A Practical Approach for Electricity Load Forecasting

T. Rashid and T. Kechadi

Abstract—This paper is a continuation of our daily energy peak load forecasting approach using our modified network which is part of the recurrent networks family and is called feed forward and feed back multi context artificial neural network (FFFB-MCANN). The inputs to the network were exogenous variables such as the previous and current change in the weather components, the previous and current status of the day and endogenous variables such as the past change in the loads. Endogenous variable such as the current change in the loads were used on the network output. Experiment shows that using endogenous and exogenous variables as inputs to the FFFB-MCANN rather than either exogenous or endogenous variables as inputs to the same network produces better results. Experiments show that using the change in variables such as weather components and the change in the past load as inputs to the FFFB-MCANN rather than the absolute values for the weather components and past load as inputs to the same network has a dramatic impact and produce better accuracy.

Keywords—Daily Peak Load Forecasting, Feed Forward and Feed Back Multi-Context Neural Network.

I. INTRODUCTION

THE electrical load forecasting in energy power plant systems is the most significant task because it secures the reliability and reduces the operational cost of the plant. The daily peak load determines the operational scheme and scheduling for the next day. Different techniques have been implemented by researchers to solve the load forecasting task. However, two techniques are widely used, namely; regression and time series. The regression technique [1 and 2] is based on finding the functional relationship between weather components and the load demand. Therefore the load is affected by the weather components that were used in the regression. A disadvantage of this technique is that the relationship between the weather components and the load demand is not stationary but rather depends on spatial-temporal components and the regression technique is unable to address this temporal variation [1]. The time series technique [3] is a type of regression and therefore it has the same problem.

T. Rashid Author is in University College Dublin, Computer Science Belfield, Dublin 4, Ireland.
T. Kechadi Author is in University College Dublin, Computer Science, Belfield, Dublin 4, Ireland.

This technique takes a load pattern as a signal in a time series and forecasts the future load. In other words, the future load is only a function of the previous loads. The absence of weather components which strongly effect the energy consumption result in the forecasting being inaccurate and unstable especially when there is a drastic change in the environment (sociological variables) [1]. The ARMA models are the best example of this technique which assumes that the future load at any particular time can be estimated by a linear combination of a few previous times. In this paper exogenous and endogenous input variables that affect the load are mapped non-linearly to the load using artificial neural networks. The use of these networks allows for the avoidance of the limitations of the techniques described above by employing the non-linear modeling and adaptation.

A. Neural Networks

The proposed FFFB-MCANN as depicted Figure 1 is based on the feed forward neural network and the simple recurrent network (SRN) architectures [4]. Our Feed Back-Multi Context Artificial Neural Network (FB-MCANN) overcomes the limitation of SRN [5]. In this paper a simple modification is done on the FBMCANN by dividing its hidden layer into two parts. The first hidden layer acts as a feed forward to the output layer and the second hidden layer acts as both feed forward and feed back to the output layer and context layers respectively. This is called FFFB-MCANN. This simple modification will improve the speed of the training session due to a reduction of the recurrent connections.
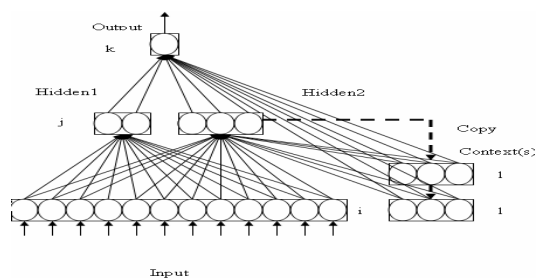

Fig. 1 The FFFB-MCANN Network

B. Learning Algorithm

Neural networks are commonly categorized in terms of their corresponding training algorithms: fixed weight, supervised and unsupervised. Supervised learning networks have been the mainstream of neural model development. The training

World Academy of Science, Engineering and Technology
International Journal of Energy and Power Engineering
Vol:1, No:5, 2007

data consists of many pairs of input/output training patterns. Therefore, the learning will benefit from the assistance of a teacher. Examples of this are FF-ANNs and the FB-ANNs [1, 4, and 5]. For an unsupervised learning rule, the training set consists of input training patterns only. Therefore, the network is trained without the benefit of teacher, such as the Kohonen network [6]. The Fixed weight networks as is suggested by its name, have fixed weights. No learning occurs, i.e. the weights cannot be adapted. An example of this type is the Hopfield network [7]. For supervised learning networks, there are several learning techniques that are widely used by researchers. The main three are real time; back propagation and back propagation through time, all of which were used for our FB-MCANN [5] depending on the application. In our application the data sequence length is specified, therefore we select the back propagation learning algorithm to train our recurrent network.

