Statistics and Applications
Volume 9, Nos. 1&2, 2011 (New Series), pp. 55-69

Weather based forecasting model for crops yield using neural network approach

Ratna Raj Laxmi[1] and Amrender Kumar[2]
[1] *Reader, Department of Statistics, Maharshi Dayanand University, Rohtak, INDIA*
[2] *Scientist, Indian Agricultural Statistics Research Institute, New Delhi*

Abstract

Application of Neural Networks (NNs) for crop yields (rice, wheat and sugarcane) forecasting using Multi-Layer Perceptron (MLP) architecture with different learning algorithm has been attempted. For development of neural network based forecast models, yields of crop at district level (Uttar Pradesh state, India) was considered as output variable and indices of weather variables *viz.* maximum and minimum temperatures, rainfall and morning relative humidity were considered as input variables. Forecasts based on MLP architecture using Conjugate gradient descent algorithm for learning have been found to be close to the observed ones in most of the cases. The findings of the study substantiates that neural networks possess potential for prediction purposes.

Key words: Neural network; Multi-Layer Perceptron (MLP); Crop yield forecasting; Weather variables

1 Introduction

Reliable and timely forecasts provide important and useful input for proper, foresighted and informed planning, more so, in agriculture which is full of uncertainties. Agriculture now-a-days has become highly input and cost intensive. Without judicious use of fertilizers and plant protection measures, agriculture no longer remains as profitable as before. Uncertainties of weather, production, policies, prices, etc. often lead to losses to the farmers. Under the changed scenario today, forecasting of various aspects relating to agriculture are becoming essential. But in-spite of strong need for reliable and timely forecasts, the current status is far from satisfactory. Weather variability both within and between seasons is uncontrollable source of variability in yields. The impact of weather and climate on food production is of vital importance. Weather variables affect the crop differently during different stages of development. Thus extent of weather influence on crop yield depends not only on the magnitude of weather variables but also on the distribution pattern of weather over the crop season. Thus there is need to develop statistically sound objective forecasts of crop yield based on weather variables so that reliable forecasts can be obtained. Most of the earlier workers have utilised regression models (taking data as such or suitable transformation of data or some indices), discriminant function analysis, agro-meteorological models, etc. for crop yield forecasting (to cite a few, Choudhary and Sarkar (1981), Agrawal *et al.* (1986), Prasad and Dudhane (1989), Kumar and Bhar (2005).

Recently, Artificial Neural Network (ANN) has received a great deal of attention, because complicated problems can be treated by this even if the data are imprecise and noisy. Preliminary work on ANNs has been done by many workers, Cheng and Titterington (1994) made a detailed study of ANN models vis-a-vis traditional statistical models. They have shown that some statistical procedures including regression, principal component analysis, density function and statistical image analysis can be given neural network expressions. Warner and Misra (1996) reviewed the relevant literature on neural networks, explained the learning algorithm and made a comparison between regression and neural network models in terms of notations, terminologies and implementation. Kaastra and Boyd (1996) developed neural network model for forecasting financial and economic time series. Dewolf and Francl (1997,200) demonstrated the applicability of neural network technology for plant diseases forecasting. Zhang *et al.* (1998) provided the general summary of the work in ANN forecasting, providing the guidelines for neural network modeling, general paradigm of the ANNs especially those used for forecasting. They have reviewed the relative performance of ANNs with the traditional statistical methods, wherein in most of the studies ANNs were found to be better than the latter. Chakraborty *et al*. (1998) utilized the ANN technique for predicted severity of anthracnose diseases in legume crop. Gaudart *et al.* (2004) compared the performance of Multilayer perceptron (MLP) and that of linear regression for epidemiological data with regard to quality of prediction and robustness to deviation from underlying assumptions of normality, homoscedasticity and independence of errors and it was found that MLP performed better than linear regression. ANNs are data driven self adaptive methods in that there are a few prior assumptions about the models for problems under study. On the basis of examples, subtle functional relationalships among the data are captured even if the underlying relationships are unknown or hard to describe. ANNs can identify and learn correlated patterns between input data sets and corresponding target values through training. After training, ANNs can be used to predict the outcome of new independent input data and have great capacity in predictive modelling, i.e. all the characters describing the unknown situation can be presented to the trained ANNs, and then prediction of agricultural system may be feasible. In this article, an attempt has been made to study application of artificial neural networks for crops (rice, wheat and sugarcane) yield forecasting using weather variables for various districts of Uttar Pradesh.

