



MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV TEORETICKÉ FYZIKY A ASTROFYZIKY



Klasifikace světelných křivek s pomocí neuronových sítí

Zdeněk Janák

Diplomová práce

Vedoucí dip. práce: Mgr. Viktor Votruba, Ph.D. Brno 2012

Bibliografický záznam

Autor: Bc. Zdeněk Janák
Přírodovědecká fakulta, Masarykova univerzita
Ústav teoretické fyziky a astrofyziky

Název práce: Klasifikace světelných křivek
s pomocí neuronových sítí

Studijní program: Fyzika

Studijní obor: Teoretická fyzika a astrofyzika

Vedoucí práce: Mgr. Viktor Votruba, Ph.D.

Rok obhajoby: 2012

Klíčová slova: klasifikace, predikce, světelné křivky,
chaos, umělá neuronová síť, Be hvězdy

Bibliographic entry

Author: Bc. Zdeněk Janák
Faculty of Science, Masaryk University
Department of Theoretical Physics and Astrophysics

Title of thesis: Classification of light curves
using neural networks

Degree programme: Physics

Field of study: Theoretical Physics and Astrophysics

Supervisor: Mgr. Viktor Votruba, Ph.D.

Year of defence: 2012

Keywords: classification, prediction, light curves,
chaos, artificial neural network, Be stars

Poděkování

Tato práce vznikla v rámci výzkumu zaměřeného na studium Be hvězd vědeckého týmu pracovníků stelárního oddělení Astronomického ústavu Akademie věd České republiky v Ondřejově. Chtěl bych zde poděkovat především Viktoru Votrubovi za volnou ruku a pevné nervy při vedení práce, Lence Kotkové za sestavení katalogu Be hvězd a Janu Fuchsovi za předpřípravu dat.

Dále bych chtěl poděkovat svým rodičům Jarmile a Milanu Janákovým za podporu při studiu a nechápavé kroucení hlavou při probdělých nocích strávených programováním. A také své přítelkyni Dominice Kalasové za ustavičné popichování.

V neposlední řadě patří můj dík rozsáhlé komunitě vývojářů otevřeného a svobodného softwaru, s jehož výhradní pomocí jsem tuto práci připravoval.

Klasifikace světelných křivek s pomocí neuronových sítí, jejímž autorem je
Zdeněk Janák, podléhá licenci [Creative Commons Uveďte autora 3.0 Česko](#).

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně
s použitím citovaných pramenů.

V Brně, dne 11. ledna 2012

Zdeněk Janák, v. r.

Abstrakt

V této práci se věnuji možnostem využití umělých neuronových sítí v astronomii pro analýzu a klasifikaci světelných křivek hvězd. Demonstruji použití umělé neuronové sítě k předpovídání chaotických časových řad. Na základě statistických a geometrických parametrů popisujících světelné křivky Be hvězd tyto klasifikuji v obsáhlém fotometrickém katalogu hvězd. Ukazují, že umělé neuronové sítě představují efektivní nástroj pro masové zpracování dat a třeba je brát jako účinnou alternativu i doplněk klasických statistických metod dobývání znalostí z dat.

Klíčová slova: klasifikace, predikce, chaos, světelné křivky, umělá neuronová síť, Be hvězdy

Abstract

In this thesis, I will be studying the different possibilities of using artificial neural networks in astronomy for analysis and classification of star's light curves. I will demonstrate the use of artificial neural network to predict chaotic time series. Based on statistical and geometric parameters describing light curves of Be stars, I will classify them in large photometrics catalogue of stars. I will present artificial neural networks as an effective tool for batch processing of data. It is necessary to consider them to be an efficient alternative and supplement to conventional statistical methods of data mining.

Keywords: classification, prediction, light curves, chaos, artificial neural network, Be stars

Obsah

1	Úvod	10
1.1	Zadání	10
1.2	Struktura práce	11
1.3	Motivace	11
1.4	Schopnosti umělých neuronových sítí	12
1.5	Využití umělých neuronových sítí	12
1.6	Inspirace přírodou	13
1.7	Historie výzkumu umělých neuronových sítí	14
1.8	Budoucnost využití umělých neuronových sítí	15
2	Umělá neuronová síť	16
2.1	Matematický model neuronu	16
2.1.1	Aktivační funkce	17
2.1.2	Hebbův zákon učení	19
2.1.3	Učící algoritmus neuronu	20
2.2	Vícevrstvá síť perceptronů	20
2.2.1	Topologie umělé neuronové sítě	20
2.2.2	Učení s učitelem	21
2.3	Backpropagation	22
2.3.1	Moment setrvačnosti	23
2.4	Přeučení	23
3	FANN	25
3.1	Fast Artifical Neural Network library	25
3.2	XOR test	26
3.3	Normalizace dat	30
4	Predikce časových řad	31
4.1	Chaos	31
4.2	Časová řada	32
4.3	Rösslerův atraktor	32

5	Klasifikace světelných křivek	37
5.1	Be hvězdy	37
5.2	Katalogy hvězd	37
5.3	Metodika klasifikace	41
5.3.1	Parametry světelných křivek	41
5.3.2	Trénovací množina	43
5.3.3	Testovací množina	43
5.3.4	Topologie umělé neuronové sítě	43
5.3.5	Výsledek klasifikace	47
6	Závěr	48

Kapitola 1

Úvod

You said: Who are you?
A.L.I.C.E.¹ My name is ALICE.

1.1 Zadání

Oficiální zadání sepsané vedoucím práce.

V průběhu posledních let se ukázalo, že metody umělé inteligence se stávají nejen osvědčeným, ale také nezbytným nástrojem pro analýzu astronomických dat v celé širší obecnosti. Mohou pomoci při klasifikaci objektů v situaci, kdy klasické statistické metody selhávají, při předpovídání dalšího chování objektu, zejména jeho světelných křivek či spektrálních změn apod. Praktickým příkladem může být hledání period pulsujících proměnných hvězd, analýza světelných křivek extra-galaktických objektů, či spektrální klasifikace hvězd z dat pořízených družicí. Není proto divu, že se metody umělé inteligence staly také součástí standardních nástrojů VO (virtuální observatoře).

U uchazeče se předpokládá dobrá znalost práce s počítačem, programovacího jazyka a programování obecně. Také znalosti matematické statistiky a zkušenosti s její aplikací jsou výhodou. Úkolem studenta bude seznámit se s problematikou neuronových sítí. Také bude nezbytné se naučit základy práce s virtuální observatoří a způsoby masového zpracování dat. Poté aplikuje získané poznatky na fotometrická data pořízená některou z družic popřípadě z VO a navrhne metodu klasifikace těchto dat s pomocí neuronových sítí. Dále porovná efektivitu a spolehlivost klasifikace s jinými používanými metodami.

¹A. L. I. C. E. The Artificial Linguistic Internet Computer Entity

1.2 Struktura práce

V této úvodní kapitole uvedu motivaci pro potřebu nasazení umělých neuronových sítí při zpracování astronomických dat a letmo nastíním jejich schopnosti. Představím jejich historický i ideový původ a směr dalšího vývoje.

V následující kapitole od základu popíši principy fungování umělých neuronových sítí, a to od popisu modelu její základní jednotky až po komplexní popis celé sítě.

Ve třetí kapitole představím softwarovou knihovnu **fann** implementující funkce umělé neuronové sítě a na jednoduchém příkladu budu demonstrovat její použití.

V následující kapitole ukáži použití umělé neuronové sítě k předpovědi vývoje chaotického systému jako jednu z jejich možných aplikací v astronomii.

Konečně v páté kapitole budu demonstrovat použití umělé neuronové sítě ke klasifikaci světelných křivek hvězd v rozsáhlém katalogu.

Na závěr podrobím předložené výsledky kritické analýze a shrnu zjištěné poznatky. Součástí práce je i příloha obsahující vstupní data a mnou vyvinuté nástroje použité k jejich zpracování, a to včetně zdrojových kódů přístupných k volnému použití při zachování podmínek stanovených všeobecnou veřejnou licencí GNU GPL¹.

Typografická poznámka Text sázený strojovým písmem představuje názvy programů, souborů a parametrů použitých v datových souborech. Znak „>“ ve výpisech výstupů programů označuje výzvu interpretu příkazů k jejich zadávání. Obsažené zdrojové kódy programů jsou pro co největší jednoduchost záměrně prosty komentářů a jindy nezbytných testů. Při jejich psaní jsem se snažil o sebedokumentující pojmenování proměnných stejně jako v případě používaných funkcí. Případné nejasnosti rychle rozptýlí konzultace s obsáhlou dokumentací. Někdy se může zdát jednoduchost v duchu Unixového hesla „KISS!“ být předřazena dobrým programátorským zvyklostem. Vězte, že to byl můj záměr.

1.3 Motivace

Pojmem *umělá neuronová síť* se rozumí výpočetní model inspirovaný fungováním lidského mozku (tedy biologickou neuronovou sítí), skládající se ze vzájemně propojených výpočetních jednotek zvaných *neurony* a řídící se matematickým modelem, jak bude popsán v následující kapitole. V této práci je umělá neuronová síť reprezentována její softwarovou implementací v podobě počítačového programu. Umělé neuronové sítě jako jedna z metod *strojového učení* jsou součástí výzkumu *umělé inteligence* a dnes již běžným nástrojem všudypřítomného *dobývání znalostí z dat* (data mining). Výzkum

¹<http://www.gnu.org/copyleft/gpl.html>

umělé inteligence kombinuje a aplikuje poznatky neurologie, informatiky a matematické analýzy.

V prvním přiblížení, které bude dále podrobněji rozšířeno, můžeme na umělou neuronovou síť nahlížet jako na program, který během opakovaného *učení* upravuje svůj vnitřní stav tak, aby jeho výstup co nejlépe odpovídal požadovanému stavu pro daná vstupní data. Takto získaný vnitřní stav pak při *vybavení* aplikuje na zkoumaná data.

Motivací pro použití umělých neuronových sítí (nejen) v astronomii je stále rostoucí objem dat, pocházejících z rozsáhlých přehlídkových pozorování a nejrůznějších experimentů, a potřeba jejich efektivního (často statistického) zpracování. Neuronové sítě umožňují analýzu dat bez nutnosti zdlouhavého programování výpočetně složitých modelů a přenášejí tak podstatnou část řešení zadaného problému z člověka na, v mnoha ohledech výkonnější, počítač.

1.4 Schopnosti umělých neuronových sítí

Převzato z [2].

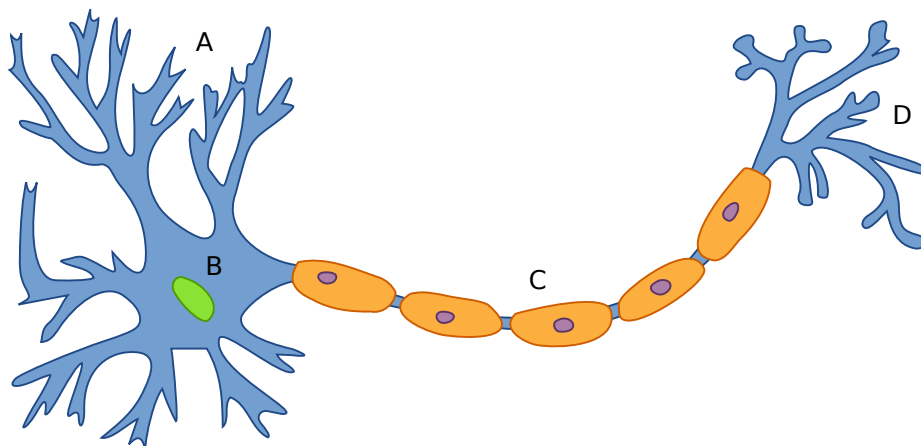
Neuronové sítě v závislosti na konkrétním typu a konfiguraci mohou řešit rozličné úkoly. Jednoduchá síť dokáže aproximovat různorodé funkce. Jeden samotný neuron je schopný klasifikovat lineárně separabilní data. *Vícevrstvá síť perceptronů (MPL)* slouží jako univerzální klasifikátor schopný klasifikovat komplexní data a časové řady do potřebného množství tříd. Sítě vycházející z modelu *funkcí s radiální bází (RBF)* poskytují dobré výsledky v oblasti aproximace a interpolace dat. *Samoorganizující mapy (SOM)* dokáží odhalit dosud neznámé vzory v předložených datech. V předpovídání na základě analýzy předchozích hodnot v časové řadě jsou úspěšné rekurentní sítě.

Hlavní předností, pro kterou jsou neuronové sítě využívány, je jejich schopnost *generalizace*. Pokud je síť správně naučena na reprezentativním vzorku dat, je schopna podávat adekvátní výsledky i pro zašumělá a nekompletní data, se kterými se během učení nesetkala.

1.5 Využití umělých neuronových sítí

Převzato z [3].

