Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 197–205
ISSN 1896-7094

Neural modeling of steam turbines

Paulina Sobanska and Piotr Szczepaniak

Institute of Computer Science, Technical University of Lodz
Wolczanska 215, 93-005 Lodz, Poland
pola@mnc.pl

Abstract. This study shows that neural networks are useful for creation of precise models of industrial objects such as turbosets. The complex dependencies of two cooperative turbines were successfully grasped by the neural network what is enormously beneficial with the lack of precise mathematical model. The estimated error of the neural model varies within the range of a few percents, which is a satisfactory result.

1 Introduction

This paper introduces the use of neural networks to modeling steam turbines [1]. It contains the description of modelled objects and the essence of relationships between the parameters of the model (chapter two). The third chapter comprises the description of the network and subsequent stages of creating the model. In the fourth chapter the research results are presented. The last chapter summarizes the application of neural model of the turboset and gives the final evaluation of the model's precision.

There are some examples in literature on using neural networks in power management systems. One of them is the seasonal prediction with error estimation of the Columbia River stream flow in British Columbia done by B. C. Hydro in Vancouver.

2 Technical problem

The steam turbine [2] is a flow machine, which transforms thermal energy into mechanical energy. The most important part of this machine is a blade system which enables the above mentioned transformation. The steam powering the turbine flows through the main stop valve which cuts off the steam in case of any emergency. Afterwards the steam flows through control valves which are manipulated by the regulator of rotational speed and, depending on the type of turbine, the regulator of power or steam pressure.

The object to be modeled (Fig. 1) consists of two components which cooperate with each other in a unique configuration [3]. It was designed and installed in one of Polish power plants. One of these components is a condensing steam turbine of type EKM and power of 6MW marked with a working number TG3. It is powered by the outlet steam from the second component which is the back-pressure steam turbine of type OPR and power of 10.5 MW (the working number is TG1). TG1 powers also the heat exchangers W1 and W2.

The live steam (from boilers) is supplied to TG1 which has valves at its inlet, regulating the flow of incoming steam and thereby its load. The electric power of TG1 depends on the amount of incoming steam (turbine's load). The outlet steam from TG1 is directed to the heat exchangers W1 and W2 which heat the water used in the district heating network up to the required temperature. The temperature of district heating water depends on the flow of steam reaching the heat exchangers, thereby on the steam's pressure. The regulation of this flow to gain the needed temperature is performed with control valves (2). The rest of outlet steam from TG1 is directed to TG3, where additional electrical energy is produced, depending on the flow of steam through TG3.

Before the modernization of this system there was only the back-pressure turbine which heated the district heating water with its outlet steam and produced electrical energy. In winter the need of outlet heat was great enough to maintain the incoming flow of steam at the stable level required for safe and continuous operation of this turbine. On the contrary, in summer this need diminished meaningly. As a result, the flow of incoming steam decreased disabling the operation of the turbine at all from time to time. It caused unacceptable breaks in the production of electrical energy in the generator. Using another condensing steam turbine supplied with the outlet steam from the first one, solved the problem. This turbine not only made the other one work continuously, but also produced additional electrical energy. Although the development of the system turned out to be favourable, it made the dependencies

