

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2014

Jakub Hlavica

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského
inženýrství

Modely stanovení vývoje
Parkinsonovy choroby z řečového
signálu

Models of Assessment of Parkinson's
Disease Progression Using Speech
Signal

2014

Jakub Hlavica

Zadání diplomové práce

Student: **Bc. Jakub Hlavica**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2601T004 Měřicí a řídicí technika

Téma: **Modely stanovení vývoje Parkinsonovy choroby z řečového signálu**
Models of Assessment of Parkinson's Disease Progression using Speech Signal

Zásady pro vypracování:

1. Metody měření a filtrace řečového signálu.
2. Popis možností analýzy řečového signálu pomocí moderních výpočetních metod.
3. Parkinsonova choroba a její vliv na řečový signál.
4. Sestavení modelu stanovení vývoje Parkinsonovy choroby.
5. Testování modelu na poskytnutých datech "Parkinsons Telemonitoring Data Set".
6. Zhodnocení funkčnosti modelu a jeho aplikace.
7. Závěr a diskuze dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] BRONZINO, Joseph D. *The biomedical engineering handbook*. Boca Raton: IEEE Press, c1995. xxxii, 2862 p. ISBN 0-8493-8346-3.
- [2] ADAMEC, Ondřej. *Návrh a realizace biozesilovače s digitalizací a signálovým předzpracováním*. Ostrava, 2008. Diplomová práce. VŠB - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, Katedra měřicí a řídicí techniky.
- [3] *Medical instrumentation: application and design*. 3rd ed. Editor John G. WEBSTER. New York: John Wiley, 1998. 691 s. ISBN 0-471-15368-0.
- [4] TSANAS, Athanasios, et. al. Accurate telemonitoring of Parkinson's disease progression using non-invasive speech tests. *IEEE Trans. Biomed. Eng.*, Apr. 2012, vol. 57, no. 4, pp. 884–893. ISSN 0018-9294. DOI: 10.1109/TBME.2009.2036000. Dostupné také z: <http://people.maths.ox.ac.uk/tsanas/Preprints/TBME2010.pdf>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Michal Prauzek, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.



V Ostravě, dne 25. 4. 2014

Jakub Hlavica

Poděkování

Chtěl bych tímto poděkovat vedoucímu mé diplomové práce Ing. Michalovi Prauzkovi, Ph.D. za trpělivý a ochotný přístup během konzultací týkajících se vypracování této práce a za celkovou spolupráci při řešení. Dále bych chtěl poděkovat Ing. Tomáši Peterkovi za poskytnutou inspiraci při řešení komplexních problémů spojených s návrhem a vývojem software pro stanovení vývoje Parkinsonovy choroby.

Abstrakt

Cílem této diplomové práce je vyvinout výpočetní model, který je schopen efektivně mapovat šestnáct vstupních příznaků dat měření kolísání řečového signálu na požadovaný výstup. Vzorky řečového signálu pochází z měření ve spolupráci s pacienty postiženými Parkinsonovou chorobou. Jedná se o 5875 vzorků měření řečového signálu. Výstup ze systému stanovuje míru závažnosti tímto onemocněním, vyjádřenou prostřednictvím klinické diagnostické stupnice UPDRS. Navržený systém musí být schopen generalizace tak, aby v případě použití budoucích měření správně určil hodnotu UPDRS. Metodami použitými v této práci jsou neuronové sítě, především modifikace algoritmu zpětného šíření chyby Error Backpropagation, a dále metoda Adaptive Neuro Fuzzy Interference System (ANFIS). Výpočty jsou v maximální možné míře paralelizovány tak, aby byl realizován co nejvyšší počet simulací, a tím bylo nalezeno optimální řešení. V případě úspěchu by bylo možné vyvinout vestavěný systém, který by umožňoval průběžně diagnostikovat vývoj Parkinsonovy choroby z domova, čímž by byly pacientům sníženy náklady spojené s cestováním do lékařských zařízení.

Klíčová slova: Parkinsonova choroba, řečový signál, neuronové sítě, Error Backpropagation, fuzzy logika, ANFIS, UPDRS.

Abstract

This diploma thesis deals with software model development, which is capable of mapping sixteen input elements of speech signal oscillation measurements to clinical outputs. Samples of speech signal were obtained by measurements in collaboration with patients suffering from Parkinson's disease. There are provided 5875 measurement samples. The system's output determines severity of Parkinson's disease progression, qualified by clinical diagnostic rating scale UPDRS. Designed system must be able to generalize in order to correctly assess UPDRS scale value from future speech signal measurements. Utilized methods in this thesis are artificial neural networks, particularly modifications of Error Backpropagation algorithm, along with Adaptive Neuro Fuzzy Interference System (ANFIS). Calculations are parallelized as much as possible, so that many model simulations are performed in order to find the optimal solution. In case of success, it would be possible to develop an embedded system, which could continuously diagnose Parkinson's disease progression from home. That would reduce patient's travelling expenses to medical centres.

Key words: Parkinson's disease, Speech Signal, Neural Networks, Error Backpropagation, fuzzy logic, ANFIS, UPDRS.

Seznam použitých symbolů a zkratek

AHTD	At-Home Testing Device
ANFIS	Adaptive Neuro Fuzzy Inference System
BPG	algoritmus zpětného šíření chyby Error Backpropagation
CGB	algoritmus typu Conjugate Gradient Backpropagation
DCS	MATLAB Distributed Computing Server™
FFN	neuronová síť s dopředným šířením Feed-Forward Network
FLT	MATLAB Fuzzy Logic Toolbox™
GUI	grafické uživatelské rozhraní
ID	identifikační číslo
LOGSIG	aktivační funkce logaritmická sigmoida
MAE	střední absolutní chyba
Motor UPDRS	mapování na motorické hodnoty v rámci stupnice UPDRS
MSE	střední kvadratická chyba
NNT	MATLAB Neural Network Toolbox™
PCT	MATLAB Parallel Computing Toolbox™
PS	parkinsonský syndrom
RADBAS	aktivační funkce radiální báze
TANSIG	aktivační funkce hyperbolická tangenta
Total UPDRS	mapování na celkové hodnoty v rámci stupnice UPDRS
UNS	umělé neuronové síť
UPDRS	Unified Parkinson's Disease Rating Scale

Obsah

1. Úvod.....	1
2. Měření a zpracování řečového signálu.....	3
2.1 Analýza současného stavu.....	3
2.2 Zařízení AHTD	3
2.3 Měření řečového signálu	4
2.3.1 Výběr pacientů pro experiment	4
2.3.2 Klinické vyšetření pacientů.....	5
2.4 Zpracování a filtrace řečového signálu	5
2.5 Rešerše výsledků jiných vědeckých prací.....	6
3. Parkinsonova choroba a její vliv na řečový signál.....	7
3.1 Parkinsonova choroba a parkinsonský syndrom	7
3.1.1 Původ Parkinsonovy choroby	8
3.1.2 Symptomatická léčba – substituce dopaminu	8
3.1.3 Stupnice UPDRS.....	8
3.2 Projevy na řečovém signálu	10
4. Možnosti analýzy a zpracování řečového signálu moderními výpočetními metodami	11
4.1 Zpracování řečového signálu neuronovými sítěmi	11
4.1.1 Popis neuronových sítí	12
4.1.2 Základní prvky neuronu	12
4.1.3 Algoritmus zpětného šíření chyby Backpropagation	13
4.1.4 Alternativy a modifikace algoritmu BPG.....	15
4.2 Zpracování řečového signálu prostřednictvím metody ANFIS.....	18
4.2.1 Základy fuzzy logiky.....	18
4.2.2 Struktura modelu ANFIS	19
4.2.3 Algoritmus pro adaptaci parametrů modelu ANFIS	21
5. Sestavení modelu stanovení vývoje Parkinsonovy choroby	22
5.1 Statistická analýza a rozdělení poskytnutých dat.....	22
5.1.1 Statistická analýza vstupních signálů.....	22
5.1.2 Statistická analýza požadovaných výstupů	23
5.1.3 Rozdělení vstupních dat pro neuronové sítě	23
5.1.4 Rozdělení vstupních dat pro metodu ANFIS	25
5.2 Paralelizace iterativních výpočtů	25
5.2.1 Charakteristika Parallel Computing Toolboxu™.....	26
5.2.2 Clusterové profily pro paralelní výpočty	27

5.3	Neuronové sítě v prostředí MATLAB Neural Network Toolbox™	28
5.3.1	Režimy trénování neuronových sítí	28
5.3.2	Neuronové sítě s dopředným šířením signálu	29
5.3.3	Kaskádové neuronové sítě	30
5.3.4	Funkce thes_neural.m	30
5.3.5	Iterativní ladění neuronových sítí	32
5.3.6	Aktivační funkce neuronů	33
5.4	Neuronové sítě v programu Rapidminer 5	34
5.4.1	Konverze vstupních dat	34
5.4.2	Vytvoření modelu neuronové sítě	34
5.4.3	Operátor Neural Net	36
5.5	Model ANFIS v prostředí MATLAB Fuzzy Logic Toolbox™	36
5.5.1	Charakteristika implementované metody ANFIS	36
5.5.2	Funkce thes_anfis.m	37
5.5.3	Základní typy funkcí příslušnosti	38
6.	Testování modelu na poskytnutých datech „Parkinsons Telemonitoring Data Set“	40
6.1	Testování modelů neuronových sítí	41
6.1.1	Mapování vstupů na hodnoty Motor UPDRS	41
6.1.2	Mapování vstupů na hodnoty Total UPDRS	43
6.2	Testování modelů ANFIS	45
6.2.1	Výběr příznaků vstupních dat pro simulace modelů ANFIS	45
6.2.2	Mapování vstupů na hodnoty Motor UPDRS	46
6.2.3	Mapování vstupů na hodnoty Total UPDRS	47
7.	Zhodnocení funkčnosti modelu a jeho aplikace	48
7.1	Specifika funkčnosti výpočetních modelů	48
7.2	Srovnání s rešeršními výsledky	49
7.3	Využitelnost v praktických aplikacích	50
7.4	Možnosti dalšího výzkumu	51
8.	Závěr	52
	Literatura a použité zdroje	53
	Seznam příloh	56

Seznam obrázků

Obr. 2.1 Zařízení Intel AHTD.....	4
Obr. 4.1 Model McCulloch-Pittsova neuronu.....	12
Obr. 4.2 Příklad funkcí příslušnosti	19
Obr. 4.3 Struktura modelu ANFIS se dvěma vstupy.....	20
Obr. 5.1 Histogramy hodnot stupnice UPDRS	23
Obr. 5.2 Rozdělení vstupních dat pro návrh neuronových sítí.....	25
Obr. 5.3 Příklad neuronové sítě s dopředným šířením signálu	29
Obr. 5.4 Příklad kaskádové neuronové sítě.....	30
Obr. 5.5 Algoritmus naprogramovaného modelu neuronové sítě	31
Obr. 5.6 Rapidminer 5 – úroveň Main Process.....	34
Obr. 5.7 Rapidminer 5 – úroveň Validation.....	35
Obr. 5.8 Rapidminer 5 – proces pro nezávislý test	35
Obr. 5.9 Algoritmus naprogramovaného modelu ANFIS	37
Obr. 5.10 Příklad průběhu zvonovité funkce (<i>gbellmf</i>).....	38
Obr. 5.11 Příklad průběhu dvojité Gaussovy funkce (<i>gauss2mf</i>)	39
Obr. I.1 Experimentální hodnocení míry závažnosti onemocnění Parkinson. chorobou	I
Obr. II.1 Rozsahy příznaků Jitter (vstupy).....	II
Obr. II.2 Rozsahy příznaků Shimmer (vstupy)	II
Obr. II.3 Rozsahy příznaků poměrů (vstupy).....	III
Obr. II.4 Celkový přehled rozsahů 16 příznaků vstupních dat.....	III
Obr. III.1 Histogramy požadovaných výstupů určených pro trénování, validaci a testování	IV
Obr. III.2 Histogramy požadovaných výstupů pro nezávislý test	IV
Obr. IV.1 Rozdělení vstupních dat pro metodu ANFIS	V
Obr. VI.1 Aktivační funkce logaritmická sigmoida a její derivace.....	VI
Obr. VI.2 Aktivační funkce hyperbolická tangenta a její derivace	VI
Obr. VI.3 Aktivační funkce radiální báze a její derivace.....	VII
Obr. X.1 Regresní analýza – neuronová síť s algoritmem Levenberg-Marquardt.....	XV
Obr. X.2 Hodnoty MAE – neuronová síť s algoritmem Levenberg-Marquardt	XV

Seznam tabulek

Tab. 2.1 Výsledky mapování hodnot UPDRS z jiných vědeckých prací.....	6
Tab. 2.2 Výsledky mapování 16 příznaků řečového signálu	6
Tab. 4.1 Fáze adaptace parametrů modelu ANFIS	21
Tab. 6.1 Statistická analýza výstupů neuronových sítí – Motor UPDRS	42
Tab. 6.2 Statistická analýza výstupů neuronových sítí – Total UPDRS	44
Tab. 6.3 Statistická analýza modelu ANFIS – Motor UPDRS	46
Tab. 6.4 Statistická analýza výstupu modelu ANFIS – Total UPDRS	47
Tab. VIII.1 CGB s Powell-Bealeovými restarty – Motor UPDRS	IX
Tab. VIII.2 CGB s Fletcher-Reevesovými updaty – Motor UPDRS	IX
Tab. VIII.3 CGB s Polak-Ribiéryho updaty – Motor UPDRS.....	IX
Tab. VIII.4 Scaled Conjugate Gradient – Motor UPDRS.....	IX
Tab. VIII.5 Resilient Backpropagation – Motor UPDRS	X
Tab. VIII.6 Adaptivní míra rychlosti učení – Motor UPDRS	X
Tab. VIII.7 Momentum – Motor UPDRS	X
Tab. VIII.8 Adaptivní míra rychlosti učení a momentum – Motor UPDRS	X
Tab. VIII.9 One-Step Secant – Motor UPDRS	XI
Tab. VIII.10 Levenberg-Marquardt – Motor UPDRS.....	XI
Tab. VIII.11 Kaskádová neuronová síť – Levenberg-Marquardt – Motor UPDRS.....	XI
Tab. IX.1 CGB s Powell-Bealeovými restarty – Total UPDRS.....	XII
Tab. IX.2 CGB s Fletcher-Reevesovými updaty – Total UPDRS	XII
Tab. IX.3 CGB s Polak-Ribiéryho updaty – Total UPDRS	XII
Tab. IX.4 Scaled Conjugate Gradient – Total UPDRS	XII
Tab. IX.5 Resilient Backpropagation – Total UPDRS.....	XIII
Tab. IX.6 Adaptivní míra rychlosti učení – Total UPDRS	XIII
Tab. IX.7 Momentum – Total UPDRS	XIII
Tab. IX.8 Adaptivní míra rychlosti učení a momentum – Total UPDRS	XIII
Tab. IX.9 One-Step Secant – Total UPDRS	XIV
Tab. IX.10 Levenberg-Marquardt – Total UPDRS.....	XIV
Tab. IX.11 Kaskádová neuronová síť – Levenberg-Marquardt – Total UPDRS.....	XIV

1. Kapitola

Úvod

Počítačové zpracování naměřených dat je v inženýrské praxi velmi významnou oblastí, pomocí které můžeme správně interpretovat fyzikální, chemické a biologické jevy. V oblasti medicíny je zpracování dat jednou z důležitých cest vedoucích k vývoji a inovacím moderních lékařských metod. Jedním z onemocnění, ke kterému prozatím šetrné kvalitní a časově nenáročné diagnostické metody neexistují, je Parkinsonova choroba.

Parkinsonova choroba je velmi závažné onemocnění centrálního nervového systému postihující přibližně 1% lidské populace. U lidí ve věku nad 60 let nemoc postihuje dokonce 1% lidí. Poprvé byla popsána až v roce 1817 a mnoho lékařů se ve svých publikacích kloní k názoru, že za rozšířením Parkinsonovy choroby stojí nástup průmyslové revoluce na začátku 19. století, kdy došlo ke skokovému zhoršení životního prostředí. Přestože se diagnostické medicínské metody v poslední době vyvíjí velmi rychle, důvod vzniku Parkinsonovy choroby prozatím není znám. Tuto chorobu je možné aplikací tzv. dopaminergní léčby v určité míře zpomalit. Včasná a efektivní diagnostika je však stále rozhodující pro zachování kvalitního života jí postižených lidí. [1]

Tato diplomová práce je věnována návrhu a realizaci softwarových modelů pro stanovení vývoje a závažnosti onemocněním Parkinsonovou chorobou. Pomocí statistických metod jsou analyzována poskytnutá data, a následně na míru sestaveny výpočetní modely, založené především na metodách umělé inteligence, konkrétně se jedná o modely umělých neuronových sítí a model ANFIS založený na fuzzy logice. Navržené modely jsou schopny mapovat 16 příznaků řečového signálu na výstup, jímž je hodnota na klinické stupnici UPDRS, která vyjadřuje míru závažnosti onemocněním Parkinsonovou chorobou. Kvalita modelů je hodnocena na základě statistické analýzy výstupů z nezávislého testu, kdy jsou modelům předkládána speciálně vyčleněná data, která v průběhu adaptace, resp. tréninku, nebyla použita.

Data, na základě kterých budou tyto modely navrženy a testovány, pochází z výzkumu amerických lékařských ústavů. Ve spolupráci těchto ústavů se společností Intel Corporation bylo vyvinuto zařízení At-Home Testing Device (AHTD) sloužící k realizaci série testů. Jedná se o testy řečové, motorické, testy svalového třesu a další. Toto zařízení mohou neléčení pacienti postižení Parkinsonovou chorobou používat doma, čímž je omezeno nákladné cestování do klinických zařízení. Měření řečového signálu pacienti podstupovali jednou týdně.

Kapitola druhá pojednává o metodách měření a zpracování řečového signálu, jehož parametry (příznaky) jsou posléze ve výpočetních modelech mapovány na výstupy. Je zde popsán průběh měření řečového signálu pacientů trpících Parkinsonovou chorobou a pre-processing řečového signálu. Součástí této kapitoly je dále popis zařízení AHTD společnosti Intel Corporation, pomocí kterého byl řečový signál měřen. Na konci kapitoly je obsažena rešerše publikací jiných autorů v oblasti zpracování řečového signálu pacientů trpících Parkinsonovou chorobou.

Třetí kapitola se zabývá Parkinsonovou chorobou, o původu jejího vzniku a typických příznacích, především pak o projevech na řečovém signálu. Je zde popsána stupnice UPDRS, podle níž se klinicky stanovuje míra onemocnění touto chorobou, a dále tzv. dopaminergní léčba zpomalující průběh Parkinsonovy choroby.

Obsahem čtvrté kapitoly je popis možností analýzy a zpracování řečového signálu. Stěžejní nástroje používané v této diplomové práci jsou umělé neuronové sítě (UNS) a zpracování dat metodou Adaptive Neuro Fuzzy Inference System (ANFIS). Je zde položen teoretický základ pro návrh těchto modelů. U neuronových sítí je detailně popsán algoritmus zpětného šíření chyby Error Backpropagation a jeho efektivnější modifikace, v rámci modelu ANFIS jsou uvedeny možnosti fuzzy logiky.

Stěžejní kapitolou této diplomové práce je kapitola pátá, ve které jsou statisticky analyzována data měření řečového signálu, představeny možnosti paralelizace výpočtů simulací modelů a sestaveny modely pro stanovení vývoje Parkinsonovy choroby v návaznosti na nástroje popsané v kapitole 4. Použitými softwarovými prostředky jsou prostředí MATLAB (především Neural Network Toolbox™, Fuzzy Logic Toolbox™, Parallel Computing Toolbox™), a dále software Rapidminer 5 určený pro data mining.

Šestá kapitola je věnována simulacím a testování jednotlivých modelů na poskytnutých datech „*Parkinson's Telemonitoring Data Set*“. Nejlepší výpočetní modely jsou srovnány a detailně popsány na základě výstupů z nezávislého testu.

V sedmé kapitole je zhodnocena funkčnost vytvořených výpočetních modelů, jejich výstupy jsou srovnány s výsledky publikovanými jinými autory, dále jsou nastíněna využití systému v praktických aplikacích a možnosti dalšího výzkumu.

V závěru této práce je celkové hodnocení jejího průběhu a jsou zdůrazněny nejdůležitější poznatky, ke kterým autor této práce dospěl.

2. Kapitola

Měření a zpracování řečového signálu

2.1 Analýza současného stavu

Pacienti trpící Parkinsonovou chorobou musí v současné době podstupovat časově náročná vyšetření, která provádí lékaři specializovaní na neurodegenerativní onemocnění, tedy například také na Alzheimerovu chorobu.

Hodnocení míry onemocněním Parkinsonovou chorobou lékaři určují subjektivně na základě série testů, přičemž tato míra je vyjadřována stupnicí UPDRS (detailní popis v podkapitole 3.1.3). Je zřejmé, že toto hodnocení je zatíženo určitou chybou, protože každý lékař hodnotí jednotlivé příznaky choroby trochu odlišně. [2]

Z těchto důvodů vznikla iniciativa vývoje nové diagnostické metody spočívající v měření řečového signálu. Jedním z příznaků Parkinsonovy choroby je totiž chvění hlasu, přičemž na základě hlasových oscilací je možné s určitou mírou přesnosti stanovit míru tohoto onemocnění. Toto měření je realizováno prostřednictvím speciálního zařízení, které pacient může používat doma. Nová diagnostická metoda samozřejmě nemůže zcela nahradit vyšetření prováděné lékařem, avšak vzdálené vyšetření umožňuje lékařům získávat data o průběhu nemoci mnohem častěji a mimo jiné tak předvídat náhlé zhoršení zdravotního stavu pacientů.

2.2 Zařízení AHTD

At-Home Testing Device (AHTD, obchodní název OMDM-Dexterity 1.0) je vestavěný systém pro neinvazivní měření příznaků raného stádia Parkinsonovy nemoci, který vyvinula společnost Intel Corporation ve spolupráci se sdružením Kinetics Foundation a šesti výzkumnými centry univerzit v USA. Jeho hlavním cílem je snížit pacientům náklady spojené s cestováním do lékařských center a s pracovní neschopností v průběhu vyšetření. Zařízení mimo jiné obsahuje LCD obrazovku (pro zobrazování instrukcí pro pacienty), vestavěný reproduktor, dále USB port, mikrofonní vstup, několik zařízení pro motorické testy rukou a ovládací tlačítka. [3] Společně se zařízením je pacientům zúčastněným v experimentu dodán také kvalitní mikrofon pro měření řečového signálu. [2]

Na následujícím obrázku je zobrazeno vestavěné zařízení AHTD společnosti Intel. Prozatím se jedná o prototyp, jehož komerční využití závisí na výsledcích experimentálního výzkumu.



Obr. 2.1 Zařízení Intel AHTD [3]

2.3 Měření řečového signálu

Měření progresu Parkinsonovy choroby probíhá jednou týdně a trvá přibližně 30 minut. V jeho průběhu jsou zaznamenávány výsledky motorických testů, testy svalového třesu, a především řečový test. Základní řečový test (tzv. *sustained phonations*) probíhá tak, že pacient musí vyslovit písmeno „A“ a držet jeho zněllost co nejdéle bez nádechu. Z míry kolísání naměřeného řečového signálu vyjádřené 16 parametry (jejich význam uveden v části Příloha VII) lze neinvazivně stanovit stupeň závažnosti Parkinsonovy choroby pomocí stupnice UPDRS. [3]

Kromě základního řečového testu je prováděn také řečový test spočívající v popisu fotografií zobrazovaných na LCD obrazovce zařízení AHTD (tzv. *running speech test*). [2]

Oba řečové testy jsou samplovány na 24kHz s 16-bitovým rozlišením, přičemž mikrofon připojený k zařízení AHTD je umístěn 5 cm od pacientových rtů. Řečový signál je měřen do doby, než jeho intenzita klesne pod stanovenou prahovou hodnotu (měření je ukončeno jednu sekundu po překročení prahu), anebo je ukončeno po 30 sekundách měření. [2] [4]

Zařízení AHTD řečový signál nahrává a průběžně ukládá přes USB port na paměťové zařízení. Jakmile je test skončen, šifrovaná data jsou z pacientova počítače odeslána přes Internet do centrálního databázového systému [4], viz Obr. I.1.

2.3.1 Výběr pacientů pro experiment

Podmínky účasti pacientů v experimentálním vyšetření jsou následující:

- rané stádium Parkinsonovy choroby,
- diagnóza byla stanovena v posledních 5 letech,

- pacienti vykazují příznaky klidového třesu, bradykinezie či svalové ztuhlosti,
- nepodstupují dopaminergní léčbu. [3]

Důležitou skutečností je, že tento způsob experimentálního vyšetřování mohou podstupovat pouze neléčení pacienti. Jakmile pacient začne z důvodu náhlého zhoršení zdravotního stavu podstupovat tzv. dopaminergní léčbu (viz podkapitola 3.1.2), je z testování vyřazen. [3]

Ve výsledku tato kritéria splnilo 42 pacientů (z toho 28 mužů), avšak před vyřazením z testování z důvodu zahájení dopaminergní léčby bylo účastníků experimentu více. [2]

2.3.2 Klinické vyšetření pacientů

Pacienti účastníci se experimentálního vyšetřování z domu prostřednictvím měření řečového signálu podstoupili také tři klinická vyšetření u lékaře. První bylo provedeno na začátku experimentu, druhé tři měsíce po zahájení experimentu a třetí pak šest měsíců po zahájení. Lékař v každém z těchto vyšetření klinicky stanovil míru závažnosti onemocnění Parkinsonovou chorobou na stupnici UPDRS.

2.4 Zpracování a filtrace řečového signálu

Data získaná měřením řečového signálu pacientů s Parkinsonovou chorobou jsou po ukončení řečových testů a odeslání na server (viz Obr. I.1) dále zpracována, přičemž extrakce příznaků řečového signálu je realizována dvěma způsoby.