II. FORECASTING SYSTEM

The load-forecasting task depends on past and current information about variables that affect the load for a period of time. A Forecasting system can be carried out as follows: obtain and analyze the historical data; pre-processing and normalizing of the information; choosing the training and testing set; choosing the type of network and its parameters; choosing a suitable learning algorithm; and finally, implementation. A Futher detail is shown below:

A. Historical Data

Two historical data sets were collected to perform the forecasting task:

1. The first set which we term data set (A) was obtained from the EUNITE 2001 symposium, a forecasting competition. It reflects the behavior of the East Slovakia Electricity Corporation. This data recorded the load at half hour intervals every day from Jan 1997 to Jan 1999 and daily average temperature from Jan 1995 to Jan 1999.
2. The second set which we term data set (B), was obtained from the ESB Company. It reflects the behavior of the Electricity Supply Board in the Republic of Ireland. The data recorded the load, temperature, cloud rate, wind speed and humidity at fifteen minute intervals every day from Jan 1989 to Jan 1999

B. Training and Testing Data

The training and testing data sets for both data sets (A) and (B) were selected to perform the daily peak load forecasting. For each of which data set (A or B), the training set was a period from Jan 1997 to Dec 1998 and the test set was period of Jan 1999.

C. Input/Output Data Selection

Cross validation techniques were carried out in order to select the appropriate input data for the network. The future load in this paper is a function of the availability of significant variables in data sets.

For data set (A) the future load is a function of the calendar, the status of the day (social events), the past and current change in the temperature T and past change in the load L. The future load in the data set (B) is a function of the calendar, the status of the day, the past and current change in the weather components (such as temperate T, cloud rate C, wind speed W and humidity H) and past change in load L. Further details are shown below:

1. The future load for data set (A) was as function of:

$$\Delta L_t = f(\text{ past and current calendar; past and}$$

current social events; $\Delta T_t, ..., \Delta T_{t-n}; \Delta L_{t-1}, ..., \Delta L_{t-n}$)

2. The future load for data set (B) was as function of

$$\Delta L_t = f(\text{ past and current calendar; past and}$$

current social

events; $\Delta T_t, ..., \Delta T_{t-n}; \Delta C_t, ..., \Delta C_{t-n}; \Delta W_t, ...,$

$\Delta W_{t-n}; \Delta H_t, ..., \Delta H_{t-n}; \Delta L_{t-1}, ..., \Delta L_{t-n}$)

t is the index of the day.

The change in load and the change in weather components (temperature, cloud rate, wind speed, humidity) can be described as follows:

$$\Delta L_t = L_t - L_{t-1} / L_{t-1}; \Delta T_t = T_t - T_{t-1}; \Delta C_t = C_t - C_{t-1}$$

$; \Delta W_t = W_t - W_{t-1}; \Delta H_t = H_t - H_{t-1};$

The input selections for data set (A) were the following 12 inputs:

- Four binary input neurons I_1 to I_4 as the index of the month.

- Three binary input neurons I_5 to I_7 as the index of the day of the week. Thus the network can identify the seasonal periods of the year and can also distinguish the days with high temperature from those with low temperatures.

- One binary input neuron I_8 to find out whether the forecasted day is a working day or a holiday.

- One binary input neuron I_9 to find out whether the day prior to the forecasted day was a working day or a holiday. Usually this will affect the next day's load.

- One input neuron for the change in the temperature between the current day and the previous day $I_{10} : \Delta T_t = T_t - T_{t-1}$.

World Academy of Science, Engineering and Technology
International Journal of Energy and Power Engineering
Vol:1, No:5, 2007

- One input neuron for the change in the temperature over the last two consecutive days $I_{11} : \Delta T_{t-1} = T_{t-1} - T_{t-2}$.

- And one input I_{12} for the change in the load over the last two days $\Delta L_{t-1} = L_{t-1} - L_{t-2} / L_{t-2}$.

Figure 2, shows a sample of input data selected from data set (A).

