VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ROZPOZNÁVÁNÍ ŘEČI (ČÍSLICE)
SPEECH RECOGNITION (DIGIT)

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — MARTIN KANTAR
AUTHOR

VEDOUCÍ PRÁCE — Ing. RADOMIL MATOUŠEK, Ph.D.
SUPERVISOR

BRNO 2012

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Martin Kantar

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Rozpoznávání řeči (číslice)

v anglickém jazyce:

Speech Recognition (digit)

Stručná charakteristika problematiky úkolu:

V oblasti umělé inteligence a zpracování signálu existuje mnoho oblastí výzkumu. Pro současnost a hlavně budoucnost bude zásadní schopnost "inteligentních" strojů rozpoznat mluvené slovo. Daná práce, v rámci svého rozsahu a domény BP, rešeršním způsobem pojedná o možnostech a současných trendech v oblasti rozpoznávání řeči. Prakticky bude v práci učiněn pokus o rozpoznání číslic pomocí zvolené metody.

Cíle bakalářské práce:

- Popsat základní model tvorby řeči.
- Rešerše tématiky rozpoznávání řeči, zejména parametrizace signálu (redukce dat) s využitím FFT, LPC, MFCC, delta příznaků, kepstrální příznaky aj.
- Pomocí zvolené metody (např. LPC, ANN) se pokusit o rozpoznání číslic 0-9.

Seznam odborné literatury:

1. Uhlíř, J. - Sovka, P. - Pollák, P. - Hanžl, V. - Čmejla, R.: Technologie hlasových komunikací. Nakladatelství ČVUT, 2007.
2. Huang, X. - Acero, A. - Hon, H.-W.: Spoken Language Processing. Prentice Hall 2001

Vedoucí bakalářské práce: Ing. Radomil Matoušek, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne 3.5.2012

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

ANOTACE

V bakalářské práci vysvětluji, jak samotná řeč vypadá a co jí ovlivňuje. Zmiňuji zde nejčastěji používané metody, kterými si řečové signály můžeme připravit pro rozpoznávání. Na názorných příkladech ukazuji, na jakých principech dnešní rozpoznávače řeči pracují, jaké mají výhody a nevýhody. Pro metodu založenou na učení neuronových sítí jsem vytvořil v prostředí Matlabu řečový rozpoznávač číslovek 0-9.

ANNOTATION

The aim of this diploma thesis is to explain what speech is and what are its constituents. I mention commonly used methods which are used for preparation of signals which we use for recognition. Schematic examples show principles of current recognizers of speech, their advantages and disadvantages. I made speech recognition program for 0-9 numerals in Matlab for neural nets learning.

KLÍČOVÁ SLOVA

Rozpoznávání řeči, parametrizace, FFT, LPC, PLP, MFCC, DTW, HMM, neuronové sítě, Matlab.

KEYWORDS

Speech recognition, parameterization, FFT, LPC, PLP, MFCC, DTW, HMM, neural network, Matlab.

PROHLÁŠENÍ O ORIGINALITĚ

Prohlašuji, že jsem bakalářskou práci na téma Rozpoznávání řeči (číslice) vypracoval samostatně s využitím odborné literatury a pramenů, uvedených v Seznamu použité literatury.

Datum

Martin Kantar

BIBLIOGRAFICKÁ CITACE

KANTAR, M. *Rozpoznávání řeči (číslice)*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 32 s. Vedoucí bakalářské práce Ing. Radomil Matoušek, Ph.D..

PODĚKOVÁNÍ

Děkuji tímto panu Ing. Radomilu Matouškovi, Ph.D. za odborné rady a připomínky při vypracovávání bakalářské práce. Dále bych chtěl poděkovat svojí rodině za vytvoření příjemného prostředí domova, ve kterém byla práce tvořena.

Obsah:

SEZNAM POUŽITÝCH SYMBOLŮ

ANN	Artificial neural network
DTW	Dynamic Time Warping
FFT	Fast Fourier transform
HMM	Hidden Markov Model
LPC	Linear Prediction Coefficients
MAP	Maximum a posteriori estimatio
MATLAB	Matrix laboratory
MLLR	Maximum Likelihood Linear Regression
MFCC	Mel Frequency Cepstral Coefficients
PLP	Perceptual Linear Prediction
Purelin	Lineární funkce
Tansig	Hyperbolický tangens
VAD	Voice Activity Detectors

ÚVOD

Rozpoznávání řeči je rychle se rozvíjející oblast, skrývající velký potenciál pro usnadnění lidského života. Už v nepříliš vzdálené budoucnosti by mohlo nastat, že budeme vše ovládat pouhým hlasem. Takže by jednou mohla i odpadnou nutnost ručního zapisování textu. Jednoduše bychom si jen zapnuli určitý program, který by dokázal převádět vše - co slyší do textové podoby. Toto by mohlo jít až tak daleko, že by se program mohl i zaměřit pouze na jednu osobu a její mluvu převádět na text, za předpokladu, že ostatní mluvící osoby by byly ignorovány. Jedním ze směrů, kterým by se mohlo rozpoznávání řeči vyvíjet v budoucnu je využitím neuronových sítí, které mají slibný potenciál v oblasti umělé inteligence.

První úspěšné snahy o osvojení rozpoznávání řeči strojem sahají až do první poloviny dvacátého století. Kolem roku 1920 se podařilo dostat do prodeje první přístroj, který byl schopen při vyslovení určitého slova vykonat daný pokyn. Tento přístroj uměl ovšem reagovat pouze na jedno slovo popřípadě i jiné s podobnou frekvenční charakteristikou. V té době se ještě nejednalo o příliš úspěšný rozpoznávač, ale jeho úspěch mezi širokou veřejností určitě výrazně přispěl k dalšímu rozvoji v této oblasti [1].

V první kapitole vysvětluji, jak model řeči vypadá a čím je tento vzhled ovlivňován. Uvádím zde také, do jakých tříd si obecně řeč můžeme rozdělit.

V druhé kapitole uvádím co to parametrizace je a proč je tak důležitá pro rozpoznávač. Jsou zde předvedeny nejčastější způsoby, kterými se parametrizace provádí.

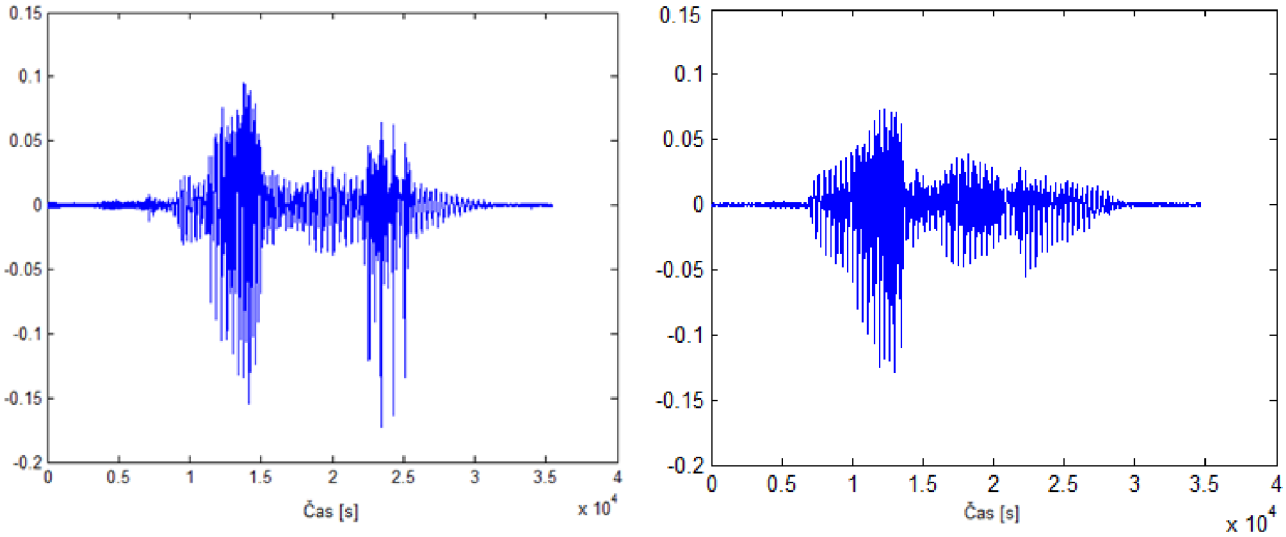
V třetí kapitole uvádím možnosti samotného způsobu rozpoznávání. Pro co se jaká metoda hodí a jak v nich probíhá samotné vyhodnocování.

Ve čtvrté kapitole realizuji rozpoznávač slov (číslic). Vysvětluji, proč jsem si zvolil tento typ parametrizace a metodu rozpoznávání.

