STUDY OF ARTIFICIAL NEURAL NETWORKS FOR DAILY PEAK LOAD FORECASTING

TARIK RASHID AND TAHAR KECHADI
University College Dublin
Department of Computer Science
Belfield Dublin 4
Republic of Ireland

ABSTRACT

The idea is to forecast the daily electricity peak load for a month of a power station based on the data of previous years. The forecasting approach is carried out by using artificial neural networks (ANNs). In this paper two types of ANNs are studied and implemented for forecasting namely; the feed forward-ANNs and feed back-ANNs (FF-ANNs and FB-ANNs). Our modified network is part of the FB-ANNs family. It is called feed back multi context neural network (FB-MCANN) which introduced and invoked for this task. The back propagation learning algorithm is implemented for all networks. Exogenous input data variables and endogenous output data were imposed upon all networks. The experiments show that the FB-MCANN outperformed the FF-ANN. Another experiment also shows that using the change in temperature as an input to the FB-MCANN rather than the absolute temperature itself as an input to the same network can have a dramatic impact and produce better accuracy. This work also proposes steps to improve the future line of online load forecasting with the FB-MCANN.

KEYWORDS: ANNs, Feed FB-MCANN, Daily Peak Load Forecasting.

1. INTRODUCTION

The most important problem that power plants face is the prediction of future load. The prediction of energy demand ought to be very close to the actual energy demand. In order to operate the power plant efficiently the forecasting of energy usage must meet the actual future energy usage; therefore 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.

Several techniques have been implemented by researchers to solve 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. The 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 the temporal variation [1]. The time series technique [3] is a sort of regression, thus it has the same problem. 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 (weather components such as temperature, wind speed, wind dir., hum. and cloud rate do not have an influence on future load). The absence of weather components which strongly effect the energy consumption, result in the forecasting being inaccurate and unstable specially when there is a drastic change in the environment (sociological variables) [1]. The ARMA models are the best example of this technique which assumes the future load at any particular time can be estimated by a linear combination of previous few times. In this paper exogenous input variables that are affecting the load are mapped nonlinearly to the load using artificial neural networks that are able to perform non-liner modeling and adaptation which is a better option to avoid the limitation of above techniques.

2. ARTIFICIAL NEURAL NETWORKS

ANNs, as in [1 and 4], are information processing paradigms inspired by the way biological nervous systems, such as the brain, process information. ANNs are a form of multiprocessor computer system with simple process elements, a high degree of interconnection, simple scalar messages and adaptive interaction between elements: ANNs, are like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. In this paper two types of ANNs are studied namely; the FF-ANNs and FB-ANNs. Below is a further detail about these types of networks:

2.1 FF-ANNs

The FF-ANNs [1 and 4] as in figure 1 (a) consist of three interconnection layers; one input layer, one or more hidden layers and one output layer. The FF-ANNs allow signals to travel one way only; from the input to the hidden layer and then to the output layer. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. FF-ANNs tend to be straightforward networks that associate inputs with outputs.

2.2 FB- ANNs

The FB-ANNs can have signals travelling in both directions by introducing loops into the network. FB-ANNs are very powerful but slower than FF-ANNs, due to loops, and can get extremely complicated. The simple recurrent network (SRN) [5] as in Figure 1 (b) is an example of these networks. The SRN is widely used by researchers, nevertheless the network faces difficulties due to the architecture of the network itself: The architecture of the SRN network includes the network memory, which consists of one context layer (relatively small), the mapping of hidden layer neurons to the output layer neurons and an increased computation cost due to the need for more hidden neurons [6, 7, and 8].

2.3 FB-MCANN

The proposed FB-MCANN as in Figure 1 (c) is based on the SRN architecture. The FB-MCANN overcomes the limitation of the SRN. The network architecture consists of fully interconnected layers. The network has feedback connections from the hidden layer to context layers. This is a further improvement of the single-step recurrent network concept for sequential processing. The functionality of the network relies heavily on the context layers, which weight the influence of shorter or longer histories in the sequence differently [6, 7, 8, 9 and 10] and the forward connected context layers to both the hidden and output layers, which speed up the learning of the network and reduce the number of neurons in the hidden layer. These modifications enable the FB-MCANN to outperform the SRN and ensure the stability of the output, and provide it with the ability to memorize and adapt based on previous data [10].

