

APPLICATION OF ARTIFITIAL NEURAL NETWORKS IN BULGARIAN LANGUAGE VOICE RECOGNITION

Hasanov Hasan, Burgas Free University, hasan.mehmedov@gmail.com

Georgieva Penka V., Burgas Free University, pgeorg@bfu.bg

Abstract: Natural language processing is one of the main areas of modern artificial intelligence. Voice recognition is an element of natural language processing and aims at transforming spoken words into written text by various techniques. Researchers in this area face many challenges that have different sources.

In this article Bulgarian Language Speech Recognition System 1.0 (BLSRS 1.0) is proposed and test results are presented. BLSRS 1.0 is based on an artificial neural network, trained to recognize the corresponding spectrograms.

Keywords: voice recognition, speech recognition, Bulgarian language processing, artificial neural networks, artificial intelligence, under-resourced language

ПРИЛОЖЕНИЕ НА ИЗКУСТВЕНИ НЕВРОННИ МРЕЖИ ЗА ГЛАСОВО РАЗПОЗНАВАНЕ НА БЪЛГАРСКИ ЕЗИК

Хасан Хасанов, Бургаски свободен университет, hasan.mehmedov@gmail.com

Пенка В. Георгиева, Бургаски свободен университет, pgeorg@bfu.bg

Абстракт: Обработването на естествени езици е една от основните области на съвременния изкуствен интелект. Гласовото разпознаване е елемент на обработката на естествени езици, при който изречените думи се преобразуват в писмен текст с помощта на различни техники. Изследователите в тази област се изправят пред множество предизвикателства от разнообразен характер.

В тази статия е представена система за гласово разпознаване на български език Bulgarian Language Speech Recognition System 1.0 (BLSRS 1.0), създадена с изкуствена невронна мрежа, тренирана да разпознава съответните спектрограми.

Ключови думи: гласово разпознаване, речово разпознаване, обработка на български език, изкуствени невронни мрежи, изкуствен интелект, език с негостатъчно ресурси

I. INTRODUCTION

The term *voice recognition* (also known as *automatic voice recognition* and *computer voice recognition*) is mainly used in two aspects: 1) converting speech into written text, and 2) training a software system to recognize a particular voice. Voice recognition applications include voice user interfaces such as voice dialling (e.g. *Call home!*), call handling (e.g. *I would like to make a call.*), managing home appliances, searching (e.g., find a podcast when certain words are spoken), entering data (e.g. entering a credit card number), preparation of structured documents (e.g. reports), voice speech during text processing (e.g. word processing or e-mails) and others.

The existing voice recognition applications are numerous, however they recognize mainly widespread languages (English) and very few them are for the so called *under-resourced*

languages. The term *under-resourced language* is introduced in [1] and refers to a language with some of the following features: lack of a unique writing system or stable orthography; limited presence on the web; lack of linguistic expertise; lack of electronic resources for speech and language processing such as monolingual corpora, bilingual electronic dictionaries, transcribed speech data, pronunciation dictionaries, vocabulary lists, etc. It is important to note that under-resourced language is not the same as a language spoken by a minority of the population of a territory [2]. One of the projects that try to fill in that gap is *Basic Language Resource Kit 7* - a joint project between *European Network of Excellence in Language and Speech* and *European Language Resources Association*. The major goal of this project is to objectively define the status of a given language and then to provide as many languages as possible with a minimal set of available language resources. Bulgarian language is considered to be an under-resource language for the lack of electronic resources for speech and language processing. Building a speech recognition system for an under-resourced language requires techniques that go far beyond the basics – there is need for new phonological systems, the grammatical structure is different and so on. The lack of resources requires innovative data collection methodologies and creative models.

The growth of available resources - *LinguisticData Consortium*, *European Language Resources Association*, *SpeechOcean*, *GlobalPhone*, etc. does not solve all the specific problems in Bulgarian language voice recognition.

This article presents an application of artificial neural networks (ANN), as part of Soft Computing [3], in *Bulgarian Language Speech Recognition System 1.0*.

II. DEVELOPMENT OF VOICE RECOGNITION TECHNIQUES

The first automated speech recognition systems are created around 1960 and are strongly influenced by the acoustic phonetics theory, which describes the phonetic elements of speech (phonemes) and explaining the implementation of these elements into a spoken language.

