

## ARTIFICIAL NEURAL NETWORKS AS MODELS NEURONAL ELECTRONIC STATE OFFICES

JERZY TCHÓRZEWSKI <sup>a)</sup>, KONRAD LESZKO <sup>b)</sup>

<sup>a)</sup> *Institute of Computer Sciences, Siedlce University of Natural Sciences and Humanities*

<sup>b)</sup> *Students Computer Science Branch GENBIT, Faculty of Science, UPH in Siedlce*

The paper presents selected results of research on learning design and artificial neural network (ANN) models paperless office as a state defined as a document repository. A review of selected issues on artificial neural network, and environments to support their generation and learning. In particular, attention was drawn to the new modeling capabilities leading to obtaining neural models of electronic systems. Artificial neural network is designed and taught her electronic office model based on the size of the input 11 and 9 variables, par 72 trainees on the actual size of government agencies for the year 2007. The model was obtained in MATLAB and Simulink and using the Neural Network Toolbox. Showing the possibilities of using the model to test sensitivities and simulation in Simulink.

Keywords: e-document, the office of the state as a repository of documents document, artificial neural networks, neural modelling, MATLAB and Simulink

### 1. Introduction

Currently, electronic documents workflow is one of the systems supporting management of government offices, institutions and organizations. For the reason of efficiency and security of documents workflow, models of e-systems are developed. Various methods of modelling, i.a. analytic modelling, identification (experimental modelling) and so called neural modelling (using artificial neural network), lead to a model of electronic documents workflow system.

Neural modelling of electronic documents flow requires development of a model of a system, a process or a phenomenon in the process of designing and training an artificial neural network (ANN). Designing an ANN requires selecting an appropriate programming environment. At present, there are numerous neural modelling-oriented environments, including MATLAB and Simulink by an American company MathWorks [1, 3, 12, 14], used in this work to present examples. Other interesting environments include SAP by StatSoft, or SPHINX by a Polish company AITECH [11]. Each of the above mentioned environments has an appropriate toolbox oriented towards neural modelling and equipped with an appropriate GUI, e.g. MATLAB and Simulink is equipped with the Neural Network Toolbox, and SPHINX is equipped with Neuronix [1, 9], etc.

The above mentioned environments differ in respect of the method of design and further use of the ANN as a model of the electronic documents workflow system. If it is not possible to create an appropriate model, select appropriate parameters of the ANN and learning rules as well as the required accuracy using the solutions available in the GUI, in both the above mentioned programming environments, reprogramming certain solutions, customizing the existing functions, or even adding one's own functions using available programming languages such as Matlab or Sphinx is possible [3, 9, 12, 14].

Apart from the architecture, it is also important to choose an appropriate training method for the ANN (supervised learning, unsupervised learning, learning with a critic, etc.), and especially a learning rule [4-8, 10-11, 13, 15-18]. In order to teach the ANN the document workflow, it is required to determine whether the model is to be related to documents workflow as one system, or individual subsystems modelled separately, in order to connect the developed neural models into one system, e.g. using Simulink.

## **2. A system of electronic documents workflow**

After learning the models, artificial neural networks are aggregated using weighted adders of input signals transformed into output signals using the activation function. Measurement of signals that are characteristic for the documents workflow depends on the implemented system of electronic documents workflow. There are many programs available on the market, implemented in various offices, institutions or organizations, such as M-Files, SiDAS, TETA HRM or OfficeObjects@Manager<sup>[1]</sup>.

These programs allow for full automation of documents workflow in the organization, i.e. collecting, storing and finding documents of any given structure. They enable efficient distribution and registration of electronic documents workflow. They may well be used in medium and large companies or government and

local government offices that work within the information society, and are responsible for the integrated electronic workflow of documents and information.

Documents workflow in the form of the implemented IT system in an office, company, organization, etc. significantly facilitates work of, i.a. a law office, a front office, a department (and people employed there), as well as other organizational units responsible for controlling documents workflow, so called Information Centres. Electronic repository (Lat. *repositorium*) is defined as a place of orderly storage of electronic documents to be made available.

