UDC 681.325

A. Vasyura, Cand. Sc. (Eng.), Prof.; T. Martynyuk, Cand. Sc. (Eng.), Assist. Prof.; L. Kupershtein

SIMULATION OF THE NEURAL NETWORK TO SOLVE TASKS OF SYMBOLS IDENTIFICATION

Modeling of the three layer feed forward neural network is executed for symbols identification in context of using this network in the neural part of the fraud management system in telecommunications. Modeling is executed by using MatLAB 6.5 and toolkit Neural Networks Toolbox.

***Keywords**: fraud management system, neural network, symbols identification.*

Introduction

Neural networks (NN) are used for the solution of various tasks in many areas of human activity. Among them are: economy and business, medicine, aviation, telecommunications, safety and protective systems, computer-aided manufacturing etc. [1 - 3].

The decision of the most of the above-mentioned tasks is based on classification and recognition of patterns, the example of which can be a task of letters recognition, language recognition, electrocardiogram signal classification, determination of solvency of bank customer, medical tasks, task of management the brief-case of securities, task of determination of viable and predisposed to bankruptcy firms etc. [4, 5].

Another practical and perspective direction of the NN using, as an universal classifier, is detection of different sort of fraud in the telecommunication systems. It is explained by the fact, that the operators all over the world lose up to 100 billion dollars exactly on this reason, and this sum increases every year [6]. Main activities directed on overcoming such losses are aimed at creation of the fraud management systems (FMS), which are component of the control and monitoring system of telecommunication network. FMS are based on the use of intellectual technologies: neuron networks and fuzzy logic [7].

Statement of problem

Identification of the dynamic system (process) is a receipt or clarification the mathematical model of this system (process) from experimental data, which (system) is described by mathematics [8]. If an object is consider in all of his operating modes, the receipt or clarification the object model from experimental data, effective for all of the operating modes, is called the identification of an object.

The task of identification is generally solved in two stages. On the first stage, which is named the structural identification, the rough model of the object, which approximates the intercommunication input-output and contains the influenced parameters, is formed. On the second stage, which is named parametrical identification, the values of parameters minimizing distance between the model and experimental outputs of object, are selected [9, 10].

In addition the tasks of identification can be considered in the same way as the tasks of patterns recognition (classifications) [11], which consist in taking a pattern to one of the few sets which do not intersect pair wise. NN is the most effective method of classification because there is the universal approximation, and also they can generate plenty of regressive models, which are used for the solution of the classification tasks by statistical methods [4, 12].

In turn, the main task to be solved by the FMS is the identification (recognition) of fraudster actions by the non-standard actions of a talker's profile, which can be described by such characteristics, as frequency and direction of calls, duration of talk, identification information of talker etc. Thus, the task of FMS is pattern recognitions by the input signals set which are taken off directly from a switching unit by the scanner of service information and information about realizable bells [7, 13]. All of this information acts in a digital kind for its next identification in FMS by the neural-fuzzy network, where preprocessing of alarm information [14] is performed in neuron part, and in fuzzy part the interpretation and logical conclusion of the got results is performed [6, 7]. In addition, the starting of such FMS designing is a construction of its simulation model.

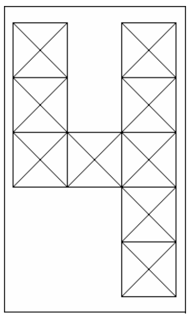
So, **the objective** of this work is a construction and researches of NN model of neural part of the neural-fuzzy fraud management system in telecommunication networks on the example of symbol identification.

Features of symbol identification

Among the large class of software for the design of NN the tool Neural Networks Toolbox of MATLAB 6.5 application package MathWorks Inc is used. It is enough flexible, provided with the wide set of commands and functions for design and research both static and dynamic NN[15]. In addition, this package is provided with a GUI-interface, which allows to execute the research of NN even to the inexperienced user.

We will consider the variant of NN model implementation by MATLAB for recognition of the 10 Arabic numerals.

As a sensor it is supposed to use the recognition system which digitizes every symbol. For example, every symbol will be presented as the template of dimension 3x5. Then symbol “ 4” can be presented, as shown on the fig. 1,a and 1,b.



a)

1	0	1
1	0	1
1	1	1
0	0	1
0	0	1

b)

Fig. 1. Presentation of the symbol “4”

However the symbols reading system usually works unideally. That is the separate elements of symbols can appear noisy (fig. 2).