Olson and Belar from the RCA Laboratories introduced a system that identifies 10 syllables from one particular speaker [4], and J. Forgie and C. Forgie from Lincoln Lab of MIT designed a speaker-independent recognition system for 10 speakers [5]. At the same time, but independently, Japanese researchers created specialized hardware for a speech recognition task execution [6], [7], [8]. The work of Sakai and Doshita [7] includes the first use of speech segmentation for speech analysis and speech recognition in different partitions of the delivered speech and is considered to be the pioneer in continuous voice recognition. Denes creates a phoneme recognition system that recognizes 4 vowels and 9 consonants [9]. Martin points out the need to cope with time heterogeneity in repetitive speech segments in [10]. Vintsyuk suggests the use of dynamic programming for assessing the similarity between two speakers [11]. Sakoe and Chiba [12] propose a model of dynamic time distortions in concordance of speech patterns, and after that dynamic programming (including the Viterbi algorithm [13]) became a mandatory technique for automatic speech recognition.

Commercial systems are the next important stage of technological development of speech recognition. Atal and Itakura independently present the basic concepts of LPC (Linear Predictive Coding) as a procedure for efficient speech wave coding, represented by time parameters associated with vocal tract response [14], [15]. By the mid-1970s, the main ideas of this technology were developed by Itakura [16], Rabiner, Levinson and others [17]. Tom Martin founded the first commercial company - Threshold Technology Inc., offering VIP-100 System - a speech recognition software system. The Advanced Research Projects Agency (ARPA) establishes Speech Understanding Research (SUR) program under which the systems *Harpy* [18], *Hearsay-II* of CMU, and *HWIM* of BBW were created [19]. However, neither system has been able to achieve ARPA's ambitious goal of reaching high efficiency. Another commercial system is *DRAGON*.

The next stage of development of voice recognition systems is the use of artificial intelligence techniques, and since 1980 there has been a change in the methodology of an intuitive-pattern-oriented approach to statistical modelling based on hidden Markov model. Another technology used in natural language processing is the artificial neural network (ANN). The primary purpose of using ANN is the recognition of several phonemes or a small number of words. Later, the ANNs are integrated with hidden Markov model. [20]

In the 1990s, the task of recognizing patterns is gradually transformed to an optimization problem for minimizing the empirical error [21]. The concept of minimal empirical error results in a set of techniques such as *support vector machines* (SVM) [21], [22]. The *Sphinx* systems of CMU [23], *BYBLOS* of BBN [24] and *DECIPHER* of SRI [25] have been developed.

Since the 1990s, there has been a significant progress in many research programs around the world. All natural language processing systems become more complex and accurate. A well-structured basic software system is indispensable in further research and development for combining new concepts and algorithms. The *Hidden Markov Model Tool Kit* (HTK), developed by a Cambridge University team led by Steve Young, is one of the most widely used software systems for automatic speech recognition [26].

Applications with realistic human-like communication capabilities are *Pegasus* and *Jupiter*, created by a team of MIT [27] and *How May I Help You* (HMIHY) by AT & T [28].

Since 2014 researchers have had a significant interest in "end-to-end" automatic voice recognition in which all the components of voice recognition are learnt and one such system is *Google DeepMind* and *Navdeep Jaitly*.

An alternative approach is the so called attention-based model. The *Listen, Attend and Spell* (LAS) model literally "listens" to the acoustic signal, "pays attention" to different parts of the signal, and "transmits" the transcription of the word letter-by-letter.

Voice recognition systems for Bulgarian language, known to the authors, are *Speechnotes* and *Google Speech API*. *Speechnotes* is an online notebook with a voice recognition capability that supports many languages including Bulgarian. It provides a dictation tool using state-of-the-art speech recognition technology in order to deliver the best results and has built-in tools to increase efficiency, productivity and user comfort. *Google's API* allows converting audio to text by applying powerful neural network models. It recognizes over 110 languages including Bulgarian.

III. TYPES OF VOICE RECOGNITION AND ALGORYTHMS

Discrete Voice Recognition is mainly used in dictation applications and in voice commands. The used technique is *Isolated Word Voice Recognition* which is a process that requires a break after each word. Pause or no sound is the basic approach for determining the start and end of a spoken word. The discreet voice recognition performs voice-to-text word translation.

Continuous Voice Recognition/Connected Words Voice Recognitions does not require a significant break between words. Instead, the speaker can speak more naturally, and yet the voice recognition software understands where the word starts and where it ends. This type of recognition is a complex one and requires a much larger resource.

Spontaneous Speech Voice Recognition/Natural Speech Voice Recognition is an advanced form of continuous recognition. The benefit of spontaneous speech recognition is that the speaker does not need to talk to the computer in a different way.

Speaker Dependent Voice Recognition is a type of recognition that depends on the voice of the speaker and so requires training, done by the speaker. The result is a voice recognition system that understands the person's specific accent and voice. Voice recognition systems that depend on the speaker may be discrete or continuous.

