STUDIA INFORMATICA
Nr 1-2 (21) Systems and information technology 2017

Dariusz RUCIŃSKI

PhD student at Institute of Computer Science, Faculty of Science,
Siedlce University of Natural Sciences and Humanities,
ul. 3 Maja 54, 08-110 Siedlce, Poland

The neural modelling in chosen task of Electric Power Stock Market

Abstract. The work contains selected results of the neural modelling for the Electric Power Exchange (EPE) for the Day Ahead Market (DAM). The paper contains description of the neural modelling method, the way of preparing (pre-processing) data used for leaning of Artificial Neural Network (ANN), description of achieved neural models of EPE, the comparative study results and the sensitivity study results. The results which was obtained was interpreted and discussed in the systemic category.

Keywords: neural modelling, neural network, electric power stock market

1. Introduction

The main target of modelling is to describe some part of reality (object, system, process, etc.), extracting the most important features according to some established criteria, by using some tools. Such approach let us to create simplified and generalized image of the original.

For building the models different methods are used, such as analytical methods, identification methods, neural methods, evolutionary algorithms, fuzzy logic etc. All of them possess specific and well-defined mathematical structures [19, 27, 33, 35]. One of the modelling method leads to obtain the neural models is Artificial Neural Networks (ANN) which is included into the artificial intelligence [20, 32, 39]. The Modelling of ANN is associated with the initial preparation of training and testing data, ANN architecture construction, choosing the ANN structure, choosing parameters and rules of learning and finally learning the system model.

In the research work is able to model systems, facilities, processes, signals, events, etc. which occur in the real world. In this paper the EPE system is modelling by using the ANN tools.

There are research woks concerning the neural modelling of Electric Power Stock Market in Poland like [11] which examine the modeling of EPE system in a classical way. It means to build the model by chose the important factors for forecasting or examining the system. Those paper does not apply such models it deals with the search of the system model as an equivalent schema similarly like in the theory of electrical circuits. Such approach don't require detail analyzing of system factors and further estimation, it trays to take the possible minimum number of factors (in this case only two) and examine the model whether it is able to represents the system.

The Day Ahead stock market data is used for learning and testing the model based on ANN, and as a result the neuronal model was obtained which can be treated as alternative schema of EPE market [8, 15]. Learning ANN model of EPE system require pre-treatment data obtained from the DAM (including their normalization), it is both the input data as the 24 size of variables being the delivered and sold volume of electricity energy (ee) in each hour of the day, as well as the size of the 24 output variables being the average price obtained in each hour of the day [22, 25] for supplied and sold energy.

For neural modelling of the EPE the ANN was selected. Selected model base on a one-way perceptron artificial network with back-propagation errors. Mean Square Error (MSE) as a measure of the model quality to the system was assume.

2. The research problem definition

The work "*Electricity markets. Selected technical and economic aspects* " emphasized the needs of seeking the new models of electric market system showing the evolution from a centralized system to a decentralized system [17].

During forecasting the price of electricity on the electricity exchange market the diffusion models of "peaks" and simulations of Monte Carlo method are willingly used. The process of determining prices by using a random variable described Markov chain with n possible states is also widely used [10]. There is also so-called general approach to statistical modeling by using parametric models such arx, armax, ar, etc.

In the case of the simulation models it is important to properly build the existing system, and the final solution is found by changing the model in such a way that it works in the revised scale in time and/or space, allowing to identify interactions and behaviors that would otherwise be lost.

Among the time series models the three general groups of models are distinguished, that is: economical stochastic models, which consists of models of autoregressive (AR), models of moving average (MA), models of autoregressive moving average (ARMA), models of autoregressive integrated moving average (ARIMA) [5,4], GARCH models [9, 14, 40] and many others.

The above models can only use the historical observations for established the forecasted variable (ie. the historical data) or exogenous variables (eg. Models ARX/Arimax) [4]. The current methods of modeling (EPE) in a comprehensive manner are written up in "*Modeindelling Polish electricity market*" [21].

Among the currently used methods other attempts take place, such as statistical modeling by design Markov models [13, 12, 6], returning to the average jump-diffusion models (mean reverting jump-diffusion models) [1,2], and even the building of hybrid models combining many classical methods [29].

Generally, in analysis and identification modeling , it is assumed that the greatest impact on the prices of electricity are: the volume of electricity trading (historical and projected), the unit price of fuel, the power of power plants, transmission capacity power grid, the weather variable (temperature, the degree of sunlight, wind speed and direction, rainfall intensity, etc.) and so-called calendar variable (season, time of day, etc.) [38].

All analytical models and identification models are very important and widely used in practice, especially in the field of forecasting electricity prices, but unfortunately they are burdened with a high degree of risk of inaccuracy, hence the needs for a different model that would be able to more accurately predict the price of electricity on the Polish Power Exchange . One of the directions of research is the ability to use models as substitute schemes for EPE system, in which the structure of model would be appropriate to the actual structure of the system, in this case, the structure of EPE for DAM.

As a promising methods considered the identification and neuronal methods which are used to obtain system models based on the input and output data (passive or active) [32]. Therefore, artificial neural networks (ANN) are designed in analogy to the structure of the delivery process and sale of electricity in the different hours of the day, and then on the basis of such data like for example Day Ahead Market (DAM) taught them the EPE system.

Preparation of data to properly learn the system model of the neural network requires a lot of preparatory processes, which is called cluster analysis.

