UDC 004.386

V. O. Yakovenko, Doctor of Technical Sciences, Professor of the Department of Information Systems and Technologies, University of Customs and Finance
Yu. V. Ulianovska, Ph.D. of Technical Sciences, Associated Professor of the Department of Information Systems and Technologies, University of Customs and Finance
O. O. Kaliaka, Student of the University of Customs and Finance

SOFTWARE IMPLEMENTATION OF NEURAL NETWORK TECHNOLOGIES FOR AUTOMATED TEXTUAL INFORMATION CLASSIFICATION

The results of the analysis of the state of highways suitable for the organization of high-speed traffic on the route Kyiv-Dnipro are presented, and the main directions of its implementation are identified. The calculations of permissible technical parameters of the movement of vehicles on certain routes. It is shown that the proportion of heavy and large-sized vehicles grows in the transport flows, which leads to the rapid destruction of roads and bridges, which are designed for much smaller volumes and loads. For the introduction of high-speed traffic proposed measures to improve the technical level of existing roads. The necessity of intensification of the modernization and repair of roads in Ukraine based on the use of modern technologies, which will create the necessary conditions for the introduction of high-speed road transport.

Key words: *neural network; perceptron; text classification.*

Розглянуто завдання створення нейронної мережі для розпізнання та класифікації текстових документів. Проведено дослідження наявних моделей нейронних мереж і методів векторного подання тексту. Виявлено переваги та недоліки досліджуваних алгоритмів. Результатом дослідження є розроблений оригінальний алгоритм класифікації текстів із використанням технологій нейронних мереж на основі перцептрона, який можна навчати методом зворотного поширення помилки. На основі запропонованого авторами алгоритму розроблено програмний додаток та проведено апробацію розробленого алгоритму.

Ключові слова: *нейронна мережа; перцептрон; класифікація текстів.*

Рассмотрено решение задачи создания нейронной сети для распознавания и классификации текстовых документов. Было проведено исследование существующих моделей нейронных сетей и методов векторного представления текста. Определены преимущества и недостатки исследуемых алгоритмов. Результатом работы является алгоритм классификации текстов с использованием технологий нейронных сетей на основе перцептрона, который можно обучать методом обратного распространения ошибки. На основе предложенного авторами алгоритма было разработано программное приложение и проведено апробацию предложенного алгоритма.

Ключевые слова: *нейронная сеть; перцептрон; классификация текстов.*

Problem formulation. The use and improvement of computer technologies opens up a range of new features to humankind, but along with this there are a number of other problems associated with the growth of processed arrays of text documents. During work with full-text databases a problem of documents searching and classification according to their content appears. Classification of documents can be regarded as one of the tasks of informational search, which consists of enrolling document in one of several categories based on its content.

There are following tasks of the classification: creation of thematic directories, spam filtering, electronic document management systems, tagging, automatic text translation [1]. Text document classification is a well known theme in the field of the information retrieval and text mining. Selection of most desired features in the text document plays a vital role in classification problem [2]. That is why the task of developing an algorithm of textual information classification in order to improve informational search in view of modern requirements for classification and characteristics of text data arrays is relevant.

Analysis of recent researches and publications. In order to solve this problem [3] there was considered the most popular classification methods such as: classification using the decision tree, the Bayesian classification method, support-vector machine method, k-nearest neighbors algorithm and the artificial neural network (NN). The support-vector machine method is a reference algorithm in terms of the quality of the classification. In traditional text classification, a document is represented as a bag of words where the words in other words terms are cut from their finer context i.e. their location in a sentence or in a document. Only the broader context of document is used with some type of term frequency information in the vector space. Consequently, semantics of words that can be inferred from the finer context of its location in a sentence and its relations with neighboring words are usually ignored [4]. However, its main disadvantage is speed. Training requires a significant amount of memory and expenses of machine time.

The advantages of classifying using the decision tree are that the constructed tree is easy to analyze and the result of the algorithm can be interpreted visually. The disadvantages of the algorithm include the fact that constructing decision trees gives the same weight to "positive" and "negative" branching in the nodes. A large number of "negative" branches in the description of the rubric can lead to hard-to-interpret rules and re-learning of the classification algorithm. The main feature in thek-nearest neighbors method is the absence of a learning process. This method focuses on more similar to the analyzed text from the collection during classification. The main disadvantage of the algorithm is the low speed of classification.