Speaker Independent Voice Recognition is the opposite of speaker dependent voice recognition. In this case training is not required and the system can understand the speech of various speakers.

Natural Language Voice Recognition is an add-on to continuous voice recognition, and refers to the ability of a software system to understand a question or command that is spoken in a natural way. The natural voice recognition is the technology that the modern virtual assistants like *Siri*, *Alexa* or *Cortana* use.

ALGORITHMS IN BLSRS 1.0

Hidden Markov models

Hidden Markov models are widely used in modern general speech recognition systems as the speech signal can be considered to be a near-to-stationary process for a small-time interval (e.g. 10 milliseconds).

From a probability point of view, a random process is a Markov process if the conditional distribution of the probabilities of future states of the observed process depends only on the present state and not on the previous states. Markov most popular processes are Markov chains that are defined on discrete domain. These chains are used in speech recognition theory because they need little memory to model the dynamic processes.

Hidden Markov model is a Markov process with unknown parameters. Each particular parameter is a function of the probabilities of a given state so the sequence of states cannot be determined until the moment of observation. Hidden Markov models are particularly well-known for their applications in speech recognition, manuscript, gestures, and more.

The standard hidden Markov models have a number of limitations:

- the use of continuous exponential distribution;
- the transition probability depends only on the source and purpose;
- all observed conditions are dependent only on the condition that causes them, regardless of the observed adjacent conditions.

Researchers have proposed a number of techniques to address these limitations, although this does not guarantee significant improvements in speech recognition accuracy, such as continuous modelling, conditionally independent prerequisites, and so on.

Dynamic Programming

Dynamic Time Warping (DTW) is an effective method for recognizing systems with a small vocabulary of words. The algorithms for dynamic programming are published in [29].

Error evaluation

In speech recognition, three distinct types of errors are distinguished:

- *substitution* - the correct word is replaced with an incorrect word;
- *deletion* - a word is skipped;
- *insertion* - an additional word is added.

The maximizing compliance task consists of matching the number of words recognized to the correct word and calculating the number of *substituted* (*Subs*), *deleted* (*Dels*) and inserted words (*Ins*). Then the *Word Error Rate* is defined as:

$$\text{Word Error Rate} = 100\% * \frac{\text{Subs} + \text{Dels} + \text{Ins}}{N},$$

where N is the overall number of words in the sentence.

IV. BULGARIAN LANGUAGE SPEECH RECOGNITION SYSTEM 1.0 (BLSRS 1.0)

BLSRS 1.0 first records the voice of the speaker, saves it as a .wav file and sends it to a web application, which accepts the audio, clears the header and sends it to the Fourier Transform. After finding the weight of the elements, the spectrogram is generated and a black and white filter is applied. The image is displayed in the form of a binary array and is placed on the pre-trained artificial neural network for recognition. The neural network returns a number that is the index of the word in the dictionary.

Architecture of the Application

In BLSRS 1.0 the client-server concept is used for connecting two programs. The first program, called *test client*, makes a request to the second program called *a server* and awaits its response. The server accepts the request, processes it and returns a response.

The *test client* is a *UWP* application that writes the word and sends it to the server. *UWP* is a platform created by Microsoft and introduced for the first time in Windows 10, designed as an extension to the Windows Runtime platform first introduced in Windows Server 2012 and Windows 8. *UWP* supports application development for Windows using C ++, C #, VB.NET, or XAML. API is implemented in C ++, and supported in C ++, VB.NET, C #, and JavaScript. The *server* is an *ASP.NET Core* application written in C # language and is hosted on *Internet Information Services*. *REST* was used for the application's architectural style.

Converting Sound into Bits

The first step in voice recognition is to convert audio into bits. There is a basic format for uncompressed audio, *PCM*, which is usually stored as .wav in Windows or as .aiff in Mac OS. WAV is a flexible file format designed to store more or less any combination of sampling frequency or bitrate. This makes it an appropriate format for storing and archiving original records. WAV, like any uncompressed format, encodes all sounds, no matter whether it is a complex signal or absolute silence, with the same number of bits per unit of time. For example, a file containing one minute of a symphony orchestra performance will have the same size as one minute with absolute silence if both files are stored as WAV. If the two files are encoded in a compressed sound format, the first file will be a bit smaller than the original file, but the second file will take almost no place. Of course, compression formatting takes significantly longer than WAV encoding. The WAV format is based on the RIFF file format, which is similar to IFF. (Table 1)

