Design of artificial neural networks for short-term load forecasting. Part I: Self-organising feature maps for day type identification

Yuan-Yih Hsu, PhD
Chien-Chuen Yang, MSc

Indexing terms: Artificial intelligence, Artificial neural networks, Power system operation, Self-organising feature maps, Short-term load forecasting, Unsupervised learning method

Abstract: A new approach using artificial neural networks (ANNs) is proposed for short-term load forecasting. To forecast the hourly loads of a day, the hourly load pattern and the peak and valley loads of the day must be determined. In paper I a neural network based on self-organising feature maps to identify those days with similar hourly load patterns is developed. These days with similar load patterns are said to be of the same day type. The load pattern of the day under study is obtained by averaging the load patterns of several days in the past which are of the same day type as the given day. In Part II of the paper a feedforward multilayer neural network is designed to predict daily peak load and valley load. Once the peak load and valley load and the hourly load pattern are available, the desired hourly loads can be readily computed. The effectiveness of the proposed neural network is demonstrated by the short-term load forecasting of the Taiwan Power Company.

List of symbols

$L(i)$ = load for hour i
$L_n(i)$ = normalised load for hour i
L_v = daily valley load
L_p = daily peak load
w_{ij} = connection weight between input node i and output node j
N = number of input nodes
M = number of output nodes
N_c = radius of the neighbourhood
η = step size for updating

1 Introduction

In the past two decades, considerable efforts have been made to develop effective approaches to forecasting the hourly loads for the next day or next week [1, 2]. This work can be roughly divided into two categories: the statistical approach [3–10] and the expert system approach [11–13]. The statistical approach can yield good results for normal days. The expert system approach has received much attention in recent years because it can combine human experts' experience with statistical techniques to generate accurate load forecasts for special days with unusual load patterns.

Paper 8141C (P9), first received 16th July 1990 and in revised form 15th April 1991

The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Recently, a rule-based expert system [14] has been developed by the authors to forecast the hourly loads of the Taiwan Power Company (TPC). In the development of that expert system, it was observed that the most important factors that affect load forecast accuracy are the hourly load patterns and the daily peak and valley loads. In this part of the paper we will deal with the problem of how to derive a suitable hourly load pattern for the day of which the hourly loads are to be forecasted. The problem of predicting daily peak load and valley load will be discussed in Part II of the paper.

To reach a proper load pattern for a given day, historical load data are analysed to divide all days into several groups, with each group comprising the days with similar load patterns. These days with similar patterns will be referred to as the days of the same day type. Once all the day types have been identified from the load data in the past, each day including the day under study can be assigned a day type. Therefore, the desired load pattern for the given day can be obtained by averaging the load patterns for several recent days of the same day type as the given day.

In Reference 14 it was mentioned that several important day types such as weekdays, Saturdays, Sundays, holidays and the Chinese New Year had been identified by the operators through their previous operational experience. Because the load pattern is a function of the social habits of customers, which change with time, new patterns may emerge and some original patterns may lose their significance or even disappear. It takes time for the operators to notice these changes in load pattern. Therefor it is desirable to develop a tool to help the operators to identify new load patterns and update the present load pattern. A pattern recognition method has been employed in Reference 14 to serve this purpose. Indeed, some other day types such as Mourning Day have been identified through this approach.

In this paper, a new approach using artificial neural networks (ANNs) is proposed for the classification of hourly load patterns. Artificial neural networks [15, 16] have been given much attention by power engineers in the past few years. Many interesting applications of neural nets in the power field have been reported, such as transient stability analysis [17], capacitor control [18], alarm processing [19], contingency selection [20], static security assessment [21], harmonic load identification

[22], observability analysis [23] and dynamic stability analysis [24]. Most of this work has used the multilayer feedforward network with a back-propagation/momentum learning technique [15]. This kind of neural network falls into the category of supervised learning [16] because the output for each output pattern must be specified during training. For our problem of load pattern classification, we are not able to specify the output (day type) for an input pattern (hourly load pattern) during training because we want the neutral net to yield potential 'new' day types and merge some existing day types which now have similar patterns. In this case, we do not know in advance how many solutions (types of day) there will be for the output units of the neural network. Thus a neural net with an unsupervised learning method is necessary. In this paper, Kohonen's self-organising feature map [16, 25] is employed to classify the 365 days in the year into several groups according to the hourly load pattern of each day, with each group consisting of those days with similar load patterns. Each such group is referred to as a day type.

