SHORT TERM LOAD FORECASTING BY NEURAL NETWORK IN MASHAD (IRAN) POWER SYSTEM

Naser MAHDAVI TABATABAEI[1], Seyed Reza MORTEZAEEI[2], Navid TAGHIZADEGAN[3], Ahmet NAYIR[4]

[1)] Electrical Engineering Department, Azarbaijan University of Tarbiat Moallem, Tabriz, Iran
Electrical Engineering Department, Islamic Azad University - Tabriz Branch, Tabriz, Iran
Electrical Engineering Department, Seraj Higher Education Institute, Tabriz, Iran
n.m.tabatabaei@gmail.com
[2)] Elect. Eng. Dept., Azarbaijan University of Tarbiat Moallem, Tabriz, Iran, rmortezaeei@yahoo.com
[3)]Azarbaijan Higher Education And Research Complex (AHERC), Tabriz, Iran,
ntaghizadegan@yahoo.com
[4)] Electrical Engineering Department, Fatih University, Istanbul, Turkey, ahmetnayir@hotmail.com

Keywords: Short Term Load Forecasting, Weather-Electricity Demand Relation, Artificial Neural Network (ANN), Multi-Layer Perceptron (MLP)

Abstract: *The paper illustrates supervised neural networks for the electric energy demand forecasting of an area with a prediction time of 1 - 24 h. The actual forecast is obtained using a two layered feed forward neural network, trained with the back propagation of momentum learning algorithm. In order to investigate the influence of climate variability on the electricity consumption, the neural network is trained using weather data (temperature, relative humidity, global solar radiation) along with historical load data available for a part of the electric grid of the town of Mashad (Iran). The model validation is performed by comparing model predictions with load data that were not used for the network's training. The results obtained bear out the suitability of the adopted methodology for the short term load forecasting (STLF) problem also at so small a spatial scale as the suburban one.*

1. INTRODUCTION

Many studies have pointed out the overwhelming sensitivity of electricity consumption to weather variables, in particular focusing attention on the forecast limited to 24 hour ahead. The prediction of the system load over an interval ranging from 1 hour to 1 week is known as short term load forecasting (STLF).

Recent research activities have also focused on the impacts of climatic changes both on supply (studies about the potential impact of climate change on renewable energy resources, such as wind power [1,2] and hydroelectric power [3]) and demand (studies about electricity demand and natural gas demand correlation with weather variability [4,5]) for energy.

Weather sensitivity has also been examined in order to correlate electricity consumptions to the increases in market saturation of air conditioning induced by long term climatic changes [6].

A very common approach in STLF problems is constituted of algorithms based on artificial neural networks (ANNs). Most of the suggested models for STLF use neural network architectures known as multi-layer perceptrons (MLPs) [7–13, 15].

Some of the proposed approaches present modular solutions in different networks are suitably arranged to provide the forecast [12, 13].

In addition to MLPs for load forecasting, there are also other models based on unsupervised learning, using self organizing maps (SOMs) [14, 16-18]. The appeal of neural networks can be explained by the ability of the network to learn complex relationships between input and output patterns that would be difficult to model with conventional algorithms.

In particular, referring to STLF, the results of application method to Mashad (Iran) power system shows that ANNs are able to learn the properties of the electric load that would otherwise require deep and careful analysis to be discovered. In neural network models, past load and weather data generally represent the network inputs, and forecasted load values represent the outputs. The networks training is conducted by providing input-output pairs extracted from historical data.

2. THE MULTI-LAYER PERCEPTRON MODELS

MLPs are constituted of a set of interconnected basic processing units (neurons) organized in layers. Each neuron produces its output by taking a linear combination of the input signals and transforming it using a function called the activity function. The weights of this linear combination are those associated with the numeric connections (synaptic weights) linking the neuron with all of the neurons belonging to the upper layer.