1 MODEL ŘEČI

Nejdůležitější je si definovat samotnou řeč a faktory, které jí ovlivňují. Jedná se o nejběžnější způsob, kterým si lidé dovedou vyměňovat informace. Možná se to nezdá, ale i taková samotná řeč obsahuje v sobě více informací, než se na první pohled zdá. Lidský sluch se vyvíjel tisíce let, takže si z řeči bere jen ty důležité informace. Ale co například takový stroj na rozpoznávání řeči? Tomu evoluce příliš nepřála, proto mu musíme pomoc my a říct mu na co konkrétního se v řeči zaměřit – zvolením například vhodné parametrizace. Samozřejmě to není tak lehké, každý člověk mluví nepatrně jinak než ostatní lidé. Z tohoto důvodu tu vzniká problém spojený se složitostí rozpoznávání. Na mluvené řeči se podepisuje spousta faktorů, které řeč ovlivňují. Jsou to například: únava, nálada, intonace, věk, přízvuk, jestli jde o muže či ženu a spousta dalších věcí. Ani dokonce ten samý člověk nevysloví stejné slovo dvakrát po sobě vždy úplně stejně.

Na Obr. 1 je například znázorněno dvakrát slovo „jedna“ od stejného člověka s nepatrně pozměněnou intonací. U obrázku vpravo nebyl kladen dostačující důraz na výslovnost jako u obrázku nalevo. Samozřejmě se na rozdílu mohl podepsat i fakt, že pokud člověk bude opakovat nějaké slovo vícekrát za sebou, tak se bude zhoršovat kvalita jeho vyslovení.



Obr. 1 Zvukové záznamy číslovky „jedna“

1.1 Rozdělení řeči

Řeč se dá rozdělit do určitých tříd, ke kterým následně budeme snadněji moc stanovit pravidla a metodiku řešení. Nejčastěji se můžeme setkat s rozdělením snímané řeči na:

- Izolovaná slova - slova jsou vyslovována izolovaně. Jedná se spíše o krátké příkazy nebo jednoslovné odpovědi, protože mluvčí musí dělat pauzy mezi slovy.

- Spojená slova - slova jsou vyslovována za sebou s pomlkami. Dokáže mimo jiné si v reálném čase uvědomit, kdy slovo začíná a kdy končí.

- Plynulá řeč - nejtěžší na rozpoznávání. Nejčastěji se používá rozdělování jednotlivých slov na menší části, jako jsou například fonémy nebo slabiky, které následně rozpoznáváme [2].

2 PARAMETRIZACE

Při snaze vytvořit kvalitní rozpoznávač řeči nemůžeme pracovat s neupravenými vzorky, jako jsou například na Obr. 1, proto je potřeba nejprve zvukový signál upravit provedením tzv. procesu parametrizace. Tento proces nám slouží k redukci dat a k vytvoření nějaké charakteristické rozlišitelnosti mezi ostatními nahrávkami řeči. Chceme tedy získat co nejzajímavější informace z co nejmenšího množství dat. Pokud bychom parametrizaci neprovedli, tak by bylo pro rozpoznávač velice složití pracovat s tak velkým počtem neuspořádaných dat. Každá metoda parametrizace se snaží vytvořit jakési odlišení, které bude pro daný zvukový signál charakteristické.

Nejznámější a nejrozšířenější metody parametrizace v dnešní době jsou:

- Fourierova transformace
- Linear Prediction Coefficients (LPC)
- Perceptual Linear Prediction (PLP)
- Mel Frequency Cepstral Coefficients (MFCC)

Samozřejmě můžeme parametrizaci provést i jinými způsoby, popřípadě modifikací některé z těchto metod. Způsob zvolené metody se velmi často také odvíjí od metody, kterou chceme daná data následně vyhodnocovat.

2.1 Fourierova transformace

Fourierova transformace je algoritmus, používající matematických výpočtů k analýze zvukového signálu. Fourierova transformace vznikla z Fourierovy řady. Jejím úkolem je převedení diskrétního signálu z časové oblasti na frekvenční. Pro realizaci této parametrizace bylo použité prostředí Matlabu a jeho efektivnější forma – Rychlá Fourierova transformace (FFT), která je definovaná jako [3]:

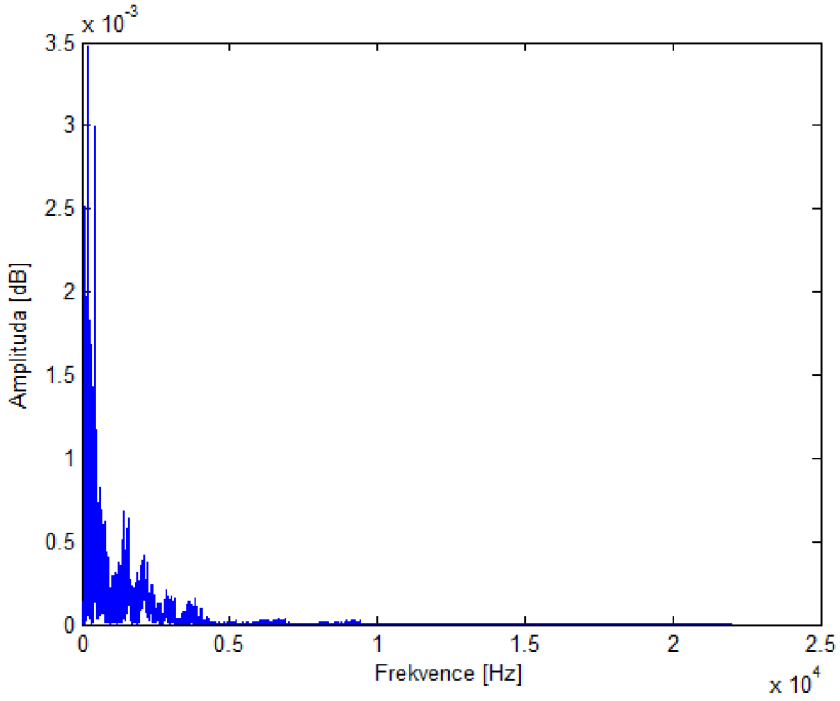
$$X(k) = \sum_{j=1}^{N} x(j)\varpi_N^{(j-1)(k-1)} \tag{1}$$

kde

$$\omega_N = e^{(-2\pi i)/N} \tag{2}$$

Implementace tohoto způsobu parametrizace je velmi jednoduchá. V dnešní době se samotná Fourierova transformace už příliš nepoužívá a pracuje se spíše s různými modifikacemi doplněnými o sady filtrů.

Pro názornost jsem na zvukový signál vlevo z Obr. 1 aplikoval Rychlou Fourierovu transformaci. Jak je patrné z Obr. 2. Výsledná data už vypadají mnohem srozumitelněji pro potencionální rozpoznávač – pohybujeme se v kladných hodnotách, došlo zde k drobné redukci dat a objevují se tu už první známky charakteristického rysu daného slova. Ovšem tento způsob obsahuje stále příliš mnoho zbytečných údajů na velmi krátké frekvenční oblast, proto by bylo vhodné použít ještě po FFT vhodný filtr, který by tento problém dokázal odstranit.

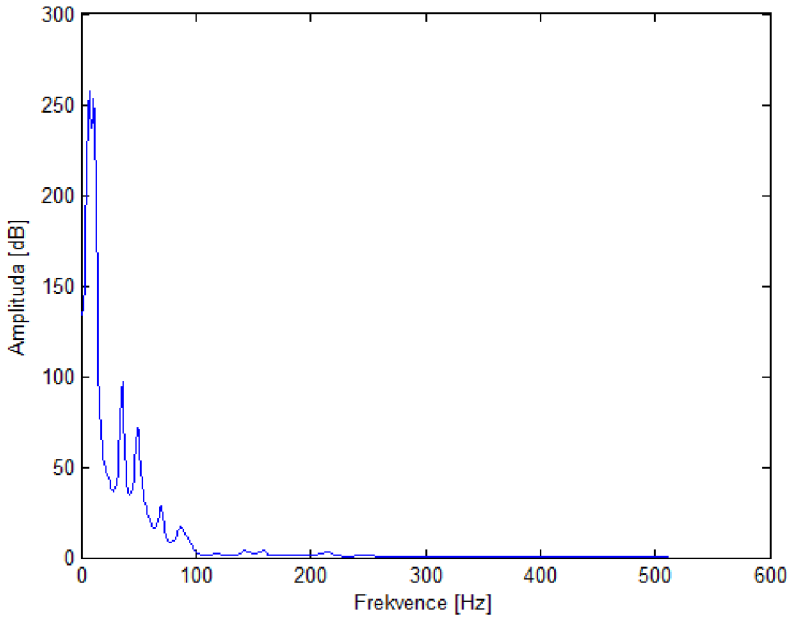


Obr. 2 Číslovka „jedna“ po FFT

2.2 *Linear Prediction Coefficients (LPC)*

Jak už název napovídá, jedná se o předpovídání vzorku signálu jako lineární kombinaci. Tato úprava se v podstatě provádí tak, že si nahrazujeme nelineární části vzorku lineárními, což vede k výrazné redukci dat. Při této metodě dochází také k potlačení drobných odchylek vlivem filtru a k zvýraznění charakteristických rysů signálu. Hlavní nevýhodou této metody je špatná odolnost proti hluku, která nám může nepříznivě ovlivnit charakteristické znaky pro daný signál.

