Electrical System Load Forecasting with Polynomial Neural Networks
(based on Combinatorial Algorithm)

Huseynov A.F.
Head of ICT and innovations division
Azerbaijan State Economic University
Baku, Azerbaijan
aydinhuseyn1945@gmail.com

Yusifbeyli N.A.,
Director
SR and DP Power Engineering Institute
Baku, Azerbaijan
yusifbeyli.n@azerenerji.com

Hashimov A.M.
First Vice-President
Azerbaijan National Academy of Science
arif@physics.ab.az

***Abstract*— A polynomial neural network model for short term electrical load forecasting (STLF) is developed. Several models use past weekly and monthly system loads to forecast future electrical demands. All models are validated with actual system load data from the Azerbaijani Power Company. Combinatorial algorithm is elaborated to find efficiently the coefficients of regression type model. The paper presents the results, conclusions and points out some directions for future work.**

Keywords - short-term load forecasting (STLF), time series, polynomial neural model, combinatorial algorithm.

I. Introduction

Electrical load forecasting plays a central role in the operation and planning of electric power. The countrywide energy estimation, the planning of new plant, the routine maintaining and scheduling of daily electrical generation are all depended on accurate load forecasting in the future [1].

Accurate models for electric power load forecasting are essential to the operation and planning of an utility company. Load forecasts are extremely important for energy suppliers, ISOs, financial institutions, and other participants in electric energy generation, transmission, distribution, and markets.
Most important factors in load forecasting includes past load history, calendar information (weekday, weekend, holiday, season, etc.) and weather information (instant temperature, average temperature, peak temperature, wind speed, etc.), economical and social events.

Due to different aim of forecasting, the load-forecasting problem can be classified into some kinds.
Spatial forecasting is mainly about forecasting future load distribution in a special region, such as a country region, a state, or the whole country.
Temporal forecasting is dealing with forecasting load for a specific supplier or collection of consumers in future hours, days, months, or even years. According to the forecasting length, there are three different kinds of temporal forecasting.

1. *Long-term load forecasting* (LTLF): mainly for system planning, typically the long term forecast covers a period of 5 to 20 years. Key factors in LTLF includes stock of electricity-using equipment, level and type of economic activity, price of electricity, price of substitute sources of energy, non-economic factors such as marketing and conservation campaigns, and global weather conditions.

2. *Medium-term load forecasting* (MTLF): mainly for the scheduling of fuel supplies and maintenance programs. It usually covers a period of a few months up to 1-2 year. Medium-term load forecasting necessarily represents regular tendencies in power system development.

3. *Short-term load forecasting* (STLF): for the day-to-day operation, weekly and monthly scheduling of the power system. STLF represents the influence of weather and seasonal factors (for example, longitude of day) and several social, economical and maybe technological parameters.

4. *Operative prognosis*: daily prognosis (hour by hour), generally for immediate decisions, that guarantees reliability of energy supply.

In this paper we concerns mainly with the STLF. The short-term forecaster calculates the estimated load for each hours of the next day, the daily peak load, or the daily or weekly energy generation. A large variety of techniques have been investigated in STLF, and the different approaches are still actively referenced.
Among them, the statistical approaches, including the classical multiple linear regression and ARMA (automatic regressive moving average), and Neural Network approach are the most

detailed studied approaches. During past ten years, neural network approaches (feed forward net, back propagation network, radical basis function network, recurrent network (e.g. Hopfield), competitive network and etc.) have been applied to the forecasting problem. Some of them are hybrid with genetic or fuzzy methods.

Our paper is devoted to comparatively new approach – constructing neural polynomial type model with structure self-organization [4, 5], that could be applied to all kinds of forecasting tasks.

II. GMDH Approach

Mathematical models of the multivariable system require a large number of parameters, identification of these parameters is very time-consuming task. Group Method of Data Handling (GMDH) [2, 3] is one of the methods of identifying nonlinear systems with many input variables. Its mathematical model is represented by a hierarchical network of the partial descriptions PDs (i.e., the basic building blocks) and discovers the structure of empirical models as well as performing the task of fitting model coefficients on the basis of observational data.

