Load Forecasting For Practical Power Systems by Using Artificial Neural Networks and Classical Techniques – A Case Study

Mr. N.M.G. Kumar[1], Dr.P.S.Sangameswara Raju[2], Mr.P.Venkatesh[3]

[1]*Research scholar, Department of EEE, S.V.U. College of Engineering.&*
Associate Professor, Department of EEE., Sree Vidyanikethan Engineering College.
[2]*Professor, Department of E.E.E., S.V.U. College of Engineering, Tirupati, Andhra Pradesh, India.*
3Asst.Prof, Dept of EEE, Sree Vidyanikethan Engineering College, Tirupati, Andhra Pradesh, India.

ABSTRACT:

This paper presents a method of STLF (Short Term Load Forecasting) in electrical power system using ANN. In this a multilayered feed forward network with back propagation learning algorithm is used because of its god generalizing property. The input to the Neural network is in terms of past load data which are heuristically chosen such that they reflect the trend, load shape as well as some influence of weather, but here the weather data is not used to train the network. The network is trained to predict one hour ahead load forecasting in both ANN as well as Conventional (Regression) methods. This method is trained for one month load forecasting data and two months load forecasting data and found that the MAPE(Mean absolute percentage error) obtained is small especially for prediction of load for hourly in considering two months data than one month data to train the network.

Keywords: *STLF, ANN, Load Forecasting, Load demand*

I. INTRODUCTION

Forecasting of load is what will happen in the future by simulating the loads for certain conditions or trends with the help of available past data. Over the years forecasting has been refined considerably and has now reached a stage where it is more precise and unbiased. In Electrical Power Systems, there is a great need for accurately forecasting the load and energy requirements because electricity generation and distribution are a great financial liability to the State Corporation. Accurate Forecasting is also necessary because availability of electricity is one of the several important factors for agricultural and industrial development. If the Energy Forecast is too conservative, then there will be a chance of generating capacity falling short of the actual demand, resulting in the restrictions being imposed on the power supply, which will affect the economic aspects of well being of the corporation of the state. If the Forecast is too optimistic, it may lead to excess generating capacity resulting in part of the investment not giving immediate returns.

A developing country like India the financial resources are minimum, it cannot afford to face the two situations mentioned above. So it must be appropriate to consider Electrical Load Forecasting importance. In order to do proper estimation of load and analyzing its pattern, to receive adequate rate of return of the investment for maximum utilization of the plant, it is important to consider both i) magnitude of the system load and ii) location of these loads. Several electric power companies have adapted the conventional methods for forecasting the future load. In conventional methods, the models are designed based on the relationship between load power and factors influencing load power. The conventional method has the advantage that we can forecast load power with a simple prediction model. However, since the relationship between load power and factors influencing load power is nonlinear, it is difficult to identify its non-linearity by using conventional methods. It is observed that perfect prediction is hardly ever been possible.

The thought of using the Artificial Neural Networks gives reasonably good predictions while estimation of loads and model can be utilized effectively when compared with the statistical techniques. It is mainly due to the ability of ANN in supporting a non-linear mapping between input and output variables along with its special features such as robust performance in the noisy and incomplete data environments, high parallel computation and self-learning. General software for Forecasting the load for different scenarios is developed and variables are trained using the MATLAB. STLF is an essential tool in operation and planning of the power system. It helps in coordinating the generation and area interchange to meet the load demand. It also helps in security assessment, dynamic state estimation, load management and other related functions. STLF are primarily used for economic load dispatch, daily operation and control and assurance of reliable power supply. In the last few decades, various methods for STLF have been proposed which are

Statistical Methods

Statistical methods include multiple linear regression, autoregressive models, stochastic time series, state space methods4, general exponential smoothing, etc. A detailed review of these methods is given in Gross and Galiana and Moghram and Rahman7. These methods though having a sound mathematical understanding are generally based on statistically assumed model and different environmental factors. The accuracy of such methods depends on the accuracy and relevance of this heuristically modeled load.

Artificial Neural Networks

The weights on the interconnections are estimated iteratively by a non-linear optimization method using known sets of input and output data, such adaptation of the network is referred to as the 'Training' or the 'Learning' of the network. The underlying idea is the biological nervous system-like performance of the network in learning complex processes. The main characteristic features are the network architecture, the non-linearity associated with the nodes and the training method used. The designs are not unique, and there may be heuristics associated with the specification of the network structure. The diagram shown below depicts the basic functioning of a neural network Figure 1. The output of the neural network is compared with the target (desired output) and the weights are adjusted so as to minimize the error between the output and the target. This process is repeated till the desired level of accuracy is achieved

