A hybrid "k-means, VSS LMS" learning method for RBF network in short-term load forecasting

Ehsan Mostafapour[1], Mehdi Panahi[2], and Morteza Farsadi[1]

[1]Department of electrical and computer engineering, Urmia University, Urmia-Iran
Ehsan_mostafapour@yahoo.com, m.farsadi@urmia.ac.ir
[2]Saveh Azad University, Saveh, Iran
mehtipanah @gmail.com

Abstract

In this paper we investigate the performance of a hybrid learning algorithm for RBF network in the application of short-term load forecasting. In this method the algorithm for finding radial basis function centers of hidden layer is k-means and the algorithm for training the weights of output layer is adaptive variable step-size algorithm. We proved this method is both accurate and fast in comparison with other presented schemes. Also we demonstrated that this method requires less computational processing and can perform well when amount of the input data is large. Our simulation results show there is up to 30 percent improvement in processing time and 37% improvement in prediction accuracy when compared with previously improved k-means learning.

Keywords: RBF network, load forecasting, variable step-size LMS algorithm, hybrid learning

1. Introduction

Short-term load forecasting or briefly STLF is a vital part of power distributing system. Multilayer neural networks are used for a long time in this task and this is due to their ability of predicting and simulation of nonlinear relations between input data and desired outputs [1]-[6]. Various factors contribute in changing the load demand of a power grid; weather conditions and date information are examples of them. Some of these factors may cause nonlinear relation functions between input data and prediction load size. Adaptive algorithms cannot trace these nonlinearities, but with the help of multilayer networks this goal is reached. In [2] a hybrid learning scheme is proposed in order to surpass the accuracy and processing rate of multilayer perceptron and ordinary RBF networks. For this approach two adaptive algorithms were presented, LMS and RLS algorithms. RLS algorithm is much faster and accurate than LMS algorithm, but this comes with the price of more processing requirements. When input data size is huge, which is a fact in load forecasting task, this processing requirement can lead to more and more computational demand and computer upgrades which are not economical. To this reason we need a fast and comparable algorithm in accuracy with RLS. In this paper we presented variable step-size LMS (VSSLMS) algorithm [7] and proved that this method can perform faster by choosing a proper step-size value automatically and in this way can compete with RLS. The rest of this paper is organized as follows:

In section 2 we will examine the factors affecting the power demand in a network. In section 3, we review common methods used in load forecasting. In section 4, RBF neural network and the learning algorithms used in it are reviewed and then we present our hybrid learning scheme, and explain the new algorithm presented in this paper. In Part 5 numerical simulations are presented Section 6 contains our concluding remarks.

2. Factors affecting the network load

Various factors influence the power demand of the network. These factors can be divided into the following main categories:

* The weather

* The time

* Economic factors

* Random disturbances

Here we will review these factors briefly:

- Weather factor

Weather factors include temperature, humidity, rainfall, wind speed, cloud coverage and etc. changes in each of these factors can affect the power usage in a network. For example increase in temperature can force power users to turn on their air conditioners and this will increase network load demand. All in all temperature usually is considered an independent variable in load forecasting.

- Time factor

Amount of load usage for a user may differ by time factor during the day. For example the amount of electrical power demand is at peak when sun sets and home lights are on. The type of the day like being a holiday or a normal week day can affect power demand too. Periodicity is a feature of load curve, but in certain times this feature may not be correct. Load curve may stay nearly the same for same days, weeks or even years.

- Economical factors

Electrical power is a commodity and the economic situation of distribution area is important in using it. The degree of industrialization of the area and electrical power costs are examples of economic factors. In modern distribution networks, power costs change with time and amount of usage and users will want to avoid peak times of electrical costs.

- Random disturbances

Modern power distribution networks have a large number of consumers. Although it is not possible to predict each user's power demand, the overall load usage of all users has good statistical features and we can extract mildly changing load curves from them. But sometimes the startup and shutdown of large loads like steel mills can add abrupt changes in load of the network. These kinds of sudden changes are usually predictable but not always.

3. Common methods for STLF

Some of the main methods STLF are as follows:

- Regression method

Regression method is a statistical procedure and it is based on modeling relation between load values and other features like weather changes. Engel et al. [12] have proposed several regression models for forecasting load of a day ahead.

- Time series method.