V astronomii lze nalézt hned několik oblastí aplikací umělých neuronových sítí. Analýzou světelných křivek, které reprezentují časové řady, lze odhalit trendy ve změnách jasnosti, rekonstruovat chybějící časové úseky, klasifikovat světelné křivky a předpovídat další vývoj i chaotických systémů. Kromě detekce objektů na astronomických snímcích a identifikace neznámých objektů v případě, že není k dispozici trénovací množina, je možná i jejich klasifikace na hvězdy nebo galaxie. Při spektroskopickém studiu vlastností hvězd lze



Obrázek 1.1: Schéma neuronu. A – dendrity, B – tělo neuronu s jádrem, C – axon obalený myelinovou pochvou tvořenou Schwannovými buňkami oddělenými Ranvierovými zářezy, D – axonální zakončení. *Kredit: Quasar Jarosz.*

umělou neuronovou síť použít pro hledání parametrů profilů spektrálních čar při jejich prokládání modelem a rovněž pro klasifikaci spekter. V neposlední řadě lze umělou neuronovou síť použít i na identifikaci a odstranění šumu.

1.6 Inspirace přírodou

Převzato z [1] a [2].

Předlohou pro matematický model neuronové sítě je fungování lidského mozku. Ten je považován za nejsložitější známou strukturu ve vesmíru a poznatky o jeho fungování jsou stále jen povrchní a neúplné. Mozek je tvořen z nervových buněk zvaných *neurony*. Popsané schéma typického neuronu je zobrazeno na obrázku 1.1.

Neuron zpracovává signál, který je do jeho těla přiváděn v podobě akčního potenciálu mnoha *dendrity*. Pokud tento překoná jistou prahovou hodnotu, stává se neuron aktivní a signál se z něj dále šíří *axonem*, který se na svém konci dělí a napojuje na dendrity, těla a axony dalších neuronů. Spojení neuronů je realizováno asi 20 nm širokou *synaptickou štěrbinou*, kterou se šíří neurotransmitery přenášejícími signál chemickou cestou z jednoho neuronu na druhý.

Mozek obsahuje řádově 10^{11} neuronů, z nichž každý může být spojen s několika sty až tisíci dalšími neurony a vytvářet tak celkově 10^{14} synaptických spojení. Právě *synapse* a nikoliv samotné neurony jsou dnes považovány za základní jednotku schopnou uchovávat informaci v podobě paměti a umožňující „výpočetní“ operace v mozku jako rozhodování či rozpoznávání podnětů od smyslových receptorů. Neurony jsou organizovány do strukturovaných sítí čítajících řádově 10^4 neuronů. Stovky propojených sítí tvoří asi 1 000 hlavních skupin. V závislosti na svém určení lze v mozku rozlišit neurony s rozdílnou stavbou uspořádané do sítí s rozličnou topologií.

1.7 Historie výzkumu umělých neuronových sítí

Převzato z [1].

Počátky výzkumu umělých neuronových sítí, motivované snahou vytvořit umělý mozek, lze datovat do roku 1943, kdy Warren McCulloch a jeho student Walter Pitts vytvořili jednoduchý matematický model neuronu pracující s diskrétními parametry z číselné množiny $\{-1, 0, 1\}$. V roce 1949 navrhl Donald Hebb učící pravidlo pro synapse neuronů inspirované pozorováním podmíněných reflexů. Hebbův zákon učení se později stal základem učících algoritmů umělých neuronových sítí. V roce 1957 Frank Rosenblatt zobecnil McCullochův a Pittsův model neuronu pro reálný číselný obor parametrů a vytvořil tak perceptron včetně učícího algoritmu. V roce 1959 vyvinul Bernard Widrow společně se svými studenty model neuronu nazvaný ADELIN. Jejich spojením vzniká umělá neuronová síť MADELINE.

V roce 1969 vydali Marvin Minský a Seymour Papert práci, ve které prokázali, že jediný perceptron není schopný simulovat jednoduchou logickou funkci vylučovací disjunkci (XOR). Tento svůj závěr mylně interpretovali na umělé neuronové síti jako takové, čímž celý obor na dlouhá léta zdiskreditovali a zapříčinili přesun zájmu vědců a hlavně finančních prostředků k výzkumu expertních systémů.

Až v roce 1986 popsali David Rumelhart, Geoffrey Hinton a Ronald Williams algoritmus pro učení umělé neuronové sítě na základě zpětného šíření chyby – *Backpropagation*, který umožnil vícevrstvé síti perceptronů simulovat logický operátor XOR a prolomit tak bezmála dvacet let trvající paradigma a stal se základem dnes používaných učících algoritmů. Ve stejné době představil svůj model rekurentní sítě John Hopfield a Teuvo Kohonen, který popsal samoorganizující mapy (SOM).

Za zmínku rovněž stojí, že z laboratoří umělé inteligence Massachusettského technologického institutu (MIT) pochází i projekt GNU založený Richardem Stallmanem v roce 1983.

1.8 Budoucnost využití umělých neuronových sítí

V dnešní době se nachází pro umělé neuronové sítě široké rozmezí uplatnění. Jejich rozšíření napomáhá i současný trend ve vývoji výpočetní techniky, který vede k nasazování vícejadrových a vláknových procesorů. Ty umožňují paralelní zpracování výpočtů, které jsou umělým neuronovým sítím z jejich podstaty vlastní. Umělé neuronové sítě je tak možné počítat i na výkonných grafických kartách, které představují oproti klasickým distribuovaným paralelním systémům typu grid levné a dostupné paralelní výpočetní prostředí. Tím umožňují zrychlit výpočty i složitých a rozsáhlých sítí.

Do popředí zájmu výzkumníků se znova vrací i hardwarové implementace umělých neuronových sítí, například v podobě kognitivního čipu SyNAPSE vyvíjeného firmou IBM.

Kapitola 2

Umělá neuronová síť

You said: Are you real?

A.L.I.C.E.: Yes I am a real robot.

2.1 Matematický model neuronu

Převzato z [2].

Formální neuron je základní samostatnou výpočetní jednotkou umělé neuronové sítě. Schématicky je jeho reprezentace znázorněna na obrázku 2.1. Matematicky lze činnost neuronu popsat následujícím vztahem

$$y = f(u - \theta), \quad (2.1)$$

kde y je *výstup* neuronu daný funkční hodnotou *aktivační funkce* f pro *aktivační potenciál* u počítaný jako vážený součet N obecně reálných *vstupů* x_i a jim odpovídajících *vah spojení* w_i

$$u = \sum_{i=1}^N w_i x_i \quad (2.2)$$

zmenšený o *prahovou hodnotu* (bias) θ .

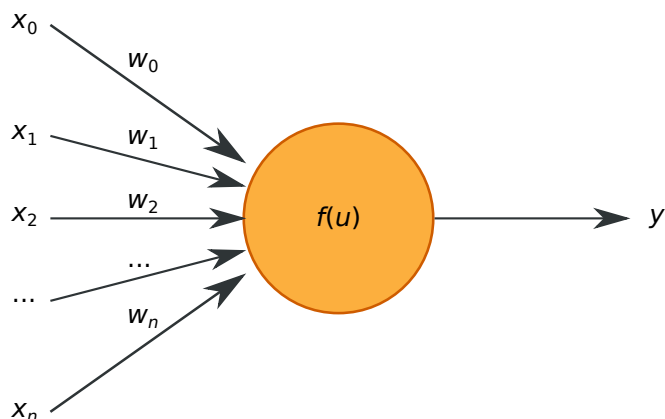
Formálně lze práh θ nahradit vahou $w_0 = -\theta$ pro jednotkový vstup $x_0 = 1$ a vztah (2.1) přepsat do tvaru

$$y = f\left(\sum_{i=0}^N w_i x_i\right), \quad (2.3)$$

případně ve vektorové podobě

$$y = f(\mathbf{w}\mathbf{x}), \quad (2.4)$$

kde $\mathbf{w} = [w_0, w_1, \dots, w_n]$ představuje řádkový vektor vah spojení a $\mathbf{x} = [x_0, x_1, \dots, x_n]^T$ sloupcový vektor vstupů neuronu.



Obrázek 2.1: Schéma formálního neuronu. Výstup neuronu y je funkční hodnotou aktivační funkce f pro aktivační potenciál u počítaný ze vstupů x a jim odpovídajících vah w .

2.1.1 Aktivační funkce

Aktivační funkce neuronu převádí aktivační potenciál do oboru hodnot neuronu. Ten zpravidla bývá omezen na interval $\langle 0; 1 \rangle$, respektive $\langle -1; 1 \rangle$, kde hodnota 1 odpovídá aktivnímu neuronu zesilujícímu signál a hodnota 0, respektive -1 , neaktivnímu neuronu signál dále nepřenášejícímu. Tvary nejčastěji používaných aktivačních funkcí zobrazuje graf 2.1.

Nejběžněji používanou aktivační funkcí je *logistická funkce* neboli *sigmoida* s funkčním předpisem

$$\sigma(x) = \frac{1}{1 + e^{-tx}}. \quad (2.5)$$

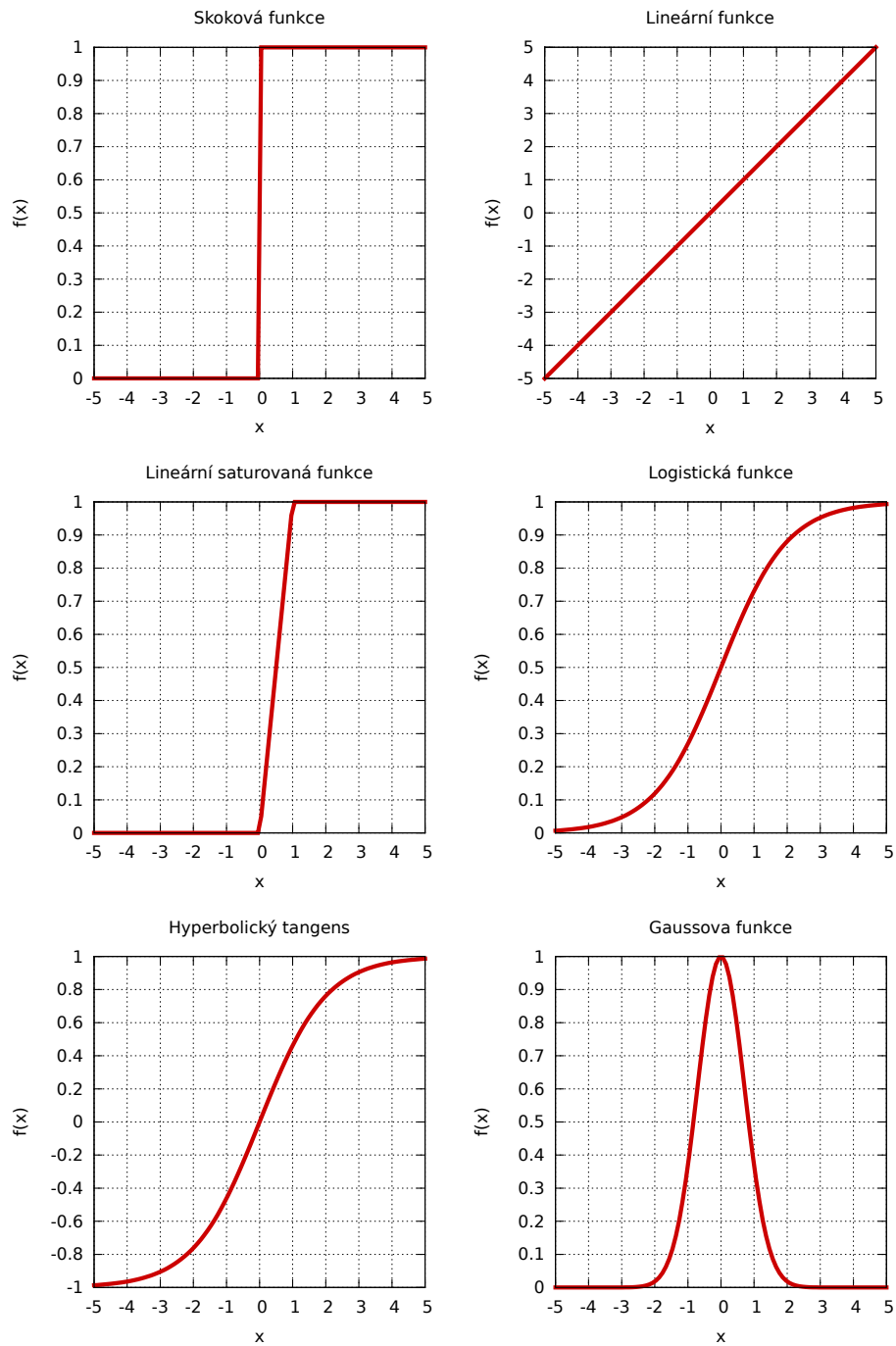
Její tvar v závislosti na hodnotě parametru t označovaného jako *strmost* ukazuje graf 2.2. Její lichou obdobou s rozšířeným oborem hodnot do záporných čísel je funkce *hyperbolický tangens*

$$\tanh(x) = \frac{1 - e^{-tx}}{1 + e^{-tx}}. \quad (2.6)$$

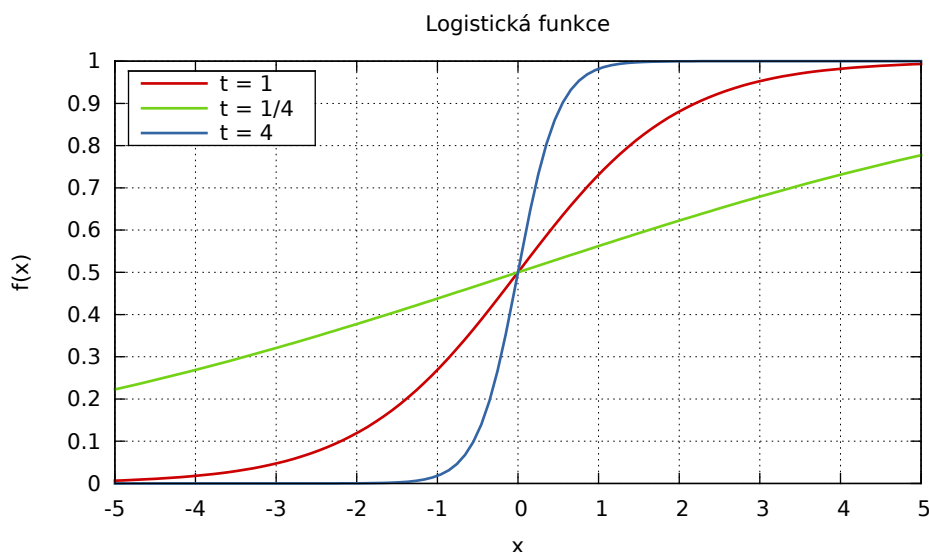
Mezi nejjednodušší aktivační funkce patří *skoková funkce* používaná při bipolárním rozhodování