Prvním způsobem je výpočet klasických (lineárních) příznaků řečového, potažmo akustického, signálu, a to prostřednictvím hodnot *Jitter*, *Shimmer*, *NHR* (Noise-to-Harmonics Ratio) a *HNR* (Harmonics-to-Noise Ratio). Hodnoty *Jitter* obecně vyjadřují rozptyl základní frekvence řečového signálu, hodnoty *Shimmer* pak vyjadřují rozptyl amplitud řečového signálu. Klasické lineární příznaky řečového signálu tedy v podstatě vyjadřují míru oscilace základní frekvence f_0 . Oscilace řečového signálu jsou projevem Parkinsonovy choroby. Průměrná základní frekvence řečového signálu je u mužů 120Hz a u žen 200Hz. [2] Tyto příznaky (celkem 13) lze z řečového signálu poměrně snadno extrahovat použitím SW nástroje Praat vyvinutého na Amsterodamské univerzitě. [5]

Druhým způsobem je výpočet tří nelineárních příznaků *RPDE*, *DFA* a *PPE*. Tyto příznaky poskytují informaci o hlasivkových vibracích z fyziologického hlediska, mimo jiné jsou jimi hodnoceny tyto parametry – míra turbulencí vzduchu v hlasivkovém ústrojí, chraptění hlasu, uzavírání hlasivkových svalů atp. [2] Celkový počet příznaků řečového signálu je tedy 16.

V rámci zpracování měřených dat (pre-processing) je použito několika filtračních algoritmů, například standardní dolní propust, inverzní filtrační metody GNE nebo výše uvedené PPE. [4]

2.5 Rešerše výsledků jiných vědeckých prací

V rámci experimentálního výzkumu týkajícího se mapování příznaků řečového signálu bylo publikováno několik prací, ve kterých autoři implementují různé výpočetní a statistické metody, jejichž kvalita je posléze hodnocena prostřednictvím statistických ukazatelů, především MAE (střední absolutní chyba), MSE (střední kvadratická chyba) a koeficient korelace mezi požadovanými výstupy a výstupy z výpočetních modelů. Ne vždy ovšem autoři použili všech 16 dostupných příznaků řečového signálu.

Co se týče výpočetních metod, autoři použili především metody *Classification and Regression Trees* (CART), *Iteratively Reweighted Least Squares* (IRLS), *Least Absolute Shrinkage and Selection Operator* (LASSO) a *Least Square Support Vector Machine* (LS-SVM).

V následující tabulce jsou obsaženy výsledky z vědeckých prací, které jsou věnovány mapování vstupních příznaků (měření řečového signálu) na hodnoty UPDRS. Hodnotami v tabulce jsou střední absolutní chyby (MAE).

Tab. 2.1 Výsledky mapování hodnot UPDRS z jiných vědeckých prací [2] [5] [6]

Autor	Výpočetní metoda	Počet použitých příznaků	Motor UPDRS (MAE)	Total UPDRS (MAE)
Tsanas et al. [2]	CART	6	5,95	7,52
Tsanas et al. [2]	IRLS	6	6,80	8,47
Tsanas et al. [5]	LASSO*	13	6,57	8,38
Eskidere et al. [6]	LS-SVM	6	5,53	6,99
Eskidere et al. [6]	LS-SVM*	13	5,61	7,17
Eskidere et al. [6]	LS-SVM*	16	4,87	6,18

* U těchto výpočetních metod byla navíc realizována logaritmická transformace uvedená v [5], která výsledky nepatrně zlepšuje.

Autoři používající všech 16 příznaků řečového signálu bez úpravy logaritmickou transformací využili kromě metody LS-SVM také modely neuronových sítí (MLPNN). [6]

Tab. 2.2 Výsledky mapování 16 příznaků řečového signálu [6]

Metoda	Motor UPDRS			Total UPDRS		
	MSE	MAE	Corr. Coeff.	MSE	MAE	Corr. Coeff.
LS-SVM	39,47	4,96	0,63	66,01	6,32	0,65
MLPNN	48,59	5,61	0,51	83,08	7,19	0,52
SVM	46,73	5,46	0,54	82,67	7,02	0,53

3. Kapitola

Parkinsonova choroba a její vliv na řečový signál

Parkinsonova nemoc je druhým nejrozšířenějším neurodegenerativním onemocněním na světě. Touto nemocí trpí přibližně 1‰ lidí na světě (po Alzheimerově chorobě). Sleduje-li se prevalence Parkinsonovy nemoci pouze u osob starších 60 let, postihuje jich až cca 1%. [1] Neurodegenerativní onemocnění obecně způsobují poruchy centrálního nervového systému v lidském těle.

Vzhledem k tomu, že především v rozvinutých zemích věk populace převážně stoupá, lze očekávat, že výskyt Parkinsonovy nemoci se bude v průběhu následujících let zvyšovat, takže hledání způsobu její včasné diagnostiky a následné kvalitní léčby je na místě. V úvodu této kapitoly je vhodné předznamenat, že příčiny vzniku Parkinsonovy nemoci prozatím nejsou známy.

3.1 Parkinsonova choroba a parkinsonský syndrom

Parkinsonský syndrom (PS) je porucha hybnosti charakterizovaná třesem, rigiditou, hypokinezi (bradykinezi) a posturálními abnormalitami (poruchami stoje a chůze). [1]

Uvedené příznaky jsou definovány následovně:

- **Třes** – jedná se o rytmický sinusoidální oscilační pohyb části těla působený střídavými stahy svalů. PS je provázen především klidovým třesem (frekvence přibližně 5 Hz).
- **Rigidita** – svalová ztuhlost kladoucí odpor aktivnímu i pasivnímu pohybu v celém jeho rozsahu
- **Hypokineze** – snížení rozsahu (amplitudy) pohybů, bradykineze je charakterizována jako zpomalený průběh pohybů
- **Poruchy stoje a chůze** – ohnuté držení trupu a končetin, šouravá chůze krátkými krůčky – jedná se o nejcharakterističtější příznaky Parkinsonovy nemoci [1].

Parkinsonova nemoc je nejčastější příčinou PS, který je v tomto případě podmíněn nedostatečnou tvorbou dopaminu (DA) v části mozku zvané *substantiae nigrae*. Dopamin je chemická látka, která funguje jako tzv. neurotransmitter (neboli přenašeč) a realizuje přenos elektrických impulsů v mozku mezi nervovými buňkami. [7] První klinické symptomy se začínají projevovat v momentě, kdy množství dopaminu klesne pod 20% původního množství. [1]

PS může být způsoben řadou dalších příčin (jiná degenerativní onemocnění), zpravidla však v těchto případech bývá vyjádřena další symptomatika, na které je tzv. dopaminergní léčba neúčinná. [1]

3.1.1 Původ Parkinsonovy choroby

Důvod nedostatečné produkce neurotransmiteru dopaminu není bohužel v současné době znám, nicméně v průběhu zkoumání Parkinsonovy nemoci byly vytvořeny čtyři hypotézy původu jejího vzniku. Nemoc je tedy způsobena:

- endotoxiny – nedostatek detoxikačních mechanismů v mozku vede k nekróze buněk části mozku *substantiae nigrae*,
- exotoxiny – exotoxiny se dostanou do mozku přes fyziologické detoxikační mechanismy a naruší činnost zdravých mitochondrií (tzv. „buněčné elektrárny“)
- genetickou poruchou – dědičná choroba v rodině (asi 5% pacientů udává, že alespoň jeden příbuzný byl obdobně postižen).
- apoptózou – v důsledku tzv. programované buněčné smrti. [1]

Samozřejmě je nutné vzít v úvahu variantu, kdy Parkinsonova nemoc je způsobena kombinací výše uvedených faktorů.

3.1.2 Symptomatická léčba – substituce dopaminu

Základní možností léčby je podávání L-3,4-dihydroxyfenylalaninu (L-DOPA), ze kterého enzymatickou přeměnou v mozku vzniká chybějící dopamin. Existuje však také možnost náhrady dopaminu bez účasti presynaptických metabolických pochodů, a to podáváním agonistů dopaminu, kteří účinkují přímo na příslušné receptory. [1]

Podávání látky L-DOPA však není bez problémů. Zatímco časným vedlejším účinkům léčby lze zabránit nebo je odstranit, pozdní komplikace Parkinsonovy nemoci vznikající v souvislosti s léčbou L-DOPA jsou mnohdy obtížně ovlivnitelné. Časnými vedlejšími účinky je například kolísání krevního tlaku, srdeční arytmie, nauzea, vomitus atp. Tyto vedlejší účinky se nejčastěji projevují na počátku léčby. Parkinsonovu léčbu nelze vyléčit, je však možno dlouhodobě a účinně potlačovat její příznaky. Dostupné metody léčby však nepotlačují primární patologický proces. Chronická medikace bývá spojena s řadou pozdních komplikací. Léčebné postupy by proto měly být koncipovány tak, aby se nástup pozdních komplikací co nejvíce oddálil. [1]

3.1.3 Stupnice UPDRS

Unified Parkinson's Disease Rating Scale (UPDRS) je v současnosti v klinické praxi nejčastěji využívanou stupnicí hodnocení závažnosti onemocněním Parkinsonovou chorobou. Byla vytvořena v roce 1987 kombinací několika již používaných stupnic závažnosti tohoto onemocnění, čímž vznikl výkonný, flexibilní a srozumitelný nástroj k hodnocení. [8]

Původní UPDRS sestává z šesti částí hodnocení závažnosti onemocnění. V rámci dat měření pacientů, která jsou zpracovávána v této práci, jsou ovšem využívány pouze první tři části stupnice UPDRS.

Jedná se o následující:

- I. Mentální aktivita, chování a nálada
- II. Aktivity každodenního života
- III. Motorika [2]

V části I vyhodnocována určitými klinickými, popřípadě psychiatrickými metodami míra poruchy myšlení, motivace a deprese. Ve II. části se vyhodnocují potíže nastávající s progresí Parkinsonovy nemoci v každodenním životě, tedy například slinění, polykání, psaní rukou, sensorické obtíže apod. Poslední III. část je z hlediska diagnostiky závažnosti choroby nejvýznamnější. V části motorika je měřena frekvence třesu, popřípadě jeho amplituda, vyhodnocovány jsou řečové potíže, postoj a chůze, svalová ztuhlost, rychlost běžných pohybů, jemná manipulace s předměty pomocí rukou apod. [8]

Každá z uvedených kategorií (například problémy s polykáním) je bodově hodnocena v rozsahu 0 – 4, přičemž bodové hodnocení 0 indikuje normální (správnou) funkci a hodnocení 4 představuje závažné potíže. [9]

Celkové hodnocení míry závažnosti onemocnění Parkinsonovou chorobou na stupnici UPDRS (v této práci označováno Total UPDRS) sestavené z tří výše uvedených částí (I, II a III) nabývá hodnot v rozsahu 0 – 176. Samotná III. část – motorika (označována Motor UPDRS) – je v rámci celkového UPDRS hodnocena v rozsahu 0 – 108, maximální hodnota v intervalu reprezentuje totální nepohyblivost. Co se týče kategorie řečových potíží, v rámci části motorika je hodnocena body v intervalu 0 – 8, přičemž hodnota 8 vyjadřuje absolutní nesrozumitelnost. [2]

Je nutné si uvědomit, že uvedená bodová hodnocení jsou z větší míry stanovována poměrně subjektivně konkrétním lékařem, takže míra závažnosti onemocnění téhož pacienta by se mohla při diagnostice několika různými lékaři mírně lišit. Tedy i uvedená kategorie řečových potíží je hodnocena subjektivně. Nicméně při zpracování řečového signálu se používá exaktních metod, které umožňují míru závažnosti choroby mapovat velmi přesně, jak bude vysvětleno v následujících kapitolách.

3.2 Projevy na řečovém signálu

V lehčích případech onemocnění je často patrné omezení řečových funkcí, které může zčásti spočívat v poruše převážně neuromechanických dějů artikulace. Ve velmi pokročilých případech Parkinsonovy choroby pak může dojít k úplné ztrátě řeči. [1]

Řeč se postupně stává tichá, monotónní, není provázena mimikou obličeje. Polykání je pomalejší a obtížnější. To může přispívat k tomu, že se hromadí sliny v ústech. [10] Ztuhlost v obličeji je obecně častý a typický příznak. Postihuje především kruhový sval ústní. Postiženy bývají i další skupiny neméně důležitých svalů podílejících se na řeči. Jsou to svaly jazyka, čelisti, měkkého patra a hltanu. [7]

Mezi další časté příznaky se řadí dysprosodie, tj. snížení schopnosti modulovat hlasitost a melodii řeči, doprovázená akcelerací tempa řeči (tachyfémie) a nutkavým opakováním slabik, slov či celých vět (palilalie). Příčinou těchto příznaků je kromě uvedené svalové ztuhlosti také zpomalená hybnost dýchacího a artikulačního svalstva (tedy projevy parkinsonské rigidity a bradykineze), ale zřejmě též porucha řečového plánu (tedy řídicích mechanismů řeči). [1]

S uvedenými dýchacími potížemi souvisí také chraptění, zadýchávání a sípavý hlas. Tyto hlasové projevy jsou obvykle předznamenáním nastupujícího závažnějšího onemocnění. [2]

Právě chraptění a zadýchávání je při zpracování řečového signálu největším problémem, protože vnáší do řečového signálu šum, který je nutné filtrovat.

4. Kapitola

Možnosti analýzy a zpracování řečového signálu moderními výpočetními metodami

V této diplomové práci jsou pro analýzu a zpracování řečového signálu použity dva nástroje. Jsou jimi umělé neuronové sítě a metoda Adaptive Neuro Fuzzy Inference System (ANFIS). Metoda ANFIS je neuronovým sítí podobná co do adaptace parametrů během trénování systému, nicméně její princip spočívá v použití fuzzy logiky. V následujících podkapitolách jsou oba systémy a jejich specifika popsány v teoretické rovině. Softwarová implementace obou nástrojů je pak součástí kapitoly 5.

4.1 Zpracování řečového signálu neuronovými sítěmi

Umělé neuronové sítě (UNS) jsou zjednodušeným matematickým modelem biologických neuronových sítí, které jsou součástí centrální nervové soustavy, tedy především součástí mozku. UNS jsou používány především v případech analýzy nelineárních dat a podobných komplexních problémů, kde selhávají klasické analytické metody. Analýza dat obsahujících nelinearity je v současnosti nezbytná v mnoha aplikacích, především při zpracování řečového signálu, u strojové syntézy hlasu, rozpoznávání písma a gest, v biomedicinských aplikacích, v telekomunikacích, v automobilovém průmyslu a obecně v mnoha dalších případech, kde se v určité míře významnosti vyskytuje rozhraní člověk-stroj (*Human-Machine Interface*) a systém se tak musí adaptovat na konkrétní osobu.

Při implementaci umělých neuronových sítí ve výpočetních systémech s výhodou využíváme dvou jejich charakteristických vlastností:

- Získávání znalostí učením pomocí množiny vzorů (tj. známých hodnot vstupních a výstupních parametrů) bez nutnosti získat znalosti explicitně.
- Schopnost generalizace, kdy do správných tříd jsou klasifikovány i vstupní vzory, které nejsou součástí tréninkové nebo učící množiny. Platí, že jestliže jsou v tréninkové množině rovnoměrně obsaženy všechny významné vzory a tato tréninková množina obsahuje 10% všech možných vzorů, pak UNS může klasifikovat správně i zbylých 90% vzorů. [11]

Je ovšem nutné si uvědomit, že tato uvedená generalizace na nová data je možná pouze po správném naprogramování neuronové sítě. Problémem však je skutečnost, že neexistuje žádný

obecný postup, jak nastavit tréninkové parametry nebo topologii a vzájemné vazby v UNS a je tedy nezbytné u každé aplikace postupovat individuálně.

4.1.1 Popis neuronových sítí

Neuronové sítě sestávají z tzv. neuronů, které si lze představit jako paralelně pracující výpočetní procesory, které mají paměť, tj. dokáží si pamatovat předchozí stavy (zkušenosti). Podle jejich uspořádání (topologie) je dělíme do několika kategorií, přičemž nejpoužívanějšími jsou tzv. vícevrstvé sítě s dopředným šířením informace (*feedforward neural networks*), jejichž princip je využíván i v této diplomové práci. Učení, tedy adaptace jednotlivých komponent sítě na předkládaná data, je realizováno formou tzv. učení s učitelem (*supervised learning*), kdy je síť s vektorem vstupních dat předkládán současně také požadovaný výstup.

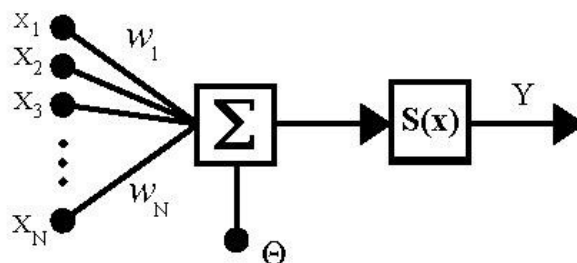
Neuronové sítě využívají distribuované, paralelní zpracování informace při provádění výpočtů. [12] Při učení (trénování) neuronové sítě jsou vstupní data rozdělena do třech částí – tréninková množina, validační množina a testovací množina. [13] Obvyklý procentuální poměr rozdělených množin je 70/15/15, 70/10/20, 60/20/20, 80/10/10 a podobně.

4.1.2 Základní prvky neuronu

Neuron jako matematický procesor je charakterizován svými vstupy, tj. vektorem vstupního signálu $x_1 \dots x_n$, dále svým výstupem, jímž je skalární hodnota y_j (výstup j-tého neuronu), vektorem ω synaptických vah ω_{ji} , hodnotou prahu θ_j a typem aktivační funkce $S(\varphi)$. Takto definovaný neuron je nazýván McCulloch-Pittsov neuron. [14]

Kombinace vstupního vektoru a $x(t)$ a vektoru synaptických vah $\omega(t)$ se přiřazuje jednotlivým složkám vektoru $x(t)$ jako určitá váha odpovídající uložení zkušeností. Prahem θ_j je myšlena minimální hodnota vážené sumy vstupů, aby byl neuron tzv. aktivní. [11]

Aktivační funkce je v podstatě přenosovou funkcí, která stanovuje způsob výpočtu výstupního skaláru z vektoru vstupních dat. Obecně se rozlišují aktivační funkce spojité (nejčastěji využívány jsou hyperbolická tangenta a logaritmická sigmoida, popřípadě lineární funkce) a nespojitě (skoková funkce, Heavisideova funkce).



Obr. 4.1 Model McCulloch-Pittsova neuronu [15]

Znalosti neuronové sítě jsou ukládány především prostřednictvím síly vazeb mezi jednotlivými neurony. Vazby vedoucí ke „správné odpovědi“ jsou posilovány a naopak, vazby vedoucí k „špatné odpovědi“ jsou oslabovány pomocí opakované expozice příkladů popisujících problémový prostor. [12]

4.1.3 Algoritmus zpětného šíření chyby Backpropagation

Algoritmu zpětného šíření chyby, anglicky Error Backward Propagation (standardní zkratka BPG) je pro svou relativní jednoduchost a flexibilitu jedním z nejpoužívanějších algoritmů pro učení neuronových sítí. Pro nastavování vah na vstupech neuronů využívá gradientní metodu, z čehož vyplývá, že aktivační funkce neuronu musí být diferencovatelná, tedy například logaritmická sigmoida (*logsig*). [16]

BPG je založen na minimalizaci energie neuronové sítě. Energie je zde myšlena jako míra naučenosti neuronové sítě, tedy odchylka mezi skutečnou hodnotou a hodnotou získanou z výstupu sítě pro danou tréninkovou množinu. [14] Tato energetická funkce se v literatuře též nazývá funkcí chybovou.

Princip BPG spočívá v tom, že výstup z neuronové sítě je porovnán s požadovaným výstupem, je spočítána odchylka a v závislosti na její velikosti je modifikován vektor synaptických vah předchozích vrstev, čímž dochází ke zpětnému šíření. Na modifikovanou neuronovou síť je opět přiveden vektor vstupních dat a dochází ke stejné proceduře. Proces je iterativně opakován do doby, než je energetická funkce počítaná ze vzorku validační množiny menší, než určité stanovené kritérium, popřípadě do doby, než chybová funkce začne opět růst, což značí, že bylo dosaženo minima. [13]

Algoritmus BPG lze shrnout v následujících šesti krocích: [14]

- **Krok 1. Inicializace**

Všechny váhy v síti nastavíme náhodně na hodnoty v doporučeném rozsahu $(-0.3, 0.3)$.

- **Krok 2. Předložení vzoru**

Vybereme vzor z tréninkové množiny a přiložíme na vstupy sítě. Dále po vrstvách směrem od vstupu k výstupům sítě počítáme výstupy jednotlivých neuronů dle vztahů:

$$y = S\left(\sum_{i=1}^n \omega_i x_i + \theta\right), \quad S(\varphi) = \frac{1}{1 + e^{-\gamma\varphi}} \quad (4.1)$$

kde ω_i je synaptická váha i -tého vstupu neuronu, x_i je i -tý prvek vstupního vektoru. Funkce $S(\varphi)$ je aktivační funkce sigmoida a γ je strmost aktivační funkce.

- **Krok 3. Srovnání**

V kroku srovnání nejprve vypočítáme energii (chybu) podle vztahu

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - d_i)^2 \quad (4.2)$$

kteřá bude využita jako přírůstek k celkové energii počítané přes všechny vzory. Proměnná y_i je i -tý výstup neuronové sítě, d_i je i -tý požadovaný výstup neuronové sítě. Dále vypočítáme chybu pro výstupní vrstvu:

$$\delta_i^o = (d_i - y_i^o) \gamma y_i^o (1 - y_i^o) \quad (4.3)$$

- **Krok 4. Zpětné šíření chyby**

Pro všechny neurony ve vrstvě vypočteme:

$$\omega_{ij}^l(t) = \alpha \delta_i^l(t) y_j^{l-1}(t) + \mu \Delta \omega_{ij}^l(t-1) \quad (4.4)$$

$$\theta_i^l(t) = \alpha \delta_i^l(t) + \mu \Delta \theta_i^l(t-1) \quad (4.5)$$

kde parametr α je tzv. *learning rate* a μ je momentum (setrvačnost). Podle vztahu

$$\delta_i^{h-1} = y_i^{h-1} (1 - y_i^{h-1}) \sum_{k=1}^n \omega_{ki}^h \delta_k^h \quad (4.6)$$

zpětně šíříme chybu do vrstvy, která je blíže vstupům. Na závěr kroku 4 modifikujeme váhy:

$$\omega_{ij}^l(t+1) = \omega_{ij}^l(t) + \Delta \omega_{ij}^l(t) \quad (4.7)$$

$$\theta_i^l(t+1) = \theta_i^l(t) + \Delta \theta_i^l(t) \quad (4.8)$$

Krok 4 opakujeme pro všechny sítě tak, že začínáme vrstvou výstupní ($l = o$), pak následují skryté vrstvy ($l = h$). V případě, kdy zpracováváme skrytou vrstvu, která je nejbližší vstupní vrstvě, nahradíme y_j^{l-1} v rovnici pro výpočet $\Delta \omega$ za odpovídající vstupní hodnotu, tedy x_j .

- **Krok 5. Ukončení výběru vzorů z tréninkové množiny**

Jestliže jsme předložili síti všechny vzory z tréninkové množiny, pokračujeme krokem 6, jinak se vracíme na krok 2.

Pozn.: při sekvenčním výběru je realizace tohoto testu jednoduchá, avšak vybíráme-li vzory náhodně, musíme v průběhu učení poznamenávat, které vzory byly použity a které nikoliv.

- **Krok 6. Ukončení proces učení**

Jestliže energie (chyba) neuronové sítě za poslední epochu (nebo stanovený určitý počet epoch) byla menší než námi zvolené kritérium, pak ukončíme učení. Jinak pokračujeme krokem 2.

Při použití algoritmu BPG lze v podstatě modifikovat pouze jeden parametr, a tím je míra rychlosti učení (*learning rate*), označovaná α . Tento parametr definuje velikost kroku, tedy v podstatě ovlivňuje míru maximální změny nastavení synaptických vah mezi neurony. Problematika nastavení tohoto parametru spočívá v tom, že pokud je míra rychlosti učení příliš vysoká, systém má tendenci oscilovat a stát se nestabilním. Pokud je naopak nízká, systém velmi pomalu konverguje. [13]

4.1.4 Alternativy a modifikace algoritmu BPG

Samotný základní algoritmus Backpropagation (BPG) je ve své podstatě náchylný při minimalizaci chybové funkce k tzv. uvíznutí v lokálním minimu. To znamená, že proces trénování neuronové sítě se zastaví v momentě, jakmile začne chybová funkce validační množiny opět růst. Toto minimum chybové funkce ovšem může být pouze lokální. Kdyby hypoteticky trénování sítě pokračovalo, je možné, že by chybová funkce nabyla ještě menší hodnoty, což by značilo nalezení globálního minima. [13] Navíc metoda BPG konverguje pomalu.

Z těchto důvodů byly vyvinuty efektivnější algoritmy, které z algoritmu BPG vycházejí, avšak vztahy pro výpočet chybové funkce a její zpětné šíření se liší v závislosti na použité metodě.

Algoritmus BPG je v těchto algoritmech používán pouze k výpočtu gradientu, tedy parciálních derivací chybové funkce (ve vztahu k parametrům neuronové sítě, tj. k synaptickým vahám a prahům neuronů). To, v čem se tyto modifikované algoritmy liší, je způsob výpočtu změn těchto nastavitelných parametrů neuronové sítě. [13]

Heuristické optimalizační metody

Algoritmy Gradient Descent

Tyto tréninkové algoritmy spočívají v zavedení adaptivních parametrů, které jsou měněny v závislosti na aktuální hodnotě gradientu chybové funkce. Těmito parametry jsou míra rychlosti učení a *momentum*, což je parametr představující setrvačnost systému.