Such repositories are gradually replacing so called main or central warehouses as well as archives of paper documents. They are designed so as the access to all their resources can be equally easy. Thus, repository mainly relates to the place of storage, but also to uploading, storing, providing access to, or even transferring electronic documents. Then, electronic repository becomes a system that tends to replace traditional warehouses such as main, central or temporary-storage warehouses, etc.

It is worth noting that a system of documents workflow is integrated with calendars and all kinds of subsystems for planning tasks in specified time. If the system has this option, new programming possibilities open up for the user.

### **3. Designing and teaching the ANN a model of a government office**

A substantial problem in modern government offices is constantly rising number of e-documents, which requires that they adapt to the rapidly changing environment. They strive at providing access to data regardless of the time and place of employees' location.

Data indispensable for designing a model of the e-documents system were prepared based on the numerical data related to the electronic government office [9, 17]. It was assumed, that a system of a government office type is a subsystem of a government office responsible for collection of state budget income. Eleven input variables:

- number of full time positions available [items],
- number of employees [items],
- number of IT specialists [items],
- government office area [m<sup>2</sup>],
- number of clients [items],
- number of computers [items],
- number of printers [items],
- number of servers [items],
- office income [PLN],
- office's own income [PLN],

- financial means to be returned as a result of overruled decisions [PLN] (Table 1) and 9 output variables:
  - volume of taxes [PLN],
  - number of inspections [items],
  - number of decisions issued [items],
  - number of overruling decisions [items],
  - number of issued certificates and explanations [items],
  - number of complaints about the work of the government office [items],
  - number of post-inspection decisions [items],
  - number of identification numbers issued [items]
- (Table 2) were assumed.

**Table 1.** Structure of input variables for training the ANN

Government office code	Number of full time positions available	Number of employees	Number of IT specialists	Government office area	Number of clients	Number of computers	Number of printers	Number of servers	State budget income	Company's own income	Financial means to be returned as a result of overruled decisions
No	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$u_7$	$u_8$	$u_9$	$u_{10}$	$u_{11}$
X	[items]	[items]	[items]	[m <sup>2</sup> ]	[items]	[items]	[items]	[items]	[PLN]	[PLN]	[PLN]
1	99	97	5	1 097.75	49 000	97	47	9	7844	0	26 296
2	27	27	4	560.90	46 000	35	17	2	2384	0	23
...	...	...	...	...	...	...	...	...	...	...	...
34	52	51	2	584.86	51 032	70	26	3	65 263	5 789	8

Source: [9]

MATLAB and Simulink environment is equipped with, i.a., GUI supporting the design of ANNs, i.e. Neural Network Toolbox (NNT), which significantly supports the designer in designing the architecture of the ANN and training the network. Additionally, it is an extensible, customizable environment. An open system in which to create programs using the Neural Network Toolbox facilitates analysis and customization of existing functions as well as adding one's own functions to meet the system requirements, whose model is to be developed. Modular representation of the ANN provides extreme flexibility allowing to teach ANNs models of technical, economic or electronic systems. The program is intended for efficient training of neural networks and easy visualization of the obtained neural models.

In the neural model of the system of e-documents, a "Feed-Forward Back-propagation" ANN was used. This method uses a delta rule and is used to train perceptron networks (multi-layer, one-directional) [6]. An ANN consists of three layers: input, hidden and output layer. The layers contain 11, 30 and 9 neurons, respectively. Neurons in the layers are equipped with the "tansig" activation function, and the function "trainlm" was used to train the ANN network, which function updates weights and biases based on the Levenberg-Marquardt Backpropagation algorithm. This algorithm is usually used for optimization of non-linear neurons.

