ISBN: 978-1-948012-16-4

Contemporary Issues in Computing (CIC)

DOI: <http://doi.org/10.26480/cic.01.2020.76.79>

CrossMark

SHORT TERM LOAD FORECASTING USING OPTIMIZED TIME DELAY NEURAL NETWORK

Nilanjana Kashyap^{a*}, Partha Kayal^a, Saheli Ray^a^aDepartment of Electrical Engineering, NIT Silchar, Assam 788010, India*Corresponding Author e-mail: nilanjana@rs.ee.student.nits.ac.in

This is an open access article distributed under the Creative Commons Attribution License CC BY 4.0, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ARTICLE DETAILS

Article History:

Received 26 October 2020
Accepted 27 November 2020
Available online 03 December 2020

ABSTRACT

Load forecasting is an essential element for reliable and effective operation of power system. It assists in taking conclusive decisions regarding various aspects of power system which includes energy generation and purchasing, load switching, infrastructure development etc. Predicting electrical load demand with minimum error is vitally important for planning and operation of a utility company. With the introduction of deregulated power market, the significance of short term load forecasting is growing faster. Number of techniques have been presented in past few years for prediction of load. Among them artificial neural network has been emerged as popular one due to its ability in mapping complex non-linear relationships. This paper proposes a model based on time delay neural networks in which the delay parameters are being optimized with differential evolution technique. The proposed model targets to reduce error between the forecasted result and actual output. The efficiency of the presented model has been tested by real time load data.

KEYWORDS

Short term load forecasting, Time delay neural network, Differential evolution.

1. INTRODUCTION

In competitive power market, the principal aim of the utilities is to supply uninterrupted electricity to the customers. On the outset of this scenario, future demand analysis has become very crucial. Therefore, load prediction is one of the essential tasks for power suppliers, financial organizations, independent system operators (ISO) and other participants of power market. Different time horizon forecasts are being used depending on the nature of power system operations. The forecasting job has become more challenging in now a-days due to the introduction of deregulated energy market and more complicated consumption pattern of the end consumers.

In recent years, different conventional techniques have been utilized for short term load forecasting (STLF). But the main drawback is that inability to represent complex nonlinear correlation between electrical power demand and a set of stochastic factors influencing on it. Since 1980s, artificial neural networks (ANNs) have been broadly used in load forecasting due to its ability to give better result by handling all nonlinear relationships among various input variables (Bashir et.al, 2009). Nonlinear relationships of the inputs are realized by learning from training data.

Although considerable volume of literature has been observed on load forecasting research using ANN. However, most of the papers (Hippert et al., 2001; Banda et al., 2007; Bashir et al., 2009; Taylor, 2009; Farhat et al., 2015) mainly used back propagation (BP) training algorithm for various load forecasting model. But there is a limitation of BP that it can acquire knowledge of only static patterns of an input-output mapping that are independent of time. Mainly two methods are used to overcome this problem: time delay neural network (TDNN) and recurrent network.

TDNN is very complex as it needs to store delays and error back-propagated signals for every neuron and entire time delays (Kim et al., 2005).

The motivation of this work is to model and predict load demand using a hybrid method using TDNN along with differential evolution (DE) to obtain reduced training error. Performance of the network will be enhanced when optimized time delay parameters are selected. Hence, suitable methods are needed to select time delays for TDNN to get better accuracy. Here, DE algorithm is used to optimize time delays for the TDNN model.

This paper is arranged as follows. In section 2, there is a brief description about various load forecasting schemes. Section 3 explains the methodology of the proposed approach. Results and discussions are given in section 4. In section 5, the paper is finally concluded.

2. LOAD FORECASTING

Load forecasting is a method to estimate the load demand using past load data and weather information. The load demand forecasting is an important research topic in electrical engineering. It is divided into following three major classes: short-term load forecasting which are mainly done from one hour to one week, medium term load forecasting where the period is from a week to a month and long-term load forecasting predicts the electrical load for one or more than one years.

