ISCIT 2004

International Symposium on Communications and Information Technologies 2004 (ISCIT 2004)
Sapporo, Japan, October 26- 29, 2004

An Adaptive RBF Network optimised using a Genetic Algorithm applied to Rainfall Forecasting

[1]C. Jareanpon ,[2,3]W. Pensuwon,[3]R. J. Frank and [3]N.Davey

[1]Department of Informatics ,
Mahasarakham University, Thailand.
E-mail: *lotte_ce_kku@yahoo.com*

[2]Department. of Computer Engineering,
Khon Kaen University , Thailand,
E-mail : wanida@kku.ac.th

[3]Department of Computer Science,
University of Hertfordshire, United Kingdom.
E-mail: R.J.Frank@herts.ac.uk and N.Davey@hert.ac.uk

Abstract

Rainfall prediction is a challenging task especially in a modern world facing the major environmental problem of global warming. The proposed method uses an Adaptive Radial Basis Function neural network mode with a specially designed genetic algorithm (GA) to obtain the optimal model parameters. A significant feature of the Adaptive Radial Basis Function network is that it is able create new hidden units and solve the spread factor problem using a genetic algorithm. It is shown that the evolved parameter values improved performance.

Keyword: RBF neural network, Forecasting, Genetic Algorithm, Time series, Rainfall

1. Introduction

Rain is one of nature's greatest gifts and in third world countries like Thailand; the entire agriculture depends upon rain. It is thus a major concern to identify any trends for rainfall to deviate from its periodicity, which would disrupt the economy of the country. The goal of prediction is to predict the future behavior of a time series from a set of known past values [1]. In some cases, apparent irregularity and randomness in time series may be due to chaotic behavior of a nonlinear of a nonlinear but deterministic system. This is done by first reconstructing a state space and then using nonlinear function approximations to create a dynamic model
The radial basis function (RBF) neural network became one of the most popular artificial neural networks, which used to approximate an unknown function. Most RBF network learning paradigms train sequentially the fixed size hidden layer first and then the output layer. While learning of the output layer is rather straightforward, the hidden layers must be carefully designed to achieve good performances. We propose a new sequential learning algorithm, which is able to adapt the structure of the network. Using this algorithm it is possible to create new hidden units. A genetic algorithm finds the width or spread factor of the hidden units radial basis function.

2. Radial basis function (RBF) Network

The Radial Basis Function (RBF) Networks is a single hidden layer feed forward neural network. Each node of the hidden layer has a parameter vector called the center. This center is used to compare with the network input vector to produce a radial symmetrical response. The response of the hidden layer are scaled by the connection weights of the output layer and then combined to produce the network output. The response of the jth hidden node to input data vector x_i, dimensionality M, is given by [3,4].

$$\phi_{ij} = \exp(-\alpha \|x_i - c_j\|^2) \quad (1)$$

where c_j is an M-dimensional center and α is a positive constant which determines the spread factor of the symmetric response of the hidden node. The network output is defined as

$$\hat{\gamma}_i = \sum_{j-1}^{k} \varphi_{ij} h_j \quad (2)$$

where h_j are the network's second layer connection weights and k is the number of hidden nodes.

The widely used RBF network, may use other functions e.g. piecewise linear, cubic approximation, the thin plate spline, the multiquadratic, and the inverse multiquadratic function [2] in place of the Gaussian.

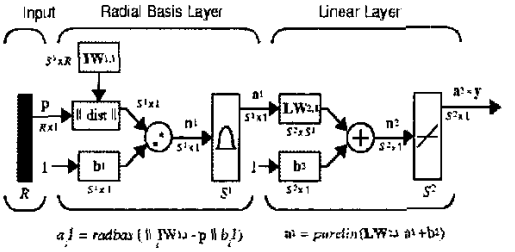


Figure 1. Architecture of an RBF neural network

Where : R = number of element in input vector
S^1 = number of neurons in layer 1
S^2 = number of neurons in layer 2
Dist = the Euclidean distance function

The performance of the networks are measured by a Mean-Squared-Error (MSE). The main objective of the training procedure is to approximate the underlying function of the system.

