VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

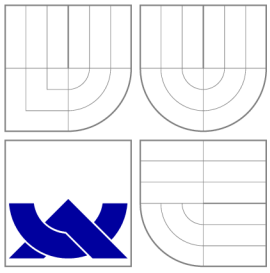
PŘEDPOVÍDÁNÍ VÝVOJE VÍCE ČASOVÝCH ŘAD PŘI BURZOVNÍM OBCHODOVÁNÍ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

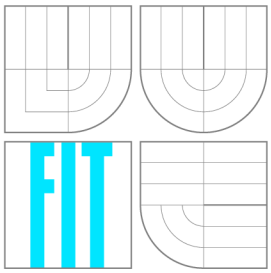
AUTOR PRÁCE
AUTHOR

Bc. PETER PALČEK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PŘEDPOVÍDÁNÍ VÝVOJE VÍCE ČASOVÝCH ŘAD PŘI BURZOVNÍM OBCHODOVÁNÍ

PREDICTION OF MULTIPLE TIME SERIES AT STOCK MARKET TRADING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETER PALČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2012

Abstrakt

V diplomové práci je uveden všeobecný postup používaný pro předpověď časových řad, jejich rozdělení, základní charakteristiky a základní statistické metody pro jejich předpovídaní. Spomenuty jsou také neuronové sítě a jejich dělení s ohledem na vhodnost k předpovídaní časových řad. Je navrhnut a implementován program pro predikci vývoje více časových řad při burzovním obchodování, kterého základem je model flexibilního neuronového stromu, kterého struktura je optimalizována pomocí imunitního programování a parametry pomocí modifikované verze simulovaného žíhání anebo pomocí optimalizace hejnem částic. Program je nejdříve testován na schopnosti předpovídat jednoduché časové řady a nakonec je testována jeho schopnost předpovídat více časových řad.

Abstract

The diploma thesis comprises of a general approach used to predict the time series, their categorization, basic characteristics and basic statistical methods for their prediction. Neural networks are also mentioned and their categorization with regards to the suitability for prediction of time series. A program for the prediction of the progress of multiple time series in stock market is designed and implemented, and it's based on a model of flexible neuron tree, whose structure is optimized using immune programming and parameters using a modified version of simulated annealing or particle swarm optimization. Firstly, the program is tested on its ability to predict simple time series and then on its ability to predict multiple time series.

Klíčová slova

predikce, předpovídání, časová řada, vícerozmřná časová řada, autoregrese, klouzavé průměry, ARMA, neuronová síť, flexibilní neuronový strom, FNT, MIMO-FNT, simulované žíhání, optimalizace hejnem částic

Keywords

predicton, forcasting, time series, multivariate time series, autoregression, moving averages, ARMA, neural network, flexible neural tree, FNT, MIMO-FNT, immune programming, simulated annealing, particle swarm optimization

Citace

Peter Palček: Předpovídání vývoje více časových řad při burzovním obchodování, diplomová práce, Brno, FIT VUT v Brně, 2012

Předpovídání vývoje více časových řad při burzovním obchodování

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Peter Palček
23. mája 2012

Poděkování

Ďakujem svojmu vedúcemu Ing. Jaroslavovi Rozmanovi, Ph.D. za prejavenú ochotu a odbornú pomoc počas riešenia mojej diplomovej práce.

Obsah

1 Úvod **3**

2 Predpovedanie vývoja časových radov **5**
2.1 Časový rad . . . 5
2.1.1 Delenie časových radov . . . 5
2.1.2 Zložky časového radu . . . 6
2.1.3 Časové rady vývoja cien akcií . . . 6
2.2 Analýza a predspracovanie časových radov . . . 7
2.2.1 Stacionarita . . . 7
2.2.2 Odstránenie trendovej a sezónnej zložky . . . 8
2.3 Predpovedanie časových radov . . . 9
2.3.1 Interval predikcie . . . 10
2.4 Najpoužívanejšie štatistické modely . . . 10
2.4.1 Autoregresia . . . 10
2.4.2 Kĺzavé priemery . . . 10
2.4.3 ARMA . . . 11
2.4.4 ARIMA . . . 11
2.4.5 Nelineárne modely . . . 11
2.4.6 Predikcia viacerých časových radov . . . 11
2.4.7 VAR . . . 12
2.5 Vyhodnotenie metód predikcie . . . 12
2.5.1 Cena . . . 12
2.5.2 Presnosť . . . 12

3 Neurónové siete **14**
3.1 Neurón . . . 14
3.2 Neurónové siete pre predpovedanie časových radov . . . 15
3.2.1 Dopredné neurónové siete . . . 15
3.2.2 Rekurentné siete . . . 16
3.2.3 Porovnanie presnosti predikcie časových radov . . . 18
3.2.4 Automatická úprava topológie siete . . . 19

4 Návrh riešenia problému **21**
4.1 Flexible neural tree . . . 21
4.1.1 Fitness funkcia . . . 22
4.1.2 Optimalizácia štruktúry - algoritmus IP . . . 23
4.1.3 Optimalizácia parametrov . . . 23
4.1.4 Algoritmus SA . . . 23

4.1.5 Algoritmus PSO . . . 24
4.1.6 Hybridný algoritmus pre FNT model . . . 25
4.1.7 Výber vstupného vektoru dát . . . 25
4.1.8 MIMO-FNT . . . 26
4.2 Vstupné dáta . . . 27
4.2.1 Predspracovanie . . . 27
4.2.2 Rozdelenie vstupných dát . . . 27
4.3 Výstup programu . . . 27
4.3.1 Horizont predikcie . . . 28

5 Implementácia **29**
5.1 Architektúra klient-server . . . 29
5.1.1 Server . . . 29
5.1.2 Klient . . . 29
5.2 Implementácia FNT . . . 30

6 Predikcia jedného časového radu **32**
6.1 Logistická rovnica . . . 32
6.1.1 Horizont predikcie . . . 32
6.1.2 Veľkosť dátovej sady . . . 33
6.1.3 Testovanie optimalizácie štruktúry . . . 33
6.1.4 Porovnanie algoritmov optimalizácie parametrov . . . 35
6.1.5 Testovanie vplyvu pomeru dát . . . 38
6.1.6 Predspracovanie . . . 39
6.2 Predikcia časovej rady ceny akcií . . . 39

7 Predikcia viacerých časových radov **45**
7.0.1 Vstupné dáta . . . 45
7.1 Predpovedanie jedného časového radu na základe viacerých . . . 45
7.1.1 Predpovedanie iba pomocou cien akcií . . . 46
7.1.2 Predpovedanie aj s objemom obchodovaných akcií . . . 46
7.2 Predpovedanie viacerých časových radov súčasne . . . 46

8 Záver **49**

A Obsah CD **53**

Kapitola 1

Úvod

Pre predpovedanie cien pri burzovom obchodovaní existuje viacero dôvodov, avšak ten najčastejší je finančný zisk. Každý systém, ktorý by dokázal vyberať víťazov a porazených na dynamickej burze by priniesol majiteľovi veľké bohatstvo. Preto mnohý ľudia, vrátane výskumníkov a investorov, stále hľadajú systém, ktorý by im priniesol veľké zisky.

Predpovedanie cien na burze je však veľmi obtiažne. Ceny akcií, vykazujú vysokú nestálosť, zložitosť a šum, ktoré vyplývajú z nepoľapiteľného mechanizmu burzy. Sú ovplyvnené množstvom vysoko korelovaných ekonomických, politických, ale aj psychologických faktorov. Tieto faktory na seba vzájomne pôsobia veľmi zložitým spôsobom. V dôsledku toho je vhodné uvažovať o predpovedaní vývoja cien viacerých akcií súčasne, v najlepšom prípade bez nutnosti analyzovať vzájomné vzťahy medzi akciami.

Ceny akcií sú reprezentované časovými radmi, ktorých predikciu zastrešujú mnohé metódy, medzi nimi aj neurónové siete.

Využitie neurónových sietí je v poslednej dobe čoraz častejšie. Aplikačné úlohy, na ktoré sú neurónové siete používané, ako sú napr. spracovanie informácií, klasifikácia vzorov alebo situácií, optimalizačné problémy a predikčné úlohy nachádzajú uplatnenie v rôznych oblastiach priemyslu, managementu, finančníctva a inde.

Výhodou neurónových sietí je to, že sa správajú ako univerzálne aproximátory funkcií, to znamená, že nepotrebujeme vedieť presnú charakteristiku problému, ktorý nimi chceme modelovať. Pre predikciu viacerých časových radov cien akcií, medzi ktorými nepotrebujeme poznať súvislosti, je táto vlastnosť veľmi významná.

Cieľom tejto práce je implementovať program, ktorý na základe vstupných časových radov cien akcií natrénuje neurónovú sieť za účelom predpovedania ich vývoja.

V druhej kapitole je popísané základné rozdelenie časových radov, ich analýza a predspracovanie za účelom predikcie, špecifiká týkajúce sa časových radov vývoja cien akcií, problém predpovedania viacerých časových radov, niektoré najpoužívanejšie štatistické metódy a spôsob hodnotenia metód predikcie časových radov.

Tretia kapitola pojednáva o neurónových sietiach so zameraním na tie, ktoré sa používajú na predpovedanie vývoja časových radov.

Vo štvrtej kapitole je navrhnuté riešenie programu, ktorý predpovedá vývoj viacerých časových radov. Jeho základom je flexibílny neurónový strom, ktorý je možné brať ako doprednú neurónovú sieť s premennou topológiou.

Piata kapitola popisuje implementáciu navrhnutého programu ako klient-server aplikácie.

V šiestej kapitole je testovaná schopnosť modelu predpovedať jeden jednoduchý časový rad a je v nej tiež zahrnuté aj testovanie časového radu priebehu vývoja cien akcií.

Experimentovanie s predpovedaním viacerých časových radov je zastrešené v siedmej kapitole, pričom je presnosť predpovede pomocou viacerých radov porovnaná s presnosťou pri predpovedaní samotného časového radu.

Kapitola 2

Predpovedanie vývoja časových radov

Za účelom predpovedania vývoja cien akcií na burze je najskôr potrebné definovať pojem časového radu, jeho vlastností a špecifických vlastností radov udávajúcich vývoj ceny akcií na burze. Ďalej budú v tejto kapitole popísané klasické prístupy k predikcia časových radov.

2.1 Časový rad

Časový rad je rad vecne a priestorovo zrovnateľných pozorovaní (dát), ktoré sú jednoznačne usporiadané z hľadiska času v smere minulosť - prítomnosť. [19]

V časových radoch vývoja cien akcií je často aktuálna hodnota radu nielen závislá od svojich minulých hodnôt, ale aj od hodnôt iných časových radov, ktoré spolu súvisia. Môže byť napr. objem obchodovaných dát alebo vývoj kurzu meny, na ktorej sú dané akcie silne závislé. V takom prípade je vhodné, namiesto predpovedania každého radu zvlášť, predpovedať závislé rady spolu.

Matematicky je možné predstaviť si časový rad ako postupnosť vektorov, ktoré závisia od času t:

$$\vec{y}_t, \quad t = 0, 1, 2, \ldots, T$$

kde T je dĺžka časového radu a každý vektor $\vec{y}_t$ (alternatívne budem tiež zapisovať $\vec{y}(t)$) odpovedá pozorovaniu v čase t. Zložky vektoru môžu byť akékoľvek pozorovateľné premenné (pozorovateľné v čase t).

Vector $\vec{y}_t$ je teda viacrozmerný (angl. multivariate) a jeho hodnota, ktorá reprezentuje K jednorozmerných časových radov je pre čas t zapísaná ako

$$\vec{y}_t = (y_{1t}, y_{2t}, \ldots, y_{Kt})'$$

kde y_{it} značí hodnotu i-tého radu v čase t.

Ak nie je explicitne uvedený vektorový tvar, tak zápis y_t bude znamenať jednorozmerný (angl. univariate) časový rad.

2.1.1 Delenie časových radov

Časové rady sa delia na dva základné typy, a to

- *spojité* - hodnoty časového radu poznáme a môžeme ich merať v každom okamihu

- *diskrétne* - hodnoty poznáme len v určitých nespojitých časových bodoch

Podľa periodicity, s akou zaznamenávame údaje pozorovanej premennej, rozdeľujeme časové rady na

- *dlhodobé* - s periodicitou jedného roku, napr. ročné hodnoty HDP
- *krátkodobé* - s periodicitou kratšou ako jeden rok, napr. rady štvrťročných, mesačných, týždenných alebo denných hodnôt, napr. vývoj kurzu meny.

Podľa vyjadrovaných veličín časové rady členíme na

- *intervalové* časové rady - viažu sa na určitý časový interval a ich hodnota rastie s dĺžkou intervalu (napr. stav na bankovom účte ku koncu kalendárneho mesiaca)
- *okamihové* časové rady - sledovaná veličina nadobúda pozorované hodnoty v určitom časovom okamihu (napr. mzdové náklady za obdobie jedného mesiaca)
- časové rady *odvodených veličín* - obvykle obsahujú podiely dvoch absolútnych veličín (okamihových aj intervalových)

Keďže ceny akcií sú merané v určitých časových bodoch s rovnakým intervalom kratším ako jeden rok (napr. koniec obchodného dňa), ďalej sa zamerám len na časové rady, ktoré sú *diskrétne*, *krátkodobé* a *okamihové*.

2.1.2 Zložky časového radu

Pri klasickej analýze časových radov sa vychádza z predpokladu, že každý časový rad y_t pre $t = 1, 2, ..., n$ je možné rozložiť na štyri zložky: trendovú, cyklickú, sezónnu a nesystematickú (reziduálnu). [1]

- Trendová zložka (T_t) vyjadruje dlhodobú tendenciu vývoja skúmanáho javu.
- Cyklická zložka (C_t) vyjadruje kolísanie okolo trendu, v ktorom sa striedajú fáze rastu a poklesu. Jednotlivé cykly (periódy) sa vytvárajú v období dlhšom ako jeden rok a majú nepravidelný charakter, tj. rôznu dĺžku a amplitúdu.
- Sezónna zložka (S_t) vyjadruje pravidelné kolísanie okolo trendu v rámci kalendárneho roku.
- Nesystematická zložka (a_t) (reziduálna) vyjadruje náhodné a iné nesystematické výkyvy, ale tiež chyby merenia a podobne.

Keďže sa budem zaoberať krátkodobými časovými radmi, tak cyklická zložku nebudem uvažovať, prípadne bude uvažovaná ako sezónna zložka.

2.1.3 Časové rady vývoja cien akcií

Predpovedanie chovania kurzov cenných papierov nie je priamočiare, existujú k nemu rôzne postoje.

Jedným z nich je, že ceny akcií sú *náhodnou prechádzkou*. Tá uvádza, že každý krok je prevádzaný náhodným smerom, čiže na ceny akcií predchádzajúce hodnoty vplyv nemajú.

Iným postojom je *teória efektívnych trhov*. Teória efektívnych trhov predpokladá, že ceny akcií plne reflektujú všetky dostupné informácie, a že sa prispôsobia v momente, kedy

sa nová informácia stáva dostupnou. Z toho vyplýva, že úspešnosť obchodovania nie je možné zvýšiť žiadnou fundamentálnou či technickou analýzou, ani štúdiom historických údajov, čiže predikcia by nemala v tomto prípade zmysel.

Tretím pohľad je technická analýza, ktorá predpokladá, že burza s cennými papiermi sa pohybuje v trendoch a tieto trendy môžu byť zachytené a použité na predikciu. Predpokladom je, že sa v historických dátach vyskytujú vzory, ktoré sú predvýdateľné. Na odhad sa používajú rôzne techniky, ako sú napr. Elliotove vlny a Fibonacciho postupnosť. Nanešťastie väčšina techník používaná technickou analýzou nepreukázala štatistickú valídnosť. [16]

Štvrtým, najrozšírenejším, pohľadom na priebeh cien akcií je, že sú *dynamickým systémom* s *chaotickým* správaním. To, že je systém dynamický, značí, že sa vyvíja v čase. Chaotické správanie systému znamená, že systém že deterministický a ako nedeterministický sa javí preto, že je citlivý na vstupné podmienky, čiže malá zmena vstupných podmienok môže značiť veľkú odchýlku správania systému v čase. Chaotické správanie dynamického systému poukazuje na to, že je systém *nelineárny*, čiže že medzi vstupmi a výstupmi je zložitejšia závislosť ako lineárna. Takýto systém je možné, aj keď v mnohých prípadoch zložité, popísať modelom (napr. neurónovou sieťou). Preto budem časové rady vývoja cien akcií v tejto práci uvažovať ako chaotický systém.

2.2 Analýza a predspracovanie časových radov

Vo väčšine prípadov sa pre predikciu nepoužívajú priamo reálne dáta, ale upravené do formy, aby metóda predikcie dosahovala najlepšie výsledky. Dáta je vhodné zanalyzovať a predspracovať, pričom pre každú metódu predikcie je vhodná iná metóda predspracovania dát. [10]

Všeobecne sa analýza časového radu prevádza podľa [2] nasledovne:

1. Vykreslenie časového radu a preskúmanie hlavných vlastností grafu, konkrétne trendu, sezónnej zložky, očividných náhlych zmien v priebehu a odľahlé hodnoty.

2. Odstránenie trendovej a sezónnej zložky za účelom získania *stacionárnej* reziduálnej zložky.

3. Výber modelu na modelovanie reziduálnej zložky

4. Predikcia bude dosiahnutá predpovedaním reziduálnych hodnôt a následnou inverziou tranformácií popísaných v bode 2.

2.2.1 Stacionarita

Stacionarita je dôležitá vlastnosť časového radu v súvislosti s výberom metódy predikcie, pretože mnoho metód požaduje, aby bol predikovaný časový rad *stacionárny*. [7]

Časový rad y_t je striktne stacionárny ak rozdelenia pravdepodobnosti $(y_{t_1}, \ldots, y_{t_k})$ sú identické s $(y_{t_1+t}, \ldots, y_{t_k+t})$ pre všetky t, kde k je ľubovoľné kladné celé číslo a $(t_1, \ldots, t_k)$ je množina k kladných čísiel. Inými slovami, striktná stacionarita vyžaduje, aby rozdelenia pravdepodobnosti $(y_{t_1}, \ldots, y_{t_k})$ boli invariantné v čase (tj. aby sa štatistické vlastnosti radu v čase nemenili).