**Table 2.** Structure of output variables for training the ANN

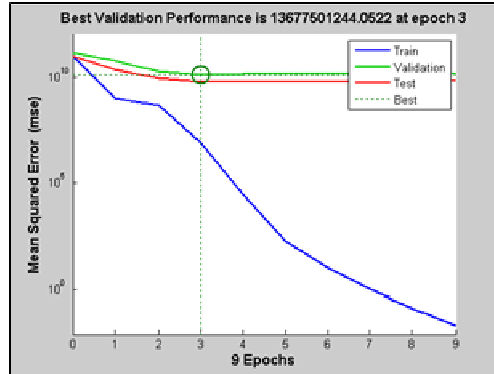
Government office code	Volume of taxes	Number of inspections	Number of decisions issued	Number of overruling decisions	Number of decisions brought to Higher Administrative Court	Number of issued certificates and explanations	Number of complaints about the work of the government office	Number of post-inspection decisions	Number of identification numbers issued
No	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>	y <sub>5</sub>	y <sub>6</sub>	y <sub>7</sub>	y <sub>8</sub>	y <sub>9</sub>
X	[PLN]	[items]	[items]	[items]	[items]	[items]	[items]	[items]	[items]
1	30 510	637	25	535	188	16 641	12	70	4
2	30 510	633	16	258	126	16 641	0	70	25
...	...	...	...	...	...	...	...	...	...
34	870 798	154	96	19	6	1 298	0	58	1 123

Source: [9]

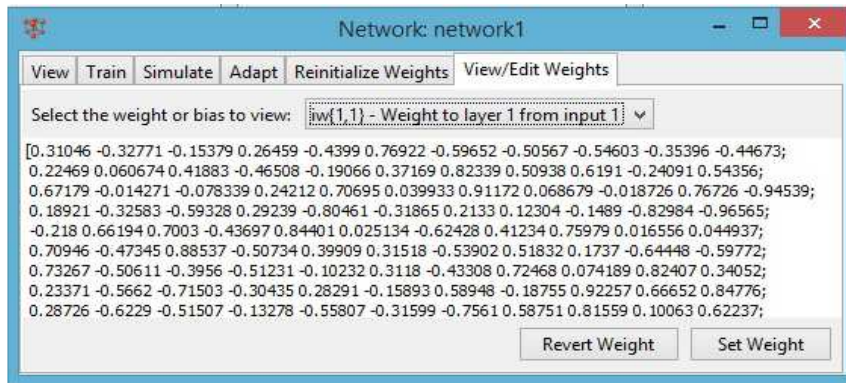
It combines two methods - the method of steepest descent and the Gauss-Newton method [6]. In the process of training, a defined objective function is MSE (Mean Squared Error).

Training the ANN was performed using a training file of numerical data as presented in table 1 and table 2. As a result of training the artificial neural network, which lasted 5 seconds, 9 epochs, etc. (Fig. 1), matrices in the form of weights were obtained as in Fig. 2.

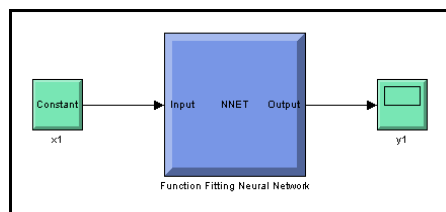
As the number of epochs increased, the above mentioned error decreased until the number of epochs reached 9, when it became stable and its value decreased much more slowly. After teaching the ANN a model of electronic documents system, a neural model of electronic office was generated that represents a system of electronic documents as in Fig. 3.



**Figure 1.** The graph illustrating the process of teaching the ANN the model of electronic documents system, Denotations: Mean Squared Error(mse), Train - graph of training the ANN, Validation- graph of train data validation, Test - graph of testing the ANN, Best- the best expected training parameters. *Source:* [9]

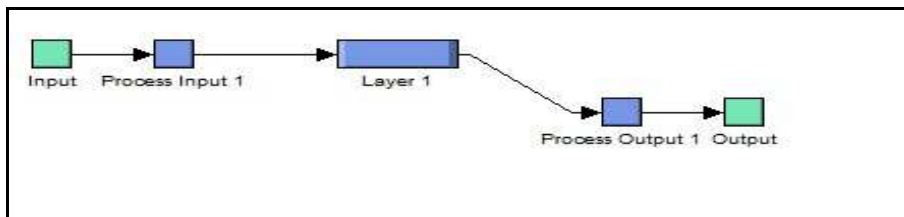


**Figure 2.** Matrix of weights of the ANN taught the neural model of the system of electronic documents using data obtained from such a system as a government office as an example. *Source:* [9]