There are two methods which are mainly used for medium-term and long-term load forecasting: end-use and econometric approach. For STLF many methods e.g. similar day approach, regression models (Kyriakides et al., 2007), time series (Heiko et al., 2009), neural networks, fuzzy logic

Quick Response Code	This paper was presented at	Access this article online	
	International Conference on Contemporary Issues in Computing (ICCIC-2020) - Virtual IETE Sector V, Salt Lake, Kolkata From 25th-26th July 2020	Website: www.intelcomp-design.com	DOI: 10.26480/cic.01.2020.76.79

(Rustum et al., 2009), expert systems and statistical learning algorithms are mostly used (Heiko et al., 2009). Mean square error (MSE), root mean square error (RMSE) and mean absolute percentage error (MAPE) are used as measure of performance for each pair of training set at a particular mode.

$$MSE = \frac{1}{n} \sum_{t=0}^n [z(t) - z'(t)]^2 \quad (1)$$

$$RMSE = \left[\frac{1}{n} \sum_{t=0}^n (z(t) - z'(t))^2 \right]^{1/2} \quad (2)$$

$$MAPE = \left[\frac{1}{n} \sum_{t=0}^n \frac{z(t) - z'(t)}{z'(t)} \right] \times 100\% \quad (3)$$

n = number of samples

$z(t)$ = actual result at time t

$z'(t)$ = forecasted result at time t

3. METHODOLOGY

The goal of this paper is to carry out STLF for a particular facility using MATLAB and the facility studied is the Silapathar Substation Assam, Feeder No-01 (Silapathar town feeder). Hourly load data has been collected for 6 months from 07.04.2019 to 07.10.2019. For STLF 6 months' hourly load data is sufficient to train the network. These historical load data and time for 24 hours are fed as input into the ANN and after proper training it predicts the load demand for 24 hours. The main aim of this work is to find minimum training error (MSE) to get best performance for the forecasted output. The flow chart of proposed TDNN-DE method is shown in Fig. 1.

3.1 Neural Network matrix formulation

TDNN architecture is used to carry out STLF in this paper. It is a multi-layer, feed-forward network architecture comprising of input layer, hidden layer and output layer. In TDNN time delays d are used to the input vector $y(k)$ which produce a series of input to the network. Equation (4) shows the input-output relation of TDNN.

$$z(t+1) = f(w_x [y(t)y(t-1)....y(t-d)]^T + b_1) \times w_y + b_2 \quad (4)$$

where z is the output, w_x and w_y are weight matrices and b_1, b_2 are bias vectors. These two are variable parameters and are updated during training algorithm in order to reduce error between output and target (Molina et al. 2011).

The first step of load forecasting using NN is to load the data. After that the data are normalized and converted into a time series. Then inputs and targets vectors of the network are configured. Three different matrices of input and target for training, testing and validation of the model are formed. Out of all the input data 70% are used for training, 15% for testing and remaining 15% for validation. For different delay sets for different training, testing and validation errors are obtained. Since most of the data are used in training, improvement of training performance is taken into consideration i.e. to reduce MSE between training output and training target by varying the value of delays.

3.2 DE optimization

In this work, DE is used to find the optimal delay set for which the MSE value of training error is minimum. The optimization performance depends on DE parameters.

In 1997, DE algorithm was first proposed by Price and Storn. It is a good evolutionary algorithm based on natural biological evolution mechanism (Storn et al., 1997). Due to its simple principle and robustness, DE can be successfully used to various types of optimization problems, e.g. in image classification, neural network (Storn et al., 1997), constrained global optimization (Kim et al., 2007), images segmentation and other areas.

The main steps involve in differential evolution (DE) algorithm are:

1. Initialisation of total population number, maximum number of iteration, cross-factor and scale factor. Population of every iteration P consist of N chromosomes. Let at generation P , chromosome j is represented as

$$\chi_{jP} = (\chi_{jP}^1, \chi_{jP}^2, \dots, \chi_{jP}^q) \quad j = 1, 2, 3, \dots \quad (5)$$

2. Initialisation of the population.

3. (a) Mutation operation: The formula for generation of new individual can be illustrated by following equation

$$\chi_{r1,P+1} = \chi_{r1,P} + F * (\chi_{r2,P} - \chi_{r3,P}) \quad (6)$$

$r1, r2$, and $r3$ are chromosomes in the range $[1, N]$. F is the variation factor. It is a real number between $[0, 1]$ and controls amplification degree of the differential variable.

