

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FUZZY LOGIKA A PREDIKCE VÝVOJE AKCIOVÉHO TRHU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN HAVIAR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FUZZY LOGIKA A PREDIKCE VÝVOJE AKCIOVÉHO TRHU

FUZZY LOGIC AND THE PREDICTION OF STOCK MARKET

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN HAVIAR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PATRIK PETŘÍK

BRNO 2011

Abstrakt

Bakalářská práce se zabývá využitím fuzzy logiky, fuzzy systémů a fuzzy neuronových sítí při predikci vývoje akciového trhu. Práce obsahuje návrh a popis implementace aplikace na předpověď vývoje akciového trhu.

Abstract

This thesis deals with fuzzy logic, fuzzy systems and fuzzy neural networks used for prediction of stock market. Thesis contains design and implementation description of application for stock market forecast.

Klíčová slova

Fuzzy logika, Fuzzy množiny, Fuzzy systémy, Fuzzy neuronové síte, Akciové trhy, Přibližné usuzování, Predikce

Keywords

Fuzzy logic, Fuzzy sets, Fuzzy systems, Fuzzy neural networks, Stock market, Fuzzy deduction, Prediction

Citace

Martin Haviar: Fuzzy logika
a predikce vývoje akciového trhu, bakalářská práce, Brno, FIT VUT v Brně, 2011

Fuzzy logika a predikce vývoje akciového trhu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Patrika Petříka.

.....
Martin Haviar
12. května 2011

Poděkování

Chcel by som sa poďakovať vedúcemu práce pánu Ing. Petříkovi za konzultácie a odborné vedenie práce. Ďakujem tiež mojej rodine a priateľom za podporu, ktorú mi poskytli.

© Martin Haviar, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Cieľ práce a metódy spracovania	4
2.1 Vymedzenie cieľa práce	4
2.2 Metódy spracovania	4
3 Teoretické východiská práce	5
3.1 Fuzzy logika	5
3.1.1 Fuzzy množiny	5
3.1.2 Funkcia príslušnosti	6
3.1.3 Približné usudzovanie	6
3.1.4 Predikcia pomocou fuzzy logiky	8
3.2 Fuzzy systémy	8
3.2.1 Fuzzifikácia	9
3.2.2 Defuzzifikácia	10
3.3 Fuzzy neurónové siete	10
3.3.1 Umelé neurónové siete	10
3.3.2 Neurónové siete a fuzzy logika	12
4 Aplikácia na predikciu akciového trhu	13
4.1 Zdrojové dáta	13
4.1.1 NASDAQ-100	14
4.1.2 Inflácia	14
4.2 Návrh modelu	15
4.2.1 Fuzzifikácia	15
4.2.2 Fuzzy neurónová sieť	16
4.2.3 Defuzzifikácia	16
4.3 Technológie	16
4.4 Implementácia	19
4.4.1 Načítanie dát	19
4.4.2 Normalizácia dát	19
4.4.3 Lineárna regresia	21
4.4.4 Fuzzifikácia	21
4.4.5 Fyzzyneurónová sieť a jej tréovanie	22
4.4.6 Defuzzifikácia a výstup aplikácie	24
4.4.7 Určenie chyby	24

5 Prípadové štúdie	25
5.1 Vymedzenie problému	25
5.2 Prípadová štúdia 1 - Akcie International Business Machines Corporation (IBM)	26
5.3 Prípadová štúdia 2 - Akcie Google Inc.	29
5.4 Prípadová štúdia 3 - Akcie Honeywell International Inc.	32
5.5 Zhodnotenie prípadových štúdií	35
5.5.1 Možnosti rozšírenia	35
6 Záver	36

Kapitola 1

Úvod

Už od čias barterových obchodov sa obchodníci snažili odhadnúť vývoj cien rôznych komodít v snahe získať obchodnú výhodu. Výrazný pokrok v tejto oblasti nastal v 18. storočí s rozvojom teórie pravdepodobnosti. Éra počítačov ďalej umožnila urýchliť výpočty spojené s predikciou a taktiež použiť nové štruktúry a modely, ktoré do tej doby nemali obdoby. Žiadny z nich však nebol natoľko úspešný, aby splnil sen každého obchodníka. Vytvoriť systém, ktorý by dokázal dokonale predikovať správanie trhu totiž nie je možné.

Dôvodom je fakt, že trh sa správa ako živý organizmus a aj jeho zmeny sú živými organizmami podnietené. Dôležitú úlohu v tomto procese hrá ľudská psychika, ktorá ovplyvňuje rozhodnutia obchodníkov nakupovať a predávať. Ponúka sa preto otázka, či má vôbec zmysel navrhovať systémy schopné predikcie trhu, ktorý je tak silne nedeterministický.

V súčasnosti však existujú techniky, ktoré to dokážu v takej miere, že sú pri analýze trhu nápomocné a majú z dlhodobého hľadiska uspokojivú úspešnosť. Medzi tieto techniky patrí aj fuzzy logika, ktorej sa venuje táto práca.