Pro názornost jsem opět na zvukový signál vlevo z Obr. 1 aplikoval LPC parametrizaci v prostředí Matlab. Nejprve bylo za potřebí na zvukový signál použít LPC filtr s daným řádem polynomu, následně tato data převést na frekvenční charakteristiku a dále na data použít absolutní hodnotu [4]. Takto získaná data jsou vykreslena na Obr. 3. S takovýmito daty by mohl rozpoznávač hlasu už bez problémů pracovat.



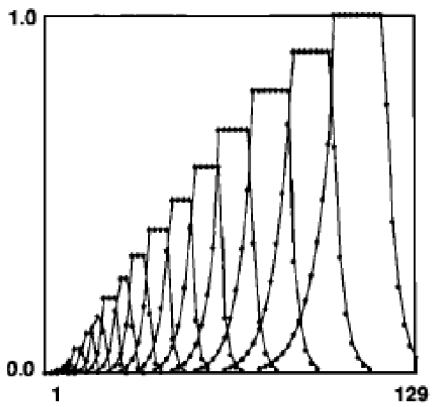
Obr. 3 Číslovka „jedna“ po LPC

2.3 *Perceptual Linear Prediction (PLP)*

Tato metoda, i když se to nezdá, je velmi podobná metodě LPC, je ovšem rozšířená o schopnost přizpůsobení se lidskému sluchu, takže se zde pracuje se spektrální analýzou. Schopnost přizpůsobit se, vychází z poklesu spektrální rozlišitelnosti s rostoucí frekvencí a z faktu, že sluch je nejcitlivější ve středu slyšitelné frekvenční oblasti. Lineární frekvenční osa je zde tedy převedena do Barkovy frekvenční osy pomocí vztahu:

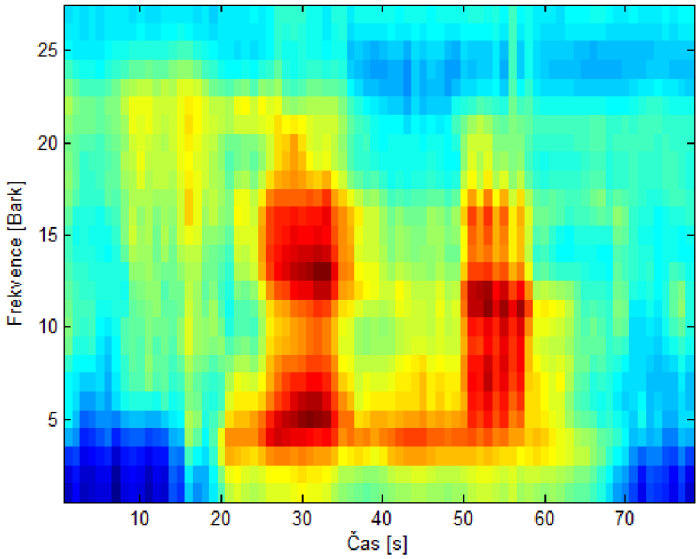
$$\Omega(f) = 6\,ln\left(\frac{f}{600} + \sqrt{\left(\frac{f}{600}\right)^2 + 1}\right) \tag{3}$$

Následně je zavedena po aproximaci Barkova banka filtrů Obr. 4 [5].



Obr. 4 Barkova banka filtrů z [5]

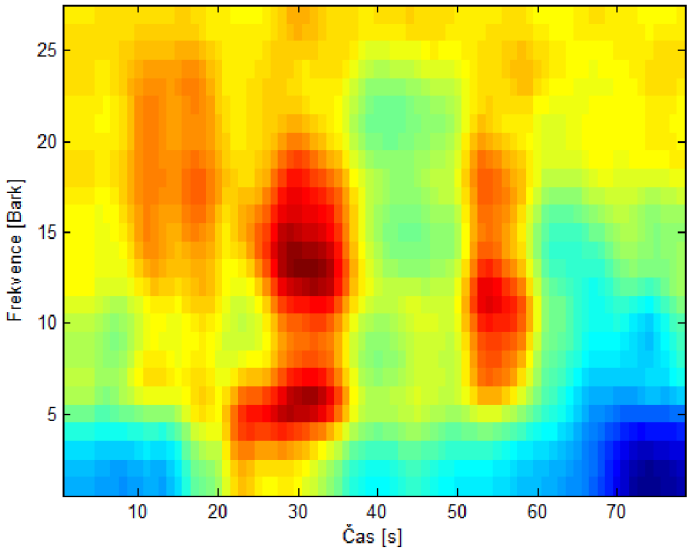
Pro představu spektra, jsem znovu na zvukový signál vlevo z Obr. 1 aplikoval PLP parametrizaci v prostředí Matlab, jak jde vidět na Obr. 5. Bohužel by bylo příliš složité takováto spektra mezi sebou porovnávat, protože se jedná o velmi datově obsáhlé matice. Proto by bylo zapotřebí vhodně převést spektrum na kepstrální koeficienty, které nám dává například samotná LPC parametrizace. Po tomto převedení se snažíme mít co nejméně údajů s nejvíce důležitými informacemi. Velmi často se proto může provést derivace koeficientů pro vytvoření větší odlišnosti mezi jednotlivými řečovými záznamy. Koeficienty po derivaci se nazývají delta koeficienty. Můžeme se setkat také s koeficienty získanými druhou nebo třetí derivací.



Obr. 5 Spektrum číslovky „jedna“ po PLP

2.3.1 Rasta – PLP

Tento způsob je pouze modifikací běžné PLP metody. Obsahuje větší odolnost při lineárně spektrálním rušení filtrováním nízkofrekvenční složky [6]. Pro porovnání s běžnou PLP metodou jsem znovu na signál vlevo z Obr. 1 aplikoval Rasta – PLP parametrizaci, jak je patrné z Obr. 6.



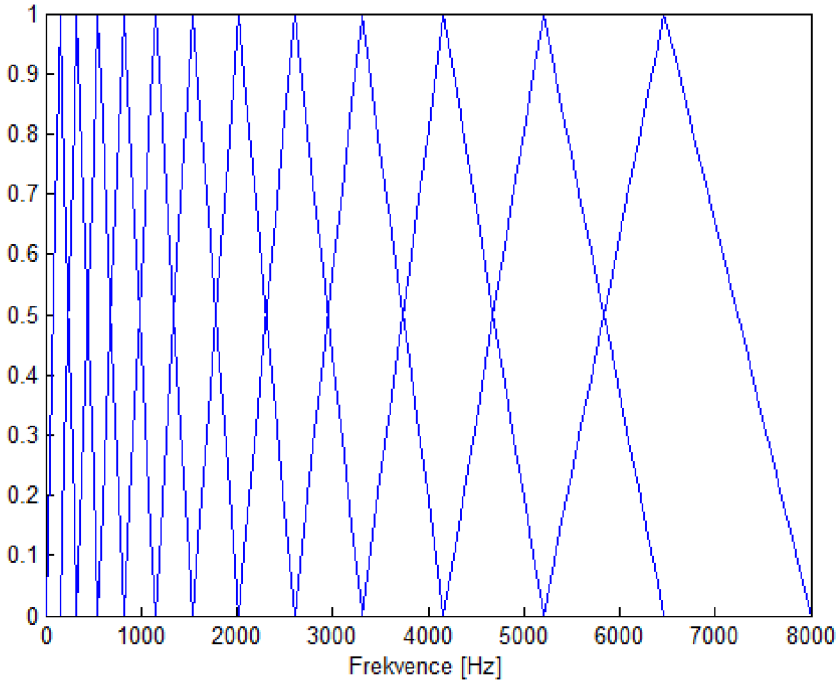
Obr. 6 Spektrum číslovky „jedna“ po RASTA-PLP

