to analyse the CSI 300 index. It concluded that EMD-LSSVM model with GS (grid search) algorithm is a promising option. Abbot et al. (2015) came up with an independent artificial neural network model, which achieved a more skilful medium-term rainfall forecasts for the Bowen Basin in Queensland. Ahmed et al. (2015) proposed a multilayer perceptron neural network to downscale rainfall in an arid region. Multilayer Perceptron (MLP), Functional-link Artificial Neural Network (FLANN) and Legendre Polynomial Equation (LPE) were used to predict the time series data. It turned out that FLANN got lower absolute average percentage error (AAPE) than the other two models (Nanda et al., 2013). Shrivastava et al. (2013) found that BPN model had capacity to learn features of monsoon rainfall data time series. Rajkumar et al. (2015) combined Fuzzy Neural Network (FNN) and Hierarchical Particle Swarm Optimization (HPSO) technique to reduce the size and improve training speed of network. Mislan et al. (2015) tested rainfall data using two-hidden layers of BPNN architectures with three different epochs. Furthermore, BPNN algorithm provided a good model to predict rainfall in Tenggarong, East Kalimantan - Indonesia. Namitha et al. (2015) put forward an idea about Artificial Neural Network implemented on Map-reduce framework for short term rainfall prediction. The result showed that implement this solution on Hadoop makes it faster and scalable. Dubey (2015) utilized three different training algorithms, feed-forward back propagation algorithm, layer recurrent algorithm and feed-forward distributed time delay algorithm to create ANN. It suggested that the best accuracy was obtained by feed-forward distributed time delay algorithm. The hybrid learning of MLP with CAPSO algorithm provided higher rainfall forecasting accuracy, lower error and higher classification accuracy (Beheshti et al., 2015). There are also many optimization algorithms to improve the performance in this field recently. A prediction model based on an optimized Kernel-based Extreme Learning Machine algorithm is proposed for faster forecast of job execution duration and space occupation (Liu et al., 2016). Xue et al. (2017) proposed a self-adaptive ABC algorithm based on the global best candidate (SABCGB) for global optimization. Tian et al. (2017) developed a so-called correlation component manifold space learning (CCMSL) to first learn a common feature space by capturing the correlations between the heterogeneous databases in human age estimation. Chen et al. (2016) solved the stereo matching problem effectively by a robust model that outputs an accurate and dense reconstruction as the final result from an input of multiple images captured by a normal camera.

This paper is organized as follows. The concept of Artificial Neural Network 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 2 is explained in Section 4.

2 Preliminary

In this section, conception of Artificial Neural Network is described, what's more, training algorithms in ANN are indispensable, which are introduced in this part as well.

2.1 The Concept of Artificial Neural Network

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. A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use (Nayak et al., 2013). The basic 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 Fig 1. As shown in this figure, after inputs are pre-processed, they are sent to an artificial neural network. 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 artificial neural network is adjusted to a suitable one.

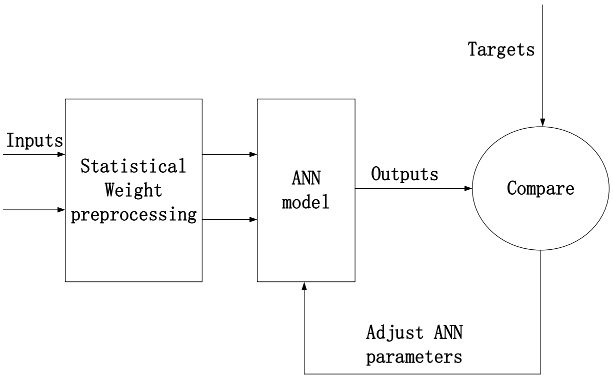

Fig. 1. Basic working process of artificial neural network

Just like the biological neural networks, an artificial neural network has similar structure and functions, which are represented by a mathematical model. Every artificial neural network 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 passing through an activation function, which is also called a transfer function.

