Application of Generalized Neuron in Electricity Price Forecasting

Sanam Mirzazad-Barijough, *Student Member*, *IEEE*, and Ali Akbar Sahari, *Student Member*, *IEEE*

Abstract-- **With recent deregulation in electricity industry, price forecasting has become the basis for this competitive market. The precision of this forecasting is essential in bidding strategies. So far, the artificial neural networks which can find an accurate relation between the historical data and the price have been used for this purpose. One major problem is that, they usually need a large number of training data and neurons either for complex function approximation and data fitting or classification and pattern recognition. As a result, the network topology has a significant impact on the network computational time and ability to learn and also to generate unseen data from training data. To overcome these problems, a new structure using generalized neurons (GN) is adapted in this paper. The proposed structure needs a smaller data set for training. So this property of GN can be very useful for price forecasting. The data such as historical prices are not available enough for most markets. The significance, viability and efficiency of the proposed approach, in electricity price forecasting, are shown using Ontario market data points and various GN models are compared.**

Index Terms-- **back-propagation, generalized neuron, price forecasting**

I. INTRODUCTION

In recent years, the electricity industry is going toward a competitive framework. Thus the market environments are being replaced with the long-established regulated frameworks [1]. This deregulation is done to maximize the efficient generation, consumption and also to reduce the energy prices in this competitive market. So, more attention should be paid to electricity price forecasting with high and acceptable precision. Bidding strategies are based fundamentally on this forecasting. With good forecast of next-day price, the producers can maximize their own benefits and the consumers can make a good plan to maximize their own utility [2].

Generally both hard and soft computing techniques could be used for this price forecasting. Hard computing techniques such as regression analysis need an exact model of the system. Although the result is very accurate here, but a lot of information is needed for prediction. The computational cost is also high in such methods.

Soft computing methods, such as neural networks, are used more in recent years. These methods do not need an explicit model of the system. They act as a kind of universal approximation and can find a mapping between input and output data points. Among these methods, Multi-layer Neural Networks are powerful tools for forecasting the price using historical data [1].

However, as it can be seen in this paper, simulation results show that performance of our new approach based on generalized neurons (GN) is better than those of back propagation neural network (BPNN) systems.

The remainder of this paper is organized as follows. Commonly used neurons in BPNNs are briefly described in Section II. In Section III, the description of GN models can be found. The training algorithm of GN is presented in section IV. Problem formulation of the GN-based price forecasting is given in Section V. Comparison of GN and ANN based price forecasting is reported in VI. The index used for forecasting effectiveness is introduced in VII. Simulation results are discussed in Section VII. Finally Section IX concludes the paper.

Sanam Mirzazad-Barijough is with the Department of Electrical Engineering, Pennsylvania State University, University Park, USA (e-mail: mirzazad@psu.edu).

Ali Akbar Sahari is with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: sahari@ee.sharif.edu).

II. COMMON NEURON MODEL

The common neuron model used in BPNN systems is shown in Fig. 1. As can be seen from the figure, it is constructed on a basis function and activation function. The basis function is summation in the common model. The functions which can be used for activation can either be Sigmoid, tangent hyperbolic or linear limiters.

III. DESCRIBING A GN MODEL

In real life problems like price forecasting, mapping between inputs and outputs, faces failure using ordinary model. To overcome this problem the GN model uses fuzzy compensatory operators, which are partly sum and partly product. It also has two activation functions which are related to each other with weight sharing. GN model has flexibility at both basis and activation functions with respect to application of model to obtain the best mapping.