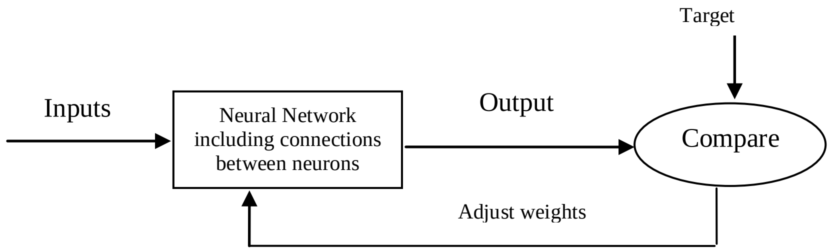


Figure 1 functioning of a neural network

Recently, Artificial Neural Networks (ANN) has been used for STLF. In this paper, a short term load forecasting method using the ANN is proposed. A multilayered feed forward (MLFF) neural network with back propagation learning algorithm has been used because of its simplicity and good generalization property. These methods used various network architectures and supervised as well as unsupervised learning rules. A brief review of performance of popular architectures is given by Parll, et al8. Though standard design procedures differ from author, networks are mostly designed on trail and error and engineering judgments. This is not a major obstacle because these trial and error methods can be easily implemented. Feed forward networks are extensively used for the problem because of easier implementation. A large number of architectures based on feed forward models used for daily Average, daily Peak, and daily hourly load for both one month and two months are reported. The input, to the neural network is based only on past load data and are heuristically chosen in such a manner that they inherently reflect all the major components, such as, trend, type of day, hour of the day and load.

II. MATAMATICAL MODEL:

Regression Models:

Regression models fall under the category of casual models of forecasting. In many models there are two or more variables are related, and it is of interest to model and explore this relationship. In general, suppose that there is a single dependent variable or response variable that depends on one or more independent variables. The relationship between these variables is characterized by a mathematical model called a 'Regression Model'. The following are the various regression model equations that are available in the literature.

Simple linear regression:

$$Y_t = a + bX……………..(1)$$

Multiple linear regressions:

$$Y_t = a + bX^1 + cX^2 + ………………..(2)$$

Curvy linear regression:

$$Y_t = a + bX + cX^2 + dX^3 +….……………..(3)$$

In the present work simple linear regression models are developed for daily average load forecasting and multiple linear regression models are developed for hourly load forecasting.

III. ESTIMATION OF PARAMETERS IN LINEAR REGRESSION MODELS SIMPLE REGRESSION:

Where X is independent variable and Y is dependent variable. Where “a” indicates intercept

$a = \bar{y} - b\bar{x}$ .Where “b” indicates $b = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n\bar{x}^2}\ where\bar{x} = \frac{\sum x}{n}\ and\bar{y} = \frac{\sum y}{n}$ = slope……………..(4)

Multiple Linear Regressions

The method of least squares is typically used to estimate the regression coefficients in a multiple linear regression model. Suppose that n>k observations on the response variable are available say, y_1, y_2… y_n. Along with each observed response y_i, we will have an observation on each regressed variable and let x_{ij} denote the ith level of variable x_j. We assume that the error term € in the model has E(€) =0 and V(€) = σ^2 and that the {€ i} are uncorrelated random variables. We may write the model equation

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + €_I \text{……………..(5)}$$

$$y_i = \beta_o + \sum_{j=1}^{k} \beta_j x_{ij} + \in_i \text{…………………………..(6)}$$

i = 1, 2… n

The method of least squares chooses the β’s in equation (6) so that the sum of the squares of errors, $€_I$, is minimized. The least squares function is

$$L = \sum_{i=1}^{n} \in_i^2 = \sum_{i=1}^{n} (y_i - \beta_o - \sum_{j=1}^{k} \beta_j x_{ij})^2 \text{……………..(7)}$$

The function L is to be minimized with respect to β_0, β_1… β_k. The least square estimators, say β_0, β_1… β_k, must satisfy

$$\frac{\partial L}{\partial \beta} = -2\sum_{i=1}^{n}(yi - \hat{\beta}0 - \sum_{j=1}^{k}\hat{\beta}jxij) = 0 \text{ ……………..(8)}$$

$$\frac{\partial L}{\partial \beta} = -2\sum_{i=1}^{n}(yi - \hat{\beta}0 - \sum_{j=1}^{k}\hat{\beta}jxij)xij = 0 \text{ ……………..(9)}$$

j = 1, 2, ------, k.

Simplifying equation (9), we obtain

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} + \ldots.. + \hat{\beta}_k \sum_{i=1}^{n} x_{ik} = \sum_{i=1}^{n} y_i \text{ ……………..(10)}$$

$$\hat{\beta}_0 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 + \hat{\beta}_2 \sum_{i=1}^{n} x_{i1}x_{i2} + \ldots.. + \hat{\beta}_k \sum_{i=1}^{n} x_{i1}x_{ik} = \sum_{i=1}^{n} x_{ik}y_i \text{ ……………..(11)}$$

$$\hat{\beta}_0 \sum_{i=1}^{n} x_{ik} + \hat{\beta}_1 \sum_{i=1}^{n} x_{ik}x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{ik}x_{i2} + \ldots.. + \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 = \sum_{i=1}^{n} x_{ik}y_i \text{ ……………..(12)}$$

These equations are called the least squares normal equations. Note that there are p = k + 1 normal equations, one for each of the unknown regression

Defining Problem:

The problem of load forecasting is approached by making the forecasts for one whole day at a time. The approach is static in the sense that the forecast is not updated during the day. The forecasting of the load on the daily basis with neural network techniques has been reported in many variations. Single feed forward artificial neural network architecture is used in forecasting the daily peak, valley, and average loads.