The idea of GMDH is the following: trying to build an analytical function (called “model”) which would behave itself in such a way that the predicted value of the output would be as close as possible to its actual value. For many applications such an analytical model is much more convenient than the “distributed knowledge” representation that is typical for neural network approach.

Polynomial Network (PN) is a flexible architecture whose structure (topology) is developed through learning. In particular, the number of layers of the PN is not fixed in advance but becomes generated on the fly. In this sense, PN is a self-organizing network. Each node of the PN exhibits a high level of flexibility and realizes a polynomial type of mapping (linear, quadratic, and cubic) between input and output variables.

For example, neuron can be presented with full quadratic polynomial function:

$$y=f(x_i,x_j)=a_0+a_1x_i+a_2x_j+a_3x_ix_j+a_4x_i^2+a_5x_j^2 \qquad (1)$$

Another partial descriptions may be different as:

$$y=f(x_i,x_j)=a_0+a_1x_i+a_2x_j$$

$$y=f(x_i,x_j)=a_0+a_1x_i+a_2x_j+a_3x_ix_j$$

So, in principle, constructed network is the composition of neurons with the mapping fnctions $f(x_i,x_j)$ that are not known beforehand (mixture of linear and nonlinear views may be selected). The fixed number of neurons is selected (or filtered) at each layer and the outputs of these neurons are used on the next layer.

Number of neurons at the first layer is $k=\frac{m(m-1)}{2}$, where m is the number of network inputs (system inputs), and k is the number of pair wise combinations of these inputs.

Only elements that fit the selection criteria (pass the threshold value), are permitted to move to the next layer.

Selection of the best models is illustrated below (see Fig.1). One way to realize models selection is defining a threshold value subjectively, or tuning it heuristically. Using a threshold value, the intermediate models are selected that are the best models in the meaning of a given quality function, they are stored as inputs for the next generation's pair wise model synthesis.

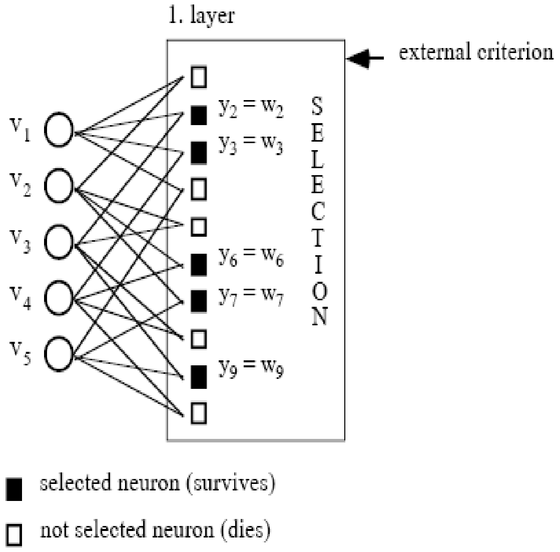

Figure 1. Development of PN topology – 1-st step

III. Design procedure of electrical Load Forecasting Algorithm ELFA

The algorithm is based on the GMDH and utilizes a class of polynomials such as linear, quadratic, and modified quadratic types. By choosing the most significant input variables and polynomial types among various types available, the partial descriptions PDs in each layer can be obtained [5].

The sample is divided into three sets. The training (learning) sub sample is used to derive estimates for the coefficients of the polynomials, the testing sub sample is used to choose the structure of the best models, and small examining sub sample is used to select the optimal model, that is, one with the minimal value of the external regularity criterion.

The framework of the design procedure of the ELFA comes as a sequence of the following steps.

Step 1. *Determine model’s parameters.*

Selection of all model variables, which are divided into input, endogenous and control.