The Bayes Classification method provides ease of implementation and low computing costs for training and classification [3; 5; 6]. However, this method has a significant disadvantage such as low quality of classification in the most real tasks [3]. In addition, performance depends on the accuracy of the estimated conditional probable conditions. Sometimes it's difficult to estimate accurately, especially when there is a lack of training data. That is why in paper [5] it is proposed to transform the probability estimation problem into an optimization problem and use three methodological approaches to solve it.

Over the past few years, neural networks have re-emerged as powerful machine-learning models, yielding state-of-the-art results in fields such as image recognition and speech processing. More recently, neural network models started to be applied also to textual natural language signals, again with very promising results [7]. An artificial neural network is an adaptive system that consists of a group of connected artificial neurons. Neural networks have high accuracy in processing both linear and nonlinear samples, but classification decision-making is difficult to formalize due to the nature of the neural networks organization and represent a non-trivial task, taking into account scalability with limited computing resources [2; 3].

As a result of analysis of classification methods, the neural network was selected. The neural network compared to other methods of classification has the greatest accuracy, but requires high computing power.

Purpose of the article is to develop automated analytical system for analysis of prepared arrays of text information documents using neural networks and their classification by categories.

Main material. An artificial (mathematical) neuron performs the transformation of the input signals vector, as follows [8]:

$$y = I(S); S = \sum w_i x_i, \quad (1)$$

where w_i – is the weight vector of the neuron (the weight of the synaptic bonds);

S – the result of the weighted summation;

I – neuron activation function.

Functionality of a neuron is simple, therefore, neurons are united in the network for solving specific problems. Training a neural network is reduced to the selection of the weight coefficients of each neuron provided that the network topology is chosen and activation function is selected.

There are the main types of classifying neural networks such as: singlelayer and multilayer perceptron, Gaussian neural classifier, Kohonen network, integrated distribution network, cascade network [9]. Perceptron is a neural network of direct distribution, the principle is transmission of the signal through N additional hidden layers to the output. Each node in the network is connected by M-number of scales from the previous layer. This creates the associations between the input and the output layers. In the classification tasks perceptron shows good accuracy but it depends on the number of layers in the network. Perceptron is a time-tested very flexible tool for solving many tasks.

Kohonen neural network differs from the perceptron in that there used unsupervised learning as learning method. Learning data is consist of only input variables and doesn't have relevant output values. It is using access threshold to solve classification problems with Kohonen networks, it plays the role of the maximum distance where the recognition takes place. If the activation level has "won" neuron that exceeds this threshold, then considered that the system did not make decisions [10].

Convolutional neural network was initially not purposed to work with text information and was used to "computer vision". This network, like perceptron, is network of direct distribution. The main feature of this network is the convolutional layers which may be many. The convolution operation is the multiplication of the input signal into a small matrix of weights and summation the results. The result is a map of signs. This method is very effective in classification tasks [11].

After selecting the type of neural network, the objective was to select an activation function. To the role of the activation function of the neural network were considered sigmoidal function and hyperbolic tangent [8; 9].

Sigmoid function is an S-shaped, continuously differentiated, monotone increasing function that can take values from 0 to 1. Sigmoid function (2) has the ability to amplify weak signals better than strong and prevents saturation from large signals.

$$f(x) = \frac{1}{1+e^{-x}}. \qquad (2)$$

Another widely used activation function is the hyperbolic tangent. By form it is similar to the sigmoid function and is often used by biologists as a mathematical model of the nerve cell activation [12]. It can take values from –1 to 1.

As an activation function of the artificial neural network, it has the form given in the expression (3):

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}. \quad (3)$$

Solution. Finally the perceptron was chosen as a result of analysis of neural networks. All considered networks have good classification accuracy, but the perceptron is simpler and more flexible in setting up.

A sigmoid function was selected as an activation function in the neural network. Compared to a hyperbolic tangent, it can take less values, but for the task of classification this will be enough. Also, the derivative of the sigmoid function is easier to find and easier to program.

