



Fakulta matematiky, fyziky a informatiky  
Univerzity Komenského  
v Bratislave

---

**Porovnanie úspešnosti rozpoznávania  
izolovaných slov rôznymi prístupmi  
pri telefónnej kvalite mobilných  
telefónov**

Diplomová práca  
2002

Autor: Jaroslav Ondriska  
Vedúci diplomovej práce: Mgr. Marek Nagy

Čestne prehlasujem, že túto diplomovú prácu som vypracoval samostatne,  
s použitím literatúry a elektronických dokumentov uvedených v zozname.

---

Jaroslav Ondriska

# Pod'akovanie

Rád by som poďakoval všetkým, ktorí stáli pri mne a podporovali ma počas štúdia, písania tejto práce a vývoja programov. Ďakujem svojej rodine a priateľom, hlavne

ALICI ONDRISKOVEJ, TIBOROVI ONDRISKOVI, MARIANOVI ONDRISKOVI  
a MARTINE ŠEBOVEJ.

Osobitne by som chcel poďakovať aj môjmu diplomovému vedúcemu

MAREKOVI NAGYOVI

za čas, ktorý mi venoval a motiváciu, ktorou vo mne vzbudil záujem o túto tému.

Ďalej by som chcel poďakovať všetkým príbuzným, priateľom, známym a neznámym, ktorí mi pomohli získať hlasové vzorky do tejto práce, špeciálne učiteľskému zboru a žiakom tercie A a B na Evanjelickom lýceu v Bratislave.

Za umožnenie nahrávania hlasových vzoriek by som chcel poďakovať spoločnosti Globtel a špeciálne za ochotu

MIROSLAVOVI PANČÍKOVI,

ktorý mi venoval svoj čas a poskytol technickú podporu.

Nakoniec by som chcel poďakovať firme Mayhem Studios, ktorá mi poskytla hardware na beh experimentu.

Ďakujem.

Bratislava, máj 2002

Jaroslav Ondriska

## **Abstrakt**

V práci upriamime pozornosť na tri z množstva modelov umožňujúcich klasifikáciu izolovaných slov a porovnáme ich z hľadiska úspešnosti, pri kvalite zvuku mobilných telefónov. V úvodných kapitolách si načrtujeme základné formy spracovania signálu a stručne si popíšeme princípy hľadania hraničných slov v hovore. V nadväznosti na to si predstavíme niektoré z najčastejšie používaných príznakových koeficientov a ukážeme si niektoré základné algoritmy predspracovania obrazov slov. Potom zameriame našu pozornosť na tri modely rozpoznávačov izolovaných slov, ktoré sú jadrom neskoršieho experimentu. V skratke podiskutujeme o kvalite signálu mobilných telefónov a potom sa zameriame na podstatnú časť tejto práce, a to popisu implementácií, a príprave, popisu a vyhodnoteniu experimentu. Naším cieľom je ponoriť sa do hĺbky problematiky z implementačného hľadiska a poukázať na niektoré implementačné problémy, komplikácie a detaily, ktoré možno neboli spomenuté v literatúre. Čomu sa v tejto práci nebudeme venovať je akákoľvek vyššia úroveň analýzy (ako syntaktická, sémantická, atď.), ani porozumeniu prirodzeného jazyka, ktoré majú nepochybne veľký význam v tejto oblasti umelej inteligencie. Taktiež zachádzanie do prílišných teoretických detailov je mimo rozsah tejto práce.

# Obsah

<b>1 Úvod</b> .....	<b>1</b>
<b>2 Určovanie hraníc slov</b> .....	<b>3</b>
2.1    Krátkodobé analýzy v časovej oblasti.....	3
2.2    Hľadanie hraničných bodov .....	5
<b>3 Predspracovanie signálu</b> .....	<b>8</b>
3.1    Analýzy vo frekvenčnej oblasti.....	8
3.1.1    Fourierova transformácia .....	8
3.1.2    Kepstrálna analýza .....	9
3.1.3    Lineárna prediktívna analýza .....	9
3.1.4    Kepstrálna LPC analýza.....	10
3.2    Miery odlišnosti (metriky) .....	11
3.3    Úprava obrazov slov.....	12
3.4    Vektorová kvantizácia.....	14
<b>4 Tri teoretické modely klasifikátorov</b> .....	<b>16</b>
4.1    Dynamické krivenie časovej osi (DTW).....	17
4.1.1    Klasifikácia.....	17
4.1.2    Trénovanie.....	19
4.2    Skryté Markovovské modely (HMM).....	20
4.2.1    Zisťovanie pravdepodobnosti generovania postupnosti.....	22
4.2.2    Trénovanie.....	24
4.3    Neurónové siete (NN) .....	27
4.3.1    Klasifikácia.....	28
4.3.2    Trénovanie.....	28
4.3.3    Predspracovanie vstupov a spracovanie výstupov .....	32
<b>5 Kvalita zvuku mobilných telefónov</b> .....	<b>33</b>
5.1    Kódovanie reči v GSM.....	33
5.2    Kódovanie dátových kanálov a modulácia.....	35
5.3    Dialogic Vox .....	36
5.4    Možné skreslenia signálu .....	36
<b>6 Popis aplikácií</b> .....	<b>38</b>
6.1    Návrh a spoločné črty.....	38
6.1.1    Zdroj vlny (signál).....	39
6.1.2    Primárne spracovanie signálu.....	39
6.1.3    Zisťovanie hraníc slov.....	39
6.1.4    Sekundárne spracovanie signálu .....	39
6.1.5    Vzory.....	39

## Obsah

---

6.1.6	Predspracovanie vzoriek .....	40
6.1.7	Klasifikátor .....	40
6.1.8	Ukladanie vzoriek .....	40
6.1.9	Výsledok .....	40
6.2	Popis implementácií .....	40
6.2.1	Spoločné časti .....	42
6.2.2	Časti špeciálne pre SrDTW .....	45
6.2.3	Časti špeciálne pre SrHMM .....	45
6.2.4	Časti špeciálne pre SrNN .....	46
6.3	Popis z užívateľského hľadiska .....	48
<b>7</b>	<b>Experiment .....</b>	<b>53</b>
7.1	Príprava .....	53
7.2	Priebeh experimentu .....	56
7.3	Problémy .....	57
7.4	Výsledky .....	58
7.5	Interpretácia výsledkov .....	62
7.6	Záver .....	63
<b>8</b>	<b>Záver .....</b>	<b>65</b>
<b>Dodatky</b>		
A.	Trénovací papier na rozpoznávanie izolovaných slov .....	I
B.	Výsledky prvých 100 DTW klasifikátorov .....	II
C.	Výsledky prvých 100 NN klasifikátorov .....	VI
D.	Výsledky HMM klasifikátorov .....	X
E.	Priložené CD .....	XII
<b>Literatúra</b>		

# Kapitola 1

## Úvod

Jednou z najprirodzenejších vlastností človeka je reč. Je pre neho tak bežná, že si niekedy ani neuvedomuje, aká je pre neho samozrejماً. V modernej dobe je efektívna komunikácia veľmi dôležitá zložka života. Nie je tomu inak ani v dorozumievaní s prístrojmi. V ostatnom čase sa začíname bežne stretávať s prístrojmi, ktoré možno istým spôsobom ovládať hlasom. Používanie reči ako formy zadávania vstupov prístrojom je sen ľudí už od dávna. Prvým krokom k automatickému rozpoznávaniu reči bol experiment v päťdesiatych rokoch, kedy vedci Davis, Biddulph a Balashek skonštruovali systém rozpoznávajúci číslice. Vedecké kruhy v tej dobe boli názoru, že sa im čoskoro podarí zrealizovať prístroj, ktorý bude rozumieť ľudskej reči. Od tej doby už uplynulo veľa času a aj napriek veľkému pokroku v tejto oblasti umelej inteligencie sa ľuďstvu takýto prístroj zatiaľ nepodarilo skonštruovať. Prelomiť tento problém je aj v súčasnosti jedna z popredných tém umelej inteligencie.

Jeden z jednoduchších problémov v automatickom rozpoznávaní reči je problém klasifikácie izolovaných slov. Odpadá tu množstvo problémov spojených s vyššími analýzami, ktoré nastávajú pri "naslúchaní" plynulej reči. Ak sú tieto slová len povely, nemá ani zmysel používať žiadne kontextové analýzy, ktoré inak nepochybne zmysel majú v rozpoznávaní plynulej reči. Klasifikácia slov spadá pod všeobecný klasifikačný problém. Ten spočíva v nájdení takých parametrov modelu s využitím tréningových dát, aby chyba ich klasifikácie bola čo najmenšia. Potom je pravdepodobné, že model bude vedieť klasifikovať aj prípady, ktoré sa v tréningových dátach nenachádzali.

Počas v dnešnej dobe už dlhej histórie sa podarilo vedcom a inžinierom navrhnuť a realizovať množstvo rôzne úspešných výpočtových modelov na klasifikáciu izolovaných slov. Prvý z nich bol prístup ovplyvnený využitím techniky dynamického programovania, ktorý sa stal najfrekvencovanejším v sedemdesiatych a na začiatku osemdesiatych rokov na klasifikáciu slov a slovných spojení v dnešnej dobe známy ako DTW. V priebehu osemdesiatych rokov sa vyvinul a zdokonalil nový prístup, ktorý je založený na modelovaní slov, resp. subslovných jednotiek tzv. skrytými Markovovskými modelmi. Atraktivnosť tohoto prístupu spočíva tak v dostatočnej pružnosti modelovania slov ako aj v schopnosti jeho využitia v prípade klasifikácie slov z rozsiahlych slovníkov, prípadne v rozpoznávaní plynulej reči. V päťdesiatych rokoch začal vývoj jedného všeobecnejšieho prístupu klasifikácie - neurónových sietí. Ich výskum sa však po uvedení dôkazu neschopnosti naučiť perceptrón klasifikovať lineárne neseparovateľné problémy Minskym a Papertom zastavil až do polovice osemdesiatych rokov, kedy bol tento problém pokorený.

## Úvod

---

V krátkom čase sa stali neurónové siete centrom záujmu a začali byť úspešne používané na rôzne klasifikačné problémy.

Nás bude v tejto práci zaujímať úspešnosť klasifikácie izolovaných slov s využitím každého zo spomenutých prístupov.

# Kapitola 2

## Určovanie hraníc slov

Určiť správne hranice slova je jedna z najdôležitejších častí rozpoznávania izolovaných slov. Na druhej strane aj jedna z najťažších, pretože rečník sa často nachádza v prostredí s podstatným rušivým pozadím. Vychádzajúc z [PSU] možno povedať, že je veľmi obtiažne definovať presný začiatok a koniec slova, ak

- sú na začiatku, či na konci slova neznelé okluzívy, t.j. "p", "t", "k"
- sú na konci slova znelé okluzívy "b", "d", "g", pretože sa menia na svoje neznelé protajšky "p", "t", "k"
- sú na začiatku, či na konci slova slabé neznelé frikatívy, ako napr. "f", "s", "š", "ch"
- sú na konci slova znelé frikatívy "v", "z", "ž", "h", pretože sa menia na ich neznelé protajšky "f", "s", "š", "ch"
- sú na konci slova nosové spoluhlásky "m", "n", "ň"

Našťastie nie vždy je treba nájsť presné hranice slov. My si v tejto kapitole ukážeme, ako sa dajú nájsť dostatočne dobré hranice aj pri šume a vzdialenom hovore.

### 2.1 Krátkodobé analýzy v časovej oblasti

Na to, aby sme mohli popísať niektoré základné techniky a techniku použitú v experimentoch, potrebujeme sa oboznámiť s niektorými metódami krátkodobej analýzy v časovej oblasti. V celej tejto časti budeme vychádzať z [PSU] a pracovať s predstavou približnej stacionarity signálu. Odtiaľ pochádza názov "krátkodobé analýzy".

Prvou z nich je krátkodobá energia signálu  $E_n$  definovaná vzťahom

$$E_n = \sum_{k=-\infty}^{\infty} (s(k)w(k-n))^2,$$

kde  $s(k)$  je vzorka signálu získaný PCM<sup>1</sup> v čase  $k$ ,  $w(n)$  je váhová postupnosť alebo tzv. okienko, ktorým sa vyberajú resp. vážia vzorky  $s(k)$  a  $n$  je začiatok mikrosegmentu, ktorý sa obvykle volí  $n = Ni$  pri mikrosegmentoch, ktoré na seba nadväzujú a  $n = Ni/2$  pri mikrosegmentoch, ktoré sa prekrývajú o polovicu, pričom  $N$  je počet vzoriek v mikrosegmente a  $i = \dots, -1, 0, 1, \dots$  je číslo mikrosegmentu.

---

<sup>1</sup> Pulse Code Modulation - štandardné nekomprimované digitálne kódovanie zvuku

## Určovanie hraníc slov

---

Úlohou okienka je vybrať príslušné vzorky signálu a prideliť im určitú váhu. Najčastejšie sa používa pravouhlé a Hammingovo okienko. Pri pravouhlom okienku je aplikovaná rovnaká váha na všetky vzorky mikrosegmentu. Je definované vzťahom

$$w(n) = 1, \text{ pre } 0 \leq n \leq N - 1$$
$$w(n) = 0, \text{ pre ostatné } N.$$

Často je treba potlačiť váhu vzoriek na koncoch mikrosegmentu. Potom je vhodné použiť Hammingovo okienko definované vzťahom

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi}{N-1}n\right), \text{ pre } 0 \leq n \leq N - 1$$
$$w(n) = 0, \text{ pre ostatné } N.$$

Okienko sa v podstate pohybuje pozdĺž časovej osi s nejakým krokom. Keď zvažíme dôsledky jeho pôsobenia na signál, t.j. že vzorky mimo okienka sú vážené 0, môžeme vzťah na výpočet krátkodobej energie prepísať do tvaru

$$E = \sum_{k=0}^{N-1} (s(k)w(k))^2,$$

pričom predpokladáme, že vzorky  $s(k)$  sa vzťahujú na konkrétny mikrosegment (prvá vzorka každého mikrosegmentu je  $s(0)$ ). Tento tvar je už z implementačného hľadiska lepší.

Jednou z nevýhod krátkodobej energie je jej vysoká citlivosť na veľké zmeny signálu. Reč samotná má veľkú dynamiku a mocnina v jej predpise ju ešte umocňuje. Z týchto dôvodov sa často využíva krátkodobá energia, ktorá tento nedostatok nemá

$$I_n = \sum_{k=-\infty}^{\infty} |s(k)|w(k-n),$$

alebo po úprave na konkrétny mikrosegment

$$I = \sum_{k=0}^{N-1} |s(k)|w(k).$$

V literatúre sa zvykne označovať ako  $M_n$ .

Ďalšou z charakteristík signálu je funkcia stredného počtu priechodov signálu nulou. Je možné ju definovať nasledovným spôsobom

## Určovanie hraníc slov

---

$$Z_n = \sum_{k=-\infty}^{\infty} |\operatorname{sgn}(s(k)) - \operatorname{sgn}(s(k-1))| w(k-n),$$

kde  $\operatorname{sgn}(x) = 1$ , pre  $x \geq 0$  a  $\operatorname{sgn}(x) = -1$ , pre  $x < 0$  a  $w(n)$  je pravouhlé okienko. Podobne sa dá prepísať do tvaru bližšieho implementácii

$$Z = \sum_{k=1}^{N-1} |\operatorname{sgn}(s(k)) - \operatorname{sgn}(s(k-1))|$$

V literatúre možno nájsť ešte rôzne modifikácie tejto funkcie.

Poslednou krátkodobou analýzou v čase, ktorú si uvedieme, je krátkodobá autokorelačná funkcia. Je definovaná vzťahom

$$R_n(m) = \sum_{k=-\infty}^{\infty} s(k)w(k-n)s(k+m)w(k+m-n).$$

Vykazuje niektoré vlastnosti, pre ktoré je používaná hlavne pri indikácii periodicity signálu. Ak je totiž spracovávaný signál periodický s periódou  $P$ , nadobúda  $R_n(m)$  maximálne hodnoty práve pre  $m = 0, P, 2P, \dots$ . Z uvedeného dôvodu sa používa aj pri výpočte koeficientov lineárnej prediktívnej analýzy, ktorú si vysvetlíme v jednej z nasledujúcich kapitol.

## 2.2 Hľadanie hraničných bodov

Spomenuté dôvody obtiažnosti určovania hraníc slova naznačujú, že samostatný priebeh intenzity signálu na určenie hranice slov nestačí. Preto je pre korektné vymedzenie hraničných bodov potrebné využitie spektrálnych charakteristík signálu.

Jedna z takýchto metód, navrhnutá Psutkom a Chánom [PCH] využíva napr. priebehy krátkodobej intenzity a krátkodobej funkcie stredného počtu priechodov signálu nulou. Je v nej pre určenie začiatku slova najprv stanovený horný prah intenzity  $H_I$  a horný prah počtu priechodov nulou  $H_Z$ . Ak prekročí intenzita signálu prah  $H_I$  v  $n (=3..5)$  mikrosegmentoch za sebou, dá sa uvažovať o výskyte začiatku slova. Ak je začiatok slova tvorený hláskou s vysokou frekvenciou, ale malou intenzitou (napr. hlásky "f", "s"), prejaví sa to zvýšenou hodnotou stredného počtu priechodov signálu nulou, takže počiatočný hraničný bod bude dodatočne odvodený z prekročenia  $H_Z$ . Podobne je určovaný aj koniec slova. Sú stanovené dolné prahy  $D_I$  a  $D_Z$  a koniec slova je určený, keď obe zodpovedajúce charakteristiky poklesnú pod tieto prahy a nejde pritom o oklúziu. Pripomeňme, že oklúzia je energetická pauza vznikajúca v signále pred vyslovením okluzívy, napr. hlások "p", "t", "k". Dá sa preto konštatovať, že je vo všeobecnosti možné koniec slova určiť až po časovom intervale vymedzenom pre prípadnú okluzívu, teda približne 150-200 ms. Z dôvodov

## Určovanie hraníc slov

---

dodatočného určovania začiatočného bodu slova je treba priebežne uchovávať príznakovú reprezentáciu minulých segmentov v súhrnnej dĺžke aspoň 200 až 250 ms a priebežne zabúdať staršie. Hodnoty  $H_I$  a  $D_I$  je dobré na základe experimentálneho overenia položiť

$$H_I = I_S + 4\sqrt{I_D} \quad \text{a} \quad D_I = I_S + 2\sqrt{I_D},$$

kde  $I_S$ , resp.  $I_D$ , je stredná hodnota intenzity šumu, resp. jej disperzia, určená vhodným počtom mikrosegmentov v pauze medzi slovami. Analogicky sa dá vypočítať aj  $H_Z$  a  $D_Z$ .

Keďže sa práve pauza medzi slovami ťažko určuje, pretože oddelovač slov ešte len budujeme a v našich experimentoch nepotrebujeme až takú presnosť určovania hraníc slov, uvedieme ešte jednu metódu, ktorú sme navrhli a použili. Ide o metódu, v ktorej je snaha prispôbovať dané prahové hodnoty šumu okolia. Dolné a horné hranice sú pritom rovnaké.

Princíp metódy spočíva v tom, že si pamätáme krátkodobú intenzitu  $I^{t-T+1} \dots I^t$ , stredný počet priechodov signálu nulou  $Z^{t-T+1} \dots Z^t$  a stredný počet priechodov signálu nulou vyhladený konvolučným jadrom  $\{1,1,2,3,4,4,4,4,3,2,1,1\}$   $Z^{t-T+1} \dots Z^t$  posledných  $T$  mikrosegmentov a podľa nich určujeme prahové hranice. Tie upravujeme každý časový úsek nasledovným spôsobom

$$H_I^t = c_m^I \min_{k=0..T-1} \{I^{t-k}\} + c_a^I N$$
$$H_Z^t = c_m^Z \min_{k=0..T-1} \{Z^{t-k}\} + c_a^Z N,$$

kde  $N$  je počet vzoriek v mikrosegmente a  $c_m^I, c_m^Z, c_a^I, c_a^Z$  sú multiplikatívne a aditívne konštanty, ktorými sa dá regulovať citlivosť odstupe signálu od šumu. Možný začiatok slova je určený vtedy, ak

$$\forall k \in \{0, \dots, c_{cons}^Z\} : Z^{t-k} > H_Z^t \quad \text{alebo}$$
$$\forall k \in \{0, \dots, c_{cons}^I\} : I^{t-k} > H_I^t,$$

kde  $c_{cons}^Z, c_{cons}^I$  je počet za sebou idúcich mikrosegmentov, ktoré majú spĺňať podmienku o strednom počte priechodu signálu nulou, resp. intenzite. Koniec slova je určený vtedy, keď  $c_{max,s}$  za sebou idúcich mikrosegmentov spĺňa podmienku

$$\forall k \in \{0, \dots, c_{cons}^Z\} : Z^{t-k} \leq H_Z^t \quad \text{a}$$
$$I^t \leq H_I^t.$$

## **Určovanie hraníc slov**

---

Použitie tejto metódy predpokladá, že disperzia šumu veľmi nekolíše, pretože je skrytá v konštantách. Táto metóda taktiež nehľadá natoľko presné hranice slova ako predošlá, a preto je vhodné rozšíriť určený úsek o pevné časové intervaly pred začiatkom a po konci slova. Je pritom vhodné voliť interval  $\approx 200 - 300$  ms.

# Kapitola 3

## Predspracovanie signálu

### 3.1 Analýzy vo frekvenčnej oblasti

Čisto teoreticky by sa dalo rozpoznávať reč priamo zo signálu získaného PCM. Táto reprezentácia sa však pre vysokú informačnú rýchlosť a značnú redundanciu príliš nehodí a zvyčajne sa tento signál považuje za prvotný a spracúva sa ďalej rôznymi analýzami vo frekvenčnej oblasti. Klasifikátory izolovaných slov zvyčajne používajú jednu z metód popísaných v tejto časti. Obraz slova je potom vyjadrený ako postupnosť krátkodobých charakteristík (vektorov príznakov), najčastejšie kepstrálnych LPC koeficientov, kepstrálnych koeficientov alebo výstupov z pásmových filtrov popísaných v tejto kapitole. Nebudeme však tieto metódy popisovať do detailov, ukážeme len, ako sa dajú tieto príznaky počítať. Čitateľ ich môže nájsť podrobne popísané v literatúre [PSU].

#### 3.1.1 Fourierova transformácia

Krátkodobá Fourierova transformácia  $S(\omega, n)$  vyjadruje, nakoľko vystihuje periodický signál s uhlovou rýchlosťou (frekvenciou)  $\omega$  daný signál  $s(k)$  získaný PCM. Jej predpis vyzerá nasledovne

$$S(\omega, n) = \sum_{k=-\infty}^{\infty} s(k)h(k-n)e^{-i\omega k},$$

kde  $n$  je diskretný čas a  $\omega$  je (spojitá) frekvencia,  $h(n)$  je funkcia okienka, ktorá vyberá a váži vzorky signálu. Podľa [RAS] je túto funkciu vhodné vyčíslňovať v rovnomerných intervaloch  $\omega_r = 2\pi/R, r = 0, 1, \dots, R-1$  pre  $R = N$ , kde  $N$  je počet vzoriek v mikrosegmente. Preto sa využíva tzv. diskretná Fourierova transformácia

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-i\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x(n)W_N^{-kn},$$

kde  $x(k), k = 0, \dots, N-1$  sú vzorky mikrosegmentu pre násobené okienkom a  $W_N = e^{i\frac{2\pi}{N}}$ . K tejto transformácii sa dá inverzná napísať takto

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{i \frac{2\pi}{N} kn} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{kn}.$$

Kvôli vysokým výpočtovým nárokom pri výpočte  $N$  koeficientov DFT bola odvodená tzv. rýchla Fourierova transformácia (FFT), ktorá využíva periodicitu členu  $W_N^{-kn}$ . Jej základnou myšlienkou je previesť výpočet  $N$ -bodovej DFT na niekoľko výpočtov  $N_1$  bodových DFT, kde  $N_1 \ll N$ . Metódou decimovania vo frekvencii prevod vyzerá nasledovne

$$X(2k) = \sum_{n=0}^{N/2-1} [x(n) + x(n + N/2)] W_{N/2}^{-nk}$$
$$X(2k + 1) = \sum_{n=0}^{N/2-1} [x(n) - x(n + N/2)] W_N^{-n} W_{N/2}^{-nk}$$

Z týchto výrazov už je zrejmé, že sa dá pôvodná  $N$ -bodová DFT previesť na dve  $N/2$ -bodové DFT. Ak je  $N$  mocnina dvoch, má FFT časovú zložitosť len  $\frac{1}{2} N \log_2 N$ . Výstupy DFT aj FFT sú vo všeobecnosti komplexné čísla vyjadrujúce intenzitu a fázu. Pre väčšinu aplikácií je zaujímavá hlavne intenzita, a preto sa ešte prevádzajú na reálne čísla vyjadrujúce jej veľkosť. Vtedy sa na tieto výstupy dá pozerat' ako na výstupy  $N$  pravidelne rozmiestnených pásmových filtrov.

### 3.1.2 Kepstrálna analýza

Kepstrálna analýza je veľmi vhodná na rozpoznávanie reči, pretože dobre vystihuje hlasový trakt. Je to transformácia zložená z Fourierovej transformácie, logaritmu z absolútnej hodnoty a inverznej Fourierovej transformácie, teda tzv. kestrum  $c(n)$  je možné vypočítať nasledovne

$$c(n) = \frac{1}{N} \sum_{k=0}^{N-1} \log |X(k)| W_N^{kn},$$

kde  $X(k)$  je výstup z Fourierovej transformácie a  $W_N$  je premenná definovaná v predošlej časti.

### 3.1.3 Lineárna prediktívna analýza

Lineárne prediktívne kódovanie (LPC) je jednou z najefektívnejších metód analýzy akustického signálu, pretože má schopnosť zabezpečiť veľmi presné odhady

## Predspracovanie signálu

---

parametrov modelu vytvárania reči priamo z rečového signálu. Nech  $s(k)$  je signál získaný PCM, potom tento signál je možné popísať lineárnou kombináciou  $Q$  predošlých vzoriek

$$s(k) = -\sum_{i=1}^Q a_i s(k-i),$$

kde  $Q$  je rád prediktora. Chybu prediktora  $E_n$  môžeme vyjadriť nasledovným spôsobom

$$E_n = R_n(0) + \sum_{i=1}^Q a_i R_n(i),$$

kde  $R_n(i)$  je autokorelačná funkcia. Koeficienty  $a_i$  je možné vypočítať Levinsonovým algoritmom, ktorý neskôr modifikoval Durbin na tvar, kde je riešenie vyjadrené rekurzívne

$$\begin{aligned} E_n^0 &= R_n(0) \\ k_i &= -\frac{R_n(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R_n(i-j)}{E_n^{(i-1)}} \\ a_i^{(i)} &= k_i \\ a_j^i &= a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}, 1 \leq j \leq i-1 \\ E_n^{(i)} &= (1 - k_i^2) E_n^{(i-1)}, \end{aligned}$$

kde  $a_j^{(i)}$  je  $j$ -ty koeficient prediktora rádu  $i$ . Niektoré implementácie používajú miesto koeficientov  $a_j^{(i)}$  medzivýsledky  $k_i$  známe pod označením PARCOR<sup>2</sup>, pretože majú vlastnosť  $k_i \in \langle -1, 1 \rangle$ .

### 3.1.4 Kepstrálna LPC analýza

Ďalšími koeficientami, ktoré veľmi dobre modelujú hlasový trakt, sú kepstrálne koeficienty LPC. Hodnotami sú podobné, ako kepstrálne koeficienty, ale nie totožné, pretože ich výpočet sa vzťahuje na spektrálnu obálku odvodenú analýzou LPC. Aby kepstrálne koeficienty LPC  $c(k)$  dobre reprezentovali spektrálnu obálku analyzovaného mikrosegmentu, je ich treba vyčísl'ovať vždy  $Q^* \geq Q$ , kde  $Q$  je rád

---

<sup>2</sup> alebo odrazové koeficienty (reflection coefficients)

prediktora LPC. Nech  $a_0 = 1$ , potom vzťahy na výpočet kepstrálnych koeficientov LPC vyzerajú takto

$$\begin{aligned}c(1) &= -a_1 \\c(k) &= -a_k - \sum_{i=1}^{k-1} \frac{i}{k} c(i) a_{k-i}, \text{ pre } 2 \leq k \leq Q \\c(k) &= -\sum_{i=1}^Q \frac{k-i}{k} c(k-i) a_i, \text{ pre } Q+1 \leq k \leq Q^*\end{aligned}$$

Tieto koeficienty sú asi najpoužívanejšie aj pre ich ľahké porovnávanie, ktoré si ukážeme v nasledujúcej kapitole.

### 3.2 Miery odlišnosti (metriky)

V rozpoznávaní reči je jedna z kľúčových rolí otázka odlišnosti medzi dvoma mikrosegmentami. Keď označíme vektory príznakov  $x = (x_1, \dots, x_Q)$  a  $y = (y_1, \dots, y_Q)$ , môžeme mieru odlišnosti označiť  $d(x, y)$ . Intuitívne je rozumné predpokladať, aby to bola metrika. Nie je to však podmienka.

Prvú mieru odlišnosti si predstavíme klasickú Euklidovskú metriku

$$d_E = \sqrt{\sum_{i=1}^Q (x_i - y_i)^2},$$

ktorá sa používa prioritne na porovnávanie výstupov zo signálových filtrov. Pre jej výpočtovú náročnosť sa však používa metrika

$$d_1 = \sum_{i=1}^Q |x_i - y_i|.$$

Na kepstrálnu analýzu bola v [PSU] odvodená kepstrálna miera odlišnosti

$$d_{CEP}(x, y) = \sum_{i=1}^Q (x_i - y_i)^2.$$

Zo štatistických vlastností kepstrálnych koeficientov sa však ukázalo, že je do výpočtu miery odlišnosti účelné zahrnúť jednotlivé koeficienty s rôznou váhou. To umožňuje tzv. Mahalanobisova miera

$$d_{MCEP}(x, y) = (x - y)^T V^{-1} (x - y),$$

## Predspracovanie signálu

---

kde  $V$  je kovariančná matica určená pomocou kepstrálnych koeficientov. Kvôli veľkému počtu operácií nutných na vyčíslenie tejto miery sa často používa jej modifikácia

$$d_{WCEP}(x, y) = \sum_{i=1}^Q w(i)(x_i - y_i)^2,$$

kde  $w(i)$  je inverzia  $i$ -teho diagonálneho prvku kovariančnej matice  $V$ . Poslednou mierou, ktorú si popíšeme je “uhlová” miera

$$d_A(x, y) = \arccos\left(\frac{x \cdot y}{|x||y|}\right),$$

ktorá však nespĺňa 2. podmienku metriky. Túto mieru sme navrhli špeciálne pre výstupy z pásmových filtrov a DTW. K jej vzniku viedla predstava, že pomery výstupov pásmových filtrov pri vyslovovaní zvuku rôznymi intenzitami sú konštantné alebo veľmi podobné. Preto na rozlíšenie dvoch rôznych zvukov je podstatné pozeráť sa na uhol medzi nimi v  $Q$ -rozmernom priestore.

### 3.3 Úprava obrazov slov

Takmer pri všetkých klasifikátoroch je vhodné použiť predspracovanie celých postupností príznakových vektorov - obrazov slov. Je to vhodné hlavne kvôli tomu, že nie vždy sa podarí určiť správne hraničné body slova. V publikácii Kuhna a Tomaschewského [KUT] bol navrhnutý nasledujúci postup, ktorý je vhodný hlavne na príznaky získané z pásmových filtrov. Pri jeho popise budeme predpokladať, že máme spracovať obraz slova  $A^{(0)} = (A_1, \dots, A_I)$ , kde  $A_i = (a_{i1}, \dots, a_{iQ})$ . Postup je nasledovný:

1. V tomto kroku pripočítame jednotkový šum ku každému mikrosegmentu obrazu slova

$$A^{(1)} = (A_1^{(0)} + u, \dots, A_I^{(0)} + u),$$

kde  $u = (1, \dots, 1)$ . Tento krok je nutný, pretože sa môže stať, že signál je veľmi čistý a niektorý komponent môže byť nulový, čo by spôsobovalo problémy v ďalších krokoch a v niektorých mierach skreslenia.

2. Tu pridáme jednotkový šum na začiatok a koniec slova. Matematicky sa dá vyjadriť takto

$$A^{(2)} = (u, A_1^{(1)}, \dots, A_I^{(1)}, u).$$

## Predspracovanie signálu

---

Tento krok spôsobí to, že aj keď hranice slova nie sú dobre zachytené a je zachytený až mikrosegment s výrazným spektrom, je schopný algoritmus spektrálnej stopy (popísaný ďalej) tieto stratené mikrosegmenty približne interpolovať v úseku ticha a prvým zachyteným mikrosegmentom (príp. posledným zachyteným mikrosegmentom a tichom). V prípade správneho zachytenia hraničných bodov má tento krok len minimálny vplyv, pretože sa slovo začína a končí tichom.