V prostředí MATLAB jsou implementovány tři algoritmy typu gradient Descent:

- **Adaptivní míra rychlosti učení** (příkaz *traingda*) – algoritmus je schopen na základě aktuální hodnoty gradientu chybové funkce měnit míru rychlosti učení, přičemž na rovné ploše gradientu chybové funkce míra rychlosti učení zvyšována (o určitý faktor, standardně 1,05), čímž dochází k rychlejšímu překonání těchto rovných ploch gradientu (což by při použití statické míře rychlosti učení trvalo velmi dlouho). Na strmých částech povrchu gradientní funkce je míra rychlosti učení snižována (standardně o hodnotu 0,7), aby nedocházelo k oscilaci chybové funkce. Pokud je velikost chyby

MSE výrazně vyšší (standardně o více než 4%), než v předchozí epoše, je aktuální krok zrušen a opakován s jinou hodnotou míry rychlosti učení. [11] [13]

- **Momentum** (příkaz *traingdm*) – parametr *momentum* představuje v systému setrvačnost, pomocí níž je neuronová síť schopná minimalizovat chybovou funkci, protože setrvačnost sleduje aktuální trend této funkce. Při dosažení lokálního minima je schopná pomocí setrvačnosti tento extrém překonat a pokračovat v klesajícím trendu chybové funkce. Hodnota parametru *momentum* se pohybuje v rozmezí $0 \div 1$. Nelze ovšem nastavit hodnotu 1, protože v takovém případě je neuronová síť necitlivá ke změnám gradientu chybové funkce a nelze ji tak natrénovat. Míra rychlosti učení je v tomto algoritmu statickým parametrem. Výhodou tohoto algoritmu je mimo jiné skutečnost, že konvergence se zrychluje, pokud se trajektorie chybové funkce nemění. [13]
- **Adaptivní míra rychlosti učení a momentum** (příkaz *traingdx*) – jedná se o kombinaci obou výše popsaných algoritmů. [11]

Algoritmus Resilient Backpropagation (příkaz *trainrp*)

Algoritmus je založený na myšlence změn synaptických vah v závislosti na znaménku parciálních derivací (gradientu) chybové funkce. [11] Velikost změny synaptické váhy je zvětšena o určitou velikost (defaultně 1,2), pokud je v následující iteraci znaménko parciální derivace stejné, naopak zmenšena (o hodnotu 0,5), pokud je znaménko opačné. Algoritmus vůbec nepracuje s amplitudou (velikostí) parciálních derivací [17], z čehož vyplývají malé nároky na paměť výpočetního systému, jelikož znaménka derivací lze ukládat do proměnných typu boolean (standardně jsou hodnoty gradientu ukládány v proměnných typu double). Z toho vyplývá velká rychlost konvergence při minimalizaci chybové funkce.

Numerické optimalizační metody

Tyto algoritmy v zásadě vycházejí z metody nejprudšího poklesu gradientu (*steepest descent*). Z důvodu existence lokálních minim povrchu chybové funkce ovšem tato metoda nevede vždy k nejrychlejší konvergenci.

Algoritmy Konjugovaného gradientu

Tyto algoritmy v prvním kroku používají metodu *steepest descent*, tedy směr poklesu chybové funkce je podél nejstrmějšího gradientu. V dalších krocích je vektor směru poklesu (metoda *line search*) stanovován prostřednictvím matematického výrazu obsahujícího parametr β_k , který je vypočítán z předchozí a aktuální hodnoty gradientu. Poté je lokalizováno minimum chybové funkce, přičemž intervaly hledání jsou v rámci uživatelem stanovených konstant postupně zmenšovány. Pokud dojde ke specifické předpokládané situaci (ortogonalita mezi prvky vektoru směru poklesu), dochází k tzv. restartu a proces se opakuje od první iterace.

Způsob, jakým je vypočítán parametr β_k je závislý na zvoleném tréninkovém algoritmu. Implementovány jsou čtyři algoritmy *Conjugate Gradient Backpropagation* (CGB) [13]:

- **CGB s Powell-Bealeovými restarty** (příkaz *traincgb*) – během trénování používá šest vektorů gradientů a směru poklesu,
- **CGB s Fletcher-Reevesovými updaty** (příkaz *traincgf*) – používá tři vektory,
- **CGB s Polak-Ribièreovými updaty** (příkaz *traincgp*) – používá čtyři vektory,
- **Algoritmus Scaled Conjugate Gradient** (příkaz *traincsg*) – od ostatních liší tím, že metoda *line search* není prováděna při každé iteraci algoritmu.

Z důvodu malého množství kalkulovaných parametrů jsou tyto algoritmy vhodné pro velké neuronové sítě s mnoha neurony, tudíž s mnoha synaptickými vahami a prahy.

Newtonův algoritmus

Základní Newtonova metoda je založena na výpočtu čtvercové Hessovy matice, která obsahuje druhé parciální derivace chybové funkce (ve vztahu k synaptickým vahám a prahům). Výpočet této matice je ovšem u větších neuronových sítí velmi náročný a konvergence takových sítí by byla extrémně pomalá. [13] Existují dva algoritmy, které proces konvergence zrychlují:

- **Kvazi-Newtonova BPG** (příkaz *trainbfg*) – v tomto algoritmu je Hessova matice v každé iteraci pouze aproximována (nejsou počítány druhé derivace). I přesto se však nehodí pro velmi velké neuronové sítě, a to z důvodu uložení aproximace velké Hessovy matice (velikost matice je $n \times n$, přičemž n je počet synaptických vah a prahů neuronové sítě). [13]
- **Metoda One-Step Secant** (příkaz *trainoss*) – tento algoritmus nepočítá celou Hessovu matici, ale předpokládá, že tato matice byla v předchozí iteraci jednotkovou maticí. [18]

Algoritmus Levenberg Marquardt (příkaz *trainlm*)

Tento algoritmus stejně jako Kvazi-Newtonův algoritmus obchází výpočet výpočetně náročné Hessovy matice tímto způsobem:

$$H = J^T J \quad (4.9)$$

kde J je tzv. Jacobiho matice, která obsahuje první parciální derivace chybové funkce (ve vztahu k synaptickým vahám a prahům). Problémem je skutečnost, že matice H nemusí být regulární (takže její determinant může být teoreticky nula). Z toho důvodu se zavádí tzv. Marquardtův parametr μ . Výsledný vztah pak vypadá následovně:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (4.10)$$

kde J je Jacobiho matice, I je jednotková matice a e je vektor chyb neuronové sítě. Jakmile je Marquardtův parametr zvyšován (o předem stanovenou hodnotu), algoritmus se podobá metodě *steepest descent*, v opačném případě (snižování jeho hodnoty) se blíží Newtonově metodě, což je cílem tréninku, protože v takovém případě vykazuje nejrychlejší konvergenci. [13]

Tento tréninkový algoritmus je výhodný pro návrh malých až středně velkých neuronových sítí, protože vyžaduje hodně paměti. V případě velkých neuronových sítí probíhá výpočet velmi pomalu. [16]

Algoritmus Bayesian Regularization (příkaz *trainbr*)

Tento algoritmus je založený na algoritmu Levenberg-Marquardt, avšak zavádí parametr regularizace zohledňující hodnoty synaptických vah a prahů neuronové sítě. Chybovou funkcí není standardní MSE, ale funkce

$$MSE_{reg} = \gamma \cdot MSE + (1 - \gamma) \cdot ms\omega \quad (4.11)$$

kde γ je parametr regularizace a člen

$$ms\omega = \frac{1}{n} \sum_{j=1}^n \omega_j^2 \quad (4.12)$$

V této konfiguraci je neuronová síť schopná dosáhnout lepší generalizace pro nově předložená budoucí data. Hodnota parametru γ je během tréninku sítě nastavována automatizovaně. [16]

4.2 Zpracování řečového signálu prostřednictvím metody ANFIS

Adaptive Neuro Fuzzy Inference System (ANFIS) je druh výpočetního modelu kombinující vlastnosti neuronových sítí a vlastnosti fuzzy logických systémů. Založen je na fuzzy modelu typu Takagi-Sugeno.

Významnou výhodou fuzzy modelu Takagi-Sugeno je skutečnost, že se může jednat o jakkoliv nelineární n-rozměrný systém, který je rozložen na r-lineárních subsystémů. [19] Další výhodou je fakt, že zisk výstupů z tohoto modelu je výpočetně nenáročný, čímž dochází k výraznému zrychlení systému, především ve výkonově omezených výpočetních systémech, například ve vestavěných systémech.

4.2.1 Základy fuzzy logiky

Fuzzy logika je postavena na premise, že množiny nejsou ostře ohraničeny tak, jako v klasické výrokové logice, ve které prvek do množiny patří (log 0) anebo nepatří (log 1). Ve fuzzy logice je zaveden pojem částečné příslušnosti, která je v podstatě vyjádřením určité neurčitosti.

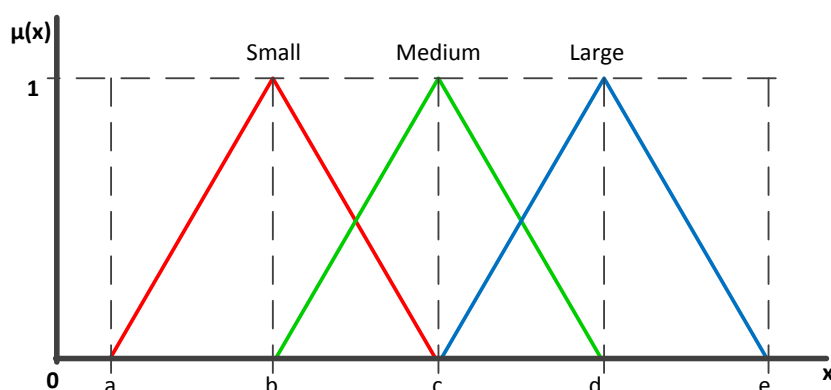
Částečná příslušnost je formalizována zavedením tzv. míry příslušnosti. Míra příslušnosti může nabývat jakéhokoliv čísla v intervalu $\langle 0,1 \rangle$. [19]

Například výraz:

$$\mu_A(x) = 0,7 \quad (4.13)$$

vyjadřuje míru (funkci) příslušnosti prvku x (prvek x je součástí tzv. univerza X , tj. hodnot, kterých může nabývat) do fuzzy množiny A . V tomto případě prvek do množiny „spíše patří“ (míra příslušnosti je bližší hodnotě 1 než 0), avšak nelze to s jistotou určit. Výhodou tohoto řešení je skutečnost, že můžeme vágní (neurčité) jazykové proměnné, (například Small, Medium, Large) formalizovat do počítačové podoby prostřednictvím funkcí příslušnosti (*membership functions*). [19] Někdy se setkáváme s pojmem „crisp“ funkce příslušnosti. [20]

Tyto funkce mohou mít různé průběhy, například trojúhelníkový, průběh Gaussovy křivky apod. Jsou vždy definovány určitým množstvím parametrů. [19] Například trojúhelníková funkce příslušnosti Small zobrazená na obrázku je definována parametry a, b, c , které určují, kde míra příslušnosti nabývá hodnoty vyšší než 0, dále kde má funkce maximum a opět kde je míra příslušnosti nulová.



Obr. 4.2 Příklad funkcí příslušnosti

Fuzzy model Takagi-Sugeno implementovaný v prostředí MATLAB musí být modelem nultého nebo prvního řádu. Znamená to, že výstup ze systému může být konstanta nebo lineární funkce. V obou případech je ve výsledku výstupem ostré číslo, tzv. singleton. [21]

4.2.2 Struktura modelu ANFIS

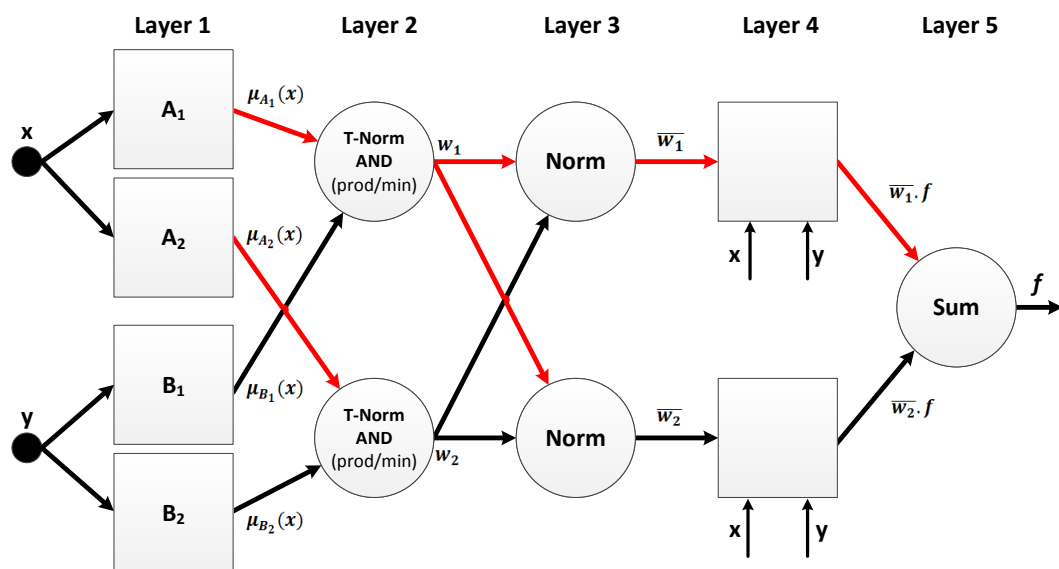
Model ANFIS sestává z pěti vrstev, z nichž každá má specifickou roli, přičemž pouze uzly v první a čtvrté vrstvě jsou adaptovány v průběhu procesu trénování modelu, obdobně jako v případě neuronových sítí. Ostatní uzly jsou statické a slouží jen ke zpracování dat. [22]

Na obrázku je znázorněna struktura modelu ANFIS se dvěma vstupy, přičemž každý z nich může náležet s určitou mírou příslušnosti k jedné ze dvou funkcí příslušnosti (A_1, B_1 pro vstup x a A_2, B_2 pro vstup y). Je nutné definovat tzv. fuzzy pravidla typu IF-THEN. Část před operátorem THEN je nazývána antecedent. Druhá část pravidla je nazývána konsekvent. [19] Pro model Takagi-Sugeno prvního řádu mohou pravidla vypadat například takto:

IF x is A_1 **and** y is B_1 , THEN $f_1 = p_1x + q_1y + r_1$.

Taková fuzzy pravidla jsou jednou z nejpřirozenějších reprezentací znalostí. [20]

V tomto příkladu by bylo nutné definovat čtyři fuzzy pravidla (kombinace A_1, B_1 ; A_1, B_2 ; A_2, B_1 ; A_2, B_2). Dále v textu následuje popis jednotlivých vrstev, více zde [22].



Obr. 4.3 Struktura modelu ANFIS se dvěma vstupy [23]

Vrstva 1

Zde jsou určeny míry příslušnosti vstupů k jednotlivým funkcím příslušnosti. Uzly v této vrstvě jsou adaptivní, což znamená, že jsou zde na základě chyby systému (rozdíl mezi požadovanými výstupy a výstupy ze systému) adaptovány parametry funkcí příslušnosti (*premise parameters*), například výše uvedené parametry a, b, c .

Vrstva 2

Tato vrstva je statická, uzly realizují fuzzy logickou operaci AND. Může se jednat o jakýkoliv operátor T-normy, například operátor Produkt (*Prod*) nebo Minimum (*Min*), přičemž výstup z operátoru Produkt vypadá následovně:

$$w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(x), \quad i = 1, 2 \quad (4.14)$$

Vrstva 3

Tato vrstva je statická a provádí normování výstupů z předchozí vrstvy takto:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \quad (4.15)$$

Vrstva 4

Uzly této vrstvy jsou adaptovány v průběhu trénování modelu ANFIS, konkrétně se jedná o adaptaci parametrů p_i, q_i, r_i . Tyto jsou označovány výrazem *consequent parameters*.

$$\bar{w}_i \cdot f_i = \bar{w}_i(p_i x + q_i y + r_i), \quad i = 1,2 \quad (4.16)$$

Vrstva 5

V poslední statické vrstvě je vypočítán celkový výstup systému:

$$\sum_i \bar{w}_i \cdot f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, \quad i = 1,2 \quad (4.17)$$

4.2.3 Algoritmus pro adaptaci parametrů modelu ANFIS

Standardně je používá hybridní algoritmus adaptace parametrů modelu ANFIS. Celý proces adaptace je rozdělen na dvě fáze.

V první fázi dochází k tzv. dopřednému průchodu (*forward pass*). Systému jsou postupně předloženy všechny tréninkové vzory a následně identifikovány parametry ve čtvrté vrstvě prostřednictvím metody LSE (*Least-Squares Estimation*). [22]

Ve druhé fázi dochází ke zpětnému průchodu (*backward pass*). Chyby systému jsou rozšiřovány z výstupní vrstvy směrem ke vstupní vrstvě prostřednictvím metody Backpropagation, konkrétně se jedná o algoritmus Gradient Descent (využívaný také ve standardních neuronových sítích). Během tohoto zpětného šíření jsou updatovány parametry v první vrstvě. [22]

Tab. 4.1 Fáze adaptace parametrů modelu ANFIS [22]

	Forward pass	Backward pass
Premise parameters	Fixed	Gradient Descent
Consequent parameters	Least-Squares Estimation	Fixed
Signals	Node outputs	Error signals

5. Kapitola

Sestavení modelu stanovení vývoje Parkinsonovy choroby

Pro účely návrhu systému pro vyhodnocování míry závažnosti onemocnění Parkinsonovou chorobou je v této práci použito především programovací prostředí MATLAB, společně s rozšířeními Neural Network Toolbox™ (NNT), Fuzzy Logic Toolbox™ (FLT). Alternativou k těmto výpočetním nástrojům je například software Rapidminer 5, jehož možnosti jsou obsahem podkapitoly 5.4.

5.1 Statistická analýza a rozdělení poskytnutých dat

Před započítím návrhu systému je nutné provést základní statistickou analýzu dat, která jsou posléze používána k trénování, validaci a testování výpočetních modelů, a to z toho důvodu, aby bylo možné optimalizovat jejich tréninkové parametry. Výstupy ze statistické analýzy jsou zobrazeny v podobě histogramů a krabicových grafů z důvodu snadné orientace a přehlednosti.

Pro statistickou analýzu je vytvořen skript *thes_dataset_statistics.m*, který pro jednotlivé příznaky datového souboru graficky a numericky vyhodnocuje minima, maxima, střední hodnoty, směrodatné odchylky, z grafických výstupů zobrazuje o krabicové grafy a histogramy.

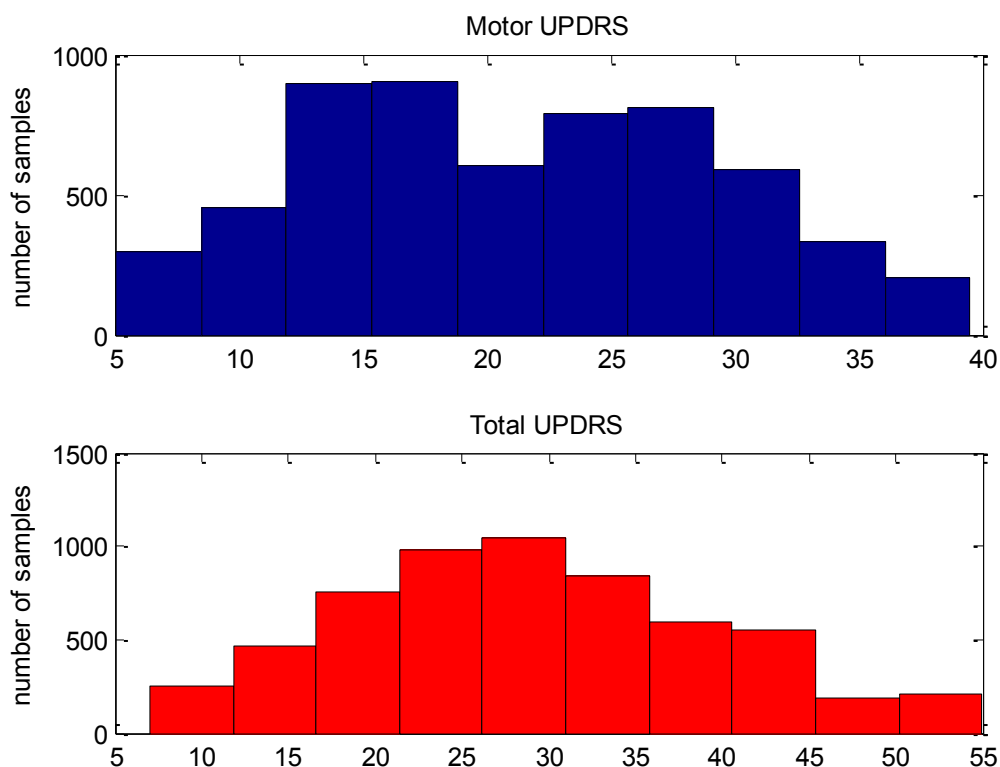
5.1.1 Statistická analýza vstupních signálů

Poskytnutý data set obsahuje 5875 vzorků měření kolísání řečového signálu, přičemž každý signál je určen 16 příznaky. Grafické výstupy této analýzy jsou uvedeny na Obr. II.1 až Obr. II.4. Význam jednotlivých příznaků je uveden v části Příloha VII.

Z krabicových grafů je patrné, že jednotlivé příznaky nabývají řádově jiných hodnot. Zatímco příznak *Jitter(abs)* nabývá hodnot řádově $\pm 10^{-4}$, hodnoty příznaku *HNR* se pohybují v rozmezí $5 \div 35$. Tato skutečnost předznamenává potřebu normování vstupních dat před jejich přeložením výpočetním modelům. Dále je z grafů evidentní, že mnoho příznaků obsahuje významný počet odlehlých hodnot, což implikuje skutečnost, že chyby systému (MSE a MAE) budou nabývat větší směrodatné odchylky. Obr. II.4 názorně demonstruje rozdíly v rozsazích hodnot jednotlivých příznaků.

5.1.2 Statistická analýza požadovaných výstupů

Data set obsahuje dva druhy požadovaných výstupů – Motor UPDRS (nabývá hodnot $0 \div 108$, viz kapitola 2) a Total UPDRS (nabývá hodnot $0 \div 176$). Význam těchto výstupů je vymezen v podkapitole 3.1.3. Na Obr. 5.1 je v podobě histogramů zobrazeno rozložení hodnot UPDRS. Zatímco rozložení hodnot Motor UPDRS nepřipomíná žádné známé rozložení, rozložení hodnot Total UPDRS připomíná Gaussovo rozložení.



Obr. 5.1 Histogramy hodnot stupnice UPDRS

5.1.3 Rozdělení vstupních dat pro neuronové sítě

Parkinson's Telemonitoring Data Set použitý v této práci obsahuje velké množství dat, z nichž ne všechna lze použít. Před zpracováním je nutné odstranit data týkající se identifikace pacientů (ID pacientů, jejich pohlaví, věk atp.), přičemž samotný řečový signál reprezentuje zmiňovaných 16 příznaků. Importování a úprava poskytnutého datového souboru je prováděna ve skriptu *thes_1_import.m*.

Celý datový soubor (na Obr. 5.2 vlevo) uložený v souboru *Tsanas_TBME2010_data.mat* je rozdělen do dvou velkých celků:

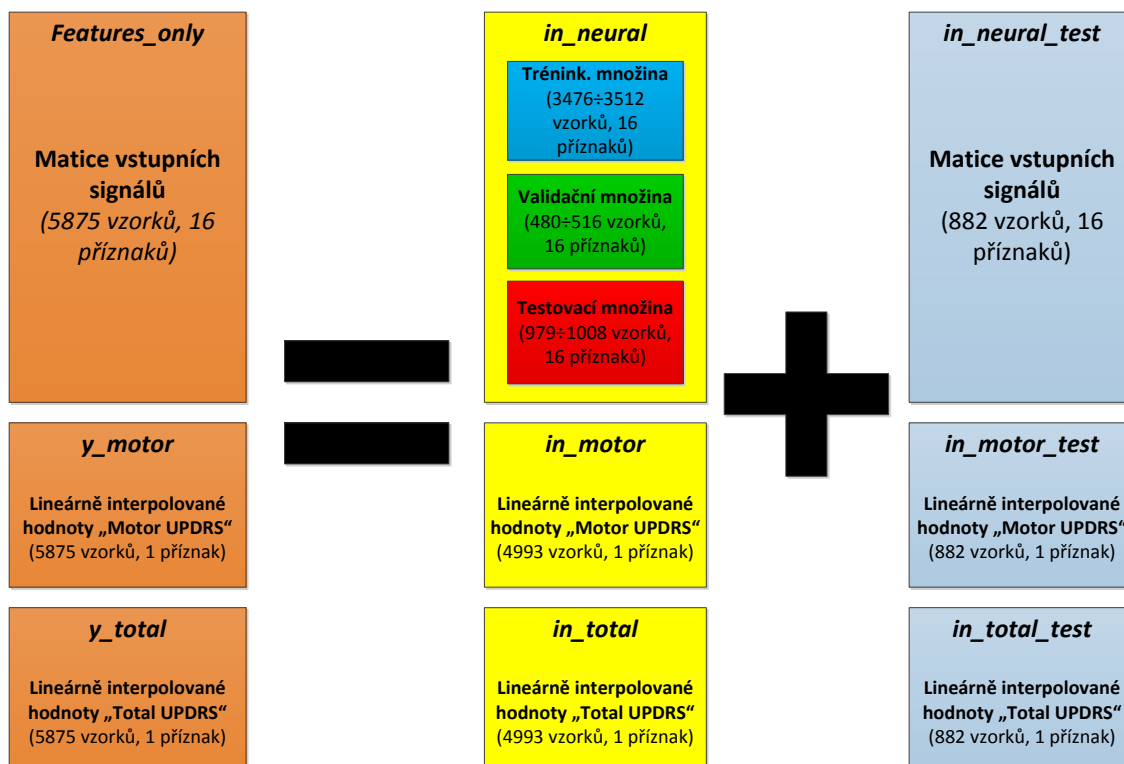
- první část (na Obr. 5.2 uprostřed) tvoří 85% dat z celkového datového souboru a je určena pro trénování, validaci a testování výpočetních modelů pro stanovení vývoje Parkinsonovy choroby z řečového signálu. Tato data jsou uložena v proměnných *in_neural*, *in_motor* a *in_total*.
- druhá část (na Obr. 5.2 vpravo) sestává z 15% celkového datového souboru a je explicitně určena pro nezávislý test, na základě kterého je v této diplomové práci hodnocena kvalita výpočetního modelu. Tato testovací data jsou uložena obdobně v proměnných *in_neural_test*, *in_motor_test* a *in_total_test*.

