Application of Artificial Neural Networks for Rainfall Forecasting in Fuzhou City

GONG Zhenbin* **Supervisor: Prof. A. W. Jayawardena****
MEE09203

ABSTRACT

Accurate rainfall prediction is of great interest for water management and flood control. In reality, physical processes influencing the occurrence of rainfall are highly complex, uncertain and nonlinear. Rainfall prediction is one of the most difficult tasks in operational meteorology. In order to forecast rainfall in Fuzhou City, Artificial Neural Networks (ANNs), a data driven technique based on the working principle of biological neurons are applied in this study. Objectives are set to differentiate what kinds of atmospheric situation can trigger rainfall, and find out how much rainfall can be generated in such situation based on the available meteorological parameters records and real rainfall data. Two different kinds of ANNs, i.e. Back Propagation Neural Networks (BPNNs) and Support Vector Machines (SVMs) are applied and compared with each other. The results show that both BPNNs and SVMs can classify rainy days and non-rainy days satisfactory, but the network ability of calculating the quantitative precipitation is very poor because of their poor generalization ability. The results of sensitivity analysis for BPNNs show that several free parameters, such as numbers of hidden neurons, learning rates and momentum terms, have great influences on the network performance. By suitable data pre-processing, such as Principal Component Analysis (PCA), data normalization and data re-sampling, the network performance can be improved a lot. Increasing the input data information, changing the real rainfall data into suitable rain levels, applying more complicated network topology and trying different activation functions are also recommended to improve the quantitative precipitation prediction for future works.

Key words: *ANNs, BPNNs, SVMs, quantitative precipitation prediction, generalization ability, Principal Component Analysis, data pre-processing, sensitivity analysis*

INTRODUCTION

Accurate rainfall prediction is of great interest for water management and flood control. In reality, physical processes influencing the occurrence of rainfall are highly complex, uncertain and nonlinear. Rainfall prediction is one of the most difficult tasks in operational meteorology. There are several rainfall forecasting methods, such as statistical methods and Numerical Weather Prediction (NWP) models. Because of the nonlinear character of rainfall process, statistical methods cannot generate good results. NWP models are based on very complicated physical dynamics. NWP have improved the forecast accuracy of large-scale weather systems greatly, such as wind field and mass field, but not good at meso-scale or small-scale weather systems. Rainfall is mainly caused by meso-scale and small-scale weather systems. Local climate situations also can affect the rainfall generation process a lot. NWP models cannot solve this local problem.

Artificial Neural Networks (ANNs) are a type of data driven technique based on the working principle of biological neurons. Since the mechanisms of rainfall are still not understood well, ANNs are a good choice worth trying to analyze the relationship between meteorological parameters and rainfall. The objectives are to differentiate what kinds of atmospheric situation can trigger rainfall, and find out how much rainfall can be generated in such situations. After finding the optimal network parameters,

* Engineer, Fuzhou Meteorological Bureau, Fujian, China
** Research and Training Advisor, International Centre for Water Hazard and Risk Management (ICHARM), Japan.

by inputting the prediction results of various meteorological parameters which come from NWP models, the network can decide if it has rainfall and quantify the amount of precipitation.

Fuzhou (福州) is the capital city of Fujian province. The city is located in south-eastern China, the west side of Taiwan Strait. The jurisdiction area is around 12,000 sq kilometers, and the population is 6.39 million. The rainy season is from April to June. During the rainy season, the continuous heavy rainfall can cause basin wide flood in Min River. Because of the flood control facilities along the river, nowadays, this kind of flood is not the major threat to Fuzhou. The period from July to September is the typhoon season. During this period, except the severe typhoon rainstorm, the strong convective rainstorms, such as thunderstorm, occur very frequently. Most of this kind of rainfall can cause severe waterlogging in the city area, and also can trigger landslides or debris flows in western and northern hilly region.