The output of a neuron as a function of the input signals can thus be written

$$y_j = f\left(\sum_{i=1}^{n} w_{ji} x_i - b\right) \quad (1)$$

where y_j is the output of the generic neuron belonging to layer *j*; x_i par are the input signals to the neuron; w_{ji} is the synaptic weight associated with the connection between the generic neurons belonging to layers *j* and *i*, respectively; *b* is the bias term (another neuron weight); *f* is the activity function.

A MLP can also be composed of several layers as Fig.1, but the most often employed architecture consists of three layers: an input layer, one hidden layer and an output layer [7,8,16].

First the input patterns of each node are provided at the input layer; then this signal is converted at each node and transferred forward until the output layer, which generates the network's output. At each step the error is calculated by comparing the network's output with the known target.

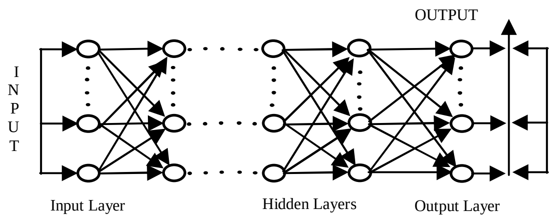

Fig. 1. Schematic representation of a generic MLP

In a fully connected MLP, each neuron of a layer is connected to each neuron of the next layer. There are no feedback connections.

Learning is the most important phase in utilizing an ANN based model. During learning, sets of known input-output patterns are presented to the network, and the connection weights among processing units are adjusted according to the imposed learning rules. The most widely used weight updating algorithm for the training of MLP networks is the so called error back propagation (EBP).

The basic idea of the back propagation learning algorithm consists of the repeated application of the rule for computing the influence of each weight in the network with respect to an arbitrary error function

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} \quad (2)$$

where w_{ji} is the weight associated with the connection from neuron *i* to neuron *j*, o_j is the output and net_j is the weighted sum of inputs for neuron *j*.

Once the partial derivative for each weight is known, the aim of minimizing the error function is achieved by performing a simple gradient descent, according to the updating rule

$$w_{ji}(t+1) = w_{ji}(t) - \eta \frac{\partial E}{\partial w_{ji}}(t) \quad (3)$$

where η is the learning rate, a term that reduces the derivative and has an important effect on the time needed till convergence is reached. If η is too small, the convergence requires too many steps to reach an acceptable solution. On the other hand, a large learning rate could lead to oscillations, preventing the error from falling under a fixed threshold value. The value of $E(t)$ is related to the difference, at time of iteration *t*, between the output and target values.

3. MLP LEARNING PROCEDURE

The learning algorithm utilized for the MLP's training is the adaptive back propagation with momentum. It constitutes a little variation of the basic back propagation algorithm. In order to prevent oscillations of the learning algorithm around the solution, a momentum term was introduced into the basic updating related rule, obtaining the following equation

$$w_{ji}(t+1) = w_{ji}(t) - \eta \frac{\partial E}{\partial w_{ji}}(t) + \alpha \Delta w_{ji}(t-1) \quad (4)$$

where the momentum parameter reduces the influence of the previous step on the current. This additional term makes the learning procedure more stable and accelerates the convergence process in the shallow regions of the error surface.

Furthermore, seeing that it is not possible to determine the optimal learning rate value before the beginning of the training phase (because its value changes during the training process as the algorithm moves across the error surface), the adaptive learning rate technique attempts to maintain the learning step size Δw_{ji} as large as possible while keeping the learning stable.

An adaptive learning rate requires some changes in the basic training procedure. At each training step, new weights and biases are updated using the current learning rate, so outputs and errors are calculated. If the new error exceeds the error calculated in the previous step by more than a predefined value (maximum performance increase), the new weights and biases are discarded, and the learning rate is decreased by multiplying by a learning rate decay factor. Otherwise, the new weights and biases are kept, and the learning rate is increased by multiplying by a learning rate increasing factor. The initial learning rate utilized was $\eta = 0.01$, and the momentum term was set to $\alpha = 0.85$.

Learning rate decay and increase factors were chosen equal to 0.8 and 1.1, respectively. Finally, the maximum performance increase was set to 5%.