Pro trénování a testování výpočetních modelů je zásadní, aby obě proměnné *in_neural* a *in_neural_test* zahrnovaly veškerá možná specifika, včetně minimálních a maximálních hodnot, které se v dostupném datovém souboru vyskytují. Jednotlivá měření řečového signálu (každé měření představuje 1 řádek transponované matice *in_neural*) byla totiž zaznamenána chronologicky, z čehož vyplývá, že v řádcích s vyšším pořadovým číslem je míra závažnosti onemocnění Parkinsonovou chorobou obecně vyšší z toho důvodu, že nemoc se v mezidobí u pacientů rozvinula.

Z toho důvodu jsou pomocí náhodných indexů data v první a druhé části vybrána tak, aby bylo rozložení dat z měření rovnoměrné. Referencí pro rovnoměrné indexování byly požadované hodnoty Motor UPDRS a Total UPDRS. Na Obr. III.1 a Obr. III.2 je z histogramů zřejmé, že rozložení těchto vektorů je téměř totožné, a lze tedy předpokládat, že rozložení dat pro vývoj výpočetních modelů a pro nezávislý test je rovnoměrné.

Co se týče dat pro trénování, validaci a testování, která jsou souhrnně uložena v proměnné *in_neural*. Jedná se o 4993 vzorků měření, přičemž pro potřeby desetinásobné křížové validace jsou pomocí náhodných indexů rozdělena přibližně v poměru 70% dat pro trénování, 10% dat pro validaci a 20% dat pro testování výpočetních modelů. Toto rozdělení je realizováno prostřednictvím vytvořených funkcí *thes_crossval.m* a *thes_trainvaltest.m*. Výraz přibližně je použit proto, že počty měření řečového signálu v množinách pro trénování, validaci a testování pro jednotlivé iterace desetinásobné validace nejsou vždy shodné. Rozsahy počtů měření jsou znázorněny v následujícím obrázku.

Aby byly pro návrh výpočetních modelů zajištěny vždy stejné podmínky, indexování pro rozdělení do tréninkových, validačních a testovacích množin je uloženo do souboru *indices.mat*, který je v rámci úpravy dat realizované ve skriptu *thes_I_import.m*.



Obr. 5.2 Rozdělení vstupních dat pro návrh neuronových sítí

5.1.4 Rozdělení vstupních dat pro metodu ANFIS

Pro návrh systému ANFIS je rozdělení souboru dat obdobné. Jediný rozdíl spočívá v rozdělení dat obsažených v proměnné *in_neural*, která tvoří jednu matici bez indexování, protože metoda ANFIS nevyžaduje rozdělení dat na tréninkovou, validační a testovací množinu. Grafické znázornění těchto rozdělených dat je součástí přílohy. Návrh probíhá tak, že proměnná *in_neural* (na Obr. IV.1 prostřední sloupec) slouží pro adaptaci (trénování) modelu ANFIS, a posléze je proveden nezávislý test na datech uložených v proměnné *in_neural_test* (na Obr. IV.1 pravý sloupec).

5.2 Paralelizace iterativních výpočtů

Trénink a testování výpočetních modelů je mnohdy velmi náročnou procedurou vyžadující i při vysokém výpočetním výkonu mnoho strojového času. Pokud je k dispozici systém s více jádry/processory, je možné výpočet provést paralelně, a tím celý proces učení zrychlit. Především při implementaci neuronových sítí, které jsou ze své podstaty vhodné pro paralelizaci, lze dosáhnout výrazného zrychlení. Jednotlivé neurony, z nichž jsou neuronové sítě sestaveny, jsou totiž samy o sobě jednoduché paralelní procesory. Avšak i při použití metod

nevhodných pro paralelizaci jako takovou, lze přistoupit tímto způsobem, pokud je nutné provést mnoho nezávislých iterací tréninku téhož modelu.

V prostředí MATLAB[®] je možné paralelní výpočty realizovat pomocí Parallel Computing Toolboxu[™] (PCT) ve spojení s nástrojem Distributed Computing Server[™] (DCS). Tyto nástroje nabízí mnoho vestavěných funkcí, které lze ovšem přizpůsobit k potřebám konkrétního modelu. Princip paralelního zpracování dat je založen na spuštění určitého množství relací (*sessions*) prostředí MATLAB, přičemž tyto jsou spuštěny na pozadí. Hlavní relace, se kterou uživatel pracuje, tedy ta, s níž má interakci prostřednictvím grafického uživatelského rozhraní, je nazývána klient (*client*). Z této hlavní relace jsou spuštěny další relace nazývané *worker*, které společně s klientem realizují paralelní zpracování dat ve formě tzv. svazku relací, anglicky *pool*. [24] Relace *worker* tedy pracují v neinteraktivním režimu.

V případě použití 4 jádrového procesoru jsou tedy spuštěny 4 relace MATLAB. Teoreticky to znamená, že paralelní výpočet nad požadovanými daty je čtyřikrát rychlejší než výpočet sekvenční. Je však nutné vzít v potaz, že komunikace mezi jednotlivými relacemi také zabírá určitý strojový čas, takže faktické zrychlení výpočtu je menší.

5.2.1 Charakteristika Parallel Computing Toolboxu[™]

Při paralelizaci je možné využít více typů výpočetních zdrojů. Kromě standardních procesorů se dvěma a více jádry umožňuje PCT využívat také grafické karty. V současné době ovšem PCT podporuje pouze grafické procesory společnosti Nvidia s technologií CUDA. [24]

Základní příkazy používané pro paralelní výpočty v této práci:

- ***matlabpool*** – spouští relace *worker* pro paralelní výpočty v rámci PCT, jeho parametry specifikují mimo jiné počet spuštěných relací a typ clusterového profilu (defaultně profil *local*). Tento příkaz je od verze MATLAB R2013b nahrazen příkazem ***parpool***, přičemž ze strany vývojářů je kladen důraz na objektovou podstatu svazku relací (*pool*)
- ***parfor*** – paralelně vykonávaná smyčka *for*. Její použití je podmíněno skutečností, že jednotlivé iterace smyčky musí být na sobě vzájemně nezávislé, aby mohly být realizovány samostatně.
- parametr ***'useParallel'*** v případě trénování neuronových sítí příkazem *train* a v případě výpočtu výstupů této sítě prostřednictvím vytvořené struktury *net*. Tento parametr tyto výpočty paralelizuje, přičemž výpočetními zdroji jsou jádra procesoru, respektive více procesorů.
- parametr ***'useGPU'*** v případě trénování neuronových sítí příkazem *train* a v případě výpočtu výstupů této sítě prostřednictvím vytvořené struktury *net*. Tento parametr tyto výpočty paralelizuje, přičemž výpočetními zdroji jsou grafické procesory.

Je třeba brát v úvahu, že spuštění více relací MATLABu trvá určitý čas, mnohdy v řádu minut. Je to způsobeno tím, že je na pozadí spouštěno tolik programů MATLAB v neinteraktivním režimu, kolik je stanoveno relací *worker* v parametrech příkazu *matlabpool*.

5.2.2 Clusterové profily pro paralelní výpočty

Každý paralelní výpočet v prostředí MATLAB musí být v rámci počítačového clusteru (popřípadě jednoho PC při využití více jader procesoru) spravován určitým clusterovým profilem. Tyto profily mají za úkol plánovat výpočetní úlohy a postupovat je jednotlivým výpočetním uzlům. Zároveň clusterové profily zajišťují komunikaci mezi klientem (hlavní relací MATLABu) a relacemi *worker* a synchronizují získané výsledky. V této podkapitole následuje popis dvou clusterových profilů použitých v rámci této diplomové práce a jejich konfigurace.

Local

Defaultně aktivovaný profil při použití příkazu *matlabpool* bez parametrů. Počet spuštěných relací se implicitně rovná počtu jader procesoru. Maximální počet relací *worker* je u lokálního profilu omezen na 12 (toto omezení je však odstraněno od verze MATLABu R2014a). Výhodou tohoto clusterového profilu je skutečnost, že není třeba provádět žádnou konfiguraci. V případě potřeby více než 12 relací je však nutné použít některý jiný clusterový profil. Nicméně pro potřeby paralelních výpočtů na běžných PC je zcela vyhovující.

MPIEXEC

Jedná se o clusterový profil využívající knihovnu MPI (*Message Passing Interface*) vytvořenou pro paralelní výpočty realizované na počítačových clusterech. Na rozdíl od lokálního profilu neomezuje počet paralelních relací. [24] Pro tuto vlastnost je tedy často využíván při náročnějších výpočtech, a také pro potřeby této diplomové práce.

Konfigurace sestává z několika kroků. V první řadě je nutné vytvořit konfigurační skript *mpiLibConf.m*, který má za úkol MATLABu určit umístění knihovny MPI v daném výpočetním systému. Příklad tohoto konfiguračního skriptu je uveden v části Příloha V, přičemž zdrojem je citovaná reference, viz [25].

Dále musí být výpočetní skript typu *mfile* doplněn o nastavení plánovače paralelních výpočtů. V něm je specifikován typ plánovače (*mpiexec*) a počet spouštěných relací. Na závěr výpočtu je nutné všechny spuštěné relace ukončit. Celý výpočet v rámci profilu MPIExec je počítačovému clusteru postoupen prostřednictvím standardního PBS skriptu. Všechny relace (tedy hlavní relace *client* i relace *worker*) jsou spuštěny v rámci jednoho PBS. [25]

5.3 Neuronové sítě v prostředí MATLAB Neural Network Toolbox™

Stěžejním nástrojem k vývoji modelu pro mapování vstupních příznaků na požadované výstupy, jsou v této diplomové práci neuronové sítě. Výše uvedené rozšíření Neural Network Toolbox™ (NNT) disponuje mnoha prostředky, které jsou schopny efektivně nalézt výpočetní model, včetně nástrojů pro ladění prostřednictvím změn parametrů neuronových sítí, jejich topologie, kvalitativní validace a vyhodnocení výsledků. Vzhledem k tomu, že dostupná měřená data použitá v této práci jsou výrazně nelineární, je nutné defaultní nastavení v NNT výrazně modifikovat, přidat určité elementy a neposlední řadě výsledky validovat prostřednictvím desetinásobné křížové validace z toho důvodu, že neuronové sítě jsou ze své podstaty velmi náchylné na inicializaci synaptických vah mezi neurony. Důležitou úlohu hraje také pre-processing vstupních dat z toho důvodu, že jednotlivé vstupní příznaky datového souboru nabývají řádově odlišných hodnot.

5.3.1 Režimy trénování neuronových sítí

V NNT je možné realizovat trénování neuronových sítí ve dvou zcela odlišných režimech. Jsou jimi tyto režimy:

- **Dávkový režim** (*batch mode*) – synaptické váhy a prahy neuronů jsou adaptovány až poté, co jsou neuronové sítě předloženy všechny dostupné tréninkové vzory (vstupy i požadované výstupy). Metoda Backpropagation je tedy v rámci této práce použita ke zpětnému šíření chyby po průběhu přibližně 3500 tréninkových vzorů, což definuje jednu tréninkovou iteraci (epochu). Při použití dávkového režimu musí být data uložena ve formě vektorů a matic. Standardně se tento režim trénování spouští příkazem *train*. Výhodou tohoto režimu je fakt, že gradienty chybové funkce vypočítané po předložení jednotlivých tréninkových vzorů jsou posléze zprůměrovány, takže výsledný gradient vypočítaný v jedné tréninkové epoše je přesnější. Tento režim se také označuje *offline*.
- **Inkrementální režim** (*incremental mode*) – v tomto režimu jsou synaptické váhy a prahy neuronů aktualizovány po průchodu každého tréninkového vzoru. Data musí být uložena ve formátu typu *cell*. Typicky se tento režim používá u neuronových sítí se zpoždovacími prvky (*delays*) nebo se zpětnými vazbami. Proces trénování se spouští příkazem *adapt*. Tento režim se také označuje *online*. [16]

V rámci této diplomové práce je používán pouze dávkový režim, který při zpracování dat měření řečového signálu vykazoval výrazně lepší výsledky.

K ukončení tréninku modelu neuronové sítě dochází standardně v případě:

- dosažení maximálního stanoveného počtu tréninkových epoch (iterací tréninku),
- uplynutím maximálního stanoveného času pro trénink,

- dosažením stanovené minimální chyby při minimalizaci chybové funkce validační množiny,
- dosažením stanoveného minima hodnoty gradientu chybové funkce,
- při růstu chyby validační množiny během stanoveného počtu po sobě jdoucích tréninkových epoch (což je nejčastější případ, tento jev se nazývá *early stopping*). Toto ovšem není implementováno u algoritmu Bayesian Regularization. [16]

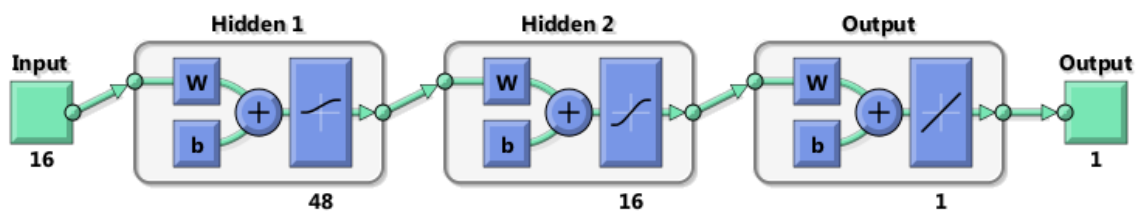
Některé tréninkové algoritmy implementují i jiné jevy ukončující trénink neuronové sítě.

5.3.2 Neuronové sítě s dopředným šířením signálu

Tyto neuronové sítě, v angličtině označované *feedforward neural networks*, se vyznačují tím, že užitečný signál (nejedná se o přenos chyby) je šířen pouze ve směru ze vstupní vrstvy sítě do výstupní vrstvy. Neuronová síť tedy neobsahuje žádnou zpětnou vazbu (kterou mají tzv. rekurentní neuronové sítě). Zároveň je signál ze vstupní vrstvy přiveden pouze na vstupy první skryté vrstvy neuronů. Tento typ neuronových sítí je schopen naučit se v podstatě jakoukoliv vstupně-výstupní závislost, včetně nelineárních závislostí. Je tedy vhodný pro potřeby této diplomové práce. V rámci metody Backpropagation je neuronovou sítí zpětně šířena pouze velikost chyby, ovšem ne vstupní signál jako takový.

Tento typ neuronové sítě lze vytvořit příkazem *fitnet* (popřípadě příkazem *feedforwardnet*, který je v podstatě stejný). Argumentem příkazu je počet neuronů ve skrytých vrstvách. Zároveň při vytvoření topologie sítě jsou inicializovány synaptické váhy a prahy neuronů. Lze použít také GUI (spouští se příkazem *nftool*), které ovšem nabízí mnohem méně možností ladění neuronové sítě, než příkazy v příkazové řádce.

Na Obr. 5.3 je zobrazena neuronová síť s 16 vstupy, 48 neurony v první skryté vrstvě (aktivační funkce neuronů je logaritmická sigmoida), 16 neuronů ve druhé skryté vrstvě (aktivační funkce hyperbolická tangenta), a 1 výstupem (aktivační funkce je lineární funkce).



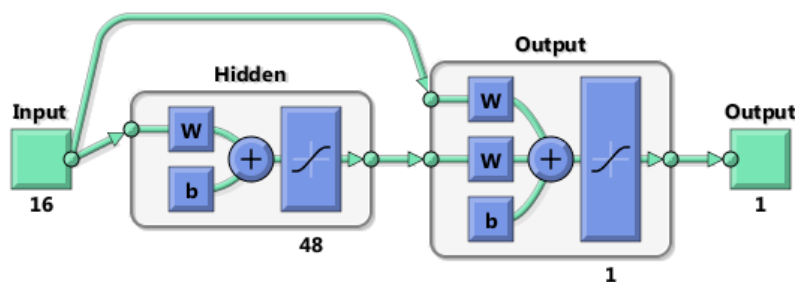
Obr. 5.3 Příklad neuronové sítě s dopředným šířením signálu

Počty neuronů ve skrytých vrstvách standardně nebývají stejné. V této diplomové práci jsou používány neuronové sítě s jednou skrytou vrstvou (počet neuronů je z důvodů velkých výpočetních nároků omezen na 96) a se dvěma skrytými vrstvami, přičemž poměr mezi počty neuronů v první a druhé skryté vrstvě je $1/2$, $1/3$ a $1/4$.

5.3.3 Kaskádové neuronové sítě

Na rozdíl od neuronových sítí s dopředným šířením signálu jsou vrstvy kaskádových neuronových sítí propojeny komplexněji. Její architektura je vytvořena příkazem *cascadeforwardnet*, vstupním argumentem je počet neuronů ve skrytých vrstvách. Obecně je každá vrstva neuronů propojena s výstupy všech předchozích vrstev a navíc je také propojena přímo se vstupem neuronové sítě. Při aplikaci kaskádových neuronových sítí na datový soubor měření oscilace řečového signálu lidí postižených Parkinsonovou chorobou se osvědčily pouze ty, které mají jen jednu skrytou vrstvu. Kaskádové neuronové sítě s více skrytými vrstvami u tohoto data setu vykazovaly velmi velké chyby, takže nebyly dále využívány.

Příklad může být neuronová síť znázorněná na Obr. 5.4, která má 16 vstupů, jednu skrytou vrstvu se 48 neurony s aktivační funkcí typu hyperbolická tangenta a výstupní vrstvu s jedním neuronem. Na výstupní vrstvu je přiveden výstup ze skryté vrstvy a přímo také signál ze vstupu neuronové sítě.

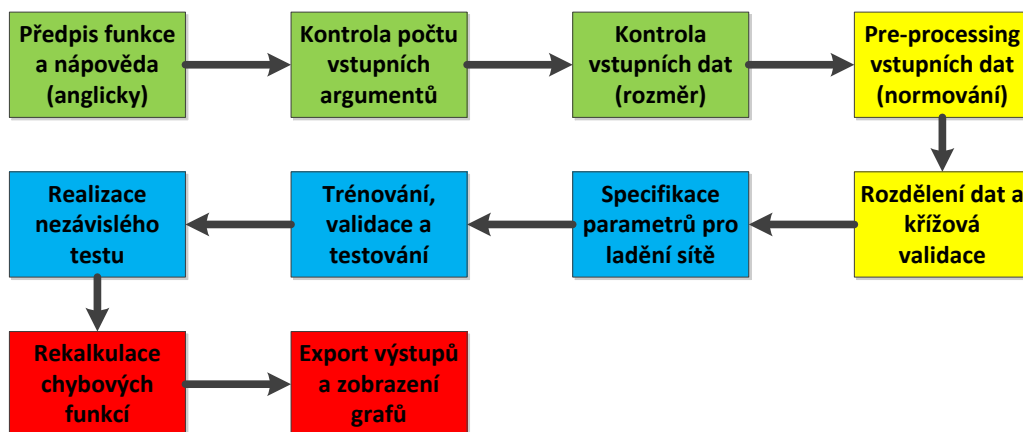


Obr. 5.4 Příklad kaskádové neuronové sítě

Aktivační funkcí neuronu ve výstupní vrstvě standardně nebývá lineární funkce, ale hyperbolická tangenta (*tansig*) nebo logaritmická sigmoida (*logsig*).

5.3.4 Funkce `thes_neural.m`

Tato funkce tvoří v této práci základ návrhu modelů neuronových sítí. Je vytvořena tak, aby vyhověla požadavku na ladění všech standardních typů neuronových sítí typu *feedforward net*, včetně specifikací parametrů všech hlavních tréninkových algoritmů. Její jádro je založeno na funkcích implementovaných v NNT, avšak přináší plno vylepšení, mimo jiné v podobě struktury pro ladění sítě (intuitivní editaci parametrů), rekalkulace chybových funkcí a provedení nezávislého testu, na základě kterého je hodnocena kvalita jednotlivých modelů neuronových sítí. Funkce sestává z deseti bloků s tím, že zelené bloky provádějí kontrolu vstupních dat, žluté bloky upravují tato data před použitím, bloky modré barvy realizují ladění neuronové sítě a výpočet výstupů a červené bloky slouží k závěrečnému hodnocení modelů neuronových sítí.



Obr. 5.5 Algoritmus naprogramovaného modelu neuronové sítě

Popis bloků funkce:

- **Předpis funkce a nápověda** – definuje požadované vstupní argumenty a možné výstupy z funkce. Uživatel musí poskytnout alespoň dvě proměnné typu *struct*. První (nazvaná *in_data*) obsahuje matici dat pro trénování, validaci a testování společně s vektorem požadovaných výstupů, popřípadě také další data pro nezávislý test. Druhá (nazvaná *in_arg*) specifikuje parametry pro ladění sítě. Dále musí poskytnout vektor *indices* pro určení počtu neuronů ve skrytých vrstvách.
- **Kontrola počtu vstupních argumentů** – stanovuje minimální a maximální počty vstupních argumentů a použitých proměnných na výstupu ze systému. Uživatel může kromě povinných vstupů poskytnout také proměnnou typu *struct* obsahující indexy pro křížovou validaci a dále může určit, zda mají být zobrazovány grafické výstupy.
- **Kontrola vstupních dat** – transponuje a upravuje vstupní data v případě nevyhovujícího formátu.
- **Pre-processing vstupních dat** – odstraňuje lineárně závislé vzorky a normuje vstupní data do rozsahu $\langle -1, 1 \rangle$. V proměnné *in_arg* lze zvolit i jinou metodu normování dat.
- **Rozdělení dat a křížová validace** – implementuje dva režimy rozdělení dat, a to buď náhodné rozdělení dat pro ladění sítě, anebo pomocí indexování pro křížovou validaci v případě, že je poskytnuta proměnná obsahující indexy pro trénování, validaci a testování.
- **Specifikace parametrů pro ladění sítě** – importuje parametry uložené ve vstupní proměnné *in_arg*. Tato proměnná definuje mimo jiné počet tréninkových epoch, aktivační funkce vrstev neuronů, maximální dobu tréninku a další. V případě, že nějaký parametr pro konkrétní tréninkový algoritmus není specifikován, je nastavena defaultní hodnota. V případě požadavku na nastavení defaultní hodnoty je také možné konkrétní parametr inicializovat na hodnotu *NaN*. Dále je zde určeno, zda má být vývoj modelu realizován paralelně prostřednictvím Parallel Computing Toolboxu™.

- **Trénování, validace a testování** – jádro funkce založené na vestavěných příkazech NNT, především na příkazu *train*. Trénování, validaci a testování lze provést paralelně, včetně následného výpočtu výstupů ze systému. Trénink je ukončen v momentě, kdy hodnota chybové funkce validační množiny roste v definovaném počtu po sobě jdoucích tréninkových iterací (epoch). Poté se hodnot synaptických vah a prahů sítě nastaví na hodnoty dosažené v epoše, kdy bylo nalezeno minimum chybové funkce.
- **Realizace nezávislého testu** – provedena po ukončení vývoje modelu neuronové sítě na datech, která nebyla součástí datového souboru pro trénování, validaci a testování. Model neuronové sítě je tedy zpracovává poprvé. Na základě nezávislého testu je určována kvalita výpočetního modelu.
- **Rekalkulace chybových funkcí** – provedena z důvodu verifikace výsledného nejefektivnějšího modelu neuronové sítě. Hodnoty MSE, MAE a korelačního koeficientu tréninkové, validační, testovací množiny a dále výstupy z nezávislého testu jsou uloženy do proměnné typu *struct* nazvané *recal*.
- **Export výstupů a zobrazení grafů** – grafické výstupy zobrazují průběh chybových funkcí jednotlivých množin, regresní analýzu, histogramy chybové funkce a další. Z dalších výstupů se jedná o proměnné *net* (výsledný model neuronové sítě), dále proměnná *tr* (*training record* neboli tréninkový záznam, obsahuje veškeré údaje o průběhu vývoje modelu). Dále je možné (volitelně) exportovat hodnoty UPDRS pro určení závažnosti onemocnění Parkinsonovou chorobou, včetně hodnot UPDRS z nezávislého testu, a vektor chyb systému.

Pro kaskádové neuronové sítě je vytvořena funkce *thes_neural_cascade.m*, která je v podstatě stejná, avšak pro vytvoření architektury neuronové sítě je použit příkaz *cascaforwardnet*.

5.3.5 Iterativní ladění neuronových sítí

Pro nalezení modelu neuronové sítě, který je schopen co nejefektivněji mapovat data měření řečového signálu na požadované výstupy, jimiž jsou hodnoty na klinické stupnici UPDRS, je nezbytné provést velké množství simulací. V těchto simulacích jsou iterativně měněny tréninkové parametry stanovující mimo jiné počty neuronů ve skrytých vrstvách neuronových sítí, dále typy aktivačních funkcí jednotlivých skrytých vrstev, minimální hodnoty gradientu chybové funkce, a další parametry.

Vzhledem k tomu, že určité skupiny tréninkových algoritmů mají společné parametry pro ladění sítí, je pro potřeby této práce vytvořen soubor několika skriptů, které modely neuronových sítí iterativně simulují, pokaždé s odlišnými tréninkovými parametry. Z důvodu velké závislosti neuronových sítí na počáteční inicializaci synaptických vah a prahů neuronů je každý model simulován prostřednictvím křížové validace vícekrát. Tyto skripty obsahují několikanásobně vnořené smyčky *for* pro realizaci simulace neuronových sítí s mnoha iterativně měněnými parametry.

5.3.6 Aktivační funkce neuronů

Prakticky všechny tréninkové algoritmy neuronových sítí vyžadují, aby byly aktivační funkce (též označovány výrazem přenosové funkce) neuronů diferencovatelné, tedy aby měly spojité derivace, viz podkapitola 4.1.3. V úvahu tedy v podstatě přicházejí pouze tři typy aktivačních funkcí. Průběhy těchto aktivačních funkcí a jejich derivací jsou součástí přílohy této práce.

