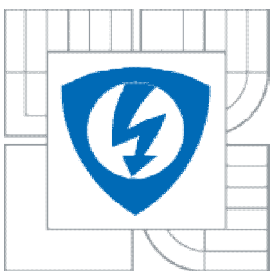




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

PREDIKCE DATOVÉHO TOKU V POČÍTAČOVÝCH SÍTÍCH

PREDICTION OF DATA FLOW IN COMPUTER NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

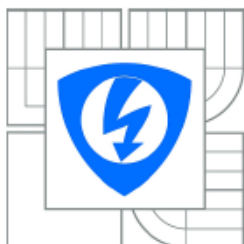
AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ ZVĚŘINA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR VYCHODIL

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Lukáš Zvěřina
Ročník: 2

ID: 106895
Akademický rok: 2012/2013

NÁZEV TÉMATU:

Predikce datového toku v počítačových sítích

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou predikce výskytu dat v počítačových sítích. Zamyslete se nad možnostmi predikce síťového provozu, uveďte jejich přehled a vyberte vhodné řešení aplikovatelné na reálný síťový provoz. Svůj výběr zdůvodněte a vybranou metodu podrobně popište.

DOPORUČENÁ LITERATURA:

- [1] JIRSÍK, V. Umělá inteligence: skripta. Brno: FEKT VUT v Brně, 2012
- [2] VONDRÁK, I. Neuronové sítě: skripta. Ostrava: VŠB v Ostravě, 2001. 56 s
- [3] KACÁLEK, J., MÍČA, I. Nelineární analýza a predikce síťového provozu. Brno: Ústav telekomunikací, FEKT VUT v Brně, 2009. 6 s

Termín zadání: 11.2.2013

Termín odevzdání: 29.5.2013

Vedoucí práce: Ing. Petr Vychodil

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor semestrální práce nesmí při vytváření semestrální práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Předmětem této diplomové práce bylo seznámit se s problematikou predikce výskytu dat v počítačových sítích. Dále se tato práce zabývala síťovým provozem a analýzou jeho vlastností. V této práci byly rozebrány možnosti predikce síťového provozu pomocí FARIMA modelu, teorii chaosu s Lyapunovým exponentem a pomocí neuronových sítí. Nejpodrobněji zde byly probrány možnosti predikce, se zaměřením na neuronové sítě. Konkrétně tedy hlavně na rekurentní neuronové sítě. Predikce se prováděly pomocí programu Matlab ve vývojovém prostředí Neural Network Toolbox, kde byly vytvořeny, natrénovány a otestovány neuronové sítě pro predikci konkrétních druhů síťového provozu. Pro testování byly vybrány Elmanova síť, NARX síť a obecná LRN rekurentní síť. Získané výsledky byly přehledně zpracovány do tabulek a vyneseny do grafických závislostí před a po použití navržené predikční techniky se závěrečným zhodnocením.

Klíčová slova: Síťový provoz, dynamická alokace šířky pásma, predikce, rekurentní neuronová síť, Neural Network Toolbox

ABSTRACT

The aim of this thesis was to study problems of prediction of data in computer networks. Furthermore, this work deals with network traffic and analyzing its properties. In this study were analyzed the possibilities of network traffic prediction using Farima model, the theory of chaos with Lyapunov exponents and neural networks. Possibilities of prediction with the focus on neural network were discussed in detail here, mainly on recurrent neural networks. Prediction was performed in Matlab development environment in Neural Network Toolbox, where they were created, trained and evaluated neural network to predict specific types of network traffic. For testing were selected Elman network NARX network and general LRN recurrent network. The results were clearly organized into tables and plotted in graphical relationships before and after the use of predictive techniques designed to final evaluation.

Keywords: Network traffic, dynamic bandwidth allocation, prediction, recurrent neural network, Neural Network Toolbox

Bibliografická citace práce

ZVĚŘINA, L. *Predikce datového toku v počítačových sítích*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 80 s. Vedoucí diplomové práce Ing. Petr Vychodil.

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Predikce datového toku v počítačových sítích jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne.....

.....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Petrovi Vychodilovi za metodické vedení a cenné rady, které mi poskytl během psaní mé práce. Děkuji rodičům za podporu a za to, že mně umožnili studovat a dojít takto daleko. V neposlední řadě děkuji své přítelkyni za podporu během celého studia.

V Brně dne

.....

podpis autora

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|--------|-----------------------------------------------------|
| ARMA | Autoregressive Moving Average |
| CDM | Code Division Multiplex |
| FARIMA | Fractional Autoregressive Integrated Moving Average |
| FDM | Frequency Division Multiplex |
| IP | Internet Protocol |
| LAN | Local Area Network |
| LRN | Layer Recurrent Network |
| NARX | Nonlinear Autoregressive Exogenous |
| PC | Personal Computer |
| QoS | Quality of Service |
| TDM | Time Division Multiplex |
| VoIP | Voice over Internet Protocol |
| WDM | Wavelength Division Multiplexing |
| WWW | World Wide Web |

OBSAH

| | |
|-------------------------------------------------------------|----|
| ÚVOD..... | 12 |
| 1 SÍŤOVÝ PROVOZ | 13 |
| 1.1 Přenosová rychlost | 13 |
| 1.2 Přenosový kanál | 14 |
| 1.3 Šířka pásma a její sdílení | 14 |
| 1.3.1 Časový multiplex – TDM..... | 15 |
| 1.3.2 Frekvenční multiplex – FDM..... | 16 |
| 1.3.3 Kódový multiplex – CDM..... | 16 |
| 1.3.4 Vlnový multiplex – WDM | 17 |
| 1.4 Alokace šířky pásma..... | 18 |
| 2 SAMOPODOBNOST..... | 19 |
| 2.1 Statické vlastnosti síťového provozu | 19 |
| 2.2 Matematické vyjádření samopodobnosti | 21 |
| 3 MOŽNOSTI PREDIKCE SÍŤOVÉHO PROVOZU | 22 |
| 3.1 FARIMA model..... | 23 |
| 3.2 Teorie chaosu | 24 |
| 3.2.1 Matematické vyjádření Lyapunova exponentu | 26 |
| 3.3 Neuronové sítě | 27 |
| 4 ZÁKLADNÍ VLASTNOSTI NEURONOVÉ SÍTĚ | 29 |
| 4.1 Biologický neuron | 30 |
| 4.2 Formální neuron | 32 |
| 4.3 Aktivační funkce | 33 |
| 4.3.1 Lineární satureovaná funkce | 34 |
| 4.3.2 Standardní sinoida | 35 |
| 4.3.3 Hyperbolický tangens..... | 35 |
| 4.3.4 Funkce radiální báze..... | 36 |
| 4.3.5 Tan-sigmoidální funkce | 37 |
| 5 NEURONOVÉ SÍTĚ VHODNÉ PRO PREDIKCI SÍŤOVÉHO PROVOZU | 38 |
| 5.1 Základní rozdělení neuronových sítí..... | 38 |
| 5.2 Dopředné sítě | 39 |
| 5.3 Rekurentní sítě | 39 |
| 5.3.1 Elmanova síť | 40 |
| 5.3.2 NARX síť..... | 41 |

| | | |
|-------|---------------------------------------------------------------------------------|----|
| 5.3.3 | Obecná rekurentní síť – LRN..... | 42 |
| 6 | VYUŽITÍ NEURONOVÝCH SÍTÍ V PRAXI | 42 |
| 7 | VÝHODY A NEVÝHODY PREDIKCE SÍŤOVÉHO PROVOZU POMOCÍ NEURONOVÝCH SÍTÍ | 44 |
| 8 | NÁVRH PRAKTICKÉ ČÁSTI..... | 44 |
| 8.1 | Zachycení testovacích dat..... | 45 |
| 8.2 | Agregace a normalizace zachycených dat..... | 49 |
| 8.3 | Návrh predikčního systému | 51 |
| 9 | VYTVOŘENÍ, NATRÉNOVÁNÍ A OTESTOVÁNÍ NEURONOVÝCH SÍTÍ V PROSTŘEDÍ NN TOOLBOX ... | 53 |
| 9.1 | Vytvoření neuronové sítě..... | 53 |
| 9.2 | Trénování neuronové sítě | 57 |
| 9.3 | Predikce neznámých hodnot (princip generalizace)..... | 60 |
| 10 | PARAMETRY A METODY VYTVOŘENÝCH SÍTÍ | 64 |
| 10.1 | Metoda Min – max | 64 |
| 10.2 | Střední čtvercová chyba MSE | 64 |
| 10.3 | Trénovací funkce TRAINBFG..... | 65 |
| 10.4 | Transformační funkce vrstvy | 65 |
| 11 | VÝBĚR VHODNÉHO MODELU SÍTĚ..... | 66 |
| 11.1 | Zhodnocení praktické části | 68 |
| 12 | ZÁVĚR | 71 |
| | POUŽITÁ LITERATURA | 73 |
| | SEZNAM PŘÍLOH | 75 |
| B | OBSAH PŘÍLOŽENÉHO MÉDIA | 80 |

SEZNAM POUŽITÝCH OBRÁZKŮ

| | |
|----------------------------------------------------------------------------|----|
| Obr. 1 Přenosový řetězec | 14 |
| Obr. 2 Princip časového multiplexu..... | 15 |
| Obr. 3 Princip frekvenčního multiplexu..... | 16 |
| Obr. 4 Princip kódového multiplexu..... | 17 |
| Obr. 5 Princip vlnového multiplexu..... | 17 |
| Obr. 6 Samopodobnost list kapradí..... | 19 |
| Obr. 7 Agregovaný síťový provoz (stupeň agregace 1 a 100)..... | 20 |
| Obr. 8 Konvergující dráhy atraktoru..... | 26 |
| Obr. 9 Divergující dráhy atraktoru | 26 |
| Obr. 10 Realizace predikce pomocí neuronové sítě | 28 |
| Obr. 11 Realizace predikce pomocí neuronové sítě | 28 |
| Obr. 12 Biologický neuron..... | 31 |
| Obr. 13 Spojení neuronů | 31 |
| Obr. 14 Schématické propojení neuronů..... | 32 |
| Obr. 15 Formální neuron..... | 33 |
| Obr. 16 Průběh Lineární saturační funkce neuronu..... | 34 |
| Obr. 17 Průběh sigmoidální aktivační funkce neuronu..... | 35 |
| Obr. 18 Průběh aktivační funkce hyperbolické tangenty neuronu | 36 |
| Obr. 19 Průběh aktivační funkce radiální báze neuronu | 37 |
| Obr. 20 Průběh tan-sigmoidální aktivační funkce neuronu..... | 38 |
| Obr. 21 Dopředná topologie neuronové sítě | 39 |
| Obr. 22 Rekurentní topologie neuronové sítě..... | 40 |
| Obr. 23 Elmanova síť..... | 41 |
| Obr. 24 Zachycený provoz ICQ komunikace pomocí analyzátoru Wireshark | 46 |
| Obr. 25 Program Wireshark | 47 |
| Obr. 26 Uložení provozu do souboru celkovy_download.txt. | 47 |
| Obr. 27 Zachycený upload pomocí analyzátoru Wireshark..... | 48 |
| Obr. 28 Zachycený download pomocí analyzátoru Wireshark | 49 |
| Obr. 29 Import dat v programu Matlab..... | 50 |
| Obr. 30 Zobrazení vstupních a výstupních trénovacích vzorů..... | 51 |
| Obr. 31 Princip učení | 52 |
| Obr. 32 Princip generalizace | 52 |

| | |
|--------------------------------------------------------------------|----|
| Obr. 33 Výběrové okno Network/Data Manager | 54 |
| Obr. 34 Import dat | 54 |
| Obr. 35 Vytvoření nové neuronové sítě | 55 |
| Obr. 36 Nastavení Network/Data Manageru | 56 |
| Obr. 37 Informace o vytvoření nové neuronové sítě | 56 |
| Obr. 38 Zobrazení topologie navrhnuté sítě | 57 |
| Obr. 39 Nastavení trénovacích dat..... | 57 |
| Obr. 40 Nastavení trénovacích parametrů..... | 58 |
| Obr. 41 Grafická podoba trénovacího procesu..... | 59 |
| Obr. 42 Závislost trénovací chyby na počtu epoch..... | 60 |
| Obr. 43 Simulace - Předložení neznámých vstupních hodnot..... | 61 |
| Obr. 44 Informativní okno o provedení simulace..... | 61 |
| Obr. 45 Export dat do okna Workspace | 61 |
| Obr. 47 Predikce neznámých hodnot provozu | 62 |
| Obr. 46 Informativní okno úspěšného exportu dat | 62 |
| Obr. 48 Blokové schéma praktické části..... | 63 |
| Obr. 49 Transformační funkce Tansig a Purelin | 65 |
| Obr. 50 Predikce downloadu pomocí sítě LRN..... | 69 |
| Obr. 51 Predikce uploadu pomocí Elmanovy sítě | 69 |
| Obr. 52 Predikce downloadu aplikace ICQ pomocí Elmanovy sítě | 70 |
| Obr. 53 Predikce uploadu aplikace ICQ pomocí Elmanovy sítě..... | 70 |

ÚVOD

Modely a predikce síťového provozu hrají významnou roli v analýze reálného síťového provozu a výkonnosti sítě. V případě dostatečně přesné predikce by bylo možné zlepšit využití a výkonnost sítě. Tato diplomová práce se zabývá možností predikce síťového provozu pomocí FARIMA modelu, teorii chaosu s Lyapunovým exponentem a pomocí neuronových sítí. V této práci je popsán síťový provoz a analýza jeho vlastností. Velká část této práce byla věnována predikci síťového provozu za pomoci neuronových sítí.

Pomocí programu Wireshark byly zachyceny na lokální síti (LAN) konkrétní druhy síťových přenosů. A to celkový upload/download, upload/download aplikace ICQ, které nám budou sloužit jako trénovací a testovací data pro predikci síťového provozu pomocí rekurentních neuronových sítí v programu Matlab s využitím Neural Network Toolbox.

K testování predikce síťového provozu byla vybrána dle teoretických předpokladů Elmanova síť, LRN a NARX síť díky svým vhodným topologiím a specifickým vlastnostem, které jsou analyzovány nelineárního chování síťového provozu, schopnost predikce samopodobných jevů, které jsou popsány v druhé kapitole. Vlastnosti síťového provozu se neustále mění, tak tyto sítě díky svým zpětným vazbám tvoří dynamické systémy. Dalšími důležitými parametry těchto sítí jsou rychlost učení, malá střední chyba predikovaných dat a modifikace sítě pro lepší výsledek predikce. O návrhu a popisu postupu samotné predikce pomocí neuronových sítí nám pojednává kapitola 8.

Pomocí neuronových sítí lze u síťového provozu předpovídat konkrétní hodnoty, které jsou z blízké budoucnosti. Dále také umí tyto sítě předpovídat trend vývoje, což je ale obecně složitější, než předpovídat konkrétní blízké hodnoty. Tento způsob by mohl být implementován v praxi například pro dynamickou alokaci šířky pásma jednotlivých front u frontového systému ve směrovačích. Neuronová síť totiž může predikovat zvýšený provoz sítě, který je při reálných přenosech, (jako jsou například internetová telefonie, video konference, video telefonie, video na vyžádání) v jehož důsledku by mohlo dojít k zahazování paketů v jedné frontě, zatímco druhá by byla poloprázdná. Dynamická alokace šířky pásma by se mohla uplatnit na směrovačích v přenosové cestě, které by zajišťovaly potřebnou kvalitu služeb (QoS - Quality of Services). Obecně lze tedy říci, že neuronové sítě slouží svou činností ke zlepšení systémů, které ale nejsou bohužel v tomto oboru dostatečně prozkoumány, otestovány a využívány.

1 SÍŤOVÝ PROVOZ

První kapitola se zabývá vlastnostmi síťového provozu. S tímto tématem jsou úzce spojeny pojmy, jako je přenosová rychlost, přenosový kanál a šířka pásma. Dále je zde rozebráno dělení šířky pásma a její možná alokace.

1.1 Přenosová rychlost

Nám udává, jaký objem informace se přenesení za jednotku času. Základní jednotkou je bit za sekundu (bit/s, b/s, nebo se značí anglicky bps = bits per second). Jednotka udává, kolik bitů informace je přeneseno zvoleným přenosovým kanálem za jednu sekundu. Také se zde používá jednotka bajty za sekundu (B/s, Byte/s, nebo Bps) a její násobky, přičemž platí, že $1 \text{ B/s} = 8 \text{ b/s}$.

S velikostí přenosové rychlosti souvisí modulační rychlost udávaná v Baudech (Bd), která vyjadřuje počet možných změn stavu signálu za jednotku času, udává tedy počet signálových prvků přenesených za sekundu. Modulační rychlost je dána vztahem

$$M = \frac{1}{a} [\text{Bd}] , \quad (1)$$

kde a je šířka impulzu (doba jeho trvání).

Čím je větší modulační rychlost, tím je menší doba trvání signálového prvku. Z Nyquistova kritéria pak vyplývá, že maximální modulační rychlost je dána vztahem:

$$M = 2B [\text{Bd}] , \quad (2)$$

kde B je šířka pásma. Modulační rychlost M je s přenosovou rychlostí R vázána vztahem:

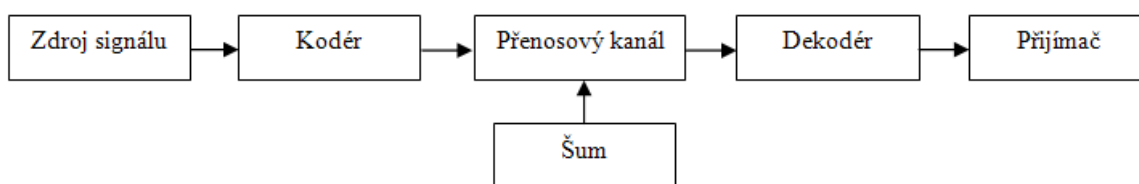
$$R = M \log_2 Q [\text{b/s}] , \quad (3)$$

kde Q je počet všech možných stavů signálu (jeden signál může teoreticky nabývat 2^n stavů), poté hovoříme o více stavovém signálu.

Z uvedených vztahů (1) až (3) vyplývá, že přenosovou rychlost lze zvyšovat pouze zvýšením počtu stavů signálu a zvýšením šířky pásma, přičemž oba tyto parametry mají své limity. [1]

1.2 Přenosový kanál

Je soubor prostředků mezi dvěma uzly sítě umožňující přenos signálu a skládá se z vysílacího rozhraní, přenosového média a přijímacího rozhraní. Přenosový kanál vytvářejí přenosové technologie, které specifikují např. rychlost přenosu, pořadí bitů, přenosové médium, fyzikální charakteristiku signálu, zařízení pro regeneraci signálu atd. Každý přenosový kanál je vždy schopen přenášet jen signály o frekvenci z určitého omezeného intervalu. Je to komunikační zařízení se vstupem a výstupem jak je znázorněno na obrázku 1. [2]



Obr. 1 Přenosový řetězec

[4]

Následující kapitola je věnována šířce pásma, jejímu efektivnímu využití, respektive rozdělení.

1.3 Šířka pásma a její sdílení

Šířka pásma určuje šířku intervalu frekvencí, které je přenosový kanál schopen přenést. Jednotka šířky pásma je stejná jako jednotka frekvence, čili Hz. Obecně platí, že čím větší je šířka pásma přenosového kanálu, tím větší je přenosová rychlost, kterou na něm lze dosáhnout. Pokud by byl přenosový kanál s velkou šířkou pásma využíván pouze ke komunikaci dvou uživatelů, daný postup by nebyl efektivní. Proto se vyvinuly metody, které umožňují sdílení daného kanálu a je tedy možné tento kanál využívat více uživateli. V této souvislosti jsou zde zaváděny pojmy multiplexování a demultiplexování.

Multiplexování je termín popisující proces, ve kterém je více analogových signálů, nebo digitálních datových toků kombinováno do jednoho výsledného signálu vhodného k dalšímu přenosu komunikačním kanálem. Cílem je co možná nejefektivnější využití daného přenosového média. Zařízení provádějící multiplexování se nazývá se multiplexor. Zařízení provádějící převod signálu zpět na jednotlivé původní signály se nazývá demultiplexor a jedná se o jev demultiplexování.

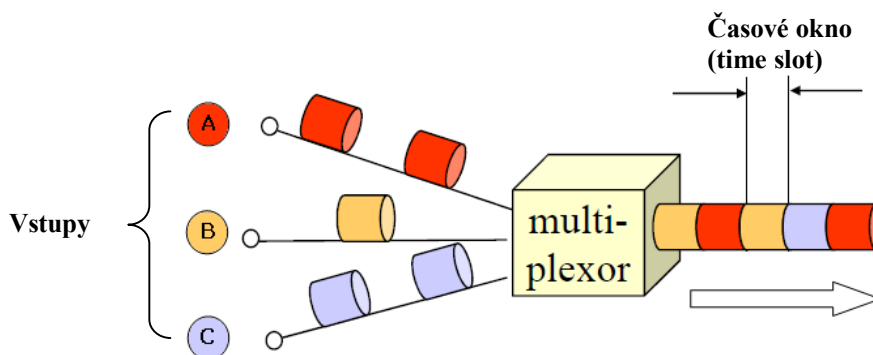
V praxi se používají čtyři základní techniky multiplexování a to konkrétně:

- Časové multiplexování (TDM – Time Division Multiplexing),
- Frekvenční multiplexování (FDM – Frequency Division Multiplexing),
- Kódové multiplexování (CDM – Code Division Multiplexing),
- Vlnové multiplexování (WDM – Wavelength Division Multiplexing).

Uvedené metody se velmi často používají ve vzájemných kombinacích. Vytvářejí se tak kombinované systémy typu např. FDM-CDM, FDM-TDM. [5]

1.3.1 Časový multiplex – TDM

Princip je založený na tom, že se přenosová cesta rozdělí v čase na "časová okna" (time slots) a ty se napevno přiřadí jednotlivým vstupům. V našem případě se jedná o vstupy A, B, C, jak je vidět na obrázku 2. Během každého time slotu se celá přenosová cesta věnuje jenom přenosu dat z daného vstupu. Celková přenosová kapacita se tak dělí v poměru v jakém jsou rozdělena jednotlivá časová okna. Rozdělení časových oken mezi jednotlivé vstupy nemusí být rovnoměrné. Rozdělení oken mezi jednotlivé vstupy je pevné, nemění se v čase a je dáno předem. Proto nemusí být přenášená data opatřena žádným identifikátorem. Každý vstup má vyhrazenou, pevně danou přenosovou kapacitu. Pokud tuto kapacitu však nevyužije, nemůže být tato kapacita přenechána jinému vstupu.

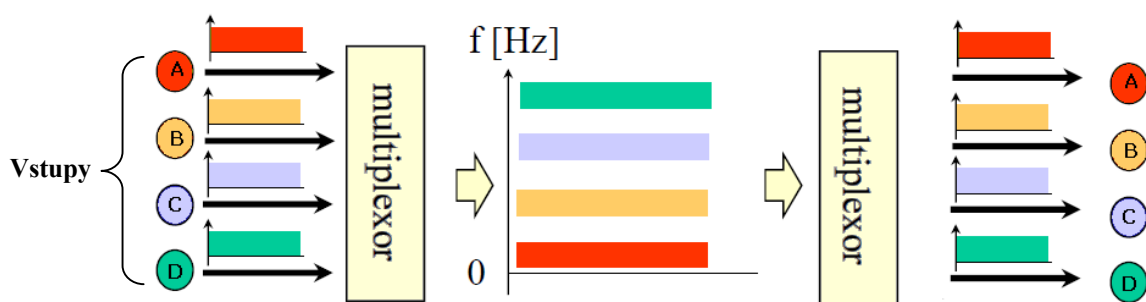


Obr. 2 Princip časového multiplexu

[2]

1.3.2 Frekvenční multiplex – FDM

Realizace je taková, že analogový signál na každém ze vstupních kanálů A, B, C, D je posunut do jiné frekvenční polohy, do jiného rozsahu frekvencí a to tak, aby se žádné tyto polohy vzájemně nepřekrývaly. Pak je možné takto frekvenčně uspořádané signály jednoduše sečíst (sloučit) do jednoho širšího signálu, a ten přenést skrze přenosový kanál. Na druhé straně se pak provede přesně obrácený postup. Jednotlivé dílčí signály se zase od sebe oddělí, a vrátí do původní frekvenční polohy, do svého původního frekvenčního rozsahu. Jak je zobrazeno na obrázku 3.



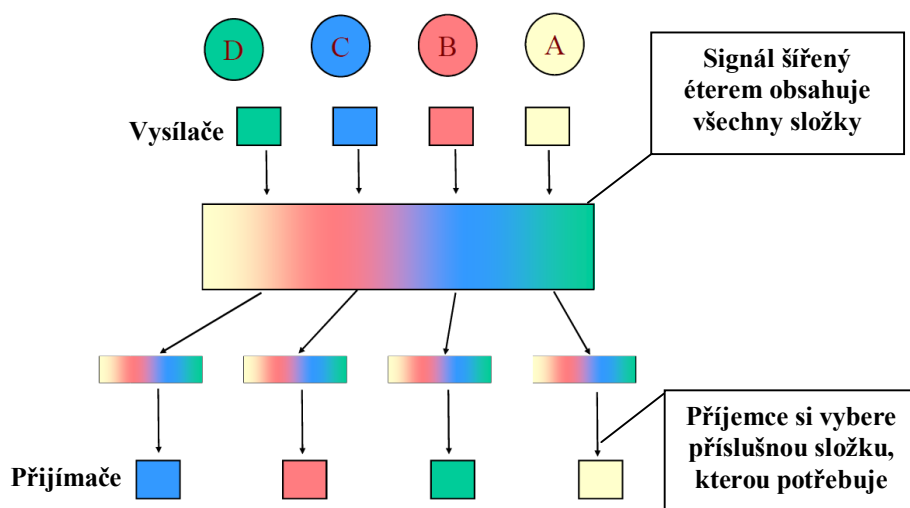
Obr. 3 Princip frekvenčního multiplexu

[2]

V praxi není příliš výhodný frekvenční multiplex právě kvůli tomu, že jde o analogovou techniku. Různé frekvenční posuny, vzájemné slučování a následné oddělování nejsou nikdy ideální, a vždy určitým způsobem znehodnocují přenášený signál.

1.3.3 Kódový multiplex – CDM

Tato technika je maximálně efektivní, využívá celé frekvenční spektrum, nedělí ho. U této metody sdílejí všichni uživatelé stejné kmitočtové pásmo, komunikují ve stejném čase. Vše funguje tak, že každý vysílač svá data nejprve vhodně zakóduje, a pak rovnou vysílá, na stejné frekvenci jako ostatní vysílače. V éteru sice pak dochází ke smíchání všech takto vysílaných signálů, ale je důležité, že toto smíchání je obnovitelné. Každý přijímač má možnost přijímat vysílání od všech vysílačů a je schopen si z toho vybrat (vyextrahovat) právě a pouze ta vysílání, která potřebuje. Představu ilustruje obrázek 4, který znázorňuje různé signály (složky) prostřednictvím barev. [2]



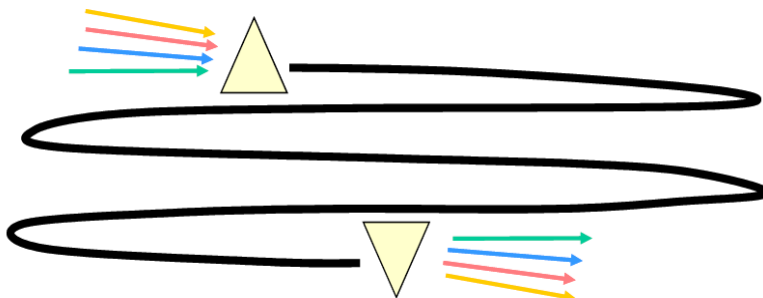
Obr. 4 Princip kódového multiplexu

[2]

1.3.4 Vlnový multiplex – WDM

Metoda se týká vedení světla skrz optická vlákna. Původně nešlo rozlišit jednotlivé barvy světla (světlo o různé frekvenci), dnes je to již možné. Dříve bylo možné použít optické vlákno pouze k jednomu přenosu všemi barvami současně. V dnešní době může již být každá barva přenášena samostatně. Každá barva má určitou vlnovou délku a na ní se mohou přenášet data. Může přenášet samostatná data a tím představovat samostatný přenosový kanál a to dokonce i v opačném směru, optické vlákno se stává obousměrné. Celková přenosová kapacita optického vlákna se tudíž tím násobí. V praxi se používají desítky a stovky barev současně. Vlnový multiplex nám ukazuje obrázek 5. [2]

Předností WDM je možnost transparentního přenosu optických kanálů o rychlostech od 140 Mbit/s až po 10 Gbit/s. S použitím této technologie je tedy možné budovat páteřní síť ATM či IP, ve kterých by jednotlivé uzly byly propojeny vyššími rychlostmi, než nabízí starší technologie SDH. [3]



Obr. 5 Princip vlnového multiplexu

[2]

1.4 Alokace šířky pásma

Alokace = vhodné přidělení omezené šířky pásma přenosovým kanálům. Z předchozí kapitoly je jasné na jakých principech probíhá sdílení šířky pásma mezi uživateli. Pro alokaci šířky pásma je z výše popsaných metod nejvhodnější frekvenční multiplex, jelikož ten jako jediný rozdělí šířku pásma na požadovaný počet menších kanálů, které mají menší šířky pásem. Protože časový i kódový multiplex s šířkou pásma nic nedělají, pouze využívají její kapacitu takovým způsobem, aby ji mohlo využívat více uživatelů.