To demonstrate the effectiveness of the proposed approach, day type identification has been performed on the Taiwan power system using the hourly load data in the database of the Taiwan Power Company. Extensive studies have been made to evaluate the effect on the final results of various parameters in the training process, such as initial weights and the step size η. It is found that the developed neural network can identify the major day types automatically. Thus the developed neural net can serve as a valuable aid to system operators for day type identification in short-term load forecasting.

2 Problem formulation

Day type identification begins with compiling the 24 hourly loads $L(i)$, $i = 1, \ldots, 24$, of each day in the five-year period. These data are available from the database of the Taiwan Power Company. These hourly loads are then normalised by using the following equation

$$L_n(i) = \frac{L(i) - L_v}{L_p - L_v} \quad i = 1, 2, \ldots, 24 \tag{1}$$

where $L_n(i)$ is the normalised load for hour i, $L(i)$ is the load for hour i, L_v is the daily valley load and L_p is the daily peak load.

These normalised hourly loads $L_n(i)$, $i = 1, \ldots, 24$, are provided to the neural network as the inputs. Thus each input pattern contains the 24 normalised hourly loads of a day. In other words, each input pattern is described by the input pattern vector

$$X' = [x'_1, x'_2, \ldots, x'_{24}]^T \tag{2}$$

$$= [L_n(1), L_n(2), \ldots, L_n(24)]^T \tag{3}$$

The input pattern vector X' is again normalised to obtain a vector X of unity length before training

$$X = [x_1, x_2, \ldots, x_{24}]^T \tag{4}$$

where

$$x_i = x'_i \bigg/ \left(\sum_{i=1}^{24} x_i'^2 \right)^{1/2} \quad i = 1, 2, \ldots, 24$$

Given a number of input pattern vectors $X(1)$, $X(2)$, …, $X(P)$, our objective is to divide them into several clusters with each cluster comprising similar pattern vectors X.

3 Kohonen's self-organising feature maps

The self-organising feature map originated from the observation that the fine structure of the cerebral cortex in the brain is self-organised during training. It maps the N inputs $x_1, x_2, \ldots, x_N$ ($N = 24$ in our study) on to the M output nodes $y_1, y_2, \ldots, y_M$ ($M = 324$ in our study), as shown in Fig. 1. The output nodes are arranged on a two-dimensional grid-like network. Each output node is described by a pair of rectangular co-ordinates. Note that the number of output nodes or the size of the grid (M) has been arbitrarily chosen to be 18×18 ($=324$). Other grid sizes can be employed, as long as computer memory capacity permits.

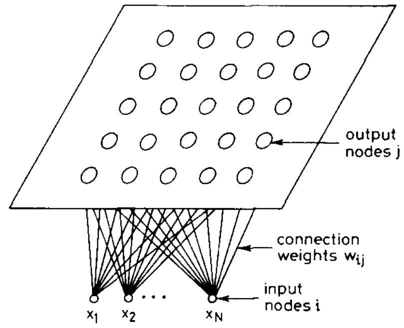

Fig. 1 *Structure of self-organising feature map*

As shown in Fig. 1, each input unit i is connected to an output unit j through connection weight w_{ij}. The continuous-valued input patterns and connection weights will give the output node j a continuous-valued activation value a_j as follows

$$a_j = \sum_{i=1}^{N} w_{ij} x_i \tag{5}$$

$$= W_j X \tag{6}$$

where X is the input pattern vector as given in eqn. 4, and $W_j = [w_{1j}, w_{2j}, \ldots, w_{Nj}]^T$ is the connection weight vector for output node j. Note that $N = 24$ in our study.

When an input pattern is presented to the neural net without specifying the desired output, the neural net computes the activation value for each output node based on present connection weights. The input pattern is said to be mapped to the output node with maximum activation value. After enough input pattern vectors have been presented, input patterns with similar features will be mapped to the same output unit or to output units within a small neighbourhood. Because the clustering of input pattern vectors is self-organised in the learning process, and the ordering of the output node of an input pattern vector is based on that feature, this kind of neural network is called the self-organising feature map by Kohonen [16, 25].

Fig. 2 shows the flowchart of the proposed algorithm to produde the self-organising feature map. The algorithm begins with reading in the input pattern vectors $X(1), X(2), \ldots, X(P)$ (block 1). It is assumed that we use P different input patterns in the training process.