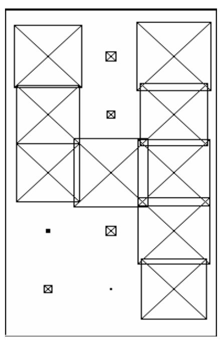


Fig. 2. Presentation of the noisy symbol “4”

The designed NN must exactly recognize the ideal input vectors and reproduce noisy vectors with maximal exactness. M-function *prprob_n* determines 10 input vectors, each of them contains 15 elements. This array is named *numbers*. M-function forms the primary variables *numpad* and *targets*, that determine the arrays of numbers and purpose vectors.

So, the 15 elements input vector goes to the input of the network; the output vector contains 10 elements, only one of which is equal to 1, and others to - 0. A correctly functioning network must return a vector with a value 1 for an element, which equals to the number of the numbers from 0 to

9. In addition, a network must be able to recognize symbols in noisy conditions. It is assumed that noise is a random variable with a mean value 0 and standard deviation, less or equals to 0.2 [15].

the operation of NN requires 15 inputs and 10 neurons in an output layer. The solution of the task involves three-layered feedforward NN with a threshold, logarithmic sigmoid and linear activating functions in input, hidden and output layers respectively. Such activating functions are selected because an input and output layers are mainly realizing the information preparation for the further use, as well as the transformation of the got results. So, the threshold and the linear functions are wholly usable for this purpose [16]. Regarding the hidden layer, it executes all of basic calculations, and his activating function must be more “natural”, for example, logistic sigmoid or hyperbolical tangent.

Simulation model of neuron network

The model of NN structure for recognition of the Arabic numerals is presented on a fig. 3, where IW{1,1}, LW{2,1}, LW{3,2} are weight matrices of the first, hidden and output layers respectively; b{1}, b{2}, b{3} are bias vectors of the first, hidden and output layers respectively; p1 is an input informative vector array; y is an output vector array; a1, a2, a3 are values arrays of activating functions of the first, hidden and output layer respectively.

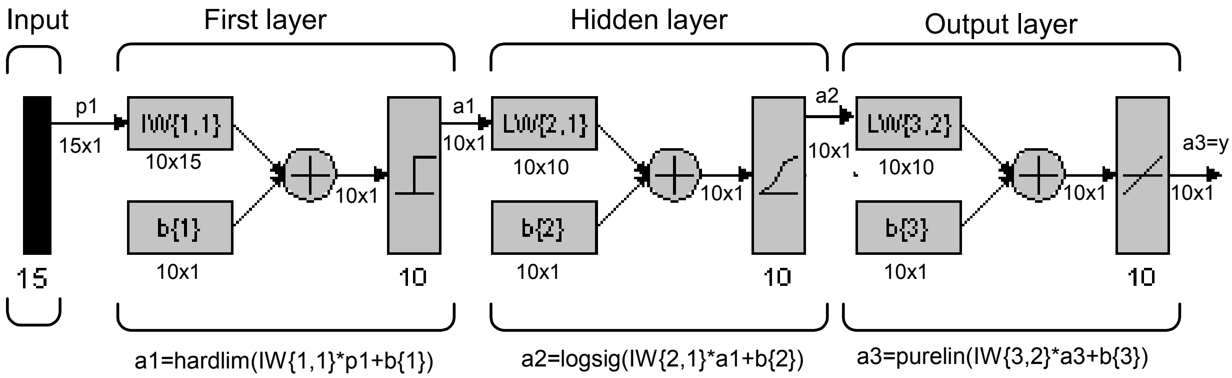


Fig. 3. Flow diagram of neuron network

The hidden layer consist of 10 neurons, nevertheless, if there are complications during the teaching of network, it is possible to increase the number of neurons of this level. The network should be learned in the way of forming of the “one” in the single element of output vector, the position of which corresponds to the symbol number, and filling the other part of the vector by "zeros". However the presence of noises can make it so that NN will be unable to form the vectors of the output, which consist exactly of “ones” and “zeros”. Therefore on completion of the training stage an output signal is processed by m-function *compet*, which appropriates a value 1 to the single element of the output vector, and to the other – the value 0.

Initializing of network. We run m-file *prprob_n*, that forms the vectors array of inputs *numpad* size 15x10 with the templates of symbols of numbers from 0 to 9 and array of having a special purpose vectors *targets*:

```
[numpad,targets]=prprob_n;
[R,Q]=size(numpad);
[S3,Q]=size(targets);
```

The three-layered feed forward neuron network is create by the command of *newff*:

```
S1=10;
S2=10;
net=newff(minmax(numpad),[S1 S2 S3],{'hardlim'... 'logsig' 'purelin'},'traingdx');
net.LW{3,l}=net.LW{3,l}*0.01;%weights initialization
net.b{3}=net.b{3}*0.01;%biases initialization
```

Training. For the NN to process the noisy input vectors, let's execute its training both on ideal and on noisy vectors. At first a network trains on ideal vectors, while there will not be provided minimum root-sum-square uncertainty. Then the network trains on 10 sets of ideal and noisy vectors. Training executed by the function *traingdx* that will realize the back propagation method with indignation and adaptation of tuning speed parameter.