Striktná stacionarita sa však v praxi veľmi ťažko overuje, preto sa väčšinou používa slabá stacionarita. Časový rad y_t je slabo stacionárny, ak stredná hodnota y_t a kovariancia

medzi y_t a $y_{t-\ell}$ sú nemenné v čase, kde ℓ je ľubovoľné celé číslo. Presnejšie, y_t je slabo stacionárny, ak

1. $\mu_t = E(y_t) = \mu$, čiže μ_t je nezávislé od času
2. $Cov(y_t, y_{t-\ell}) = \gamma_\ell$ je nezávislé od času pre každé ℓ

Stacionárny časový rad je rovnomerne vyvážený (t. j. s konštantným rozptylom) okolo konštantnej úrovne (t. j. má konštantnú strednú hodnotu), pričom závislosť (korelácia) medzi jeho dvoma ľubovoľnými pozorovaniami závisí len od ich vzájomnej časovej vzdialenosti a nie na ich skutočnej polohe v časovom rade. [21]

Biely šum

Príkladom stacionárneho procesu je *biely šum*, pre kotrý platí:

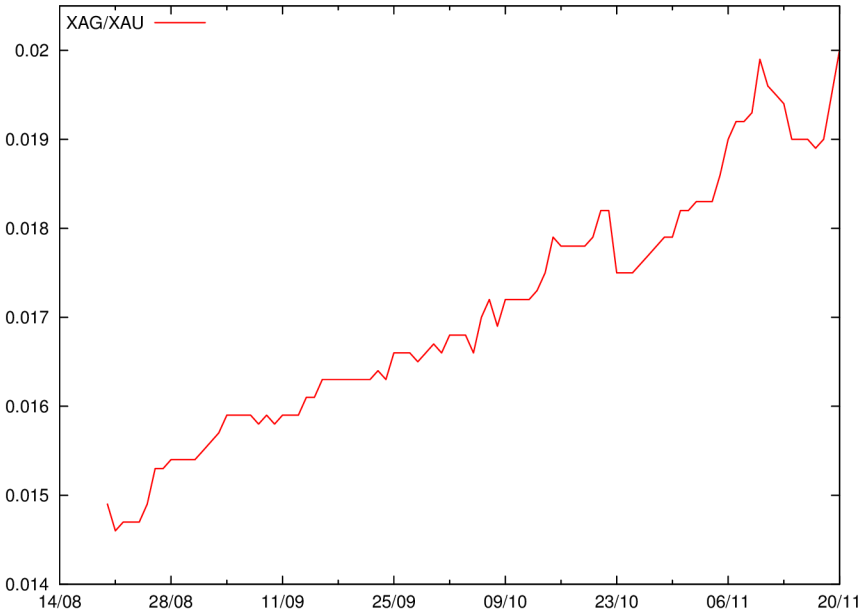
$$E(a_t) = 0,\ D(a_t) = \sigma_a^2,\ cov(a_t, a_{t-k}) = 0 \text{ a } a_t \approx N(0, \sigma_a^2)$$

tj. náhodné veličiny a_t majú v čase t nulovú strednú hodnotu, konštantný rozptyl, sú vzájomne lineárne nezávislé a majú normálne rozdelenie. [1]

Mnoho metód predikcie predpokladá, že reziduálna zložká má charakter bieleho šumu. [7]

2.2.2 Odstránenie trendovej a sezónnej zložky

V prípade, ak sú trend a sezónnosť jasne viditeľné vlastnosti časového radu, ich odstránenie by malo byť prevádzané manuálne, pretože inak by sa metódy predikcie snažili predpovedať práve tieto najviditeľnejšie charakteristiky namiesto toho, aby sa zamerali na skryté vlastnosti systému. Príklad jasne viditeľného trendu je zobrazený na obrázku 2.1.



Obr. 2.1: Časový rad vývoja pomeru ceny striebra a zlata

Odstránenie trendovej a sezónnej zložky môže byť vykonané rôznymi spôsobmi, jedným z nich je napr. *diferenciácia*. Pri odstraňovaní oboch zložiek pomocou diferenciácie sa najskôr odstráni sezónna zložka na takto upravený časový rad sa nasledovne aplikuje ľubovoľná metóda pre odstránenie trendu s časového radu, ktorý pozostáva len z trendovej a reziduálnej zložky.

Sezónnosť je možné odstrániť nahradením radu y_t radom y'_t, ktorý pozostáva z rozdielov prislúchajúcich hodnôt v rámci periódy, ktorej efekt sa snažíme odstrániť. Rad y'_t je vyjadrený vzťahom (2.1), kde s je dĺžka periódy (napr. ak sú údaje merané denne a každý týždeň vykazujú podobný vzor, potom $s = 7$).

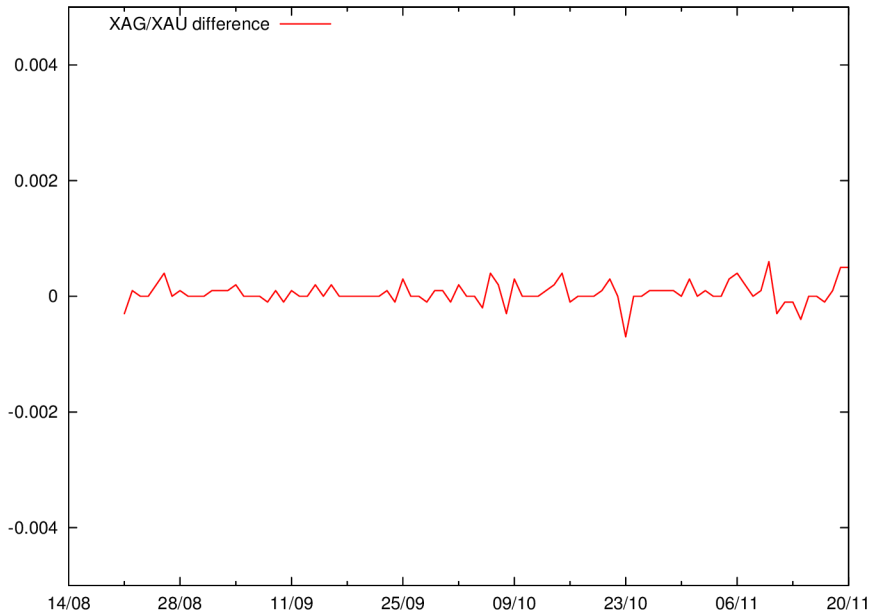
$$y'_t = y_t - y_{t-s} \tag{2.1}$$

Lineárny trend môže byť potom odstránený nahradením časového radu y_t radom y'_t) zobrazeným vo vzťahu (2.2). Tento rad je zložený z rozdielov po sebe idúcich hodnôt.

$$y'_t = y_t - y_{t-1} \tag{2.2}$$

Pri nelineárnom trende sa môže použiť parametrický model (napr. exponenciálna krivka) pre estimáciu trendu a na základe tejto estimácie trendovú zložku odstrániť.

Príklad diferenciácie trendu priebehu vývoja pomeru ceny striebra a zlata daného grafom 2.1 je zobrazený na obrázku 2.2.



Obr. 2.2: Časový rad vývoja pomeru ceny striebra a zlata po diferenciácii trendu

2.3 Predpovedanie časových radov

Predpovedanie (predikcia) časových radov je vo všeobecnosti prevádzaná pomocou štatistických modelov poskytujúcich prevdepodobnú hodnotu v budúcnosti, pričom tieto modely sú zostavené na základe znalosti historických hodnôt daných časových radov.

Predikciu hodnoty y s budúcnosti od času t môžeme zapísať ako

$$\hat{y}_{t+s} = f(y_t, y_{t-1}, \dots) \tag{2.3}$$

kde s je nazývaný *horizont predikcie* a $\hat{y}$ je predikovaná hodnota.

2.3.1 Interval predikcie

Vzťah (2.3) zobrazuje *bodovú predikciu*, ktorá predstavuje odhad hodnoty skúmaného časového radu v stanovenom budúcom okamihu. Takáto predpoveď je vždy zaťažená určitou chybou, takže konkrétne číslo, ktoré poskytuje, je nutné brať s určitou rezervou. Preto je často užitočné počítať s takzvaným *intervalom predikcie*, ktorý je analogický intervalu spoľahlivosti z matematickej štatistiky. Pri predpoklade normálneho rozloženia chýb s nulovou strednou hodnotou, čo vo väčšine prípadov platí, môžeme horný a dolný interval spoľahlivosti zapísať ako vzťah predpovedanej hodnoty $\hat{y}$ a odmocniny zo strednej štvorcovej chyby $RMSE$ (popísaná vzťahom (2.10)) nasledovne

$$\hat{y}_t \pm z \cdot RMSE \tag{2.4}$$

kde z je hodnota, na ktorej závisý požadovaný stupeň spoľahlivosti (napr. pre 95% spoľahlivosť je hodnota $z = 1.96$). [11]

2.4 Najpoužívanejšie štatistické modely

Základom klasickej predikcie dát je vytvorenie štatistického modelu z historických dát, pomocou ktorého sú predpovedané nové hodnoty. Existuje mnoho modelov, vhodnosť použitia sa líši v závislosti od charakteristiky časovej rady. Nižšie sú uvedené niektoré najviac používané modely.

2.4.1 Autoregresia

Autoregresný model (angl. autoregression) je lineárny model používaný na modelovanie stacionárnych procesov a označuje sa AR*(p)*, kde p značí rád modelu.

V prípade časového radu y_t je model AR*(p)* zapísaný formou

$$y_t = \phi_0 + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \tag{2.5}$$

kde $\{\varepsilon_t\}$ je biely šum s nulovou strednou hodnotou a rozptylom σ_a^2, p je celé kladné číslo a ϕ_i sú parametre modelu. Jedná sa v podstate o filter s nekonečnou impulznou odozvou.

Pre zachovanie stacionarity je potrebné vhodne nastaviť parametre (napr. pre AR(1) musí byť $|\phi_1| \leq 1$), na čo sa môže použiť napr. metóda najmenších štvorcov.

2.4.2 Kĺzavé priemery

Model kĺzavých priemerov (angl. moving-average model) je lineárny model často používaný na modelovanie jednorozmerných časových rád. Označuje sa MA*(q)*, kde q značí rád modelu kĺzavých priemerov.

V prípade časového radu y_t je model MA*(q)* zapísaný formou

$$y_t = c_0 + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2.6}$$

kde $\{\varepsilon_t\}$ je biely šum, q je celé kladné číslo, θ_i sú parametre modelu a c_0 je konštanta (často rovná nule). Jedná sa v podstate o filter s konečnou impulznou odozvou.

Modely kĺzavých priemerov sú vždy slabo stacionárne, pretože sú konečnou lineárnou kombináciou bieleho šumu.

2.4.3 ARMA

ARMA (autoregressive moving average) model, niekedy nazývaný Box-Jenkins model, je lineárny model, ktorý sa používa na modelovanie stacionárnych časových radov. Pozostáva z dvoch častí - autoregresnej časti (AR) a z časti kĺzavých priemerov (MA). Obvykle sa označuje ARMA*(p,q)*, kde p je rád autoregresie a q je rád kĺzavých priemerov.

V prípade časového radu y_t je model ARMA*(p,q)* zapísaný formou

$$y_t = \phi_0 + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2.7}$$

kde symboly majú rovnaký význam ako vo vzťahoch (2.5) a (2.6).

2.4.4 ARIMA

ARIMA (autoregressive integrated moving average) model je zovšeobecnením ARMA modelu pre použitie pri časových radoch vykazujúcich nestacionaritu. V ekonomike sú nestacionárne napr. časové rady cien produktu.

Vstupom pre model ARIMA je časový rad obsahujúci trend spôsobujúci nestacionaritu, ktorá je následne odstránená diferenciáciou uvedenou v rovnici (2.2) a získaný časový rad je modelovaný ARMA modelom.

Obvykle sa ARIMA model zapisuje ARIMA*(p,d,q)*, kde p je rád autoregresie, d je rád diferenciácie (koľkokrát bola aplikovaná diferenciácia na časový rad) a q je rád kĺzavých priemerov.

2.4.5 Nelineárne modely

Doposiaľ boli spomenuté len lineárne modely, ale mnohé časové rady (medzi nimi aj časové rady vývinu cien akcií) vykazujú nelineárnu závislosť predikovanej hodnoty od predchádzajúcich hodnôt, vďaka čomu môžu lineárne vykazovať veľkú nepresnosť. Pre modelovanie takýchto časových radov sa používajú rôzne druhy modelov, mnohé z nich odvodené z lineárnych, ako napr. Treshold AR, Smooth Transition AR, non-linear GARCH a iné. [21]

Stále populárnejším spôsobom predikcie nelineárnych časových radov sú tiež neurónové siete, ktoré dokážu aproximovať ľubovoľnú lineárnu alebo nelineárnu funkciu k žiadanej úrovni presnosti.

2.4.6 Predikcia viacerých časových radov

Modely pre predikciu viacrozmerných časových radov skúmajú vnútorné závislosti medzi jednotlivými radmi a s ohľadom na tieto závislosti predpovedajú ďalšie hodnoty. Mnoho

modelov používaných pre jednorozmerné časové rady môže byť zovšeobecnených na priame použitie s viacrozmernými časovými radmi. Jedná sa napr. o modely VAR, VARMA a podobne.

2.4.7 VAR

Ako príklad uvediem často používanú metódu pre predikciu viacrozmerných časových radov. Jedná sa o vektorovú autoregresiu (VAR), ktorá generalizuje jednorozmerné autoregresné (AR) modely. Každú hodnotu popisuje rovnica, ktorá berie do úvahy nielen jej oneskorenie, ale aj oneskorenia ostatných hodnôt v modele.

Viacrozmerný časový rad $\mathbf{y}_t$ modelovaný VAR procesom rádu 1, označovaným VAR(1), je možné zapísať

$$\mathbf{y}_t = \boldsymbol{\phi}_0 + \boldsymbol{\Phi}\mathbf{y}_{t-1} + \mathbf{a}_t \tag{2.8}$$

kde $\boldsymbol{\phi}_0$ je k-rozmerný vektor, $\boldsymbol{\Phi}$ je matica o rozmeroch $k \times k$ a $\{\mathbf{a}_t\}$ je postupnosť sériovo nekorelovaných náhodných vektorov s nulovou strednou hodnotou a kovariančnou maticou $\boldsymbol{\Sigma}$, ktorá musí byť pozitívne definitná. [21]

2.5 Vyhodnotenie metód predikcie

Existuje mnoho metód predikcie časových radov, preto je potrebné určiť parametre, podľa ktorých sa tieto metódy porovnajú. Samozrejme, veľmi dôležitou charakteristikou je presnosť metódy pri predpovedaní sledovaného radu, ale tiež zaváži cena, čiže úsilie, ktoré je potrebné vynaložiť na vývoj a údržbu metódy.

2.5.1 Cena

Pri určovaní ceny metódy sú zahrnuté tri dôležité aspekty:

- Cena vývoja metódy - zahrňuje zdroje požadované pre definíciu aktuálnej hodnoty, ktorá má byť predpovedaná.
- Cena spojená s uložením a získaním dát
- cena výpočtu a údržby - závisí hlavne na množstve procesorového času, ktorý je potrebný na výpočet jednej predpovede a na frekvencii, s akou sú predpovede prepočítavané

Ďalej budem ako cenu hlavne uvažovať dobu výpočtu jednej predpovede časového radu.

2.5.2 Presnosť

Na porovnávanie metód predikcie s ohľadom na presnosť existuje viacero spôsobov, ktorých základom je väčšinou chyba ϵ daná rozdielom predpovedanej hodnoty od skutočnej:

$$\epsilon = y_t - \hat{y}_t \tag{2.9}$$

Rovnice (2.10) až (2.13) zobrazujú najpoužívanejšie spôsoby určenia chyby, kde RMSE je odmocnina zo strednej štvorcovej chyby (angl. root mean square error), MSE je stredná štvorcová chyba (angl. mean square error), MAE je stredná absolútna chyba (angl. mean absolute error), MAPE je percentuálna chyba (angl. mean absolute precentage error), y_t je

aktuálna hodnota a $\hat{y}_t$ je predpovedaná hodnota v čase t, pričom n je počet predpovedaných (skúmaných) hodnôt.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{2.10}$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{2.11}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|(y_t - \hat{y}_t)| \tag{2.12}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|(y_t - \hat{y}_t)|}{y_t} \cdot 100 \tag{2.13}$$

V práci budem väčšinou používať RMSE, ktorá má oproti MSE výhodu v tom, že udáva chybu v rovnakých jednotkách ako sú hodnoty časového radu.

Kapitola 3

Neurónové siete

Neurónové siete sú matematický aparát, ktorý má napodobniť paralelizmus a výpočtovú silu ľudského mozgu. Používanjú sa na mnohé účely, akými sú klasifikácia, spracovávanie dát, riadenie systémov, predikcia a iné. Jedna z definícii neurónovej siete, ktorá je akceptovaná komunitou, znie: [20]

Neurónová sieť je masívne paralelný procesor, ktorý má sklon k uchovávaniu experimentálnych znalostí a ich ďalšieho využívania. Napodobňuje ľudský mozog v dvoch aspektoch:

- poznatky sú zbierané v neurónovej sieti počas učenia
- medzineurónové spojenia (synaptické váhy) sú využívané na ukladanie znalostí

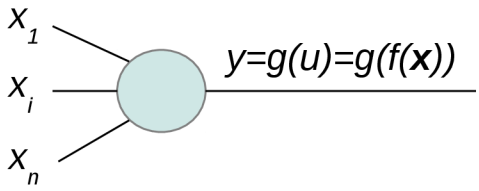
Veľmi významnou vlastnosťou neurónovej siete je, že svojím spôsobom je tzv. *univerzálnym aproximátorom funkcií*. Znamená to, že nepotrebujeme poznať podrobnosti sledovaného systému, ktorý sa snažíme popísať, stačí mať dáta, ktoré do systému vstupujú a k nim odpovedajúce výstupy systému. Podľa týchto dát môžeme natrénovať vhodnú neurónovú sieť a tým sa ju pokúsiť naučiť správať ako sledovaný systém.

