МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2020
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2020
Proceedings of the Forty-ninth Spring Conference
of the Union of Bulgarian Mathematicians
2020

MATHEMATICAL METHODS IN AUTOMATIC SPEECH RECOGNITION[*]

Stoyan Mihov

We present the mathematical methods which are used in the process of Automatic Speech Recognition. The presentation is divided in three parts. We start with a short overview of the vocal tract and the corresponding acoustics equations. Afterwards we introduce the digital signal processing, which is performed over the speech signal in order to extract Mel-frequency cepstrum coefficients, corresponing to the articulation configuration. In the second part we present an approach for acoustic modeling based on time-delayed deep neural networks. We discuss the methodology for the machine learning of the acoustic model. In the third part we describe the use of finite-state f-transducers for representing the language model. For decoding the signal we shortly present the Viterby and the beam-search algorithm over a Hidden Markov Model represented as a f-transducer. Finally, we show experimental results for automatic speech recognition of Bulgarian language.

1. Introduction. Because of its practical importance, the area of Automatic Speech Recognition (ASR) has been extensively studied and developed in the recent years [4, 3, 6]. During the last three decades various approaches have been applied and tested in order to improve the ASR performance in respect to preciseness and efficiency [1]. In this paper we intent to present a concise description of the mathematical methods behind a typical implementation of a modern speech recognition system. We start with the processing of the input signal in Section 2. Afterwards we describe the acoustic modeling in Section 3. In Section 4 we show the representation and implementation of the language model. Section 5 presents the speech decoding algorithm. Finally, in Section 6 we show experimental results for ASR for Bulgarian language.

2. Digital signal processing for automatic speech recognition. The vocal tract is the general name of the cavities above the larynx (throat) through which the air passes when speech is produced. In humans, it consists of a laryngeal cavity (containing the larynx and vocal cords), an oral cavity, and a nasal cavity (Fig. 1). The vocal tract is responsible for generating different sounds with the current configuration of its individual components determining what the sound itself will be like.

[*] **2020 Mathematics Subject Classification:** 68T10, 94A12, 68T07, 68Q45.

Key words: speech corpus, automatic speech recognition, low resource language.

The research presented in this paper is partially funded by the Bulgarian Ministry of Education and Science via grant DO1-200/2018 “Electronic healthcare in Bulgaria” (e-Zdrave) and grant DO1-205/2018 “Information and Communication Technologies for Unified Digital Market in Science, Education and Security”.

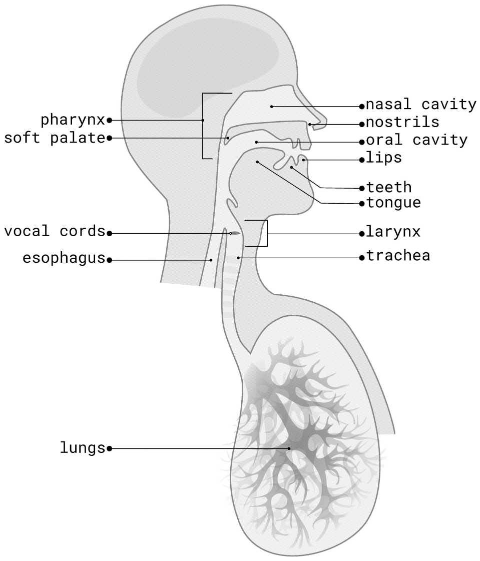

Fig. 1. Vocal tract

Let's take a closer look at Fig. 1. Speech, in fact, is simply the acoustic wave obtained at the end of the system – the lips and nostrils – as a result of expelled air from the lungs. The lungs act as an energy source for this system – the air flow resulting from contraction of the intercostal muscles and the diaphragm propagates up the trachea and through the glottis (the opening between the vocal cords). The vocal cords let out the propagated air. The oscillation of the vocal cords depends on the anatomical features. In men, the vocal cord oscillation frequency (F0) averages 125 Hz, and in women – 210 Hz. The acoustic wave resulting from the oscillation passes through the vocal tract, where it generates turbulence (swirls) when encountering barriers such as lips and teeth, and eventually leaves the system through one of the openings.