3. 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 data consist of many pairs of input/output training patterns. Therefore, the learning will benefit from the assistance of a teacher, FF-ANNs and FB-ANNs are examples of this [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 any teacher. The network learns to adapt based on the experiences collected through the previous training patterns examples of this is Kohonen network [11]. And fixed weight, as suggested by its name has fixed weights, so that no learning occurs, that means the weights cannot be adapted; an example of this type is the Hopfield network [12].

For supervised learning networks, there are several learning techniques that are widely used by researchers. The main three are; the real time recurrent learning, back propagation and back propagation through time, all of which were used for our network [10] depending on the application. In this application the data sequence length is specified, therefore, we selected the back propagation learning algorithm to train our recurrent network to predict the daily peak load.

3.1 BACK PROPAGATION

The back propagation algorithm is an example of supervised learning [13]. It is based on the minimization of error by gradient descent. A new network is trained with BP. When a target output pattern exists, the actual output pattern is calculated. The gradient descent acts to modify each weight in the layers to reduce the error between the target and actual output patterns. The modification of the weights is accumulated for all patterns and finally the weights are updated.

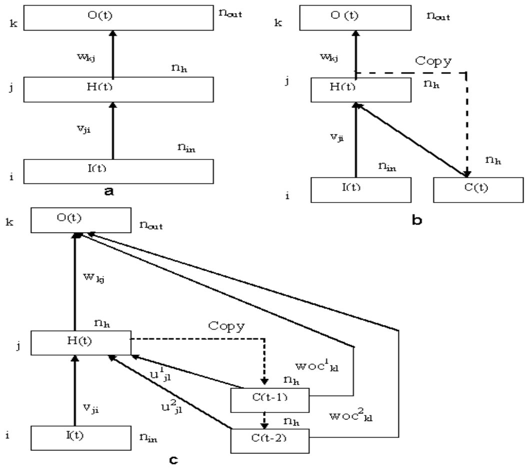

Figure 1: (a) is the FF-ANN, (b) is the simple recurrent network and (c) is the FB-MCANN.

4. LOAD FORECASTING SYSTEM

The load-forecasting task depends on the past and current information about variables that affect the load for a period of time. Our forecasting system can be processed as follows: obtain and analyze the historical data; pre-processing and normalizing information; choosing the training and testing set; choosing the type of network and it's parameters; choosing a suitable learning algorithm; and finally implementation.

4.1 HISTORICAL DATA

The data set for load forecasting was obtained from the EUNITE 2001 symposium, a forecasting competition. It reflects the behavior of the East Slovakia Electricity Corporation. The details of the historical data are shown in table 1.

Table 1: Details of the historical data.

Historical data	Period
Load	Every half hour interval from Jan1997-Dec1998. Daily peak load of Jan 1999.
Daily Average. Temperature.	from Jan 1995- Jan1999

4.2 TRAINING AND TESTING SETS

The training and testing data sets were selected to perform the forecasting as shown in table 2. Given the nature of our generic selection for the training set, our system is in fact able to predict any arbitrary month in 1999.

Table 2: Training and testing sets.

Data	Training set	Test set
Period	Jan 1997- Dec 1998	Jan 1999

4.3 INPUT/OUTPUT DATA

In accordance with the analysis in [14, 15, 16, 17 and 18] of the historical data in the literature, many concepts about both the application and ANNs were obtained. In addition several experiments using cross validation were carried out in order to select the appropriate input data to the networks. This paper displays only two cases to show how significant the input data selection factor is on the network's performance. The first case to consider is that the future load is a function of the calendar, the status of the day (social events) and the temperature. The second case to consider is that the future load is a function of the calendar, the status of the day (social events) and the change in the temperature. Further details of the two cases are shown below:

Case 1: Load= f (calendar, social events, temperature). Seven input neurons (I1-I7) to the networks: 1) The forecasted day of the year (I1:1 to 365), with one input neuron, the network can identify the seasonal periods of the year and also can identify the days with high temperatures from those with low temperatures; 2) The forecasted month (I2:1 to 12 / Jan to Dec), the network can figure out the month of the year and again can identify the days with high temperatures from low temperatures; 3) The forecasted day of the week (I3:1 to 7 / Sun-Mon); 4) A digit (I4: 0 or 1) for a working day or a holiday; 5) The daily average temperature of the forecasted day (I5:Td); 6) The daily average temperature of the day before the forecasted day (I6:Td-1); 7) And the daily average temperature for the same day as the forecasted day of the week before (I7:Td-7). One network output: The daily peak load (O1: DPL).