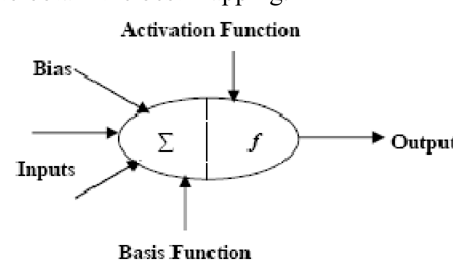

Fig. 1. Common neuron model

Some of the possible models for GN form are described and used in this paper. All the models have two basic functions. Usually $\sum$ and $\prod$ are used for this purpose but other fuzzy operators such as max, min … can be used. In this paper $\sum$ and $\prod$ are used for simplicity. As mentioned above the model also has two activation functions. These functions may either be Sigmoid, Gaussian or Ramp. The different types of GN model are as follows:

a) The first model has two parts. One part sums the product of the inputs and weights belonging to this part. The result is transformed through a ramp function to produce first term of the result.

$$O_{\Sigma} = f_1\left(\sum_{i=1}^{n} W_{\Sigma i} \times X_i + X_{o\Sigma}\right) \tag{1}$$

Where X_i and $X_{o\Sigma}$ represent the input and bias respectively and $W_{\Sigma i}$ is the weight which is related to the input of $\sum$ part. f_1 is a ramp function in this model. The other part multiplies all inputs and their associated weights and then transfers it through a ramp function to produce the second term.

$$O_{\Pi} = f_2\left(\prod_{i=1}^{n} W_{\Pi i} \times X_i + X_{o\Pi}\right) \tag{2}$$

$X_{o\Pi}$ is bias and $W_{\Pi i}$ is the weight for this part. f_2 is also a ramp function.

The final output is a summation of the two calculated terms multiplied by the weights W and 1-W. The obtained result can be seen in Fig. 2 too.

$$O_f = O_{\Sigma} \times W + O_{\Pi} \times (1 - W) \tag{3}$$

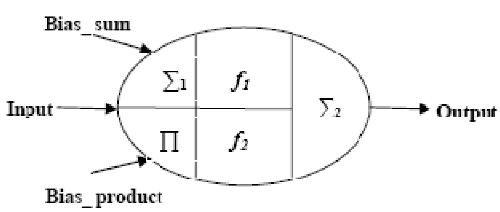

Fig. 2. The first GN model

Because of the summation in the last part, this model is named as summation type neuron model [3].

b) The second model is similar to the first model but $f_1$and f_2 are not ramp. f_1 is a Sigmoid function and f_2 is a Gaussian function. Thus the outputs of two parts will be as follows:

$$O_{\Sigma} = \frac{1}{1 + e^{-S_{net}}} \tag{4}$$

where $S_{net} = \sum_{i=1}^{n} W_{\Sigma i} \times X_i + X_{o\Sigma}$, and:

$$O_{\Pi} = e^{-P_{i-net}^{2}} \tag{5}$$

where $P_{i-net} = \prod_{i=1}^{n} W_{\Pi i} \times X_i + X_{o\Pi}$.

c) In the third model the output of the first part ($\sum$ part) is different from previous cases and is described in (6).

$$O_{\Sigma} = f_1\left(\sum_{i=1}^{n} \left(W_{\Sigma i} + X_i\right)^2 + X_{o\Sigma}\right) \tag{6}$$

Instead of 2 in the power term of (6), other numbers can be used. f_1 and f_2 are ramp functions in this case.

d) The fourth model is the same as the third model with a slight difference that f_1 is Sigmoid and f_2 is Gaussian [4].

IV. Training The GN

The training process requires a set of examples which include inputs and target outputs. During training, the weights and biases are iteratively updated to minimize the difference between the desired output and the output which is obtained from neural network. Here, the first model is applied to explain the GN training. It should be noted that the training for other models will be the same.

For the GN training process these steps should be followed:

Step 1) The output of the first part of GN (O_{Σ}) is calculated.

Step 2) The output of $\prod$ part of GN (O_{Π}) is calculated.

Step 3) The final output of the GN is intended.

Step 4) After calculating the output, it is compared with desired output. Using back-propagation algorithm the training is done to minimize the error. Thus the next step will be the calculation of the error. The sum-squared error which is used for analyzing the convergence of all patterns is defined as (7).

$$E = 0.5 \times \left(Y - O_f\right)^2 \tag{7}$$

Y is the target output and O_f is the output obtained from GN. The factor 0.5 is used to simplify the calculations.