Table 1. Structure of a .wav file

Location	Space	Description
0...3 (4 bytes)	ChunkId	Contains the symbols <i>RIFF</i> in ASCII format
4...7 (4 bytes)	ChunkSize	The size of the rest of the file
8...11 (4 bytes)	Format	Contains the symbols <i>WAVE</i>
12...15 (4 bytes)	Subchunk1Id	Contains the symbols <i>fmt</i>
16...19 (4 bytes)	Subchunk1Size	16 for PCM format
20...21 (2 bytes)	AudioFormat	PCM=1 (a value $\neq 1$ means compression of the file)
22...23 (2 bytes)	NumChannels	Mono=1, Stereo = 2

24...27 (4 bytes)	SampleRate	8000, 44100 and so on
28...31 (4 bytes)	ByteRate	$= \frac{SampleRate * NumChannels * BitsPerSample}{8}$
32...33 (2 bytes)	BlockAlign	$= \frac{NumChannels * BitsPerSample}{8}$
34...35 (2 bytes)	BitsPerSample	8 bits = 8, 16 bits = 16 and so on
36...39 (4 bytes)	Subchunk2Id	Contains the symbols <i>data</i>
40...43 (4 bytes)	Subchunk2Size	$= \frac{NumSamples * NumChannels * BitsPerSample}{8}$
44...	data	The data of the sound

Fourier Transformation

Sound wave data is divided into its components for easier processing. Then, considering how much energy there is in each of these frequency bands, a fingerprint of the audio is created. All this is done through a mathematical operation called Fourier Transformation. The end result contains the weight of each frequency range in the form of a binary array.

Generating a Spectrogram

Once the weight of each frequency range has been calculated, the result is kept in a two-dimensional array and a colour for the spectrogram is generated. In this way, the spectrogram depicts the weights of the frequency ranges.

Applying a Black and White Filter to the Spectrogram

After the spectrogram generation a black and white filter is applied with GDI +, which is a Windows-based API that provides classes such as Image and Bitmap for easier image manipulation. Image and Bitmap objects store the colour of each pixel as a 32 bit number - 8 bits for each colour from the RGB palette - red, green, blue and alpha in the range [0, 255]. One additional "false" column is added to the matrix for any linear and non-linear operation. To translate the colours of the image into black and white, the original colour vector is multiplied by the following colour matrix:

$$\begin{array}{ccccc}
 0.299 & 0.299 & 0.299 & 0 & 0 \\
 0.587 & 0.587 & 0.587 & 0 & 0 \\
 0.114 & 0.114 & 0.114 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 1
 \end{array}$$

Converting the Spectrogram into Binary

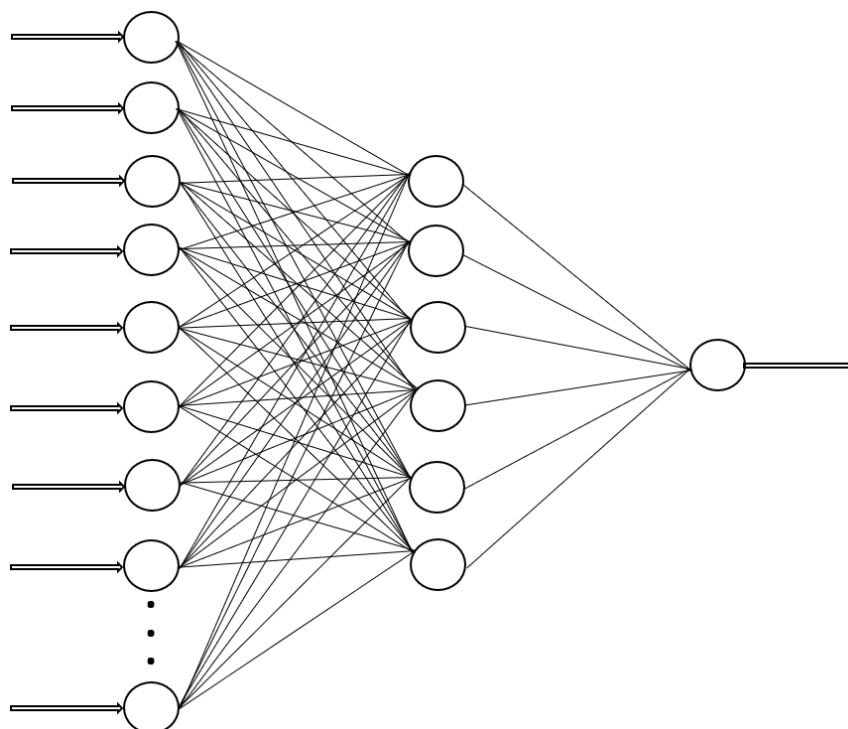
Converting the spectrogram into ones and zeros is done as follows: the image is resized to 100x100 pixels; then 1 is written if all four values (red, green, blue and alpha) are 255 and 0 is written otherwise. The total number of ones and zeros is 10000.

