International Journal of Advanced Culture Technology Vol.2 No.2 11-14 (2014)
IJACT.2014.2.2.11

IJACT 14-2-3

RNN NARX Model Based Demand Management for Smart Grid

Sang-Hyun Lee[1], Park Dae-Won[2*], Kyung-Il Moon[3]

[1, 3]*Dept. of Computer Engineering, Honam University, Korea*
{leesang64, kimoon}@honam.ac.kr

[2*]*Darwin Company, Korea*
Sunlion82@naver.com

Abstract

In the smart grid, it will be possible to communicate with the consumers for the purposes of monitoring and controlling their power consumption without disturbing their business or comfort. This will bring easier administration capabilities for the utilities. On the other hand, consumers will require more advanced home automation tools which can be implemented by using advanced sensor technologies. For instance, consumers may need to adapt their consumption according to the dynamically varying electricity prices which necessitates home automation tools. This paper tries to combine neural network and nonlinear autoregressive with exogenous variable (NARX) class for next week electric load forecasting. The suitability of the proposed approach is illustrated through an application to electric load consumption data. The suggested system provides a useful and suitable tool especially for the load forecasting.

Keywords: *Demand response, Electric load, NARX, Smart grid*

1. Introduction

Among the objectives of the smart grid, demand management will play a key role in increasing the efficiency of the grid [7]. In the smart grid, demand management extends beyond controlling the loads on the demand-side. Controlling demand side load is known as demand response, and it is already implemented in the traditional power grid for large-scale consumers although it is not fully automated yet. The demand response directly aims to control the load of the commercial and the industrial consumers during peak hours. Peak hours refer to the time of day when the consumption exceeds the capacity of the base power generation plants that are built to accommodate the base load. In smart grid, automated demand response is being considered. In automated demand response programs, utilities send signals to buildings and industrial control systems to take a pre-programmed action based on the specific signal. This paper presents the formulation of a demand response model identification method based on recurrent neural network (RNN) Nonlinear Auto-Regressive with external input (NARX) model derived from dynamic feed-forward neural network (DFNN) by adding feedback connection between output and input layers. The proposed identification method identifies the neural network (NN) model of an input-output system. The identified NN model is then validated using 5-step prediction simulations. The suggested algorithm is applied to forecast electric load

Manuscript Received: Oct. 15, 2014 / Revised: Nov. 14, 2014 / Accepted: Dec. 5, 2014
Corresponding Author: Sunlion82@naver.com
Tel:+82-62-940-5977,Fax: +82-62-940-5779
Darwin Company, Korea

consumption. Simulation results show that the RNN models the electricity demand to a high degree of accuracy.

2. NARX model class

NARX models are the nonlinear generalization of the well-known ARX models, which constitute a standard tool in linear black-box model identification [4]. These models can represent a wide variety of nonlinear dynamic behaviors and have been extensively used in various applications. ANARX model is formulated as a discrete time input–output recursive equation: $y(t) = f(y(t-1),\ldots,y(t-n_a),x(t-1),\ldots,x(t-n_b)) + \varepsilon(t)$.

Here x and y are the model input and output, n_a and n_b are the respective maximum lags, and ε is noise term, generally assumed Gaussian and white. The optimal predictor form of this model is $\hat{y}(t) = f(y(t-1),\ldots,y(t-n_a),x(t-1),\ldots,x(t-n_b))$. Here $\hat{y}(t)$ denotes the one-step ahead prediction of $y(t)$. Depending on how function f is represented and parameterized, different NARX model structures and, consequently, identification algorithms are derived. The function f for NARX models can also be modeled by means of artificial neural networks. A huge literature is available on NN model theory and applications [1]. Here, only a brief description of NNs is presented. Neural networks are composed by simple connected elements (denoted as neurons) operating in parallel. Each neuron's output is obtained by filtering a weighted sum of its inputs through a usually nonlinear function, i.e., the so-called activation function. The weights associated with the network connections are tuned during the training (learning) phase in order to reduce a given cost function, such as MSE (or a modified cost function). The neural network structure used in many studies is a standard feed-forward neural network. This kind of network computes a vector function $f : R^Q \rightarrow R^L$ where Q and L are the dimensions of the input and output vectors of the net, respectively, the l-th element of the vector function f for the n-th pattern($v_n \in R^Q$) is defined as (M is the number of the neurons in the hidden layer) $f(v^n) = af_2\left(\sum_{m=1}^{M}(OW_{lm} \cdot a_m) + g_l\right)$, $a_m = af_1\left(\sum_{q=1}^{Q}(IW_{mq} \cdot v_q) + b_m\right)$.