The next step is to select the initial values for the connection weights w_{ij} ($i = 1, \ldots, N, j = 1, \ldots, M$) and the radius of the neighbourhood N_c. Kohonen suggested that the connection weights be initialised to small random values [16, 25]. The approach works well for the cases where the input vectors are widely spread over the whole pattern space. But the input vectors in the present work are restricted to a small portion of the space. Therefore we propose to set the initial weight vectors to be

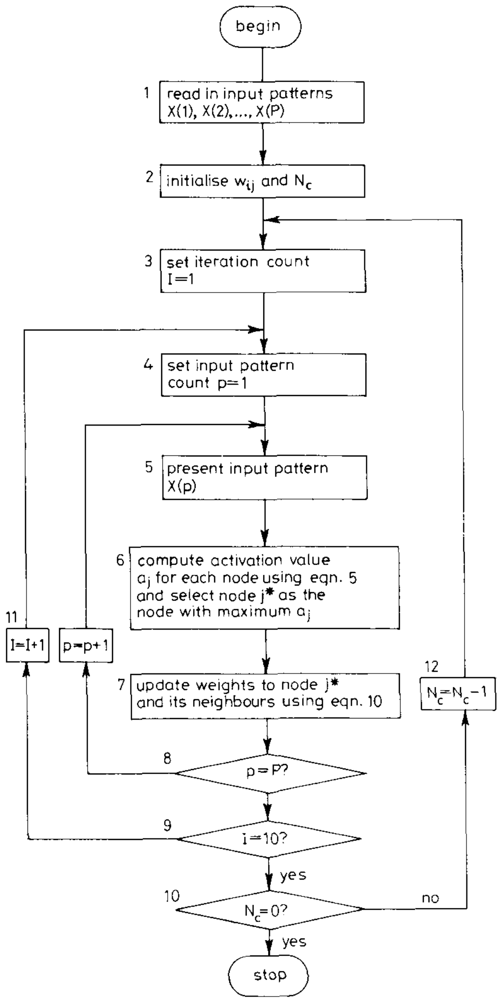

Fig. 2 *Flowchart for algorithm to produce self-organising feature maps*

around the mean of these vectors. In other words, we first define

$$W'_j = \text{mean of } (X(1), X(2), \ldots, X(P) \quad j = 1, \ldots, M \tag{7}$$

or

$$w_{ij} = \text{mean of } (x_i(1), x_i(2), \ldots, x_i(P)) \quad i = 1, \ldots, N \tag{8}$$

Then perturb W'_j with a random noise as follows:

$$W''_j = W'_j + [5r \times \text{variance of } (X(1), X(2), \ldots, X(P))] \tag{9}$$

where r is a random number uniformly distributed in the range from -0.5 to 0.5. Finally, normalise W''_j to give unity vector length

$$W_j = W''_j \Bigg/ \left(\sum_{i=1}^{24} w_{ij}^{2''} \right)^{1/2} \tag{10}$$

The radius of neighbourhood N_c is useful in updating the weights (see block 7 of Fig. 2). As shown in Fig. 3, the

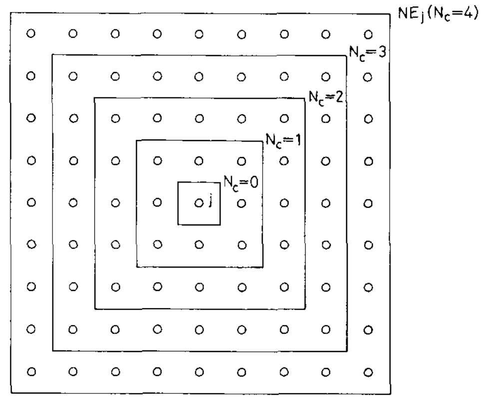

Fig. 3 *Neighbours for node j, $NE_j(N_c)$*

neighbours of an output node j are defined to be the $(2N_c + 1) \times (2N_c + 1)$ output nodes within the square block, with node j being the centre of the block. For example, the neighbours for output node j with $N_c = 4$ include 81 nodes in the largest square block NE_j ($N_c = 4$) of Fig. 3.

In this study, the radius of neighbourhood N_c is initialised to be 4. In the training process, N_c is decreased by 1 after ten iterations (see block 12 of Fig. 2) until N_c equals zero. Here, by one iteration we mean the P input pattern vectors $X(1)$, $X(2)$, ..., $X(P)$ have all been presented once. Detailed procedures within one iteration are described as follows.

After the initial connection weights have been specified, the activation value of each output node can be computed using eqn. 5. The node j^* with the maximum activation value is picked up. This is what the training algorithm performs in block 6 of Fig. 2.

In block 7, the connection weights for node j^* and all nodes in the neighbourhood defined by NE_{j^*} as shown in Fig. 3 are updated using the following equation

$$w_{ij,\,\text{updated}} = w_{ij} + \eta(x_i - w_{ij}) \quad \text{for } i = 1, \ldots, N, j \in NE_{j^*} \tag{11}$$

where η is the step size for the updating. Note that η can be fixed or variable throughout the iterations. A value of η from 0.003 to 0.008 has been found to give satisfactory results in this study.