4. ANN ARCHITECTURE

The model implemented consists of a totally connected two layer network (input, hidden and output layers) with the following structural characteristics. 136 inputs, namely

• hourly load values of the day preceding the forecast day ($y(i-1),\ t_j$), with $j = 1, \cdots, 24$;

• 24 hourly load values of the day preceding the foregoing ($y(i-2),\ t_j$), with $j = 1, \cdots, 24$;

• 9 elements vector flagging the day preceding the forecast day and identifying the cluster its load curve belongs to (e.g. 100000000 cluster no.1; 010000000 cluster no. 2; etc.);

• 24 mean hourly dry bulb temperatures of the day preceding the forecast day ($T(i-1),\ t_j$), $j = 1, \cdots, 24$;

• 24 mean hourly global solar radiations of the day preceding the forecast day ($R(i-1),\ t_j$), $j = 1, \cdots, 24$;

• 24 mean hourly air relative humidity's of the day preceding the forecast day ($U(i-1),\ t_j$), $j = 1, \cdots, 24$;

• 7 elements vector indicating the day preceding the forecast day (e.g. 1000000 Sunday and 0000001 Saturday).

The number of outputs was set to 1, representing the simultaneous load forecast for the day concerned.

All components of the data vectors presented to the network (except for the vectors of size 9 and 7) were normalized with the same technique utilized for the load data in the preprocessing (Normalization of load data vectors).

Before splitting the load data into different clusters using a SOM, they were normalized using the technique described by the following equations

$$x' = \frac{x - \bar{x}}{\mathrm{std}(x)}$$
$$x_{norm} = \frac{1}{1+e^{-x'}} \quad , \quad f(x) = \frac{1}{1+e^{-x}} \quad (5)$$

where x is the value of the generic component of the data vectors before normalization; $\bar{x}$ and $\mathrm{std}(x)$ are, respectively, the mean and standard deviation of the same component; x_{norm} is the component value after normalization.

Phase in order to prevent the neurons activity function from being driven into saturation, causing convergence problems for the learning algorithm.

The activity function utilized for the neurons of the hidden layer was the logistic sigmoid function that is described by the following equations and depicted in Figure 2.

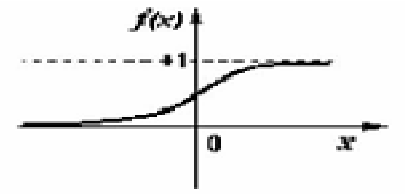

Fig. 2. Shape of logistic sigmoid function

For the neurons of the output layer, the linear transfer function $f(x) = x$ was utilized. Obviously, before comparing the network outputs with the actual electric loads of the period they are referred to, the first were renormalized. A simplified expression of the realized model can be summarized as follows

$$y_i = F[W_i, y(i-1), y(i-2), T(i-1), R(i-1), U(i-1), cc, dc] \quad (6)$$

Where W_i represents the generic synaptic weight (or bias value) of the net, *cc* the cluster code of the data vectors (assigned to them during the preprocessing phase) and *dc* the code identifying the day of forecast.

Figure 3 reports a schematic representation of the architecture of the MLP as implemented. The MLP was trained with the load data vectors labeled in the preprocessing phase and with the vectors identifying the day of forecast. In order to investigate the influence of climatic variables on the electricity demand, also the correspondent temperature, global solar radiation and humidity data vectors were utilized for the network's training.

In fact, although any ANN based forecasting model, because of its "black box" nature, is able to provide a numerical relation between weather variables and electric energy consumption, it is well known that the utilization of significant weather variables has a positive effect on the forecasting results achieved using a neural architecture [7, 8].

The whole available data set was subdivided into a training set (from 80/02/01 to 83/12/29 Hijri Shamsi) and a test set (from 84/01/01 to 84/12/29 H.S.). The training set was used for computing the gradient and updating the network weights and biases. The test set was used as a reference set to quantify the general performance and the error calculated on its basis. The test error normally decreases during the initial phase of training in step with the training set error, but if the learning phase is per formed for a too large number of epochs, it happens, usually in the later stages of learning, that the test error starts to increase, even though the training error continues to decrease.