These equations are called the least squares normal equations. Note that there are p = k + 1 normal equations, one for each of the unknown regression coefficients. The solution to the normal equations will be the least squares estimators of the regression coefficients $\hat{\beta}_0, \hat{\beta}_1, \ldots\ldots, \hat{\beta}_k$.

It is simpler to solve the normal equations if they are expressed in matrix notation. We now give a matrix development of the normal equations that parallels the development of equation (10, 11, 12). The model in terms of the observations, of the above equation may be written in matrix notation as Y = X β + € Where,

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_n \end{bmatrix}, \ldots\ldots\ldots\ldots\ldots..(13)$$

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{1k} \\ 1 & x_{21} & x_{22} & x_{2k} \\ 1 & x_{31} & x_{32} & x_{3k} \\ 1 & x_{n1} & x_{n2} & x_{nk} \end{bmatrix}, \ldots\ldots\ldots\ldots\ldots..(14)$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_k \end{bmatrix} \quad \text{and} \quad \ldots\ldots\ldots\ldots\ldots..(15)$$

$$€ = \begin{bmatrix} \in_1 \\ \in_2 \\ \in_n \end{bmatrix} \ldots\ldots\ldots\ldots\ldots..(16)$$

In general, y is an (n x 1) vector of the observations, X is an (n x p) matrix of the levels of the independent variables, β is a (p x 1) vector of the regression coefficients, and € is an (n x 1) vector of random errors. We wish to find the vector of least squares estimators, $\hat{\beta}$ that minimizes

$$L = \sum_{i=1}^{n} \in_i^2 = \in' \in = (y - x\beta)'(y - x\beta) \ldots\ldots\ldots\ldots\ldots..(17)$$

Note that L may be expressed as

$$L = Y^1 Y - \beta^1 X^1 Y - Y^1 X \beta + \beta^1 X^1 X \beta$$
$$= Y^1Y - 2\beta^1X^1Y + \beta^1X^1X\beta \ldots\ldots\ldots\ldots\ldots..(18)$$

Because β^1X^1Y is a (1 x 1) matrix, and its transpose $(\beta^1X^1Y)^1 = Y^1X\beta$ is the same matrix. The least squares estimators must satisfy

$$\left.\frac{\partial L}{\partial \beta}\right|_{\hat{\beta}} = -2X^1Y + 2X^1X\hat{\beta} = 0 \ldots\ldots\ldots\ldots\ldots..(19)$$

This simplifies to $X^1X\hat{\beta} = X^1Y$

equation (19) is the matrix form of the least squares normal equations. It is identical to equation (18). To solve the normal equations, multiply both sides of equation (15) by the inverse of X^1X. Thus, the least squares estimator of β is

$$\hat{\beta} = (X^1X)^{-1}X^1Y \ldots\ldots\ldots\ldots\ldots..(20)$$

It is easy to see that the matrix form of the normal equations is identical to the scalar form. Writing out the above equation in detail we get If the indicated matrix multiplication is performed, the scalar form of the normal equations (i.e equations (3.4)) will result. In this form it is easy to see that X^1X is a (p x p) symmetric matrix and X^1Y is a (p x 1) column vector. Note the special structure of the X^1X matrix. The diagonal elements of X^1X are the sum of squares of the elements in the columns of X, and the off-diagonal elements are the sums of cross-products of the elements in the columns of X. Furthermore, note that the elements of X^1Y are the sums of cross-products of the columns of X and the observations $\{Y_i\}$.

$$\begin{bmatrix} n & \sum_{i=1}^{n} X_{i1} & \sum_{i=1}^{n} X_{i2} \cdots & \sum_{i=1}^{n} X_{ik} \\ \sum_{i=1}^{n} X_{i1} & \sum_{i=1}^{n} X_{i1}^2 & \sum_{i=1}^{n} X_{i1}X_{i2} \cdots & \sum_{i=1}^{n} X_{i1}X_{ik} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{n} X_{ik} & \sum_{i=1}^{n} X_{ik}X_{i1} & \sum_{i=1}^{n} X_{ik}X_{i2} \cdots & \sum_{i=1}^{n} X_{ik}^2 \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} Y_i \\ \sum_{i=1}^{n} X_{i1}Y_i \\ \vdots \\ \sum_{i=1}^{n} X_{ik}Y_i \end{bmatrix} \ldots\ldots\ldots\ldots\ldots..(21)$$

The fitted regression model is

$$\hat{Y} = X\hat{\beta} \ldots\ldots\ldots\ldots\ldots..(22)$$

In scalar notation, the fitted model is

$$\hat{Y}_i = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_j X_{ij} \qquad \text{……………..(23)}$$

j = 1, 2… n.