Funkce logaritmická sigmoida (logsig)

Tato funkce se v $-\infty$ limitně blíží hodnotě 0, naopak v $+\infty$ se limitně blíží hodnotě 1. Přechod z hodnoty 0 na 1 nastává v momentě, kdy nezávislá proměnná nabývá hodnot kolem 0. V podstatě to tedy znamená „smrštění“ hodnot vstupů do intervalu (0,1). Nevýhodou této funkce je fakt, že pokud vstupy do funkce nabývají velkých hodnot, neuronová síť se stává necitlivou na rozdíly mezi takto vysokými hodnotami a prodlužuje se doba konvergence.

$$\text{logsig}(x) = \frac{1}{(1 + e^{-x})} \quad (5.1)$$

Funkce hyperbolická tangenta (tansig)

Hyperbolická tangenta se v $-\infty$ limitně blíží hodnotě -1 , naopak v $+\infty$ se limitně blíží hodnotě 1. Přechod z hodnoty 0 na 1 nastává v momentě, kdy nezávislá proměnná nabývá hodnot kolem 0. V podstatě to tedy znamená „smrštění“ hodnot vstupů do intervalu $(-1,1)$. Na rozdíl od logaritmické sigmoidy, tato funkce lépe koresponduje s normovanými vstupy do neuronové sítě, které nabývají hodnot z intervalu $(-1,1)$. Dá se říct, že zachovává intervaly, ve kterých jsou normována vstupní data.

$$\text{tansig}(x) = \frac{2}{(1 + e^{-2x})} - 1 \quad (5.2)$$

Funkce radiální báze (radbas)

Funkce nabývá maxima, tedy hodnoty 1, v bodě $x = 0$, dále v $\pm\infty$ se limitně blíží hodnotě 0. Lze tedy tvrdit, že čím jsou hodnoty rozdíly mezi synaptickými vahami (resp. prahy) menší, tím je výstup z funkce větší. Tato funkce mnohdy vykazuje dobré výsledky v kombinaci s výše uvedenými funkcemi.

$$\text{radbas}(x) = e^{-x^2} \quad (5.3)$$

5.4 Neuronové sítě v programu Rapidminer 5

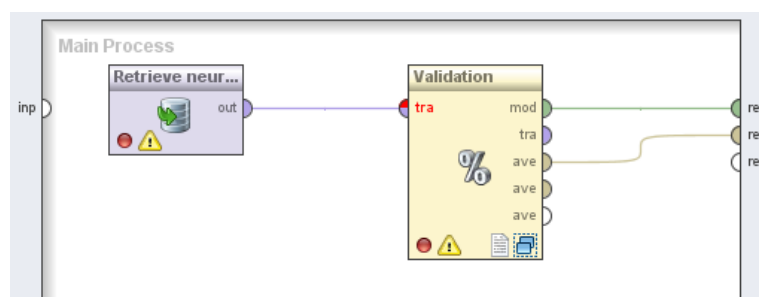
Software Rapidminer vytvořený stejnojmennou společností představuje alternativu k prostředí MATLAB v oblasti prediktivní analýzy dat a data miningu. Jedná se o Open-Source produkt, přičemž pro potřeby této diplomové práce je použita verze 5.3.015. Na rozdíl od prostředí MATLAB nejsou v programu Rapidminer používány skripty obsahující zdrojové kódy. Celý proces návrhu výpočetního modelu je realizován prostřednictvím spojování funkčních bloků, tzv. operátorů. Funkce těchto operátorů, obsazení vstupních a výstupních portů či příklady použití jsou sepsány v přehledných manuálech vydaných společností Rapidminer.

5.4.1 Konverze vstupních dat

Data předkládaná výpočetnímu modelu je před použitím nutné převést na formát vhodný pro program Rapidminer. Tato dílčí úloha je v zásadě jednoduchá. V prvním kroku je nutné datový soubor ve formátu *.xls* (popřípadě ve formátech *.csv*, *.xml* či jiných) importovat a prostřednictvím rozhraní *Data Import Wizard* jsou označena importovaná data, určeny typy a názvy proměnných, a především je zde stanoveno, která data slouží jako požadované výstupy (pro potřeby tréninku výpočetních modelů). Standardní vstupy jsou označovány přívlástkem *attribute*. Požadované výstupy jsou označovány přívlástkem *label*. Jakmile je konverze dat dokončena, v programovém repozitáři jsou vytvořeny tři soubory s příponami *.iioo*, *.md* a *.properties*.

5.4.2 Vytvoření modelu neuronové sítě

Návrh výpočetního modelu je realizován v několika úrovních procesů. Nejvyšší úroveň se nazývá *Main Process*. V této úrovni v podstatě stačí vytvořit dva operátory. Operátor *Retrieve* načítá zvolená data z programového repozitáře, přičemž tato data již musí být konvertována, viz předcházející podkapitola. Operátor *Validation* realizuje křížovou validaci, v defaultním nastavení je nad výsledky provedena desetinasobná křížová validace.

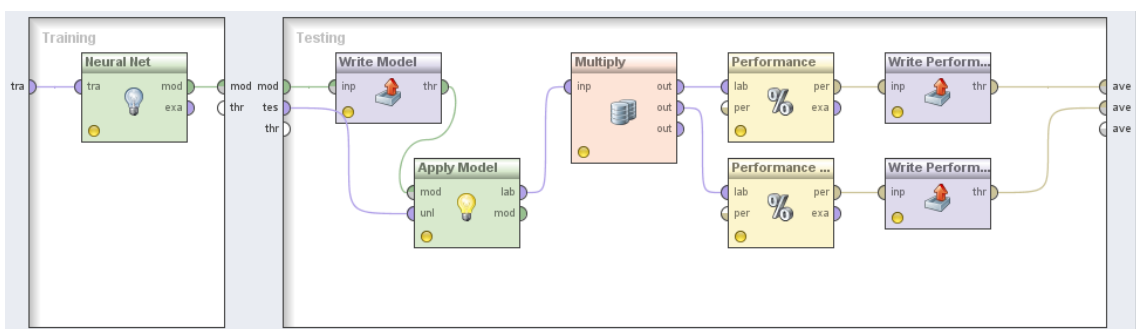


Obr. 5.6 Rapidminer 5 – úroveň Main Process

Uvnitř operátoru *Validation* je implementován model neuronové sítě. Veškeré v něm obsažené operátory tak v podstatě tvoří nižší úroveň procesů. V sekci *Training* se určuje typ výpočetního modelu. Defaultně sekce *Training* obsahuje operátor *Linear Regression*. Co se týče modelů

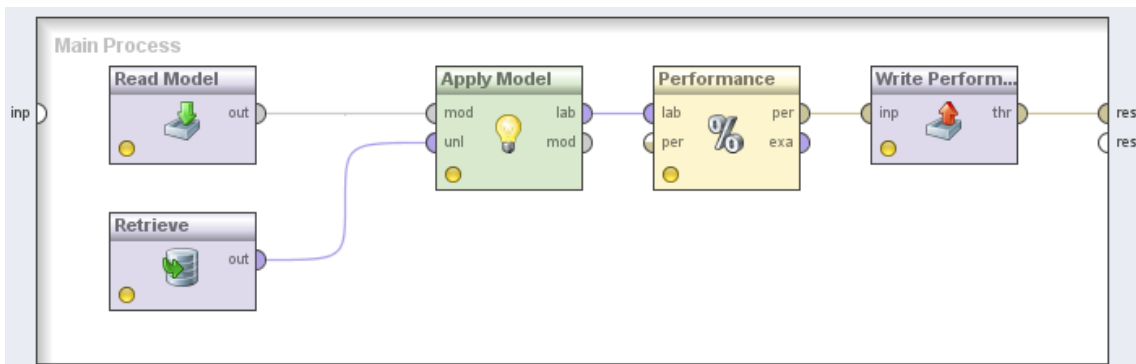
neuronové sítě, ze seznamu operátorů je možné vybrat operátory *Neural Net* (standardní neuronová síť s dopředným šířením signálu), a dále operátory *Perceptron* a *AutoMPL* (umožňuje iterativně měnit míru rychlosti učení a topologii neuronové sítě). Program Rapidminer nabízí kromě neuronových sítí i nespočet jiných klasifikačních a regresních modelů, například modely Random Forest či Support Vector Machine.

V sekci Testing je model prostřednictvím operátoru *Apply Model* validován a testován; výstupy jsou pak v operátorech *Performance* přepočítány na požadované statistické údaje (například MSE, MAE, koeficient korelace a další uživatelem zvolené statistické ukazatele) a zapsány do souboru. Operátor *Write Model* ukládá výsledný model do textového souboru (avšak ve formě archivu .xml souboru).



Obr. 5.7 Rapidminer 5 – úroveň Validation

Pro realizaci nezávislého testu je nezbytné vytvořit nový projekt (proces), ve kterém je načítán natrénovaný model neuronové sítě, a dále data určená pro nezávislý test. Výsledky a statistické údaje jsou opět ukládány do souboru.



Obr. 5.8 Rapidminer 5 – proces pro nezávislý test

5.4.3 Operátor Neural Net

V této práci je za výpočetní tréninkový model zvolen operátor *Neural Net*. V parametrech operátoru je nutné mimo jiné stanovit topologii neuronové sítě, tedy počty neuronů v jednotlivých skrytých vrstvách, dále pak počet tréninkových cyklů (epoch), míra rychlosti učení, parametr *momentum* (setrvačnost pro překonání lokálních minim chybové funkce), a požadovanou chybu výstupů. Jako aktivační funkce skrytých neuronů je použita sigmoida, proto je nezbytné vstupy normovat do rozsahu $(-1, 1)$. Normování vstupů je možné provést výběrem parametru *Normalize* v rámci operátoru *Neural Net*.

Pokud model realizuje regresní úlohu, aktivační funkcí ve výstupní vrstvě neuronové sítě je lineární funkce, v případě klasifikační úlohy se jedná opět o sigmoidu. [26]

Synaptické váhy mezi neurony a jejich prahy jsou modifikovány metodou Backpropagation. Návoděda ani manuály k programu Rapidminer bohužel explicitně nespecifikují, jaký konkrétní tréninkový algoritmus je k modifikaci vah využíván.

Numerické výstupy z neuronové sítě implementované v programu Rapidminer 5, včetně jejich popisu, jsou uvedeny v následující kapitole.

5.5 Model ANFIS v prostředí MATLAB Fuzzy Logic Toolbox™

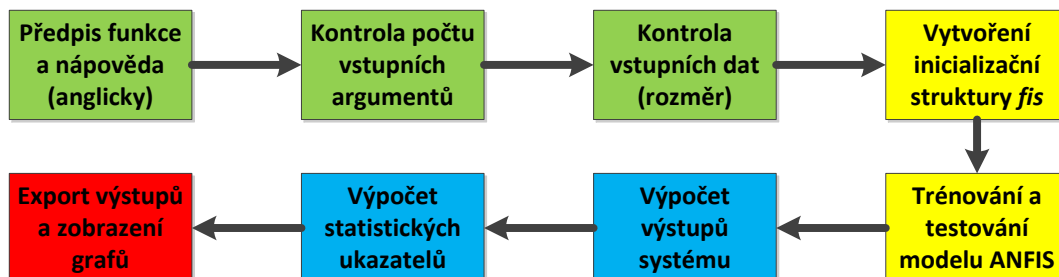
Metoda *Adaptive Neuro Fuzzy Inference System* (ANFIS) je v prostředí MATLAB implementována v nástroji Fuzzy Logic Toolbox™ (FLT). Jak je uvedeno v předchozí kapitole, realizace ANFIS je založena na modelu Takagi-Sugeno, jehož výhodou je fakt, že výstupem ze systému je skalár (ve formě konstanty nebo lineární funkce), což vyhovuje požadavku na požadovaný výstup, jímž je hodnota (taktéž skalár) na lékařské stupnici UPDRS sloužící pro hodnocení závažnosti onemocnění Parkinsonovou chorobou.

5.5.1 Charakteristika implementované metody ANFIS

Použití modelu ANFIS je podmíněno splněním několika skutečnostmi. Předně se musí jednat o model mající pouze jeden požadovaný výstup. Fuzzy model Takagi-Sugeno, jak již bylo zmíněno, musí být nultého nebo prvního řádu. Výstupní funkce příslušnosti tedy musí být konstantami anebo lineárními funkcemi, přičemž všechny musí být stejného druhu. Standardně se výstup kalkuluje formou lineární funkce. Dále specificky u ANFIS, na rozdíl od ostatních fuzzy modelů, není možné sdílení jednotlivých fuzzy pravidel. Znamená to, že více antecedentů pravidel (část pravidla před operátorem THEN) nemůže sdílet stejnou funkci příslušnosti na výstupu (konsekvent). Z toho vyplývá, že počet výstupních funkcí příslušnosti musí být stejný, jako počet pravidel. Navíc musí mít všechna fuzzy pravidla stejnou váhu (respektive prioritu). [21]

5.5.2 Funkce `thes_anfis.m`

Pro potřeby této diplomové práce je vytvořena v prostředí MATLAB funkce `thes_anfis.m`. Tato funkce přináší plno vylepšení usnadňující práci s implementací metody ANFIS, která je v původní verzi poměrně neintuitivní a je nutné provést plno komplikovaných nastavení.



Obr. 5.9 Algoritmus naprogramovaného modelu ANFIS

Popis bloků funkce:

- **Předpis funkce a nápověda** – definuje požadované vstupní argumenty a možné výstupy z funkce. Uživatel musí funkci poskytnout alespoň matici vstupů a vektor požadovaných výstupů určených pro trénování systému. Další vstupní argumenty jsou volitelné. Uživatel může dále poskytnout matici vstupů a požadovaných výstupů pro testování systému, stanovit počet funkcí příslušnosti a určit jejich typ pro jednotlivé vstupy systému, stanovit počet tréninkových epoch, a v neposlední řadě určit, zda mají být zobrazeny grafické výstupy.
- **Kontrola počtu vstupních argumentů** – stanovuje minimální počet vstupů do funkce, přičemž nepoužité vstupní argumenty jsou inicializovány na defaultní nastavení. Zároveň jsou v tomto bloku definovány dva výstupní argumenty, které uživatel musí použít. Jsou jimi vektor výstupů ze systému a struktura obsahující vypočtené statistické veličiny sloužící pro hodnocení kvality výpočetního modelu.
- **Kontrola vstupních dat** – kontrola rozměr vstupních matic, přičemž v případě nevyhovění jsou matice upraveny pro další zpracování. Dále je zde definováno omezení počtu fuzzy pravidel, které se vypočítá následovně:

$$N_{fr} = (N_{mf})^{N_{in}},$$

kde N_{fr} je počet fuzzy pravidel, N_{mf} je počet funkcí příslušnosti pro jednotlivé vstupní příznaky a N_{in} je počet vstupních příznaků. Defaultní omezení je 256 fuzzy pravidel, což znamená použití nejvýše 2 funkcí příslušnosti pro nejvýše 8 vstupních příznaků (anebo 3 funkcí příslušnost pro nejvýše 5 vstupních příznaků). Toto omezení je zavedeno z důvodu velmi vysoké výpočetní (a tím i časové) náročnosti trénování systému při použití více fuzzy pravidel.

- **Vytvoření inicializační struktury fis** – na základě vstupních argumentů je vytvořena struktura fis prostřednictvím příkazu `genfis1`. Tato struktura je poté během procesu

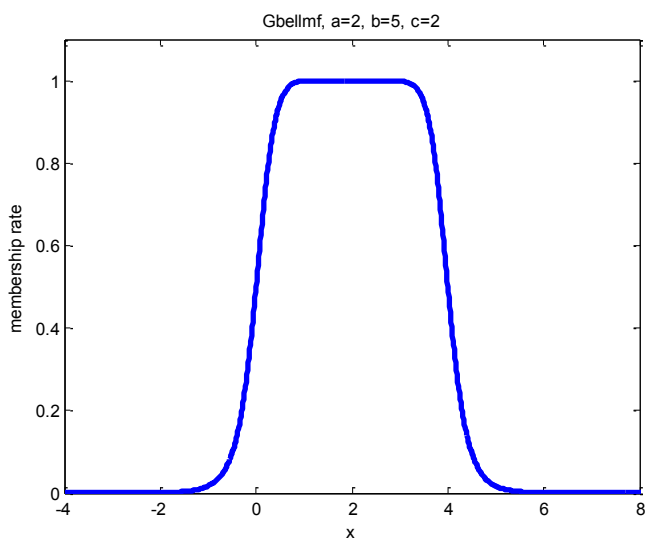
trénování adaptována tak, aby byla schopna mapovat vstupy na požadované výstupy. Defaultně je logická operace AND mezi proměnnými realizována operátorem Produkt (*Prod*). Lze ovšem zvolit také operátor Minimum (*Min*).

- **Trénování a testování modelu ANFIS** – zde je v uživatelsky stanoveném počtu epoch systém adaptován na konkrétní vstupy a požadované výstupy. Pro tento účel je použit hybridní algoritmus popsany v předchozí kapitole. Po procesu testování je uložena struktura *best_fis* představující nejefektivnější nastavení parametrů systému.
- **Výpočet výstupů systému** – realizováno vestavěným příkazem *evalfis*.
- **Výpočet statistických ukazatelů** – pro data určená pro testování jsou vypočítány statistické ukazatele MSE, MAE a koeficient korelace pro hodnocení kvality regresního modelu.
- **Export výstupů a zobrazení grafů** – dosažené výsledky jsou přehledně uloženy do proměnných (typu *struct*). Z grafů je mimo jiné zobrazeno průběhy inicializačních a adaptovaných funkcí příslušnosti, průběh chybové funkce a další.

5.5.3 Základní typy funkcí příslušnosti

Funkce příslušnosti pro antecedenty mají defaultně zvonovitý průběh (příkaz *gbellmf*). Průběh je možné v rámci procesu trénování systému, ve fázi zpětného průchodu metodou Backpropagation (algoritmus Gradient Descent), adaptovat změnou třech parametrů: *a*, *b*, *c*. Matematicky je funkce vyjádřena takto:

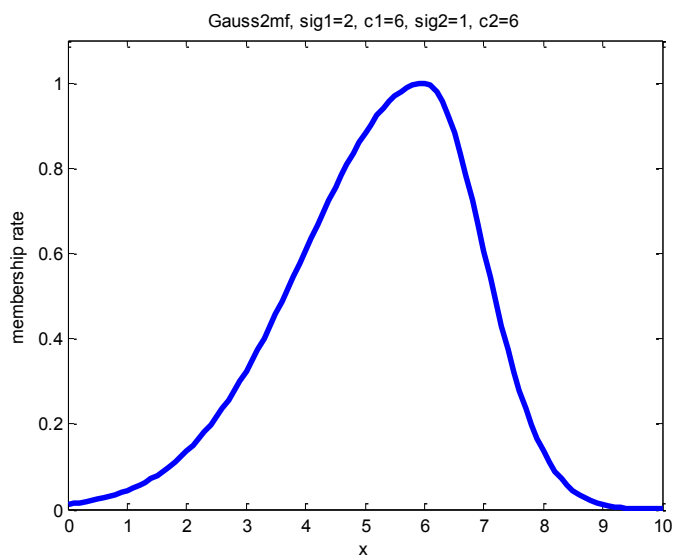
$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad (5.4)$$



Obr. 5.10 Příklad průběhu zvonovité funkce (*gbellmf*)

Během návrhu modelů pro stanovení závažnosti onemocnění Parkinsonovou chorobou se osvědčily také funkce příslušnosti mající průběh dvojité Gaussovy funkce (příkaz *gauss2mf*), přičemž adaptovány jsou čtyři parametry. Funkce je vyjádřena následovně:

$$\mu_A(x) = e^{\frac{-(x-c)^2}{2\sigma^2}} \quad (5.5)$$



Obr. 5.11 Příklad průběhu dvojité Gaussovy funkce (*gauss2mf*)

6. Kapitola

Testování modelu na poskytnutých datech „Parkinsons Telemonitoring Data Set“

Výpočetní modely pro stanovení míry závažnosti onemocnění Parkinsonovou chorobou, navržené v souladu s metodami uvedenými v předchozích kapitolách jsou realizovány a testovány na poskytnutém souboru dat měření oscilací řečového signálu, který byl vytvořen v rámci výzkumu na šesti amerických výzkumných lékařských centrech. Tento datový soubor je volně stažitelný v několika internetových uložiscích, například zde [27].

Datový soubor je rozdělen, jak je uvedeno v podkapitole 5.1.3, na dvě části. První část je určena pro trénování, validaci a testování výpočetních modelů. Druhá část slouží k realizaci nezávislého testu, přičemž se jedná o data, se kterými systém v průběhu tréninku nepřišel do styku. Na základě nezávislého testu je tedy ověřována schopnost generalizace na nová data, tedy budoucí měření oscilací řečového signálu pacientů postižených Parkinsonovou chorobou.

Na základě výstupů a statistických ukazatelů z nezávislého testu je určována kvalita výpočetního modelu. Statistické údaje z nezávislého testu jsou v rámci této kapitoly a příslušných příloh přehledně zobrazeny v tabulkách. Každý řádek tabulky reprezentuje jeden výpočetní model neuronové sítě nebo ANFIS, přičemž v prvních sloupcích jsou obsaženy vybrané tréninkové parametry modelu, poslední tři sloupce pak obsahují statistické údaje. Tréninkových parametrů použitých pro ladění sítě je samozřejmě celá řada. Nej kvalitnější modely, včetně veškerých parametrů, záznamů o průběhu tréninku a kompletního statistického vyhodnocení, jsou součástí elektronické přílohy.

V rámci testování byly vytvořeny dvě velké skupiny modelů. Prvním modelům byly předkládány jako požadované výstupy hodnoty Motor UPDRS, druhým pak byly předkládány hodnoty Total UPDRS. Rozdíl mezi těmito požadovanými výstupy je objasněn v podkapitole 3.1.3. Tyto požadované výstupy byly lékařsky stanoveny v nultém, třetím a šestém měsíci vyšetření pacientů, viz podkapitola 2.3.2. Vzhledem k tomu, že data měření řečového signálu byla zaznamenávána jednou týdně, jsou požadované výstupy Motor a Total UPDRS lineárně interpolovány tak, aby každému měření řečového signálu odpovídala jedna interpolovaná hodnota UPDRS.

6.1 Testování modelů neuronových sítí

Modely umělých neuronových sítí byly navrženy v prostředí MATLAB Neural Network Toolbox™ (NNT) a v programu Rapidminer 5. V obou SW byla každá architektura neuronové sítě trénována vícenásobně prostřednictvím křížové validace. Je to z důvodu náhodných inicializací synaptických vah a prahů, takže model s naprosto stejnými tréninkovými parametry může dosahovat zcela odlišných výsledků, viz podkapitola 5.3. Přestože jsou vytvořené skripty a funkce připraveny pro desetinásobnou křížovou validaci, v průběhu testování byly z časových důvodů modely trénovány pouze trojnásobně či pětinasobně. Desetinásobnou křížovou validací prošly jen nejkvalitnější výsledné modely. Aktivačními funkcemi neuronů (popř. v podkapitole 5.3.6), jsou buď logaritmické sigmoidy (*logsig*), hyperbolické tangenty (*tansig*), radiální báze (*radbas*) či jejich vzájemné kombinace. Co se týče počtu neuronů ve skrytých vrstvách, v rámci hledání nejoptimálnějších modelů byly tyto počty iterativně měněny v rozsahu $8 \div 96$ v první skryté vrstvě neuronů. Pokud byly použity druhé skryté vrstvy neuronů, jejich počet byl vzhledem k počtu neuronů v první vrstvě v poměru $1/2$, $1/3$ a $1/4$.

Každá kombinace tréninkových parametrů (mimo jiné aktivačních funkcí) tak byla trénována 48x (tedy s různými kombinacemi počtů neuronů ve skrytých vrstvách). Navíc každá z těchto iterací prošla trojnásobnou či pětinasobnou křížovou validací. Aby bylo možné realizovat tak rozsáhlý počet simulací, bylo použito paralelních výpočtů na počítačích s mnoha procesory, potažmo jádry procesorů. Z toho důvodu byl použit Parallel Computing Toolbox™, viz podkapitola 5.2. V rámci této diplomové práce tak bylo celkově nasimulováno přes 70 000 neuronových sítí. Některé simulace trvaly řádově sekundy, avšak trénink náročnějších a rozsáhlejších neuronových sítí mnohdy zabral mnohem více času.

Nebyla realizována simulace neuronových sítí s tréninkovým algoritmem Bayesian Regularization, viz podkapitola 4.1.4, z důvodu omezeného mechanismu pro ukončení tréninku (simulace by bylo nutné sledovat a manuálně zastavit v případě dosažení minima chybové funkce), a dále neuronové sítě Kvazi-Newtonova BPG (zde byly simulace časově velmi náročné z důvodu aproximace složitých matic druhých parciálních derivací chybové funkce v závislosti na hodnotách synaptických vah a prahů neuronů).

Výpočetní modely v následujících tabulkách (Tab. 6.1 a Tab. 6.2) jsou nejkvalitnějšími zástupci daných tréninkových algoritmů. Podrobnější výstupy z každého tréninkového algoritmu jsou obsaženy v příloze této práce, a dále v elektronické příloze.

6.1.1 Mapování vstupů na hodnoty Motor UPDRS

Hodnoty na lékařské stupnici UPDRS při hodnocení motoriky pacientů s Parkinsonovou chorobou (označovány Motor UPDRS) jsou obecně menší, než celkové hodnocení pacientů (označovány Total UPDRS). Je to dáno tím, že hodnoty Motor UPDRS jsou podmnožinou hodnot Total UPDRS. S touto skutečností souvisí fakt, že statistické ukazatele střední

kvadratické chyby (MSE) a střední absolutní chyby (MAE) obecně nabývají u simulací s požadovanými hodnotami Motor UPDRS výrazně menší hodnoty v porovnání se simulacemi s Total UPDRS. Korelační koeficient vyjadřující vzájemný vztah mezi výstupy z neuronové sítě a požadovanými výstupy (tj. klinicky zjištěnými lineárně interpolovanými výsledky lékařských vyšetření) nabývá u nejkvalitnějších sítí hodnoty přibližně 0,60 (přičemž maximum je 1,00).

Trénink prakticky všech nejkvalitnějších modelů neuronových sítí byl ukončen metodou *early stopping* (aneb *validation stop*, viz podkapitola 5.3.1), což značí, že s velkou pravděpodobností bylo opravdu nalezeno globální minimum chybové funkce.