	Month			Week		today on/off	Yest. On/off	ΔT (t)	ΔT (t-1)	Δ L(t-1)
0	0	0	1	1	1	0	0	1	1.3	3.3	-0.0251
0	0	0	1	1	1	1	0	0	3.3	3.7	0.02574
0	0	0	1	0	0	1	0	0	3.7	-2.6	-0.0502

Fig. 2 Shows a sample of input data selected from data set (A) to the network

The input selections for data set (B) were 18 inputs as follows:

1. Include the items 1 to 4 above (which include input neurons I_1 to I_9).

2. One input neuron for the change in the temperature between the current day and the previous day I_{10} :
$$\Delta T_t = T_t - T_{t-1}$$

3. One input neuron for the change in the temperature over the last two consecutive days $I_{11} : \Delta T_{t-1} = T_{t-1} - T_{t-2}$

4. One input neuron for the change in the cloud rate between the current day and the previous day I_{12} :
$$\Delta C_t = C_t - C_{t-1}$$

5. One input neuron for the change in the cloud rate over the last two consecutive days $I_{13} : \Delta C_{t-1} = C_{t-1} - C_{t-2}$

6. One input neuron for the change in the wind speed between the current day and the previous day I_{14} :
$$\Delta W_t = W_t - W_{t-1}$$

7. One input neuron for the change in the wind speed over the last two consecutive days $I_{15} : \Delta W_{t-1} = W_{t-1} - W_{t-2}$

8. One input neuron for change in the humidity between the current day and the previous day I_{16} :
$$\Delta H_t = H_t - H_{t-1}$$

9. Input neuron for change in the humidity over the last two consecutive days $I_{17} : \Delta H_{t-1} = H_{t-1} - H_{t-2}$

10. And one input neuron for change in the load over the last two days $I_{18} : \Delta L_{t-1} = L_{t-1} - L_{t-2} / L_{t-2}$

Both data sets (A and B) have one network output neuron.

- The current change of the daily peak load, which is the difference between the forecasted daily peak load and the previous daily peak load, O_1 :
$$\Delta L_t = L_t - L_{t-1} / L_{t-1}$$

Inputs I_1 to I_9 are binary coded and inputs I_{10} to I_{18} be scaled between 0-1. One output to the network was also normalized between 0-1.

The forecasting in [8] takes only time and change in previous loads into account. Here, the change in weather components, the change in the load and the details of status of the day and the calendar have been taken as inputs to the networks. This sort of selection gives the network a dramatic improvement in terms of accuracy and stability. The average error of the network performance decreased from (3.87-4.55)% to (1.58-1.99)%. This is because the variation of the differences between the loads for 2 consecutive days is less than the differences between the loads factors themselves for 2 consecutive days. Thus the network takes inputs in time series with values that are close to each other. This allows the network to learn more easily than presenting the network with inputs whose values are not close. The same thing applies to the other variables (weather components). This is shown in Figures 3 (for variables relating to load and change in load).
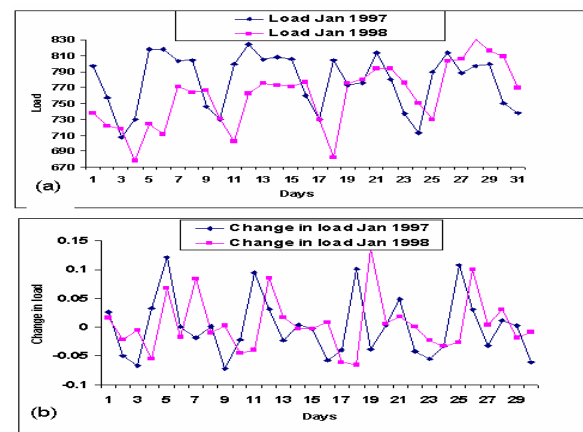

Fig.3 (a), is the daily peak load for the Jan 1997 and 1998,
(b), is the difference between daily peak load over consecutive days for Jan 1997 and 1998 (data set (A)).

D. Selection of Network Structure

For each data set (A and B), one network model was made with different parameters. This means for each data set no exception was made in terms of separate models for weekend,

World Academy of Science, Engineering and Technology
International Journal of Energy and Power Engineering
Vol:1, No:5, 2007

weekday, and holiday or even for the days with unusual behavior e.g. high temperature with load did not decrease and low temperature with load did not increase (no distinction was made for weekdays, weekends, winter season etc).

The FFFB-MCANN structure for data set (A) consisted of 12-1-2-2*2-1; 12 neurons, 1 neuron in the first hidden layer, 2 neurons in the second hidden layer, 2 context layers each of which has 2 neurons and 1 output neuron. The FFFB-MCANN structure for data set (B) consisted of 18-2-3-2*3-1; 18 neurons, 2 neurons in the first hidden layer, 3 neurons in the second hidden layer, 2 context layers each of which had 3 neurons and 1 output neuron. These parameters relied heavily on the size of the training and testing sets. Learning rates, momentum and the training cycles were varied. The type of activation function was a logistic function.