When the error on the test set begins to rise, the network begins to over fit the data. This means that the network simply memorizes the training patterns, thus having good performances during training but failing when presented with inputs slightly different from that utilized in the learning phase.

Table 1 reports the values of the errors for each hour t_i of the forecasting day considered

$$\varepsilon(t_i) = 100 * \left| \frac{\hat{y}(t_i) - y(t_i)}{y(t_i)} \right| \quad (7)$$
$$e(t_i) = \hat{y}(t_i) - y(t_i)$$

Where $\hat{y}(t_i)$ indicates the forecasted load at hour t_i; $y(t_i)$ indicates the actual load at the same hour. The results obtained are quite acceptable, considering that the spatial scale we worked with in this case study is very small with respect to the regional or national scale generally adopted for similar applications [15-17].

This fact implies that to equal values of absolute error correspond higher values of relative error. Neural networks reveal, therefore, a useful instrument in tackling STLF problems also at the suburban scale and could became a precious decision supporting tool in energy planning, especially for periods (as the summer one, in particular for areas with Mediterranean climate) in which the influence of weather conditions on electric consumptions is certainly overwhelming, but very difficult to evaluate precisely.

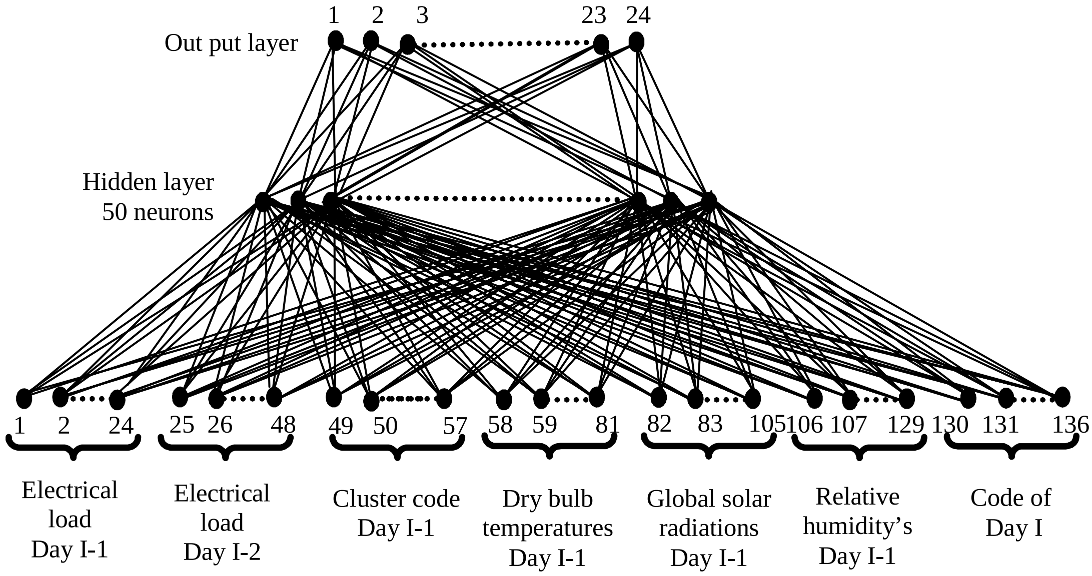

Fig. 3. The MLP structure

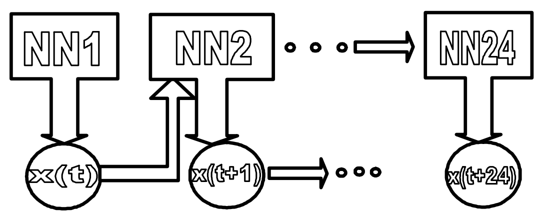

Fig. 4. Proposed modular architecture for the STLF problem

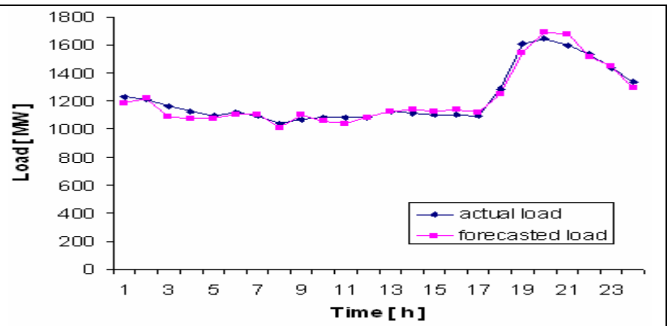

Fig. 5. Comparisons between actual and forecasted loads for 1384/ 01 /01 H.S.