The first treatment on the basis of the data standardization concerned the missing and duplicate data eg. Due to the change from summer time to winter and vice versa[1]. But in this case the time series for specific hours were not built and the identification of the structure of the stochastic time series were not performed as well, and thus the certain hypotheses regarding the specific tests ware not defined, eg. stationary time series.

In this place a completely different research approach based on artificial neural networks were used, to determine to see how much (on what level of accuracy) the ANN is able to learn the EPE system as a whole, without the needs for analysis of the components of the system.

The problem of neural modeling of EPE system was taken some tines ago in [30] and it is still valid, just mention here work "*Electricity price forecasting of deregulated market using Elman Neural Network*" [37], in which the authors shared the results of research on trying to build a neural model for forecasting electricity prices using artificial Elman neural networks, or work "*Test the quality of prediction of load electro-energy*" [20], which shows the results of comparison of prediction of loads methods in a small power system in Poland with the use of neural models SVM, RBF and MLP in relation to the naive prediction. Research work was carried out in MATLAB . Two tasks ware carried out for one and 24 hours prediction. The authors concluded that neural networks are an effective tool in predicting loads in the power system, especially in small areas, where significant fluctuations of charges by the hour are observed.

Similarly the objective of conducted research was, among others, to check whether well designed artificial neural networks, it means adjusted to the structure of EPE, can learn the EPE system sufficiently to fit the model to the real system without needs for fragmentally analyzing the system and without statistical calculations burdened with large computational errors. As already mentioned the data used to learn the neural model of EPE system were taken from fixing of DAM 1, Polis Power Exchange SA and include transactions with hourly period from 01.01.2015 to 30.06.2015, in order to maintain six full monthly cycles. Therefore a different point of view were adopted than in the previously used statistics research. The approach used in this paper base on control and systems theory somewhat similar to the

[1]In the case of missing values determined average of two adjacent values, and in the case of the occurrence of duplication of data was erased one of the values.

approach used in the work of Ciechulski and Osowski [3]. It was assumed that the model of EPE system in terms of equivalent diagram of the system is searched.

Consequently as the input side only the volume of electricity delivered and sold on the stock exchange was taken, and as the output side only the average price achieved for each hour of the day was taken. The fact that the volume of power and price were formed in a suitable manner as at any hour of the day as well as on each day of the examined period was dependent on the number of factors, including and the temperature, requirement, etc.

3. Input and output data for ANN learning

The data used for ANN learning were achieved from Electric Power Exchange for Day Ahead Market [42]. As an input data stream the volume of delivered and sold electric energy (ee) in particular hours of the day in 2015 was assumed, it is indicated as $\mathbf{u}_i$ [i=1-24], as an output data stream the average price obtained in the particular hours of the day for sold volume of the (ee) was assume, it is indicated as $\mathbf{y}_j$ (j=1-24) [PLN/kWh].

Quotations on the DAM take place according to schedule every day, contracts are finalize for 24 or 48 hours prior to delivery to the buyer ee [17, 31]. Contracts are finalized to deliver for a pointed hours of the day. Each participant of EPE possess own account (portfolio) and may submit any number of orders. Each order specifies, among others, type of code instrument, type of order (buy or sell), a portfolio of orders, volume (the amount of bought or sold energy), and so on [17,16].

The rules of concluding exchange transactions are defined, among others, in *Trading Rules Exchange Commodity Market of the Polish Power Exchange SA*, *the Trading Rules for the weekly program forward instruments for electricity and the Detailed Rules for the markets operated by PPE SA* [42].

Due to the hardware and software limitations, and available research tools (MATLAB and Simulink environment) in this study we used a period of 6 months (181 days) of DAM trading, that is, from 01.01.2015 r. to 30.06. 2015. The input volume delivered and sold to DAM is a 24 x 181 dimensions matrix (**u** [kWh]), the output size is the average electricity price obtained for each hour of the day and it is a 24 x 181 dimensions matrix (**y** [PLN/kWh]). Above **y** matrix are elements of the learning and testing sets.

4. Architecture and learning of ANN the EPE model

For implementation of EPE model the ANN was constructed. Implemented as one direction , three layer perceptron's network consisted with input layer, hidden layer, and output layer. Assumption of the ANN construction is, that ANN is the system in which architecture is its own structure and processes on that structure is a result of stream flow from input to output by the structure [22,33].

In a consequence the layers of ANN are treated as a subsystems where weights and biases are considered as proportional members from the systems and stirrings theory point of view and links between the layers are treated as feedback connecting subsystems [18,36].

As a learning method the back propagation algorithm was taken [20,28]. The simulation was performer in the MATLAB and Simulink environment with usage of Neural Network Toolbox(NTT) and with embedment MATLAB script language [7,26]. For learning model the neural perceptron ANN was taken, which is shown on Figure 1. It possesses three layer of neurons with 24 neurons in each of it, with a consequences it gives the structure of neural model consisted with two layer of weights.

The following designations was:

W1 – The first layer is consisting with weights which connect all neurons with the input layer and the hidden layer neurons,

W2 – The second layer is consisting with weights which connect all neurons with the hidden layer and the output layer neurons,

$u_1 \div u_{24}$ – input quantities for the volume of electricity sold to EPE in the next hours a day,

net_{ij} – the sum of the products of weights and values of input quantities for the i-th hour and for the j -th layer,

$f(net_{ij})$ –activation function for net_{ij},

y_{ij} – the output vector for the j -th layer, the layer 2 are the average unit prices obtained in each hour of the day for sold ee on the PPE S.A. , for first layer the vector are unnamed.