2.4 *Mel Frequency Cepstral Coefficients (MFCC)*

Tato metoda pracuje podobně jako PLP s vlastností přizpůsobování se lidskému sluchu. Takže se zde provádí spektrální zobrazení. Oproti PLP metodě se zde ale převádí frekvenční osa z herců na osu v melech. Pro tento převod se používá vztah:

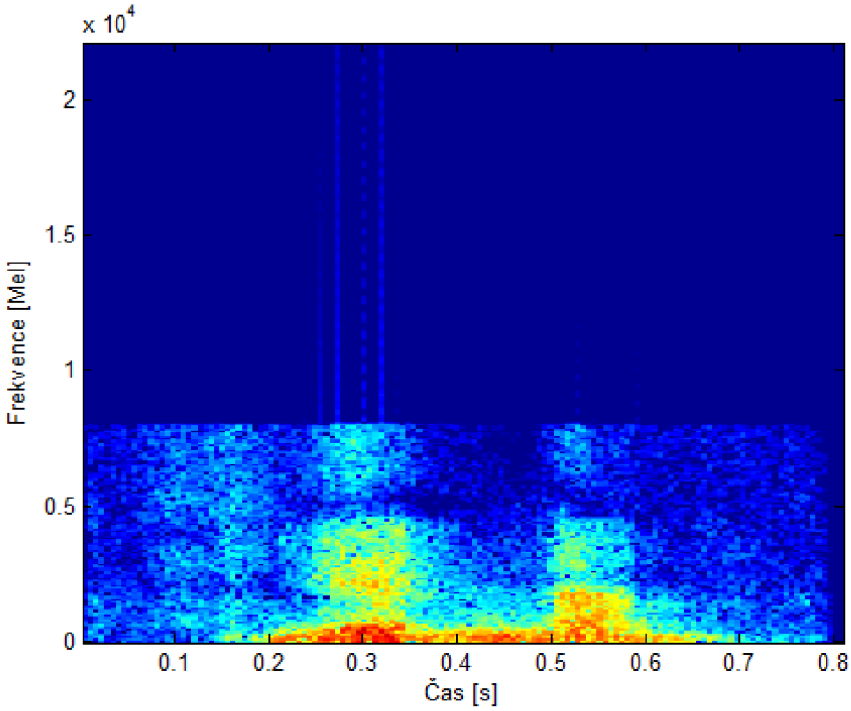
$$f_{mel} = 2595 \cdot log_{10}\left(1 + \frac{f_{HZ}}{700}\right) \tag{4}$$

Po tomto převodu se na vzorek aplikuje melovská banka filtrů, která je převedena na lineární frekvenční měřítko (Obr. 7). Trojúhelníky v melovské bance filtrů mají stejnou šířku pásma a vzájemné padesátiprocentní překrytí mezi sebou [7], [8].



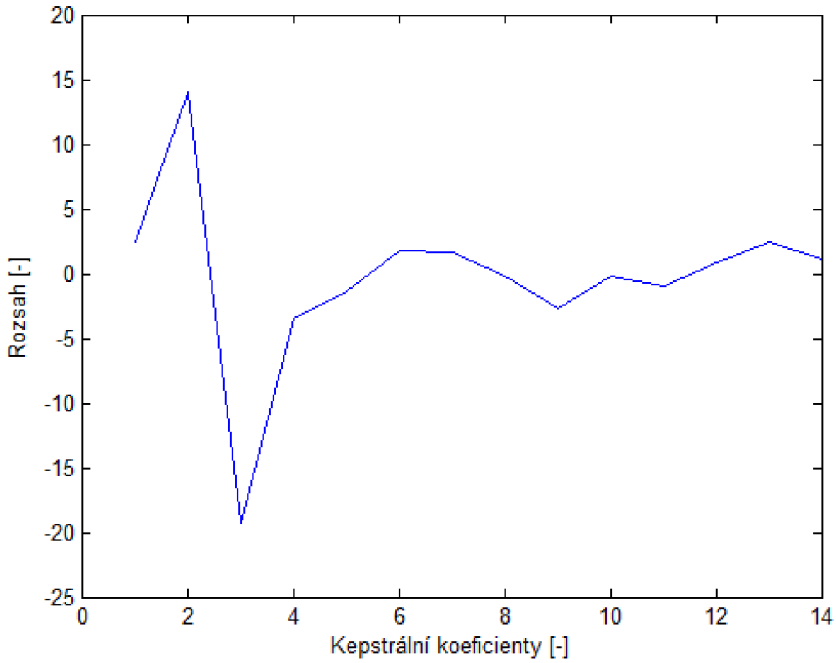
Obr. 7 Melovská banka filtrů

Se získaným spektrem (Obr. 8), stejně tak jako u PLP, by se velmi obtížně pracovalo, proto je potřeba spektrum převést na kepstrální koeficienty a následně například na delta koeficienty.



Obr. 8 Číslo „jedna“ po MFCC

Pouze pro ilustraci jak by takovéto kepstrální koeficienty mohly vypadat u MFCC. Na Obr. 9 je jeden ze způsobů realizace [9].



Obr. 9 Převedené spektrum na kepstrální koeficient [9]

3 METODY ROZPOZNÁVÁNÍ

V dnešní době existuje několik metod, kterými se dá rozpoznávat řeč. Mezi nejznámější metody patří Dynamické Borcení času (DTW) a Skryté Markovovy Modely (HMM). Každá z těchto metod pracuje na jinačím principu a s jiným typem dat. Další alternativou, kterou se dá rozpoznávat řeč je metoda využívající Neuronové sítě. Existují i kombinace více metod dohromady pro lepší přesnost.

Důležitou věcí spojenou s metodami rozpoznávání je i to, zda jsou namluvené nahrávky od stejné osoby, jako od osoby, jejichž zvukový vzorek chceme rozpoznávat. Podle tohoto si můžeme rozpoznávání rozdělit dle řečníka na:

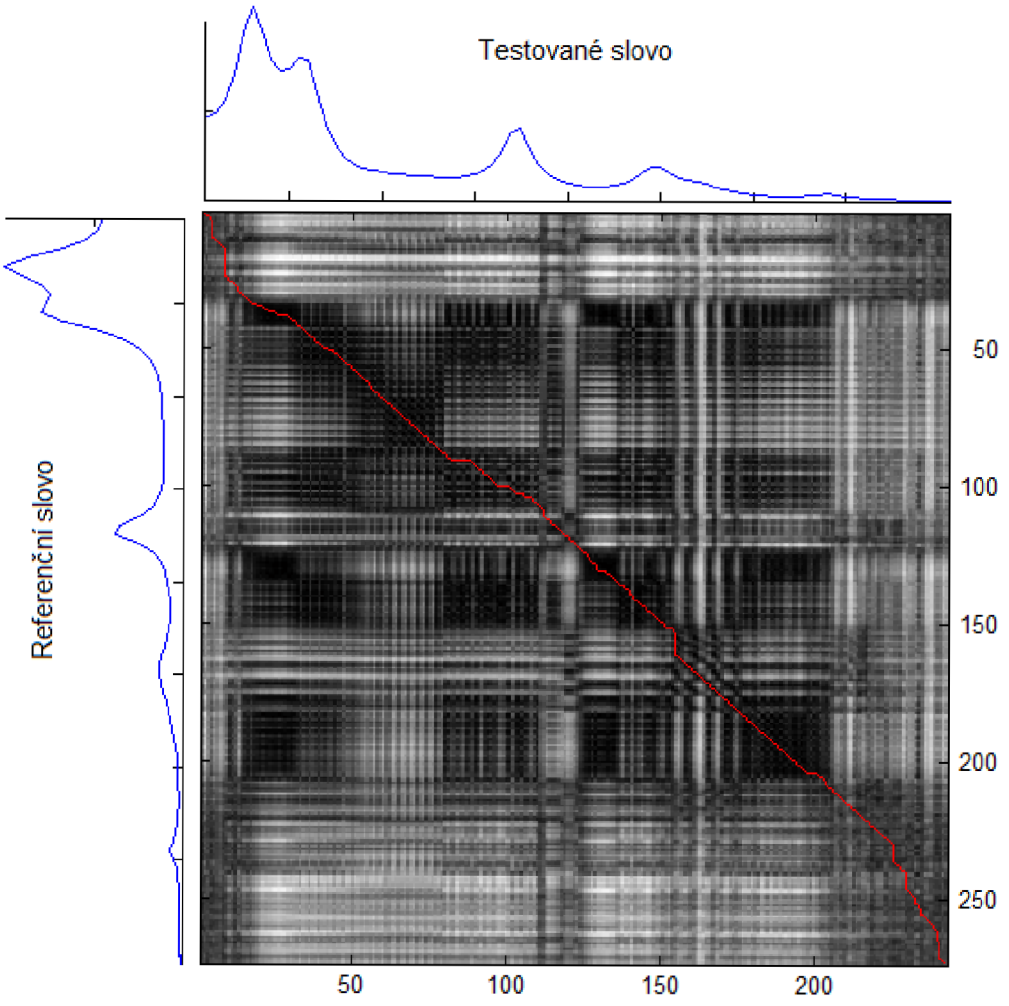
- Závislé na řečníkovi – dobrá varianta pro rozpoznávání u jedné osoby, na jehož hlas je rozpoznávač natrénován. S rozpoznáváním hlasu jiných osob jsou mnohem horší výsledky.

