SELFORGANIZING NEURAL NETWORKS FOR SHORT TERM LOAD FORECASTING IN POWER SYSTEM

S. Osowski, K. Siwek
Institute of the Theory of Electrical Engineering and Electrical Measurements
Warsaw University of Technology, Warsaw, POLAND
email: sto@iem.pw.edu.pl

1 INTRODUCTION

Prediction of the electrical energy consumption is an important subject from the power plant economy point of view. Especially useful is the daily planning of the power demand. When forecasting the next days electrical load the previous load curves are used. The conventional load forecast approaches make use of time series methods and stochastic approaches for processing the numerical information [3, 4]. The need to deal with today's increasingly complex information system, forces to develop modern, more reliable methods of prediction. Recently great attention has been paid to the neural network application. Three general approaches have been studied and develop. The first is based on the multilayer perceptron and its generalization ability [2, 5, 9]. The second is the use of selforganizing mechanism of Kohonen network and its ability to represent the clusters of data by the weights of the neurons [2, 8, 11]. The third one applies the recurrent neural networks, especially Elman or RTRN and exploits the internal dynamics built into these kinds of networks [6, 7].

In our opinion the most interesting and promising is the application of the selforganizing networks, representing directly the statistical distribution of data. Thanks to the simple and robust learning mechanism they could model the data with good overall accuracy and deliver the predicted values of the load of the power system in the future with the minimum error.

The paper will consider different solutions of 24 hour load prediction using selforganising neural networks. Three different structures will be studied: the conventional Kohonen network and the prediction based on the statistics of the winning neurons only, the extended form of Kohonen network taking into account the activity of winner and neighbouring neurons and the Kohonen layer cooperating with the postprocesing feedforward one. The results of the numerical experiments will be given and discussed in the paper.

2 SELFORGANIZING NEURAL NETWORK

Selforganizing neural network, often called the Kohonen network, is composed of a single layer of neurons working in a self organizing competitive mode [12]. Each neuron is fed the N components of the input vector through the weighting coefficients, forming the N-dimensional vector $\mathbf{w}_i$. In the process of learning, the neurons are self organizing, where the self organization algorithm is formed by the sequence of the following operations [1, 2, 12]

- present the input vector $\mathbf{x}$ to the network
- find the area in the network where the specific neuron responds most strongly to the previously presented vector $\mathbf{x}$; the winner unit N_w is the one, whose weight vector is nearest in the sense

of assumed distance measure to the input vector

- update the weights of the winner and selected neurons in close neighbourhood area of winner in the direction towards the vector $\mathbf{x}$.

Repeating these sequence many times brings the network to an organized state, in which each neuron represents one separate cluster of data. In the generalized Kohonen algorithm we update the weights of the neurons found in the neighbourhood around the winning neuron N_w, according to the following rule

$$\mathbf{w}_i \leftarrow \mathbf{w}_i + \eta_i G(i, \mathbf{x})[\mathbf{x} - \mathbf{w}_i] \quad (1)$$

where $\mathbf{w}_i$ is the vector of weights of ith neuron found in the neighbourhood of the winner and $\eta_i(t)$ is the adaptation coefficient (learning constant), decreasing with time. Usually it is the linear decrease, starting from some initial value and ending with $\eta_i = 0$.

The neighbourhood $G(i, \mathbf{x})$ of the winner is also decreasing with time and is adjusted in special way. One of the solution is application of the gaussian function defined around the winner.

$$G(i, \mathbf{x}) = exp\left(-\frac{d^2(i, w)}{2\lambda^2}\right) \quad (2)$$

where $d(i, w)$ is the distance between the winner and ith neuron while λ is the parameter. To get good organization of neurons it is important to keep all neurons active in the process of learning.The efficient way to do it is to apply the conscience mechanism [12]. In this mechanism too active neuron (winning too often) is punished. In practice this punishment was implemented in the competition process by modifying the measure of distance between the input vector $\mathbf{x}$ and the weight vector of the neuron. The real distance is multiplied by the number of winnings of each neuron, i.e.,

$$d(i, w) \leftarrow d(i, w) \cdot N_w(i) \quad (3)$$

where $N_w(i)$ is the number of winnings of ith neuron. Thanks to such modification the passive neurons, whose initial weights were placed in infavourable region, have the chances to win and to modify their weights. As a result of such organization of the process of learning, the whole net is generally better organized and the total error is smaller.

3 PREPARATION OF LEARNING DATA

The main task of the selforganizing network is to learn the characteristics of the daily loads of the system. Each day of the year has its own 24 hour load characteristic. The days of the same type belonging to the same seasons of the year have similar load characteristics and form clusters, grouping the similar data. Each cluster can be represented by one neuron acting in the competitive mode. Once the network is trained it can be used in the retrieval mode to find out the winner after applying the input data to the input of the network.