DATA

Daily rainfall data of Fuzhou Meteorological Observation Station from 1948 to 2008 are selected to be the desired output data for training, validating and testing. The NCEP/NCAR (National Center for Atmospheric Research) Re-analysis data downloaded from NOAA website are chosen as input data. The input data contain various meteorological parameters, including air temperature, specific humidity, relative humidity, geopotential height, wind and precipitable water. Each parameter contains three different layers' variables ranging from 1000mb to 850mb, except precipitable water, which is measured for the whole air column. Besides these measured data, the input data also include the geopotential height difference between Fuzhou and three other surrounding grids data to express the information of pressure gradient. Total number of input variables is 31.

THEORY AND METHODOLOGY

According to Haykin (1999), a neural network contains large amounts of parallel distributed processors made up of simple processing units. These units have a natural propensity for storing experiential knowledge and making it available for use. Neural networks acquire knowledge from its environment through a learning process, and use its interneuron connection strengths, i.e. synaptic weights, to store the acquired knowledge. By giving a set of measured data, neural networks can learn the stimulus-response relationship within the data set through the training process (Minns and Hall, 1996). There are several kinds of ANNs. BPNN and SVMs are selected in this study.

Multilayer Perceptrons with back propagation algorithm, i.e. BPNN, is the most popular network architecture in use today. BPNN includes two distinct passes of computation, the forward pass and the backward pass. In the forward pass the synaptic weights remain unaltered throughout the network, and the function signals of the network are computed on a neuron-by-neuron basis and propagate through the activation function. Any non-linear function that is differentiable can be used as an activation function. From input layer to hidden layer, the logistic sigmoid function is often used as the activation function because of its very simple derivative that makes the subsequent implementation of the learning algorithm much easier (Flood and Kartam, 1994). The logistic sigmoid function is shown as below:

$$Logsig(x) = \frac{1}{1+e^{-ax}}$$, which varies between the limits [0,1].

The hyperbolic function is also often used in many circumstances, which is shown as:

$$Hyerbolic(x) = a\tanh(bx)$$, which varies between the limits [-1,1].

According to Lecun, suitable values for the constants a and b are: a=1.7159 and b=2/3.When signal is propagated from hidden layer to output layer, sometimes a linear function can be used as the activation function, which is more simple than the logistic function or the hyperbolic function. The backward pass is in opposition to the forward pass, which starts at the output layer by passing the error signals back through the network layer by layer and recursively computing the local gradient for each neuron, and then adjusting the synaptic weights of the connections with a hope that in the next iteration the error will be reduced (Haykin, 1999). The error-correction rule used in this study is the Delta rule, which is also named as Least Mean Squared (LSM) algorithm. According to the delta rule, the correction $\Delta w_{ji}(n)$ applied to the synaptic weight connecting neuron i to neuron j is defined as: $\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n)$, where α is the momentum term with the range between 0 and 1, η denotes learning rate parameter, usually we can set $\alpha = 0.5, \eta = 0.1$, $\delta_j(n)$ denotes the local gradient of nth training example, $\Delta w_{ji}(n)$ is the synaptic weights correction of current training example, $\Delta w_{ji}(n-1)$ is the synaptic weights correction of previous training example. The local gradient $\delta_j(n)$ can be calculated by using the derivative of the activation function. Then the synaptic weights and bias can be updated step by step.

Before applying the BPNN, the input data and desired output data need to be pre-processed. In this study, there are 31 input variables, so the dimensionality should be reduced and the input variables also should be decorrelated. The Principal Component Analysis (PCA) is a good choice to solve this problem. After PCA, the input data should be normalized according to different output range of different activation functions. For example, for logistic function, the input data and the desired output data should be scaled into range of 0 to 1. The final data pre-processing step is data balancing. As the number of training examples increases, in many applications, the neural networks would face with imbalanced data sets, which can cause the network to be biased towards one class containing more examples (Kotsiantis et al, 2006). Common solutions of imbalanced data set include: re-sampling the examples, i.e. duplicating training examples of the under-represented class (Ling, C. and Li, C., 1998) and downsizing the examples, i.e. removing training examples of the over represented class (Kubat, M. and Matwin, S., 1997). In this study, re-sampling the examples is selected in order to present as many data as possible to the network.