3. The new RBF Network with Proposed adaptive Structure

To determine an appropriate structure for the RBF network, a modified RBF network with adaptive structure is proposed. Initially, in the proposed structure, the radial basis layer has one hidden neuron. Afterward, the network iteratively appends one RBF node to the hidden layer at each training epoch until the error falls beneath an error goal or the maximum number of neurons has been reached. Unlike a traditional network with a fixed structure, the proposed network gradually searches for a minimum number of hidden nodes needed to meet the performance goal. The overall algorithm is given as the following:

Step 1. Initialise the network using the structure having a single neuron in the hidden layer.

Step 2. For each training epoch, feed all input vectors to the network and train the network according to the RBF training algorithm.

Step 3. Find the input vector in which the network output yields the greatest error.

Step 4. Add one neuron to the hidden layer with its weight vector equals to the vector obtained from Step 3.

Step 5. Repeat Step 2 until the performance goal is met or the maximum number of hidden nodes has been reached.

4. Genetic Algorithms

A genetic algorithm (GA) is an optimisation method [3,5] that manipulates a string of numbers in a manner similar to the way chromosomes are changed in biological evolution. Each string of numbers is called a "chromosome" or an "individual," and each number slot is called a "gene."

A set of chromosomes forms a population. Each chromosome represents a given number of traits that are the actual parameters that are being varied to optimise the "fitness function". The fitness function is a performance index that is expected to be maximised.

The operation of the GA proceeds in steps. Beginning with the initial population, "selection" is used to choose which chromosomes should survive to form a "mating pool." Chromosomes are chosen based on how fit they are (as computed by the fitness function) relative to the other members of the population.

More fit individuals end up with more copies of themselves in the mating pool so that they will more significantly effect the formation of the next generation. Next, several operations are taken on the mating pool. First, "crossover" (which represents mating, the exchange of genetic material) occurs between parents.

To perform crossover, a random spot is picked in the chromosome, and the genes after this spot are switched with the corresponding genes of the other parent. Following this, "mutation" occurs. This is where some genes are randomly changed to other values. After the crossover and mutation operations occur, the resulting strings form the next generation and the process is repeated. A termination criterion is used to specify when the GA should end (e.g., the maximum number of generations or until the fitness stops increasing).

The state of the algorithm in each generation consists of a population if bit string ϕ_i ,i = 1, ..., m, where m is the size of the population, and each bit string ϕ_i encodes the spread factor α_i of one RBF ϕ_i of (1) .

A fitness – based selection procedure is used to generate the population G_{k+1} form G_k . The training set in generation G_k will consist of a set of pairs, $(x_j, (fx_j))$, j = 1, ... p, in which $x_j \in \Re^n$ is a point in the domain of the function f to be approximated, and $f(x_j)$ is the known value of the function at that point.

The algorithm employs proportionate selection in a fixed-sized population, meaning that each slot to be filled in G_{k+1} is drawn by sampling from G_k with replacement, and with each encoded RBF ϕ_i in G_k having a probability of selection proportional to its performance. Thus, the individual RBF performance measure will

specify the expected number of copies of ϕ_i to occur in the next generation G_{k+1} before these copies are altered by genetic operators. After filling the slots in G_{k+1} with these selected copies, genetic operators of crossover and mutation.

The GA's parameter values used in this paper are shown in Table 1.

Table1. GAs Parameters for Time series prediction

Population Size	100
Generation Rate Replacement	0.25
Number of Traits	2
Crossover Rate	0.8
Mutation Rate	0.3
Maximum Generation	2,000

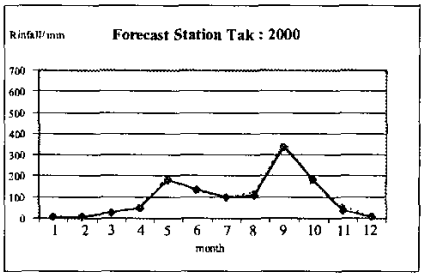
3. Experimental Results

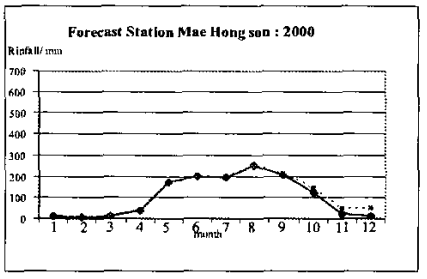
In this section, experimental results are presented in Table 2. The Department of Meteorology of Thailand provided the database used in this study. The database consists of 30 years' (1971 – 2000) of monthly rainfall values at of 12 climate stations.