The wave propagation in the vocal tract can be modeled with the Navier-Stokes flow equations. Under some simplifications from the equations we derive that in the time domain the output signal $y(t)$ can be expressed as the convolution of the "source" $g(t)$ with the "envelope" or "filter" $h(t)$:

$$y(t) = g(t) * h(t).$$

The source corresponds to the signal generated by the vocal cord oscillations combined with the noise caused by the turbulence. The source determines the presence of noise, voice (the oscillation of the vocal cords), and the pitch. The envelope characterises the articulation configuration of the vocal tract – the relative positioning of the tongue, lips, soft palate etc. It determines the perceived sound (*phoneme*). Therefore, for the task of speech recognition we shall extract the envelope from the original signal. Applying the Fourier transform we derive the following equation in the frequency domain:

$$Y(x) = G(x).H(x).$$

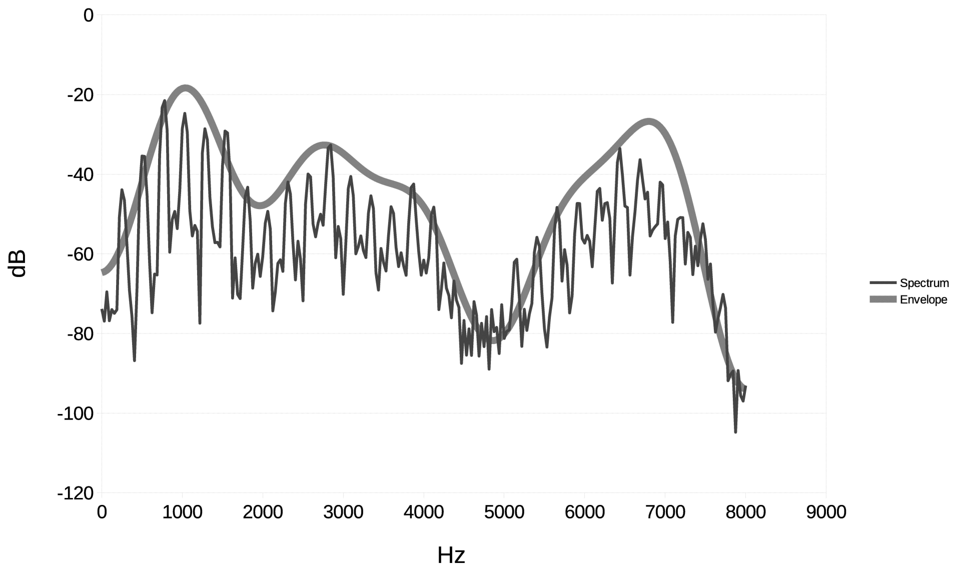

Fig. 2. Logarithm of the power of speech signal (thin line) and envelope (bold line) in the frequency domain

Taking the logarithm of the magnitude we obtain:

$$\log(|Y(x)|) = \log(|G(x)|) + \log(|H(x)|).$$

In Figure 2 the power spectrum of a vowel is shown by a thin line, which has a well expressed period of local extremums at the multiples of 280 Hz. This corresponds to the pitch of the source signal (F0). The global fluctuations shown by the bold line correspond to the envelope. If we consider the speech signal (the thin line) as a signal in time domain, then the higher frequencies correspond to the source and the lower frequencies correspond to the envelope. Hence we approximate the envelope by extracting the low frequency coefficients of the real cosine transform of power spectrum. Those are the so-called *cepstrum coefficients*.

The formulae for the digital signal processing are given in Table 1. For calculating the characteristic vectors of a speech signal we start with the audio input $s[n]$, sampled at $F_s = 16$ KHz. First an emphasis filter is applied for emphasizing the higher frequencies and for compensating the lip's radiation effect. Afterwards, the signal is sliced into frames using a 25 ms Hamming window in 10 ms steps rolling. The next step is extracting the envelope features of the signal. For that purpose we extract the Mel-frequency cepstrum coeffiecients. We compensate for some specifics of the human ear like logarithmic energy and frequency perception. To obtain the Mel-frequency power spectrum we slice the linear-frequency power spectrum using triangle windows spaced according to the Mel-scale. The characteristic feature vector we obtain from each frame consists of the 13 Mel-frequency cepstrum coefficients and their first and second derivatives (finite differences). Thus for every 10 ms of the input signal we obtain a 39-dimensional characteristic vector.