Case 2: Change of the load=f (calendar, social events, and change in the temperature). Five inputs (I1-I5) to the networks: 1) The forecasted day of the year (I1:1 to 365); 2) The forecasted month (I2:1 to 12 / Jan to Dec); 3) The forecasted day of the week (I3:1 to 7 / Sun-Mon); 4) A digit (I4: 0 or 1) to represent a working day or a holiday; 5) And the difference between the daily average temperature of the forecasted day and the previous day (I5: ∇ Td = Td2-Td1).

One network output: The difference between the forecasted daily peak load and previous the daily peak load (O1: ∇ DPL = DPL2-DPL1). All input/output data to the networks in both cases were normalized between zero and 1.

Case 2 is similar to [19], but it takes the change in temperature instead of just absolute load or even the change in load as input to the networks. This sort of selection gives the network a dramatic improvement in terms of accuracy and stability. This because the variation of the differences between the load/temperature for 2 consecutive days is less than the differences between the load/temperature factors themselves for 2 consecutive days. Thus the network takes inputs in time series with values that are close to each other. This makes the network learn more easily than presenting the network with inputs whose values are not close as shown in Figure 2.

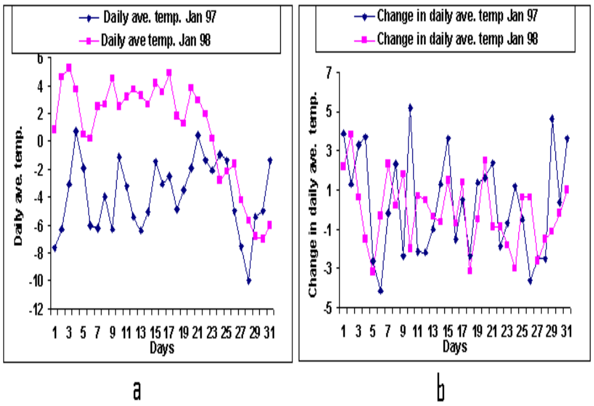

Figure 2: (a) is the daily average temperature for Jan 1997 and 1998; Figure (b) is the difference between daily average temperature of consecutive days for Jan 1997 and 1998.

4.4 STRUCTURES AND PARAMETERS

Several network structures were used with different parameters for both FF-ANN and FB-MCANN. A generic model was selected to include all the data (no exception was made in terms of separate models for weekend, weekday holiday and for the days with unusual behavior e.g. high temperature with load did not decrease or low temperature with load did not increase). There were only one network model for all days (no weekday, weekend networks neither day time slot's network models). The FF-ANN consisted of 7-15- 1 (7 input neurons, 15 hidden neurons, and 1 output neuron). The FB-MCANN case 1 structure consisted of 7-7-7*2-1 (7input neurons, 7 hidden neurons, two context layers each of which has 7 neurons and 1 output neuron). For the FB-MCANN case 2, the structure consisted of 5-5-5*2-1. Each network structure used relatively different network parameters. These parameters relied heavily on the size of training and testing sets. Learning rates and momentum were varied. The training cycles were also varied. The type of activation function was a logistic function. Table 3 shows the details of our selections.

For the FB-MCANN structures patterns of training data were divided into subsets. Each subset was presented to the FB-MCANN in a time series. Every time a pattern was presented to the model, the weight connection was modified and the history of the states was updated automatically and this was continued until the piece of data, when the stored activations in the context layers were cleared out. The same process was then repeated for the next subset. The processes were iterative. It is important to mention that a specific value of tolerance should be declared to stop training. This threshold was chosen so that it ensured that the model fitted to the training data, and it also did not guarantee good out-of-sample performance.