As a result of modeling was created the network, shown at fig. 1:

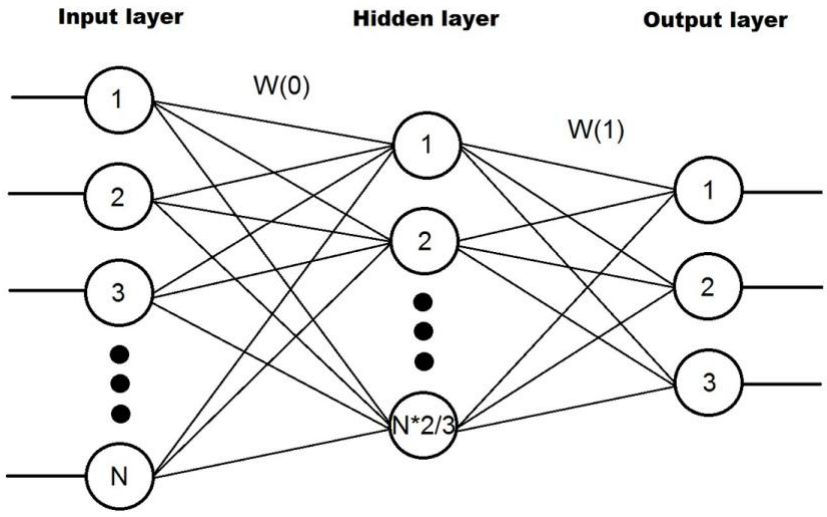

Fig. 1. Model of created neural network

In the developed model of neural network there are 3 layers of neurons: input, hidden and output.

The first two layers are dynamic, that means that they change the number of neurons depending on the size of input vector. The hidden layer is setting up depending on the input vector in the proportion of 2: 3. This gives to the neural network some flexibility and adaptability.

The output layer has a fixed number of neurons of 3, but may be changed depending on the needs of the user. Neuronal network with 3 output neurons can

takes up to 8 different results, but to facilitate learning in the developed model, each output neuron corresponds to separate classification category.

There are following steps to thoroughly make out the algorithm of recognizing the text as:

1. Create a dictionary.
2. Neural network training.
3. Documents recognition and information output.

Let's check described steps more detailed. Creating a dictionary is necessary for network training and text recognition and is one of the most important parts of both algorithm and subsequent program realization. Excel was chosen as the basis for the dictionary, there are several reasons for this:

– easy to edit;

– ability to use without installing special software;

– Ease of recovery.

The algorithm of creating a dictionary is down below:

– first you need to select and open the document that was chosen as the basis for the dictionary in the program;

– after loading the document is split into separate words, punctuation marks and duplicates are deleted and the obtained array of words are copied into an array of type **`List <string>`.**

– **`<string>`** array is copied to MS Excel document which is creating using the Microsoft.Office.Interop.Excel namespace.

If it is need, the created document can be edited and supplemented with new words.

To create the input signal vector it is necessary to convert it into a signal. N-gram was used as a conversion method, on the basis of which the dictionary was created.Statistical n-gram language models (LM) have been used successfully for speech recognition and many other applications/ they compute the probability of the nth word by conditioning on the previous n-1 history (h) words. They suffer from the shortage of long-range information, which limits performance. To capture the long-range information. One of the earliest attempts was a cache-based LM that took advantage that a word observed earlier in a document could occur again [13]. The dictionary is manually created previously, or it can automatically be create by program (you need to upload document with data). For productivity, it's necessary to use similar subjects documents with analyzed texts. The uploaded document is split into separate words, the repetitions and symbols removes. The next step is to copy data into a document that can be edited manually.

During the program execution, the documents that will be classified will be compared with this dictionary, and as a result, document vector will be created (fig. 2).

```
class Vector
{
    ссылок: 2
    public int[] VectorBuilder(string docName, List<string> dictionary)
    {
        int[] vector = new int[dictionary.Count];
        string fi = null;
        Word.Application app = new Word.Application();
        Object fileName = docName;
        var doc = app.Documents.Open(fileName);
        string rangeText = doc.Range().Text;
        fi = rangeText;
        app.ActiveDocument.Close();
        app.Quit();

        for (int j = 1; j < dictionary.Count; j++)
        {
            if (fi.Contains(dictionary[j]))
            {
                vector[j] = 1;
            }
            else
            {
                vector[j] = 0;
            }
        }
        return vector;
    }
}
```

Fig. 2. C# code with the method of creating a text vector

The project uses the backpropagation method. This method is the most popular for multilayer perceptron learning. In general, the backpropagation algorithm is sequence of following steps [14]:

1. Initiate weights with small random variables.
2. If the stop condition is not satisfied, follow steps 3–10.
3. For each training pair, complete steps 4–9.