3. Všetky nasledujúce kroky vrátane tohoto sú nepovinné. Tento krok spôsobuje vyhladenie dlhodobého spektra a jeho normalizáciu. Má priaznivé dôsledky na presnosť rozpoznávania hlavne v systémoch s viacerými rečníkmi, alebo pri malej zmene hlasu. Najprv je treba určiť stredné hodnoty jednotlivých príznakov

$$\bar{a}_q^{(2)} = \frac{1}{I} \sum_{i=1}^I a_{iq}^{(2)},$$

kde  $q = 1, \dots, Q$ , a potom spraviť prípadné vyhladenie konvolučným jadrom  $K = (1, 2, 4, 2, 1)$

$$\tilde{a}_q^{(2)} = \sum_{i=-2}^2 \bar{a}_{q+i}^{(2)} K(i+3),$$

kde  $\bar{a}_{-1}^{(2)} = \bar{a}_0^{(2)} = \bar{a}_{Q+1}^{(2)} = \bar{a}_{Q+2}^{(2)} = 0$  a  $q = 1, \dots, Q$ . Normalizáciu spektra urobíme potom podľa vzťahu

$$a_{iq}^{(3)} = \frac{a_{iq}^{(2)}}{\bar{a}_q}, \text{ resp. } \tilde{a}_{iq}^{(3)} = \frac{a_{iq}^{(2)}}{\tilde{a}_q},$$

kde  $i = 1, \dots, I$  a  $q = 1, \dots, Q$ .

4. V tomto kroku spravíme amplitúdovú normalizáciu vzhľadom na priemernú amplitúdu v mikrosegmente, čo má za následok, že všetky mikrosegmenty budú mať pri klasifikácii rovnakú váhu. Transformácia je nasledovná

$$A_i^{(4)} = \frac{A_i^{(3)}}{\sum_{q=1}^Q a_{iq}^{(3)}},$$

kde  $i = 1, \dots, I$ .

## Predspracovanie signálu

---

5. Posledný krok. V ňom vyhladáme amplitúdové obálky jednotlivých pásiem vzhľadom na čas, čo odstráni skreslenia, ktoré sa môžu objaviť po aplikácii 3. kroku. Využijeme rovnaké konvolučné jadro ako v 3. kroku

$$a_{iq}^{(5)} = \sum_{j=-2}^2 a_{(i-j)q}^{(4)},$$

pričom  $a_{(-1)q}^{(4)} = a_{0q}^{(4)} = a_{(I+1)q}^{(4)} = a_{(I+2)q}^{(4)} = 0$ ,  $i = 1, \dots, I$  a  $q = 1, \dots, Q$ .

### 3.4 Vektorová kvantizácia

Vektorová kvantizácia je proces aproximácie vektorov spojitých veličín na konečný počet vektorov číselných hodnôt. Pôvodne bola uplatňovaná pri kompresii dát a pri minimalizácii prenosovej rýchlosti vokodérov, no dnes sa využíva aj v systémoch rozpoznávania reči a to hlavne pri modelovaní slov Markovovskými reťazcami (viď ďalej). Keď máme množinu vektorov ktoré chceme kvantovať, problém spočíva v tom, aké vektory je vhodné vybrať (pri ich známom počte), aby čo najlepšie reprezentovali všetky tieto vektory. Na riešenie tohoto problému je ľahká odpoveď - vyskúšajme všetky možnosti a ukáže sa, ktorá je najlepšia. Pri takomto prístupe však narážame na neprekonateľný výpočtový problém. Preto vzniká otázka ako to urobiť efektívne aj čiastočne na úkor výsledku. V roku 1967 navrhol MacQueen algoritmus, ktorý rieši tento problém elegantne a jednoducho. Nehľadá síce najlepšie riešenie, ale pokúša sa hľadať riešenie veľmi blízke optimálnemu v relatívne krátkom čase.

Formálne nech  $T$  je tréningová množina - množina vektorov, ktorú by sme chceli čo najlepšie aproximovať  $L$  výslednými vektormi. Chceli by sme  $T$  rozdeliť na podmnožiny alebo zhluky  $T_l, l = 1, \dots, L$  tak, aby bola minimalizovaná kritériálna funkcia  $J$ , ktorá má predpis

$$J = \sum_{l=1}^L J_l = \sum_{l=1}^L \sum_{x \in T_l} d(x, v_l), \text{ kde } v_l = \frac{1}{|T_l|} \sum_{x \in T_l} x,$$

pre danú mieru odlišnosti (príp. metriku)  $d$ . Označme

$T_i^k$  - množinu vektorov  $i$ -teho zhluku v  $k$ -tom kroku algoritmu,

$v_i^k$  - centroid  $i$ -teho zhluku v  $k$ -tom kroku algoritmu,

$J_i^k$  - dielčiu hodnotu kritéria  $i$ -teho zhluku v  $k$ -tom kroku algoritmu,

$n_i^k$  - počet vektorov v zhluku  $T_i$  v  $k$ -tom kroku algoritmu.

## Predspracovanie signálu

---

Potom MacQueenov zhlukovací algoritmus<sup>3</sup> vyzerá nasledovne:

1. Náhodne alebo cielene vyberieme  $L$  počiatočných centroidov  $v_i^{(1)}$

2. Nech sme v kroku  $k$ . Vytvoríme zhluky  $T_i^k$  takto

$$x \in T_i^k \Leftrightarrow d(x, v_i^k) < d(x, v_j^k), \forall i, j = 1, \dots, L, i \neq j$$

3. Vypočítame nové centroidy  $v_i^{k+1}$  tak, aby boli minimálne dielčie hodnoty kritérií

$$J_i^{k+1} = \sum_{x \in T_i^k} d(x, v_i^{k+1}), i = 1, \dots, L,$$

čo možno urobiť nasledovne

$$v_i^{k+1} = \frac{1}{n_j^k} \sum_{x \in T_i^k} x, i = 1, \dots, L.$$

4. Ak je pokles celkového kritéria (celkovej chyby)

$$J^{k+1} = \sum_{i=1}^L J_i^{k+1}$$

pod zadaným prahom, tak skončíme, inak pokračujeme bodom 2.

Výstupom algoritmu sú centroidy z posledného kroku, ktoré vytvárajú nový rozklad celého vektorového priestoru. Kvantizácia potom prebieha tak, že kvantovanému vektoru priradíme ten centroid, ku ktorému je najbližšie berúc do úvahy mieru odlišnosti  $d$ . Ešte treba spomenúť, že tento proces hľadá len lokálne minimum funkcie  $J$ , a preto je vhodné ho spustiť niekoľkokrát pre lepšiu šancu nájdenia globálneho minima.

---

<sup>3</sup> v anglickej literatúre známy ako K-means

# Kapitola 4

## Tri teoretické modely klasifikátorov

Porozumieť reči znamená spracovať signál, zistiť, či sa niečo hovorí, zistiť, čo sa asi hovorí, a potom výsledok opraviť analyzátormi na vyšších úrovniach - syntaktickými sémantickými a inými. Na zisťovanie čo sa hovorí sú známe dva prístupy: rozpoznať celé slová naraz a rozpoznať najprv menšie časti (napr. fonémy) a podľa slovníka ich pospájať do slov. Súčasný výskum je sústredený na druhý prístup, pretože je novší a umožňuje kvalitnejšie rozpoznávanie plynulej reči. Prvý prístup je celkom dobre preskúmaný. Momentálne sa mu už natoľko nevenuje pozornosť, ale využíva sa v komerčnej sfére. Napriek tomu nás zaujíma, pretože ho využijeme na porovnanie modelov klasifikátorov. Všetky tri modely klasifikátorov sú dobre preskúmané z teoretického aj praktického hľadiska.

- Klasifikácia izolovaných slov pomocou dynamického krivenia časovej<sup>4</sup> osi (Dynamic Time Warping - DTW) bola po prvý raz publikovaná už v roku 1969. Je vyčerpávajúco popísaná v [PSU].
- Neurónové siete (Neural Networks - NN) sú ďalším modelom klasifikátora. I keď ich výskum sa začal už v 50-tych rokoch, ich využitie v klasifikovaní slov nie je rozšírené. Vo väčšine prípadov sa používajú na predspracovanie dát a zmenšenie rozmeru príznakových vektorov, príp. na analýzy na vyšších úrovniach. Sú dobre popísané v [ÚNS] a detailne v [HRI].
- Tretím modelom klasifikátora je model založený na využití pravdepodobnostného prístupu skrytých Markovovských modelov (Hidden Markov Models - HMM). Po prvý raz bol predstavený v polovici 70-tych rokov. Je najnovší a v súčasnosti už úplne vytlačil DTW. Používal ho napr. systém TANGORA od IBM a SPHINX z CMU. Veľmi dobre je rozobraný v [RJI] a [RHF] a venujú sa mu v [PSU] a [PIC], tréning je zhrnuté v [SUN].

V tejto kapitole si stručne zhrnieme teóriu všetkých troch spomenutých prístupov rozpoznávania izolovaných slov, ktorú sme použili pri implementácii programov a následne v experimentoch popísaných v ďalších kapitolách.

---

<sup>4</sup> poprípade len dynamického programovania

### 4.1 Dynamické krivenie časovej osi (DTW)

Táto metóda je založená na porovnávaní vzoriek príznakov tak, že ich vnútorne naťahuje (borí jednu z časových osí) a hľadá najlepšiu zhodu k vzorovej vzorke vzhľadom na nejakú metriku, ktorá závisí od metriky určujúcej vzdialenosť dvoch vektorov príznakov. Rozhodnutie o zaradení neznámeho slova do určitej triedy potom spraví na základe minimálnej vzdialenosti obrazu neznámeho slova od niektorého referenčného obrazu tejto triedy. Názov dynamické naťahovanie v čase teda nevznikol od kritéria klasifikácie, ale je odvodený z mechanizmu určovania vzdialeností dvoch obrazov slov. Vnútorne naťahovanie vyplynulo z dôkladného rozboru signálu získaného niekoľkonásobným vyslovením rovnakého slova jedným rečníkom. V tomto rozbere sa ukázalo, že hlavné rozdiely medzi týmito signálmi neboli v spektrálnej oblasti, ale v časovom členení, teda v rôznej dĺžke vyslovovania slova a jeho vnútorných častí (foném, hlások). Táto metóda sa stala obľúbenou, pretože je atraktívna jej jednoduchosťou.

#### 4.1.1 Klasifikácia

Majme postupnosti príznakových vektorov  $A = \{a_1 \dots a_I\}$ ,  $B = \{b_1 \dots b_J\}$ , kde  $A$  je referenčná postupnosť a  $B$  je testovaná postupnosť, a nejakú metriku  $d(u, v)$  určujúcu vzdialenosť príznakových vektorov. Pomocou tejto metriky sa budeme snažiť vytvoriť metriku  $D(A, B)$  celých obrazov slov (postupností vektorov príznakov). Metriku  $D(A, B)$  spravíme tak, že postupne vyskúšame všetky (obmedzené) možnosti naťahovania vstupnej postupnosti aj referenčnej postupnosti a zakaždým vypočítame ich vzdialenosť, pričom najmenšiu hodnotu vrátíme ako výsledok. To zabezpečuje funkcia DTW, ktorú si označme  $g(i, j)$ , kde  $1 \leq i \leq I, 1 \leq j \leq J$ . Táto funkcia počíta najlepšiu zhodu (natiehnutie) postupností až po  $i$ -ty príznakový vektor v  $A$  a  $j$ -ty príznakový vektor v  $B$ . Keď už takúto funkciu máme, tak  $D(A, B)$  môžeme zapísať v tvare

$$D(A, B) = \frac{g(I, J)}{N}, \text{ kde } N \text{ je normalizačný faktor.}$$

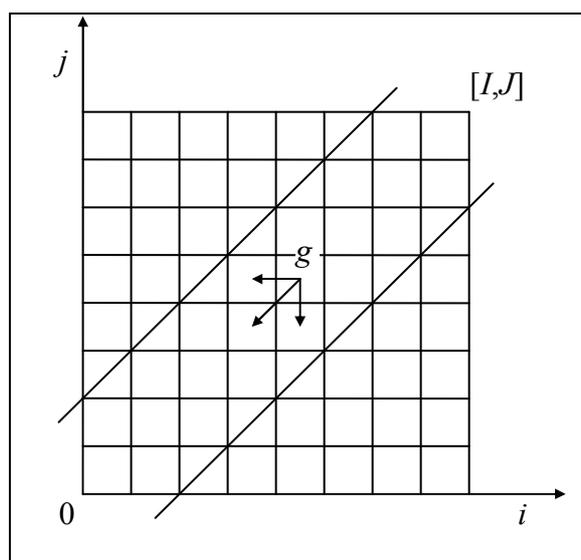
Existuje veľa predpisov pre  $g(i, j)$  a  $N$ , v [PSU] je ich niekoľko uvedených, avšak pre naše účely úplne postačí  $g_1$  a  $N_1$ , nakoľko sme pri pokusoch s ostatnými žiadne rozdiely nepostrehli:

$$g_1(1, 1) = d(a_1, b_1)$$
$$g_1(i, j) = \min\{g_1(i-1, j-1) + 2d(a_i, b_j), g_1(i-1, j) + d(a_i, b_j), g_1(i, j-1) + d(a_i, b_j)\}$$

## Tri teoretické modely klasifikátorov

$$N_1 = I + J$$

Hraničné prípady si už ľahko domysliť - stačí vynechať výrazy so zlými indexmi. Výpočet tejto funkcie sa zvykne robiť dynamickým programovaním, ktorého výsledkom je tabuľka podobná nasledujúcemu obrázku (Obrázok 1).



**Obrázok 1** - Náčrt prehľadávania priestoru možných natiahnutí pri DTW. Šikmé priamky znázorňujú ohraničenie šírky prehľadávaného priestoru.

Kvôli pamäťovej a časovej náročnosti použijeme preškáľovanie slovníka pomocou počítania spektrálnej stopy tak, ako ju popísal Psutka v [PSU]. Tým odstránime aj potrebu normalizačného faktora.

Samotná klasifikácia je vcelku jednoduchá - použijeme pravidlo k-NN. Majme slovník tvaru  $S = \{A_1 \dots A_R\}$ , kde  $A_r$  sú triedy vzorových postupností  $A_{r1}, \dots, A_{rS_r}$  pre  $r$ -tému slovo a  $B$  testovanú postupnosť, ktorú chceme klasifikovať. Vypočítame všetky  $D(A_{rs}, B)$  a usporiadame ich pre každé  $r$  tak, aby platilo

$$D(A_{r1}, B) \leq D(A_{r2}, B) \leq \dots \leq D(A_{rS_r}, B)$$

O klasifikácii obrazu  $B$  neznámeho slova do triedy  $A_r^*$  potom rozhodneme podľa minima priemernej vzdialenosti  $k$  najbližších susedov

$$A_r^* = \arg \min_r \left\{ \frac{1}{k} \sum_{i=1}^k D(A_{ri}, B) \right\},$$

kde  $r = 1, \dots, R$ .

### 4.1.2 Trénovanie

Trénovanie tohto modelu je jednoduché - stačí vytvoriť slovník tvaru  $S = \{A_1 \dots A_R\}$ , kde  $A_r$  sú triedy vzorových postupností  $A_{r,1}, \dots, A_{r,S_r}$  pre  $r$ -té slovo. My však do  $A_r$  dáme predspracované slová rovnakej dĺžky. To dosiahneme buď natiahnutím, čím si príznakové vektory udržia časovú ekvidistantnosť, alebo tzv. segmentáciou spektrálnej stopy, ktorá túto vlastnosť síce nemá, ale lepšie pokrýje zmeny v príznakových vektoroch. Obidve možnosti sú popísané v [PSU]. My uprednostníme druhú možnosť.

Predpokladajme, že predspracovaním slova bol získaný jeho obraz  $A$  ako postupnosť  $Q$  dimenzionálnych časovo ekvidistantných vektorov príznakov  $a_i$

$$A = \{a_1, \dots, a_i, \dots, a_I\}.$$

Spektrálna stopa je spojnica  $I$  koncových bodov vektorov príznakov  $a_i$  a algoritmus segmentácie spektrálnej stopy nahrádza vzorkovanie v časovej oblasti vzorkovaním v  $Q$  dimenzionálnom priestore príznakov, t.j. výsledné za sebou idúce vektory príznakov majú rovnakú vzdialenosť v danej metrike. Pretože nepoznáme skutočný priebeh spojitú spektrálnu stopu, budeme ju lineárne aproximovať medzi jednotlivými bodmi. Princíp segmentácie potom spočíva v rozdelení tejto interpolovanej stopy do  $I^*$  segmentov a určení  $I^* + 1$  nových príznakových vektorov  $a_i^*$ , ktorých postupnosť bude novým obrazom slova.

Najprv bude treba určiť celkovú dĺžku pôvodnej stopy  $L$

$$L = \sum_{i=1}^{I-1} d(a_i, a_{i+1}),$$

potom zvolíť nový počet segmentov  $I^*$  a určiť dĺžku nového segmentu

$$M = \frac{L}{I^*}.$$

Položme  $a_1^* = a_1$ ,  $j^1 = 2$ ,  $a_{I^*+1}^* = a_I$ . Teraz určíme  $a_i^*, 2 \leq i \leq I^*$ . K tomu potrebujeme určiť také  $j^i$ , že

$$\begin{aligned} d(a_{i-1}^*, a_{j^{i-1}}) + d(a_{j^{i-1}}, a_{j^{i-1}+1}) + \dots + d(a_{j^{i-1}-1}, a_{j^i}) &\geq M \\ d(a_{i-1}^*, a_{j^{i-1}}) + d(a_{j^{i-1}}, a_{j^{i-1}+1}) + \dots + d(a_{j^{i-2}-2}, a_{j^{i-1}}) &< M, \end{aligned}$$

## Tri teoretické modely klasifikátorov

---

teda segment  $[a_{j^{i-1}}, a_{j^i}]$ , v ktorom sa bude nachádzať  $a_i^*$ . Vzdialenosť  $a_i^*$  od  $a_{j^{i-1}}$  v tomto segmente si označme

$$M^* = d(a_{j^{i-1}}, a_i^*) = M - d(a_{i-1}^*, a_{j^{i-1}}) - d(a_{j^{i-1}}, a_{j^{i-1}+1}) - \dots - d(a_{j^{i-2}}, a_{j^{i-1}})$$

Potom súradnice bodu  $a_i^*$  vypočítame zo vzťahu

$$a_i^* = a_{j^{i-1}} + (a_{j^i} - a_{j^{i-1}}) \frac{M^*}{d(a_{j^i}, a_{j^{i-1}})}$$

Výsledná postupnosť príznakových vektorov je  $A^* = \{a_1^*, \dots, a_{I^*+1}^*\}$ . Pri vhodnom zvolení  $I^*$  nedochádza k podstatnej strate informácií, aj keď  $I^* \ll I$ .

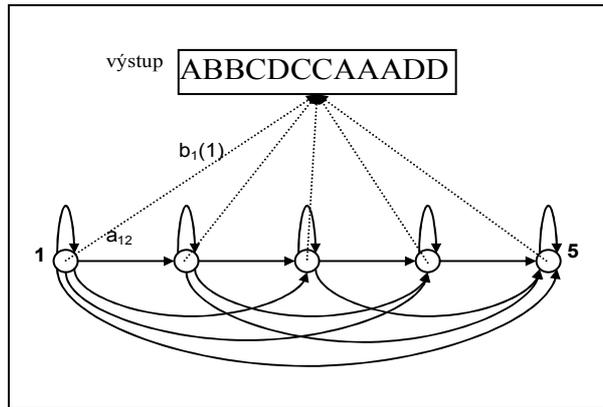
## 4.2 Skryté Markovovské modely (HMM)

Používanie tohoto prístupu je motivované tým, že priebeh reči sa dá dobre modelovať ako náhodný parametrický proces meniaci sa v čase a na to sú vhodné dopredné HMM, ktoré sú založené na štatistike a Bayesovskej pravdepodobnosti. Známe sú dve triedy HMM:

- generujúce symboly v stavoch
- generujúce symboly pri prechode po hranách

My sa budeme zaoberať len prvou triedou, tá nám na klasifikáciu izolovaných slov postačí. Čo je to Markovovský model? Je to automat, ktorý prechádza do iných stavov s istou pravdepodobnosťou. Pri prechode do tohto stavu generuje náhodne s určitou pravdepodobnosťou symbol na výstup (Obrázok 2). Ak nie je známa postupnosť stavov, ktorá vygenerovala danú postupnosť symbolov, hovoríme o skrytom Markovovskom modeli - HMM. Ako to súvisí s rozpoznávaním reči? Keď sa takýto model natrénuje, dá sa ho "spýtať", s akou pravdepodobnosťou by generoval zadanú postupnosť. Pri popise HMM budeme vychádzať z [RJF] a zameriame sa len na také HMM generujúce symboly v stavoch, ktoré majú konečnú množinu možných vygenerovaných symbolov a teda diskrétnu rozdelenie pravdepodobnosti ich generovania.

## Tri teoretické modely klasifikátorov



Obrázok 2 - Náčrt dopredného Markovovského modelu.

Aby sme si HMM priblížili viac, musíme zaviesť zopár označení:

$\{1..N\}$  - stavy modelu

$q_t$  - stav modelu v čase  $t$

$\{v_1..v_M\}$  - množinu výstupných symbolov

$A = \{a_{ij}\}, a_{ij} = P(q_{t+1} = j | q_t = i), 1 \leq i, j \leq N$  - maticu pravdepodobnosti prechodov zo stavu  $i$  do stavu  $j$

$B = \{b_j(k)\}, b_j(k) = P(o_t = v_k | q_t = j)$  - maticu pravdepodobnosti generovania symbolov v stave  $j$ , kde  $o_t$  je výstupný symbol v čase  $t$ .

HMM je v každom čase  $t$  v stave  $q_t$  a v každom kroku prechádza náhodne do stavu  $q_{t+1}$  s pravdepodobnosťou  $a_{q_t, q_{t+1}}$ , kedy vygeneruje náhodný symbol  $v_k$  s pravdepodobnosťou  $b_{q_{t+1}}(k)$ . Takto vygeneruje náhodnú postupnosť symbolov s určitou pravdepodobnosťou. Na druhej strane, vieme aj zistiť, s akou pravdepodobnosťou by vygeneroval istú postupnosť symbolov. To je presne tá vlastnosť, ktorú potrebujeme a využijeme. Nás ďalej budú zaujímať len tie HMM, ktoré sú progresívne (dopredné), teda také, ktoré sa nemôžu vracat' do stavov, v ktorých už boli (s výnimkou toho, v ktorom sa nachádzajú), teda  $a_{ij} = 0, \forall i > j$ .

Ďalej označme:

$\pi = \{\pi_i\}, \pi_i = P(q_1 = i), 1 \leq i \leq N$  - vektor pravdepodobnosti počiatočného stavu

$\lambda = (A, B, \pi)$  - parametre, ktoré treba naučiť

Pretože nás zaujímajú len dopredné HMM, je  $\pi_1 = 1$  a  $\pi_i = 0, i \neq 1$ , nakoľko chceme, aby model začínal vždy v prvom stave a časom prechádzal do stavov s vyšším indexom, presne tak, ako sa vyslovuje slovo - nikdy nezačína v strede. To, aké

postupnosti symbolov HMM generuje s akými pravdepodobnosťami, je určené  $\lambda$ . Ako hľadať tieto parametre si ukážeme v jednej z nasledujúcich podkapitol.

### 4.2.1 Zisťovanie pravdepodobnosti generovania postupnosti

Majme HMM s parametrami  $\lambda$  a postupnosť príznakových vektorov  $O = \{o_1, \dots, o_T\}$ . Chceme zistiť  $P(O | \lambda)$ , teda s akou pravdepodobnosťou by HMM generoval  $O$ . Táto pravdepodobnosť sa dá určiť dvoma základnými spôsobmi - doprednou procedúrou a spätnou procedúrou. Popíšeme si obe, lebo ich budeme potrebovať.

Dopredná procedúra je rekurzívny proces, v ktorom sa postupne zisťujú pravdepodobnosti  $\alpha_t(i)$  definované ako  $\alpha_t(i) = P((o_1, \dots, o_t, q_t = i) | \lambda)$ , teda pravdepodobnosti, že bola pozorovaná postupnosť  $o_1..o_t$  a model skončil v stave  $q_t = i$ . To sa dá vo všeobecnosti zapísať indukčným nasledujúcim spôsobom:

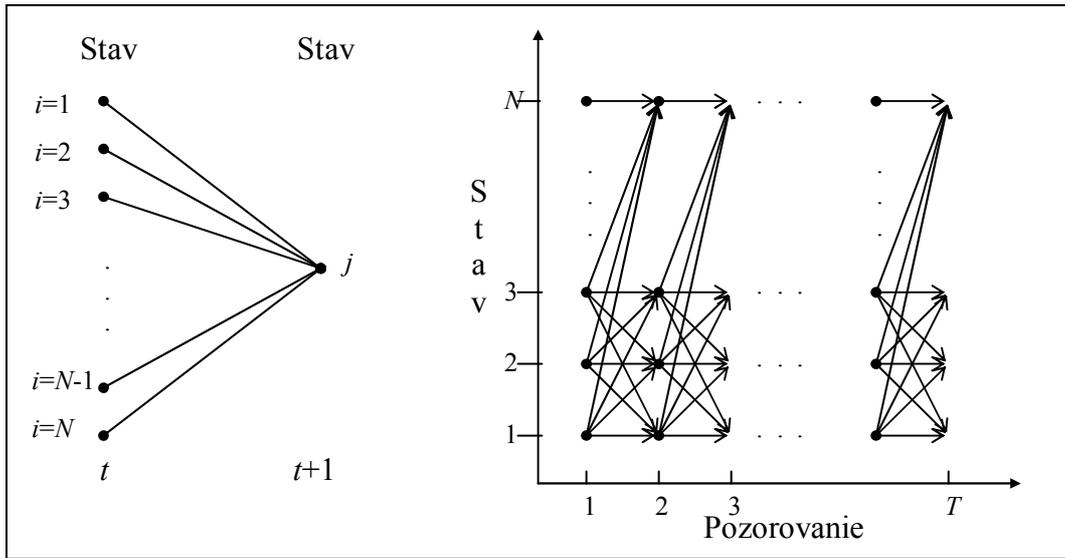
$$\begin{aligned}\alpha_1(i) &= \pi_i b_i(O_1), 1 \leq i \leq N \\ \alpha_{t+1}(j) &= \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), t = 0..T-1 \\ P(O | \lambda) &= \sum_{i=1}^N \alpha_T(i) ,\end{aligned}$$

keď môže model skončiť v ľubovoľnom stave, no pre naše účely sa viac hodí, aby model skončil v poslednom stave, preto

$$P(O | \lambda) = \alpha_T(N)$$

Rekurzívny výpočet doprednej premennej môžeme pozorovať na nasledovnom obrázku (Obrázok 3).

## Tri teoretické modely klasifikátorov



Obrázok 3 - Výpočet doprednej premennej.

Podobne môžeme odvodiť výpočet spätnej premennej  $\beta_t(i) = P(o_{t+1}, \dots, o_T | q_t = i, \lambda)$ , teda pravdepodobnosť toho, že bola pozorovaná postupnosť  $o_{t+1}, \dots, o_T$ , pričom model začína v stave  $i$ :

$\beta_T(i) = 1, 1 \leq i \leq N$  pre model, ktorý môže skončiť v ktoromkoľvek stave a

$\beta_T(i) = 0$ , ak  $i \neq N$ , inak 1, pre model, ktorý musí skončiť v stave  $N$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \text{ pre } t = T-1, T-2, \dots, 1, 1 \leq i \leq N$$

$$P(O | \lambda) = \sum_{i=1}^N \pi_i b_i(o_1) \beta_1(i)$$

Nás táto teória zaujíma z implementačného hľadiska. Pri výpočte doprednej aj spätnej premennej by sme už pri krátkych postupnostiach narazili na problém príliš malých čísel, ktoré počítač nie je schopný reprezentovať. Preto obe premenné znormalizujeme na  $\tilde{\alpha}_t(i)$  a  $\tilde{\beta}_t(i)$  nasledovným spôsobom:

Označme

$$\bar{\alpha}_t(i) = \sum_{j=1}^N \tilde{\alpha}_{t-1}(j) a_{ji} b_i(o_t), \text{ pre } 2 \leq t \leq T \text{ a } \bar{\alpha}_1(i) = \alpha_1(i)$$

$$c_t = \frac{1}{\sum_{i=1}^N \bar{\alpha}_t(i)}$$

## Tri teoretické modely klasifikátorov

---

$$\bar{\beta}_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \tilde{\beta}_{t+1}(j), \text{ pre } 1 \leq t \leq T-1 \text{ a } \bar{\beta}_T(i) = \beta_T(i)$$

Potom

$$\begin{aligned}\tilde{\alpha}_t(i) &= c_t \bar{\alpha}_t(i) \\ \tilde{\beta}_t(i) &= c_t \bar{\beta}_t(i)\end{aligned}$$

Pravdepodobnosť generovania postupnosti (pri skončení v ktoromkoľvek stave) sa dá vypočítať nasledovne:

$$P(O | \lambda) = \prod_{t=1}^T c_t, \quad \text{alebo lepšie} \quad \log(P(O | \lambda)) = \sum_{t=1}^T \log(c_t)$$

Pre naše účely potrebujeme, aby model skončil v poslednom stave:

$$\begin{aligned}P(O | \lambda) &= \tilde{\alpha}_T(N) \prod_{t=1}^{T-1} c_t, \quad \text{alebo lepšie} \\ \log(P(O | \lambda)) &= \log(\tilde{\alpha}_T(N)) + \sum_{t=1}^{T-1} \log(c_t)\end{aligned}$$

Posledný výraz je už dobre realizovateľný aj pri dlhších postupnostiach a preto postačí na naše účely, aj keď nie je optimalizovaný vzhľadom na výpočtovú cenu operácií.

Keď už máme natrénované HMM, pre každé slovo jeden, klasifikácia je nasledovná: vyberieme to slovo, ktorého model má najväčšiu pravdepodobnosť, že generoval danú postupnosť.

### 4.2.2 Trénovanie

Učenie je najzložitejšia a najdôležitejšia časť teórie HMM, pretože neexistuje priama metóda ako dospieť k najlepším parametrom. HMM sa dá učiť viacerými spôsobmi, ale najpoužívanejší je Baum-Welchov iteračný (reestimačný) proces. Aj keď nenachádza globálne najlepšie parametre, má dobré vlastnosti - po každej iterácii platí  $P(O | \hat{\lambda}) > P(O | \lambda)$ , až kým nedôjde do lokálneho minima, kde nastáva rovnosť. Tento algoritmus pritom hľadá lokálne minimum pomerne rýchlo.