Table 3: this table shows the selection of network's structures and parameters for both cases (c1 and c2). α and β are the learning rate and momentum.

Network	ff-ann-c1	fb-mcann-c1	fb-mcannc2
Input. No	7	7	5
Hidden. no	15	7	5
Output. no	1	1	1
α and β	0.1- 0.3	0.005-0.02	0.005-0.02
Activ. fun.	Logistic	Logistic	Logistic
Epoch	30000	2000-20000	2000-20000

5. RESULTS

The performances of the networks were evaluated with two measurement formulae, namely: The Mean Absolute Percentage Error (MAPE) and Maximum Error (MAX), as in the equations (1) and (2):

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{lr_i - lp_i}{lr_i}\right| ----(1),$$

$$MAX = \max(|lr_i - lp_i|)----(2)$$

In the above equation n is the number of outputs predicted from the network, lr_i is the desired value of the daily peak load and lp_i is the predicted value of the daily peak load for the i^{th} day. Figure 3 and Table 4 show the predicted values for the month of Jan 1999, of FF-ANN with case 1 and FB-MCANN with both case 1 and case2. Table 5 shows the performance error for both networks.

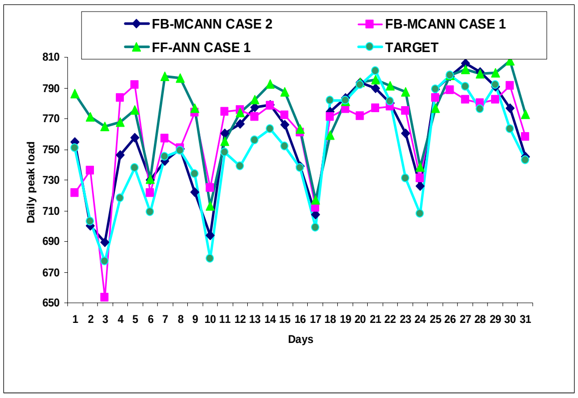

Figure 3: Prediction values for both networks are plotted against the target.

Table 4: This table shows the predicted values for Jan 1999 produced by FF-ANN for case 1 and FB-MC-ANN for cases 1 and 2.

day	ff-ann-c1	fb-ann-c1	fb-ann-c2	Target
1	786.61	721.83	754.86	751
2	770.99	736.4	699.92	703
3	764.95	653.51	689.18	677
4	767.5	783.67	746.18	718
5	775.91	792.13	757.57	738
6	730.67	721.57	729.8	709
7	797.71	756.96	742.16	745
8	796.69	750.73	750.46	749
9	776.86	774.12	722.2	734
10	713.04	725.07	693.77	679
11	755.34	774.73	760.47	748
12	773.67	775.48	766.58	739
13	782.32	770.97	777.36	756
14	792.76	778.36	778.96	763
15	787.41	772.16	766.12	752
16	763.01	760.72	738.8	738
17	717.04	711.69	707.26	699
18	759.48	771.18	774.73	782
19	781.07	776.12	783.7	782
20	793.66	771.51	793.63	792
21	795.15	776.78	789.6	801
22	791.2	778.06	779.96	781
23	787.49	774.99	760.31	731
24	738.52	731.39	725.96	708
25	776.93	783.44	788.99	789
26	798.33	788.73	797.55	798
27	802.01	782.58	805.85	791
28	799.27	780.2	800.22	776
29	799.61	782.46	790.63	792
30	807.56	791.19	776.83	763
31	772.77	758.2	745.47	743

Table 5: This table shows the values of MAPE and MAX over the days of the week for Jan 1999. The column Month shows the MAX and MAPE for the whole month Jan 1999.