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

In order to fully reflect the powerful function of artificial neural networks, 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 artificial neural network 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)

Back propagation algorithm (BPA) is a traditional training algorithm of ANN.

2.2.1 The Concept of Back Propagation Algorithm (BPA)

Back Propagation Algorithm is one of the most popular training algorithms in neural network (Rumelhart et al., 1986). BPA consists of two channels: a forward pass and a backward pass. In the forward pass, vectors are transmitted into the neurons in input layer, then output real responses through layers at last. In the backward pass, the main purpose is adjusting weights of synapses to make actual responses close to expected responses. Specifically, actual responses are subtracted from the desired response to generate an error signal. Then the error signal is propagated along the opposite direction with synapses.

2.2.2 Optimization Methods of BPA

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

1. Gradient Descent

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

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

w_0 is an original point, $w_i$is moving to next pointw_{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

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

φ_i denotes learning rate. This parameter can be set to a fixed value, 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,

$$f = \sum e_i^2, i = 0,\cdots,m \tag{3}$$

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

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

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

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

e is a vector consists of error terms. Finally, we can use this formula to estimate the Hessian matrix

$$HF \approx 2J^{TJ+\lambda I} \tag{6}$$

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

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

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

Newton's method belongs to the second order algorithm because this method uses Hessian matrix. 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 $f(w_i) = f_i$, $\nabla f(w_i) = g_i$ 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 equation at $g = 0$

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

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

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

$H_{i-1} \cdot g_i$ 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 reduced. In order to avoid this problem, we usually modify the equation of the Newton algorithm:

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

Learning rate η can be set to a fixed value or dynamically adjusted. The vector$d = H_{i-1} \cdot g_i$ is called the 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 does not need to calculate the Hysen matrix, we also recommend that when the neural network model is large.

4. Conjugate gradient

The conjugate gradient method can be considered as the intermediate of the gradient descent method and the Newton method. 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 optimization information.

In the conjugate gradient training algorithm, because the search is performed along 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 initial parameter vector w_0 and the initial training direction vector $d_0 = -g_0$. The conjugate gradient method constructs the training direction sequence as follows:

$$d_{i+1} = g_{i+1} + d_i \cdot Y_i, i = 0,1,\cdots\cdots \tag{12}$$

In the above equation, γ is called a conjugate parameter, and there are some methods to calculate this parameter. The two most common methods are derived from Fletcher, Reeves and Polak, Ribiere. For all conjugate gradient algorithms, the training direction is periodically reset to a negative gradient.

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

$$w_{i+1} = w_i + d_i \cdot \eta_i, i = 0,1,\cdots\cdots \tag{13}$$

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

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

Learning rate η can be set to a fixed value or dynamically adjusted. Hessian matrix inverse matrix estimation G has many different types. Two commonly used types are Davidon-Fletcher-Powell formula (DFP) and Broyden-Fletcher-Goldfarb-Shanno formula (BFGS).

In many cases, this is the default choice algorithm: it is faster than the gradient descent method and the conjugate gradient method, 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 artificial neural network 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 optimization method, namely HPSOGA, which combined particle swarm optimization (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 (RBF-NN)

Radial basis function (RBF) networks were introduced by K Aydin et al. (2014) RBF-NN is a kind of feed forward neural network, 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 with nonlinear processing neurons and an output layer. It is shown in Fig 2.