Direct pass:

4. Each input neuron $x_i = 1 \dots n$ receives an input signal and distributes it to all hidden layer neurons.
5. Each hidden layer neuron $v_i = 1 \dots q$ summarizes its weighted input signals: $h_j = \sum_j^n w_{ij} x_i$, applies the activation function to the received sum, forming an output signal: $v_j = f(h_j)$, which is sent to all the neurons of the output layer.

6. Every output neuron $y_k, k = 1 \dots m$ sums up the weighted signals: $h_k = \sum_j^q w_{jk} v_j$, forming after the activation function the output signal of the network: $y = f'(h_k)$.

Backpropagation:

7. Each output neuron equates its value with the objective function and calculates it $\delta_k = (t_k - y_k) f'(h_k)$, after what determines the corrective member of the weights:$\Delta w_{jk} = \mu \delta_k v_j$,and the parameters δ_k sent to the hidden layer neurons.

8. Each hidden neuron$v_j$sums itsδ −inputsfrom the output layer neurons: $h_k = \sum_k^m \delta_k w_{jk}$, the result is multiplied by the derivative of the activation function for expression δ_j : $\delta_j = f(h_j) \sum_k^m \delta_k w_{jk}$ and calculate the correction member: $\Delta w_{jk} = \mu \delta_k w_{jk}$.

Weights adjustment:

9. Weights between hidden and output layers are modified as: $w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$.Weights between input and hidden layers are adjusted in a similar way: $w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$.

10. Checking the stop condition to minimize the error between the ideal and the actual output of the network.

The algorithm of document classification with the proposed principles of classification consists of the following steps:

Step 1. To the program input comes one or more MS Word documents;

Step 2. The document is opens in the background and copies to the array;

Step 3. The array is handled by the Vector class;

Step 4. The text vector comes to the input of the neural network;

Step 5. We get the text classification in the output.

The documents are copied to the appropriate folders and the classification type is added to the title.

To test the algorithm, there was created a software application that is able to execute a number of experiments with designed neural network. The program setting are flexible and you can set the number of neurons at the NN levels, the number of training periods, the number of training data and the number of experiments.

During the software product development were set tasks the solution of which allowed to increase the comfort of user experience of the program, to automate some of the program processes and improve performance.

One of these tasks is the choice of dictionary data store. The essence of the task is to make the dictionary easy to transfer, edit by user without using of special software and create a new one if it necessary. Several modern DBMSs have been tested, including Oracle and MS SQL Server 2017. However, to complete

the task it was necessary to refuse from DB and choose MS Excel. This program has installed on most modern computers and most users to their needs without special skills can edit files.

The automated system allows:

– create and edit a dictionary;

– train the neural network for their needs;

– classify documents in batch;

– search the documents key words and quickly open them.

The class diagram of the projected system with a graphical representation of the structural interrelations of the logical model of the system is shown in fig. 3.