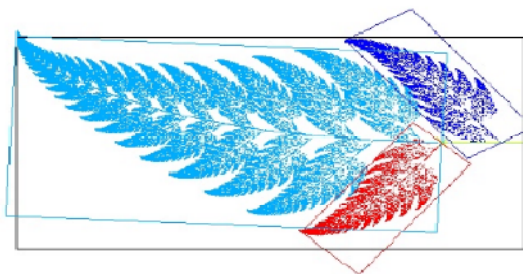
Existují dvě metody alokace šířky pásma, statická a dynamická. U statické alokace šířky pásma probíhá dělení tak, že podle určitých předpokladů je jednomu přenosovému kanálu přiřazena větší část šířky pásma a druhému menší část šířky pásma. Problém nastává v tom, že v některých případech je těžké dopředu odhadnout, který přenosový kanál bude zrovna více vytížen, tedy si zaslouží větší šířku pásma, a který bude zatížen méně, jemuž by stačila přidělit menší šířka pásma. Při chybném odhadu pak dochází k situacím, kdy jeden přenosový kanál je vytížen maximálně, zatímco jiný není vytížen třeba vůbec. A když se na tuto problematiku podíváme z hlediska času, je situace volby šířky pásma pro zvolené kanály ještě složitější. Může totiž nastat situace, že jeden den v určitý čas bude kanál maximálně zatížen a druhý den ve stejný čas bude zatížen jen minimálně, nebo vůbec.

Řešením popsaného problému by mohla být dynamická alokace šířky pásma. Zde však nastává jiný problém, podle čeho tedy přiřazovat šířku pásma? Spíše na základě čeho a jak často ji měnit, když nevidíme do vzdálené budoucnosti a nevíme, jak bude i nadále přenosový kanál zatížen. Možné řešení jak nahlédnout do budoucnosti tu však je. Síťový provoz je možné totiž krátkodobě predikovat (předpovídat). K tomu slouží metody, kterým je věnována následující kapitola. V praxi by však dynamická alokace šířky pásma pro konkrétní uživatele nebyla moc výhodná. Uživatelé jsou totiž schopni přidělenou šířku pásma maximálně využít i při opakovaném navýšení přidělené šířky pásma a jelikož celková šířka pásma, která je sdílena mezi uživateli je přenosovými vlastnostmi použitého média omezena, tak by po určitém čase došlo k jejímu maximálnímu vytížení několika uživateli na úkor ostatních uživatelů, na které by už nezbyla žádná šířka pásma. Dynamická alokace šířky pásma by se však mohla uplatnit na směrovačích v přenosové cestě, které by zajišťovaly potřebnou kvalitu služeb (QoS - Quality of Services) u tzv. reálných aplikací jako je například internetová telefonie, video konference, video telefonie, video na vyžádání a jiné.

[6], [8]

2 SAMOPODOBNOST

Samopodobnost je pojem z oboru fraktální geometrie. Je to charakteristika tvaru objektů. Útvar považujeme za samopodobný, pokud tento útvar vypadá stejně v jakémkoliv měřítku či zvětšení. U dokonalé samopodobnosti platí, že ať pozorujeme samopodobný objekt v jakémkoliv měřítku nebo v jakémkoliv rozlišení, pozorujeme stále se opakující charakteristický tvar. Takovým obecným příkladem samopodobnosti může být například čtverec, tento útvar můžeme rozdělit na konečný počet částí podobajících se původnímu (originálnímu) tvaru. Sítový provoz má také charakteristiky samopodobnosti, protože časový průběh toku dat totiž vykazuje podobné charakteristiky v různých časových měřítkách. Příklad samopodobnosti nám ilustruje obrázek 6. [7]



Obr. 6 Samopodobnost list kapradí

[9]

Samopodobné charakteristiky je velmi obtížné modelovat či predikovat pomocí Markovových, Poissonových a ARMA modelů, které řadíme mezi klasické matematické modely. Za tímto účelem můžeme použít FARIMA model, či neuronové sítě, nebo jej můžeme predikovat pomocí teorie chaosu. Těmto metodám bude věnována kapitola 3. [10]

2.1 Statické vlastnosti sítového provozu

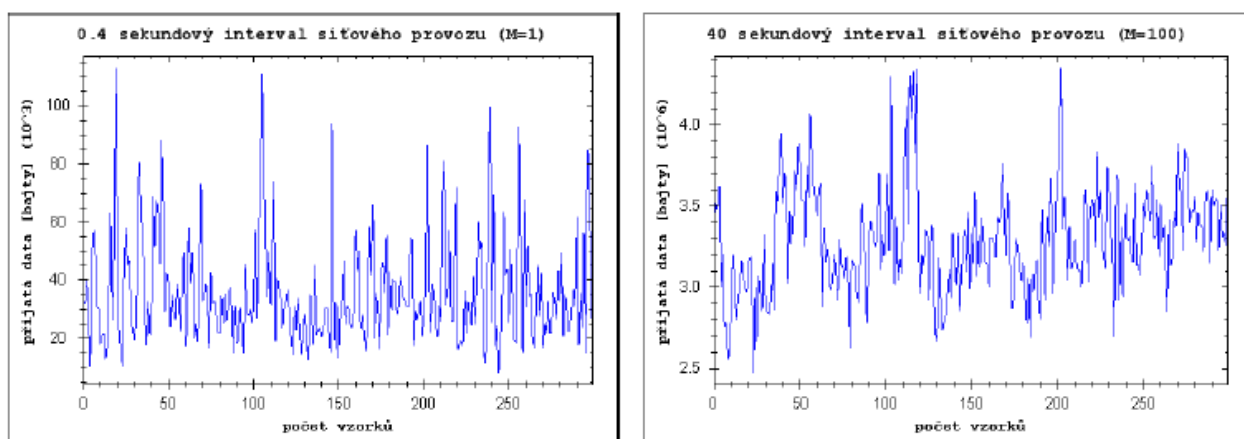
Sítový provoz tzv. (traffic) v ethernetové LAN síti (Local Area Network) je statisticky samopodobný. Shlukující proudy takového provozu typicky nabývají na intenzitě samopodobnosti ("protržení") místo vyhlazování. Tyto studie vlastností sítového provozu jsou podloženy statistickou analýzou vysoké kvality z obrovského množství paketů v ethernetovém provozu sesbíraných v letech 1989 - 1992. Ty jsou samozřejmě spojeny s diskusí o podložení matematických a statistických vlastností samopodobnosti a její vztah s aktuálním chováním sítí. Zde jde také o prezentaci modelu provozu, založeného

na samopodobných náhodných procesech, které jsou jednoduché, přesné a jsou realisticky popsány provozními scénáři s očekávanou dobou trvání.

První krok učinili pánové Wilson a Leland, kteří byli schopni zaznamenat stovky milionů ethernetových paketů beze ztrát a to nezávisle na provozním zatížení, včetně přesně zaznamenaných časových značek. Zachycená data byla v rozmezí srpna 1989 a února 1992 v několika ethernetových LAN sítích. Wilson a Leland prezentovali předběžnou statistickou analýzu těchto jedinečných vysoce kvalitních dat a detailně komentovali momentální situaci "protržení" napříč extrémně širokému časovému měřítku. Provozní "špičky" trvají delší dobu a vytvářejí "zvlnění", které ve své podstatě "roste".

Samopodobnost agregovaného ethernetového LAN provozu je velmi odlišná od konvenčního telefonního provozu a od aktuálních formálních modelů pro pakety provozu například výše zmíněné Markovy, Poissonovy a ARMA modely, či jejich varianty. Samopodobnost vyžaduje nový pohled na modelování provozu a výkonu širokopásmových sítí.

V dnešní době se ukazuje, že je možné prezentovat princip samopodobnosti jak na velkém tak i na lokálním měřítku. Existuje hypotéza vysvětlující možnosti samopodobnosti použitím partikulárních podmnožin široké oblasti provozu. Provoz v oblasti celosvětového webu (World Wide Web), neboli také zkráceně WWW. Při použití rozsáhlých sad přenosu, které představují přes půl milionu žádostí o WWW dokumenty je možné hledat strukturu WWW přenosu. Nejdříve je však třeba dokázat, že WWW provoz má vlastnosti charakteristické pro samopodobnostní modely přenosu. Agregovaný síťový provoz je zobrazen na obrázku 7. [10], [11]



Obr. 7 Agregovaný síťový provoz (stupeň agregace 1 a 100)

[10]

2.2 Matematické vyjádření samopodobnosti

Samopodobná časová řada má tu vlastnost, že pokud je agregována novou řadou, kde každý bod tvoří součet originálních bodů, tak má stejnou autokorelační funkci jako její originál. Představme si kovariační stochastický stacionární proces příchodů paketů vyjádřený časovou řadou $X = (X_t : t = 0, 1, 2, \dots)$, která má směrodatnou odchylku $\sigma^2 = \text{Var}(X_t)$ a střední hodnotu $\mu = E(X_t)$. Kde (X_t) je velikost příchozích paketů v t -ém časovém slotu. Dále je definována autokorelační funkcí $\gamma(k) = \text{cov}(X_t, X(t+n))/\sigma^2$, pro $k = 0, 1, 2, 3, \dots$.

Zprůměrováním vzorků původní série X lze získat novou kovarianční stacionární časovou posloupnost, která je dána vztahem:

$$X^{(m)} = (X_k^{(m)} : k = 1, 2, 3, \dots), \quad (4)$$

kde $m = 1, 2, 3, \dots$.

Nově vzniklá kovariační stacionární časová posloupnost, která představuje součet vzorků originální řady, nepřesahující velikost m . Pro $m = 1, 2, 3, \dots$, je X_k dáno vztahem:

$$X_k^{(m)} = \frac{1}{m} (X_{km-m+1} + \dots + X_{km}), \quad k \geq 1 \quad (5)$$

Časová posloupnost $X^{(m)}$ pro všechny $m = 1, 2, 3, \dots$ je stejný kovarianční stacionární stochastický proces jako časová posloupnost X . Pro výpočet střední hodnoty, směrodatné odchylky a autokorelační funkce je možné použít následující vztahy:

$$E(X^{(m)}) = \mu, \quad (6)$$

$$\text{VAR}(X^{(m)}) = \frac{\sigma}{m} + \frac{2\sigma}{m^2} (m-j)\gamma(j), \quad (7)$$

$$\gamma^{(m)}(k) = \frac{\sigma^2}{m^2 \text{VAR}(X^{(m)})} \left\{ m\gamma(m \cdot k) + 2 \sum_{j=1}^{m-1} (m-j)\gamma(m \cdot k + j) \right\}. \quad (8)$$

Pokud samopodobná struktura $X_{(m)}$ je stejná jako u X pak platí rovnice (9) a (10):

$$\gamma^{(m)}(k) = \gamma(k) , \quad (9)$$

$$VAR(X^{(m)}) = \gamma^2 m^{-\beta} . \quad (10)$$

V takovémto případě nazýváme tento proces jako samopodobnost II. řádu s Hurstovým parametrem:

$$H = 1 - \frac{\beta}{2} . \quad (11)$$

Hurstovým parametrem určíme míru samopodobnosti časové řady. Když je Hurstův parametr $H = 0,5$, tak mají procesy malou nezávislost na měřítku. U samopodobných procesů pro H platí $0,5 < H < 1$. U síťového provozu bylo zjištěno, že $H > 0,6$. Nejvýznamnější metody pro výpočet Hurstova parametru jsou Variance Time Plot (VPT) analýza, Rescaled Range (R/S) analýza, Metoda absolutní hodnoty, Metoda rozptylu, Periodogram metoda a Abry – Weitch metoda.

Kdyby bylo m dostatečně velké, tak je možné definovat asymptotickou samopodobnost, která se v teorii síťového provozu velmi často využívá. Pro tento případ je samopodobnost definována dle následujícího vztahu: [12], [13]

$$VAR(X^{(m)}) \approx c \cdot m^{-\beta} , \quad (12)$$

kde $m \rightarrow \infty$, $c = \text{konst.}$

3 MOŽNOSTI PREDIKCE SÍŤOVÉHO PROVOZU

Síťový provoz je velmi obtížné modelovat či predikovat pomocí Markových, Poissonových a ARMA modelů, které řadíme mezi klasické matematické modely. Existují metody (modely) podle kterých můžeme síťový provoz predikovat. A patří sem zejména FARIMA model, neuronové sítě, nebo jej můžeme predikovat pomocí teorie chaosu. Jednotlivé metody jsou popsány v následujících podkapitolách.

3.1 FARIMA model

FARIMA model (Fractional Autoregressive Intergrated Moving Average), který vychází z obecnějšího ARIMA modelu. Skládá se z autoregresního procesu a klouzavého průměru. FARIMA proces patří do skupiny velkorozsahových samopodobných procesů. Nevýhodou tohoto modelu je neschopnost zachytit nelineární síťový provoz. Dále je zde obtížná volba počátečních parametrů, na které závisí kvalita predikce. Tato volba je zásadní a velmi obtížná. FARIMA (p, d, q) model, kde p nám udává autoregresivní řád, d představuje rozdílový řád a q zastupuje řád klouzavého průměru je reprezentován vztahem: [14]

$$\Phi(B)\Delta^d X_t = \Theta(B)a_t, \quad (13)$$

kde $\Phi(B)$ je komplexní proměnná představující autoregresivní proces, který odpovídá rovnici:

$$\Phi(B) = 1 - \Phi_1 B - \Phi_2 B^2 - \dots - \Phi_p B^p, \quad (14)$$

kde $\Theta(B)$ je komplexní proměnná klouzavého poměru:

$$\Theta(B) = 1 - \Theta_1 B - \Theta_2 B^2 - \dots - \Theta_q B^q, \quad (15)$$

B je zpětnovazební operátor pro který platí $BX_t = X_{t-1}$, $B^2 X_t = X_{t-2}$, a_t je sekvenční bílého šumu ($a_t : t = \dots, -1, 0, 1, \dots$) s nulovou průměrnou hodnotou a rozptylem, který je rovný σ^2 . Δ^d je rozdílová část:

$$\Delta^d = (1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k, \quad (16)$$

$$\binom{d}{k} = \frac{\Gamma(d+1)}{\Gamma(d-k+1)\Gamma(k+1)}, \quad (17)$$

kde Γ je definováno jako $\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt$, $x > 0$.

U modelů ARIMA i FARIMA je rovnice (13) totožná. Jediný rozdíl v nich je a to takový, že pro každý model nabývá diferenciální parametr d jiných hodnot. U ARIMA modelu může nabývat pouze celočíselných hodnot. Pokud $d = 0$, tak se jedná o klasický ARIMA model. U FARIMA modelu se hodnoty parametru d mohou pohybovat v rozmezí $0 < d < 0,5$. Poté je hodnota Hurstova parametru rovna: [14]

$$H = d + \frac{1}{2} . \quad (18)$$

3.2 Teorie chaosu

Další možnou metodou je predikce pomocí Teorie chaosu, která využívá hlavně rysu samopodobnosti síťového provozu. Vlastnosti provozu lze zjistit již ze zadané časové řady a není tak potřeba vytvářet model, jako tomu bylo u předchozí FARIMA metody. To má za následek zkvalitnění predikce, zejména přesnost a spolehlivost.

Tato teorie popisuje chování nelineárních systémů, které mají skrytý řád, ale přesto se jeví jako systémy řízené náhodnými jevy. Tyto systémy jsou velmi citlivé na počáteční podmínky, malá změna v současnosti může představovat velkou změnu v budoucnosti. V důsledku této citlivosti se chování těchto fyzikálních systémů, vykazujících chaos, jeví jako náhodné, i když model systému je deterministický v tom smyslu, že je dobře definovaný a neobsahuje žádné náhodné parametry.

Systémy, které vykazují deterministický chaos, jsou v jistém smyslu složitě uspořádané. Tím je význam slova v matematice a fyzice v jistém nesouladu s obvyklým chápáním slova chaos jako totálního nepořádku.

Základy teorie chaosu

Nelineární dynamický systém může obecně vykazovat jeden z následujících typů chování, které jsou:

- Vždy v klidu,
- Vždy expandující,
- Periodický pohyb,
- Kvaziperiodický pohyb,
- Chaotický pohyb.

Typ chování, jaký systém může vykazovat, závisí na počátečním stavu systému a hodnotách jeho parametrů, jestliže nějaké existují. Nejsložitějším typem chování je chaotický pohyb (neperiodický složitý pohyb), který pojmenoval tuto teorii.

Chaotický pohyb

Abychom mohli považovat chování systému za chaotické, musí systém vykazovat následující vlastnosti:

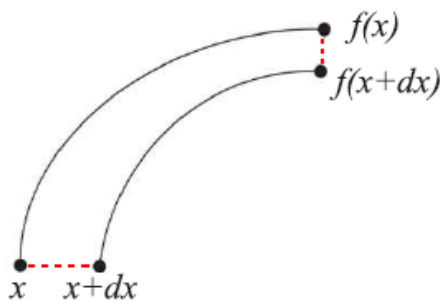
- Musí být citlivý na počáteční podmínky,
- Musí být topologicky tranzitivní,
- Musí mít husté periodické orbity.

Chaotické časové řady můžeme vizualizovat pomocí abstraktního prostoru stavů, který se nazývá fázový prostor. V tomto prostoru každá osa představuje jednu dimenzi stavů a čas je zde implicitní. Po určitém čase vyvíjení systému vznikne ve fázovém prostoru křivka, kde tato křivka po dostatečně dlouhé době začne zvýrazňovat strukturu, které se nazývá atraktor. Atraktor je konečný stav nějakého systému.

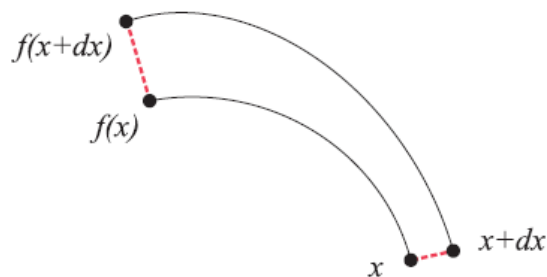
Vlastností chaotických atraktorů je jejich veliká citlivost na počáteční podmínky. Citlivost k počátečním podmínkám znamená, že dvě blízké trajektorie ve fázovém prostoru se s rostoucím časem rozbíhají (exponenciálně). Jinak řečeno, malá změna v počátečních podmínkách vede po čase k velmi odlišnému výsledku. Systém se chová identicky pouze, když je jeho počáteční konfigurace úplně stejná. Stupeň chaotičnosti lze číselně vyjádřit pomocí parametru, který se jmenuje Lyapunovův exponent. Tento exponent je základním nástrojem pro popis dynamického systému.

Výpočet exponentu je poměrně obtížně numericky zvládnutelný, získané informace však tento problémový úkon vyvažují. Lyapunovy exponenty totiž svými hodnotami přímo popisují, jak se systém chová. Jiné způsoby popisu systému jsou ještě více náročnější.

Lyapunovův exponent vyjadřuje, zda blízké dráhy atraktoru k sobě konvergují, nebo divergují. Pro každý systém existuje několik Lyapunových exponentů, přesněji řečeno právě tolik kolik má dimenzí. Chování systému nejvíce ovlivňuje právě ten největší exponent ze všech Lyapunových exponentů. Na obrázku 8 jsou vidět konvergující dráhy atraktoru a naopak na obrázku 9 jsou zobrazeny divergující dráhy atraktoru. [15]



Obr. 8 Konvergující dráhy atraktoru



Obr. 9 Divergující dráhy atraktoru

Když je Lyapunovův exponent záporný, tak dráhy v čase konvergují, jak je vidět na obrázku 8 a dynamický systém není citlivý na počáteční podmínky. Jestliže je daný exponent kladný, pak dráhy atraktoru mezi sebou divergují a daný systém vykazuje citlivost vůči počátečním podmínkám. U chaotického systému se musí vždy alespoň v jednom případě trajektorie drah atraktoru exponenciálně vzdalovat (divergovat). Musí mít tedy alespoň jeden Lyapunovův exponent kladný. Příklad divergujících drah je ukázán na obrázku 9.

Lyapunovův exponent můžeme vypočítat díky Wolfova algoritmu. Jehož princip spočívá v tom, že se zvolí několik blízkých bodů, které se nechají v čase rozvíjet a při tom je sledována rychlost růstu jejich vzájemné vzdálenosti.

Za pomoci Lyapunova exponentu můžeme určit horizont predikce, tedy maximum budoucích bodů, které je možné predikovat. Tento horizont lze vypočítat dle následující rovnice:

$$T_{max} = \frac{1}{\lambda_{max}} \ln \frac{\Delta}{\delta_0}, \quad (19)$$

kde Δ je požadovaná maximální chyba. λ_{max} je maximální Lyapunovův exponent a δ_0 je neurčitost v měření počátečních podmínek. [15]

3.2.1 Matematické vyjádření Lyapunova exponentu

Máme daný rekonstruovaný fázový prostor a pro jeho bod X_i je potřeba najít jeho nejbližšího souseda X_{near} , který splňuje následující podmínku: [7]

$$\|X_{near} - X_i\| = \min \|X_j - X_i\|, \quad (20)$$

kde $(d - 1)\tau < |j - i| < R(d - 1)\tau$ d nám představuje dimenzi vložení, τ je časové zpoždění a R reprezentuje heuristickou hranici, pro kterou platí $1 < R < 10$.

Nyní můžeme vypočítat Lyapunovův exponent $\lambda(i)$:

$$\lambda(i) = \frac{1}{jh\delta_t} \log \frac{\|X_{near+jh} - X_{i+jh}\|}{\|X_{near} - X_i\|} = \max_h \left\{ \frac{1}{h\Delta t} \frac{\|X_{near+h} - X_{i+h}\|}{\|X_{near} - X_i\|} \right\}, \quad (21)$$

kde Δt je vzorkovací perioda časové řady. Hodnota parametru h musí splňovat $1 < h < (d - 1)\tau$.

Konečná hodnota Lyapunova exponentu se vypočítá pomocí vztahu:

$$\lambda = \frac{1}{N} \sum_{i=1}^N \lambda(i). \quad (22)$$

3.3 Neuronové sítě

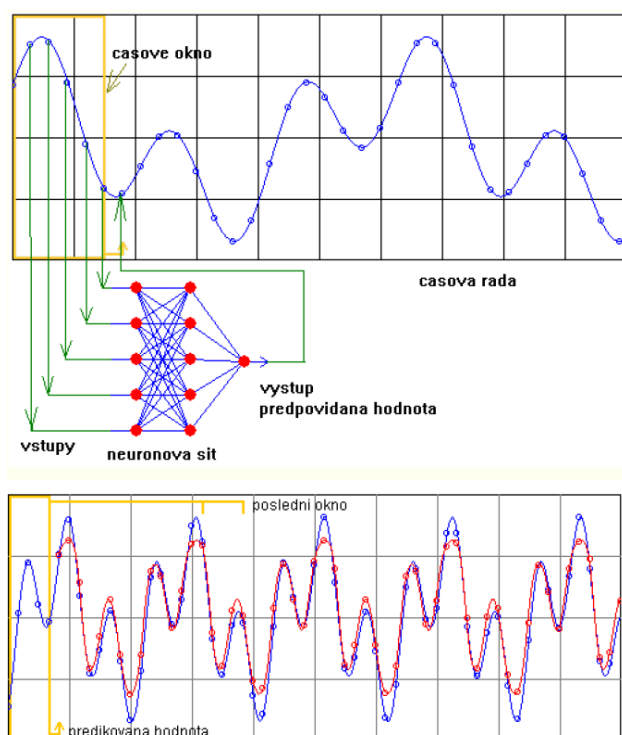
Další způsob jak predikovat síťový provoz je za pomoci neuronových sítí. Touto problematikou se bude nadále zabývat tato práce.

Podívejme se na neuronovou síť jako na černou skříňku, která má dva konce. Na prvním konci vkládáme příklady a na druhém jejich řešení. Pokud to děláme v jistém smyslu výchovně, tak je naše skříňka schopna do sebe vstřebat předkládané znalosti. To má za následek, že po jistém učícím čase, sama na vstupní příklad vygeneruje správný výsledek. Tuto vlastnost sítě nazýváme učící schopnost. Možnosti neuronové sítě zde však nekončí. Pokud síti zadáme příklad, který ale nebyl v množině učících příkladů, neuronová síť je přesto schopna vygenerovat správnou odpověď. Tento jev se jmenuje generalizace. Neuronová síť může pak mít abstraktní chování. Výzkum prokázal, že tyto dvě schopnosti sítě jdou částečně proti sobě. Neuronová síť, která je schopna pamatovat si velké množství příkladů přesně, má malou schopnost generalizace (zobecnování) a opačně.

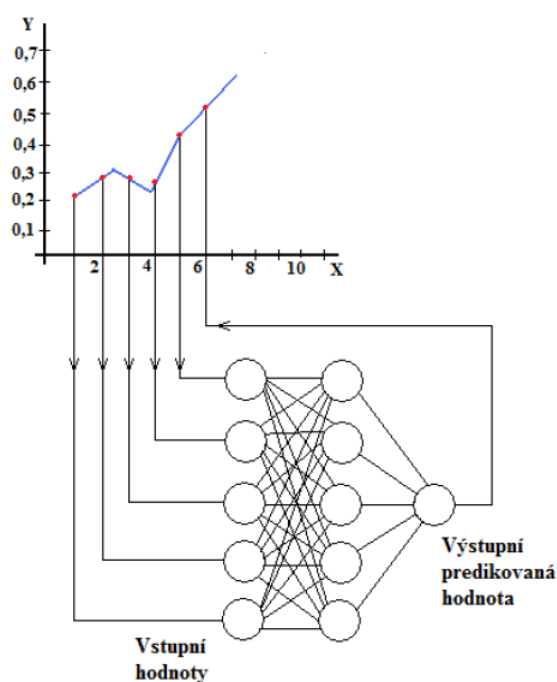
Zatímco dnešní počítače mají velmi sofistikované architektury, kde každá jednotka plní specializované funkce, neuronová síť se skládá z mnoha jednoduchých jednotek (neuronů), které jsou mezi sebou vzájemně propojeny spoji, které jsou ohodnoceny váhami.

Míra a struktura tohoto propojení se nazývá architekturou neuronové sítě. Jedno z paradigmat neuronové vědy dokonce tvrdí, že veškerá síla neuronových sítí spočívá v množství a kvalitě vzájemných vazeb neuronů. Stejnou roli jako architektura hraje také vhodný typ neuronu.