Strictly speaking, Support Vector Machines (SVM) is one kind of neural networks, and is another category of universal feed forward networks pioneered by Vapnik, which can also be used for pattern classification and nonlinear regression (Haykin, 1999). Because of its remarkable characteristics such as good generalization performance, the absence of local minima, and sparse representation of solution, SVM has been receiving increasing attention in areas ranging from pattern recognition to regression estimation (Cao and Tay, 2003). The SVM is based on Cover's theorem that a non-linearly unseparable multidimensional space may be transformed into a new feature space where the patterns are linearly separable with high probability (Haykin, 1999). What SVM need to do is through an inner-product kernel function to construct an optimal hyperplane which can separate the different features, in such a way that the margin of separation between positive and negative examples is maximized. Briefly speaking, the SVM is an approximate implementation of the method of structural risk minimization which seeks to minimize an upper bound of the generalization error consisting of the sum of the training error and a confidence interval.

In SVM, the solution to avoid getting stuck into local minima is dependent on a subset of training data points which are referred to as support vectors (Cao and Tay, 2003). In order to solve pattern recognition problems, the first step is to choose a suitable kernel function for nonlinear mapping of an input vector into a high-dimensional feature space that is hidden from both the input and output. There are three common types of support vector machines: polynomial learning machine, radial-basis function network, and two-layer perceptron. For radial-basis function network, for example, the kernel function is shown as: $K(x, x_i) = \exp(-\frac{1}{2\sigma^2}\|x - x_i\|^2)$. By using this kernel function, a decision surface is constructed. This decision surface is nonlinear in the input space, but its image in the feature space is linear. Given the training sample $\{(x_i, d_i)\}_{i=1}^{N}$ (x_i is the input vector, d_i is the desired output), then the classification problem can be transferred to find the Lagrange multipliers $\{(\alpha_i)\}_{i=1}^{N}$ that maximize the objective function:

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j K(x_i,x_j) ,$$

This function subject to the constraints: (1) $\sum_{i=1}^{N}\alpha_i d_i=0$ and (2) $0\le\alpha_i\le C$, for i=1,2,……,N, where C is a penalty parameter, which is a user-specified positive parameter. By solving function of $Q(\alpha)$ with its constraints, we can determine the Largrange multipliers, and a hard classifier implementing the optimal separating hyperplane in the feature space, which is given by:

$$f(x)=\mathrm{sgn}[\sum_{i=1}^{N}\alpha_i d_i K(x_i,x_j)+b] ,$$

According to this function, the classification purpose can be realized. The regression problem can be solved by similar ways, but need to define a Є-insensitive loss function first.

RESULTS AND DISCUSSION

Findings from classification problems: After performing PCA, the input variables are reduced from 31 to 9. Feeding the input data to the BPNN with sequential mode to solve classification problem is the first task in this study. In classification problem, different training examples are classified into two types, rainy days and non-rainy days, so the desired output data have only two values. Taking the logistic function in BP neural networks as example, 0.1 denotes the day with no rainfall (precipitation=0), and 0.9 denotes the day with rainfall (precipitation>0). In order to solve this problem, BP neural networks with different activation functions are attempted.

Table 1: Results of Classification Problem by applying BPNN

Logistic Function in Hidden Layer						
Number of hidden neurons	Logistic Function in output layer			Linear Function in output layer		
	Accuracy			Accuracy		
	Training	Validation	Testing	Training	Validation	Testing
5	0.77	0.81	0.81	0.76	0.77	0.78
6	0.77	0.80	0.81	0.76	0.76	0.77
7	0.78	0.80	0.81	0.76	0.76	0.78
Hyperbolic Function in Hidden Layer						
Number of hidden neurons	Hyperbolic Function in output layer			Linear Function in output layer		
	Accuracy			Accuracy		
	Training	Validation	Testing	Training	Validation	Testing
5	0.75	0.80	0.80	0.75	0.79	0.80
6	0.75	0.80	0.79	0.74	0.80	0.80
7	0.75	0.80	0.79	0.75	0.79	0.80