Tab. 6.1 Statistická analýza výstupů neuronových sítí – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgb	80 /27	radbas /radbas	152	Validation stop.	24	47,397	5,606	0,573
traincgf	48 /16	radbas /radbas	178	Validation stop.	17	47,719	5,655	0,571
traincgp	48 /12	radbas /logsig	217	Validation stop.	22	48,142	5,666	0,564
trainscg	80 /27	radbas /logsig	222	Validation stop.	16	47,625	5,648	0,571
trainrp	96 /24	tansig /logsig	230	Validation stop.	11	46,548	5,553	0,584
traingda	40 /14	radbas /tansig	93	Validation stop.	2	63,469	6,763	0,331
traingdm	40 /14	radbas /logsig	241	Validation stop.	5	58,496	6,411	0,417
traingdx	80 /20	radbas /radbas	108	Validation stop.	4	56,559	6,232	0,448
trainoss	80 /40	radbas /radbas	156	Validation stop.	38	49,830	5,830	0,542
trainlm	16 /6	logsig /logsig	58	Validation stop.	18	44,689	5,326	0,611
cascade trainlm	32	tansig	54	Validation stop.	29	46,228	5,489	0,593
rapidminer BPG	24 /8	logsig /logsig	500	Max. epoch	58	48,784	5,678	0,565

Pokud není v tabulce explicitně uveden výraz „cascade“, jedná se o neuronové sítě s dopředným šířením signálu (podkapitola 5.3.2). Ve sloupci Neurons # vyjadřuje počty v první a případně ve

druhé skryté vrstvě. Stejně jsou znázorněny použité aktivační funkce ve sloupci Transfer functions. Tři nejlepší modely neuronových sítí jsou zvýrazněny tučným písmem.

Z tabulky je patrné, že nejkvalitněji vstupní data s 16 příznaky mapuje na požadované výstupy Motor UPDRS algoritmus Levenberg-Marquardt (*trainlm*) v klasické i kaskádové topologii neuronové sítě (podkapitola 5.3.3). Dobré výsledky rovněž vykazuje algoritmus Resilient BPG (*trainrp*).

Z předchozí tabulky i z Tab. VIII.10 v příloze vyplývá, že algoritmus Levenberg-Marquardt nevyžaduje ke kvalitnímu mapování velký počet neuronů (nejlepší model má 16 neuronů v první skryté vrstvě a 6 ve druhé). Naopak algoritmus Resilient BPG a taky algoritmy konjugovaného gradientu (první čtyři řádky Tab. 6.1) vyžadují použití podstatně vyššího počtu neuronů. Co se týče výstupů modelů optimalizovaných metodami variabilní míry rychlosti učení a momenta (algoritmy *traingda*, *traingdm* a *traingdx*), na těchto konkrétních nelineárních datech měření oscilací řečového signálu nejsou schopny kvalitně mapovat vstupy na výstupy. Avšak tyto modely nejrychleji konvergují ke globálnímu minimu chybové funkce.

Z tabulky Tab. 6.1 je rovněž evidentní, že zatímco algoritmy Levenberg-Marquardt jsou nejefektivnější při použití aktivačních funkcí typu *logsig* a *tansig*, modely trénované jinými optimalizačními algoritmy jsou nejefektivnější při použití aktivační funkce *radbas*.

Grafické výstupy (regresní analýza a hodnoty MAE) neuronové sítě s algoritmem Levenberg-Marquardt jsou zobrazeny v Obr. X.1 a Obr. X.2.

Model neuronové sítě navržený v programu Rapidminer 5 vykazuje průměrné výsledky. Jeho nevýhodou je ovšem doba tréninku, která je výrazně vyšší, než u ostatních neuronových sítí navržených v prostředí MATLAB NNT. Jedná se však o dostatečně kvalitní alternativu k NNT při mapování vstupů na výstupy Motor UPDRS.

6.1.2 Mapování vstupů na hodnoty Total UPDRS

Celkové klinické hodnocení závažnosti onemocnění Parkinsonovou chorobou (označováno Total UPDRS) sestává z více testů různého druhu, které musí pacienti podstoupit. Toto hodnocení je tedy komplexnější, čímž se zvyšuje pravděpodobnost, že vyšetřující lékař (hodnotící pacienta subjektivně) udělá při hodnocení chybu. Tato skutečnost se projevuje i na výstupech modelů neuronových sítí, jejichž statistické ukazatele MSE a MAE nabývají vyšších hodnot, než při mapování na hodnoty Motor UPDRS zhodnocení v předchozí podkapitole.

Naopak korelační koeficient je u mapování na hodnoty Total UPDRS prakticky u všech odpovídajících modelů vyšší, než při mapování na Motor UPDRS.

Také zde byl trénink téměř všech neuronových sítí ukončen metodou *early stopping*, což opět indikuje pravděpodobné nalezení globálního minima chybové funkce.

Pro tuto tabulku platí stejný popis, jako ten umístěný pod Tab. 6.1, tedy například, že tři nejlepší zástupci modelů neuronových sítí jsou zvýrazněni tučným písmem.

V případě mapování vstupů na hodnoty Total UPDRS je nejlepší model optimalizovaný algoritmem Levenberg-Marquardt (opět v klasické i v kaskádové). Dále dobré výsledky vykazuje model optimalizovaný metodou konjugovaného gradientu (*traincgb*).

Zmíněný model neuronové sítě s algoritmem Levenberg-Marquardt má naprosto stejnou architekturu (počty neuronů ve skrytých vrstvách) a tréninkové parametry (například aktivační funkce neuronů), jako předchozí model pro mapování hodnot Motor UPDRS, což svědčí o možnosti kvalitní univerzální implementace pro lékařské účely.

Z Tab. IX.10 (v příloze) opět vyplývá, že tento algoritmus nevyžaduje mnoho neuronů ve skrytých vrstvách. Doba konvergence je zde vyšší, než v případě mapování na Motor UPDRS.

Tab. 6.2 Statistická analýza výstupů neuronových sítí – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgb	88 /22	radbas /tansig	198	Validation stop.	33	78,115	7,077	0,590
traincgf	88 /44	radbas /radbas	156	Validation stop.	30	79,717	7,193	0,580
traincgp	32 /16	radbas /tansig	260	Validation stop.	18	78,710	7,121	0,586
trainscg	48 /12	radbas /radbas	156	Validation stop.	6	79,182	7,089	0,583
trainrp	32 /8	logsig /logsig	294	Validation stop.	5	78,827	7,099	0,585
traingda	24 /6	radbas /logsig	74	Validation stop.	1	105,86	8,550	0,360
traingdm	24 /6	logsig /radbas	144	Validation stop.	2	103,15	8,287	0,378
traingdx	64 /32	radbas /radbas	158	Validation stop.	6	94,520	8,043	0,462
trainoss	96 /48	radbas /radbas	216	Validation stop.	60	81,368	7,277	0,567
trainlm	16 /6	logsig /logsig	62	Validation stop.	22	72,709	6,683	0,631
cascade trainlm	88	tansig	58	Validation stop.	135	75,577	6,869	0,612
rapidminer BPG	24 /8	logsig /logsig	500	Max. epoch	58	123,07	8,746	0,232

Naopak u kaskádové neuronové sítě s algoritmem Levenberg-Marquardt (předposlední řádek Tab. 6.2) je nutné pro mapování na hodnoty Total UPDRS použití výrazně většího počtu neuronů, s čímž souvisí i doba konvergence a samozřejmě mnohem větší výpočetní náročnost. U zmíněného modelu neuronové sítě optimalizované metodou konjugovaného gradientu (*traincgb*) je také využito velkého počtu neuronů (88 v první skryté vrstvě a 22 ve druhé), avšak zde nejsou výpočetní nároky tak vysoké, protože systém v podstatě používá pouze šest vektorů obsahující hodnoty gradientu chybové funkce, viz podkapitola 4.1.4.

Algoritmy s adaptivní mírou rychlosti učení a momentem se při mapování na hodnoty Total UPDRS taktéž neosvědčily.

Model vytvořený v prostředí Rapidminer 5 (viz poslední řádek Tab. 6.2) v případě mapování na hodnoty Total UPDRS vykazoval velmi vysoké hodnoty MSE a MAE, a není tedy použitelný.

6.2 Testování modelů ANFIS

Druhou metodou využitou v této diplomové práci pro mapování vstupních dat měření oscilací řečového signálu na hodnoty lékařské stupnice UPDRS je metoda Adaptive Neuro Fuzzy Inference System (ANFIS), která je detailně popsána v podkapitolách 4.2 a 5.5.

Od neuronových sítí se liší v mnoha ohledech, nejdůležitějším je ovšem výpočetní náročnost, která v podstatě určuje dobu adaptace systému na předložená data. Zatímco drtivá většina simulací neuronových sítí v rámci této práce trvala řádově sekundy či minuty, trénink modelů ANFIS v současné podobě implementace trvá řádově hodiny. Je to způsobeno mimo jiné tím, že proces adaptace parametrů funkcí příslušnosti prozatím nelze paralelizovat.

Doba adaptace modelu navíc velmi výrazně závisí na počtu vstupních příznaků datového souboru a počtu funkcí příslušnosti. Definujeme-li 2 funkce příslušnosti pro každý vstup ANFIS, při použití 6 vstupů může trvat adaptace systému (při 40 tréninkových epochách) přibližně 10 minut, při použití 7 vstupů může adaptace trvat přibližně hodinu a půl a při použití 8 vstupů, což je maximální simulovaný počet vstupů v této práci, může adaptace jednoho výpočetního modelu trvat přes 17 hodin. Z tohoto důvodu proto nebyla použita metoda křížové validace, která by výpočetní čas ještě znásobila.

V případě použití všech 16 dostupných příznaků měřených dat, jako je tomu u neuronových sítí, by byla doba simulace modelu ANFIS enormní.

6.2.1 Výběr příznaků vstupních dat pro simulace modelů ANFIS

Z výše uvedených důvodů je nezbytné vybrat pro simulace pouze určité vstupní příznaky řečového signálu. Obecně byly zvoleny ty, které nejlépe reprezentují datový soubor, k čemuž

lze dojít na základě korelační analýzy, více zde [2]. Příznaky a jejich popis jsou uvedeny v části Příloha VII.

Při návrhu modelu ANFIS je použita jedna z následující konfigurací:

- 5 příznaků vstupních dat
 - 3 funkce příslušnosti pro každý vstup,
 - 243 fuzzy pravidel
 - vybrány příznaky *Shimmer*, *NHR*, *HNR*, *DFA* a *PPE*.
- 8 příznaků vstupních dat
 - 2 funkce příslušnosti pro každý vstup,
 - 256 fuzzy pravidel
 - vybrány příznaky *Jitter(abs)*, *Shimmer*, *Shimmer:DDA*, *NHR*, *HNR*, *RPDE*, *DFA* a *PPE*.

6.2.2 Mapování vstupů na hodnoty Motor UPDRS

Stejně jako u neuronových sítí při mapování vstupů na výstupy na stupnici Motor UPDRS nabývají u ANFIS statistické ukazatele MSE a MAE menších hodnot, než při mapování na výstupy Total UPDRS. Co se týče Tab. 6.3 a Tab. 6.4, sloupec „AND Operator“ představuje použitou metodu při návrhu fuzzy pravidel (viz podkapitola 5.5.2), sloupec „MF type“ pak obsahuje typ použité funkce příslušnosti, které jsou popsány v podkapitole 5.5.3.

Tab. 6.3 Statistická analýza modelu ANFIS – Motor UPDRS

Inputs #	Fuzzy rules #	AND Operator	MF type	Epochs #	Training time	MSE	MAE	Corr. coeff
8	256	prod	gbellmf	40	58355	53,297	5,903	0,504
8	256	min	gbellmf	40	55189	54,110	6,099	0,485
5	243	min	gbellmf	40	19956	58,658	6,153	0,436

Z Tab. 6.3 je patrné, že nejlepších výsledků dosahují modely ANFIS s 8 vstupy (s osmi vybranými vstupními příznaky) než modely s 5 vstupy. Toto lze očekávat v souvislosti s tím, že systému je předloženo větší množství dat, přestože při použití 2 funkcí příslušnosti pro každý z osmi vstupů je omezena citlivost na adaptaci parametrů. Zároveň je ovšem zjevné, že při použití osmi vstupů je doba adaptace systému přibližně trojnásobná. Také hodnoty MSE, MAE a korelačního koeficientu jsou při použití osmi vstupů výrazně lepší.

Ve srovnání s výstupy z neuronových sítí jsou ovšem výstupy z ANFIS horší. Je to dáno především tím, že neuronové sítě mohou zpracovávat mnohem větší množství dat (všech 16 dostupných příznaků datového souboru), a mohou se tak lépe naučit specifika datového souboru. Co se týče zvoleného typu funkcí příslušnosti, nejlepší výsledky vykazovaly modely, jejichž funkce příslušnosti byly zvonovitého tvaru, viz Obr. 5.10.

6.2.3 Mapování vstupů na hodnoty Total UPDRS

Při mapování vstupů na výstupy Total UPDRS je situace analogická. Nej kvalitnějšími modely jsou opět ty, které mají osm vstupů, přičemž je zároveň prodloužena doba adaptace. Nicméně zde není rozdíl mezi výstupy ze systémů z osmi vstupy a ze systémů z pěti vstupy tak markantní. Stejně jako u mapování na hodnoty Motor UPDRS mají nejlepší modely funkce příslušnosti zvonovitého tvaru.

Tab. 6.4 Statistická analýza výstupu modelu ANFIS – Total UPDRS

Inputs #	Fuzzy rules #	AND Operator	MF type	Epochs #	Training time	MSE	MAE	Corr. coeff
8	256	prod	gbellmf	40	54206	87,904	7,475	0,527
8	256	min	gbellmf	40	56337	89,486	7,700	0,506
5	243	min	gbellmf	40	19780	92,842	7,746	0,481

7. Kapitola

Zhodnocení funkčnosti modelu a jeho aplikace

Simulací výpočetních modelů pro mapování vstupních dat měření řečového signálu na hodnoty stupnice UPDRS bylo provedeno velké množství. Obecně jsou modely hodnoceny na základě statistických ukazatelů z nezávislých testů realizovaných po ukončení adaptace parametrů jednotlivých modelů, viz kapitola 6. V této kapitole jsou zhodnocena specifika jednotlivých metod, jejich použitelnost v praktických aplikacích a nastíněny možnosti dalšího výzkumu a zefektivnění implementace algoritmů.

7.1 Specifika funkčnosti výpočetních modelů

V rámci testování systému bylo vyvinuto mnoho výpočetních modelů. Je nutné si uvědomit skutečnost, že ačkoliv při letném pohledu na tabulky nejkvalitnějších modelů (Tab. 6.1, Tab. 6.2, Tab. 6.3 a Tab. 6.4) se zdá, že výsledky jsou velmi podobné, každý z těchto modelů byl navržen naprosto odlišnou metodou. Ať už se jedná o adaptaci synaptických vah a prahů neuronů, či o architekturu modelu (počty neuronů v jednotlivých skrytých vrstvách a způsob propojení vrstev neuronů), o druh aktivačních funkcí neuronů a v neposlední řadě také o specifické tréninkové parametry konkrétních tréninkových algoritmů. Metoda ANFIS je navíc založena na zcela jiném principu. Přes tato fakta je evidentní, že chyby výstupů (hodnot UPDRS) se pohybují přibližně ve stejných rozsazích, z čehož vyplývá, že datový soubor obsahuje značné množství nelinearit.

Nejlepšími algoritmy pro mapování na hodnoty Motor UPDRS jsou Levenberg-Marquardt a Resilient Backpropagation. Nejlepšími algoritmy pro mapování na hodnoty Total UPDRS jsou Levenberg-Marquardt a CGB s Powell-Bealeovými restarty. Algoritmus Levenberg-Marquardt se prokázal jako nejkvalitnější, avšak zároveň mající velmi velké výpočetní nároky.

Obecně všechny modely se vyznačují tím, že hodnoty MSE jsou poměrně vysoké. Implikuje to skutečnost, že datový soubor (respektive jednotlivé příznaky) obsahuje mnoho odlehlých měření, která zanášejí do zpracování šum, který je obtížné odstranit. Řešením by bylo zkompletovat větší datový soubor (větší počet měření řečového signálu), na základě kterého by bylo možné výpočetní modely lépe adaptovat.

Variantou k návrhu neuronových sítí v prostředí MATLAB je software Rapidminer 5. Jak je vidět z Tab. 6.1, pro mapování vstupů na hodnoty Motor UPDRS jsou modely z Rapidminer v rámci mezí postačující. Je nutné uvést, že z důvodu komplikované implementace iterativních

změn parametrů neuronových sítí bylo simulováno malé množství modelů. Je pravděpodobné, že pokud by byly realizovány simulace výrazně většího množství neuronových sítí, byly by nalezeny mnohem kvalitnější výpočetní modely pro zpracování řečového signálu.

Metoda ANFIS se také osvědčila vzhledem ke skutečnosti, že pro trénink byla použita pouze polovina poskytnutého datového souboru (pouze 8 vstupů z 16 možných). I přesto výsledky nejsou výrazně horší a lze předpokládat, že v případě použití všech vstupů by byly výstupy lepší, než v případě neuronových sítí.

7.2 Srovnání s rešeršními výsledky

Srovnáním výsledků této diplomové práce s výsledky publikovanými mimo jiné v [2], [5] a [6] lze posoudit kvalitu výpočetních modelů v obecnější rovině z toho důvodu, že v těchto publikacích byly použity jiné výpočetní a statistické metody. Autoři obvykle nepoužívali pokaždé všech 16 příznaků řečového signálu, ale na základě korelační analýzy volili ty příznaky, které mají s hodnotami požadovanými UPDRS největší korelační závislost, viz Tab. 2.1. Autoři práce [6] ovšem použili v jednom z testů všech 16 příznaků řečového signálu, takže výstupy lze přímo porovnat s výstupy této práce, viz Tab. 2.2.

Ze srovnání nejlepších výpočetních modelů uvedených v Tab. 6.1 a Tab. 6.2 s modely popsány v Tab. 2.1 a Tab. 2.2 vyplývá, že pouze jeden model z jiných publikací dosahuje lepších výsledků. Jedná se o model založený na metodě *Least Square Support Vector Machine* (LS-SVM), jehož vstupem je zmíněných 16 příznaků řečového signálu. Rozdíl mezi tímto modelem a modelem neuronové sítě adaptované prostřednictvím algoritmu Levenberg-Marquardt (s 16 neurony v první skryté vrstvě a 6 ve druhé skryté vrstvě) ovšem není velmi výrazný a nejvíce se projevuje v hodnotě MSE. Je tomu stejně v případě mapování na hodnoty Motor i Total UPDRS.

Konkurenceschopné jsou kromě neuronové sítě s algoritmem Levenberg-Marquardt také neuronové sítě s algoritmem Resilient BPG a s algoritmem CGB s Powell-Bealeovými restarty, které jsou ovšem vhodnější pro implementaci ve vestavěných zařízeních, viz následující kapitola. Tyto dvě neuronové sítě konkurují (s výjimkou zmíněného LS-SVM) ve většině statistických ukazatelů všem ostatním výpočetním modelům popsáným v ostatních publikacích, v některých charakteristikách je i překonávají.

Otázkou ovšem zůstává, zdali autoři v jiných publikacích také vyčlenili data měření řečového signálu určená explicitně pro nezávislý test, nebo zdali pouze zprůměrovali výsledky z křížové validace. Pravdou je, že v případě průměrování výstupů křížové validace v této diplomové práci dosahovaly výpočetní modely kvalitnějších výsledků, než v případě následného nezávislého testu.

7.3 Využitelnost v praktických aplikacích

Obecnou podmínkou pro použití vyvinutých modelů na budoucí data je fakt, že tato klinická data musí být získána stejným způsobem, tedy stejným měřicím postupem a musí být stejným způsobem zpracována (pre-processing) před jejich přeložením výpočetním systémům. Jedná se o to, že modely neuronových sítí i ANFIS jsou adaptovány na konkrétní počty vstupních příznaků, na konkrétní rozsahy hodnot těchto příznaků atp. Aby byl systém schopen efektivně mapovat měřená data na výstupy na lékařské stupnici UPDRS, musí být tyto podmínky dodrženy.

Co se týče realizace systémů pro stanovení závažnosti onemocnění Parkinsonovou chorobou, nabízí se dvě varianty. První by byla varianta čistě softwarová. Tato by byla využitelná v nemocnicích ve formě standardní desktopové aplikace, se kterou by pracoval lékař specializující se na neurodegenerativní choroby, který by výstupy ze SW porovnával se svým subjektivním hodnocením získaným na základě vyšetření pacienta. Data získaná měřením z domu by byla přeposílána Internetem stejně jako doposud, přičemž vyhodnocení by probíhalo přímo u lékaře. V tomto případě by bylo nutné aplikaci převést z prostředí MATLAB (který není standardním SW vybavením nemocnic) například kompilací do aplikace s příponou .exe, popřípadě model implementovat v jiném programovacím jazyce, například v C#. Pro tyto účely se jako nejvhodnější řešení nabízí použití neuronové sítě s tréninkovým algoritmem Levenberg-Marquardt, který je na základě testování systémů uvedeného v předchozí kapitole nejvýkonnějším modelem vykazujícím nejlepší výsledky. Je to z toho důvodu, že desktopové systémy v podstatě nemají taková výpočetní omezení, jako vestavěné systémy. Velký výpočetní výkon je u algoritmu Levenberg-Marquardt vyžadován kvůli tomu, že výpočet Jacobiho matice prvních parciálních derivací a Marquadtova parametru je velmi náročný, viz podkapitola 4.1.4. Možnost adaptace systému prostřednictvím nového tréninku by bylo možné s výhodou použít při aktualizaci datového souboru (nová měření řečového signálu, nový způsob měření apod.).

Druhou variantou je vývoj aplikace ve formě vestavěného systému. Ten by mohl mít pacient doma, přičemž vyhodnocení by probíhalo bezprostředně po ukončení měření řečového signálu a výsledek by byl lékařům přeposílán společně s měřenými daty přes Internet. Zde by pravděpodobně bylo nezbytné pro případnou adaptaci systému použít jiný tréninkový algoritmus, tedy takový, který má výrazně nižší výpočetní nároky. Tímto algoritmem by mohl být například Resilient BPG, který minimalizuje chybovou funkci pouze pomocí znaménka derivace (gradientu), rychle konverguje a má malé výpočetní nároky. Nebo by bylo možné implementovat algoritmus CGB s Powell-Bealeovými restarty, který využívá šest vektorů gradientů a směru poklesu. Pokud by možnost adaptace systému nebyla nutná, stačilo by pouze implementovat hotový výpočetní model, tedy již vypočtené synaptické váhy a prahy neuronů a správné propojení vrstev (nebylo by tak nutné implementovat tréninkový algoritmus), čímž by došlo k značnému zjednodušení vývoje vestavěného systému.

7.4 Možnosti dalšího výzkumu

Dvěma základními faktory, ve kterých lze spatřit možnosti zlepšení výkonu výpočetních modelů, jsou zmenšení chyb systému a rychlost konvergence při minimalizaci chybové funkce (tedy rychlost adaptace systému na nové vstupy).

První faktor se týká především návrhu neuronových sítí. Pro efektivnější návrh by bylo vhodné mít k dispozici výrazně větší množství měřených dat řečového signálu, pokud možno s co nejmenším počtem odlehlých měření. Modely by tak byly schopny naučit se více specifik datového souboru (především v oblasti rozsahu jednotlivých příznaků), čímž by došlo k minimalizaci chyb na výstupu (hodnoty Motor a Total UPDRS). V současné podobě jsou hodnoty MSE poměrně vysoké, což indikuje, že systém stále produkuje mnoho výstupů, které jsou daleko od požadovaných hodnot. V případě většího počtu měření by bylo možné tento nedostatek omezit.

Rychlost konvergence při minimalizaci chybové funkce je problémem především současné podoby implementace metody ANFIS. I přesto, že nejkvalitnější modely ANFIS používají pouze 8 vstupů (z dostupných 16), doba adaptace (tréninku) systému je v řádu hodin. Je to z důvodu prozatím nemožnosti paralelizace procesu adaptace parametrů funkcí příslušnosti. Kdybychom chtěli použít všech 16 vstupních příznaků datového souboru, doba adaptace by mohla činit řádově stovky či tisíce hodin, což je pro standardní aplikace nepřijatelné. Jakmile dojde k paralelizaci výpočtů parametrů ANFIS, bude možné použít více vstupních příznaků, čímž dojde k výraznému zpřesnění výstupů, protože systému bude v rámci tréninku předloženo výrazně větší množství dat. Zároveň by bylo vhodné pro každý vstup využít více funkcí příslušnosti (v současné podobě jsou používány 2), v důsledku čehož by byla zvýšena citlivost systému při adaptaci. Tímto se výpočetní model naučí mnohem více specifik datového souboru měření řečového signálu a bude schopen efektivněji mapovat vstupy na hodnoty UPDRS.

8. Závěr

Ve své diplomové práci jsem se zabýval možnostmi stanovení vývoje Parkinsonovy choroby založenými především na modelech umělých neuronových sítí a modelu Adaptive Neuro Fuzzy Inference System (ANFIS) vycházejícího z fuzzy logiky. Dostupná data měření řečového signálu pacientů s Parkinsonovou chorobou jsou prostřednictvím výpočetních modelů mapována na hodnoty na klinické stupnici UPDRS, na základě které je hodnocen vývoj onemocnění u pacientů.

V průběhu zpracování této práce byly realizovány simulace přibližně 70 000 modelů neuronových sítí. Toho bylo dosaženo paralelizací výpočtů během adaptace parametrů neuronových sítí s použitím výpočetních systémů s mnoha procesory. Za nejkvalitnější lze považovat model neuronové sítě, jehož parametry byly adaptovány algoritmem Levenberg-Marquardt, který ovšem vyžaduje velký výpočetní výkon. V případě potřeby implementace modelu do vestavěného systému by bylo pravděpodobně nutné zvolit modely adaptované méně náročným tréninkovým algoritmem, například algoritmem Resilient Backpropagation či metodami konjugovaného gradientu, které se vyznačují velkou rychlostí konvergence při minimalizaci chybové funkce.

Co se týče použití metody ANFIS, v současné podobě implementace má hodně omezení, například velmi dlouhá doba adaptace systému, což znemožňuje použití všech dostupných dat měření řečového signálu. Z toho důvodu vykazuje větší chyby na výstupech, protože není schopen naučit se všechna specifika datového souboru obsahujícího velké množství nelinearit. Myslím si však, že v případě implementace paralelní adaptace parametrů systému ANFIS by došlo k výraznému zvýšení výkonu, a bylo by tak možné překonat neuronové sítě. Toto může být podnětem k dalšímu výzkumu.

