Optimization of Climate Dowscaling Using Gradient Descent with Momentum and Quasi-Newton Methods

Wachiraporn Permpoonsinsup[1*], Dusadee Sukawat[2]

Received: 15 March 2013 Accepted: 15 June 2013

Abstract

This paper presents two optimization methods in training algorithm to minimize error of the feed forword neural network. Gradient descent with momentum and Quasi-Newton methods are applied to optimize weights in iteration of a training network model. Data from a global model are downscaled to four provinces in Thailand namely: Chaingmai, Bangkok, Ubonratchathani and Phuket. The results of experiments show that the Quasi-Newton method can minimize the error better than Gradient descent with the momentum method. Moreover, the number of hidden nodes of the network structure also affected the regression between the output and observed data.

Keywords: component, Downscaling, Artificial Neural Network, Gradient Descent, Quasi-Newton

Introduction

Artificial neural networks (ANNs) are mathematical models which can identify and represent nonlinear relationship between input and output data[1]. ANNs are trained by supervised learning. There are many optimization methods available for minimizing the error of objective function in ANN models. In this study, the gradient descent with momentum and the quasi-Newton back propagation method are presented. They are applied to optimize the weight between the input and hidden layers, and between the hidden and output layers. The data used in training are obtained from the 20th Century Reanalysis V2 reanalysis data of National Oceanic and Atmospheric Administration (NOAA)[2]. Although, the global climate models (GCM) can simulate temperature changes, the models are mainly project at coarse resolutions. Because data from the global model has coarse resolution, the data are interpolated before feeding to ANN model. Furthermore, interpolation is used to downscale the air temperature from a global model at 850 hPa (about 1.5 km above the ground) to Chaingmai, Bangkok, Ubonratchatani and Phuket of Thailand. Performance of the network is measured by mean square error. To minimize the error, ANNs are adjusted for the connection weights. Output data from the model are compared with observation data of the Thai Meteorological Department and then retraining the network to optimize the weight and reduce the error of the network.[3] has proposed the model that applied a neural network with a backpropagation algorithm for forecasting hourly water levels in the Chao Phraya River at Bangkok. Furthermore,[4] presented a new approach using an artificial neural network technique to improve rainfall forecast performance and ANN model were used for real time rainfall forecasting and flood management in Bangkok, Thailand.[5]has utilized a neural network model for monthly rainfall prediction for Chao Phraya River. In addition,[6] developed a time series forecasting model for a case study. The type of ANN implemented was multilayer perceptron with the quick-propagation training algorithm using time series factors.

[1] *Department of Mathematics, Faculty of Science King Mongkut's University of Technology Thonburi, Bangkok, Thailand E-mail: wachirapond@gmail.com*

[2] *Department of Mathematics, Faculty of Science King Mongkut's University of Technology Thonburi, Bangkok, Thailand E-mail: dusadee.suk@kmutt.ac.th*

** Corresponding author: E-mail: wachirapond@gmail.com*

This paper is organized as follows. Section II shows structure of the neural network model. Section III defines optimization methods. Next, Section IV designs methodology and data. Section V discusses the results. Finally, Section VI concludes the results of the models.

Artificial Neural Network

Neural Network Model Structure

The feedforward neural networks are the most widely used form of neural network in many practical applications. In this paper, a multilayer feedforward neural network is proposed in three layers; input, hidden and output layers[7]. Mathematically, the output network of the structure can be written as

$$y = f_2\left[\sum_{j=0}^{m} w_{jk} f_1\left(\sum_{i=0}^{n} w_{ij}x_i + b_1\right) + b_2\right] \quad (1)$$

where x_i is the *i*-th net input, y is output, w_{ij} is weight connection between input *i* and hidden neuron *j*, w_{jk} is weight connection between hidden *j* and output neuron *k*, *n* is the number of neuron in the input layer, *m* is the number of neurons in the hidden layer, f_1 and f_2 are the transfer functions, b_1 and b_2 are bias.

The activation function of neural network which is most used is the sigmoid function. It is very useful in neural networks trained by backpropogation and is defined as [8]

$$f(x) = 1/(1+\mathrm{e}^{-x}) \quad (2)$$

$$f'(x) = f(x)[1-f(x)] \quad (3)$$

where x is the net input, f is the sigmoid function.

Performance of the Network

The training of a neural network produces a small error on the training data set. A back propagation algorithm is used for the training of the neural networks. The objective of training is to reduce the error between the desired output and the neural network output. The performance of the network is defined as[3]

$$E = MSE = \frac{1}{N}\sum_{i=1}^{n}\left(y_i - t_i\right)^2 \quad (4)$$

where E is mean square error, y_i is network output data, t_i is observed data, n is the number of output data.