V nasledujúcich sekciách stručne popíšem základné prvky neurónových sietí a prehľad typov neurónových sietí, ktoré sa používajú pre predpovedanie časových radov. Podrobnejší prehľad neurónových sietí je možné nájsť napr. v [4].

3.1 Neurón

Základným prvkom neurónovej siete je umelý neurón, ktorého obecný model je zobrazený na obrázku 3.1. Vo všeobecnosti má niekoľko vstupov od iných nerónov alebo z okolitého prostredia a jeden výstup. Operácia, ktorou neurón transformuje svoje vstupy na výstup je spravidla veľmi jednoduchá. Pozostáva z výpočtu *aktivačnej* funkcie, ktorej vstupom je výsledok aplikácie *bázovej* funkcie na vstupný vektor. Na obrázku 3.1 je táto operácia znázornená ako $y = g(f(\mathbf{x}))$, kde f je bázová funkcia, g je aktivačná funkcia, $\mathbf{x}$ je vstupný vektor a y je výtup.

Výpočetná charakteristika neurónových sietí je ovplyvnená použitými aktivačnými funkciami v jej neurónoch a spôsobom ich prepojenia. Dôležitým parametrom siete je tiež učiaci algoritmus, ktorý upravuje váhy prepojení. V tejto práci budem používať *učenie s učiteľom* (angl. supervised learning), ktoré je založené na tom, že sieť má pri svojom učení k dispozícii množinu vstupov a k nim požadovaných výstupov. V procese učenia sa upravujú váhy spojení tak, aby sa minimalizoval rozdiel medzi obdržanou odozvou siete na dané vstupy a výstupmi požadovanými pre tieto vstupy.



Obr. 3.1: Obecný model umelého neurónu [23]

3.2 Neurónové siete pre predpovedanie časových radov

Je zrejmé, že na účel predpovedania časových radov nie sú vhodné všetky druhy sietí. Nižšie uvedené typy sietí, ktoré tvoria základy väčšiny modelov odhadujúcich časové rady, sú uvedené v súvislosti s klasickými metódami predikcie uvedenými v sekcii 2.4.

3.2.1 Dopredné neurónové siete

V dopredných neurónových sietiach (angl. feed-forward neural network, FFNN) sa signál šíri po orientovaných synaptických prepojeniach len jedným smerom, a to dopredu. Medzi príklady takýchto sietí patrí napr. viacvrstvý perceptron (angl. multilayer perceptron - MLP) a siete s radiálnou bázovou funkciou (angl. radial basis function - RBF). MLP pozostáva z viacerých skrytých vrstiev, kde každá vrstva je plne prepojená s nasledujúcou a každý neurón má lineárnu bázovú funkciu (vážený súčet vstupov) a nelineárnu aktivačnú funkciu (okrem neurónov vstupnej vrstvy). RBF siete majú typicky jednu skrytú vrstvu pozostávajúcu s neurónov s nelineárnou RBF ako aktivačnou funkciou (najčastejšie Gaussova funckia) a lineárnu výstupnú vrstvu.

S pevným počtom skrytých neurónov môžu byť dopredné siete brané ako *semiparametrické* aproximátory funkcií, tj. nepredpokladajú určitý tvar funkcie (ako by to bolo pri *parametrických* modeloch), ale nie sú schopné aproximovať ľubovoľnú funkciu, keďže majú pevne daný počet neurónov (nie sú *neparametrické*). Matematický zápis funkcie $\mathbf{F}^{MLP}(\vec{p}) : \mathbb{R}^n \to \mathbb{R}^m$ aproximovanej pomocou MLP je uvedený v rovnici (3.1), kde σ je nelineárna aktivačná funkcia, k je počet neurónov v skrytých vrstvách, $v_{jl}aw_{ij}$ sú váhy a θ_i sú prahy.

$$\mathbf{F}^{MLP}(\vec{p}) = \left(\sum_{j=1}^{k} v_{jl}\sigma \left(\sum_{i=1}^{n} w_{ij}p_i - \theta_j \right) - \theta_l \right), l = 1 \ldots m \tag{3.1}$$

Zápis (3.1) jasne ukazuje súvislosť s rovnicou (2.5). Je viditeľné, že z tohto uhľa pohľadu sú dopredné siete *nelineárnou* autoregresiou. Oproti lineárnemu AR modelu majú výhodu v tom, že môžu modelovať o mnoho zložitejšie charakteristiky časových radov a teoreticky nemusia predpokladať stacionaritu radu. Nevýhodou je potreba veľkého množstva trénovacích dát, a to, že sa môžu u nich vyskytnúť problémy ako je pretrénovanie, sub-optimálne minimum, atď.

Pre predpovedanie časových radov pri burzovom obchodovaní sú dopredné neurónové siete najbežnejšie používané siete, pretože poskytujú dobré generalizáčné schopnosti a sú relatívne ľahko implementovateľné. Aj keď je niekedy obtiažne určiť optimálnu konfiguráciu siete a jej parametrov, tieto siete vykazujú, pri správnom natrénovaní, dobré výsledky.

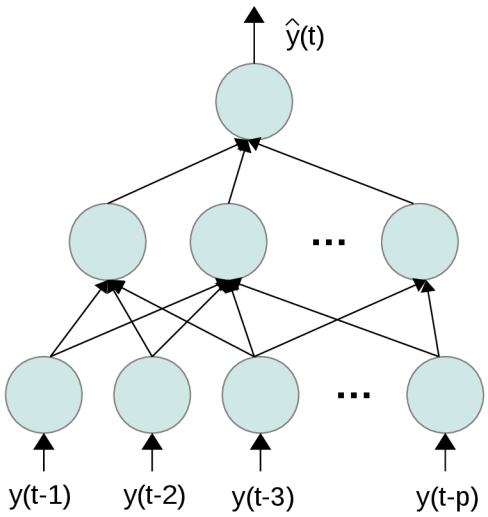
Obecne dokážu RBF siete predpovedať s presnosťou vyššou ako MLP, avšak sú náchylné na problém nerelevantných vstupných hodnôt. Tieto hodnoty zvyšujú dimenzionalitu vstupného priestoru a pri výpočte sa sieť sústredí prevažne na výpočet týchto hodnôt. Možné riešenia zahrnujú redukciu vstupných dát na základe analýzy ich významnosti [13].

Pohyblivé časové okno

Pre predpovedanie sekvenčných dát je potrebné zviesť určitú formu pamäte, aby sa zohľadnili vnútorné medzi-časové súvislosti v týchto dátach. Keďže dopredné siete nemajú spätné väzby ako rekurentné siete (uvedené ďalej), tak sa táto pamäť nahrádza vytvorením trénovacích vzorov z trénovacích dát pomocou *pohyblivého časového okna*. Každú hodnota y_t je predpovedaná na základe p predchádzajúcich hodnôt, čiže

$$\hat{y}_t = f(y_{t-1}, y_{t-2}, \dots y_{t-p})$$

Príklad takejto siete je zobrazený na obrázku 3.2.



Obr. 3.2: Dopredná neurónová sieť s veľkosťou pohyblivého časového okna p

3.2.2 Rekurentné siete

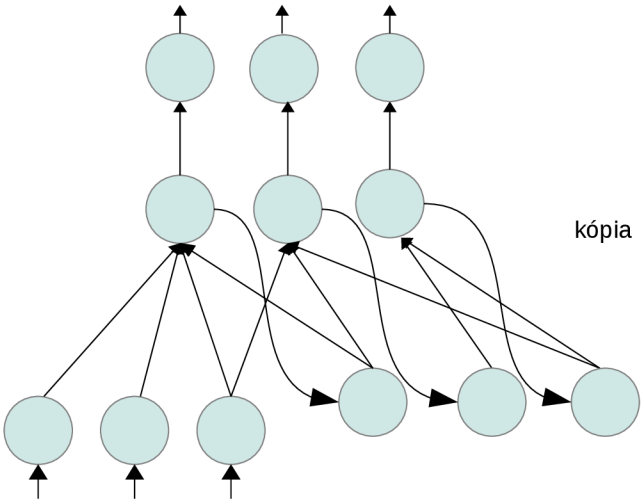
V rekurentných neurónových sietiach (RNN) sa signál môže pohybovať aj smerom od výstupov neurónov späť ku skrytým častiam alebo k vstupom - prepojenia tvoria cyklus. Tým je zabezpečený prvok pamäte, avšak kvôli spätnej väzbe môžu trpieť nestabilitou a citlivosťou na šum. [13]

Zvlášť vhodné na predikciu časových radov sú Elmanove a Jordanove siete, tzv. *jednoduché rekurentné siete* (SRN).

Elmanove siete

Elmanove siete (obrázok 3.3) sú MLP siete s pridanou vstupnou vrstvou, zvanou *stavová*

vrstva, ktorá prjíma ako spätnú väzbu kópiu hodnôt skrytej vrstvy z predchádzajúceho kroku.



Obr. 3.3: Elmanova sieť

Tieto siete môžeme, pre potreby predikcie, odvodiť od možnosti stavového popisu časovej rady, tj.:

$$\vec{y}(t) = \mathbf{C}\vec{s}(t) + \vec{\epsilon}(t) \tag{3.2}$$

kde $\mathbf{C}$ je matica väzieb výstupu na stav reprezentovaný stavovým vektorom $\vec{s}(t)$:

$$\vec{s}(t) = \mathbf{A}\vec{s}(t-1) + \mathbf{B}\vec{\eta}(t) \tag{3.3}$$

kde $\mathbf{A}$ a $\mathbf{B}$ sú matice a $\eta(\vec{t})$ je šum ako v rovnici $\vec{\epsilon}(t)$ v (3.2). Jedná sa v podstate o model ARMA[1,1].

Ak ďalej pradpokladáme, že sú stavy závislé aj na minulom vstupnom vektore a zanedbáme kĺzavý priemer, dostaneme vzťah popisujúci Elmanovu sieť:

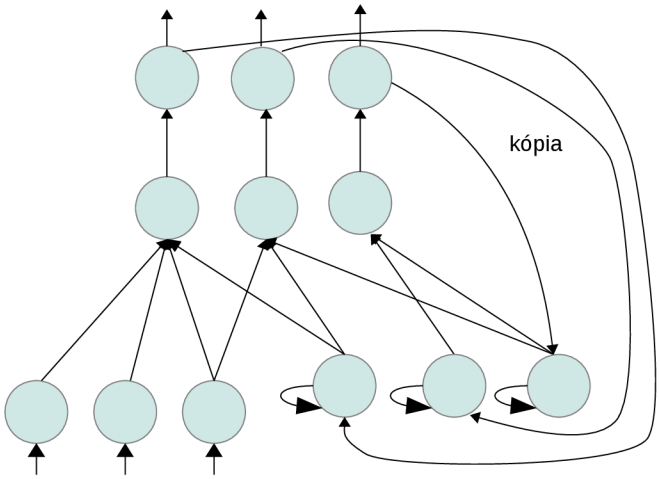
$$\vec{s}(t) = \mathbf{A}\vec{s}(t-1) + \mathbf{D}\vec{y}(t-1) \tag{3.4}$$

Elmanova sieť môže byť natrénovaná akýmkoľvek algoritmom použiteľným pre MLP, ako je napríklad *backpropagation*. Pri trénovaní je však nutné dbať na to, že je pred trénovaním potrebné previesť niekoľko krokov s iniciálnymi hodnotami stavovej vrstvy (väčšinou 0), až kým stavová vrstva nebude vracať vhodné výstupy.

Jordanove siete

Jordanove siete (obrázok 3.4) pozostávajú z MLP s jednou skrytou vrstvou a prepojením z výstupu na pridanú vstupnú (alebo tiež *kontextovú*) vrstvu. Naviac má každý neurón kontextovej vrstvy vlastnú smyčku, tj. je spojený sám zo sebou, a to s váhou $v_i < 1$.

Jordanove siete predstavujú nelineárny ARMA[p,q] model. Zložka autoregresie je zrejmá z toho, že Jordanove siete pozostávajú z MLP. MA model je možné predviesť na špeciálnom prípade Jordanovej siete , ktorý je uvedený na obrázku (3.5). V tomto prípade sú v pridanej vstupnej vrstve namiesto vlastných smyčiek odpočítavané skutočné vstupné hodnoty od



Obr. 3.4: Jordanova sieť

predpovedaných, čím sa ako vstup získava chyba $\hat{\epsilon}(t) = y(t) - \hat{y}(t)$. Tým je zrejmá zložka MA[q] modelu ARMA[p,q], keďže kĺzavé priemery sú lineárnou kombináciou šumu.

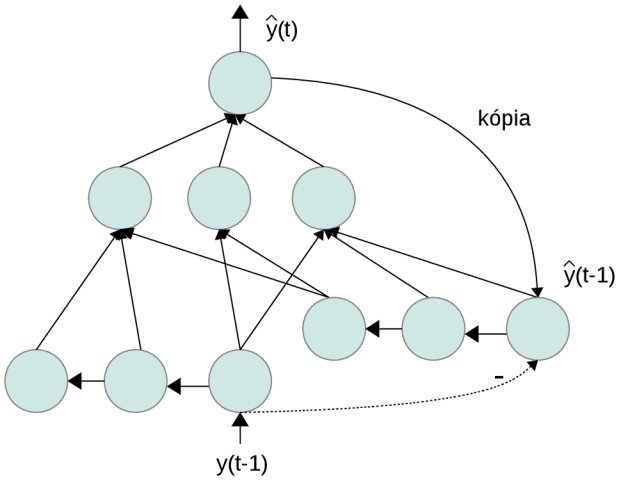
Tak isto ako Elmanova sieť, aj Jordanova sieť môže byť natrénovaná akýmkoľvek algoritmom použiteľným pre MLP, a tiež je potrebné pred trénovaním previesť niekoľko krokov s počiatočnými hodnotami hodnotami kontextovej vrstvy.

3.2.3 Porovnanie presnosti predikcie časových radov

Vo väčšine prípadov dávajú neurónové siete obecne lepšie výsledky ako klasické metódy (zväčša porovnávané s lineárnou, multiregresnou analýzou, ale vyrovnajú sa aj nelineárnym modelom [6]).

Jednoznačné porovnanie presnosti predikcie rôznych typov neurónových sietí je však veľmi obtiažne, pretože, okrem typu sietí (tj. typ prepojenia, počet neurónov a použitá aktivačná funkcia) závisí od množstva iných faktorov, ako sú napríklad:

- horizont predikcie, čiže či sa jedná o krátkodobú (1-3 kroky), strednodobú (4-8 krokov) alebo dlhodobú (viac ako 8 krokov) predpoveď - schopnosť neurónovej siete predpovedať klesá s rastúcim horizontom predikcie [22]
- frekvencia dát (denne, týždenne, mesačne alebo štvrťročne) - dlhšie intervaly majú obecne menej variabilný priebeh
- veľkosť trénovacej množiny - nemala by byť ani príliš malá, ani príliš veľká (napr. pre kurzy mien dvojročné dáta [22]) a tiež by mala byť rozdelená na dve až tri časti (pre predídenie pretrénovaniu siete)
- veľkosť a zloženie vstupného vektora - veľmi dôležitý faktor, vhodné vstupy a ich optimálny počet je veľmi často smerodatný pre presnosť predpovedania pomocou NN [17]



Obr. 3.5: Jordanova sieť bez vlastných smyčiek

- použitý trénovací algoritmus - okrem rýchlosti naučenia siete a schopnosti nájdenia optimálneho riešenia tiež ovplyvňuje schopnosť siete generalizovať
- spôsob vyhodnotenia

Z porovnaní presnosti rôznych typov sietí v [22],[18] vyplýva, že za účelom krátkodobej a strednodobej predikcie sú dopredné neurónové siete rovnako vhodné ako rekurentné siete, dokonca v [14] je ukázané, že najlepšiu presnosť (porovnané boli MLP, RF, Elmanova a Jordanova sieť) pri predpovedaní chaotickej časovej rady dosahuje MLP s dvoma skrytými vrstvami. Z literatúry je tiež zrejmé, že najväčším úskalím je zvolenie vhodného počtu neurónov vo vstupnej a v skrytých vrstvách. Použitie malého počtu môže znamenať nedostatočné natrénovanie (angl. underfitting) a naopak, veľký počet môže spôsobovať pretrénovanie siete (angl. overfitting) - tj. natrénovanie siete špecificky na trénovaciu sadu dát, tým pádom nedostatočnú generalizácium ktorá vedie k nepresným výsledkom pri vyhodnotení.

3.2.4 Automatická úprava topológie siete

Odpoveďou na predídenie problému nesprávneho počtu neurónov je upravovať tento parameter automaticky. K automatickej úprave topológie existuje množstvo rôznych prístupov, ako je napr. zmenšovanie veľkej naučenej siete (pruning algoritmus), zväčšovanie malej nenaučiteľnej siete, metódy založené na kanonickom rozklade a iné. Tieto prístupy môžeme rozdeliť nasledovne:

- Empirické alebo štatistické metódy, ktoré sa používajú na skúmanie efektu vnútorných parametrov a vyberajú pre ne vhodné hodnoty na základe výkonnosti modelu
- Hybridné metódy, medzi ktoré patrí napr. fuzzy inferencia, kde umelá neurónová sieť môže byť interpretovaná ako adaptívny fuzzy systém alebo môže pracovať s fuzzy hodnotami namiesto reálnych čísel

- Algoritmy zväčšujúce resp. zmenšujúce neurónovú sieť, ktoré na základe pôvodnej topológie pridávajú resp. uberajú neuróny a skúmajú, ako je neurónová sieť ovplyvnená zmenami
- Evolučné stratégie, ktoré prehľadávajú priestor topológií tým, že menia počet a prepojenie neurónov aplikovaním genetických operátorov.