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}. \quad (2.7)$$

Klasifikace světelných křivek s pomocí neuronových sítí



Graf 2.1: Tvary nejčastěji používaných aktivačních funkcí.



Graf 2.2: Průběh logistické funkce v závislosti na strmosti t .

Uplatnění má i prostá *lineární funkce*

$$f(x) = tx. \quad (2.8)$$

Funkce podobná skokové funkci (2.7) je *lineární satureovaná funkce*

$$f(x) = \begin{cases} 0, & x < 0 \\ tx, & 0 \leq x < 1 \\ 1, & x \geq 1 \end{cases} . \quad (2.9)$$

Používanou sudou funkcí je *Gaussova funkce*

$$f(x) = e^{-tx}. \quad (2.10)$$

Do výčtu používaných aktivačních funkcí lze zařadit i goniometrické funkce a liché varianty výše popsaných funkcí.

2.1.2 Hebbův zákon učení

Váhy spojení w neuronů představují volné parametry, které je možné měnit. Hebbův zákon učení vychází z představy: jestliže vstup neuronu x s váhou w způsobí aktivaci neuronu reprezentovanou jeho výstupem y , je jeho váha w zvětšena, v případě neaktivního neuronu zmenšena. Míra změny váhy Δw je dána jako

$$\Delta w = \beta xy, \quad (2.11)$$

kde konstantní parametr β nabývající hodnoty z intervalu $\langle 0; 1 \rangle$ představuje rychlost učení ovlivňující rychlost změny váhy. Novou hodnotu váhy spojení w' pak dostaneme jako

$$w' = w + \Delta w. \quad (2.12)$$

2.1.3 Učící algoritmus neuronu

Učení samotného neuronu, stejně jako celé sítě, je proces, při kterém dochází k adaptaci vah spojení neuronů. V případě učení s učitelem¹ máme k dispozici trénovací množinu obsahující soubor vektorů vstupů \mathbf{x} a jim odpovídající výstup z . Po zpracování každého vzoru z trénovací množiny se z očekávaného výstupu z a skutečného výstupu neuronu y spočte chyba

$$E = z - y \quad (2.13)$$

a pomocí modifikovaného Hebbova zákona učení (2.12) se vypočte nový vektor vah spojení

$$\mathbf{w}' = \mathbf{w} + \beta \mathbf{x} E. \quad (2.14)$$

2.2 Vícevrstvá síť perceptronů

Převzato z [2].

Jeden samotný neuron dokáže řešit jen omezené množství jednoduchých úloh jako je klasifikace lineárně separabilních dat. Pro řešení složitějších problémů lze použít síť neuronů. Nejběžněji používanou sítí jako univerzální klasifikátor je vícevrstvá síť perceptronů učená s učitelem pomocí algoritmu zpětného šíření chyby – *Backpropagation*.

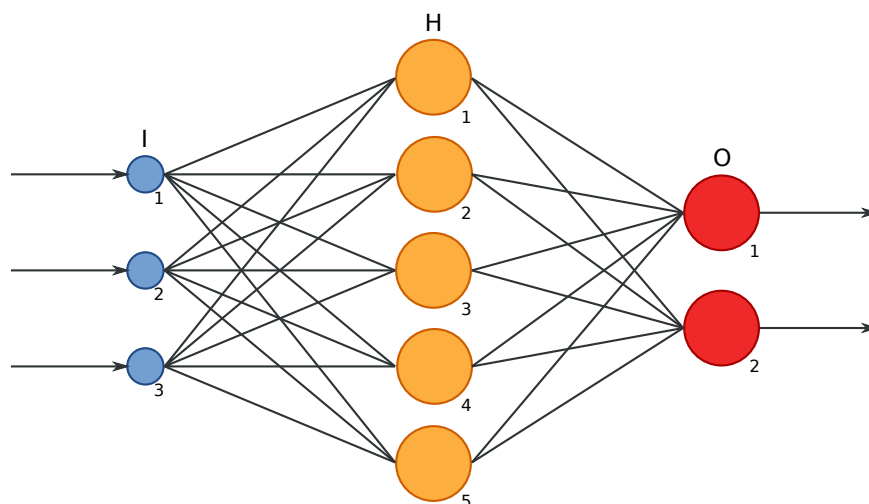
2.2.1 Topologie umělé neuronové sítě

Vícevrstvá síť perceptronů je organizována do vrstev, jak je schematicky znázorněno na obrázku 2.2. Neuronů ve vstupní vrstvě se samy nepodílejí na výpočtu, nemají měnitelné váhy, ale jen zajišťují přenos dat do sítě. Jejich počet odpovídá počtu vstupních parametrů.

Neuronů ve výstupní vrstvě představují samotný výstup ze sítě. Jejich počet se odvíjí od požadovaného počtu skupin pro klasifikaci. Pro bipolární klasifikaci stačí jeden neuron. Pro klasifikaci do více než dvou skupin je třeba odpovídající počet výstupních neuronů, kde aktivita daného výstupního neuronu odpovídá příslušnosti k dané skupině.

Mezi vstupní a výstupní vrstvou může být obecně více skrytých vrstev. Pro většinu aplikací je dostatečná jedna skrytá vrstva. Pro počáteční odhad

¹Alternativou je učení bez učitele, kdy síť v předložených datech sama hledá vztahy. Tento model ale není v této práci použit a proto ani dále rozebírán.



Obrázek 2.2: Schéma vícevrstvé sítě perceptronu typu 3-5-2 s třemi neurony ve vstupní vrstvě (I), pěti neurony v jedné skryté vrstvě (H) a dvěma neurony ve výstupní vrstvě (O). Spojení mezi neurony odpovídají vahám spojení. Jednotkové vstupy s prahovými vahami nejsou vyznačeny.

počtu neuronů p ve skryté vrstvě pro síť s n vstupními a m výstupními neurony lze využít následující vzorce [4]

$$p = \sqrt{mn} \quad \text{nebo} \quad p \geq n + m. \quad (2.15)$$

V dopředné síti jsou neurony z jedné vrstvy spojeny jen s neurony v následující vrstvě. Kromě výstupní vrstvy jsou výstupy z neuronů rozmnoženy na všechny neurony v následující vrstvě. Vyjma neuronů ve vstupní vrstvě má každý neuron přiřazen i jednotkový vstup s prahovou vahou.

2.2.2 Učení s učitelem

Během učící fáze umělá neuronová síť opakovaně zpracovává trénovací množinu tvořenou sadou vektorů vstupních parametrů \mathbf{x} a jim odpovídajících vektorů výstupních hodnot \mathbf{z} . Po zpracování celé trénovací množiny – jedné *epoše* – se spočte *střední kvadratická chyba* sítě

$$E = \frac{1}{2NK} \sum_{k=1}^K \left[\sum_{n=1}^N (y_{nk} - z_{nk})^2 \right], \quad (2.16)$$

kde N je počet vzorů v trénovací množině a K je počet neuronů ve výstupní vrstvě. Cílem učení umělé neuronové sítě je minimalizace střední kvadratické chyby E . Toho se dosáhne úpravou vah spojení jednotlivých neuronů v síti.

Učení probíhá tak dlouho, dokud chyba nedosáhne požadované přesnosti nebo neproběhne předem stanovený počet epoch.

Vhledem ke způsobu zpracování trénovací množiny umělou neuronovou sítí se rozlišují dva způsoby učení.

Průběžné učení Střední kvadratická chyba je počítána po zpracování každého z N vzorů trénovací množiny. K úpravě vah spojení mezi neurony dochází N -krát za epochu.

Dávkové učení Střední kvadratická chyba je počítána a k následné úpravě vah spojení mezi neurony dochází až po zpracování celé trénovací množiny sítí, tedy jen jednou na konci každé epochy.

2.3 Backpropagation

Převzato z [2].

Algoritmus zpětného šíření chyby je založen na úpravě vah spojení mezi neurony, jak byl popsán v části 2.1.3, a je základem pro další pokročilejší učící algoritmy. Novou hodnotu váhy spojení pro následující epochu učení po epoše m získáme jako

$$w_{m+1} = w_m + \Delta w_m, \quad (2.17)$$

kde změna váhy spojení mezi neurony Δw_m je dána jako

$$\Delta w_m = -\epsilon d_m. \quad (2.18)$$

Parametr ϵ odpovídá rychlosti učení β ve vzorci (2.11). Celkový gradient d_m pro epochu m udává směr změny střední kvadratické chyby E vzhledem k dané váze spojení w_m

$$d_m = \sum_{n=1}^N \left[\frac{\partial E}{\partial w_m} \right]_n \quad (2.19)$$

počítané pro každý z N vzorů trénovací množiny.

Pro třívrstvou umělou neuronovou síť získáme derivaci střední kvadratické chyby E vzhledem k váze spojení b neuronu ve výstupní vrstvě jako

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial b}, \quad (2.20)$$

kde y je výstup neuronu ve skryté vrstvě a v je vážený součet vstupů tohoto neuronu podle vztahu (2.2) pro aktivační potenciál neuronu.

Derivaci střední kvadratické chyby E vzhledem k váze spojení a neuronu ve skryté vrstvě získáme jako

$$\frac{\partial E}{\partial a} = \left(\frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial h} \right) \cdot \frac{\partial h}{\partial u} \cdot \frac{\partial u}{\partial a}, \quad (2.21)$$

kde oproti předchozímu vztahu (2.20) je navíc h výstup z neuronu ve skryté vrstvě a u jeho aktivační potenciál.

Z rovnic (2.20) a (2.21) plyne podmínka pro derivovatelnost aktivační funkce neuronu použité během učení umělé neuronové sítě.

2.3.1 Moment setrvačnosti

Dávkové učení oproti průběžnému učení poskytuje stabilnější řešení a urychluje hledání minima střední kvadratické chyby. Další možností zvýšení robustnosti učícího algoritmu je zavedení *momentu setrvačnosti* μ do vztahu (2.18) pro změnu váhy spojení neuronu, který tak můžeme rozšířit do tvaru

$$\Delta w_m = \mu \Delta w_{m-1} - (1 - \mu) \epsilon d_m^w. \quad (2.22)$$

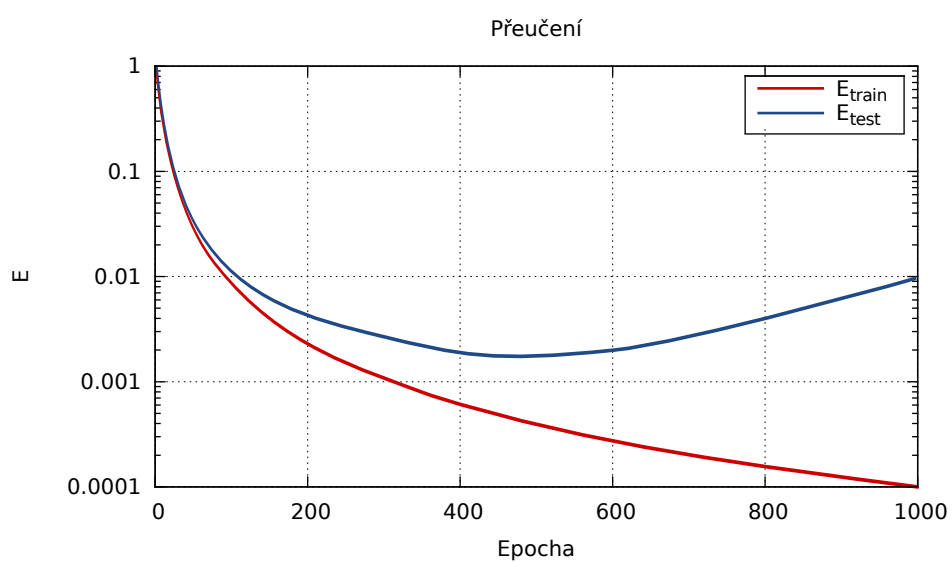
Moment setrvačnosti μ nabývá hodnot z intervalu $(0; 1)$. Vhodná volba parametrů rychlosti učení a momentu setrvačnosti ovlivňuje rychlost učení a omezuje možnost zastavení procesu učení při dosažení lokálního minima nebo naopak překročení globálního minima střední kvadratické chyby.