3. (b) Crossover operation: Crossover or recombination produces new individual from existing population. It introduces diversity to the new population. After crossover new individuals are represented as

$$\chi_{j,P+1} = (\chi_{1j,P+1}, \chi_{2j,P+1}, \dots, \chi_{Dj,P+1}) \quad j = 1, 2, \dots \quad (7)$$

Where,

$$\chi_{jl,P+1} = \begin{cases} V_{jl,P+1} & \text{if } (rndb(j) \leq CP) \text{ or } (j = nbr(l)), \\ V_{jl,P} & \text{if } (rndb(j) > CP) \text{ or } (j \neq nbr(l)) \end{cases} \quad j = 1, 2, \dots \quad (8)$$

$l = 1, 2, \dots, D$

$rndb(j)$ is uniformly distributed within the range $[0, 1]$, CP is the crossover probability between $[0, 1]$ and $nbr(l)$ is random integer within the range $[0, D]$.

3.(c) Selection Operation: The individual which are generated by genetic operations are compared with target individual. Then individuals with best fitness value are selected.

$$\chi_{j,P+1} = \begin{cases} U_{j,P} & \text{if } (f(U_{j,P}) > f(\chi_{j,P+1})), \\ \chi_{j,P+1} & \text{if } (f(U_{j,P}) \leq f(\chi_{j,P+1})) \end{cases} \quad (9)$$

where f is the fitness function.

4. Steps 3(a)-3(c) are repeated till the termination criterion is met.

Figure 1: Flow chart of TDNN-DE approach

4. RESULT AND DISCUSSION

Total number of samples or hourly load data are 4392 because 6 months' hourly load data are used in the model ($30 \times 24 + 31 \times 24 + 30 \times 24 + 31 \times 24 + 31 \times 24 + 30 \times 24 = 4392$). The data are normalized within the range 0-1. The input to the network is hourly load demand. The networks are simulated by separating the input data to training, validation and test sets. This means out of 4392 hourly load data, 3052-hourly load data is used for training, 720-hourly load data (30 days) is used for validation and 168-hourly load data (1 week) is used for test sets. Levenberg–Marquardt (trainlm) supervised training algorithm and ten hidden layers are used to train the network.

Two cases are considered, firstly STLF using TDNN and secondly STLF using TDNN-DE. Finally, the result shows the comparison of both the cases.

4.1 Case 1

In this case, TDNN with 5 delays are used for load forecasting and training performance obtained is 0.02816, Fig. 2.

Figure 2: Training error with TDNN

Figure 3: Training error with respect to no. of maximum delays applied

4.2 Case 2

For the same network, based on trial and error method no. of delays are changed and corresponding training performance is checked. It is seen that the training performance or error (MSE) is decreased with increase in no. of delays and it is almost constant (0.03) after 10 delays, as can be observed in Fig. 3. Hence maximum 10 delays are used i.e. 10 sets of input data. Then DE is used to optimize these delay sets of TDNNs within the range [1,1340]. The main aim of using DE is to find the minimum training error for different sets of 10 delays within that range. Fig. 4 shows good convergence property of DE algorithm. For TDNN-DE approach the minimum MSE value of training error obtained is, 4.0686×10^{-5} for the delay set = [198 261 125 1330 744 346 29 1146 1179 1016], as shown in Fig. 3.

Figure 4: Convergence of DE algorithm

For the same delay set the validation, test, overall performance of the TDNN-DE approach is observed as shown in Fig. 5, 6, 7 respectively. Overall performance of TDNN-DE approach is reduced to 0.0023713 from 0.028085 in TDNN.

Figure 5: Validation set performance

Figure 6: Test set performance

Figure 7: Overall performance of the ANN model

Figure 8: Validation performance of training, validating, and testing sets

The training will be stopped by the validation set before at a particular number of iteration if further training of input set will degrade the performance (Bashir et. al., 2009). Thus, performance of test vector can be used to check the ability of a network to generalize beyond primary and validation vectors. Fig. 8 shows training, test and validation set error patterns. All three curves have similar features. This means the ANN model gives better accuracy.