Optimization Learning

Gradient Descent with Momentum Algorithm (GDMA)

Gradient descent with momentum algorithm is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate[9].

The simplest gradient descent algorithm which is known as the steepest descent modifies the weights at time step *k* according to

$$w_{k+1} = w_k - \alpha g_k \quad (5)$$

A momentum term is added into the neural network model in learning algorithms. The new weight vector *w* is adjusted as

$$w_{k+1} = w_k - \alpha g_k + \mu w_k \quad (6)$$

where *w* is a weight vector in the network, k is iterative number, α is the learning rate which is a small positive number, g is the gradient operator with respect to the weights, μ is the momentum parameter.

Quasi-Newton Algorithm (QNA)

The quasi-Newton method is based on Newton's method and it approximates an inverse Hessian. The quasi-Newton algorithm operates in the BFGS (Broyden, Fletcher, Golfarb, Shanno) formula for updating the Hessian matrix. Iteratively, the update learning for this algorithm is[10]

$$w_{k+1} = w_k - \eta_k d_k \quad (7)$$

$$d_k = -B_k g_k \quad (8)$$

Where B_k is a positive definite matrix, d_k is the directions for approximating Newton's direction. η_k is the step size. w_k and w_{k+1} are iterated on the gradients $\Delta f(w_k)$ and $\Delta f(w_{k+1})$ then it can be written as

$$\Delta f(w_{k+1}) - \Delta f(w_k) \approx H(w_k)(w_{k+1} - w_k) \quad (9)$$

In iteration, B_{k+1} is defined as

$$B_{k+1}q_k = z_k \tag{10}$$

where

$$z = w_{k+1} - w_k \tag{11}$$

$$q = \Delta f(w_{k+1}) - \Delta f(w_k) \tag{12}$$

To calculate the matrix B_{k+1} from previous B_k by vector *q* and *z*, it can be written as

$$\begin{aligned} B_{k+1} = B_k + zz^T / \left(q^T q - B_k q q^T B_k\right) / \left(q^T B_k q\right) \\ + qB_k q^T + \left(z / z^T z\right) - \left(B_k q / q^T B_k q\right) \end{aligned} \tag{13}$$

Methedology and data

Interpolation Method

The method to downscale the large-scale to small-scale data in this paper is linear interpolation. Linear interpolation can calculate the value at an unknown data point between each pair of data points on a straight line. If a pair of data points is given by the coordinates (x_0, y_0) and (x_1, y_1) linear interpolation is[11]

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} \tag{14}$$

The Steps of Approach by Backpropagation Method

Phase 1. The air temperature data are downscaled by linear interpolation method. Then the data are passed to the neural network model.

Phase 2. Forward processes, the input data are fed into feedforward neural network and compute the network output. Then the network output are compared with the observed data by Eq. (4) to calculate the error performance.

Phase 3. Backward processes, if error of the network does not satisfy the predefined value (Epoch=1000), then optimize the weights by Eq. (6) or Eq. (7) and go to Phase 2. Otherwise stop iteration.

Experiment Design

In this paper, the parameter is monthly air temperature data from the 20th Century Reanalysis V2 reanalysis data of National Oceanic and Atmospheric Administration (NOAA), Department of Commerce, USA at 850 hPa from summer (March to June) and winter (November to February) from 2001 to 2010[2] are processed. The domain in this study is between latitude $0^\circ N$ to $20^\circ N$ and longitude $90^\circ E$ to $105^\circ E$. By linear interpolation, data are downscaled from 2° lat × 2° long grid to 0.1° lat × 0.1° long grid. The data from 2001 to 2008 are used for training, the data of 2009 for testing and the data of 2010 for validation in the neural network model. Eventually, the results are compared with the observed data of the Thai Meteorological Department. Table 1 presents the positions of the stations for downscaling.

Table 1 The positions of the stations for downscaling

Provinces	Position	
	Latitude	Longitude
Chiang Mai	18.4 $^\circ N$	98.5 $^\circ E$
Ubon Ratchathani	15.1 $^\circ N$	104.5 $^\circ E$
Bangkok	13.4 $^\circ N$	100.3 $^\circ E$
Phuket	8.8 $^\circ N$	95.1 $^\circ E$

Table II and Table III show the model design. The model design consists of algorithm, architecture network and season. There are two algorithms in the model that is GDMA and QNA. Furthermore, architectures of the models have the pattern as input node-hidden node-output node.