Determination of time lags, characterizing the process dynamics, choosing the suitable criteria for model’s evaluation.

The normalization of the input data is also performed (if required).

Step 2. *Form training, testing and examining data.*

The input - output data set (x_{ij}, y_j) j = 1, 2 ,…, N is separated into training (N_{tr}) data set, testing (N_{te}) data set, and examining (N_{ex}) data set. Obviously we have N= N_{tr} + N_{te} ∔ N_{ex}. The training data set is used to construct models, the testing data set is used to evaluate the models, and examining data set is used to select one final model and equation.

Step 3. *Choose a structure of the models.*

The structure of model is strongly dependent on the number of input variables and the order of PD in each layer. Two kinds of structures, namely, the basic ELFA structure and the modified ELFA structure can be available. The modification of ELFA structure is dependent on some restrictions imposed by prognosis problem – full protection of variables for normative projections or partial involvement of some variables into higher level models (to improve the performance of modeling).

Step 4. *Estimate the coefficients of the PD.*

The vector of coefficients of the PDs is determined using a training data set and minimizing standard mean squared errors (MSE).

$$ER(A) = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} (y_i - y_i(A))^2 \rightarrow \min \qquad (2)$$

where A is the unknown vector of model coefficients.

This step is fulfilled repeatedly for all nodes in the current layer. The total number of polynomials to be constructed is equal to $M = \frac{N(N-1)}{2}$.

Step 5. *Select PDs with the good predictive capability.*

The predictive capability of each PD is evaluated by performance index using the testing data set.

$$CR = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} (y_i - y_i(A))^2 \leq \textit{threshold value} \qquad (3)$$

The F PDs from M with the best predictive capability (the lowest value of the performance index) are choosen. Here, F is the pre-defined number of PDs that must be preserved to next layer. At this step the "best" variables are remaining for the next iteration while the "weakest" ones are eliminated.

Step 6. *Check the stopping criterion.*

The next layer iteration (or the next iteration) begins immediately by repeating steps 4 and 5 as described above. The lowest value of the selection criterion obtained during the iteration is compared with the smallest value obtained at the previous one.

If an improvement is achieved, the algorithm goes back and repeats steps 4 and 5, otherwise the iterations terminate and the task of the network realization is considered to be completed. The ELFA algorithm can also terminates when the number of layers predetermined is reached.

Step 7. Visualization and analysis of the results.

One optimal model, selected according to the minimum value of criterion on examination sub sample, is used further for decision making.

IV. Combinatorial Algorithm COMBI

This is the main GMDH algorithm. Two kinds of models are considered: linear and nonlinear.

The input data sample is a matrix containing N levels (points) of observations over a set of M variables (main variables + lag variables). The input variables are designated as x_i , i = 1, 2 ,…, m and related to output variables y.

Linear model.

Consider the general equation of form

$$y = \theta(x_1, x_2, ..., x_m) \qquad (4)$$

with the coefficients that are introduced as vector with components $a_0, a_1, ..., a_M$. and is linear relative to variables x_i.

The first layer model is obtained by using the information in the every column of the training sub sample of observations. The candidate models for the first layer have the form:

$$y = a_0 + a_1 x_i \ , \quad i = 1, 2, ..., M \qquad (5)$$

To compute the values of the coefficients a0 and a1 for each of the M models, a system of Gauss normal equations is solved. On the first layer, the system of Gauss normal equation for ith model will be

$$\begin{bmatrix} N_{tr} & \sum_{k=1}^{N_{tr}} x_{ki} \\ \sum_{k=1}^{N_{tr}} x_{ki} & \sum_{k=1}^{N_{tr}} x_{ki}^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{N_{tr}} y_k \\ \sum_{k=1}^{N_{tr}} x_{ki} y_k \end{bmatrix} \qquad (6)$$

where N_{tr} is the number of observations in the training set.