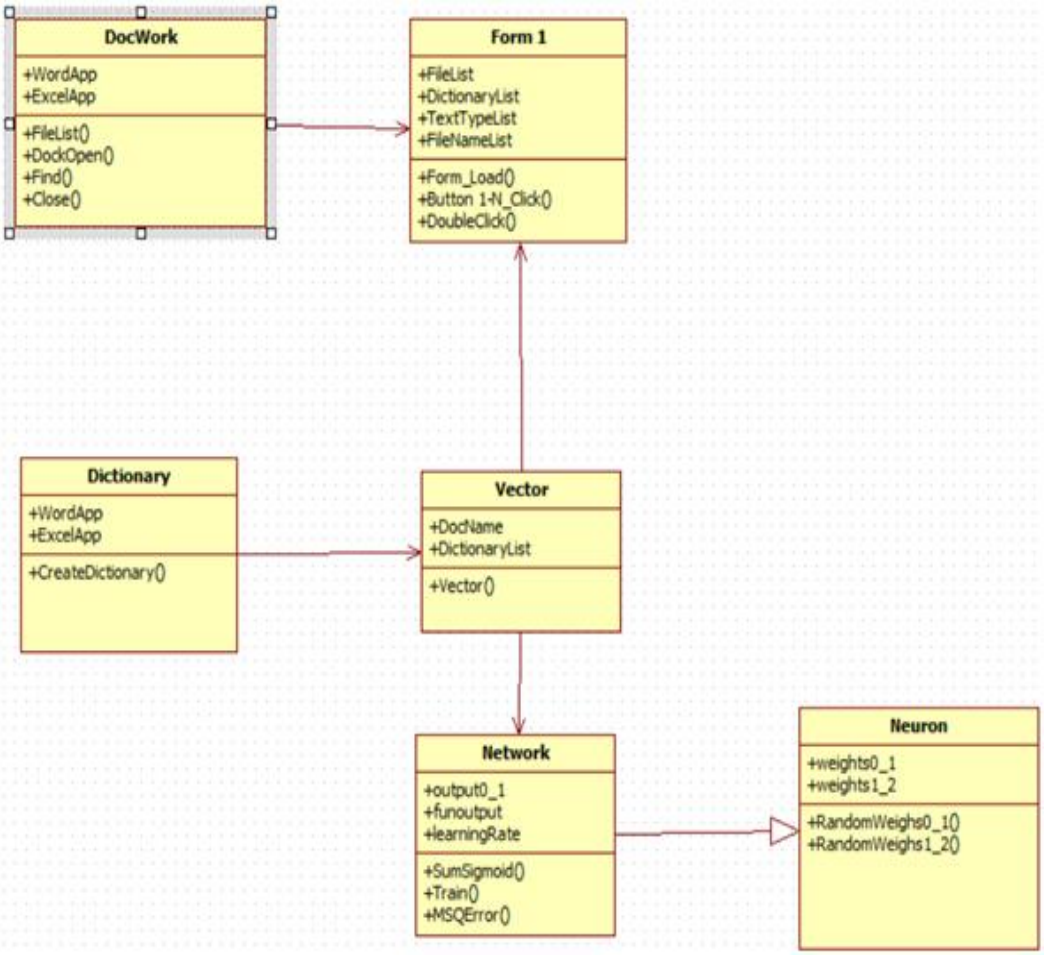

Fig. 3. Class diagram

To train the program you don’t need text documents. Vectors are automatically generated and randomly assigned to a type. You can specify their number in the Trainset field. The Resultsets field determines quantity of vectors that will be generated to test the trained network (fig. 4).

