Proceedings of the V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 55–59, April 2–5, 2001 - Rio de Janeiro - RJ - Brazil

Short Term Load Forecasting via a Hierarchical Neural Model

Otávio A. S. Carpinteiro, Alexandre P. Alves da Silva, Agnaldo J. Rocha Reis
Instituto de Engenharia Elétrica, Escola Federal de Engenharia de Itajubá
Av. BPS, 1303 - Itajubá, MG, 37500-903, Brazil
E-mails: {otavio, alex, agnreis}@iee.efei.br

Abstract

This paper proposes a novel neural model to the problem of short term load forecasting. The neural model is made up of two self-organizing map nets – one on top of the other. It has been successfully applied to domains in which the context information given by former events plays a primary role.

The model is trained and compared to a Multi-Layer Perceptron load forecaster. It is required to compute the one to twenty four steps ahead recursive load forecasts. The paper presents the results, and evaluates them.

1. Introduction

With power systems growth and the increase in their complexity, many factors have become influential to the electric power generation and consumption (e.g., load management, energy exchange, spot pricing, independent power producers, non-conventional energy, generation units, etc.). Therefore, the forecasting process has become even more complex, and more accurate forecasts are needed. The relationship between the load and its exogenous factors is complex and non-linear, making it quite difficult to model through conventional techniques, such as time series and linear regression analysis. Besides not giving the required precision, most of the traditional techniques are not robust enough. They fail to give accurate forecasts when quick weather changes occur. Other problems include noise immunity, portability and maintenance[1]

Neural networks (NNs) have succeeded in several power system problems, such as planning, control, analysis, protection, design, load forecasting, security analysis, and fault diagnosis. The last three are the most popular [2]. The NN ability in mapping complex non-linear relationships is responsible for the growing number of its application to the short-term load forecasting (STLF) [3, 4, 5, 6]. Several electric utilities over the world have been applied NNs for load forecasting in an experimental or operational basis [1, 2, 4].

So far, the great majority of proposals on the application of NNs to STLF use the multi-layer perceptron (MLP) trained with error backpropagation algorithm. Besides the high computational burden for supervised training, MLPs do not have a good ability to detect data outside the domain of the training data. The Kohonen's self-organizing map (SOM) has been proposed to overcome these shortcomings [7]. However, the SOM does not work appropriately for regression problems, because it does not preserve the functionality of the regression surface [8].

In this work, an extension of the original self-organizing map to STLF is introduced. Reference [7] adds a new input variable to the original explanatory set in order to forecast based on the interconnection weight between the extra input and the winner in the neural grid. In the proposed approach there is no extra input, so that the dimensionality of the explanatory variables is preserved. The neurons in the output grid are associated with different load values, preserving the functionality of the regression surface as well. The proposed model has been applied to load data extracted from a Brazilian electric utility and compared to a MLP model.

This paper is divided as follows. In Section 2, load forecasting models are discussed. These models are compared through forecasting simulations in Section 3. Finally, Section 4 presents the main conclusions of this paper and indicates some directions for future work.

2. Load forecasting models

This section describes the load forecasting models.

2.1 The HNM

The model is made up of two self-organizing maps (SOMs). Its features, performance, and potential are better evaluated in [9, 10].

The input to the model is a sequence in time of m-dimensional vectors, $\mathbf{S}1 = \mathbf{V}(1), \mathbf{V}(2), ..., \mathbf{V}(t), ..., \mathbf{V}(z)$, where the components of each vector are non-negative real values. The sequence is presented to the input layer of the bottom SOM, one vector at a time. The input layer has m units, one for each component of the input vector $\mathbf{V}(t)$, and a time integrator. The activation $\mathbf{X}(t)$ of the units in the input layer is given by

$$\mathbf{X}(t) = \mathbf{V}(t) + \delta_1 \mathbf{X}(t-1) \qquad (1)$$

where $\delta_1 \in (0,1)$ is the decay rate. For each input vector $\mathbf{X}(t)$, the winning unit $i^*(t)$ in the map is the unit which has the smallest distance $\Psi(i,t)$. For each output unit i, $\Psi(i,t)$ is given by the Euclidean distance between the input vector $\mathbf{X}(t)$ and the unit's weight vector $\mathbf{W}_i$.