2.4 Přeučení

Variabilní počet neuronů ve skrytých vrstvách a od nich se odvíjející počet vah spojení jakožto volných parametrů umělé neuronové sítě ovlivňuje generalizační schopnosti sítě a může vést až k jejímu *přeučení*. To je stav, kdy síť při učení sice dosáhne minima střední kvadratické chyby a správně reaguje na data z trénovací množiny, ale pro odlišná data z testovací množiny podává chybné výsledky.

Přeučení lze předejít včasným zastavením učícího procesu, redukcí počtu neuronů ve skrytých vrstvách nebo i počtu parametrů na vstupu do sítě. Vhodnou metodikou při učení umělé neuronové sítě je z trénovací množiny vyčlenit testovací množinu, která se nebude podílet na samotném učení, ale poslouží k otestování dosaženého stavu sítě.

Graf 2.3 zobrazuje modelový vývoj střední kvadratické chyby pro trénovací E_{train} a testovací E_{test} množinu. Ukazuje, že ačkoliv střední kvadratická chyba při učení stále klesá, pro testovací data, která jsou síti neznáma, začne střední kvadratická chyba po čase opět růst a síť se přeučí.



Graf 2.3: Modelový vývoj střední kvadratické chyby během učící E_{train} a vybavovací E_{test} fáze.

Kapitola 3

FANN

You said: Can you love?

A.L.I.C.E.: As a robot I lack human emotions.

3.1 Fast Artificial Neural Network library

Pro implementaci umělé neuronové sítě jsem zvolil knihovnu `fann`¹. Její dostupnost pod licencí GNU LGPL umožňuje její bezplatné použití a rovněž zaručuje jak přístup ke zdrojovým kódům knihovny, napsané v programovacím jazyce C, tak možnost jejich modifikace i následného šíření takto upravených zdrojových kódů.

Knihovna implementuje potřebné základní funkce umělé neuronové sítě. S její pomocí je možno vytvořit plně i částečně propojenou vícevrstvou dopřednou síť. Nabízí několik nejčastěji používaných aktivačních funkcí. K dispozici je jak průběžné, tak i dávkové učení založené na algoritmu Backpropagation i na něm založených pokročilejších algoritmech iRprop a Quickprop.

Zvolený standardizovaný programovací jazyk C zaručuje přenositelnost knihovny mezi různými platformami a operačními systémy. Výkonnostní optimalizace na úrovni zdrojového kódu přispívá k rychlosti běhu a paměťové nenáročnosti výsledného programu. Srovnávací testy jejího výkonu v porovnání s jinými implementacemi je popsána jejím autorem v [5]. Knihovna se vyznačuje rovněž jednoduchostí použití, které obnáší převážně volání jejich funkcí. Programovací jazyk C byl zvolen i pro realizaci samotných demonstračních programů. Pro knihovnu existuje mnoho nadstavbových vrstev umožňujících její použití i v jiných programovacích jazycích.

¹<http://leenissen.dk/fann/>

Tabulka 3.1: Pravdivostní tabulka pro logickou operaci XOR.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

3.2 XOR test

Demonstraci funkčnosti a použití knihovny předvedu na řešení jednoduché logické operace vylučovací (exkluzivní) disjunkce (XOR). Tabulka 3.1 shrnuje pravdivostní tabulku logické operace XOR pro binární vstupy A a B .

Hodnoty v tabulce 3.1 tvoří zároveň i trénovací množinu pro umělou neuronovou síť. Níže je uvedený obsah souboru `xor.dat` s přeepsanou tabulkou 3.1 do formátu pro použití s knihovnou `fann`. Na prvním řádku je postupně uveden počet trénovacích vzorů, počet vstupů sítě a počet jejich výstupů. Na dalších řádcích se střídají vstupy sítě s jim odpovídajícími výstupy.

Výpis 3.1: Obsah souboru s trénovací množinou `xor.dat`.

```
1 4 2 1
2 0 0
3 0
4 0 1
5 1
6 1 0
7 1
8 1 1
9 0
```

Učení umělé neuronové sítě vykonává program `train-xor`. Výpis jeho zdrojového kódu zobrazuje program 3.1. Je použita třívrstvá síť typu 2–3–1 s dvěma neurony ve vstupní vrstvě, třemi neurony v jedné skryté vrstvě a jedním výstupním neuronem. Aktivační funkcí pro všechny neurony je logistická funkce (2.5). Pro učení je použit algoritmus `iRprop`, který je rozšířením algoritmu `Backpropagation`, jak je popsán v části 2.3 na straně 22, ale který oproti němu výrazně rychleji konverguje k minimu střední kvadratické chyby.

Níže uvedený ukázkový výstup [3.2](#) programu `train-xor` zobrazuje počet uběhlých epoch učení (`Epochs`), aktuální střední kvadratickou chybu (`Current error`) a počet výstupních neuronů (`Bit fail`), jejichž výstup pro každý vzor z trénovací množiny se liší od požadovaného výstupu více, než je povolená tolerance (výchozí hodnota 0,35). Stav neuronové sítě je uložen do souboru `xor.net`.

Výpis 3.2: Ukázkový výstup programu `train-xor`.

```
1 > ./train-xor
2 Max epochs      1000. Desired error: 0.0000100000.
3 Epochs         1. Current error: 0.2500626743. Bit fail 4.
4 Epochs        10. Current error: 0.2382091135. Bit fail 4.
5 Epochs        20. Current error: 0.1681263000. Bit fail 3.
6 Epochs        30. Current error: 0.0813056082. Bit fail 1.
7 Epochs        40. Current error: 0.0016979991. Bit fail 0.
8 Epochs        50. Current error: 0.0004712551. Bit fail 0.
9 Epochs        60. Current error: 0.0001502323. Bit fail 0.
10 Epochs       69. Current error: 0.0000062181. Bit fail 0.
```

Pro ověření správnosti fungování umělé neuronové sítě slouží program `run-xor` jehož zdrojový kód je vypsán v programu [3.2](#). Program `run-xor` načte dříve uložený stav sítě ze souboru `xor.net`. Aktivační funkce výstupního neuronu je změněna na skokovou funkci ([2.7](#)), aby na výstupu sítě byly jen bipolární hodnoty z množiny $\{0; 1\}$. Kvůli podmínce pro derivovatelnost aktivační funkce použité během procesu učení nemůže být skoková funkce při něm použita, ale musí být nahrazena logistickou funkcí s případně změněnou strmostí t , jak je zobrazeno v grafu [2.2](#).

Použití programu `run-xor`, který přijímá dva parametry, pro něž určuje jejich exkluzivní disjunkci, `i` s jeho výstupem zobrazuje následující výpis [3.3](#). Z něho plyne správnost vybavování umělé neuronové sítě pro předložené vstupní hodnoty.

Výpis 3.3: Výstup programu run-xor.

```
1 > ./run-xor
2 Pouziti: ./run-xor INPUT INPUT
3 > ./run-xor 0 0
4 xor(0, 0) = 0
5 > ./run-xor 0 1
6 xor(0, 1) = 1
7 > ./run-xor 1 0
8 xor(1, 0) = 1
9 > ./run-xor 1 1
10 xor(1, 1) = 0
```

Program 3.1: Zdrojový kód programu train-xor.c.

```
1  #include "fann.h"
2  #include <stdlib.h>
3
4  int main(void)
5  {
6      int num_layers = 3;
7      int num_input = 2;
8      int num_hidden = 3;
9      int num_output = 1;
10     int max_epochs = 1000;
11     int epochs_between_reports = 10;
12     float desired_error = 0.00001;
13
14     struct fann *ann = fann_create_standard(num_layers,
15         num_input, num_hidden, num_output);
16
17     fann_set_training_algorithm(ann, FANN_TRAIN_RPROP);
18     fann_set_activation_function_hidden(ann, FANN_SIGMOID);
19     fann_set_activation_function_output(ann, FANN_SIGMOID);
20     fann_train_on_file(ann, "xor.dat", max_epochs,
21         epochs_between_reports, desired_error);
22     fann_save(ann, "xor.net");
23     fann_destroy(ann);
24
25     return EXIT_SUCCESS;
26 }
```

Program 3.2: Zdrojový kód programu run-xor.c.

```
1  #include "fann.h"
2  #include <stdlib.h>
3  #include <stdio.h>
4
5  int main(int argc, char *argv[])
6  {
7      if (argc != 3) {
8          fprintf(stderr, "Pouziti: %s INPUT INPUT\n", argv[0]);
9          exit(EXIT_FAILURE);
10     }
11
12     fann_type input[2];
13     fann_type *output;
14
15     struct fann *ann = fann_create_from_file("xor.net");
16
17     input[0] = (fann_type) atoi(argv[1]);
18     input[1] = (fann_type) atoi(argv[2]);
19
20     fann_set_activation_function_output(ann, FANN_THRESHOLD);
21     output = fann_run(ann, input);
22     printf("xor(%d, %d) = %d\n",
23         (int) input[0], (int) input[1], (int) output[0]);
24     fann_destroy(ann);
25
26     return EXIT_SUCCESS;
27 }
```

3.3 Normalizace dat

Jak je vidět z grafů průběhů aktivačních funkcí 2.1, dochází k přenosu signálu pro nejčastěji používané funkce – logistickou funkci, funkci hyperbolický tangens a Gaussovou funkci v blízkém okolí středové hodnoty 0. Pro vzdálenější hodnoty se jejich funkční hodnota asymptoticky blíží jejich limitě v nekonečnu. Z toho důvodu je vhodné vstupní data normalizovat.

Jednou z metod je převod každého vstupního parametru x_i na normalizovaný tvar \bar{x}_i z intervalu $\langle 0; 1 \rangle$ pomocí vzorce

$$\bar{x}_i^n = \frac{x_i^n - x_i^{\min}}{x_i^{\max} - x_i^{\min}}, \quad (3.1)$$

kde x_i^{\min} , respektive x_i^{\max} , odpovídá minimální, respektive maximální, hodnotě parametru x_i .

Alternativou je standardizace parametru x_i vzhledem k jeho střední hodnotě \hat{x}_i a střední kvadratické odchylce σ_i danými vztahy

$$\hat{x}_i = \frac{1}{N} \sum_{n=1}^N x_i^n, \quad \sigma_i^2 = \frac{1}{N-1} \sum_{n=1}^N (x_i^n - \hat{x}_i)^2 \quad (3.2)$$

pro celou vstupní množinu čítající N členů podle vzorce

$$\bar{x}_i^n = \frac{x_i^n - \hat{x}_i}{\sigma_i}. \quad (3.3)$$

Knihovna **fann** sama obsahuje funkce pro normalizaci vstupních parametrů. V současné verzi 2.1.0 β knihovny z roku 2007 však tyto funkce obsahují chybu a nepracují správně. K nápravě by mělo dojít po vydání v současnosti vyvíjené verze 2.2.0, která by měla přinést i mnoho nových a užitečných funkcí. Z toho důvodu provádím normalizaci dat externím programem.

Kapitola 4

Predikce časových řad

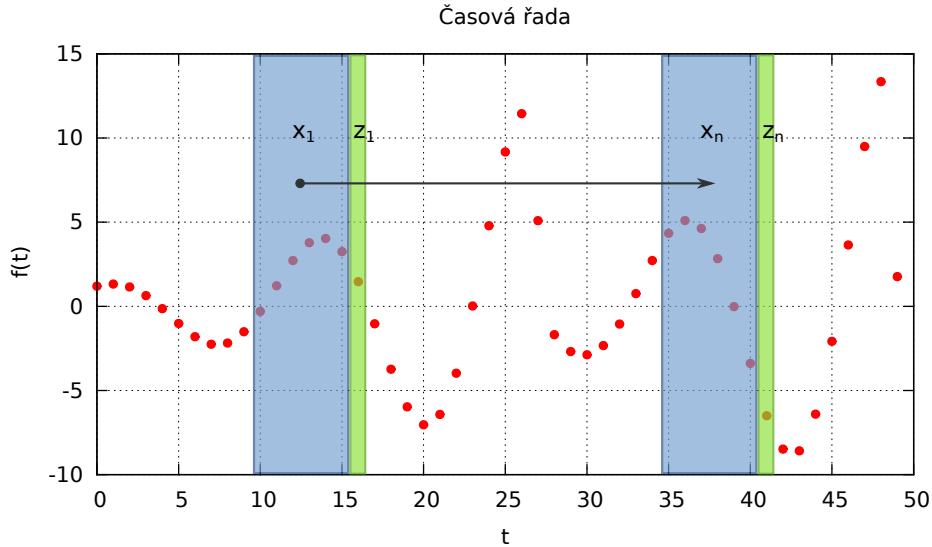
You said: How old are you?