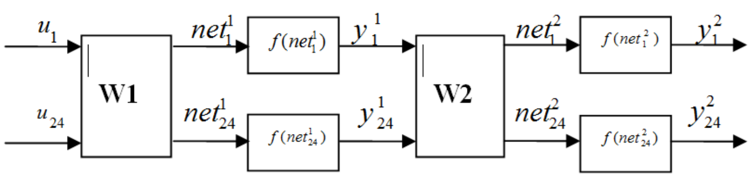

Figure 1. EPE neural model for DAM as a ANN perceptron network schema. Symbols in the text. Source: own work with NNT notation in MATLAB environment.[7].

The neural model consists of adder and the mapper, wherein the adder is the sum of products of weight and input values for each layer, and the role of the mapper fulfill the activation function, which are described in [24,28].

Generally learning of ANN lead to modify the weights of the respective layers in order to obtain a model similar to the real system. The object of analysis can be the weights after the learning process. They are therefore (the weights) an important element in defining the relationship between the input elements, the space which defining the hidden layers, and its output. The weight transforms m-dimensional input space to the n dimensional output space, which corresponds to "synapse" biological brain model [28].

Generally, the relationship between the elements of the system are defined as closely related (such a relationship describes the function), independent (in this case the relationships are random) or dependent in some range, such object is a subject of artificial intelligence research, The artificial intelligence research includes, among the other, algorithms based on artificial neural networks, evolutionary algorithms, and fuzzy systems [24].

An important question is, what information about the system possessed the model weights. If the same weights can be considered as the subject of the system analysis in order to find or to discover its essential properties. From a mathematical point of view, they are matrix defining the relationship between the values of input vector, hidden vector and output vector, so it is assumed that the individual elements can be the base to determine the most important connections between the model neurons (knowledge is represented by the values of the respective weights). It can show, which of them has the greatest effect on the neuron activation.

It is also important question if the weights are only unnamed coefficients (in terms of theory of proportional members steering), or depending on the model they have specific name, and therefore the specific properties as well which enable their physical interpretation, not to mention the fact, that in terms of control and systems theory the weights can be treated as other members, such as. Integral members, Derivative members, oscillating members, etc. [18, 34, 36]. The knowledge interpretation which is contained in respective weights can be carried out by using the activation function of the neuron model. In this paper it is the sigmoid function defined as follows [30]:

$$y(net) = \frac{2}{1+e^{-2\cdot net}} - 1, \tag{1}$$

which, after the logarithm and the transformation can be represented

$$net = \frac{1}{2} \cdot [\ln(1+y) - \ln(1-y)]. \tag{2}$$

The equation (2) can be further developed in the Maclaurin series [23,31] expressions:

$$\ln(1+y) = y - \frac{y^2}{2} - \frac{y^3}{3} - \ldots, \tag{3}$$

$$\ln(1-y) = -y - \frac{(-y)^2}{2} - \frac{(-y)^3}{3} - \ldots,$$

which limited to three major elements and after some simplifications lead to dependence as follow:

$$y = net = \sum_{i=1, j=1}^{n} w_{ij} \cdot u_i = w_{11} \cdot u_1 + w_{12} \cdot u_2 + \ldots + w_{124} \cdot u_{24}, \tag{4}$$

from equation (4) the nature of the knowledge can be interpret by examining the individual weights (weights can be also be seen as elements which transforming the inputs value into outputs value, in this case [MWh] into [PLN]), which requires the appropriate assumptions about the nature of the analysed phenomena. However, due to the fact that the individual inputs ($\mathbf{u}_i$) represent the volume ee delivered and sold into EPE and **y** received average prices, so the weights in this particular case can be interpreted as [PLN/kWh^2].

5. EPE neural model distinction

As a result of the project of perceptron artificial neural network architecture and perform the learning process using backpropagation the neural model was obtained. ANN is composed of two types of equations: the equations defining the stimulation of artificial neural network (combiner weights multiplied by the input signals) and the equation of response of the network on the output signal. Because of the design model in a single hidden layer neurons obtained by learning described two layers of weight matrices **W1** and **W2**, both measuring 24 x 24, which possess the corresponding values for the layer 1:

$$\mathbf{W}_1 = \begin{bmatrix} -0{,}039464 & -0{,}16003 & 0{,}064382 & ,\ldots, & 0{,}2364 & -0{,}61574 & -0{,}11246 \\ \ldots, & \ldots & \ldots & ,\ldots, & \ldots & \ldots & ,\ldots \\ 0{,}34381 & 0{,}22651 & -0{,}18017 & ,\ldots, & 0{,}16676 & -0{,}072057 & 0{,}13359 \end{bmatrix}, \tag{5}$$

and for the layer 2:

$$\mathbf{W}_2 = \begin{bmatrix} 0{,}28888 & 0{,}58709 & -0{,}41224 & ,\ldots, & 0{,}2202 & 0{,}60631 & 0{,}21501 \\ \ldots, & \ldots & \ldots & ,\ldots, & \ldots & \ldots & ,\ldots \\ -1{,}3183 & 0{,}090753 & -0{,}3071 & ,\ldots, & 0{,}73966 & -0{,}76636 & 0{,}2740 \end{bmatrix}, \tag{6}$$

wherein the learning process was carried out by using the input and output normalized values. The normalized input data for the first layer:

$$u_{i(1-24)} = [0{,}08499483 \quad 0{,}08512609 \quad 0{,}08696089 \quad ,..., \quad 0{,}05593292 \quad 0{,}0385510 \quad 0{,}0492389]$$

and corresponding normalized output learned values:

$$s_{i(1-24)} = [0{,}0366824 \quad 0{,}03531393 \quad 0{,}03471317 \quad ,..., \quad 0{,}05128824 \quad 0{,}05169401 \quad 0{,}04245519]$$