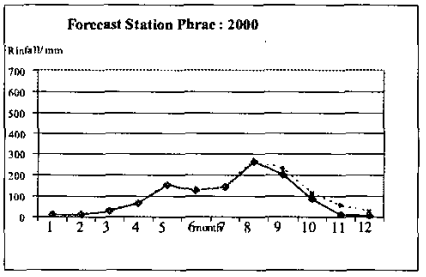
Table2. Results of the prediction errors obtained from different methods using 1971-1999 training data sets and 2000 test sets

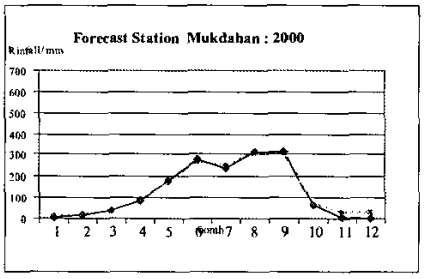
Rainfall Station	RBF Network with Adaptive using δ Obtained form GA	
	SSE	Sum of EDF with in 10 steps
Stations of Northern part Thailand		
Tak	148E-21	7.58E+01
Mae Hong Son	9.85E-03	5.32E+01
Phrae	1.02E-21	8.07E+01
Stations of North-Eastern part Thailand		
Mukdahan	1.93E-02	7.14E+01
Khon Kaen	2.07E-21	1.02E+02
Surin	1.92E-02	9.76E+01
Stations of Central part Thailand		
Suphan Buri	2.43E-21	1.09E+02
Nakhon Sawan	7.67E-02	1.02E+02
Bangkok metropolis	3.74E-21	1.66E+02
Stations of Southern part Thailand		
Songkhla	1.08E-02	1.24E+02
Trang	5.92E-21	1.53E+02
Phuket	1.45E-20	1.40E+02

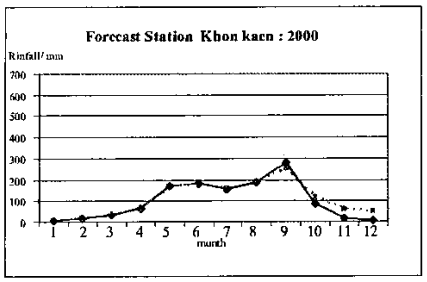
For training our network, the database from 1971- 1999 and test set of 2000 from the sliding windows of size 6 were feed in to the network repeated until end of training data set. The network goal is to minimise the value of sum square error (SSE) below than 0.002[7] After training, the network was used to predict the result of 2000. The output from the network was compared with the actual data in the time series. Figure 2 illustrates the spectral analysis on the test dataset to obtain a quantitative appreciation of the performance of the network.