2 Materials and Methods

Models for forecasting crop yields based on weather parameters utilize data for past several years for a location. Crop yield in different years is affected by technological change and weather variability. It can be assumed that the technological factors will increase yield smoothly through time and, therefore, years or some other parameter of time can be used to study the overall effect of technology on yield. Weather variability both within and between seasons is uncontrollable source of variability in yields. Weather variables affect the crop differently during different stages of development. Thus extent of weather influence on crop yield depends not only on the magnitude of weather variables but also on the distribution pattern of weather over the crop season which, as such, calls for the necessity of dividing the whole crop season into fine intervals and studying the relationship in these periods. This will increase number of variables in the model and in turn a large number of model parameters will have to be evaluated from the data. This will require a long series of data for precise estimation of the parameters which may not be available in practice. Thus, a technique based on relatively smaller number of manageable parameters and at the same time taking care of entire weather distribution may solve the problem.

District-wise data on productivity in respect of the crops (rice, wheat and sugarcane) from 1970-71 to 2000-01 were obtained from Directorate of Agriculture, U.P. Weekly data on weather variables for Allahabad, Varanasi, Ballia, Lucknow, Fatehpur, Hardoi, Banda and Jalaun from 1970 onwards were procured from India Meteorological Department whereas for Kanpur, Jhansi and Faizabad data were obtained from Central Research Institute for Dryland Agriculture, Hyderabad. For this study, weather data on maximum & minimum temperature, morning relative humidity and rainfall (MAXT, MINT, RHI and RF) were considered. For each agro-climatic zone, data for all districts were taken together. For development of forecast models weekly weather data for rice and wheat whereas for sugarcane fortnightly data were used. As weather during pre-sowing period is important for establishment of the crop, data starting from two weeks before sowing have been included in model development (for rice and wheat). Further, as the objective was to forecast yield well in advance of harvest, weather data till about 2½ months before harvesting have been considered. Therefore, for rice, weather data from 23rd standard meteorological week (smw) to 35th smw, for wheat data from 40th smw to 52nd smw and for sugarcane data from 2nd fortnight of March to 2nd fortnight of September were used for development of models. Models utilized weekly weather data for rice and wheat whereas for sugarcane fortnightly data were considered. In this study, models were developed through weather indices (WI) based regression approach and artificial neural networks approach using Multilayer perceptron (MLP) architecture with different learning algorithm *viz.* Backpropagation (BP), Levenberg – Marquardt(L-M) and Conjugate gradient descent (CGD). Forecasts have been obtained for subsequent years not included in model development. The performances of the developed models with different learning algorithm were compared with weather indices regression models based on mean absolute percentage error (MAPE).

2.1 Weather Indices (WI) based regression models

In this type of model, for each weather variable two indices have been developed, one as simple total of values of weather parameter in different weeks and the other one as weighted total, weights being correlation coefficients between variable to forecast and weather variable in respective weeks. The first index represents the total amount of weather parameter received by the crop during the period under consideration while the other one takes care of distribution of weather parameter with special reference to its importance in different weeks in relation to the variable to forecast. On similar lines, indices were computed with products of weather variables (taken two at a time) for joint effects (Agrawal *et al.* (1986) and Agrawal and Mehta (2007)).

2.2 Multilayer perceptron artificial neural network

Multilayer perceptron technique is very popular and is used more often than other neural network types. MLP is neural network in which the non-linear elements (neurons) are arranged in successive layers, and the information flows uni-directionally from input layer to output layer through hidden layer(s). MLP architecture can have a number of hidden layers with a variable number of hidden units per layer. The network thus has a simple interpretation as a form of input-output model, with weights and thresholds (biases) as free parameters of the model. Such networks can model functions of almost arbitrary complexity, with the number of layers, and the number of units in each layer, determining the function complexity. In a training of the network, through learning algorithm, produces its own output and tries to minimize the discrepancies between its own output and the target output. The minimization of discrepancies is done by weight adjustment during the learning phase. In this study, different learning algorithm viz. Backpropagation, Levenberg - Marquardt and Conjugate gradient descent were used for minimize the discrepancies between its own output and the target output. Neural network models with different hidden layers (one & two) and different number of neurons (4, 5 and 6) in a hidden layer with hyperbolic tangent(tanh) function as activation function i.e.