The candidate models for the second layer have the form:

$$y = a_0 + a_1 x_i + a_2 x_j \ , \quad i = 1, 2, ..., F-1; \quad j = i+1, ... F \quad i \neq j \quad F \leq M \qquad (7)$$

At this level, as a result of pair wise combinations of x_i , x_j variables, the total number of models generated is equal to C_F^2.

On the second layer, the system of Gauss normal equation for ith model will be

$$\begin{bmatrix} N_{tr} & \sum_{k=1}^{N_{tr}} x_{ki} & \sum_{\substack{k=1 \\ j \neq i}}^{N_{tr}} x_{kj} \\ \sum_{k=1}^{N_{tr}} x_{ki} & \sum_{k=1}^{N_{tr}} x_{ki}^2 & \sum_{\substack{k=1 \\ j \neq i}}^{N_{tr}} x_{ki} x_{kj} \\ \sum_{k=1}^{N_{tr}} x_{kj} & \sum_{\substack{k=1 \\ i \neq j}}^{N_{tr}} x_{kj} x_{ki} & \sum_{k=1}^{N_{tr}} x_{kj}^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{N_{tr}} y_k \\ \sum_{k=1}^{N_{tr}} x_{ki} y_k \\ \sum_{k=1}^{N_{tr}} x_{kj} y_k \end{bmatrix} \qquad (8)$$

When all possible models from this layer have been formed, the F best are chosen on the test sub sample according to the values of *regularity criterion*.

. The variables of the second layer that give best results will proceed to form third layer candidate models. This procedure is carried out as long as the criterion decreases in value, and candidate models at the m th layer will have the form:

$$y = a_0 + a_1 x_i + a_2 x_j + \ldots + a_m x_z, \; i = 1, 2, \ldots, F - m + 1; \; j = i + 1, \ldots, F - m + 2; \; i \neq j \qquad (9)$$
$$m = j + 1, \ldots, F \quad i \neq j \neq \ldots \neq m \; F \leq M$$

At this level, as a result of m elements ombinations of variables, the total number of models generated is equal to C_F^m.

The removal of "bad" models automatically causes the removal of the corresponding variables from the training and testing sub samples. These variables do not participate in the combinatorial process of models generation for the next layers.

Fig. 2 below illustrates Combinatorial GMDH algorithm (COMBI), where

1 - layers of partial descriptions complexing,
2 - form of partial descriptions,
3 - choice of optimal models,
4 - additional model definition by discriminating criterion.

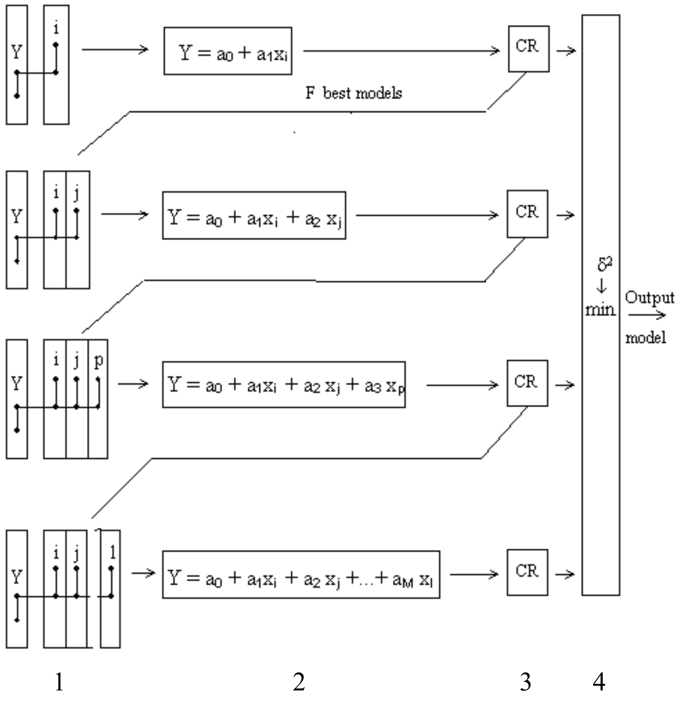

Figure 2. Combinatorial algorithm – illustration.