Initializing the ANN

The next step is to initialize the ANN with Encog. Encog is an open source library that supports artificial intelligence algorithms. The training of the ANN is done with parallel algorithms for fast performance.

The ANN consists of three layers: input layer with 10000 neurons representing the ones and zeros of the 100x100 spectrograms one hidden layer and one output layer with 1 neuron (Fig.

1). The activation function selected for neural network architecture is ReLU (rectified linear unit) based on a series of tests.



*Input Layer with
10 000 Neurons*

*Hidden Layer
with 6 Neurons*

*Output Layer
with 1 Neuron*

Fig. 1. Architecture of the ANN in BLSRS 1.0

Generating a Dictionary

The dictionary is generated from the samples and is a folder on the server machine. For example, the user can generate 100 spectrograms for all words that the application is going to recognize. The name of the dictionary consists of an index and the word itself - "0 No", "1 Yes" and so on. The spectrograms are stored in the folders (Fig. 2).



Fig. 2. Generated dictionary with two words "He" and "Да"

ANN Training

For training the neural network, all pre-generated spectrograms are taken and recorded in a single object, along with the value they must match - in this case, the word and index. The object is stored in a list and is placed in the Encog training algorithm. The training continues until the error index is less than or equal to 0.0001 or until the number of the epochs exceeds 5000.

For verification of the training the same list is run and if all the words are recognized without a single mistake, the neural network is ready for use. The user can also see if the ANN is well trained by requesting <http://URL/api/Home/>. The received result contains a number and a word. If the number is close enough to the glossary index, the neural network well-trained. (Fig. 3).

```
- {
  NeuralNetworkValue: 0,
  Word: "He"
},
- {
  NeuralNetworkValue: 0.9996871014058952,
  Word: "Да"
},
```





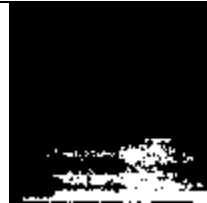




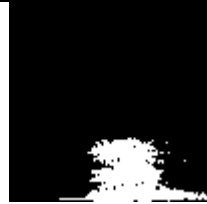
Fig. 3. Output after the ANN training for the two words “He” and “Да”

