Journal of Engineering and Applied Sciences 3 (2): 149-154, 2008
ISSN: 1816-949X

Electricity Load Forecasting Using Artificial Neural Networks

A.O. Afolabi, B.O. Olatunji and A.O. Ajayi
Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso Nigeria

Abstract: Load forecasting is an essential part of an efficient power system planning and operation. This research work is on short term electricity load forecasting using Artificial Neural Network (ANN) and Ogbomoso a city in Nigeria is considered as a case study. Input variables considered are past loads history, hours of the day and days of the week, while the output is the forecasted load for 24 h ahead. The training tool Neurosolution was employed in simulating and designing the feed forward back propagation forecasting network. Result obtained shows that electricity load can be predicted ahead of time also, it can also be inferred from this research that, load forecast using neural network is somewhat intelligent in that it gives real values even when the past load history is of zero value. This shows that areas without constant supply of electricity can still forecast future loads with a reasonable error margin so as to help in better load distribution and effective load shedding planning.

Key words: LTLF, MTLF, SLTF, load forcasting, neural networks

INTRODUCTION

Over the years optimal planning of power generation with minimal loss and costs has been a major concern in power distribution. Consequently, load forecasting has become an essential prerequisite for efficient power system planning and operations. The objectives of the study is essentially o be able to forecast amount of power of electricity needed for better load distribution in areas within Ogbomoso city, Oyo State in Nigeria Typically, load forecasting can be classified into three:

Long Term Load Forecasting (LTLF): Primarily for system development planning, it usually covers a period of years.

Medium Term Load Forecasting (MTLF): Mainly for maintenance and scheduling programs.

Short Term Load Forecasting (SLTF): Primarily for day to day operation of controlling and scheduling of power system.

The neural network approach helps reduce the problem associated with the conventional method, in that it has approximation ability for non-linear mapping and generalization. Neural network has the advantages of learning directly from historical data (Avelino, 1993). The neural network uses data such as past load history, calendar information (weekday, weekend, holiday, season etc) and weather information (constant temperature, average temperature, peak temperature, wind speed etc). In the past 10 years, neural network approach is the most frequently used approach. Different kind of neural networks; feed forward net (perceptron, back propagation network, radial basis function network) recurrent network (e.g., opfield), competitive network (e.g., self-organization map) have been applied to this problem, some of them are hybrid with genetic or fuzzy method.

The focus of this project is to apply Artificial Neural Network (ANN) to forecast electricity load demand of Ogbomoso town in Oyo State, Nigeria on a shot-term (daily) basis considering factors like past load history, weather and calendar information (days of the week, holidays). However, holidays and weekend days are being treated as the same.

Load forecasting: Most forecasting techniques used today are based upon traditional linear or nonlinear statistical models (Raymond and Ballard, 2002). However, despite the usefulness of these models as a forecast tool, they are limited in their ability to forecast in certain situations. Considering the changing nature of technology and the quest for optimal planning and distribution of electric power, it is becoming increasingly important to be able to easily and accurately predict trend and patterns in data. More so, it is now very essential

Corresponding Author: A.O. Afolabi, Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Nigeria

for forecasting models to be able to detect nonlinear relationships with consideration for high levels of noisy data and chaotic inputs.

Having realized the deficiencies of the aforementioned forecasting techniques, there is a great need for an advanced and improved forecasting method. One specific forecasting method being focused upon today utilizes a system that mirrors the organization and structure of the human nervous system (Daniel, 2002). This system consists of multiple processing elements aligned to operate in parallel to process information. This forecasting toll is refereed to as the artificial neural network.

Artificial neural networks: Artificial Neural Networks (ANN) is recognized as a massively parallel distributed processor that has a natural tendency for storing experiential knowledge and making it available for use (Daniel Klerfors, 2002).

Neural network is one of the most rapidly expanding areas of current research, attracting people from a wide variety of disciplines. It is one of the research fields in evolutionary computing or computational intelligence. The original idea of neural networking as a more flexible and accurate method of forecasting data is not a recent development. It began in the mid 1940 = s, but not until during the 1980 = s, after the breaking of some theoretical barriers and the growth in available computing power; did these networks become widely accepted as useful tools. There are many definitions of Artificial Neural Networks, based or different view of individual scholars. These include.