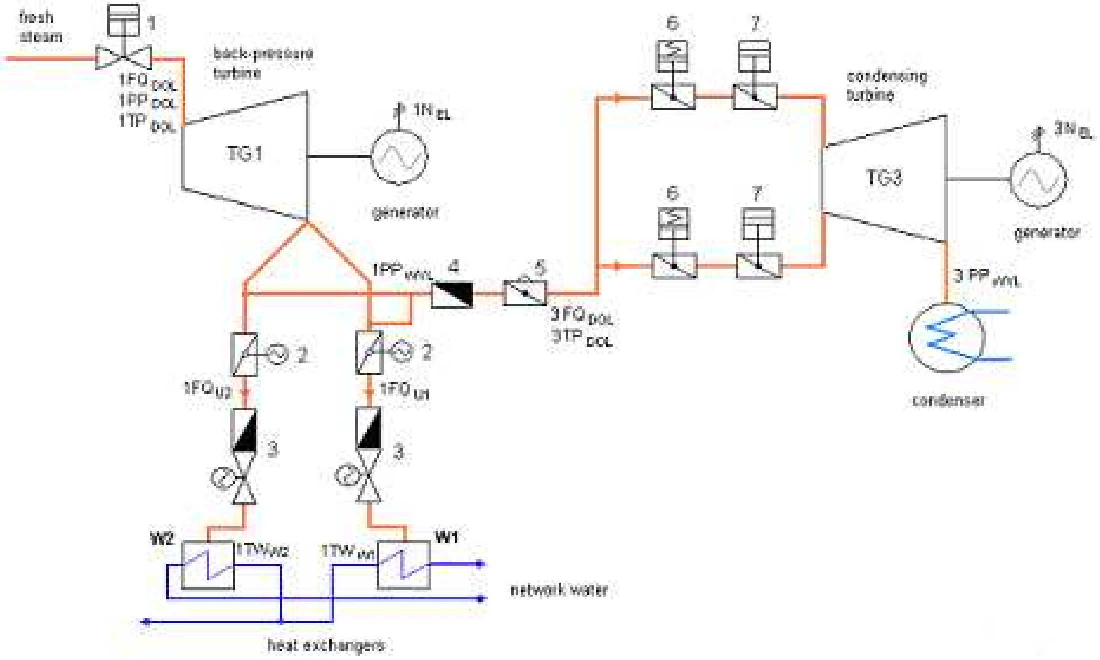

Fig. 1. The modelled turboset. 1- control valves in the steam inlet 2- control throttle valve in the steam outlet 3- cut off-check gate valve 4- non-return gate valve 5—cut-off gate valve 6—quick shut-off valve 7—control throttle valve of inlet steam to TG3

between the parameters more complex. Furthermore the lack of strict mathematical equations involving these parameters made their estimation difficult. For these reasons an effort was made to create a neural model of this object.

Table 1. The parameters of turbine TG1

Name of parameter	Description of turbine parameter	Type of signal
1 FQDOL	Steam flow to TG1	input
1 PPDOL	Pressure of steam powering TG1	input
1 TPDOL	Temperature of incoming steam	input
1 PPWYL	Pressure of outlet steam	input
1 NEL	Power of TG1	output
1 FQU1	Flow of extraction stream 1	input
1 FQU2	Flow of extraction stream 2	input
1 TWW1	Temp. of water in heat exchanger W1	input
1 TWW2	Temp. of water in heat exchanger W2	input

The dependencies between input and output parameters are very often complex and ambiguous. We have to deal here with non-linear and non-linearizable functions. The power of turbine, assumed to be the output signal, is described by equation 1:

$$1N_{EL} = 1FQ_{DOL} * 1H * 1\eta, \tag{1}$$

where $1FQ_{DOL}$—the flow of steam powering TG1, $1H$—the drop of enthalpy 1η—the efficiency of TG1.

The drop of enthalpy and the efficiency of TG1 are not constant values. These are described by equation 2:

$$1H = f_t(1PP_{DOL}, 1TP_{DOL}) - f_{ts}(1PP_{WYL}), \tag{2}$$

where f_t, f_{ts}—thermodynamic functions (steam tables), $1\eta = f(1PP_{DOL}, 1TP_{DOL}, 1FQ_{DOL}, 1PP_{WYL})$.

Table 2. The parameters of turbine TG3

Name of parameter	Description of turbine parameter	Type of signal
3 FQDOL	Steam flow to TG3	input
3 TPDOL	Temperature of incoming steam	input
3 PPWYL	Pressure of outlet steam	input
3 NEL	Power of TG3	input

General equation describing the power of TG3:

$$3N_{EL} = 3FQ_{DOL} * 3H * 3\eta, \tag{3}$$

where: $3FQ_{DOL}$—the flow of steam powering TG3, $3H$—the drop of enthalpy, 3η—the efficiency of TG3.