Kapitola 2

Ciel' práce a metódy spracovania

Úlohou tejto kapitoly je vymedziť rámec tejto bakalárskej práce a objasniť čitateľovi postup, ktorým sa bude práca riadiť.

2.1 Vymedzenie cieľa práce

Cieľom práce je zoznámiť čitateľa so základmi fuzzy logiky a fuzzy systémov v spojení s predikciou akciového trhu. Budú mu predstavené základné postupy a modely, ktoré využívajú fuzzy logiku na predikciu vývoja hodnôt akcií. Ďalej sa čitateľ dozvie o súčasných trendoch v predikcii pomocou fuzzy neurónových sietí.

Cieľom praktickej časti je poskytnúť čitateľovi návod na aplikovanie teoretických poznatkov a postupov do formy počítačovej aplikácie. Tá na základe historických hodnôt akcie a pomocných ukazovateľov predikuje (predvída) ďalší vývoj. V závere tejto časti je zhodnotená úspešnosť a vhodnosť tohoto postupu s odkazmi na možné rozšírenia.

Práca sa nezaobrá podrobnejšími princípmi fungovania akciového trhu a investovaním, nakoľko to rozsah práce to nedovoľuje. Pre záujemcov odporúčam [7].

2.2 Metódy spracovania

Teoretická časť

V teoretickej časti práce sú obsiahnuté teoretické základy fuzzy logiky, fuzzy systémov a fuzzy neurónových sietí. Tieto poznatky sú dôležité pre porozumenie praktickej časti, ktorá sa na kvôli prehľadnosti spätne odkazuje na teóriu.

Praktická časť

Obsahuje ucelený popis procesu tvorby aplikácie na predikciu od návrhu, cez implementáciu až po prípadové štúdie, skúmajúce výkonnosť aplikácie. Prípadové štúdie majú jednotnú štruktúru, po stručnom popise spoločnosti nasleduje:

- určenie optimálnych parametrov aplikácie,
- predikcia a
- záver, v ktorom sú zhodnotenú dosiahnuté výsledky.

Po prípadových štúdiách nasleduje ich zovšeobecnené zhodnotenie.

Kapitola 3

Teoretické východiská práce

Táto kapitola obsahuje základné teoretické východiská z oblasti fuzzy logiky a fuzzy neuronových sietí, potrebné pre porozumenie praktickej časti práce. Pre podrobnejšie štúdium danej oblasti budú na konci kapitol uvedené referencie.

3.1 Fuzzy logika

Klasická teória množín (angl. *crisp sets* - ostré množiny) priniesla matematike formalizmus, vďaka ktorému možno jednotným prístupom riešiť veľa typov úloh. Existujú však aj situácie, ktoré tento model nedokáže pokryť. Príkladom môže byť paradox, ktorým sa zaoberali grécki filozofi: [4]

Je zvykom veľké množstvo piesku označovať slovom „hromada“. Keď z hromady odoberieme jedno zrnko, je zrejmé, že neprestane byť hromadou. Ak v tom však budeme pokračovať dosť dlho, nezostane nič. To už však nie je hromada. Nevieme však určiť, v ktorom kroku hromada prestala byť hromadou pri odobratí jediného zrnka. Z toho vyplýva, že vlasnosť „býť hromadou“ (definovaná pre akékoľvek množstvo piesku) nemôže byť klasická, dvojhodnotová a teda ju nemôžeme vyjadriť jednou z možností je alebo nie je.

Ukazuje sa, že ľudia zvyknú do svojich úvah a vyjadrení zahrňovať aj vlasnosti, ktoré sú vágne, nepresné a bez ostrých hraníc. Napr. vyjadrenie, „v miestnosti je teplo“ obsahuje vágnu informáciu. Teplo môže znamenať 20 ale pre niekoho iného aj 30°C. Je však zrejmé, že nepôjde o hodnoty pod nulou. Práve tento druh informácie sa snažia popisovať fuzzy množiny s ktorými následne narába fuzzy logika.

3.1.1 Fuzzy množiny

Teóriu fuzzy množín a fuzzy logiky vytvoril v roku 1965 L. A. Zadeh [17]. Pri klasickej teórii množín, tzv. ostré množiny (*crisp sets*), sa na rozdiel od fuzzy množín hovorí, či daný prvok do množiny patrí, alebo nie. Fuzzy množiny zosširujú ostré množiny v tom zmysle, že prvok do nich môže partiť do istej miery. Táto miera, daná funkciou príslušnosti môže nadobúdať hodnoty z reálneho intervalu $\langle 0, 1 \rangle$. Keďže však ide o ohraničenie neurčitosti, na vyjadrenie tohto intervalu sa používajú tzv. *konfidenčné intervaly*, značené hranatými zátvorkami $[0, 1]$. [3](s. 106)

3.1.2 Funkcia príslušnosti

V teórii fuzzy množín je funkcia príslušnosti (*membership function*) μ_F takou funkciou, ktorá mapuje celé univerzum na interval $[0, 1]$, teda