The artificial neuron: Like the human brain, the basic unit of neural networks is the artificial neuron. The neurons have connections between them and are able to simulate the four basic functions of neural neurons (Bearle and Jackson, 1992). The Fig. 1 shows the basics of an artificial neuron.

It is obvious that, artificial neuron looks similar to a biological neural cell and it functions the same way. The various inputs to the network are represented by the mathematical symbol, X (n) and each of these inputs are multiplied by a connection weight. These weights are represented by W (n)., the products are summed, fed through a transfer function to generate a result and then an output. In a neural net, the neurons are grouped in layers, called neuron layers (Purnima and Swen, 1997). Usually each neuron of one layer is connected to all neurons of the preceding and the following layer (except the input layer and the output layer of the net). The information given in a neural net is propagated layer-by

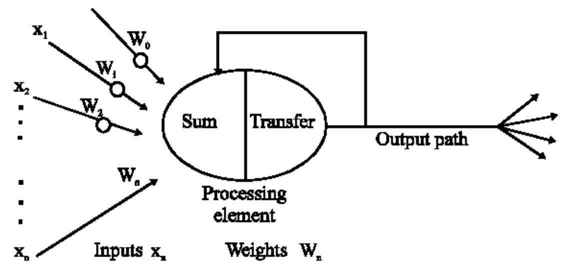

Fig. 1: Structure of an artificial neuron

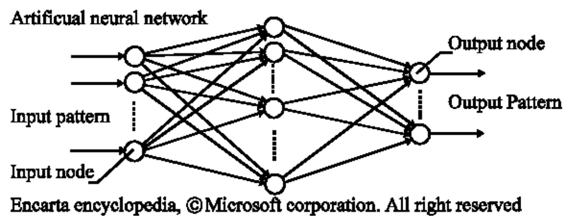

Fig. 2: A typical neuron network with one hidden layer

layer from input layer to output layer through, either, none, one or more hidden layers (Senaike *et al*., 1993). Depending on the learning algorithm, it is also possible that data is propagated backwards through the net.

Figure 2 shows a simple structure with only one hidden layer. To determine the number of hidden neurons the network should have to perform its best, one are often left out to the method of trial and error via training. If the number of hidden neurons is increased too much, there will be an over fit, that is the net will have problem to generalize (Lee *et al*., 1992). The training set of data will be memorized making the network useless on new data sets.

Mathematical neural model: If there are n inputs (X_1, X2,...Xn) to a neuron and there are n associated weights (W_1, W_2, YW_n) on the input lines. The model neuron calculates the weighted sum of inputs, that is (Raymond, 2003).

$$\text{Total input} = W_1 X_1, + Y.. + W_n X_n \qquad 2.1$$

In a compact form,

$$\text{Total input} = \Sigma W_i X_i \qquad 2.2$$

The sum (total input above) is then compared with a certain value in neuron, the threshold value. If the sum of the input exceeds the thresholds, the neuron outputs a 1, otherwise it output a O. That is,

$$Y = fh\left[\sum_{i=1}^{n} Wi \quad Xi\right] \qquad 2.3$$

Where, Y is the output and /h is a step function or the activation function.

$$fh(x) = O \begin{cases} 1 & \text{if} \quad x > \quad \Theta \\ 0 & \text{if} \quad x \geq \quad \Theta \end{cases} \qquad 2.4$$

where, Θ is the threshold value

Alternatively, the threshold value can be subtracted from the weighted sum and the resulting value compared to output the result is positive, then it outputs a 1, else it output is O. Thus, we have

$$Y = fh\left[\Sigma\, Wi\, Xi - \Theta\right] \qquad 2.5$$

$$fh(x) = O \begin{Bmatrix} 1 & \text{if} \quad x > \quad \Theta \\ 0 & \text{if} \quad x \leq \quad \Theta \end{Bmatrix} \qquad 2.6$$

Neural network design: Designing neural networks is complex and of major concern to developers. It is an iterative process that involves making a number of design decisions (Leslie, 2000). However, design decisions are mainly in the areas of

- Connections and communication: Deciding the types of connections among neurons, both within and among the layers.
- Learning: Deciding the appropriate learning method and algorithm for the network. In other words, deciding how a neuron receives input and produces output.