The drop of enthalpy and the efficiency can be described like in the case of TG1:

$$3H = f_t(1PP_{DOL}, 3TP_{DOL}) - - - f_{ts}(3PP_{WYL}) \tag{4}$$

where: f_t, f_{ts}—thermodynamic functions (steam tables), $3\eta = f(1PP_{DOL}, 3TP_{DOL}, 3FQ_{DOL}, 3PP_{WYL})$.

The flow of steam $3FQ_{DOL}$ to TG3 is the final result and depends on the values of flows going through the extractions of TG1—$1FQ_{U1}$ and $1FQ_{U2}$. The changes in flows $1FQ_{U1}$ and $1FQ_{U2}$ is the effect of the regulating system, maintaining the temperature of the network water at the set level. The relationship between these temperatures and the flows is ambiguous. The unregistered changes in network parameters (flows and pressures) have impact on fluctuations of the temperature of district heating water.

3 Method

There are many types of neural networks appropriate for modelling which makes the selection of the most suitable one difficult. In this research the dependencies between the parameters of the model are non linear so the application of non-linear neurons was undoubted. The complexity of modelled objects is great and the main objective was to achieve a model as thorough as possible so we decided to use a multilayer perceptron because this network has proved in a number of studies its usefulness in modelling of complex objects. Perceptron is a universal aproximtor which can project very complicated functions.

The network was taught with original data from the visualisation system of the power plant which can be seen in Fig. 2. The measurements were taken every second over the period of five months which gave an enormous number of records. Each record consists of all parameters, describing the object at the given moment, listed in tables 1 and 2. The preliminary data processing comprised the elimination of incomplete or incorrect records. Then the records were scaled according to the equation 5:

$$x_{skal} = \frac{x - z_{min}}{z_{max} - z_{min}}, \tag{5}$$

where x—scaled value, x_{skal}—value after scalling, z_{min}, z_{max}—translated minimal and maximal value (the translation guarantees scaling to open interval $(0, 1)$).

The training set included 315 records chosen from the database—every 1000^{th} record was taken into account. As the parameters of the model depend on the season of the year, this method of selection provided the variety of data (the records came from different months) which is so crucial for teaching neural networks. The test set included 96 records from the database—every 10000^{th} record was chosen. The diversity of the test records and the fact that none of them appeared in the training set, enabled preparing authoritative tests.

The necessary computation was done with the software written by the author of this study in the C# language. The trainig set as well as the test set filled the database (Microsoft SQL Server) which was also prepared by the author of this study.

At first, we decided to model the system with two neural networks of identical structure. It has much simplified the process of network's study. The input signals of both networks were exactly the same, but their output signals where different. The output signal of the first network was the power of TG1 and the output signal of the other one was the power of TG3.

Then we determined the initial architecture of the network which is illustrated in Fig. 3. We used only one hidden layer in our network because it has been proved in many works that one hidden layer