Step 5) The weights are updated with respect to the error following to steps a, b, c.

a) The total weight W is updated using (8):

$$\Delta W = \eta \times \left(Y - O_f\right) \times \left(O_{\Sigma} - O_{\Pi}\right) + \alpha W(j-1) \tag{8}$$

b) The weights for the $\sum_1$ part of GN are updated as:

$$\begin{gathered} W_{\Sigma i}(j) = W_{\Sigma i}(j-1) + \Delta W_{\Sigma i} \\ \Delta W_{\Sigma i} = \eta\left(Y - O_f\right) \times W \times X_i + \alpha W_{\Sigma i}(j-1) \end{gathered} \tag{9}$$

c) The weights for the $\prod$ part of GN are updated as:

$$\begin{gathered} W_{\Pi i}(j) = W_{\Pi i}(j-1) + \Delta W_{\Pi i} \\ \Delta W_{\Pi i} = \frac{\eta\left(Y - O_f\right) \times (1 - W) \times P_{i-net}}{W_{\Pi i}} + \alpha W_{\Pi i}(j-1) \end{gathered} \tag{10}$$

j is the iteration number and i indicates the number of input. α is the momentum factor which is used for better convergence to the global minimum and η is the learning rate. Both of these factors can be selected in the range of 0 to 1 [3].

V. GN-BASED PRICE-FORECASTING

Input selection is an important factor in using neural networks which should be considered carefully to obtain a precision result. Using optimal inputs will improve the accuracy and convergence speed. Several inputs may influence the price predicting as inputs. Among these factors, the most important parameters are historical prices and demand data. The importance of these parameters is achieved by experience. The historical price data with more effect on predicted price can be selected with correlation [5].

The selection of the number of inputs is done with respect to construction of the GN model. Because just one neuron is used, the number of inputs required for this model is less than ANN. This is one of the advantages of GN. Using more input data will result in over fitting.

The number of inputs in this paper is obtained by the trial and error method. The best result is obtained with following scheme and the whole structure is shown in Fig. 3. The identification of inputs is as follows:

1) Historical system prices:
   - The price for an hour before the desired hour: P(d,t-1)
   - The price for the same hour a day before: P(d-1,t)
   - The price for the same hour a week before the day: P(d-7,t)
2) System demand:
   - The system demand in the desired hour: L(d ,t)
   - The system demand in the hour before the desired hour: L(d,t-1)
   - The system demand in the same hour for a day before: L(d-1,t)
   - The system demand in the same hour a week before the day: L(d-7,t)

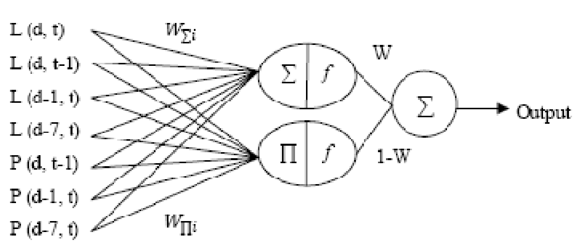

Fig. 3. Structure of GN forecasting model

VI. COMPARISON OF GN AND ANN BASED PRICE FORECASTING

The GN model is less complex in comparison with a multi layer ANN. For a comparison purpose, the purposed GN model with 7 inputs is compared with a 7-11-5-1 ANN in table 1. It is clear from the table that the number of interconnections and therefore the weights which must be updated are fewer in GN model than ANN. It is clear that this property will reduce the training time and data. Reducing the number of data is an essential factor in price forecasting because the huge amount of data needed for price forecasting in ANN is not available.

Table 1. Comparison of the complexity of ANN and GN models

MODEL	ANN	GN
Construction	7-11-5-1	1
Number of layers	3	1
Number of neurons	17	1
Number of weights to be updated	390	15

VII. EVALUATING INDEX OF FORECASTING EFFECTIVENESS

For the purpose of evaluating the forecasting effectiveness, the factor "Mean Absolute Percentage Error" (MAPE) is used in this paper. MAPE is defined as following:

$$MAPE = \frac{1}{N} \times \sum_{i=1}^{N} APE_i \tag{11}$$

$$APE_i = |PE_i| \tag{12}$$

Percentage Error (PE) = 100 * (forecasted price-actual price) / (actual price)
where N is the number of forecasted data [6].