Byla by velká chyba, myslet si, že neuronová síť je schopna samostatného myšlení. Přesto však výsledky, kterých může tato síť dosáhnout, jsou často překvapivé i pro samotné vědce a badatele. Toto je jedna z nepřímých indicií, že se výzkum lidského myšlení, z něhož studium neuronových sítí vzešlo, ubírá správným směrem. Na obrázcích 10 a 11 je znázorněn obecný princip predikce pomocí neuronových sítí. [16]



Obr. 11 Realizace predikce pomocí neuronové sítě [17]



Obr. 10 Realizace predikce pomocí neuronové sítě [17]

4 ZÁKLADNÍ VLASTNOSTI NEURONOVÉ SÍTĚ

Neuronové sítě jsou jednou z oblastí, které se v poslední době věnuje zvýšená pozornost. To je zapříčiněno hlavně tím, že mají mnoho vlastností, které je možno s úspěchem využít v řadě oborů. Specifické vlastnosti a parametry těchto sítí si přiblížíme v následující kapitole. Bude zde popsán biologický neuron, obecné schéma umělého neuronu a následně zde budou zmíněny aktivační funkce.

Specifické vlastnosti neuronových sítí lze zjednodušeně ve zkratce shrnout v následujících bodech:

- Jsou inspirovány biologickými neuronovými sítěmi. Tuto vlastnost určitým způsobem předurčuje, že uměle vytvořené neuronové sítě by měly být schopny, z hlediska základních principů, chovat se stejně nebo alespoň podobně jako jejich biologické vzory. Je patrné, že vytvoření umělého lidského mozku se všemi jeho schopnostmi je velmi těžce řešitelná věc ať už z hlediska kvantity neuronů, jejich způsobu propojení či chování jednotlivých typů neuronů atd. Nicméně je tu možnost simulovat alespoň některé funkce lidského myšlení a tyto pak implementovat.
- Využívají distribuované, paralelní zpracování informace při provádění výpočtů. Zjednodušeně řečeno ukládání, zpracování a předávání informace probíhá prostřednictvím celé neuronové sítě, než pomocí určitých paměťových míst. Paměť a zpracování informace v neuronové síti je tedy ve své přirozené podstatě spíše globální než lokální.
- Informace jsou ukládány především prostřednictvím síly vazeb mezi jednotlivými neurony. Vazby mezi neurony vedoucí ke správné odpovědi jsou posilovány. A naopak vazby vedoucí k špatné odpovědi jsou oslabovány pomocí opakované expozice příkladů popisujících problémový prostor.
- Základní a podstatná vlastnost neuronových sítí je učení. Když jsme doposud veškeré své úsilí při tvorbě uživatelských programů soustředili na vytvoření algoritmů, které transformují vstupní množinu dat na množinu dat výstupních, pak při použití neuronových sítí tato náročná operace odpadá. Způsobem jakým se budou vstupní data

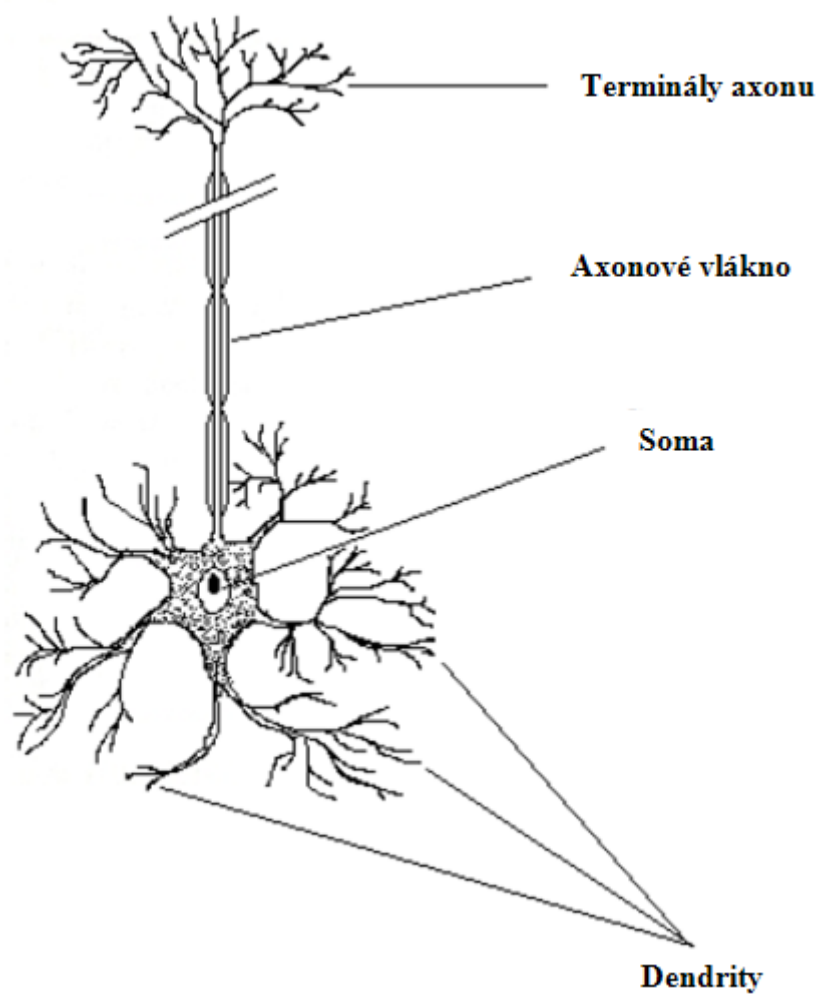
transformovat na data výstupní určuje právě fáze učení založená na již dříve uvedené expozici příkladů popisující řešenou problematiku, této expozici se také říká trénovací množina. Odpadá tedy operace algoritmizace úlohy, které je nahrazena předložením trénovací množiny neuronové síti a jejím učením. [18]

4.1 Biologický neuron

Jen velmi těžko lze zastírat pravý účel vzniku neuronových sítí. Od začátku se vědci snažili modelovat činnost lidského mozku. Proto se obraceli k fyziologii a neurologii. Nastiňme si nyní celou problematiku z pohledu neurofyziologa.

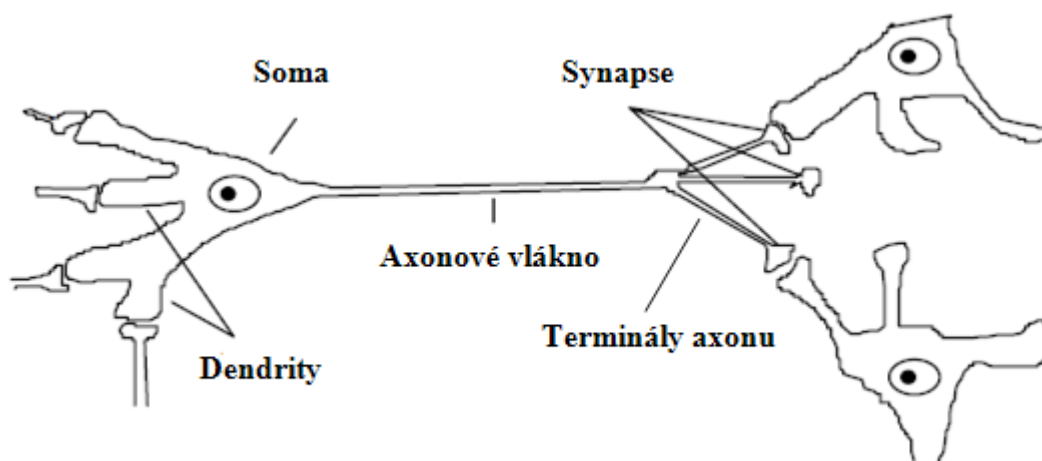
Nervová soustava člověka je složena z několika složek. Nejdůležitějšími z nich jsou neurony a glie. Funkce glií není dodnes plně prozkoumána, ale všeobecně se ví, že tvoří jakýsi podpůrný mechanismus celé nervové soustavy. Zájem o glie stále více roste a zdá se, že jejich úloha je mnohem větší, než se dříve mínilo. My se dále však o nich už zmiňovat nebudeme. Podstatnější úlohu v našich modelech budou hrát samotné neurony. Jen v samotném mozku jich je okolo 100 miliard (10^{11}).

Na popisu biologického neuronu, jakožto základního stavebního kamene nervové soustavy, si můžeme všimnout jeho stavby, uzpůsobené pro přenos signálů. Neuron se skládá z těla, které se nazývá soma, dále pak výstupního axonu a vstupních dendritů. Axon je zakončen terminály, které jej spojují s dendrity ostatních neuronů. Toto uspořádání v podstatě tvoří přenosové rozhraní. Takto může být neuron pospojován až s dalšími 5 000 neurony. Vrstva mezi terminály a dendrity se jmenuje synapse. Propustnosti synapse ovlivňuje přenos vzruchu od terminálů k následným dendritům. Schopnost synaptické vrstvy přenášet vzruch nervovou soustavou nazýváme excitací a naopak tlumící schopnost inhibicí. Šíření vzruchu v nervové soustavě probíhá následujícím způsobem. Dendrity neuronu jsou podrážděny a neuron pak po dosažení určitého stupně podráždění sám produkuje vzruch, který předává terminály axonu přes synaptickou vrstvu na dendrity dalších neuronů atd. Právě míra synaptické propustnosti, tj. schopnosti předávat vzruch, hraje významnou roli při vytváření paměťových stop. Její změnou je vysvětlována schopnost učení. Zjednodušené schéma neuronu je vidět na obrázku 12. Na dalším obrázku 13 je pak ukázka propojení více neuronů. [16]



Obr. 12 Biologický neuron

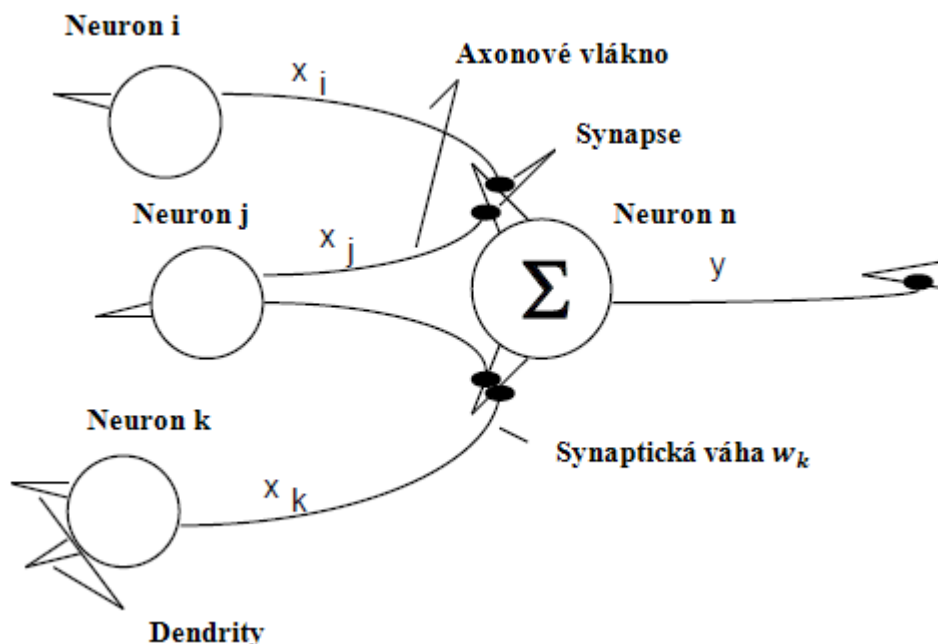
[16]



Obr. 13 Spojení neuronů

[19]

Výše uvedené spojení více neuronů mezi sebou můžeme pak vyjádřit schematicky takto, jak je vidět na obrázku 14,



Obr. 14 Schématické propojení neuronů

[19]

kde x_i, x_j, x_k jsou výstupní signály neuronů i, j, k

w_i, w_j, w_k jsou synaptické váhy upravující výstupní signál neuronů i, j, k

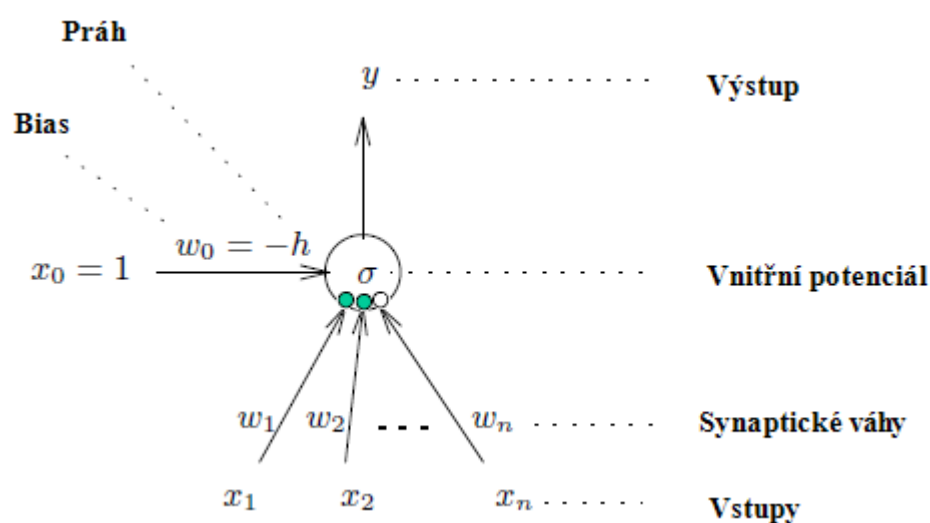
y je výstupní excitační signál neuronu n

4.2 Formální neuron

Jestliže vyjádříme biologický neuron matematickým modelem, dostáváme tak formální neuron, který je zobrazen na obrázku 15. Neuron sčítá vstupní podněty x_i (u biologického neuronu by to odpovídalo dendritům) vynásobené (synaptickými) vahami w_i . Pokud tento součet přesáhne určitou mez (práh) vystřelí, tj. vrátí na výstup y hodnotu 1. V opačném případě vrátí hodnotu 0. Matematicky bychom mohli tento jev zapsat takto:

$$y = \sigma \left(\sum_{i=1}^n w_i x_i + w_0 \right), \quad (22)$$

kde x_i jsou hodnoty vstupů a w_i váhy. Prahová hodnota se označuje w_0 a určuje nám při jaké hodnotě vážené sumy má neuron vystřelit. Funkce σ je aktivační funkce. O této funkci budeme podrobněji hovořit v následující podkapitole.



Obr. 15 Formální neuron

[16]

Vstupy x_i : 0/1 (nepřichází vzruch / přichází vzruch)

Váhy w_i : +1, -1 (● excitační spoj přispívá +1, ○ inhibiční spoj-1)

Výstup y : 0/1

4.3 Aktivační funkce

Jedná se o matematickou funkci, která převádí vstupní hodnoty (pod prahem aktivační funkce) na relativně nízké výstupní hodnoty a vstupní hodnoty nad prahem na poměrně vysoké výstupní hodnoty. Je to prahová funkce. Hodnota prahu pak určuje, jestli je neuron

v aktivním (aktivační funkce je větší než prahová hodnota), či pasívním stavu (aktivační funkce je menší než prahová hodnota).

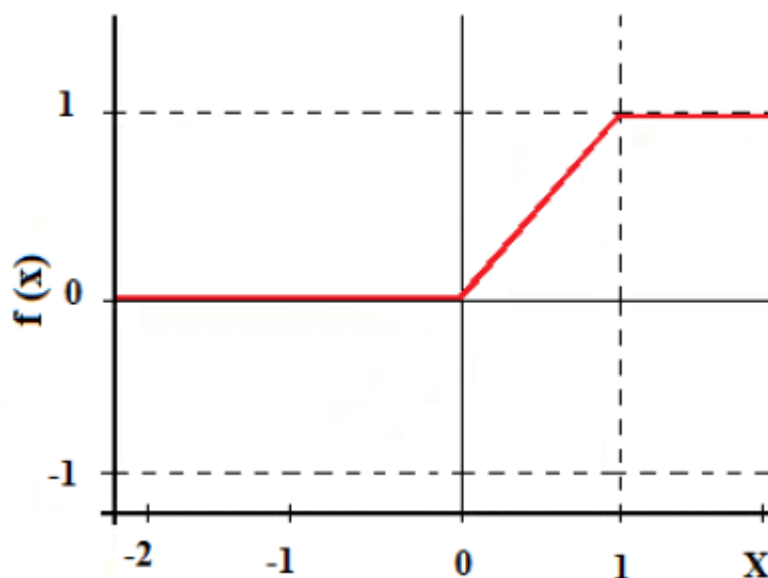
Jako aktivační funkce v jednovrstvém perceptronu je použita funkce $f = \text{sign}(x)$. Funkce signum, ale není diferencovatelná, a proto se u vícevrstvých sítí používají jiné funkce, které se jí blíží svým průběhem a diferencovatelné jsou. Aktivační funkce můžeme definovat různými způsoby a ty pak ani nemusí odpovídat biologické formě neuronu. Nejvýznamnější funkce jsou:

4.3.1 Lineární satureovaná funkce

Tato funkce omezuje velikost vstupního signálu v rozsahu 0 až +1. Kopíruje vstupní hodnoty na výstupu, pouze při záporných hodnotách vstupu je tato funkce nulová. V matematickém zápisu ji můžeme vyjádřit:

$$f(x) = \begin{cases} 1 & (\text{pokud } x > 1) \\ x & (\text{pokud } 0 \leq x \leq 1) \\ 0 & (\text{pokud } x < 0) \end{cases} \quad (23)$$

Tuto funkci můžeme vyjádřit grafem, jak je vidět na obrázku 16.



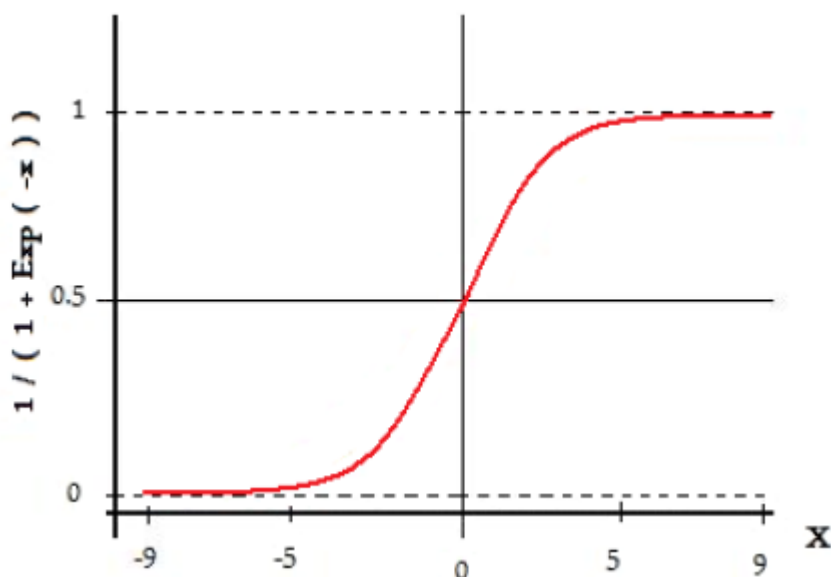
Obr. 16 Průběh Lineární saturační funkce neuronu

4.3.2 Standardní sinoida

Funkční hodnoty se blíží k nule v minus nekonečnu ($-\infty$) a k jedničce v plus nekonečnu ($+\infty$). Jestliže je vstup nula, náleží mu výstupní hodnota 0,5. Výhodou sigmoidální funkce oproti skokové, je výskyt přijatelné derivace v každém bodě. V matematickém zápisu ji můžeme vyjádřit:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (24)$$

Tuto funkci můžeme vyjádřit grafem, jak je vidět na obrázku 17.



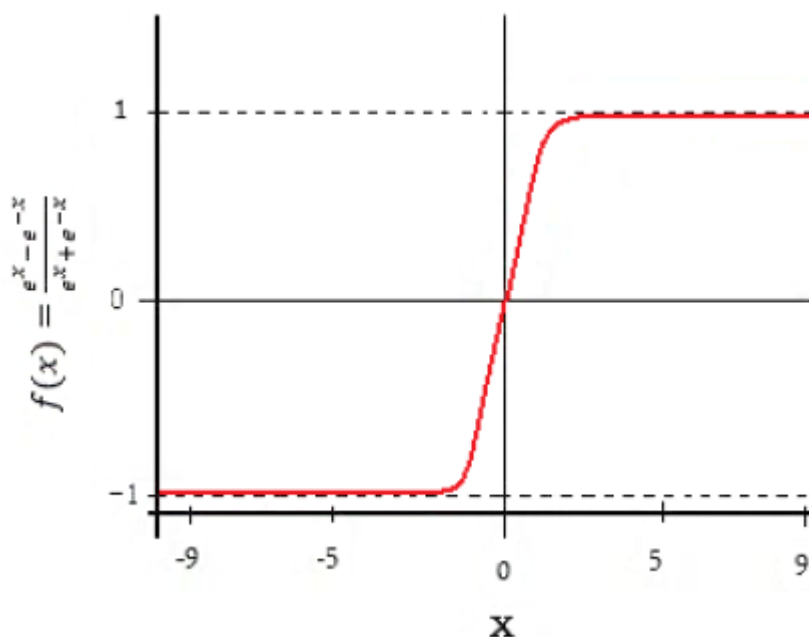
Obr. 17 Průběh sigmoidální aktivační funkce neuronu

4.3.3 Hyperbolický tangens

Pro minus nekonečno ($-\infty$) se blíží k minus jedničce, a naopak pro plus nekonečno ($+\infty$) se blíží k plus jedničce. V nule nabývá rovněž nulové hodnoty. V matematickém zápisu ji můžeme vyjádřit:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (24)$$

Tuto funkci můžeme vyjádřit grafem, jak je vidět na obrázku 18.



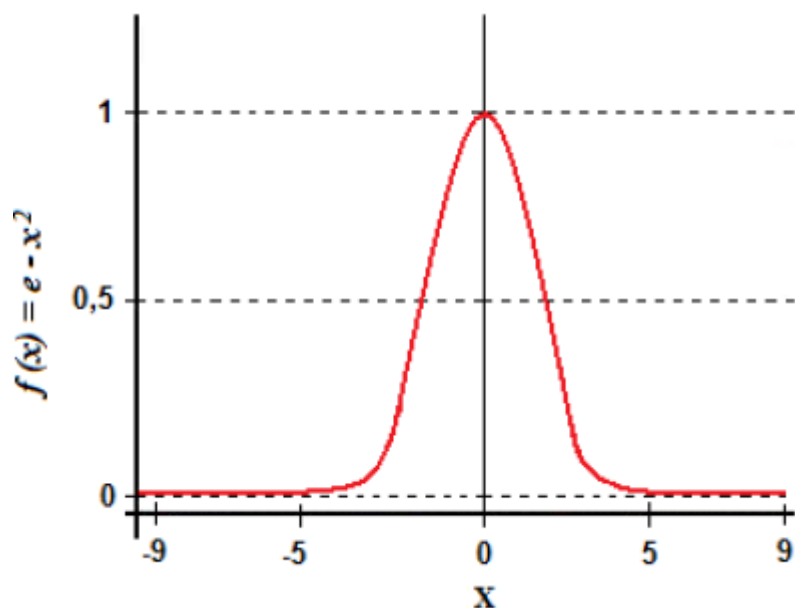
Obr. 18 Průběh aktivační funkce hyperbolické tangenty neuronu

4.3.4 Funkce radiální báze

V minus nekonečnu ($-\infty$) se hodnoty blíží k nule. Stejně je tomu i v plus nekonečnu ($+\infty$). V nule nabývá funkce hodnoty 1. V matematickém zápisu ji můžeme vyjádřit:

$$f(x) = e^{-x^2} \quad (25)$$

Tuto funkci můžeme vyjádřit grafem, jak je vidět na obrázku 19.



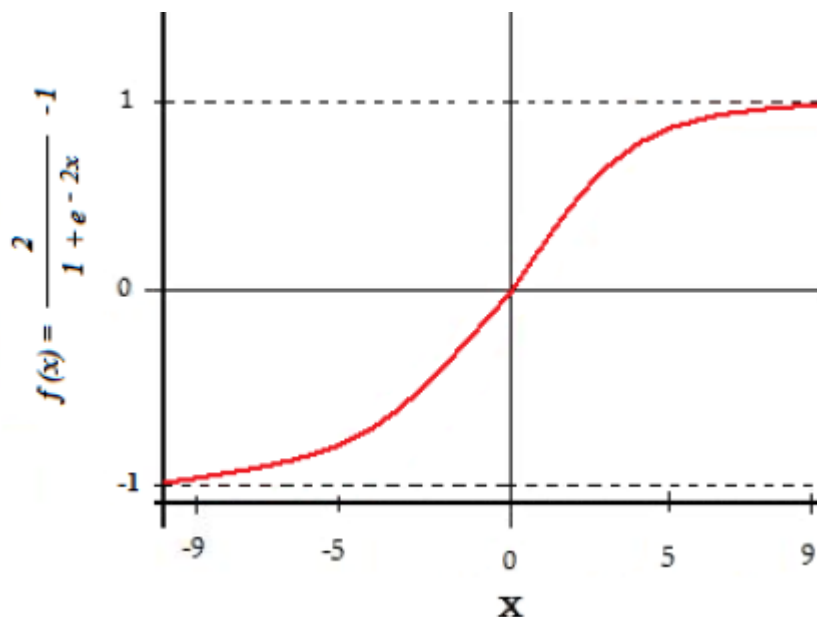
Obr. 19 Průběh aktivační funkce radiální báze neuronu

4.3.5 Tan-sigmoidální funkce

Je to nejpoužívanější aktivační funkce v oboru neuronových sítí. Vznikla kombinací funkcí tangens a sinoidy. V mínus nekonečnu ($-\infty$) se její hodnoty blíží k minus jedné, naopak v plus nekonečnu ($+\infty$) k plus jedné. Zkráceně se této funkci říká tansig. V matematickém zápisu ji můžeme vyjádřit:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (26)$$

Tuto funkci můžeme vyjádřit grafem, jak je vidět na obrázku 19.



Obr. 20 Průběh tan-sigmoidální aktivační funkce neuronu

Často býváme v praxi nuceni řešit problém, musíme matematicky popsat daný proces s dostatečnou přesností. Pokud se sestavení modelu podaří, tak je třeba brát ohledy na náročnost algoritmizace. Ať už časovou, či programovou.

5 NEURONOVÉ SÍTĚ VHODNÉ PRO PREDIKCI SÍŤOVÉHO PROVOZU

Existuje široká škála neuronových sítí, ale ne však každá síť je vhodná kvůli své topologii, specifickým vlastnostem, způsobu učení atd., právě na predikci samopodobných charakteristik, kterými se vyznačuje síťový provoz. Pro predikci síťového provozu jsou vhodné tzv. Rekurentní sítě a patří sem zejména Elmanova síť, Obecná rekurentní síť (LRN) a NARX síť. V následujících podkapitolách se na tyto sítě podíváme detailněji.

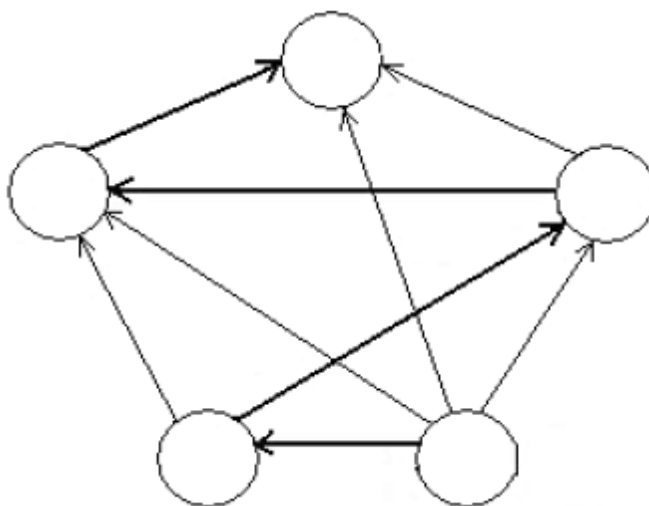
5.1 Základní rozdělení neuronových sítí

Z hlediska způsobu toku dat sítí rozdělujeme neuronové sítě na dvě základní skupiny a to na Dopředné sítě (Feed-forward networks) a Rekurentní sítě (Recurrent networks).