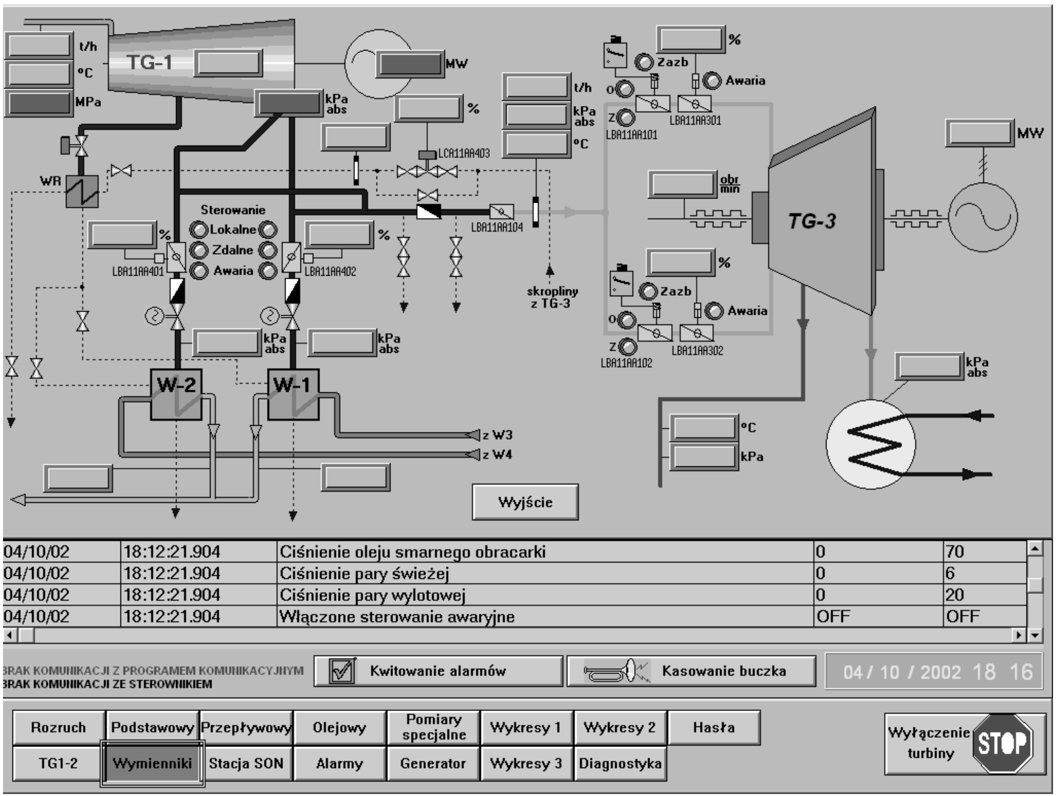

Fig. 2. The screen of the system of turbines' visualisation.

in perceptrons is entirely sufficient to project complicated functions. The number of input and output neurons was determined by the number of input and output parameters of the model. Another problem was the choice of the activation function and the number of hidden neurons. The initial number of hidden neurons was the effect of our experience and ideas found in literature [5] such as:

$$N_U = \sqrt{N_{WE} * N_{WY}}, \tag{6}$$

where: N_U—the number of hidden neurons, N_{WE}—the number of input neurons, N_{WY}—the number of output neurons.

The next step was to teaching the network according to the back propagation algorithm with coefficient momentum according to equation 7:

$$\Delta w_j^{(i)} = -\eta \frac{\partial Q^{(j)}}{\partial w_i} * \alpha \Delta w_i^{(j-1)}, \tag{7}$$

where: $\Delta w_j^{(i)}$—change of weight i in epoch j, η—the coefficient of teaching, Q — the error function, α—the coefficient momentum, $\Delta w_i^{(j-1)}$—the change of weight in epoch $j-1$.

We analysed the course of the network's error in successive epochs (where the epoch means one presentation of the whole training set of records), depending on the values of the BP algorithm parameters, to get the optimal model of the turboset, thus to select the most proper values of these parameters. There were the following parameters of the BP algorithm that we tried to match:

- the coefficient of teaching,
- the coefficient momentum,
- the number of hidden neurons,
- the activation function,
- the way of initialization of network weights.