Here af_1 and af_2 are two real continuous functions, called activation function of the hidden layer (af_1) and of the output layer (af_2). The matrices IW ($M \times Q$) and OW ($L \times M$) are the input and output weight matrix, respectively, and b ($M \times 1$) and g ($L \times 1$) vectors are the bias terms. Neural networks learn on a training data set, tuning the parameters IW, OW, b and g by means of the well-known Levenberg-Marquardt back-propagation (BP) algorithm. The choice of the best neural network weights has been done minimizing a selected cost function (MSE or WMSE).

3. Multilayer perceptron RNN identification scheme

The proposed architecture for the nonlinear NN model identification scheme is a multilayer perceptron recurrent neural network (MLP RNN) using the teacher forcing training method [2][3]. This architecture places the true system in parallel with the neural network, as illustrated in Fig. 1.

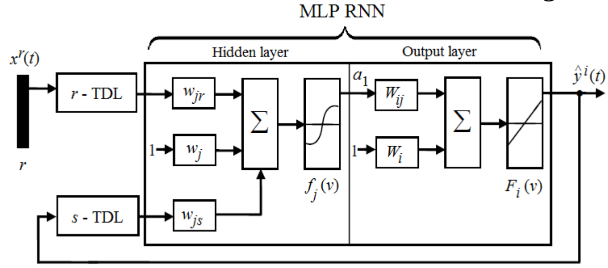

Figure 1. MLP RNN NARX network

According to this method one can train the MLP RNN as a feed-forward network by using the LMA proposed in [5]. Thus, for the MLP RNN, given the input data $x^l(t)$ and output observations $y^i(t)$ as Z={[$x^l(t)$, $y^i(t)$], t=1, … , N}. The first m inputs $x^r(t)$ is applied to r tapped-delay-lines (r-TDL) memory.

At time t+1, the first n predicted outputs $y^s(t)$ is fed back to the input via another s tapped-delay-lines

(s-TDL) memory. The contents of these two TDL memories are used to feed the input layer of the MLP RNN to predict the state of the system. m and n corresponds to the number of past inputs and outputs used by the network for prediction; N is the total number of input-output data pair; m and n are also the orders of the line memories applied to the input and feedback output signals respectively; l and i are the respective number of inputs and outputs vectors of the true system; r and s are the respective number of inputs and outputs fed to the MLP RNN. Thus the output is ahead of the input by one time unit; r and s are the respective number of inputs and outputs fed to the MLP RNN. Thus the output is ahead of the input by one time unit. The present and past values of the network inputs represent exogenous inputs originating from outside the network. The delayed values of the outputs on which the model output is regressed are $y^s(t)$, $y^s(t-1)$, … , $y^s(t-n)$.

The NARX model structure for an input-output RNN model has the following structure in the predictor form: $\hat{y}^i(t) = g^i\left[\theta^i, \varphi^q(t), t\right] + e^i(t)$, $\quad \varphi^q(t) = \left[x^r(t-1), \ldots, x^r(t-m), y^s(t-1), \ldots, y^s(t-n)\right]^T$

Here $\hat{y}^i(t)$ denotes the predicted output of the network; $\varphi^q(t)$ denotes the regressor vector made up of $\varphi^r(t)$ for past inputs and $\varphi^s(t)$ for past outputs; g is an unknown unidentified nonlinear function; θ is an unknown vector of d elements containing adjustable parameters of the network (i.e., the joint weights of the network); e is a white noise independent of the inputs.

The predicted model output of Fig. 1 can be expressed mathematically in terms of the network parameters as: $\hat{y}^i(t) = F_i \cdot \left[\sum_{j=1}^{n_k} W_{ij} f_j \cdot H_j + W_i\right]$, $\quad H_j = \sum_{r=1}^{m} w_{ja}\varphi^r(t-1) + \sum_{s=1}^{n} w_{jb}\varphi^s(t) + w_j$

Here j is the number of hidden neurons, (w_{ja} and w_{jb}) and W_{ij} are the hidden and output weights respectively; w_j and W_i are the hidden and output biases; F_i is a linear activation function for the output layer and f_j is a fast hyperbolic tangent function for the hidden layer.

