

A Spike Train Neural Network Model for Short-term Electricity Load Forecasting

K Santosh¹⁺ and Sishaj P Simon²

¹ PG Research Scholar, Department of EEE, NIT-Trichy

² Assistant Professor, Department of EEE, NIT-Trichy

Abstract. Short term load forecasting (STLF) is very important because of its essentiality in power system planning and operation. Due to the restructuring of power markets, several electric power companies need to forecast their load requirement with maximum accuracy. Though various forecasting models are developed, accurate forecasting with zero error is still a challenging task. Here in this paper, the forecast engine of the proposed strategy is a spike train based neural network. The Spiking Neural Network (SNN) uses the neural code of precisely timed spikes and exploits the rich dynamics in its structural representation. Therefore, the proposed load forecasting system is known as the SNNSTLF (Spiking Neural Network Short-term Load Forecaster). The suitability and the validation of the proposed SNNSTLF model is investigated and compared with back-propagation artificial neural network (BPANN). The method is tested based on historical load data of Hyderabad.

Keywords: STLF, ANN, SNN, Spike Response Model (SRM).

1. Introduction

Electricity demand forecasting is mainly classified into three ranges: short, medium and long-term forecasts. Each range of forecasts has its own merits to the utilities. The short-term load forecast (STLF) is useful in determining electricity price in the deregulated markets, commitment of generating units, etc., medium range forecasting is required for fuel procurement, maintenance scheduling and diversity interchanges; whereas the long-term forecasting is essential for system expansion planning and financial analysis. Prediction of the system load over an interval usually from one hour to one week is called short term load forecasting. It has become more important after power system deregulation since forecasted load is used by the operators to determine market prices of next day. However, STLF is very difficult due to the various factors influencing the dynamic nature of load consumption. First of all, the load series is complex and exhibits several levels of seasonality. Secondly, load at any particular hour is not only dependent on the load at previous hour but also load at the same hour of the previous day. As indicated above, STLF is an important area and this is reflected in the literature with many traditional and non-conventional techniques like regression analysis [1], time series approach [2], neural networks [3-4], and machine learning techniques [5] etc.

The artificial neuron that has found closeness to its biological counterpart is known as Spiking Neuron and the network based upon this type of neurons is known as Spiking Neural Network (SNN). This was first proposed as Hodgkin-Huxley [6] model in 1952 however, poses several problems for use such as encoding/decoding of information, control of information flow and computational complexity. However, recently researchers are trying to find its usefulness in various engineering application. SNN in recent times have found its application for pattern recognition and classification purposes such as electricity price forecasting etc. [7].

⁺ Corresponding author. Tel.: +91-431-2503250; fax: +91-431-2503252.
E-mail address: santosh.kulkarni1988@gmail.com.

In this paper, a general methodology based on approximate gradient descent based backpropagation ANN is used for SNN modeling [8] which is applied for load forecasting and compared with feed forward back-propagation ANN. Section 1 gives an introduction about load forecasting and the justification in applying SNN to STLF. The details about SNN, spike propagation algorithm and the proposed model for STLF are given in section 2. Section 3 discusses about the historical data, the results of testing and training of SNN. The conclusions and scope of the future works are presented in section 4. Acknowledgement and References are mentioned in section 5 and section 6 respectively.

2. Spiking Neural Network

The SNN are the third generation artificial neural networks which are more detailed models and use the neural code of precisely timed spikes. The input and the output of a spiking neuron are described by a series of firing times known as the spike train. One firing time means the instant the neuron has fired a pulse. The potential of a spiking neuron is modeled by a dynamic variable and works as a leaky integrator of the incoming spikes: newer spikes contribute more to the potential than the older spikes. If this integrated sum of the incoming spikes is greater than a predefined threshold level then the neuron fires a spike. Though its likeness is closer to the conventional back propagation ANN, the difference lies in having many numbers of connections between individual neurons of successive layers as in Figure.1. Formally between any two neurons ‘i’ and ‘j’, input neuron ‘i’ receives a set of spikes with firing times t_i . The output neuron ‘j’ fires a spike only when membrane potential exceeds its threshold value θ . The relationship between the input spikes and the internal state variable is described as ‘Spike Response Model’ (SRM) [8].