Learning: One important feature of neural networks is its ability to learn the solution to a problem (Leslie, 2000). The network stores and learns by changing of the connection weights. The strength of connection between the neurons has a weight value for the specific contention.

Learning method: The learning ability of a neural network depends on its architecture and the learning method adopted.

Back propagation: It is a form of supervised learning. This method is proven highly successful in training of multi-layered neural nets (Shuja and Qurban, 2007). The network is not just given reinforcement, but information about errors is also filtered back through the network and is used to adjust the connections between the layers, thus improving performance.

MATERIALS AND METHODS

Data collection and analysis: The hourly electricity load readings of NEPA/33/11kv Injection Sub-Station, Ilorin Road, Ogbomosho, Oyo State, Nigeria were used as input data for the network training. The entire city of Ogbomosho is supplied by three main electricity Feeders, namely, the Takie Feeder, Ilorin road Feeder and Oke-Ado Feeder. The hourly readings from these Feeders are summed up to have the total electricity load supply of the entire city at any given hour. Data used spanned through the 1st of April 2004 and 31st of May 2004. However, owing to the epileptic nature of the Nigerian power supply there are quite a number of black outs and these are denoted in the readings with 0 voltage.

The feed forward backpropagation network model was used. The model has eight inputs and one desired output referred to as the target. However, the model structure is of two layers.

Forecasting procedure: These are the major steps to be followed for an effective forecasting process. Basically, these include the following:

Input variable selection: Input variables such as past electricity load, temperature, day type are selected.

Data processing: Improperly recorded data and other observation error are inevitable. Hence, bad and abnormal data are identified and discarded or adjusted.

Scaling: Considering the difference in ranges of the input variables, the network data have to be scaled so as to avoid convergence problem. In this project, we scaled our load inputs by dividing through with the highest load reading (680 V) and then represent the days of the week with digits.

Network training: In this project work, the Neural Network training tool Neuro Solution was used for the short term load forecasting.

RESULTS AND DISCUSSION

Implementation of ANN for electricity load forecasting: Simulation of feed forwardbackpropagation using neurosolution: By using the NeuroSolutions to train the neural network, the first eight columns are specified for use as inputs, by selecting the columns and then Tag data|column as Input from the neurosolutions menu

Table 1: Training report table showing the output and the target output

24 h	12 h	6 h	L 2 h	1 h	Day	Temp	h	Tag	Tag ouput	Tag in kv	Tag output in kv
0.4098	0.3607	0.4098	0.0164	0.0328	6	27.8	1	0.0328	0.258079	223.041	175.49
0.4098	0.3607	0.4262	0.0328	0.0328	7	28.7	2	0	0.216573	0	147.27
0.4098	0.3607	0.4098	0.0328	0	1	29	3	0	0.207870	0	141.35
0.4098	0.6721	0.3771	0	0	2	28.4	4	0	0.306505	0	208.42
0.4262	0.3443	0.0164	0	0	3	26.7	5	0	0.260543	0	177.16
0.4262	0.3115	0.0328	0	0	4	26.3	6	0	0.265221	0	180.350
0.459	0.4098	0.0328	0	0	5	24.6	7	0	0.271012	0	184.288
0.7377	0.4262	0	0	0	6	24.8	8	0	0.222024	0	150.977
0.6885	0.4098	0	0	0	7	25.5	9	0	0.194320	0	132.137
0.6393	0.3771	0	0	0	1	26	10	0	0.268569	0	182.627
0.6721	0.0164	0	0	0	2	26.2	11	0	0.335503	0	228.142
0.7049	0.0328	0	0	0	3	25.3	12	0	0.324528	0	220.679
0.3607	0.0328	0	0	0	4	27	13	0	0.342603	0	232.970
0.3607	0	0	0	0	5	29.3	14	0	0.353474	0	240.362
0.3607	0	0	0	0	6	29.1	15	0	0.285238	0	193.962
0.6721	0	0	0	0	7	27.3	16	0.3279	0.241255	222.972	164.053
0.3443	0	0	0	0.3279	1	26.3	17	0.3279	0.444473	222.972	302.241
0.3115	0	0	0.3279	0.3279	2	26.1	18	0.7213	0.552590	490.484	375.761
0.4098	0	0	0.3279	0.7213	3	24.6	19	0.4918	0.713435	334.424	485.136
0.4262	0	0	0.7213	0.4918	4	24.8	20	0.4918	0.497957	334.424	338.611
0.4098	0	0	0.4918	0.4918	5	26	21	0.541	0.560158	367.88	380.908
0.3771	0	0.3279	0.4918	0.541	6	26.6	22	0.5246	0.511772	356.728	348.005
0.0164	0	0.3279	0.541	0.5246	7	27.3	23	0.8361	0.526630	568.548	358.109
0	0	0.7213	0.5246	0.8361	1	27.9	24	0.7213	0.686818	490.484	467.036