Table 1: Comparison of TDNN with TDNN-DE		
	TDNN	TDNN- DE
Hidden layers	10	10
Transfer function	Trainlm	Trainlm
Training error (MSE value)	0.02816	4.068×10^{-5}
Validation error (MSE value)	0.027983	4.3365×10^{-5}
Test error (MSE value)	0.020724	3.4053×10^{-5}
Overall performance (MSE value)	0.028154	0.0023699

5. CONCLUSION

This paper demonstrated the ability of a TDNN to predict hourly load using DE technique to adjust the delays of the model for better accuracy. The main goal of this work is to minimize the error between forecasted result and target. The model uses past load hours' data as input variables. The results suggest that with appropriate delays in input vector, the

performance of TDNN is improved. The performance of TDNN-DE is better than TDNN in aspect of precision. In further analysis, other factors such as weather information can be imposed on the proposed model. Also, performance of TDNN with other optimization techniques can be checked.

REFERENCES

- Banda. E, Folly K. A., 2007, "Short Term Load Forecasting Using Artificial Neural Network". Power Tech, 2007 IEEE, Laussane
- Bashir Z. A, M. E. El-Hawary. M. E, Applying Wavelets to Short-Term Load Forecasting Using PSO-Based Neural Networks, IEEE Transactions on Power Systems, VOL. 24, NO. 1, FEBRUARY 2009
- Farahat. M.A, Elgawed. A.F. Abd, Ibrahim. A, Mustafa. H. M. M., 2015, "Short Term Load Forecasting Using Neural Networks and Particle Swarm Optimization." Journal of Electrical Engineering, pp 1-10.
- Feinberg Eugene A., Genethliou Dora, "Load Forecasting, chapter 12" State University of New York, Stony Brook Eugene.
- Heiko Hahn, Silja Meyer-Nieberg, Stefan Pick, 2009 Electric load forecasting methods: Tools for decision making, European Journal of Operational Research 199 (2009) 902-907.
- Hippert Henrique Steiner, Pedreira Carlos Eduardo Pedreira, Souza Reinaldo Castro, 2001 Neural Networks for Short-Term Load Forecasting: A Review and Evaluation, IEEE Transactions on Power Systems, Vol. 16, No. 1, February 2001
- Kim H. K., Chong J. K., Park K. Y., Lowther D. A., 2007, "Differential evolution strategy for constrained global optimization and application to practical engineering problems," IEEE Transactions on Magnetics, vol. 43, no. 4, pp. 1565-1568, 2007.
- Kim H., Shin K., Park K. (2005) Time Delay Neural Networks and Genetic Algorithms for Detecting Temporal Patterns in Stock
- Kyriakides, E., Polycarpou, M., 2007. Short term electric load forecasting: A tutorial. In: Chen, K., Wang, L. (Eds.), Trends in Neural Computation, Studies in Computational Intelligence, vol. 35. Springer, pp. 391-418 (Chapter 16).
- Markets. In: Wang L., Chen K., Ong Y.S. (eds) Advances in Natural Computation. ICNC 2005. Lecture Notes in Computer Science, vol 3610. Springer, Berlin, Heidelberg.
- Molina Diogenes, Liang Jiaqi, Harley Ron, Venayagamoorthy Ganesh K., 2011, Comparison of TDNN and RNN performances for neuro-identification on small to medium-sized power systems, 2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG), April 2011.
- Rustum Mamlook, Omar Badran, Emad Abdulhadi, 2009, A fuzzy inference model for short-term load forecasting, Energy Policy 37, pp. 1239-1248.
- Storn R., Price K., 1997 "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341-359.
- Taylor Eric Lynn, 2013 "Short-term Electrical Load Forecasting for an Institutional/Industrial Power System Using an Artificial Neural Network." Master's Thesis, University of Tennessee.
- Tee Chin Yen, Cardell. Judith B, Ellis. Glenn W., 2009, Short-Term Load Forecasting Using Artificial Neural Networks. 41st North American Power