VIII. SIMULATION RESULTS

We use MATLAB environment for simulating the GN model. This model is performed on Ontario electricity market. The data for Ontario power market both the clearing price and load for the 26th of January 2003 is modified. (α=0.01) and (η=0.8) are chosen. 120 training patterns are used for training GN and the training epochs are 100. As mentioned, this number is very low compared to ANN. The simulation results can be seen for the models described in section III. An ANN with the same number of inputs and training epochs needs about 350 input patterns to obtain the same MAPE. The results for all models can be seen in figures 4-7.

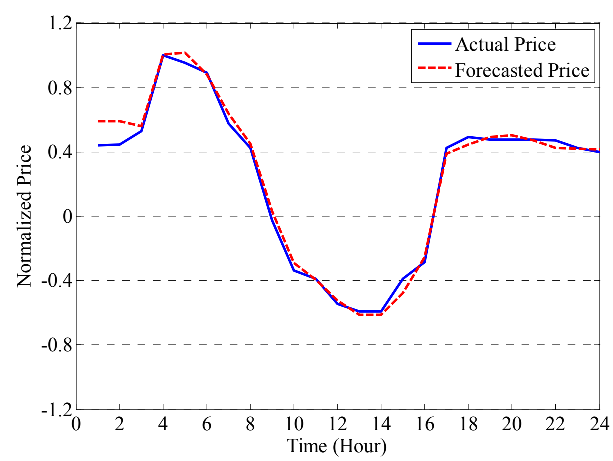

Fig. 4. The results for the first GN model

In table 2 a comparison between these models can be seen. It is clear that the best result is achieved using the first model. The other three models do not give acceptable results. They also cannot predict the spikes well which arise from unknown bidding strategies from participants in the market.