Table 1. Formulae for obtaining the Mel-frequency cepstrum coefficients

Sampled signal:	$s[n],\ n = 1, 2, \ldots$
Signal pre-emphasis:	$s' = s[n] - \alpha s[n-1],\ \alpha = 0.95,\ n = 1, 2, \ldots$
Hamming window:	$w[n] = 0.54 - 0.46\ \cos\left(\frac{2\pi(n-1)}{N}\right),\ n = 1, 2, \ldots, N$
Windowed frame:	$y_t[n-1] = s'[(t-1)N + n].w[n],\ n = 1, 2, \ldots, N$
Fourier transform:	$Y_t[k] = \sum_{n=0}^{N-1} y_t[n] \exp(-2\pi i \frac{kn}{N}),\ k = 0, 1, \ldots, N-1$
Triangle windows:	$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{k - f[m-1]}{f[m] - f[m-1]} & f[m-1] \leq k \leq f[m] \\ \frac{f[m+1] - k}{f[m+1] - f[m]} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases}$ $f[m] = \frac{N}{F_s} HerzToMel^{-1}(m \frac{HerzToMel(F_s/2)}{M+1})$
Mel-frequency power spectrum:	$C_t[m] = \log\left(\sum_{k=0}^{N-1} \lvert Y_t[k]\rvert^2 H_m[k]\right),\ m = 0, 1, \ldots, M-1$
Mel-frequency cepstrum coefficients:	$c_t[n] = \sum_{m=0}^{M-1} C_t[m] \cos\left(\pi n \frac{m+1}{2M}\right),\ n = 0, 1, \ldots, M'$

The sequence of characteristic vectors is passed to the next stages. More details on the articulation model and the DSP part might be found in [3, 6].

3. Acoustic modeling. The purpose of the acoustic model is to represent the relation between a given part of the audio signal and the phonemes in the target language. Given the characteristic vector of a frame and its surrounding frames, the acoustic model returns the posterior probability estimates for the observed speech signal $\mathbf{o}_{u,t}$ for the time frame t in utterance u, to correspond to the phoneme state s: $P(s|\mathbf{o}_{u,t})$.

A typical realization[1] of an acoustic model with a time delayed deep neural network (TD-DNN) is shown on Figure 3.The input to the network at moment t is the sequence $\mathbf{o}_{u,t}$ of the 23 characteristic vectors generated from the signal processing stage at the moments $t-13, \ldots, t+9$. Each of the four layers consists of an affine transform followed by a non-linear activation function. In other words, if the input of the layer i is $\mathbf{x}_{i-1} \in \mathbb{R}^{n_i}$ then $\mathbf{x}_i = \sigma(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i)$, where $\mathbf{A}_i \in \mathbb{R}^{m_i \times n_i}, \mathbf{b}_i \in \mathbb{R}^{m_i}$. The applied activation function σ is e.g. the ReLU nonlinearity:

$$\sigma(x) = \max(x, 0).$$

In a TD-DNN architecture the initial layer transforms are learnt on narrow contexts and

[1] For the sake of simplicity the given description does not cover some additional details like iVectors, variants of the activation and objective functions, drop-out, and many others.

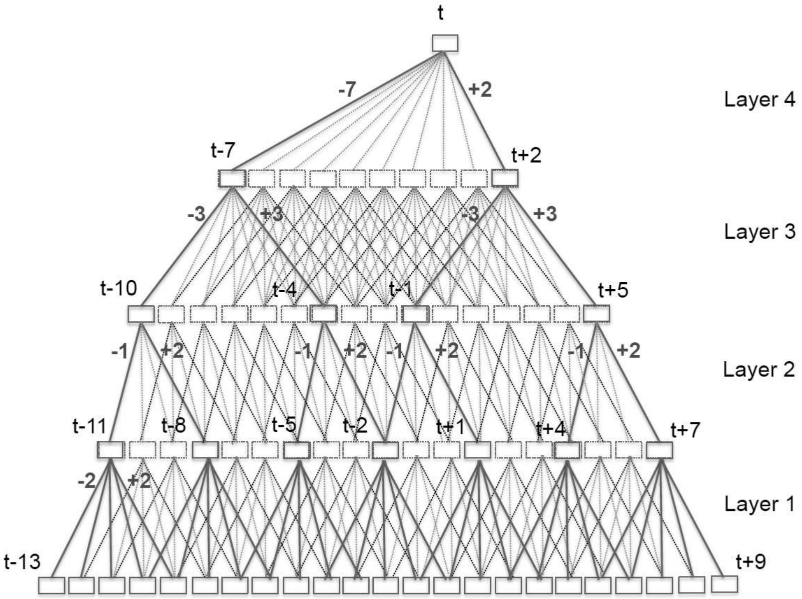

Fig. 3. Computation scheme of a sample time delayed deep neural network [5]