$$f(x) = (\exp(x) - \exp(-x))/(\exp(x) + \exp(-x))$$

with varying learning rate, were obtained and selected the best architecture, which having lowest Mean Absolute Percentage Error (MAPE). Klimasauskas (1991) suggests logistic activation functions for classification problems which involve learning about average behaviour, and to use the hyperbolic tangent functions if the problem involves learning about deviations from the average such as the forecasting problem. Therefore, in the present study, hyperbolic tangent (tanh) function has been used as activation function for neural networks model based on MLP architecture.

2.3 Partition of data

For models development, data set is divided into three distinct sets called training, testing and validation sets. The training set is the largest set and is used by neural network to learn patterns present in the data. In a training of the network, through different learning algorithm viz. Backpropagation, Levenberg - Marquardt and Conjugate gradient descent were used for the detecting the pattern in the data. The testing set is used to evaluate the generalization ability of a supposedly trained network. A final check on the performance of the trained network is made using validation set.

2.4 Learning algorithm

2.4.1 Back propagation algorithm

Backpropagation is the most commonly used method for training multilayered feed-forward networks. It can be applied to any feed-forward network with differentiable activation functions. For most networks, the learning process is based on a suitable error function, which is then minimized with respect to the weights and bias. If a network has differential activation functions, then the activations of the output units become differentiable functions of input variables, the weights and bias. If we also define a differentiable error function of the network outputs such as the sum of square error function, then the error function itself is a differentiable function of the weights. Therefore, we can evaluate the derivative of the error with respect to weights, and these derivatives can then be used to find the weights that minimize the error function by either using optimization method. The algorithm for evaluating the derivative of the error function is known as *backpropagation*, because it propagates the errors backward through the network. Multilayer feed forward neural network or multilayered perceptron (MLP), is very popular and is used more than other neural network type for a wide variety of tasks. MLP learned by backpropagation algorithm is based on supervised procedure, i.e. the network constructs a model based on examples of data with known output. The Backpropagation Learning Algorithm is based on an error correction learning rule and specifically on the minimization of the mean squared error that is a measure of the difference between the actual and the desired output. As all multilayer feedforward networks, the multilayer perceptrons are constructed of at least three layers (one input layer, one or more hidden layers and one output layer), each layer consisting of elementary processing units (artificial neurons), which incorporate a nonlinear activation function, commonly the logistic sigmoid or hyperbolic tangent(tanh) function.

The algorithm calculates the difference between the actual response and the desired output of each neuron of the output layer of the network. Assuming that $y_j(n)$ is the actual output of the j^{th} neuron of the output layer at the iteration n and $d_j(n)$ is the corresponding desired output, the error signal $e_j(n)$ is defined as:

$$e_j(n) = d_j(n) - y_j(n)$$

The instantaneous value of the error for the neuron j is defined as $e_j^2(n)/2$ and correspondingly, the instantaneous total error $E(n)$ is obtained by summing the neural error $e_j^2(n)/2$ over all neurons in the output layer. Thus,

$$E(n) = \frac{1}{2}\sum_j e_j^2(n)$$

In the above formula, j runs over all the neurons of the output layer. If we define N to be the total number of training patterns that consist the training set applied to the neural network during the training process, then the average squared error E_{av} is obtained by summing E(n) over all the training patterns and then normalizing with respect to the size N of the training set. Thus,

$$E_{av} = \frac{1}{2}\sum_{n=1}^{N} E(n)\,.$$

It is obvious, that the instantaneous error E(n), as well as the average squared error E_{av}, is a function of all the free parameters of the network. The objective of the learning process is to modify these free parameters of the network in such a way that E_{av} is minimized. To perform this minimization, a simple training algorithm is utilized. The training algorithm updates the synaptic weights on a pattern-by-pattern basis until one epoch, that is, one complete presentation of the entire training set is completed. The correction (modification) $\nabla w_{ji}(n)$ that is applied on the synaptic weight w_{ij} (indicating the synaptic strength of the synapse originating from neuron i and directing to neuron j), after the application of the n^{th} training pattern is proportional to the partial derivative $\frac{\partial E(n)}{\partial w_{ji}(n)}$.