Table 1. Load forecast results for 84 /01 /01 H.S.

	H1	***H2***	***H3***	***H4***	***H5***	***H6***
Real load	1228.72	1211.33	1162.78	1126.08	1093.5	1116.63
Forecasted load	1190.48	1218.52	1091.06	1074.12	1075.53	1109.8
Error	38.24	-7.19	71.72	51.96	17.97	6.83
% Error	3%	0.6%	6%	4.5%	1.7%	0.5%

	H7	***H8***	***H9***	***H10***	***H11***	***H12***
Real load	1096.14	1040.84	1069.48	1081.42	1082.41	1080.31
Forecasted load	1104.89	1006.83	1103.56	1056.68	1039.43	1079.6
Error	-8.75	-34.01	-34.08	24.74	42.98	0.71
% Error	0.57%	2.5%	2.5%	2.1%	4%	0%

	H13	***H14***	***H15***	***H16***	***H17***	***H18***
Real load	1123.09	1110.99	1103.41	1099.98	1093.83	1283.72
Forecasted load	1128.22	1137.74	1124.8	1137.9	1121.33	1250.19
Error	-5.13	-26.75	-21.39	-37.92	-27.5	33.53
% Error	0.05%	2.1%	2%	3%	2.5%	2.5%

	H19	***H20***	***H21***	***H22***	***H23***	***H24***
Real load	1606.09	1642.6	1594.95	1534.53	1436.97	1338.98
Forecasted load	1549.41	1690.46	1674.78	1513.31	1446.92	1294.72
Error	56.68	-47.86	-79.83	21.22	-9.95	44.26
% Error	3.5%	3%	5%	1.4%	0.67%	3.3%

5. CONCLUSIONS

The paper describes an application of artificial neural networks to the forecasting of the daily electric load profiles of a suburban area. The studied model consists of a multi-layer perceptron (MLP) with only one hidden layer, taking as inputs load and weather data.

To distinguish among different types of load profiles, the load data vectors of the training set were first subdivided into clusters using an unsupervised neural network. This is a useful operation, especially when the training set is constituted by widely non-homogeneous data.

In spite of the small number of available training patterns, the proposed model gave fine results; bearing out the suitability of ANN based models for urban energy planning problems. Accuracy improvement will certainly be obtained by utilizing a larger data library and making use of other important meteorological and climatic variables (like wind speed).

Moreover, combined with information related to the market penetration of splits, an opportune neural model could be realized for studying the correlation between electric consumptions of the same suburban area and the use of heating, ventilating and air conditioning (HVAC) systems. To this aim, a wide research activity is already in progress.

REFERENCES

[1] P. BRESLOW and D.J. SAILOR, "*Vulnerability of Wind Power Resources to Climate Change in the Continental United States*", Renew Energy, No. 4, Vol. 27, 2001, Pages 585-598.

[2] D.J. SAILOR, T. HU, X. LI and J.N. ROSEN, "*A Neural Network Approach to Local Downscaling of GCM Wind Statistics for Assessment of Wind Power Implications of Climate Variability and Climatic Change*", Renew Energy, Vol. 19, 2000, Pages 359-78.

[3] J.R. MUNOZ and D.J. SAILOR, "*A Modeling Methodology for Assessing the Impact of Climate Variability and Climatic Change on Hydroelectric Generation*", Energy Converse Manage, Vol. 39, 1998, Pages 1459-69.