To make the prediction independent on the general trend, changing from year to year, we transform the input data by cutting out the mean value and dividing the result by the variance of the data for this day. Instead of real data we use in this way the so called profiles [11], defined in the way

$$p(d, t) = \frac{P(d, t) - P_m(d)}{\sigma(d)} \quad (4)$$

where $P(d, t)$ is the real load of dth day at tth hour, $P_m(d)$ is the mean value of the load of dth day and $\sigma(d)$ is the variance of the load of this day. The set of profiles for each day of the years taking part in learning process forms the training data of the network. Once the network is trained, each neuron represents the data closest to its weight vector in the chosen metric sense.

4 SELFORGANIZING NEURAL NETWORK STRUCTURES FOR PREDICTION

4.1 Classic competitive approach

In this approach we train the selforganizing neural network on the profiles and find the winners representing the individual clusters.

In testing mode, when we want to make prediction for 24-hour load of the particular day, we simply deliver the actual date of the day. On the basis of this, the type of the day is determined automatically and the appropriate winning neuron vectors, corresponding to this type of the days in the past (for example all Tuesdays in April) are found out. The prediction of the load profile $\hat{\mathbf{p}}(d)$ is then the weighted average of all found vectors

$$\hat{\mathbf{p}}(d) = \frac{\sum_{i=1}^{n} k_{di} \mathbf{w}_i}{\sum_{i=1}^{n} k_{di}} \tag{5}$$

where k_{di} is the number of appearances of ith neuron among the winners in the past for this particular day. The real load forecast is then calculated from equation (4) in the form

$$P(d,t) = \sigma(d) p(d,t) + P_m(d) \tag{6}$$

The variance $\sigma(d)$ is estimated using the data base corresponding to the same winning neurons. The mean value $P_m(d)$ is also subject to estimation, this time the closest days of the same type of the known load characteristic may be chosen and their mean values used in prediction of $P_m(d)$. In practical implementation we have used the weighted mean calculation, in which the winning days of the same year are weighted by 1, and the days of the previous years are weighted with decreasing values.

We have applied here the linear model of prediction, taking into account the appropriate values from the last 4 years. Limiting ourselves to the mean value of the load let us denote by W_i and P_i the appropriate weights and mean loads, respectively, where i is the notation of the year. Then the linear prediction of the P_i is given in the form

$$P_i = ai + b \tag{7}$$

The coefficients a and b are obtained from the solution of quadratic programming problem, for which the minimized cost function is defined as

$$E = \sum_{k=i}^{i+3} \left[(ak + b) - P_k \right] W_k \tag{8}$$

In this formulation W_k is the weighting coefficient adjusted for each year, where we have taken 4 last years in definition of E. Solving this quadratic problem we get

$$b = \frac{\sum_{k=i}^{i+3} kW_k \frac{\sum_{k=i}^{i+3} kP_kW_k}{\sum_{k=i}^{i+3} k^2W_k} - \sum_{k=i}^{i+3} P_kW_k}{\sum_{k=i}^{i+3} kW_k \frac{\sum_{k=i}^{i+3} kW_k}{\sum_{k=i}^{i+3} k^2W_k} - \sum_{k=i}^{i+3} W_k} \tag{9}$$

$$a = \frac{\sum_{k=i}^{i+3} kP_kW_k - b\sum_{k=i}^{i+3} kW_k}{\sum_{k=i}^{i+3} k^2 W_k} \tag{10}$$

Different simulations have been made to find the best values for weighting coefficients W_k. As a result of these experiments we have assumed $W_1 = 1$ for the data of actual year, $W_2 = 0.9$, $W_3 = 0.7$, and $W_4 = 0.5$ for the previous years, respectively. Similar estimation is repeated for the variance σ.

These relations have been implemented in the form of C++ program, doing automatically prediction for any date of the year. Note that predicting stage is separated from the activity of Kohonen neural network. As a matter of fact it uses only the data base generated by the Kohonen network for finding the profile values of the appropriate days taking part in prediction. Observe that this scheme of load forecasting allows to predict the whole 24-hour load for each day of the year for any days ahead.

4.2 Fuzzy selforganizing neural network

Further increase of the accuracy can be achieved by taking fuzzy model of the selforganizing neural network. This is also one-layer network, however this time in prediction process we take into account not only the winner but also the activity of some loosers, closest to the winner. The learning phase is performed in the same way as it was done in the first case (Kohonen algorithm). However as a result of learning we memorize not only the winner but also some limited number q (in practical implementation it was number 5) of neurons closest to the winner. At the same time we keep also their relative activities, that represent the membership values.