The difference between the actual observation Y_i and the corresponding fitted value $\hat{Y}_i$ is the residual; say $e_i = Y_i - \hat{Y}_i$. the (n x 1) vector of residual is denoted by

$$e = Y - \hat{Y} \text{ ……………..(24)}$$

ANN DESIGN:

ANN models work in the manner similar to human brain. In fact the basic ANN element is simulation of biological neuron. A biological neuron is composed of different parts performing different functions. Input information is received by the dendrites and act similar to the input node. These dendrites pass the information to soma the cell body or the processing node. The output of the soma is then transferred through the axon to the synapse. The synapse then gives a weight called synoptic weights to this information. This weighted input serves as the input to the dendrite of the next neuron. Brain consists of large number of such neurons with complex interconnections among them. The processing node in the human brain is a hard limiter. Since it is practically impossible to have such a large number of neurons and interconnection a continuous sigmoid function is used as a processing node in artificial simulation.

ANNs works on the basis of mapping of input/output spaces. The relationship between the input variables and the resulting responses there from are determined in an implicit manner. This is done through the training process. Training of the network is accomplished through the correction of the network weights given to each of variables using steepest gradient or other gradient method so as to minimize the sum squared error on responses. The general method of arriving at a neural network design is shown in figure 1. After gathering sample set of inputs and the related output responses (training samples) the network is trained. The learning parameters (learning constant, momentum etc) are adjusted till the network is trained successfully ie, with lowest possible sum squared error. The trained network is tested on the samples other than the training samples. The network parameters ie, number of layers and neurons in each layer are adjusted till successful testing of the network.For any neural network the design procedure involves, (i) feature selection (selection of input variables) and (ii) parameter selection for the network (number of layers, neurons etc). For the current load forecast problem the two procedures go hand in hand. In this case the dependence of the future load could be related to the load at various previous hours based on the past experience and judgment. However the suitability of selected set in modeling the said input/output relationship can only are verified by observing the performance of the network. A network which requires number of layers and neurons in a layer is selected which is fit to capture the input/output relationship. There is no fixed rule to decide the number of layers and neurons in each layer for a given problem. The number of layer depends on the complexity of the problem. It has been observed that a single hidden layer is sufficient for the problem. It is also found that 8-20 neurons in a single hidden layer capture the load pattern closely. For this appropriate number of training and testing patterns are selected from the historical data. It is also important to note that a network with low training error may not always lead to a good prediction, since forecasting not only requires good pattern matching but generalization as well. This balance between training and prediction error can be achieved by training the network of the said training patterns with a few number of epochs (say few thousand) then it is tested on test patterns. This training and testing continues till the prediction error or training error improves. This procedure may be repeated for other combination and one with lowest error is selected.

ANN Method :(Applications)

Selection of Network Architecture: (Maximum, Minimum, Average load)

There are numerous ways to choose the architecture of the models. Here a decision is made to use one network for all day types. It was concluded that networks with only one output node gives better results than forecasting peak-, valley-, and average loads with one multi-output node. Other features to be decided about the architecture of the network are the input variables and the number of hidden layer neuron variables. To forecast the maximum load L_{max} (*i*) (minimum and average loads) of a certain day, the maximum load (minimum, average loads) of the previous day can be considered as potential input variable. Therefore the network configuration can be summarized as given below: Note: number of neurons in a hidden layer can approximately be determined from the form H = T / {5(N+M)} Where, H= number of hidden layer neurons. N= size of the input layer M= size of the output layer T= size of the training set Training set: The training set for the networks consists of the data over one year, from April $1^{st,}$ 1999 to March 31st, 2000.

The idea is that all load conditions are representative in the training in order to enable the model to adapt to all conditions. Therefore the size of the training set is 1×366Test data: The test data consisting data of one month from April 1st, 2000 to April 30th, 2000. Initialization of weights: Weights are opted to be initialized randomly Activation functions: The activation function used in the hidden layer is ‘tan sigmoid’. The activation function used in output layer is ‘purelinear’. Training algorithm: Batch training is adopted in training the said network. Gradient descent with momentum, which is a variation of back propagation algorithm, is used.

$$\mathrm{E}_{\mathrm{avg}} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\left|\hat{L}_i - L_i\right|}{L_i}\right] * 100\% \cdots\cdots\cdots\cdots\cdots..(25)$$

Where, N = the number of cases to be forecast (number of days in the cases of peak-, valley- and average load forecasting, and number of hours in the case of hourly load forecasting). L_i = the ith load value $\hat{L}_i$ = the ith load forecast

Defining the Problem (Hourly):

The problem of load forecasting is approached by making the forecasts for one hour at a time. The hourly load forecasts can be obtained either by forecasting the whole daily load curve at one time with a multiple-output network or by forecasting the load with a single-output network for one hour at a time. Here it has been used a network which has only one output node, which provides the forecasts for the next hour.