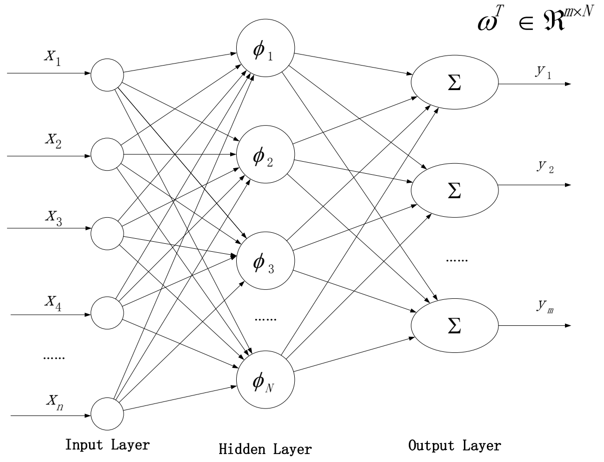

Fig. 1. Architecture of RBF-NN

The output can be obtained using the following formula

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

Where $\mathrm{x} \in R^{n\times1}$ denotes an input vector, $\varphi_k(\cdot)$ means a radial function from $R^{n\times1}$, $\omega_{ti}$are equal to weights of the links from the number of hidden neuron i to the number of output neuron t in output layer. N represents the number of neurons in the hidden layer.

3.1.2 Hybrid of PSO and GA for RBF-NN Design

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

GA is a computational model, which is derived from Darwin's theory of biological evolution and biological evolution process of natural selection. What's more, it is a kind of method to search the optimal solution through simulating the process of natural evolution. PSO is inspired by social behaviours among animals, such as bird flocking, fish schooling and so on. 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 have ability to remember information of good solutions in all particles, which is different with genetic algorithm. The hybrid algorithm is displayed in Fig 3.

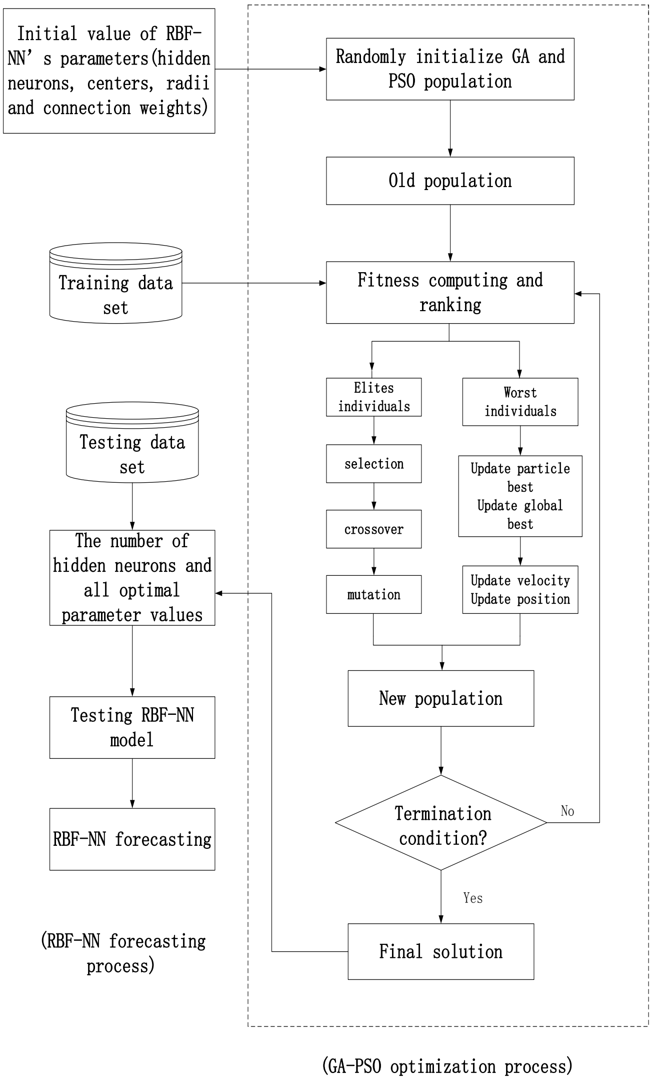
Initial value of RBF-NN' s parameters(hidden neurons, centers, radii and connection weights)
Randomly initialize GA and PSO population
Old population
Training data set
Fitness computing and ranking
Elites individuals
Worst individuals
Testing data set
selection
Update particle best
Update global best
crossover
The number of hidden neurons and all optimal parameter values
mutation
Update velocity
Update position
New population
Testing RBF-NN model
RBF-NN forecasting
Termination condition?
No
Yes
(RBF-NN forecasting process)
Final solution
(GA-PSO optimization process)