A.L.I.C.E.: I was activated in 1995.

4.1 Chaos

Jeden ze základních úkolů fyziky vůbec je předpovídat budoucnost chování fyzikálního systému. V případě periodického procesu lze tento úkol splnit bez výraznějšího problému a to i v případě, že máme k dispozici pouze pozorování daného systému a neznáme stavové rovnice popisující systém. Jsou však systémy vykazující mnohem komplikovanější, nepravidelné chaotické chování, u nichž předpovídání budoucího chování je problematické i pokud jsme schopni tyto systémy matematicky popsat. Jedná se o celou třídu nelineárních procesů vykazujících chaotické chování, známé také jako systémy s deterministickým chaosem. Takovéto děje jsou popsány a identifikovány i v astrofyzice a jedním z jejich kandidátů jsou hvězdné pulsace.

Mnoho hvězd nacházejících se na HR diagramu v pásu nestability vykazuje nepravidelné chování, jejich světelné křivky jsou velmi komplikované a nelze u nich určit správně periodu. Teoretické studie [6] ukázaly, že dynamika těchto pulsací je podobná Rösslerově atraktoru, známému jednoduchému systému vykazujícímu deterministický chaos. Nepravidelnost světelné křivky je pak logickým důsledkem chaotičnosti procesu. Předpovídat chování průběhu světelné křivky takovéto hvězdy je výzvou, ne však neřešitelnou úlohou. Předpovědět chování zcela jistě lze, nicméně pouze na určitou dobu. A ukazuje se, že nejlépe si s tímto úkolem poradí právě neuronová síť.



Obrázek 4.1: Tvorba trénovací množiny tvořené vstupními vektory \mathbf{x} a výstupem z posunem vstupního okna po časové řadě.

4.2 Časová řada

Časová řada je série po sobě v čase následujících měření. Následující hodnota je závislá na předchozích hodnotách

$$x_{t+1} = F(x_t, x_{t-1}, \dots, x_{t-n}) \quad (4.1)$$

Pro dynamické a chaotické systémy může být nalezení závislosti F obtížné, což komplikuje předpovídání dalšího vývoje systému. Dále ukážu, že i v této oblasti, stejně jako pro doplňování chybějících dat do časových řad, lze s výhodou použít umělou neuronovou síť jako alternativu k jiným metodám [7].

Princip učení umělé neuronové sítě na časové řadě spočívá v předkládání jí jako vstupu vektor hodnot $\mathbf{x} = [x_t, x_{t-1}, \dots, x_{t-n}]$ a jako požadovaný výstup následující hodnotu $z = x_{t+1}$. Počet vstupních parametrů n udává šířku okna, které postupně prochází celou známou časovou řadou a tvoří tak trénovací množinu, jak je schématicky znázorněno na obrázku 4.1.

4.3 Rösslerův atraktor

Predikci časové řady budu demonstrovat na příkladu složky x Rösslerova atraktoru [8], jež je popsán trojicí diferenciálních rovnic (4.2)

$$\frac{dx}{dt} = -y - z, \quad \frac{dy}{dt} = x + a y, \quad \frac{dz}{dt} = b + z(x - c) \quad (4.2)$$

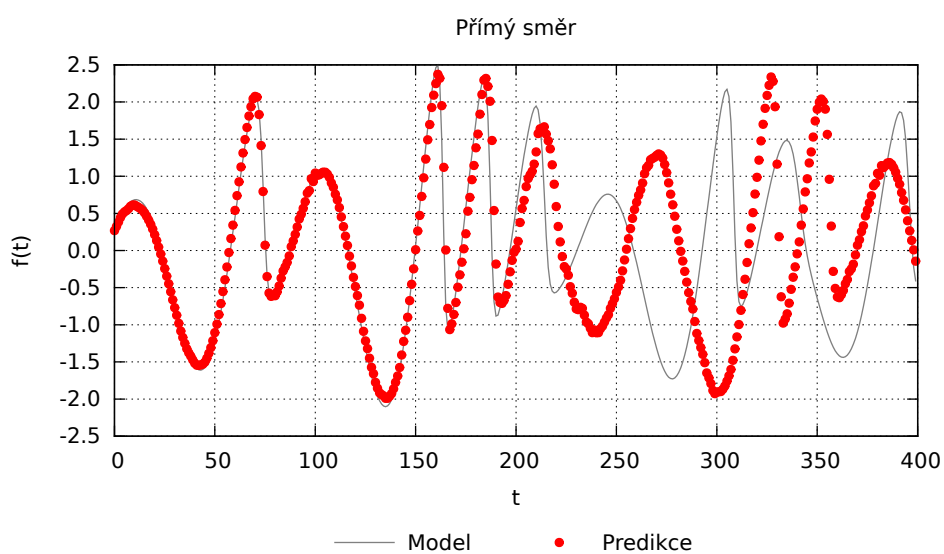
Trénovací množinu jsem vytvořil výpočtem 110 000 hodnot podle uvedených vztahů (4.2) se zvolenými parametry $a = 0,35$, $b = 0,2$, $c = 5,7$, časovým krokem $dt = 0,01$ a počátečními podmínkami $x_0 = 1$, $y_0 = -1$ a $z_0 = 1$. Následně jsem pro složku x vybral každou desátou hodnotu, z nich prvních 1000 zahodil a zbylou množinu čítající 10 000 hodnot standardizoval podle vztahu (3.3). Z prvních 4 800 prvků této množiny jsem pak připravil trénovací množinu pro přímý směr a po ní následujících 400 prvků nechal určit umělou neuronovou síť. Stejně jsem postupoval i v případě trénovací množiny pro zpětný směr, ale s časovou řadou v obráceném pořadí. Umělá neuronová síť tedy dvakrát předpovídala stejný časový úsek, ale učila se ze dvou protějších směrů na rozdílných datech.

Vstupní okno jsem zvolil o šířce 25, z čehož vyplývá i topologie použité sítě 25–50–1 inspirované modelem RBF s 25 vstupními neurony, 50 neurony v jedné skryté vrstvě s Gaussovou aktivační funkcí (2.10) a jedním výstupním neuronem s lineární aktivační funkcí (2.8). Učící algoritmus jsem zvolil iRprop.

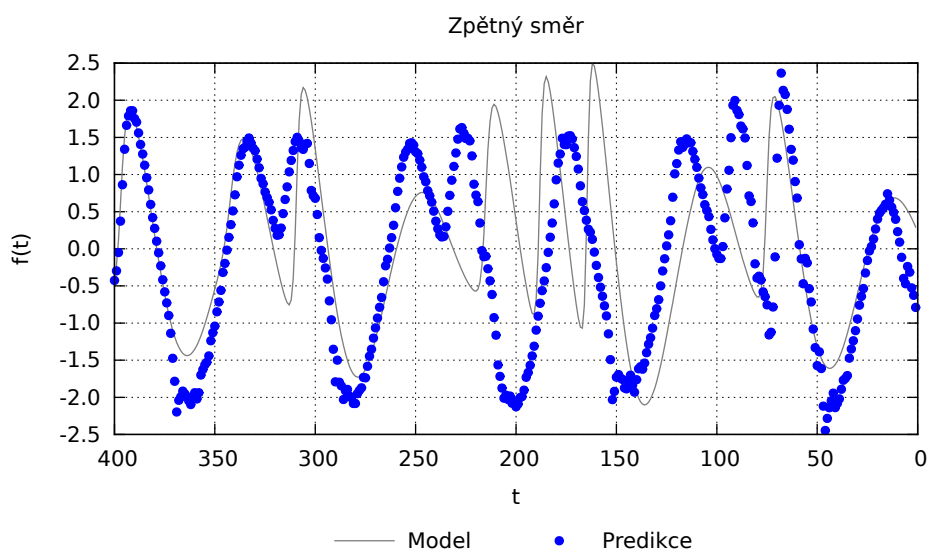
Zdrojový kód učícího programu `train-pred` je zobrazen v programu 4.1. Program přijímá dva argumenty, kde první je název souboru, kam se uloží získaný stav sítě, a druhý je název souboru s trénovací množinou. Výstup programu je obsáhlejší, ale jinak obdobný jako u výpisu 3.2, proto jej zde ani neuvádím.

Program `run-pred`, jehož zdrojový kód je uvedený v programu 4.2, přijímá rovněž dva argumenty, z nichž první je opět název souboru s uloženým stavem sítě a druhý je název souboru s testovací množinou, ze které získává inicializační vektor vstupu. Na výstupu pak poskytuje jak vypočítaný výstup, tak i modelový výstup a jejich rozdíl.

Výsledek predikce v obou směrech i s modelovým průběhem zobrazují grafy 4.1 a graf 4.2. Z nich je patrné, že pro přímý směr je úspěšnost predikce umělou neuronovou sítí větší než pro zpětný směr. Příčina tohoto jevu mi však není v tuto chvíli známá, ale bude předmětem dalšího zkoumání.



Graf 4.1: Predikce časové řady umělou neuronovou sítí v přímém směru.



Graf 4.2: Predikce časové řady umělou neuronovou sítí v zpětném směru.

Program 4.1: Zdrojový kód programu train-pred.c.

```
1  #include "fann.h"
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      if (argc != 3) {
7          fprintf(stderr, "Pouziti: %s NETWORK TRAIN_DATA\n", argv[0]);
8          exit(EXIT_FAILURE);
9      }
10
11     int num_layers = 3;
12     int num_input = 25;
13     int num_hidden = 50;
14     int num_output = 1;
15     int max_epochs = 10000;
16     int epochs_between_reports = 10;
17     float desired_error = 5e-4;
18
19     struct fann *ann = fann_create_standard(num_layers,
20         num_input, num_hidden, num_output);
21
22     fann_set_training_algorithm(ann, FANN_TRAIN_RPROP);
23     fann_set_activation_function_hidden(ann, FANN_GAUSSIAN);
24     fann_set_activation_function_output(ann, FANN_LINEAR);
25     fann_train_on_file(ann, argv[2], max_epochs,
26         epochs_between_reports, desired_error);
27     fann_save(ann, argv[1]);
28     fann_destroy(ann);
29
30     return EXIT_SUCCESS;
31 }
```

Program 4.2: Zdrojový kód programu run-pred.c.

```
1  #include "fann.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int main(int argc, char *argv[])
6  {
7      if (argc != 3) {
8          fprintf(stderr, "Pouziti: %s NETWORK TEST_DATA\n", argv[0]);
9          exit(EXIT_FAILURE);
10     }
11
12     struct fann *ann = fann_create_from_file(argv[1]);
13     struct fann_train_data *data = fann_read_train_from_file(argv[2]);
14     fann_type input[data->num_input];
15     fann_type *output;
16
17     for (int i = 0; i < (int) data->num_input; i++) {
18         input[i] = data->input[0][i];
19     }
20
21     puts("# N model network diff");
22
23     for (int i = 0; i < (int) data->num_data; i++) {
24         output = fann_run(ann, input);
25         printf("%i %e %e %e\n", i, data->output[i][0],
26             output[0], output[0]-data->output[i][0]);
27         for (int j = 0; j < (int) data->num_input-1; j++) {
28             input[j] = input[j+1];
29         }
30         input[data->num_input-1] = output[0];
31     }
32
33     fann_destroy_train(data);
34     fann_destroy(ann);
35
36     return EXIT_SUCCESS;
37 }
```

Kapitola 5

Klasifikace světelných křivek

You said: What is the meaning of life?

A.L.I.C.E.: The meaning of life is part of God's mysterious plans.

5.1 Be hvězdy

Klasifikaci světelných křivek s pomocí umělé neuronové sítě budu demonstrovat na příkladu identifikace Be hvězd v obsáhlých hvězdných katalozích.

Dle definice jsou Be hvězdy rychle rotující hvězdy spektrálního typu B a luminozitních tříd III–V, v jejichž spektru je, nebo v minulosti byla, pozorována alespoň jedna emisní čára Balmerovy série. Přesný mechanismus vzniku disku u Be hvězd není doposud znám. Existuje několik alternativních hypotéz, od binární teorie až po komplikovaný mechanismus spojující vliv magnetického pole, rychlé rotace, pulzací a hvězdného větru. Be hvězdy vykazují jak fotometrickou, tak spektrometrickou proměnnost na všech časových škálách.