Training without a noise. A network at first trains without a noise with the maximal amount of training cycles 5000 or to achievement of permissible root-sum-square uncertainty that is equal to 0.11 (fig. 4):

```
R=numpad;
T=targets;
net.performFcn='mse;
net.trainParam.goal=0.11;
net.trainParam.show=20;
net.trainParam.epochs=5000;
net.trainParam.me=0.95;
[net,tr]=train(net,P,T)
```

As is obvious from fig. 4, a network learned already after implementation of 84 epoches (cycles).

Training with the noise presence. To project NN, immune to influencing of noise, we will teach it with the use of two ideal and two noisy copies of vectors of numbers set from 0 to 9. Target vectors consists of four copies of vectors.

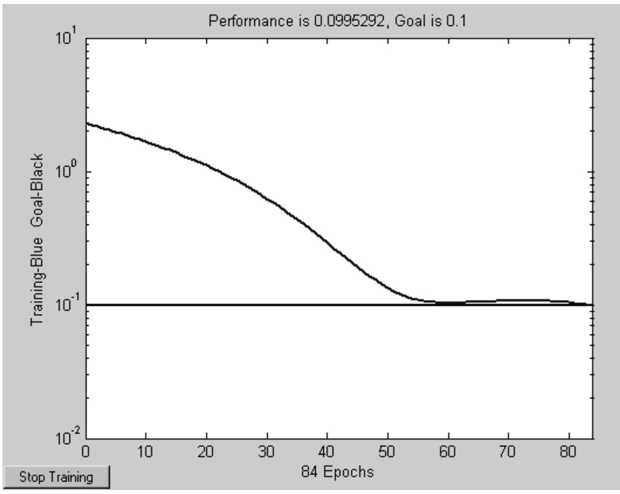


Fig. 4. Network training without noise

The noisy vectors have noise with a average value 0.1 and 0.2. It teaches NN correctly recognize the noisy symbols and ideal vectors. At training with noise will shorten the maximal amount of teaching cycles to 300, and will increase a permissible error by 0.6 (fig. 5):

```
netn=net;
netn.trainParam.goal=0.6;
netn.trainParam.epochs=300;
T = [targets targets targets targets];
for pass=1:10
P=[numpad,numpad,(numpad+randn(R,Q) *0.1),
(numpad+randn(R,Q) *0.2)];
[netn,tr]=train(netn,P,T);
end
```

As NN trained with the presence of noise, it makes sense to repeat it’s training without noise, to guarantee that the ideal input vectors are classified correctly:

```
netn.trainParam.goal=0.1;%Permissible error
netn.trainParam.epochs=500;%Maximal numbers of training cycles
```

net.trainParam.show=5;%Frequency output of results to a screen
[netn,tr]=train(netn,P,T)

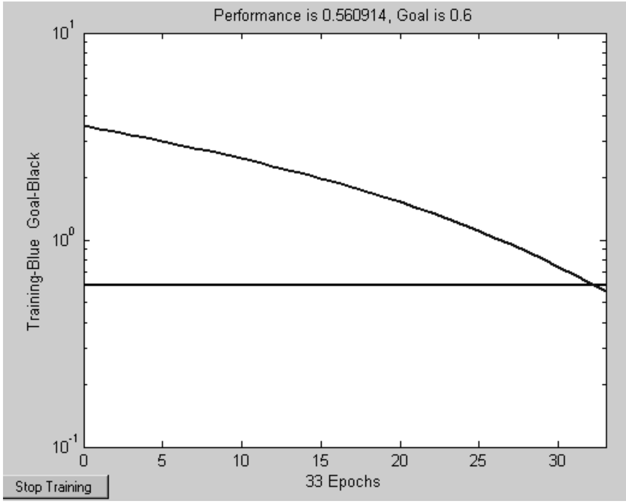


Fig. 5. Teaching of network at presence of noise

If we need a higher accuracy of recognition, the network can be trained during a longer period of time, or with the use of greater number of neurons in the hidden layer. It is also possible to increase the size of vectors in order to take advantage of template with more fine-meshed net, for example 5x7 or 10x14 points in place of 3x5.