Fig. 2. RBF-HPSOGA hybrid algorithm

Details of Fig 3 are described as follows:

1. Population initialization: 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 length of binary code of the number of hidden nodes and H is the length of real-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 elites are selected, basic crossover and mutation operations are used to the binary code, namely, if a hidden node is deleted (added) according to mutation operation, the corresponding control code is encoded 0(1). The crossover and mutation operators of real value are as follows. Crossover operation with probability p_c :

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

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 :

$$X_i^{t+1} = X_i^t + c_i \qquad (17)$$

Where c_i is the random number in the interval $\lfloor u_{\min} - \delta_1 - x_i^t, u_{\max} + \delta_1 + x_i^t \rfloor$. This will ensure that the mutated individual is still within the searching range.

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 termination condition, go back to step 2, otherwise 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 validation accuracy, the training process is supposed to the end.
8. Recall: According to the termination condition of training procedure in the previous step, the number 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*

The equipment used to develop the RBF-HPSOGA is a computer, which owns following characters: Intel Pentium E3.20 GHz CPU, 1.96GB RAM, a Windows XP operating system and the Matlab 9.0 development environment. Parameters in this method are as follows: population size= 40, the number of iteration= 100, crossover probability= 0.8 and mutation probability= 0.2.

Data sources are monthly rainfall data of 24 stations of Liuzhou Meteorology Administration rain gauge networks from 1949 to 2011.

Training set is total of 480 samples in the period from January 1949 to December 1988, and validation set is total of 180 samples in the period from January 1989 to December 2004, and 84 samples were used as a testing set from January 2005 to December 2011.

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	1.0059	170.4612	0.6978
RBF-GA	0.8954	111.9055	0.8477
RBF-HPSOGA	0.6050	67.7286	0.9331

In Table 1, evaluation indexes, average absolute relative error (AARE), root mean square error (RMSE) and correlation coefficient (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 made to produce an effective way for rainfall forecasting by combining wavelet technique with ANN (Ramana et al., 2013). In this hybrid model, input signal was processed by using wavelet analysis. To predict precipitation at Darjeeling rain gauge station, monthly rainfall and temperature data of 74 years was used.

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 a local transform of space (time) and frequency, so it can effectively extract information from signal. By scaling and translation operation, the function or signal can be refined and analyzed in many scales, which can solve many difficult problems that can’t be solved by Fourier transform. By decomposing a time series into time-frequency-space, one is able to determine both the dominant modes of variability and how those modes vary in time. Wavelet have proven to be a powerful tool for the analysis and synthesis of data from long memory processes (Ramana et al., 2013).

1. Discrete Wavelet Transform (DWT)

Discrete Wavelet Transform (DWT) is useful in numerical analysis and time-frequency analysis. The first discrete wavelet transform is invented by Hungarian mathematicians. Discrete wavelet transform is the discrete input and discrete output, but there is not a simple and clear formula to represent the relationship between input and output, only in a hierarchical architecture.

2. Continuous Wavelet Transform (CWT)

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 fixed for the WFT, and it cannot be changed as the signal component analyzed is high-frequency information or low-frequency information, and the non-stationary signal is rich of the frequency components, so their ability to analyze 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 important feature of wavelet transform is to calculate 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. Fig 4 shows the structure of wavelet based MLP.