Selections of Network Architecture:

There are many alternative ways to build a multilayer feed forward artificial neural network for forecasting the hourly load. The most important decisions concern the selection of the input variables and the network architecture. The model consists of one feed forward artificial neural network which has one output node. This corresponds to L (*i*), the load at hour ‘*i*’. The input contains load data of the previous day as well as previous week in addition to the most recent hours. The load inputs therefore are:

L_k = L (*i*-k), where k=1, 2, 3
L_4 = L (*i*-24)
L_5 = L (*i*-168)

In addition to the load values, the network is also given with the inputs indicating the day type. Each day type is assigned with a binary value that gets the corresponding value of the target hour. The binary value assignments for each day are given below.

Sun001, Mon010, Tue011, Wed 1 0 0, Thu101, Fri1 1 0, Sat1 1 1

It was also concluded that including the input variables to inform the network of the hour of the day improves the forecasting accuracy significantly. The 24 hours of the days are encoded in five bit binary representation i.e. the first hour is encoded as 00001, second hour as 00010 and so on the twenty fourth hour as 11000.Training set: in the previous chapter, the date over the whole year was used for training a network to predict peak-, valley-, and average loads. The idea was to train the network to recognize the characteristics of all seasons. However, in forecasting the hourly load the training set is 24 times larger, and the networks also have to be larger. The training can become very exhaustive.

On the other hand, with a short training period the network can be trained fast. So short training sets consisting of one month and two months data are employed separately to train the same network architecture. For one month training, the training set is from December 1st, 2001 to December 31st, 2001. and for two months training, the training set is from November 1st, 2001 to December 31st,2001Test data: The test data consisting, data of one week from January 1st, 2002 to January 7th, 2002.Initialization of weights: Weights are opted to be initialized randomly Activation functions: The activation function used in the hidden layer is ‘tan sigmoid’. The activation function used in output layer is ‘log sigmoid’.

Test Results:

Defining the Problem (daily)

The problem of load forecasting is approached by making the forecasts for one hour at a time. The hourly load forecasts can be obtained either by forecasting the whole daily load curve at one time with a multiple-output network or by forecasting the load with a single-output network for one hour at a time. Here it has been used a network which has only one output node, which provides the forecasts for the next hour.

Selections of Network Architecture:

There are many alternative ways to build a multilayer feed forward artificial neural network for forecasting the hourly load. The most important decisions concern the selection of the input variables and the network architecture. The model consists of one feed forward artificial neural network which has one output node. This corresponds to L (*i*), the load at hour '*i'*. The input contains load data of the previous day as well as previous week in addition to the most recent hours. The load inputs therefore are:

L_k = L (*i*-k), where k=1, 2, 3
L_4 = L (*i*-24)
L_5 = L (*i*-168)

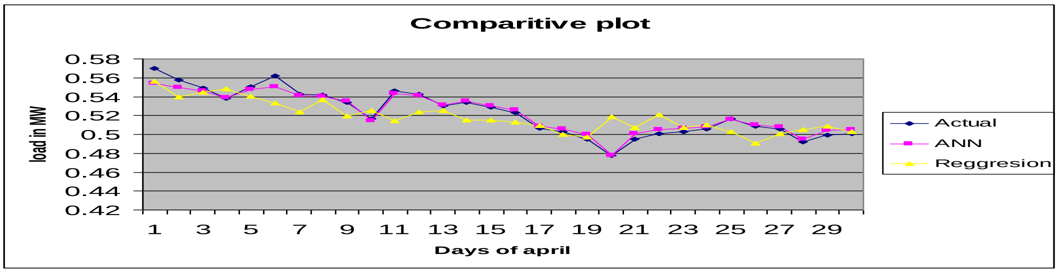
It was also concluded that including the input variables to inform the network of the hour of the day improves the forecasting accuracy significantly. The 24 hours of the days are encoded in five bit binary representation i.e. the first hour is encoded as 00001, second hour as 00010 and so on the twenty fourth hour as 11000.Training set: in the previous chapter, the date over the whole year was used for training a network to predict peak-, valley-, and average loads. The idea was to train the network to recognize the characteristics of all seasons. However, in forecasting the hourly load the training set is 24 times larger, and the networks also have to be larger.

The training can become very exhaustive. On the other hand, with a short training period the network can be trained fast. So short training sets consisting of one month and two months data are employed separately to train the same network architecture. For one month training, the training set is from December 1st, 2001 to December 31st, 2001. And for two months training, the training set is from November 1st, 2001 to December 31st, 2001.Test data: The test data consisting, data of one week from January 1st, 2002 to January 7th, and 2002.Initialization of weights: Weights are opted to be initialized randomly.