So, the general form of the output value y (t) for each layer k of neurons is expressed as follows:

$$\mathbf{y}_i^k(\mathbf{t}) = f_i^k(\mathbf{net}_i^k) \tag{7}$$

wherein:

$$\mathbf{net}_i^k = \sum_{j=1}^{j=m} w_{ij}^k \cdot u_j^{kT}. \tag{8}$$

6. Calculation example

For the first layer of neurons (hidden layer) the value obtained of ANN simulation of the first neuron is:

$$net_1 = -0{,}039464 \cdot 0{,}08499483 + ... + 0{,}34381 \cdot 0{,}04923891 = -0{,}01273394\ 709429 \quad (9)$$

so the activation function as a function of reproducing the result of stimulation of the neuron on the output of the first neuron takes the form of a sigmoidal according to the formula:

$$y_1^1(net_1^1) = \frac{2}{1+e^{-2\cdot net_1^1}} - 1 = \frac{2}{1+e^{-2\cdot(-0{,}01273394709429t)}} - 1 = \text{-}0.0127\,. \quad (10)$$

Due to the fact that the output signals from the first weight layer W_1 are also the input value to the second layer W_2, in the same way the neurons of output layer and the output signals of the second layer are determined. The second layer outputs is also a neural model of EPE output, thus:

$$net_i^2 = \sum_{i=1,j=1}^{i=n,j=m} w_{ij}((\frac{2}{1+e^{-2\cdot net_i^2}} - 1)_j^2)^T \quad (11)$$

The activation function of the second layer is a linear function, thus obtained:

$$y_i^2(net_i^2) = net_i^2\,, \quad (12)$$

therefore, the output of the first neuron in the second layer of neurons (output layer) can be expressed as follows:

$$y_1^2(net_1^2) = \sum_{i=1,j=1}^{i=n,j=m} w_{ij}(\frac{2}{1+e^{-2\cdot net_1^2}} - 1)_1^2. \quad (13)$$

The next step is comparing the result value of the model output y (net) with the reference value (learning with the teacher) to determine the error learning, the error MSE:

$$MSE = \sum_{i=1}^{i=n}(y_i - s_i)^2\,, \quad (14)$$

It is worth to note that the learning process is to change the weights of the two layers. Changing the weights by using the backward propagation method is to calculate the decline of the value of the error, and more precisely the speed of decreasing of the error, which involves with the determination of Hessian. Because of use (during the learning) Levenberg-Marquardt algorithm the approximated value is determined as the product of the first derivative of the Jacobian (J) and the transposed J:

$$H = J^T J \,, \tag{15}$$

the gradient of the error has been determined as the product of the transpose of the first derivative and scalar mean square error MSE:

$$grad = J^T MSE \,, \tag{16}$$

change of the value of individual weights in the learning process by using Levenberg-Marquardt algorithm has been determined by the following equation:

$$w_{ij(k+1)} = w_{ij(k)} - [J^T J + \mu I]^{-1} J^T MSE \,, \tag{17}$$

where:
μ – is arbitrarily selected factor (scalar) decreasing in the course of learning.

7. MATLAB and Simulink environment, including Neural Network Toolbox and MATLAB language

For the design and training the ANN the MATLAB and Simulink environment was selected. Due to its high computing power and the practical implementation ability. There are also other environments used in the designing models using artificial intelligence methods, eg. MATHEMATICA, STATISTICA, SAS, STATA, or SAP. Due to the implemented solutions, engineering and economic considerations, as well as access to papers and user manuals to help through the process of creating model development of EPE computing environment MATLAB and Simulink was selected.

In the process of designing the architecture of ANN and learning it the model of EPE in MATLAB and Simulink environment MATLAB language and a library NNT was used, which requires to define the following algorithm steps:

- declare ANN, in this case using the function: feedforwardnet (hiddenSizes, trainFcn), which has two optional arguments: hiddenSizes (number of neurons in the hidden layers) and trainFcn (learning function),
- select the activation function for each layers in this case for the first and second layers, respectively, „*tansing*[2]” and „*purelin*[3]” activation function was selected,

[2] tansing(n)= 2/(1+exp(-2*n))-1, where: n – input signal value.

[3] purelin(n) = n (linear function), where: n – input signal value.

−declaration of ANN property, eg. using configure, which arguments are the input data, output data as learning data. For example net=configure(net,'outputs',t,i), which initializes the weight and makes shifting of the activation function (the so-called. "bias").

The reinitialization of weights is possible by using the net=init(net), and the command used to initialize learning ANN is train(), [net, tr]=train(net,X,T,Xi,Ai,EW)[4], which returns ANN object available under the net variable, net object returns among the others such properties like, matrices values, vectors values, weights values and biases values.

In the MATLAB environment is possible to initialize and implement different types of networks, uni- and multi-directions, single and multi-layer, for example cascadeforwardnet which forms the unidirectional cascade, it wok similarly to feedforwardnet. The difference lies on the connection weights to each input layer of the network and the weight of each layer to the each next layer of the network. This architecture sometimes improves the learning speed network. Another type of ANN is patternnet network, which is also similar to the feedforwardnet network except that the last network layer is initialize by ‘tansig’ transfer function instead of 'purelin' [20].

The process of learning ANN with a teacher, implemented at this work, requires two data sets, a set of input data and set of output data. The output data consists of the so-called two subsets learning data and testing data.