Podľa výsledkov v literatúre som došiel k rozhodnutiu použiť doprednú neurónovú sieť. Vzhľadom na veľký vplyv štruktúry na presnosť výsledkov som dospel k záveru, že najvýhodnejšie bude topológiu optimalizovať automaticky. Ako optimalizačnú techniku siete som zvolil evolučný algoritmus, aby nebolo nutné dodávať odborné znalosti.

Kapitola 4

Návrh riešenia problému

V tejto kapitole popíšem návrh programu, ktorého úlohou je predpovedanie časových radov pri burzovom obchodovaní. Jadrom programu je model založený na neurónovej sieti, na základe ktorého je prevádzaná predikcia. Vstupom do modelu sú historické dáta, na základe ktorých model na výstup predá predpovedanú hodnotu.

4.1 Flexible neural tree

Táto podkapitola je zväčša prevzaná z [5].

Použitým modelom na predikciu je flexibilný neurónový strom (ang. flexible neural tree, FNT), ktorý je možné brať ako flexibilnú, viacvrstvú doprednú neurónovú sieť s prepojeniami cez vrstvy a voľnými parametrami aktivačných funkcií.

Tento model rieši v predchádzajúcej kapitole spomenutý problém, že pri dopredných sietiach je ťažké určiť optimálnu konfiguráciu siete a jej parametrov. FNT využíva evolučné stratégie k vytvoreniu neurónového stromu na základe poskytnutých množín operátorov (inštrukcií). Týmto spôsobom sú povolené prepojenia cez vrstvy, rôzne aktivačné funkcie pre každý neurón, a je automatizovaný výber vstupného vektoru. Hierarchická štruktúra je evolvovaná za použitia evolučného algoritmu založeného na stromovej štruktúre. Vyladenie parametrov (čiže natrénovanie siete) zakódovaných v štruktúre môže byť dosiahnuté pomocou optimalizačných algoritmov.

Za pomoci množiny použitých funkcií F a množiny terminálnych inštrukcií T je možné popísať FNT model nasledovne:

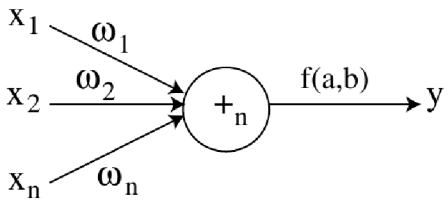
$$S = F \cup T = \{+_2, +_3, \ldots, +_N\} \cup \{x_0, \ldots, x_n\}$$

kde $+_i$, $i = 2, 3, \ldots, N$ značí uzlovú inštrukciu s i argumentami a $x_0, x_1, \cdots, x_n$ sú listové inštrukcie bez parametrov. Výstup z nelistového uzlu je počítaný ako flexibilný neuronový model (zobrazený na obrázku 4.1). Inštrukcia $+_i$ je tiež nazývaná flexibilný neuronový operátor s i vstupmi.

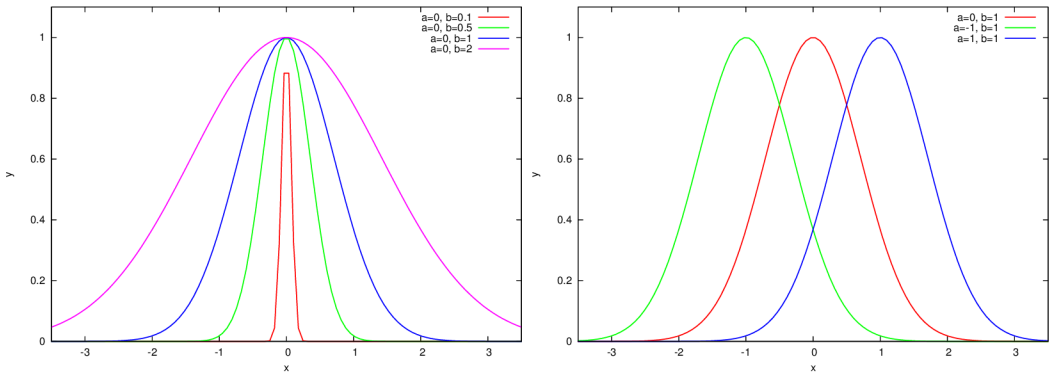
Ak je počas tvorby stromu vybraná uzlová inštrukcia, napr. $+_i$, je generovaných i-náhodných váh $w_1, \ldots, w_i$ a dve nádné hodnoty a_i a b_i použité ako parametre akitvačnej funkcie

$$f(a_i, b_i, x) = e^{-\left(\frac{x-a_i}{b_i}\right)^2} \tag{4.1}$$

Priebeh funkcie a vplyv jej paranetrov a a b je zobrazený na obrázku 4.2.



Obr. 4.1: Flexibilný neurónový operátor [5]



Obr. 4.2: Použitá aktivačná funkcia daná vzťahom (4.1) a vplyv parametrov a a b na jej priebeh

Výstup flexibilného neurónu $+_n$ je vypočítaný ako excitácia vstupov $+_n$ daná nasledovne:

$$net_n = \sum_{j=1}^{n} w_j x_j \tag{4.2}$$

kde x_j $(j = 1, 2, \ldots, n)$ sú vstupy uzlu $+_n$. Výstup uzlu $+_n$ je potom možné zapísať ako

$$out_n = f(a_n, b_n, net_n) = e^{-\left(\frac{net_n - a_n}{b_n}\right)^2}$$

4.1.1 Fitness funkcia

Fitness funkcia vyjadruje v evolučných algoritmoch kvalitu riešenia reprezentovaného daným jedincom. V prípade FNT by fitness funkcia mala v prvom rade odrážať presnosť predikcie, napr. pomocou RMSE resp. MSE (uvedené v rovnici (2.10) resp. (2.11)) a v druhom rade veľkosť neurónového stromu určenú ako počet neurónov. FNT používa dve optimalizačné metódy a trénovanie oboch metód na rovnakej dátovej sade by mohlo viesť k malej generalizácii modelu. Preto sú pre výpočet fitness funkcie použité dve zložky - presnosť po optimalizácii parametrov použitím trénovacej sady dát a presnosť predikcie stromu pri použití testovacích dát. Tým vzniká nutnosť väčšieho počtu historických dát, preto je pomer dát použitých pre optimalizáciu štruktúry a parametrov parametrizovaný.

4.1.2 Optimalizácia štruktúry - algoritmus IP

Na optimalizáciu štruktúry modelu FNT je použitý algoritmus IP [15]. Algoritmus je založený na koncepte evolúcie súboru protilátok, ktoré zakódovávajú kandidátne riešenia k danému problému. Na začiatku sú kandidátne riešenia náhodne generované, čím poskytujú počiatočný súbor s dostatočnou rozmanitosťou. Evolúcia súboru je vedená klonovaním, mutáciou a nahradzovaním protilátok. Tieto procesy uchovávajú rozmanitosť súboru a rozširujú prehľadávaný priestor riešení. Algoritmus je možné popísať nasledovne:

1. Inicializácia. Je vygenerovaný počiatočný súbor (populácia) AB, pozostávajúci z n protilátok Ab_i, $(i = 1, \ldots, n)$. Číslo generácie je nastavené na $G = 1$.

2. Evaluácia. Je predstavený antigén Ag, ktorý reprezentuje riešený problém. Ag je porovnaný s každou protilátkou $Ab_i \in AB$ a je určená ich afinita (pomocou fitness funkcie) f_i.

3. Nahradenie. S pravdepodobnosťou P_r je vygenerovaná a do nového súboru vložená nová protilátka. Týmto spôsobom sú protilátky s nízkou afinitou implicitne nahradené.

4. Klonovanie. Ak nebola generovaná nová protilátka, je vybratá zo súboru protilátka Ab_i s pravdepodobnosťou priamo úmernou jej afinite. Nasledovne je vybraná protilátka s pravdepodobnosťou P_c klonovaná a vložená do nového súboru.

5. Mutácia. Ak nebola žiadna z protilátok naklonovaná, potom je vybraná zo súboru protilátka s pravdepodobnosťou nepriamo úmernou jej afinite. Ak je protilátka vybraná na mutáciu, potom je každá časť jej reťazca zmutovaná s pravdepodobnosťou mutácie P_m.

6. Iterácia - súbor. Opakuj kroky 3–5 pokiaľ nie je zaplnený nový súbor AB o veľkosti n.

7. Iterácia - algoritmus. Inkrementuj číslo generácie (G=G+1) a nový súbor je predaný na evaluáciu v kroku 2. Proces sa opakuje až pokiaľ nie sú splnené ukončovacie podmienky.

4.1.3 Optimalizácia parametrov

Pre nájdenie optimálnej množiny parametrov (váhy a parametre aktivačných funkcií) FNT modelu je možné použiť mnoho algoritmov, medzi ktorými sú genetické algoritmy, gradientné metódy a podobne. Pre porovnanie sú v tejto práci použité dve metódy. Jednou z nich je varianta simulovaného žíhania, nazvaná *degraded ceiling* [3] a druhou je *optimalizácia hejnom častíc* (angl. *particle swarm optimization*, PSO). Počiatočné hodnoty parametrov sú pri oboch algoritmoch nastavené náhodne rovnomerným rozložením v intervale $[-1, 1]$ pre váhy a $[0, 1]$ pre parametre a a b.

4.1.4 Algoritmus SA

Základná myšlienka simulovaného žíhania je, že akceptuje horšie riešenia s pravdepodobnosťou $p = e^{-\frac{\delta}{T}}$, kde $\delta = f(s^*) - f(s)$, s a s^* sú starý a nový vektor riešenia, $f(s)$ značí ohodnocovaciu funkciu, parameter T značí teplotu v procese žíhania. Algoritmus začne hľadanie z vysokej teploty a túto teplotu redukuje za pomoci vzorca $T_{i+1} = T_i - T_i \cdot \beta$. Avšak

rýchlosť chladnutia β a počiatočná hodnota T musia byť vhodne nastavené, pretože ich optimálna hodnota závisí ma skúmanom probléme.
Algoritmus *degraded ceiling* tiež akceptuje horšie riešenia, avšak iným spôsobom. Akceptuje každé riešenie, ktorého hodnotiaca funkcia je rovná alebo menšia ako horný limit B, ktorý je monotónne znižovaný počas prehľadávania. Algoritmus je nasledovný:

1. Nastav počiatočné riešenie s
2. Vypočítaj počiatočnú fitness funkciu $f(s)$
3. Počiatočná hodnota stropu $B = f(s)$
4. Špecifikuj parameter ΔB
5. Pokiaľ nie je splnená podmienka ukončenia
   (a) Definuj okolie $N(s)$
   (b) Náhodne vyber kandidátne riešenie s^* z $N(s)$
   (c) Ak je $(f(s^*) < f(s))$ alebo $(f(s^*) \leq B)$, potom príjmi s^*
   (d) Zníž strop $B = B - \Delta B$

Presnosť a schopnosť konvergencie algoritmu SA silne závisí od výberu susedných riešení. Keďže prehľadávaný priestor je obrovský, vyberám náhodného suseda len v okolí, ktoré je dané určitým, dostatočne malým krokom zmeny parametrov.

4.1.5 Algoritmus PSO

Optimalizácia hejnom častíc je optimalizačná meta-heuristická technika inšpirovaná chovaním hejna vtákov pri hľadaní potravy.
PSO na prehľadávanie priestoru riešení používa populáciu častíc (zvaná hejno), ktoré korešpondujú jedincom v evolučných algoritmoch. Populácia častíc (S) je na začiatku inicializovaná náhodne, pričom každá častica predstavuje kandidátne riešenie - pozíciu, ktorá je reprezentovaná vektorom $\vec{x}_i$. Hejno častíc sa pohybuje v prehľadávanom priestore s rýchlosťou pohybu reprezentovanou rýchlostným vektorom $\vec{v}_i$. Každá častica i si uchováva svoje najlepšie riešenie $\vec{p}_i$ a pozná najlepšie riešenie celého hejna $\vec{g}$. Pohyby častíc v priestore riešení sú ovplyvnené oboma riešeniami, čo má za následok, že sa hejno pohybuje k lepšie ohodnoteným pozíciam. Táto verzia PSO sa nazýva *globálna.*
Algoritmus pri použití cenovej funkcie f je nasledovný:

- Pre každú časticu $i = 1, \ldots, S$:
  - Inicializuj pozície častíc náhodným vektorom $\vec{x}_i \approx U(\vec{b}_{lo}, \vec{b}_{up})$, kde $\vec{b}_{lo}$ a $\vec{b}_{up}$ sú dolná resp. horná hranica prehľadávaného priestoru.
  - Inicializuj najlepšiiu pozíciu častice $\vec{p}_i = \vec{x}_i$
  - Ak f($\vec{p}_i$) $< f(\vec{g})$) aktualizuj najlepšiu pozíciu hejna $\vec{g} = \vec{p}_i$
  - Inicializuj rýchlosť častice $\vec{v}_i \approx U(-|\vec{b}_{up} - \vec{b}_{lo}|, |\vec{b}_{up} - \vec{b}_{lo}|)$
- Pokiaľ nie je splnené kritérium ukončenia:
  - Pre každú časticu $i = 1, \ldots, S$:

* Aktualizuj rýchlosť častice:

$$\vec{v}_i = \omega\vec{v}_i + c_1 r_1(\vec{p}_i - \vec{x}_i) + c_2 r_2(\vec{g} - \vec{x}_i), \tag{4.3}$$

kde ω, c_1, c_2 sú konštanty a r_1, r_2 sú náhodné čísla z intervalu [0,1]

* Aktualizuj pozíciu častice pomocou novej rýchlosti:

$$\vec{x}_i = \vec{x}_i + \vec{v}_i$$

* Ak je f($\vec{x}_i$) < $f(\vec{p}_i)$):
  - Aktualizuj nejlepšiu pozíciu častice $\vec{p}_i = \vec{x}_i$
  - Ak (f($\vec{p}_i$) < $f(\vec{g})$), aktualizuj najlepšiu pozíciu hejna $\vec{g} = \vec{p}_i$

- V $\vec{g}$ sa teraz nachádza najlepšie nájdené riešenie

Výber parametrov

Za najdôležitejší parameter konvergencie PSO je považovaná ω z rovnice (4.3), zvaná inerciálna váha, ktorá kontroluje dopad predchádzajúcich rýchlosti na súčasnú. Tým reguluje vyváženie medzi globálnou (veľká hodnota - prehľadávanie nových oblastí) a lokálnou explo-ráciou (malá hodnota - dolaďovanie aktuálne prehľadávanej oblasti). Pôvodne je ω uvažo-vaná ako vhodne zvolená konštanta, avšak niektoré pokusy ukazujú, že k lepším výsledkom je možné dôjsť zvolením $w \approx 1.2$ (za účelom prehľadania veľkého priestoru) a postupným lineárnym znižovaním jej hodnoty na 0.
Parametre c_1 a c_2 v rovnici (4.3) nie sú kritické pre konvergenciu, avšak môžu ju urýchliť. Zvyšajne sa určujú hodnoty $c_1 = c_2 = 2$, ale niektoré pokusy ukazujú, že hodnoty $c_1 = c_2 = 1.49$ môžu priniesť lepšie výsledky.

4.1.6 Hybridný algoritmus pre FNT model

Pre nájdenie optimálneho alebo takmer optimálneho modelu FNT, sú striedavo používané optimalizácie štruktúry a parametrov. Kombináciou argoritmu IP s algortimom SA resp. PSO, vzniká hybridný algoritmus pre evolúciou FNT modelu:

1. Definícia parametrov. Najskôr je potrebné definovať parametre oboch optimalizačných algoritmov, ako napr. veľkosť populácie.
2. Inicializácia. Náhodne vytvor N FNT modelov $A(t)$. Nastav $t = 0$.
3. Optimalizácia parametrov. Pre každý FNT model $A_0(t), A_1(t), \ldots, A_N(t)$, optimalizuj váhy pomocou SA alebo PSO.
4. Optimalizácia štruktúry. Pomocou IP vytvor novú populáciu A(t+1). Nastav $t = t+1$.
5. Iterácia. Novým súborom pokračuj v bode 3. Pokiaľ je splnená podmienka ukončenia, ukonči.