5.2 Dopředné síť

V tomto typu uspořádání sítí jde tok dat striktně od vstupu k výstupu. Data postupují v síti po vrstvách, nikdy síť však neobsahuje zpětnou vazbu. U dopředné neuronové sítě neexistuje cyklus. Uspořádání neuronů je do vrstev a vytváří následující schéma, jak je vidět na obrázku 21. Cesty mezi neurony vedou pouze z nižších vrstev do vyšších. Je zde možné ale také přeskakování vrstev.

Rekurentní síť zpětné vazby, nebo smyčky obsahují. Velmi důležité je zde také dynamické chování sítě. Může dojít k takovému případu, že výstup je do jisté doby neměnný a stabilní, protože přechodový stav trval krátkou dobu. U typů rekurentních sítí je tomu jinak, a výstup lze popisovat jako dynamický děj.



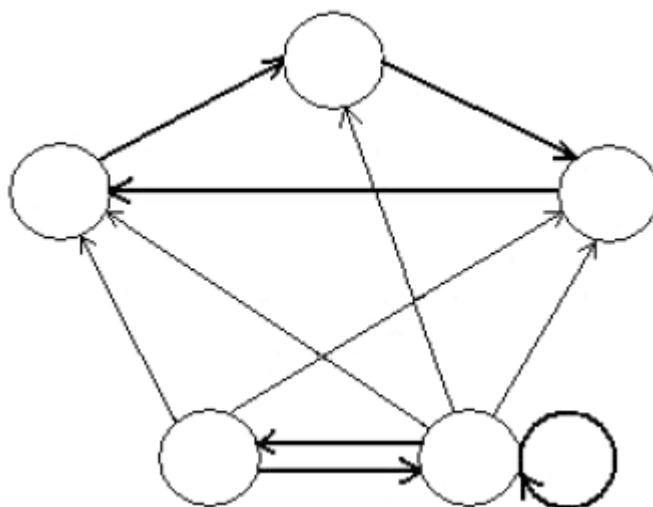
Obr. 21 Dopředná topologie neuronové sítě

5.3 Rekurentní síť

Rekurentní síť (Recurrent networks) obsahují zpětné vazby z výstupních bodů zpět na vstup sítě či vrstvy. U této skupiny sítí výstupní data nezávisí jen na vstupních datech, ale také na stavu, ve kterém se systém nachází. Tyto zpětné vazby způsobují dynamické chování sítě.

U architektury této sítě, existuje množina neuronů v síti, která je propojená v kruhu, řetězcově za sebou. Přičemž výstup posledního neuronu z této skupiny představuje vstup prvního neuronu. Dohromady toto uspořádání tvoří tzv. cyklus. Zvláštním případem je úplná topologie neuronové rekurentní sítě. Kdy výstup libovolného neuronu je vstupem každého neuronu (představuje největší možný počet cyklů). Nejjednodušším případem je zpětná vazba

neuronu (jehož výstup je zároveň vstupem). Architektura rekurentní sítě je vidět na obrázku 22. [19]



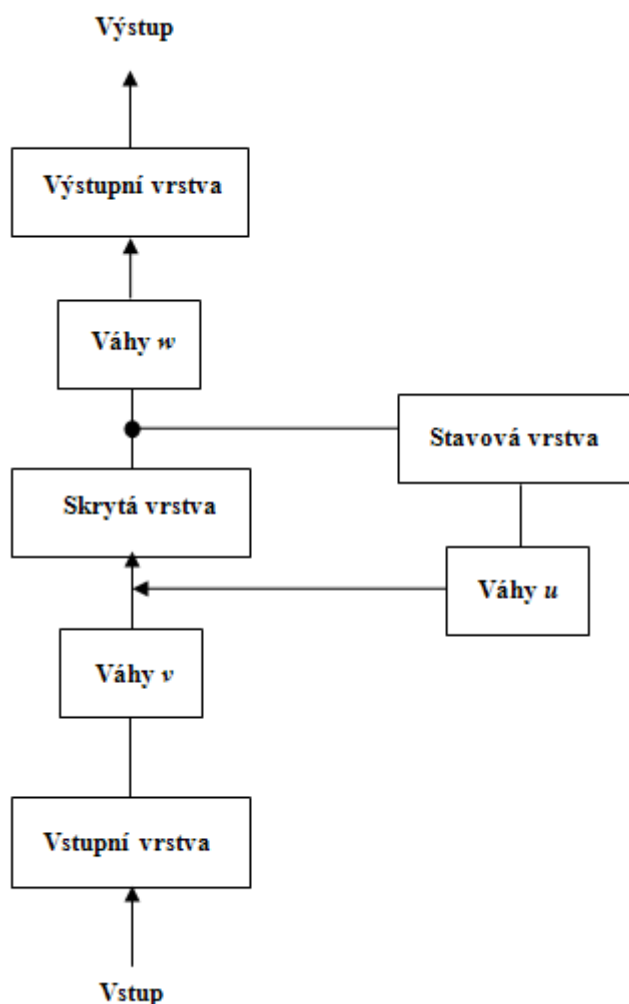
Obr. 22 Rekurentní topologie neuronové sítě

5.3.1 Elmanova síť

Jedná se o neuronovou síť s dvouvrstvou architekturou, která obsahuje zpětnovazební smyčky. Takto uspořádaný systém má setrvačné paměti, které si pamatují hodnotu předchozího stavu. Tyto paměti ovlivňují pak budoucí výstup. V této síti se kromě klasických neuronů nacházejí i neurony stavové, které jsou obsaženy ve stavové vrstvě. Výstup z tzv. skryté vrstvy je napojen na výstupní vrstvu a také na vstup stavové vrstvy. Výstup stavové vrstvy je pak veden na vstup skryté vrstvy, jak je vidět na obrázku 23. Skrytá vrstva má tedy na svůj vstup přiveden výstup ze vstupní a stavové vrstvy. Skryté neurony jsou spojené se stavovými neurony přes váhy, které mají hodnotu 1. Tento způsob zajistí, že výstupy skryté vrstvy v čase t ovlivňují své výstupy v čase $t + 1$. Počet neuronů ve skryté vrstvě je stejný jako počet neuronů ve stavové vrstvě. Stavové neurony jsou propojeny se všemi neurony ve skryté vrstvě. Pro natrénování Elmanovy sítě se využívá algoritmus Backpropagation, kde se provádí rozvinutí rekurentní sítě v čase do sítě s dopředným šířením.

Velkou předností Elmanovy sítě a i obecně všech dynamických sítí je, že nabízejí větší výkonnost, než statické sítě. Nevýhodou však je, že vyžadují velký výpočetní výkon a potřebný paměťový prostor, aby pracovaly s uspokojujícími parametry. Dynamické sítě umožňují pracovat s časově závislými vzory, což je důležité pro mnoho praktických aplikací. Tato vlastnost se využívá při predikci síťového provozu a obecně pro predikci časových řad. Podle

teoretických předpokladů má Elmanova síť podobné výsledky jako síť NARX, ale s jedním podstatným rozdílem. Tato síť by měla lépe kopírovat nástupné hrany laloků průběhu intenzity provozu, kdežto model NARX spíše lépe reaguje na hrany sestupné. [19]



Obr. 23 Elmanova síť

5.3.2 NARX síť

NARX síť (Nonlinear Autoregressive Exogenous) je nelineární autoregresní síť s vnějšími vstupy. Jedná se o dynamickou rekurentní síť, která má zpětnou vazbu na jednu či několik vrstev. NARX model je založen na lineárním ARX modelu, který je běžně používán k modelování časových řad.

Z teoretických předpokladů je patrné, že se síť dokáže velmi rychle naučit trénovacím vzorům a předkládat relativně uspokojivé výsledky. Tento typ sítě bychom pravděpodobně zvolili v situaci, kde máme velké množství vzorků a potřebujeme je přikládat v reálném čase na neuronovou síť. Výhodou NARX sítě je rychlost učení a vybavování, efektivita predikce a v poslední řadě jednoduchost modelu. Její slabinou je velmi těžká gradace řešení, jelikož modifikací tohoto modelu síť se výsledky nezlepšují.

5.3.3 Obecná rekurentní síť – LRN

LRN síť (Layer Recurrent Network) jedná se v podstatě o obecný zápis struktury Elmanovy sítě, která tvoří podmnožinu LRN. Zde máme poměrně velkou volnost při návrhu topologie sítě oproti Elmanově síti, která se vyznačuje jen dvouvrstvou architekturou.

Z teoretických zkušeností lze obecně říci, že změna aktivačních funkcí předepsaného modelu se nevyplácí, protože zde dochází k nárůstu výpočetní náročnosti a ke zvětšení střední chyby. Výhody této sítě jsou velká variabilita a volnost při návrhu architektury sítě. Dobrá hodnota střední chyby, avšak nevýhodou je výskyt větších chyb.

6 VYUŽITÍ NEURONOVÝCH SÍTÍ V PRAXI

Neuronové sítě mají širokou škálu využití v hodně oborech, používají se všude tam, kde nám nevadí případná chyba. Úspěšnost neuronových sítí je obecně cca 95%. Využívají se tam, kde je náročný přesný algoritmus, ať už na finanční prostředky (hardware, experti), anebo čas, kde je požadována nutnost rychlých rozhodnutí.

Nabízí se hned několik typů neuronových sítí. A každý z nich je vhodný pro něco jiného. Zatímco samostatný neuron umí rozlišit prvky do dvou skupin, vrstevnaté neuronové sítě umí rozdělit prvky do libovolného množství skupin. Jedna neuronová síť je vhodná pro kompresi dat např. pro přenos televizního signálu a nepoužitelná pro predikci síťového provozu.

Další možnost predikce je v letecké dopravě. Systém AMT, který optimalizuje rezervace letenek. Kde v první fázi systém předpovídá poptávku po volných místech, zde je právě implementována neuronová síť. A ve druhé fázi navrhuje rozložení letů, to se provádí již tradičními způsoby.

Obor, který hojně využívá neuronové sítě je zdravotnictví. Používá se zde u filtrací EKG signálu. Jsou vyvinuty systémy, které na základě EKG křivky umožňují odhad nemoci pacienta. Další využití je např. při aplikaci antibiotik, kdy neuronová síť pracovala až 4x přesněji než lékaři. Síť např. nepředepisovala antibiotika, na které mohl být pacient alergický. Lékařům sloužila pouze pro konzultační účely.

Nejprestižnější využití je ve finančnictví. Předpovídá se změna kurzů měn, nebo změna kurzů akcií. Při použití klasických metod je úspěšnost cca 55%, tak při použití neuronových sítí se dosahuje až 75%. Existují projekty, které dokázaly při využití i jiných informací než pouhých hodnot kurzu predikovat s úspěšností cca 85%. Dále se předpovídá vzájemné spojení, či krach podniků. Dalším využitím ve finančnictví může být testování, zda daný klient bude schopen splácet půjčku, nebo dokonce navrhnutí výšky půjčky a doby splácení.

Za zmínku také stojí situace, kdy se neuronová síť nainstalovala a učila se reagovat na nějaké podněty od "učitele - Experta". Poté, co byla dostatečně naučena, tak dokázala reagovat správně i na neznámé podněty. Praktické využití bylo např. v pivovarech, atomových elektrárnách. Dělal se i pokusy při řízení automobilů, kdy síť reagovala podobně jako "její" řidič - jezdila agresivně plyn, brzda, nebo naopak klidně a opatrně.

Jiným oborem využití neuronových sítí je robotika. Prakticky každý řídicí systém v robotice lze nahradit neuronovou sítí. Ať už jde o rozpoznávání obrazců, rozhodování jednoho, případně skupiny robotů, nebo jakoukoliv jinou automatizaci. Neuronová síť se může učit doslova jen pouhým "koukáním". Představme si, že člověk řídí auto na dálkové ovládání s pár senzory kolem překážek a síť sleduje jeho reakce na dané situace. Navíc, pokud jí k tomu dáme dostatečné výpočetní prostředky, tak bude síť schopna reagovat správně i za ztížených podmínek, které jsou např. větší rychlosti. Neuronová síť se tak dostane za hranici lidského reflexu.

Predikce pomocí neuronových sítí se také používá v chemickém průmyslu. Kdy místo najímání zkušených a drahých pracovníků může chemický závod naučit neuronovou sít řídit chemickou reakci (proces, apod.) stejně spolehlivě, jako by to dělal sám specializovaný pracovník. [20]

7 VÝHODY A NEVÝHODY PREDIKCE SÍŤOVÉHO PROVOZU POMOCÍ NEURONOVÝCH SÍTÍ

V poslední teoretické kapitole této práce bych chtěl shrnout v následujících pár bodech předpověď samopodobných charakteristik síťového přenosu za pomoci neuronových sítí.

Výhody:

- Učení - není potřeba znát přesný matematický popis vstupů, neuronová síť si zjistí potřebný algoritmus sama,
- Generalizace - na základě předchozí vlastnosti (učení) je neuronová síť schopná správně reagovat i na neznámé data, které ji přijdou na vstup,
- Cenová dostupnost - pro předpověď potřebujeme program Matlab, který obsahuje Neural Networks Toolbox.

Nevýhody:

- Při aplikaci trénování jsou velké hardwarové nároky na počítač,
- Se složitostí sítě se zvyšuje doba trénování, ale výsledek predikce se nemusí zlepšovat,
- Dopředu není jasné, jaký bude výsledek predikce (úspěch/neúspěch).

8 NÁVRH PRAKTICKÉ ČÁSTI

Tento návrh můžeme rozdělit na dvě části. V první části se seznámíme, jak byla data zachycena a zpracována (agregována, znormalizována) pro potřebnou predikci. V druhé části bude vysvětlen návrh predikčního systému s těmito zpracovanými daty.

Wireshark - (Ethereal) je protokolový analyzátor, umožňující odposlouchávání všech protokolů, které počítač přijímá/odesílá. Tento program dokáže paket (balíček informací putující sítí) dekodovat a ukázat celý, jak ho počítač poslal. Mezi jeho nejčastější použití patří zejména analýza a ladění problémů v počítačových sítích, vývoj komunikačních protokolů a studium síťové komunikace. Jedná se o open source program, je tedy šířen zcela zdarma.

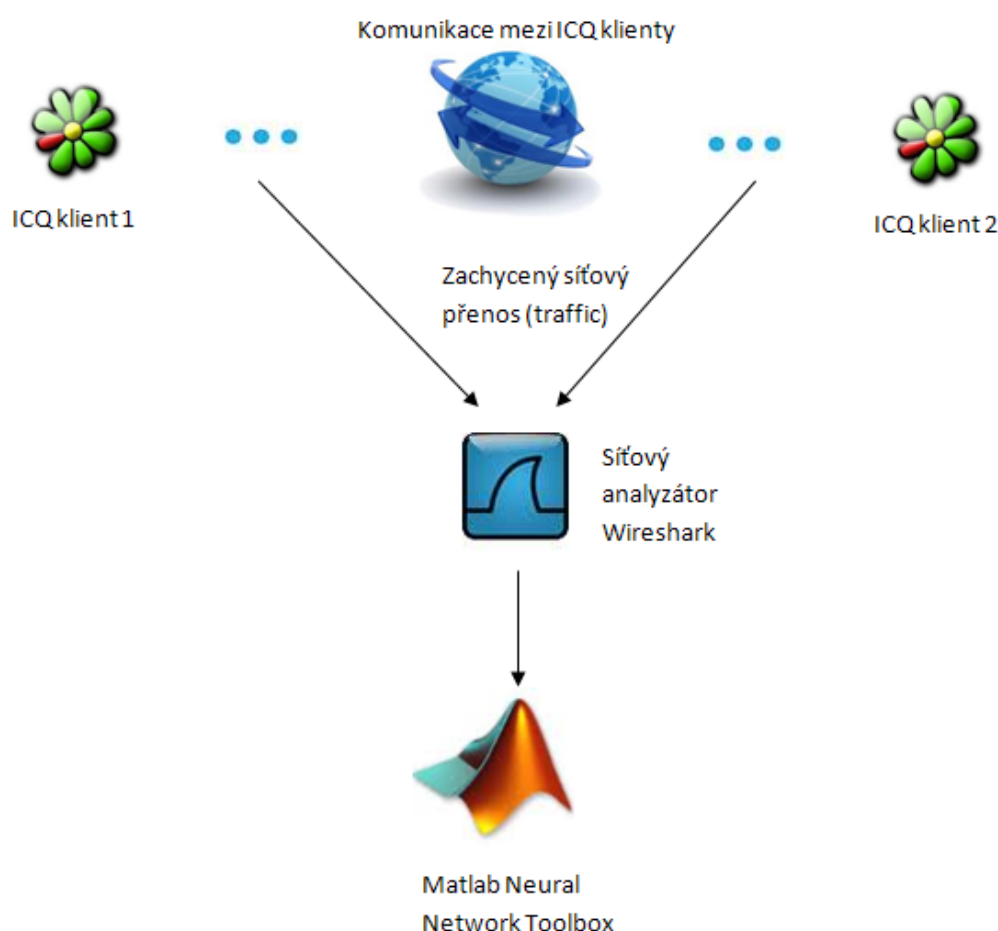
MATLAB - je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. Je to nástroj jak pro pohodlnou interaktivní práci, tak pro vývoj širokého spektra aplikací. Tento program poskytuje nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti. Dalšími moduly lze funkce rozšiřovat například o simulační prostředí k modelování dynamických systémů, která využívají algoritmy Matlabu pro numerické řešení především nelineárních diferenciálních rovnic. Jedním z takových rozšíření je například Neural Network toolbox. Zobrazovací a simulační prostředí neuronových sítí.

Neural Networks Toolbox - poskytuje nástroje pro tvorbu, návrh, trénování, simulaci a vizualizaci neuronových sítí. Neuronové sítě nacházejí uplatnění zejména v oblastech, kde je použití formálních analytických nástrojů obtížné, či dokonce nemožné, jako je vzorové rozpoznávání nebo řízení a identifikace nelineárních soustav. Toto prostředí nabízí grafická uživatelská rozhraní, která usnadňují návrh a práci s neuronovými sítěmi. Modulární, otevřená a rozšiřitelná stavba toolboxu zjednodušuje tvorbu vlastních funkcí a sítí. Grafické rozhraní má široké možnosti nastavování architektury sítě a učících algoritmů. Jednoduše umožňuje zobrazovat a pochopit architekturu sítě. Neural Networks Toolbox také obsahuje funkce pro automatické vytváření simulačních bloků neuronových sítí do Simulinku. [21]

8.1 Zachycení testovacích dat

Pomocí síťového analyzátoru Wireshark byla odchycena komunikace dvou ICQ klientů (ICQ klient 1 a ICQ klient 2), jak je vidět na obrázku 24. Odchycený síťový provoz byl na lokální síti (LAN), který bude sloužit k natrénování vybrané neuronové sítě. Byl uskutečněn hovor o délce 60 minut. Na tento způsob komunikace se lze podívat jako na celkový upload/download aplikace ICQ.

V programu Wireshark jsou zachycené pakety přehledně řazeny pod sebe do řádků. V jednotlivých řádcích můžeme zvolit zobrazované hodnoty a ty pak pomocí zkratky (*ctr + p*) vyexportovat/vytisknout do souboru, který lze načíst a dále zpracovávat v programu Matlab. Pro predikci síťového provozu musíme jednoznačně vědět velikost (*length*) a čas (*time*) příchodu/odchodu paketu v síti, proto byly vyexportovány právě tyto dvě hodnoty. Pomocí tohoto programu byl odchycen další následující síťový provoz. Jednalo se konkrétně o nahrání (*upload*) série fotek z PC na server <http://www.picasa.cz>. Poslední odchycený síťový provoz byl (*download*) programu Linux Debian, který se nacházel na serveru Ulozto.cz na adrese <http://www.ulozto.cz/xNuTsph/mnsb-debian-zip>. Zachycený *upload/download* pomocí síťového analyzátoru Wireshark ilustrují obrázky 27 a 28.



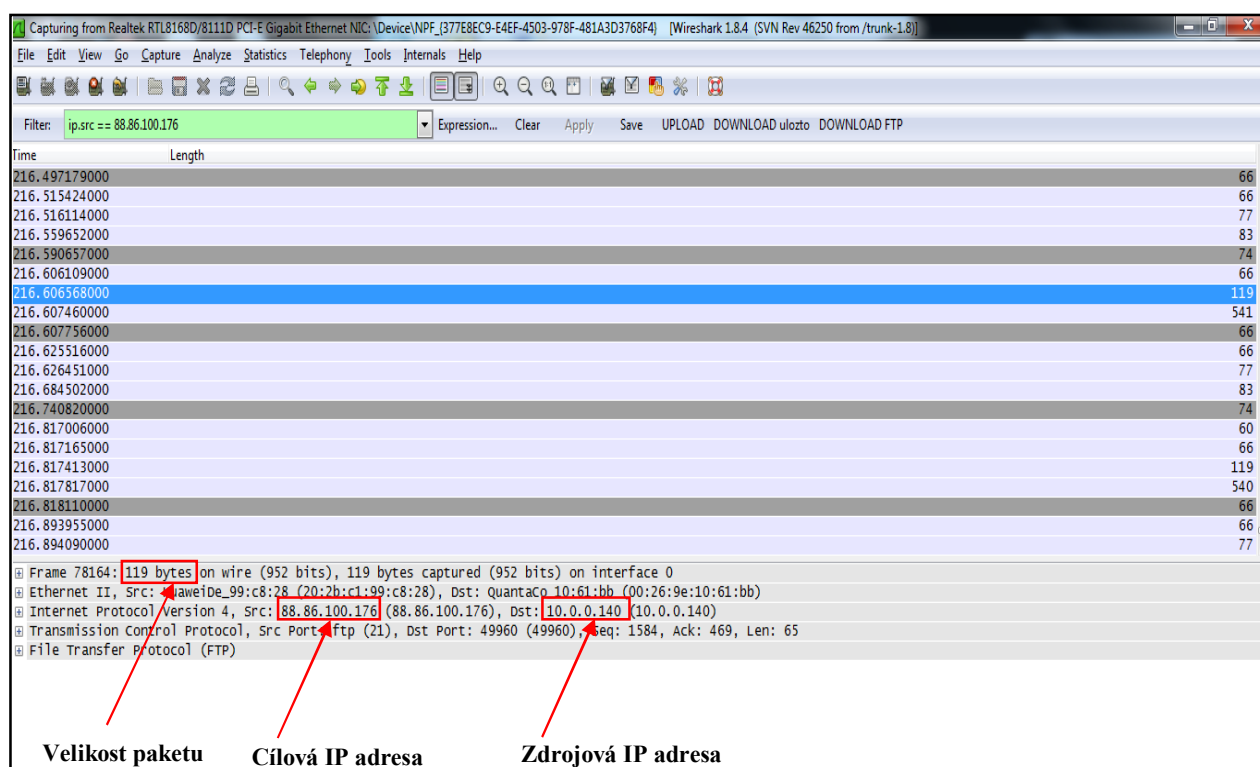
Obr. 24 Zachycený provoz ICQ komunikace pomocí analyzátoru Wireshark

[9], [22]

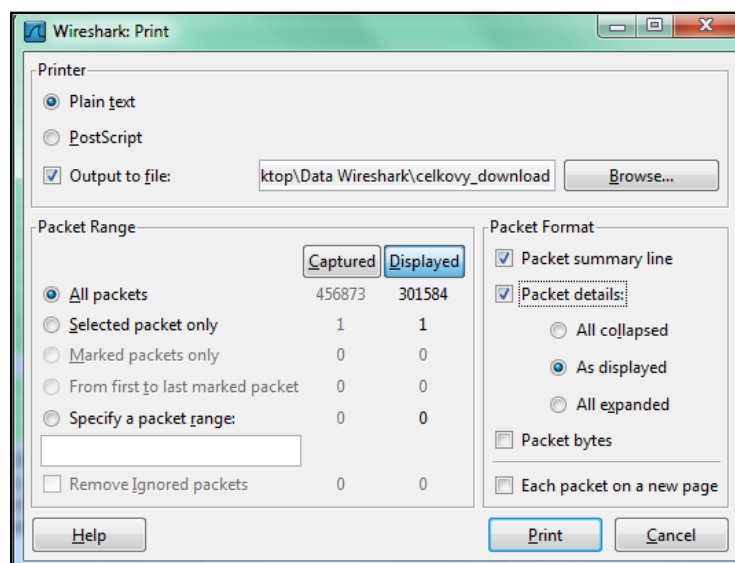
Program Wireshark umí filtrovat zachycený síťový provoz podle námi zvolených pravidel. Protože při možném využití neuronových sítí k predikci síťového provozu a následném dynamickém upravování velikosti jednotlivých front na směrovači nás nezajímá celkový síťový provoz, ale za to provoz konkrétní. Celkový *upload/download*, nebo

upload/download aplikace (ICQ), tak je zapotřebí konkrétní provoz pomocí funkce (filter) vyfiltrovat.

Celkový upload byl vyfiltrován tak, že byly zobrazeny pakety pouze se zdrojovou IP adresou PC v našem případě `ip.src == 10.0.0.140`. Celkový download byl vyfiltrován s cílovou IP adresou PC. Upload a download paketů komunikace pomocí programu ICQ byl filtrován podle zdrojového/cílového portu a protokolu UDP, pod kterým celá tato komunikace běží `udp.srcport==51835`. Na obrázku 25 je zobrazeno okno programu Wireshark, kde je použit filtr na zobrazení paketů pouze z cílové adresy, tudíž se jedná o celkový download.