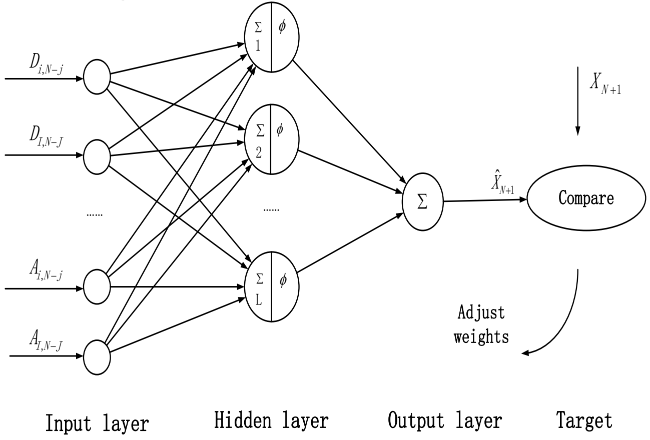

Fig. 3. Wavelet based MLP neural network structure

The decomposed details (D) and approximation (A) are as input signal into neural network structure as displayed in Fig 4. As shown in Fig 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 length of time series.

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". The hills of Darjeeling are part of the Mahabharat Range or Lesser Himalaya. Kanchenjunga, the world’s third-highest peak, 8,598 m (28,209 ft) high, is the most prominent mountain visible from Darjeeling.

To prove the superiority of WNN intuitively, auto-regression (AR）model and ANN model were used to make comparison with WNN model. 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 root mean square error and a higher correlation coefficient as shown in Table 2.

Table 2. Performance of Calibration and Validation of WNN and ANN for Rainfall Prediction

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

3.3 FLANN

Nanda et al. (2013) compared different ANN models, namely, Multi-Layer Perceptron (MLP), Functional-link Artificial Neural Network (FLANN) 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 (FLANN)

Patra et al. (2008) originally proposed Functional link artificial neural network (FLANN) and it is a single-layer of ANN structure emerging in recent years. It is able to form discretional and complex decision regions through generating 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 contains a subset of orthogonal sin and cos basis functions, original pattern with its outer products. The FLANN structure is displayed in Fig. 5. For example, suppose a two dimensional input pattern, $\mathrm{X} = (x_1, x_2)^T$, the improved pattern is adjusted to $\mathrm{X} = [x_1, \cos(\pi x_1), \sin(\pi x_1), \cdots, x_2, \cos(\pi x_2), \sin(\pi x_2), \cdots, x_1 x_2]^T$.

Wherein, BPA was as a training algorithm in this literature, which turned to be very simple because of the absence of hidden layers.

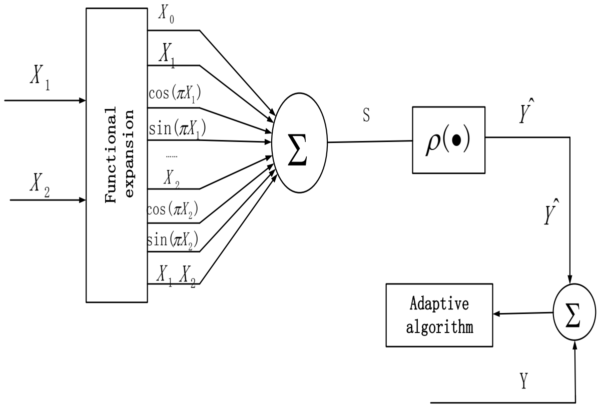

Fig. 4. FLANN structure

3.3.2 Rainfall Forecasting using FLANN

Here is the procedure of rainfall estimation using FLANN.

Step 1: Start

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

Step 3: Initialize weights ω_i , i=1,2,…,l where i represents the number of functional elements.

Step 4: A functional block is produced

$$X_i = [1, x_1, \sin(\pi x_1), \cos(\pi x_1), x_2, \sin(\pi x_2), \cos(\pi x_2), \cdots\cdots] \quad (18)$$

Step 5: The output is calculated as

$$O_i = \sum_{i=1}^{N} \omega_i * X_i \quad (19)$$

Step 6: The output error is calculated as

$$e_i = d_i - O_i \quad (20)$$

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

Step 7: Weights are updated as follow

$$\omega_i(k+1) = \omega_i(k) + \alpha e_i(k) X_i(k) \quad (21)$$

k means the time index $\mathrm{error} \le \varepsilon(0.01)$ and α is the momentum parameter.