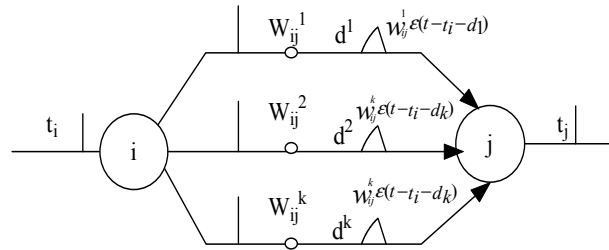


Figure.1 Connectivity between i^{th} and j^{th} with Multiple Delayed Synaptic Terminals

The internal state variable $x_j(t)$ is described by the spike response function $\epsilon(t)$ weighted by the synaptic efficacy. i.e., the weight between the two neurons is given as

$$x_j(t) = \sum_i \sum_{k=1}^m Y_i^k W_{ij}^k \quad (1)$$

Where, W_{ij}^k is the weight of the sub-connection k , and Y gives the un-weighted contribution of all spikes to the synaptic terminal. Y is given as

$$Y_i^k(t) = \epsilon(t - t_i - d^k) \quad (2)$$

Where, m is the number of synaptic terminals between any two neurons of successive layers, d^k is the delay between the two nodes in the k^{th} synaptic terminal and $\epsilon(t)$ is the spike response function shaping the Post Synaptic Potential (PSP) of a neuron which is as follows,

$$\epsilon(t) = \frac{t}{\tau} e^{-\frac{t}{\tau}} \quad (3)$$

Where, τ is the membrane potential decay time constant.

2.1. Training Mechanism of SNN Model

The step by step procedure of the SNN learning algorithm for a 3 layer network with a single hidden layer is given below.

Step 1: Prepare an initial dataset normalized between 0 and 1.

Step 2: Generate the weights randomly to small random values. Initialize the SNN parameters such as Membrane potential decay time constant (τ) and learning Rate (α).

Step 3: For the proposed SNN architecture with 3 layers, input layer (i), hidden layer (h) and output layer (o), do the steps 4-8 'itt' times (itt=no of iterations).The network indices represents the number of neurons and are given below

i=number of neurons in the input layer. Here i=1, 2, 3.....p.

h=number of neurons in the hidden layer. Here h=1, 2, 3....q

j=number of neurons in the output layer. Here j=1, 2, 3.....n.

k=number of synaptic terminals. Here k=1, 2, 3.....m.

Step 4: For each pattern in the training set apply the input vector to the network input.

Step 5: Calculate the internal state variable of the hidden layer neurons.

$$x_h(t) = \sum_i \sum_{k=1}^m Y_i^k W_{ih}^k \quad (4)$$

Step 6: Calculate the network output which is given by the equation

$$x_j(t) = \sum_h \sum_{k=1}^m Y_h^k W_{hj}^k \quad (5)$$

Step 7: Calculate the error, the difference between the actual spike time and the desired spike time.

$$E = 0.5 * \sum_{j=1}^n (t_j^a - t_j^d)^2 \quad (6)$$

Where t_j^a is the actual output spike and t_j^d is the desired output spike and n is the number of neurons in the output layer.

Step 8: Adjust weights of the network in such a way that minimizes this error. Then repeat Steps 4-8 for each pair of input and output vectors of the training set until error for the entire system is acceptably low. The equations for change in hidden and output layer are given below. Change in weights from hidden layer (h) to output layer (o) is given as

$$\Delta W_{hj}^k = -\alpha \delta_j Y_{hj}^k(t_j^a) \quad (7)$$

$$\text{Where, } \delta_j = \frac{t_j^d - t_j^a}{\sum_{k,h} W_{hj}^k \left(\frac{\partial Y_{hj}^k(t_j^a)}{\partial t_j^a} \right)} \quad (8)$$

Change in weight from input (i) to hidden layer (h) is given as

$$\Delta W_{ih}^k = -\alpha \delta_h Y_{ih}^k(t_h^a) \quad (9)$$

$$\text{Where, } \delta_h = \frac{\sum_{j \in \Gamma^h} \delta_j \sum_k W_{hj}^k \left(\frac{\partial Y_{hj}^k(t_j^a)}{\partial t_h^a} \right)}{\sum_{k,i} W_{ih}^k \left(\frac{\partial Y_{ih}^k(t_h^a)}{\partial t_h^a} \right)} \quad (10)$$

Where, Γ^h is the set of immediate successors of hidden layer neurons.