[4] D.J. SAILOR and J.R. MUNOZ, "*Sensitivity of Electricity and Natural Gas Consumption to Climate in the US - Methodology and Results for Eight States*", The International Journal of Energy, No. 10, Vol. 22, 1997.

[5] D.J. SAILOR, "*Relating Residential and Commercial Sector Electricity Loads to Climate Evaluating State Level Sensitivities and Vulnerabilities*", The International Journal of Energy, Vol. 26, 2001, Pages 645-657.

[6] D.J. SAILOR and A. PAVLOVA, "*Air Conditioning Market Saturation and Long Term Response of Residential Cooling Energy Demand to Climate Change*, The International Journal of Energy, No. 9, Vol. 28, 2001, Pages 941-951.

[7] D.C. PARK, M.A. EL-SHARKAWI, R.J. MARKS II, L.E. ATLAS and M.J. DAMBORG, "*Electric Load Forecasting Using an Artificial Neural Network*", IEEE Transactions on Power System, No. 2, Vol. 6, 1991, Pages 442-449.

[8] K.Y. LEE, Y.T. CHA and J.H. PARK, "*Short-Term Load Forecasting Using an Artificial Neural Network*", IEEE Transactions on Power System, No. 1, Vol. 7, 1992, Pages 124-132.

[9] K.L. HO, Y.Y. HSU and C.C. YANG, "*Short-Term Load Forecasting Using a Multilayer Neural Network with an Adaptive Learning Algorithm*", IEEE Transactions on Power System, No. 1, Vol. 7, 1992, Pages 141-149.

[10] S.T. CHEN, D.C. YU and A.R. MOGHADDAMJO, "*Weather Sensitive Short-Term Load Forecasting Using Nonfully Connected Artificial Neural Network*", IEEE Transactions on Power System, No. 3, Vol. 7, 1992, Pages 1098-1105.

[11] C.N. LU, H.T. WU and S. VEMURI, "*Neural Network Based Short-Term Load Forecasting*", IEEE Transactions on Power System, No. 1, Vol. 8, 1993, Pages 336-342.

[12] M. SFORNA and F. PROVERBIO, "A *Neural Network Operator Oriented Short-Term and Online Load Forecasting Environment*", Electric

Power System Research, Vol. 33, 1995, Pages 139–149.

[13] A. KHOTANZAD, R.C. HWANG, A. ABAYE and D. MARATUKULAM, "*An Adaptive Modular Artificial Neural Network Hourly Load Forecaster and its Implementation at Electric Utilities*", IEEE Transactions on Power System, No. 3, Vol. 10, 1995, Pages 1716–1722.

[14] Y.Y. HSU and C.C. YANG, "*Design of Artificial Neural Networks for Short-Term Load Forecasting, Part I: Self-Organising Feature Maps for Day Type Selection*", IEE Proceedings-C, No. 5, Vol. 138, 1991, Pages 407-413.

[15] Y.Y. HSU and C.C. YANG, "*Design of Artificial Neural Networks for Short-Term Load Forecasting, Part II: Multilayer Feedforward Networks for Peak Load and Valley Load Forecasting*", IEE Proceedings-C, No. 5, Vol. 138, 1991, Pages 414-418.

[16] M. DJUKANOVIC, B. BABIC, D.J. SOBAJIC and Y.H. PAO, "*Unsupervised/Supervised Learning Concept for 24-Hour Load Forecasting*", IEE Proceedings-C, No. 4, Vol. 140, 1993, Pages 311-318.

[17] R. LAMEDICA, A. PRUDENZI, M. SFORNA, M. CACIOTTA and V. ORSOLINI CENCELLI, "*A Neural Network Based Technique for Short Term Forecasting of Anomalous Load Periods*", IEEE Transactions on Power System, No. 4, Vol. 11, 1996, Pages 1749-1756.

[18] M. HAYATI and Y. SHIRVANI, "*Artificial Neural Network Approach for Short Term Load Forecasting for Ilam Region*", International Journal of Electrical, Computing and System Engineering, No. 2, Vol. 1, 2007, Pages 121-125.