## Tri teoretické modely klasifikátorov

---

Aby sme vysvetlili tento proces, potrebujeme zdefinovať  $\gamma_t(i) = P(q_t = i | O, \lambda)$ , pravdepodobnosť toho, že model je v stave  $i$  v čase  $t$ , keď je pozorovaná postupnosť  $O$  a má parametre  $\lambda$ . Vyjadriť ju môžeme takto:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} = \frac{\tilde{\alpha}_t(i)\tilde{\beta}_t(i)}{\sum_{j=1}^N \tilde{\alpha}_t(j)\tilde{\beta}_t(j)}$$

Platnosť poslednej rovnosti je zrejmá z toho, že normalizačné koeficienty môžeme ľahko vybrať pred sumu v menovateli a vykrátiť ich s normalizačnými koeficientami v čitateli. Ďalej potrebujeme definovať  $\xi_t(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda)$ , teda pravdepodobnosť toho, že model je v stave  $i$  v čase  $t$  a prejde do stavu  $j$  v čase  $t+1$  za predpokladu, že je pozorovaná postupnosť  $O$  a má parametre  $\lambda$ . S využitím predošlých výrazov môžeme napísať

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{k=1}^N \sum_{l=1}^N \alpha_t(k)a_{kl}b_l(o_{t+1})\beta_{t+1}(l)} = \frac{\tilde{\alpha}_t(i)a_{ij}b_j(o_{t+1})\tilde{\beta}_{t+1}(j)}{\sum_{k=1}^N \sum_{l=1}^N \tilde{\alpha}_t(k)a_{kl}b_l(o_{t+1})\tilde{\beta}_{t+1}(l)}$$

Vzťah medzi  $\gamma_t(i)$  a  $\xi_t(i, j)$  sa dá vyjadriť jednoducho

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j)$$

Teraz už máme všetko na to, aby sme mohli odvodiť reestimačné formuly. Nech máme  $S$  postupností označených  $O = [O^1, \dots, O^S]$ , pričom chceme maximalizovať

$$P(O | \lambda) = \prod_{s=1}^S P(O^s | \lambda).$$

Keď označíme dĺžky jednotlivých postupností  $T^1, \dots, T^S$ , tak reestimačné formuly môžeme napísať takto:

$$\hat{\pi}_1 = 1 \text{ a } \hat{\pi}_i = 0, 2 \leq i \leq N$$

$$\hat{a}_{ij} = \frac{\text{ocakavany počet priechodov zo stavu } i \text{ do stavu } j}{\text{ocakavany počet priechodov zo stavu } i} = \frac{\sum_{s=1}^S \sum_{t=1}^{T^s-1} \xi_t^s(i, j)}{\sum_{s=1}^S \sum_{t=1}^{T^s-1} \gamma_t^s(i)}$$

$$\hat{b}_j(k) = \frac{\text{ocakavany počet pozorovani symbolu } v_k \text{ v stave } j}{\text{ocakavany počet prichodov do stavu } j} = \frac{\sum_{s=1}^S \sum_{t=1, o_t^s=v_k}^{T^s} \gamma_t^s(j)}{\sum_{s=1}^S \sum_{t=1}^{T^s} \gamma_t^s(j)}$$

Tieto formuly platia tak pre nenormalizované, ako aj pre normalizované dopredné a spätné premenné, pretože sme ich použili na výpočet  $\gamma_t(i)$  a  $\xi_t(i, j)$  pre jednotlivé postupnosti a normalizačné koeficienty sa vykrátili už vo vnútri vnútorných súm.

Jedným z problémov, ktoré sme ešte nevyriešili, je problém konečnej trénovacej množiny, t.j. vynulovanie pravdepodobnosti  $b_j(k)$  pre taký stav  $j$ , v ktorom sa v žiadnej trénovacej postupnosti nevyskytuje symbol  $v_k$ . V praxi to spôsobí vynulovanie pravdepodobnosti každej postupnosti symbolov, ktorá je čo len trochu (v jedinom neštandardnom symbole) odlišná od trénovacích. Keďže ale v reči isté podobné chyby nastávajú, potrebujeme tomu predísť. Dobré riešenie je po každom kroku trénovania nahradiť tie  $b_j(k)$ , ktoré sú pod nejakou hranicou  $\kappa$ , práve touto hranicou a ostatné upraviť tak, aby bol súčet pravdepodobností generovania symbolov rovný 1. Označme  $i^*$  počet pravdepodobností generovania symbolov v stave  $j$  menších ako  $\kappa$ . Potom úprava vyzerá nasledovne

$$b_j^* = \sum_{k=1, b_j(k) \geq \kappa}^M b_j(k)$$

$$m_j = \frac{1 - i^* \kappa}{b_j^*}$$

$$\hat{b}_j(k) = \kappa, \text{ ak } b_j(k) < \kappa$$

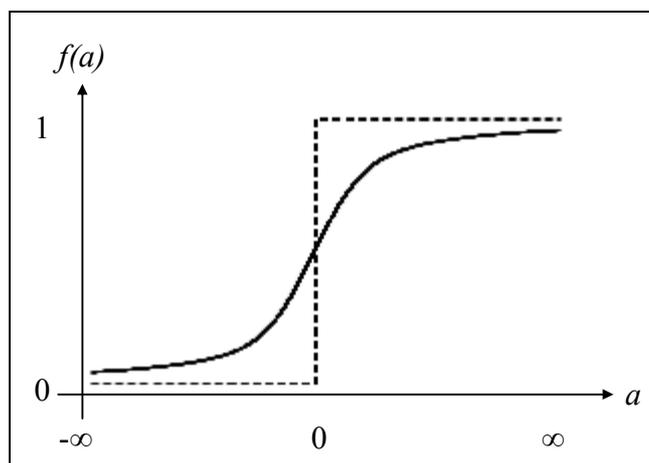
$$\hat{b}_j(k) = b_j(k) m_j \text{ inak}$$

Žiaľ takáto úprava má za následok to, že proces učenia stráca pozitívnu vlastnosť stáleho zlepšovania.

Trénovanie HMM je treba opakovať niekoľkokrát, aby sme mali väčšiu šancu priblížiť sa ku globálnemu minimu.

### 4.3 Neurónové siete (NN)

Použitie tohto prístupu je motivované tým, že NN sa dokážu naučiť akúkoľvek hladkú funkciu. Túto podmienku reč spĺňa, pretože sa mení spojitou a hladkou množinou priebehov slov jednotlivých rečníkov je tiež možné aproximovať spojitými a hladkými plochami. Ďalší dôvod je ten, že neurónové siete sa snažia byť modelom biologických neurónov a ich spojení a modelovať ich najdôležitejšiu schopnosť - učiť sa. Na druhej strane klasifikačná architektúra NN na rozpoznávanie reči bola často odmietaná a v niektorých publikáciách označovaná za nevhodnú [SUR], pretože dopredné NN nevedia zvládnuť dynamiku reči z pohľadu toho, že proces reči je závislý od času a slová môžu byť vyslovované rôznym tempom, a teda majú rôznu dĺžku. My skúsime tento problém prekonať natiahnutím vzoriek použitím segmentácie spektrálnej stopy.



Obrázok 4 - Náčrt priebehu sigmoidálnej aktivačnej funkcie.

Náš prístup a použitie NN sú založené na priamej klasifikácii postupností rovnakej dĺžky. Na vstup sa dajú paralelne všetky vektory príznakov a výstupov je toľko, koľko klasifikovaných tried. Budeme sa teda zaoberať len s doprednými (Feed Forward) 3-vrstvovými sieťami (spolu so vstupnými neurónmi) a konkrétnou architektúrou (klasifikačnou), kde na výstupe je práve jedna jednotka a ostatné nuly. Aby sme si bližšie popísali neurónovú sieť, musíme zaviesť niekoľko označení:

$L$  - počet vrstiev bez vstupnej vrstvy,  $l$  číslo vrstvy od 0 - vstupná vrstva

$N_l$  - počet neurónov v  $l$ -tej vrstve,

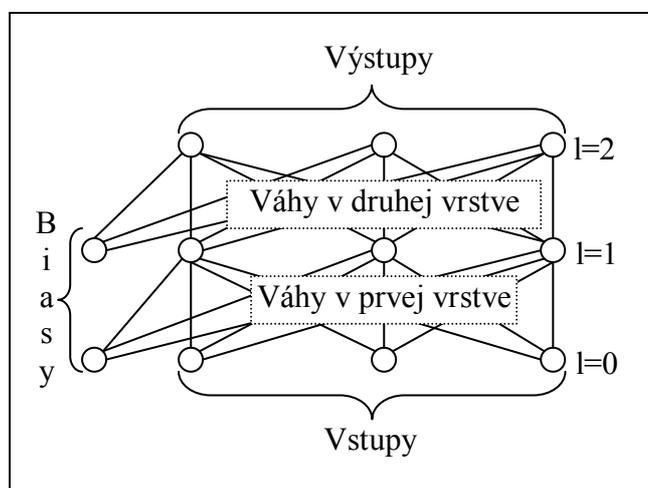
$w_{ij}$  - váhu vo vrstve  $l$  z neurónu  $i$  do neurónu  $j$

$a_{li}$  - výstupnú aktiváciu neurónu  $i$  vo vrstve  $l$

$f_l$  - aktivačnú funkciu vo vrstve  $l$  (Obrázok 4)

## Tri teoretické modely klasifikátorov

Pre lepšiu názornosť uvádzame náčrt 3-vrstvovej siete ( $L=2$ ) na nasledujúcom obrázku (Obrázok 5).



Obrázok 5 - Náčrt 3-vrstvovej neurónovej siete.

### 4.3.1 Klasifikácia

Klasifikácia spočíva v určení výstupného neurónu s najväčšou aktivitou. Najprv však musíme dať na vstup testovací vektor - teda všetky príznakové vektory testovacieho zvuku - paralelne na všetky vstupné neuróny. Vstupné neuróny sú reprezentované premennými  $a_{0i}$ , kde  $1 \leq i \leq N_0$ . Aktivitu neurónu vypočítame nasledovným vzťahom

$$a_{li} = f_l \left( \sum_{j=0}^{N_{l-1}} a_{(l-1)j} w_{lij} \right)$$

Aktivity propagujeme z vstupnej vrstvy až na výstupnú. Slovo asociované s neurónom, ktorý má najväčšiu aktivitu je výsledok  $v$ :

$$v = \arg \max_{j=1 \dots N_L} \{a_{Lj}\}$$

### 4.3.2 Trénovanie

Trénovanie neurónovej siete spočíva v nastavení jej váh tak, aby sa na testovacích vektoroch správala potrebným spôsobom. Na trénovanie sa používajú gradientové

## Tri teoretické modely klasifikátorov

---

metódy, nakoľko neexistuje metóda, ktorá by určila váhy priamo, pokiaľ sú aktivačné funkcie iné ako lineárne. Gradientové metódy hľadajú globálne minimum zadanej chybovej funkcie iteratívnym prehľadávaním chybovej plochy vždy smerom k menšej energetickej hladine.

V literatúre [HRI] možno nájsť viacero chybových funkcií, no nás bude zaujímať jednoduchá sum-of-squares chybová funkcia

$$E_{(x,t)}(W) = \frac{1}{2} \sum_{i=1}^{N_L} (a_{Li}(x) - t_i)^2,$$

kde  $x = \{x_1, \dots, x_{N_0}\}$  je vstupný vektor, z ktorého boli počítané aktivity a  $t = \{t_1, \dots, t_{N_L}\}$  je trénovací vektor. Trénovaciu množinu si označme ako

$$T_{Set} = \{(x^1, t^1), \dots, (x^P, t^P)\},$$

kde  $x^p$  sú vstupné vektory,  $t^p$  sú trénovacie výstupné vektory a  $P$  je ich počet. Celková chybová funkcia vyzerá potom nasledovne

$$E_{tot}(W) = \sum_{p=1}^P E_{T_{Set}(p)}(W),$$

kde  $W = \{w_{ij}\}$ . Na to, aby sme mohli prejsť k samotnému učeniu, potrebujeme ešte vedieť počítať gradient chybovej funkcie  $\nabla E = \left\{ \frac{\partial E}{\partial w_{ij}} \right\}$ . Štandardný prístup

$$\{\nabla E\}_{ij} \approx \frac{\Delta E}{\Delta w_{ij}},$$

pre malé  $\Delta w_{ij}$  by mal časovú zložitosť  $O(N_w^2)$ , pretože každý výpočet chyby má časovú zložitosť  $O(N_w)$  a museli by sme ju počítať  $N_w$ -krát. Naproti tomu algoritmus Backpropagation (algoritmus spätného šírenia chyby) na výpočet gradientu redukuje časovú zložitosť na  $O(N_w)$  a preto je často používaný.

Nech funkcia  $g$  má nasledujúcu vlastnosť

$$\forall x : g(f(x)) = f'(x),$$

potom môžeme parciálnu deriváciu  $\frac{\partial E}{\partial w_{ij}}$  vyjadriť rekurentným vzťahom

## Tri teoretické modely klasifikátorov

$$\frac{\partial E}{\partial w_{lij}} = \frac{\partial E}{\partial a_{li}} \frac{\partial a_{li}}{\partial w_{lij}}$$

$$\begin{aligned} \frac{\partial a_{li}}{\partial w_{lij}} &= \frac{\partial}{\partial w_{lij}} \left[ f \left( \sum_{m=1}^{N_{l-1}} w_{lmi} a_{(l-1)m} \right) \right] = f' \left( \sum_{m=1}^{N_{l-1}} w_{lmi} a_{(l-1)m} \right) \frac{\partial}{\partial w_{lij}} \left( \sum_{m=1}^{N_{l-1}} w_{lmi} a_{(l-1)m} \right) = \\ &= g(a_{li}) a_{(l-1)j} \end{aligned}$$

$$\frac{\partial E}{\partial a_{Li}} = a_{Li} - t_i$$

$$\begin{aligned} \frac{\partial E}{\partial a_{li}} &= \sum_{m=1}^{N_{l+1}} \frac{\partial E}{\partial a_{(l+1)m}} \frac{\partial a_{(l+1)m}}{\partial a_{li}} = \sum_{m=1}^{N_{l+1}} \frac{\partial E}{\partial a_{(l+1)m}} \frac{\partial}{\partial a_{li}} \left[ f \left( \sum_{n=1}^{N_l} w_{(l+1)nm} a_{ln} \right) \right] = \\ &= \sum_{m=1}^{N_{l+1}} \frac{\partial E}{\partial a_{(l+1)m}} f' \left( \sum_{n=1}^{N_l} w_{(l+1)nm} a_{ln} \right) \frac{\partial}{\partial a_{li}} \left[ \sum_{n=1}^{N_l} w_{(l+1)nm} a_{ln} \right] = \\ &= \sum_{m=1}^{N_{l+1}} \left[ \frac{\partial E}{\partial a_{(l+1)m}} g(a_{(l+1)m}) w_{(l+1)im} \right] \end{aligned}$$

Spomenutá funkcia  $g$  existuje pre všetky štandardné aktivačné funkcie, ako znázorňuje Tabuľka 1. Zapríčiňuje ďalšie zrýchlenie celého procesu výpočtu, pretože eliminuje drahé matematické operácie ako počítanie reálnej mocniny.

**Tabuľka 1** - Prehľad štandardných aktivačných funkcií a k nim prislúchajúcich funkcií  $g$ .

Názov	Aktivačná funkcia	Derivácia	$g$
Sigmoida	$f(a) = \frac{1}{1 + e^{-ca}}$	$f'(a) = cf(a)(1 - f(a))$	$g(fa) = cfa(1 - fa)$
Hyperbolický tangens	$f(a) = \tanh(ca) = \frac{e^{ca} - e^{-ca}}{e^{ca} + e^{-ca}}$	$f'(a) = c(1 - f^2(a))$	$g(fa) = c(1 - fa)$
Lineárna	$f(a) = ca$	$f'(a) = c$	$g(fa) = c$

Keď už vieme počítať  $\nabla E$ , môžeme prejsť k učeniu. Gradient  $\nabla E$  je vektor, ktorý smeruje k lokálnemu maximu chybovej plochy, preto musíme váhy  $\{w_{ij}\}$  zmeniť opačným smerom. Pri diskretnej časovej aproximácii sa váhy menia z času  $t$  na čas  $t + 1$  nasledovným spôsobom

$$w_{ij}(t + 1) = w_{ij}(t) - \mu \frac{\partial E}{\partial w_{ij}}$$

## Tri teoretické modely klasifikátorov

---

teda v maticovom tvare

$$W(t+1) = W(t) - \mu \nabla E(W(t)),$$

kde  $\mu > 0$  je učiaca konštanta ovplyvňujúca rýchlosť a kvalitu učenia. Jej zvyčajná hodnota sa pohybuje v intervale (0.1,1). Ak v predchádzajúcom výraze použijeme gradient parciálnej chybovej funkcie  $\nabla E$ , teda meníme váhy po každej jednej tréningovej vzorke, označujeme takéto učenie ako on-line. Ak použijeme  $\nabla E_{tot}$ , hovoríme, že je to batch učenie, alebo off-line. V tomto prípade však gradient treba predeliť počtom tréningových dvojíc, aby jeho veľkosť nebola od tohoto počtu závislá, pretože inak by bola od tohoto počtu závislá aj hodnota učiacej konštanty. Spomenutý vzťah sa tiež nazýva delta pravidlo.

V experimentoch sme použili ešte rôzne vylepšenia učenia podľa [HRI]. Prvé z nich je momentum. Tento spôsob predpokladá, že sa smer k lokálnemu minimu nemení príliš prudko, a preto pridáva istú zotrvačnosť. Tým dokáže preskočiť lokálne minimum a rôzne nerovnosti na chybovej ploche. V maticovom tvare sa dá zapísať takto

$$\begin{aligned} W(t+1) &= W(t) + \Delta W(t), \\ \Delta W(t) &= -\mu \nabla E(W(t)) + \alpha \Delta W(t-1), \end{aligned}$$

kde  $\alpha$  je momentum, väčšinou z intervalu (0.5,1). Ďalším vylepšením je odstránenie problému plytkej chybovej plochy (flat spot elimination). Používa sa vtedy, keď je chybová funkcia veľmi plytká  $\nabla E \approx 0$ . To môže viesť k veľmi pomalému učeniu. Aby sme predišli tomuto problému, upravíme počítanie gradientu nasledovne

$$\frac{\partial E}{\partial a_{ij}} = \sum_{m=1}^{N_{l+1}} \left[ \frac{\partial E}{\partial a_{(l+1)m}} (g(a_{(l+1)m}) + c) w_{(l+1)im} \right],$$

kde  $c$  je tzv. eliminačná konštanta plytkých miest (flat spot elimination constant), ktorej hodnota býva zvyčajne  $c \approx 0.25$ . Robili sme experimenty aj s inými zlepšeniami ako Adaptive Backpropagation a Super SAB [HRI], ale tieto neboli úspešné.

Ešte sme nespomenuli ako celú tréningovú procedúru naštartovať. Iniciálne hodnoty váh je vhodné voliť náhodne z intervalu (-1,1).

### 4.3.3 Predspracovanie vstupov a spracovanie výstupov

Jedna z užitočných transformácií vstupov je ich normalizácia. Spôsobuje to, že všetky zložky vektora budú mať rovnakú váhu. Najprv vypočítame strednú hodnotu a disperziu každej zložky vstupných vektorov

$$\langle x_i \rangle = \frac{1}{P} \sum_{p=1}^P x_i^p$$
$$\sigma_i^2 = \frac{1}{P-1} \sum_{p=1}^P (x_i^p - \langle x_i \rangle)^2$$

Následne môžeme transformovať vektory podľa vzťahu

$$\hat{x}_i^p = \frac{x_i^p - \langle x_i \rangle}{\sigma_i}$$

Vlastnosťou transformovaných vektorov je, že majú strednú hodnotu 0 a disperziu 1:

$$\begin{aligned} \langle \hat{x}_i \rangle &= \frac{1}{P} \sum_{p=1}^P \hat{x}_i^p = \frac{1}{P\sigma_i} \sum_{p=1}^P (x_i^p - \langle x_i \rangle) = \frac{1}{P\sigma_i} \left( P\langle x_i \rangle - \sum_{p=1}^P (x_i^p) \right) = \\ &= \frac{1}{P\sigma_i} (P\langle x_i \rangle - P\langle x_i \rangle) = 0 \\ \hat{\sigma}_i^2 &= \frac{1}{P-1} \sum_{p=1}^P (\hat{x}_i^p - \langle \hat{x}_i \rangle)^2 = \frac{1}{P-1} \sum_{p=1}^P (\hat{x}_i^p)^2 = \frac{1}{P-1} \sum_{p=1}^P \left( \frac{x_i^p - \langle x_i \rangle}{\sigma_i} \right)^2 = \\ &= \frac{\sigma_i^2}{P-1} \sum_{p=1}^P (x_i^p - \langle x_i \rangle)^2 = 1 \end{aligned}$$

Podobnú transformáciu môžeme urobiť aj na tréningových vektoroch, treba si však dať pozor, aby tieto hodnoty boli v rozsahu koncových aktivačných funkcií. Rovnakú transformáciu je dobré používať tak pri učení, ako aj pri klasifikácii.

# Kapitola 5

## Kvalita zvuku mobilných telefónov

Keďže sme sa rozhodli urobiť experiment, v ktorom budeme používať nahrávky z mobilných telefónov, popíšeme si stručne v tejto kapitole, ako funguje kódovanie a prenos signálu systémom GSM<sup>5</sup>, a kde môžu nastávať straty alebo skreslenia signálu.

### 5.1 Kódovanie reči v GSM

GSM môžeme označiť za určitý druh vokódera, ktorý kóduje zvuk, prenáša ho nižším dátovým tokom a na druhej strane ho dekóduje. GSM je digitálny systém, takže zvuk, ktorý je sám o sebe analógový, musí byť digitalizovaný (vzorkovaný a kvantovaný). Metóda kódovania zvukových dát použitá v ISDN<sup>6</sup> a v terajších telefónnych systémoch na multiplexovanie hlasových liniek cez vysokorýchlostné vodiče a optické vlákna je PCM. Pri štandardnej telefónnej vzorkovacej frekvencii 8000Hz je výstupný dátový tok PCM 64kbps, čo bolo pri vývoji systému GSM priveľa na prenášanie rádiovým spojením. Hoci je tento 64kbps signál ľahko implementovateľný, obsahuje veľa redundantných informácií. Preto dostala skupina GSM za úlohu zostrojiť zariadenie, ktoré by tento neduh odstránilo, bolo relatívne ľahko a efektívne zostrojiteľné a lacné. Skupina GSM študovala niekoľko algoritmov na kódovanie zvuku z pohľadu subjektívnej kvality zvuku a komplexity (ktorá sa týka nákladov, časovej náročnosti pri spracovaní a spotreby energie po implementácii), kým nenarazila na algoritmus s názvom Lineárny Prediktívny Kóder Excitovaný Pravidelnými Pulzmi s Dlhodobým Prediktorom - RPE/LTP<sup>7</sup>.

Popíšme si teraz bližšie proces kódovania signálu týmto algoritmom. GSM kóder kóduje 13 bitový (lineárne kvantovaný) digitálny signál vzorkovaný na 8000Hz rozdelený na 20 milisekundové bloky dĺžky 160 vzoriek na bloky dĺžky 260 bitov. RPE/LTP kódovací algoritmus patrí do triedy algoritmov lineárnych prediktívnych analýz syntézou (LPAS). Tak ako LPAS algoritmy, GSM kóder/dekóder reprezentuje rečový signál dvomi parametrami - informáciou o LPC filtri (vo forme kvantovaných Log Area Ratios alebo Q-LARS<sup>8</sup>) a informáciou o komprimovanom zvyškovom signále (vo forme kvantovaných parametrov RPE/LTP). Q-LARSov posiela osem, čo

---

<sup>5</sup> pôvodne Groupe Spécial Mobile, v súčasnosti Global System for Mobile Communication

<sup>6</sup> Integrated Services Digital Network

<sup>7</sup> Regular Pulse Excited - Linear Predictive Coder (RPE-LPC) with a Long Term Predictor loop alebo Residual Pulse Excitation/Long Term Prediction codec (RPE/LTP)

<sup>8</sup> sú to vlastne kvantované PARCOR koeficienty, alebo pomocné premenné pri výpočte LPC označené ako k

## Kvalita zvuku mobilných telefónov

je zároveň rád prediktora. Zo zvyškového signálu vyberie procedúra RPE/LTP dva dôležité parametre - vzdialenosť impulzov (Lag) a zosilnenie (Gain), a následne odpočíta signál vypočítaný z týchto parametrov. Zostane šumový signál, ktorý už nemusí byť prenesený dôkladne (všetky hlasivkové pulzy už určite pokryl signál predikovaný premennými Lag a Gain). Preto stačí, keď sa tento signál odhadne jednou zo štyroch podpostupností (každá štvrtá vzorka) s najväčšou energiou. Tá sa zakóduje 3-bitovou APCM<sup>9</sup> s 6-bitovým zosilnením. Kompresia zvyškového signálu je stratový proces, ktorý do neho vnáša skreslenie. Počas dekompresie je najprv zvyškový signál rekonštruovaný z parametrov RPE/LTP a následne excituje krátkodobý syntézový filter, ktorého parametre sú odvodené z prijatých LARS-ov. Diagram 1 zobrazuje schematickú reprezentáciu všeobecného LPAS kódera. V prípade Full-Rate GSM kódera, ako sa popisovaný algoritmus nazýva, komponent, ktorý minimalizuje rozdiel medzi zvyškovým signálom a rekonštruovaným zvyškovým signálom počíta jeho kvantovanú RPE/LTP reprezentáciu. Okrem stratovej reprezentácie zvyškového signálu hrá rolu v degradácii rečového signálu ešte kvantovanie LAR koeficientov.

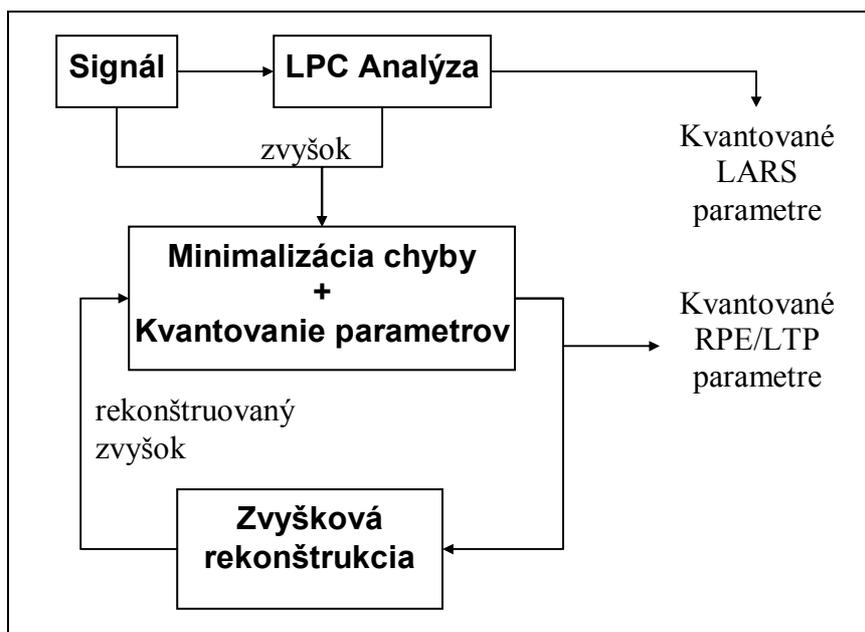


Diagram 1 - Schematická reprezentácia všeobecného LPAS kódera.

V súčasnosti je už niektorými severoamerickými operátormi používané Enhanced Full-Rate kódovanie, ktoré poskytuje zvýšenú kvalitu zvuku pri nezmenenej prenosovej rýchlosti 13kbps. Detailný popis RPE/LTP je možné nájsť v [DEG] a zdrojové kódy je možné nájsť v [DB].

<sup>9</sup> Adaptive Pulse Code Modulation - vzorky sú zakódované menším počtom bitov a spolu s nimi je zakódované aj zosilnenie na nejaký úsek vzoriek, ktorým sa pri dekompresii pre násobí

### 5.2 Kódovanie dátových kanálov a modulácia

Kvôli prirodzenej a človekom vyrobenej elektromagnetickej interferencii musí byť signál, ktorý prenáša zakódované dáta, chránený pred vznikom chýb. GSM používa k tomuto účelu konvolučné kódovanie a prelínanie blokov. Konkrétne algoritmy sú odlišné pre ochranu zvuku a iných dátových tokov. My si popíšeme len metódu, ktorá má za úlohu chrániť prenášané bloky zvuku. Spomeňme si, že RPE/LPT kóder produkuje 260 bitový blok každých 20 milisekúnd. Pri rôznych testoch sa zistilo, že niektoré bity blokov sú dôležitejšie ako ostatné. Preto sú jednotlivé bity bloku rozdelené do troch tried dôležitosti:

- **Trieda Ia** 50 bitov - najviac citlivé na chyby
- **Trieda Ib** 132 bitov - stredne citlivé na chyby
- **Trieda II** 78 bitov - najmenej citlivé na chyby

Bity triedy Ia majú pridaný 3 bitový CRC<sup>10</sup> na detekciu chýb. Ak je detegovaná chyba, úsek bitov tejto triedy je označený za príliš poškodený a nezrozumiteľný a je nepoužitý. Namiesto neho sa použije jemne stlmená verzia rovnakého úseku bitov z posledného korektne prijatého bloku. Týchto 53 bitov spolu so 132 bitmi Triedy Ib a 4 bitovou koncovou postupnosťou (spolu 189 bitov) vstupujú do polorýchlostného konvolučného kódera obmedzeného na dĺžku 4. Každý vstupný bit je kódovaný dvomi výstupnými bitmi v závislosti od predošlých 4 bitov. Konvolučný kóder teda vyprodukuje 378 bitov na výstupe, ku ktorým sa pridá zvyšných 78 bitov Triedy II, ktoré sú nechránené. Výsledkom je 465 bitov každých 20 milisekúnd, ktoré sa vysielajú, teda dátový tok 22.8 kbps.

Aby boli vysielané dáta ešte lepšie chránené voči bežným chybám pri impulznom prenášaní rádiovým spojením, bloky sa prelínajú. Výstupných 465 bitov z konvolučného kódera je rozdelených na 8 menších 57 bitových miniblokov a tie sú vysielané v ôsmich za sebou idúcich časových impulzoch. Keďže každý časový impulz môže preniesť dva 57 bitové minibloky, každý impulz prenáša dáta z dvoch rozličných blokov.

Autori (Degenerová a Bormann) implementácie RPE/LTP z nemeckej vedeckej skupiny Communications and Operating Systems Research Group, TU Berlín tvrdia, že táto kompresia je natoľko dobrá, že je možné spoľahlivo rozpoznať rečníka. Podľa ich slov aj prenášaná hudba často prežije transkódovanie v rozoznateľnej kvalite.

---

<sup>10</sup> Cyclic Redundancy Code

### 5.3 Dialogic Vox

Pretože v experimente sme použili hardware a software od výrobcu Dialogic, ktorý používal ako výstup komprimované súbory, a keďže je táto kompresia ďalším možným zdrojom skreslenia, popíšeme si na tomto mieste veľmi stručne aj túto kompresiu.

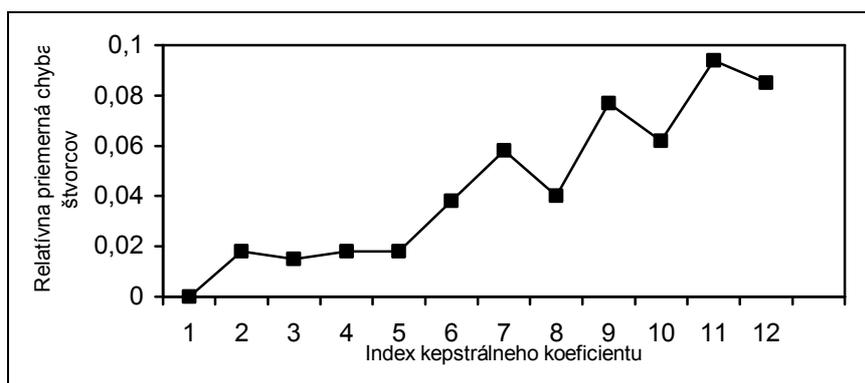
Súbory Dialogic Vox sú čisté binárne súbory obsahujúce kódované digitalizované zvukové vzorky kvantované 12 bitmi. Každý bajt pozostáva z dvoch zakódovaných vzoriek, teda každá 12 bitová vzorka je zakódovaná 4 bitmi. Pozícia od začiatku súboru je úzko spätá s časom, kedy bola vzorka na danom mieste nahraná. Zakódované vzorky sú uložené v bajte v poradí podľa času ich vzniku od najvýznamnejšieho bitu. Použitý kódovací algoritmus je ADPCM. Je to diferenčná kódovacia schéma, v ktorej každá zakódovaná vzorka aproximuje diferenciu medzi súčasnou hodnotou vzorky a predošlou hodnotou odhadu. Váhovanie veľkosti diferencie je pri tom adaptívne (nelineárne), teda sa môže meniť po každej vzorke. Presný popis kódovacieho aj dekódovacieho algoritmu je možné nájsť v [DAA] a zdrojové kódy na priloženom CD.

### 5.4 Možné skreslenia signálu

Kódovanie signálu reči vplýva na presnosť rozpoznávania reči poškodením presnosti vyjadrenia preslovu so znižujúcim dátovým tokom. S čím ďalej väčším používaním komunikačných aplikácií, ktoré používajú kódovacie algoritmy a ich interakciou s aplikáciami automatického rozpoznávania reči sa môže stať kódovanie rečového signálu významným problémom, ktorý ohraničuje ich úspešnosť. Degradácia presnosti rozpoznávania je väčšia, keď signál použitý na tréning rozpoznávača neprešiel rovnakým procesom spracovania, ako rozpoznávaný signál [HUE]. Avšak použitie rovnakého procesu kódovania signálu na tréning a testovanie neeliminuje celkom degradáciu presnosti rozpoznávania v porovnaní s nekódovaným signálom. Vychádzajúc zo štúdie [HUE] sa pokúsime načrtnúť, ako vplýva GSM (resp RPE/LTP) kompresia na parametre reči. GSM kompresia vplýva na rôzne kepstrálne koeficienty rôznymi podielmi. Graf 1 znázorňuje relatívnu priemernú chybu štvorcov (NMSE) medzi korešpondujúcimi kepstrálnymi koeficientami originálneho a GSM-kódovaného signálu reči. Ak by sme označili efekt skreslenia ako aditívny šumový signál, NMSE by zhruba odpovedala prevrátenej hodnote odstupe signálu od šumu. Môžeme si všimnúť, že NMSE spôsobená GSM kódovaním vo všeobecnosti rastie so stúpajúcim indexom koeficientu.