Each output unit i in the neighborhood $N^*(t)$ of the winning unit $i^*(t)$ has its weight W_i updated by

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + \alpha \ (i)[\mathbf{X}(t) - \mathbf{W}_i(t)] \quad (2)$$

where $\alpha \in (0,1)$ is the learning rate. (i) is the *neighborhood interaction function* [11], a Gaussian type function, and is given by

$$(i) = \kappa_1 + \kappa_2 e^{-\frac{\kappa_3[\Phi(i,i^*(t))]^2}{2\sigma^2}} \quad (3)$$

where κ_1, κ_2, and κ_3 are constants, σ is the radius of the neighborhood $N^*(t)$, and $\Phi(i,i^*(t))$ is the distance in the map between the unit i and the winning unit $i^*(t)$. The distance $\Phi(i',i'')$ between any two units i' and i'' in the map is calculated according to the maximum norm,

$$\Phi(i',i'') = max\left\{\left|l' - l''\right|, \left|c' - c''\right|\right\} \quad (4)$$

where (l',c') and (l'',c'') are the coordinates of the units i' and i'' respectively in the map.

The input to the top SOM is determined by the distances $\Phi(i,i^*(t))$ of the n units in the map of the bottom SOM. The input is thus a sequence in time of n-dimensional vectors, $\mathbf{S_2} = \Lambda(\Phi(i,i^*(1))), \Lambda(\Phi(i,i^*(2))), ..., \Lambda(\Phi(i,i^*(t))), ..., \Lambda(\Phi(i,i^*(z)))$, where $\mathbf{\Lambda}$ is a n-dimensional *transfer function* on a n-dimensional space domain. Λ is defined as

$$\Lambda(\Phi(i,i^*(t))) = \begin{cases} 1 - \kappa\Phi(i,i^*(t)) & \text{if } i \in N^*(t) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where κ is a constant, and $N^*(t)$ is a neighborhood of the winning unit.

The sequence S_2 is then presented to the input layer of the top SOM, one vector at a time. The input layer has n units, one for each component of the input vector $\Lambda(\Phi(i,i^*(t)))$, and a time integrator. The activation $\mathbf{X}(t)$ of the units in the input layer is thus given by

$$\mathbf{X}(t) = \Lambda(\Phi(i,i^*(t))) + \delta_2 \mathbf{X}(t-1) \quad (6)$$

where $\delta_2 \in (0,1)$ is the decay rate.

The dynamics of the top SOM is identical to that of the bottom SOM. Figure 1 shows the HNM.

The input data in the experiment consists of sequences of load data. Seven neural input units are required. The first unit represents the load at the current hour. The second, the load at the hour immediately before. The third, fourth and fifth units represent respectively the load at twenty-four hours behind, at one week behind, and at one week and twenty-four hours behind the hour whose load is to be predicted. The sixth and seventh units represent a trigonometric coding for the hour to be forecast, i.e., sin(2π.hour/24) and cos(2π.hour/24). The load data are linear transformed according to pre-specified minimum and maximum values. The normalized data ranges from 0 to 1. Each unit receives these normalized values.

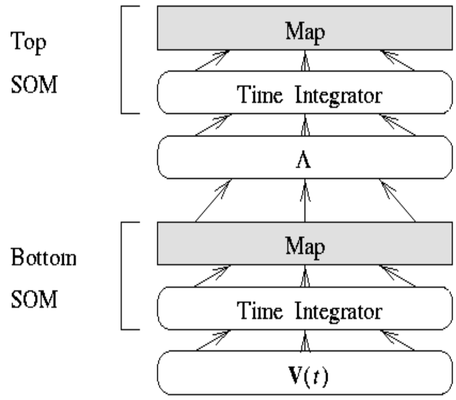

Figure 1: The HNM

2.2. The MLP model

One single hidden layer has been utilized. One to three hidden neurons have been used depending on the period of the year. Hyperbolic and linear activation functions have been adopted in the hidden and output layers, respectively.