Step 8: If $\mathrm{error} \le \varepsilon(0.01)$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 real time data. Absolute Average Percentage Error (AAPE) was adopted as an evaluation index. From the experimental result, it was observed that FLANN owned a lower AAPE than MLP and LPE which is displayed in Table 3. So FLANN was proved to be an excellent neural network for precipitation prediction.

Table 3. Absolute Error Analysis of MLP, FLANN & LPE over Actual Rainfall Data

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

4 Conclusion

Because of the ability of distributed storage and nonlinear data processing, artificial neural network 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, background knowledge and techniques of artificial neural network for rainfall forecasting are introduced, and three methods are elaborated in detail. First, GAPSO algorithm is used to construct the architecture of RBF-NN by determining parameters of radial basis function (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. 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. 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. We can see that these three methods improve the artificial neural network from different aspects. And they can achieve satisfactory results when applying to precipitation forecasting.

Acknowledgements

This work is supported by Marie Curie Fellowship (701697-CAR-MSCA-IF-EF-ST), 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 & Planning, Vol. 10 No. 1, pp.66-75

Abhishek Kumar, Kumar Abhay, Ranjan Rajeev and Kumar Sarthak . (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 SB and Wang XJ. (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

Beheshti Zahra, Firouzi Morteza, Shamsuddin Mariyam Siti, Zibarzani Masoumeh and Yusop Zulkifli. (2015) 'A new rainfall forecasting model using the CAPSO algorithm and an artificial neural network', Neural Comput & Applic, pp.1-15

Chai J, Du J, Lai KK and Yan PL. (2015) 'A Hybrid Least Square Support Vector Machine Model with Parameters Optimization for Stock Forecasting', Mathematical Problems in Engineering, pp.1-7

Chen Yadang, Hao Chuanyan, Wu Wen and Wu Enhua. (2016) 'Robust dense reconstruction by range merging based on confidence estimation', SCIENCE CHINA Information Sciences, vol. 59 No. 9, pp.1-11

Dubey D Akash. (2015) 'Artificial Neural Network Models for Rainfall Prediction in Pondicherry', International Journal of Computer Applications, Vol. 120 No. 3, pp.30-35

Liu Qi, Cai Weidong, Shen Jian, Fu Zhangjie, Liu Xiaodong and Linge Nigel. (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, Haviluddin, Hardwinarto Sigit, Sumaryono and Aipassa Marlon. (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., 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 GS. (2015) 'Rainfall Prediction using Artificial Neural Network on Map-Reduce Framework', International Symposium on Women in Computing & Informatics (ISWCI), pp.492-495

Nanda Kumar Santosh, Tripathy Prasad Debi, Nayak Kumar Simanta and Mohapatra Subhasis. (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

Nayak Ranjan Deepak, Mahapatra Amitav and Mishra Pranati. (2013) 'A Survey on Rainfall Prediction using Artificial Neural Network', International Journal of Computer Applications, Vol. 2 No. 16, pp.32-40

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 Neu-ral 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

Report, http://www.njdaily.cn/ (2017)

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

Shrivastava Gyanesh, Karmakar Sanjeev, Kowar Kumar Manoj and Guhathakurta Pulak. (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

Tian Qing and Chen Songcan. (2017) 'Cross-Heterogeneous-Database Age Estimation Through Correlation Representation Learning', Neurocomputing, Vol. 238, pp.286–295

Wu Jiansheng, Long Jin and Liu Mingzhe. (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 Yu, Jiang Jiongming, Zhao Binping and Ma Tinghuai. (2017) 'A self-adaptive articial bee colony algorithm based on global best for global optimization', Soft Computing, pp.1-18

Young CC and Liu WC. (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