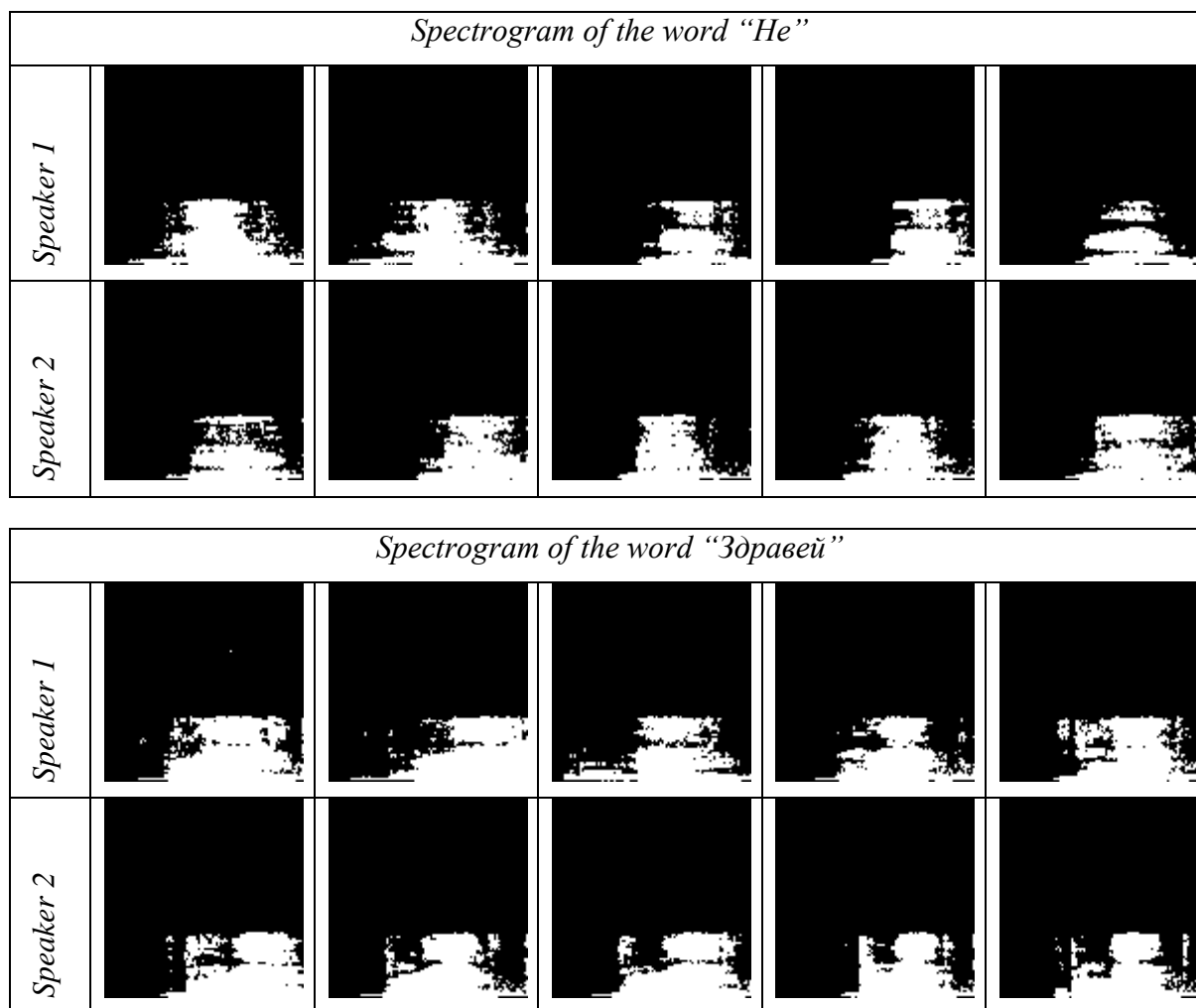
V. TESTS AND RESULTS

1. Testing Data

The initial tests are conducted for three Bulgarian words – *Да, He, Здравей*. Then two speakers are used to make 100 records for each word. In this way 300 spectrograms for each word are used. Each spectrogram has size of 100x100 pixels and is a collection of 10,000 ones and zeros. Some spectrograms are shown on Table 2.

Table 2. Spectrograms generated by the voices of two different speakers

Spectrogram of the word “Да”					
Speaker 1					
Speaker 2					



2. Comparing Activation Functions

There is no theoretical rule for determining the activation function of the ANN and therefore the best way is to go through a series of tests. For BLSRS 1.0 four types of activation functions are tested the error indexes are compared.

The type of the activation function plays a major role in the work of the neural network. Each neuron receives signals from the others, in the form of numbers, sums them, and goes through an activation function and thus determines its activation that is transmitted through connections to other neurons. Each connection has a weight that is multiplied by the signal. These weights are analogous to the strength of the synaptic pulses transmitted between the biological neurons. A negative weight corresponds to a suppressive impulse and a positive - to a stimulating one.

1. *Sigmoidal activation function* is defined by the equation:

$$y = \frac{1}{1 + e^{-a \cdot x}}$$

The sigmoidal activation function behaves best when the ANN has 6 neurons in the hidden layer with an error index of 0.004824224 and worst when there are 100 neurons in the hidden layers with an error index of 9.115.

2. *Cardinal sine activation function* is defined by:

$$f(x) = \begin{cases} 1, & \text{if } x = 0 \\ \frac{\sin(x)}{x}, & \text{if } x \neq 0 \end{cases}$$

From the tests cardinal sine function shows the best results when the neural network has 100 neurons in the hidden layer with an error index of 0.000164248898749968 and worst when there are 40 neurons in the hidden layers with an error index of 4.40920665310783.

3. *Hyperbolic tangent activation function* is defined by:

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

The hyperbolic tangent shows almost identical results for all cases. It behaves best when the neural network has 6 neurons in the hidden layer - 0.2500000000000852 and worst when there are 40 neurons in the hidden layer - 0.2500000000620091.

4. *Rectified linear activation function* is defined by:

$$f(x) = \max(x, 0)$$

According to the tests performed so far, this activation function shows the best result among all activation functions with an error index 0.000109571559015781 in case of 6 neurons.