- Nezávislé na řečníkovi – dobrá varianta u rozpoznávání pro více osob, i u kterých se netrénovalo. Jeho potenciál tkví v rozmanitosti trénovaných vzorků. V dnešní době se problém s nerozpoznatelností řeší výpočetními operacemi – převáděním hlasu na nejvíce podobný hlas v databázi. Tyto nejčastější přizpůsobovací metody jsou MAP a MLLR [10].

3.1 Dynamické Borcení Času (DTW)

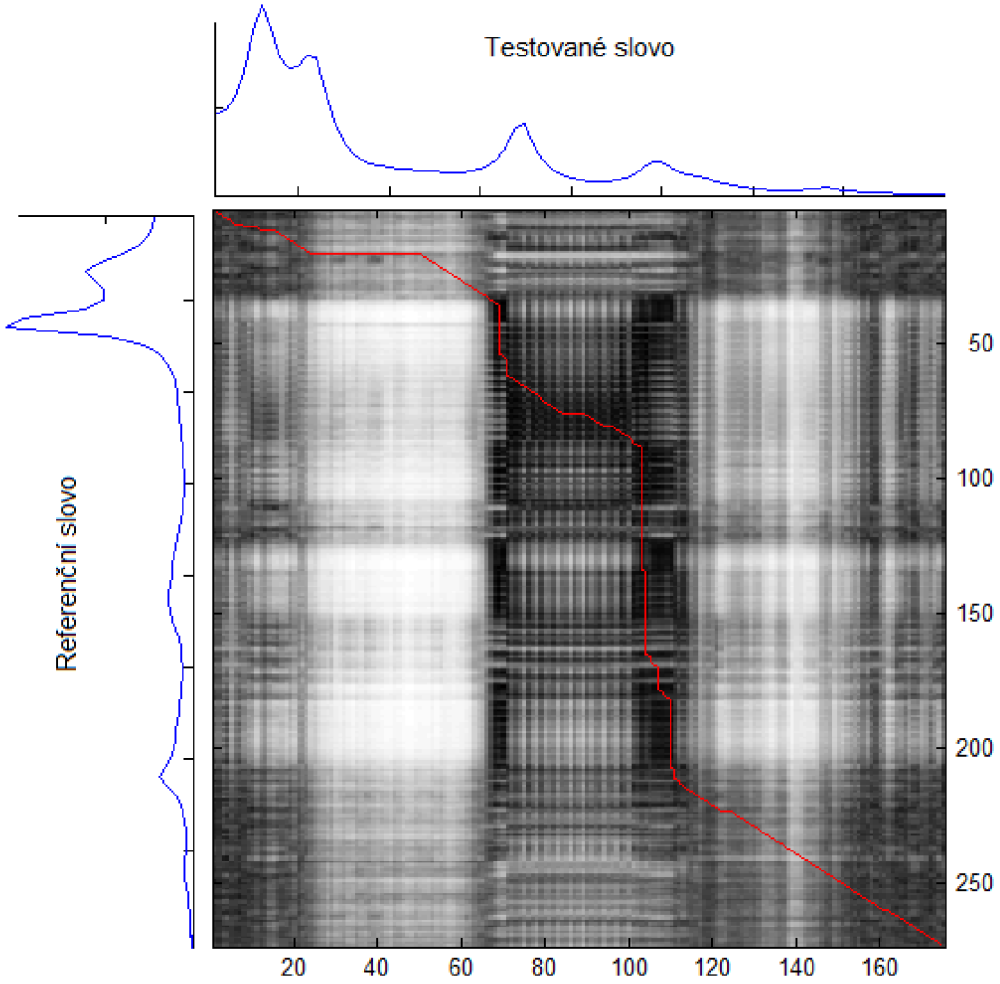
Tato metoda spočívá v porovnávání zvukových vektorů, které získáme z matic pro jednotlivé nahrávky. Dochází zde k hledání největší podobnosti mezi dvěma slovy, a to referenčního slova, které se nachází v databázi namluvených slov, se slovem testovaným, které je pro nás neznáme. Princip spočívá vytvořením si tzv. cesty přes spektrální oblast, která nám znázorňuje rozdíly v jednotlivých částech mezi referenčním slovem a testovaným. Tyto cesty jsou patrné na Obr. 10 a Obr. 11. Samozřejmě zde dochází k výpočtu optimální cesty od počátku ke konci s největší podobností. Čím je tato cesta menší, tím je větší podobnost mezi danými zvukovými vzorky.

Tato metoda se používá především k rozpoznávání izolovaných slov, kde dosahuje výborných výsledků. Rozpoznávač je potřeba vybavit detektorem řečové aktivity (například VAD). Hlavní nevýhodou této metody je potřeba velké databáze namluvených slov. Těchto slov nesmí být ovšem příliš moc, protože by se nám to nepřijatelně promítlo do doby vyhodnocování testovaného slova [11], [12].



Obr. 10 Číslovka „jedna“ a „jedna“ v DTW

Na Obr. 10 je zobrazeno využití metody DTW v Matlabu. Testované slovo „jedna“ bylo porovnáno s náhodným slovem „jedna“ z databáze. Cesta se snaží nalézt nejkratší cestu od počátku do konce po co nejvíce tmavých místech – čím tmavší místo, tím větší shoda. Nejkratší cesta u tohoto porovnání činila 29.6035 jednotek.

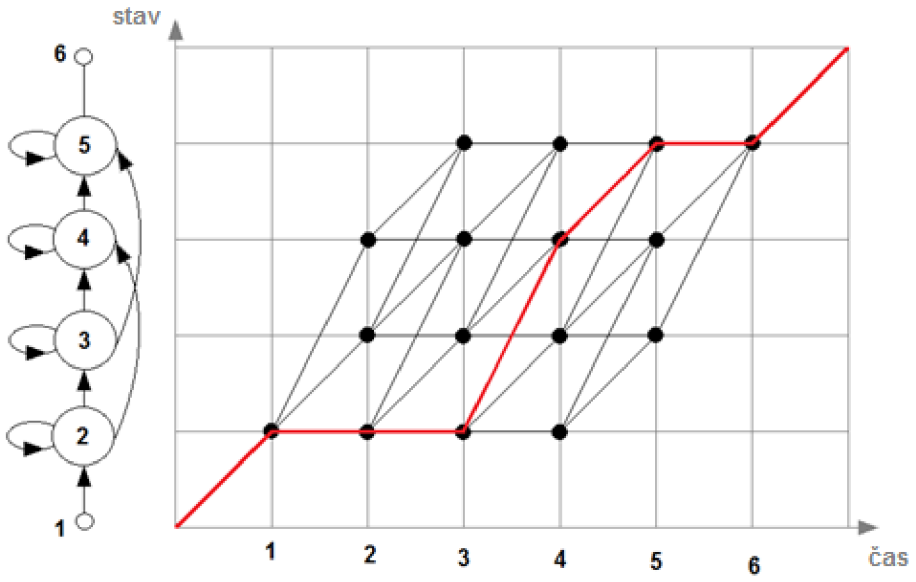


Obr. 11 Číslovka „jedna“ a „šest“ v DTW

Na Obr. 11 byla použita opět stejná metoda a realizace jako u předchozího názorného příkladu. Tentokrát bylo ovšem testované slovo „jedna“ porovnáno se slovem „šest“ z databáze. Jak je patrné, tak tmavých míst na hlavní diagonále rapidně ubylo a to vedlo k mnohem kostrbatější cestě. Velikost nejkratší cesty v tomto příkladu je 72.3946 jednotek.

3.2 Skryté Markovovy Modely (HMM)

Jedná se o statickou metodu, která simuluje proces, při kterém dochází k přechodu mezi jednotlivými stavy. Při každém přechodu je vypočítána tzv. přechodová pravděpodobnost, že daný stav nastane. Jedná se v podstatě o automat s konečným počtem stavů, kdy s rostoucím časem přecházíme mezi jednotlivými stavy a snažíme se najít nejpravděpodobnější cestu viz Obr. 12.