Specifically, the correction applied is given by:

$$\Delta w_{ij} = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}$$

where **J** is the Jacobian matrix for the system, λ is the Levenberg's damping factor, δ is the weight update vector that we want to find and E is the error vector containing the output errors for each input vector used on training the network. The δ tell us by how much we should change our network weights to achieve a (possibly) better solution. The $\mathbf{J}^t\mathbf{J}$ matrix can also be known as the approximated Hessian. The λ damping factor is adjusted at each iteration, and guides the optimization process. If reduction of E is rapid, a smaller value can be used, bringing the algorithm closer to the Gauss–Newton algorithm, whereas if iteration gives insufficient reduction in the residual, $\boldsymbol{\lambda}$ can be increased, giving a step closer to the gradient descent direction. This algorithm requires computation of the Jacobian **J** matrix at each iteration step and the inversion of $\mathbf{J}^t\mathbf{J}$ square matrix, the dimension of which is N´N. This is the reason why for large size neural networks the LM algorithm is not practical.

2.4.3 Conjugate gradient descent algorithm

The back-propagation (BP) training algorithm is a supervised learning method for multi-layered feed-forward neural networks. It is essentially a gradient descent local optimization technique which involves backward error correction of network weights. Despite the general success of back-propagation method in the learning process, it has a number of serious drawbacks. Firstly, it is required to select three arbitrary coefficients: learning rate, momentum, and the number of hidden nodes. An unfortunate choice could cause slow convergence. Secondly, the network may become trapped in a local minimum of the error function, resulting in an unacceptable solution. Finally, it requires a large number of iterations to optimally adjust the network weights, hence it becomes unsuitable for large problems. Numerical optimization theory such as conjugate gradient offers a rich and robust set of techniques, which can be applied to train neural networks. These techniques not only focus on the local gradient of the error function, but also make use of its second derivative. The first derivative measures the slope of the error surface at a point, while the second measures the curvature of the error surface at the same point. This information is very important for determining the optimal update direction. As these methods make use of the second derivatives of the function to be optimized, they are typically referred to as second-order methods which also reduced the training time for feedforward neural networks (Barnard, 1992 and Johansson *et al.*, 1992). Hence, the Conjugate gradient learning algorithm is another alternative procedure to overcome these problems faced by back-propagation (BP) training and Levenberg Marquardt algorithm algorithm.

Let J(w) be the objective function to be minimized as the mean squared error

$$J(w) = \frac{1}{N}\sum_{n} E_n$$

where N is the number of patterns in the training set corresponding to the trajectory from time t_0 to t_1, E_n is the output error for the nth pattern and w is the weight vector. E_n is defined as

$$E_n = \frac{1}{2}\sum_{k}(d_{kn} - o_{kn}(w))^2$$

where o_{kn} and d_{kn} are the actual and desired output of the k^{th} output unit in response to the n^{th} input pattern respectively. Thus the objective function becomes

$$J(w) = \frac{1}{2N}\sum_{n}\sum_{k}(d_{kn} - o_{kn}(w))^2$$

A single evaluation of above equation requires the entire training set to be presented to the network, the errors to be calculated for each pattern, and the results to be summed and averaged. When the number of weights and training patterns increase, the cost of computing *J* (w) also increases. For computation of the gradient of the objective function, $\nabla J(w)$, differentiate the above equation with respect to w, yielding

$$g(w) = \frac{1}{N}\sum_{n} \nabla E_n(w)$$

Since the Conjugate Gradient algorithm requires both error function and gradient to be evaluated, the calculations should be performed together to maximize efficiency. Johansson *et al.* (1992) proposed the use the conjugate gradient method to train neural networks was utilised in this study.