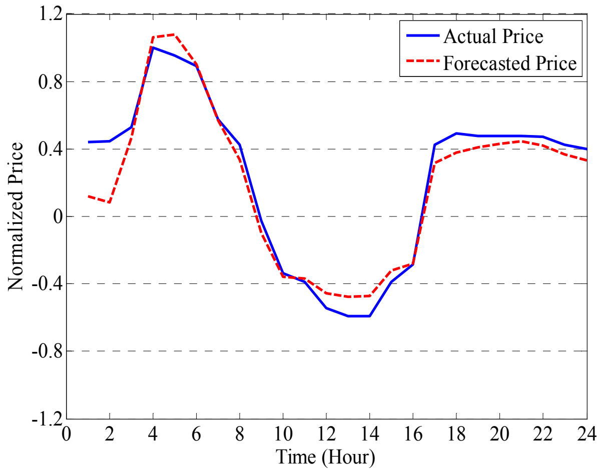

Fig. 5. The results for the second GN model

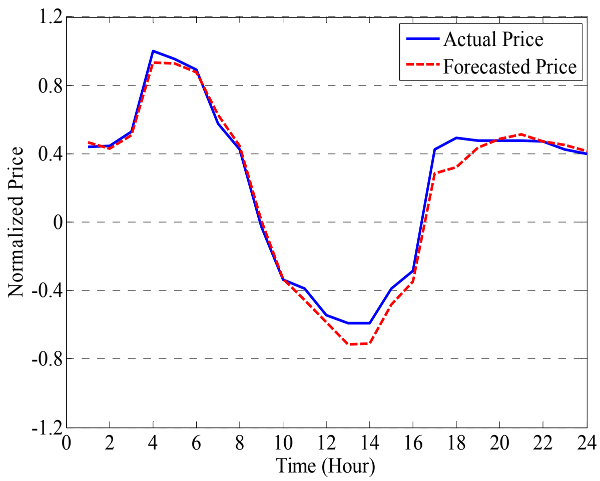

Fig. 6. The results for the third GN model

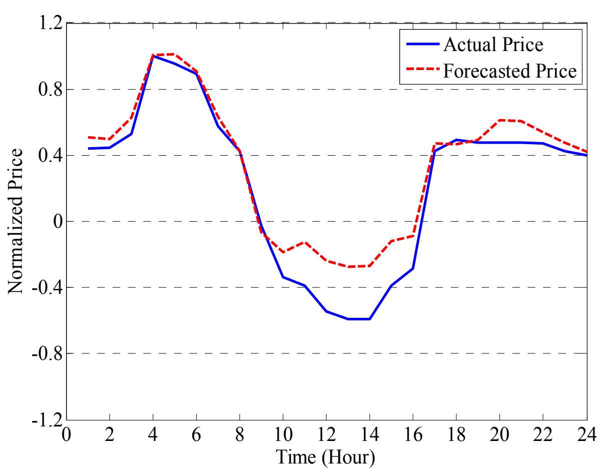

Fig. 7. The results for the fourth GN model

IX. CONCLUSION

Because of recent changes in electric industry which led to a competitive market, price forecasting has become the basis for all bidding strategies.

A GN model is used in this paper. It can map preciously inputs to outputs in complex nonlinear problems. It uses only one neuron and the back-propagation algorithm is used for training. Therefore it has small number of weights which must be trained. Thus unlike the ANN, which is used until now for forecasting, the required training data is low significantly which is very important in price forecasting because of lacking data. Training time is also reduced as the result of the GN structure.

Various models of GN are introduced and the simulation results are compared for the models. The simulations show the illustrated properties of GN. Although the first model shows better performance for price forecasting but we can use other models for different cases.

Table 2. Comparison of the errors of GN models

MODEL	MAPE %	Max. Percentage error %	Min. Percentage error %
First	2.02	0.06	-0.066
Second	2.89	0.102	-0.020
Third	4.45	0.161	-0.098
Fourth	7.36	0.162	-0.262

X. REFERENCES

[1] J.P.S. Catalão, S.J.P.S. Mariano, V.M.F. Mendes, and L.A.F.M. Ferreira, "Short-term electricity prices forecasting in a competitive market: A neural network approach," *Electrical Power Energy Systems*, pp. 1297-1304, 2007.

[2] P. Mandel, T. Senjyu, and T. Funabashi, "Neural network models to predict short-term electricity prices and loads," *IEEE Industrial Technology*, pp. 164-169. Dec. 2005.

[3] D.K. Chaturvedi, O.P. Malik, and P.K. Kalra, "Experimental studies with a generalized neuron-based power system stabilizer," *IEEE Trans. Power Systems,* vol. 19, no. 3, August 2004.

[4] D.K. Chaturvedi, M. Mohan, R.K. Singh, and P.K. Kalra, "Improved generalized neuron model for short-term load forecasting," *Soft Computing 8*, pp. 10-18, July 2003.

[5] M. Ranjbar, S. Soleimani, N. Sadati, and A.M. Ranjbar, "Electricity price forecasting using artificial neural networks," *IEEE Power Electronics, Drivers and Power Syst.*, Dec. 2006.

[6] H.Y. Yamin, S.M. Shahidehpour, Z. Li, *Market operations in electric power systems,* 1thed., Chap. 3.

[7] Q. Ding, J. Tang, and J. Liu, "Application of new FCMAC neural network in power system marginal price forecasting," *IEEE Power Engineering*, Dec. 2005.

[8] B.R. Szkuta, L.A. Sanabria, and T.S. Dillon, "Electricity price short-term forecasting using artificial neural networks," *IEEE Trans. Power Systems,* vol. 14, no. 3, August 1999.

[9] J. Bastin, J. Zhu, V. Banunarayana, and R. Mukerji, "Forecasting energy prices in a competitive market," *IEEE Computer Applications in Power*, pp. 40-45, July 1999.

XI. Biographies

Sanam Mirzazad Barijough (S'06) received the B.S. and M.S. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 2006 and 2008 respectively. She is currently a Ph.D. student in Electrical Engineering at the Pennsylvania State University, University Park. Her research interests are Power systems, Hybrid and Switched systems.

Ali Akbar Sahari (S'07) received the B.S. degree in Power engineering from Sharif University of Technology, Tehran, Iran, in 2006, and the M.S. degree in electrical engineering from the same university in 2008. His current research interests include power electronics, power systems and control systems.