If the activities of the winner and the neighbouring neurons are denoted by u_w and u_i respectively, we introduce their fuzzyfied activities, that are defined by

$$y_i = e^{-\alpha(u_w - u_i)^2} \tag{11}$$

This value for the winner is $y_w = 1$ and for all other neurons $0 \leq y_i < 1$. The coefficient α is the decaying parameter of this transformation. On the basis of these fuzzyfied activities of neurons we can define the memebership value for ith neuron in the form

$$\mu_i = \frac{y_i}{\sum_{i=1}^{q} y_i} \tag{12}$$

The phase of prediction is performed in the similar way as it was done in the first solution, however this time we take into account not only the winners but also their neighbours and their relative acitivities, described by μ_i. As a result the prediction of the profile for any dth day is given in the form

$$\hat{\mathbf{p}}(d) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{q} \mu_{di}^{(j)} \mathbf{w}_i^{(j)}}{\sum_{i=1}^{n} \sum_{j=1}^{q} \mu_{di}^{(j)}} \tag{13}$$

The parameter $\mu_{di}^{(j)}$ denotes the membership value of jth neuron taking part in prediction of the load for dth day. The number of past days influencing the prediction is denoted here by n. The relations (11) - (13) resemble the fuzzy relationships, thus the method is called fuzzy selforganization.

4.3 Two-layer selforganizing neural network

Another approach to load forecasting is application of the two layer network. The first is the classic selforganizing layer, acting in the standard , described above way while the second is the feedforward linear layer, acting in the supervised mode. The first layer performs the clusterization and the second - real prediction of the profiles of the load. The learning process is composed of two parts: first the selforganizing layer is trained and then after freezing the weights of this layer - the training of the second layer is performed using supervised learning algorithm. The real distinction to the classic method is is the fact that now not only winners take role in prediction but also (to some degree) the loosers, according to their level of activity. At the same time learning is fast, because the initialization of the second layer may be made close to optimum. The initial weights of the output neurons represent now the mean values of the load prediction represented by the winners. Taking into account that winners are most influencing the prediction process, such initialization provides the initial weights very close to optimum. Thanks to this the learning process of the feedforward layer changes the weights only slightly, making this phase of learning very fast and globally optimal.

However this time the prediction phase (the retrieval mode) is slightly different than before. Instead of summing up the weights of the winning neurons, corresponding to the type of the day, we sum up the activities y_i of the output neurons

$$\hat{\mathbf{p}}(d) = \frac{\sum_{i=1}^{n} k_{di} \mathbf{y}_i}{\sum_{i=1}^{n} k_{di}} \tag{14}$$

The structure of this equation is the same as (5), however this time the activity of output neurons is influenced by all neurons of the selforganizing layer. Thanks to this the method resembles rather the fuzzy prediction, described above.

5 RESULTS OF NUMERICAL EXPERIMENTS

The numerical experiments have been carried out on the data of the Polish Power System and used the real data for years 1986 - 1995. The data have been first converted to the profiles and these profiles have been used in training of the network. The Kohonen network of 100 neurons has been used in experiments. The weights of each neuron or the activity of neurons (in the case of 2-layer selforganizing network) represent the profile vector corresponding to the energy consumption of the particular day of the year.

Table 1 The MAPE errors of prognosis

Date	CSO	FSO	2LSO
1990	3.35%	3.24%	3.32%
1991	3.41%	3.29%	3.31%
1992	2.74%	2.55%	2.53%
1993	2.41%	2.29%	2.32%
1994	2.45%	2.18%	2.17%
1995	2.34%	2.12%	2.14%

On the basis of predicted profiles the prognosis for 24 hours is made by applying equation (6). The important point is to estimate the mean value $P_m(d)$ of the power of the particular dth day and the variance $\sigma(d)$ for the same day. These have been done by using the linear model of prediction described by equation (7).

Table 1 presents the obtained results in the form of MAPE (Mean Absolute Percentage Error) and table 2 the maximum errors for the years 1990 - 1995. The data corresponding to 1990 - 1994 have been used also in learning phase while 1995 was used only in testing mode. The first column of data denotes the year of prediction, second column - the results of classical selforganizing network (CSO), third column - the fuzzy selforganizing network (FSO) and the fourth one - the 2 layer selforganizing network (2LSO).

Table 1 The maximum errors of prognosis

Date	CSO	FSO	2LSO
1990	21.45%	19.24%	20.32%
1991	18.44%	17.12%	17.35%
1992	17.78%	17.67%	16.93%
1993	13.21%	12.59%	12.36%
1994	15.45%	14.39%	14.46%
1995	13.34%	11.89%	12.04%

As it is seen, in all cases both MAPE and maximum errors have been reduced by applying more sofisticated methods of selforganization (fuzzy and 2-layer networks). The relative improvement of accuracy is in the range of 7-15%. Both fuzzy and 2-layer networks are similar and comparable in results. The interesting is that the accuracy of prediction for the most actual years is improving from year to year. The second important point of the results are small values of the maximum errors of prediction. All the time they stay on the acceptable level and are much smaller than obtained by using classical methods.