2.2. Proposed SNN Model for STLF

The network model consists of 1 input layer, 1 hidden layer and 1 output layer. The most important issue in the design of SNN model is the selection of the input variables. In this paper, load pattern of consecutive days are correlated for training input and target data. For example, during training the load pattern of Monday is taken as input data and load profile of Tuesday is taken as target data. So when a test data is given as input, a day-ahead load profile is forecasted. In this paper, the input to the SNN network consist a total of 41 neurons representing the hourly load consumption on any day, the day of the week, month number and week number. The topology of the network included following variables as inputs to the proposed SNN to forecast load at any hour h consists: one day Lagged load (L_h-1), one day lagged minimum and maximum temperature values (W-1), actual historical maximum and minimum temperatures on the day of forecast (W),

day of the week encoded in binary form (001 to 111), month number: (0001 to 1100) and week number (000001 to 110100).

3. Results and Discussion

The program for SNN is coded on MATLAB version 2010a. The hardware configuration of the computer used is Intel Core I5 processor with 2.53 GHz CPU, 4 GB RAM and the operating system used is Windows 7 Home Premium.

3.1. Setting of Control Parameters

The proposed forecast strategy has the following adjustable parameters: Number of hidden nodes (h), weights (W_{ih} and W_{hj}), learning rate (α) and decay time constant (τ). Mean Square Error (MSE) is considered as the performance index for adjustment of the control parameters. The range for each of the parameter is selected as $W \in [0.01, 3]$, number of hidden layer neurons $h \in [12, 30]$, number of synaptic connections $k \in [10, 20]$, decay time constant $\tau \in [4, 8]$, and $\alpha \in [0.0005, 0.01]$. Initially, training samples are constructed based on the data of 100 days training period. Keeping the other parameters constant, within the selected neighborhood, the weights from input to hidden layer and hidden to output layer are varied and the training process is stopped when MSE obtained is minimum. Now with the weights being set, similarly the other parameters are varied with respect to the order of their precedence. The best possible result for decay time constant is obtained when the coding interval is equal to the decay time constant. For high values of α , the MSE obtained after SNN training is very large and did not converge. Therefore, small values of α are considered. The final combination of parameters that provide the best results are $W_1 \in (1, 2)$, $W_2 \in (2, 3)$, $h=20$, $k=16$, $\tau=5\text{ms}$ and $\alpha=0.0006$. The accuracy of the forecasting results is evaluated by Mean Absolute Percentage Error (MAPE). Definition of MAPE is as follows:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{P_i - A_i}{A_i} \right| \quad (11)$$

Where, P_i and A_i are the i^{th} predicted and actual values respectively and n is the total number of predictions.

3.2. Numerical Results

The effectiveness of the proposed SNNSTLF model is evaluated by conducting 2 experiments and comparing the results obtained with BPANN. A BPANN based model has been designed for comparative study. The training set for all the case studies included hourly data from 1st November 2009 to 31st October 2011. In the first case study, complete hourly load data for the two years is included in the same training set. However in the second case study, the SNN is first trained with data for the first year (from 1st Nov 2009 to 31st October 2010) and then for the second year i.e., from November 2010 to October 2011. This type of training mechanism is called alternative training strategy (ATS) [9]. The test set included hourly load demand from December 1st 2011 to 14th December 2011. The historical data set of temperature is obtained from website www.wunderground.com.