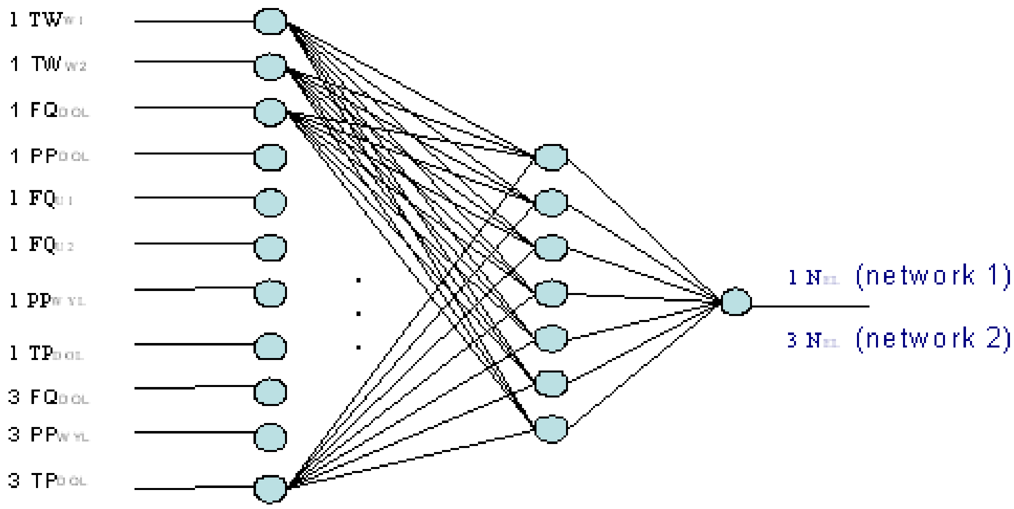

Fig. 3. Perceptron modelling the cooperation of turbines TG1 and TG3 All parameters in the picture are described in tables 1 and 2.

Changing these parameters had impact on the quality and speed of teaching the network. The error of the network in every epoch was calculated according to equation 8:

$$Q = \frac{1}{2}\sum_{j=1}^{N}(z^{(j)} - y^{(j)})^2, \tag{8}$$

where: Q—error function, $z^{(j)}$—expected output value, $y^{(j)}$—output value determined by the network.

The initial values of the BP algorithm parameters are shown in table 3.

Table 3. The initial values of the BP algorithm parameters

The number of hidden neurons	5
The activation function	Sigmoid unipolar function
The coefficient beta	0.8
The way of initiallization the weights	Random from the interval [0,1]
The way of teaching	On-line
The number of epochs	2000
The coefficient of teaching	0.9
The coefficient momentum	0.9
The acceptable error of the network	0.001

After finishing the process of teaching, the network was tested. The network error for the test set calculated according to equation 9 (Relative Root Mean Squares) was the measure of the correctness of the model:

$$Q = \sqrt{\frac{\sum_p \sum_j (t_j^p - y_j^p)^2}{\sum_p \sum_j (t_j^p - \overline{t_j})^2}}, \tag{9}$$

where: t_j^p—the expected value, y_j^p—the value determined by the network, $\overline{t_j} = \frac{1}{n}\sum_{p=1}^{n} t_j^p$—the mean expected value.

This method of defining the network error makes it independent of the number of test records and the order of magnitude of test values. Thanks to it, it is more authoritative than the mean squares error and helps better evaluate the model precision. In case of a good model, this error should fit in the interval $[0, 1]$. The smaller the error, the better the model.

4 Results

The sample final weights of neurons are illustrated in table 4:

Table 4. Final weights for seven hidden neurons in the model of the turboset

Nr	bias	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11
						Hidden layer						
1	-4.87	5.57	4.91	5.35	1.02	0.89	-0.31	-5.35	-1.83	-2.19	2.52	-1.42
2	-0.87	-0.31	-0.09	-0.65	-0.42	-0.32	-1.51	-0.05	-1.51	-0.44	-0.19	-0.14
3	1.72	0.48	-2.36	-6.59	-0.31	0.47	-0.58	1.78	0.47	-1.04	0.09	-0.57
4	-2.59	-2.55	-1.98	3.80	-0.59	1.20	-1.31	-1.32	-0.11	1.41	-3.93	-1.69
5	-0.36	2.43	-0.77	-5.90	-0.72	1.22	0.91	0.62	-2.04	-2.83	2.47	0.60
6	-9.23	1.13	0.09	9.72	0.89	-3.09	-0.04	-2.04	2.71	1.80	-1.97	-0.71
7	-4.43	-2.73	-2.96	9.81	0.01	1.09	-0.25	-0.96	0.08	-2.83	3.39	0.42
						Output layer						
1	-0.53	1.06	-1.62	-2.78	2.72	-2.61	2.15	1.28	-	-	-	-

The network precision evaluation (9) for the test set was 0,0674 and the network error (8) was 0.0100.

The sample course of the network error (8) depending on the number of hidden neurons is illustrated in Fig. 4 and Fig. 5.