Srovnáním výsledků této diplomové práce a výsledků autorů publikací zabývajících se podobnou problematikou lze dojít k závěru, že pouze jedna výpočetní metoda z těchto publikací dosahuje kvalitnějších výsledků, než výše popsaná neuronová síť navržená v rámci této práce. Toto lze považovat za úspěch, protože jiných použitých metod je velké množství a je zřejmé, že výsledky této práce jim mohou směle konkurovat, v určitých charakteristikách také překonat.

Obecně jsou výpočetní modely v této práci schopny s poměrně velkou přesností diagnostikovat míru závažnosti onemocnění Parkinsonovou chorobou prostřednictvím mapování na hodnotu klinické stupnice UPDRS. Pacienti by tak nebyli nuceni podstupovat časově a finančně náročné cesty do lékařských zařízení a průběžná diagnostika Parkinsonovy choroby by byla prováděna z pohodlí jejich domova. Nicméně žádný výpočetní systém nemůže zcela nahradit diagnostický posudek lékaře specializujícího se na Parkinsonovu chorobu.

Literatura a použité zdroje

- [1] ROTH, J., RŮŽIČKA, E., KAŇOVSKÝ, P. *Parkinsonova nemoc a parkinsonské syndromy*. 1. vyd. Praha: Galén, 2000, 293 s. ISBN 80-7262-048-7.
- [2] TSANAS, A., LITTLE, M.A., McSHARRY, P.E., RAMIG, L.O. *Accurate Telemonitoring of Parkinson's Disease Progression by Noninvasive Speech Tests*. Biomedical Engineering, IEEE Transactions on, vol.57, no.4, pp.884,893, April 2010.
- [3] GOETZ, C.G., STEBBINS, G.T., WOLFF, D., DeLEEuw, W., BRONTE-STEWART, H., ELBLE, R. *Testing objective measures of motor impairment in early Parkinson's disease: Feasibility study of an at-home testing device*. Movement Disord., vol. 24, no. 4, pp. 551–556, 2009.
- [4] TSANAS, Athanasios. *Accurate telemonitoring of Parkinson's disease symptom severity using nonlinear speech signal processing and statistical machine learning*. [dissertation]. Oxford, UK: University of Oxford, St. Cross College, June 2012.
- [5] TSANAS, A., LITTLE, M.A., McSHARRY, P.E., RAMIG, L.O. *Enhanced classical dysphonia measures and sparse regression for telemonitoring of Parkinson's disease progression*. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pp.594,597, 14-19 March 2010..
- [6] ESKIDERE, Ö., ERTAŞ, F., HANILÇI, C. *A comparison of regression methods for remote tracking of Parkinson's disease progression*. Expert Systems with Applications: An International Journal, vol.39 no.5, pp.5523-5528, April, 2012.
- [7] ROTH, J., SEKYROVÁ, M. a RŮŽIČKA, E., *Parkinsonova nemoc*, 2. rozšířené vyd. Praha: Maxdorf, 1999, 142 s. ISBN 80-85800-63-2.
- [8] *EPDA* [online]. 2013 [cit. 2013-11-14]. Unified Parkinson's Disease Rating Scale. Dostupné z WWW:
<http://www.epda.eu.com/en/parkinsons/in-depth/parkinsonsdisease/rating-scales/updrs/>.
- [9] *International Parkinson and Movement Disorder Society* [online]. 2010 [cit. 2013-11-14]. MDS-UPDRS. Dostupné z WWW:
<http://www.movementdisorders.org/UserFiles/New%20UPDRS%207%203%2008%20final.pdf>.

- [10] VLADYKA, Vilibald. *Jak čelit Parkinsonově nemoci*. 1. vyd. Praha: Avicenum, 1977, 114 s.
- [11] TUČKOVÁ, Jana. *Vybrané aplikace umělých neuronových sítí při zpracování signálů*. Vyd. 1. Praha: České vysoké učení technické v Praze, 2009, 224 s. ISBN 978-80-01-04229-8.
- [12] VONDRÁK, Ivo. *Umělá inteligence a neuronové sítě*. 2. vyd. Ostrava: Vysoká škola báňská - Technická univerzita, 2002, 139 s. ISBN 80-7078-949-2.
- [13] HAGAN, M.T., DEMUTH, H.B., BEALE, M.H. *Neural Network Design*. 1. vyd. Stillwater: Oklahoma State University, 2002, 726 s. ISBN 0-9717321-0-8.
- [14] JIRSÍK, V., HRÁČEK, P. *Neuronové sítě, expertní systémy a rozpoznávání řeči*. Studijní skripta. Brno: VUT v Brně, Fakulta elektrotechniky a komunikačních technologií. 106 s.
- [15] *Wikipedie* [online]. 2008 [cit. 2013-10-29]. Model neuronu. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Soubor:NeuronModel.jpg>.
- [16] *Mathworks* [online]. 2014 [cit. 2014-01-12]. MATLAB Neural Network Toolbox™ User's Guide. Dostupné z WWW: http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf.
- [17] *Mathworks* [online]. 2014 [cit. 2014-01-12]. Resilient backpropagation. Dostupné z WWW: <http://www.mathworks.co.uk/help/nnet/ref/trainrp.html>.
- [18] *Mathworks* [online]. 2014 [cit. 2014-01-12]. One-step secant backpropagation. Dostupné z WWW: <http://www.mathworks.co.uk/help/nnet/ref/trainoss.html>.
- [19] POKORNÝ, M., SROVNAL, V. *Znalostní systémy řízení*. Studijní skripta. Ostrava: VŠB-TUO, Fakulta elektrotechniky a informatiky. 2012, 230 s.
- [20] BÍLA, Jiří. *Umělá inteligence a neuronové sítě v aplikacích*. 1. vyd. Praha: České vysoké učení technické, 1995, 115 s. ISBN 80-01-01275-1.
- [21] *Mathworks* [online]. 2014 [cit. 2014-01-29]. MATLAB Fuzzy Logic Toolbox™ User's Guide. Dostupné z WWW: http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf.
- [22] CRUZ, A.O. [online]. 2013 [cit. 2014-02-08]. ANFIS: Adaptive Neuro-Fuzzy Inference Systems. Dostupné z WWW: <http://equipe.nce.ufrj.br/adriano/fuzzy/transparencias/anfis/anfis.pdf>.

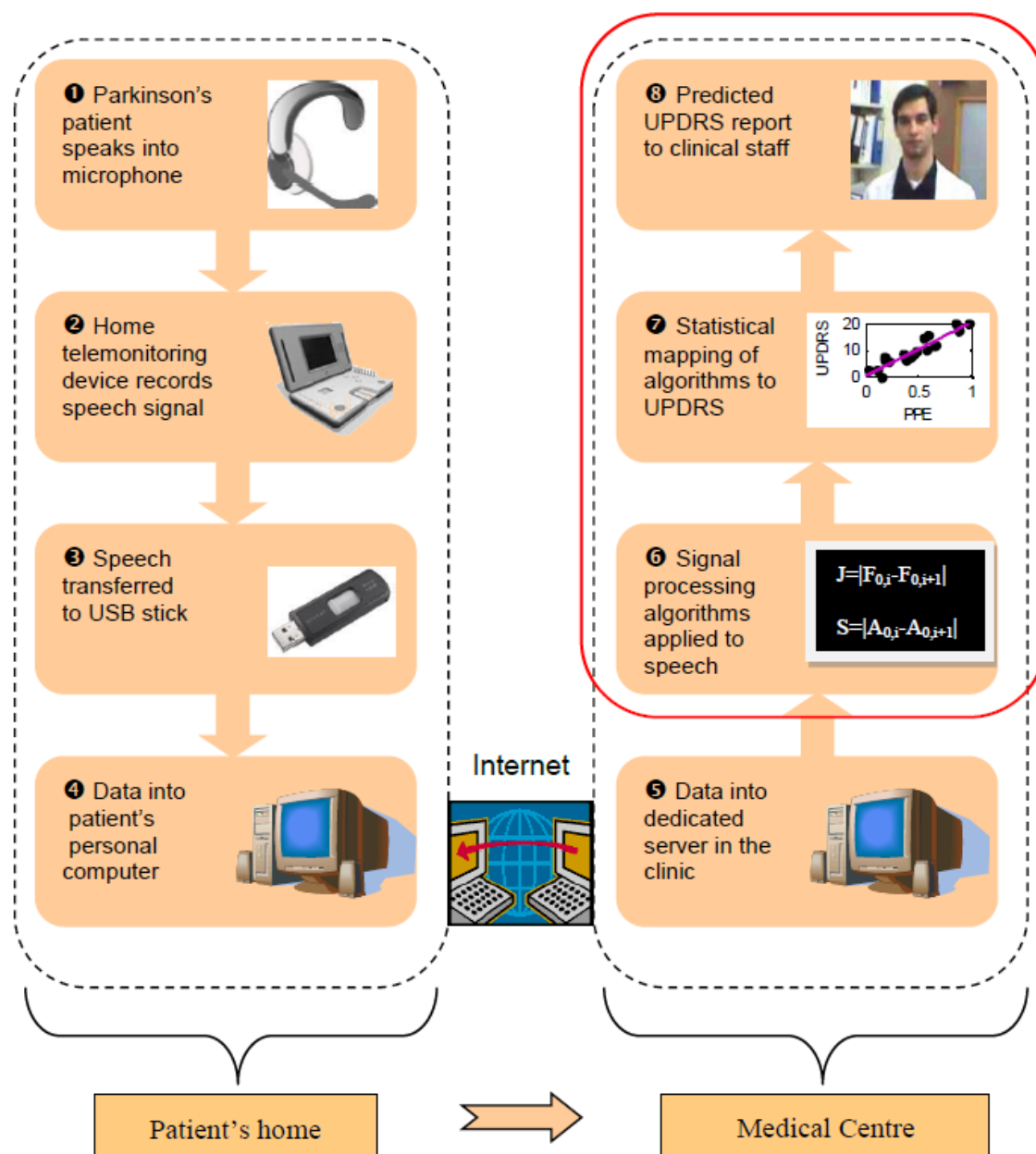
- [23] *KOIVO, H.* [online]. 2000 [cit. 2014-02-08]. ANFIS: Adaptive Neuro-Fuzzy Inference Systems. Dostupné z WWW:
<ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Control%20Inteligente/Parte%20III/clase%2023%20int/FVAnfis2.pdf>.
- [24] *Mathworks* [online]. 2014 [cit. 2014-01-26]. MATLAB Parallel Computing Toolbox™ User's Guide. Dostupné z WWW: http://www.mathworks.com/help/pdf_doc/distcomp/distcomp.pdf.
- [25] *IT4Innovations* [online]. 2014 [cit. 2014-01-26]. Running parallel Matlab using Distributed Computing Toolbox / Engine. Dostupné z WWW:
<https://support.it4i.cz/docs/anselm-cluster-documentation/software/numerical-languages/matlab>.
- [26] *Rapidminer 5* [online]. 2012 [cit. 2014-03-02]. Operator Reference. Dostupné z WWW:
http://1xltkxylmzx3z8gd647akcdvov.wpengine.netdna-cdn.com/wp-content/uploads/2013/10/RapidMiner_OperatorReference_en.pdf.
- [27] *UCI Machine Learning Repository* [online]. 2009 [cit. 2013-09-18]. Parkinsons Telemonitoring Data Set. Dostupné z WWW: <http://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring>.

Seznam příloh

Příloha I	Průběh experimentálního vyšetření.....	I
Příloha II	Statistická analýza vstupních dat	II
Příloha III	Histogramy požadovaných výstupů výpočetních modelů.....	IV
Příloha IV	Rozdělení vstupních dat pro metodu ANFIS	V
Příloha V	Příklad skriptu mpiLibConf.m (pro paralelní výpočty)	V
Příloha VI	Typy aktivační funkcí neuronů	VI
Příloha VII	Popis příznaků měřeného řečového signálu.....	VIII
Příloha VIII	Statistická analýza výsledků výpočetních modelů – Motor UPDRS	IX
Příloha IX	Statistická analýza výsledků výpočetních modelů – Total UPDRS.....	XII
Příloha X	Grafické výstupy – Levenberg-Marquardt – Motor UPDRS	XV

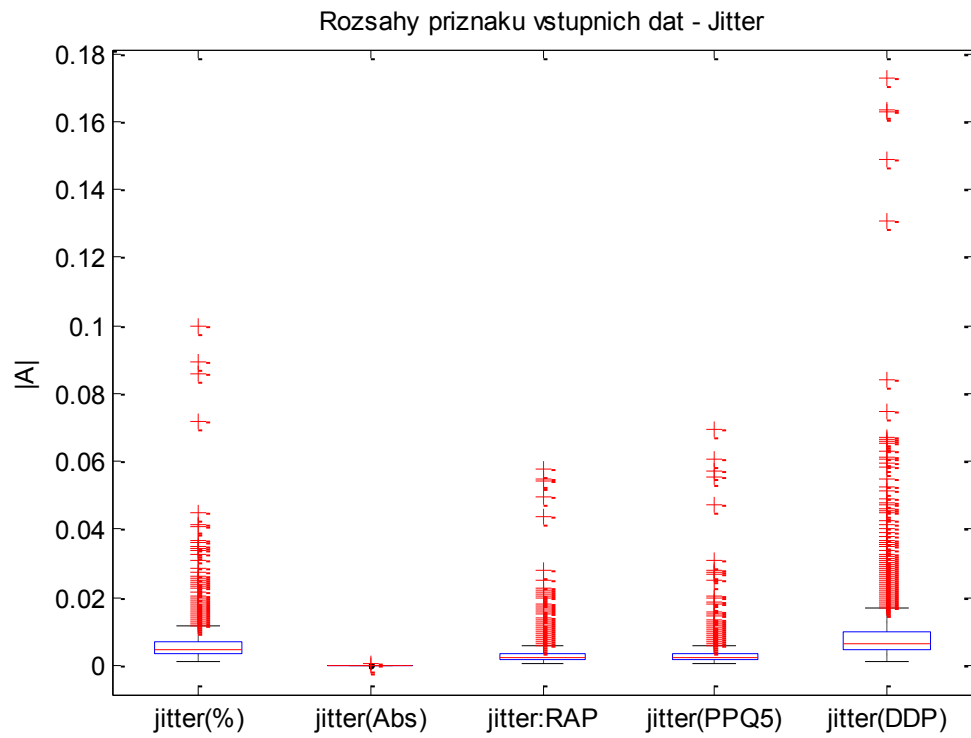
Přílohy

Příloha I Průběh experimentálního vyšetření

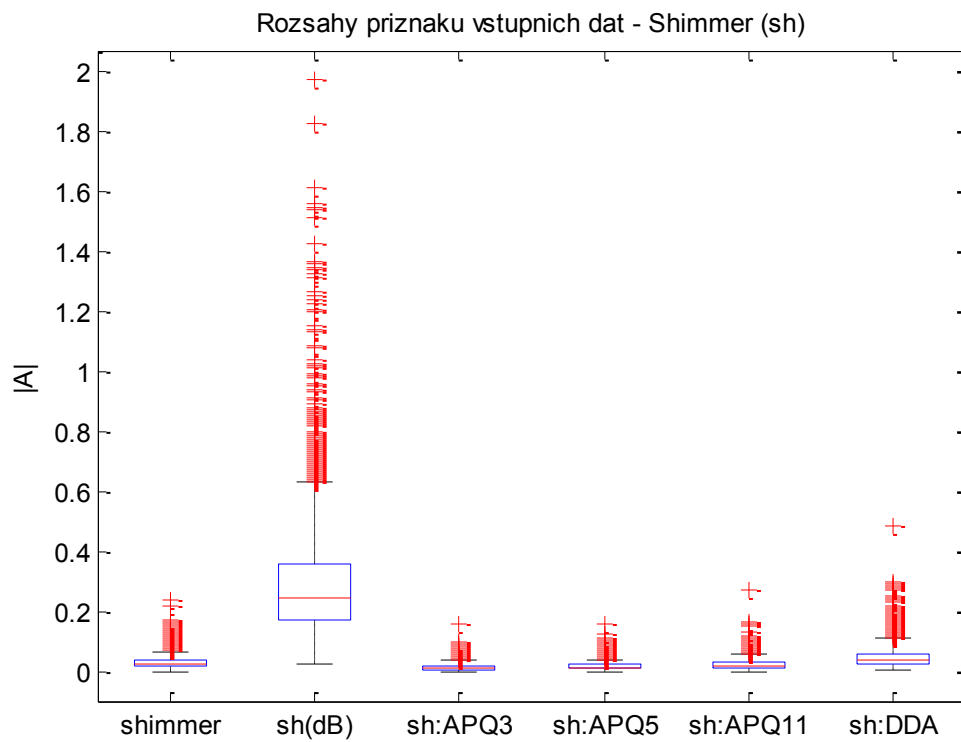


Obr. I.1 Experimentální hodnocení míry závažnosti onemocnění Parkinson. chorobou [4]

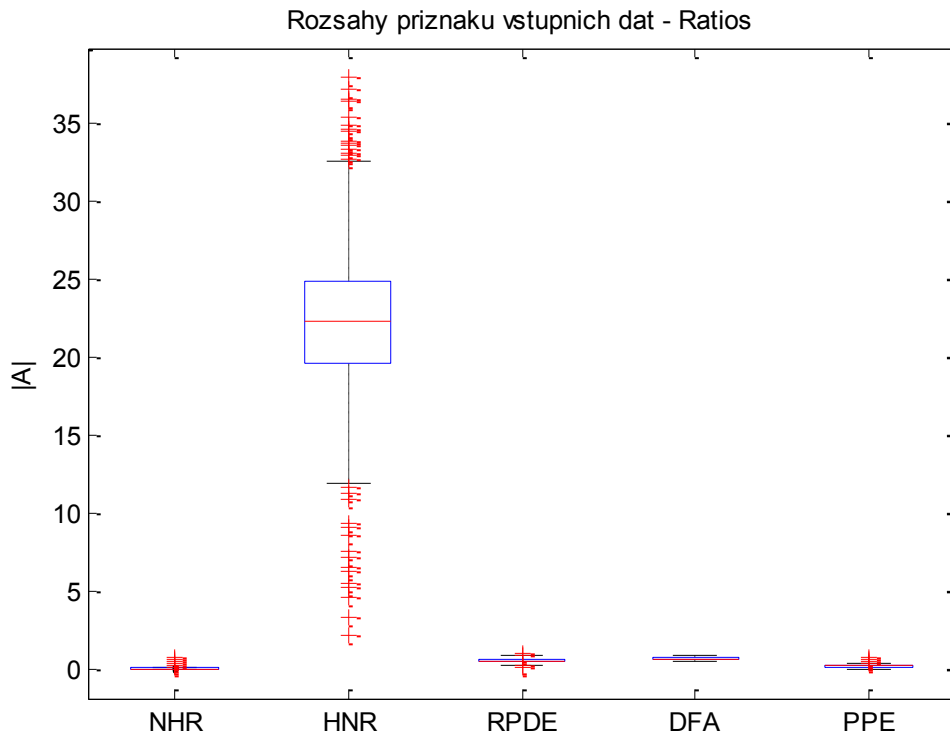
Příloha II Statistická analýza vstupních dat



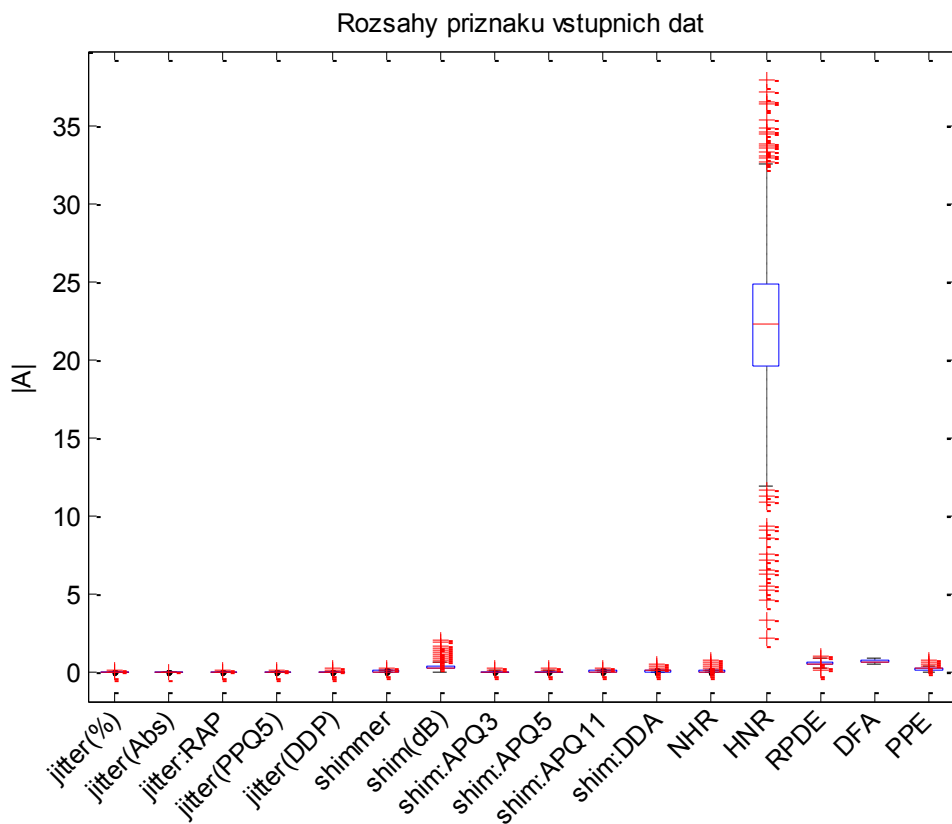
Obr. II.1 Rozsahy příznaků Jitter (vstupy)



Obr. II.2 Rozsahy příznaků Shimmer (vstupy)

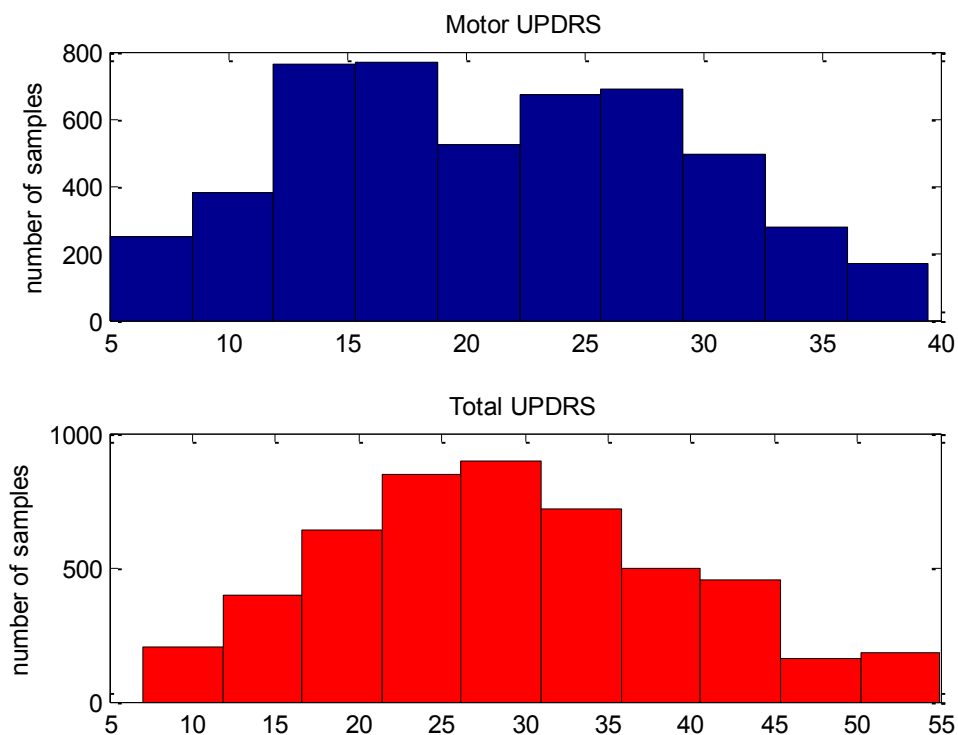


Obr. II.3 Rozsahy příznaků poměrů (vstupy)