## Kvalita zvuku mobilných telefónov

---



**Graf 1** - Zobrazenie závislosti medzi indexom kepstrálneho koeficientu a skreslenia pri kompresii GSM.

Z popisu kódovania dátových kanálov vyplýva, že autori sa snažili v čo najväčšej miere predísť rôznym prenosovým chybám. No skreslenia signálu okrem samotnej kompresie RPE/LTP môžu spôsobovať aj poškodené bity jednotlivých tried. Po detekcii chyby v Triede Ia môže nastať skreslenie z dôvodu opakovania informácií z predošlého bloku (LTP). Kompresia ADPCM v súboroch VOX môže byť tiež zdrojom skreslenia, avšak nebudeme predpokladať, že významne ovplyvňuje výstupný signál.

# Kapitola 6

## Popis aplikácií

V tejto kapitole si bližšie popíšeme všetky tri implementácie z rôznych pohľadov. Budeme sa pri tom odvolávať na zdrojové kódy, ktoré sú priložené na CD k tejto práci.

Pre lepšiu zrozumiteľnosť ďalšieho textu si vopred definujme názvoslovie. Pod pojmom *nahrávka* budeme rozumieť celý kontinuálny blok signálu získaného nahrávaním z telefónu. Slovom *vzorka* označíme potenciálne slovo s automaticky určenými hranicami z nahrávky. Pojmom *vzor* označíme zaradený obraz slova s jednoznačným významom a *slovom* označíme významovú zložku vzorky, alebo celú triedu *vzorov* reprezentujúcich slovo.

### 6.1 Návrh a spoločné črty

Pri navrhovaní architektúry sme vychádzali z návrhu jednoduchého klasifikátora izolovaných slov využívajúceho DTW v [PSU] a návrhu klasifikátora využívajúceho HMM v [RJF]. Pre aplikácie sme navrhli jednotnú štruktúru, ktorá je dostatočne abstraktná, aby pokryla ľubovoľný klasifikátor izolovaných slov. Túto štruktúru zobrazuje Diagram 2, kde šípky označujú dátové toky medzi jednotlivými komponentmi a prerušované čiary určujú istú funkčnú spätosť jednotlivých komponentov. Jednotlivé komponenty sú veľmi blízke častiam, ktoré sme načrtli v teórii.

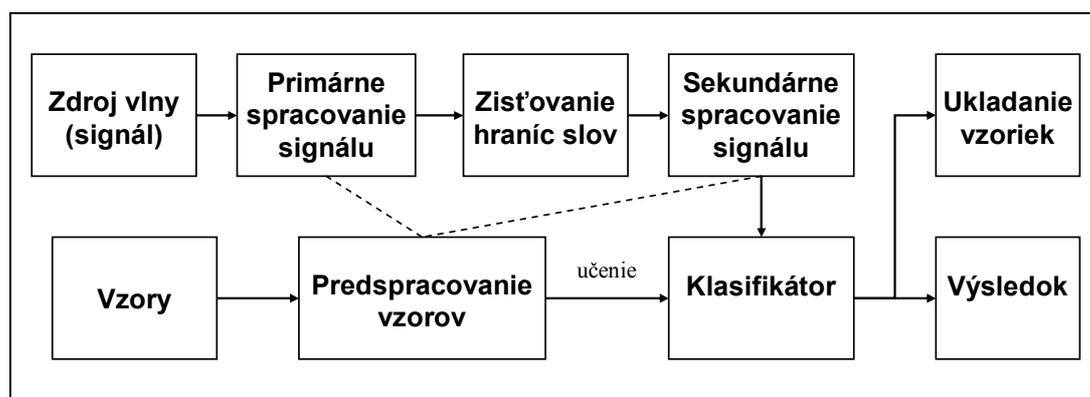


Diagram 2 - Štruktúra všeobecného klasifikátora izolovaných slov.

## Popis aplikácií

---

Popíšme si teraz jednotlivé komponenty štruktúry v poradí, akom určujú šípky. Najprv si popíšeme hornú vetvu až po "Klasifikátor" a potom pripojíme spodnú, ktorou budeme pokračovať až k popisu "Ukladania slov" a "Výsledku".

### 6.1.1 Zdroj vlny (signál)

Pojem zdroj vlny označuje akýkoľvek zdroj zvukového signálu v tvare PCM. Snažili sme sa ho navrhnuť tak, aby ho bolo možné jednoducho upraviť na ľubovoľný zdroj dát.

### 6.1.2 Primárne spracovanie signálu

Týmto označením sme mali na mysli nasledovné činnosti:

- rozdelenie signálu na mikrosegmenty
- získavanie základných charakteristík pomocou krátkodobých charakteristík v časovej oblasti
- prípadné odstraňovanie šumu z prostredia
- získavanie ďalších charakteristík pomocou krátkodobých analýz vo frekvenčnej oblasti

### 6.1.3 Zisťovanie hraníc slov

Tento pojem je už známy z úvodnej teoretickej časti, takže ho na tomto mieste nebudeme bližšie vysvetľovať. Poznamenajme len, že slovom sa tu myslí *vzorka*, čiže potenciálne slovo. Samotný proces hľadania hraníc slov sme implementovali tak, ako sme načrtli v kapitole 2, pričom sme sa pokúsili zaviesť aj experimentálne odstraňovanie šumu z prostredia.

### 6.1.4 Sekundárne spracovanie signálu

Pod týmto pojmom myslíme procesy predspracovania signálu vyššej úrovne ako

- úpravu obrazov slov
- segmentáciu spektrálnej stopy
- vektorovú kvantizáciu

### 6.1.5 Vzory

Týmto pojmom sme označili všetky triedy slov, teda kompletný slovník *vzorov*. Navrhli sme ho tak, aby vzory mohli byť uložené v súboroch.

## Popis aplikácií

---

### 6.1.6 Predspracovanie vzoriek

Pod týmto pojmom si predstavujeme proces ekvivalentný primárnemu a sekundárnemu spracovaniu signálu, pretože zisťovať hraničné body *vzorov* už nie je potrebné.

### 6.1.7 Klasifikátor

Naším hlavným predmetom záujmu je klasifikátor. V teoretickej časti sme si predstavili tri prístupy, akými možno klasifikovať izolované slová. Práve v tomto komponente sme mali na mysli možnosť voľby niektorého z nich a preto sme ho spravili dostatočne všeobecným a abstraktným. Za ním sa skrýva najrozsiahlejšia časť každej implementácie a ním sa aj jednotlivé aplikácie najviac a takmer výlučne líšia.

### 6.1.8 Ukladanie vzoriek

Týmto pojmom myslíme procedúru, ktorá dokáže ukladať zachytené *vzorky* do súborov, tým nemyslíme spracovaný signál, ale priamo čistý PCM signál.

### 6.1.9 Výsledok

Pod pojmom výsledkom si budeme predstavovať nejakú vhodnú reprezentáciu *slova*, ktoré najvýstižnejšie reprezentuje rozpoznanú *vzorku*.

## 6.2 Popis implementácií

Implementovali sme tri programy - ich aplikácie majú názvy SrDTW, SrHMM a SrNN. Všetky sme písali v prostredí Borland C++ Builder 5.0 a bežia na platformách Win32<sup>11</sup>. Zámerne sme nepoužili žiadnu špeciálnu knižnicu, pretože sme chceli preniknúť do hĺbky problematiky. Sledovali sme tým aj fakt, že na FMFI UK neexistuje a ani nie je vo vývoji žiadna knižnica, ktorá by sa zaoberala rozpoznávaním reči, aj keď je takmer na každej dôležitejšej univerzite vývoj takejto knižnice bežný a táto práca by mohla položiť prvý kameň do jej základu. Hlavnú časť sme mali snahu písať čo najviac v ANSI C++, aby bolo možné ju preniesť aj na iné platformy. Naším cieľom nebolo písať algoritmy čo najviac optimálne, ale čitateľne, aby bol ľahko pochopiteľný aj pre prípadného čitateľa. Na druhej strane, snažili sme sa neplyvať výpočtovou silou, takže algoritmy sú v rámci možností relatívne optimálne. V nasledujúcej časti si popíšeme jednotlivé triedy, z ktorých sa skladajú jednotlivé aplikácie. Pomôže nám pri tom Diagram 3.

---

<sup>11</sup> aplikácie sme testovali na platformách Windows 98, Windows NT, Windows 2000 a Windows XP



### 6.2.1 Spoločné časti

Všetky dôležité parametre, ktoré nebolo z nejakých dôvodov vhodné definovať ako premenné, sme vyňali do jedného hlavičkového súboru `defines.h`. Všetky sme nastavili na hodnoty, ktoré sú vhodné na telefónnu kvalitu zvuku, ktorá je predmetom nášho záujmu. Tieto hodnoty sú uvedené v nasledujúcej tabuľke (Tabuľka 2).

**Tabuľka 2** - Hodnoty základných parametrov, ktoré sme z rôznych dôvodov nezahrnuli medzi premenné.

vzorkovacia frekvencia	<code>SAMPLING_FREQUENCY</code>	8000Hz
veľkosť okienka	<code>BUFFER_SAMPLES</code>	128
rozmer vektora príznakov	<code>MATCHING_VECTOR_SIZE</code>	12
rozlíšenie zvuku	-	16 bit

Zdrojové kódy sme navrhli tak, aby bolo možné jednoducho modifikovať aj rozsah použitých reálnych čísel. Ich definíciu je možné nájsť tiež v tomto hlavičkovom súbore. Niektoré konštanty sme vyňali do inicializačného súboru, aby boli ľahšie meniteľné. Sú to najmä konštanty používané pri určovaní hraničných bodov slov a definície množín vzoriek pre slová. Podrobnejšie sa týmto parametrom venujeme v kapitole 6.3. V nasledujúcich riadkoch si popíšeme všetky pre nás významné triedy.

```
class TMVector
```

Týmto názvom sme nazvali základnú triedu, ktorá zaobaluje takmer elementárnu jednotku celého programu, a síce porovnávací vektor príznakov<sup>12</sup>. Jeho rozmer určuje parameter `MATCHING_VECTOR_SIZE`, pretože dynamická verzia implementácie tejto triedy bola kvôli veľkému počtu alokácií pamäte príliš pomalá. Pre ľahšie a frekventované narábanie s touto triedou sme preťažili jej niektoré aritmetické operátory. Keďže sa využíva ako základná stavebná jednotka pri budovaní slovníka DTW, naprogramovali sme jeho univerzálnu serializáciu<sup>13</sup>, ktorá umožňuje zapisovanie v aktuálnom formáte a čítanie z ľubovoľného formátu reálnych čísel.

```
class TWord
```

Reťazec príznakových vektorov (obraz slova) sme zaobalili do triedy `TWord`. Implementovali sme základné zoznamové operácie (ako `Insert`, `Add`, `Delete`, a `pod`). Pretože sa s touto triedou ďalej často manipulovalo, preťažili sme niektoré operátory. Taktiež v implementácii tejto triedy možno nájsť univerzálnu serializáciu.

```
class TFourier
```

Táto trieda zaobaluje Fourierovu transformáciu a kepstrálnu analýzu. Popri čitateľnosti sme sa snažili dbať aj na efektívnosť výsledného kódu. Najdôležitejšie metódy sú `DoPrecomputedFFT`, ktorá počíta Fourierovu transformáciu algoritmom

---

<sup>12</sup> jej meno pochádza z anglického `Matching Vector`

<sup>13</sup> ukladanie do a čítanie z prúdov (angl. `streams`)

## Popis aplikácií

---

FFT s využitím predpočítaných hodnôt, `DoCA` a `DoCAFromFFT`, ktoré počítajú kepstrálne koeficienty. Nevýhodou je len skutočnosť, že metódy pracujú správne len s veľkosťami okienok ohraničenými na mocniny dvoch.

```
class TTriangulation
```

Pomocou tejto triedy je možné simulovať výstupy z pásmových filtrov. Ako vstupy zadáme výstupy z Fourierovej transformácie a metóda `DoTriangulation` upraví vstupný vektor na vektor zadaného rozmeru tak, že pre každý komponent výstupného vektora prenásobí vstupný vektor okienkom tvaru trojuholníka, ktorý sa so zvyšujúcim indexom rozširuje a posúva sa vždy o polovicu šírky predošlého trojuholníka. Zvolili sme trojuholníky, aby výstupy neboli až také citlivé na presnosť rozmiestnenia daných pásiem.

```
class TLPA
```

Trieda zapuzdruje lineárnu prediktívnu analýzu. Jednoduchý výpočet koeficientov LPC zabezpečuje metóda `DoLPCCoefs`, z nich zabezpečuje výpočet kepstrálnych LPC koeficientov metóda `DoCepstralLPCCoefs`, a pomocou metódy `DoSpectralEnvelope` je možné získať hodnoty spektrálnej obálky hlasového traktu.

```
class TWaveFile
```

Túto triedu využívajú všetky aplikácie na čítanie vzorov zo súborov typu Windows PCM Wave. Umožňuje nielen čítanie, ale aj zápis týchto súborov, čo sa využíva prednostne pri ukladaní *vzoriek*.

```
class TWaveSource
```

Táto abstraktná trieda zaoberá sa spomínaným ľubovoľným zdrojom signálu vo formáte PCM. Ponúka virtuálne členské funkcie `SetFormat`, `GetFormatAddress`, `SetBufferInfo`, `Start`, `Stop`, `Pop`, `BytesInBuffer` a `Running`, ktoré stačí v potomkovi upraviť pre príslušný zdroj signálu. Tento súbor metód je dostatočne široký, aby pokryl potreby rôznych zdrojov. Zo spomenutých je najdôležitejšia funkcia `Pop`, ktorá prečíta zo zdroja zadaný počet bajtov.

```
class TWaveFileSource
```

Trieda zaoberá sa zdrojom signálu zo súboru typu Windows PCM Wave. Je odvodená z triedy `TWaveSource`. Vnútorne sa v nej využíva trieda `TWaveFile`.

```
class TVoxFileSource
```

Táto trieda umožňuje použiť ako zdroj signálu súbor typu Dialogic Vox. Zabezpečuje správnu dekompresiu dát, ktoré sú skomprimované ADPCM<sup>14</sup> kompresiou. Túto triedu sme zaviedli po získaní možnosti využiť komunikačný hardvér firmy Dialogic. Je taktiež odvodená od triedy `TWaveSource`.

---

<sup>14</sup> Adaptive Delta Pulse Code Modulation

## Popis aplikácií

---

`class TWaveInSource`

Implementácia tejto triedy zaobalujúcej štandardný audio vstup systémov Windows bola najkomplikovanejšia. Museli sme použiť pomocné vlákno<sup>15</sup>, aby sme zabezpečili korektné ukladanie dát. Táto trieda je taktiež odvodená od triedy `TWaveSource`.

`class TIntQueue`

V tejto triede reprezentuje frontu celých čísel typu `int`. Využíva sa pri určovaní hraničných bodov slov, presnejšie pri udržiavaní hodnôt intenzít a stredného počtu priechodov nulou, v triede `TProcessor`.

`class TProcessor`

Asi najkomplikovanejšia trieda spoločná pre všetky tri aplikácie je trieda `TProcessor`. Je tomu tak, pretože jej hlavná časť beží vo vlákne a neustále spracúva signál zo zadaného zdroja, automaticky určuje hraničné body slov, nahráva ich signálovú PCM reprezentáciu a zároveň signál po spracovaní ukladá to inštancie triedy `TWord`. Popri tom volá po spracovaní každého mikrosegmentu užívateľsky<sup>16</sup> definovanú "callback" funkciu `OnDrawAll`, pomocou ktorej je aplikácia schopná bezpečne čítať a zobrazovať jej dáta. Po zaznamenaní možného slova skontroluje jeho dĺžku, aby zodpovedala aspoň minimálnej dĺžke slova a potom bezpečne<sup>17</sup> zavolá užívateľsky definovanú "callback" funkciu `OnRecognize`. Jedinou nevýhodou tejto triedy je jej značná závislosť na prostredí Borland C++ Builder.

`function type TMetric`

Týmto názvom sme označili prototyp funkcie miery odlišnosti, ktorá je využívaná v mnohých ďalších funkciách a triedach.

`function Preprocess`

Táto funkcia vykoná proces úpravy obrazu slova, teda predspracuje `TWord` zvolenými spôsobmi.

`function ResizeWord`

Ako už názov tejto funkcie nahovára, zmení táto funkcia počet mikrosegmentov inštancie triedy `TWord` na zvolenú hodnotu, pričom zachová časovú ekvidistantnosť medzi jednotlivými mikrosegmentami.

`function SpectralPath`

Na rozdiel od predošlej funkcie, táto zmení dĺžku inštancie `TWord` (teda počet jej mikrosegmentov) využívajúc metódu segmentácie spektrálnej stopy spomenutú v teoretickej časti 4.1.2.

---

<sup>15</sup> angl. thread

<sup>16</sup> pod užívateľom teraz myslíme programátora aplikácie

<sup>17</sup> tak, aby bola zavolaná z hlavného vlákna aplikácie - to zabezpečuje trieda `TThread` z VCL, z ktorej je `TProcessor` odvodená

### 6.2.2 Časti špeciálne pre SrDTW

```
class TDTWModel
```

Táto trieda, ako už názov naznačuje, zaoberá sa triedou (množinou) *vzorov* pre jedno *slovo* a proces určovania vzdialenosti predspracovanej *vzorky* od tohto *slova* použitím dynamického krivenia časovej osi a zadanej miery odlišnosti typu `TMetric`. Teda trieda pracuje nad zoznamami tried `TWord`. Jej najdôležitejšia metóda je `GetDistance`, ktorá vracia hodnotu tejto vzdialenosti použitím pravidla k-NN spomenutého v teoretickej časti 4.3.1. Táto metóda predpokladá, že predpracované *vzory* (obrazy *vzorov*) v zozname aj *vzorka* majú rovnaké dĺžky (z hľadiska počtu mikrosegmentov). V tejto metóde je možné zadať ako argument aj DTW funkciu  $g$ , ale pre nepresvedčivosť použitia iných sme zásadne používali iba funkciu  $g_1$ . Pretože táto trieda zaoberá sa modelom slova, je pre ňu samozrejماً implementácia univerzálnej serializácie.

### 6.2.3 Časti špeciálne pre SrHMM

```
class TIntList
```

Podobne ako trieda `TIntQueue` aj táto trieda pracuje s množinou celočíselných hodnôt typu `int`. Táto trieda však udržiava tieto hodnoty v dynamickom zozname. Implementovali sme štandardné funkcie na prácu so zoznamom ako `Insert`, `Add`, `Delete`, `Clear`, atď. Navyše sme pre lepšiu prístupnosť k dátam preťažili niektoré operátory. Túto triedu sme ďalej využili v implementácii HMM a vektorového kvantizéra, kde reprezentujeme obrazy slov číselnými postupnosťami.

```
class TQuantizer
```

Vektorový kvantizér sme enkapsulovali do triedy `TQuantizer`. MacQueenov<sup>18</sup> adaptačný algoritmus reprezentuje asi najdôležitejšia metóda `MakeCenters`. Ďalšia dôležitá a pravdepodobne najpoužívanejšia metóda je metóda `VectorToInt`, ktorá kvantuje zadaný vektor k najbližšiemu centru a vracia jeho index. Celá trieda pracuje s abstraktnou funkciou miery odlišnosti, ktorú je možné nastaviť. Keďže adaptačný proces nie je jednoduchý, ale naopak časovo náročný a súvisí s tréňovaním HMM, implementovali sme univerzálnu serializáciu.

```
class TStateHMM_Params
```

Táto trieda reprezentuje parametre učenia skrytých Markovovských modelov. Zahrnuli sme do nej aj väčšinu výpočtových častí učenia. Keďže táto trieda je agregovaná v `TStateHMM` a udržiava v nej aj naučené parametre, implementovali sme do nej univerzálnu serializáciu. V metóde `Randomize` sme obmedzili HMM

---

<sup>18</sup> v anglickej literatúre známy aj ako K-means algorithm

## Popis aplikácií

---

na nielen dopredný, ale znemožnili sme mu pohyb do iných stavov ako do nasledujúceho a toho, v ktorom sa v ten moment nachádza. Táto zmena je pre prípadného vývojára ľahko odstrániteľná.

```
class TStateHMM
```

Dopredné HMM, ktoré generujú diskkrétne symboly v stave, ako sme popísali v teoretickej časti 4.2, zaobaluje trieda `TStateHMM`. Každý takýto naučený model reprezentuje jedno *slovo*. Najdôležitejšími metódami sú `LearnFrom`, ktorou sa model naučí zo zoznamu vzorov slov reprezentovaných zoznamami prirodzených čísel (vo všeobecnosti rôznej dĺžky) a `GetProbability`, ktorá vracia logaritmus pravdepodobnosti generovania zadanej postupnosti daným modelom. Ako pre každý model slov sme aj v tejto triede implementovali univerzálnu serializáciu.

### 6.2.4 Časti špeciálne pre SrNN

```
function type TActivation
```

Tento prototyp aktivačnej funkcie sme zaviedli, pretože sme chceli držať celú implementáciu neurónových sietí na abstraktnej úrovni. Rovnaký predpis má aj funkcia  $g$  z teórie, ktorá dokáže počítať deriváciu funkcie z jej funkčnej hodnoty. Preto sme ju použili aj pre tento prípad.

```
class TNetwork
```

Abstraktnú triedu `TNetwork` sme navrhli tak, aby zaobalovala funkčnosť neurónovej siete, ale na druhej strane neurčovala jej vnútornú architektúru, ktorá je pre aplikáciu pri jej použití zväčša nezaujímavá. Najdôležitejšie sú vždy len prístupy k vstupom, k výstupom a ku komponentom tréningového vektora, ktoré zabezpečujú metódy `Inputs`, `Outputs` a `Trainers` a metóda `ComputeActivities`, ktorá odpropaguje vstupné hodnoty až na výstup, a tým vlastne vypočíta funkciu, ktorú aproximuje neurónová sieť. Ako ďalšie dôležité abstraktné metódy uveďme `ZeroWeights`, ktorá vynuluje váhy, `RandomizeWeights`, ktorá inicializuje váhy malými náhodnými hodnotami, `ZeroGradient`, ktorá vynuluje celkový gradient, `ComputeGradient`, ktorá z aktivít neurónov, vstupov a výstupov vypočíta gradient chybovej funkcie, `ApplyLearningRule`, ktorá aplikuje učiace pravidlo, a `ComputeError`, ktorá vypočíta chybu na základe výstupných hodnôt a tréningového vektora.

```
class TCFFNN
```

Táto trieda reprezentuje kompletnú doprednú neurónovú sieť<sup>19</sup>, t.j. sieť, v ktorej existujú vždy všetky spojenia medzi každými dvomi vrstvami neurónov. Pri jej implementácii sme sa nechali inšpirovať teóriou v [HRI]. Je odvodená z triedy

---

<sup>19</sup> názov tejto triedy vznikol z angl. Complete Feed Forward Neural Network

## Popis aplikácií

---

TNetwork, teda implementuje jej funkčnosť. V jednotlivých vrstvách musia mať všetky neuróny zhodnú aktivačnú funkciu.

```
class TNeuron
```

Táto trieda zaoberá sa samotným neurónom, ktorý využijeme v inom prístupe k implementácii neurónovej siete, konkrétne v triede TFFNN. Tento prístup je viac konekcionistický, všeobecnejší a predurčenejší k paralelnému výpočtu, alebo chápaniu neurónových sietí. Neurón pozná vždy všetky váhy, ktoré vedú z neho a doň, môže mať ľubovoľnú aktivačnú funkciu a vie si počítať svoju aktiváciu metódou ComputeActivity.

```
class TNeuronConnection
```

Každý neurón triedy TNeuron je spojený s ostatnými pomocou tejto jednoduchšej triedy. Táto trieda obsahuje okrem základných metód aj základy výpočtov gradientu.

```
class TFFNN
```

Táto trieda pristupuje k implementácii už spomenutým viac konekcionistickým prístupom. Neuróny nemusia byť striktné zoskupené do pekných vrstiev s kompletnými váhami, ale ich váhy môžu viesť do ľubovoľnej nasledujúcej vrstvy tak, aby spĺňali definíciu doprednej neurónovej siete<sup>20</sup>. Vnútorne je táto sieť tvorená z neurónov typu TNeuron a váh triedy TNeuronConnection. Pre jej nie úplne striktnú vnútornú architektúru je táto trieda všeobecnejšia ako trieda TCCFNN. Pri jej implementácii sme vychádzali z [ÚNS]. Je tiež zdedená z triedy TNetwork, čiže implementuje jej plnú funkčnosť. (Pretože sme nakoniec túto triedu v experimente nevyužili, neimplementovali sme ani jej serializáciu.)

```
class TTrainingPair
```

Trénovaciu dvojicu neurónovej siete (vektor vstupov, vektor chcených výstupov) sme zapuzdрили do tejto jednoduchšej triedy.

```
class TTrainingSet
```

Množinu dvojíc triedy TTrainingPair sme potrebovali uložiť do dynamického zoznamu. K tomu nám poslúžila táto jednoduchá trieda, ktorá obsahuje len základné zoznamové operácie.

```
class TTransformer
```

Táto abstraktná trieda má za úlohu zabstraktovať všetky typy predspracovania vstupov a postspracovania výstupov neurónovej siete. Preto jediné dve jej metódy slúžia na vykonanie transformácie a jej inverzie.

```
class TNormalizer
```

Táto trieda, zdedená z triedy TTransformer, vykonáva transformáciu, ktorú sme popísali v teoretickej časti 4.3.3. Vychádzali sme pritom z teoretického základu

---

<sup>20</sup> odtiaľ aj názov tejto triedy - Feed Forward Neural Network

## Popis aplikácií

---

v [HRI]. Okrem zdedených metód implementuje ešte metódy `ComputeInputTransformation` a `ComputeOutputTransformation`, ktoré počítajú vstupné a výstupné transformácie, ktoré sa líšia potrebou upraviť rozptyl vzoriek na veľkosť výstupného rozsahu aktivačnej funkcie, ako sme spomenuli v teoretickej časti. Keďže táto trieda je súčasťou rozpoznávacieho procesu implementovali sme do nej univerzálnu serializáciu.

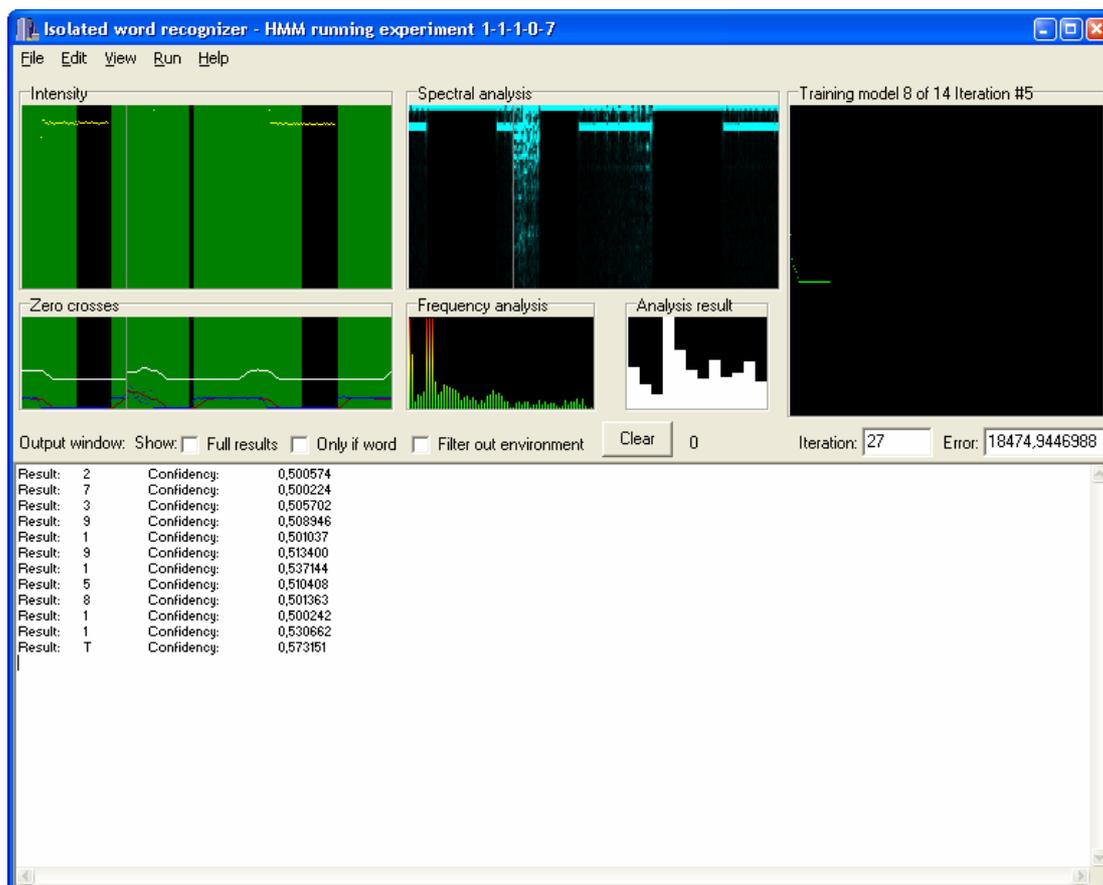
```
class TTrainer
```

Na tréovanie abstraktnej neurónovej siete `TNetwork` sme potrebovali nejaký adaptačný proces. Keďže ich existuje viacero, rozhodli sme sa vytvoriť triedu `TTrainer`. Táto trieda dokáže trénovať neurónové siete on-line aj off-line spôsobom, pričom sme umožnili používať momentum, elimináciu plochých miest chybovej plochy a toleranciu chyby výstupu, ako sme uviedli v teórii. Ako vstup používa tréovaciu množinu triedy `TTrainingSet`, ľubovlnú neurónovú sieť odvodenú od triedy `TNetwork` a ľubovlné vstupné a výstupné transformácie `TTransformer`.

### 6.3 Popis z užívateľského hľadiska

Všetkým trom implementovaným aplikáciám sme navrhli jednotný vzhľad, ktorý zobrazuje Obrázok 6. Snažili sme sa dbať na vizuálnu stránku, aby si užívateľ vedel jednoduchšie utvoriť predstavu o práci programov a dynamike jednotlivých analýz signálu. Do hornej polovice aplikačného okna sme sústredili vizualizačnú časť, do jeho dolnej polovice textový výstup a do stredu sme umiestnili ovládacie prvky.

## Popis aplikácií



Obrázok 6 - Vzhľad jednej z aplikácií.

Popíšme si teraz jednotlivé vizualizačné komponenty v poradí zľava doprava, zhora nadol. Prvé dva vizualizačné komponenty sme navrhli tak, aby užívateľ jasne videl, kedy program začal a kedy skončil nahrávanie *vzorky*, teda slúžia na vizualizáciu procesu zisťovania hraničných bodov slov. V oboch sa totiž za bežiacou čiarou označujúcou okamžitý časový moment mení farba pozadia z čiernej na zelenú, keď zachytil začiatok potenciálneho slova a zo zelenej na čiernu, keď zaznamenal koniec slova. V komponente *Intensity* sa navyše žltou farbou zobrazuje krátkodobá intenzita v danom časovom úseku a bielou farbou jej prah  $H_I'$ . V komponente *Zero crosses* sa navyše zobrazuje modrou farbou stredný počet prechodov signálu nulou, fialovou jeho vyhladená hodnota a bielou jeho prah  $H_Z'$ . Ďalšie tri vizualizačné komponenty slúžia na vytvorenie predstavy o priebehu rečového signálu vo frekvenčnej oblasti. Prvý z nich označený ako *Spectral analysis* vykresľuje cyklicky frekvenčnú analýzu signálu, čím umožňuje užívateľovi dobre sledovať formantovú štruktúru reči. Ďalší komponent označený ako *Frequency analysis* zobrazuje výstup Fourierovej transformácie mikrosegmentu v danom okamihu, resp. intenzitu každej zo 128 rovnomerne rozmiestnených frekvencií od 0 do 4000Hz. V poslednom zo spomínaných troch komponentov, označenom ako

## Popis aplikácií

---

*Analysis result*, vykresľujú aplikácie hodnoty koeficientov zvolenej analýzy (pásmových filtrov, kepstrálnej a LPC kepstrálnej). Posledný vizualizačný komponent, ktorý sa nachádza úplne na pravej strane slúži na vykresľovanie rôznych štatistík pri učení alebo výstupov pri behu programu. Jeho obsah závisí od konkrétnej aplikácie. Tento komponent sme zaviedli, pretože pri väčšom množstve dát užívateľ v tréningovom procese nemal prehľad o jeho rýchlosti a postupe. SrDTW do tohoto komponentu pri učení nevykresľuje nič, ale pri klasifikácii znázorňuje najlepšiu cestu krivenia časovej osi, SrHMM pri priebehu adaptačného procesu vektorového kvantizéra do tohoto komponentu vykresľuje celkové skreslenie a pri tréningovom procese jednotlivých Markovovských modelov celkovú chybu pre všetky vzory a nakoniec SrNN pri tréningovom procese celkovú chybu pre všetky tréningové dáta. S týmto komponentom sú zviazané textové polia, ktoré sme umiestnili pod neho. Pomáhajú zobrazovať konkrétne hodnoty, ktorých reprezentácia je zobrazovaná graficky.