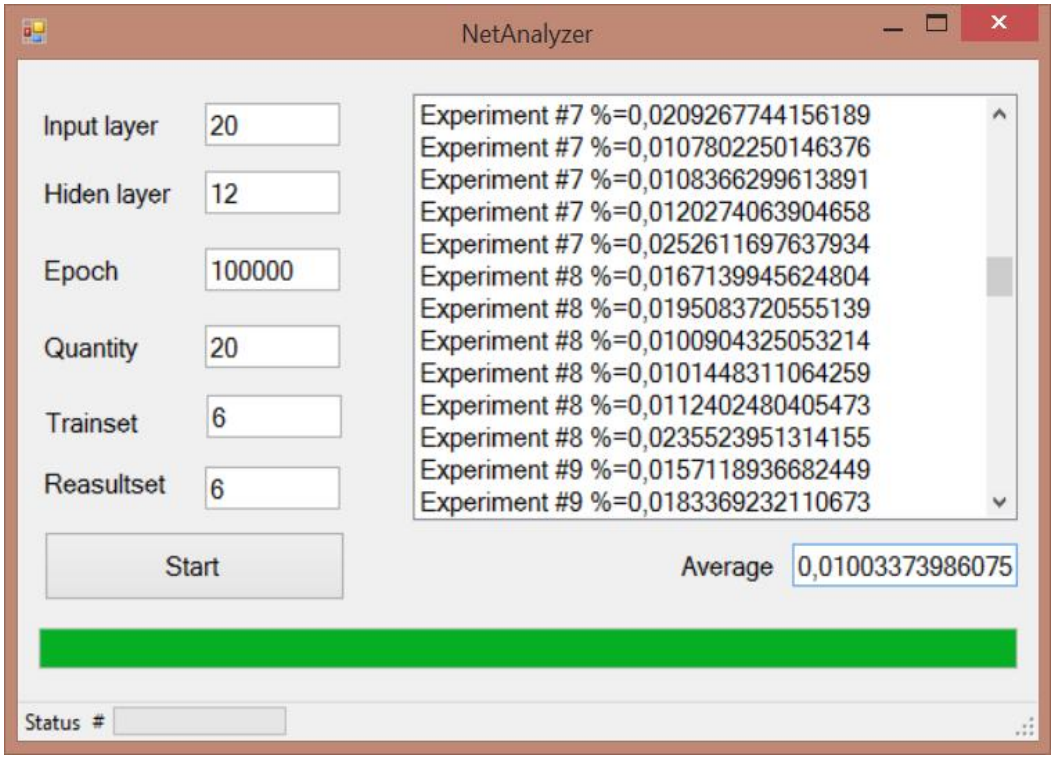

Fig. 4. Test

Thanks to the developed automated classification system there were performed several experiments with different data and the accuracy of the system was determined depending on the number of neurons in the layers of the neural network. Experiment was conducted several times for the best accuracy of the results. As a result, it was found that with increasing quantity of input neurons the accuracy of the result were reduced. With 20 Input Neurons and 12 Hidden, the error was 1 % and the training lasted for 10 seconds. With 40 Input Neurons and 26 Hidden Neurons, the error was already 6 % and training lasted for 5 minutes 20 seconds. At 100 and 66 neurons, the error was 20 % and the duration of training exceeded to 8 hours (table 1). According to the obtained data, we can conclude that the neural network will best recognize the small documents.

Table 1

Dependence of neurons quantity to the time of training and the accuracy of the neural network

Experiment number	Results		
	Neuron Input/Hidden	*Time*	*Accuracy*
1	20/12	10s	99 %
2	30/20	1m	96 %
3	40/26	5m20s	94 %
4	50/33	9m30s	92 %
5	60/40	45m	88 %
6	70/46	2h	86 %
7	100/66	8h	80 %

Thanks to the data obtained from the analysis of the test program results there was created a final software application that can categorize the real documents (fig. 5).

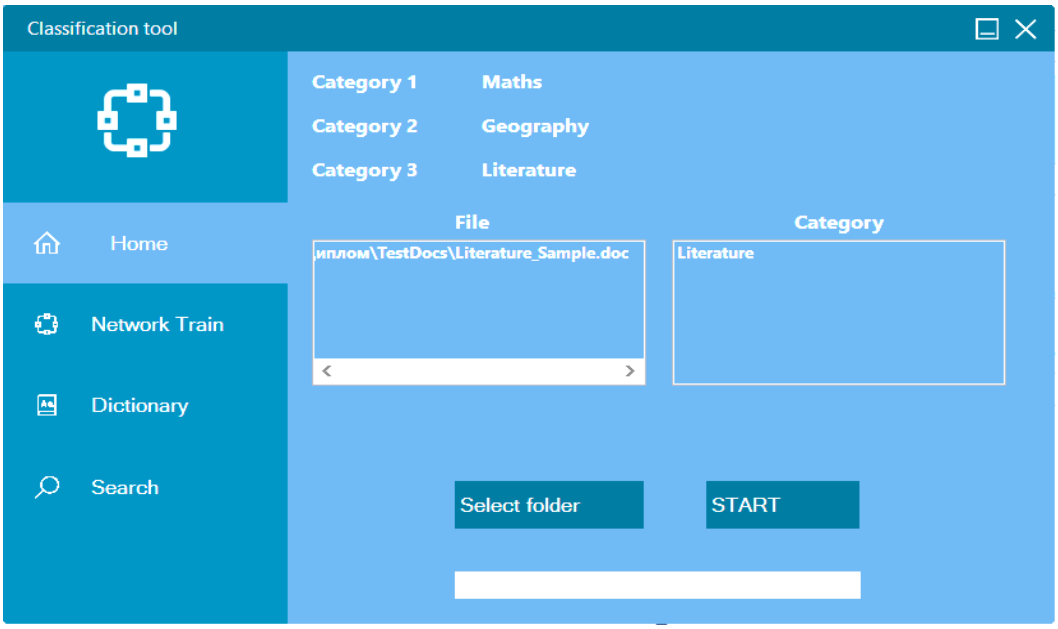

Fig. 5. Program interface

Conclusions and further researches directions. In this paper was proposed an algorithm for categorizing texts by category using a neural network with backpropagation as a learning method. Also was solved the task of profiteering input texts from repetitions and punctuation marks.