the deeper layers process the hidden activations from a wider temporal context. Each layer in a TD-DNN operates at a different temporal resolution, which increases as we go to higher layers of the network. Typically, there are large overlaps between input contexts of activations computed at neighbouring time steps. Therefore, they can be sub-sampled in order to increase the efficiency. A typical sub-sampling scheme is shown in bold on Figure 3. Let us denote the output of the network for a given phoneme state s for the input sequence $\mathbf{o}_{u,t}$ as $\mathbf{N}_s(\mathbf{o}_{u,t})$. The posterior probability for the phoneme state s is obtained using the softmax activation function:

$$P(s|\mathbf{o}_{u,t}) = \frac{\exp(\mathbf{N}_s(\mathbf{o}_{u,t}))}{\sum_{s'} \exp(\mathbf{N}_{s'}(\mathbf{o}_{u,t}))}.$$

From the Bayes' theorem we obtain:

$$P(\mathbf{o}_{u,t}|s) = \frac{P(s|\mathbf{o}_{u,t})P(\mathbf{o}_{u,t})}{P(s)} \propto \frac{P(s|\mathbf{o}_{u,t})}{P(s)}.$$

The networks are trained to optimize a given training objective function using the standard error back-propagation procedure. Typically, cross-entropy is used as the objective and the optimization is done through stochastic gradient descent (SGD). In our case the cross entropy is given by the equation

$$\mathcal{F}_{CE} = -\sum_u \sum_t \log P(s_{u,t}|\mathbf{o}_{u,t}),$$

where $s_{u,t}$ is the reference phoneme state label at time t for the utterance u. More details for the use of deep neural networks for ASR can be found in [6, 1].

4. Building the language model for ASR. The word level language model defines a probability distribution over all sequences of words in the target language. A typical approach for building a language model is by assuming that the probability of observing the i^{th} word w_i in the context history of the preceding $i-1$ words can be approximated by the probability of observing it in the shortened context history of the preceding $n-1$ words (n^{th}order Markov property). In that case

$$P(w_1 \dots w_m) = \prod_{i=1}^{m} P(w_i|w_1 \dots w_{i-1}) \approx \prod_{i=1}^{m} P(w_i|w_{i-(n-1)} \dots w_{i-1}).$$

The empirical estimations of the conditional probabilities are given by the equation:

$$\hat{P}(w_i|w_{i-(n-1)} \dots w_{i-1}) = \frac{count(w_{i-(n-1)} \dots w_{i-1}w_i)}{\sum_{w'} count(w_{i-(n-1)} \dots w_{i-1}w')},$$

where $count(w_{i-(n-1)} \dots w_{i-1})$ is the number of occurrences of the given sequence of words in a representative corpus of texts in the target language. The concept of *weighted finite-state transducer* provides an efficient structure for representing a language model. The top diagram in Fig. 4 presents a weighted finite state transducer for a very simple language model.

The model estimated with empirical probabilities returns zero probability for every sequence, which contains a subsequence of n words, which does not occur in the representative corpus. For instance, $\hat{P}(\alpha\alpha\gamma) = 0$ for the simple model shown in Fig. 4. To avoid this deficiency and to improve the performance of the language model a smoothing of the empirical distribution is applied. A common approach for smoothing is the back-off smoothing model. The estimation of the conditional probabilities of unseen sequences is done by backing off through progressively shorter history. Formally

$$\bar{P}(w_i|w_{i-(n-1)} \dots w_{i-1}) =$$

$$\begin{cases} d.\hat{P}(w_i|w_{i-(n-1)} \dots w_{i-1}) & \text{if } \hat{P}(w_i|w_{i-(n-1)} \dots w_{i-1}) > 0 \\ \alpha.\bar{P}(w_i|w_{i-(n-2)} \dots w_{i-1}) & \text{otherwise} \end{cases}.$$

Essentially, this means that if the n-gram has been seen in the corpus, the conditional probability of a word is proportional to the empirical estimate of that n-gram. Otherwise, the conditional probability is equal to the back-off conditional probability of the $(n-1)$-gram. To efficiently represent a smoothed back-off language model a finite-state transducer with failure transitions (f-transducer) is used. The failure transitions, which correspond to the back-off case, are followed only in case there is no regular transition with the current label. The middle diagram in Fig. 4 shows a finite-state f-transducer representing the back-off smoothed language model.