The above procedures are repeated for each input pattern. When all P input patterns have been presented, we say that one iteration has been completed. Because there are ten iterations sharing one common radius of neighbourhood and we use five different values of N_c (N_c = 4, 3, 2, 1, 0) in the training process, there will be 50 iterations in all.

4 Application to day type identification in short-term load forecasting

The developed neural net has been applied to identify the day types of the Taiwan power system using the five-year hourly load data in the database of the Taiwan Power Company. Owing to limited space, only the results from some typical studies are given.

4.1 Example 1: load patterns in April 1987

The 30 load patterns in April 1987 are mapped to a two-dimensional output network with 324 nodes. Figs. 4, 5, 6 and 7 depict the feature maps after 1, 5, 25 and 50 iterations of training, respectively. The step size η is decreased from 0.008 to 0.001 throughout the iterations. Table 1 summarises the results in Figs. 4–7. To see how the step size η affects the resultant feature maps, Figs. 8, 9 and 10 show the feature maps after 50 iterations of training for fixed step sizes of 0.002, 0.005 and 0.008, respectively. These maps are summarised in Table 2.

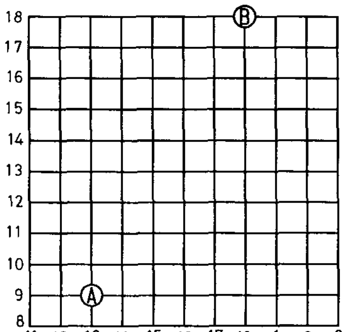

Fig. 4 *Feature map for load patterns in April 1987*
(variable step size, 1 iteration)

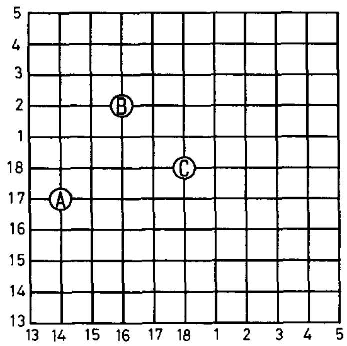

Fig. 5 *Feature map for load patterns in April 1987*
(variable step size, 5 iterations)

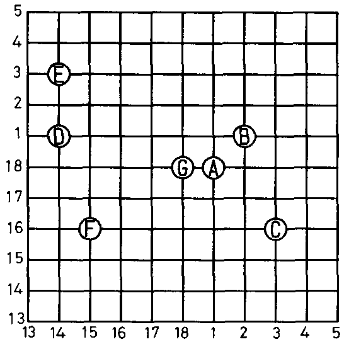

Fig. 6 *Feature map for load patterns in April 1987*
(variable step size, 25 iterations)

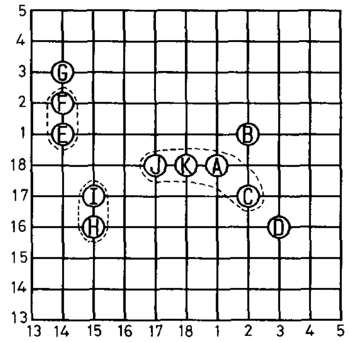

Fig. 7 *Feature map for load patterns in April 1987*
(variable step size, 50 iterations)

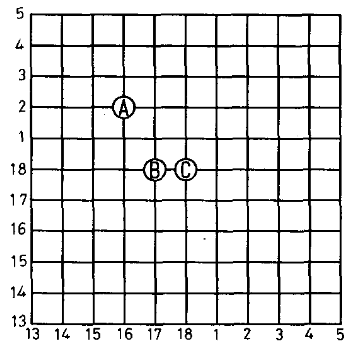

Fig. 8 *Feature map for load patterns in April 1987*
(η = 0.002, 50 iterations)

Table 1: Summary of the feature map for April 1987 (variable step size)

1 iteration (Fig. 4)		5 iterations (Fig. 5)		25 iterations (Fig. 6)		50 iterations (Fig. 7)	
Output node	Date (day)	Output node	Date (day)	Output node	Date (day)	Output node	Date (day)
A(13, 9)	5(SU)*	A(14, 17)	19(SU)	A(1, 18)	2(TH), 3(F), 14(T) 15(W), 16(TH), 17(F)	A(1, 18)	1(W), 2(TH), 3(F) 14(T), 15(W), 16(TH) 17(F)
B(18, 18)	Others	B(16, 2)	5(SU), 12(SU), 26(SU)	B(2, 1)	6(M)	B(2, 1)	6(M)
		C(18, 18)	Others	C(3, 16)	7(T), 13(M), 20(M) 27(M)	C(2, 17)	8(W), 9(TH), 10(F) 21(T), 23(TH), 29(W)
				D(14, 1)	12(SU), 19(SU), 26(SU)	D(3, 16)	7(T), 13(M), 20(M) 27(M)
				E(14, 3)	5(SU)	E(14, 1)	12(SU), 19(SU)
				F(15, 16)	4(S), 11(S), 18(S) 25(S)	F(14, 2)	26(SU)
				G(18, 18)	Others	G(14, 3)	5(SU)
						H(15, 16)	11(S), 18(S), 25(S)
						I(15, 17)	4(S)
						J(17, 18)	30(TH)
						K(18, 18)	22(W), 24(F), 28(T)