Algorithms for text preparation, the neural network and the dictionary creation were implemented programmatically using C# language. The algorithms were successfully tested and tuned. Optimal algorithm settings were set up as a result of the performed experiments to provide the best result of text categorization.

However, the algorithm for converting text into vector is not perfect. He gives an accurate result only if the required word is in the dictionary. This feature requires an increase in the size of the dictionary, which can lead to an increasing CPU usage of the computer.

The result of the work is the creation of a software product capable of classifying text documents by predefined categories. The system tests were performed, the accuracy checked and all the advantages and disadvantages of the perceptron, the method of training the network and the vectorization of information were revealed. It has been proved experimentally that the system is better able to classify small documents.

In the future, the algorithms can be further elaborated using other methods of text vector creating, such as Word2Vec.

List of sources used:

1. *Енрев А. С.* Автоматическая классификация текстовых документов // Математические структуры и моделирование. 2010. № 21. С. 65–81.

2. *Bharath Bhushan S. N., Danti A.* Classification of textdocumentsbased on score level fusion approach // Pattern Recognition Letters. 2017. Vol. 94. P. 118–126. URL: https://doi.org/10.1016/j.patrec.2017.05.003

3. *Волосюк Ю. В.* Методи класифікації текстових документів в задачах TextMining // Наукові записки Українського науково-дослідного інституту зв'язку. 2014. № 6 (34). С. 76–81.

4. *Altınel B., Ganiz M. C.* Semantic text classification: a survey of past and recent advances // Information Processing & Management. 2018. Vol. 54. Issue 6. P. 1129–1153. URL: https://doi.org/10.1016/j.ipm.2018.08.001

5. *Hadi W., Al-Radaideh Q. A., Alhawari S.* Integrating associative rule-based classification with Naive Bayes for text classification // Applied Soft Computing. 2018. Vol. 69. P. 344–356. URL: https://doi.org/10.1016/j.asoc. 2018.04.056

6. *Diab D. M., El Hindi K.* Using ifferential evolution for fine tuning naive Bayesian classifiers and its application for text classification // Applied Soft Computing. 2017. Vol. 54. P. 183–199. URL: https://doi.org/10.1016/ j.asoc.2016.12.043

7. *Yang Y., Liu X.* A re-examination of text categorization methods // Proc. of Int. ACM Conference on Research and Development in Information Retrieval (SIGIR-99). 2007. P. 42–49. URL: https://doi.org 10.1145/312624.312647

8. *Корнеев В. В., Гареев А. Ф., Васютин С. В., Райх В. В.* Базы данных. Интеллектуальная обработка информации. Москва: Нолидж. 2000. 352 с.

9. *Новотарський М. А., Нестеренко Б. Б.* Штучні нейронні мережі: обчислення // Праці Інституту математики НАН України. 2004. Т. 50. 408 с.

10. *Сетлак Г.* Использование искусственных нейронных сетей для решения задач классификации в менеджменте // Радіоелектроніка. Інформатика. Управління. Нейроінформатика та інтелектуальні системи. 2004. № 1. С. 127–135.

11. *Воробьев Н. В., Пучков Е. В.* Классификация текстов с помощью сверточных нейронных сетей. URL: https://cyberleninka.ru/article/v/klassifikatsiya-tekstov-s-pomoschyu-svertochnyh-neyronnyh-setey

12. *Приходченко С. Д.* Искусственный нейрон со сложной функцией активации // Науковий вісник національного гірничого університету. 2004. Вип. № 10. С. 92–95.

13. *Akmal Haidar Md., O'Shaughnessy D.* Novel topic n-gram count LM incorporating document-based topic distributions and n-gram counts // 22nd European Signal Processing Conference. 2014. P. 2310–2314.

14. *Калініна І. О.* Дослідження алгоритмів навчання нейронних мереж в задачах прогнозування // Наукові праці [Чорноморського державного університету імені Петра Могили]. Сер.: Комп'ютерні технології. 2009. Т. 117. Вип. 104. С. 160–171.

References:

1. Eprev A. S (2010), "Automatic classification of text documents" // Mathematical structures and modeling, vol. 21, pp. 65–81 [Russia].