The basic set of input variables correspond to the lagged values of the hourly load series (P) by 1h, 2h, 24h and 168h. The trigonometric coding for the hour to be forecast is used too (HS and HC).

The load values are preprocessed using two different techniques; ordinary normalization (minimum and maximum values in the [0;1] range) and single differencing. The differencing process computes the differences of adjacent values of a load series, i.e. the new series represents the variations of the original one. According to [12], adding differenced variables to that basic set of input improves the forecaster's performance.

A. MLP1 Model

The MLP1 model is the simplest one. It uses the basic set of input variables with ordinary normalization. The output variable is also normalized with the same procedure.

B. MLP2 Model

Differenced variables are added to the MLP1's set of inputs. Therefore, two time series are employed; the normalized and the differenced one. Differencing is applied to the normalized series. Only the first-order time differences (D) are considered. Fig. 2 illustrates the MLP2 load forecaster.

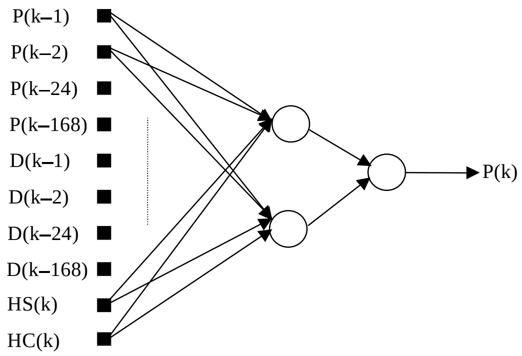

Figure 2: The MLP2 model

3. Tests

3.1 The HNM training process

Two different hierarchical neural models have been conceived. The first one is required to foresee the time horizon 1-6h. The training of the two SOMs of this model takes place in two phases – coarse-mapping and fine tuning. In the coarse-mapping phase, the learning rate and the radius of the neighborhood are reduced linearly whereas in the fine-tuning phase, they are kept constant. The bottom and the top SOMs were tested and trained respectively with map sizes of 15x15 in 700 epochs, and 18x18 in 850 epochs. It was used low values for decay rates - 0.4 and 0.7 for the bottom and top SOMs, respectively. According to [9], low decay rates reduce the memory size for past events, and that seems to be critical for predictions on distant horizons. The initial weights are given randomly to both SOMs.

Forecasting the remaining time leads (i.e., 7-24h) is addressed by the second model. The same training process previously described is applied to this model too. Nevertheless, medium values for decay rates - 0.5 and 0.8 for the bottom and top SOMs, respectively, were used instead. These new values for decay rates extend the memory size for past events [9], and consequently, yield more accurate predictions on large horizons.

One single model has been estimated to deal with the different days of the week.

3.2 MLP training process

Six-week windows have been taken for training (and testing), with data grouping according to the day of the week. For each day of the week, a MLP has been trained, applying the backpropagation algorithm with cross-validation. Different partitions for the training and testing sets are randomly created every 50 epochs. During the NNs training, there is no particular treatment for holidays. Special days have been excluded from the training set.

A comparison of HNM and MLP models have been performed. After the one-step ahead training, the one to twenty four steps ahead recursive load forecasts are computed. The load forecasters are retrained at the end of the day. The training window is moved one day forward, and the forecasts for the next 24 hours are performed (predictions always start at midnight). A load series from an electric utility in Rio de Janeiro has been used (maximum load around 3,900 MW). This validation procedure is repeated for one year.

The Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Mean Error (ME) and Maximum percentage error (MAX) has been used to evaluate the load forecasting models.

The following tables present the performance of the forecasters for 1-24 steps ahead predictions (Tables 1, 3), and a global average evaluation (Tables 2, 4). Figures 3 and 4 show actual and forecast loads for two particular days. The first one is a typical weekday (Friday, 02/03/1995). On the other hand, the second one is a special weekday (Tuesday, 02/07/1995).

The results from HNM are very promising, considering that no data segmentation and preprocessing has been applied to it.