Nonlinear model.

Linear partial description can be extended up to full nonlinear polynomial description. Full polynomial with p variables of degree q consists of C_{p+q}^q terms.

The candidate models for the first layer are one variable quadratic polynomials and have the form ($q = 2, p = 1,$ 3 terms):

$$y = a_0 + a_1 x_i + a_2 x_i^2, \quad i = 1, 2, \ldots, M \qquad (10)$$

Adding of x_i^2 term formally brings to linear model of the second layer, but without combinatorial enumeration. The equation (8) is used where x_{kj} is replaced by x_{ki}^2.

Some of the models (after selection) is moving to the next layer.

The candidate models for the second layer are two variable full quadratic polynomials that have the form ($q = 2, p = 2,$ 6 terms):

$$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + a_4 x_j^2 + a_5 x_i x_j, \quad i = 1, 2, \ldots, F - 1; \quad j = i + 1, \ldots, F \quad i \neq j \quad F \leq M \qquad (11)$$

Six unknown coefficients of the equation (11) are determined using the same approach that in linear case, where additional variables are introduced instead of nonlinear terms. Description (11) is taken as a basic for the nonlinear modeling. The further iterations could increase the number of variables and in parallel the number of nonlinear terms in partial descriptions. However, the number of terms in model's partial descriptions is increasing so fast that actually the number of variables could not exceed 10 and the polynomial degree - 3. Otherwise, we are forced to turn down full enumeration algorithm and use multilayered iterative GMDH algorithm.

V. Electrical Load Data Analysis

Time series prognosis problem requests for three main parameters: number of variables M, size of the observed sample data in time series N, and the considered depth of the process dynamics - τ (time lag).

The input data set for electrical load prediction consists of three variables: electrical load hour-by-hour, daily maximal and minimal load, average daily temperatures and several secondary indexes, such as load coverage, ratio P/Q and some other parameters. These data that covers 2005-2006 years was extracted from load calculations in Azerbaijani power system and separated according to seasons (Fig.1 and Fig.2).

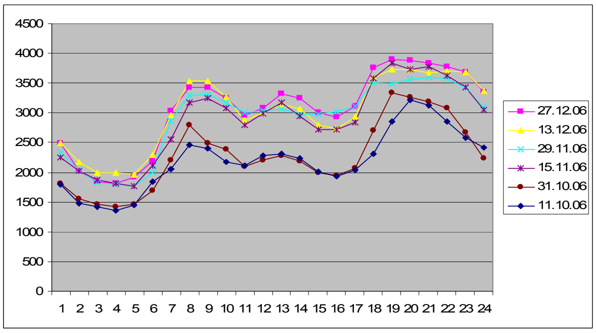

Figure 3. 2006 – winter, Wednesdays

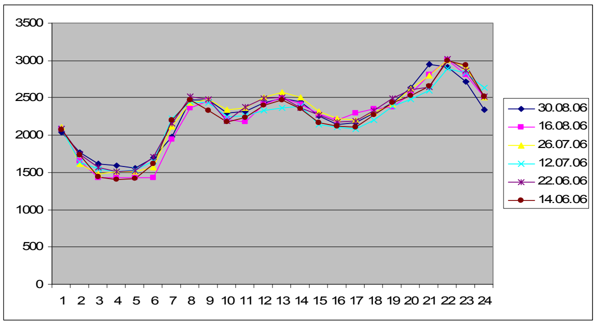

Figure 4. 2006 – summer, Wednesdays

Several factors could be taken into account on the data collecting step: daily load changes (daytime, nighttime), internal weekly cycles (Saturday, Sunday), seasonal changes (summer, winter).

The objective is: one day ahead prognosis.