MAPE	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Month
ff-ann-c1	3.07%	2.49%	2.03%	3.67%	3.61%	6.49%	6.22%	7.10%
fb-ann-c1	3.70%	3.54%	1.85%	1.79%	1.67%	4.60%	3.84%	5.25%
fb-ann-c2	1.63%	1.66%	1.96%	1.75%	0.57%	1.59%	1.90%	1.51%
MAX (MW)	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Month
ff-ann-c1	49.49%	37.9	26.32	52.7	47.69	67.98	87.94	87.94
fb-ann-c1	65.66%	54.12	20.48	24.22	29.16	43.98	46.06	65.66
fb-ann-c2	28.17%	27.57	21.35	24.21	14.11	29.31	17.96	29.31

6. CONCLUSIONS

In this paper FF-ANNs and FB-ANNs are studied and used for daily peak electricity load forecasting. Our network, the FB-MCANN outperformed the SRN [8] and the FF-ANNs as shown in table 4 and 5. The input data selection, when we used the change in the temperature between two consecutive days as input variable into the FB-MCANN as shown in case 2, had a key impact on producing higher accuracy against the FB-MCANN with current's, yesterday's and last week's temperatures as input variables to the FB-MCANN. However one may argue that the day related information that is given as an input of the network is very important. It is clear that if we are on Monday and want to forecast Tuesday load, the Monday's temperature is important. But knowing the day nearly gives the solution of the problem. Tuesday is like any other Tuesday, and last week's Tuesday temperature is known. This is what we do in case 1, while we included Td-7. However given the day of the week, knowing this day is working day or not is the most important. Therefore it is important to know how the past days are selected as input for the model. What would the behaviors of the FB-MCANN be when we include Td-7, in terms of increasing accuracy or not? Experiments in this way could be more fascinating. Finally the obtained results validate this approach and compare favorably with those examples mentioned in [14, 15, 16, 17 and 18]. The errors associated with each method heavily depend on homogeneity of data information, choice and size of training sets, the network's type and its parameters.

7. FUTURE WORK

We pursue our research work in order to develop the FB-MCANN to deal with electricity load forecasting by combination of online training and testing load. We intend to design a system where by the FB-MCANN can be trained online through adapting the training sets and updating the weight connections of the network. The back propagation is relatively unsteady and slow, and the FB-MCANN is slow too therefore we recommend using Optical Back Propagation (OBP), which is developed in [20]. This is very fast, and very efficient.

8. REFRENCE

[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] A.Khotanzad, R.Afkami, T.L.Lu, A.Abaye, M.Davis and D.J.Maratukulam, “ANNSTLF-A neural network-based electric load forecasting system,” *In IEEE Tran. On Neural networks*, vol~8, no. 4, pp. 835-846, 1997.

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

[6] Dit-Yan, K. W. Yeung, “A Locally Recurrent Neural Network Model for Grammatical Interface,” In Info. Processing Proceedings of the International Conference on Neural Information Processing, 1468-1473, 1995.

[7] W H. Wilson, “Learning Performance of Networks like Elman's Simple Recurrent Networks but having Multiple State Vectors,” In the 7th Australian Conf. on Neural Networks, Canberra, Australia1996.

[8] C. Ulbricht, “Multi-Recurrent Networks for Traffic Forecasting,” In International Conference on Artificial Intelligence, pp 883-888, 1994.

[9] G. Dorffner, “Neural Networks for Time Series Processing,” *In Neural Network World*, vol~6, no. 4, pp 447--468, 1996.

[10] 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.

[11] T, Kohonen. “*Self-Organizing Maps*”. In Springer, Berlin, Heidelberg, 1995.

[12] 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.

[13] P.J. Worbs, “Back propagation through time: What is dose and how to do it” In Preceding of IEEE, Volume 78, pp 1550-1560, 1990.

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

[15] D. Esp, “Adaptive Logic Networks for East Slovakian Electrical Load Forecasting,” In EUNITE 2001 symposium, a forecasting competition, 2001.

[16] I. King and John Tindle, “Storage of Half Hourly Electric Metering Data and Forecasting with Artificial Neural Network Technology,” In EUNITE 2001 symposium, a forecasting competition, 2001.

[17] W.Kowalczyk, “Averaging and data enrichment: two approaches to electricity load forecasting,” In EUNITE 2001 symposium, a forecasting competition, 2001.

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

[19] 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.

[20] M. Otair and W. A. Salameh “Optical Back-Propagation Neural Networks with Momentum Factor – A Case Study**”** In Proceedings of IASTED International Conference on Artificial Intelligence and Application, Innsbruck, Austria, 2004.