Test Results:

Defining the Problem (Hourly)

Hourly Load Forecasting With One Month Training Set Load data of the month of December 2001 is selected for one month training set. As December month contains 31days, altogether we have 744 hours (i.e. 31days*24hours). For each hour a column will be generated. As we have opted for 13 inputs each column contains 13 elements. So the size of input vector will be 13×744 and that of target vector will be 1×744. Testing is done for January 1st, 2002 to January 7th, 2002 (i.e., one week load data). So altogether testing is performed for 168 hours



Comparative Plot :(Avg. Load Demand)

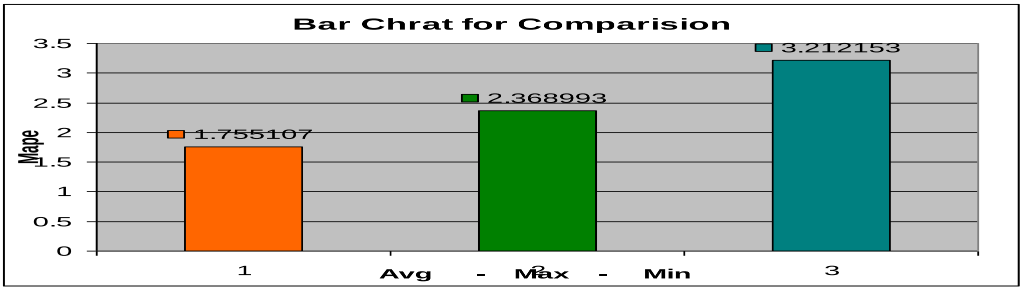


Figure2. Average Load Comparative between actual, ANN, Regression plot.

Figure 3. Mean Absolute Percentage Errors: The individual mean absolute percentage errors for all the days of test week are calculated separately. They are given in the table below:

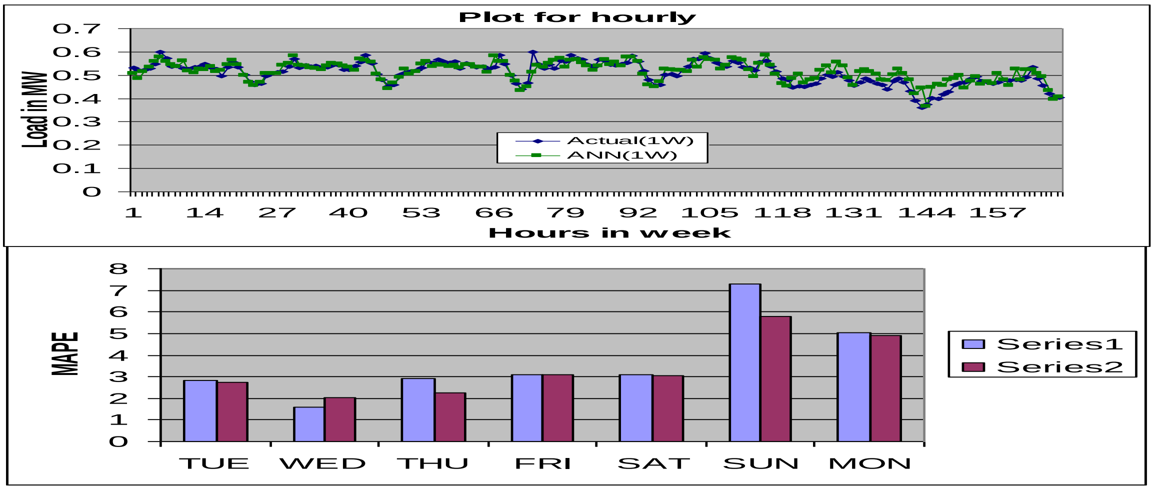


Figure4. In Bar Chart Plot of a week with one month train set

With 2 months training set:

Load data of the months November and December of 2001 are selected for two month training set. Now as the total numbers of days in the two months are 61, the total numbers of hours are 1464(i.e. 61 days*24 hours) for each hour a column will be generated. As we have opted for 13 inputs each column contains 13 elements. So the size of input vector will be 13×1464 and that of target vector will be 1×1464. Testing is done for January 1st, 2002 to January 7th, 2002 same week, on which the testing is performed with network using one month training set. So altogether testing is performed for 168 hours.