E. Cross Validation, Training and Testing

A simple algorithm was used to select the optimized parameters such as learning rate, momentum, hidden neurons and the threshold value to stop training. Assuming that our training set is called *TR* and testing set is called *TS*. The algorithm in general was as follows:

1. Invoke the training data set *TR* only.
2. Divide the training data set *TR* by n, so we have P_i validation set of data, for all $i = 1,2...n$ validation sets of data.
3. Let $P_i^{'}$ be the outcome of subtracting the P_i set from the *TR* set. Consider $P_i^{'}$ is a training set and P_i is validation set. For all $i = 1,2...n$.
4. Train the n networks independently, each with its training set $P_i^{'}$ and P_i test set. For all $i = 1,2...n$.
5. Compute the mean square error for each network $MSE_i, i = 1,2...n$.
6. Optimize each network parameter (such as hidden neurons, learning rate, momentum etc). Repeat step 4.
7. Choose the best performance amongst the networks in terms of prediction and accuracy from step 5. Save the best MSE_i and the best weight connections as the optimized network weight connections OW_i and mean square error $OMSE_i$.

Testing of the network can be done in two ways as follows:
1) Invoke the testing data set *TS*.
2) Load the network with the saved OW_i from above. Then, present the *TS* data set to the network. Obtain the forecasting results.

Or
1) Train the network with *TR*.
2) Stop the training when MSE of the network is equal to or less than the $OMSE_i$.

3) Present the *TS* data set to the network. Obtain the forecasting.

Lastly compare the forecasting results obtained from the two techniques above.

III. RESULTS

The performance of the training and the validation of the network are evaluated by computing the sum of MSE_i averaged over the number of training and validation sets using the equation below:

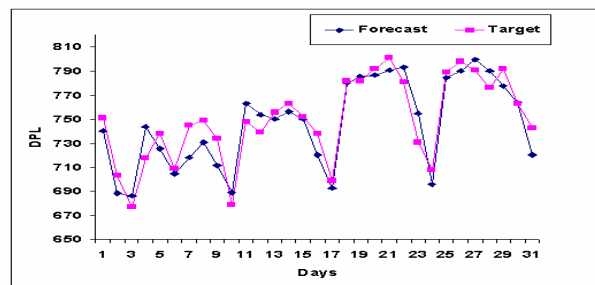$$MSE(perform.) = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

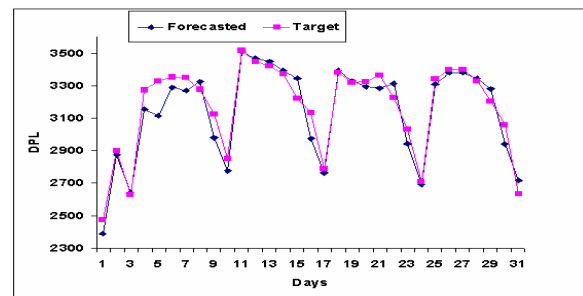Fig. 4 Displays the forecasting results for the data set (A)

Fig. 5 displays the forecasting results for the data set (B)

The performance of the network forecasting was evaluated with two measurement formulae, namely: The Mean Absolute Percentage Error (MAPE) and Maximum Error (MAX), as shown below:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{Lr_i - Lp_i}{Lr_i}\right|; MAX = \max(Lr_i - Lp_i)$$

Where n, is the number of outputs forecasted from the network, Lr, is the target value of the daily peak load, Lp_i, is the forecasted of the daily peak load and i, is the index of the day. Figure 4 displays the forecasting results of the network on data set (A). Figure 5 displays the forecasting results of the network on data set (B). It can be seen that the results of error forecasting using the cross validation of 10 training and testing sets as in Figure 6 for data set (A)), are quite similar to the results of error forecasting using the main training and testing sets as in Figure 7 for data set (A)).

World Academy of Science, Engineering and Technology
International Journal of Energy and Power Engineering
Vol:1, No:5, 2007

Training & validation.sets	Data set (A)			
	MAX	MAPE	TrainingError	ErrorTest
(P'1,P1) sets	60.30	2.21	1.05E-04	0.019051276
(P'2,P2) sets	48.30	1.81	1.10E-04	0.092288542
(P'3,P3) sets	39.24	1.88	1.12E-04	0.063174716
(P'4,P4) sets	25.89	1.65	1.14E-04	0.057318069
(P'5,P5) sets	35.65	1.52	1.17E-04	0.071417407
(P'6,P6) sets	40.69	1.54	1.11E-04	0.034073225
(P'7,P7) sets	42.24	1.96	1.33E-04	0.06424313
(P'8,P8) sets	33.94	1.47	1.02E-04	0.1064
(P'9,P9) sets	55.93	1.70	1.15E-04	0.1499
(P'10,P10) sets	42.05	1.72	1.28E-04	0.092254687

Fig. 6 Displays the results of various error performances of the n numbers of training and validation sets for the data set (A).