Table 1 shows the results of computation. The input data set is divided into three parts, 80% for training, 10% for validation, and the rest 10% for testing. The network is run with three different numbers of hidden neurons. There are four kinds of activation function combinations. The results show that the network performance with different activation function combination and different number of hidden neurons has no big difference. For training part, the accuracy range from 74% to 78%, and for testing part, the accuracy range from 77% to 81%. This accuracy is within the acceptable range.

Table **2**: Results of classification by using SVM

For Classification				
Kernel Function	Penalty parameter	Kernel function	Accuracy	
	C	σ	Training	Testing
Linear Function	8	-	74.75%	77.47%
Polynomial Function	4	1	78.24%	77.73%
Radial Basis Function	4	1	79.16%	78.94%
Sigmoid Function	8	0.01	73.77%	79.96%

Table 2 shows the results calculated by using SVM with different kernel functions. The penalty parameter and kernel function parameter are selected by trying from 2^{-5} to 2^{5} which the interval is based on powers of 2. The accuracy is the only performance measurement for this classification problem. From the table, the accuracy is also within the acceptable range, and the radial basis function has the best performance. The SVM method can be used to classify this complicated nonlinear inseparable problem well, but the results are not as good as BPNN.

Findings from regression problems: After deciding whether there will be rainfall or not, the next task is to figure out the quantity of precipitation that will be. After data pre-processing, the data set is also divided into three parts for training, validation and testing. The network is run by using different combinations of activation functions in hidden layer and output layer with numbers of hidden neurons ranging from 5 to 14. Table 3 only shows the best performance of the four combinations.

Table 3. Statistical performance of BPNN using different activation functions

Activation Functions	Hidden neurons	Training Part				Testing Part			
		R	CD	EI	RMSE	R	CD	EI	RMSE
Logistic & Logistic	8	0.91	0.85	0.84	0.08	0.33	1.28	-0.52	0.1
Logistic & Linear	9	0.88	0.86	0.77	0.09	0.30	1.58	-0.83	0.11
Hyperbolic & Linear	6	0.73	0.75	0.51	0.17	0.29	1.65	-0.91	0.14
Hyperbolic & Hyperbolic	6	0.81	0.88	0.64	0.15	0.32	2.41	-1.42	0.16

From Table 3, all of the statistical performances of training parts for different activation function combination are far better than testing parts. The range of correlation coefficient of training is 0.73 to 0.94, and the best performance comes from combination of two logistic functions, but the combination of hyperbolic function and linear function shows relatively poor performance. Results obtained by comparing these performance indices show that logistic function has the best performance, and the linear function has the poorest performance. For testing part, all of statistical performances are not good, that means the generalization ability of the network this study selected is very poor, and cannot interpret the relationship between the input variables and the precipitation. This may be because the complication of the problem itself or the input variables this study selected might not be sufficient to model the mechanism of rainfall generation.

Table 4: Results of regression problems by using SVMs

For Regression							
Kernel Function	Penalty parameter	Kernel function parameter	Loss function parameter	Training		Testing	
	C	σ	ϵ	MSE	R	MSE	R
Linear	0.1	-	0.0001	1.78	0.13	1.64	0.15
Polynomial	0.1	0.125	0.01	1.78	0.28	1.62	0.38
Radial Basis	0.125	0.125	0.01	1.72	0.25	1.58	0.35
Sigmoid	0.125	0.125	0.0001	2.29	0.11	2.36	0.16

Table 4 shows the results of SVM regression problem. Similar to BP neural network, SVM cannot simulate the rainfall mechanism correctly. All correlation coefficients of observed rainfall and calculated output are not good enough. In this case, polynomial kernel function has the best performance, but the performance of sigmoid function is very poor.