Table 1: Hourly percentage error for 02/03/1995.

Time (h)	MLP1 error	MLP2 error	HNM error
1	0,92	1,63	1,70
2	0,28	1,62	0,99
3	1,92	1,73	2,19
4	3,57	1,20	0,70
5	4,48	0,68	0,68
6	3,97	1,67	0,71
7	2,70	2,40	1,47
8	2,32	2,54	0,76
9	5,12	1,02	2,26
10	4,78	2,26	0,94
11	2,79	0,83	0,74
12	1,37	0,41	2,37
13	2,50	0,48	3,90
14	2,92	3,61	4,01
15	4,51	0,96	3,19
16	4,58	0,06	3,89
17	0,54	3,84	5,36
18	3,48	3,26	2,49
19	3,75	0,05	2,24
20	1,31	0,41	2,74
21	1,62	0,00	0,75
22	0,96	1,80	2,77
23	1,16	2,62	4,12
24	1,91	2,85	4,89

Table 2: Overall evaluation of the predictors for 02/03/1995.

INDEX	MLP1	MLP2	HNM
MAPE (%)	2,64	*1,58*	2,33
MSE (MW^2)	10156	*4211*	8675
ME (MW)	-51,50	*39,04*	52,99
MAX (%)	5,12	*3,84*	5,36

Table 3: Hourly percentage error for 02/07/1995.

Time (h)	MLP1 error	MLP2 error	HNM error
1	7,88	3,22	0,11
2	7,98	4,12	1,14
3	8,76	5,42	0,41
4	6,14	3,25	0,58
5	6,09	3,63	0,12
6	6,37	4,47	1,04
7	4,23	2,81	2,98
8	4,07	3,27	0,43
9	2,72	3,11	1,31
10	1,99	2,49	1,75
11	4,28	4,50	1,90
12	4,83	5,26	1,02
13	6,18	7,17	1,04
14	7,49	9,87	3,54
15	6,57	9,26	0,90
16	5,41	7,36	0,63
17	6,76	8,67	1,86
18	6,27	7,36	6,11
19	6,28	7,01	1,85
20	3,72	5,51	1,81
21	2,90	4,69	2,19
22	5,74	5,76	1,24
23	7,78	5,93	6,38
24	11,56	8,53	8,30

Table 4: Overall evaluation of the predictors for 02/07/1995.

INDEX	MLP1	MLP2	HNM
MAPE (%)	5,92	5,53	*2,03*
MSE (MW^2)	36259	35174	*7885*
ME (MW)	180,65	172,69	*8,70*
MAX (%)	11,56	9,87	*8,30*

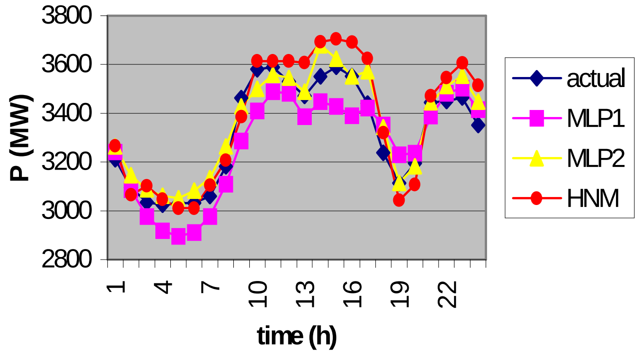

Figure 3: Actual and forecast load for 02/03/1995.

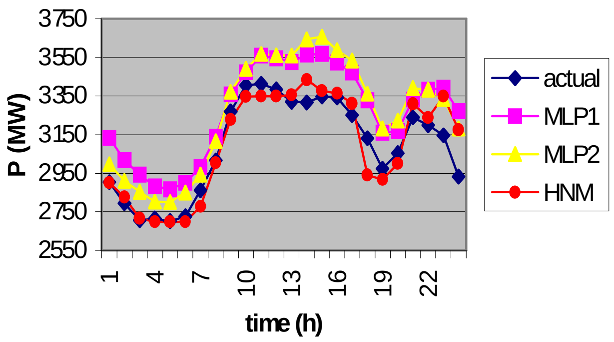

Figure 4: Actual and forecast load for 02/07/1995.