* Abbreviations: SU, Sunday; M, Monday; T, Tuesday; W, Wednesday; TH, Thursday; F, Friday; S, Saturday

Table 2: Summary of the feature map for April 1987 (fixed step size, 50 iterations)

$\eta = 0.002$		$\eta = 0.005$		$\eta = 0.08$	
Output node	Date (day)	Output node	Date (day)	Output node	Date (day)
A(16, 2)	5(SU), 12(SU), 19(SU), 26(SU)	A(1, 17)	24(F)	A(1, 1)	2(TH), 15(W)
		B(1, 18)	1(W), 2(TH), 3(F) 14(T), 15(W)	B(1, 3)	16(TH), 17(F)
				C(1, 15)	23(TH), 29(W)
B(17, 18)	6(M)	C(2, 17)	8(W), 9(TH), 10(F) 21(T), 23(TH), 29(W)	D(1, 16)	9(TH)
C(18, 18)	Others			E(1, 18)	1(W)
		D(3, 1)	6(M)	F(2, 15)	21(T)
				G(3, 17)	27(M)
				H(3, 18)	7(T)
		E(3,16)	7(T), 13(M), 20(M), 27(M)	I(4, 16)	6(M)
				J(4, 18)	13(M), 20(M)
		F(14, 2)	5(SU), 12(SU), 19(SU), 26(SU)	K(14, 2)	26(SU)
				L(14, 3)	5(SU), 12(SU), 19(SU)
		G(15, 17)	4(S), 11(S)	M(15, 15)	30(TH)
		H(16, 17)	18(S)	N(15, 16)	25(S)
				O(15, 17)	4(S), 11(S), 18(S)
				P(17, 17)	22(W)
		I(16, 18)	25(S)	Q(18, 1)	3(F), 28(T)
		J(17, 18)	30(TH)	R(18, 2)	14(T)
		K(18, 1)	16(TH), 17(F) 28(T)	S(18, 15)	8(W), 10(F)
				T(18, 16)	24(F)
		L(18, 18)	22(W)		

Abbreviations: see Table 1

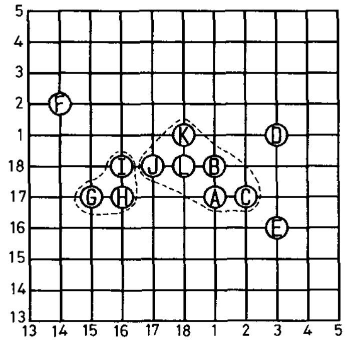

Fig. 9 *Feature map for load patterns in April 1987*
($\eta = 0.005$, 50 iterations)

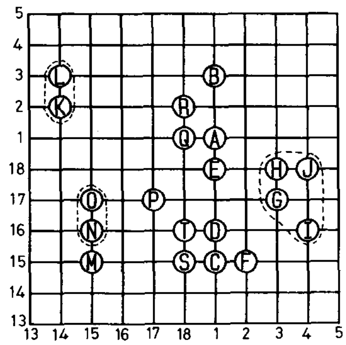

Fig. 10 *Feature map for load patterns in April 1987*
($\eta = 0.008$, 50 iterations)

4.2 Example 2: load patterns in May 1986 and November 1986

To examine the effectiveness of the proposed neural network in different seasons, day type identification is also performed on other months of the year. Figs. 11 and 12 show the feature maps for the load patterns in May 1986 and November 1986, after 50 iterations of training. The maps are summarised in Tables 3 and 4, respectively.