Pod vizualizačné komponenty sme umiestnili ovládacie prvky. Popíšeme si ich v poradí zľava doprava. Zaškrtávacie políčko *Full results* umožňuje prepínať formu textového výstupu medzi jednoduchou, kde program vypisuje len rozpoznané slovo a istotu (vierohodnosť), a kompletnou, kedy program vypíše aj miery podobnosti všetkých slov s danou vzorkou. Zaškrtávacím políčkom *Only if word* je možné prepínať filtrovanie výstupu iba na slová, ktoré sú definované v inicializačnom súbore ako plnohodnotné<sup>21</sup>. Zaškrtávacím políčkom *Filter out environment* môže užívateľ ovládať experimentálne filtrovanie šumu z prostredia zo vstupného signálu. Za spomenutými ovládacími prvkami program vypisuje, koľko dát má zdroj vlny pripravených na spracovanie. Podľa tejto hodnoty sa dá ľahko určiť, či program má alebo nemá problém dostatočne rýchlo spracovávať dáta zo zdroja vlny<sup>22</sup>.

Do spodnej polovice okna sme umiestnili výstup. Urobili sme tak, pretože dáta v kompletnom výstupe vyžadujú pre prehľadné čítanie veľkú plochu. Do tohto textového poľa program vypisuje výsledok klasifikácie a jeho výslednú vierohodnosť, ktorá závisí vždy od konkrétnej implementácie.

Ponuky<sup>23</sup> aplikácií sme navrhli podľa štandardných pravidiel. Za zmienku stoja len neštandardné položky, ktoré v prehľadnej forme zobrazuje Tabuľka 3.

**Tabuľka 3** - Popis položiek z ponuky.

Položka	Význam
<i>File</i> → <i>Parse Experiment Results</i>	Spustí proces spracovania všetkých výsledkov, ktoré daný program vyprodukoval, a ktorý vyextrahuje z daných výsledkov počet chýb v každom teste a uloží ho do prehľadnej tabuľky v textovom súbore.
<i>Edit</i> → <i>Settings</i>	Otvorí okno s nastaveniami programu, kde je možné prepínačmi nastavovať spracovanie vo frekvenčnej oblasti

<sup>21</sup> t.j. ich index je menší alebo rovný ako parameter RealWords nastaviteľný v inicializačnom súbore

<sup>22</sup> Ak toto číslo rastie, počítač nestíha spracovávať dáta v reálnom čase.

<sup>23</sup> možno známejšie pod angl. názvom menu

## Popis aplikácií

	a používané metriky, zaškrtnutými poliami predspracovanie obrazov slov a pomocou textových polí programovo špecifické nastavenia.
ponuková skupina <i>View</i>	Sem sme umiestnili prepínače, ktoré umožňujú zabrániť zobrazovaniu informácií jednotlivých vizualizačných komponentov.
<i>Run</i> → <i>Run from WaveIn</i>	Spustí proces klasifikácie slov použitím štandardného audio vstupu systému (napr. mikrofón).
<i>Run</i> → <i>Run from file</i>	Spustí proces klasifikácie slov z užívateľom zadaného súboru, ktorý je typu <i>Windows Wave</i> alebo <i>Dialogic Vox</i> .
<i>Run</i> → <i>Save recognized as Waves</i>	Umožní ukladať zvuky, ktoré boli detegované ako slová, do súborov typu *.wav
<i>Run</i> → <i>Save recognized to Data</i>	Umožní ukladať zvuky rozpoznané ako jedno zo slov s vierohodnosťou väčšou ako <i>ConfidenciyTreshold</i> v inicializačnom súbore uložiť priamo do dát, pričom pri ďalšom tréningu sa už tieto budú používať.

Ku každej aplikácii sme vytvorili inicializačný súbor, ktorým je možné upravovať jej nastavenia. Prehľad významov jednotlivých položiek ponúkame v nasledujúcej tabuľke (Tabuľka 4).

**Tabuľka 4** - Popis parametrov v inicializačnom súbore.

Parameter	Význam
GroupCount	Počet slov, do ktorých chceme, aby program klasifikoval jednotlivé vzorky.
DataPath	Absolútna alebo relatívna <sup>24</sup> cesta k dátam, čiže jednotlivým vzorom.
ITresholdMul	Multiplikatívna konštanta pre intenzitu $c_m^I$ používaná pri hľadaní hraničných bodov vzoriek.
ITresholdMin	Aditívna konštanta pre intenzitu $c_a^I$ používaná pri hľadaní hraničných bodov vzoriek.
ZTresholdMul	Multiplikatívna konštanta pre počet priechodov nulou $c_m^Z$ používaná pri hľadaní hraničných bodov vzoriek.
ZTresholdMin	Aditívna konštanta pre počet priechodov nulou $c_a^Z$ používaná pri hľadaní hraničných bodov vzoriek.
IConsecutive	Počet za sebou idúcich mikrosegmentov prekračujúcich prah intenzity potrebných na začiatok „neticha“ $c_{cons}^I$ .
ZConsecutive	Počet za sebou idúcich mikrosegmentov prekračujúcich prah počtu priechodov nulou potrebných na začiatok „neticha“ $c_{cons}^Z$ .
MinWordLength	Minimálny počet mikrosegmentov slova.
MaxSilenceInWord	Maximálny počet segmentov „ticha“ vo vzorke.
PrecachedFrames	Počet pamätaných a cyklicky zabúdaných mikrosegmentov pred začiatkom vzorky.
PostcachedFrames	Počet pamätaných mikrosegmentov „ticha“ po konci vzorky.

<sup>24</sup> vzhľadom na aplikačný súbor

## Popis aplikácií

---

ConfidenciyTreshold	Program pri zapnutom <i>Run</i> → <i>Save to data</i> ukladá <i>vzorky</i> medzi <i>vzory</i> , ak ich rozpoznaná vierohodnosť je vyššia ako tento prah.
RealWords	Program pri zapnutom ovládacom prvku <i>Only if word</i> vypisuje rozpoznáný výstup, len ak index jeho <i>slova</i> , do ktorého bol zaklasifikovaný, je nie je väčší ako táto hodnota.
WaveCount	Index, od ktorého má začať program ukladať <i>vzorky</i> do súborov v tvare <i>wave [Index] .wav</i> pri zapnutom <i>Run</i> → <i>Save as waves</i> .
WavePath	Absolútna alebo relatívna cesta, kam bude program ukladať <i>vzorky</i> pri zapnutom <i>Run</i> → <i>Save as waves</i>
[GroupX]	Tento atribút určuje začiatok informácií o definícii <i>slova</i> s poradovým číslom X, ktoré má byť v rozmedzí 1...GroupCount. Nasledujúce parametre sú zviazané s týmto atribútom.
Word	Znakový reťazec reprezentujúci dané <i>slovo</i> X.
WaveCount	Počet súborov <i>vzorov</i> , ktoré program načíta pred procesom tréovania modelu.
WaveNameBody	"telo" súboru <i>vzoru</i> , ktoré bude program pred procesom učenia čítať v tvare [telo]+i+".wav", kde i je 1...WaveCount

# Kapitola 7

## Experiment

V tejto kapitole si podrobne popíšeme experiment, v ktorom sme porovnávali úspešnosť klasifikácie izolovaných slov jednotlivými prístupmi. Rozhodli sme sa pri tom rozpoznávať čísllice od 0 po 9, slová "áno"  $A$  a "nie"  $N$ , koncový tón telefónneho hovoru  $T$  a šum  $S$ .

### 7.1 Príprava

Po absolvovaní dlhej cesty vývoja aplikácií sme začali prípravnú fázu experimentu. Prvým krokom bolo získanie malej množiny *nahrávok* potrebných pre utvorenie predstavy o kvalite a typickom priebehu zvukového signálu z mobilného telefónu. Preto sme celý proces (aj ďalšieho) nahrávania vykonali v spolupráci so spoločnosťou Globtel<sup>25</sup>, ktorá nám vyšla v ústrety a poskytla potrebný hardvér a softvér, ktorý mala k dispozícii.

Po získaní prvých *nahrávok* sme už mohli vhodne nastaviť inicializačné súbory jednotlivých programov a zaručiť, aby hodnoty slúžiace k určovaniu hraníc slov boli totožné pre všetky aplikácie. Po krátkom skúšaní na malej množine telefónnych vzoriek sme empiricky zistili vhodné nastavenia konštánt na určovanie hraničných bodov slov. Pre telefónnu kvalitu signálu sme dospeli k nastaveniam, ktoré prehľadne zobrazuje Tabuľka 5.

**Tabuľka 5** - Prehľad empiricky zistených hodnôt nastavení programov.

Parameter	Hodnota
počet pamätaných a cyklicky zabúdaných mikrosegmentov pred začiatkom <i>vzorky</i>	18
počet pamätaných mikrosegmentov „ticha“ po konci <i>vzorky</i>	30
maximálny počet segmentov „ticha“ vo <i>vzorke</i>	30
minimálny počet mikrosegmentov slova	25
multiplikatívna konštanta pre intenzitu $c_m^I$	1
aditívna konštanta pre intenzitu $c_a^I$	2000
multiplikatívna konštanta pre počet priechodov nulou $c_m^Z$	1
aditívna konštanta pre počet priechodov nulou $c_a^Z$	40

<sup>25</sup> v súčasnosti už známou pod názvom Orange

## Experiment

---

počet za sebou idúcich mikrosegmentov prekračujúcich prah intenzity potrebných na začiatok „neticha“ $C_{cons}^I$	3
počet za sebou idúcich mikrosegmentov prekračujúcich prah počtu priechodov nulou potrebných na začiatok „neticha“ $C_{cons}^Z$	6

Jednotlivé parametre sme už počas prípravy aj behu experimentu nemenili. Mali sme k tomu dva dôvody. Prvý bol, aby dosiahnuté výsledky boli objektívne z hľadiska určovania hraníc *vzorov* a testovacích *vzoriek* a dali sa objektívne porovnávať, a druhý, aby bol totožný počet *vzoriek* v testovacích súboroch s vopred určenými, analyzovanými a popísanými v textových súboroch (viď ďalej).

Aby sme mohli získať kvalitné *vzory*, potrebovali sme analyzovať, ako ľudia jednotlivé *slová* vyslovujú. Zistili sme, že človek vyslovuje *slová* s rôznou intenzitou a intonáciou hlasu, ba dokonca niektoré hláskovo odlišnými slovami<sup>26</sup>, v závislosti od pozície daného slova medzi inými slovami, na začiatku textu alebo na jeho konci. Preto sme vytvorili tzv. trénovací papier, ktorý možno nájsť v dodatku A. Sú v ňom štyri telefónne čísla a osem otázok. Telefónne čísla sú už na prvý pohľad neštandardné. Snažili sme sa o to, aby sa v každom vyskytoval každá číslica práve raz a aby boli pozície číslic vzhľadom na skupiny čísel rôzne. Dôležitý psychologický efekt pri tom zohral fakt, že čísla aj naďalej vyzerajú ako telefónne. Otázky sme sa snažili vytvoriť tak, aby zhruba polovica odpovedí bola "áno" a polovica "nie". Koncipovali sme ich tak, aby mal rečník pocit, že odpovedá na otázky a nesústredoval sa na fakt, že je nahrávaný. Z uvedeného vyplýva, že vo výslednej nahrávke sa mal vyskytovať *vzor* každého *slova* približne štyrikrát. Tento počet sme zvolili z dôvodu zabezpečenia dostatočnej redundancie *vzoriek*, aby sme zabránili absencii niektorých *vzorov* od každého rečníka už pri bežnej chybovosti reči a signálu a prítomnosti šumu. Naproti tomu tento počet nie je natoľko vysoký, aby spôsoboval väčšie časové problémy pri získavaní dostatočného počtu nahrávok.

Časovo a organizačne náročným procesom bolo získavanie *nahrávok*. Rozhodli sme sa, ak to bude možné, získavať nahrávky v prostrediach s relatívne nízkou hladinou šumu. Takéto prostredia zaručujú lepšiu a vyrovnanú kvalitu *vzorov* a z toho vyplývajúcu presnosť reprezentácie hlasového ústrojenstva rečníka a teda aj kvalitu natrénovania jednotlivých modelov. Keďže sme chceli systémy urobiť nezávislé na rečníkovi, vyberali sme rečníkov oboch pohlaví a rôzneho veku. Najväčší počet *nahrávok* sa nám však podarilo získať na Evanjelickom Lýceu v Bratislave, ktorého vedenie a žiaci nám vyšli v ústrety a umožnili zorganizovať nahrávanie, teda najviac rečníkov bolo vo veku okolo 13 rokov. Počet rečníkov s daným vekom bol nepriamo úmerný ich veku, preto najstarších bolo najmenej. Rečníkov sme upozorňovali, aby pokiaľ možno hovorili *slová* prirodzene a nie roboticky, pričom si však museli dávať pozor na dostatočné medzery medzi ich vyslovovaním. Nahrávací proces bol časovo náročný, pretože každý rečník musel byť dostatočne oboznámený s nahrávacou procedúrou a niekedy bolo proces potrebné

---

<sup>26</sup> napr. číslo 1 možno vysloviť ako "jedna" alebo "jeden"

## Experiment

---

opakovať z technických dôvodov<sup>27</sup>. Aj keď samotné nahovorenie potrebných vzorov trvalo okolo 90 sekúnd, stávalo sa, že sme potrebovali aj 5 minút na získanie jedinej nahrávky, čo je minimum vzhľadom na čas, ktorý sme potrebovali za zabezpečenie rečníkov. Celkovo sa nám podarilo získať okolo 130 rôznych nahrávok rôznej kvality. Na záznam nahrávok sme použili mobilné telefóny Ericsson A1018s a Nokia 3210. Firma Globtel nám poskytla na nahrávanie stanicu Sun s komunikačnou kartou od firmy Dialogic.

Zo zaznamenaných nahrávok sme niektoré vyčlenili do tzv. testovacej skupiny. Ostatné sme spracovali tak, že sme z nich automaticky vyextrahovali jednotlivé vzorky jedným z programov (zapnutím *Menu* → *Run* → *Save recognized as Waves* a behom zo súboru). Uložené vzorky sme ďalej manuálne<sup>28</sup> roztriedili podľa ich významu a premenovali, pričom nepodarky sme ignorovali. Medzi nepodarky patrili vzorky so zle určenými hraničnými bodmi slova a zvuky nezatriediteľné medzi slová. Následne na to sme upravili konfiguračné súbory tak, aby pokrývali všetky novo vytvorené vzorky.

Tabuľka 6 - Prehľad popisných hodnôt testovacích dát.

	Počet nahrávok	Počet vzoriek	Počet otáznikov v popise
Netrénované	16	746	5
Trénované	6	311	2
Rovnaký rečník	1	17	0
Celkovo	23	1074	7

Do testovacej skupiny sme zahrnuli ešte niektoré spracované nahrávky, aby bolo možné testovať chybovosť jednotlivých programov aj na trénovacích dátach. Zo všetkých nahrávok z takto upravenej testovacej skupiny sme taktiež automaticky vyextrahovali vzorky. Vzorky sme následne počúvali a ukladali ich význam v textovej forme do textových súborov, aby sme tieto mohli v experimente použiť na automatické porovnávanie úspechov klasifikácie. Aj tu sa vyskytli nepodarky, tak sme ich v textových súboroch označili symbolom ?, ktorý zabezpečil, aby sa tam "pomýlil" každý z rozpoznávačov rovnako. Celkovo sme použili 23 testovacích nahrávok, v ktorých sa nachádzalo 1074 testovacích vzoriek, z ktorých bolo 7 popísaných symbolom ?. Prvých 16 nahrávok sme zvolili nespracovaných, z ktorých 8 bolo dievčenských hlasov, 4 chlapčenské a 4 mužské. Ďalších 6 nahrávok sme vybrali z trénovacích dát (hlasy chlapcov a dievčat), a za poslednú nahrávku sme zvolili nahrávku rečníka, ktorého iná nahrávka sa už v trénovacích dátach nachádzala. Prehľadne to zobrazuje Tabuľka 6. Celý popísaný proces bol taktiež značne časovo náročný.

---

<sup>27</sup> keď bol rečník nečinný niekoľko sekúnd, nahrávacia služba sa vyplá

<sup>28</sup> pre zaujímavosť, je ich okolo 5000

### 7.2 Priebeh experimentu

V SrDTW sme použitím segmentácie spektrálnej stopy pri učení po úprave každého obrazu slova zmenili jeho dĺžku na vopred stanovenú. Rovnakým predspracovaním sme upravili aj každú *vzorku* pri testovaní a porovnali ju so všetkými *vzormi* v naučenom slovníku, pričom sme vybrali výsledok pomocou pravidla k-NN.

V SrHMM sme použili dopredné diskkrétne HMM generujúce v stavoch s obmedzenými prechodmi vždy len o jeden vpred. Modely sme učili vždy tri krát, pričom sme vybrali vždy najlepšie naučený. Vektorový kvantizér sme učili len raz, pretože rôzne iterácie dávali z pohľadu celkového skreslenia porovnateľné výsledky. Učili sme ho, pokiaľ sa zlepšoval, bez iného kritéria zastavenia.

V SrNN sme použili kompletne dopredné neurónové siete s klasifikačnou architektúrou, ktoré sme učili 400 iterácií s parametrami

- rýchlosť učenia 0.7
- momentum 0.9
- konštanta na odstránenie plochých miest 0.2.

Pre každý program sme zvolili množinu všetkých relevantných nastavení. Pri prvých priebehoch experimentu sme narazili na niekoľko problémov, ktoré sú popísané v nasledujúcej kapitole. Kvôli značnej časovej náročnosti tréningového procesu najmä pri učení HMM sme museli spomenuté množiny upraviť (zúžiť) tak, že sme vynechali tie nastavenia, ktoré sa nám zdali byť (po preštudovaní literatúry) málo pravdepodobné alebo nepravdepodobné na dobrý úspech. Výsledné relevantné nastavenia jednotlivých programov možno vidieť v tabuľkách 7-9 (pričom sme použili len relevantné metriky pre každý typ analýzy).

**Tabuľka 7** - Prehľad relevantných nastavení programu SrDTW.

Parameter	Hodnoty
Dĺžky slov	12-20 po krokoch 4
Analýzy	Fourierova <sup>29</sup> , kepstrálna LPC
Metriky	Euklidova, uhlová, kepstrálna, váhovaná kepstrálna
Úpravy obrazov slov - predspracovania	žiadne, potrebné (body 1 a 2), kompletne (body 1-5)
Šírka prehľadávaného priestoru	1-1/4 dĺžky slova
Počet susedov v k-NN	1-9 po krokoch 2

**Tabuľka 8** - Prehľad relevantných nastavení programu SrHMM. Nepoužili sme všetky kombinácie týchto hodnôt.

Parameter	Hodnoty
Analýzy	Fourierova, kepstrálna LPC
Metriky	Euklidova, váhovaná kepstrálna, kepstrálna

---

<sup>29</sup> týmto sa myslí simulácia pásmových filtrov

## Experiment

---

Úpravy obrazov slov - pedspracovania	žadne, potrebné (body 1 a 2), kompletne (body 1-5)
Pocet centier	128, 256, (1024)
Pocet stavov HMM	5, 7, (9)

**Tabuľka 9** - Prehľad relevantných nastavení programu SrNN.

Parameter	Hodnoty
Dĺžky slov	12-32 po krokoch 4
Analýzy	Fourierova, kepstrálna LPC, kepstrálna
Úpravy obrazov slov - pedspracovania	žadne, potrebné (body 1 a 2), kompletne (body 1-5)
Pocet skrytých neurónov	15-35 po krokoch 10

Celý experiment sme spustili so všetkými 23 testovacími *nahrávkami* v každom programe, pričom sme experiment aplikácie SrHMM upravili a rozdelili tak, aby mohli rôzne jeho časti bežať na viacerých počítačoch paralelne. Použili sme niekoľko počítačov s procesormi Athlon XP 1500-1700+ a operačným systémom Windows XP. Experiment sme nechali bežať niekoľko dní. Po celý čas bolo zapnuté ukladanie modelov (okrem SrDTW, kde sa dajú modely generovať na počkanie), aby sme ich mohli dať k dispozícii na priloženom CD.

V každom kroku experimentu sa najprv spustil trénovací proces s danými nastaveniami, potom sa spustil rozpoznávací (klasifikačný) proces na všetkých testovacích nahrávkach a nakoniec sa inicioval vyhodnocovací proces. V ňom sa výstupy klasifikácie porovnali s manuálne zostrojenými dátami v textových súboroch a na základe toho sa generoval výstupný textový súbor s výsledkami spracovanými do prehľadnej tabuľky pre každú testovaciu *nahrávku*<sup>30</sup>.

Po skončení experimentov jednotlivých aplikácií sme ich výsledky spracovali danou aplikáciou do dvoch prehľadných tabuliek, s ktorými už bolo možné jednoducho manipulovať.

### 7.3 Problémy

Kvôli problémom s množstvom dát sme boli nútení urobiť úpravy jednotlivých aplikácií.

Vektorový kvantizér sme učili nie zo všetkých vektorov, ale asi len z ich jednej pätiny, pretože vytvorené dáta spôsobovali pamäťové a časové problémy. Tie spočívali v dynamickom menení jedného pamäťového bloku, ktorý udržiaval smerníky na všetky vektory príznakov. Pred každým pridaním sa pamäť zväčšila a pri hranici okolo osemdesiatich tisíc vektorov sa už dynamicky zmeniť veľkosť nepodarilo. To všetko aj napriek tomu, že číselne bolo fyzickej pamäte dostatok (nastala vysoká fragmentácia pamäte).

---

<sup>30</sup> Tieto súbory je potom možné dobre spracovať v programe MS Excel.

## Experiment

---

Markovovské modely sme trénovali len tri krát, na rozdiel od pôvodných päť, pretože experiment trval príliš dlho. Vo všetkých programoch sme pracovali s reálnymi číslami typu *long double*, pretože pri učení (s použitím normalizácie) HMM a použití dlhších slov hodnoty  $\beta_i(t)$  narastali do neprijateľných veľkostí (presahujúcich rozsah *double*). Po dôkladnom preštudovaní zdrojového kódu a porovnaní s inými implementáciami sme vylúčili možnosť implementačnej chyby. Ukázalo sa, že problém vznikol pri trénovaní s dlhšími *vzormi*, pretože hodnoty  $\beta_i(t)$  sú porovnateľné s hodnotami  $\alpha_i(t)$ , ako sa predpokladá v teórii, ale nie až natoľko, aby bol počítač ich hodnoty schopný reprezentovať aj pri vzoroch dĺžky asi ako celá veta. Na tento problém sme narazili v podstate náhodou, a to po zaradení do *vzorov* šumu dlhší zvuk, kvôli vektorovému kvantizéru.

Ďalším dôvodom tohoto rozhodnutia bolo, že adaptačný proces neurónovej siete tiež niekedy skončil výnimkou. Dôvodom bolo zrejme použitie sigmoidálnych aktivačných funkcií a diskrétna tréningová výstup 0 a 1. Pri dôkladnom trénovaní sa pravdepodobne snažila sieť priblížiť jednej z týchto hodnôt, až sa dostal výsledok funkcie *exp* mimo rozsah *double*. Toto sa dalo riešiť obmedzením argumentu funkcie *exp* alebo trénovaním hodnotami 0.1 a 0.9, no my sme zvolili iný prístup - vypli sme výnimky spôsobované FPU a tým sme umožnili procesoru pracovať so symbolmi  $\infty$  a  $-\infty$ .

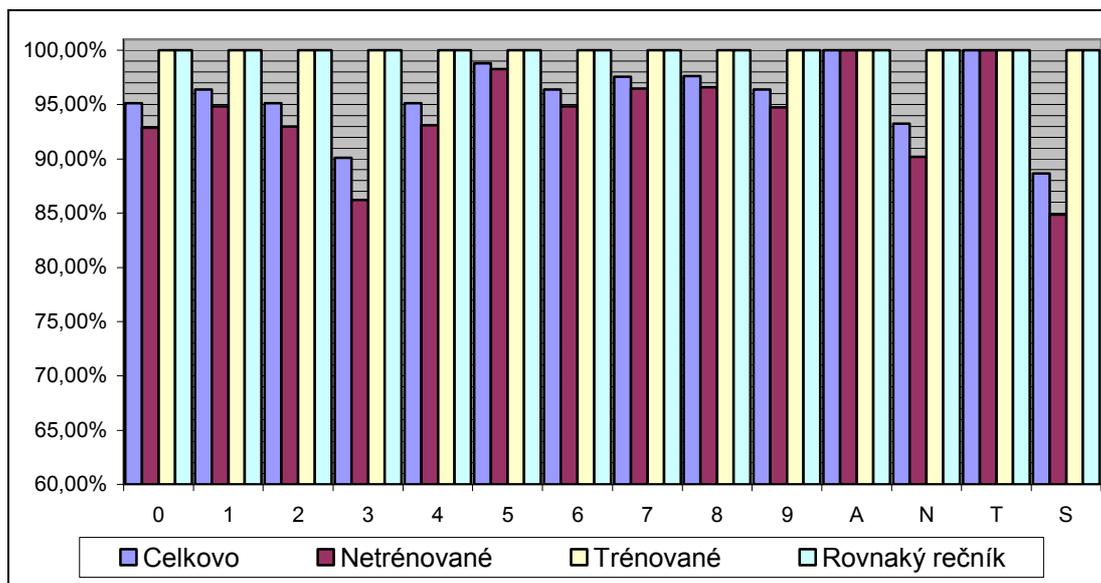
Narazili sme ešte na zopár technických problémov s neuvolňovaním pamäte. Tento zdanlivo nie až tak podstatný problém sa ukázal pri takýchto dátových rozmeroch ako fatálny. Vďaka výbornému nástroju, akým CodeGuard od firmy Borland nepochybne je, sa nám však podarilo s ľahkosťou odstrániť aj tieto problémy.

## 7.4 Výsledky

Zopakujme, že sme klasifikovali číslice od 0 do 9, slová Áno a Nie, Tón a Šum. Testovacie nahrávky boli zložené z nenatréovaných, natréovaných a hlasu rovnakého rečníka, ako v jednej z tréningových nahrávok.

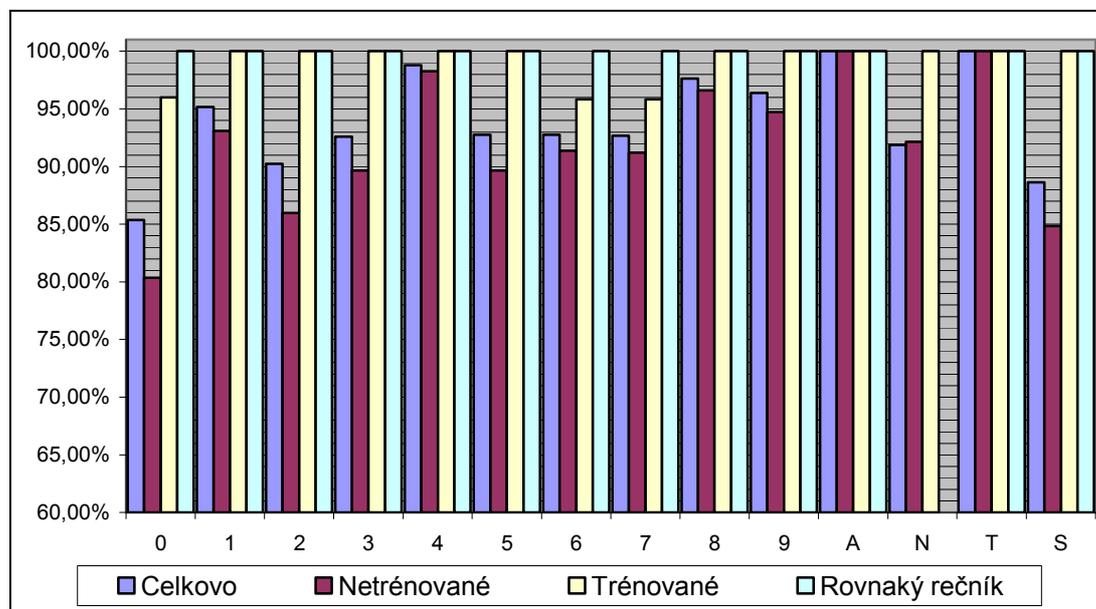
Výsledky úspešností prvých 100 najúspešnejších rozpoznávačov DTW je možné vidieť v prílohe B. V prvej tabuľke (Tabuľka 10) uvádzame chybovosť a úspešnosť klasifikácie spolu s popisom parametrov jednotlivých klasifikátorov a v druhej tabuľke (Tabuľka 11) prezentujeme ich úspešnosť rozpoznávania jednotlivých slov. Úspešnosť najlepšieho DTW rozpoznávača (s kódom 12-2-3-3-0-3) na jednotlivých slovách a typoch rečníkov zobrazuje Graf 2.

## Experiment



**Graf 2** - Úspešnosť najlepšieho DTW klasifikátora s kódom 12-2-3-3-0-3 na jednotlivých slovách podľa ich tried.

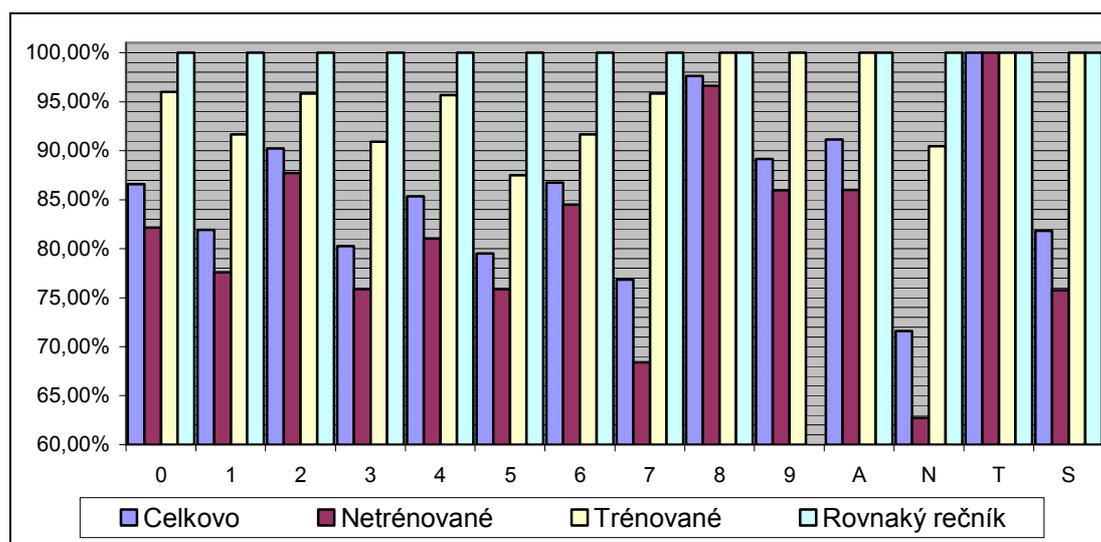
Výsledky úspešností prvých 100 najúspešnejších NN klasifikátorov možno nájsť v prílohe C. Tabuľka 12 zobrazuje chybovosť a celkovú úspešnosť jednotlivých klasifikátorov a Tabuľka 13 zobrazuje úspešnosť jednotlivých klasifikátorov pri rozpoznávaní každého zo slov podľa tried zaradenia. Graf 3 zobrazuje úspešnosť rozpoznávania jednotlivých slov podľa ich tried najúspešnejšieho NN rozpoznávača (s kódom 32-1-3-2).