The time series method is used for a long time in load forecasting. ARMA, ARIMA and ARIMAX models are from popular schemes in load forecasting. ARMA model is used for modeling stationary random processes. ARIMA model is the extended version of ARMA and can model non-stationaries. ARMA and ARIMA models only use time and load variables as input but ARIMAX model can use weather variables too. Chu et al. [15] and fan et al. [14] explained the usage of AIRMAX model in load forecasting.

- Neural Networks

Artificial neural networks (ANNs) are used pervasively in load forecasting from 1990s. These networks have the ability to model nonlinear relations and this feature can be used in load forecasting for relating input data with output load value. Several neural structures have been presented true these years of which we can name MLP (multiple layer perceptron) and RBF (radial basis function) networks. The prediction in these networks has two steps: 1. Training and 2. Test. After training step with known input and output data, we ask network to predict output load for a given day by using input features.

4. RBF neural network

The RBF network uses radial basis functions to model classification. It has 3 layers: 1) the input layer and has neurons that are associated with the input signal. 2) Hidden layer which handles the image (Mapping) is the input signal to the new environment, and 3) the output layer which handles relations between signals input is detected. The structure of RBF network is given in Figure 1.

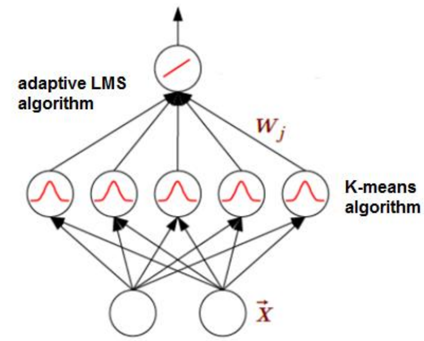

Fig.1. RBF network and its layers

Gaussian function is a commonly used radial basis function but other functions can be used in this application. Gaussian function is given as below:

$$\varphi(x_i) = exp\left[-\frac{1}{2\sigma_i^2}\|x_i - t_i\|\right] \qquad (1)$$

Where x_i are training samples, σ_i variance or width of the Gaussian functions, t_i the central points of function (i) in the hidden layer and $\|x_i - t_i\|$ distance of samples from the center of the functions. The function $\phi(x_i)$ is maximum at t_i and reduces as the distance increases from center. For a given input data x_i the function only has significant amount in a small range, so if the input is closer to the center, the output of neurons in the hidden layer will be larger. Mapping of data from the input layer to the hidden layer is done by function $\phi(x_i)$ and thus we have a non-linear transformation, but the data transfer takes place from the middle layer to output layer linearly:

$$Y = \sum_{i=1}^{m} w_i \phi(x_i) \qquad (2)$$

RBF network learning starts with selecting the appropriate radial function (in this study, Gaussian) and finding its variance by the following equation:

$$\sigma = \sqrt{\frac{d_{max}}{m+1}} \qquad (3)$$

Where d_{max} is maximum distance between the inputs and (m) is the number of neurons or nodes in the hidden layer. So far, several methods for learning the hidden layer of RBF network is presented, some of which are self-organizing map algorithm (self-organizing algorithm), random algorithms and (K-means) algorithm. Self-organizing and randomized algorithms are suitable only for the static and dynamic teaching methods to be variable and online (online) should be used. Here we briefly review k-means algorithm:

4.1 K-means algorithm

The nonlinear mapping ability of RBF neural network is reflected in the hidden layer and its properties are mainly determined by the centers of radial basis functions. Many methods can determine the centers but we commonly use K-means clustering algorithm or its improved scheme. The complete explanation of k-means is given in [5] so we will only have a brief review of this algorithm:

K-means clustering algorithm is a widely used heuristic classification method in clustering analysis; it is a simple and fast Algorithm idea: k-means algorithm divides the data set into k clusters based on the input parameter k. The algorithm uses the iterative update approach: In each round, we divide the points around into k clusters based on k reference respectively, and the centroid of each cluster (the average of all points in this cluster, that is, the geometric Center) will be used as the new reference point for the next round of iteration. Iteration makes the selected reference point get closer and closer to the true cluster centroid, so that clustering results getting better. The main steps of this algorithm are:

1) Select arbitrary k data objects as initial cluster centers from the n data objects;

2) Calculate the distances between each object and each cluster center, assign the object to the nearest cluster;

3) Re-calculate the centers of k clusters after the distribution of all objects is completed;

4) Compare with the previous k-cluster centers, if the new cluster centers change, turn 2), Otherwise turn 5);

5) Output the clustering result.