Very interesting is the distribution of 24-hour prediction of power for different seasons of the year. Fig. 1 presents chosen curves of the load prediction for four different days belonging to different

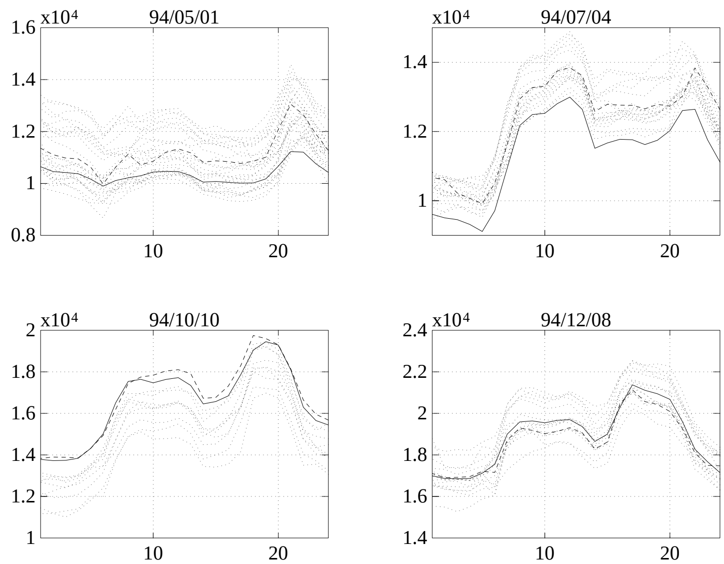

Figure 1: *The load curves of the days forming the base for prediction of the load for 4 different days of the year*

seasons of the year, taking part in prognoses. The solid lines represent the final prediction and the dashed line – the actual load of the appropriate days. The dot lines show the variability of the loads corresponding to the days taking part in final prediction. The prediction (solid line) is the result of the weighted average of the dotted lines and the actual trend. As it is seen from the figures the prediction of the data follows the real load changes and the discrepancy between both curves is acceptable, especially for workdays.

6 CONCLUSIONS

It has been shown, that selforganizing neural networks represent very good methods for short term load forecasting in the power system. The obtained average accuracy is above the level of 97% and exceed both classical method and multilayer perceptron approaches. The method based on the so called profiles is universal, flexible and insensitive to the global changes following from the development of the economy of the country.

It can be used for prediction of the energy consumption at any date ahead (even one year ahead). The accuracy of prediction is dependent on the availability of the past data. The network can be retrained any time with the additional actually obtained data base, allowing in this way to refresh the weights and adapt the system to the new conditions of the power system performance.

References

[1] S. Haykin, Neural networks, a comprehensive foundation, Macmillan College Publishing Company, 1994, N. Y.

[2] Osowski S., Neural network - algorithmic approach (in Polish), Warsaw, 1996

[3] Hagan M. T., Behr S. M., The time series approach to short term load forecasting, IEEE Trans. PWRS-2, 1987, pp. 785-791

[4] Brace M. C., Schmidt J., Hadlin M., Comparison of forecasting accuracy of n.n. with other techniques, Proc. Int. Forum on Appl. of N. N. to power systems, Seattle, 1991

[5] K. Lee, T. Choi, C. Ku, J. Park, Neural network architectures for short term load forecasting, IEEE ICNN, Orlando, 1994

[6] H. Mori, T. Ogasawara, A recurrent n. n. approach to short term load forecasting in electric power systems, IEEE ICNN, Portland, 1993

[7] T. Onoda, Next day peak load forecasting using ANN with modified backpropagation learning algorithm, IEEE ICNN, Orlando, 1994

[8] A. Germond, N. Macabrey, T. Baumann, Application of artificial neural networks to load forecasting, 1992 NATO Conference, Brussels, 1992

[9] S. Osowski , K. Siwek, A. Kadzielawa, Neural network approach to load forecasting, II National Conference on Neural Networks and their Applications, Karpacz, 1996

[10] M. Cottrell, B. Girard, Y. Girard, C. Muller, P. Rousset, Daily electrical power curve: classification and forecasting using a Kohonen map, in "From Natural to Artificial Neural Computation", J. Mira, F. Sandoval, eds., IWANN, Malaga, 1995, pp. 1107-1113

[11] T. Kohonen, Self-organizing maps, Springer Verlag, 1995, Berlin

[12] DeSieno D., Adding a consience to competitive learning, Neural Networks, 1988, pp. 117 - 124