**Graf 3** - Úspešnosť najlepšieho NN klasifikátora s kódom 32-1-3-2 na jednotlivých slovách podľa ich tried.

## Experiment

Kompletné výsledky úspešnosti klasifikátorov používajúcich skryté Markovovské modely možno nájsť v dodatku D. V prvej tabuľke (Tabuľka 14) uvádzame v poradí od najúspešnejšieho výsledky chybovosti a úspešnosti jednotlivých klasifikátorov spolu s popisom ich nastavení. Úspešnosť HMM klasifikátorov na jednotlivých slovách v tom istom poradí zobrazuje Tabuľka 15. Graf 4 zobrazuje úspešnosť rozpoznávania jednotlivých slov podľa ich tried zaradenia najúspešnejšieho HMM klasifikátora (s kódom 2-4-3-7-9).

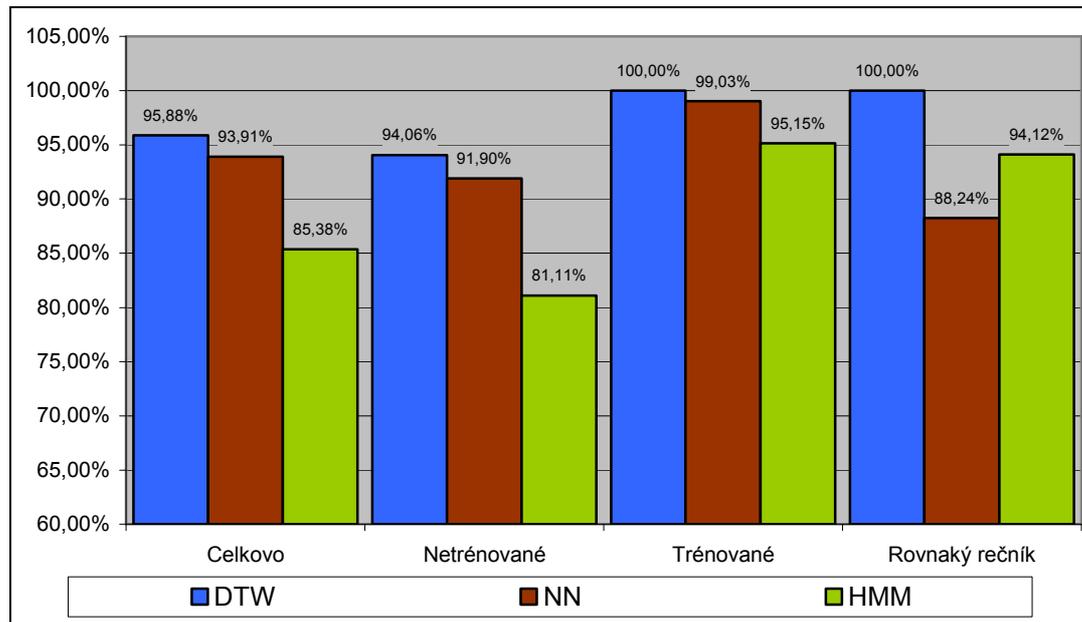


**Graf 4** - Úspešnosť najlepšieho HMM klasifikátora s kódom 2-4-3-7-9 na jednotlivých slovách podľa ich tried.

Vo všetkých troch grafoch sme použili úmyselne rovnakú mierku od 0 do 100, aby bolo lepšie vidieť rozdiely úspešnosti jednotlivých prístupov. V grafe 4 neuspel klasifikátor pri klasifikovaní slova "9" a v grafe 3 neuspel klasifikátor na slove "Nie" v nahrávke rovnakého rečníka a preto je úspešnosť nulová. Nebolo našim cieľom zameriavať sa na úspešnosť rovnakého rečníka, pretože sme použili len jednu nahrávku a výsledky by neboli objektívne, preto bledomodré majú len orientačný charakter.

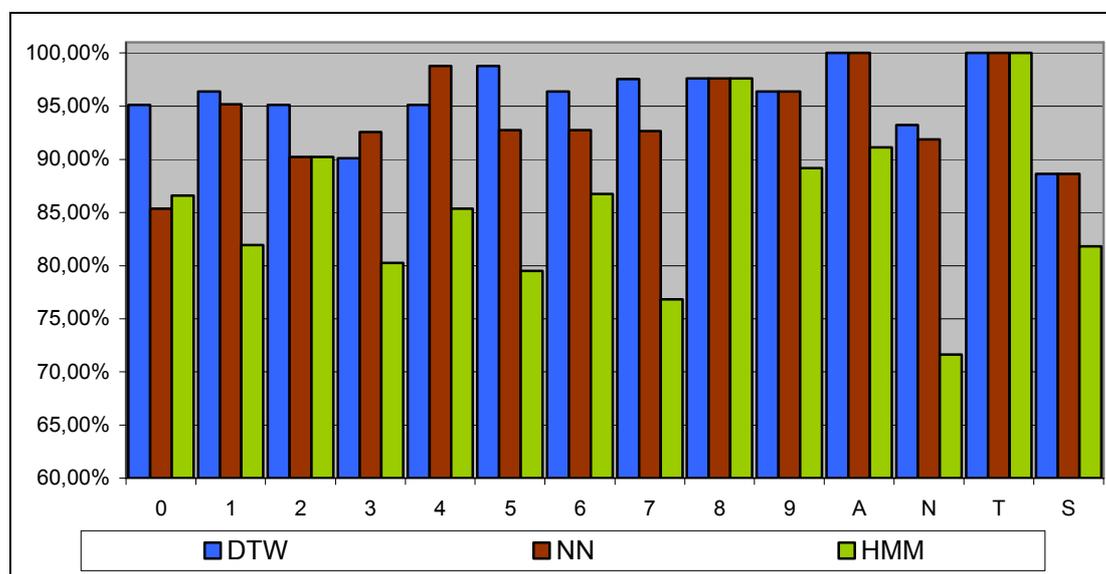
Porovnanie úspešností najúspešnejších klasifikátorov jednotlivých prístupov zobrazujeme v nasledujúcom grafe (Graf 5).

## Experiment



**Graf 5** - Porovnanie úspešností víťazných klasifikátorov jednotlivých prístupov na všetkých typoch nahrávok.

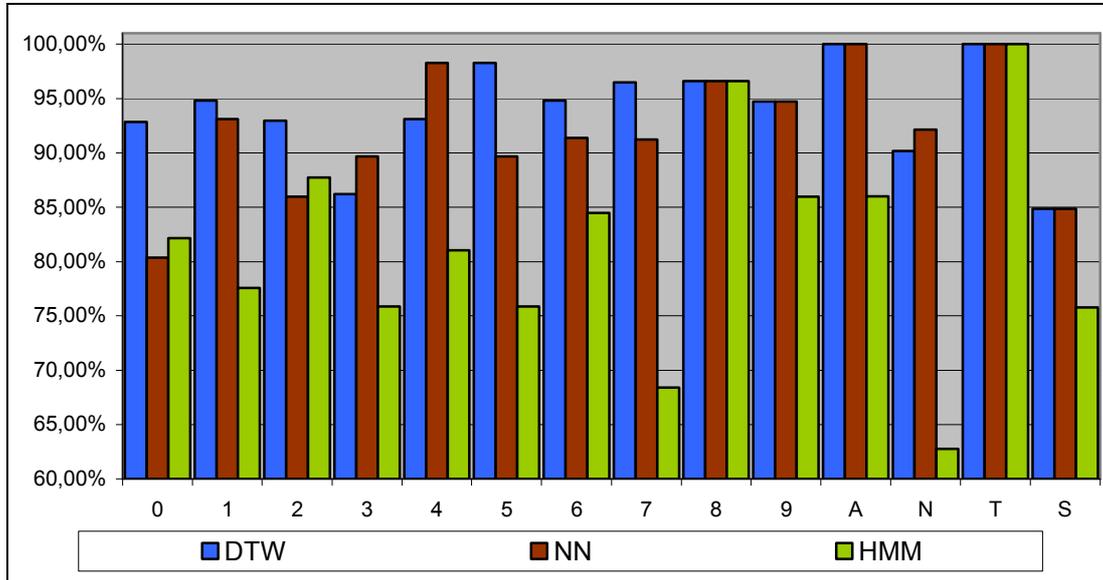
Na nasledujúcom grafe (Graf 6) prezentujeme porovnanie celkovej úspešnosti víťazných klasifikátorov jednotlivých prístupov na každom slove.



**Graf 6** - Porovnanie celkovej úspešnosti víťazov všetkých troch typov na jednotlivých slovách.

## Experiment

Pre lepšiu objektivitu a ukážku schopnosti generalizácie uvádzame ešte porovnanie úspešnosti víťazov na každom slove z nahrávok, z ktorých sme modely netrénovali (Graf 7).



**Graf 7** - Porovnanie úspešnosti rozpoznávania nenatrénovaných slov víťazov všetkých troch typov na jednotlivých slovách.

## 7.5 Interpretácia výsledkov

Výsledky DTW nás svojim víťazstvom neprekvapili. Keď ich porovnáme s výsledkami Rabinera a Juanga v [RJF], zistíme, že sme dosiahli vyššiu chybovosť - jeho systém mal chybovosť len 2% pri testovaní na nenatrénovaných rečníkoch. Pravdepodobne je to spôsobené kvalitou signálu.

Príjemne nás prekvapili výsledky úspešnosti neurónových sietí. Aj napriek ich nepoužívaniu na rozpoznávanie izolovaných slov majú úspešnosťou veľmi blízko DTW. Žiaľ tieto výsledky nemáme s čím porovnať, pretože literatúra, v ktorej by sa tento prístup skúmal, nám nebola dostupná. Všimli sme si však, že nami dosiahnuté hodnoty by mohli byť ešte lepšie, pretože sme pozorovali zlepšovanie siete s rastúcim počtom skrytých neurónov. Z časových dôvodov sme už nezaradili do tejto práce experimenty s vyšším počtom skrytých neurónov v skrytej vrstve. Nevylučujeme pri väčšom množstve tréningových dát ani takú možnosť naučenia siete, s ktorým by dokázala NN klasifikovať slová lepšie ako DTW, pretože si sama vyberie najvhodnejšiu funkciu g a mieru skreslenia.

## Experiment

---

Aj keď výsledná úspešnosť Markovovských modelov je v zhode s výsledkami v [HUE], je tento výsledok pre nás istým sklamaním, pretože Rabiner dosiahol pri svojich testoch úspešnosť len o 3.5% horšiu ako DTW pri testovaní na rovnakých rečníkoch. Spočiatku sme používali na vektorovú kvantizáciu 128 a 256 centier, čo odpovedá počtom centier v systémoch TANGORA a SPHINX, no neboli sme s týmito výsledkami spokojný. Preto sme použili neskôr 1024 centier, ako použil Rabiner vo svojich testoch [RJF]. Na naše sklamanie, nepomohla ani takáto úprava. Počet centier teda zrejme nie je dôvodom k horšej úspešnosti. Núka sa otázka, čo je týmto dôvodom. Jednou možnosťou je nevhodnosť použitia tohoto prístupu pri tejto kvalite signálu. Druhá možnosť môže byť v obmedzení tohoto prístupu na diskrétno rozdelenie pravdepodobnosti generovania výstupných symbolov. V [HUE] použili spojité rozdelenia pravdepodobnosti a použili parametre priamo z kompresie na výpočet kepstrálnych koeficientov, pri ktorých ukázali, že sú ako zdroj lepšie. Úspešnosť ich systému nepresiahla 90%. Naproti tomu Rabiner dosiahol pri použití spojitých rozdelení pravdepodobnosti chybovosť len 1.3%, no použil štandardný zdroj signálu. Preto druhú možnosť môžeme považovať za menej dôveryhodnú. Naskytá sa ešte jedno možné vysvetlenie - nedostatok tréningových dát. To však spochybňuje počet použitých tréningových dát v porovnaní s tým, čo použili Rabiner a Juang (použil vzorky od 100 rečníkov).

Venujme sa chvíľku úspešnosti rozpoznávania jednotlivých slov. Z grafov porovnávajúcich úspešnosti klasifikátorov na jednotlivých slovách (Graf 6 a Graf 7) vidieť, že nie všetky slová robili pri klasifikácii programom rovnaké ťažkosti. Asi najväčšie problémy mali všetky prístupy so slovom "Nie". Pripisujeme to tomu, že bolo vyslovované rôznymi rečníkmi značne odlišne. Všimnime si, že NN bola úspešnejšia na slovách "3" a "4" ako DTW. Vo všeobecnosti sa dá povedať, že jednotlivé programy mali ťažkosti s rôznymi slovami. Preto by bolo možno zaujímavé pozorovať ich v spolupráci.

Zaujímavé je aj pozorovanie tabuliek výsledkov v dodatkoch. Vo všeobecnosti sa dá povedať, že využitie všetkých piatich krokov pri úprave obrazu slova zlepšovalo úspešnosť jednotlivých rozpoznávačov. Ďalšou zaujímavosťou je použitie v zásade rôznych koeficientov víťaznými programami. Zamerajme sa ešte na našu uhlovú mieru skreslenia. Mala pomerne vysokú úspešnosť, no jej výpočtová náročnosť je príliš vysoká, a preto nie je vhodná na väčšie slovníky. Na druhej strane chyby robí na iných slovách, a preto môže byť výhodne využitá v kombinácii s iným prístupom.

## 7.6 Závěry

Z grafu 5 je možné odčítať víťaza celého experimentu. Je ním podľa očakávania DTW. Na prekvapivom druhom mieste sa umiestnili NN a nakoniec sklamaním<sup>31</sup> skočili HMM. Úspešnosť jednotlivých prístupov je pomerne slušná. Z porovnania

---

<sup>31</sup> aspoň podľa nášho názoru

## Experiment

---

výsledkov úspešnosti našich systémov so systémami v literatúre môžeme odhadnúť zvýšenie chybovosti spôsobené prenosom signálu systémom GSM. V našom prípade sa pohybuje okolo hranice 3%. Z uvedenej diskusie môžeme konštatovať, že kvalita signálu mobilného telefónu je postačujúca na rozpoznávanie rečových povelov, no nie v prípadoch, kde by systém rozoznával dôležité pokyny, alebo jednoduchšie, kde by už relatívne malá chybovosť mohla zapríčiniť vážne následky. Všimli sme si rôznu efektivitu jednotlivých modelov na jednotlivých slovách, preto by bolo nepochybne zaujímavé skúmať ich kooperatívnu architektúru. Spoločne by mohli dávať solídnu úspešnosť. Z prípravy a priebehu experimentu vyplýva, že nami navrhnutá metóda určovania hraníc slov bola postačujúca, no zďaleka nie optimálna.

Aj keď úspešnosť bola pre nás najdôležitejšia, spomeňme ešte výhody a nevýhody jednotlivých prístupov. Nevýhoda DTW spočíva v komplikovanosti rozšírenia prístupu na frázy a celé vety. Ďalším nie príliš príjemným faktom je jeho časová náročnosť, ktorú je možné do značnej miery zredukovať. Naopak výhodnosť je v jednoduchej implementácii a v možnosti celkom úspešného fungovania už pri jedinom slove v slovníku. Nevýhod prístupu NN je viacero, a preto sa zrejme tento prístup nepoužíva. Ťažko ho totiž rozšíriť na frázy a celé vety a už pri zmene, pridaní, alebo odobratí jediného slova zo slovníka je potrebné celý model trénovať odznova. Pri väčších slovníkoch by bolo treba zrejme viac skrytých neurónov, čo znamená veľkú priestorovú a v dôsledku toho aj časovú náročnosť. Naproti tomu výhoda je v celkom dobrej výkonnosti a priestorovej a časovej náročnosti pri menších slovníkoch. Výhod prístupu HMM je najviac. Ľahko sa ním dajú modelovať subslovné stavebné jednotky, ktoré potom možno spájať do slov a takto ich učiť, čím je možné vytvoriť systém rozpoznávajúci celé vety pri nie veľkých priestorových a časových nárokoch. Je robustný a flexibilný, čo viedlo aj k jeho širokému použitiu nielen v rozpoznávaní reči. Nepotrebuje špeciálne predspracovania obrazov slov a nie je natoľko náchylný na chyby pri zlom určení hraničných bodov slova. Jeho nevýhoda je však v pomerne vysokej časovej a algoritmickej náročnosti pri učení a malej schopnosti generalizácie - treba čo najviac trénovacích dát.

# Kapitola 8

## Záver

V úvodnej časti sme stručne prezentovali spôsoby spracovania signálu a určovania možných hraníc slov, kde sme uviedli vlastný spôsob, ktorý sa prispôsobuje intenzite šumu v okolí rečníka. V teoretickej časti o mierach odlišnosti sme sa popri popisovaní známych metrík pokúsili navrhnúť uhlovú mieru špeciálne pre DTW a výstup z pásmových filtrov, ktorú sme neskôr úspešne implementovali. Ukázali sme si, ako sa čiastočne vysporiadať s malými zmenami hlasu rečníka pomocou úpravy obrazov slov a pre úplnosť sme uviedli aj MacQueenov zhlukovací algoritmus na hľadanie centier vektorového kvantizéra.

V hlavnej časti teórie sme si z množstva existujúcich výpočtových modelov klasifikátorov izolovaných slov predstavili tri, ktorých úspešnosť sme sa rozhodli experimentálne porovnať. Prvý z nich, klasifikáciu pomocou dynamického programovania - DTW, sme považovali za najlepšieho kandidáta na klasifikáciu izolovaných slov. Markovovské modely sme na základe ich mohutnosti považovali za vážneho protivníka DTW. Za outsidera sme považovali neurónové siete, pretože sa na takýto problém takmer nepoužívajú a v literatúre ich označujú na rozpoznávanie reči za nevhodné.

V piatej kapitole sme popísali, ako pracuje systém GSM z hľadiska prenosu dát a prezentovali sme vplyv kompresie GSM na kepstrálne koeficienty. Následne sme predostreli možnosti skreslenia signálu touto kompresiou a ADPCM kompresiou vo výstupných súboroch Dialogic Vox.

V šiestej kapitole sme sa z viacerých strán venovali návrhu a popisu aplikácií, ktoré sme využili v neskoršom experimente.

Hlavnú časť tejto práce, prípravu a priebeh experimentu, sme popísali v siedmej kapitole. Výsledky v nás vyvolali zmiešané pocity. Najúspešnejší prístup bol v súlade s predpokladom DTW, no poradie ostatných dvoch prístupov bolo obrátené. Pôvodný outsider - neurónové siete - hrali rolu vážneho konkurenta DTW a robustné HMM zostali v úzadí, aj napriek nie zlému výsledku. Po zvážení výsledkov experimentu môžeme konštatovať, že kompresia GSM a prenos signálu vzduchom negatívne ovplyvňujú úspešnosť klasifikácie slov.

Záverom ešte poznamenajme, že nami získané výsledky sú do značnej miery závislé od kvality implementácie, a preto heslovite spomenieme ešte rôzne vylepšenia a ďalšie námety pre jednotlivé prístupy:

## Záver

---

- aplikovanie iných predspracovaní signálu vo všetkých prístupoch
- využitie diskriminatívneho učenia pri tréovaní HMM
- použitie HMM so spjitými rozdeleniami pravdepodobností, alebo HMM generujúcich symboly na hranách, príp. ich modifikácií
- využitie heuristik pri hľadaní optimálnej cesty na urýchlenie DTW, zmenšenie počtu vzoriek pre jedno slovo v slovníku pomocou MacQueenovho zhlukovacieho algoritmu, atď.
- urýchlenie NN sofistikovanejšími algoritmami alebo lepším nastavením parametrov
- zvýšenie úspešnosti NN viacerými neurónmi v skrytej vrstve
- použitie iných výpočtových modelov neurónových sietí (napr. rekurentných) pri modelovaní časového priebehu slov

Dúfame, že táto práca prispeje k rozvoju umelej inteligencie a špeciálne k napredovaniu výskumu rozpoznávania reči na FMFI UK.

# Dodatky

## A. Trénovací papier na rozpoznávanie izolovaných slov

Táto akcia slúži na zber hlasových vzoriek od bežných ľudí. Získané hlasové vzorky budú použité v mojej diplomovej práci. Preto prosím prečítajte tento text POZORNE a držte sa pokynov v ňom napísaných.

### Časť 1.:

Prečítajte nahlas nasledovné telefónne čísla po jednotlivých číslicach: (Robte medzi vyslovovaním číslic malé - asi sekundové – pauzy)

0934 85 61 27  
421 7 50 69 38  
0815 34 76 92  
507 9 42 18 63

### Časť 2.:

Spontánne odpovedajte na nasledujúce otázky. Najprv si otázku potichu prečítajte a následne odpovedzte „Áno“ lebo „Nie“. Odpovedajte pokiaľ možno pravdivo.

1. Narodili ste sa v Austrálii?
2. Chodili ste (chodíte) na základnú školu?
3. Hovoríte slovensky?
4. Žije niekto z vašej rodiny v Mongolsku?
5. Ste ženského pohlavia?
6. Pijete kávu?
7. Vlastníte mobil?
8. Pozeráte v televízii telenovely?

A teraz ukončíte hovor (stlačte na telefóne tlačidlo No).

Ďakujem Vám

Jaroslav ONDRISKA

## B. Výsledky prvých 100 DTW klasifikátorov

Tabuľka 10 - Úspešnosť prvých 100 najlepších rozpoznávačov DTW. Názvy analýz, mier skreslenia a úprav obrazov slov sú v angličtine.

Kód klasifikátora	Dĺžka preškáľovaného obrazu slova	Typ analýzy	Miera skreslenia	Úpravy obrazov slov	Šírka prehľadávaného priestoru	Počet najbližších susedov v k-NN	Chybovosť netréňovaných	Chybovosť tréňovaných	Chybovosť rovnakého rečníka	Chyby celkovo	Celková úspešnosť v %	Úspešnosť v % netréňovaných	Úspešnosť v % tréňovaných	Úspešnosť v % rovnakého rečníka
12-2-3-3-0-3	12	CepLPC	Cepstral	All	0	3	44	0	0	44	95,88	94,06	100,00	100,00
20-2-4-3-0-3	20	CepLPC	WCepst.	All	0	3	46	0	0	46	95,69	93,79	100,00	100,00
12-2-3-3-1-3	12	CepLPC	Cepstral	All	1	3	47	0	0	47	95,60	93,66	100,00	100,00
12-2-4-3-0-3	12	CepLPC	WCepst.	All	0	3	47	0	0	47	95,60	93,66	100,00	100,00
12-2-4-3-0-5	12	CepLPC	WCepst.	All	0	5	47	0	0	47	95,60	93,66	100,00	100,00
12-2-3-3-1-5	12	CepLPC	Cepstral	All	1	5	49	0	0	49	95,41	93,39	100,00	100,00
12-2-3-3-1-7	12	CepLPC	Cepstral	All	1	7	49	0	0	49	95,41	93,39	100,00	100,00
12-2-4-3-0-7	12	CepLPC	WCepst.	All	0	7	49	1	0	50	95,31	93,39	99,68	100,00
24-1-1-3-1-5	24	Fourier	Euklid	All	1	5	49	1	0	50	95,31	93,39	99,68	100,00
12-2-3-3-0-5	12	CepLPC	Cepstral	All	0	5	51	0	0	51	95,22	93,12	100,00	100,00
12-2-3-3-2-3	12	CepLPC	Cepstral	All	2	3	51	0	0	51	95,22	93,12	100,00	100,00
16-2-3-3-1-3	16	CepLPC	Cepstral	All	1	3	51	0	0	51	95,22	93,12	100,00	100,00
20-2-3-3-0-3	20	CepLPC	Cepstral	All	0	3	51	0	0	51	95,22	93,12	100,00	100,00
20-2-3-3-1-3	20	CepLPC	Cepstral	All	1	3	51	0	0	51	95,22	93,12	100,00	100,00
20-2-3-3-2-5	20	CepLPC	Cepstral	All	2	5	51	0	0	51	95,22	93,12	100,00	100,00
20-2-4-3-0-5	20	CepLPC	WCepst.	All	0	5	51	0	0	51	95,22	93,12	100,00	100,00
24-1-1-3-1-3	24	Fourier	Euklid	All	1	3	51	0	0	51	95,22	93,12	100,00	100,00
24-1-1-3-1-7	24	Fourier	Euklid	All	1	7	50	1	0	51	95,22	93,25	99,68	100,00
24-1-1-3-1-9	24	Fourier	Euklid	All	1	9	50	1	0	51	95,22	93,25	99,68	100,00
12-2-3-3-2-5	12	CepLPC	Cepstral	All	2	5	52	0	0	52	95,13	92,98	100,00	100,00
12-2-4-3-1-3	12	CepLPC	WCepst.	All	1	3	52	0	0	52	95,13	92,98	100,00	100,00
20-2-3-3-0-5	20	CepLPC	Cepstral	All	0	5	52	0	0	52	95,13	92,98	100,00	100,00
20-2-3-3-2-7	20	CepLPC	Cepstral	All	2	7	52	0	0	52	95,13	92,98	100,00	100,00
16-2-3-3-1-5	16	CepLPC	Cepstral	All	1	5	53	0	0	53	95,03	92,85	100,00	100,00
20-2-4-3-0-1	20	CepLPC	WCepst.	All	0	1	53	0	0	53	95,03	92,85	100,00	100,00
12-2-4-3-0-9	12	CepLPC	WCepst.	All	0	9	53	1	0	54	94,94	92,85	99,68	100,00
12-2-4-3-1-5	12	CepLPC	WCepst.	All	1	5	54	0	0	54	94,94	92,71	100,00	100,00
20-1-1-3-1-5	20	Fourier	Euklid	All	1	5	53	1	0	54	94,94	92,85	99,68	100,00
20-2-3-3-0-7	20	CepLPC	Cepstral	All	0	7	54	0	0	54	94,94	92,71	100,00	100,00
20-2-3-3-1-5	20	CepLPC	Cepstral	All	1	5	54	0	0	54	94,94	92,71	100,00	100,00
12-1-2-3-0-5	12	Fourier	Angle	All	0	5	54	1	0	55	94,85	92,71	99,68	100,00
12-2-3-3-0-7	12	CepLPC	Cepstral	All	0	7	54	1	0	55	94,85	92,71	99,68	100,00
16-2-3-3-0-3	16	CepLPC	Cepstral	All	0	3	55	0	0	55	94,85	92,58	100,00	100,00
16-2-3-3-0-5	16	CepLPC	Cepstral	All	0	5	55	0	0	55	94,85	92,58	100,00	100,00
20-1-1-3-1-7	20	Fourier	Euklid	All	1	7	54	1	0	55	94,85	92,71	99,68	100,00
12-1-2-3-0-7	12	Fourier	Angle	All	0	7	55	1	0	56	94,75	92,58	99,68	100,00
12-2-4-3-1-7	12	CepLPC	WCepst.	All	1	7	56	0	0	56	94,75	92,44	100,00	100,00
16-2-3-3-0-7	16	CepLPC	Cepstral	All	0	7	56	0	0	56	94,75	92,44	100,00	100,00
16-2-3-3-2-3	16	CepLPC	Cepstral	All	2	3	56	0	0	56	94,75	92,44	100,00	100,00
16-2-3-3-2-5	16	CepLPC	Cepstral	All	2	5	56	0	0	56	94,75	92,44	100,00	100,00
16-2-4-3-0-3	16	CepLPC	WCepst.	All	0	3	56	0	0	56	94,75	92,44	100,00	100,00

Kód klasifikátora	Dĺžka preškátovaného obrazu slova	Typ analýzy	Miera skreslenia	Úpravy obrazov slov	Sírka prehľadávaného priestoru	Počet najbližších susedov v k-NN	Chybovosť netrénovaných	Chybovosť trénovaných	Chybovosť rovnakého rečníka	Chýb celkovo	Celková úspešnosť v %	Úspešnosť v % netrénovaných	Úspešnosť v % trénovaných	Úspešnosť v % rovnakého rečníka
16-2-4-3-1-3	16	CepLPC	WCepst.	All	1	3	56	0	0	56	94,75	92,44	100,00	100,00
20-1-1-3-1-9	20	Fourier	Euklid	All	1	9	55	1	0	56	94,75	92,58	99,68	100,00
20-1-2-3-1-5	20	Fourier	Angle	All	1	5	55	1	0	56	94,75	92,58	99,68	100,00
20-1-2-3-1-7	20	Fourier	Angle	All	1	7	55	1	0	56	94,75	92,58	99,68	100,00
20-2-3-3-2-3	20	CepLPC	Cepstral	All	2	3	56	0	0	56	94,75	92,44	100,00	100,00
20-2-3-3-3-3	20	CepLPC	Cepstral	All	3	3	56	0	0	56	94,75	92,44	100,00	100,00
12-1-2-3-0-3	12	Fourier	Angle	All	0	3	57	0	0	57	94,66	92,31	100,00	100,00
12-2-3-3-1-1	12	CepLPC	Cepstral	All	1	1	57	0	0	57	94,66	92,31	100,00	100,00
12-2-4-3-2-3	12	CepLPC	WCepst.	All	2	3	57	0	0	57	94,66	92,31	100,00	100,00
16-2-3-3-1-7	16	CepLPC	Cepstral	All	1	7	57	0	0	57	94,66	92,31	100,00	100,00
16-2-3-3-2-1	16	CepLPC	Cepstral	All	2	1	57	0	0	57	94,66	92,31	100,00	100,00
16-2-4-3-0-1	16	CepLPC	WCepst.	All	0	1	57	0	0	57	94,66	92,31	100,00	100,00
16-2-4-3-1-1	16	CepLPC	WCepst.	All	1	1	57	0	0	57	94,66	92,31	100,00	100,00
20-2-3-3-0-1	20	CepLPC	Cepstral	All	0	1	57	0	0	57	94,66	92,31	100,00	100,00
20-2-3-3-3-5	20	CepLPC	Cepstral	All	3	5	57	0	0	57	94,66	92,31	100,00	100,00
24-1-1-3-2-5	24	Fourier	Euklid	All	2	5	56	1	0	57	94,66	92,44	99,68	100,00
12-2-3-3-1-9	12	CepLPC	Cepstral	All	1	9	57	1	0	58	94,56	92,31	99,68	100,00
12-2-3-3-2-7	12	CepLPC	Cepstral	All	2	7	57	1	0	58	94,56	92,31	99,68	100,00
16-2-3-3-0-9	16	CepLPC	Cepstral	All	0	9	57	1	0	58	94,56	92,31	99,68	100,00
16-2-3-3-3-3	16	CepLPC	Cepstral	All	3	3	58	0	0	58	94,56	92,17	100,00	100,00
16-2-4-3-0-5	16	CepLPC	WCepst.	All	0	5	58	0	0	58	94,56	92,17	100,00	100,00
20-1-1-3-0-5	20	Fourier	Euklid	All	0	5	56	2	0	58	94,56	92,44	99,35	100,00
20-1-1-3-1-3	20	Fourier	Euklid	All	1	3	57	1	0	58	94,56	92,31	99,68	100,00
20-1-2-3-1-3	20	Fourier	Angle	All	1	3	57	1	0	58	94,56	92,31	99,68	100,00
20-1-2-3-1-9	20	Fourier	Angle	All	1	9	57	1	0	58	94,56	92,31	99,68	100,00
20-2-3-3-0-9	20	CepLPC	Cepstral	All	0	9	57	1	0	58	94,56	92,31	99,68	100,00
20-2-4-3-0-7	20	CepLPC	WCepst.	All	0	7	58	0	0	58	94,56	92,17	100,00	100,00
20-2-4-3-1-1	20	CepLPC	WCepst.	All	1	1	58	0	0	58	94,56	92,17	100,00	100,00
24-1-1-3-0-7	24	Fourier	Euklid	All	0	7	56	2	0	58	94,56	92,44	99,35	100,00
12-2-3-3-0-9	12	CepLPC	Cepstral	All	0	9	56	3	0	59	94,47	92,44	99,03	100,00
12-2-4-3-1-9	12	CepLPC	WCepst.	All	1	9	58	1	0	59	94,47	92,17	99,68	100,00
16-2-3-3-1-1	16	CepLPC	Cepstral	All	1	1	59	0	0	59	94,47	92,04	100,00	100,00
16-2-4-3-1-5	16	CepLPC	WCepst.	All	1	5	59	0	0	59	94,47	92,04	100,00	100,00
20-1-2-3-0-3	20	Fourier	Angle	All	0	3	59	0	0	59	94,47	92,04	100,00	100,00
20-1-2-3-0-5	20	Fourier	Angle	All	0	5	58	1	0	59	94,47	92,17	99,68	100,00
20-2-3-3-2-9	20	CepLPC	Cepstral	All	2	9	59	0	0	59	94,47	92,04	100,00	100,00
20-2-3-3-3-7	20	CepLPC	Cepstral	All	3	7	59	0	0	59	94,47	92,04	100,00	100,00
20-2-4-3-2-5	20	CepLPC	WCepst.	All	2	5	59	0	0	59	94,47	92,04	100,00	100,00
24-1-1-3-0-5	24	Fourier	Euklid	All	0	5	57	2	0	59	94,47	92,31	99,35	100,00
24-1-1-3-2-3	24	Fourier	Euklid	All	2	3	59	0	0	59	94,47	92,04	100,00	100,00
12-2-4-3-1-1	12	CepLPC	WCepst.	All	1	1	60	0	0	60	94,38	91,90	100,00	100,00
16-2-3-3-2-7	16	CepLPC	Cepstral	All	2	7	60	0	0	60	94,38	91,90	100,00	100,00
16-2-3-3-3-5	16	CepLPC	Cepstral	All	3	5	60	0	0	60	94,38	91,90	100,00	100,00
20-1-1-3-0-7	20	Fourier	Euklid	All	0	7	58	2	0	60	94,38	92,17	99,35	100,00
20-2-3-3-1-1	20	CepLPC	Cepstral	All	1	1	60	0	0	60	94,38	91,90	100,00	100,00
20-2-3-3-3-9	20	CepLPC	Cepstral	All	3	9	60	0	0	60	94,38	91,90	100,00	100,00
24-1-1-3-2-7	24	Fourier	Euklid	All	2	7	59	1	0	60	94,38	92,04	99,68	100,00
12-1-2-3-0-9	12	Fourier	Angle	All	0	9	60	1	0	61	94,28	91,90	99,68	100,00

Kód klasifikátora	Dĺžka preškáľovaného obrazu slova	Typ analýzy	Miera skreslenia	Úpravy obrazov slov	Šírka prehľadávaného priestoru	Počet najbližších susedov v k-NN	Chybovosť netrénovaných	Chybovosť trénovaných	Chybovosť rovnnakého rečníka	Chýb celkovo	Celková úspešnosť v %	Úspešnosť v % netrénovaných	Úspešnosť v % trénovaných	Úspešnosť v % rovnnakého rečníka
12-2-4-3-0-1	12	CepLPC	WCepst.	All	0	1	61	0	0	61	94,28	91,77	100,00	100,00
12-2-4-3-2-5	12	CepLPC	WCepst.	All	2	5	61	0	0	61	94,28	91,77	100,00	100,00
16-2-4-3-2-3	16	CepLPC	WCepst.	All	2	3	61	0	0	61	94,28	91,77	100,00	100,00
16-2-4-3-2-9	16	CepLPC	WCepst.	All	2	9	61	0	0	61	94,28	91,77	100,00	100,00
20-1-2-3-0-7	20	Fourier	Angle	All	0	7	60	1	0	61	94,28	91,90	99,68	100,00
20-2-3-3-1-7	20	CepLPC	Cepstral	All	1	7	61	0	0	61	94,28	91,77	100,00	100,00
20-2-3-3-4-3	20	CepLPC	Cepstral	All	4	3	61	0	0	61	94,28	91,77	100,00	100,00
12-2-3-3-0-1	12	CepLPC	Cepstral	All	0	1	62	0	0	62	94,19	91,63	100,00	100,00
12-2-3-3-2-9	12	CepLPC	Cepstral	All	2	9	61	1	0	62	94,19	91,77	99,68	100,00
16-1-2-3-0-7	16	Fourier	Angle	All	0	7	61	1	0	62	94,19	91,77	99,68	100,00

**Tabuľka 11** - Úspešnosť rozpoznávania jednotlivých slov prvých 100 najúspešnejších rozpoznávačov DTW. Sú zoradené podľa celkovej úspešnosti.