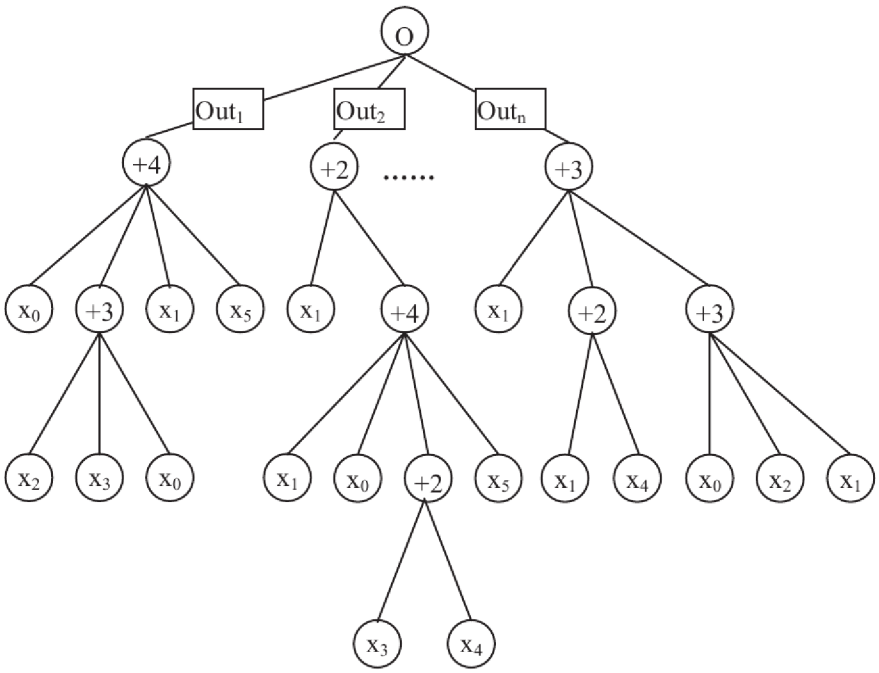
4.1.7 Výber vstupného vektoru dát

V predchádzajúcej kapitole bol spomenutý problém výberu vstupného vektoru príznakov. V prípade modelu FNT je identifikácia dôležitých vstupných hodnôt daná priamo charakterom konštrukcie modelu, ktorý dovoľuje flexibílnemu neurónovému stromu identifikovať tieto hodnoty pri zostavovaní výpočetne efektívneho a účinného modelu predikcie. Mechanizmus výberu môžeme popísať nasledovne:

- Na začiatku sú vstupné hodnoty vybrané s rovnakou pravdepodobnosťou
- Vstupy, ktoré prispievajú k lepšiemu výsledku siete majú vyšiu pravdepodobnosť že sa evolučným procesom dostanú do ďalšej generácie
- Evolučné operátory (ako je napr. kríženie alebo mutácia) poskytujú metódu výberu vstupov, podľa ktorej by si mal FNT vybrať vhodné hodnoty automaticky

4.1.8 MIMO-FNT

Keďže práca je zameraná na predikciu viacerých časových radov, bolo by vhodné, mať okrem vstupu z viacerých radov aj viaceré výstupy, pričom každý z nich by prislúchal jednému radu na vstupe. Takýto model v podstate odpovedá modelu viacrozmerného časového radu. FNT má v základe len jeden výstup, ale je možné ho ľahko upraviť na model s viacerými vstupmi a viacerými výstupmi (angl. multiple input multiple output), v skratke MIMO FNT.
Úprava spočíva vo vytvorení m uzlov, ktoré predstavujú výstupy modelu. Príklad je zobrazený na obrázku 4.3, kde $Out_1, Out_2, \ldots, Out_m$ reprezentujú výstupy modelu. Za povšimnutie stojí, že v koreni nie je použitá žiadna aktivačná funkcia, vracia len váženú sumu výstupov modelu.



Obr. 4.3: Typická reprezentácia neurónového stromu pre MIMO s inštrukčnou sadou $F = \{+_2, +_3, +_4\}$ a so šiestimi vstupmi $x_0, x_1, \ldots, x_5$ [5]

Model MIMO-FNT je možné bez straty presnosti použiť aj pre predikciu jednorozmerného časového radu, ale v prípade, ak sú na vstupe viaceré rady a je požadovaný len jeden výstup, je výpočetne výhodnejšie použiť model FNT. Preto výber medzi FNT a MIMO-FNT nechávam na užívateľovi aplikácie.

4.2 Vstupné dáta

Program by mal užívateľovi na jednej strane poskytovať voľnosť výberu vstupných dát, čiže možnosť importovať rôzne časové rady za účelom skúmania súvislostí a experimentácie, no na druhej strane by mal byť schopný poskytnúť automatické získanie historických dát z webových služieb firiem, ktoré tieto dáta zverejňujú. Preto je užívateľovi poskytnutá možnosť výberu a vloženia dát ručne, pomocou textového súbora alebo získať ceny obchodovaných akcií podľa ich identifikátora z webovej služby *finance.yahoo.com*, ktorá poskytuje k danému účelu jednoduché rozhranie.
Zo vstupných dát je pred načítaním do modelu odstránený predom vybraný počet hodnôt určený na vyhodnotenie.

4.2.1 Predspracovanie

Použitý model pre správnu funkcionality požaduje, aby boli pre každý meraný časový okamih dáta všetkých radov dostupné, tj. aby boli dáta kompletné. Program prípadné chýbajúce alebo chybné hodnoty neošetruje, očakáva korektný a kompletný vstupný súbor dát.
Keďže je použitá neurónová sieť s aktivačnou funkciou, ktorá vracia na výstupe neurónov hodnotu v intervale $[0, 1]$, je nutné dáta normalizovať, čiže tranformovať ich do určitého malého intervalu (v tomto prípade $[0, 1]$). Jedným zo spôsobov je min-max normalizácia, ktorá z intervalu $[min_A, min_B]$ transformuje dáta do intervalu $[min_B, max_B]$:

$$v' = \frac{v - min_A}{max_A - min_A}(max_B - min_B) + min_B$$

kde v je aktuálna hodnota, v'je normalizovaná hodnota.
Normalizácia bude mať za následok, že sa výstupné hodnoty z modelu budú musieť upraviť späť do podoby pred predspracovaním.

4.2.2 Rozdelenie vstupných dát

Vstupné dáta sú rozdelené do troch skupín, a to:

- trénovacie dáta pre paramterickú optimalizáciu
- testovacie dáta, pričom presnosť modelu vyhodnotená na týchto dátach je zložkou celkovej fitness funkcie štruktúrnej optimalizácie
- prípadné vyhodnocovacie dáta, použité pre určenie presnosti modelu

Ďalej sú trénovacie a testovacie dáta spracované pohyblivým časovým oknom, ktoré je rovné podielu počtu vstupov siete s počtom časových radov, pričom veľkosť sa pre každý časový rad určuje experimentálne.

4.3 Výstup programu

Výstupom programu budú štatistické udaje o presnosti metódy, grafické zobrazenie odchýlky predpovedaných a skutočných hodnôt predikovaných časových rád a štruktúra FNT (alebo MIMO-FNT), ktorá bola evolvovaná a pomocou ktorej bola vykonaná predpoveď.

4.3.1 Horizont predikcie

V predchádzajúcej kapitole som spomenul, že predpoveď časových radov pomocou neurónových sietí je najpresnejšia krátkodobá, a že pri dlhodobej predpovedi sú presnejšie klasické metódy (popísané v časti 2.4). Pre dosiahnutie najlepších výsledkov je preto zvolený horizont predikcie jeden krok (tzv. one-step ahead prediction). V časti 6.1.1 je tiež ukázaná predikcia dlhšieho časového úseku s tým, že sú ako vstupné dáta brané predpovedané hodnoty z minulých krokov.

Kapitola 5

Implementácia

V nasledujúcich častiach popíšem implementáciu aplikácie navrhnutej v predchádzajúcej kapitole.

5.1 Architektúra klient-server

Program som sa snažil implementovať tak, aby bol použiteľný nie len pre testovanie presnosti počas vývoja, ale aj v praxi. Preto je program navrhnutý ako klient-server aplikácia - tým je možné prevádzať výpočetne náročné operácie na výkonnejšom serveri (prípadne distribuovane) a užívatelia môžu pracovať s intuitívnejším grafickým rozhraním. Pomocou nastavenia parametrov je potom možné voliť medzi modelmi FNT a MIMO-FNT a tiež voliť medzi vyhodnotením predikcie na vyhodnocovacích dátach a samotnou predikciou.

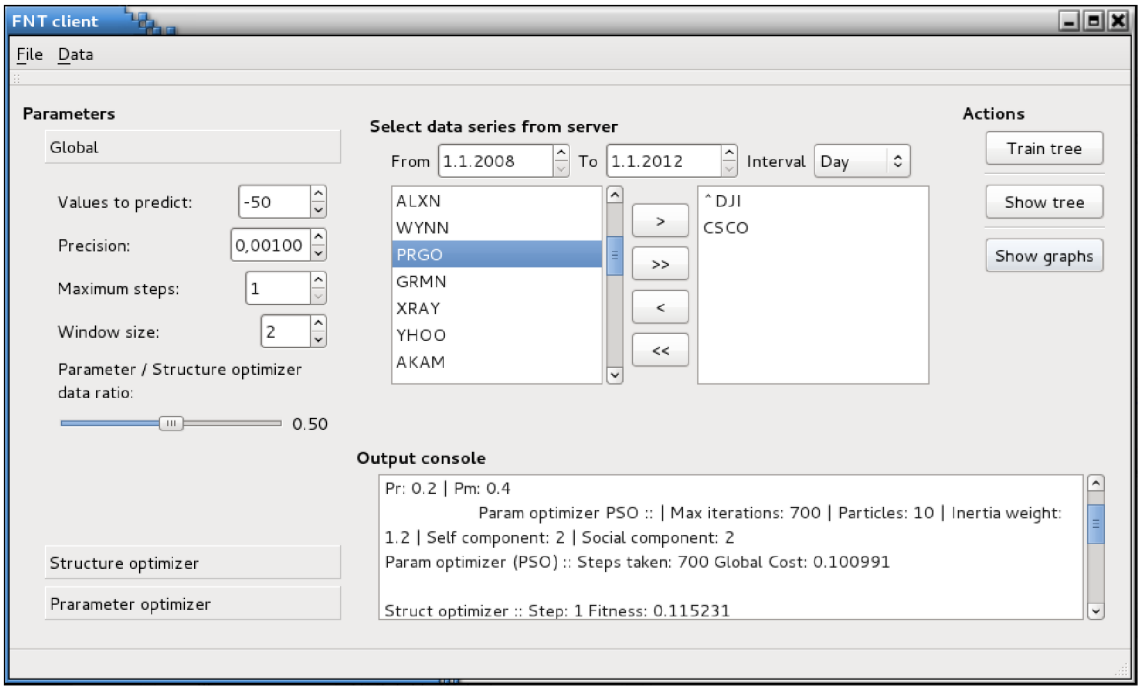
5.1.1 Server

Serverová časť je implementovaná v jazyku C++ za použitia knižníc *boost* pre sieťovú komunikáciu, prácu so súbormi a pre spracovávanie parametrov príkazovej riadky. Pre účely jednoduchšej distribúcie a konfigurácie sú pre preklad použité nástroje *autotools*.
Server implementuje modely FNT a MIMO-FNT, medzi ktorými je možné si vybrať pri započatí komunikácie so serverom. Tiež poskytuje rozhranie pre získavanie dát uverejňovaných portálom *finance.yahoo.com*, ktoré sťahuje do lokálneho adresára len v prípade, ak ešte nie sú prítomné. Z týchto dát program dokáže získať ceny akcií v daný deň, týždeň, alebo mesiac. K doplneniu väčšej voľnosti je možné do aplikácie importovať dáta vo formáte `csv`, kde každý stĺpec reprezentuje jeden časový rad.
Na základe viacerých vstupných radov je možné predpovedať vývoj len jedného radu (model FNT), alebo tiež predpovedať hodnoty viacerých časových radov súčasne.
Komunikácia medzi klientmi a serverom prebieha pomocou soketov, takže je možné sa k serveru pripojiť aj pomocou jednoduchého konzolového klienta, akým je napr. `telnet`.

5.1.2 Klient

Klient s grafickým užívateľským rozhraní (obrázok 5.1) je, podobne ako server, implementovaný v C++. Konkrétne sú využité multiplatformné knižnice Qt. Rozhranie sprístupňuje všetky funkcie implementované na serveri, čiže výber akcií pre predpovedanie z portálu *finance.yahoo.com*, import vlastných dát, trénovanie FNT alebo MIMO-FNT modelov na základe zvolených parametrov.

Implementovaný klient ďalej umožňuje zobrazenie grafu priebehu spolu s predpovedanými hodnotami, k čomu využíva knižnicu `qwt`. Tiež umožňuje, za pomoci nástrojov *graphviz* použitých ako knižnice, grafické zobrazenie štruktúry evolvovaného stromu. Súčasťou klienta je tiež textové okno, ktoré zobrazuje správy od serveru informujúce o priebehu trénovania modelu.



Obr. 5.1: Grafické užívateľské rozhranie klienta

5.2 Implementácia FNT

V rámci automatickej úpravy topológie siete pomocou evolučných algoritmov má FNT veľkú výhodu v tom, že je stromom. Tým pádom odpadá réžia pre zostavovanie fenotypu na základe evolvovaného genotypu za účelom vyhodnotenia riešenia. Inou výhodou je, že je možné výstupnú hodnotu celého stromu získať pomocou prehľadávania do hĺbky (depth first search).
Nad stromom je tiež možné jednoducho vykonávať operácie kríženia a mutácie. V algoritme je použitá len mutácia uzlov stromu, a to tak, že je n náhodných potomkov uzlu odstránených a m náhodných potomkov pridaných. Tým sa zabezpečí čiastočné uchovanie evolvovaného riešenia. Kvôli spomenutým výhodám pri štruktúrnej optimalizácii narábame priamo zo stromom a nie s jeho reprezentáciou pomocou reťazca.
Pri optimalizácii štruktúry je použitý *elitizmus*, čiže najlepšie riešenie sa vždy dostane do nasledujúcej populácie. Do nasledujúcej populácie sa tiež ale dostane mutácia najlepšieho jedinca, aby sa zlepšila šanca vyviaznutia z lokálneho minima.
Optimalizácia prebieha iteratívne, tj. pri každom kroku algoritmu optimalizácie štruktúry sa optimalizujú aj parametre. Tým je možné zvoliť menší krok pri parametrickej optimalizácii a ponechať priestor optimalizácii štruktúry.

Pri parametrickej optimalizácii nie je potrebné pracovať so štruktúrov stromu, ale je potrebné efektívne meniť jeho parametre tak, aby bolo božné dané riešenie vyhodnotiť. Preto každý strom obsahuje štruktúru mapujúcu parametre všetkých uzlov stromu, inštrukčných aj vstupých. Tým je urýchlená práca najmä pri optimalizácii veľkých stromov.

Kapitola 6

Predikcia jedného časového radu

V nasledujúcich sekciách bude otestovaná schopnosť modelu predpovedať jeden časový rad. Pre potreby určenia vplyvu parametrov a doladenia návrhu riešenia otestujem najskôr implementovaný model na jednoduchom, umelom časovom rade. Zvolená bola logistická rovnica, ktorá vykazuje chaotické správanie, takže je vhodná pre predbežné ohodnotenie metódy. Ďalej, za účelom určenia schopnosti predpovedať vývin akcií na burze, bol otestovaný časový rad ceny akcií indexu Dow Jones. Tento rad bude tiež použitý pri predikcii viacerých časových radov, aby boli preskúmané zmeny presnosti a stability predikcie pri použití viacerých časových radov ako vstupu.

6.1 Logistická rovnica

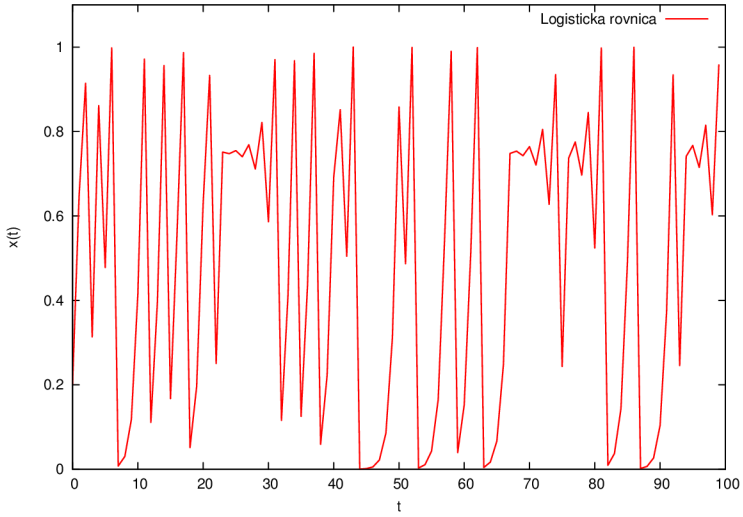
Logistická rovnica je nelineárna rovnica, ktorá simuluje rast populácie a pri určitých hodnotách svojich parametrov vykazuje chaotické strávanie. Je daná vzťahom:

$$x(t+1) = rx(t)(1-x(t)), \qquad x(0) \in (0,1), \tag{6.1}$$

kde $r \in [1,4]$ je preddefinovaný parameter, ktorý definuje charakter rovnice. V tomto tvare sú výstupom rovnice pre každý časový okamih t hodnotoy v intervale $[0,1]$. Pre dosiahnutie chaotického správania sú zvolené parametre $x = 0.2027$ a $r = 4$, rovnako ako v [14]. Časový rad logistickej rovnice pre tieto parametre je zobrazený na obrázku 6.1).

6.1.1 Horizont predikcie

Ako bolo v predchádzajúcej kapitole spomenuté, otestoval som schopnosť siete predpovedať hodnoty viac ako o jeden krok dopredu, a to tak, že ako vstup pre predpovedanie krokov väčších ako 1 som bral výstup z krokov predchádzajúcich. Z obrázku 6.2 je zreteľná značná odchýlka predikcie od skutočných hodnôt. Je ale tiež vidieť, že je týmto spôsobom s dostatočnou presnosťou možné dopredu predpovedať krátky úsek (1-8) hodnôt jednoduchého časového radu. Po úseku dobrej predpovede vidieť postupné, rapídne zhoršenie presnosti a úplné odklonenia sa od modelovaného systému.
Pre dosiahnutie najlepšej presnosti predikcie budem preto používať predikciu s jedným krokom dopredu, pomocou ktorej boli predpovedané hodnoty zobrazené na obrázku 6.3.