Zaměření právě na Be hvězdy plyne z řešení pracovního balíku CU7 projektu Gaia Evropské kosmické agentury (ESA) realizovaného vědeckým týmem pracovníků stelárního oddělení Astronomického ústavu Akademie věd v Ondřejově. Jeden z úkolů v rámci tohoto projektu je na základě světelné křivky pozorované hvězdy, respektive jejích parametrů, rozhodnout, s určitou pravděpodobností, o její příslušnosti k Be hvězdě. Tento úkol je poměrně složitý, nicméně je však velmi vhodný právě pro použití umělých neuronových sítí.

5.2 Katalogy hvězd

Jako zdroj dat pro klasifikaci byly použity dva obsáhlé fotometrické katalogy. Použitá data byla nejprve vyčištěna od hrubých chyb, tedy světelné křivky byly zbaveny odlehlých bodů, na základě Grubbova testu [9], který odstraní

odlehle hodnoty, pokud jeho hodnota G pro krajní body x_{\min} a x_{\max}

$$G = \frac{\hat{x} - x_{\min}}{\sigma} \quad a \quad G = \frac{x_{\max} - \hat{x}}{\sigma} \quad (5.1)$$

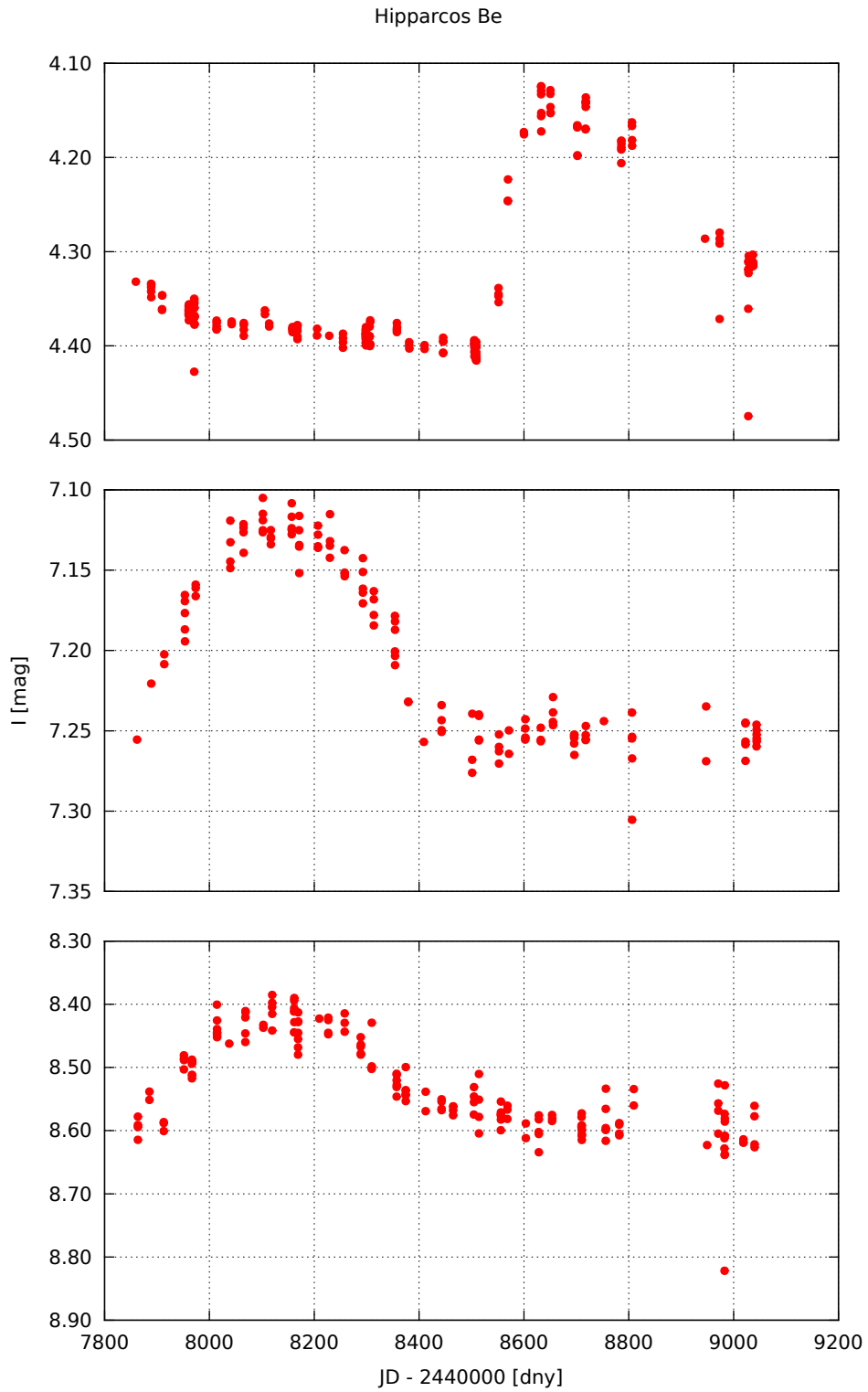
přesahuje hodnotu danou kritickým parametrem Studentova rozdělení pro požadovanou hladinu spolehlivosti. Poté byly spočteny níže popsané statistické a geometrické parametry.

Hipparcos Z hvězdného katalogu pořázeného družicí Hipparcos [10] byl sestaven katalog *Hipparcos OBA* čítající 29 335 hvězd spektrálních typů O, B a A. Výběrem hvězd na základě dostupné literatury byl sestaven katalog *Hipparcos Be* obsahující prokazatelně 603 Be hvězd. Je však třeba poznamenat, že již dle definice Be hvězd je zřejmé, že se v tomto vzorku mohou vyskytovat také Be hvězdy v poklidné fázi, tedy bez přítomnosti případné emise ve spektru. Graf 5.1 zobrazuje tři vybrané světelné křivky z katalogu Hipparcos Be, včetně ponechaných odlehklých bodů.

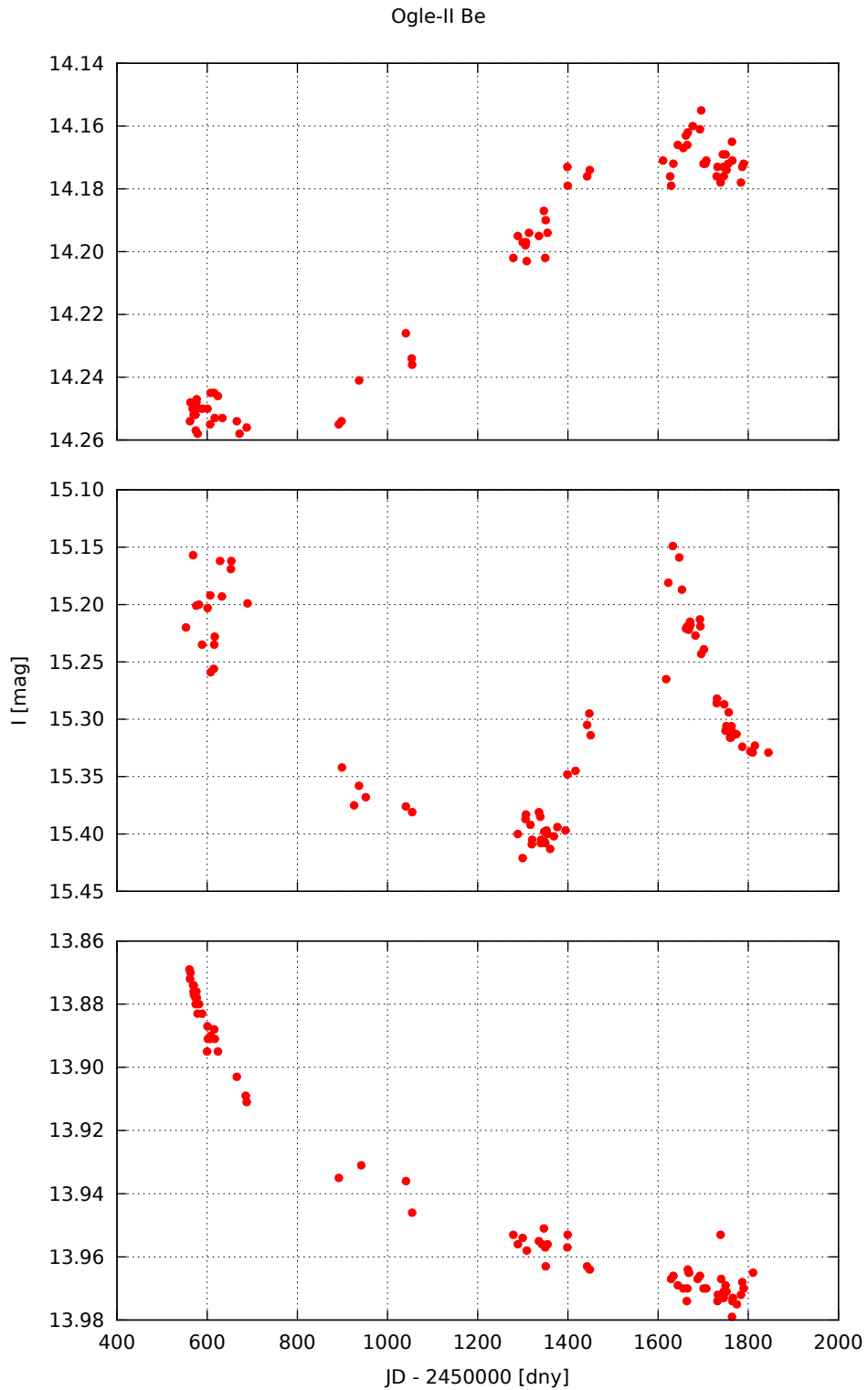
Ogle-II V rámci projektu Ogle-II [11] byl sestaven katalog fotometrických měření přibližně 30 milionů hvězd ve 49 hvězdných polích ve směru Galaktické výdutě. V článku [12] popisuje Sabogal, jak z tohoto katalogu na základě jasnosti hvězd ve fotometrickém filtru I a jejich barevném indexu $V - I$ sestavil seznam *prekandidátů* na Be hvězdy. Z něj na základě robustní střední hodnoty jasnosti hvězdy \widehat{I}_R a její robustní střední kvadratické odchylce σ_R a amplitudě změn jasnosti A vybral seznam *kandidátů* na Be hvězdy. Z jeho malého vzorku vizuální prohlídkou světelných křivek vybral zástupce typických Be hvězd, u kterých určil hodnoty statistických parametrů jako klasická a robustní střední hodnota jasnosti \widehat{I} a \widehat{I}_R a klasická a robustní střední kvadratická odchylka σ a σ_R a z nich odvozené další statistické parametry skewness S a S_R a kurtosis K a K_R . Na základě těchto parametrů, respektive jejich kombinací, určil rozsah parametrů zaznamenaný v tabulce 5.1, které potvrdily 70 % dříve vizuální prohlídkou určených Be hvězd. Tuto statistiku pak aplikoval na celý seznam kandidátů, výsledkem čehož získal katalog *nejpravděpodobnějších* Be hvězd. Graf 5.2 zobrazuje tři vybrané světelné křivky z tohoto katalogu.

Tabulka 5.1: Definice a hodnoty rozsahu statistických parametrů určených Sabogalem.

Parametr	Definice	Rozsah hodnot
p_1	$\widehat{I}_R/\widehat{I}$	[0,9970; 1,0011]
p_2	σ_R/σ	[0,30; 1,22]
p_3	S_R/S	[-1,00; 7,15]
p_4	$ p_1 \times p_2 \times p_3 $	[0,29; 3,82]



Graf 5.1: Příklady světelných křivek Be hvězd z katalogu Hipparcos Be.



Graf 5.2: Příklady světelných křivek nejpravděpodobnějších Be hvězd z katalogu Ogle-II.

5.3 Metodika klasifikace

Světelné křivky hvězd jsem klasifikoval pomocí umělé neuronové sítě na základě devíti vybraných robustních statistických a geometrických parametrů popisujících dostupné světelné křivky z katalogů Hipparcos a Ogle-II. Dále podrobněji objasním jednotlivé kroky vedoucí k výsledné klasifikaci.

Zvolené statistické parametry popisující světelnou křivku nebyly zvoleny zcela náhodně, ale po pečlivém výběru, a to jak na základě zkušeností předchozích prací (zejména [12]), tak i na základě našich zkušeností a experimentů. Kriterium bylo zvoleno takové, aby se tyto parametry nejvíce odlišovaly pro vzorky Be hvězd a vybrané hvězdy z OBA katalogu. Tyto parametry využijeme v budoucnu také pro klastrovací data mininigové algoritmy.