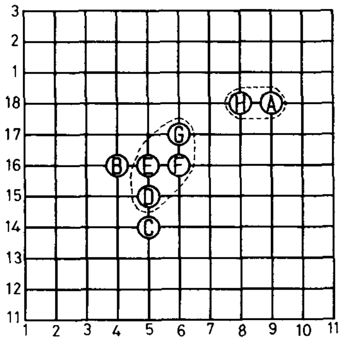

Fig. 11 *Feature map for load patterns in May 1986*

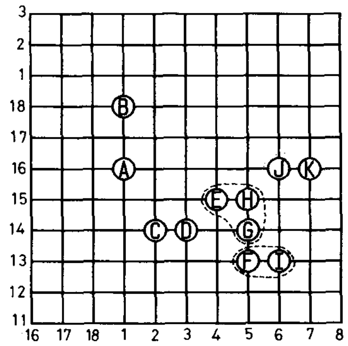

Fig. 12 *Feature map for load patterns in November 1986*

Table 3: Summary of the feature map for May 1986

Output node	Date (day)
A(9, 18)	1(TH), 11(SU)
B(4, 16)	2(F), 5(M), 12(M), 19(M), 26(M)
C(5, 14)	3(S), 10(S), 17(S), 24(S), 31(S)
D(5, 15)	6(T), 7(W), 9(F), 16(F), 20(T), 28(W), 29(TH)
E(5, 16)	8(TH), 13(T)
F(6, 16)	14(W), 15(TH), 21(W), 22(TH), 27(T), 30(F)
G(6, 17)	23(F)
H(8, 18)	4(SU), 18(SU), 25(SU)

Abbreviations: see Table 1

Table 4: Summary of the feature map for November 1986

Output node	Date (day)
A(1, 16)	2(SU), 9(SU), 16(SU), 23(SU), 30(SU)
B(1, 18)	28(F)
C(2, 14)	12(W)
D(3, 14)	1(S)
E(4, 15)	4(T), 26(W), 27(TH)
F(5, 13)	22(S), 29(S)
G(5, 14)	5(W), 11(T), 19(W), 20(TH), 25(T)
H(5, 15)	6(TH), 7(F), 18(T), 21(F)
I(6, 13)	8(S), 15(S)
J(6, 16)	3(M), 10(M), 13(TH), 17(M), 24(M)
K(7, 16)	14(F)

Abbreviations: see Table 1

5 Discussion

From the results in Figs. 4–12 and Tables 1–4, the following observations can be made.

First, from the feature maps in Figs. 4–7 it is observed that new output nodes, each corresponding to a group of input patterns with a particular feature, emerge as input patterns, are sequentially presented, and the connection weights are updated during training. From these feature maps, the day types as listed in Table 5 can be identified. Note that days of which the load patterns are mapped to the same output node or neighbouring nodes are referred to as of the same day type.

In Table 5 it is observed that after 5 iterations of training, only the output nodes (A, B) for Sundays appear in the feature map. But all the six important load patterns appear in the feature map after 25 iterations. The basic principle for clustering the output nodes is to put neighbouring nodes in a group, but the way of clustering is by no means unique. For example, output node G, which is the map of the load pattern for Mourning Day, is regarded as a separate group in Table 5, making Mourning Day a special type of day. In fact, because Mourning Day is a national holiday and output node G is close to nodes F and E, one can also regard nodes E, F and G as the nodes in the same group. It is also noted that the day after Mourning Day (Monday 6th April) is also a holiday. Thus the output node of this day, B, is very different from output node D for Mondays. In addition, the following day (Tuesday 7th April) is mapped to node D for Mondays.

Secondly, the value of step size η affects the learning speed and the resultant feature maps to a great extent. As shown in Fig. 8, a small step size of 0.002 will make the neural network converge very slowly during training. In fact, only the output nodes for Sundays (node A) and Mondays (node B) appear after 50 iterations of training.

On the other hand, a large step size (e.g. 0.008) will improve the learning speed. But similar load patterns can be mapped to several nodes which are not close to one another. For example, the weekdays are mapped to 13 output nodes (nodes A, B, C, D, E, F, H, M, P, Q, R, S, T) in the feature map of Fig. 10. It appears that a step size of 0.005 will give good results (see Fig. 9 and Table 2) for the load patterns in April 1987. To have fast learning and to avoid the drawback of mapping a similar pattern to widespread output nodes, we use a variable step size [16] in the present study. The step size η is initialised to be 0.008 and is gradually decreased to 0.001 as the training proceeds. It was found that this variable step size would yield satisfactory results for most of the load patterns in five years.

Thirdly, from the feature map for the load patterns in May 1986 (see Fig. 11 and Table 3), the day types as shown in Table 6 can be identified. Note that the weekdays in May are mapped to four nodes (D, E, F, G) which are neighbours on the plane (see Fig. 11). Only one day type is assigned to these four nodes. It is also observed that the load pattern of 1st May (Labour Day) and of Sunday are mapped to two neighbouring nodes A and H. The day after Labour Day has a load pattern similar to that of Monday. Thus the load pattern of 2nd May is mapped to the node of Monday (node B).