Obr. 12 Přecházení mezi stavy

Čekání na výpočet všech cest by mohlo být ovšem příliš zdlouhavé, proto se používá Viterbiho metoda, která provádí eliminaci nepravděpodobných přechodů v časových úsecích.

Tato metoda je vhodná pro rozpoznávání plynulé řeči. Pro rozpoznávání izolovaných slov potřebuje rozsáhlé databáze. Princip metody rozpoznávání plynulé řeči spočívá v rozložení slov na ještě menší části. Mohou to být například fonémy (hlásky), slabiky, difóny nebo trifóny. Tyto menší části se trénují zvlášť a jejich výhodou je, že dovedou v malém množství nahrazovat rozsáhlé databáze namluvených slov. Je to jako bychom chtěli rozpoznávat slova pomocí jednotlivých písmen ve slově. Při rozpoznávání se pak hledá nejpravděpodobnější shoda daných fonémů v databázi. Při vyhodnocování je zahrnuta i pravděpodobnost přecházení mezi jednotlivými fonémy, kde každý foném reprezentuje jeden stav, takže je například nepravděpodobné, aby dva stejné fonémy byli hned za sebou [13].

3.3 Neuronové sítě (ANN)

Neuronové sítě jsou účelný prostředek pro znázornění matematických procesů, které jsou navrženy z vlastností biologických principů. Jejich široké možnosti využití se zakládají na možnosti predikovat procesy. Neuronové sítě se skládají z paralelně propojených prvků, které znázorňují jednotlivé neurony. Tyto neurony, stejně tak jako lidské neurony, se umí učit postupným trénováním. Neurony si mezi sebou poskytují data a přidělují jim váhy, jak jsou v daném řešení důležité. Výhodou neuronových sítí je lepší práce s horší kvalitou zvuku, proto v jistých případech mohou dosahovat i větší přesnosti než například HMM [14]. V práci nejsou neuronové sítě dopodrobna vysvětlené, protože se jedná o velmi rozsáhlé téma. Uvedl jsem zde pouze základní věci, se kterými dále pracuji.

3.3.1 Topologie sítě

Topologie nám určuje, jak jsou neurony mezi sebou propojeny. Nejčastěji jsou neurony uspořádány ve vrstvách, kde je každý neuron propojen s každým. Všechny sítě mají svojí vstupní a výstupní vrstvu. U jednovrstvých sítí je vstupní vrstva současně výstupní [14].

3.3.2 Způsoby učení

Nejdůležitější složkou neuronových sítí je samotná možnost učení. Učení se provádí pomocí učícího algoritmu, který dokáže během iterací měnit váhy pro jednotlivé spoje. Podle způsobu učení můžeme síť rozdělit na:

- **učení s učitelem** - jedná se o učení, kdy jsou neuronové síti poskytnuty příklady a k nim příslušné správné výsledky, na nichž se síť učí. Pokud se příklad vzdaluje od výsledku, je mu přidělaná nízká váha. Samozřejmě toto platí i obráceně. Tato operace s přidělením vah se provede pro všechny prvky. Muže zde dojít také k tzv. přeučení. Jedná se o stav, kdy po opakovaném a opakovaném učení dochází k výraznému zjednodušení. Při přílišném zjednodušení můžeme poté dostávat v lepším případě nepřesné výsledky, v horším – úplné hlouposti.

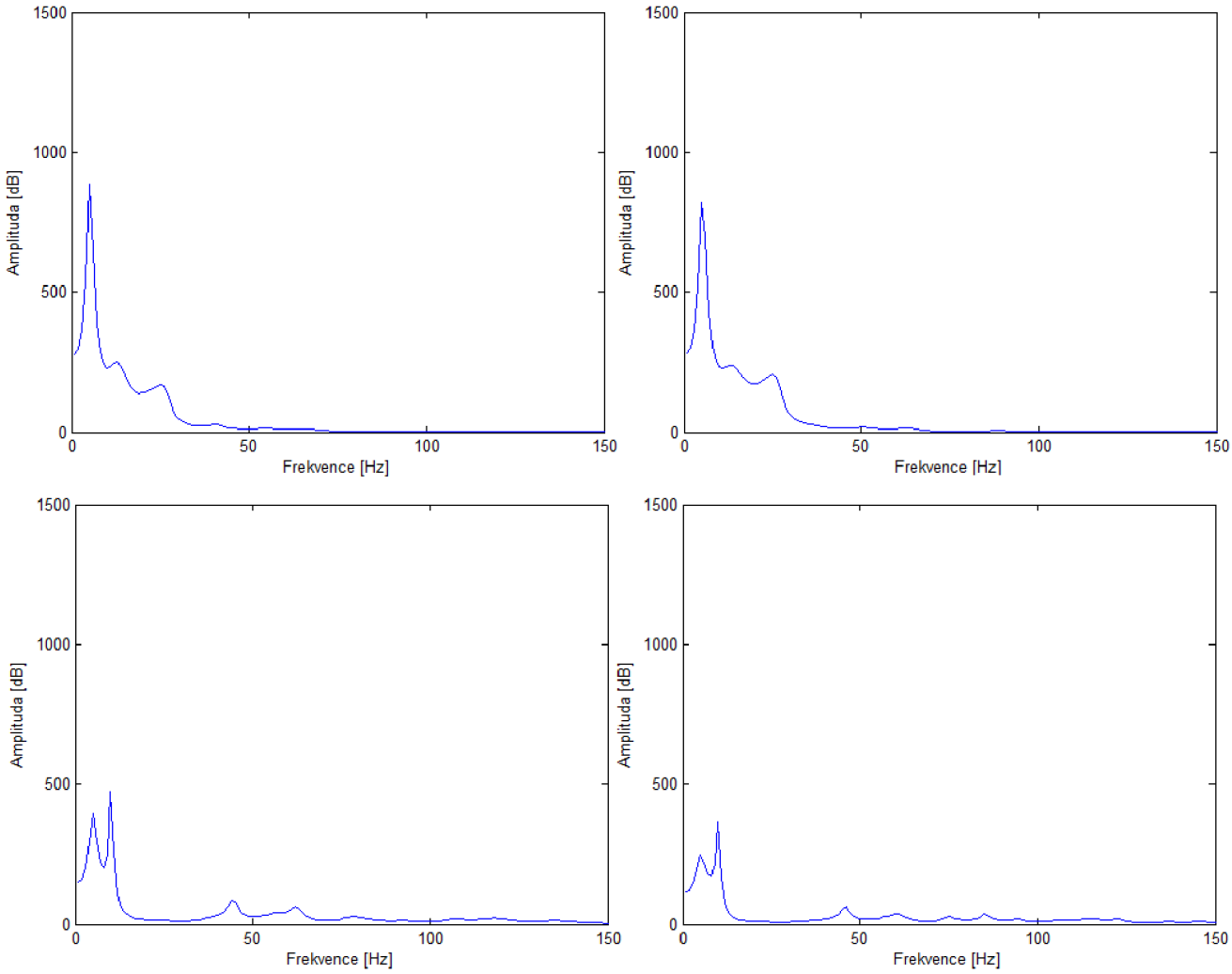
- **učení bez učitele -** v tomto způsobu jsou síti předkládané příklady bez výsledků. Absence výsledků je zde nahrazena znalostí kritéria – popřípadě vzoru, podle kterého je příklad řešen. Váhy se zde upravují podle daného vzoru [15].

4 REALIZACE ROZPOZNÁVAČE SLOV (0-9)

Pro realizaci bylo určeno, že se bude jednat o rozpoznávání 10 slov, která budou závislá na mluvčím. Podle těchto kritérií jsem zvolil způsob parametrizace a metodu řešení rozpoznávání.

4.1 Výběr parametrizace

Způsob parametrizace jsem zvolil LPC, pro její snadnou implementaci na velké množství dat. Hlavním důvodem byla ovšem výrazná odlišitelnost jednotlivých slov mezi sebou, viz Obr. 13. Na tomto obrázku jsou v horní části znázorněné dvě parametrizovaná slova „nula“. Na dolní části jsou zobrazena dvě slova „čtyři“. Už z prvního pohledu jsou patrné výrazné rozdíly mezi odlišnými slovy. Takovéto rozdíly se dají najít i u jednotlivých slov mezi sebou. Při výběru vhodných parametrů jsem pracoval s frekvencí pouze do 150 Hz, protože za touto hranicí byla frekvence už velmi nezajímavá a pouze jen těsně kolísala nad hranicí nuly.