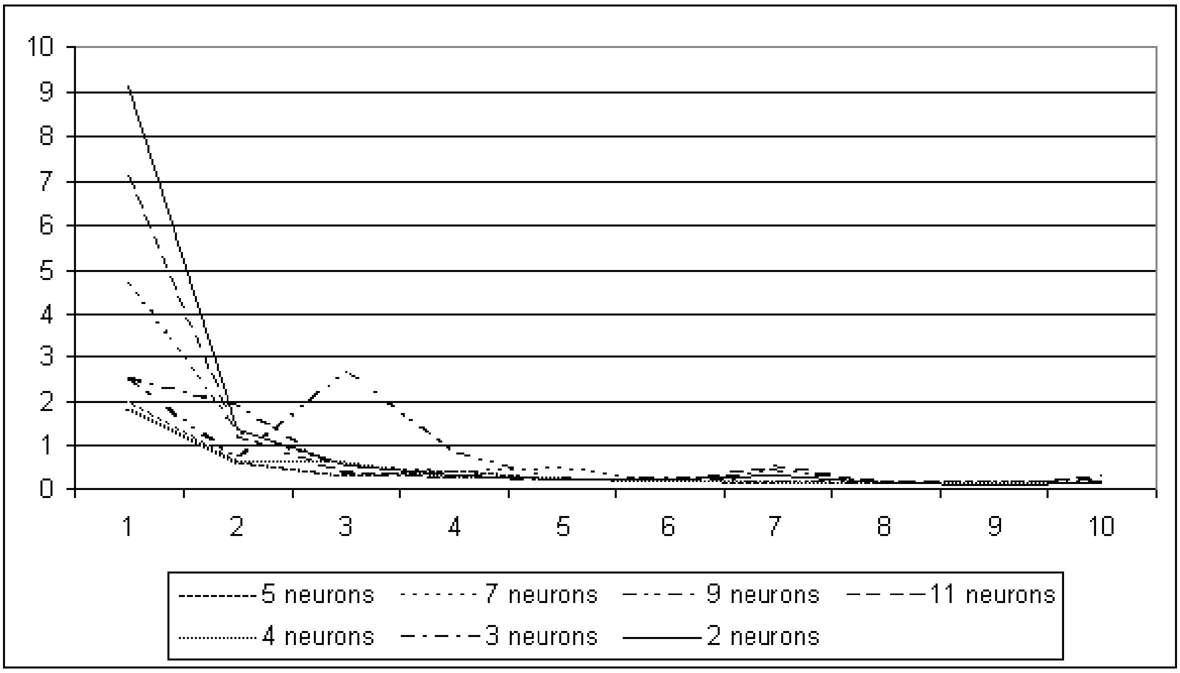

Fig. 4. The network error in epochs 1–10 in the model of the turboset for different number of hidden neurons

The results gained by the taught network are acceptable. They are illustrated in Fig. 6. The red series shows the real values of power of TG1 in the model of the turboset. The other one shows the values of power of TG1 determined by the network. The differences between these values are insignificant what testifies the high precision of the model.

The comparison of real values with these determined by the network is more visible when specific values are shown, thus some sample results of tests can be found in table 5. The first two columns contain