We will check up work of NN for recognition of symbols. For the graphic output of symbol on a display m-function *plotchar_n* is used. We will form the noisy input vector for a number “2” (fig. 6):

noisy_num=numpad(:,3)+randn(15,1)*0.2;
plotchar_n(noisy_num);% noisy number “2” (fig. 6)
A2=sim(net,noisy_num);
A2=compet(A2);
answer=find(compet(A2)==1)
answer = 3
plotchar_n(numpad(:,answer));%recognized number ”2”

NN recognized symbol “2” without errors (fig. 7).

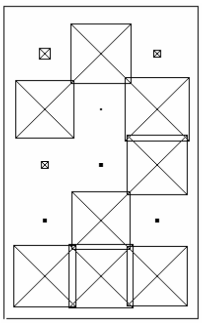


Fig. 6. Noisy symbol

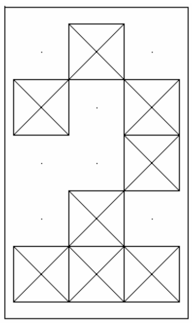


Fig. 7. Recognized symbol

Conclusions

So, the presented model of NN showed us how a system of patterns recognition can operate. The network was trained once or twice with the different input vectors. Training of NN on the different sets of the noisy vectors allowed us to teach it to work with noisy information, which is common for the real practice. In addition, the threshold neurons are used in the input layer of NN that proves possibility and expedience of hardware representation of neural structures with the use of simple perceptron neurons for designing intellectual FMS in telecommunication networks.

REFERENCES

1. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. Пер. с англ. – М.: Мир, 1992. – 240 с.
2. Комашинский В.И., Смирнов Д.А. Нейронные сети в системах управления и связи. – М.: Горячая линия – Телеком, 2003. – 94 с.
3. Галушкин А.И., Иванов В.В. Некоторые концептуальные вопросы развития нейрокомпютеров // Зарубежная радиоэлектроника. – 1997. – № 2. – С. 3 – 10.
4. Стариков А. Практическое применение нейронных сетей для задач классификации (кластеризации) // BaseGroup. – 2000 (http://www.basegroup.ru /neural/prectice.htm).
5. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применения в экономике и бизнесе. – М.: МИФИ, 1998. – 224 с.
6. Лезин В., Перминов К. Мошенничество в сетях операторов фиксированной связи и как с ним бороться // Информкурьерсвязь. – 2004. – № 3 (http://www.miks.ru/magazine/ magazine_look.php?id=192).
7. Русеев Д. Мошенничество в мобильных сетях и средства борьбы с ним // Мобильные телекоммуникации. – 2003. – № 3 (http://www.mobilecomm.ru /view.php?id=423).
8. Красовский А.А. Универсальные алгоритмы оптимального управления непрерывными процессами. –М.: Наука, 1977. – 272 с.
9. Цыпкин Я.З. Основы информационной теории идентификации. – М.: Наука, 1984. – 225 с.
10. Ротштейн А.П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. – Винница: «УНІВЕРСУМ-Вінниця», 1999. – 320 с.
11. Цыпкин Я.З. Адаптация и обучение в автоматических системах. – М.: Наука, 1968. – 400 с.
12. Неймарк Ю.И. Многомерная геометрия и распознавание образов // Математика. – 1996. – №7. – С. 119 – 123.
13. Куперштейн Л.М. До проблеми виявлення шахрайства у телекомунікаційних мережах на базі інтелектуальних технологій // Матеріали ІІІ МНТК "Сучасні проблеми радіоелектроніки, телекомунікацій та приладобудування". – Вінниця. – 2007. – С. 25 – 26.
14. Борисов Е.С. Использование языка MC# для реализаций параллельного классификатора образов // Кибернетика и системный анализ. – 2005. - № 3. – С. 179 – 182.
15. Медведев В.С., Потемкин В.Г. Нейронные сети. MATLAB 6 / Под. общ. редакцией В.Г. Потемкина. – ДИАЛОГ-МИФИ, 2002 . – 496 с.
16. Пономарев С. Нейронные сети // INFUSED BYTES OnLine (http://www.enlight.ru/ib/tech/ neural/index.html).

Anatolyy Vasyura – Director of the Institute, Professor of the Department for Automatics and Information Measuring Devices;

Tetyana Martynyuk – Assistant Professor of the Department for Laser and Optoelectronic Engineering;
The Institute for Automatics, Electronics and Computer Control Systems, Vinnytsia National Technical University.

Leonid Kupershtein – Senior Lecturer of the Department for Information Systems and Technologies.
Vinnytsia Financial Economic University