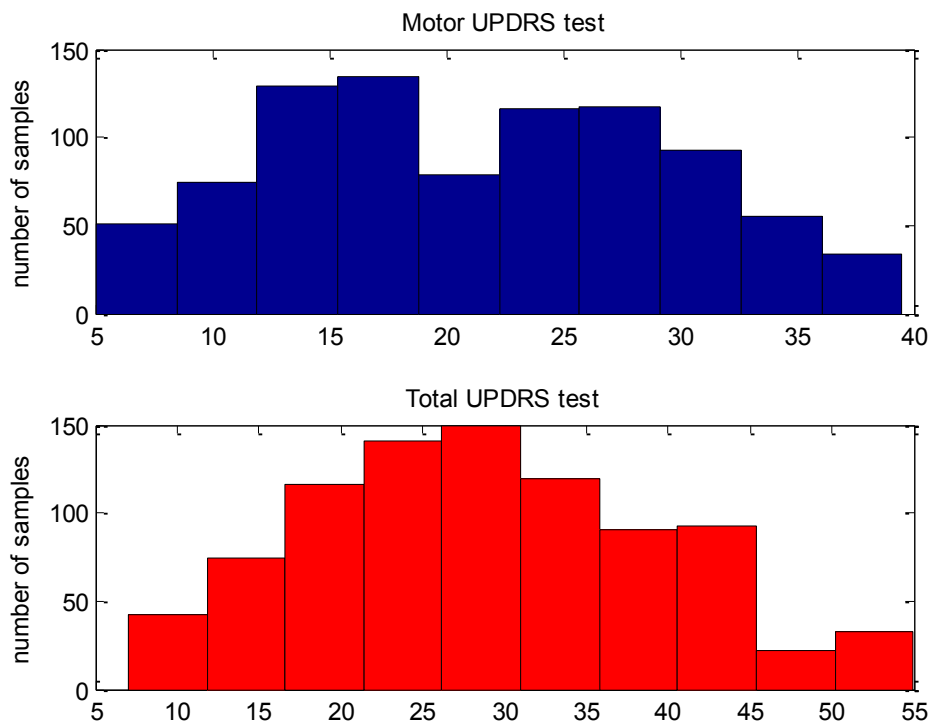


Obr. II.4 Celkový přehled rozsahů 16 příznaků vstupních dat

Příloha III Histogramy požadovaných výstupů výpočetních modelů

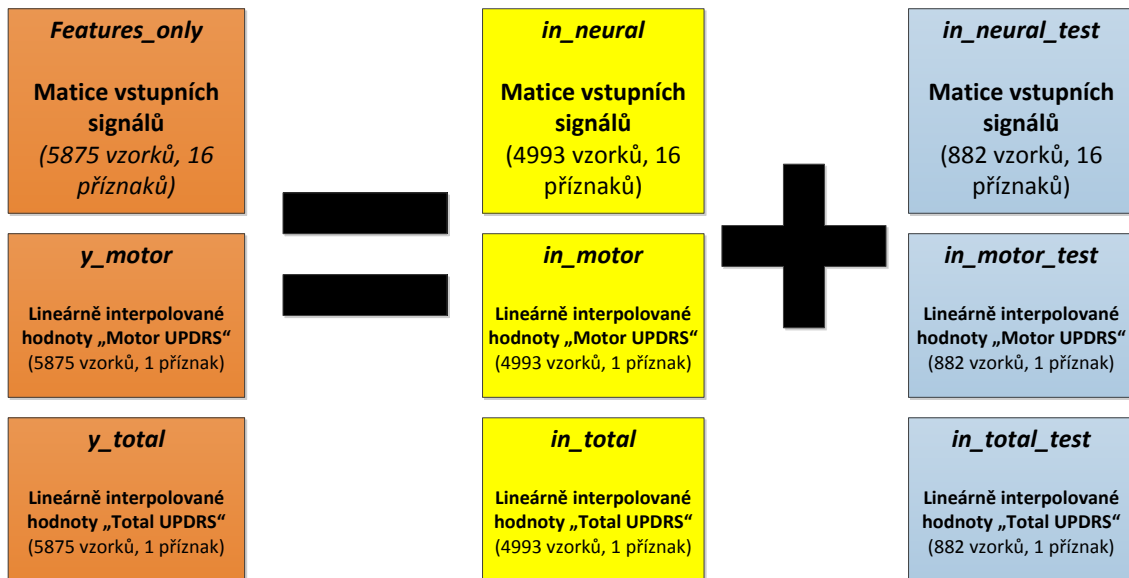


Obr. III.1 Histogramy požadovaných výstupů určených pro trénování, validaci a testování



Obr. III.2 Histogramy požadovaných výstupů pro nezávislý test

Příloha IV Rozdělení vstupních dat pro metodu ANFIS



Obr. IV.1 Rozdělení vstupních dat pro metodu ANFIS

Příloha V Příklad skriptu mpiLibConf.m (pro paralelní výpočty)

```
function [lib, extras] = mpiLibConf
%MATLAB MPI Library overloading for Infiniband Networks

mpich = '/opt/intel/impi/4.1.1.036/lib64/';

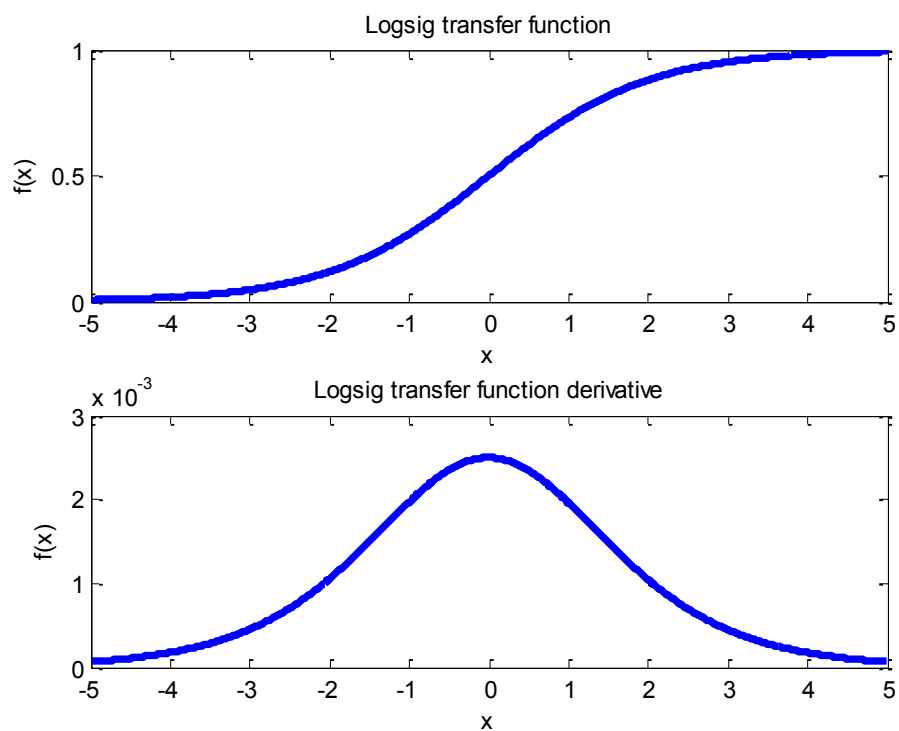
disp('Using Intel MPI 4.1.1.036 over Infiniband')

lib = strcat(mpich, 'libmpich.so');
mpl = strcat(mpich, 'libmpl.so');
opa = strcat(mpich, 'libopa.so');

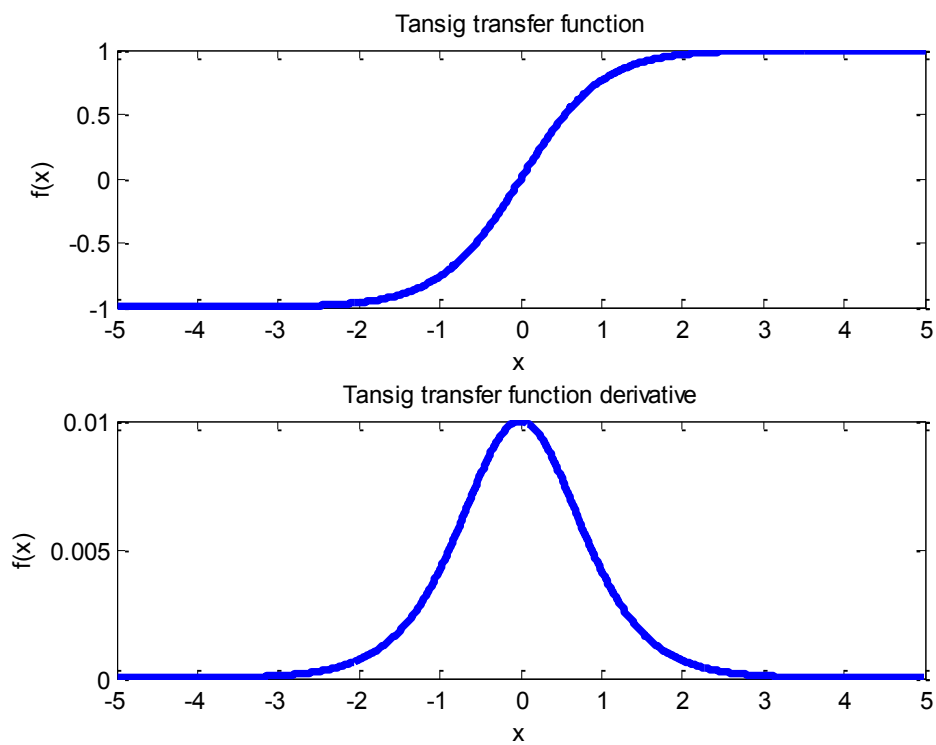
extras = {};
```

Pozn: tento skript musí být umístěn v adresáři ~/matlab. Adresa určená v proměnné *mpich* musí odpovídat umístění MPI knihovny v daném výpočetním systému. [25]

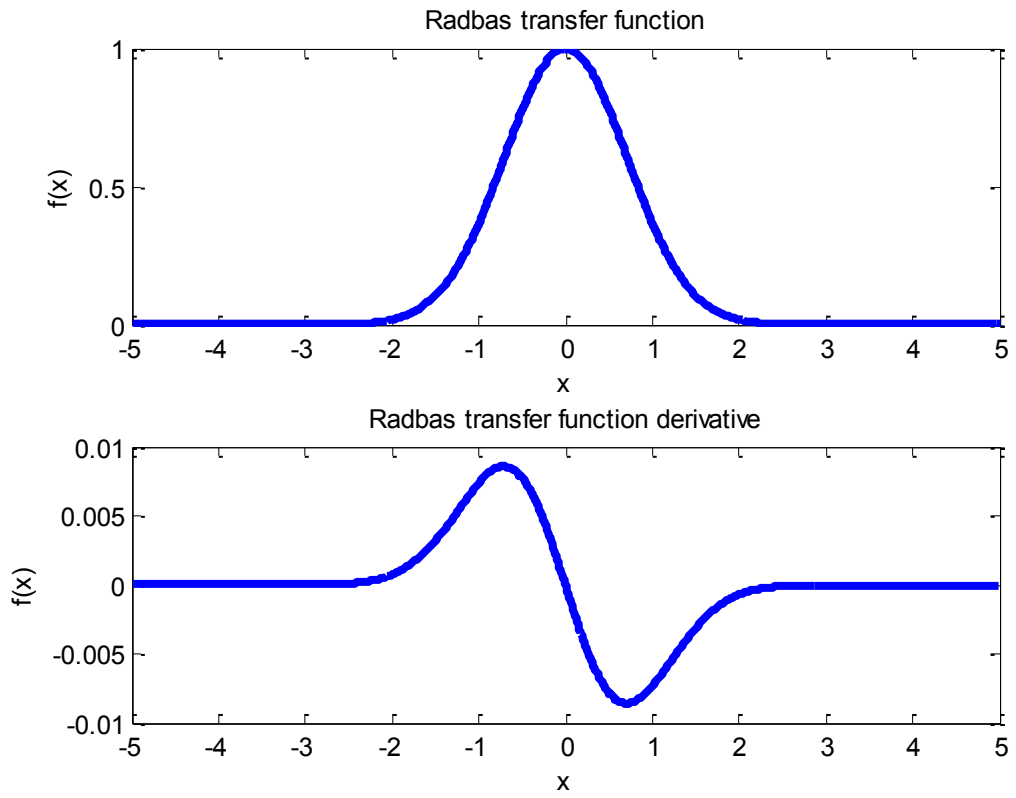
Příloha VI Typy aktivační funkcí neuronů



Obr. VI.1 Aktivační funkce logaritmická sigmoida a její derivace



Obr. VI.2 Aktivační funkce hyperbolická tangenta a její derivace



Obr. VI.3 Aktivační funkce radiální báze a její derivace

Příloha VII Popis příznaků měřeného řečového signálu

- **Subject #** ID pacienta
- **Age** Věk pacienta
- **Sex** Pohlaví pacienta; 0 = muž, 1 = žena
- **Test_time** Časové údaje o měření řečového signálu
- **Motor_UPDRS** Klinicky zjištěná hodnota motorického UPDRS, lineárně interpolovaná
- **Total_UPDRS** Klinicky zjištěná hodnota celkového UPDRS, lineárně interpolovaná
- **Jitter(%)** Střední relativní odchylka mezi konsekutivními periodami, vydělená střední periodou. Patologickou hodnotou je 1,040%.
- **Jitter(Abs)** Střední absolutní odchylka mezi konsekutivními periodami, uvedeno v sekundách. Patologickou hodnotou je 82,2 μ s.
- **Jitter:RAP** Relativní koeficient perturbace (odchylky), střední absolutní odchylka mezi periodou, mezi jejím průměrem a průměrem dvou nejbližších sousedních period, vydělen průměrnou periodou. Patologickou hodnotou je 0,68%.
- **Jitter(PPQ5)** Pětibodový koeficient perturbace (odchylky), střední absolutní odchylka mezi periodou, mezi jejím průměrem a průměrem čtyř nejbližších sousedních period, vydělen průměrnou periodou. Patologickou hodnotou je 0,84%.
- **Jitter(DDP)**
- **Shimmer** Střední absolutní odchylka mezi amplitudami konsekutivních period, vydělená střední amplitudou. Patologickou hodnotou je 3,81%.
- **Shimmer(dB)** Hodnota shimmer uvedená v dB.
- **Shimmer:APQ3** Trojbodový koeficient amplitudové perturbace (odchylky), střední absolutní odchylka mezi amplitudou periody a středními amplitudami sousedních period, vydělen střední amplitudou.
- **Shimmer:APQ5** Pětibodový koeficient amplitudové perturbace, viz shimmer:APQ3.
- **Shimmer:APQ11** Jedenáctibodový koeficient amplitudové perturbace, viz příznak shimmer:APQ3; za patologickou je považována hodnota 3,070%.
- **Shimmer:DDA** Střední absolutní odchylka mezi konsekutivními odchylkami mezi amplitudami po sobě jdoucích period.
- **NHR** Poměr šum-harmonická (příčemž harmonická = užitečný signál).
- **HNR** Poměr harmonická-šum.
- **RPDE** Recurrence Period Density Entropy, určuje opakovanost signálu.
- **DFA** Detrended Fluctuation Analysis – exponent fraktál. rozlišení signálu.
- **PPE** Nelineární parametr variace základní frekvence řečového signálu.

Pozn.: Hodnoty *Jitter* obecně vyjadřují rozptyl základní frekvence řečového signálu, hodnoty *Shimmer* pak vyjadřují rozptyl amplitud řečového signálu.

Pozn. 2: Při zpracování dat nejsou použity příznaky sloužící k identifikaci pacientů (ID apod.).

Příloha VIII Statistická analýza výsledků výpočetních modelů – Motor UPDRS

Tab. VIII.1 CGB s Powell-Bealeovými restarty – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgb	80 /27	radbas /radbas	152	Validation stop.	24	47,397	5,606	0,573
traincgb	88 /30	radbas /logsig	172	Validation stop.	33	47,623	5,607	0,571
traincgb	72 /24	radbas /radbas	130	Validation stop.	18	47,798	5,695	0,568

Tab. VIII.2 CGB s Fletcher-Reevesovými updaty – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgf	48 /16	radbas /radbas	178	Validation stop.	17	47,719	5,655	0,571
traincgf	56 /19	radbas /radbas	153	Validation stop.	17	48,276	5,669	0,562
traincgf	88 /30	radbas /tansig	171	Validation stop.	31	48,487	5,727	0,560

Tab. VIII.3 CGB s Polak-Ribiéryho updaty – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgp	48 /12	radbas /logsig	217	Validation stop.	22	48,142	5,666	0,564
traincgp	96 /24	radbas /radbas	169	Validation stop.	28	48,440	5,702	0,561
traincgp	96 /48	radbas /radbas	127	Validation stop.	27	48,477	5,684	0,560

Tab. VIII.4 Scaled Conjugate Gradient – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainscg	80 /27	radbas /logsig	222	Validation stop.	16	47,625	5,648	0,571
trainscg	56 /19	radbas /radbas	195	Validation stop.	10	47,952	5,670	0,566
trainscg	48 /24	radbas /radbas	194	Validation stop.	10	48,353	5,641	0,562

Tab. VIII.5 Resilient Backpropagation – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainrp	96 /24	tansig /logsig	230	Validation stop.	11	46,548	5,553	0,584
trainrp	56 /14	tansig /logsig	210	Validation stop.	5	47,134	5,660	0,577
trainrp	64 /32	radbas /logsig	269	Validation stop.	11	47,446	5,574	0,574

Tab. VIII.6 Adaptivní míra rychlosti učení – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traingda	40 /14	radbas /tansig	93	Validation stop.	2	63,469	6,763	0,331
traingda	64 /32	radbas /radbas	128	Validation stop.	5	63,785	6,725	0,338
traingda	64 /16	radbas /radbas	52	Validation stop.	2	63,360	6,781	0,323

Tab. VIII.7 Momentum – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traingdm	40 /14	radbas /logsig	241	Validation stop.	5	58,496	6,411	0,417
traingdm	8	logsig	300	Maximum epoch reached.	2	62,831	6,713	0,333
traingdm	8 /3	logsig /radbas	300	Maximum epoch reached.	2	63,033	6,695	0,328

Tab. VIII.8 Adaptivní míra rychlosti učení a momentum – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traingdx	80 /20	radbas /radbas	108	Validation stop.	4	56,559	6,232	0,448
traingdx	96 /32	radbas /radbas	118	Validation stop.	5	56,680	6,340	0,446
traingdx	40 /10	radbas /radbas	139	Validation stop.	3	56,960	6,321	0,443

Tab. VIII.9 One-Step Secant – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainoss	80 /40	radbas /radbas	156	Validation stop.	38	49,830	5,830	0,542
trainoss	96 /48	radbas /radbas	138	Validation stop.	40	49,883	5,815	0,542
trainoss	96 /32	radbas /radbas	201	Validation stop.	46	49,906	5,807	0,541

Tab. VIII.10 Levenberg-Marquardt – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainlm	16 /6	logsig /logsig	58	Validation stop.	18	44,689	5,326	0,611
trainlm	8 /4	tansig /logsig	69	Validation stop.	13	44,651	5,340	0,607
trainlm	32 /16	tansig /radbas	30	Validation stop.	28	45,028	5,293	0,609
trainlm	48 /16	logsig /logsig	48	Validation stop.	82	45,010	5,395	0,604
trainlm	64 /16	logsig /logsig	51	Validation stop.	173	45,042	5,318	0,606

Tab. VIII.11 Kaskádová neuronová síť – Levenberg-Marquardt – Motor UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainlm	32	tansig	54	Validation stop.	29	46,228	5,489	0,593
trainlm	56	logsig	48	Validation stop.	42	46,286	5,422	0,592
trainlm	48	tansig	48	Validation stop.	54	46,218	5,504	0,590

Příloha IX Statistická analýza výsledků výpočetních modelů – Total UPDRS

Tab. IX.1 CGB s Powell-Bealeovými restarty – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgb	88 /22	radbas /tansig	198	Validation stop.	33	78,115	7,077	0,590
traincgb	88 /30	radbas /radbas	186	Validation stop.	32	78,612	7,102	0,588
traincgb	88 /44	radbas /radbas	210	Validation stop.	42	78,984	7,087	0,585

Tab. IX.2 CGB s Fletcher-Reevesovými updaty – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgf	88 /44	radbas /radbas	156	Validation stop.	30	79,717	7,193	0,580
traincgf	96 /32	radbas /radbas	110	Validation stop.	20	80,803	7,287	0,570
traincgf	96 /48	radbas /logsig	140	Validation stop.	34	81,066	7,358	0,568

Tab. IX.3 CGB s Polak-Ribiéryho updaty – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traincgp	32 /16	radbas /tansig	260	Validation stop.	18	78,710	7,121	0,586
traincgp	64 /22	radbas /radbas	207	Validation stop.	25	79,264	7,175	0,582
traincgp	40 /10	radbas /tansig	213	Validation stop.	15	80,249	7,210	0,574

Tab. IX.4 Scaled Conjugate Gradient – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainscg	48 /12	radbas /radbas	156	Validation stop.	6	79,182	7,089	0,583
trainscg	56 /14	tansig /radbas	227	Validation stop.	11	80,195	7,213	0,575
trainscg	72 /36	radbas /radbas	151	Validation stop.	12	80,677	7,252	0,572

Tab. IX.5 Resilient Backpropagation – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainrp	32 /8	logsig /logsig	294	Validation stop.	5	78,827	7,099	0,585
trainrp	80 /40	radbas /radbas	105	Validation stop.	5	78,864	7,101	0,585
trainrp	72 /36	tansig /tansig	196	Validation stop.	8	78,913	7,075	0,584

Tab. IX.6 Adaptivní míra rychlosti učení – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traingda	24 /6	radbas /logsig	74	Validation stop.	1	105,86	8,550	0,360
traingda	56 /14	radbas /radbas	80	Validation stop.	2	105,99	8,287	0,349
traingda	48 /24	radbas /radbas	67	Validation stop.	2	106,08	8,282	0,349

Tab. IX.7 Momentum – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traingdm	24 /6	logsig /radbas	144	Validation stop.	2	103,15	8,287	0,378
traingdm	24 /6	radbas /radbas	178	Validation stop.	3	105,11	8,416	0,361
traingdm	24 /8	tansig /logsig	10	Validation stop.	1	131,67	9,109	0,138

Tab. IX.8 Adaptivní míra rychlosti učení a momentum – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
traingdx	64 /32	radbas /radbas	158	Validation stop.	6	94,52	8,043	0,462
traingdx	64 /22	radbas /radbas	113	Validation stop.	4	94,538	7,963	0,461
traingdx	56 /14	radbas /radbas	242	Validation stop.	6	96,432	7,997	0,444

Tab. IX.9 One-Step Secant – Total UPDRS

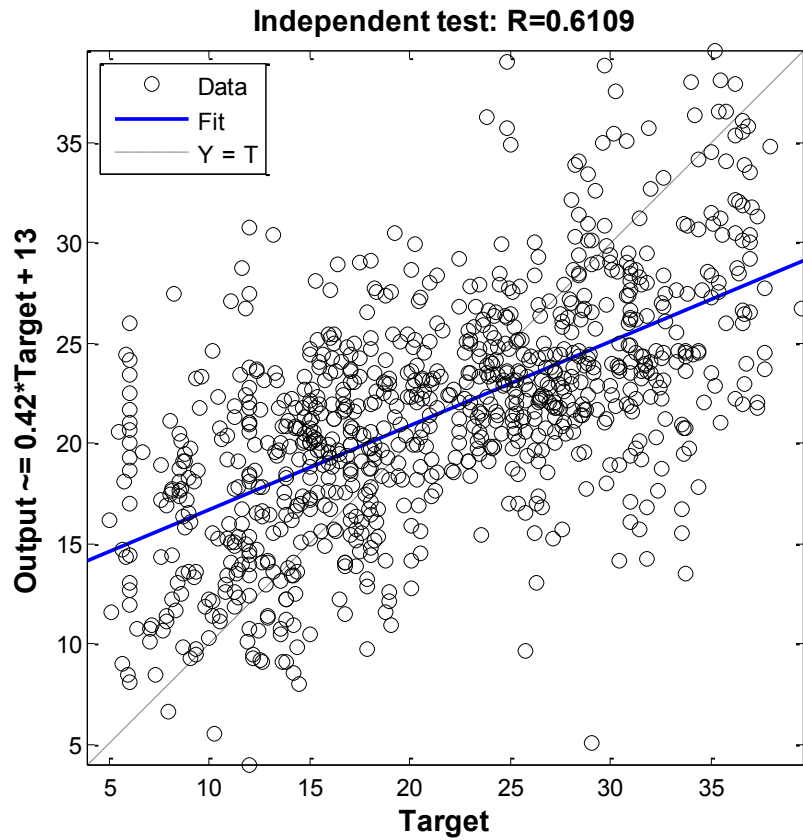
Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainoss	96 /48	radbas /radbas	216	Validation stop.	60	81,368	7,277	0,567
trainoss	40 /14	radbas /radbas	227	Validation stop.	24	81,890	7,251	0,563
trainoss	72 /24	radbas /radbas	200	Validation stop.	37	81,897	7,261	0,564

Tab. IX.10 Levenberg-Marquardt – Total UPDRS

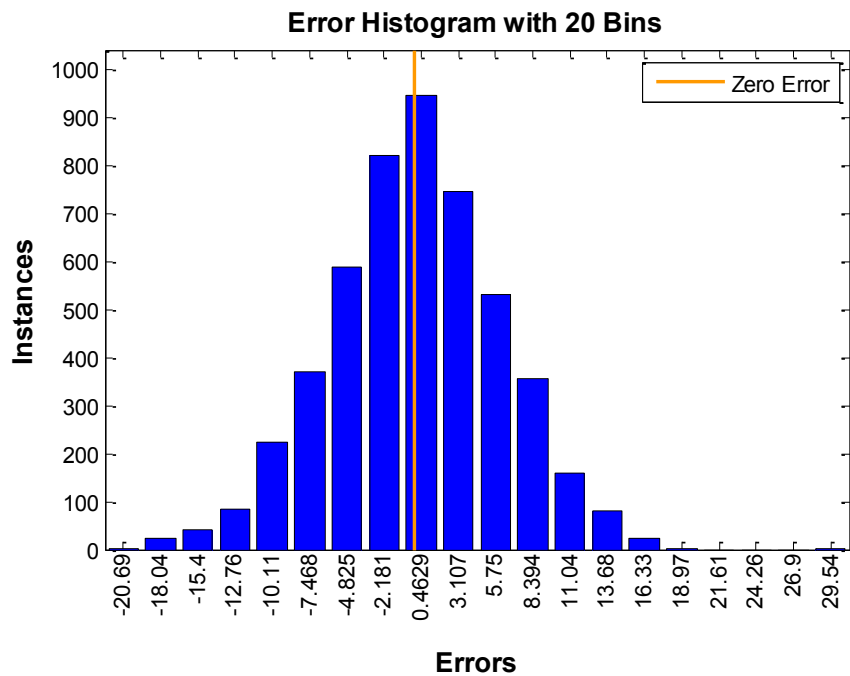
Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainlm	16 /6	logsig /logsig	62	Validation stop.	22	72,709	6,683	0,631
trainlm	24 /8	logsig /tansig	55	Validation stop.	24	73,188	6,722	0,630
trainlm	24 /6	tansig /tansig	48	Validation stop.	26	73,798	6,656	0,624
trainlm	48 /12	logsig /logsig	74	Validation stop.	146	73,766	6,815	0,623
trainlm	16 /8	logsig /tansig	60	Validation stop.	22	74,643	6,763	0,618

Tab. IX.11 Kaskádová neuronová síť – Levenberg-Marquardt – Total UPDRS

Algorithm	Neurons #	Transfer functions	Epochs #	Training stop	Training time [s]	MSE	MAE	Corr. coeff
trainlm	88	tansig	58	Validation stop.	135	75,577	6,869	0,612
trainlm	96	logsig	56	Validation stop.	134	76,093	6,938	0,613
trainlm	88	logsig	43	Validation stop.	89	75,047	7,043	0,611



Obr. X.1 Regresní analýza – neuronová síť s algoritmem Levenberg-Marquardt



Obr. X.2 Hodnoty MAE – neuronová síť s algoritmem Levenberg-Marquardt