After the tests carried out the conclusion is that the best result is achieved by the rectified linear activation function. It has an error index of 0.00010957155901 in case of 6 neurons in the hidden layer, which makes it the most efficient activation function that can be used to train the neural network with specific data. (Fig. 3)

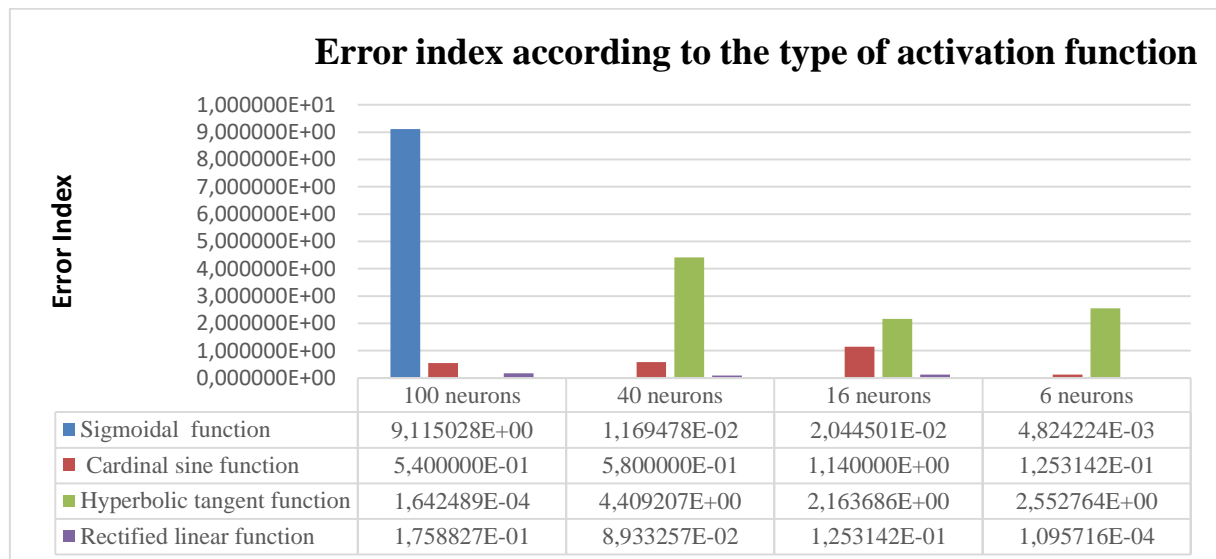


Fig. 3. Test results for different activation functions

3. Determining the Number of Epochs

For determining the number of epochs for training the ANN, a series of tests are conducted. For each number (10, 50, 100, 300, 500) the ANN has been trained 5 times and the error index and training time are measured. The received average values of the error index and training time are shown on Table 3.

After analysing the results the authors decided that 150 epochs is the most efficient number of epochs for the training process – the average error index will be small enough while the average time will be short enough.

Table. 3. Test results for different number of epochs

<i>Number of epochs</i>	<i>Average error index</i>	<i>Average training time</i>
10	19,64109	1,409135
50	0,114712	4,568209
100	0,062447	8,629199
300	0,000133	25,38428
500	0,0000465	40,72743

VI. CONCLUSION

The development of voice recognition techniques has begun in the 1960s with a small set of single words (10-100) that are recognized on the basis of the acoustic-phonetic properties of the sounds. The key techniques developed during this early period are filter analyses, time normalization methods and the beginning of sophisticated dynamic programming methodologies. In the 1970s, medium-sized vocabulary was recognized (100-1000 words), using pattern-based methods for sample recognition. The main techniques during this period are patterns recognition models, the introduction of LPC spectral presentation methods, methods for creating a cluster of patterns for speaker independent systems, and the introduction of dynamic programming methods. In the 1980s, v recognition systems using a large vocabulary (> 1000 words) are created. Key ideas implemented during this period are the hidden Markov models and the stochastic language model. In the 1990s, systems for continuous speech recognition and comprehension were introduced. Key technologies at that time are methods of stochastic understanding of language, training of acoustic and language models, and artificial neural networks. In the following decades, lexical systems with full semantic patterns, integrated with speech-synthesized systems and multi-modal inputs (keyboards, mice, etc.) are introduced. These systems perform a speech dialogue with a set of input and output modalities. Various challenges emerge in voice recognition for Bulgarian language as it is considered to be one of the under-resourced languages. The proposed in this article Bulgarian Language Speech Recognition System 1.0 (BLSRS 1.0) is a step in overcoming the lack of electronic resources for speech and language processing. The software system first converts the sound into a spectrogram, applies a Black and White Filter, and after transforming to binary, uses the previously trained ANN for recognition of the word.

Designing a machine that actually functions as an intelligent human being is still a challenge. Achievements to date are just the beginning and much more research is needed for developing a real-time speech recognition application for Bulgarian language.