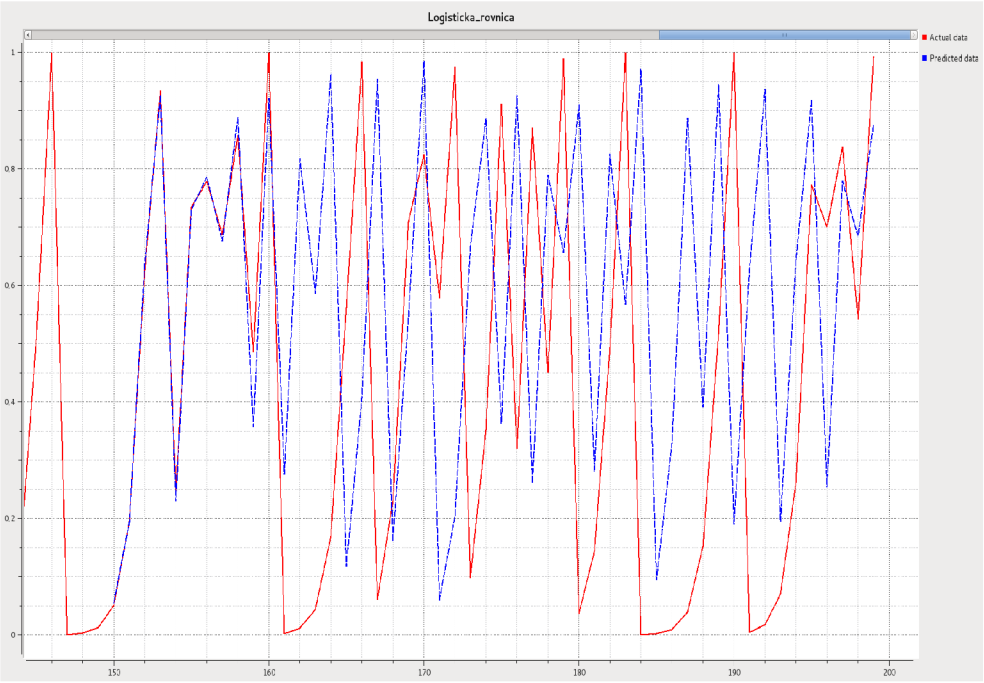
Obr. 6.1: Prvých 100 hodnôt časového radu logistickej rovnice pre parametre $x = 0.2027$ a $r = 4$

6.1.2 Veľkosť dátovej sady

Ako bolo spomenuté v časti 3.2.3, dátová sady by nemala byť ani príliš malá, ani príliš veľká. Vhodná veľkosť dátovej sady ovplyvňuje hlavne čas učenia siete, ktorý je, pre potreby zostavenia modelu FNT, potrebné znižovať, pretože sieť je potrebné optimalizovať pre každého jedinca. Optimálna veľkosť ale silne závisí od skúmaneho časového radu, preto tento parameter určím až pri testovaní časového radu ceny akcií. V literatúre sa väčšinou veľkosť dátovej sady pri testovaní logistickej rovnice pohybuje v rádoch stoviek, preto je použitá dátová sada o veľkosti 200 hodnôt. Efektívne sa pri zostavovaní modelu použije 150 hodnôt, pričom zvyšných 50 je ponechaných na vyhodnotenie predikcie.

6.1.3 Testovanie optimalizácie štruktúry

V prvotných experimentoch som sa snažil obmedziť optimalizáciu parametrov a zamerať sa na topológiu siete. Hlavným zámerom bolo otestovať schopnosť modelu optimalizovať štruktúru do takej miery, aby bol vybraný optimálny vstupný vektor. Pri logistickej rovnici som experimentálne zistil (tabuľka 6.1), že optimálny počet historických hodnôt je 1, čo je, vzhľadom na zápis funkcie, očakávané. Hybridný algoritmus ale túto skutočnosť nezachytil, a tak som sa snažil dosiahnuť prehľadanie čo najväčšieho priestoru riešení. Modifikácia hybridného algoritmu tak, aby v každom kroku optimalizoval len novo vytvorené stromy a stromy vzniknuté mutáciou optimalizoval s menším krokom, priniesola urýchlenie behu algoritmu, čím bolo možné prehľadať väčší priestor riešení topológie. Táto modifikácia však prieniesla len mierne zvýšenie výslednej presnosti a požadovaný automatický výber optimálnych vstupov siete nebol dosiahnutý. K veľkému vylepšeniu neviedli ani variácie spôsobu mutácie jedincov.



Obr. 6.2: Dlhodobá predikcia hodnôt logistickej rovnice

Vplyv počtu vstupných hodnôt

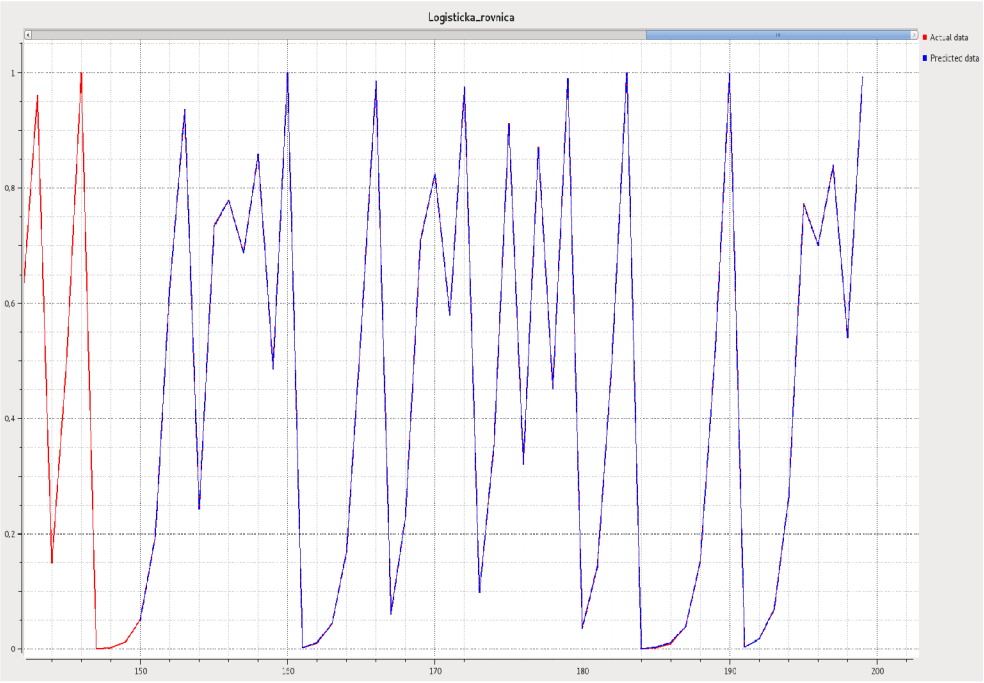
V tabuľke 6.1 je zobrazená chyba RMSE pri trénovaní a pri evaluácii predikovaných hodnôt v závislosti od dimenzie vstupného vektoru. Z tabuľky je okrem jasne viditeľnej najvyššej presnosti jednoprvkového vstupného vektoru tiež vidieť, že s rastúcim počtom vstupných hodnôt klesá chyba evaluácie až je menšia než chyba, ktorá vznikla pri trénovaní.

Počet vstupov	1	2	3	4	5
RMSE trénovanie	0.0237	0.0585	0.0594	0.0619	0.0613
RMSE evaluácia	0.0235	0.0613	0.0595	0.0579	0.0560
Čas [s]	42	45	56	62	71

Tabuľka 6.1: Vplyv počtu vstupných hodnôt pri výške stromu 3 a maximálnom počte potomkov uzlu 3

Vplyv parametrov veľkosti stromu

Optimálna veľkosť stromu je dôležitá hlavne kvôli veľkosti prehľadávaného priestoru a časovej náročnosti jeho prehľadávania. V tabuľke 6.2 je zobrazený vplyv parametrov veľkosti strom (C - maximálny počet potomkov uzlu a D - maximálna hĺbka stromu) na presnosť a čas natrénovania pri rovnakom počte krokov. Je vidieť, že pri zvyšovaní veľkosti stromu dochádza k strate presnosti, čo je spôsobené horšou konvergenciou optimalizátoru parametrov, ktorý musí prehľadávať veľké množstvo riešení. Popri znižujúcej sa presnosti sa pri veľkom strome zvyšuje čas výpočtu, pričom najlepšie výsledky boli dosiahnuté pre maximálnu hĺbku stromu 4 a maximálny počet potomkov 3. Presnosť je samozrejme závislá od



Obr. 6.3: Predikcia hodnôt logistickej rovnice s krokom rovným 1

predpovedaného radu, ale ako počiatočný bod je to dostačujúce.

Parametre stromu	D2,C3	D3,C3	D4,C3	D5,C3	D3,C2	D3,C4	D3,C5
RMSE trénovanie	0.2183	0.0594	0.0269	0.0335	0.2631	0.0343	0.0412
RMSE evaluácia	0.2186	0.0595	0.0260	0.0325	0.2678	0.0358	0.0440
Čas [s]	20	60	225	560	28	217	391

Tabuľka 6.2: Vplyv parametrov veľkosti stromu pri 3 vstupných hodnotách

Vplyv pravdepodobností

V tabuľke 6.3 sú zobrazené priemery presností meraní prevádzané po 100-krokovej optimalizácii so stromom, ktorý má jeden vstup, maximálnu hĺbku 3 a maximálny počet potomkov uzlu 3. Z tabuľky je vidieť že pre klonovanie je vhodné zvoliť väčšiu pravdepodobnosť, pre nahradenie pravdepodobnosť bližšiu nule a pre mutáciu približne strednú hodnotu. Ďalej v programe budem používať hodnoty $Pc = 0.6$ (lepšie generalizuje a po dostatočne veľkom počte krokov sa presnosť vyrovná vyššej hodnote), $P_r = 0.2$ a $P_m = 0.4$.

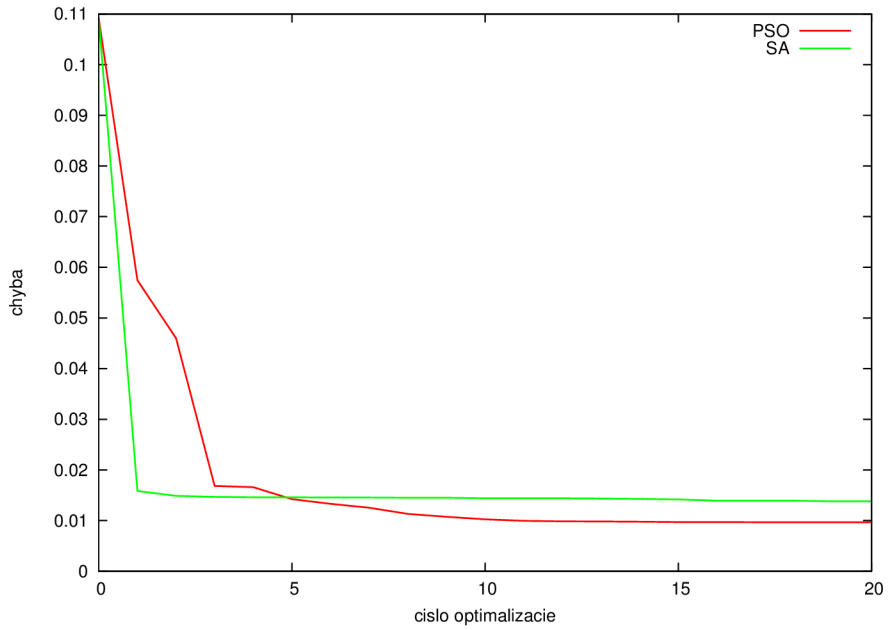
6.1.4 Porovnanie algoritmov optimalizácie parametrov

Porovnanie optimalizácie parametrov bolo prevádzané pre náhodné stromy a pre objektívne porovnanie so snahou rovnakej dĺžky behu oboch algoritmov. Obe metódy optimalizácie vykazovali pri testovaní logistickej rovnice dobré výsledky s tým, že SA bola v priemere o $3,9\%$ presnejšia ako PSO. Aj keď algoritmus SA dosahoval pri jednom behu presnejší

Pravdepodobnosti	RMSE trenovanie	RMSE vyhodnotenie	Čas [s]
Pc=0.2,Pr=0.2,Pm=0.4	0.0434	0.0426	18
Pc=0.4,Pr=0.2,Pm=0.4	0.0352	0.0341	15
Pc=0.6,Pr=0.2,Pm=0.4	0.0301	0.0291	13
Pc=0.8,Pr=0.2,Pm=0.4	0.0256	0.0259	12
Pc=0.6,Pr=0.4,Pm=0.4	0.0738	0.0722	28
Pc=0.6,Pr=0.6,Pm=0.4	0.0721	0.0718	35
Pc=0.6,Pr=0.8,Pm=0.4	0.0643	0.0623	40
Pc=0.6,Pr=0.2,Pm=0.2	0.0386	0.0377	20
Pc=0.6,Pr=0.2,Pm=0.6	0.0441	0.0435	21
Pc=0.6,Pr=0.2,Pm=0.2	0.0402	0.0386	23

Tabuľka 6.3: Vplyv pravdepodobností pri 1 vstupe, maximálnej výške stromu 3 a maximálnemu počtu potomkov 3

výsledok, jeho priemerná presnosť bola pri prevedení dvadsiatich optimalizácií toho istého stromu horšia ako presnosť PSO o $1,6\%$. Na obrázku 6.4 je zobrazený priebeh viacnásobnej optimalizácie jedného stromu. Z grafu je zreteľne vidieť rýchlejšia konvergencia algoritmu SA, ale tiež jeho obmedzená presnosť.

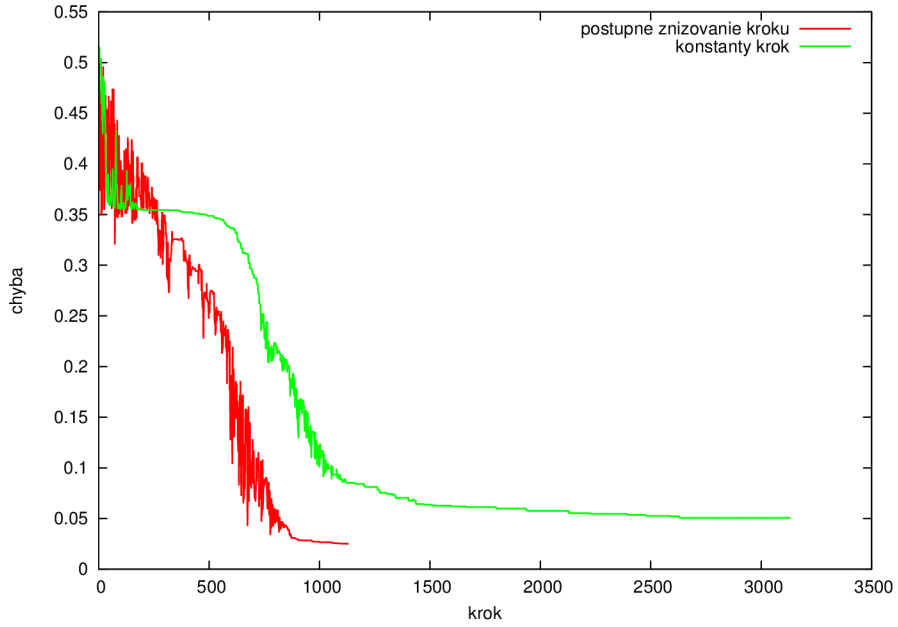


Obr. 6.4: Porovnanie priebehu viacnásobnej optimalizácie jedného jedinca pomocou algoritmov SA a PSO

Vylepšenie SA

Pri prvotnom testovaní algoritmu SA doli dosiahnuté priemerné až podpriemerné výsledky za neuspokojivú dobu behu algoritmu. Preto som skúsil pozmeniť okolie výberu suseda

spôsobom podobným pre PSO. Ako počiatočný krok zmeny parametrov sa nastaví fitness funkcia riešenia určeného pre optimalizáciu a tento krok sa postupne znižuje tak, aby pri dosiahnutí konca behu algoritmu bol rovný 0. Tým sa dosiahne prehľadanie väčšieho priestoru a ku koncu jemnejšia optimalizácia riešenia. Nastalo pár prípadov, kedy bolo riešenie optimalizované pomocou premennej veľkosti okolia (s konštantnou zmenou kroku) horšie ako riešenie s konštantou veľkosťou okolia, avšak v priemere sa dosiahlo zlepšenia o 6.5% pri testovaní logistickej rovnica. Z grafu priebehu optimalizácie pomocou SA, zobrazeného na obrázku 6.5, je tiež zrejmé, že premenný krok zmeny parametrov našiel presnejšie riešenie rýchlejšie ako konštantný.

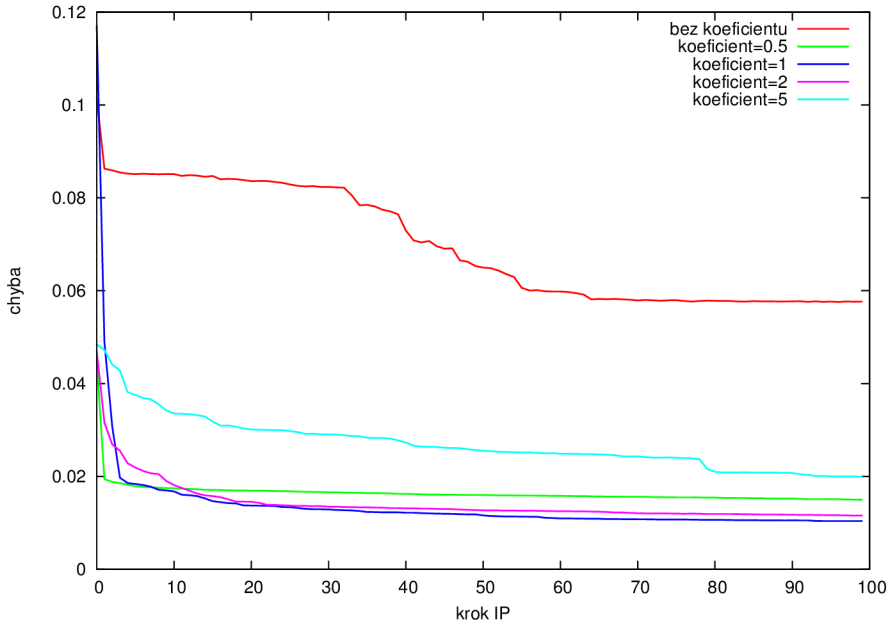


Obr. 6.5: Vplyv výberu suseda z konštantého okolia a z postupne zmenšujúceho sa okolia

Vylepšenie PSO

Štruktúrna optimalizácia sa spolieha na iteratívne volanie optimalizácie parametrov, v čom pre PSO vzniká problém v tom, ako zahrnúť optimalizované riešenie do procesu. V rámci jednoduchosti som najskôr zvolil len taký postup, kedy sa optimalizované riešenie určí za globálne najlepšie riešenie a častice reprezentujúce riešenie sa inicializujú náhodne v rozmedzí miním a maxím parametrov. Po prvotnom testovaní a úprave algoritmov tak, aby sa nemuseli zdržovať v malom, dopredu definovanom priestore, bolo potrebné zmeniť tento proces.
Namiesto úplne náhodných hodnôt sú častice počiatočne inicializované v určitom okolí predaného riešenia, ktoré už bolo optimalizované. Tým je dosiahnuté prehľadávanie väčšieho priestoru a stále je možné vymaniť sa z prípadného lokálneho minima najlepšieho jedinca, s ktorým je, na základe štruktúrnej optimalizácie, algoritmus PSO volaný každým krokom optimalizácie štruktúry.
Na obrázku 6.6 je znázornené porovnanie priebehu postupnej optimalizácie jedného jedinca pre náhodné generovanie častíc algoritmu PSO z upredom daného okolia a pre genrovanie z okolia optimalizovaného jedinca. Na obrázku je tiež vidieť, že veľkosť takéhoto okolia - daná

koeficientom, ktorý značí delenie počiatočných intervalov parametrov - je vhodné voliť v rozsahu $1 - 2$.