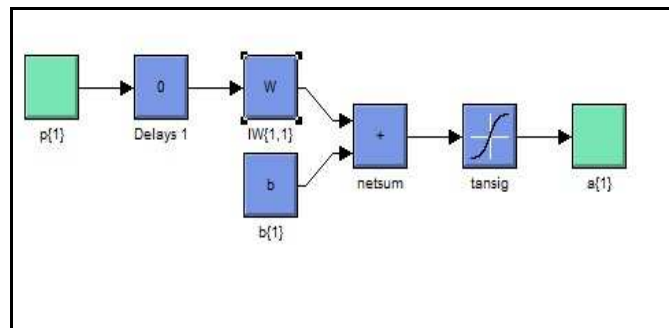
**Figure 3.** Neural model of electronic office as NNET. *Source:* [9]  
The following denotations were used:  $x_1$  – input variables,  $y_1$  – output variables, with both additional blocks (input and output) being selected so as to verify the correctness of the generated neural model

The generated model consists of three main blocks: " $x\{1\}$ " – input data, "Neural Network" – a block that contains the generated neural network, " $y\{1\}$ " – output data, i.e. the network response to the input data. In order to go deeper into the network structure one needs to click on the "Neural Network" block to open a window with an internal structure of the network, which allows to read the weights of the trained network. A neural model of an electronic government office is presented in Fig. 5.



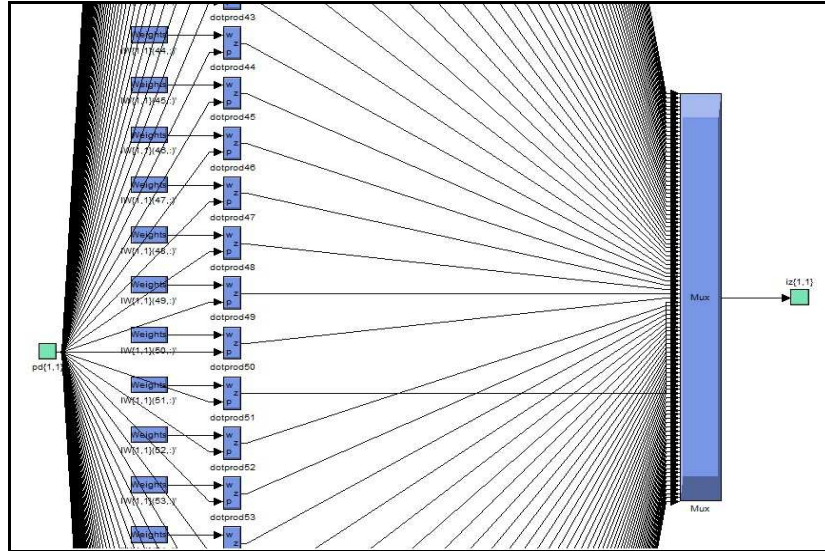
**Figure 4.** Internal structure of the ANN. *Source:* [9]

The ANN consists of the following blocks: "Process Input 1" – processes performed on input data, "Layer 1" – hidden layer, "Process Output 1" - processes performed on output data.



**Figure 5.** Structure of the hidden layer, Denotations:  $p\{1\}$  – input value of a neuron, Delays 1 – delay,  $W\{1,1\}$  – matrix of weights of the hidden layer,  $b\{1\}$  – bias value, netsum – adding block, tansig – activation function block,  $a\{1\}$  – output vale of a neuron. *Source:* [9].

The most important component of the ANN model are matrices of weights, where knowledge about the system, i.e. the government office, is gathered. The layer of input vectors (Layer 1) contains neurons, which are connected to the inputs of the ANN. The internal structure of each layer is presented in Fig. 6.