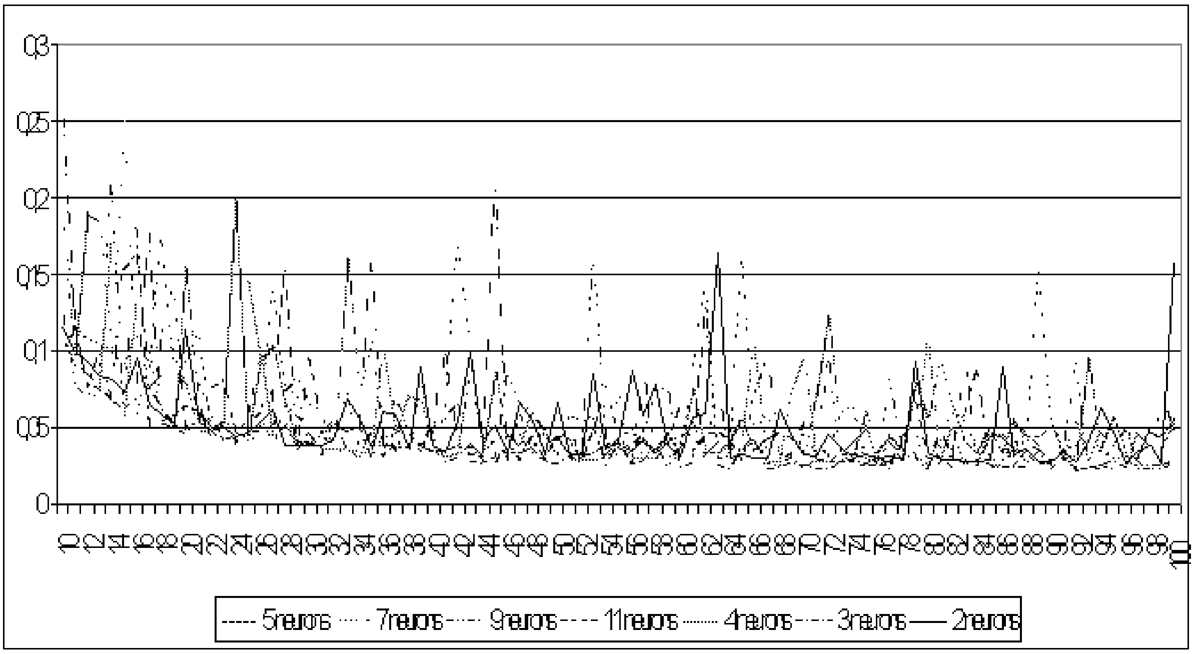

Fig. 5. The network error in epochs 10–100 in the model of the turboset for different number of hidden neurons

scaled values, in the third one there is a difference between the values from the first two columns and the last column presents unscaled values.

Table 5. The comparison between real power and the power determined by the network in the model of the turboset

Real power (scaled)	Network power (scaled)	Difference	Real power (unscaled)	Network power (unscaled)
0.3501	0.3525	0.0024	4.8120	4.8352
0.3448	0.3575	0.0126	4.7607	4.8825
0.7442	0.7330	-0.0112	8.6060	8.4979
0.4848	0.4828	-0.0020	6.1084	6.0891
0.3813	0.3909	0.0096	5.1123	5.2043
0.6750	0.6916	0.0166	7.9395	8.0996
0.6248	0.6297	0.0049	7.4561	7.5033
0.7160	0.7237	0.0077	8.3350	8.4088
0.5882	0.5722	-0.0160	7.1045	6.9502
0.7267	0.7394	0.0127	8.4375	8.5595

The sigmoid unipolar activation function in the network neurons gave acceptable results unlike the sigmoidal bipolar function or hiperbolic tagent. Other parameters of the BP momentum algorithm did not have such a significant impact on the process of teaching. However the number of hidden neurons, the coefficient of teaching and momentum had an influence on the gradient of minimizing the error function. Although it is said that the choice of initial weights should not be significant, we observed that the speed and stability of teaching was greater when initial weights were chosen from the interval $[-1, 1]$ rather than generated as the result of other initialization algorithms such as:

- initialization according to Smieja [4]—the weights of hidden neurons are chosen from the interval $[-\frac{\sqrt{n_{in}}}{2}, \frac{\sqrt{n_{in}}}{2}]$, where n_{in} is the number of neural inputs
- initialization according to Nguygen [4]—the weights of hidden neurons are chosen from the interval $[-\sqrt[n_{in}]{N_h}, -\sqrt[n_{in}]{N_h}]$, where n_{in} is the number of neural inputs, and N_h is the number of neurons in the layer