5.3.1 Parametry světelných křivek

Na tomto místě jen heslovitě uvedu definice a označení v datových souborech použitých statistických a geometrických parametrů popisujících světelné křivky hvězd a sloužících jako vstup pro umělou neuronovou síť. Grafy zobrazující histogramy a závislosti mezi jednotlivými parametry jsou dostupné na internetových stránkách této práce¹.

Střední kvadratická odchylka `stddev` jasnosti hvězdy

$$\sigma^2(x) = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{x})^2, \quad (5.2)$$

kde \hat{x} značí střední hodnotu hvězdné velikosti hvězdy

$$\hat{x} = \frac{1}{N} \sum_{i=1}^n x_i. \quad (5.3)$$

Skewness `skewness` Koeficient šikmosti charakterizuje asymetrii v rozložení veličiny

$$S(x) = \frac{1}{N} \frac{\sum_{i=1}^N (x_i - \hat{x})^3}{\sigma(x)^3}. \quad (5.4)$$

Kurtosis `kurtosis` Koeficient špičatosti porovnává dané rozdělení veličiny s normálním rozdělením

$$K(x) = \frac{1}{N} \frac{\sum_{i=1}^N (x_i - \hat{x})^4}{\sigma(x)^4} - 3. \quad (5.5)$$

¹Viz. část Přílohy

Chyba lineární regrese `stderrPoly1` Chyba při prokládání polynomem prvního řádu

$$E_L^2(x) = \frac{\sum_{i=1}^N (at_i + b - x_i)^2}{N - 2}. \quad (5.6)$$

Chyba parabolické regrese `stderrPoly2` Chyba při prokládání polynomem druhého řádu

$$E_P^2(x) = \frac{\sum_{i=1}^N (at_i^2 + bt_i + c - x_i)^2}{N - 2}. \quad (5.7)$$

Hurstův exponent `hurst` Hurstův exponent H [13] je spjat s autokorelační funkcí. Hodnota Hurstova exponentu v rozmezí $0,5 < H < 1,0$ indikuje časovou řadu s kladnou autokorelací, kdy po vysoké hodnotě členu časové řady následuje s velkou pravděpodobností opět člen s vysokou hodnotou. Nižší hodnoty indikují oscilace v časové řadě.

$$\frac{R(\tau)}{S(\tau)} = \left(\frac{\tau}{2}\right)^H. \quad (5.8)$$

Minimální relativní amplituda `ramin` Relativní minimální a maximální amplituda je ukazatel míry změny ve světelné křivce oproti střední hodnotě. U typických Be hvězd by absolutní hodnota této veličiny neměla být větší než 1,3.

$$A_r^{\min} = \frac{x_{\min}}{\hat{x}}. \quad (5.9)$$

Maximální relativní amplituda `ramax`

$$A_r^{\max} = \frac{x_{\max}}{\hat{x}}. \quad (5.10)$$

Abbe test `abbeTest` Jako jeden z parametrů popisujících světelnou křivku byl zvolen Abbe statistický test. Tato statistická veličina je schopná identifikovat spojitě a hladce se měnící časovou řadu. Platí, že čím nižší hodnota této veličiny, tím výraznější je hladký a spojitý charakter časové řady. Jak bylo ukázáno v [14] lze tuto statistiku použít i pro nepravidelně vzorkované časové řady.

Abbe test je velmi důležitou veličinou, pomáhá nám určit ve velkém množství světelných křivek křivky s velmi hladkým průběhem a výraznou změnou svého profilu. Tyto výrazné křivky pak mohou tvořit reprezentativní, dobře definovaný vzorek pro neuronovou síť a další data miningové metody vyžadující tréninkové množiny. Z našeho katalogu Hipparcos Be tento test vybral 300 reprezentativních křivek. Graf 5.3 zobrazuje dvě světelné křivky z katalogu Hipparcos Be s maximálně rozdílnými hodnotami Abbe testu.

$$r = \frac{N}{2(N-1)} \frac{\sum_{i=1}^{N-1} (x_{i-1} - x_i)^2}{\sum_{i=1}^N (x_i - \hat{x})^2} \quad (5.11)$$

5.3.2 Trénovací množina

Trénovací množinu jsem sestavil z 500 náhodně vybraných hvězd z katalogu Hipparcos Be, který sám po vynechání hvězd s chybějícími parametry obsahuje 601 hvězd a představuje pozitivní výstup sítě. Jako negativní výstup sítě jsem doplnil rovněž náhodný výběr 1 500 z celkového počtu 29 254 hvězd z katalogu Hipparcos OBA. Celá trénovací množina, uložená v souboru `train.dat`, obsahuje tedy dohromady 2 000 záznamů.

5.3.3 Testovací množina

Pro ověření správnosti naučení sítě jsem sestavil dvě testovací množiny. V souboru `test.1.dat` je uložen náhodný výběr 100 hvězd z katalogu Hipparcos Be, pro který by síť měla podávat pozitivní výstup. Soubor `test.0.dat` obsahuje náhodný výběr 100 hvězd z katalogu Hipparcos OBA, na který by síť měla reagovat negativně.

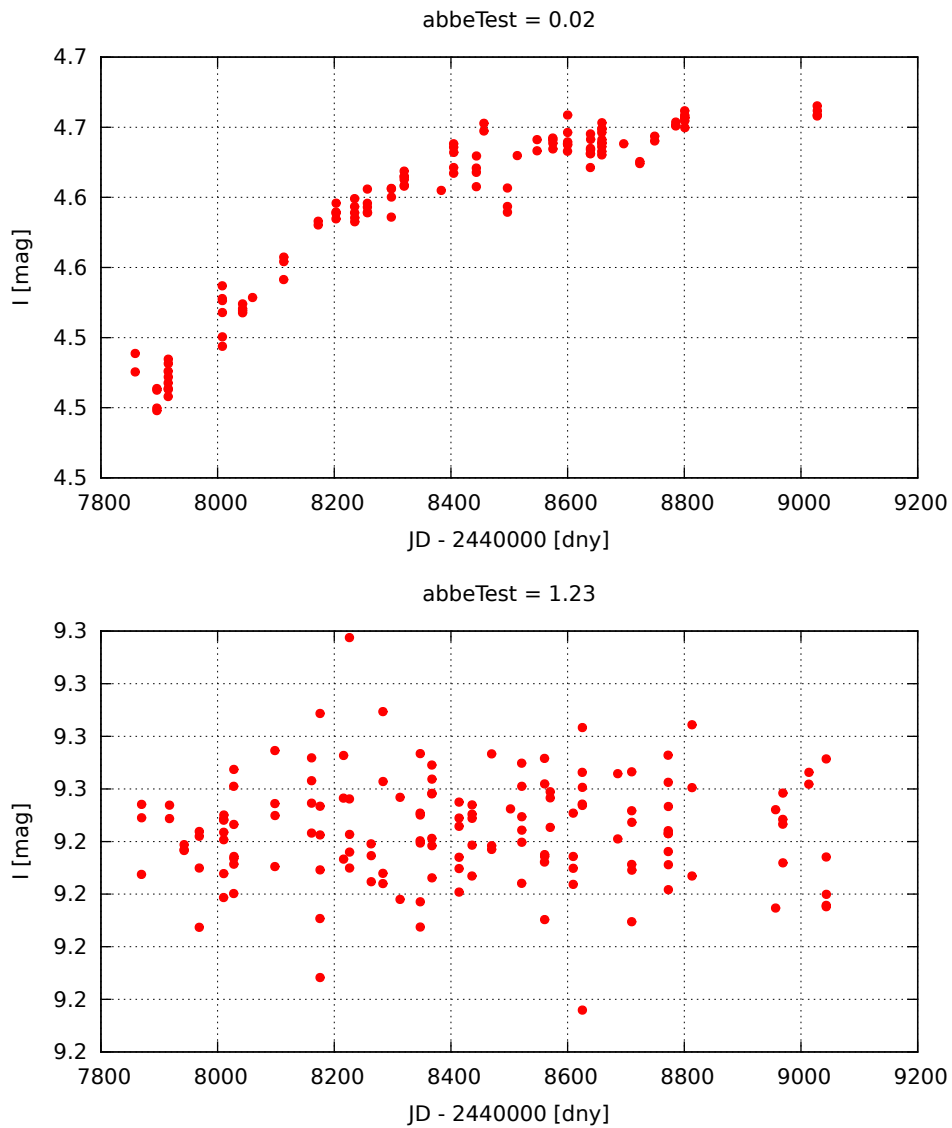
5.3.4 Topologie umělé neuronové sítě

Pro klasifikaci jsem sestavil plně propojenou dopřednou umělou neuronovou síť typu 9–30–1, tedy třívrstvou síť s 9 neurony ve vstupní vrstvě, 30 neurony s aktivační funkcí hyperbolický tangens (2.6) v jedné skryté vrstvě a jedním neuronem ve výstupní vrstvě s logistickou aktivační funkcí (2.5) a upraveným parametrem strmosti na hodnotu $t = 4$, aby svým tvarem více připomínala skokovou funkci (2.7) jak je znázorněno v grafu 2.2 na straně 19. Ve zdrojovém kódu programu `train-klas`, který je uvedený v programu 5.1 a který vykonává učení sítě pro klasifikaci, se z implementačních důvodů zadává poloviční hodnota strmosti t . Jako učící je zvolený algoritmus iRprop.

Vybavení a samotnou klasifikaci vykonává program `run-klas` se zdrojovým kódem vypsáním v programu 5.2. V něm je ve výstupním neuronu logistická funkce nahrazena skokovou funkcí (2.7), jejíž výstup s hodnotou 1 odpovídá pozitivní klasifikaci a 0 negativní klasifikaci ve smyslu *je* nebo *není* Be hvězdou.

Oba programy přijímají dva argumenty. Prvním je název souboru, kam se v případě programu `train-klas` ukládá naučený stav sítě a ze kterého ho program `run-klas` opětovně načítá. Druhým argumentem je v případě programu `train-klas` název souboru s trénovací množinou a v případě programu `run-klas` název souboru s daty pro testování sítě nebo určenými ke klasifikaci.

Samozřejmostí je standardizace trénovací, testovací i klasifikační množiny dat podle vztahu (3.3).



Graf 5.3: Dvě světelné křivky z katalogu Hipparcos Be s maximálně rozdílnými hodnotami Abbe testu.

Program 5.1: Zdrojový kód programu train-klas.c.

```
1  #include "fann.h"
2  #include <stdlib.h>
3  #include <stdio.h>
4
5  int main(int argc, char *argv[])
6  {
7      if (argc != 3) {
8          fprintf(stderr, "Pouziti: %s NETWORK TRAIN_DATA\n", argv[0]);
9          exit(EXIT_FAILURE);
10     }
11
12     int num_layers = 3;
13     int num_input = 9;
14     int num_hidden = 30;
15     int num_output = 1;
16     int max_epochs = 100000;
17     int epochs_between_reports = 100;
18     float desired_error = 1e-4;
19
20     struct fann *ann = fann_create_standard(num_layers,
21         num_input, num_hidden, num_output);
22
23     fann_set_training_algorithm(ann, FANN_TRAIN_RPROP);
24     fann_set_activation_function_hidden(ann, FANN_SIGMOID_SYMMETRIC);
25     fann_set_activation_function_output(ann, FANN_SIGMOID);
26     fann_set_activation_steepness_output(ann, 2.0);
27     fann_train_on_file(ann, argv[2], max_epochs,
28         epochs_between_reports, desired_error);
29     fann_save(ann, argv[1]);
30     fann_destroy(ann);
31
32     return EXIT_SUCCESS;
33 }
```

Program 5.2: Zdrojový kód programu run-klas.c.

```
1  #include "fann.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int main(int argc, char *argv[])
6  {
7      if (argc != 3) {
8          fprintf(stderr, "Pouziti: %s NETWORK DATA\n", argv[0]);
9          exit(EXIT_FAILURE);
10     }
11
12     fann_type *output;
13     int jednicka = 0;
14     int nula = 0;
15
16     struct fann *ann = fann_create_from_file(argv[1]);
17     struct fann_train_data *data = fann_read_train_from_file(argv[2]);
18
19     fann_set_activation_function_output(ann, FANN_THRESHOLD);
20
21     for (int i = 0; i < (int) data->num_data; i++) {
22         output = fann_run(ann, data->input[i]);
23         (int) output[0] ? jednicka++ : nula++;
24     }
25
26     printf("Ze souboru %s pozitivne klasifikovano %d z celkem %d (%d%%).\n",
27         argv[2], jednicka, data->num_data,
28         (int) ((float) jednicka/data->num_data*100));
29     printf("Ze souboru %s negativne klasifikovano %d z celkem %d (%d%%).\n",
30         argv[2], nula, data->num_data,
31         (int) ((float) nula/data->num_data*100));
32
33     fann_destroy_train(data);
34     fann_destroy(ann);
35
36     return EXIT_SUCCESS;
37 }
```

5.3.5 Výsledek klasifikace

Výstup programu `train-klas` pro jeho rozsah neuvádím, ale formálně se neliší od výpisu 3.2 a je k nahlédnutí zařazen do příloh.