**Figure 6.** Matrix of weights. *Source:* [9].

Weights are placed the table with one row and  $n$ -columns. Weights are arranged in the following way: the first element in the row is for the first input, the second one - for the second output, etc.

#### 4. Conclusions

It is possible to teach an ANN a model of the system such as a government of-fice, equipped with a system of electronic documents workflow.

The ANN was taught a model of an electronic office in the MATLAB and Simulink environment using the NNT. Further, a model in Simulink may be constructed, and used to study and analyze the functioning of the ANN taught the model of electronic documents.

In the practical part of the work shows the resulting model for the 11 neurons electronic input and four output variables using one hidden layer. Artificial neural network model has learned well the office for 34 pairs of trainees. Learning error dropped to zero after nine epochs.

#### REFERENCES

- [1] Burke M.E. (2003): *The Phenomenal Power of Business Intelligence Managerial Skills for the 21st Century*, Europa Publications Limited, Taylor & Francis Group, London, UK.



- [2] Bral W.: *Obieg i ochrona dokumentów w zarządzaniu jakością środowiskiem i bezpieczeństwem informacji*. CDiI Delfin, Warszawa 2008
- [3] Brzózka J., Dorobczyński L.: *MATLAB. Środowisko obliczeń naukowo-technicznych*. MIKOM, Warszawa 2005
- [4] Hornik K., Stunchcombe M., White H. *Multilayer feedforward networks are universal approximators*. *Neural Networks*. Vol. 2, s. 359-366, 1989
- [5] Jankowski N.: *Ontogeniczne sieci neuronowe. O sieciach zmieniających swoją strukturę*. AOW EXIT, Warszawa 2003
- [6] Kacprzak T., Ślot K.: *Sieci neuronowe komórkowe*. PWN, Warszawa 1995
- [7] Kłopotek M., Tchórzewski J: *The concept of discoveries in evolving neural net*. *Advances in Soft Computing*. IPI PAN. Warszawa 2002
- [8] Kosiński R.A.: *Sztuczne sieci neuronowe, dynamika nieliniowa i chaos*. WNT, Warszawa 2008
- [9] Leszko K.: *Implementacja modelu i metamodelu obiegu elektronicznych dokumentów z wykorzystaniem różnych środowisk sztucznych sieci neuronowych*. Praca inżynierska pod kierunkiem dr inż. Jerzego Tchórzewskiego napisana na kierunku informatyka Wydziału Nauk Ścisłych UPH w Siedlcach, Siedlce 2014
- [10] Markowska-Kaczmar U., Kwaśnicka H.: *Sieci neuronowe w zastosowaniach*. OW PWr, Wrocław 2005
- [11] Michalik K., Wprowadzenie do budowy aplikacji w hybrydowym systemie ekspertowym PC-Shell, e-book, Katowice 2012.
- [12] Mrozek B, Mrozek Z.: *Matlab i Simulink. Poradnik użytkownika*. Wydanie III, Wydawnictwo Helion, 2010
- [13] Osowski S. : *Sieci neuronowe do przetwarzania informacji*. OW PW, Warszawa 2006
- [14] Osowski S., Cichocki A., Siwek K.: *MATLAB w zastosowaniu do obliczeń obwodowych i przetwarzania sygnałów*. OW PW, Warszawa 2006
- [15] Rutkowska D., Piliński M., Rutkowski L.: *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. WN PWN,
- [16] Tadeusiewicz R.: *Sieci neuronowe*. AOW RM, Warszawa 1993
- [17] Tchórzewski J.: *Neuronowy model dokumentu tekstowego*. Materiały V Konferencji Naukowej nt. "Sztuczna Inteligencja". AP, Siedlce 2002
- [18] Tchórzewski J.: *Artificial Neural Network for Processing Knowledge Concerning the State Of Economy and Administration Development*. *Information System in Management III*, pp. 113-120, Wyd. SGGW, Warszawa 2009
- [19] Żurada J., Barski M., Jędruch W.: *Sztuczne Sieci neuronowe. Podstawy teorii i zastosowania*. Wydawnictwo PWN, Warszawa 1996