Obr. 13 Odlišnost mezi slovy

4.2 *Výběr metody rozpoznávání*

Metodu, kterou budu data rozpoznávat, jsem si vybral neuronové sítě. Neuronové sítě mají výhodu v tom, že dovedou relativně dobře pracovat s šumem poškozenými daty. Kdežto jak jsem už zmínil v kapitole LPC (3.2.), tak LPC má s šumem problémy. V tomto způsobu řešení se tedy s problémem špatného vyhodnocení zvukové nahrávky vlivem šumu téměř nesetkáme.

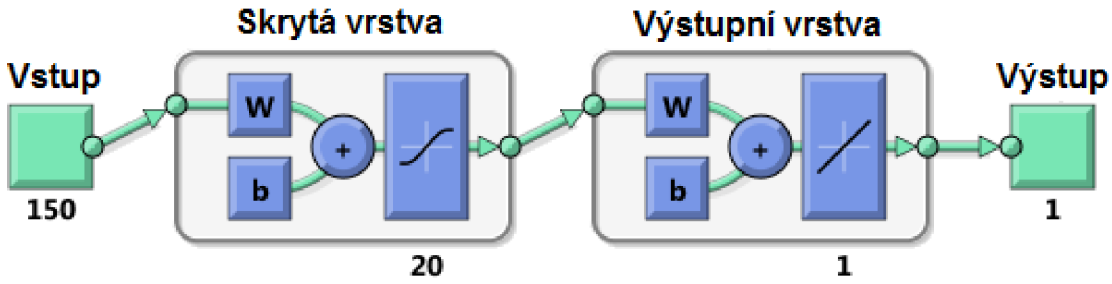
Realizaci metody jsem provedl v programové prostředí programu Matlab, pro který existuje rozšiřující Neural Network Toolbox. Tento Toolbox rozšiřuje Matlab o možnost navrhování, vizualizaci, realizaci a simulaci neuronových sítí.

4.2.1 Nastavení neuronové sítě

Způsob učení neuronové sítě jsem vybral učení s učitelem. Neuronovým sítím jsem tedy poskytl relativně velké množství nahrávek a k nim jejich správné výsledky. Každé slovo zde bylo zastoupeno rovným stem nahrávek. Při řešení s menším počtem nahrávek by bylo velmi obtížné neuronovou síť správně natrénovat, proto jsem se rozhodl k tomuto množství.

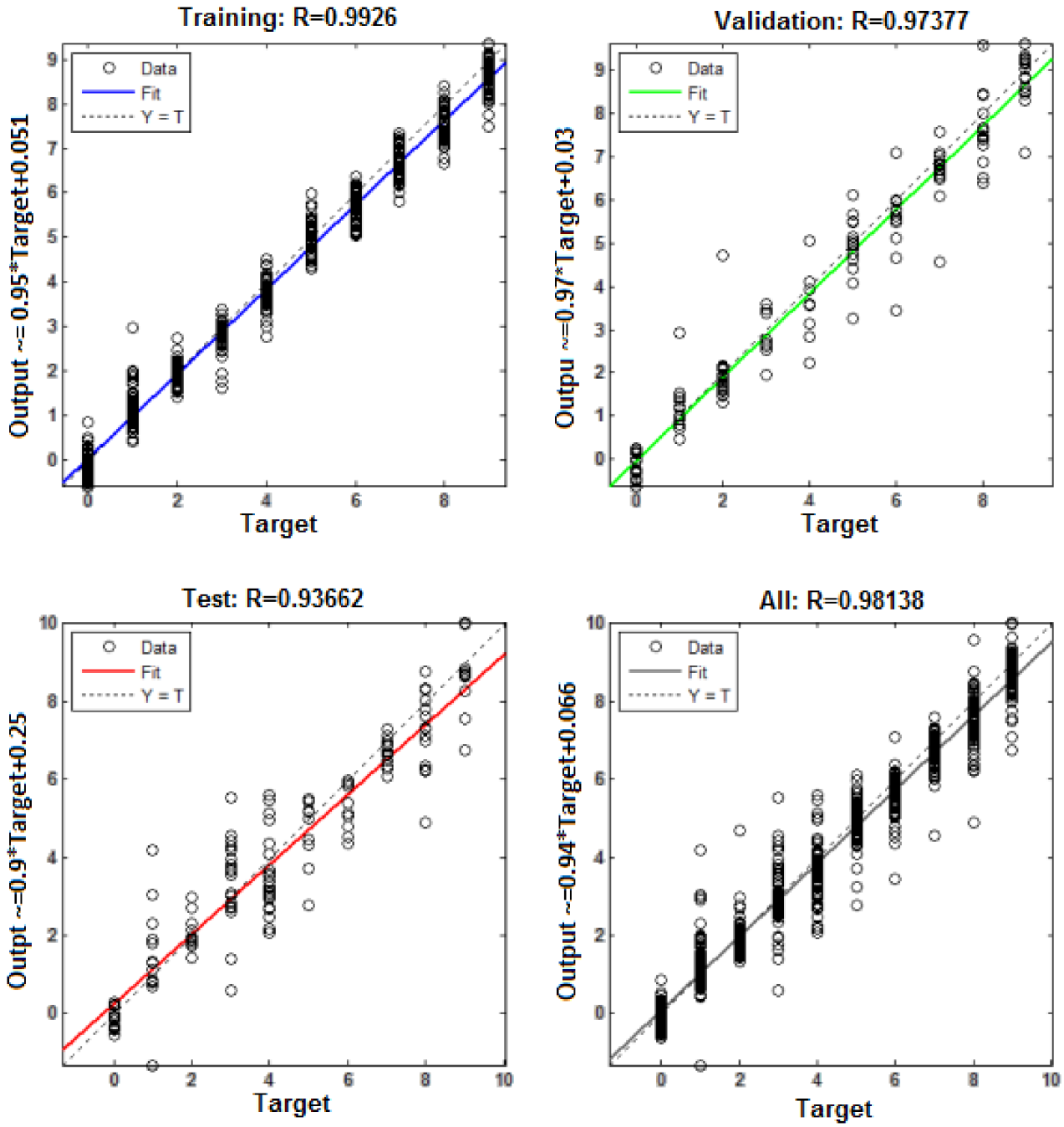
Po připravení vstupních a výstupních dat je dále potřeba si vybrat v jakém množství chceme naše data trénovat, validovat a testovat. Data určená k trénování jsou síti předávány během trénování a síť se podle nich nastavuje. Validační data slouží k zjištění jakou má síť schopnost zobecňovat. Pokud nebude možné dále síť zobecňovat, tak se trénování zastaví. Testovaná data poskytují pouze měřítko výkonnosti sítě během a po tréninku, nemají žádný vliv na trénování. Při rozdělení jsem přiřadil trénování 70% dat, validaci a testování po 15%.

Pro tuto síť jsem se rozhodl pro vrstvy. Vstupní vrstva obsahuje 20 neuronů s aktivační funkcí Tansig a výstupní jeden neuron s funkcí Purelin. 20 neuronů jsem zvolil z důvodu nejlepších výsledku při porovnání trénování sítě a zatěžování počítače. Znázornění vytvořené sítě je na Obr. 14.



Obr. 14 Model neuronové sítě

Při trénování byl použit Levenberg Marquardtův učící algoritmus, který se velmi často v oblasti neuronových sítí používá. Po dokončení trénování si můžeme graficky zobrazit, jak bylo trénování úspěšné, viz Obr. 15. Na obrázku jsou výsledky regrese pro trénovaná, validační, testovaná a celková data. U jednotlivých výsledků se snažíme dostat regresi co nejblíže k číslu 1. Čím blíže se dostaneme k hodnotě jedna, tím jsme síť lépe natrénovali. Proces trénování můžeme opakovat do doby, než dostaneme uspokojivé výsledky. Výsledná regrese tohoto trénování vyšla 0.98138, což je velmi obstojný výsledek u takového množství dat. Následně můžeme tedy přejít k vyzkoušení rozpoznávání v praxi.



Obr. 15 Výsledky regrese

4.3 *Výsledky rozpoznávání*

Pro zjištění funkčnosti jsem vytvořil databázi 30 „neznámých“ slov, kde bylo každé slovo (číslo) zastoupeno třikrát. Po vyhodnocení rozpoznávačem bylo zapotřebí tento výsledek ještě zaokrouhlit na nejbližší celé číslo pro získání konkrétního výsledku, viz Tab. 1.