Table 2 model design for summer

	Model I	Model II	Model III	Model IV
Algorithm	GDMA	QNA	GDMA	QNA
Architecture	1-2-1	1-2-1	1-10-1	1-10-1
Season	Summer	Summer	Summer	Summer

Table 3 model design for winter

	Model V	Model VI	Model VII	Model VIII
Algorithm	GDMA	QNA	GDMA	QNA
Architecture	1-2-1	1-2-1	1-10-1	1-10-1

Results and Discussion

Gradient descent with momentum and quasi Newton methods are optimization methods which are applied in the training phase to optimize the weight in the neural network. To implement the algorithms, Encog Machine Learning Framework is used. Moreover, the goal of training algorithms is to minimize the error between network output and the desired output. In this section, the error is calculated for a supervised neural network. Mean square error (MSE) is the goal to minimize. The fitting between network output and the desired output is determined by considering the regression R values which measure the correlation between output data and observed data. If R is closed to 1 it means that the model can generalize network output well. Table 4 shows the results of the experiments.

Table 4 Regression of model

Model	CHIANG MAI		UBON RATCHATHANI		BANGKOK		PHUKET	
	summer	winter	summer	winter	summer	winter	summer	winter
I	0.38727	0.35406	0.27434	0.49786	0.33228	0.35473	0.33337	0.40196
II	0.45573	0.45332	0.30519	0.56679	0.50187	0.73951	0.42972	0.46195
III	0.40971	0.31151	0.38005	0.51820	0.38894	0.41312	0.63009	0.73140
IV	0.60937	0.66156	0.54078	0.57053	0.61443	0.77826	0.85921	0.88827

Conclusions and recommendation

In this paper, the 850 hPa grid-point temperature data from NOAA global model are downscaled to Chiang Mai, Ubon Ratchathani, Bangkok and Phuket of Thailand by linear interpolation. The data between 2001-2010 are divided into summer and winter. After downscaling by linear interpolation, the data are fed into the feedforward neural network. In network learning, weights are adjusted by GDMA and QNA.

In the experiments, the model with the 1-10-1 structure and trained by QNA has the highest regression. So it can conclude that QNA is better than GDMA and the number of node in hidden layer affected the regression between output and observed data.

For recommendation, the results show that most regressions of GDMA and QNA are less than 60%. In future work, QNA should be used to improve the process for minimizing the error and increasing the regression. Furthermore, as the number of node in the hidden layer is also significantly affect the performance from Table IV, the network should be designed to appropriately cover all nodes in the hidden layer.

Acknowledgment

This paper has been financially supported by the Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi.

References

1. K. Suzuki, 2011, "Artificial Neural Networks-Methodological Advances And Biomedical Applications", India : InTech.
2. National Oceanic and Atmospheric Administration (NOAA), http://www.noaa.gov/, 2013
3. T. Tingsanchali, "Forecasting model of Chao Phraya river flood levels at Bangkok",http://std.cpc.ku.ac.th/delta/conf/Acrobat/Papers_Eng/Volume%201/Tawatchai%20Paper%20OK.pdf
4. N.Q. Hung and et.al, "An artificial neural network model for rainfall forecasting in Bangkok, Thailand", Hydrology and Earth System Sciences,Vol.13, pp1413-1425, 2009.
5. N.Chantasut and et al, "Predictive Mining of Rainfall Predictions Using Artificial Neural Networks for Chao Phraya River", Proceddings of Joint Conference The 4th International of The Asian Federation of Information Technology in Agriculture, pp 117-122, August 2004

6. P. Kunlid and U. Smutkupt, “Application of Artificial Neural Network for Derate Loss Forecasting of Lignite Power Plant in Northern Region of Thailand”, http://psrcentre.org/images/extraimages/92.pdf
7. S.D Hunt and J.R.Deller, “Selective Training of Feedforward Artificial Neural Networks Using Matrix Perturbation Theory”, Elsevier Science Ltd. Great Britain, vol. 8, pp. 931–944, 1995.
8. J. Heaton, Introduction to Neural networks with Java, 2nd, Heaton Research, Inc, United States of America, pp. 119-123, 2008.
9. N. Qian, “Neural Networks Backpropagation General Gradient Descent”, Neural Networks. PERGAMON, vol. 12, pp. 145–151, 1999.
10. O. Mahmoud and et.al., “LEARNING ALGORITHM EFFECT ON MULTILAYER FEED FORWARD ARTIFICIAL NEURAL NETWORK PERFORMANCE IN IMAGE CODING”, Journal of Engineering Science and Technology. School of Engineering, Taylor’s University College, vol. 2, pp. 188–199, 2007.
11. A. kinson and et al, Elementary numerical analysis, 3 rd ed, New Jersey : Upper Saddle River, 2006.
12. Thai Meteorological Department, http://www.tmd.go.th
13. Encog Machine Learning Framework http://www.heatonresearch.com/encog