Obr. 6.6: Vplyv inicializácie častíc PSO podľa optimalizovaného riešenia

6.1.5 Testovanie vplyvu pomeru dát

Za účelom zlepšenia generalizácie neurónovej siete je okrem výstupu parametrickej optimalizácie do fitness funkcie zahrnutý výsledok testovania na testovacej sade dát. V tabuľke 7.1 sú zobrazené výsledné chyby najlepších jedincov po trénovaní modelu, chyby vyhodnotenia a nárast v týchto chybách. Hodnoty boli merané pre rovnaké parametre a konštantný krok, čiže pri použití menšieho pomeru bola časová naročnosť nižšia. Z tabuľky je zreteľné, že spolu s veľkosťou trénovacej sady rastie presnosť metódy, avšak pri príliš veľkom pomere dochádza k zlej generalizácii a vysokej chybe vyhodnotenia. Z tabuľky vyplýva, že najvýhodnejší pomer dát určených pre trénovani siete a dát určených pre optimalizáciu štruktúry je približne 0.6.

Pomer dát	0.2	0.4	0.6	0.8
RMSE trénovanie	0.0112	0.0062	0.00251	0.00262
RMSE evaluácia	0.0216	0.0105	0.00384	0.00474
Nárast v chybe	80.87%	69.35%	52.99%	80.92%

Tabuľka 6.4: Vplyv pomeru trénovacích a testovacích dát na chybu a jej nárast pri vyhodnotení

6.1.6 Predspracovanie

I keď by neurónové siete mali byť schopné predpovedať hodnoty z akýchkoľvek vstupných dát, experimenty ukazujú, že charakter vstupných dát je nezanedbateľným parametrom predikcie.

Normalizácia

Na presnosť predikcie modelu má podľa článku [14] vplyv aj interval, do ktorého sú vstupné dáta normalizované. V danom článku je spomenutá zvýšená presnosť predikcie po normalizácii vstupných dát do intervalu $[0.2, 0.8]$.
V tabuľke 6.5 sú zobrazené priemerné výsledky pre rovnaké parametre modelu, len so zmeneným intervalom normalizácie. Pre interval $[0.2, 0.8]$ je dosiahnutá presnosť o niečo vyššia.

Interval	[0,1]	[0.2,0.8]
RMSE trénovanie	0.01662	0.01487
RMSE evaluácia	0.01795	0.01494

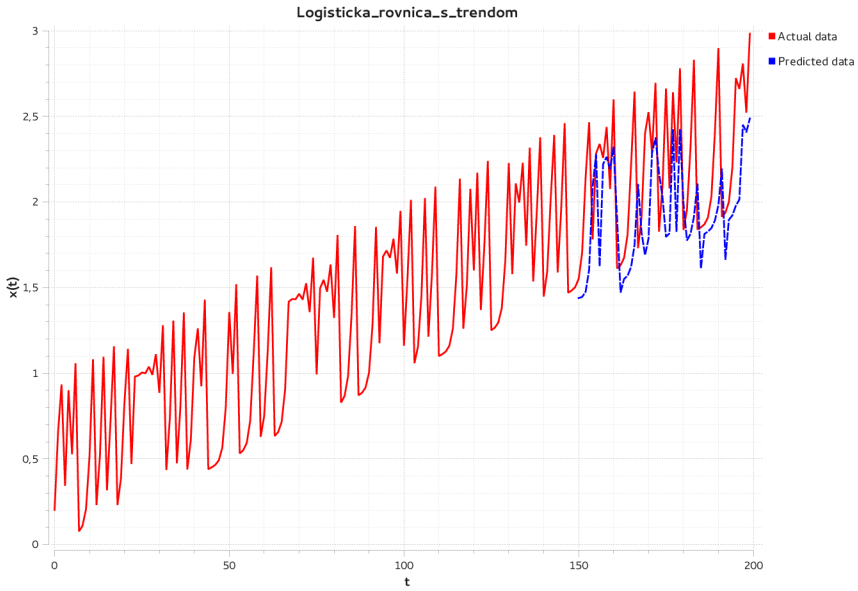
Tabuľka 6.5: Vplyv intervalu normalizácie na presnosť

Trend

Ceny akcií v sebe väčšinou obsahujú viditeľnú trendovú zložku, preto som pre potreby experimentov do logistickej rovnice zaviedol trend. Na obrázku 6.7 je zobrazený priebeh logistickej rovnice s trendom a predpovedané hodnoty modelom FNT. Z obrázku je jasne viditeľná zlá presnosť a neschopnosť modelu správne predpovedať rady s trendovou zložkou. Tento prípad je pravdepodobne spôsobený obmedzením intervalu výstupných hodnôt siete na základe aktivačnej funkcie.
Vyriešenie problému je možné dosiahnuť diferencovaním trendu spomenutým v časti 2.2.2. Takto predspracovaná logistická rovnica je zobrazená na obrázku 6.8, na ktorom je pozorovateľné viditeľné zlepšenie presnosti.

6.2 Predikcia časovej rady ceny akcií

V tejto časti sa zamerám na predpovedanie akcií indexu Dow Jones, ktorý zobrazuje akcie tridsiatich vybraných, verejne obchodovateľných spoločností, a je jedným z najznámejších identifikátorov amerického akciového trhu.
Ako vstupné dáta budem pre predikciu pomocou jedného radu používať dennú uzatváraciu cenu obchodovaných akcií za obdobie štyroch rokov. Model je natrénovaný pomocou parametrov uvedených v tabuľke 6.6.
Výsledok, zobrazený na obrázku 6.9, vyzerá na prvý pohľad veľmi uspokojivo. Odpovedá tomu aj veľkosť chyby RMSE, ktorá sa po prevedení dostatočného počtu krokov algoritmu ustáli na hodnote 0.023. Avšak bližšie preskúmanie ukazuje, že je za najpresnejšie predpovedanú hodnotu vzatá minulá hodnota z času $t-1$ a jednoducho predaná na výstup. Keďže pôvodne nebola obmedzená minimálna výška stromu, graf topológie evolvovanej štruktúry (obrázok 6.10) potvrdzuje túto skutočnosť - štruktúra bola optimalizovaná len do podoby vstupu z času $t-1$.

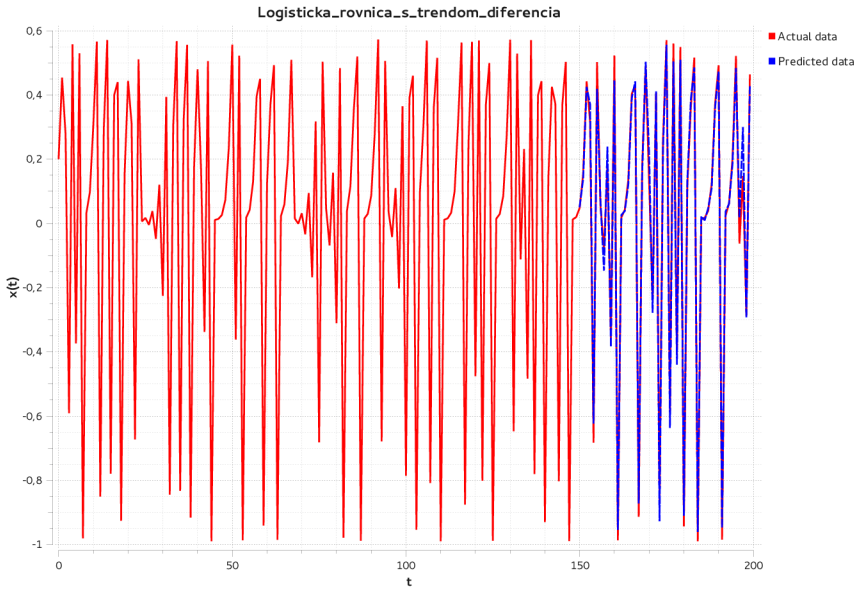


Obr. 6.7: Predikcia hodnôt logistickej rovnice s trendom

Parameter	Hodnota
Počet vstupov	6
Maximálna výška stromu	4
Maximálny počet potomkov	4
Pomer trénovacích a testovacích dát	0.65
Pravdepodobnosť klonovania	0.7
Pravdepodobnosť nahradenia	0.15
Pravdepodobnosť mutácie	0.4
Optimalizačný algoritmus	PSO, 10 častíc

Tabuľka 6.6: Nastavenie parametrov pri predikcii indexu Dow Jones

Zavedením minimálnej výšky stomu sa dosiahla mierna zmena charakteru predpovedaných hodnôt, ale predikcia stále inklinuje ku kopírovaniu predchádzajúcich hodnôt (obrázok 6.11). Tieto hodnoty boli predpovedané evolvovaným stromom zobrazeným na obrázku 6.12. Z jeho štruktúry je možné poznať, že sa algoritmus dopracováva k čo najmenšiemu počtu inštrukčných uzlov. Z toho dôvodu bola modifikovaná operácia mutácie tak, aby boli potlačené mutácie funkčných uzlov a aby sa väčšie zameranie kládlo na alteráciu vstupného vektora. Aj keď táto zmena priniesla zlepšenie charakteru štruktúrnej optimalizácie, problém pri predpovedaní časového radu indexu Dow Jones pretrval.
Príčinou vzniku tohto problému je pravdepodobne neschopnosť modelu natrénovať sieť na vyššiu presnosť, ako je tá, ktorú ponúka kopírovanie minulej hodnoty. Spôsobené to môže byť napr. normalizáciou vstupných dát na príliš malé hodnoty z dôvodu veľkého rozsahu dát. Jednoduchou úpravou intervalu normalizácie ale problém nezmizol.
Tiež by príčina mohla tkvieť v nedostatočnej veľkosti stromu. Táto možnosť by mohla byť

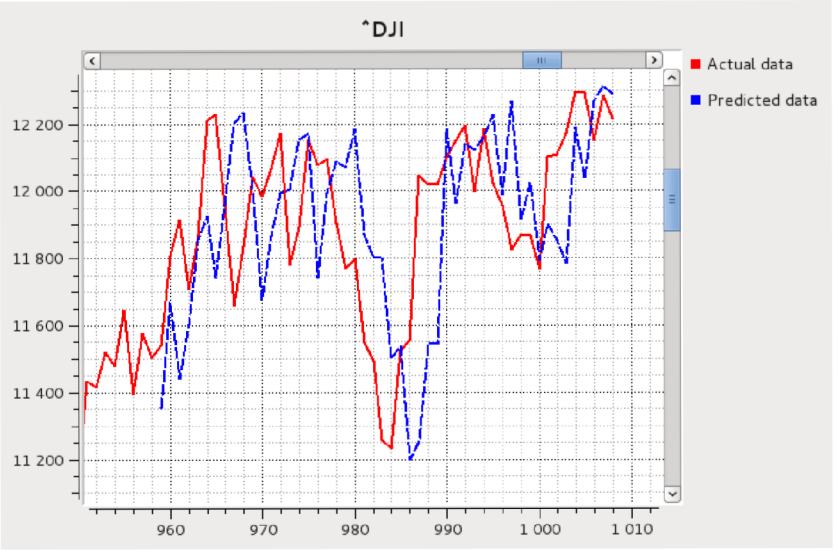


Obr. 6.8: Predikcia hodnôt logistickej rovnice s trendom po diferenciácii

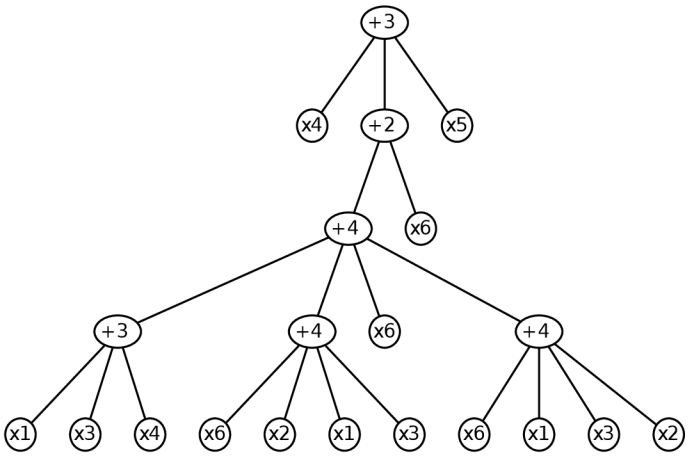
asi najreálnejšia, avšak veľké výpočetné nároky na optimalizáciu veľkého stromu by mohli spôsobiť neaktuálnosť predikovanej hodnoty. Za účelom čo najširšieho prehľadania priestoru riešení boli vykonané drobné upravy, ale charakter predpovede sa nezmenil.
Ďalší dôvod nesprávnej funkcionality môže spôsobovať charakter vstupných dát. Možným spôsobom odstránenia tejto chyby je predikcia diferencie indexu Dow Jones, čiže predpovedanie bez trendu, ktorý by mohol daný problém spôsobovať (podobne ako pri logistickej rovnici). Na obrázku 6.13 je zobrazený priebeh a predpovedané hodnoty indexu Dow Jones po diferenciácii. Z obrázku je ale jasne vidieť, že rozsahovo sú predpovedané hodnoty vo veľmi malom intervale v porovnaní s historickými dátami. Testy ďalej ukázali, že riešenie predikcie diferencií konverguje len veľmi pomaly a, čo je jasne viditeľné z grafu, že je veľmi nepresné (RMSE 0.0906).
Nepohla ani úprava vstupných hodnôt dát nahradením krajných hodnôt minimálnou a maximálnou povolenou hodnotou (priebeh takej predikcie je zobrazený na obrázku 6.14). Ten sa rozsahovo už blíži k správnym hodnotám, avšak stále dosahuje vysokú nepresnosť (RMSE 0.089%) a pomalú konvergenciu.

Obr. 6.9: Predikcia hodnôt indexu Dow Jones

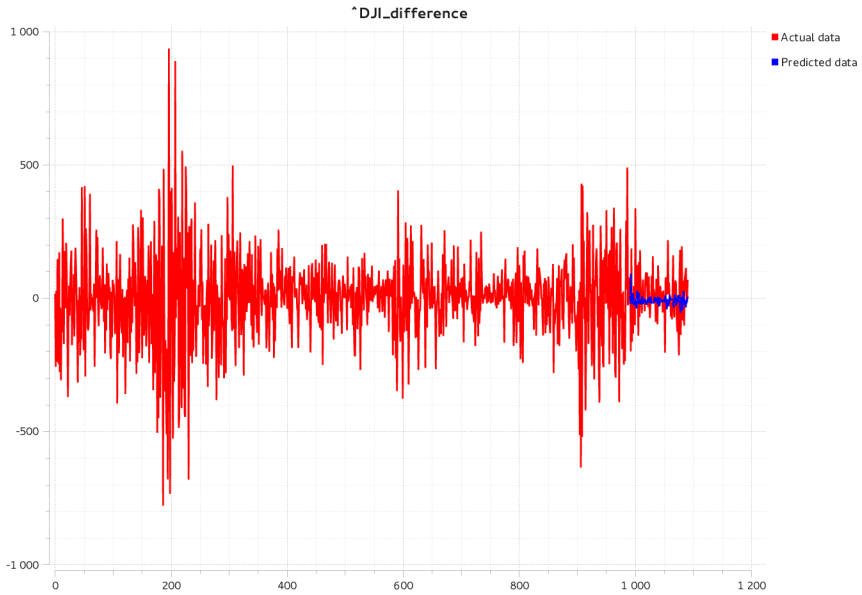
Obr. 6.10: Výsledný strom predikujúci vývoj indexu Dow Jones



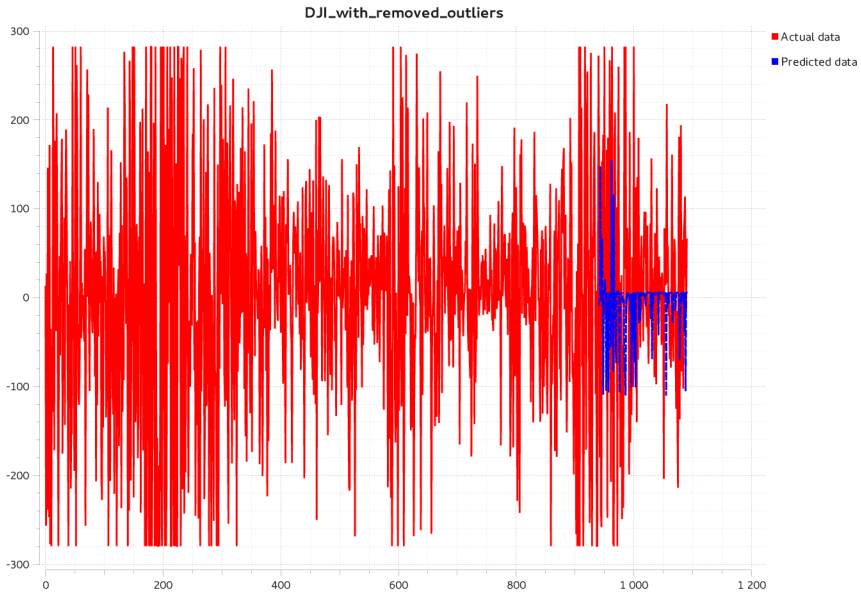
Obr. 6.11: Priblížený predpovedaný a skutočný priebeh indexu Dow Jones