Neznámé číslo	Výsledky rozpoznávače	Zaokrouhleno
0	-0.1611	0
0	0.0019	0
0	-0.4074	0
1	0.7218	1
1	2.0670	2
1	1.2236	1
2	2.0817	2
2	1.6292	2
2	2.1486	2
3	2.6823	3
3	2.4642	2
3	2.6728	3
4	3.9184	4
4	3.7511	4
4	3.8394	4
5	4.5824	5
5	4.7461	5
5	4.6932	5
6	5.3540	5
6	5.6574	6
6	5.0202	5
7	6.8364	7
7	4.5336	5
7	7.1524	7
8	8.3022	8
8	8.0838	8
8	7.6890	8
9	8.8850	9
9	9.2323	9
9	5.9688	6

Tab. 1 Výsledky rozpoznávače

5 ZÁVĚR

V bakalářské práci jsem zhodnotil dnešní typy parametrizace řečových nahrávek. Jaké má každá parametrizace výhody a kdy je vhodné jí použít.

Zmínil jsem zde nejčastější dvě metody, a jednu alternativní metodu, kterými se řeč dá v dnešní době rozpoznávat a upozornil na výhody a nevýhody jednotlivých metod.

V závěru bakalářské práce jsem vybral metodu a způsob parametrizace, které jsem posléze využil při pokusu o realizaci schopného rozpoznávače slov (číslic) 0-9. Nejprve jsem si vytvořil databázi jednotlivých slov. Pohyboval jsem se na začátku kolem 45 nahrávek na jedno slovo. Matlabovská neuronová síť, ale nebyla schopná se úspěšně učit na tak malém množství dat a výsledky rozpoznávání byly zcela nesmyslné. Z tohoto důvodu jsem databázi nahrávek pro jednotlivá slova rozšířil na 90. V celkovém výsledku byl tedy rozpoznávač trénován na 900 vzorcích. Toto číslo mohlo být i klidně menší a dosahovat stejné úspěšnosti, protože slova s výrazně odlišnou charakteristikou od jiných mohla být zastoupeno mnohem méněkrát. Já jsem ale už měl slova namluvená a nebyl důvod je tedy nevyužít. Výsledky z takto vytvořeného rozpoznávače jsou zobrazeny v Tab. 1. Z těchto 30 neznámých slov se podařilo správně určit 24 vzorků, u 4 vzorků byly výsledky téměř správně, lišili se většinou přibližně v řádu 10%. U dvou vyhodnocení byly výsledky zcela milné. Z výsledků, které byly dosaženy, bych schopnost tohoto rozpoznávače rozpoznávat i přes dobré chyby označil za velmi dobrou.

Při snaze zjistit proč došlo k chybám a odchylkám jsem dospěl k závěru, že tyto nepřesnosti byly způsobeny intonací. Při výrazné změně intonace může totiž dojít ke zkreslení údajů vzorku, kdy má neznámé slovo charakteristiku jako úplně jiné. Možným řešením by mohlo být, kdybychom neuronovou síť učili i na slovech o různých intonacích. Dalším a mnohem úspěšnějším možným řešením by bylo kladení většího důraz na výslovnost u neznámých slov.

SEZNAM POUŽITÉ LITERATURY

[1] Gold, B., Morgan, N., Ellis D. Speech and Audio Signal Processing: Processing and Perception of speech and Music. [s.l.] : Wiley, 1999.560 s. ISBN 978-0-470-19536-9.

[2] Images SI, Inc. Images Scientific Instruments [online]. 2010 [cit. 2012-22-5]. Dostupné z: http://www.imagesco.com/articles/hm2007/SpeechRecognitionTutorial02.html

[3] MatWorks. Fast Fourier transform. MathWorks [online]. 2012 [cit. 2012-22-5]. Dostupné z: http://www.mathworks.com/help/techdoc/ref/fft.html

[4] MatWorks. Linear prediction filter coefficients. MathWorks [online]. 2012 [cit. 2012-22-5]. Dostupné z: http://www.mathworks.com/help/toolbox/signal/ref/lpc.html

[5] H. Hermansky: „Perceptual linear predictive (PLP) analysis for speech.“ J. Acoust. Soc. Am., pp. 1738-1752,1990

[6] Hermansky,H. Morgan, N. Bayya, A. Kohn, P. RASTA-PLP SPEECH ANALYSIS. December 1991 [cit. 2012-22-5]. Dostupné z: http://www.icsi.berkeley.edu/pubs/techreports/tr-91-069.pdf

[7] Automatic Sign Language Detection. Mel Frequency Cepstral Coefficients Analysis [online] 2008 [cit. 2012-22-5]. Dostupné z: http://sites.google.com/site/autosignlan/algorithms-used/mfcc

[8] Milner, B. Shao, X. SPEECH RECONSTRUCTION FROM MEL-FREQUENCY CEPSTRAL COEFFICIENTS USING A SOURCE-FILTER MODEL [online]. 8th of December.2002 [cit. 2012-22-5]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.62.1968&rep=rep1&type=pdf

[9] Juráček, D. Klasifikátor izolovaných slov na bázi umělé neuronové sítě. [online]. 2005 [cit. 2012-22-5]. Dostupné z: http://dsp.vscht.cz/konference_matlab/matlab05/prispevky/juracek/juracek.pdf

[10] Wikipedie. Rozpoznávání řeči [online]. 2012 [cit. 2012-22-5]. Dostupné z: http://cs.wikipedia.org/wiki/Rozpoznávání_řeči

[11] Müller, M. Dynamic Time Warping. [online]. 2007 [cit. 2012-22-5]. Dostupné z: http://www.springer.com/cda/content/document/cda_downloaddocument/9783540740476-c1.pdf?SGWID=0-0-45-452103-p173751818

[12] Salvadorm, S. Chan, P. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time ans space. [online]. 6th of August 2004 [cit. 2012-22-5]. Dostupné z: http://www.cs.fit.edu/~pkc/papers/tdm04.pdf

[13] Smékal, Z. Atassi, H. Stejskal, V. Mekyska, J. Soubor programů pro práci se skrytými Markovovými modely (HTK) [online]. 5.3.2009 [cit. 2012-22-5]. Dostupné z: http://elektrorevue.cz/file.php?id=200000288-a19dba297a

[14] Kačenka, P. Neuronové sítě [online]. 2010 [cit. 2012-23-5]. Dostupné z: http://mks.mff.cuni.cz/library/NeuronoveSitePK/NeuronoveSitePK.pdf

[15] Veselovský, M. Neuronové sítě [online]. [cit. 2012-22-5]. Dostupné z: http://avari.cz/uir/index.php

PŘÍLOHA

Pro ukázku zde mám přiložený zdrojový kód vygenerovaný Matlabem. Po vložení tohoto kódu do příkazového řádku Matlabu bude zahájen proces trénování. Potřebné jsou pouze dvě věci. Vstupní data, která jsou zde zastoupena pod jménem *vstup* a jejich výsledky pod jménem *vysledky*. Po natrénování je potřeba ještě zkontrolovat regresi, čím vzdálenější bude od 1, tím větší nepřesnosti bude rozpoznávač prokazovat. Při nedostačující regresi stačí proces trénování zopakovat, dokud nedostaneme uspokojivé výsledky.

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by NFTOOL
% Created Sat May 05 20:35:56 CEST 2012
%
% This script assumes these variables are defined:
%
%   vstup - input data.
%   vysledky - target data.

inputs = vstup;
targets = vysledky;

% Create a Fitting Network
hiddenLayerSize = 20;
net = fitnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstan-
trows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstan-
trows','mapminmax'};


% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndi-
vide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% For help on training function 'trainlm' type: help trainlm
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainlm';  % Levenberg-Marquardt

% Choose a Performance Function
% For a list of all performance functions type: help nnperfor-
mance
net.performFcn = 'mse';  % Mean squared error
```

```
 % Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist',
...
  'plotregression', 'plotfit'};


% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
testPerformance = perform(net,testTargets,outputs)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotfit(net,inputs,targets)
%figure, plotregression(targets,outputs)
%figure, ploterrhist(errors)
```

Na přiloženém CD je vložen i samotný program pro rozpoznávání a seznam použitých neznámých vzorků, které byly zhodnoceny v Tab. 1. Vše je vytvořeno na verzi Matlabu R2010b. Tyto oba soubory *rozpoznavac* a *neznname_vzorky* soubor stačí vložit do Workspace. Pro vyhodnocování neznámého vzorku, poté stačí do příkazového řádku napsat například *x=sim(net,n1_3)*, kde *n1_3* je název jednoho neznámého vzorku. Po tomto zadání se nám vyhodnocená hodnota přiřadí pod jménem *x*. Samotné zaokrouhlení na nejbližší celé číslo se provede příkazem *x=round(x)*. Jednotlivé neznámé vzorky jsou pojmenovány tak, že první číslo za písmenem *n* znázorňuje skutečné číslo slova. Za ležatou čárkou poté následuje jen sériové označení daného čísla.