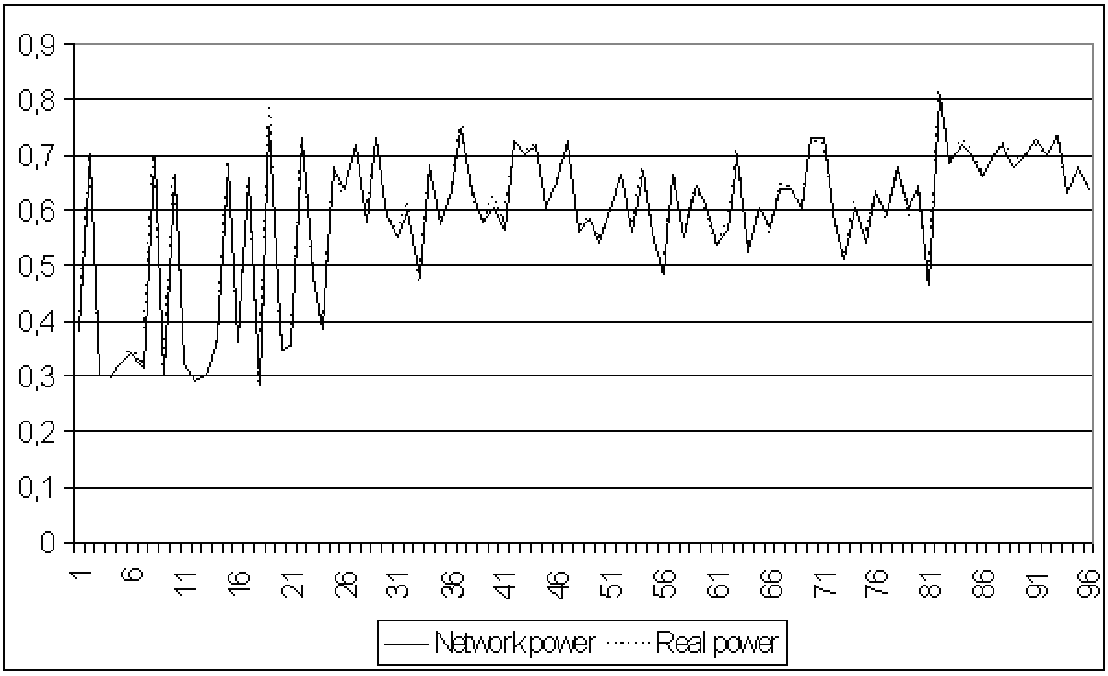

Fig. 6. The real power and the power determined by the network in the model of the turboset with seven hidden neurons

The course of the network error shows that there were oscilations at the begining of teaching which diminished in subsequent epochs to reach the minimum after about a thousand of epochs.

The research we have carried out led us to the parameters of the final model (table 6) which turned out to be more stable than other models.

Table 6. Parameters of the final model

The number of hidden neurons	4
The activation function	Sigmoid unipolar activation function
The coefficient beta	0.8
The way of initiallization the weights	Random initialization from the interval [-1,1]
The way of teaching	On-line
The number of epochs	1000
The coefficient of teaching	0.6
The coefficient momentum	0.9
The acceptable error of the network	0.01

5 Summary

The neural network has proved its usfulness for modeling of complicated objects with many parameters such as turbosets. The model quality (Relative Root Mean Squares) was 0.07 for the network with TG1 power as the output signal and0.1 when the TG3 power was the output signal.

The error of the model (defined as the difference between the expected power and the achieved one) for the real (unscaled) values of power was:

- 1.2%—when the output signal was the power of TG1
- 3.3%.—when the output signal was the power of TG3

The precision of the model is adequate for the analysis of the operating conditions of the turboset, defining optimal conditions for the turboset and predicting the heat and electical energy requirement in different seasons of the year. In the future, the experience gained during modeling steam turbines with neural networks might be essential while modelling the regulation systems of the turbosets and creating networks working on-line in the supervisory control systems.

6 Bibliography

References

1. P. Sobanska: *Neural modeling of industrial objects,* MSc. Thesis, Technical University of Lodz 2005.
2. A. Andrzejewska: *Instrukcja obslugi regulatora elektrohydraulicznego turbiny TG1 w EC Lubin,* opracowanie INWAT, Lodz 2002 r.
3. A. Andrzejewska, T. Jach: *Opis regulacji turbozespoż TG1 i TG2 oraz czonu kondensacyjnego TG3,* opracowanie INWAT nr ewid. 475/02, nie publ., Lodz 2002 r.
4. S. Osowski: *Sieci neuronowe w ujeciu algorytmicznym,* WNT Warszawa 1996.
5. `http://republika.pl/edward_ch/sneuro.html`