Table 6: Summary of day types for May 1986

Day type	Description	Output node
1	Sunday and holiday	A, H
2	Monday and day after holiday	B
3	Saturday	C
4	Weekdays except holidays	D, E, F, G

Fourthly, in Part II of this work we will give an example of load forecasting for 5th May 1987. In conducting the load forecast, we first have to identify the type of day for the given date. Thus we need a list of day types for the month of May. The list of day types in Table 6 can serve that purpose because it summarises the types of days for the same month in the preceding year (May 1986). The day type of 5th May 1987 can then be obtained by picking out a proper one from those in Table 6. After the day type of the given date has been determined, the desired hourly load pattern can be achieved by averaging the load patterns of several recent days in April and May of 1987 which are of the same day type as the given date. Detailed procedures will be described in Part II of the paper.

Fifthly, in the feature map for November 1986 (Table 4) the load pattern of 1st November (Saturday) is mapped to a separate node D, whereas the load patterns of other Saturdays are mapped to two neighbouring nodes F and I. This is because 1st November is a day following a holiday (31st October). In addition, 12th November is a national holiday, and therefore it is mapped to a node C which is separate from the nodes for weekdays (E, H, G); 13th November is mapped to node J for Mondays. The load pattern for 28th November (node B) seems to result from poor data.

Table 5: Identified day types based on the feature maps in Figs. 4–7

Day type	Description	Output nodes			
		1 iteration (Fig. 4)	5 iterations (Fig. 5)	25 iterations (Fig. 6)	50 iterations (Fig. 7)
1	Sunday	A	A, B	D	E, F
2	Weekdays except holidays			A	ACJK
3	Monday			C	D
4	Saturday			F	H, I
5	Special day: Mourning Day (Sunday 5th April)			E	G
6	Special day: the day after (Monday 6th April)			B	B

Sixthly, when a node separate from all existing clusters appears on the feature map, it can be either the mapping of poor data or the mapping of a new type of load pattern. In this case, operators have to distinguish between the two possiblities. In the event of a new load pattern, the operator may wish to update his original list of day types. Thus, the self-organising feature map is especially useful to system operators when a new load pattern emerges due to changes in the social habits of customers.

Finally, the proposed feature map is capable of generating all the day types in the Taiwan power system that have been identified using other approaches such as the expert system approach [14]. A distinct feature of the proposed approach is that the day after a public holiday (such as Labour Day) is always grouped together with Mondays. This has not been reported in previous work [14].

6 Conclusions

An artificial neural network using a self-organising feature map is developed to identify the day types which are essential to short-term load forecasting. The inputs to the neural network are the 24 hourly load data of a day. Each input load pattern is mapped to a node on the output plane according to its feature. Similar load patterns are mapped to the same node or neighbouring nodes. By examining the feature maps for a set of historical load patterns, important day types of the system can be identified. A special feature of the self-organising feature map is that it is capable of identifying a new type of load pattern before the operators can recognise the new day type. Thus it can serve as a valuable aid to system operators in short-term load forecasting.

The artificial neural net has been implemented on a personal computer. To demonstrate the effectiveness of the proposed neural net, day type identification has been performed on the Taiwan power system using the five-year hourly load data of the Taiwan Power Company. It is found that the neural net is very effective in identifying the day types. The required CPU time for training the neural network is only 10 minutes on a personal computer. Thus the developed neural net is a valuable tool for day type identification which requires only minor training effort.

7 Acknowledgments

The authors wish to express their sincere gratitude to Messrs. C.C. Liang, T.S. Lai, K.K. Chen, B.S. Chang and T.E. Lee of the System Operation Department of the Taiwan Power Company for providing valuable system data and operational experience. Financial support given to this work by the National Science Council of Taiwan under contract number NSC-78-0404-E002-30 is appreciated.

8 References

1 MOGHRAM, I., and RAHMAN, S.: 'Analysis and evaluation of five short-term load forecasting techniques', *IEEE Trans.*, 1989, **PWRS-4**, pp. 1484–1491

2 GROSS, G., and GALIANA, F.D.: 'Short term load forecasting', *Proc. IEEE*, 1987, **75**, pp. 1558–1572

3 PAPALEXOPOULOS, A.D., and HESTERBERG, T.C.: 'A regression-based approach to short-term system load forecasting'. IEEE 1989 PICA Conference, pp. 414–423

4 LU, Q.C., GRADY, W.M., GRAWFORD, M.M., and ANDERSON, G.M.: 'An adaptive nonlinear predictor with orthogonal escalator structure for short-term load forecasting', *IEEE Trans.*, 1989, **PWRS-4**, pp. 158–164