Data Set (A)				
Trainig & Test sets	MAX	MAPE	TrainigError	TestingError
	26.85	1.58	1.14E-04	0.027512874

Data Set (B)				
Trainig & Test sets	MAX	MAPE	TrainigError	TestingError
	214.37	1.99	9.06E-05	0.04511

Fig. 7 Displays the testing results of our network for both data sets.

IV. CONCLUSION

In this paper FFFB-ANNs are studied and used for daily peak electricity load forecasting. Two historical data sets have been used on our network. We summarized these main points:

- The disadvantage of using neural networks and more specifically recurrent networks for forecasting is that the design of these networks is very complex and it depends on good training. This mainly involves selecting optimized parameters; the most important are the inputs selection, hidden neurons etc. However neural networks and more specifically recurrent networks are more dynamic and flexible (because of their learning and weight adaptation capabilities) when compared with other statistical techniques.

- Our results show that exogenous and endogenous inputs to the network are better than just exogenous inputs to the network, as it is difficult for the network to learn when only exogenous inputs are presented into it.

- The main positive result of this paper is the demonstration that the change in weather components over time leads to better performance than using current absolute weather components, for the power plant peak load forecasting.

- The obtained results from our network using different data sets were steady and provided positive results. The results of our network training and testing were similar when verified with a simple cross validation algorithm. These similarity results for the algorithm proved that our technique is statistically stable.

- Given no distinction was made in terms of separate models for winter season, summer season weekend, weekday and holiday etc, our approach compares favorably with other techniques [9, 10, 11, 12 and 13]. Our technique obtained steadily ± 1.5% mean average percentage error.

REFERENCES

[1] D. Park, M. Al-Sharkawi, R. Marks, A. Atlas and M. Damborg, "Electric load forecasting using an artificial neural network," In *IEEE Tran. On Power Systems,* vol~ 6, no. 2, pp. 442-449, 1991.

[2] A. D. Papalxopoulos and T. C. Hiterbeg, "A regression-based approach to short-term load forecasting," In *IEEE Tran. On Power Systems,* vol~ 5, no. 4, pp. 1535-1547, 1990.

[3] G. Gross and F. D. Galianan, "Short-term load forecasting," In *Proceedings of the IEEE,* vol~75, no. 12, pp. 1558-1572, 1987.

[4] J. L. Elman, "Finding structure in time," In *Cognitive Science,* 14(2), 179--211, 1990.

[5] B. Q. Huang, T. Rashid and T.Kechadi, "A new modified network based on the Elman network," In *Proceedings of IASTED International Conference on Artificial Intelligence and Application,* Innsbruck, Austria, 2004..

[6] T. Kohonen, "Self-Organizing Maps". In Springer, Berlin, Heidelberg, 1995.

[7] J. J. Hopfield, "Learning algorithms and probability distributions in feed-forward and feed-back networks" In *Proceedings of the National Academy of Sciences of the USA,* 84:8429 - 8433, 1987.

[8] W. Charytoniuk, and Mo-Shing Chen, "Very Short-Term Load Forecasting Using Neural Networks," In *IEEE Tran. On Power Systems,* vol~ 15, no. 1, Feb, 2000.

[9] M. W. Chang, B. J. Chen and C. J. Lin, "Eunite network competition: Electricity load forecasting," In *EUNITE 2001 symposium,* a forecasting competition, 2001.

[10] D. Esp, "Adaptive Logic Networks for East Slovakian Electrical Load Forecasting," In *EUNITE 2001 symposium,* a forecasting competition, 2001.

[11] I. King and J. Tindle, "Storage of Half Hourly Electric Metering Data and Forecasting with Artificial Neural Network Technology," In *EUNITE 2001 symposium,* a forecasting competition, 2001.

[12] W. .Kowalczyk, "Averaging and data enrichment: two approaches to electricity load forecasting," In *EUNITE 2001 symposium,* a forecasting competition, 2001.

[13] L. .Lewandowski, F. Sandner and P. Portzel, "Prediction of electricity load by modeling the temperature dependencies," In *EUNITE 2001 symposium,* a forecasting competition, 2001.