Accordingly 24 linear (and non-linear) regression models are created in the form demonstrated below:

$$y_{1,t} = f_1(x_{1,t}, x_{2,t}, x_{3,t}, x_{1,t-1}, x_{2,t-1}, \ldots, x_{4,t-1}, \ldots, x_{1,t-3}, x_{2,t-3}, \ldots, x_{4,t-3})$$

$$y_{2,t} = f_2(x_{1,t}, x_{2,t}, x_{3,t}, x_{1,t-1}, x_{2,t-1}, \ldots, x_{4,t-1}, \ldots, x_{1,t-3}, x_{2,t-3}, \ldots, x_{4,t-3})$$

$$y_{24,t} = f_5(x_{1,t}, x_{2,t}, x_{3,t}, x_{1,t-1}, x_{2,t-1}, \ldots, x_{4,t-1}, \ldots, x_{1,t-3}, x_{2,t-3}, \ldots, x_{4,t-3})$$

The problem stated is the typical interpolation one.

Model training is executed on data of the previous day, data of the same and previous days but one week before.

In each line (equation) we have 4x3=12 lag variables, 3 separate temperature variables and in total 15 independent variables on the right side of equation.

Parameter's values were normalized using the formula

$$\bar{x}_i = \frac{x_i - x_{i\min}}{x_{i\max} - x_{i\min}}$$

The linear regression model keeps all input and lag variables in the right side of equation

$$y_t = Ax_t + \sum_{j=1}^{3} B_j x_{t-j}$$

All coefficients of 24 linear and nonlinear equations were calculated and the result of ELFA algorithm implementation is demonstrated on Fig.5.

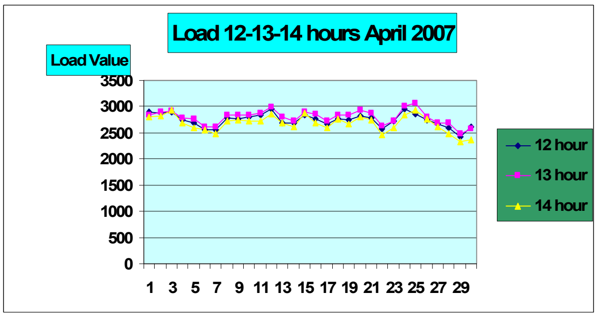

Figure 5. 2 hours ahead prediction 2007 – April

VI. Conclusion

An original combinatorial algorithm of GMDH was analyzed in detail, improved in performance and implemented. The nonlinear models produce better results than the linear ones.

The proposed method demonstrates the ability to select simple models characterized by a high prediction ability and thus the method and algorithm proposed are of a considerable interest when compared with multiple linear regressions (MLR), ARIMA and a traditional neural network models.

Future directions of investigation will be related with the analysis and implementation of MultiLayer Iterative and Objective System Analysis algorithms, selection of the best alqorithm as a result of comparative analysis .in the process of data load forecasting.

References

[1] Шумилова Г.П., Готман Н.Э., Старцева Т.Б. Прогнозирование нагрузки ЭЭС на базе новых информационных технологий. Сборник, ред. Манов Н.А., Чукреев Ю.Я., Успенский М.И. и др. Екатеринбург: Уро РАН, 2002, стр. 127-156.

[2] Mueller, J.-A., Lemke, F. Self-Organizing Data Mining Libri, Hamburg, 2000, ISBN 3-89811-861-4

[3] Madala,H.R. and Ivakhnenko,A.G. Inductive Learning Algorithms for Complex Systems Modeling. CRC Press Inc., Boca Raton, 1994, p.384.

[4] R.E. Abdel-Aal. "Modeling and forecasting electric daily peak loads using abductive networks," International Journal of Electrical Power & Energy Systems, Volume 28, Issue 2, February 2006, pp. 133-141.

[5] Sung-Kwun Oh, Witold Pedrycz. "The design of self-organizing Polynomial Neural Networks." Information Sciences, Volume 141, Issues 3-4, April 2002, pp. 237-258