5 CAMPO, R., and RUIZ, P.: 'Adaptive weather-sensitive short term load forecast', *IEEE Trans.*, 1987, **PWRS-2**, pp. 592–600

6 HAGAN, M.T., and BEHR, S.M.: 'The time series approach to short term load forecasting', *IEEE Trans.*, 1987, **PWRS-2**, pp. 785–791

7 DAVEY, J., SOACHS, J.J., CUNNINGHAM, G.W., and PRIEST, K.W.: 'Practical application of weather sensitive load forecasting to system planning', *IEEE Trans.*, 1972, **PAS-91**, pp. 971–977

8 THOMPSON, R.P.: 'Weather sensitive demand and energy analysis on a large geographically diverse power system — application to short-term hourly electric demand forecasting', *IEEE Trans.*, 1976, **PAS-95**, pp. 384–393

9 IRISARRI, G.D., WIDERGREN, S.E., and YEHSAKUL, P.D.: 'On-line load forecasting for energy control center application', *IEEE Trans.*, 1982, **PAS-101**, pp. 71–78

10 MESLIER, F.: 'New advances in short term load forecasting using Box and Jenkins approach'. Paper A78 051-5, presented at the IEEE/PES 1978 Winter Meeting

11 RAHMAN, S., and BHATNAGAR, R.: 'An expert system based algorithm for short term load forecast', *IEEE Trans.*, 1988, **PWRS-3**, pp. 392–399

12 JABBOUR, K., RIVEROS, J.F.V., LANDSBERGEN, D., and MEYER, W.: 'ALFA: automated load forecasting assistant', *IEEE Trans.*, 1988, **PWRS-3**, pp. 908–914

13 RAHMAN, S., and BABA, M.: 'Software design and evaluation of a microcomputer-based automated load forecasting system', *IEEE Trans.*, 1989, **PWRS-4**, pp. 782–788

14 HO, K.L., HSU, Y.Y., CHEN, C.F., LEE, T.E., LIANG, C.C., LAI, T.S., and CHEN, K.K.: 'Short term load forecasting of Taiwan power system using a knowledge-based expert system', *IEEE Trans.*, 1990, **PWRS-5**, pp. 1214–1221

15 RUMELHART, D.E., HINTON, G.E., and WILLIAMS, R.J.: 'Learning internal representations by error propagation', *in* 'Parallel distributed processing, vol. 1' (MIT Press, Cambridge, MA, 1986), pp. 318–362

16 LIPPMANN, R.P.: 'An introduction to computing with neural nets', *IEEE ASSP Mag.*, 1987, pp. 4–22

17 SOBAJIC, D.J., and PAO, Y.H.: 'Artificial neural-net based dynamic security assessment for electric power systems', *IEEE Trans.*, 1989, **PWRS-4**, pp. 220–228

18 SANTOSO, M.I., and TAN, O.T.: 'Neural-net based real-time control of capacitors installed on distribution systems'. Paper 89 SM 768-3 PWRD, presented at the IEEE/PES 1987 Summer Meeting

19 CHAN, E.H.P.: 'Application of neural network computing in intelligent alarm processing'. IEEE 1989 PICA Conference, pp. 246–251

20 FISCHL, R., KAM, M., CHOW, J.C., and RICCIARDI, S.: 'Screening power system contingencies using a backpropagation trained multiperception'. 1989 International Symposium on Circuits and Systems, pp. 486–489

21 AGGOUNE, M.E., ATLAS, L.E., COHN, D.A., DAMBORG, M.J., EL-SHARKAWI, M.A., and MARKS, R.J.II: 'Artificial neural networks for power system static security assessment'. 1989 International Symposium on Circuits and Systems, pp. 490–494

22 MORI, H., UEMATSU, H., TSUZUKI, S., SAKURAI, T., KOJIMA, Y., and SUZUKI, K.: 'Identification of harmonic loads in power systems using an artificial neural networks'. Second Symposium on Expert Systems Application to Power Systems, 1989, pp. 371–377

23 MORI, H., and TSUZUKI, S.: 'Power system topological observability analysis using a neural network model'. Second Symposium on Expert Systems Application to Power Systems, 1989, pp. 385–391

24 AGGOUNE, M., EL-SHARKAWI, M.A., PARK, D.C., DAMBORG, M.J., and MARKS. R.J.II: 'Preliminary results on using artificial neural networks for security assessment'. IEEE 1989 PICA Conference, pp. 252–258

25 KOHONEN, T.: 'Self-organization and associative memory' (Springer, Berlin, 1988)