Obr. 6.12: Výsledný strom predikujúci vývoj indexu Dow Jones po zavedení minimálnej výšky



Obr. 6.13: Priebeh a predikcia indexu Dow Jones po diferenciácii



Obr. 6.14: Priebeh a predikcia indexu Dow Jones po diferenciácii a po odstránení okrajových hodnôt

Kapitola 7

Predikcia viacerých časových radov

Keďže ani po prevedení testov s rôzne veľkým krokom, rôznou veľkosťou historických dát a s variáciou vstupných hodnôt sa mi nepodarilo zlepšiť presnosť predikcie časového radu vývoja ceny akcií a optimalizovať model FNT tak, aby neinklinoval ku kopírovaniu historických hodnôt, rozhodol som sa otestovať možnosť presnejšej predikcie pomocou viacerých časových radov aj pri takomto charaktere predpovedaných hodnôt.
Najskôr rozoberiem vstupné dáta, pomocou ktorých budem predpovedať vývoj cien akcií, ďalej otestujem presnosť indexu Dow Jones pri vstupných hodnotách viacerých časových radov za použitia modelu FNT so vstupmi z rôznych radov. Následne otestujem schopnosť modelu MIMO-FNT predikovať viaceré rady súčasne

7.0.1 Vstupné dáta

Predikcia časových radov cien akcií firiem na burze môže byť prevádzaná dvoma odlišnými spôsobmi - charakterizovanými odlišnými zdrojmi dát - jedná sa o *fundamentálnu* a *technickú* analýzu.
Fundamentálna analýza predpokladá, že firmy, ktorým sa vo svojom obore darí, či už vysokými ziskami, dobrým manažmentom, úspešným výskumom alebo kvôli iným faktorom, budú mať dobrú hodnotu akcií na burze. Tým je potrebné, aby fundamentálna analýza sledovala rôzne aspekty spoločnosti a ekonomickej situácie.
Technická analýza naopak predpokladá, že budúca hodnota akcie je obsiahnutá v jej historických hodnotách. Niektoré druhy technickej analýzy tiež berú pre potreby predikcie budúcej hodnoty do úvahy objem akcií - teda ako často je daná komodita v určitý časový interval obchodovaná. [8]
V práci budem využívať *technickú* analýzu. Z portálu *finance.yahoo.com* sú pre každý pracovný deň poskytnuté historické ceny akcií, a to konkrétne *otváracia*, *najvyššia*, *najnižšia*, *uzatváracia* a *uzatváracia* cena upravená o dividendy. V historických dátach sa tiež nachádza objem obchodovaných akcií v daný deň.

7.1 Predpovedanie jedného časového radu na základe viacerých

Keďže som v predchádzajúcej kapitole predpovedal index Dow Jones, pre porovnanie presnosti predpovede pomocou viacerých radov s predpoveďou pomocou jedného radu pri tomto indexe zostanem. Najskôr som sa pokúsil predpovedať cenu jeho uzatváracích cien pomocou

najvyšej, najnižšej a uzatváracej ceny akcií za minulé dni a následne som k trénovacej sade pridal objem obchodovaných akcií. Otváracie ceny nebudem uvažovať, pretože obsahujú rovnaké hodnoty ako uzatváracie ceny, len o jeden deň oneskorené.

7.1.1 Predpovedanie iba pomocou cien akcií

Priemerná chyba RMSE $(0,0225)$ bola pri predikcii pomocou viacerých časových radov jedného indexu nižšia o $0,005\%$ ako pri predpovedaní s len s jedným čsovým radom. Toto zlepšenie je nepatrné a neprinieslo riešenie problému kopírovania vstupných hodnôt, ktorý je spomínaný v predchádzajúcej kapitole. Aj napriek veľmi mierne zvýšenej presnosti hodnotím predpoveď negatívne, pretože konvergencia metódy bola pomalšia a tým aj časovo náročnejšia.

7.1.2 Predpovedanie aj s objemom obchodovaných akcií

Pri pridaní časového radu objemu obchodovaných akcií sa presnosť výsledku, ani na úkor výpočetného času, nezmenila.
V tabuľke 7.1 je zobrazené porovnanie presnosti pri použití jedného radu, viacerých radov pozostávajúcich len z cien akcií a vplyv pridaného časového radu objemu dát.

Použitá metóda	Jeden rad	3 rady ceny akcií	Objem + 3 rady	Objem + 1 rad
RMSE trénovanie	0.023	0.0225	0.0226	0.0225
RMSE evaluácia	0.0756	0.0751	0.07487	0.07496
Čas [s]	323	584	712	460

Tabuľka 7.1: Porovnanie presnosti pri predpovedaní indexu Dow Jones za použitia viacerých radov

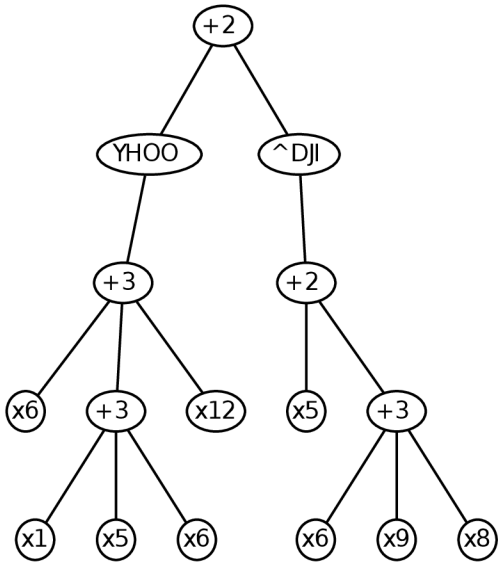
7.2 Predpovedanie viacerých časových radov súčasne

Predpovedanie viacerých radov súčasne je zabezpečené pomocou modelu MIMO-FNT.
Predpovedaný bol index Dow Jones spolu s akciami firmy Yahoo, ktorá je jednou zo zložiek tohto indexu. Parametre modelu pri testovaní sú zobrazené v tabuľke 7.2.

Parameter	Hodnota
Počet vstupov	6
Maximálna výška stromu	3
Maximálny počet potomkov	3
Pomer trénovacích a testovacích dát	0.65
Pravdepodobnosť klonovania	0.6
Pravdepodobnosť nahradenia	0.2
Pravdepodobnosť mutácie	0.4
Optimalizačný algoritmus	PSO, 10 častíc

Tabuľka 7.2: Nastavenie parametrov pri predikcii indexu Dow Jones spolu s akciami spoločnosti Yahoo

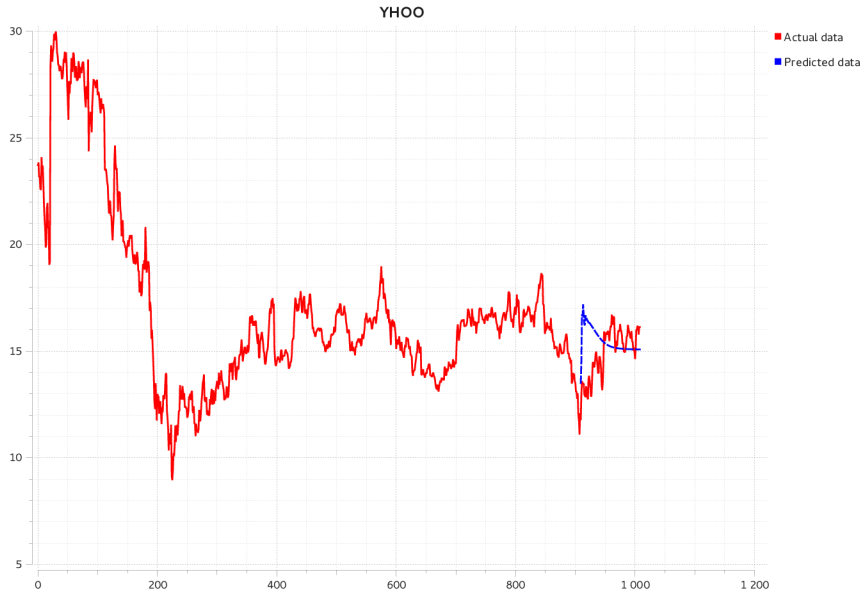
Presnosť predpovede časových radov akcií je ale stále rovnaká až horšia ako pri predikcii samostatného časového radu, pričom časová náročnosť metódy sa zvyšuje. Pravdepodobne je tento stav spôsobený tým, že predikcia viacerých časových radov je silne závislá na vhodnej voľbe vstupného vektoru. Tento vektor, ako už bolo spomenuté pri testovaní jednoduchého radu, ale metóda aj napriek svojej charakteristike neoptimalizuje.



Obr. 7.1: Evolvovaný MIMO strom

Na obrázku 7.1 je vyobrazený evolvovaný strom dávajúci výsledky zobrazené na obrázkoch 7.3 a 7.2 Tieto výsledky sú predikované dlhodobo. Z priebehov predpovede je jasne vidieť nepresnosť predikcie a negatívny vplyv vzájomného dlhodobej predikcie pri použití viacerých časových radov.

Obr. 7.2: Priebeh a predikcia indexu Dow Jones



Obr. 7.3: Priebeh a predikcia akcií spoločnosti Yahoo

Kapitola 8

Záver

V tejto diplomovej práci som sa zoznámil s problematikou predpovedania časových radov a s vlastnosťami časových radov udávajúcich ceny akcií. Keďže je burza dynamickým, chaotickým systémom, je tieto ceny veľmi obtiažne predpovedať. Preto bol na predikciu použitý model FNT, ktorý je založený na doprednej neurónovej sieti s algoritmicky optimalizeovanou topológiou. Dopredné neurónové siete, okrem toho, že sú pri predpovedaní najpoužívanejšie, doshujú pri vhodne zvolenej veľkosti rovnakú až vyššiu presnosť ako ostatné typy sietí.
Model FNT bol použitý ako základ pre implementáciu programu predpovedajúceho časové rady. Program umožňuje na základe vložených dát alebo na základe dát poskytnutých zo serveru predpovedať budúci vývoj časového radu. Program bol implementovaný ako klient-server aplikácia, aby bolo možné prevádzať predpovede na výkonnejšom stroji.
Implementovaný program bol otestovaný na jednoduchom časovom rade logistickej rovnice, ktorá, pri vhodne zvolených parametroch, dosahuje chaotické správanie. Na základe tejto predpovede boli vykonané optimalizácie modelu a predbežné odhady vplyvu parametrov na presnosť modelu. Tiež boli porovnané dva implementované algoritmy použité pre optimalizáciu parametrov. Podľa tohto porovnania dosahovala optimalizácia hejnom častíc lepšie výsledky ako simulované žíhanie, ktoré ale k nájdenému výsledku konvergovalo za kratšiu dobu.
Pri predpovedaní logistickej rovnice je celková presnosť predpovede veľmi dobrá.
Následne bola presnosť modelu testovaná na časovom rade priebehu cien akcií. V tomto prípade sa dostavili nečakané výsledky. Chyba predpovede totiž bola, ak vezmem do úvahy charakter priebehov radov akcií, viditeľne dobrá, ale pri bližšom preskúmaní bolo objavené nesprávne optimalizovanie stromu do takej podoby, aby model na výstup kopíroval hodnoty z minulého kroku. V snahe zlepšiť presnosť bol vyskýšaný spôsob predpovedania pomocou diferencovaného radu, ktorý pri testovaní jednoduchej rady dosahoval dobré výsledky, ale v tomto prípade bol nepoužiteľný.
Nesprávne chovanie modelu zostalo v aplikácii aj pri testovaní viacerých časových radov. Pri testovaní predikcie viacerých radov súčasne som neobjavil žiaden prípad, v ktorom by sa predpoveď výrazne zlepšila. Ale možné je, že by sa výsledky pri korektnom chovaní predikcie správali inak.
Pri ďalšom vývoji aplikácie by sa malo predovšetkým zamerať na spresnenie predikcie časových radov akcií, čiže na odstránenie spomínaného nesprávneho chovania predikcie, pretože v opačnom prípade by pri predpovdaní akcií program nepriniesol žiadnu výhodu. Ďalej by bolo vhodné implementovať paralelizmus aplikácie, keďže algoritmy použité pre optimalizáciu sú ľahko paralelizovateľné. Výrazne by sa tým zvýšila rýchlosť aplikácie a bolo by možné riadne otestovať schopnosť modelu predikovať veľké množstvo časových radov. Tiež

by mohlo byť zaujímavé otestovanie rôznych modifikácií flexibílneho neurónového stromu, akým by mohlo byť napr. pridanie spätných väzieb tak, aby strom reprezentoval Elmanovu alebo Jordanovu sieť.

Literatúra

[1] Arlt, J.; Arltová, M.; Rublíková, E.: *Analýza ekonomických časových řad s příklady [online]*. Praha, 2002 [cit. 2011-12-25], Dostupné z <http://nb.vse.cz/~arltova/vyuka/crsbir02.pdf>. Skriptá. VŠ Ekonomická v Praze.

[2] Brockwell, P. J.; Davis, R. A.: *Introduction to time series and forecasting*. New York: Springer, druhé vydání, 2002, 434 s., ISBN 03-879-5351-5.

[3] Burke, E.; Bykov, Y.; Newall, J.; aj.: A New Local Search Approach with Execution Time as an Input Parameter. Technická zpráva, School of Computer Science and Information Technology, University of Nottingham, 2002.

[4] Chen, C.: *Fuzzy Logic and Neural Network Handbook*. New York: McGraw-Hill, 1996, ISBN 0-07-011189-8.

[5] Chen, Y.; Abraham, A.: *Tree-structure based hybrid computational intelligence: theoretical foundations and applications*. Berlin: Springer, 2010, 206 s., ISBN 978-364-2047-381.

[6] Dhamija, A.; Bhalla, V.: Financial Time Series Forecasting: Comparison of Neural Networks and ARCH Models. *International Research Journal of Finance and Economics*, 2010.

[7] Dorffner, G.: Neural Networks for Time Series Processing [online]. *Neural Network World*, ročník 6, 1996 [cit. 2012-01-05]: s. 447–468, Dostupné z <http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15782-f06/slides/timeseries.pdf>.

[8] Gryc, W.: Neural Network Predictions of Stock Price Fluctuations. [cit. 2012-05-12], Dostupné z <i2r.org/nnstocks.pdf>.

[9] Karray, F. O.: *Soft computing and intelligent systems design: theory, tools and applications*. Essex: Pearson Education Limited, 2004, 560 s., ISBN 03-211-1617-8.

[10] Kléma, J.; Kout, J.; Vejmelka, M.: Predictive System for Multivariate Time Series [online]. [cit. 2011-12-28], Dostupné z <http://labe.felk.cvut.cz/~klema/publications/EMCSR/emcsr04.pdf>.

[11] Lembke, R.: Confidence Intervals for Forecasting [online]. [cit. 2012-01-07], Dostupné z <http://www.business.unr.edu/faculty/rtl/461/>.

[12] Lütkepohl, H.: *New Introduction to Multiple Time Series Analysis*. Berlin: Springer, 2005, 764 s., ISBN 35-404-0172-5.

[13] Mandic, D. P.: *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Chichester: Wiley, 2001, 285 s., ISBN 04-714-9517-4.

[14] Mańdziuk, J.; Mikołajczak, R.: Chaotic time series prediction with feed-forward and recurrent neural nets. *Control and Cybernetics*, 2002 [cit. 2012-05-01], Dostupné z <www.mini.pw.edu.pl/~mandziuk/PRACE/time_ser.pdf>.

[15] Musilek, P.; Lau, A.; Reformat, M.; aj.: Immune programming. *Elsevier*, 2005.

[16] Panahian, H.: Stock Market Index Forecasting by Neural Networks Models and Nonlinear Multiple Regression Modeling: Study of Iran's Capital Market. *American Journal of Scientific Research*, 2011.

[17] R.J. Frank, S. H., N. Davey: Time Series Prediction and Neural Networks. *Journal of Intelligent and Robotic Systems*, 2001 [cit. 2012-04-25], Dostupné z <www.smartquant.com/references/NeuralNetworks/neural30.pdf>.

[18] Samek, D.; Manas, D.: Artificial neural networks in artificial time series prediction benchmark. *International journal of mathematical models and methods in applied sciences*, [cit. 2012-05-10], Dostupné z <naun.org/journals/m3as/20-869.pdf>.

[19] Seger, J.: *Statistické metody v ekonomii*. Jinonice: H&H, 1993, 445 s., ISBN 80-857-8726-1.

[20] Sinčák, P.; Andrejková, G.: Neurónové siete Inžiniersky prístup (1. diel) [online]. [cit. 2012-01-08], Dostupné z <http://www.ai-cit.sk/source/publications/books/NS1/html/index.html>.

[21] Tsay, R. S.: *Analysis of financial time series*. Hoboken, N.J.: Wiley, druhé vydání, 2005, 605 s., ISBN 04-716-9074-0.

[22] Yu, L.: *Foreign-exchange-rate forecasting with artificial neural networks*. New York: Springer, první vydání, 2007, ISBN 978-0-387-71719-7.

[23] Zbořil, F. V.: Soft Computing. 2011/2012, Prednášky. FIT VUT v Brne.

Dodatok A

Obsah CD

- Zdrojové kódy tejto práce v LaTeX
- Obrázky a grafy uvedené v práci
- PDF verzia tejto práce
- Zdrojové kódy aplikácie
- Spustiteľná aplikácia
- Manuál