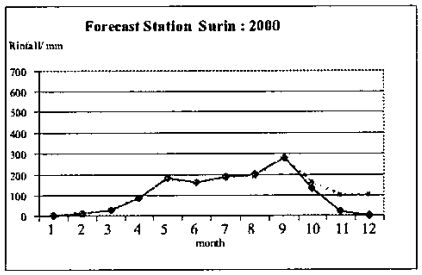


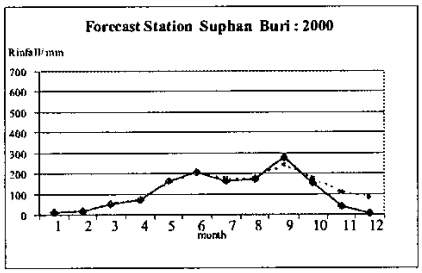


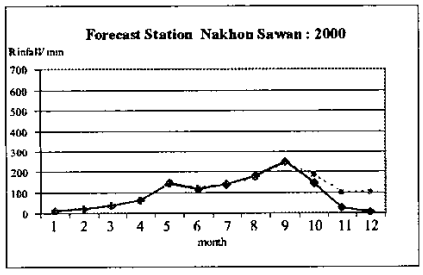


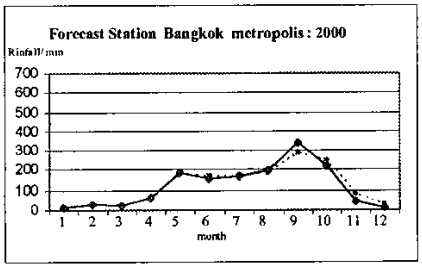


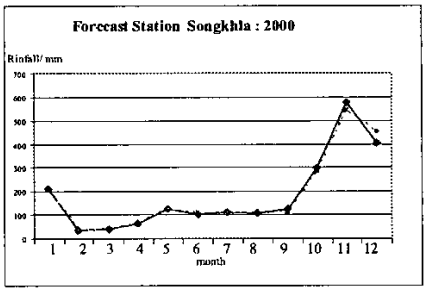


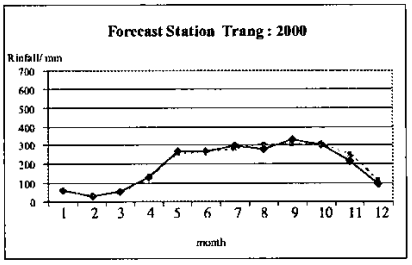












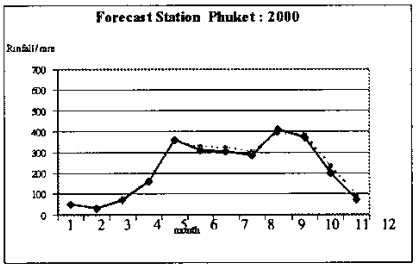
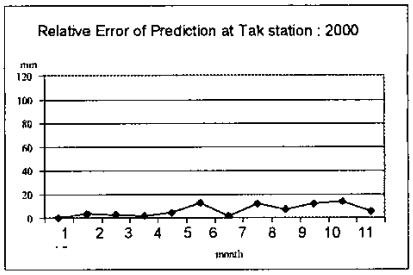
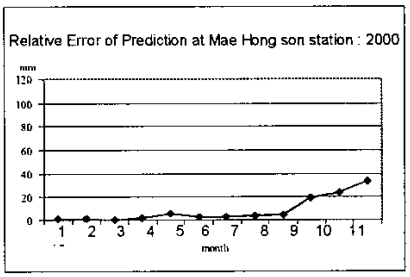
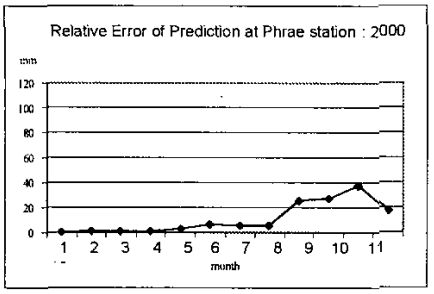
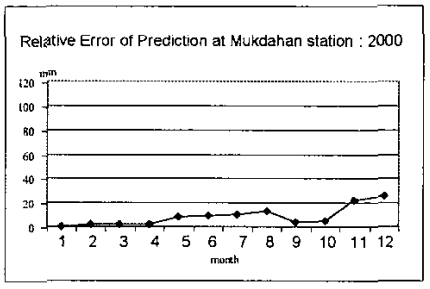


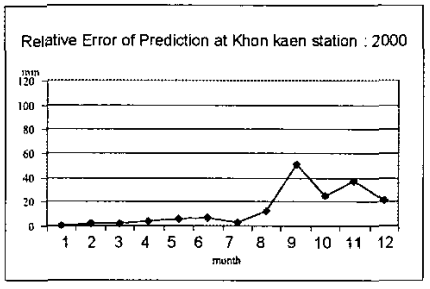
Figure 2. Actual data Rainfall 2000 (Solid Line) and Network forecasting 2000 (Dotted line)