4. Conclusions

A novel artificial neural model for sequence classification and prediction is presented. The model has a topology made up of two self-organizing map networks, one on top of the other. It encodes and manipulates context information effectively.

The results obtained have shown that the HNM was able to perform efficiently the prediction of the electric load in both very short and short forecasting horizons. We intend to do further research on the model, and study carefully the effects of the time integrators on the predictions in order to produce a better adaptability.

Future work will focus on the incorporation of input variables related to weather. Due to climatic diversity over the geographical zone of interest, many meteorological stations are necessary to establish a significant correlation with the load. Installation of such devices is still being planned by the local electric utility. Although it would be desirable to count on such information, the univariate adaptive procedure proposed in this paper implicitly tracks the weather induced load changes over the short-term.

References

[1] A. Khotanzad et al. An adaptive modular artificial neural hourly load forecaster and its implementation in electric utilities. IEEE Trans. on Power Systems, 10(3):1716-1722, August 1995.

[2] H. Mori, “State-of-the-art overview on artificial neural networks in power systems”. in A Tutorial Course on Artificial Neural Networks with Applications to Power Systems, IEEE Catalog no. 96TP112-0, Edited by M.A. El-Sharkawi and D. Niebur, 1996, pp. 51-70.

[3] K. Liu, S. Subbarayan, R.R. Shoults, M.T. Manry, C. Kwan, F.L. Lewis, and J. Naccarino, “Comparison of very short-term load forecasting techniques”. IEEE Trans. Power Systems, vol. 11, no. 2, May 1996, pp. 877-882.

[4] A.G. Bakirtzis, V. Petridis, S.J. Klartzis, M.C. Alexiadis, and A.H. Maissis, “A neural network short-term load forecasting model for the Greek power system”. IEEE Trans. Power Syst., vol. 11, no. 2, May 1996, pp. 858-863.

[5] O. Mohammed, D. Park, R. Merchant, T. Dinh, C. Tong, A. Azeem, J. Farah, and C. Drake, "Practical experiences with an adaptive neural network short-term load forecasting system". IEEE Trans. Power Systems, vol. 10, no. 1, February 1995, pp. 254-265.

[6] D.C. Park, M.A. El-Sharkawi, R.J. Marks II, L.E. Atlas, and M.J. Damborg, "Electric load forecasting using an artificial neural network". IEEE Trans. Power Systems, vol. 6, no. 2, May 1991, pp. 442-449.

[7] T. Baumann and A. Germond. Application of the Kohonen network to short-term load forecasting. In proceedings of the Second International Forum on Application of Neural Networks to Power Systems, pages 407-412, Yokohama, Japan, April 1993.

[8] V. Cherkassky and F. Mulier. Learning from data. J. Wiley & Sons, 1998.

[9] O. A. S. Carpinteiro. A hierarchical self-organizing map model for pattern recognition. In L. Caloba and J. Barreto, editors, Proceedings of the Third Brazilian Conference on Neural Networks (III CBRN), pages 484-488, UFSC, Florianópolis, SC, Brazil, 1997.

[10] O. A. S. Carpinteiro. A hierarchical self-organizing map model for pattern recognition. In L. Niklasson, M. Bóden, and T. Ziemke, editors, Proceedings of the International Conference on Artificial Neural Networks 98, pages 815-820, Skövde, Sweden, 1998. Springer-Verlag.

[11] Z. Lo and B. Bavarian. Improved rate of convergence in Kohonen neural network. In Proceedings of the International Joint Conference on Neural Networks, volume 2, pages 201-206, July 8-12 1991.

[12] A. J. Rocha Reis and A. P. Alves da Silva. Pré-processamento de séries temporais para a previsão de carga a curto prazo via redes neurais. In A. H. Bruciapaglia and A. J. Simões Costa, editors, Proceedings of the XIII Automatica Brazilian Conference (XIII CBA), pages 1138-1143, Florianópolis, SC, Brazil, 2000.