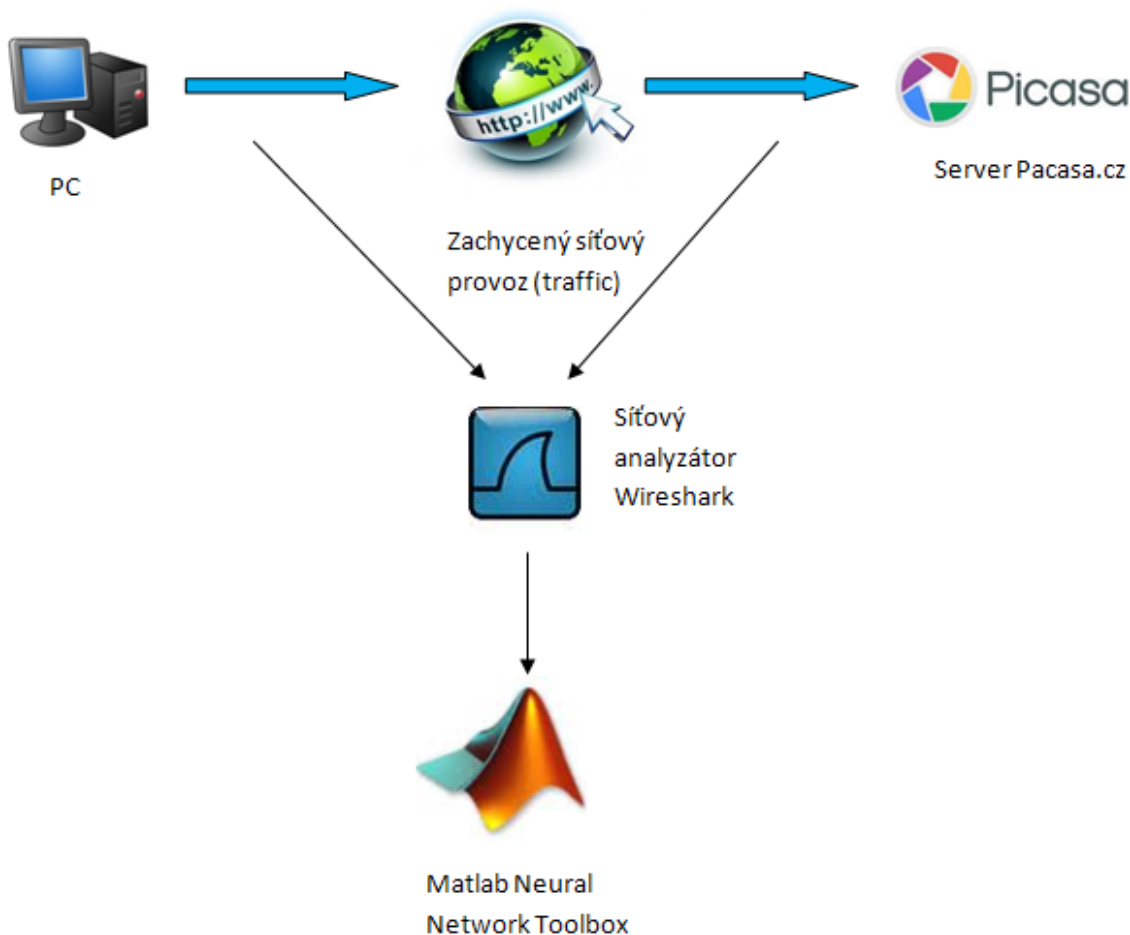


Obr. 25 Program Wireshark



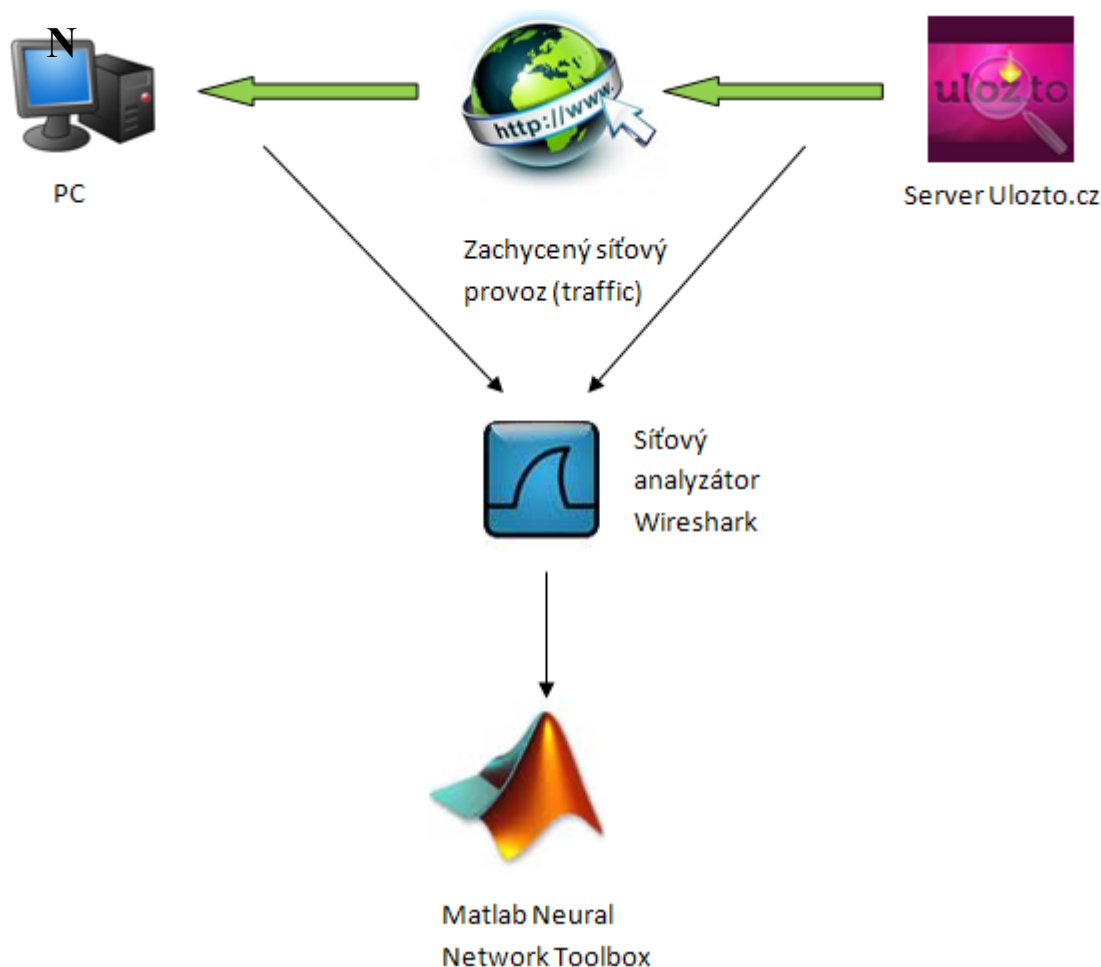
Obr. 26 Uložení provozu do souboru celkovy_download.txt.

Z odchyceného síťového provozu byly vyfiltrovány pouze tyto čtyři druhy provozu: celkový upload (nahrání fotografií na server Picasa), celkový download (stažení ISA souboru programu Linux Debian ze serveru Ulozto), dále pak download/upload aplikace ICQ. Jednotlivé provozu byly vyexportovány/vytisknuty do souborů *celkovy_download.txt*, *celkovy_upload.txt*, *icq_download.txt*, *icq_upload.txt*, jak zobrazuje obrázek 26. Tyto druhy provozu budou sloužit k testování vhodnosti predikce síťového provozu pomocí neuronových rekurentních sítí v prostředí Matlab Neural Network Toolbox.



Obr. 27 Zachycený upload pomocí analyzátoru Wireshark

[9], [23], [24]



Obr. 28 Zachycený download pomocí analyzátoru Wireshark

[9], [23], [24]

8.2 Agregace a normalizace zachycených dat

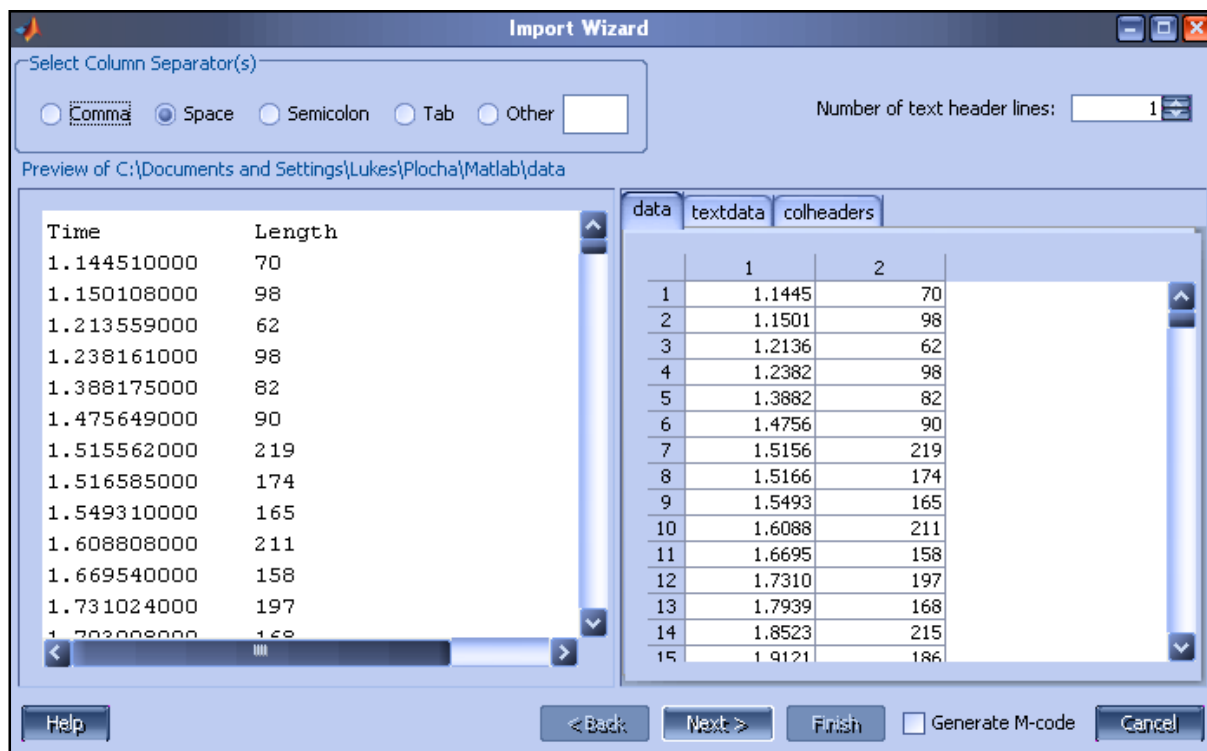
Náplní druhé kapitoly byl vysvětlen pojem samopodobnost. Bylo zde řečeno, že útvar je samopodobný pokud vypadá stejně v jakémkoliv měřítku či zvětšení. Traffic, neboli síťový provoz vykazuje charakteristiky samopodobnosti, jelikož jeho charakteristiky jsou si podobné v různých časových měřítkách. Nyní máme zachycena testovací data, jak nám popisuje podkapitola 8.1., ale ještě je musíme upravit do přijatelné podoby. Toho dosáhneme tzv. agregací (seskupením) intenzit provozu, kdy v podstatě sčítáme tyto intenzity v rámci každého intervalu.

Různé časové měřítko lze dosáhnout agregací síťového provozu v určitém námi zvoleném časovém měřítku. To znamená, že pokud je zvolené časové měřítko např. 1s, tak

každou sekundu jsou velikosti příchozích/odchozích paketů sčítány. Výstupem každou sekundu je číslo určující celkovou velikost všech paketů, které došly/odešly během dané sekundy. Když máme např. síťový provoz agregován s časovým intervalem $M=1s$, tak k naměření 100 hodnot je potřeba, aby délka síťového provozu byla 100 sekund. Tedy je jasné, že pokud bude síťový provoz agregován časovým intervalem $M=10s$, tak k naměření 100 hodnot je zapotřebí mít k dispozici síťový provoz o délce 1000 sekund.

Před samotným použitím takto upravených (agregovaných) dat v oboru neuronových sítí je velmi vhodné tyto data znormovat. Protože nás zajímá hlavně průběh intenzit (vychází se z principu samopodobnosti), než jejich skutečná hodnota.

Zachycený síťový provoz *celkovy_download.txt*, *celkovy_upload.txt*, *icq_download.txt*, *icq_upload.txt*, byl načten v programu Matlab *File – Import data...* jako matice 2 x X hodnot. V prvním sloupci byl uveden čas příchodu/odchodu paketů a druhý sloupec hodnot reprezentoval velikost paketů v Bytech, jak je vidět na obrázku 29.



Obr. 29 Import dat v programu Matlab

Takto načtený síťový provoz byl agregován s časovým intervalem 400 ms a poté byl znormován pomocí metody Min-max. Tento upravený síťový provoz byl pak rozdělen na trénovací a testovací matice. Neuronové sítě byly natrénovány s učitelem viz kapitola 4, tak k trénovací vstupní matici byl vytvořen trénovací výstupní vektor a k testovací vstupní matici byl vytvořen výstupní testovací vektor, který ale nebyl neuronové síti předkládán, sloužil pouze k porovnání správnosti predikce síťového provozu. Agregace a následná normalizace

byla vytvořena pomocí skriptu *agregace_normalizace_400ms.m*, který je součástí přiloženého CD. Pomocí tohoto skriptu se zpracovává vstupní matice, která má rozměr 2 x X hodnot, výstupem jsou matice, které obsahují agregovaná a znormovaná data. Jedná se o matice trénovací 4 x 500 hodnot, trénovací_vystupni 1x 500 hodnot, testovací 4 x 1500 hodnot a testovací_vystupni 1 x 1500 hodnot. Podle velikostí matic je patrné, že neuronová síť na základě 4 vstupních hodnot bude predikovat 1 výstupní hodnotu.

Vytvořená trénovací vstupní matice má formát 4 x 500 hodnot. Na vstup neuronové sítě jsou tedy předkládány 4 hodnoty a výstupem je jedna předpovídaná (predikovaná) hodnota. Byly zde použity neuronové s učitelem, tudíž je potřeba také konkrétním čtyřem vstupním hodnotám přiřadit jednu konkrétní výstupní hodnotu. První hodnota výstupního trénovacího vektoru je rovna páté hodnotě ve vstupní matici 4x500 hodnot. Tuto situaci nám ilustruje obrázek 30.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|--------|-----------------|--------|--------|--------|------------------|--------|--------|--------|
| 1 | 0.0694 | 0.1235 | 0.2002 | 0.7324 | 0.3125 | 0.2463 | 0.2726 | 0.2383 | 0.3132 |
| 2 | 0.1259 | 0.8755 | 0.3306 | 0.2889 | 0.2504 | 0.3082 | 0.3058 | 0.1516 | 0.2850 |
| 3 | 0.1694 | 0.1803 | 0.2378 | 0.2832 | 0.1644 | 0.2501 | 0.3090 | 0.1734 | 0.2739 |
| 4 | 0.1377 | 0.9021 | 0.2501 | 0.2286 | 0.3994 | 0.2928 | 0.2736 | 0.4650 | 0.3365 |
| 5 | | | | | | | | | |
| 6 | | | | | | | | | |
| 7 | | VSTUPNÍ HODNOTY | | | | VÝSTUPNÍ HODNOTY | | | |
| 8 | | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |
| 11 | | | | | | | | | |

Obr. 30 Zobrazení vstupních a výstupních trénovacích vzorů

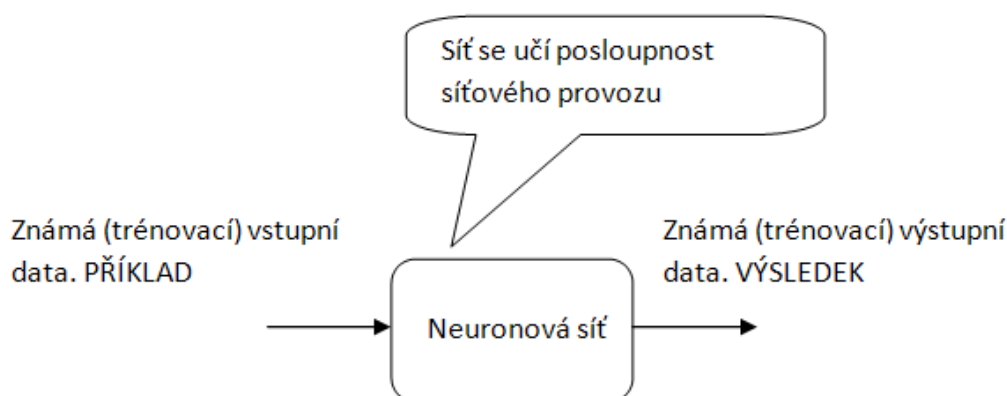
8.3 Návrh predikčního systému

Predikce síťového provozu pomocí neuronových bude testována na čtyřech typech síťového provozu a to na celkovém downloadu odchyceného provozu, celkovém uploadu, downloadu aplikace ICQ a uploadu aplikace ICQ. Nasnímaná a upravená data, jak jsou popsány v podkapitolách 8.1 a 8.2 jsou upraveny a rozděleny na dvě části. A to na část vstupních trénovacích dat a část vstupních testovacích dat. Z první části byl vytvořen výstupní trénovací vektor, který bude předán neuronové síti pro naučení a z druhé části byl vytvořen výstupní testovací vektor, který však síti předložený nebude. Bude sloužit na porovnání (ověření) naučení sítě.

Takto budou rozděleny vstupní data na trénovací část a testovací část. Jednotlivé typy sítí budou pracovat se stejnými vstupními daty (celkovy_download.txt, celkovy_upload.txt, icq_download.txt, icq_upload.txt), čímž se dosáhne objektivnějších výsledků. Bude potřeba ověření dvou principů, které jsou popsány v kapitole 4. Principu trénování a principu generalizace. První ze jmenovaných bude vyžadovat přikládání trénovacích vzorů a ověřování trénování pomocí gradientu a střední chyby. Kdy se na vstup i výstup systému vloží známá data a síť se je pokusí naučit, výsledek se pak bude moci porovnávat s výstupními testovacími daty, které nebyly neuronové síti předány a slouží právě ke zpětné kontrole. Druhý způsob bude představovat přiložení neznámých vzorů na výše naučenou neuronovou síť. A sledována právě bude reakce sítě na tyto neznámé data.

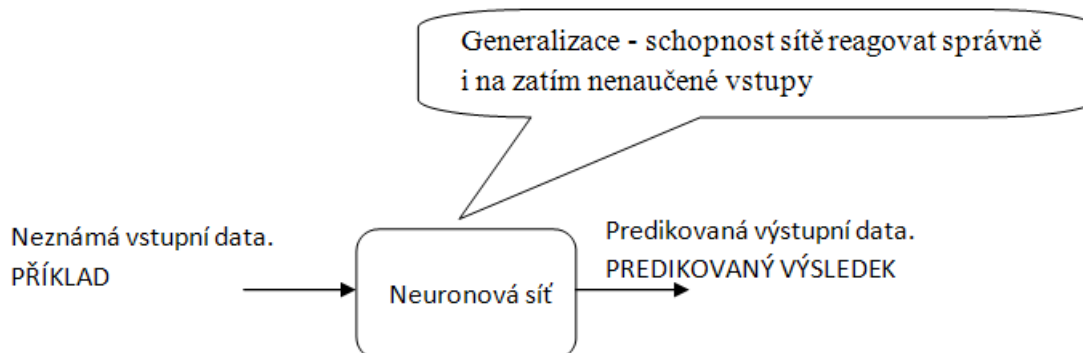
Zjednodušené schéma principu predikce je zobrazené na obrázcích 31 a 32.

a) Učení



Obr. 31 Princip učení

b) Generalizace



Obr. 32 Princip generalizace

a) V prvním případě budou sítě na vstupu předložena známá vstupní data (PŘÍKLAD). Na výstup této sítě přiložíme opět známá, ale výstupní data (VÝSLEDEK). To má za následek, že po jistém učicím čase, sama síť na vstupní příklad vygeneruje správný výsledek. Tuto vlastnost sítě nazýváme učicí schopnost.

b) Takto naučené sítě, pak na vstup předložíme neznámá vstupní data a budeme sledovat princip generalizace, kdy síť bude muset správně reagovat i na neznámé podněty. Tuto vlastnost nám ukazuje obrázek 32.

Tyto principy budou odzkoušeny na všech typech provozů a u všech tří typů vybraných sítí (Elmanova, NARX, LRN). Bude vyhodnoceno, která síť je nejvhodnější pro predikci daného typu provozu. Jelikož budeme mít původní výstupní charakteristiky průběhů, které pak poslouží k porovnání s predikovanými výstupními hodnotami navrženého systému.








9 VYTVOŘENÍ, NATRÉNOVÁNÍ A OTESTOVÁNÍ NEURONOVÝCH SÍTÍ V PROSTŘEDÍ NN TOOLBOX

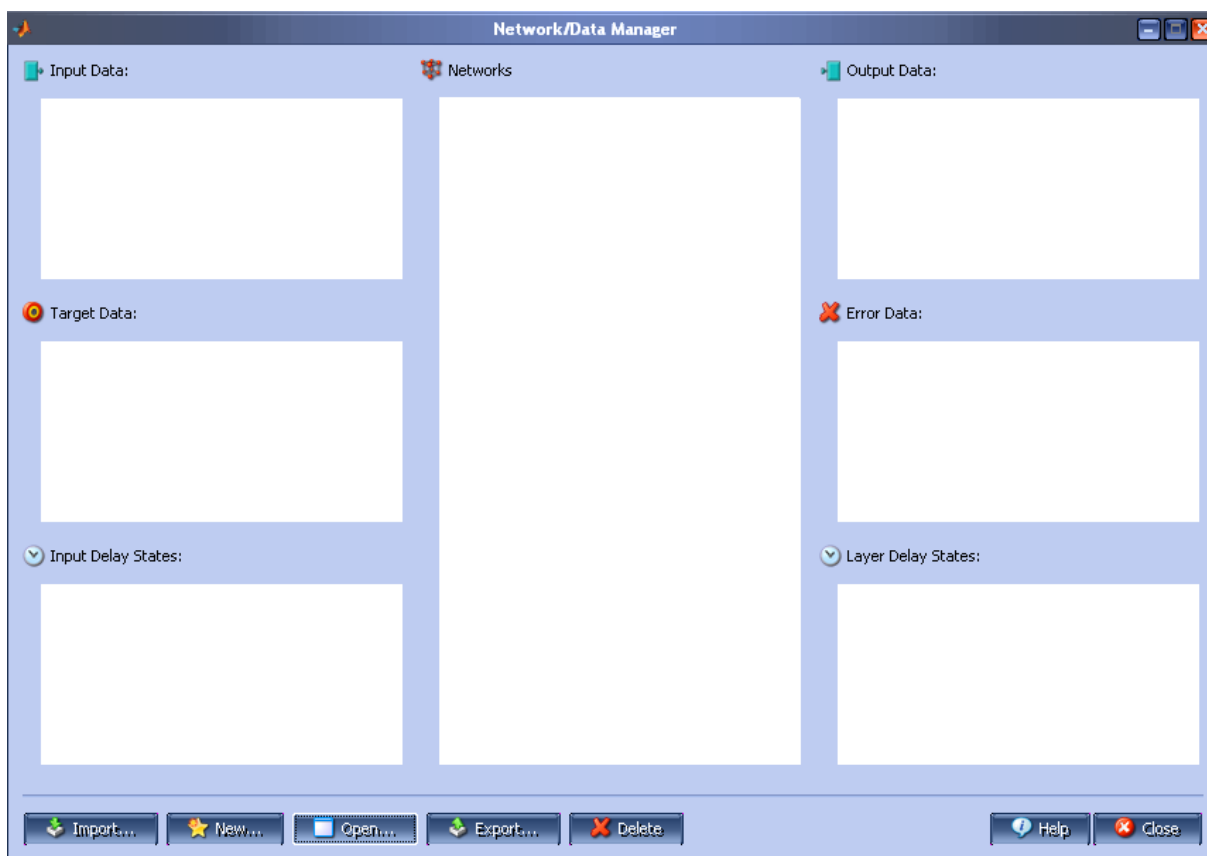
K vytvoření, trénování a následnému testování neuronových sítí byl použit program Matlab verze 7.8.0.347 (R2009a), který obsahuje vývojové prostředí NN Toolbox.

9.1 Vytvoření neuronové sítě

Po napsání příkazu `nntool` v hlavním okně programu Matlab se nám otevře okno Neural Network/Data Manager, jak je vidět na obrázku 33. Zde můžeme importovat, editovat, exportovat a mazat námi vytvořené neuronové sítě.

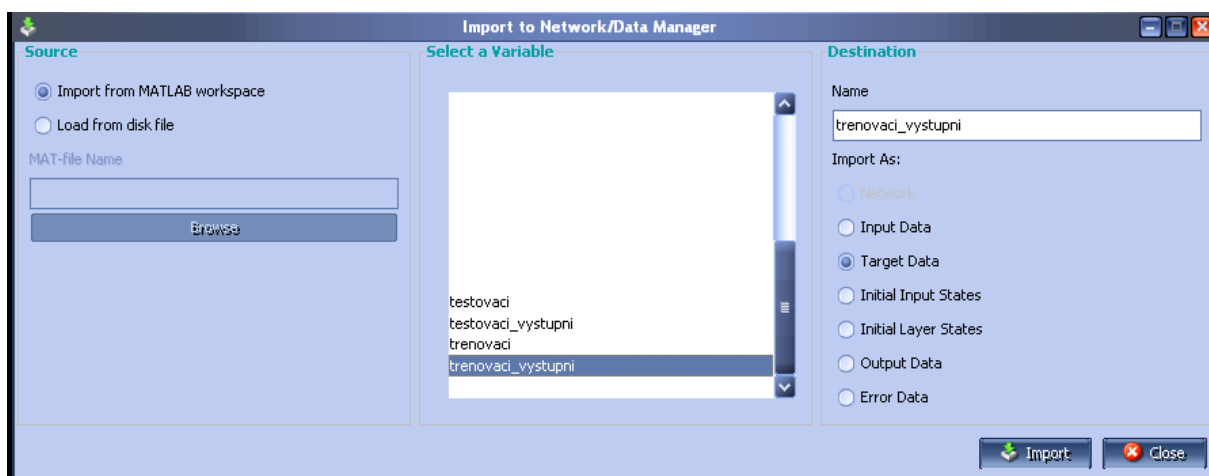
Stručný význam jednotlivých oken Neural Network/Data Manager:

-  **Input Data:** Vstupní data k trénování a testování neuronové sítě.
-  **Target Data:** Výstupní data k trénování neuronové sítě.
-  **Input Delay States:** Vstupní data zpoždění (pro síť se vstupním zpožděním).
-  **Networks** Vytvořené neuronové sítě.
-  **Output Data:** Výstupní data (reakce neuronové sítě na vstupní data).
-  **Error Data:** Rozdíl mezi Target Data a Output Data.
-  **Layer Delay States:** Zpoždění vrstvy (pro síť se zpožděním ve vrstvě).



Obr. 33 Výběrové okno Network/Data Manager

Abychom mohli vytvořit neuronovou síť, tak je zapotřebí pomocí tlačítka *Import...* importovat data, které budou sloužit k trénování a následnému testování neuronových sítí. Je potřeba naimportovat vstupní trénovací/testovací data a výstupní trénovací data. Výstupní testovací data importovány nebudou. Ty pouze poslouží k ověření úspěšnosti neuronové sítě předpovídat síťový provoz. Budou porovnávány s Output Daty. Import dat je vidět na obrázku 34.



Obr. 34 Import dat

Tímto krokem jsme úspěšně nainportovali potřebná data. Nyní je třeba ještě vytvořit samotnou neuronovou síť. V základním okně nntoolu klikneme na tlačítko *New* a objeví se okno, kde můžeme definovat nejružnější parametry naší sítě, jak je vidět na obrázku 35. Význam jednotlivých parametrů je následující:

Name: Uživatelem zadané jméno neuronové sítě (defaultně nastaveno na Network1).

Network Type: Výběr typu neuronové sítě.

Input data: Vstupní trénovací data.

Target data: Výstupní trénovací data.

Training function: Typ trénovací funkce.

Adaption learning function: Typ adaptační funkce, po dosažení minimální stanovené chyby.

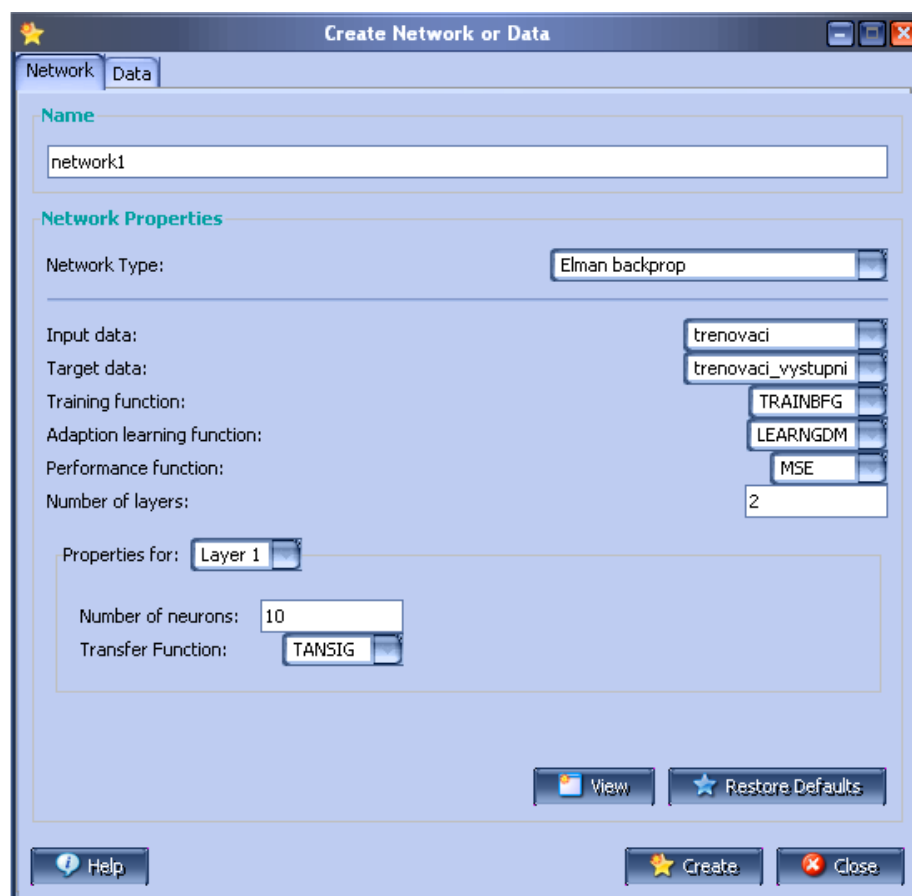
Performance function: Funkce hodnotící natrénování neuronové sítě.