2. Bharath Bhushan S. N. and Danti A. (2017), "Classification of text documents based on score level fusion approach" // Pattern Recognition Letters, vol. 94, pp. 118–126, available at: https://doi.org/10.1016/j.patrec.2017.05.003

3. Volosyuk Yu. V. (2014), *"Metody klasyfikatsiyi tekstovykh dokumentiv v zadachakh TextMining"* ["Methods of classification of text and documents in the tasks Text Mining"] // *Naukovi zapysky Ukrayins'koho naukovo-doslidnoho instytutu zv'yazku* [Scientific notes of the Ukrainian Research Institute of Communication], vol. 6(34), pp.76–81 [Ukraine].

4. Altınel B. and Ganiz M. C. (2018), "Semantic text classification: A survey of past and recent advances" // Information Processing & Management, vol. 54, issue 6, pp. 1129–1153, available at: https://doi.org/10.1016/j.ipm.2018.08.001

5. Hadi W., Al-Radaideh Q. A. and Alhawari S. (2018), "Integrating associative rule-based classification with Naive Bayes for text classification" // Applied

Soft Computing, vol. 69, pp. 344–356, available at: https://doi.org/10.1016/j.asoc.2018.04.056

6. Diab D. M. and El Hindi K. (2017), "Using ifferential evolution for fine tuning naive Bayesian classifiers and its application for text classification // Applied Soft Computing, vol. 54, pp. 183–199, available at: https://doi.org/10.1016/j.asoc.2016.12.043

7. Yang Y. and Liu X. (2007), "A re-examination of text categorization methods" // Processing of Inteligent. ACM Conference on Research and Development in Information Retrieval (SIGIR-99), pp. 42–49, available at: https://doi.org/10.1145/312624.312647

8. Goldberg Y. (2016), "A primer on neural network models for natural language processing" // Journal of Artificial Intelligence Research, vol. 57, pp. 345–420.

9. Korneev V. V., Gareev A. F., Vasyutin S. V. and Reich V. V. (2000), Databases. Intellectual information processing, Press Knowledge, Moscow, 352 p. [Russia].

10. Novatorsky M. A. and Nesterenko B. B. (2004), "*Shtuchni neyronni merezhi: obchyslennya"* ["Artificial Neural Networks: Computing"] // *Pratsi Instytutu matematyky NAN Ukrayiny* [Works of the Institute of Mathematics of the National Academy of Sciences of Ukraine], vol. 50, 408 p. [Ukraine].

11. Setlak G. (2004), *"Ispol'zovaniye iskusstvennykh neyronnykh setey dlya resheniya zadach klassifikatsii v menedzhmente"* ["The use of artificial neural networks for solving classification problems in management") // Radioelektronika. Informatyka. Upravlinnya. Neyroinformatyka ta intelektual"ni systemy [Radioelectronics. Computer science. Control. Neuroinformatics and Intelligent Systems], vol. № 1, pp. 127–135 [Ukraine].

12. Vorobev N. V. and Puckov E. B. (2017), *Klassifikatsiya tekstov s pomoshch'yu svertochnykh neyronnykh setey* [The text classification using ultra-precise networks], available at: https://cyberleninka.ru/article/v/klassifikatsiya-tekstov-s-pomoschyu-svertochnyh-neyronnyh-setey [Russia].

13. Prihodcenko S. D. (2004), *"Iskusstvennyy neyron so slozhnoy funktsiyey aktivatsii"* ["Artificial neuron with a complex activation function"] // *Naukovyy visnyk national'noho hirnychoho universytetu* [Scientific Bulleten of National Mining University], vol. 10, pp. 92–95 [Ukraine].

14. Akmal Haidar Md. and O'Shaughnessy D. (2014), "Novel topic n-gram count LM incorporating document-based topic distributions and n-gram counts" // 22nd European Signal Processing Conference, pp. 2310–2314

15. Kalinina I. O. (2009), "Doslidzhennya alhorytmiv navchannya neyronnykh merezh v zadachakh prohnozuvannya" ["Further algorithms for neural neuralz in prediction problems"] // Naukovi pratsi [Chornomors'koho derzhavnoho universytetu imeni Petra Mohyly] [Scientific works. Series: of computer science], vol. 117, issue 104, pp. 160–171 [Ukraine].