Výstup programu `run-klas` pro testovací množiny dat i pro samotnou klasifikaci je zobrazen ve výpisu 5.1. Program na výstupu zobrazuje, kolik hvězd z dané množiny klasifikoval pozitivně a kolik negativně spolu s procentuální hodnotou. Soubory `test.0.dat` a `test.1.dat` odpovídají testovacím souborům popsaným v sekci 5.3.3. Soubor `certain.dat` obsahuje katalog nejpravděpodobnějších Be hvězd sestavený Sogalem z katalogu Ogle-II. Soubor `candidate.dat` pak obsahuje katalog kandidátů na Be hvězdy ze stejného zdroje.

Výpis 5.1: Formátovaný výstup programu `run-klas`.

```
1 > ./run-klas
2 Pouziti: ./run-klas NETWORK DATA
3 > ./run-klas klas.net test.1.dat
4 Ze souboru test.1.dat pozitivne klasifikovano 94 z celkem 100 (94%).
5 Ze souboru test.1.dat negativne klasifikovano 6 z celkem 100 (6%).
6 > ./run-klas klas.net test.0.dat
7 Ze souboru test.0.dat pozitivne klasifikovano 3 z celkem 100 (3%).
8 Ze souboru test.0.dat negativne klasifikovano 97 z celkem 100 (97%).
9 > ./run-klas klas.net certain.dat
10 Ze souboru certain.dat pozitivne klasifikovano 1006 z celkem 1407 (71%).
11 Ze souboru certain.dat negativne klasifikovano 401 z celkem 1407 (28%).
12 > ./run-klas klas.net candidate.dat
13 Ze souboru candidate.dat pozitivne klasifikovano 9857 z celkem 27167 (36%).
14 Ze souboru candidate.dat negativne klasifikovano 17310 z celkem 27167 (63%).
```

Z výstupu programu je patrné, že síť korektně klasifikuje testovací množiny. Rovněž pozitivně klasifikuje a tím potvrzuje i významnou část nejpravděpodobnějších Be hvězd z katalogu Ogle-II a množství Be hvězd odhalí i v souboru kandidátů na Be hvězdy. Další diskuze výsledků je pak probírána v závěru práce.

Kapitola 6

Závěr

You said: What is the answer on The Ultimate Question of Life, the Universe and Everything?

A.L.I.C.E.: It depends on how you phrase the question.

Umělé neuronové sítě se ukazují být dobrou alternativou a doplněním klasických statistických metod. V aplikacích, kde přes nesmírný rozsah množiny studovaných objektů není možný přístup prvek po prvku, ale naopak vyžadují statistické zpracování, mohou být umělé neuronové sítě vhodnou volbou. Při prvotní analýze souboru dat mohou napomoci s jejich klasifikací a nalezením skrytých vzorů a poukázat tak na směr, kterým se má zaměřit další výzkum.

V kapitole 4 jsem ukázal, že umělou neuronovou síť lze použít k predikci časových řad a předpovídání vývoje chaotických systémů. Právě teorie chaosu se v poslední době ukazuje jako vhodný model popisu fyzikálních systémů, kde klasické modely selhávají. Praktickou aplikací této schopnosti může být rekonstrukce světelných křivek nedostatečně pokrytých měřeními.

Stěžejní pro tuto práci je kapitola 5. V ní jsem demonstroval použití umělé neuronové sítě pro klasifikaci světelných křivek na základě popisu statistickými a geometrickými parametry. Klíčovým krokem v tomto procesu se ukázal být výběr reprezentativní trénovací množiny a kvalita použitých dat. Jelikož Be hvězdy jsou podmnožinou B hvězd, může i použitý katalog Hipparcos OBA, který sloužil jako zdroj referenčních dat pro negativní klasifikaci, sám obsahovat hvězdy s parametry Be hvězd, které však takto nejsou klasifikovány.

Tato případná kontaminace trénovací množiny pak může vést k horšímu naučení sítě a špatným generalizačním schopnostem. Tomuto jevu jsem se snažil vyhnout náhodným výběrem malého vzorku hvězd z celého dostupného katalogu, kde je jen nízká pravděpodobnost výskytu irelevantních dat.

Z výpisu 5.1, zobrazujícího výstup klasifikačního programu `run-klas`, je patrné, že uvedený program správně klasifikoval hvězdy z testovacích množin, což ukazuje jeho správné naučení, ale zároveň malou část hvězd klasifikoval chybně. To může být ukazatelem kontaminace trénovací množiny chybními

daty, ale zároveň je třeba podotknout, že od umělé neuronové sítě nelze očekávat stoprocentní neomylnost, zvláště pokud neexistuje prokazatelný parametr jasně definující skupinu Be hvězd.

Robustnost a věrohodnost použité klasifikační metody by šla zvýšit opakovaným učením umělé neuronové sítě a vytvořením statistiky na základě více výsledků klasifikace.

Jednou z vlastností umělých neuronových sítí, kterou je třeba mít stále na paměti, je jistá míra náhodnosti, která plyne z principu jejich učení. Na jeho začátku se váhy spojení neuronů nastavují náhodně a každé proběhnuté učení je tak jedinečný proces s nejistým výsledkem. Správnost naučení sítě, a zvláště pak její nežádoucí přeučení, je třeba testovat na testovací množině. V nejjednodušším případě vizuální prohlídkou výsledků nebo lépe pomocí automatizovaných testů s podporou všech dostupných nástrojů, které dnešní moderní vývojové a operační systémy nabízí.

Jistou komplikací pro nasazení umělých neuronových sítí je absence exaktních pravidel pro jejich tvorbu. Každý studovaný problém je tak třeba nejprve podrobit rozvážné analýze a vybrat vhodný typ a topologii použité neuronové sítě. V této práci jsem používal vícevrstvou síť perceptronů (MPL), která je univerzálním modelem pro mnoho aplikací. V rámci optimalizace je však možné zvolit i složitější typy umělých neuronových sítí navržené pro řešení konkrétních problémů, použít komplikovanější topologie s rekurentními, cyklickými či zkratkovými spoji a pokročilejší učící algoritmy. Případně, pokud to situace dovoluje, lze použít specializovaný model jako model podpůrných vektorů – support vector machines (SVM) pro bipolární klasifikaci, nebo modely používající funkce s radiální bází (RBF).

Klíčovým, ale zároveň nesnadným, úkolem je volba počtu neuronů ve skrytých vrstvách umělé neuronové sítě. Ty sice zvyšují generalizační schopnosti sítě, ale zároveň jejich velký počet může vést až k nežádoucímu přeučení sítě. Zároveň s rostoucím počtem neuronů roste i výpočetní náročnost umělé neuronové sítě, kterou je ale možné efektivně optimalizovat paralelizací výpočtů.

K optimální topologii umělé neuronové sítě se lze dopracovat metodou pokus–omyl doplněnou rozsáhlým testováním. Až s přibývajícím zkušeností konstruktéra sítě ubývá zcestných omylů.

Výpočetně i časově náročnou částí je právě učení sítě, při kterém dochází k minimalizaci střední kvadratické chyby sítě hledáním optimálních hodnot vah spojení neuronů. V mnoha cyklech se počítají nové hodnoty vah spojení neuronů, jejichž počet je úměrný počtu skrytých a výstupních neuronů tvořících síť.

Výpočty umělých neuronových sítí použitých v této práci byly prováděny na počítači s procesorem Intel(R) Pentium(R) M processor 1,60 GHz s 767 MB operační pamětí a operačním systémem GNU/Linux. Kompilace programů proběhla pomocí překladače GCC¹ ve verzi 4.6.2.

¹<http://gcc.gnu.org>

Klasifikace světelných křivek s pomocí neuronových sítí

Práce pro mě byla velkým přínosem a naučila mě mnoho nových technik masového zpracování dat. Ukázalo se, že pro praxi moderní astronomie je takřka nutností znalost minimálně jednoho vyššího programovacího jazyka. Znalost některého z mnoha interpretovaných programovacích jazyků je jen ohromnou výhodou, která dokáže ušetřit nemálo rutinní práce.

V mojí práci je jistě mnoho míst, na kterých se dá ještě zapracovat, nicméně jsem velice rád za zkušenosti a vědomosti, které jsem v průběhu jejího psaní získal a které budou podkladem pro připravovaný článek. Průběžně opravovaná elektronická verze je k dispozici na internetových stránkách uvedených mezi přílohami spolu s rozsáhlým doprovodným materiálem, který nebylo možné do práce zařadit. Zpracovávání dat pomocí umělých neuronových sítí je směr, kterému bych se chtěl ve svém studiu dále věnovat.

Literatura

- [1] ŠÍMA, JIŘÍ; NERUDA, ROMAN. *Teoretické otázky neuronových sítí*. Vyd. 1. Praha: Matfyzpress, 1996. 390 s. ISBN 80-85863-18-9.
- [2] SAMARASINGHE, SANDHYA. *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. 1st ed. New Yourk: Auerbach Publications, 2007. 570 p. ISBN 0-8493-3375-X.
- [3] TAGLIAFERRI, ROBERTO ET AL. *Neural networks in astronomy*. Neural Networks, April–May 2003, vol. 16, iss. 3–4, p. 297–319. ISSN: 0893-6080.
- [4] TAUFER, I.; DRÁBEK, O.; SEIDL, P. *Umělé neuronové sítě – základy teorie a aplikace (5)*. CHEMagazín, 2006, roč. 16, č. 5, s. 29–31. ISSN 1210-7409.
- [5] NISSEN, STEFFEN. *Implemetation of a Fast Artificial Neural Network Library (fann): University report*. Copenhagen (Denmark): University of Copenhagen, Department of Computer Science, 2003. 88 p. Supervisor Klaus Hansen.
- [6] BUCHLER, J. R. & REGEV, O. *Oscillations of an extended ionization region in a star*. The Astrophysical journal, 1982, vol. 263, p 312–319.
- [7] SERRE, T.; AUVERGNE, M.; GOUPIL, M. J. *A new method for filling gaps in data*. Astronomy & Astrophysics, 1992, vol. 259, p. 404–411.
- [8] RÖSSLER, O. E. *An equation for continuous chaos*. Physics letters, 1976, vol. 57A, num. 5, p. 397–398.
- [9] GRUBBS, FRANK. *Procedures for Detecting Outlying Observations in Samples*. Technometrics, 1969, vol. 11, no. 1, p. 1–21.
- [10] VAN LEEUWEN, F. ET AL. *The HIPPARCOS mission: photometric data*. Astronomy & Astrophysics, 1997, vol. 323, p. L61–L64.

- [11] UDALSKI, A.; KUBIAK, M.; SZYMANSKI, M. *Optical Gravitational Lensing Experiment. OGLE-2 – the Second Phase of the OGLE Project*. Acta Astronomica, 1997, v. 47, p. 319–344.
- [12] SABOGAL, B. E. ET AL. *A catalogue of Be stars in the direction of the Galactic Bulge*. Astronomy & Astrophysics, 2008, vol. 478, p. 659–665.
- [13] MARCHENKO, S. G.; HAGEN-THORM, A. V. *On the possible fractal nature of blazar light curves*. Astrophysics, 1997, vol. 40, no. 3, p. 222–226.
- [14] SPANIO M. ET AL. *Long period Variables in the Large Magellanic Cloud from the EROS-2 survey*. Astronomy & Astrophysics, 2011, vol. 536, id. A60.

Přílohy

Přílohou této práce je i archiv obsahující použité soubory dat a veškeré zdrojové kódy uvedených programů a použitých nástrojů pro dodatečnou úpravu dat, na kterou jsem si nevystačil s běžnými unixovými nástroji. Složky programů kromě samotných zdrojových kódů obsahují i soubor `Makefile` pro snadnou kompilaci programů pomocí nástroje `make`. Podmínkou úspěšné kompilace a následného spuštění programů je samozřejmě přítomnost knihovny `fann` řádně instalované v operačním systému.

Archiv `diplomka.tar.gz` obsahuje tyto adresáře

xor Ukázkový program demonstrující použití a funkčnost knihovny `fann`.

pred Demonstrační programy pro predikci časových řad.

klas Použité programy pro klasifikaci Be hvězd.

data Datové soubory s parametry světelných křivek použitých katalogů hvězd ve formátu souborů `arff`.

mktrain Pomocné programy pro přípravu trénovacích množin.

text Elektronická verze této práce ve formátu PDF s hypertextovými odkazy.

Dodatečnou přílohou jsou také internetové stránky obsahující oprávanou elektronickou verzi této práce spolu se seznamem změn. A kromě příloh přiložených k samotné práci i obsáhlý přehled všech světelných křivek použitých katalogů hvězd.

<http://physics.muni.cz/~janak/diplomka/>

Klasifikace světelných křivek s pomocí neuronových sítí

Zdeněk Janák

Diplomová práce

Verze 1.0 ze dne 11. ledna 2012

<http://physics.muni.cz/~janak/diplomka/>

Připraveno v programem L^AT_EX2e