Number of layers: Počet vrstev neuronové sítě (hodnotu je zapotřebí potvrdit Enterem).

Properties for: Slouží pro konkrétnější nastavení jednotlivých vrstev.

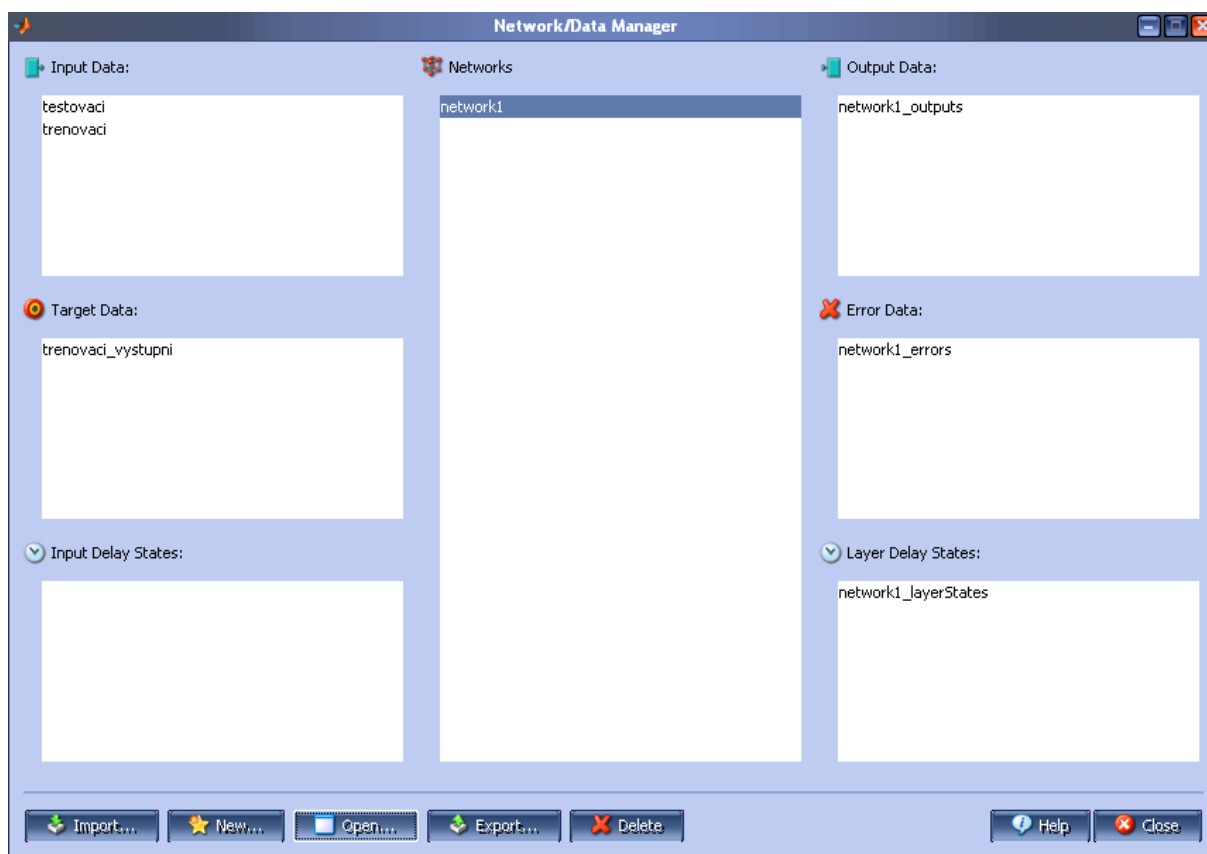
Number of neurons: Volba počtů neuronů ve vrstvě (počet je zapotřebí potvrdit Enterem).

Transfer Function: Volba přenosové funkce vrstvy.



Obr. 35 Vytvoření nové neuronové sítě

Parametry nastavíme dle obrázku 35. Za zmínku stojí MSE. Je to střední chyba naší sítě. Může být jedním z kritérií pro ukončení trénování. V záložce Data už není potřeba nic nastavovat. Po nastavení všech požadovaných parametrů v záložce Network se po kliknutí na tlačítko *Create* vytvoří nová neuronová síť, která se uloží do okna Networks, jak je vidět na obrázku 36. O této akci nás informuje dialogové okno, které je zobrazeno na obrázku 37.



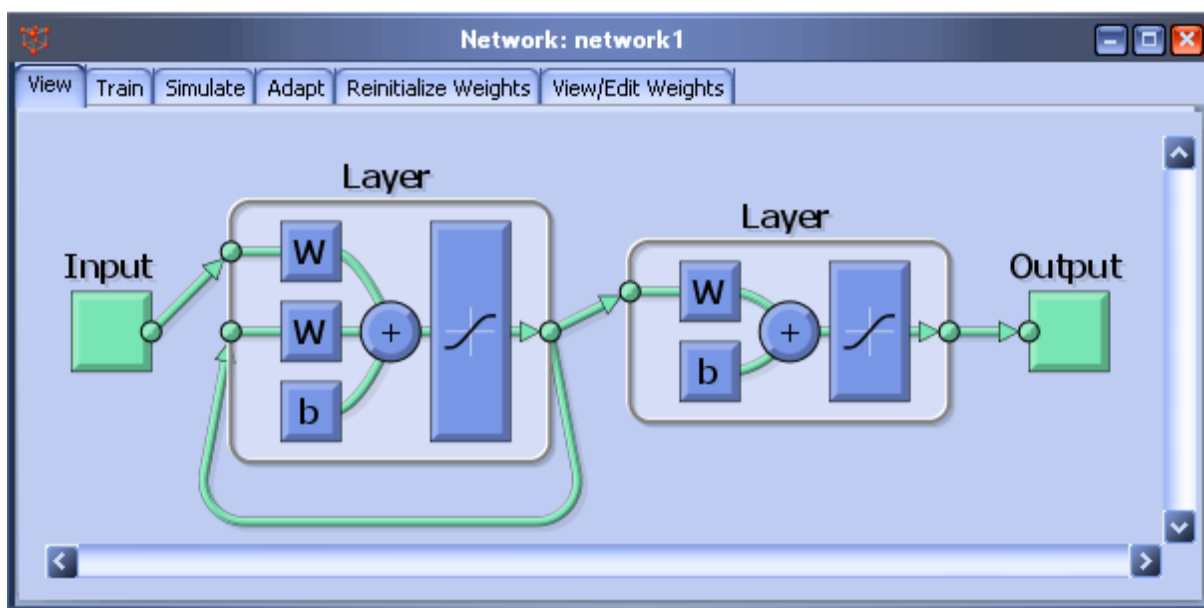
Obr. 36 Nastavení Network/Data Manageru



Obr. 37 Informace o vytvoření nové neuronové sítě

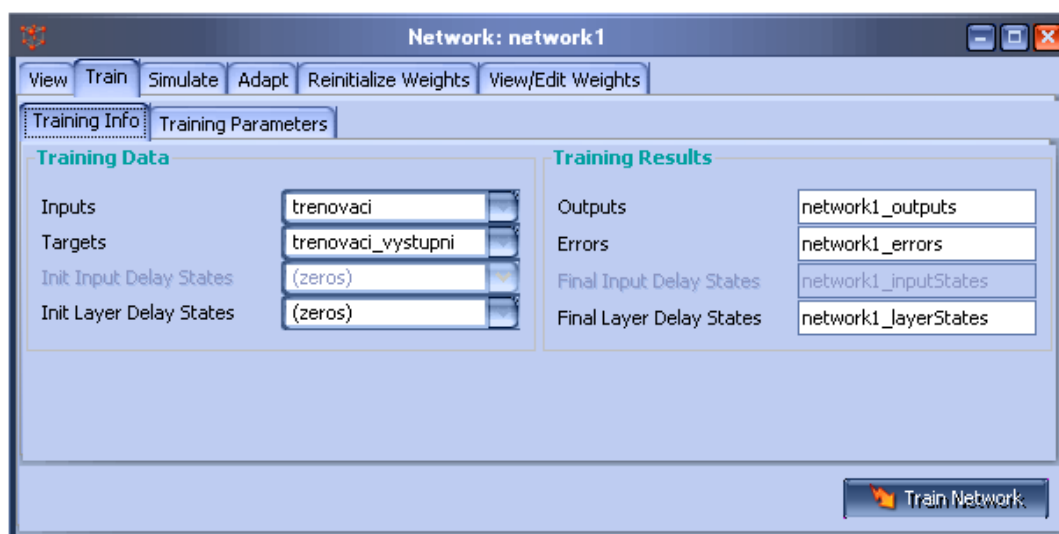
9.2 Trénování neuronové sítě

V okně Network/Data Manageru můžeme pomocí tlačítka *Open* vytvořenou síť otevřít. Na první záložce View se nachází topologie navrhnuté sítě obrázek 38. Druhá záložka Train slouží k trénování neuronové sítě. Třetí záložka v pořadí Simulate slouží k simulaci/testování generalizace neuronové sítě. Záložka Adapt umožňuje adaptaci neuronové sítě. Poslední dvě záložky se zabývají váhami mezi jednotlivými neurony.



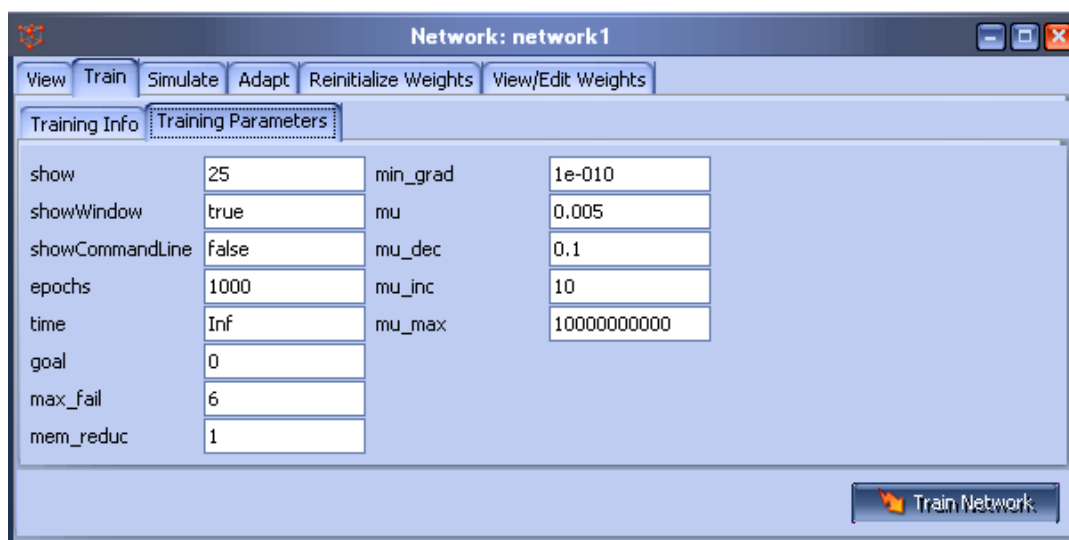
Obr. 38 Zobrazení topologie navrhnuté sítě

V záložce Train a následné podzáložce Training info opět musíme nadefinovat vstupní a výstupní data. Výsledky budou uloženy do proměnných *network1_outputs* (výstupní hodnoty) a *network1_errors* (rozdíl mezi skutečnými - Outputs a vypočtenými - Target hodnotami). Toto nám ilustruje obrázek 39.

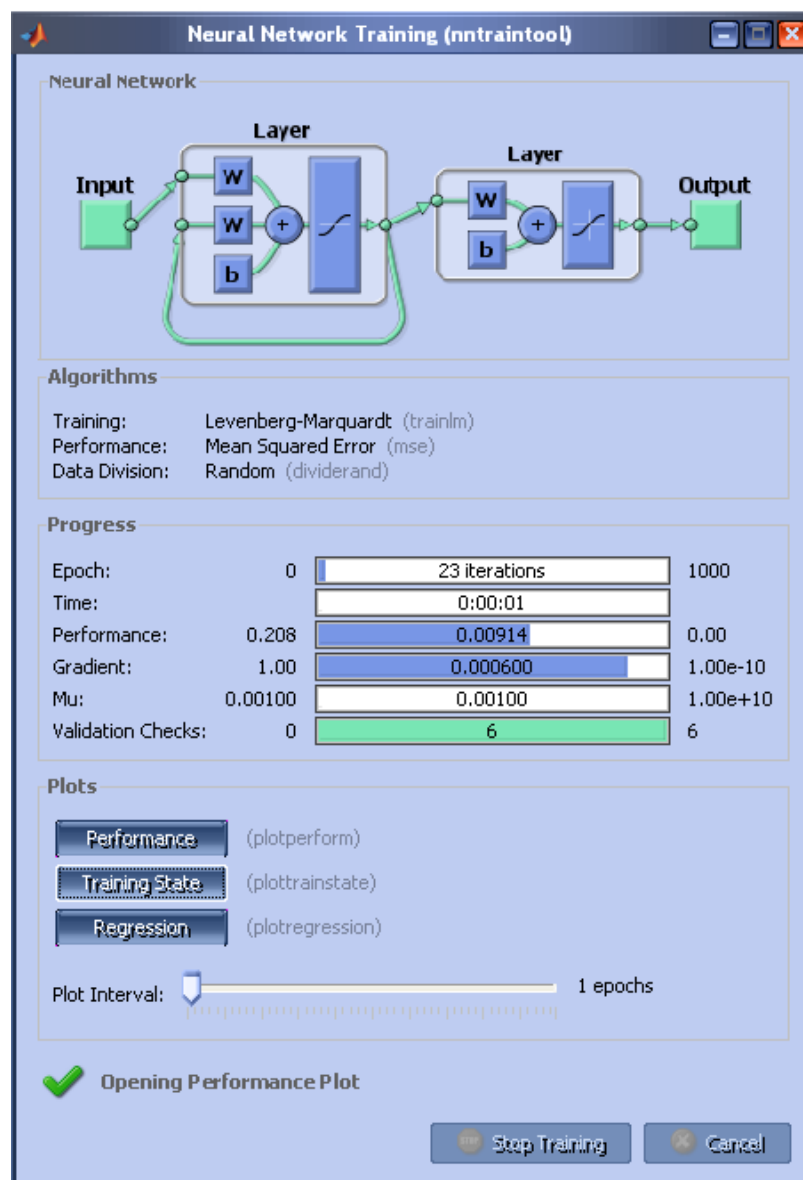


Obr. 39 Nastavení trénovacích dat

Po přepnutí do podzáložky Training Parameters obrázek 40. Zde máme potřebné podrobné parametry k trénování sítě. Z nichž nás bude nejvíce zajímat hodnota o počtu epoch. Počet epoch (v našem případě 1000) nám udává počet trénovacích epoch. Epocha je časový interval, během kterého je síti předkládán každý vzor učicí množiny minimálně jednou. Na 1000 učicích epoch se zpravidla nikdy nedostaneme, protože trénování sítě bývá ukončeno, pokud klesne střední chyba sítě pod stanovenou mez (MSE). Nyní máme vše připraveno, tak můžeme přejít k trénování neuronové sítě, které se provede zmáčknutím tlačítka Train Network. Po tomto kroku nám vyskočí přehledné grafické rozhraní ve kterém můžeme tento proces trénování pohodlně sledovat obrázek 41.

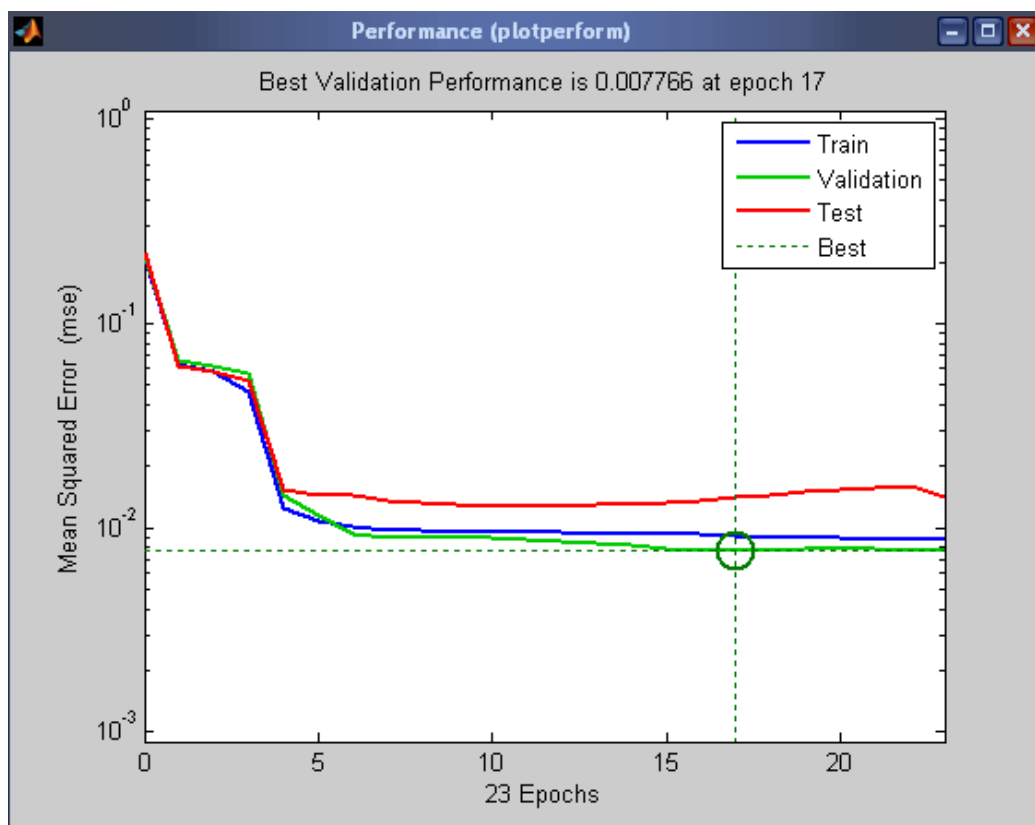


Obr. 40 Nastavení trénovacích parametrů



Obr. 41 Grafická podoba trénovacího procesu

Z tohoto grafického okna je zřejmé, že se síť natrénovala za 1 sekundu. K natrénování bylo použito 23 iterací (epoch). Údaj Performance nám udává chybu trénování, hodnota vlevo odpovídá počáteční chybě a hodnota uprostřed udává chybu sniženou trénováním sítě. Dále po kliknutí na tlačítko Performance lze z grafu vyčíst, že chyba po trénování byla 0,007766 a bylo jí dosaženo v 17. epoše. Toto nám ilustruje obrázek 42. Odchylku v odpovědích na validační řadu používáme jako kritérium pro ukončení trénování sítě.

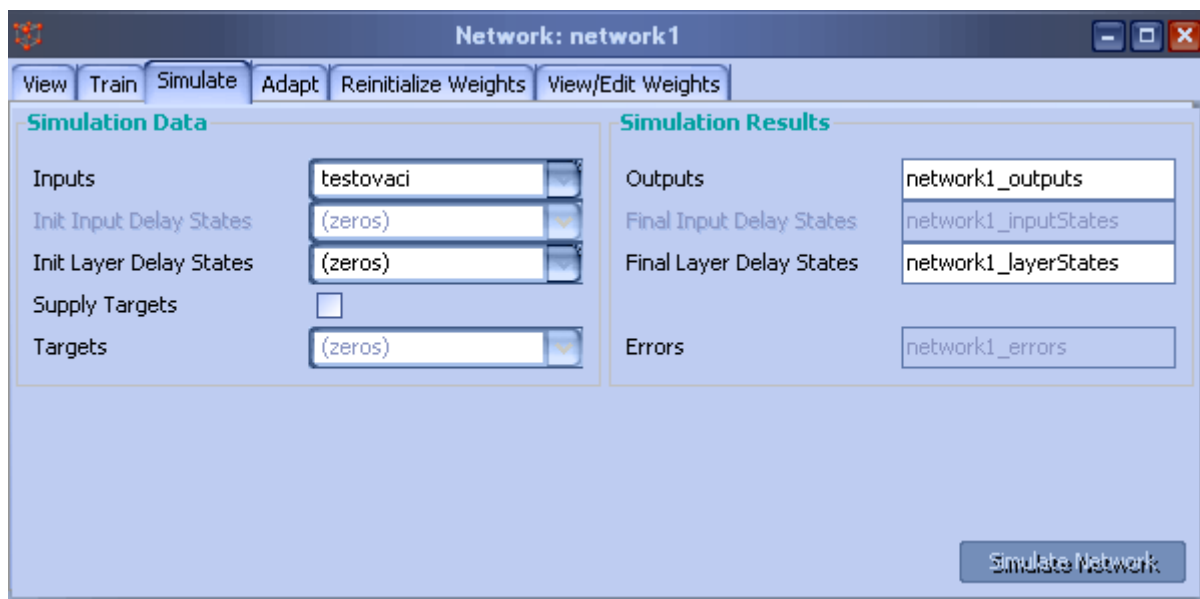


Obr. 42 Závislost trénovací chyby na počtu epoch

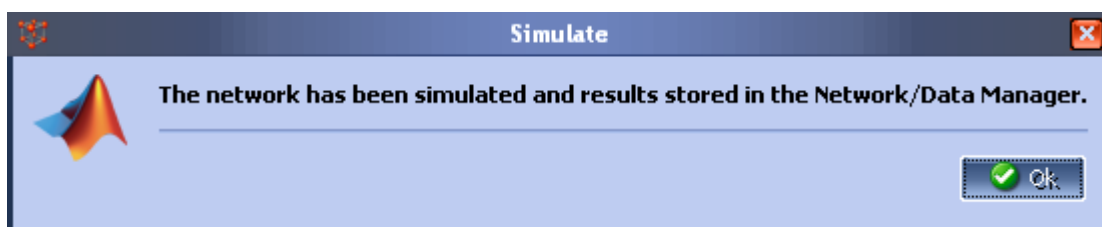
9.3 Predikce neznámých hodnot (princip generalizace)

Princip generalizace je jeden ze základních vlastností neuronových sítí viz kapitola 4. Na základě natrénování jsou totiž schopny tyto sítě více, či méně predikovat nelineární časové řady. Po natrénování sítě jsme okno Neural Network Training zavřeli a přešli se na záložku Simulate. Zde je opět nutné navolit potřebné vstupní testovací data jak je vidět na obrázku 43. Tyto data ale budou ovšem sítí neznámá a bude se zde ověřovat princip generalizace - reakce neuronové sítě na neznámý vstup. Dále stačí kliknout na tlačítko Simulate Network, tím se spustí simulace sítě. Výsledek (predikovaná data) se uloží do proměnné `network1_outputs` a bude porovnán s výstupními hodnotami v našem případě `testovací_vystupni`. Provedení úspěšné simulace nám potvrdí obrázek 44.

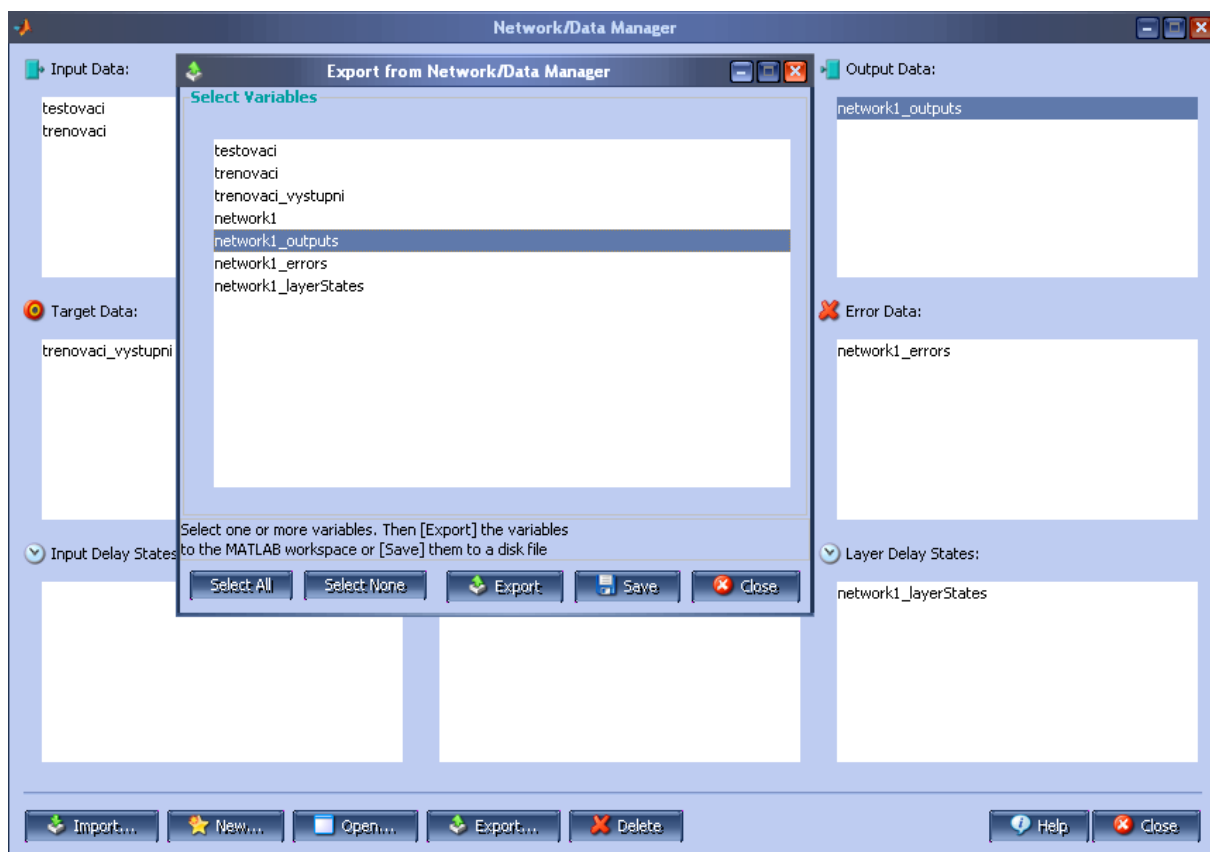
Abychom mohli tyto predikovaná data porovnat, tak je musíme nejprve z Network/Data Manageru vyexportovat do okna Workspace v prostředí Matlab, to provedeme jejich označením a kliknutím na tlačítko Export... jak je vidět na obrázku 45. Úspěšný export nám potvrdí informativní okno obrázek 46.



Obr. 43 Simulace - Předložení neznámých vstupních hodnot



Obr. 44 Informativní okno o provedení simulace.