References

- [1] Krauwer S., "The Basic Language Resource Kit (BLARK) as the First Milestone for the Language Resources Roadmap," in *International Workshop Speech and Computer*, SPECOM 2003, 2003.
- [2] Besacier L., Et. Barnard, Al. Karpov, T. Schultzd, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85-100, 2014.
- [3] Georgieva P., *Genetic Fuzzy Systems* (in Bulgarian), Burgas: Poligraph, 2016.
- [4] Olson H., H. Belar, "Phonetic Typewriter," *The Journal of the Acoustical Society of America*, vol. 28, no. 6, pp. 1072-1081, 1956.
- [5] Forgie J., C. Forgie, "Results Obtained from a Vowel Recognition Computer," *The Journal of the Acoustical Society of America*, vol. 31, no. 11, pp. 1480-1489, 1959.
- [6] Suzuki J., K. Nakata, "Recognition of Japanese Vowels—Preliminary to the Recognition of Speech," *J. Radio Res. Lab*, vol. 37, no. 8, pp. 193-212, 1961.
- [7] Sakai T., S. Doshita, "The Phonetic Typewriter," *The Journal of the Acoustical Society of America*, vol. 33, no. 11, 1961.
- [8] Nagata K., Y. Kato, S. Chiba, "Spoken Digit Recognizer for Japanese Language," *NEC Res. Develop*, № 6, 1963.
- [9] Denes P., "The Design and Operation of the Mechanical Speech Recognizer at University College London," *British Institution of Radio Engineers*, vol. 19, no. 4, pp. 211-229, 1959.
- [10] Martin T., A. Nelson, X. Zadell, "Speech Recognition by Feature Abstraction," Tech. Report AL-TDR-64-176, Air Force Avionics Lab, 1964.
- [11] Vintsyuk T., "Speech Discrimination by Dynamic Programming," *Kibernetika*, vol. 4, no. 2, pp. 81-88, 1968.
- [12] Sakoe H., S. Chiba, "Dynamic Programming Algorithm Quantization for Spoken Word," *Speech and Signal Proc.*, vol. 26, no. 1, pp. 43-49, 1978.
- [13] Viterbi A., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Informaiton Theory*, vol. 13, pp. 260-269, 1967.
- [14] Atal B., S. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.*, vol. 50, no. 2, pp. 637-655, 1971.
- [15] Itakura F., S. Saito, "A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies," *Electronics and Communications in Japan*, vol. 53, pp. 36-43, 1970.
- [16] I. F., "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. Acoustics, Speech and Signal Proc*, vol. 23, pp. 57-72, 1975.
- [17] Rabiner L., S. Levinson, A. Rosenberg, J. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. Acoustics, Speech and Signal Proc.*, vol. 27, pp. 336-349, 1979.
- [18] Lowerre B., "The HARPY Speech Understanding System," *Trends in Speech Recognition, Speech Science Publications*, 1986, reprinted in *Readings in Speech Recognition*, pp. 576-586, 1990.
- [19] Klatt D., "Review of the DARPA Speech Understanding Project (1)," *J. Acoust. Soc. Am.*, vol. 62, pp. 1345-1366, 1977.

- [20] Georgieva P., H. Hasanov, „Voice recognition - historical development and main techniques,“ *Computer Science and Communications* , том 6, № 1, pp. 20-55, 2017.
- [21] Juang B., C. Lee, W. Chou, “Minimum Classification Error Rate Methods for Speech Recognition,” *IEEE Trans. Speech & Audio Processing, T-SA*, vol. 5, no. 3, pp. 257-265, 1997.
- [22] Vapnik V., Statistical Learning Theory, John Wiley and Sons, 1998.
- [23] Lee K., Large-vocabulary Speaker-independent Continuous Speech Recognition: The Sphinx System, Ph.D. Thesis, Carnegie Mellon University, 1988.
- [24] Schwartz R., C. Barry , Y. Chow, etc., „The BBN BYBLOS Continuous Speech Recognition System,“ in Proc. of the Speech and Natural Language Workshop, Philadelphia, 1989.
- [25] Murveit H., M. Cohen , P. Price , etc., „SRI's DECIPHER System,“ in proceedings of the Speech and Natural Language Workshop, 1989, Philadelphia.
- [26] Young S., „the HTKBook,“ <http://htk.eng.cam.ac.uk/>.
- [27] Glass J., E. Weinstein, „SpeechBuilder: Facilitating Spoken Dialogue System Development,“ 7th European Conf. on Speech Communication and Technology, Aalborg Denmark, 2001.
- [28] Gorin A., B. Parker, R. Sachs, J. Wilpon, “How May I Help You?,” 1996.
- [29] Huang X., A. Acero, H. Hon, Spoken Language processing – A Guide to Theory, Algorithms and System Development, Prentice Hall PTR, 2001, pp. 375-407.