4.2 Hybrid “K-means, LMS” algorithm

It should be noted that the relationship between the hidden layer and the output of RBF network is a linear mapping and adaptive algorithms can be used to update the weights of this layer. The LMS algorithm is used in this paper for this purpose. We consider the linear regression model [2]:

$$d(n) = W^T(n)\phi(x_i) + e(n) \qquad (4)$$

Then we have:

$$e(n) = d(n) - W^T(n)\phi(x_i) \qquad (5)$$

$$W(n+1) = W(n) + 2\mu e(n)\phi(x_i) \qquad (6)$$

In these relations d(n) is the desired predicted load which is known in learning step, $W(n)$ is the vector of weights w_i and $\phi(x_i)$ is the vector of values of $\varphi(x_i)$. $e(n)$ is the detection error and step size μ is here assumed to be constant. This algorithm updates output weights in order to minimize difference between network outputs and desired predicted values.

So here we have the following two steps:

1) Using unsupervised algorithms (k-means) to obtain variances centers and central functions.

2) Using the LMS algorithm to update the weights of the output layer.

Such training scheme is called hybrid learning because it is a combination of the K-means and adaptive LMS algorithms. This method has much faster training speed and accuracy than other methods because the need for propagation of the error among layers for weight training is eliminated. Also, because the LMS algorithm is used for classification we do not have to worry about being trapped in the local minimums and will be sure that by selecting an appropriate step-size we will find the optimal point of prediction.

4.3 “K-means, variable step-size LMS” algorithm

For the weight-training task of the output layer instead of fixed step size LMS algorithm, we used the variable step size LMS that is proposed in [7]. This algorithm is defined as follows:

$$W(n+1) = W(n) + 2\mu(n)e(n)\Phi(x_i) \qquad (7)$$

$$\mu(n) = \alpha\mu(n-1) + \gamma e^2(n) \qquad (8)$$

Where α and γ are the controlling parameters. In this application, α and γ are selected 0.997 and 0.0002 respectively. The authors of [8] showed that the proposed VSS algorithm of [7] has the optimum performance among all variable step-size algorithms.

5. Numerical Simulation

In this section we consider the performance of presented learning scheme in load forecasting task. We will have an overview of train and test data and present simulation results. The multiple input structure of network is due to the multiple dimensionalities of input data which is resulted by combining load amount with weather and time variables. The network is trained with input load data of the first two months of autumn according to our country calendar and is asked to predict the load amount of one day ahead (first day of the third month of autumn). As we mentioned, input data consists of weather and time features and network can learn the relations between these features and load amount. In this paper input data have six features: maximum and minimum temperature, rain fall and load of time (t) in 3 past days. Weather data can be obtained from [16].

5.1 Data preprocessing

In training step of RBF network input data have different dimensions and each of input data features has different scale. For this reason it is necessary to preprocess these data in order to ease network training and increasing speed. First of all we must normalize input features between [0, 1]. Also bad and irrelevant data must be removed by comparison. Bad data is sometimes collected by errors in measurements and have very high or very low quantities. By considering the type of the day, on which we want to have prediction, in terms of temperature, rainfall and the load of previous days we can be sure of finding relations between features and load amount.

After preprocessing and learning steps, it is time to forecast or predict the load amount for a certain time in future. We do this task for prediction of the load for 24 hours ahead. We used MATLAB software for our simulations and estimations. Overall flowchart of used load forecasting system is given in figure 2:

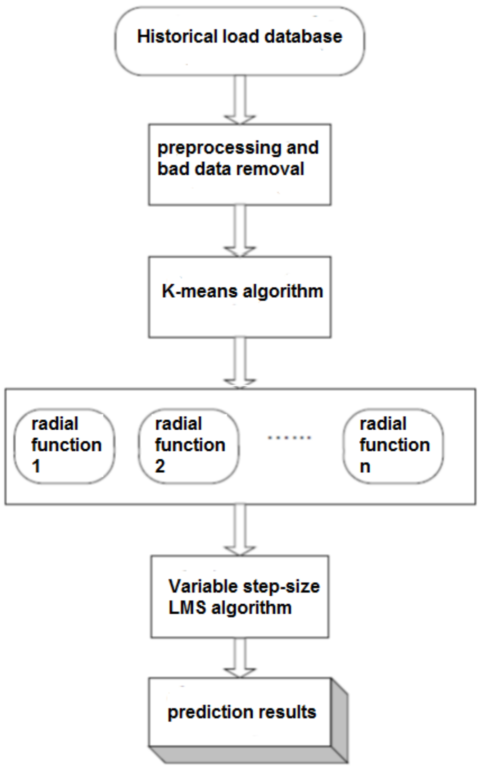

Fig.2. Flowchart of used algorithm

5-2 Simulation results

In this paper we used the hourly load data of Urmia city-Iran from 23 September to 21 November as training set for our simulations and tried to predict the load amount of 22 November. This city has an average power demand of 500 MWs and has a peak of about 750 MWS in summers. The power usage in this area never goes under 300MWs. The prediction is performed by RBF network with hybrid learning scheme and K-means algorithm of [5]. We expect that our presented RBF network will increase accuracy of prediction. The criterion of comparing the performance of prediction is MAPE (Mean absolute percentage error) which is given as below:

$$MAPE = \left(\frac{|x_i - y_i|}{x_i}\right) \times 100 \qquad (9)$$

Where x_i is the actual data amount in time i and y_i is the predicted value. The results of comparison among presented and similar networks are given in table 1. For this comparison we run the simulations with RBF network and 2 types of K-means and presented learning algorithm. The results for other algorithms are reported by respected papers and we did not test them.

It is important to mention that in some papers the prediction error of the network is reported in training step on which the test and train data are the same. This leads to some very low error values which are not credible. In table 2 we presented the comparison between actual and forecasted hourly load:

Table 1. Comparison of the performances of load forecasting methods

Forecasting type	MAPE Percentage:
SVM method [6]	3.03 %
Method of [10]	1.57 %
Method of [5]	0.8184 %
Presented hybrid learning scheme	0.511 %

Table 2. Comparison between actual and forecasted load

Time/h	Actual load/MW	Forecasted load/MW	MAPE (%)
1	427.4128	429.7228	0.540
2	394.1482	391.8382	0.586
3	375.2088	372.8988	0.615
4	367.3943	369.7043	0.628
5	364.4380	362.1280	0.633
6	365.2043	362.8943	0.632
7	384.3272	386.6372	0.601
8	409.2146	406.9046	0.564
9	437.2225	434.9125	0.528
10	474.8592	477.1692	0.486
11	488.8567	491.1667	0.472
12	485.4856	487.7956	0.475
13	478.5562	476.2462	0.482
14	457.8284	455.5184	0.504
15	447.8852	450.1952	0.515
16	453.3834	455.6934	0.509
17	467.4663	465.1563	0.494
18	550.3238	552.6338	0.419
19	585.8657	583.5557	0.394
20	570.4170	572.7270	0.405
21	552.4272	550.1172	0.418

22	531.7868	534.0968	0.434
23	505.1452	507.4552	0.457
24	485.1674	487.4774	0.476

In figure 3 we can see the load curve of actual and forecasted load. Figure 4 presents a comparison between the performance of k-means RBF network [5] and our network. We can see that the mean value of error for our network is lower than that of k-means RBF network.

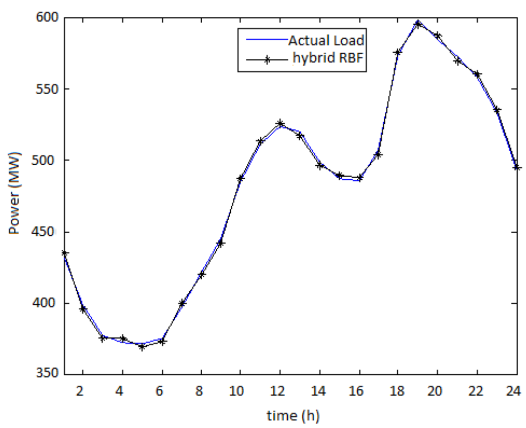

Fig.3. Comparison between actual and forecasted load.