Figure5.Plot of a week with 2 months train set

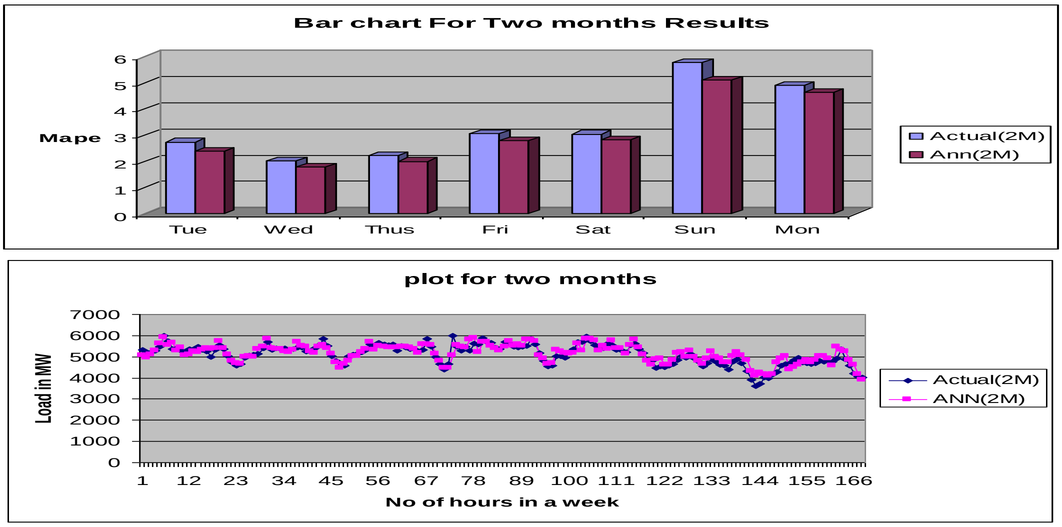
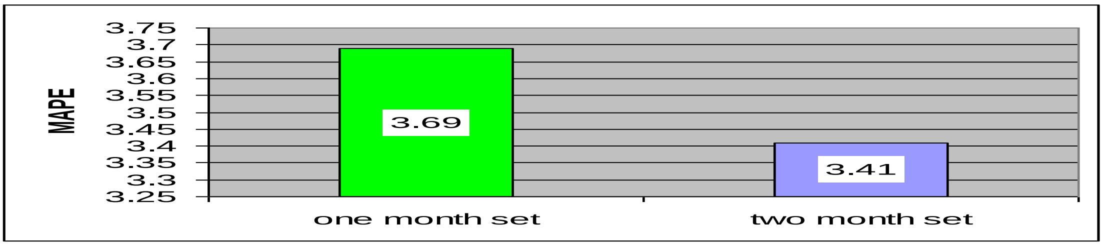


Figure 6. Bar chat comparison with one month and two months as training sets.



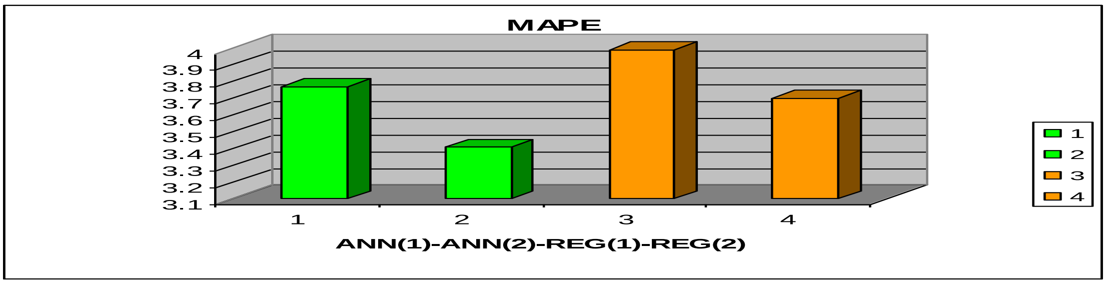


Figure 7. MAPE between ANN vs. Regression

CONCLUSIONS:

Several neural network models for short-term load forecasting were studied in this work. These techniques were divided into two classes: models for creating the forecast for one whole day at a time, and models utilizing the most recent load data and allowing the hourly forecasting. The focus was on discovering basic properties of different model types. This was considered necessary for the basis of a real forecasting application.

Hourly forecasting is perceived with average, peak, and valley loads. It was concluded that the forecasts for the average load are more accurate than those for the peak and valley loads. The problem with the daily forecasting models is that they don't allow updating during the day. If the forecast starts to go wrong, the models cannot make any changes on the basis of the most recent information, until at the end of the day. Therefore, if they were used in a real application, a method for making corrections in real time would be necessary. In this sense, the hour by hour models are more attractive. They allow hourly forecasting, and this clearly improves the accuracy for the closest hours.The idea to further improve the accuracy for the closest hours is endured by making the training set larger. It was concluded that training with two months gives in general better results than training with only one month.

Further improvement in the accuracy for the above said models can be achieved by including temperature data as input variable and also by finding some means to consider the effect, of anomalous load conditions, on forecasting during special holidays. As design of a problem in neural networks is basically a trail and error method, so several variations like changing the no. of hidden layers, hidden layer neurons, training algorithms, input variables, etc. are to be tried with. This should be done until optimum accuracy is reached, which makes these models suitable for real time energy management systems.