ANN learning process is run to adjust the weights to most closely coincided output model data and learning system data. In this paper, the quality of training model is measured by the mean square error (MSE).

The ANN learning process can be performed by using two methods: incremental and batch methods . Incremental method is to calculate the gradient of weight and how to update its value after processing the ANN for each pair learning (output,teacher) .

The batch method whole data is processed in the learning network and after then the weight are updated. For learning the ANN model of the EPE system the trainlm function was

[4] train(net,X,T,Xi,Ai,EW), where: net – net name, X – input matrix, T – source value (teacher value), X_i – initial input delay of ANN, A_i – initial input delay of hidden layer, ΔW – error weights.

selected. Trainlm function is an implementation of the Levenberg - Marquardt algorithm considered as an effective and quick method.

8. Research experiments

For ANN model of EPE system learning functioning the data of DAM were used. The data of DAM contains the period from 01.01.2015 to 30.06.2015. There was adopted 24 input variables representing the volume of electricity [kWh] sold in each hours of the day and 24 size output variables representing the average electricity prices obtained in each hours of the day, expressed in [PLN/kWh]. Input and output data were normalized according relations (2). Due to the normalization method used for both input and output values it become unnamed figure.

As a method of learning model the "trainlm" function was used, it implements of the Levenberg-Marquardt (LM) algorithm. "*Tansing*" function activated the hidden layer of ANN and "*purelin*" function activate the output layer of model. The ANN architecture was designed and learned in the MATLAB and Simulink environment with using of NNT.

The learning of ANN took place as follow, first the weights in each layers were initiated by random values, then they were subjected of the learning process in accordance with the adopted strategy, in this case the data obtained from the model(output) were compared with reference (teaching) data, the difference between the data was the basis for modifying the weights in accordance to LM back-propagation algorithm.

The model was learned until, one of the conditions of its stop occurred if is eg.: achieving relatively small MSE, achieving maximum number of transition periods, obtaining a minimum changes of drop gradient weights, obtaining a predetermined number of periods after which there is no improvement of model characteristic, etc. In the case of our learning model of EPE system the reason for stopping the learning process was to achieving a set number of periods, after which there is no improvement in the model, in this case it was 6 epochs without improvement.

As a result of ANN learning the EPE system neural model was received, which in the fourth epoch received optimal parameters, which means the next era of learning didn't significantly improved its quality, and the error learning declined in the whole process of learning about two orders of magnitude - Figure 2. The data on the basis of which the network was taught was divided by the mechanisms of MATLAB in three sets: train, validation and test (a detailed description of the mechanism of training network is beyond the scope of the work).

During the study of adaption of ANN model in depending on changes the network parameters the simulation of the learning process and the quality of the resulting model depending on the number of neurons in the hidden layer was carried out. Results of the carried simulation of network learning for different numbers of neurons in the hidden layer, that is, from 6 to 48 are shown in Figure 3. The process of training the ANN model system TGEE proceeded relatively quickly, regardless of the number of neurons in the hidden layer. Changes of the weights in each case from the fourth epoch were small, so the number of neurons in the hidden layer does not have a major impact on the speed and quality of learning.

The neuronal model of EPE was also examined from the point of view of the influence of number of neurons in hidden layer on the quality of regression. Generally the model reproduces sufficiently the real system if the value of regression is close to one. The influence of the number of neurons in hidden layer on the quality of regression is shown in Figure 4. The course of the dependence shown in Figure 4 shows that the examined model reveal a high stability in the tested range of neurons from 6 to 48 in the hidden layer. The regression is stable in the range of 0.75-0.85.

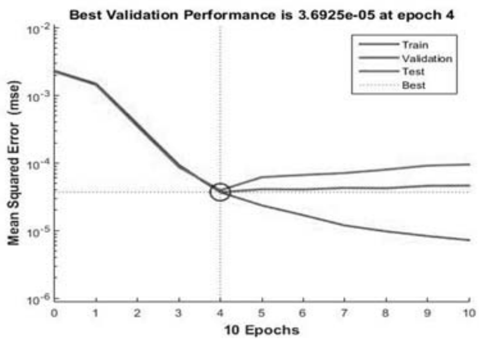

Figure 2. The course of ANN learning the EPE system. y- Mean Square Error value for a set of training, validating and testing, x-axis the number of epochs. Source: own work in the MATLAB environment using the NNT [7].

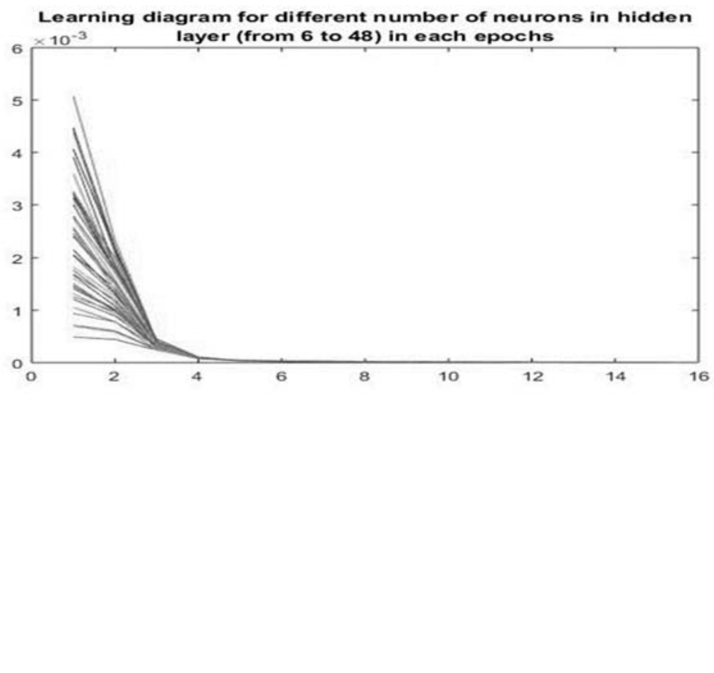

Figure 3. Mean Square Error value in depending on the number of neurons in the hidden layer. y- Mean Square Error value for a train set number of neurons in hidden layer network ANN, x- number of epochs. Source: own work in the MATLAB environment, the use of NNT [7]