Let the real system corresponding to $\theta^i = \hat{\theta}^i$ be given by $y^i(t) = g^i\left[\hat{\theta}^i, \varphi^q(t), t\right] + e^i(t)\right]$

At time t+1, the m past inputs and the n past outputs are available. The predicted model output represents the estimate of the true (desired) output. The estimate is subtracted from the real output to produce the error signal. The error is used to adjust the synaptic weights so as to minimize the error. The model identification problem reduces to searching the parameter set θ and obtaining reasonable estimate θ^* using a suitable training (nonlinear optimization) algorithm.

The RNN network training is formulated as a total square error (TSE) problem which can be expressed as:

$$J(\theta^i, Z^N) = \frac{1}{N}\sum_{t=1}^{N}\left[y^i(t) - \hat{y}^i(t)\right]^2$$

However, the effects of the modeling errors can be considered in terms of bias and variance errors. The model parameterization contains many parameters, estimating several of these models accurately may be impossible during the training process. Moreover, as θ^j contain many parameters, the minimization of TSE may be ill-conditioned due to the effects of the bias and variance errors.

4. Results

To test the performance of proposed RNN NARX identification algorithm, we consider a problem to predict electric load represented in [6]. The RNN NARX model identification and validation algorithm is implemented and simulated using MATLAB to predict electric load. The input variables consist of the times series hour weekly load data rearranged in multi input single output. For the given weekly data points RNN NARX predictor is supposed to work with six inputs and one output only. The inputs are directly extracted from the data sets. Here, the weekly load data is used. There are 168 data samples ($y(t)$, $x(t)$), from t = 1 to t = 168 corresponding of hourly load for a week. With this idea, the sample ahead is forecasted by using past samples. The graphical representations of the comparison between the desired weekly load values and the RNN NARX predicted values are presented in Fig. 2. We used ten hidden layer neurons to predict the electric load. Relatively small validation error implies that the network has been well trained. This small error also indicates that the identified NN model approximates the nonlinear dynamics of the demand response system accurately.

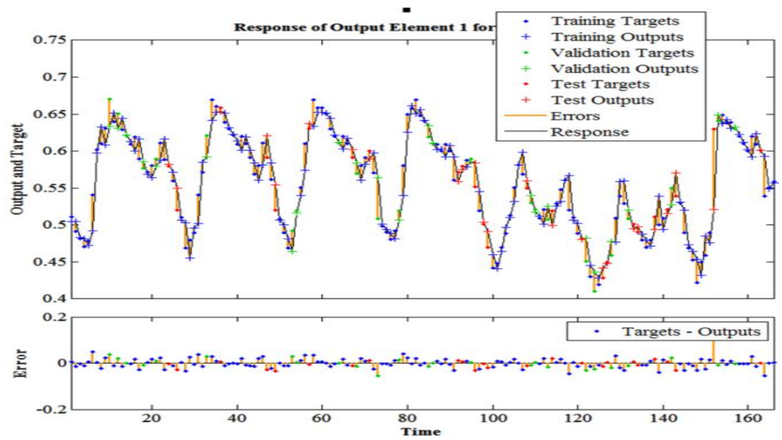

Figure 2. Comparison of load series within a week realized and predicted

5. Conclusion

The NARX model validation results have shown good agreement between the predicted and the real outputs. This study presented an application of RNN NARX model with a high forecasting accuracy that depends on previous weekly load data. The results obtained show that the RNN NARX approach can accurately predict weekly load consumption and the performance of the proposed model is not affected by rapid fluctuations in power demand which is the main drawback of neural networks and fuzzy models in the smart grid environment. The validation result also show that the proposed nonlinear model identification method modeled the electric load consumption to a high degree of accuracy. The parallelization of these algorithms for the real-time demand management system in smart grid environment is recommended for further studies.

References

[1] M. W. Gardner, and S. R. Dorling. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. Atmospheric Environment 32, 2627–2636, 1998.

[2] M. T. Hagan, H. Demuth, and M. Beale. Neural Network Design. PWS Publishing, Boston, MA, 1996.

[3] S. Haykin. Neural Networks: A Comprehensive Foundation, 2nd ed., Upper Saddle River, NJ: Prentice-Hall 1999.

[4] L. Ljung. System Identification: Theory for the User, 2nd ed., Upper Saddle River, NJ: Prentice-Hall, 1999.

[5] J. Medina, N. Muller, and I. Roytelman. Demand Response and Distribution Grid Operations: Opportunities and Challenges, IEEE Transactions on Smart Grid, vol.1, no.2, pp.193-198, 2010.

[6] M. Mordjaoui and B. Boudjema, Forecasting and Modeling Electricity Demand Using ANFIS Predictor, Journal of Mathematics and Statistics 7 (4): 275-281, 2011.