The test results were obtained using the load data of a particular case i.e. load dispatch centre located at Hyderabad. As this is only a single case, further evidence on other cases is desired. Without this, no conclusions on the generality and reliability of the model can be made. The next step related to this work is, therefore to implement the forecasting models and use it along with the current forecasting models on the data of several electric utilities. So it can be concluded that this project paves the way towards the process of creating a real time applicable neural network model.

Scope for Further Study:

1. The project stresses only the short term demand forecasting and the same approach can be extended to long term forecasting for predicting the values up to 5 years and above, for the purpose of power generation planning, system expansions etc.
2. Demand forecasting is carried out for state of Andhra Pradesh and also can be extended to all categories of consumption.

BIBLIOGRAPHY

1) A D Papalexopoulos, T Hasterberg. ‘A Regression Based Approach to Short-term System Load Forecast.’ IEEE Transactions on power systems, vol 5, no 4, November 1990, p1535.
2) S R Haung. ’Short-term Load Forecasting Using Threshold Autoregressive Models.’ IEEE Proceedings on Generation, Transmission and Distribution, vol 144, no 5, September1997, p 477.
3) M T Hagen and S M Behr.’The time series approach to time series load forecasting.’ IEEE Transactions, PRWS-2, vol 3, 1987, p 785.
4) K L S Sharma and A K Mahalnbis. ’Recursive short term load forecasting algorithm.’ IEE Proceedings, vol 121, no 1, January 1974, p 59.
5) W R C hristiaanse. ‘Short term load forecasting using general exponential smoothing’, IEEE Transactions on Power Apparatus System, PAS-3, 1988, p 900.
6) G Gross and F D Galiana.’ Short term load forecasting’ IEEE Proceedings, vol 75, no 12, December 1987, p 1558.
7) D. Singh, etal,” ANN Based Short Term Load Forecaster”, Journal of IE (I), Vol 82, sep 2001, p.p 161 – 166.
9) J.K.Mandal, et. al, “Application of ANN for Short Term Load Forecasting in Electrical Power Systems”, Journal of IE(I), vol.77, May1966, pp 40 – 43.
10) Dave Anderson and George Mc Neill (Web support) “Artificial Neural Networks Technology.”
12) Neural Network Toolbox for use with MATLAB from MATHWORKS.COM
13) Load data has been taken from LOAD DISPATCH CENTER at APGENCO, Hyderabad.
14) One Hour Ahead Load Forecasting using Neural Networks – IEEE Trans, Vol.17, Feb’02
15) Electric Power Systems Load Forecasting – IEEE DEE, FEIS, Brazil
16) Use of ANN for Electrical Load Forecasting – Journal of the Institution of Engineers
17) Load forecasting by Dr. C.Lakshminarayana
18) Electric utility transmission systems, engineering staff college of India
19) Dr. R.L.Sullivan ”Power system planning” Power System Planning,McGraw-Hill Inc., US, First (1st) Printing
20) Jacek .M. Zurada, ”Introduction to Artificial Neural Systems”, Jaico Publishing House
21) B.Yegnanarayana, ”Artificial Neural Networks”, Yegnanarayana B, PHI
22) Pabla, A. S.”Electric Power Distribution”, 6th Edition, Tata McGraw-Hill Education, 2011.
23) Turan Gonen” Electric Power Distribution Systems”, CRC Press, 2006.
24) Dr.M.K.Khedkar,Dr. G.M. Dhole,“A Textbook of Electric Power Distribution Automation”, Lakshmi Publications, Ltd., 2010

Author’s detail:

1) **Mr. N.M.G.KUMAR** Currently pursuing P.hD at SVU College of engineering at Tirupati, AP, India and Obtained his B.E in Electrical and Electronics Engineering from Bangalore University at S.M.V.I.T.S., Bangalore. Obtained M.Tech (PSOC) at S.V.U .college Engineering, Tirupati. Area of interest are power system planning, power system optimizations, power system reliability studies, Real time application of power system and like non-linear controllers applications to power systems.

2) **Dr.P.SANGAMESWARA RAJU** is presently working as professor in S.V.U. college engineering, Tirupati. Obtained his diploma and B.Tech in Electrical Engineering, M.Tech in power system operation and control and PhD in S.V.University, tirupati. His areas of interest are power system operation, planning and application of fuzzy logic to power system, application of power system like non-linear controllers.

3) **P.VENKATESH** Currently working as Assistant Professor in Sri Vidyanikethan engineering college, tirupati. Obtained his B.Tech in Electrical and Electronics Engineering from JNTU Hyderabad University at S.V.P.C.E, T. Putter. and Obtained his M.Tech in Electrical Power System from JNTU Anantapur University at Sri Vidyanikethan Engineering College, Tirupati. Areas of interest are power system analysis, application of FACTS devices using in Transmission systems.