You may also notice a single "peak" for a specific number of neurons in the hidden layer. The reason for such phenomenon is a random weight values initiating during the learning process, Due to such phenomenon the learning process of ANN should be performed a few times, in order to avoid a particularly adverse effect initiation of the weight values at the beginning of the learning process.

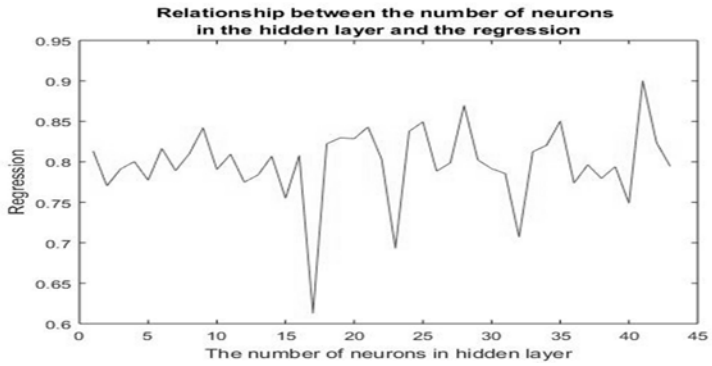

Figure 4. The course of the dependence of the regression, depending on the number of neurons in the hidden layer. y - the regression coefficient, x - number of neurons in the hidden layer. Source: own work in the MATLAB environment, the use of NNT [7]

9. Discussion and results interpretation (model assessment)

As a result of learning the modified perceptron neural network with some system solutions the EPE neuronal model was obtained The quality of model representation with respect to the source (teaching) data is presented in Figure 5.

The course of the signals shown in Figure 5 due to the difference of the individual signals obtained from the ANN model in relation to actual data of EPE system. The scope of discrepancy ranges from 0.015 to -0.015, and the value of individual vector components oscillate around zero. So the difference between the normalized actual price on the EPE of DAM and the normalized price obtained from ANN model for individual days and hours falls within the range of 0.015 to (-0.015) normalized value prices.

Selected discrepancies between responses of models and systems for selected hours, ie. the highest for 6 p.m. the smallest for 11 p.m. and average hours for 5 a.m. is given in Figure

6. The relationship between the output of the EPE system and the output of NNT model for the first vector of average prices for the 0-1 hour is shown in Figure 7. As can be seen the model reacts with a lag to system changes, but it relatively faithful reproduces its changes of value. Due to the fact that the waveforms for the other hours of the day are similar to 0-1 hour, their presentation is abandoned. Presenting the course as shown in Figure 7 shows an important feature of SSN, which is the ability to knowledge generalize, ie. reasonable interpolation of teaching signals.

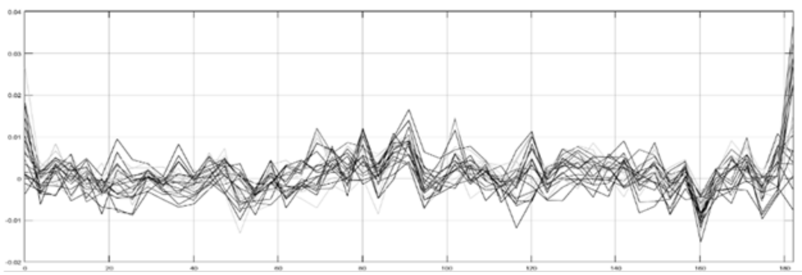

Figure 5. The degree of accuracy mapping ANN model to the real TGEE system measured as the difference between the values for the output of the system and the values for the output of the model. Symbols: x-axis - another day of the period, the y axis - Mean Square Error value for each hour day (24 hours).. Source: own work in the MATLAB environment, the use of NNT [7]

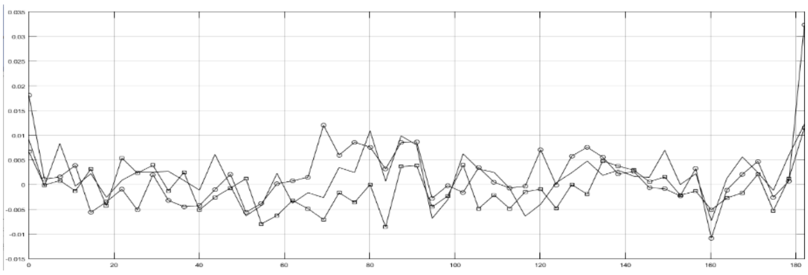

Figure 6. The difference between the reference(source) data and the data obtained from the model for three characteristics periods, the biggest5- 6 pm, the smallest 10-11 pm and mean 4- 5 am. Symbols: x - the number of learners values, y – Mean Square Error. Source: own work in the MATLAB environment, the use of NNT [7].