2.5 Evaluation criterion

The performance evaluation measure considered is Mean Absolute Percentage Error (MAPE)

$$\text{MAPE} = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{Y_i - F_i}{Y_i}\right| x100$$

where Y_i and F_i are the observed and forecast values respectively and m is the number of observations for the hold-out data set.

3 Results and Discussion

Models were developed for Rice, Wheat and Sugarcane yields for Eastern Plain zone (Allahabad, Varanasi, Faizabad and Ballia), Central Plain zone (Kanpur, Lucknow, Fatehpur and Hardoi) and Bundelkhand zone (Jhansi, Banda, and Jalaun) of Uttar Pradesh taking crop yield as output variable and weather indices of weather variables on maximum and minimum temperatures, rainfall and morning relative humidity used as input variables. For each weather variable, weather indices (WI) have been developed at district level as weighted accumulation, weights being the correlation coefficients between detrended yield and weather variable in different periods (week for rice and wheat; fortnight for sugarcane). Similarly, for joint effects of weather variables, composite weather indices have been developed as weighted accumulations of product of weather variables (taken 2 at a time), weights being correlation coefficients between detrended yield and product of weather variables considered in respective periods**.** Deviations of the data from respective district averages were obtained which was used for model development. The developments of models at zone level pooled data of these indices (taking deviation from district average) were used.

3.1 Weather indices based regression model

For regression model, both weighted and un-weighted weather indices were used for models development. In these models, deviations of the data (weather indices) from respective district averages were used as independent variables while detrended yield was used as dependent variable. Stepwise regression technique has been used for selecting significant variables in all the models. The analysis has been done by using SAS (*Statistical Analysis System)* Version 9.1 software package available at Indian Agricultural Statistics Research Institute, New Delhi. Forecasts have been obtained for subsequent years 1998-99 to 2000-01 not included in model development.

3.2 Models based on neural networks approach

For each variable (weather indices of maximum temperature, minimum temperature, rainfall and morning relative humidity only) deviations of the data from respective district averages were used as independent / input variables and detrended yield as dependent / output variable in development of models. The entire data have been divided into three distinct sets, viz. training set, validation set and testing set is presented in Table 1.

Table 1: Number of data points in three sets for different agro-climatic zones for various crops

Crop	Zone	Training set	Validation set	Testing set
Rice	Central Plain	68	20	5
	Eastern Plain	40	20	7
	Bundelkhand	40	18	4
Wheat	Central Plain	62	20	3
	Eastern Plain	40	18	4
Sugarcane	Central Plain	61	20	7
	Eastern Plain	47	20	8

Neural network models using MLP architecture with different learning algorithm were for yield of rice, wheat and sugarcane have been obtained. The analysis has been done by using Statistica Neural Networks Version 6.1, available at Indian Agricultural Statistics Research Institute, New Delhi. The trained ANN models have been implemented for prediction of yields for subsequent cases corresponding to the years not included in the model development. For this, input information pertaining to these years were supplied to the trained models as a test data set. MLP based neural networks model with different hidden layers (1 and 2) and different number of neurons (4, 5 and 6) in a hidden layer with hyperbolic function as an activation function were developed and MAPE for various models were obtained. The performance of various models in terms of Mean Absolute Percentage Error (MAPE) for different crops for various combinations of nodes and layers (4N 2L means 4 nodes each in two hidden layers) is given in Table 2, while the MAPE for best selected models is given in Table 3.

Table 2: Mean Absolute Percentage Error (MAPE) for MLP based neural network models in different crops for various zone with different nodes and layers

Crop	Zone	Node and layer	**MAPE**			
			Backpropagation (BP)	Levenberg Marquardt (L-M)	Conjugate gradient descent (CGD)	Models based on Weather Indices (WI)