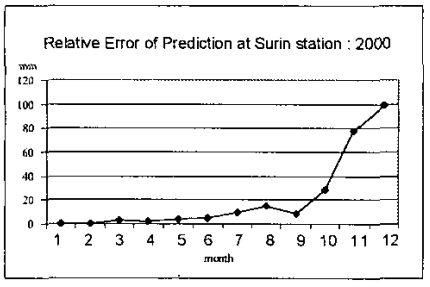


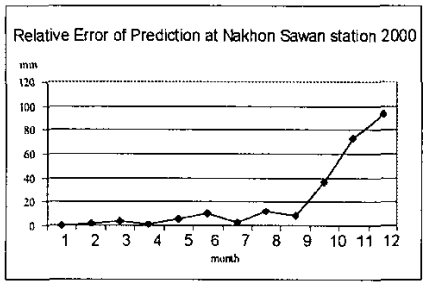


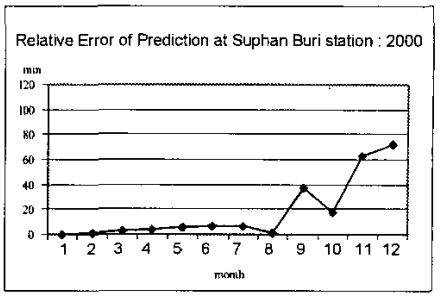


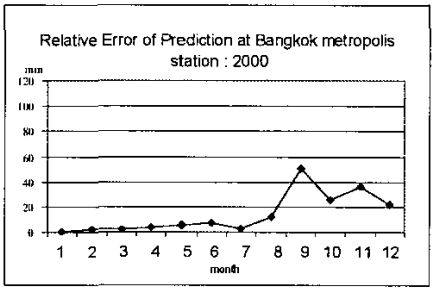


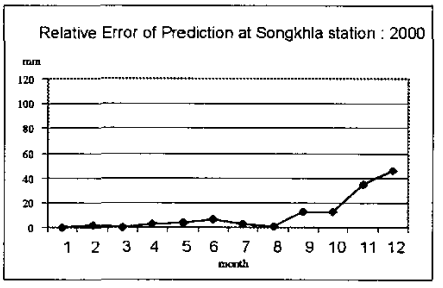


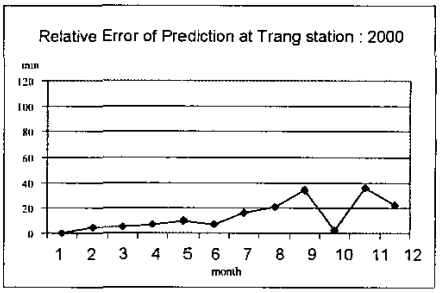












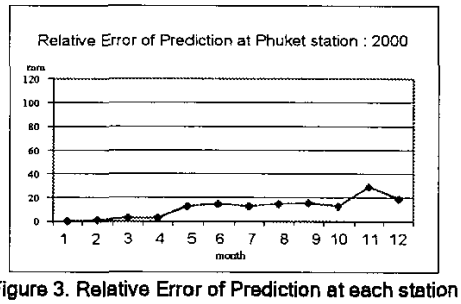


Figure 3. Relative Error of Prediction at each station

4. Conclusions

The paper presents a new method using an Adaptive RBF network model for rainfall forecasting of one year in advance, which is done by. The algorithm uses a genetic algorithm to determine an optimal value for the width (spread factor) of the hidden units. The network grows by iteratively adding one hidden node at each training epoch until the maximum performance is achieved. The suitable number of hidden neurons is thus obtained automatically. The test results show few deviations of the predicted rainfall value from the actual. Therefore, exploiting the benefit of Genetic Algorithms to optimise the adaptive RBF Network represents a very good model for the prediction of rainfall.

References

[1] A.Lapedes and R.Farber: "Nonlinear signal processing usingneural network; Prediction and system modeling," Los Alomos Net.Lab Tect. Rep:LA-UR-87-2262, 1987

[2] C. Jareanpon, N. Eua-Anant, C,Vithsupalert, and W. Pensuwon, "Radial Basis Function Networks with Adaptive Structure", The 3rd International Symposium on Communications and Information Technologies, September 2003, pp 621-625

[3] Eric P.Maillard and Didier Gueriot: "RBF Neural Network, Basis Functions and Genetic Algorithm," IEEE , pp. 2187 – 2192, 1997

[4] I.Rojas , H.Pomares , J.Gonzalez, E.Ros, M.Salmeron, J.Ortega, A.Prieto:"A New Radial Basis Functions Networks Stucture : Application to time series prediction," IEEE , pp. 449 – 454, 2000

[5] J.platt: "A resource – allocating network for function interpolation," Neural computa., Vol.3 , No.2 , pp. 213-225, 1991.

[6] S.Chen, "Combined Genetic Algorithm Optimisation and Regularised Orthogonal Least Squares Learning for Radial Basis Function Network", IEEE TRANSCTIONS OF NEURAL NETORKS, Vol.10, No.5, September 1999, pp.1239-1243

[7] Atmospheric Pressure Standard of Thailand , http://kanchanapisek.or.th/kp6/BOOK2/chapter4/t2-4-11.htm