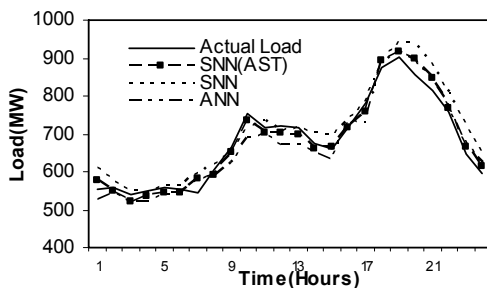


Figure.2 8th December 2011

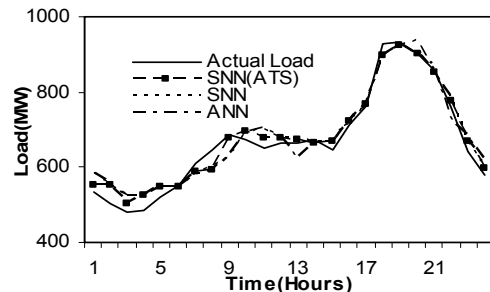


Figure.3 13th December 2011

Figures 2 and 3 compare the results obtained for all the 3 methods with the actual demand profile. The MAPE obtained from the SNN (AST) method for Figure.2 and Figure.3 are 2.06% and 2.27% respectively, while for SNN method they are 2.11% and 2.29%. As the load demand increased from the year 2009 to 2011, the AST method helped the SNN to learn the tendency of change in load. Therefore, more accurate results

are obtained when the proposed SNN model is implemented using AST. From Table 1, it is obvious that the proposed SNNSTLF model with ATS results in lowest MAPE when compared to normal spike propagation algorithm and BPANN.

Table 1 Performance Comparison for each day of a week

December 1/01/2011 to 7/01/2011	SNN MAPE (%)	SNN with ATS MAPE (%)	BPANN MAPE (%)
Thurs	2.34	2.25	3.12
Fri	2.42	2.25	3.19
Sat	2.35	2.21	3.25
Sun	2.67	2.45	3.57
Mon	2.49	2.32	2.89
Tue	2.19	2.15	3.66
Wed	2.29	2.23	3.48
Average	2.36	2.26	3.30

4. Conclusion

In this paper, a new methodology is proposed for STLF. Two different case studies for STLF are discussed to evaluate the performance of the proposed SNNSTLF model. The results obtained are compared with a feed forward BPANN. The results obtained are very encouraging and can be applied to real time applications in a large scale.

5. Acknowledgment

The authors would like to express their gratitude to the Andhra Pradesh Central Power Distribution Corporation Limited for providing the real time load demand data.

6. References

- [1] A.D. Papalexopoulos and T.C. Hesterberg. A Regression Based Approach to Short-Term System Load Forecasting. *IEEE Transactions on Power Systems*. 1990, **5** (4): 1535-1544.
- [2] M. T. Hagan and S. M. Behr. The time series approach to short-term load forecasting. *IEEE Transactions on Power Systems*. 1987, **2** (3): 785-791
- [3] H. S. Hippert, C. E. Pedreira, and R. Castro. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*. 2001, **16**(1): 44-55.
- [4] T. Senjyu, P. Mandal, K. Uezato, and T. Funabashi. Next day load curve forecasting using neural network. *IEEE Transactions on Power Systems*. 2005, **20**(1): 102-109.
- [5] B. J. Chen, M. W. Chang, and C. J. Lin. Load Forecasting using Support Vector Machines: A Study on EUNITE competition 2001. *IEEE Transactions on Power Systems*. 2004, **19** (4): 1821-1830.
- [6] A. L. Hodgkin and A. F. Huxley. A Quantitative Description of Ion Currents and its Applications to Conductance and Excitation in Nerve Membranes. *Journal of Physiology* (London). 1952, **117**(4): 500-544.
- [7] V. Sharma. D. Srinivasan. A Spiking Neural Network Based on Temporal Encoding for Electricity Price Time Series Forecasting in Deregulated Markets. *Proceedings of Neural Networks International Conference: Barcelona, Spain*. 2010, pp. 1-8.
- [8] Sergio M. Silva and Antonio E. Ruano. Application of Levenberg-Marquardt Method to the Training of Spiking Neural Networks. *Proceedings of International Conference on Neural Networks and Brain: Vancouver, BC, Canada*. 2006, pp. 3978-3982.
- [9] Daniel Ortiz-Arroyo, Morten K. Skov and Quang Huynh. Accurate Electricity Load Forecasting with Artificial Neural Networks. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation: Vienna, Austria*. 2005, **1**, pp.94-99.