Rice	Central Plain	4N 1L	4.9	7.1	5.0	**4.4**
		5N 1L	4.5	3.7	0.9	
		6N 1L	7.1	4.0	3.9	
		4N 2L	2.7	4.4	3.6	
		5N 2L	4.8	3.9	3.3	
		6N 2L	7.5	5.6	4.0	
	Eastern Plain	4N 1L	4.2	3.3	2.9	**1.9**
		5N 1L	3.3	3.1	3.5	
		6N 1L	3.2	2.1	2.5	
		4N 2L	3.2	2.4	3.8	
		5N 2L	4.7	3.8	4.4	
		6N 2L	4.2	3.3	5.4	
	Bundelkhand	4N 1L	10.5	10.2	2.9	**4.6**
		5N 1L	6.0	3.0	6.4	
		6N 1L	9.3	6.6	4.8	
		4N 2L	6.9	6.8	7.4	
		5N 2L	6.3	5.7	6.8	
		6N 2L	6.5	2.8	11.8	
Wheat	Central Plain	4N 1L	2.6	7.0	2.4	**3.9**
		5N 1L	2.1	7.1	3.5	
		6N 1L	4.3	13.0	4.7	
		4N 2L	2.3	6.5	3.2	
		5N 2L	5.3	8.1	5.9	
		6N 2L	2.3	5.6	6.0	
	Eastern Plain	4N 1L	2.8	2.7	4.0	**3.7**
		5N 1L	4.8	9.6	4.3	
		6N 1L	4.3	4.7	6.7	
		4N 2L	2.0	3.6	0.4	
		5N 2L	6.6	5.3	3.9	
		6N 2L	4.1	4.3	2.7	

Sugarcane	Central Plain	4N 1L	1.5	2.1	1.3	**1.8**
		5N 1L	3.1	3.3	2.7	
		6N 1L	2.4	3.6	3.1	
		4N 2L	3.9	4.5	2.0	
		5N 2L	5.5	1.3	0.5	
		6N 2L	2.7	2.2	3.4	
	Eastern Plain	4N 1L	5.2	11.0	6.4	**5.4**
		5N 1L	6.2	4.2	0.6	
		6N 1L	6.0	7.9	5.7	
		4N 2L	5.2	3.3	5.1	
		5N 2L	9.7	7.9	5.5	
		6N 2L	8.4	5.6	8.8	

Table 3: Mean Absolute Percentage Error (MAPE) for MLP based neural network models in different crops for various zone with different nodes and layers

Crop	**Zone**	**MAPE**		
		MLP	**WI**	**Best selected Models**
Rice	Central Plain	0.9	4.4	CGD 5N 1L
	Eastern Plain	2.1	1.9	L-M 6N 1L
	Bundelkhand	2.9	4.6	CGD 4N 1L
Wheat	Central Plain	2.4	3.9	CGD 4N 1L
	Eastern Plain	0.4	3.7	CGD 4N 2L
Sugarcane	Central Plain	0.5	1.8	CGD 5N 2L
	Eastern Plain	0.6	5.4	CGD 5N 1L

Cheng, B. and Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. *Statistical Science*, 9, 2-54.

Dewolf, E.D., and Francl, L.J. (1997). Neural network that distinguish in period of wheat tan spot in an outdoor environment. *Phytopathalogy*, 87, 83-87.

Dewolf, E.D. and Francl, L.J. (2000). Neural network classification of tan spot and stagonespore blotch infection period in wheat field environment. *Phytopathalogy*, 20, 108-113.

Johansson, . M., Dowla, F. U. and Goodman, D. M. (1992). Backpropagation Learning for Multilayer Feed-Forward Neural Networks Using the Conjugate Gradient Method. *International Journal of Neural Systems*, 2(4),pp 291 - 301.

Kaastra, I. and Boyd, M. (1996): Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3), 215-236.

Konstantinos, A. (2000). Application of Back Propagation Learning Algorithms on Multilayer Perceptrons, Project Report, Department of Computing, University of Bradford, England.

Kumar, Amrender and Bhar, Lalmohan (2005). Forecasting model for yield of Indian mustard (Brassica junea) using weather parameter. *Indian journal of agricultural science*, 75(10), 688-90.

Klimasauskas, C.C. (1991). Applying neural networks. Part 3: Training a neural network, PC-AI, May/ June, 20–24.

Prasad, R. and Dudhane, S. N. (1989). Forecasting mustard yield in gangetic West Bengal using rainfall and agricultural technology. *Mausam*, 40(4): pp 441-6.

Warner, B. and Misra, M. (1996). Understanding neural networks as statistical tools. *American Statistician,* 50, 284-93.

Zhang, G., Patuwo, B. E. and Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting,* 14, 35-62.