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
12-2-3-3-0-3	95,12	96,39	95,12	90,12	95,12	98,80	96,39	97,56	97,62	96,39	100,00	93,24	100,00	88,64	95,88
20-2-4-3-0-3	93,90	93,98	97,56	90,12	95,12	98,80	96,39	96,34	96,43	97,59	100,00	94,59	100,00	86,36	95,69
12-2-3-3-1-3	96,34	93,98	96,34	90,12	93,90	98,80	95,18	96,34	98,81	93,98	98,73	94,59	100,00	90,91	95,60
12-2-4-3-0-3	92,68	96,39	95,12	92,59	95,12	98,80	95,18	96,34	96,43	95,18	100,00	93,24	100,00	90,91	95,60
12-2-4-3-0-5	95,12	93,98	95,12	93,83	93,90	98,80	95,18	96,34	95,24	95,18	100,00	94,59	100,00	90,91	95,60
12-2-3-3-1-5	96,34	92,77	96,34	90,12	93,90	98,80	95,18	92,68	98,81	93,98	98,73	95,95	100,00	93,18	95,41
12-2-3-3-1-7	96,34	93,98	96,34	88,89	92,68	97,59	95,18	92,68	98,81	95,18	98,73	95,95	100,00	95,45	95,41
12-2-4-3-0-7	93,90	92,77	93,90	92,59	93,90	98,80	93,98	96,34	95,24	96,39	100,00	94,59	100,00	93,18	95,31
24-1-1-3-1-5	86,59	96,39	91,46	91,36	95,12	98,80	93,98	100,00	97,62	95,18	100,00	95,95	100,00	93,18	95,31
12-2-3-3-0-5	92,68	96,39	95,12	87,65	95,12	98,80	96,39	97,56	97,62	96,39	100,00	93,24	100,00	81,82	95,22
12-2-3-3-2-3	95,12	95,18	95,12	85,19	93,90	95,18	95,18	95,12	100,00	95,18	100,00	95,95	100,00	93,18	95,22
16-2-3-3-1-3	96,34	93,98	96,34	85,19	96,34	98,80	95,18	95,12	97,62	93,98	98,73	95,95	100,00	88,64	95,22
20-2-3-3-0-3	96,34	95,18	95,12	88,89	95,12	97,59	95,18	96,34	97,62	97,59	100,00	94,59	100,00	77,27	95,22
20-2-3-3-1-3	93,90	95,18	96,34	86,42	95,12	98,80	93,98	95,12	97,62	96,39	98,73	95,95	100,00	88,64	95,22
20-2-3-3-2-5	93,90	96,39	96,34	86,42	91,46	100,00	93,98	95,12	97,62	95,18	97,47	95,95	100,00	95,45	95,22
20-2-4-3-0-5	95,12	92,77	95,12	91,36	93,90	98,80	96,39	93,90	96,43	97,59	100,00	94,59	100,00	84,09	95,22
24-1-1-3-1-3	87,80	95,18	91,46	91,36	95,12	98,80	93,98	100,00	97,62	93,98	100,00	95,95	100,00	93,18	95,22
24-1-1-3-1-7	86,59	96,39	91,46	91,36	93,90	98,80	93,98	100,00	97,62	95,18	100,00	94,59	100,00	95,45	95,22
24-1-1-3-1-9	85,37	97,59	91,46	92,59	93,90	97,59	93,98	98,78	97,62	96,39	100,00	94,59	100,00	95,45	95,22
12-2-3-3-2-5	95,12	95,18	96,34	85,19	92,68	96,39	95,18	92,68	100,00	95,18	98,73	95,95	100,00	95,45	95,13
12-2-4-3-1-3	92,68	93,98	95,12	90,12	92,68	98,80	95,18	95,12	98,81	93,98	98,73	93,24	100,00	95,45	95,13
20-2-3-3-0-5	95,12	93,98	95,12	90,12	96,34	98,80	95,18	95,12	97,62	96,39	98,73	94,59	100,00	79,55	95,13
20-2-3-3-2-7	93,90	96,39	97,56	85,19	91,46	100,00	93,98	93,90	97,62	95,18	97,47	95,95	100,00	95,45	95,13
16-2-3-3-1-5	96,34	92,77	96,34	85,19	95,12	98,80	93,98	95,12	96,43	95,18	98,73	95,95	100,00	90,91	95,03
20-2-4-3-0-1	92,68	96,39	96,34	93,83	95,12	98,80	93,98	93,90	96,43	96,39	97,47	91,89	100,00	84,09	95,03
12-2-4-3-0-9	93,90	92,77	93,90	92,59	95,12	98,80	93,98	95,12	95,24	95,18	98,73	93,24	100,00	90,91	94,94
12-2-4-3-1-5	91,46	93,98	96,34	88,89	92,68	98,80	95,18	92,68	98,81	92,77	98,73	95,95	100,00	95,45	94,94
20-1-1-3-1-5	82,93	97,59	92,68	90,12	93,90	98,80	93,98	100,00	97,62	95,18	100,00	94,59	100,00	93,18	94,94
20-2-3-3-0-7	95,12	93,98	93,90	91,36	93,90	98,80	95,18	95,12	97,62	95,18	97,47	94,59	100,00	84,09	94,94
20-2-3-3-1-5	93,90	93,98	96,34	85,19	95,12	98,80	92,77	95,12	97,62	96,39	98,73	95,95	100,00	88,64	94,94
12-1-2-3-0-5	81,71	96,39	91,46	95,06	95,12	100,00	96,39	96,34	97,62	97,59	100,00	93,24	100,00	84,09	94,85
12-2-3-3-0-7	95,12	93,98	95,12	87,65	95,12	98,80	96,39	93,90	96,43	95,18	100,00	93,24	100,00	84,09	94,85

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
16-2-3-3-0-3	92,68	93,98	93,90	90,12	96,34	97,59	95,18	96,34	97,62	95,18	100,00	94,59	100,00	79,55	94,85
16-2-3-3-0-5	93,90	92,77	96,34	90,12	95,12	97,59	95,18	93,90	97,62	95,18	100,00	94,59	100,00	81,82	94,85
20-1-1-3-1-7	81,71	97,59	92,68	91,36	93,90	97,59	92,77	100,00	97,62	95,18	100,00	94,59	100,00	95,45	94,85
12-1-2-3-0-7	82,93	95,18	90,24	95,06	95,12	100,00	96,39	96,34	97,62	97,59	100,00	93,24	100,00	84,09	94,75
12-2-4-3-1-7	92,68	93,98	96,34	87,65	92,68	98,80	93,98	92,68	98,81	91,57	98,73	95,95	100,00	95,45	94,75
16-2-3-3-0-7	95,12	92,77	95,12	90,12	93,90	97,59	93,98	93,90	97,62	95,18	100,00	94,59	100,00	84,09	94,75
16-2-3-3-2-3	96,34	95,18	93,90	85,19	93,90	97,59	95,18	93,90	98,81	91,57	98,73	95,95	100,00	90,91	94,75
16-2-3-3-2-5	96,34	93,98	95,12	86,42	91,46	97,59	95,18	93,90	97,62	92,77	97,47	95,95	100,00	95,45	94,75
16-2-4-3-0-3	90,24	93,98	96,34	88,89	95,12	97,59	95,18	96,34	96,43	96,39	100,00	94,59	100,00	81,82	94,75
16-2-4-3-1-3	91,46	93,98	97,56	90,12	92,68	97,59	93,98	95,12	96,43	91,57	98,73	94,59	100,00	95,45	94,75
20-1-1-3-1-9	80,49	97,59	92,68	91,36	93,90	97,59	92,77	100,00	97,62	95,18	100,00	94,59	100,00	95,45	94,75
20-1-2-3-1-5	82,93	97,59	92,68	86,42	95,12	98,80	93,98	100,00	97,62	95,18	100,00	95,95	100,00	90,91	94,75
20-1-2-3-1-7	82,93	97,59	93,90	87,65	95,12	98,80	92,77	100,00	97,62	95,18	100,00	95,95	100,00	88,64	94,75
20-2-3-3-2-3	92,68	96,39	95,12	86,42	90,24	98,80	95,18	95,12	97,62	95,18	97,47	95,95	100,00	90,91	94,75
20-2-3-3-3-3	93,90	95,18	96,34	88,89	90,24	98,80	95,18	93,90	98,81	91,57	96,20	95,95	100,00	93,18	94,75
12-1-2-3-0-3	84,15	96,39	92,68	92,59	95,12	100,00	93,98	96,34	97,62	97,59	100,00	91,89	100,00	84,09	94,66
12-2-3-3-1-1	92,68	95,18	95,12	87,65	92,68	98,80	95,18	95,12	97,62	95,18	98,73	91,89	100,00	88,64	94,66
12-2-4-3-2-3	93,90	95,18	96,34	83,95	92,68	95,18	96,39	93,90	97,62	95,18	97,47	94,59	100,00	95,45	94,66
16-2-3-3-1-7	95,12	92,77	96,34	85,19	95,12	98,80	92,77	93,90	96,43	93,98	98,73	95,95	100,00	90,91	94,66
16-2-3-3-2-1	93,90	96,39	93,90	87,65	95,12	96,39	96,39	92,68	98,81	92,77	100,00	93,24	100,00	86,36	94,66
16-2-4-3-0-1	90,24	97,59	93,90	88,89	96,34	98,80	96,39	95,12	96,43	95,18	98,73	93,24	100,00	79,55	94,66
16-2-4-3-1-1	90,24	95,18	96,34	90,12	92,68	97,59	95,18	97,56	96,43	91,57	100,00	93,24	100,00	88,64	94,66
20-2-3-3-0-1	91,46	95,18	95,12	93,83	96,34	96,39	93,98	96,34	97,62	96,39	96,20	94,59	100,00	75,00	94,66
20-2-3-3-3-5	92,68	95,18	96,34	90,12	91,46	97,59	93,98	92,68	97,62	92,77	96,20	95,95	100,00	95,45	94,66
24-1-1-3-2-5	82,93	96,39	93,90	85,19	91,46	98,80	93,98	100,00	97,62	96,39	100,00	95,95	100,00	95,45	94,66
12-2-3-3-1-9	95,12	92,77	96,34	86,42	91,46	96,39	93,98	92,68	97,62	95,18	98,73	95,95	100,00	93,18	94,56
12-2-3-3-2-7	93,90	92,77	95,12	85,19	91,46	96,39	95,18	92,68	98,81	95,18	98,73	95,95	100,00	95,45	94,56
16-2-3-3-0-9	96,34	91,57	95,12	90,12	93,90	97,59	93,98	93,90	96,43	95,18	98,73	94,59	100,00	84,09	94,56
16-2-3-3-3-3	95,12	96,39	93,90	85,19	92,68	97,59	95,18	93,90	98,81	90,36	97,47	95,95	100,00	93,18	94,56
16-2-4-3-0-5	92,68	92,77	95,12	88,89	95,12	97,59	93,98	93,90	95,24	97,59	100,00	94,59	100,00	84,09	94,56
20-1-1-3-0-5	87,80	95,18	90,24	95,06	96,34	96,39	91,57	97,56	97,62	96,39	100,00	91,89	97,78	88,64	94,56
20-1-1-3-1-3	84,15	97,59	90,24	86,42	93,90	98,80	93,98	100,00	97,62	93,98	100,00	95,95	100,00	93,18	94,56
20-1-2-3-1-3	82,93	97,59	92,68	85,19	95,12	98,80	93,98	100,00	97,62	95,18	100,00	95,95	100,00	88,64	94,56
20-1-2-3-1-9	80,49	97,59	93,90	87,65	93,90	98,80	92,77	100,00	97,62	95,18	100,00	95,95	100,00	90,91	94,56
20-2-3-3-0-9	96,34	91,57	93,90	91,36	93,90	97,59	95,18	95,12	96,43	93,98	97,47	94,59	100,00	84,09	94,56
20-2-4-3-0-7	92,68	92,77	95,12	91,36	92,68	97,59	95,18	93,90	95,24	95,18	100,00	94,59	100,00	86,36	94,56
20-2-4-3-1-1	89,02	95,18	95,12	88,89	92,68	97,59	95,18	95,12	96,43	93,98	100,00	95,95	100,00	88,64	94,56
24-1-1-3-0-7	87,80	96,39	90,24	93,83	96,34	96,39	91,57	96,34	97,62	96,39	100,00	94,59	95,56	88,64	94,56
12-2-3-3-0-9	95,12	93,98	95,12	87,65	95,12	98,80	95,18	92,68	96,43	93,98	98,73	93,24	100,00	84,09	94,47
12-2-4-3-1-9	91,46	92,77	97,56	87,65	91,46	98,80	93,98	92,68	97,62	90,36	98,73	95,95	100,00	97,73	94,47
16-2-3-3-1-1	91,46	95,18	96,34	83,95	96,34	97,59	95,18	95,12	97,62	93,98	100,00	91,89	100,00	86,36	94,47
16-2-4-3-1-5	91,46	92,77	97,56	88,89	91,46	98,80	91,57	96,34	96,43	90,36	97,47	95,95	100,00	97,73	94,47
20-1-2-3-0-3	85,37	96,39	90,24	91,36	95,12	100,00	93,98	96,34	97,62	96,39	100,00	93,24	100,00	84,09	94,47
20-1-2-3-0-5	82,93	96,39	90,24	93,83	95,12	100,00	93,98	96,34	97,62	96,39	100,00	93,24	100,00	84,09	94,47
20-2-3-3-2-9	95,12	95,18	97,56	85,19	91,46	98,80	93,98	92,68	96,43	91,57	97,47	95,95	100,00	93,18	94,47
20-2-3-3-3-7	92,68	96,39	96,34	90,12	90,24	97,59	93,98	91,46	97,62	91,57	96,20	95,95	100,00	95,45	94,47
20-2-4-3-2-5	90,24	95,18	97,56	88,89	90,24	98,80	93,98	93,90	95,24	93,98	96,20	95,95	100,00	95,45	94,47
24-1-1-3-0-5	87,80	95,18	90,24	93,83	96,34	96,39	91,57	97,56	97,62	97,59	100,00	91,89	95,56	88,64	94,47
24-1-1-3-2-3	82,93	97,59	95,12	82,72	91,46	98,80	93,98	100,00	97,62	95,18	100,00	95,95	100,00	93,18	94,47
12-2-4-3-1-1	93,90	92,77	96,34	90,12	90,24	96,39	93,98	93,90	96,43	93,98	98,73	91,89	100,00	95,45	94,38
16-2-3-3-2-7	96,34	93,98	95,12	85,19	91,46	97,59	93,98	92,68	96,43	92,77	97,47	95,95	100,00	95,45	94,38
16-2-3-3-3-5	95,12	95,18	95,12	86,42	90,24	97,59	95,18	93,90	97,62	89,16	97,47	95,95	100,00	95,45	94,38
20-1-1-3-0-7	86,59	96,39	90,24	95,06	95,12	97,59	91,57	97,56	97,62	95,18	100,00	91,89	97,78	86,36	94,38
20-2-3-3-1-1	90,24	93,98	95,12	88,89	95,12	98,80	95,18	95,12	97,62	93,98	98,73	90,54	100,00	86,36	94,38

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
20-2-3-3-3-9	92,68	96,39	96,34	90,12	90,24	98,80	93,98	91,46	97,62	89,16	96,20	95,95	100,00	95,45	94,38
24-1-1-3-2-7	79,27	96,39	93,90	85,19	91,46	98,80	93,98	100,00	97,62	96,39	100,00	95,95	100,00	95,45	94,38
12-1-2-3-0-9	80,49	95,18	90,24	93,83	95,12	100,00	95,18	96,34	96,43	97,59	100,00	93,24	100,00	84,09	94,28
12-2-4-3-0-1	89,02	96,39	92,68	93,83	96,34	97,59	95,18	93,90	96,43	93,98	97,47	90,54	100,00	84,09	94,28
12-2-4-3-2-5	95,12	93,98	95,12	85,19	91,46	96,39	93,98	91,46	98,81	92,77	97,47	95,95	100,00	95,45	94,28
16-2-4-3-2-3	91,46	95,18	95,12	86,42	91,46	98,80	93,98	93,90	96,43	90,36	97,47	95,95	100,00	97,73	94,28
16-2-4-3-2-9	93,90	93,98	97,56	88,89	91,46	98,80	91,57	92,68	95,24	91,57	94,94	95,95	100,00	97,73	94,28
20-1-2-3-0-7	82,93	96,39	90,24	92,59	95,12	100,00	92,77	96,34	97,62	96,39	100,00	93,24	100,00	84,09	94,28
20-2-3-3-1-7	95,12	92,77	96,34	85,19	93,90	98,80	92,77	92,68	97,62	91,57	98,73	95,95	100,00	88,64	94,28
20-2-3-3-4-3	92,68	95,18	96,34	90,12	91,46	97,59	95,18	91,46	96,43	90,36	94,94	94,59	100,00	97,73	94,28
12-2-3-3-0-1	86,59	96,39	96,34	92,59	95,12	97,59	93,98	96,34	97,62	93,98	96,20	89,19	100,00	84,09	94,19
12-2-3-3-2-9	95,12	91,57	95,12	85,19	91,46	96,39	93,98	91,46	98,81	93,98	98,73	94,59	100,00	95,45	94,19
16-1-2-3-0-7	81,71	96,39	90,24	91,36	95,12	100,00	93,98	95,12	97,62	97,59	100,00	93,24	100,00	84,09	94,19

## C. Výsledky prvých 100 NN klasifikátorov

Tabuľka 12 - Úspešnosť prvých 100 najlepších rozpoznávačov využívajúcich neurónové siete. Názvy analýz a úprav obrazov slov sú v angličtine.

Kód klasifikátora	Dĺžka preskátovaného obrazu slova	Typ analýzy	Úpravy obrazov slov	Skrýtych neurónov	Chybovosť netrénovaných	Chybovosť trénovaných	Chybovosť rovnakého rečníka	Chýb celkovo	Celková úspešnosť v %	Úspešnosť v % netrénovaných	Úspešnosť v % trénovaných	Úspešnosť v % rovnakého rečníka
32-1-3-2	32	Fourier	All	35	60	3	2	65	93,91	91,90	99,03	88,24
28-1-3-2	28	Fourier	All	35	61	5	2	68	93,63	91,77	98,38	88,24
28-3-3-2	28	Cepstral	All	35	62	5	2	69	93,53	91,63	98,38	88,24
12-1-3-2	12	Fourier	All	35	62	6	2	70	93,44	91,63	98,06	88,24
20-1-3-2	20	Fourier	All	35	64	6	0	70	93,44	91,36	98,06	100,00
16-1-3-2	16	Fourier	All	35	64	7	2	73	93,16	91,36	97,73	88,24
24-1-3-2	24	Fourier	All	35	65	7	3	75	92,97	91,23	97,73	82,35
28-3-3-1	28	Cepstral	All	25	66	6	3	75	92,97	91,09	98,06	82,35
20-3-3-2	20	Cepstral	All	35	67	8	1	76	92,88	90,96	97,41	94,12
24-3-3-1	24	Cepstral	All	25	68	6	2	76	92,88	90,82	98,06	88,24
24-3-3-2	24	Cepstral	All	35	71	6	1	78	92,69	90,42	98,06	94,12
32-3-3-2	32	Cepstral	All	35	69	8	1	78	92,69	90,69	97,41	94,12
12-1-3-1	12	Fourier	All	25	70	9	1	80	92,50	90,55	97,09	94,12
16-3-3-2	16	Cepstral	All	35	71	8	1	80	92,50	90,42	97,41	94,12
16-3-3-1	16	Cepstral	All	25	71	9	1	81	92,41	90,42	97,09	94,12
20-3-3-1	20	Cepstral	All	25	71	9	1	81	92,41	90,42	97,09	94,12
20-1-3-1	20	Fourier	All	25	71	9	3	83	92,22	90,42	97,09	82,35
28-1-3-1	28	Fourier	All	25	72	9	2	83	92,22	90,28	97,09	88,24
16-1-3-1	16	Fourier	All	25	77	6	1	84	92,13	89,61	98,06	94,12
24-1-3-1	24	Fourier	All	25	76	7	2	85	92,03	89,74	97,73	88,24
12-3-3-2	12	Cepstral	All	35	76	10	0	86	91,94	89,74	96,76	100,00
32-3-3-1	32	Cepstral	All	25	79	11	2	92	91,38	89,34	96,44	88,24
32-1-3-1	32	Fourier	All	25	80	10	3	93	91,28	89,20	96,76	82,35
24-3-1-2	24	Cepstral	No	35	83	15	0	98	90,82	88,80	95,15	100,00

Kód klasifikátora	Délka přeškálovaného obrazu slova	Typ analýzy	Úpravy obrazov slov	Skrytých neuronů	Chybovost' netrénovaných	Chybovost' trénovaných	Chybovost' rovnakého řečníka	Chyb celkovo	Celková úspěšnost v %	Úspěšnost' v % netrénovaných	Úspěšnost' v % trénovaných	Úspěšnost' v % rovnakého řečníka
32-3-1-2	32	Cepstral	No	35	85	13	0	98	90,82	88,53	95,79	100,00
24-3-3-0	24	Cepstral	All	15	77	18	5	100	90,63	89,61	94,17	70,59
24-3-1-1	24	Cepstral	No	25	85	15	1	101	90,53	88,53	95,15	94,12
12-3-3-1	12	Cepstral	All	25	84	19	2	105	90,16	88,66	93,85	88,24
20-3-1-2	20	Cepstral	No	35	84	19	2	105	90,16	88,66	93,85	88,24
20-1-3-0	20	Fourier	All	15	93	10	3	106	90,07	87,45	96,76	82,35
28-3-1-2	28	Cepstral	No	35	87	18	2	107	89,97	88,26	94,17	88,24
28-3-3-0	28	Cepstral	All	15	82	20	5	107	89,97	88,93	93,53	70,59
32-2-3-2	32	CepLPC	All	35	99	9	1	109	89,78	86,64	97,09	94,12
28-1-3-0	28	Fourier	All	15	97	12	1	110	89,69	86,91	96,12	94,12
32-3-1-1	32	Cepstral	No	25	94	16	0	110	89,69	87,31	94,82	100,00
16-3-1-1	16	Cepstral	No	25	97	13	1	111	89,60	86,91	95,79	94,12
12-3-3-0	12	Cepstral	All	15	90	20	2	112	89,50	87,85	93,53	88,24
20-3-3-0	20	Cepstral	All	15	90	20	2	112	89,50	87,85	93,53	88,24
32-2-3-1	32	CepLPC	All	25	99	13	0	112	89,50	86,64	95,79	100,00
24-1-3-0	24	Fourier	All	15	95	15	3	113	89,41	87,18	95,15	82,35
28-2-3-1	28	CepLPC	All	25	103	9	1	113	89,41	86,10	97,09	94,12
28-2-3-2	28	CepLPC	All	35	101	9	4	114	89,32	86,37	97,09	76,47
16-1-3-0	16	Fourier	All	15	94	20	2	116	89,13	87,31	93,53	88,24
32-2-2-1	32	CepLPC	Required	25	96	16	4	116	89,13	87,04	94,82	76,47
12-2-3-2	12	CepLPC	All	35	102	12	3	117	89,03	86,23	96,12	82,35
32-2-2-2	32	CepLPC	Required	35	106	9	2	117	89,03	85,70	97,09	88,24
16-2-1-2	16	CepLPC	No	35	103	12	3	118	88,94	86,10	96,12	82,35
16-2-3-2	16	CepLPC	All	35	106	11	2	119	88,85	85,70	96,44	88,24
16-3-3-0	16	Cepstral	All	15	94	22	3	119	88,85	87,31	92,88	82,35
20-2-3-1	20	CepLPC	All	25	108	10	1	119	88,85	85,43	96,76	94,12
20-2-3-2	20	CepLPC	All	35	105	11	3	119	88,85	85,83	96,44	82,35
24-2-3-1	24	CepLPC	All	25	109	9	1	119	88,85	85,29	97,09	94,12
24-2-3-2	24	CepLPC	All	35	110	9	0	119	88,85	85,16	97,09	100,00
12-3-1-1	12	Cepstral	No	25	98	18	4	120	88,75	86,77	94,17	76,47
20-2-1-2	20	CepLPC	No	35	104	13	3	120	88,75	85,96	95,79	82,35
24-3-2-2	24	Cepstral	Required	35	99	20	1	120	88,75	86,64	93,53	94,12
28-3-1-1	28	Cepstral	No	25	92	26	2	120	88,75	87,58	91,59	88,24
28-2-2-2	28	CepLPC	Required	35	102	14	5	121	88,66	86,23	95,47	70,59
16-2-3-1	16	CepLPC	All	25	107	12	3	122	88,57	85,56	96,12	82,35
16-3-1-2	16	Cepstral	No	35	107	13	2	122	88,57	85,56	95,79	88,24
20-2-2-2	20	CepLPC	Required	35	111	9	2	122	88,57	85,02	97,09	88,24
28-3-2-2	28	Cepstral	Required	35	102	17	3	122	88,57	86,23	94,50	82,35
16-2-2-2	16	CepLPC	Required	35	107	14	2	123	88,47	85,56	95,47	88,24
20-3-1-1	20	Cepstral	No	25	104	17	2	123	88,47	85,96	94,50	88,24
32-3-2-1	32	Cepstral	Required	25	97	22	5	124	88,38	86,91	92,88	70,59
32-3-3-0	32	Cepstral	All	15	103	18	3	124	88,38	86,10	94,17	82,35
12-1-3-0	12	Fourier	All	15	103	19	3	125	88,28	86,10	93,85	82,35
24-2-2-2	24	CepLPC	Required	35	111	11	4	126	88,19	85,02	96,44	76,47
12-2-3-1	12	CepLPC	All	25	113	11	3	127	88,10	84,75	96,44	82,35
16-2-2-1	16	CepLPC	Required	25	105	19	3	127	88,10	85,83	93,85	82,35
16-3-2-1	16	Cepstral	Required	25	102	20	5	127	88,10	86,23	93,53	70,59
16-3-2-2	16	Cepstral	Required	35	109	19	2	130	87,82	85,29	93,85	88,24

Kód klasifikátora	Dĺžka preskávanej obrazu slova	Typ analýzy	Úpravy obrazov slov	Skrytých neurónov	Chybovosť netrénovaných	Chybovosť trénovaných	Chybovosť rovnakého rečníka	Chýb celkovo	Celková úspešnosť v %	Úspešnosť v % netrénovaných	Úspešnosť v % trénovaných	Úspešnosť v % rovnakého rečníka
24-1-1-2	24	Fourier	No	35	116	14	0	130	87,82	84,35	95,47	100,00
24-2-2-1	24	CepLPC	Required	25	114	13	4	131	87,72	84,62	95,79	76,47
28-2-1-2	28	CepLPC	No	35	112	16	3	131	87,72	84,89	94,82	82,35
28-2-2-1	28	CepLPC	Required	25	115	11	5	131	87,72	84,48	96,44	70,59
32-2-1-2	32	CepLPC	No	35	113	17	1	131	87,72	84,75	94,50	94,12
12-1-1-2	12	Fourier	No	35	118	16	0	134	87,44	84,08	94,82	100,00
12-2-2-1	12	CepLPC	Required	25	113	20	2	135	87,35	84,75	93,53	88,24
28-2-1-1	28	CepLPC	No	25	113	19	3	135	87,35	84,75	93,85	82,35
28-3-2-1	28	Cepstral	Required	25	111	22	2	135	87,35	85,02	92,88	88,24
12-1-2-2	12	Fourier	Required	35	120	16	0	136	87,25	83,81	94,82	100,00
12-2-1-2	12	CepLPC	No	35	111	22	3	136	87,25	85,02	92,88	82,35
20-3-2-2	20	Cepstral	Required	35	106	29	1	136	87,25	85,70	90,61	94,12
12-3-1-2	12	Cepstral	No	35	109	25	3	137	87,16	85,29	91,91	82,35
20-2-1-1	20	CepLPC	No	25	119	17	1	137	87,16	83,94	94,50	94,12
28-1-2-2	28	Fourier	Required	35	123	14	1	138	87,07	83,40	95,47	94,12
32-2-1-1	32	CepLPC	No	25	115	17	6	138	87,07	84,48	94,50	64,71
16-1-2-2	16	Fourier	Required	35	124	14	1	139	86,97	83,27	95,47	94,12
28-1-1-2	28	Fourier	No	35	119	19	1	139	86,97	83,94	93,85	94,12
32-1-3-0	32	Fourier	All	15	112	23	4	139	86,97	84,89	92,56	76,47
20-1-1-2	20	Fourier	No	35	123	17	0	140	86,88	83,40	94,50	100,00
32-2-2-0	32	CepLPC	Required	15	107	28	5	140	86,88	85,56	90,94	70,59
12-2-2-2	12	CepLPC	Required	35	117	21	3	141	86,79	84,21	93,20	82,35
24-3-2-1	24	Cepstral	Required	25	104	32	5	141	86,79	85,96	89,64	70,59
20-1-2-2	20	Fourier	Required	35	121	20	1	142	86,69	83,67	93,53	94,12
32-1-1-2	32	Fourier	No	35	120	20	2	142	86,69	83,81	93,53	88,24
12-1-1-1	12	Fourier	No	25	120	23	1	144	86,50	83,81	92,56	94,12
20-3-2-1	20	Cepstral	Required	25	116	23	5	144	86,50	84,35	92,56	70,59

**Tabuľka 13** - Úspešnosť rozpoznávania jednotlivých slov prvých 100 najúspešnejších rozpoznávačov NN. Sú zoradené podľa celkovej úspešnosti.