Table 2: Table of the desired output against actual output

Performance	Tag
MSE	0.031990071
NMSE	0.664014054
MAE	0.139585935
Min. Abs. error	6.42613E-05
Max. Abs. error	0.514569175
R	0.579643281

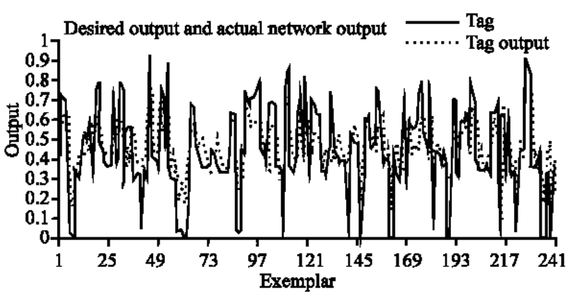

Fig. 3: Graph of the desired output against actual output

(located within the Microsoft Excel menu bar) and the last column to use as the desired outputs, the corresponding column is selected followed by Tag data|column as desired from the neurosolutions menu.

The rows of data must also be tagged before a training process can be run. Rows can be tagged as Training, Cross Validation, Testing, or Production. Only the Training tag is required for running a training process. However, cross validation is a very useful tool for preventing over-training; thus a portion of the data is tagged as Cross Validation. The rows of data needed for testing the trained network (Testing) or producing the network output for new data (Production) can be tagged before or after the training process is run. To tag rows of data, appropriate rows and the corresponding tagging operation from the NeuroSolutions menu are selected. The data available for creating the network was split into the training data and test data sections. The amount of data required for each was also decided before implementation of the ANN commenced. As a standard practice, about 75% of the data is allocated to training the network and 25% to testing/simulating the network.

Test result analysis for neurosolutions: Table 1 and various generated graphs showing the result of training done with the neurosolutions training kit. Just as in Matlab, the training was done with eight input variables and because it is a supervised learning it uses the expected output, tag for checking. The corresponding output of the training is listed in the Atag output column.

From Table 1 the first eight columns of the table contain data record from NEPA. The last 2 columns contain the outputs; the actual and desired outputs of the experiment, respectively.

Figure 3 shows all data in the network the graph is the desired output and actual outputs against all the training data. Table 2 reporting the mean-squared error (MSE) Normalized mean-squared error (NMSE), mean absolute error (MAE), minimum absolute error, maximum absolute error and correlation coefficient (r) for each output.

Figure 4 is a plot of the training mean-squared error (MSE) versus Epochs. The active worksheet contains cross validation input and cross validation desired tags as well as the cross validation MSE.

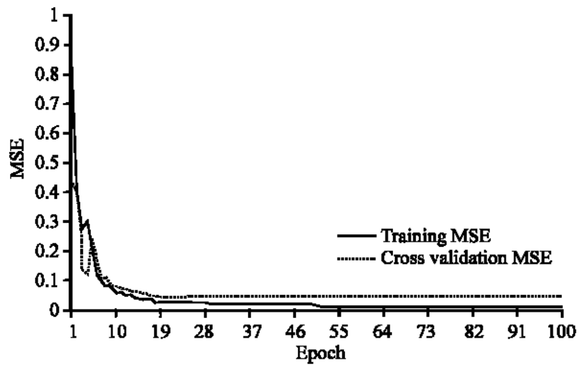

Fig 4: Graph of mean square error against Epoch

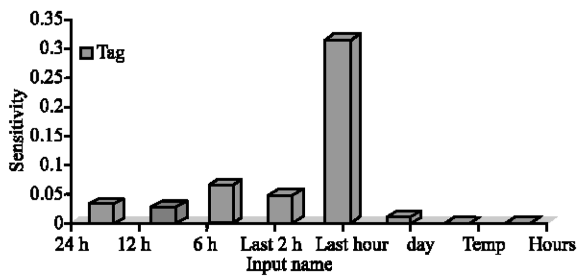

Fig 5: Sensitivity about the mean

Table 3: Table of mean square error against Epoch

Best networks	Training	Cross validation
Epoch #	100	24
Minimum MSE	0.057940189	0.054813582
Final MSE	0.057940189	0.056341805