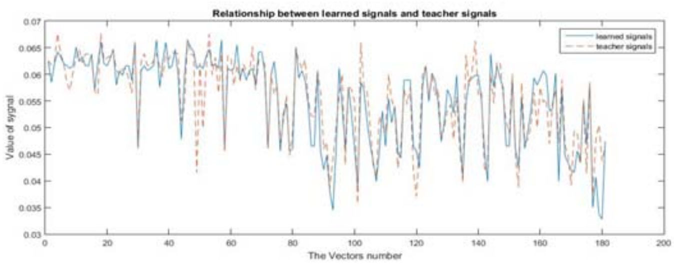

Figure 7. Comparison of the values of learners (from the system), and the values obtained from the model for the price of the first hour of the day. Symbols: x-axis - another day of the period (the number of training vectors), the y axis - normalized output value obtained from the ANN model (solid line) and the normalized output value of the EPE system (dashed line).. Source: own work in the MATLAB environment, the use of NNT [7]

10. The results, conclusions and directions for further research

Artificial neural network was designed, and taught EPE system for data for the Day Ahead Market , and the next implemented in Simulink as a neuronal model of EPE.

The ANN model was learned of EPE system and was implemented it in Simulink as a neural model for the Day Ahead Market.

Received model of EPE system was tested for sensitivity to changes in, among others, the number of weights in the hidden layer, for the different number of hidden weights, for the different activation function in the individual layers (purelin, tansig), for different learning methods (trainlm, trains), wherein the results obtained were characterized by a worse parameters of the originally selected parameters at the design stage ANN or remained unchanged, ie:

- t change of the number of neurons in the hidden layer has not changed the learning process model neither the quality measured as an error MSE nor the rate of regression but increasing the number of neurons increased of course, computing power, and lengthen the learning process of ANN,
- increasing the number of layers does not improve the quality of the final model, measured as MSE and the rate of regression, increased of course demand for computing power.
- increasing the number of layers is sensible if the system characteristics does not allow for the implementation of linear transformation in the model, in this case, increasing the number of layers allows for the line separability into hyperplane in the n-dimensional space. This issue is closely described in [7, 23, 41].
- the network activation function changing as well as learning methods do not alter the quality of the final ANN model, measured as an error MSE and the rate of regression.

As a result of testing the sensitivity of NNT , we concluded that the regardless of the different number of neurons in the hidden layer, and the different number of hidden layers NNT learns the EPE system model in a similar level of accuracy. Adopted half-year period of operation EPE for DAM, contains 181 days (the input vector and the output vector with dimensions of 24 x 181) is sufficient set of value to learn the ANN model the EPE system and don’t demand, excessive computing power.

Due to the desire to improve the accuracy of parameters of the final model, in this case improving the weights accuracy are taken under consideration, some research actions have been taken to develop a hybrid system composed from ANN model supported by Evolutionary Algorithm methods (EA).

It sets the direction for further research in the field of improving EPE system as a neural-evolutionary hybrid model. The direction of further research may be, among the others, improving the neural-evolutionary model parameters by using quantum computing, which involves, among others, with the issues or receiving mixed quantum numbers problem [30].

References

1. Bierbrauer M., Menn C., Rachev S.T., Trück S.: Spot and derivative pricing in the EEX power market, Journal of Banking & Finance, Vol. 31.4, pp. 3462-3485, 2007.

2. Cartea A., Figueroa M.: Pricing in electricity markets: a mean reverting jump diffusion model with seasonality, Applied Mathematical Finance, Vol.12(4).5, pp. 313-335, 2005.

3. Ciechulski T., Osowski S.: Badanie jakości predykcji obciążeń elektroenergetycznych za pomocą sieci neuronowych SVM, RBF i MLP", Przegląd Elektrotechniczny, R. 90, nr 8, pp.148-151, 2014.

4. Conejo A.J., Contreras J., Espínola R., Plazas M.A.: Forecasting electricity prices for a day-ahead pool-based electric energy market, International Journal of Forecasting, Vol. 21(3), pp. 435-462, 2005.

5. Contreras J., Espínola R., Nogales F.J., Conejo A.J.: ARIMA models to predict next-day electricity prices, IEEE Transactions on Power Systems, Vol. 18(3), pp. 1014-1020, 2003.

6. Eichler M., Türk D.: Fitting semiparametric Markov regime-switching models to electricity spot prices, Energy Economics, Vol. 36, pp. 614-624, 2013.

7. Embedded MATLAB™ User's Guide by The MathWorks, Inc. Natick, MA 01760-2098 (USA), 182 pages, 2016.

8. Fijorek K., Mróz K., Niedziela K., Fijorek D.: Prognozowanie cen energii elektrycznej na Rynku Dnia Następnego metodami data mining. Rynek Energii, nr. 12, pp. 1-8, 2010.

9. Garcia R.C., Contreras J., van Akkeren M., Garcia J.B.C.: A GARCH forecasting model to predict day-ahead electricity prices, IEEE Transactions on Power Systems, Vol. 20(2), pp. 867 – 874, 2005.

10. Huisman, R., & Mahieu, R. J.: Regime jumps in electricity prices. Energy Economics, 25(5), pp. 425-434, 2003.

11. Halicka K.: Wykorzystanie sztucznych sieci neurono-wych do prognozowania cen na giełdzie energii, Rynek Energii, No. 1, pp. 20-23, 2010.

12. Janczura J., Weron R.: An empirical comparison of alternate regime-switching models for electricity spot prices, Energy Economics, Vol. 32(5) No.16 pp. 1059-1073, 2010.