The next step is building the phoneme level language model. This is done by substituting the words in the word level language model with its corresponding sequences of phonemes. In this way we construct the phoneme level language model which defines a probability distribution over all sequences of phonemes in the target language. Given a sequence of phonemes $p_1p_2 \dots p_n$ the phoneme level language model returns the probability $\bar{P}(p_1p_2 \dots p_n)$ for observing the given sequence in the target language. This model represented with a corresponding f-transducer is shown at the bottom of Fig. 4.

The last step is building a Hidden Markov Model (HMM) for representing the conditional probability $P(\mathbf{O}|\mathbf{s})$ for emitting (producing) a given sequence of characteristic

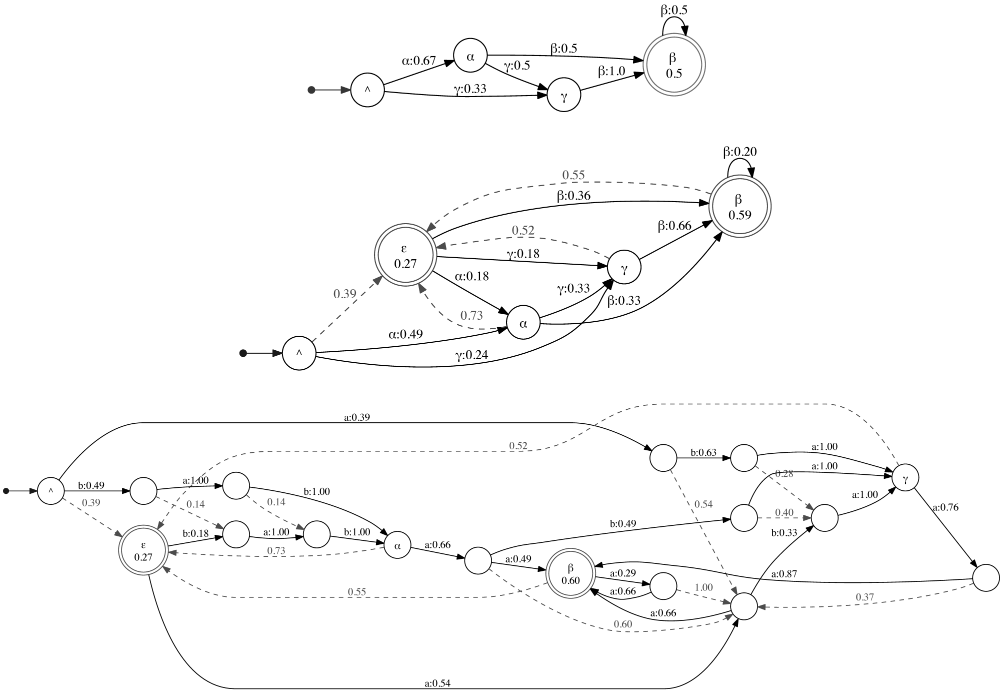

Fig. 4. *Top*: A weighted finite state transducer for representing the following second order language model. Let $\{\alpha, \beta, \gamma\}$ be the set of words in the language and the corpus for the empirical estimation of the conditional probabilities consist of the three sequences of words $\alpha\beta\beta, \alpha\gamma\beta$ and $\gamma\beta$. Then the conditional probabilities are $\hat{P}(\alpha|\wedge) = 2/3, \hat{P}(\gamma|\wedge) = 1/3, \hat{P}(\beta|\alpha) = \hat{P}(\gamma|\alpha) = 1/2, \hat{P}(\beta|\gamma) = 1, \hat{P}(\beta|\beta) = 1/2, P(\$|\beta) = 1/2$. The sign $\wedge$ marks the beginning and \$ marks the end of the sequence. *Middle*: The corresponding weighted finite state f-transducer for representing the back-off smoothed language model. *Bottom*: The corresponding f-transducer representing the phoneme level language model for the following phonetisation of the words: $\alpha = bab, \beta = aa, \gamma = aba$