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
32-1-3-2	85,37	95,18	90,24	92,59	98,78	92,77	92,77	92,68	97,62	96,39	100,0	91,89	100,0	88,64	93,91
28-1-3-2	84,15	96,39	92,68	88,89	93,90	95,18	95,18	93,90	97,62	97,59	100,0	89,19	100,0	84,09	93,63
28-3-3-2	90,24	98,80	89,02	88,89	95,12	89,16	91,57	96,34	97,62	93,98	97,47	93,24	93,33	95,45	93,53
12-1-3-2	90,24	96,39	90,24	91,36	96,34	93,98	93,98	95,12	97,62	93,98	100,0	85,14	100,0	79,55	93,44
20-1-3-2	81,71	96,39	90,24	90,12	98,78	95,18	93,98	93,90	95,24	93,98	100,0	90,54	100,0	88,64	93,44
16-1-3-2	87,80	92,77	95,12	91,36	92,68	92,77	93,98	91,46	97,62	93,98	100,0	85,14	97,78	93,18	93,16
24-1-3-2	86,59	96,39	90,24	90,12	95,12	92,77	92,77	91,46	97,62	92,77	100,0	89,19	100,0	86,36	92,97
28-3-3-1	85,37	97,59	86,59	88,89	96,34	91,57	96,39	93,90	98,81	89,16	96,20	94,59	95,56	90,91	92,97
20-3-3-2	91,46	96,39	87,80	88,89	93,90	90,36	90,36	91,46	96,43	93,98	98,73	93,24	97,78	90,91	92,88
24-3-3-1	92,68	91,57	85,37	90,12	96,34	91,57	91,57	95,12	95,24	90,36	100,0	91,89	95,56	95,45	92,88
24-3-3-2	91,46	96,39	82,93	90,12	95,12	92,77	91,57	93,90	95,24	92,77	94,94	93,24	97,78	90,91	92,69
32-3-3-2	92,68	95,18	86,59	88,89	95,12	85,54	93,98	90,24	97,62	95,18	96,20	90,54	97,78	95,45	92,69
12-1-3-1	85,37	95,18	93,90	88,89	96,34	89,16	92,77	87,80	97,62	91,57	98,73	91,89	97,78	88,64	92,50
16-3-3-2	91,46	96,39	81,71	86,42	93,90	89,16	92,77	93,90	97,62	93,98	100,0	90,54	97,78	90,91	92,50
16-3-3-1	90,24	95,18	82,93	87,65	92,68	86,75	92,77	95,12	96,43	95,18	100,0	90,54	97,78	93,18	92,41
20-3-3-1	85,37	97,59	90,24	88,89	93,90	90,36	92,77	87,80	96,43	92,77	100,0	89,19	97,78	93,18	92,41

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
20-1-3-1	80,49	96,39	91,46	93,83	90,24	89,16	96,39	91,46	96,43	95,18	100,0	86,49	97,78	84,09	92,22
28-1-3-1	86,59	91,57	86,59	91,36	93,90	96,39	92,77	92,68	96,43	92,77	96,20	87,84	97,78	88,64	92,22
16-1-3-1	86,59	92,77	86,59	86,42	95,12	90,36	92,77	90,24	96,43	96,39	100,0	89,19	97,78	90,91	92,13
24-1-3-1	82,93	93,98	90,24	90,12	92,68	92,77	92,77	92,68	96,43	91,57	98,73	87,84	100,0	86,36	92,03
12-3-3-2	87,80	95,18	85,37	81,48	98,78	90,36	92,77	90,24	96,43	93,98	100,0	90,54	93,33	90,91	91,94
32-3-3-1	85,37	95,18	78,05	92,59	93,90	85,54	96,39	92,68	97,62	87,95	97,47	90,54	97,78	90,91	91,38
32-1-3-1	74,39	96,39	85,37	90,12	96,34	92,77	91,57	85,37	98,81	96,39	100,0	86,49	100,0	84,09	91,28
24-3-1-2	81,71	95,18	91,46	86,42	91,46	84,34	97,59	93,90	95,24	89,16	98,73	87,84	95,56	79,55	90,82
32-3-1-2	85,37	93,98	81,71	86,42	91,46	90,36	96,39	95,12	95,24	83,13	100,0	90,54	97,78	84,09	90,82
24-3-3-0	87,80	92,77	85,37	91,36	93,90	86,75	91,57	90,24	94,05	87,95	98,73	89,19	93,33	84,09	90,63
24-3-1-1	85,37	93,98	91,46	86,42	90,24	81,93	96,39	93,90	94,05	86,75	98,73	86,49	95,56	86,36	90,53
12-3-3-1	84,15	91,57	85,37	80,25	93,90	92,77	91,57	91,46	96,43	80,72	94,94	91,89	97,78	95,45	90,16
20-3-1-2	81,71	89,16	90,24	83,95	95,12	85,54	93,98	90,24	95,24	89,16	98,73	87,84	97,78	84,09	90,16
20-1-3-0	90,24	95,18	80,49	86,42	87,80	90,36	87,95	89,02	98,81	89,16	98,73	83,78	97,78	86,36	90,07
28-3-1-2	80,49	91,57	82,93	91,36	85,37	84,34	100,0	92,68	96,43	86,75	100,0	87,84	100,0	79,55	89,97
28-3-3-0	82,93	92,77	80,49	88,89	93,90	85,54	87,95	92,68	96,43	86,75	97,47	91,89	93,33	90,91	89,97
32-2-3-2	86,59	89,16	90,24	85,19	92,68	85,54	92,77	90,24	92,86	86,75	100,0	86,49	95,56	81,82	89,78
28-1-3-0	90,24	95,18	79,27	91,36	95,12	74,70	97,59	79,27	97,62	89,16	100,0	86,49	95,56	84,09	89,69
32-3-1-1	81,71	95,18	86,59	80,25	92,68	84,34	96,39	91,46	95,24	84,34	100,0	85,14	95,56	88,64	89,69
16-3-1-1	86,59	91,57	82,93	85,19	91,46	85,54	95,18	92,68	94,05	85,54	96,20	90,54	97,78	77,27	89,60
12-3-3-0	93,90	96,39	80,49	76,54	95,12	87,95	92,77	90,24	96,43	78,31	97,47	85,14	97,78	86,36	89,50
20-3-3-0	86,59	92,77	76,83	90,12	95,12	84,34	78,31	91,46	97,62	87,95	98,73	87,84	95,56	95,45	89,50
32-2-3-1	85,37	95,18	79,27	86,42	93,90	86,75	95,18	82,93	96,43	89,16	98,73	86,49	95,56	79,55	89,50
24-1-3-0	67,07	96,39	89,02	87,65	90,24	91,57	85,54	90,24	96,43	91,57	100,0	89,19	95,56	79,55	89,41
28-2-3-1	81,71	90,36	89,02	87,65	89,02	87,95	92,77	84,15	95,24	87,95	98,73	90,54	97,78	77,27	89,41
28-2-3-2	85,37	86,75	91,46	90,12	91,46	81,93	92,77	85,37	95,24	84,34	100,0	86,49	97,78	81,82	89,32
16-1-3-0	67,07	95,18	87,80	80,25	96,34	86,75	93,98	92,68	96,43	90,36	96,20	85,14	97,78	81,82	89,13
32-2-2-1	81,71	85,54	85,37	87,65	89,02	91,57	92,77	84,15	94,05	85,54	98,73	89,19	97,78	88,64	89,13
12-2-3-2	84,15	89,16	86,59	86,42	91,46	83,13	91,57	84,15	94,05	85,54	100,0	90,54	97,78	84,09	89,03
32-2-2-2	82,93	91,57	85,37	83,95	89,02	86,75	95,18	87,80	94,05	84,34	98,73	86,49	97,78	84,09	89,03
16-2-1-2	75,61	91,57	87,80	82,72	92,68	85,54	95,18	81,71	95,24	81,93	100,0	93,24	97,78	88,64	88,94
16-2-3-2	80,49	91,57	86,59	81,48	89,02	85,54	93,98	80,49	95,24	90,36	100,0	90,54	95,56	84,09	88,85
16-3-3-0	87,80	90,36	81,71	79,01	93,90	80,72	90,36	91,46	92,86	86,75	92,41	94,59	95,56	90,91	88,85
20-2-3-1	81,71	92,77	84,15	81,48	91,46	87,95	93,98	87,80	89,29	87,95	98,73	83,78	97,78	88,64	88,85
20-2-3-2	82,93	89,16	85,37	80,25	90,24	89,16	93,98	87,80	96,43	79,52	98,73	86,49	100,0	88,64	88,85
24-2-3-1	84,15	90,36	86,59	86,42	93,90	80,72	91,57	85,37	94,05	86,75	100,0	89,19	91,11	81,82	88,85
24-2-3-2	86,59	89,16	86,59	80,25	89,02	83,13	92,77	82,93	95,24	91,57	100,0	89,19	95,56	81,82	88,85
12-3-1-1	82,93	87,95	89,02	81,48	91,46	87,95	93,98	90,24	95,24	85,54	96,20	82,43	95,56	81,82	88,75
20-2-1-2	90,24	91,57	85,37	83,95	90,24	83,13	96,39	80,49	96,43	77,11	98,73	85,14	95,56	93,18	88,75
24-3-2-2	86,59	93,98	81,71	86,42	96,34	85,54	87,95	93,90	89,29	75,90	100,0	87,84	95,56	81,82	88,75
28-3-1-1	86,59	93,98	81,71	86,42	91,46	86,75	90,36	91,46	94,05	78,31	98,73	82,43	97,78	84,09	88,75
28-2-2-2	82,93	89,16	86,59	80,25	91,46	81,93	92,77	85,37	97,62	81,93	98,73	91,89	95,56	88,64	88,66
16-2-3-1	90,24	86,75	84,15	81,48	91,46	83,13	92,77	85,37	94,05	91,57	98,73	87,84	91,11	77,27	88,57
16-3-1-2	90,24	93,98	73,17	81,48	90,24	80,72	96,39	93,90	96,43	85,54	98,73	83,78	97,78	75,00	88,57
20-2-2-2	81,71	90,36	84,15	81,48	86,59	84,34	93,98	89,02	96,43	81,93	98,73	91,89	93,33	88,64	88,57
28-3-2-2	84,15	87,95	76,83	81,48	96,34	85,54	95,18	90,24	96,43	84,34	94,94	86,49	95,56	86,36	88,57
16-2-2-2	79,27	90,36	89,02	81,48	92,68	83,13	93,98	85,37	94,05	81,93	98,73	86,49	100,0	86,36	88,47
20-3-1-1	85,37	91,57	84,15	76,54	92,68	79,52	95,18	91,46	95,24	84,34	100,0	87,84	93,33	79,55	88,47
32-3-2-1	86,59	93,98	76,83	80,25	97,56	89,16	89,16	86,59	92,86	83,13	94,94	87,84	95,56	84,09	88,38
32-3-3-0	82,93	97,59	82,93	85,19	95,12	84,34	85,54	85,37	94,05	85,54	97,47	85,14	86,67	88,64	88,38
12-1-3-0	71,95	89,16	90,24	90,12	90,24	87,95	86,75	87,80	96,43	80,72	97,47	87,84	95,56	86,36	88,28
24-2-2-2	81,71	86,75	81,71	83,95	90,24	84,34	95,18	82,93	95,24	86,75	98,73	86,49	95,56	88,64	88,19
12-2-3-1	79,27	86,75	87,80	88,89	86,59	80,72	91,57	86,59	94,05	86,75	100,0	89,19	95,56	79,55	88,10
16-2-2-1	80,49	89,16	85,37	87,65	91,46	83,13	95,18	78,05	95,24	78,31	98,73	90,54	97,78	86,36	88,10

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
16-3-2-1	81,71	96,39	82,93	83,95	95,12	84,34	95,18	80,49	95,24	83,13	94,94	86,49	91,11	79,55	88,10
16-3-2-2	80,49	89,16	82,93	82,72	93,90	87,95	92,77	89,02	92,86	80,72	97,47	89,19	88,89	77,27	87,82
24-1-1-2	87,80	90,36	95,12	81,48	92,68	89,16	91,57	85,37	92,86	77,11	97,47	85,14	95,56	56,82	87,82
24-2-2-1	81,71	86,75	90,24	85,19	91,46	87,95	91,57	81,71	96,43	78,31	96,20	83,78	93,33	84,09	87,72
28-2-1-2	87,80	89,16	84,15	81,48	91,46	85,54	92,77	79,27	94,05	72,29	100,0	91,89	93,33	88,64	87,72
28-2-2-1	85,37	87,95	84,15	81,48	89,02	83,13	87,95	82,93	96,43	81,93	98,73	90,54	95,56	86,36	87,72
32-2-1-2	79,27	89,16	86,59	80,25	93,90	87,95	91,57	80,49	96,43	81,93	100,0	85,14	95,56	79,55	87,72
12-1-1-2	87,80	87,95	89,02	80,25	86,59	89,16	95,18	92,68	90,48	81,93	94,94	83,78	93,33	61,36	87,44
12-2-2-1	79,27	90,36	86,59	80,25	85,37	89,16	95,18	74,39	94,05	81,93	97,47	89,19	97,78	86,36	87,35
28-2-1-1	82,93	87,95	84,15	81,48	89,02	85,54	92,77	81,71	95,24	77,11	96,20	90,54	95,56	86,36	87,35
28-3-2-1	76,83	89,16	86,59	76,54	97,56	86,75	90,36	91,46	95,24	78,31	96,20	81,08	93,33	84,09	87,35
12-1-2-2	79,27	91,57	91,46	81,48	87,80	90,36	95,18	81,71	91,67	83,13	97,47	86,49	91,11	63,64	87,25
12-2-1-2	78,05	86,75	86,59	85,19	86,59	83,13	93,98	82,93	92,86	81,93	98,73	89,19	95,56	81,82	87,25
20-3-2-2	89,02	89,16	70,73	83,95	95,12	78,31	93,98	92,68	95,24	72,29	97,47	87,84	91,11	86,36	87,25
12-3-1-2	81,71	89,16	76,83	80,25	89,02	85,54	90,36	91,46	92,86	84,34	97,47	86,49	91,11	84,09	87,16
20-2-1-1	84,15	89,16	86,59	79,01	86,59	84,34	96,39	82,93	94,05	78,31	98,73	83,78	93,33	84,09	87,16
28-1-2-2	84,15	89,16	92,68	80,25	82,93	95,18	90,36	82,93	90,48	80,72	97,47	85,14	91,11	70,45	87,07
32-2-1-1	85,37	86,75	80,49	85,19	89,02	84,34	93,98	78,05	91,67	79,52	96,20	89,19	91,11	93,18	87,07
16-1-2-2	85,37	86,75	91,46	80,25	87,80	89,16	95,18	82,93	90,48	78,31	97,47	85,14	93,33	68,18	86,97
28-1-1-2	86,59	86,75	87,80	86,42	91,46	91,57	91,57	80,49	88,10	80,72	96,20	89,19	91,11	59,09	86,97
32-1-3-0	63,41	93,98	80,49	87,65	93,90	84,34	89,16	85,37	96,43	83,13	100,0	87,84	93,33	77,27	86,97
20-1-1-2	85,37	86,75	86,59	80,25	89,02	89,16	93,98	86,59	90,48	78,31	100,0	89,19	95,56	54,55	86,88
32-2-2-0	79,27	86,75	87,80	86,42	89,02	81,93	89,16	78,05	95,24	77,11	96,20	87,84	95,56	93,18	86,88
12-2-2-2	80,49	89,16	86,59	82,72	86,59	85,54	91,57	75,61	95,24	78,31	98,73	87,84	100,0	79,55	86,79
24-3-2-1	82,93	90,36	73,17	82,72	90,24	90,36	92,77	86,59	94,05	77,11	97,47	78,38	95,56	86,36	86,79
20-1-2-2	85,37	85,54	93,90	81,48	90,24	92,77	87,95	82,93	92,86	77,11	96,20	85,14	93,33	59,09	86,69
32-1-1-2	85,37	85,54	84,15	82,72	93,90	86,75	95,18	81,71	91,67	74,70	97,47	94,59	93,33	56,82	86,69
12-1-1-1	81,71	90,36	89,02	74,07	89,02	86,75	93,98	82,93	91,67	80,72	96,20	93,24	93,33	59,09	86,50
20-3-2-1	86,59	87,95	71,95	76,54	87,80	84,34	90,36	92,68	96,43	80,72	96,20	83,78	93,33	84,09	86,50

## D. Výsledky HMM klasifikátorov

Tabuľka 14 - Prehľad výsledkov úspešnosti jednotlivých HMM klasifikátorov. Výsledky sú zoradené podľa celkovej úspešnosti. Typy analýz, miery skreslenia a názvy úprav obrazov slov sú v angličtine.

Kód klasifikátora	Typ analýzy	Miera skreslenia	Úpravy obrazov slov	Centier kvantizéra	Stavov HMM	Chybovosť netrénovaných	Chybovosť tréovaných	Chybovosť rovnakého rečníka	Chýb celkovo	Celková úspešnosť v %	Úspešnosť v % netrénovaných	Úspešnosť v % tréovaných	Úspešnosť v % rovnakého rečníka
2-4-3-7-9	CepLPC	WCepst.	All	1024	9	140	15	1	156	85,38	81,11	95,15	94,12
2-3-3-7-9	CepLPC	Cepstral	All	1024	9	148	18	1	167	84,35	80,03	94,17	94,12
1-1-3-1-7	Fourier	Euklid	All	256	7	147	23	4	174	83,69	80,16	92,56	76,47
2-1-3-7-9	CepLPC	Euklid	All	1024	9	149	25	1	175	83,60	79,89	91,91	94,12
2-4-3-1-7	CepLPC	WCepst.	All	256	7	141	34	1	176	83,51	80,97	89,00	94,12
1-1-3-7-9	Fourier	Euklid	All	1024	9	159	23	0	182	82,94	78,54	92,56	100,00
1-1-3-0-7	Fourier	Euklid	All	128	7	141	41	6	188	82,38	80,97	86,73	64,71
2-4-3-0-7	CepLPC	WCepst.	All	128	7	165	37	3	205	80,79	77,73	88,03	82,35
2-4-3-1-5	CepLPC	WCepst.	All	256	5	175	37	2	214	79,94	76,38	88,03	88,24

Kód klasifikátora	Typ analýzy	Miera skreslenia	Úpravy obrazov slov	Centier kvantizéra	Stavov HMM	Chybovosť netrénovaných	Chybovosť trénovaných	Chybovosť rovnakého rečníka	Chýb celkovo	Celková úspešnosť v %	Úspešnosť v % netrénovaných	Úspešnosť v % trénovaných	Úspešnosť v % rovnakého rečníka
1-1-3-1-5	Fourier	Euklid	All	256	5	181	37	5	223	79,10	75,57	88,03	70,59
2-4-3-0-5	CepLPC	WCepst.	All	128	5	183	47	3	233	78,16	75,30	84,79	82,35
2-4-2-1-7	CepLPC	WCepst.	Required	256	7	206	36	5	247	76,85	72,20	88,35	70,59
2-1-1-7-9	CepLPC	Euklid	No	1024	9	217	32	2	251	76,48	70,72	89,64	88,24
1-1-3-0-5	Fourier	Euklid	All	128	5	196	57	6	259	75,73	73,55	81,55	64,71
2-4-2-0-7	CepLPC	WCepst.	Required	128	7	210	52	3	265	75,16	71,66	83,17	82,35
2-1-1-7-7	CepLPC	Euklid	No	1024	7	239	36	0	275	74,23	67,75	88,35	100,00
2-4-1-1-7	CepLPC	WCepst.	No	256	7	228	60	0	288	73,01	69,23	80,58	100,00
2-4-1-0-7	CepLPC	WCepst.	No	128	7	231	61	1	293	72,54	68,83	80,26	94,12
2-4-2-1-5	CepLPC	WCepst.	Required	256	5	245	50	5	300	71,88	66,94	83,82	70,59
2-4-1-1-5	CepLPC	WCepst.	No	256	5	241	64	3	308	71,13	67,48	79,29	82,35
2-1-1-7-5	CepLPC	Euklid	No	1024	5	258	51	2	311	70,85	65,18	83,50	88,24
2-4-2-0-5	CepLPC	WCepst.	Required	128	5	253	65	4	322	69,82	65,86	78,96	76,47
1-2-1-1-7	Fourier	Angle	No	256	7	267	59	5	331	68,98	63,97	80,91	70,59
2-4-1-0-5	CepLPC	WCepst.	No	128	5	265	64	4	333	68,79	64,24	79,29	76,47
1-2-2-0-7	Fourier	Angle	Required	128	7	289	76	6	371	65,23	61,00	75,40	64,71
1-2-1-0-7	Fourier	Angle	No	128	7	289	76	10	375	64,85	61,00	75,40	41,18
1-1-1-7-9	Fourier	Euklid	No	1024	9	302	71	8	381	64,29	59,24	77,02	52,94
1-2-1-1-5	Fourier	Angle	No	256	5	301	80	8	389	63,54	59,38	74,11	52,94
1-2-1-0-5	Fourier	Angle	No	128	5	313	94	8	415	61,11	57,76	69,58	52,94
1-1-2-1-7	Fourier	Euklid	Required	256	7	340	118	9	467	56,23	54,12	61,81	47,06
1-2-2-0-5	Fourier	Angle	Required	128	5	363	100	10	473	55,67	51,01	67,64	41,18
1-1-1-1-7	Fourier	Euklid	No	256	7	399	127	11	537	49,67	46,15	58,90	35,29
1-1-1-1-5	Fourier	Euklid	No	256	5	395	130	13	538	49,58	46,69	57,93	23,53
1-1-2-0-7	Fourier	Euklid	Required	128	7	419	136	11	566	46,95	43,45	55,99	35,29
1-1-2-1-5	Fourier	Euklid	Required	256	5	427	131	10	568	46,77	42,38	57,61	41,18
1-1-1-0-7	Fourier	Euklid	No	128	7	431	137	11	579	45,74	41,84	55,66	35,29
1-1-1-0-5	Fourier	Euklid	No	128	5	447	145	12	604	43,39	39,68	53,07	29,41
1-1-2-0-5	Fourier	Euklid	Required	128	5	483	162	12	657	38,43	34,82	47,57	29,41

Tabuľka 15 - Úspešnosť HMM klasifikátorov na jednotlivých slovách v poradí od najúspešnejšieho.

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
2-4-3-7-9	86,59	81,93	90,24	80,25	85,37	79,52	86,75	76,83	97,62	89,16	91,14	71,62	100,00	81,82	85,38
2-3-3-7-9	86,59	85,54	82,93	76,54	85,37	80,72	87,95	80,49	94,05	85,54	94,94	62,16	100,00	81,82	84,35
1-1-3-1-7	84,15	86,75	75,61	81,48	79,27	75,90	86,75	75,61	97,62	73,49	89,87	90,54	91,11	90,91	83,69
2-1-3-7-9	91,46	84,34	76,83	79,01	84,15	81,93	87,95	80,49	95,24	81,93	88,61	60,81	100,00	81,82	83,60
2-4-3-1-7	85,37	83,13	87,80	66,67	79,27	81,93	84,34	69,51	96,43	85,54	89,87	82,43	100,00	84,09	83,51
1-1-3-7-9	84,15	85,54	67,07	77,78	85,37	73,49	78,31	65,85	90,48	92,77	93,67	90,54	100,00	86,36	82,94
1-1-3-0-7	69,51	81,93	71,95	75,31	78,05	72,29	78,31	91,46	97,62	85,54	92,41	81,08	97,78	90,91	82,38
2-4-3-0-7	85,37	85,54	67,07	72,84	86,59	74,70	86,75	58,54	96,43	87,95	87,34	71,62	100,00	75,00	80,79
2-4-3-1-5	87,80	83,13	84,15	54,32	86,59	78,31	77,11	64,63	92,86	68,67	88,61	78,38	100,00	86,36	79,94
1-1-3-1-5	53,66	81,93	80,49	69,14	84,15	77,11	78,31	80,49	97,62	67,47	75,95	86,49	95,56	93,18	79,10
2-4-3-0-5	76,83	86,75	73,17	62,96	86,59	73,49	80,72	68,29	95,24	55,42	83,54	85,14	100,00	75,00	78,16
2-4-2-1-7	62,20	92,77	89,02	67,90	80,49	87,95	71,08	52,44	86,90	79,52	81,01	64,86	100,00	61,36	76,85
2-1-1-7-9	76,83	87,95	78,05	62,96	80,49	90,36	74,70	75,61	94,05	83,13	72,15	37,84	91,11	59,09	76,48
1-1-3-0-5	60,98	81,93	74,39	50,62	82,93	68,67	77,11	65,85	96,43	59,04	78,48	91,89	97,78	93,18	75,73

Kód	0	1	2	3	4	5	6	7	8	9	A	N	T	S	Celkom
2-4-2-0-7	65,85	87,95	84,15	91,36	64,63	45,78	68,67	65,85	95,24	91,57	88,61	66,22	66,67	56,82	75,16
2-1-1-7-7	69,51	83,13	73,17	60,49	81,71	83,13	77,11	78,05	94,05	80,72	73,42	31,08	86,67	61,36	74,23
2-4-1-1-7	87,80	89,16	36,59	48,15	76,83	91,57	68,67	57,32	91,67	84,34	86,08	50,00	91,11	63,64	73,01
2-4-1-0-7	50,00	85,54	87,80	82,72	56,10	83,13	61,45	50,00	86,90	90,36	94,94	45,95	71,11	61,36	72,54
2-4-2-1-5	89,02	87,95	47,56	79,01	75,61	53,01	73,49	45,12	92,86	85,54	75,95	51,35	91,11	59,09	71,88
2-4-1-1-5	80,49	83,13	45,12	76,54	71,95	54,22	68,67	65,85	95,24	87,95	73,42	41,89	91,11	61,36	71,13
2-1-1-7-5	54,88	91,57	60,98	60,49	78,05	81,93	71,08	63,41	91,67	81,93	60,76	40,54	91,11	65,91	70,85
2-4-2-0-5	91,46	85,54	31,71	69,14	71,95	51,81	74,70	39,02	95,24	79,52	77,22	66,22	88,89	56,82	69,82
1-2-1-1-7	68,29	73,49	25,61	59,26	69,51	69,88	79,52	76,83	79,76	89,16	96,20	29,73	88,89	61,36	68,98
2-4-1-0-5	51,22	79,52	86,59	75,31	63,41	68,67	66,27	37,80	91,67	80,72	69,62	54,05	73,33	61,36	68,79
1-2-2-0-7	50,00	51,81	75,61	41,98	78,05	65,06	73,49	68,29	97,62	65,06	69,62	27,03	100,00	56,82	65,23
1-2-1-0-7	37,80	60,24	13,41	69,14	75,61	53,01	77,11	57,32	77,38	83,13	100,00	64,86	91,11	56,82	64,85
1-1-1-7-9	67,07	80,72	34,15	48,15	70,73	80,72	60,24	65,85	84,52	61,45	67,09	36,49	88,89	59,09	64,29
1-2-1-1-5	53,66	72,29	30,49	41,98	80,49	81,93	62,65	25,61	85,71	65,06	96,20	44,59	100,00	63,64	63,54
1-2-1-0-5	45,12	68,67	37,80	37,04	78,05	75,90	63,86	26,83	89,29	60,24	92,41	44,59	84,44	59,09	61,11
1-1-2-1-7	42,68	71,08	34,15	17,28	79,27	77,11	60,24	40,24	80,95	25,30	74,68	44,59	88,89	70,45	56,23
1-2-2-0-5	57,32	67,47	54,88	14,81	76,83	13,25	74,70	17,07	76,19	79,52	68,35	40,54	95,56	61,36	55,67
1-1-1-1-7	25,61	71,08	34,15	40,74	75,61	60,24	45,78	36,59	73,81	30,12	49,37	18,92	84,44	70,45	49,67
1-1-1-1-5	35,37	75,90	41,46	41,98	58,54	51,81	44,58	51,22	73,81	12,05	49,37	25,68	86,67	68,18	49,58
1-1-2-0-7	24,39	71,08	42,68	35,80	80,49	60,24	36,14	31,71	64,29	13,25	54,43	20,27	77,78	63,64	46,95
1-1-2-1-5	31,71	63,86	51,22	17,28	60,98	21,69	56,63	35,37	70,24	7,23	83,54	31,08	88,89	59,09	46,77
1-1-1-0-7	14,63	69,88	40,24	33,33	76,83	54,22	38,55	29,27	80,95	6,02	59,49	14,86	71,11	70,45	45,74
1-1-1-0-5	30,49	59,04	28,05	19,75	51,22	46,99	43,37	35,37	72,62	16,87	53,16	29,73	91,11	54,55	43,39
1-1-2-0-5	24,39	36,14	28,05	45,68	65,85	19,28	31,33	42,68	60,71	9,64	50,63	9,46	80,00	61,36	38,43

## E. Priložené CD

Čo možno nájsť na priloženom CD, prehľadne zobrazuje Tabuľka 16.

**Tabuľka 16** - Prehľad obsahov jednotlivých adresárov na priloženom CD.

Adresár	Obsah
Bin	Binárne súbory všetkých troch aplikácií spolu s ich inicializačnými súbormi.
Common	Spoločné zdrojové kódy pre všetky tri programy - platformovo nezávislé.
Documents	Elektronické formy tohoto dokumentu v rôznych formátoch.
Obj	Prázdny adresár na výstup pomocných výstupov kompilátora. Používaný len na kompiláciu (po skopírovaní na pevný disk).
PhoneData	Trénovacie dáta nahrané z mobilných telefónov a roztriedené podľa významu.
ProjectGroup	Súbor projektovej skupiny.
Results	Výsledky a modely získané v našom experimente.
SrDTW	Zdrojové kódy aplikácie SrDTW.
SrHMM	Zdrojové kódy aplikácie SrHMM.
SrNN	Zdrojové kódy aplikácie SrNN.
Vox	Zvuky získané nahrávaním cez mobilné telefóny.
Waves	Prázdny adresár výstupných zvukov pri automatickom oddeľovaní slov niektorým z programov.

# Literatúra

[ÚNS] Vladimír Kvasnička, Ľubica Beňušková, Jiří Pospíchal, Igor Farkaš, Peter Tiňo, Andrej Král, “*Úvod do teórie neurónových sietí*”, IRIS 1997, [Sk]

[PSU] Josef Psutka, “*Komunikace s počítačem mluvenou řečí*”, ACADEMIA PRAHA 1995, [Cz]

[SUN] Ramasubramaniam H. Sundaram, “*The Baum-Welch Algorithm*”, Mississippi State Univerzity, [En]

[RJI] L.R. Rabiner, B.H. Juang, “*An Introduction to Hidden Markov Models*”, IEEE ASSP Magazine, 1986, [En]

[RJF] L.R. Rabiner, B.H. Juang, “*Fundamentals of Speech Recognition*”, Prentice Hall Signal Processing Series, 1993, [En]

[HRI] R.M. Hristev, “*The ANN Book*”, GNU pdf, 1998, [En]

[PIC] Joseph Picone, “*Continuous Speech Recognition Using Hidden Markov Models*”, IEEE ASSP Magazine, 1990, [En]

[SUR] Ronald A. Cole, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, Victor Zue, “*Survey of the State of the Art in Human Language Technology*”, HTML a PS, 1995, CSLU OGI, [En]

[JUT] Jutta Degener, “*GSM 06.10 lossy speech compression*”, HTML, 2000, [En]

[SCO] John Scourias, “*Overview of the Global System for Mobile Communications*”, HTML, 1997, [En]

[RAS] L.R. Rabiner, R.W. Schafer, “*Digital Processing of Speech Signals*”, Prentice Hall Inc. Englewood Cliffs, New Jersey, 1978, [En]

[KUT] M.H. Kuhn, H.H. Tomaszewski, “*Improvements in Isolated Word Recognition*”, IEEE Transactions on ASSP, 1983, [En]

[DAA] Dialogic Corp., <http://www.comptek.ru/telephony/tnotes/tt1-13.html>, “*Dialogic ADPCM Algorithm*”, HTML, [En]

[DEB] Jutta Degener, Carsten Bormann, <http://www.cs.tu-berlin.de>, Source Codes, 1992

## Literatúra

---

[DEG] Jutta Degener, <http://www.ddj.com>, "*Putting the GSM 06.10 RPE-LTP algorithm to work*", HTML, 1994, [En]

[HUE] J. M. Huerta, R. M. Stern, "*Speech Recognition From GSM Codec Parameters*", DECE SCS Carnegie Mellone University, PS, [En]

[PCH] J. Psutka, E. Chán, "Experimentální systém rozpoznávání slov SiRiuS 1.3.", Automatizace, 1989, [Cz]