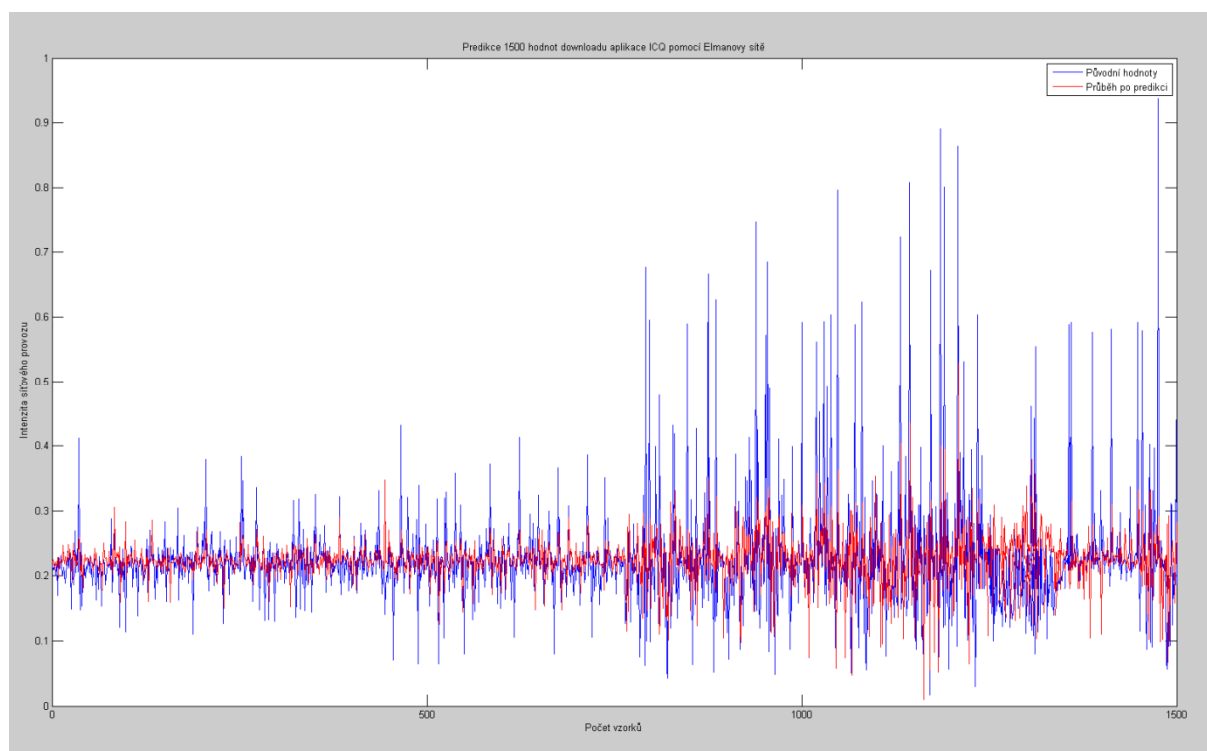


Obr. 45 Export dat do okna Workspace

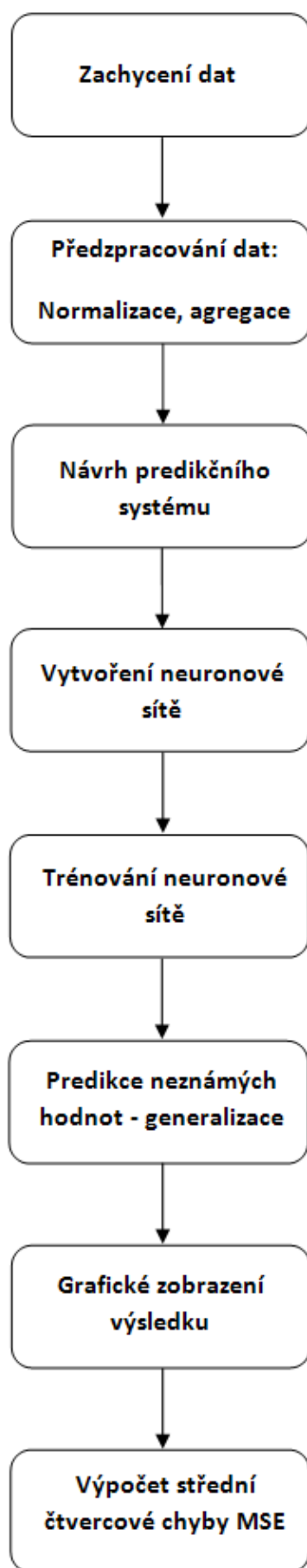


Obr. 46 Informativní okno úspěšného exportu dat

Nyní máme výstupní hodnoty i predikované, tak je můžeme pomocí jednoduchého skriptu *vysledek predikce test_vyst a network1_outputs.m* na zobrazení obou grafů mezi sebou porovnat, jak je vidět na obrázku 47. Jako poslední operaci provedeme výpočet Střední čtvercové chyby MSE (Mean Squared Error), která se opět provede pomocí skriptu s názvem *chyba.m*. Tato metoda vypočítá rozdíl mezi skutečnými a předpovídanými hodnotami, abychom si mohli ověřit přesnosti předpovídaných hodnot se skutečnými i numericky. Oba tyto skripty jsou na přiloženém médiu. Tato kapitola je ukončena zjednodušeným blokovým schématem pro upřesnění celého postupu praktické části, jak je vidět na obrázku 48.



Obr. 47 Predikce neznámých hodnot provozu



Obr. 48 Blokové schéma praktické části

10 PARAMETRY A METODY VYTVOŘENÝCH SÍTÍ

V této kapitole si shrneme, jaké metody, trénovací a transformační funkce byly použity u výše zmíněných vytvořených neuronových sítí.

10.1 Metoda Min – max

Pro normalizaci dat v podkapitole 8.2 byla použita metoda Min- max. Bylo k ní přistoupeno z důvodu schopnosti trénování a testování neuronových sítí. Protože, pokud je rozsah trénovacích hodnot příliš velký, tak není zaručeno správné natrénování sítě. Když se neuronové síti předloží množina vstupních dat z jiného rozsahu, než v jakém byla natrénována, tak dochází k nesprávné predikci. Varianta jak tento problém eliminovat je normalizace trénovacích hodnot.

Odchycená data jednotlivých provozů po operaci agregace byla znormována tak, že se po znormování hodnoty agregovaných dat pohybují v intervalu $<0, 1>$. K transformaci hodnot do intervalu $<0, 1>$ pomocí metody Min-max lze využít rovnici 27, viz skript na přiloženém médiu:

$$x'_i = \frac{x_i - \min(x_1 \dots x_i)}{\max(x_1 \dots x_i) - \min(x_1 \dots x_i)}, \quad (27)$$

kde $\min(x_1 \dots x_i)$ je nejmenší hodnota množiny $(x_1 \dots x_i)$ a $\max(x_1 \dots x_i)$ je největší hodnota množiny $(x_1 \dots x_i)$.

10.2 Střední čtvercová chyba MSE

Na porovnávání přesností predikovaných hodnot se skutečnými byla zvolena metoda MSE (Mean Squared Error). Tato metoda vypočítává rozdíl mezi skutečnými a předpovídanými hodnotami. Výpočet střední čtvercové chyby lze uplatnit pomocí vztahu 28 viz skript na přiloženém médiu[24]:

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - y(p)_t)^2, \quad (28)$$

kde y_t nám udává skutečnou hodnotu uvažované časové řady v čase t ($t = 1, \dots, n$) a $y(p)_t$ je předpověď hodnoty y_t v čase t . Rozdíl $(y_t - y(p)_t)$ reprezentuje předpovědní chybu.

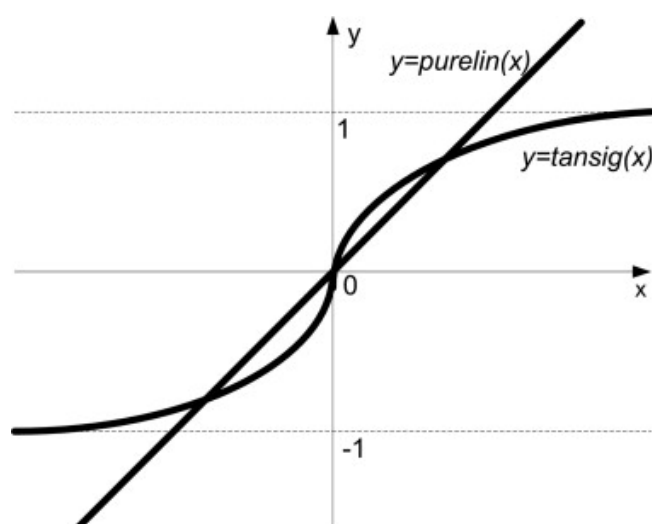
10.3 Trénovací funkce TRAINBFG

Na natrénování neuronových sítí byla zvolena trénovací funkce TRAINBFG. Hlavní účel této funkce je, že postupně mění hodnoty vah mezi jednotlivými neurony tak, aby trénovací chyba postupně klesala. Tato funkce využívá BFGS quasi-Newton metodu. Více k této metodě a k jejímu fungování můžeme nalézt v dokumentaci programu Matlab při zadání příkazu *doc trainbfg* v Command Window.

K natrénování neuronových sítí byly v průběhu experimentální části vyzkoušeny i jiné trénovací funkce, ale ty však neměly tak dobré výsledky, proto bylo do nich opuštěno a byla vybrána za nejvhodnější právě funkce TRAINBFG.

10.4 Transformační funkce vrstvy

Transformační funkce vypočítává výstup vrstvy ze vstupu vrstvy neuronové sítě. Při návrhu byly použity funkce Tansig a Purelin. První funkce ze zmíněných Tansig představuje funkci hyperbolický tangens ($y = \text{tansig}(x)$). Druhá funkce Purelin představuje lineární transformační funkci ($y = \text{purelin}(x)$). Obě tyto funkce jsou zobrazeny na obrázku 49.



Obr. 49 Transformační funkce Tansig a Purelin

[26]

11 VÝBĚR VHODNÉHO MODELU SÍTĚ

Na testování predikce síťového provozu (traffiku) pomocí rekurentních neuronových sítí v programu Matlab a vývojovém prostředí NN Toolbox byly vybrány tyto neuronové sítě: Elmanova síť, obecná rekurentní síť (LRN) a NARX síť. Tyto sítě byly natrénovány s učitelem. Pro predikci síťového provozu pomocí neuronových sítí byly odchyceny a použity čtyři typy dat a to konkrétně celkový download, celkový upload, download aplikace ICQ a upload aplikace ICQ. Jelikož před samotným testováním schopnosti neuronových sítí předpovídat síťový provoz nebylo jasné, které topologie budou nejvhodnější pro predikci, tak bylo přistoupeno k testování několika druhů různých topologií neuronových sítí. Každá neuronová síť se specifickou topologií byla natrénována 2x a poté odsimulována. Dosažené výsledky těchto dvou měření byly zprůměrovány. Tyto výsledky jednotlivých měření pak posloužily k určení nejvhodnější neuronové sítě s danou topologií k predikci zvoleného typu síťového provozu.

Zvolené testované topologie pro dvouvrstvou Elmanovu síť byly: 4-1, 8-1, 12-1, 16-1 a 20-1, kde první číslo zde udává počet neuronů v první vrstvě a druhé číslo popisuje počet neuronů ve vrstvě druhé. Ve vstupní vrstvě byla použita přenosová funkce Tansig a ve výstupní vrstvě byla zvolena přenosová funkce Purelin. Následující testované topologie pro třívrstvé LRN a NARX sítě byly: 4-4-1, 4-8-1, 4-12-1, 8-4-1, 8-8-1, 8-12-1, 12-4-1, 12-8-1, 12-12-1. Ve všech vrstvách byla vybrána přenosová funkce Tansig.

Zprůměrované dosažené Střední čtvercové chyby MSE (Mean Squared Error) vyjádřené v % z predikovaných 1500 hodnot pro jednotlivé sítě a jejich topologie pro daný síťový provoz jsou vidět z tabulek 1, 2 a 3.

Tab. 1 Dosažené Střední čtvercové chyby (MSE) v [%] pro různé topologie Elmanovy sítě

| Typ provozu | MSE 4-1 | MSE 8-1 | MSE 12-1 | MSE 16-1 | MSE 20-1 |
|--------------|---------------|---------|----------------|---------------|----------|
| Download | 7,0634 | 7,3518 | 7,4712 | 7,8791 | 8,4720 |
| Upload | 9,9069 | 8,1253 | 7,5574 | 10,5434 | 8,3146 |
| Download ICQ | 4,8632 | 5,0904 | 4,9688 | 4,8596 | 4,9104 |
| Upload ICQ | 12,3626 | 12,4821 | 11,6994 | 12,2432 | 12,2549 |

Z tabulky 1 je vidět, že při predikci downloadu bylo dosaženo nejmenší průměrné chyby pomocí topologie 4-1. Při predikci uploadu bylo nejmenší chyby dosaženo u topologie 12-1. Download aplikace ICQ nám nejlépe predikovala topologie 16-1. U uploadu aplikace

ICQ jsme dosáhli nejmenší chyby s topologií 12-1. Dále je z tabulky zřejmé, že při predikci downloadu bylo dosaženo menších chyb, než při predikci uploadu.

Tab. 2 Dosažené Střední čtvercové chyby (MSE) v [%] pro různé topologie LRN sítě

| Typ provozu | MSE 4-4-1 | MSE 4-8-1 | MSE 4-12-1 | MSE 8-4-1 | MSE 8-8-1 | MSE 8-12-1 | MSE 12-4-1 | MSE 12-8-1 | MSE 12-12-1 |
|--------------|-----------|-----------|------------|---------------|----------------|---------------|------------|------------|-------------|
| Download | 6,4068 | 6,9926 | 6,7268 | 6,0954 | 6,7030 | 7,2020 | 7,0102 | 6,7134 | 7,7084 |
| Upload | 9,4055 | 10,1346 | 8,3076 | 8,0090 | 11,9918 | 10,7809 | 12,0727 | 9,5074 | 10,0667 |
| Download ICQ | 5,1824 | 6,3322 | 5,1361 | 5,4820 | 4,8724 | 4,5391 | 4,6784 | 5,2236 | 5,8484 |
| Upload ICQ | 12,4452 | 12,0278 | 13,6412 | 11,7301 | 11,5139 | 12,1074 | 13,0847 | 14,2781 | 13,3753 |

Z tabulky 2 je vidět, že nejmenší průměrné chyby při predikci downloadu bylo dosaženo s topologií 8-4-1. Při predikci uploadu bylo dosaženo nejmenší chyby pomocí topologie 8-4-1. Dále je zde vidět, že nejmenší průměrné chyby při predikci downloadu aplikace ICQ bylo dosaženo s topologií 8-12-1, pro upload této aplikace pak s topologií 8-8-1. Když se podíváme na celkovou predikci pomocí sítě LRN, tak bylo zde dosahováno o něco vyšších chyb než při predikci pomocí Elmanovy sítě, až na predikci downloadu, kdy síť LRN lépe predikovala síťový provoz než Elmanova síť. Dále je zde ze získaných hodnot vidět, že nejmenší průměrné chyby u downloadu a uploadu bylo dosaženo stejnou topologií neuronové sítě a to 8-4-1.

Tab. 3 Dosažené Střední čtvercové chyby (MSE) v [%] pro různé topologie NARX sítě

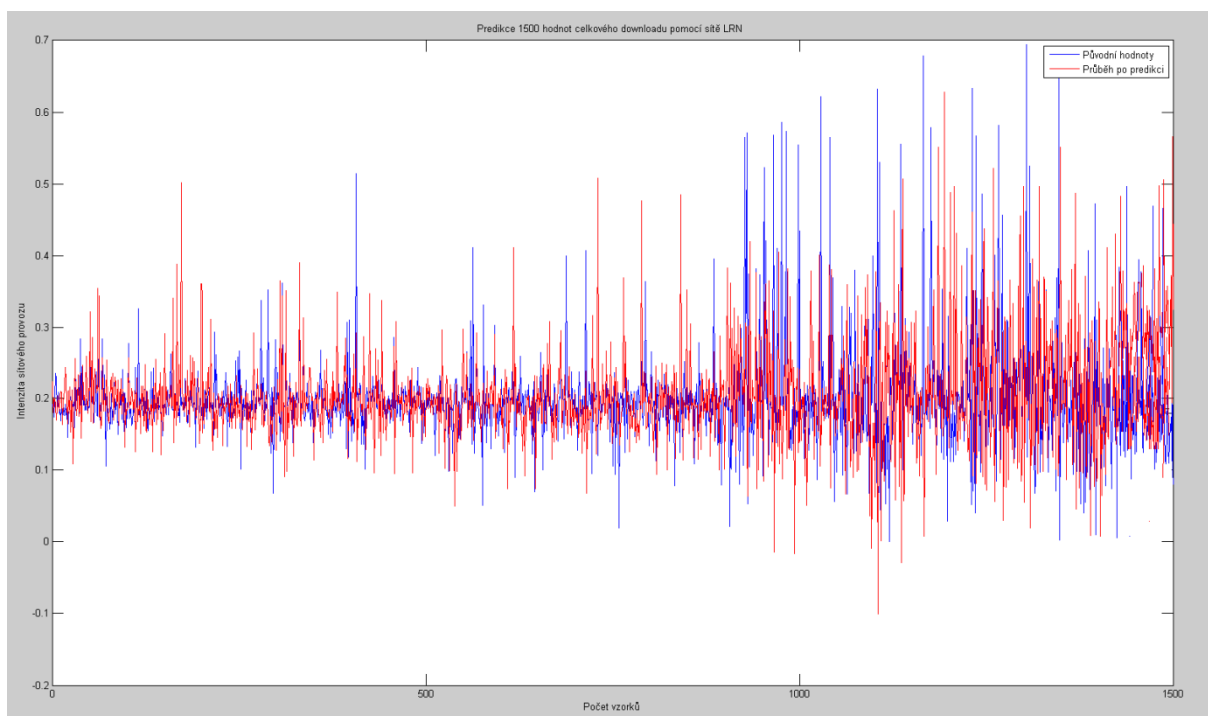
| Typ provozu | MSE 4-4-1 | MSE 4-8-1 | MSE 4-12-1 | MSE 8-4-1 | MSE 8-8-1 | MSE 8-12-1 | MSE 12-4-1 | MSE 12-8-1 | MSE 12-12-1 |
|--------------|-----------|----------------|---------------|---------------|-----------|------------|------------|------------|-------------|
| Download | 7,7654 | 7,1758 | 7,0910 | 6,9998 | 7,0682 | 7,3338 | 7,0186 | 7,6011 | 8,7794 |
| Upload | 10,6301 | 9,2235 | 8,7284 | 11,1448 | 9,9473 | 9,4436 | 9,9171 | 10,8721 | 9,3073 |
| Download ICQ | 6,2978 | 4,3148 | 4,3906 | 5,8508 | 4,3356 | 4,7572 | 4,6726 | 4,7364 | 4,9956 |
| Upload ICQ | 12,1047 | 11,5142 | 12,5883 | 13,1906 | 12,3414 | 12,6161 | 12,2718 | 11,6509 | 13,0784 |

Z naměřených výsledků v tabulce 3 můžeme vyčíst, že nejmenší průměrné chyby pro download u sítě NARX bylo dosaženo s topologií 8-4-1. Nejmenší průměrná chyba pro upload byla dosažena s topologií 4-12-1. Pro download a upload aplikace ICQ bylo dosaženo

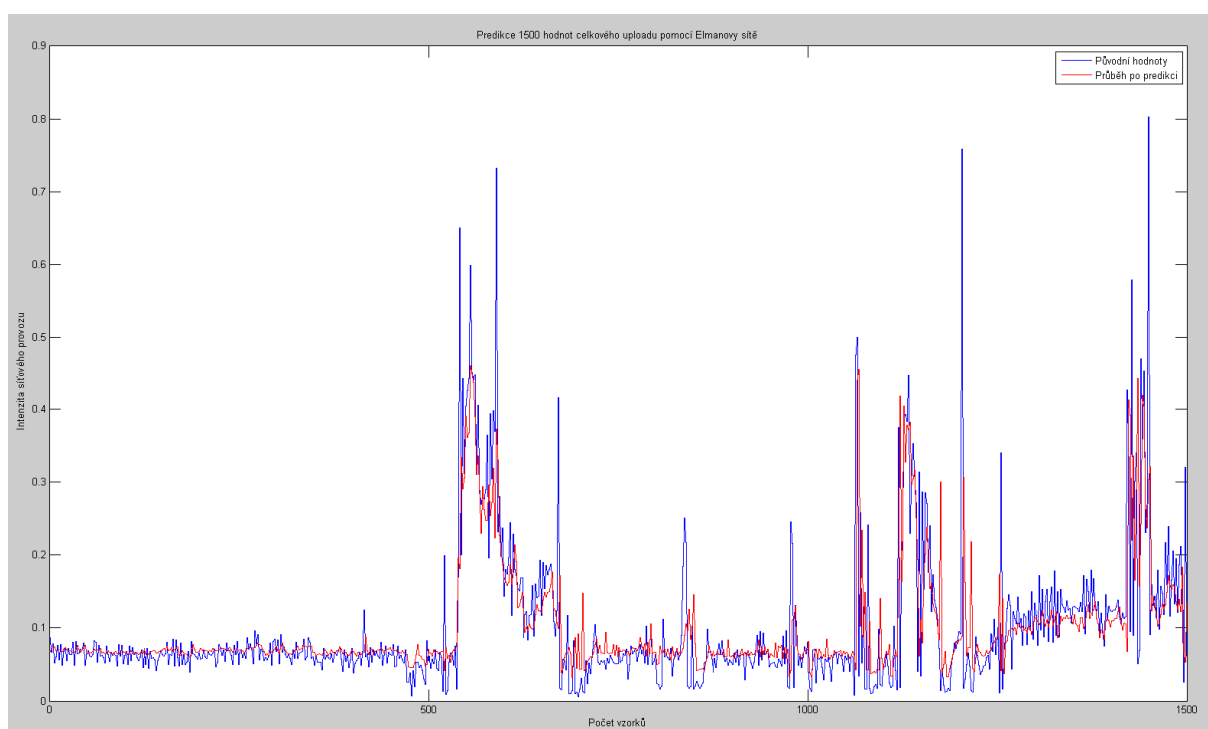
nejmenší průměrné chyby s topologií 4-8-1. Celková predikce pomocí sítě NARX měla horší výsledky, čili větší průměrné chyby než předešlá síť LRN.

11.1 Zhodnocení praktické části

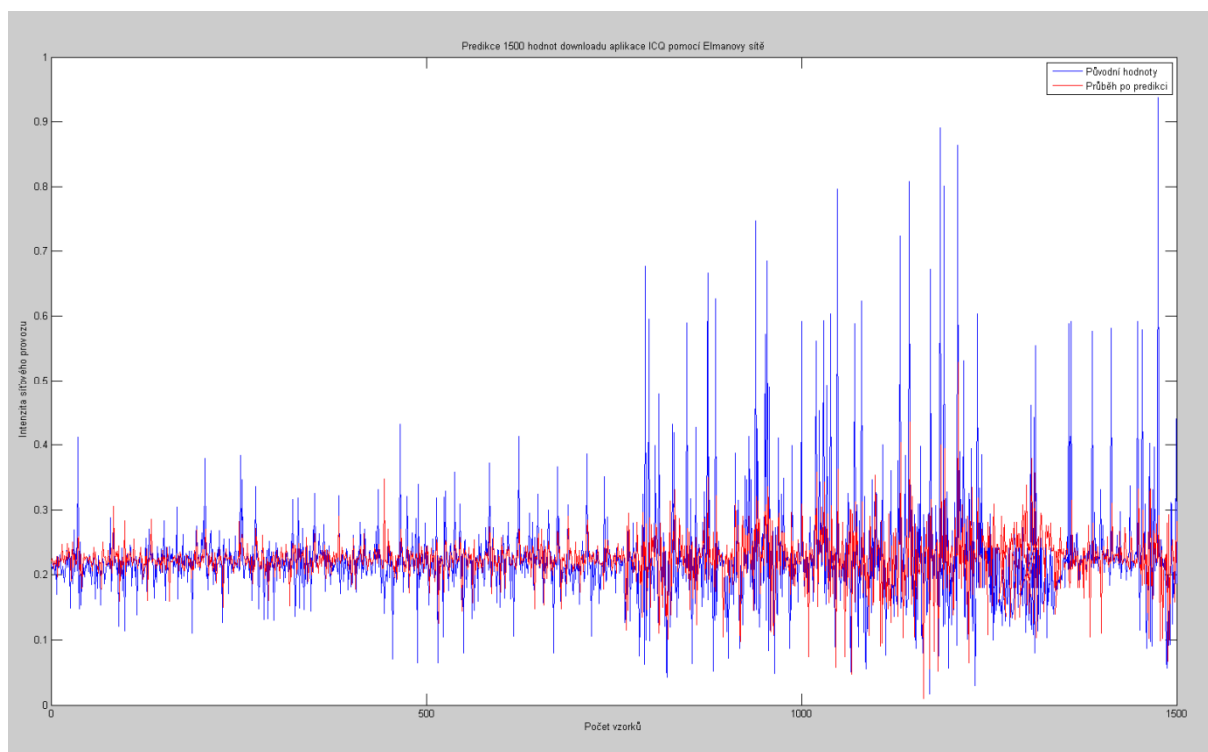
Z naměřených výsledků krátkodobé predikce na 1500 hodnotách síťového provozu je vidět, že download predikovala nejlépe třívrstvá obecná rekurentní síť LRN s topologií 8-4-1, kdy průměrná chyba byla 6,0954 %. Pro upload dosahovala nejlepších výsledků, čili nejmenších průměrných chyb Elmanova dvouvrstvá rekurentní síť s topologií 12-1 a průměrnou chybou 7,5574 %. Na další síťový provoz konkrétně download aplikace ICQ nejlépe zareagovala a predikovala Elmanova síť s topologií 16-1 a dosaženou chybou 4,8596 %. Upload této aplikace opět nejlépe predikovala tato síť s topologií 12-1 a průměrnou chybou 11,6694 %. Tento jev může být dán jednoduchostí Elmanovy dvouvrstvé sítě. U třívrstvých použitých neuronových rekurentních sítí jmenovitě LRN a NARX jejich variabilita (volnost při návrhu) může být při predikci síťového provozu na škodu, protože s větším počtem neuronů (vrstev) je totiž obtížnější nastavit váhy mezi jednotlivými neurony, tak aby síť predikovala požadovaný síťový provoz správně. Celkově můžeme tyto dosažené výsledky shrnout, že nejlepší výsledky dosahovala dvouvrstvá Elmanova síť, která až na download dosahovala nejlepších výsledků predikce. Download predikovala lépe obecná rekurentní LRN síť, což je pochopitelné, protože se jedná v podstatě o obecný zápis struktury Elmanovy sítě, která tvoří podmnožinu LRN. Dále si zde můžeme povšimnout, že všechny tři neuronové sítě lépe predikovaly download, než upload. Grafické prezentace výsledků pro nejvhodnější neuronové sítě a jejich topologie pro predikci daného typu provozu jsou na obrázcích 50-53 (příloha A1-A4).



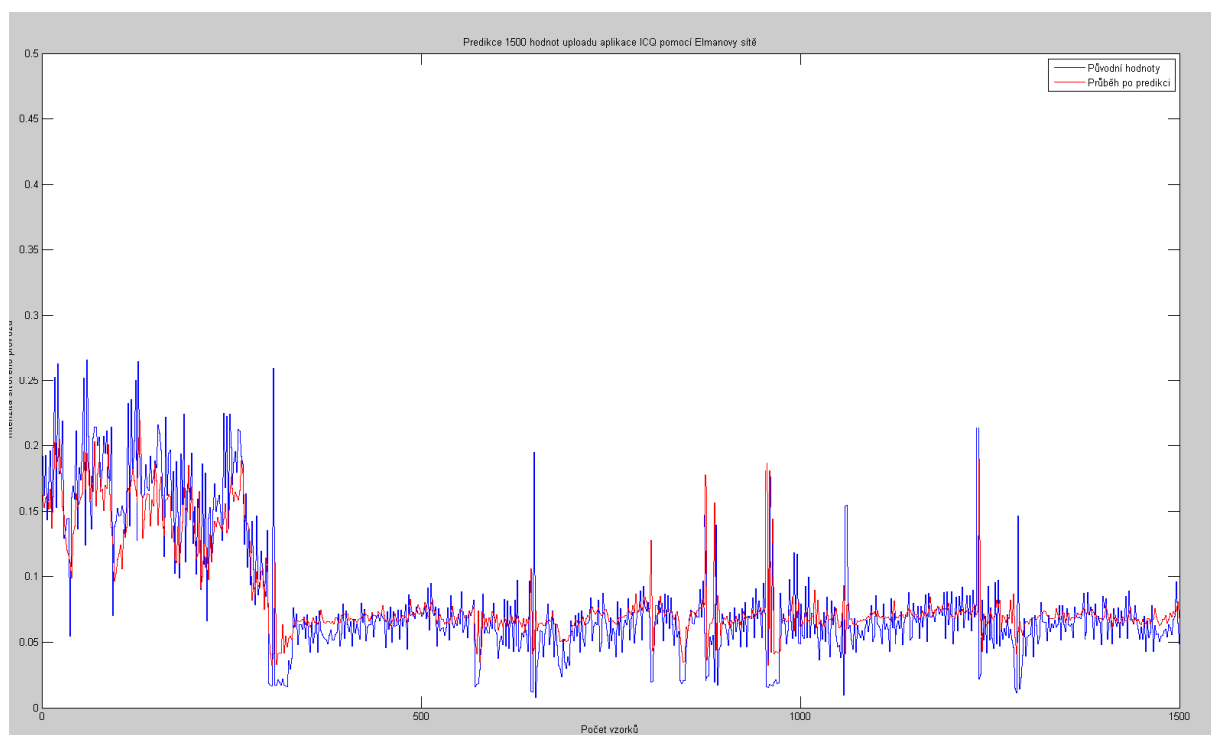
Obr. 50 Predikce downloadu pomocí sítě LRN



Obr. 51 Predikce uploadu pomocí Elmanovy sítě



Obr. 52 Predikce downloadu aplikace ICQ pomocí Elmanovy sítě



Obr. 53 Predikce uploadu aplikace ICQ pomocí Elmanovy sítě

12 ZÁVĚR

V této diplomové práci je první kapitola zaměřena na síťový provoz. Jsou zde vysvětleny pojmy, jako je přenosová rychlost, přenosový kanál a šířka pásma. Dále je zde vysvětlen princip časového, frekvenčního, kódového a vlnového multiplexu. Poslední část první kapitoly se věnuje možnosti alokace šířky pásma.