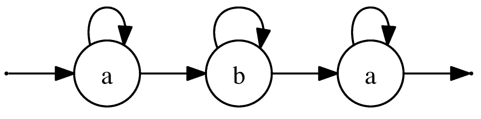

Fig. 5. A HMM model for the sequence of phonemes *aba*

vectors $\mathbf{O}$ from a given sequence of phoneme states $\boldsymbol{s}$. For that purpose each phoneme p in the phoneme level is substituted with a one-state HMM s_p. Fig. 5 shows the sub model for a subsequence of phonemes. The resulting model returns for a given sequence of phoneme states $s_1 s_2 \ldots s_n$ the probability $\bar{P}(s_1 s_2 \ldots s_n)$ for observing the given sequence in the target language. We obtain:

$$P(\mathbf{O}|\mathbf{s}) = P(\mathbf{o}_1 \dots \mathbf{o}_n | s_1 \dots s_n) = \prod_{i=1}^{n} P(\mathbf{o}_i|s_i) P(s_i|s_1 \dots s_{i-1}).$$

5. Speech decoding. The speech decoding task is reduced to finding the most likely sequence of phoneme states which emits the observed input.

$$\hat{\mathbf{s}} = \operatorname*{argmax}_{\mathbf{s}} P(\mathbf{O}|\mathbf{s}).$$

After finding the most likely phoneme state sequence, the corresponding word sequence is derived from the correspondding path in the f-transducer.

This optimization task is solved using the so-called *Viterby algorithm.* The essence of the Viterby algorithm is to calculate inductively at each time step t the probability of observing the sequence $\mathbf{o}_1 \dots \mathbf{o}_t$ and finishing in HMM state i. In other words,

$$\delta_t(i) = \max_{s_1 \dots s_{t-1}} P(\mathbf{o}_1 \dots \mathbf{o}_t | s_1 \dots s_{t-1} s_t = i).$$

Using this dynamic programming scheme the path in the HMM which maximizes the target probability is found in time quadratic to the number of states in the HMM. For providing a practical solution we apply a reduction of the full Viterby search where at each step we consider only a fixed number of states, which produce the highest probabilities. This heuristic algorithm is called *Beam search.* The theory of HMMs is presented in many textbooks (see e.g [3]).

6. ASR experiments and results. We present the ASR results using the TD-DNN based acoustic model trained on the BG-PARLAMA datasets [2]. We tested the ASR accuracy on the BG-PARLAMA development and the BG-PARLAMA test sets. Table 2 summarizes the recognition results. The best ASR result on the test data is achieved by the TD-DNN model trained on the BG-PARLAMA dataset – 6.80% word error rate (WER).

Table 2. ASR word error rate on different datasets

Acoustic model	BG-PARLAMA dev	BG-PARLAMA test
TD-DNN	7.45%	6.80%

REFERENCES

[1] L. Deng, Y. Liu. Deep Learning in Natural Language Processing, Springer, 2018.

[2] D. Geneva, G. Shopov, S. Mihov. Building an ASR Corpus Based on Bulgarian Parliament Speeches. In: Statistical Language and Speech Processing (Eds C. Martín-Vide, M. Purver, S. Pollak). Springer International Publishing, 2019, 188–197.

[3] X. Huang, A. Acero, H.-W. Hon. Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice hall PTR, 2001.

[4] F. Jelinek. Statistical Methods for Speech Recognition. MIT press, 1997.

[5] V. Peddinti, D. Povey, S. Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In: INTERSPEECH, 2015.

[6] D. Yu, L. Deng., Automatic Speech Recognition. Springer, 2016.

Stoyan Mihov
IICT – BAS
Acad. G. Bonchev Str., Bl. 2
1113 Sofia, Bulgaria
e-mail: stoyan@lml.bas.bg

Математически методи в автоматичното разпознаване на реч

Стоян Михов

Представяме математическите методи, които се използват в процеса на автоматично разпознаване на речта. Презентацията е разделена на три части. Започваме с кратък преглед на вокалния тракт и съответните уравнения на акустиката, които описват процеса. След това представяме цифровата обработка на сигнала, която се осъществява над речевия сигнал, за да се извлекат коефициентите на Мел-честотния кепструм, съответстващи на конфигурацията на артикулацията. Във втората част представяме подход за акустично моделиране, базиран на забавени във времето дълбоки невронни мрежи. Разглеждаме и методологията за машинно обучение на акустичния модел. В третата част описваме използването на монотонни стохастични f-преобразуватели за представяне на езиковия модел. За декодиране на сигнала представяме накратко алгоритъма на Витерби и алгоритъма за търсене по лъча върху стохастичния f-преобразувател. Накрая показваме експериментални резултати за автоматично разпознаване на реч на български език.