13. Karakatsani N., Bunn D.: Forecasting electricity prices: The impact of fundamentals and time-varying coefficients, International Journal of Forecasting, Vol.24, pp. 764-785, 2008.

14. Knittel C.R., Roberts M.R.: An empirical examination of restructured electricity prices, Energy Economics, Vol.27, pp. 791–817, 2005.

15. Li J., Li J.: Next-Day Electricity Price Forecasting Based on Support Vector Machines and Data Mining Technology. Proceedings of the 27th Chinese Control Conference, Kunming, Yunnan, China, pp. 630-633, 2008.

16. Malko J.: Black-out, czyli zdarzenia katastrofalne w krytycznych systemach infrastrukturalnych, Wokół Energetyki 5(15), pp. 32-36, 2003.

17. Mielczarski W.: Rynki energii elektrycznej. Wybrane aspekty techniczne i ekonomiczne, ARE, Warszawa, 308 pages, 2000.

18. Mynarski S.: Analiza rynku. Systemy i mechanizmy. Wyd. AE. Kraków, 178 pages, 1993.

19. Mulawka J.J.: Systemy ekspertowe, WNT, Warszawa, 235 pages, 1996.

20. Osowski S.: Sieci neuronowe do przetwarzania informacji, OWPW, Warszawa, 422 pages, 2013.

21. Opalski K., Kacprzak K., Maciejczyk K., Pawłowski M.: Modelowanie polskiego rynku energii elektrycznej. Matematyka Stosowana, Volume 13/54, pp. 105-114, 2011.

22. Ruciński D., Tchórzewski J.: Neural modeling of the electric power stock market in usage of MATLAB and Simulink tools for the day ahead market data, Journal of Information System in Management, Vol. 5 (2) pp. 215-226, 2016.

23. Rudin W.: Podstawy analizy matematycznej. PWN, Warszawa, 292 pages, 2016.

24. Rutkowska D., Piliński M., Rutkowski L.: Sieci neuronowe, algorytmy genetyczne i systemy rozmyte, PWN, Warszawa, 410 pages, 1997.

25. Senjyu T., Takara H., Uezato K., and Funabashi T.: One-Hour-Ahead Load Forecasting Using Neural Network, IEEE Transaction on Power Systems, vol. 17, No. 1, pp. 113-118, 2002.

26. Simulink Getting Started Guide by The MathWorks, Inc. Natick, MA 01760-2098 (USA), 92 pages, 2016.

27. Stankovic A. M., Saric A. T., and Milosevic M.: Identification of Nonparametric Dynamic Power System Equivalents with Artificial Neural Networks, IEEE Transactions on Power Systems, vol. 18, No. 4, pp. 1478-1486, 2003.

28. Tadeusiewicz R.: Sieci neuronowe. AOW RM, Warszawa, 136 pages, 1993.

29. Tan Z., Zhang J., Wang J., Xu J.: Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models, Applied Energy, Vol. 87, pp. 3606-3610, 2010.

30. Tchórzewski J., Ruciński D.: Quantum Evolutionary Algorithm to Improve Parameters of Neural Models of Polish Electricity Power Exchange. Proceedings of Electrical Power Networks, IEEE Xplore Digital Library, Szklarska Poręba. Accepted for publication, 2016.

31. Tchórzewski J., Ruciński D.: Modelowanie neuronalne bezpiecznego rozwoju TGEE z wykorzystaniem notowań RDN. Rozdział w książce [pod red.] K. Raczkowskiego pt. Wymiary zarządzania ryzykiem w obrocie gospodarczym. CeDeWu. Warszawa, pp. 465-480, 2016.

32. Tchórzewski J.: Rozwój systemu elektroenergetycznego w ujęciu teorii sterowania i systemów. OW PWr. Wrocław, 190 pages, 2013.

33. Tchórzewski J.: Identification of the Electrical Energy Stock Exchange and creating knowledge maps using MATLAB environment with SIT and NNT Toolboxes, Energy Market, EEM 2009 6th International Conference on the European, IEEE Xplore Digital Library, pp. 1-6, 2009.

34. Tchórzewski J., Kłopotek M.: A case study in neural network evolution, Prace IPI PAN, nr 943, Warszawa, pp. 1-12, 2002.

35. Tchórzewski J.: Inżynieria rozwoju systemów, Monografia naukowa nr 18, Wydawnictwa WSR-P w Siedlcach. Siedlce, 279 pages, 1990.

36. Tchórzewski J.: Cybernetyka życia i rozwoju systemów, WSR-P, Siedlce, 408 pages, 1992.

37. Vardhan N. H., Chintham V.: Electricity price forecasting of deregulated market using Elman Neural Network, 2015 Annual IEEE India Conference (INDICON), New Delhi, pp. 1-5, 2015.

38. Weron R.: Modeling and Forecasting. Electricity loads and Prices: A statistical Approach, John Wiley & Sons Ltd, England, 154 pages, 2006.

39. Wierzchoń S., Kłopotek M.: Cluster analysis. IPI PAN. Warszawa, 310 pages, 2015.

40. Włodarczyk A., Zawada M.: Analiza cen spot energii elektrycznej. Przegląd wybranych modeli szeregów czasowych, „Energetyka”, nr 7, pp. 523-535, 2008.

41. Żurada J., Barski M., Jędruch W.: Sztuczne sieci neuronowe. Podstawy teorii i zastosowania, PWN, Warszawa, 374 pages, 1996.

42. Portal Spółki TGE S.A., www.tge.pl, http://wyniki.tge.pl/wyniki/rdn/fixing-I/.