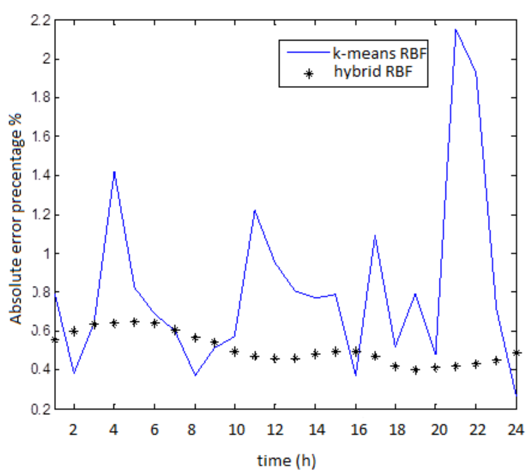

Fig.4. Comparison between presented method and method of [5].

6. Conclusion

In this paper we studied the performance of RBF networks in load forecasting of Urmia city and presented a new hybrid learning scheme which is faster and precise than previously presented schemes. Our contribution to this task is presenting variable step size LMS to the hybrid 'k-means, LMS' algorithm. This network can be used in tasks other than load forecasting. Our simulation results show that the mean value of MAPE for our network is 0.51 which is about 37% better than 0.81 performance of network in [5]. Also our network is about 30 percent faster in convergence than k-means RBF and MLP networks. This improvement in prediction rate can be useful when amount of training data is high or when we want to predict the load values of very short time in future like hours or minutes ahead. In future works we will extend the applications of presented network and try to compare it with more neural networks.

7. References

[1] A. Khotanzad, R. Afkhami-Rohani and D. Maratukulam, "ANNSTLF Artificial neural network shortterm load forecaster generation three, " *IEEE Trans. on Power Syst., 13, 4, 1413–1422, November, 1998.*

[2] Simon Haykin, *Neural networks and learning machines*. Prentice Hall, pp. 230-263, 2008

[3] I. Moghram and S. Rahman, "Analysis and evaluation of five short termload forecasting techniques," *IEEE Trans. Power Syst., vol. 4, no. 4, pp. 1484–1491, Nov. 1989.*

[4] C.N. Lu, H.T. Wu, and S. Vemuri, "Neural network based short term load forecasting," *IEEE Trans. Power Syst., vol. 8, no. 1, pp. 336–342, Feb. 1993.*

[5] N. Dongxiao et al. "Improved RBF network applied to short-term load forecasting," *IEEE* 2nd International Conference on Software Engineering and Service Science, pp. 864 – 867, 2011

[6] Z. Fan, S.; Chen, L., "Short-Term Load Forecasting Based on an Adaptive Hybrid Method," *IEEE Trans. Power Syst.* 2006, *pp.* 392.

[7] R. H. Kwong, and E.W. Johnston, "A variable step size LMS algorithm, " *IEEE Trans. Signal Process., vol. 40, no. 7,pp. 1633-1642, 1992.*

[8] C.e.G. Lopes and J.e.M. Bermudez, "Evaluation and design of variable step size adaptive algorithms," *Proc. IEEE ICASSP, Salt Lake City, UT, May 2001, pp. 3845-3848.*

[9] J.W. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *IEEE Trans. Power Syst., vol. 17, no.3, pp. 626–632, Aug. 2002.*

[10] N. Amjady, F. Keynia, "A New Neural Network Approach to Short Term Load Forecasting of Electrical Power Systems," *Energies Journal, vol. 4, pp. 488–503, March 2011.*

[11] H.S. Hippert, C.E. Pedreira, and R.C. Souza, "Neural networks for short term load forecasting: A review and evaluation," *IEEE Trans. Power Syst., vol. 16, no. 1, pp. 44–55, Feb. 2001.*

[12] R.F. Engle, C. Mustafa, J. Rice, 'Modeling peak electricity demand', *Journal of Forecasting, 1992, 11, 241 - 251*

[13] O. Hyde, P.F. Hodnett, 'An Adaptable automated procedure for short-term electricity load forecasting', *IEEE Transactions on Power Systems, 1997,12, pp. 84 - 93*

[14] J.Y. Fan, J.D. McDonald, 'A real-time implementation of short – term load forecasting for distribution power systems', *IEEE Transactions on Power Systems, 1994, 9, pp. 988 - 994*

[15] M. Y. Cho, J.C. Hwang, C. S. Chen, "Custumer short-term load forecasting by using ARIMA transfer function model," Proceedings of the International conference on Energy Management and power delivery, EMPD, 1995, 1. pp. 317-322.

[16] Weather data is obtained from website: www.weather.com.