Findings from sensitivity analysis: There are several free parameters which can influence the network performance, such as the numbers of hidden neurons, the learning rate, and the momentum term. By applying different numbers of hidden neurons, the results show that more hidden neurons always give more accurate output. In this study, when the number of hidden neurons increases from 3 to 15, the correlation coefficient (R) also improves from 0.77 to 0.9, and the root mean square error (RMSE) decreases from 0.09 to 0.062, but when the number is over 15, the performance becomes worse. This indicates that too many hidden neurons could cause over learning problem. The testing results show that more hidden neurons would decrease the generalization ability of the network.

In order to test the influence of different learning rate and momentum term, a BP neural network with 8 hidden neurons and logistic sigmoid function in hidden layer and linear function in output layer was tested. Different learning rates ranging from 0.03 to 0.3, in steps of 0.03 were tried. The results show that smaller learning rates always generates better accuracy. The correlation coefficients between training output and desired output decreases from 0.88 to 0.72 as the learning rate increases from 0.03

to 0.3, and so does the minimum RMSE. With lower learning rates, more training epochs to converge the network are needed, indicating that smaller learning rate parameters cause smaller changes to the synaptic weights in the network from one to the next iteration. After combined with suitable momentum term, the slowness of convergence and network performance can be improved. This can be proved by applying different momentum terms combined with the learning rate equal to 0.1. The results show that when the momentum term is equal to 0.5 or 0.7, the network performance is the best, and the convergence time is also shorter than others.

CONCLUSIONS

By the application of multilayer perceptron neural networks with back-propagation algorithm and support vector machines, the occurrence of a rainfall event can be predicted satisfactorily for different atmospheric conditions. Both methods are good at classification problem. For quantitative precipitation analysis, neither method can generate good results in real application. The generalization ability of artificial neural networks is poor. Before apply artificial neural networks into real practice, the network should be optimized by trying different free parameters, such as different numbers of hidden neurons, different learning rates and momentum terms. These free parameters can influence the network performance significantly. Also, suitable data pre-processing is very important and necessary before applying ANNs, such as data normalization, data balancing, and principal component analysis.

RECOMMENDATION

Future study should be conducted to improve the generalization ability of the BPNNs. The possible methods are: 1) modify the input data to include more information from surrounding area and also the data from more upper layers, 2) classify the quantitative precipitation data into different levels, such as light rain, moderate rain, and heavy rain, and replace the exact quantitative precipitation with suitable rain levels, 3) adjust the topology of networks, increase hidden layers or introduce different activation functions, and 4) find an efficient algorithm, such as Genetic Algorithm, to decide suitable network free parameters.

ACKNOWLEDGEMENT

I would like to express my solemn gratitude to my supervisor Prof. A. W. Jayawardena, Research and Training Advisor, ICHARM, PWRI, Tsukuba, Ibaraki, Japan for his overall supervision, invaluable suggestions and ardent encouragement in every aspect of this work.

REFERENCES

Cao L.J and Tay Francis E.H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE transactions on neural networks, Vol. 14, No.6* , 1506-1518.

Flood I and Kartam N. (1994). Neural networks in Civil Engineering: Principles and understanding. *Journal of Computing in Civil Engineering (ASCE), 8(2)* , 131-148.

Haykin, S. (1999). *Neural networks: A comprehensive foundation.* New York: Macmillan Publishing.

Kotsiantis S.B, Kanellopoulos D and Pintelas P.E (2006). Data pre-processing for supervised learning. *International Journal of Computer Science, Vol 1 No.2*, 111-117.

Kubat, M. and Matwin, S. (1997). Addressing the curse of imbalanced training sets: one sided selection. *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*, 179-186.

Ling, C. and Li, C. (1998). Data mining for direct marketing: problems and solutions. *American Association for Artificial Intelligence*, KDD-98, 73-79.

Minns, A. W and Hall, M. J. (1996). Artificial neural networks as rainfall-runoff models. *Hydrol. Sci. J.,41, 3* , 399-417.