V úvodu druhé kapitoly jsou nastíněny statické vlastnosti síťového provozu. A dále je zde rozebrán pojem samopodobnost. Na příkladu síťového provozu, který obecně vykazuje samopodobné charakteristiky je uvedeno matematické vyjádření samopodobnosti.

Třetí kapitola se věnuje možnostem predikce síťového provozu. Jsou zde uvedeny a popsány tři hlavní metody, pomocí kterých lze síťový provoz predikovat. Jedná se o FARIMA model, Teorii chaosu s Lyapunovým exponentem a Neuronové sítě. Těmto sítím se podrobně věnuje následující kapitola.

Ve čtvrté kapitole jsou popsány základní vlastnosti Neuronové sítě, jako je učení a schopnost generalizace. Dále je zde popsán biologický neuron a i jeho matematický model. Jsou zde uvedeny jejich schémata, vlastnosti a principy. Závěr čtvrté kapitoly nám uzavírají aktivační funkce, které řadíme mezi prahové funkce a ty nám udávají, jestli je neuron v aktivním či pasivním stavu.

V následující kapitole je zmíněno, že ne každá neuronová síť je vhodná k predikci síťového provozu. V této kapitole je rozebráno základní dělení neuronových sítí. A jsou zde zvoleny vhodné rekurentní neuronové sítě hodící se pro tuto oblast predikce. Dále je zde rozebrána rekurentní Elmanova síť, NARX síť a Obecná LRN síť.

Šestá kapitola je věnována využití neuronových sítí v praxi. Jsou zde popsány obory, které tyto sítě hojně využívají. A je to například letecká doprava, zdravotnictví, finančnictví, robotika...

Další kapitola nám pojednává o výhodách a nevýhodách predikce síťového provozu pomocí neuronových sítí. Mezi výhody jednoznačně patří princip učení, generalizace a v poslední řadě cenová dostupnost. K nevýhodám se řadí velké hardwarové nároky na počítač při trénování neuronové sítě, dále dopředu nevíme, jaký bude výsledek predikce (úspěch/neúspěch).

Praktická část osmé kapitoly se dělí na dvě části. Kde v první je popsáno zachycení a následné vhodné zpracování testovacích dat síťového provozu potřebných pro predikci, která se provede v Matlab Neural Network Toolbox. Vše je přehledně uvedeno na obrázcích. V této kapitole je zmíněno na jakých datech bylo testování jednotlivých sítí prováděno, dále je

zde uvedeno jaké vhodné sítě se pro predikci použily. Tato kapitola je pak ukončena obecným návrhem predikční části za pomoci neuronových sítí, kde vystupují do popředí hlavní jejich vlastnosti a to pojem učení a generalizace.

Devátá kapitola nám pojednává o samotném vytvoření, natrénování a otestování neuronových sítí v prostředí NN Toolbox. Vše je zde přehledně, detailně popsáno po jednotlivých krocích a uvedeno na obrázcích. Tuto kapitolu nám uzavírá celkové blokové schéma praktické části.

Náplní desáté kapitoly jsou parametry a metody vytvořených neuronových rekurentních sítí. Je zde popsána normalizační metoda Min-max, její princip a matematické vyjádření. Dále je zde uvedena Střední čtvercová chyba MSE, která slouží na porovnávání přesností predikovaných hodnot se skutečnými. V poslední řadě jsou zde zmíněny trénovací a transformační funkce.

V poslední kapitole jsou popsány zvolené testované topologie neuronových sítí potřebné pro predikci síťového provozu. Získané výsledky jsou přehledně zpracovány do tabulek a vyneseny do grafických závislostí. Dále je zde provedeno vyhodnocení jednotlivých testovaných sítí. Dosažené výsledky predikce jednotlivých neuronových sítí se zvolenými topologiemi pro jednotlivé síťové provozy (tab. 1-3) byly porovnány a výsledkem bylo zjištění, že pro predikci síťového provozu downloadu z testovaných sítí je nejvhodnější obecná rekurentní síť (LRN) s topologií 8-4-1. Pro síťový provoz upload se nejvíce hodí Elmanova síť s topologií, kde v první vrstvě má 12 neuronů a v druhé 1 neuron. Dále pak na další síťový provoz konkrétně download aplikace ICQ nejlépe zareagovala a predikovala Elmanova síť s topologií, kde v první vrstvě bylo 16 neuronů a v druhé 1 neuron. Na poslední typ síťového provozu upload aplikace ICQ se opět jevila nejvhodnější tato neuronová síť s variantou, při níž bylo v první vrstvě 12 neuronů a druhé 1 neuron.

Výsledný model těchto sítí lze v praxi například použít pro dynamickou alokaci šířky pásma jednotlivých front u frontového systému na směrovačích. Rekurentní neuronová síť může totiž predikovat zvýšený provoz sítě (např. při real-time přenosech), v jehož důsledku by mohlo dojít k zahazování paketů v jedné frontě, zatímco druhá by byla poloprázdná, či zcela prázdná. Obecně lze tedy říci, že neuronové sítě slouží svou činností k vylepšení systémů.

POUŽITÁ LITERATURA

- [1] ŠEBESTA V., SMÉKAL, Z. *Signály a soustavy: skripta*. Brno: FEKT VUT v Brně, 2002. 167 s.
- [2] PETERKA, J. Jiří Peterka: *Archiv článků a přednášek*. [online]. [cit. 2012-10-10]. Dostupné z: <http://www.earchiv.cz/a91/a143c110.php3>
- [3] Bartošek P.: *Zavedení WDM v prostředí České republiky*. Telekomunikace a podnikání, č.11/99, str. 6.
- [4] ŠENOVSKÝ, P. Pavel Šenovský: *Základy informatiky*. [online]. [cit. 2012-10-13]. Dostupné z: <http://homen.vsb.cz/~sen76/inform/inform.html>
- [5] HANUS, S. *Rádiové a mobilní komunikace: skripta*. Brno: FEKT VUT v Brně, 2005. 124 s.
- [6] ČÍKA, P. *Multimediální služby: skripta*. Brno: FEKT VUT v Brně, 2012. 127 s
- [7] KACÁLEK, J., MÍČA, I. *Nelineární analýza a predikce síťového provozu*. Brno: Ústav telekomunikací, FEKT VUT v Brně, 2009. 6 s. Dostupné na internetu: <<http://www.elektrorevue.cz/file.php?id=200000389-ae332af2d1>>.
- [8] ČLUPEK, V. *Nelineární analýza a predikce síťového provozu*. Diplomová práce, FEKT VUT v Brně, 2012.
- [9] WIKIPEDIE [online]. 8. 11. 2012 [cit. 2012-11-08]. *Soběpodobnost*. Dostupný z <http://cs.wikipedia.org/wiki/Soběpodobnost>.
- [10] KACÁLEK, J. *Metody modelování a predikce síťového provozu: pojednání o disertační práci*. Brno: FEKT VUT v Brně, 2008. 20 s.
- [11] CROVELLA, E., BESTAVROS, A.: Computer Science Department, Boston University, *Self-Similarity in World Wide Web Traffic – Evidence and Possible Causes*
- [12] LELAND, W., TAQQU, M., WILSON, D.: *IEEW/ACM transactions on networking*, vol. 2, no. 1. February 1994
- [13] LETLANT, W. E., MEMBER, IEEE.: *On the Self-similar Nature of Ethernet traffic* (Extended version) vol. 2, February 1994
- [14] XUN-YI, R., YANG-YU, JUN-FENG, Z., LI-MA, XIAO-DONG, M. *Parameter Estimation and Application of Time-varying FARIMA Model*. International Journal of Advancements in Computing Technology, vol. 3, no. 3, April 2011

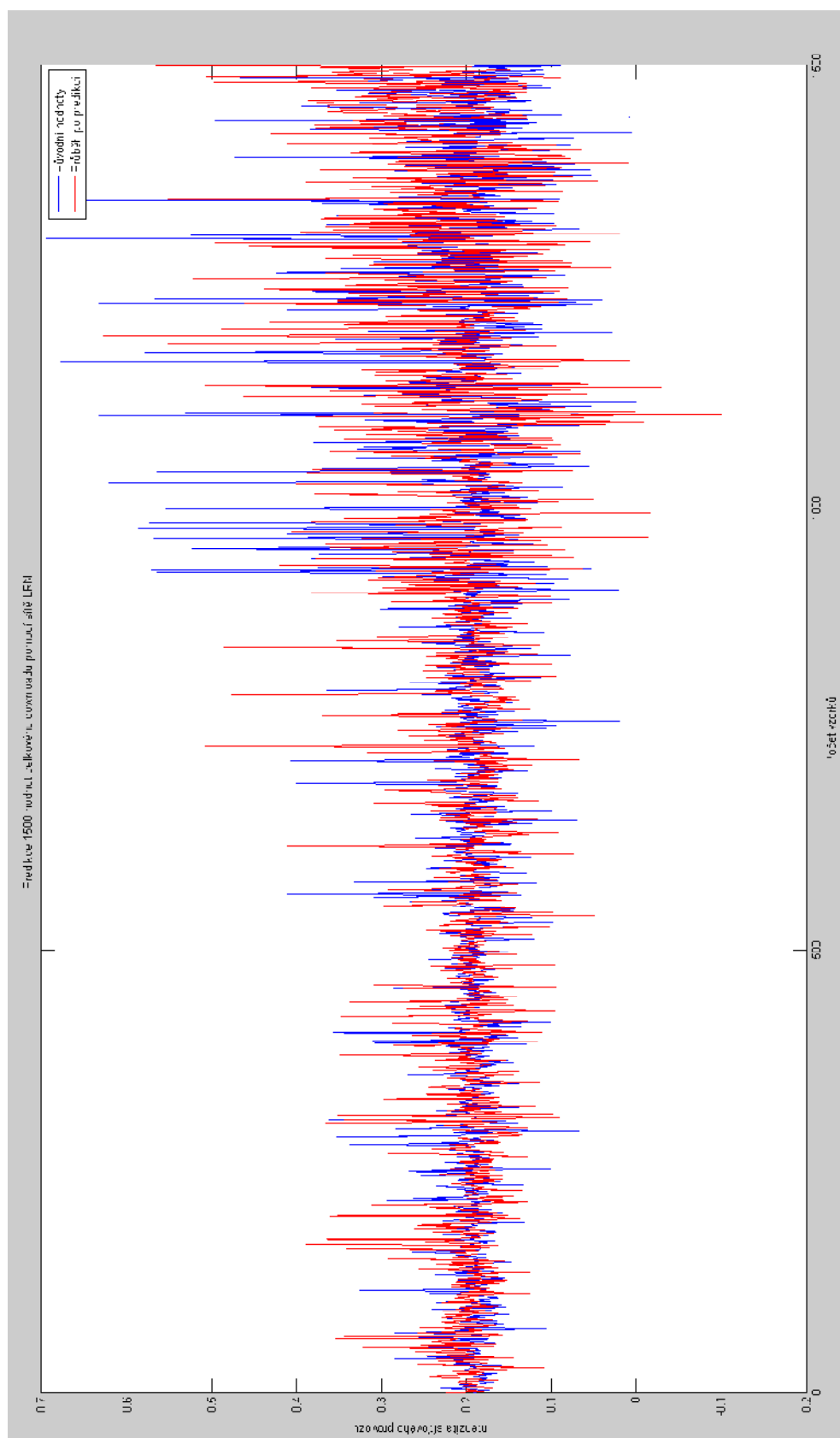
- [15] RACKOVÁ, L. *Metody vykreslování chaotických atraktorů*. Brno: Masarykova universita, fakulta Informatiky, 2009. 42 s. Dostupné na internetu: <http://is.muni.cz/th/173268/fi_b/traktory.pdf>.
- [16] KLÁN, P. Petr Klán: *Neural Networks for Chemists: Univerzitní přednášky*[online]. [cit. 2012-11-13]. Dostupné z: <http://www.uivt.cas.cz/~pklan/>
- [17] JIRSÍK, V. *Umělá inteligence: skripta*. Brno: FEKT VUT v Brně, 2012.
- [18] VONDRÁK, I. *Neuronové sítě: skripta*. Ostrava: VŠB v Ostravě, 2001. 56 s
- [19] VOLNÁ, E. *Neuronové sítě 1: skripta*. Ostrava: VŠB v Ostravě, 2002. 85 s
- [20] HORKÝ, L., BŘINDA, K. *Neuronové sítě: skripta*. Praha: ČVUT v Praze, 2009
- [21] MATLAB Central. [online]. [cit. 2012-12-06]. Dostupné z: <http://www.mathworks.com/matlabcentral/>
- [22] Professional Web Hosting. [online]. [cit. 2012-12-06]. Dostupné z: <http://www.ps-aruba.com/hosting/>
- [23] Science and Curation: the New Practice of Web 2.0. [online]. [cit. 2012-12-06]. Dostupné z: <http://blog.mysciencework.com/en/2012/10/29/science-and-curation-the-new-practice-of-web-2-0.html>
- [24] Learn Photoshop with Tut5's Adobe Photoshop Tutorials- Desktop PC: Desktop PC Icon. [online]. [cit. 2012-12-06]. Dostupné z: http://tut5.com/display.php?t=desktop_pc
- [25] HANČLOVÁ, J., TVRDÝ, L. *Úvod do analýzy časových řad*. Ostrava: Fakulta ekonomická, katedra matematických metod v ekonomice a katedra regionální ekonomiky, VŠB-TU Ostrava, 2003. 34 s. Dostupné na internetu: <http://gis.vsb.cz/panold/Skoleni_Texty/TextySkoleni/AnalyzaCasRad.pdf>.
- [26] Energy Conversion and Management: Modeling and prediction of Turkey's electricity consumption using Artificial Neural Networks [online]. [cit. 2013-04-20]. Dostupné z: http://www.sciencedirect.com/science?_ob=ShoppingCartURL&_method=add&_eid=1-s2.0-S0196890409002404&originContentFamily=serial&_origin=article&_acct=C000228598&_version=1&_userid=10&_ts=1366526744&md5=9811b8a569b99e20fc6d5d774c4e8f45

SEZNAM PŘÍLOH

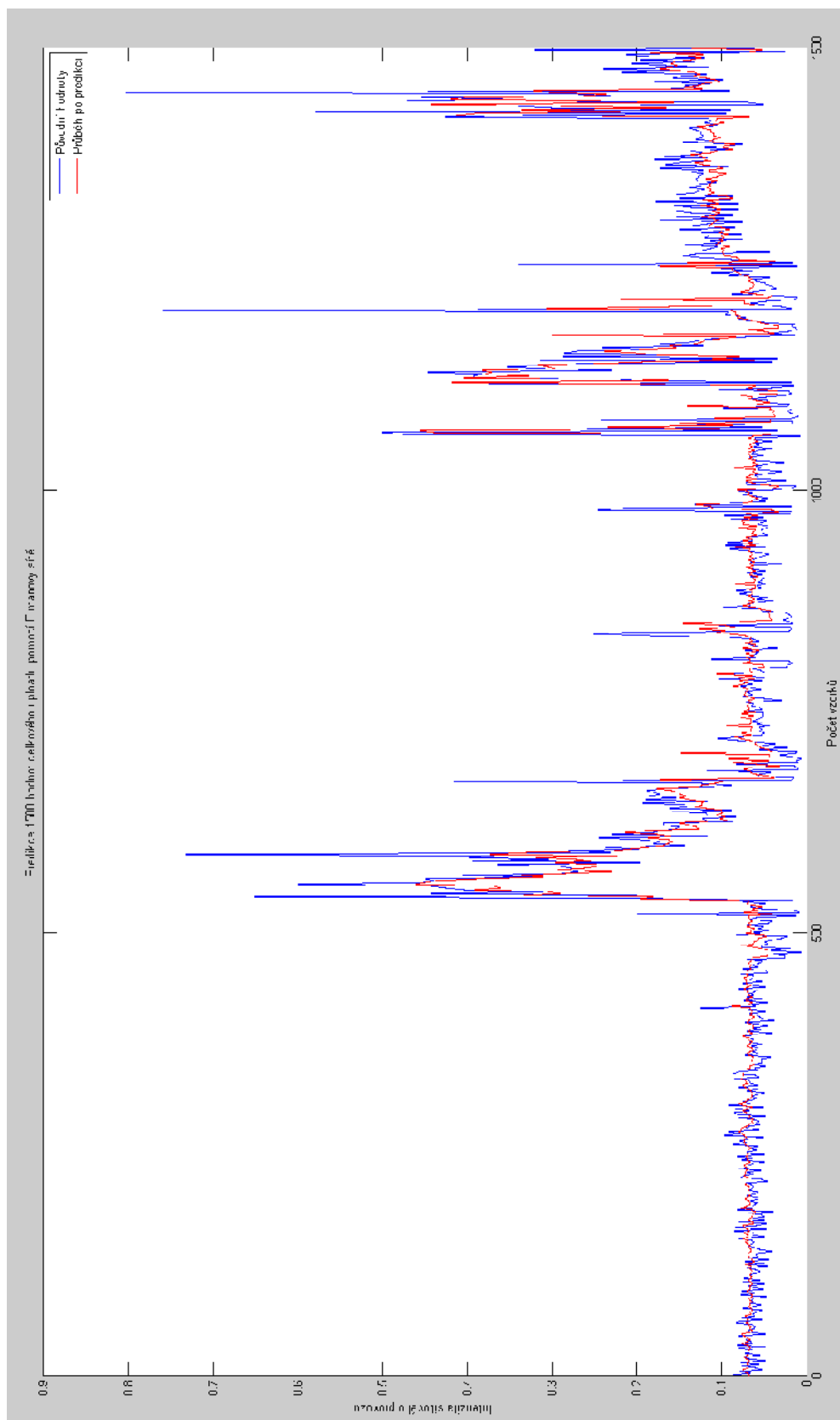
| | | |
|----------|--------------------------------------------------|-----------|
| A | Zhodnocení praktické části | 68 |
| A.1 | Predikce 1500 hodnot celkového downloadu..... | 76 |
| A.2 | Predikce 1500 hodnot celkového uploadu..... | 77 |
| A.3 | Predikce 1500 hodnot downloadu aplikace ICQ..... | 78 |
| A.4 | Predikce 1500 hodnot uploadu aplikace ICQ..... | 79 |
| | | |
| B | Obsah přiloženého média | 80 |

A PREDIKCE SÍŤOVÉHO PROVOZU

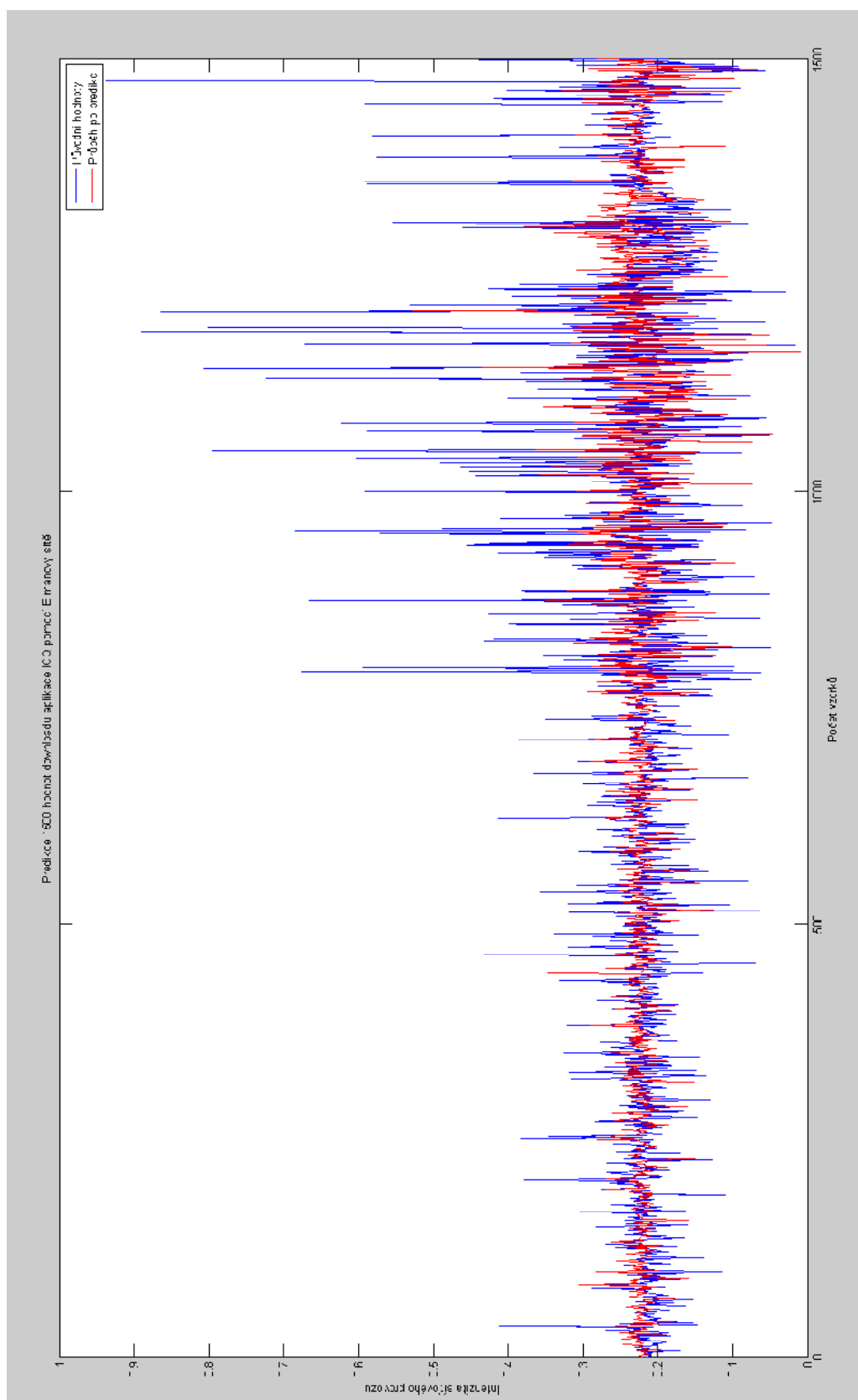
A.1 Predikce 1500 hodnot celkového downloadu pomocí sítě LRN



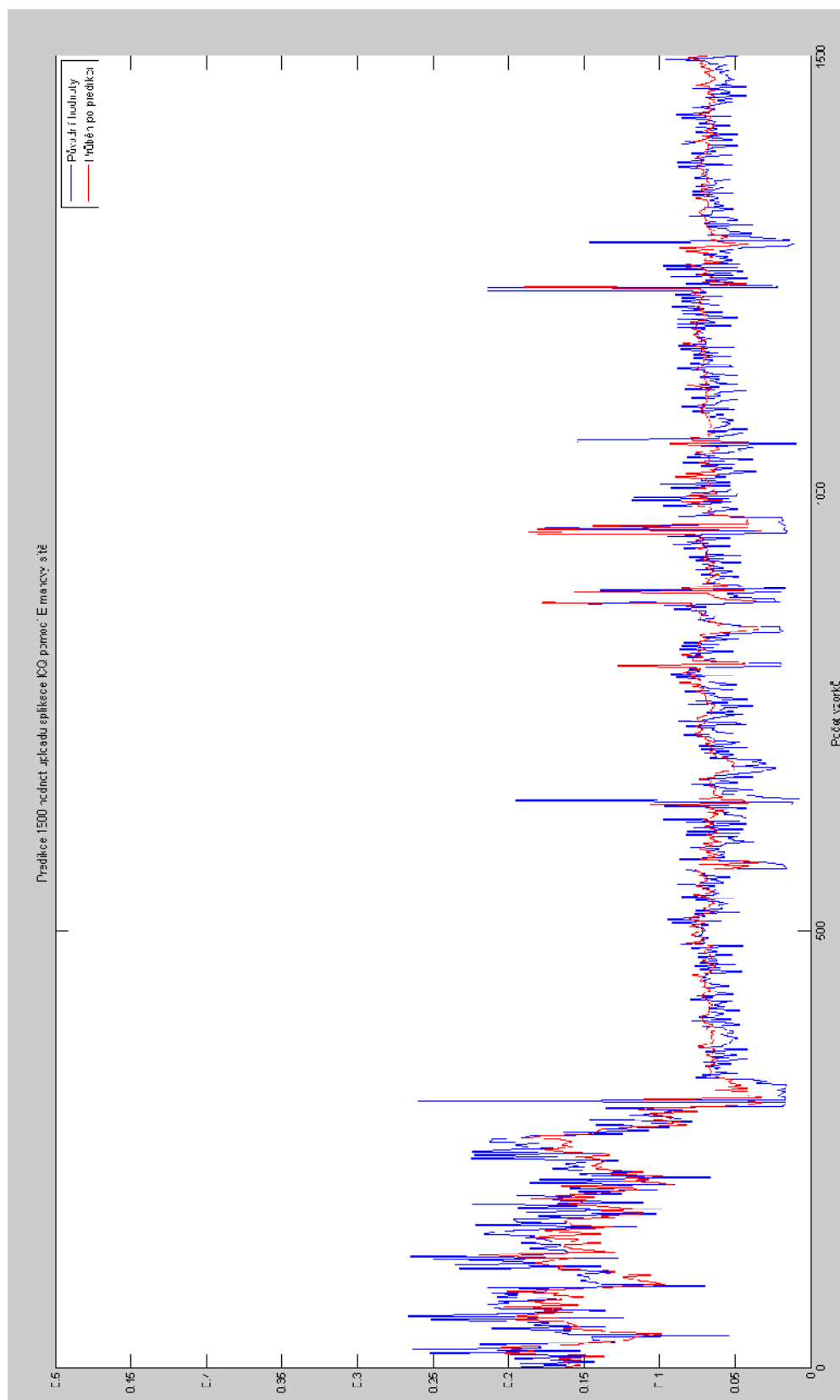
A.2 Predikce 1500 hodnot celkového uploadu pomocí Elmanovy sítě



A.3 Predikce 1500 hodnot downloadu aplikace ICQ pomocí Elmanovy sítě



A.4 Predikce 1500 hodnot uploadu aplikace ICQ pomocí Elmanovy sítě



B OBSAH PŘÍLOŽENÉHO MÉDIA

\Diplomová práce.pdf

\Matlab\ - obsahuje skripty využívané v prostředí Matlab

- `agregace_normalizace_400ms.m`
- `chyba.m`
- `vysledek_predikce_test_vyst_a_network1_outputs.m`
-

\Data\ - obsahuje soubory jednotlivých síťových provozů

- `celkovy_download`
- `celkovy_upload`
- `download_icq`
- `upload_icq`