Table 4: Table of sensitivity about the mean

Sensitivity	Tag
24 h	0.0346471
12 h	0.028888395
6 h	0.064714327
Last 6 h	0.048968423
Last hour	0.313539922
Days	0.010700372
Temperature (Temp)	0.001292306
Hour of the day (h)	0.00083611

The Table 3 shows the minimum training MSE, the epoch at which this minimum training MSE occurred and the final training MSE. This table includes the minimum cross validation MSE, the epoch at which this minimum cross validation error occurred and the final cross validation MSE.

This testing process provides a measure of the relative importance among the inputs of the load forecast and illustrates how the modeled electricity load outputs varies in response to variation of the variable data input. All inputs are varied between their mean load readings and the number of standard deviations when all other inputs are fixed at their respective means. This process is repeated for each input and a report is generated which summarizes the variation of each output with respect to the variation in each input. Table 4 and Fig. 5 show the result sensitivity about the mean.

CONCLUSION

In conclusion, one important factor to be considered is the sensitivity of load forecasting to the selected input variables. Of the eight input variables fed into the network, it was observed that past load history has the greatest influence on the overall forecast result, followed by the day of the week and then the hour of the day. The daily temperature input has the least influence on the overall output. This is largely because the 2 months (April and May) considered in training this network falls within the same season of the year, thus making the temperature readings of approximate values. Also, our case study is a developing city where power consumption is less sensitive to weather condition.

Also, it can also be inferred from this research that, load forecast using neural network is somewhat intelligent in that it gives real values even when the past load history is of zero value. This shows that areas without constant supply of electricity can still forecast future loads with a reasonable error margin. It was also observed that Neurosolution has a better documentation of the whole training process and allows for validation of the network during training so as to adjust the data for a better training.

Finally, using a feed forward backpropagation network; the short term electricity load of Ogbomosho was forecasted and the results obtained were completely satisfactory. The high values of the correlation coefficient (R-value) indicate that the network is fully trained and capable of forecasting the result with a good degree of precision.

REFERENCES

Avelino, J.G. and D.D. Dankel, 1993. The engineering of knowledge-based systems. Prentice-Hall Inc, pp: 34-45.

Bearle, R. and T. Jackson, 1992. Neural computing: An introduction. Instit. Phys. Publishing, Great Britain, pp: 56-53.

Daniel, K., 2001. Artificial neural networks. School of Business and Administration, Saint Louis University, pp: 78-87.

Data and Analysis Center for Software, Artificial Neural Technology, 1992. http://www.dacs.dtic.mil/techs/neural/neural.title.html

Lee, K.Y., Y.T. Cha and J.H. Park, 1992. Short-term load forecasting using an artificial neural network. Trans. Power Sys., 1 (7): 124-131.

Leslie, S., 2000. An introduction to neural networks, center for cognitive and computational neuroscience, university of stirling. http://www.cs.stir.ac.uk/~lss/NNIntro/Invslides.html.

Purnima, S. and M. Swen, 1997. Neural network: Is it needed for your business. India business school. http: //www.indiabschools.com/systems_004.htm.

Raymond, J.B., 2003. Forecasting with neural networks: A review http://www.nssa.us/nssajrnl/NSSJ2003%2020_2/pdf/03Ballard_Raymond.pdf.

Senaike, O.A., A.K. Ojo and A.B. Sofoluwe, 1997. Pattern recognition using single layer perceptron. J. Sci. Res. Dev., 3: 23-24.

Shuja, M. and A. Qurban, 2007. Crime investigation system using neural nets. Karachi institute of information technology. http: //www.techcom.org.pk/conference_files/spsr_cisnn.pdf.