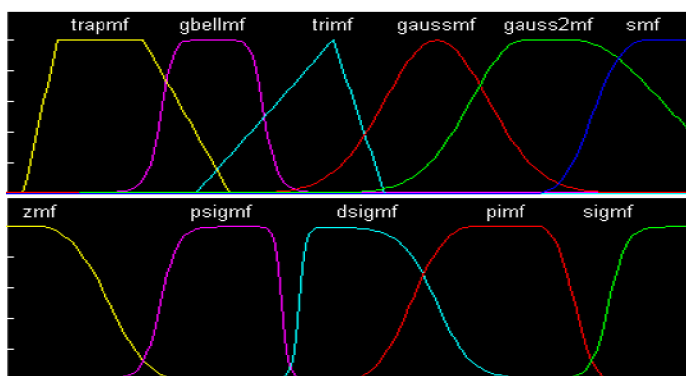
$$\mu_F : U \rightarrow [0, 1]. \quad (3.1)$$

a určuje tak mieru, do akej jednotlivé prvky univerza patria do fuzzy množiny.[2]

Fuzzy množina je potom jednoznačne určená prvkom $u \in U$ a jemu odpovedajúcej hodnote príslušnosti $\mu_F(u)$, resp. množinou týchto dvojíc $(u, \mu_F(u))$

$$F = \{(u, \mu_F(u)) / u \in U\}. \quad (3.2)$$

Rôzne tvary funkcií príslušnosti ukazuje obr. 3.1.



Obrázek 3.1: Rôzne tvary funkcií príslušnosti

3.1.3 Približné usudzovanie

Základom približného usudzovania je použitie *fuzzy výrokov* pri *fuzzy implikácii*. Fuzzy výrok je výrok, ktorý obsahuje fuzzy hodnoty. Príkladom môže byť veta „auto ide rýchlo“. Slovo „rýchlo“ predstavuje tzv. *lingvistickú premennú*, ktorá môže byť vyjadrená fuzzy číslom. Fuzzy implikácia tohoto výroku môže mať tvar napr.:

if (auto ide rýchlo) then (mierne príbrzdí).

Výrok sa skladá z predpokladov (*antecedentov*), ktoré môžu byť od seba oddelené spojками (and, or) a výsledku (*konsekventu*).

V dvojhodnotovej logike možno implikáciu vyjadriť ako

$$p \rightarrow q = (\text{not } p) \text{ or } q = (p \text{ and } q) \text{ or } (\text{not } p). \quad (3.3)$$

V prípade fuzzy výrokov p a q , môžeme miesto operátora *and* využiť tzv. *t-normu* a miesto *or* operátora *s-normu*¹. Je zrejmé, že takto môže vzniknúť nekonečne veľa fuzzy implikácií podľa toho, ktorú *t-normu* a *s-normu* zvolíme. V tabuľke 3.1 je prehľad najpoužívanejších z nich.[2]

¹t-norma a s-norma sú binárne operácie, ktoré spĺňajú tieto vlastnosti: komutatívnosť, asociatívnosť, monotónnosť a hraničnú podmienku

Názov implikácie	Použitý vzťah	Funkcia príslušnosti
Zadeh	$p \rightarrow q = (p \text{ and } q) \text{ or } (\text{not } p)$	$\max(\min(x,y), 1-x)$
Lukasiewicz	$p \rightarrow q = (\text{not } p) \text{ or } q$	$\min(1, 1-x+y)$
Kleene-Dienes	$p \rightarrow q = (\text{not } p) \text{ or } q$	$\max(1-x, y)$

Tabulka 3.1: Fuzzy implikácie

Inferencia

Inferencia je proces spojenia vstupov fuzzy systému s jeho pravidlami za účelom približného usudzovania. Inferencia obsahuje 4 fázy:

1. Fuzzifikácia vstupných hodnôt - vstupné hodnoty sa pomocou funkcií príslušnosti prevedú na hodnoty príslušnosti pre jednotlivé fuzzy množiny.
2. Vyhodnotenie pravidiel - pre aktivované pravidlá sa vypočíta výsledok implikácie.
3. Agregácia výstupov aktivovaných pravidiel - zjednotenie výstupov pravidiel do jednej fuzzy množiny.
4. Defuzzifikácia výstupu - prevod fuzzy hodnoty na ostrú hodnotu.

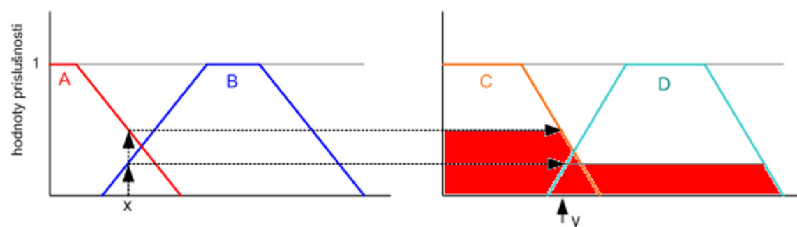
Najčastejšie používané metódy inferencie sú *Mamdani* a *Sugeno*. Uvažujme pravidlá a hodnoty

if (*X je A*) **then** (*Y je C*),
if (*X je B*) **then** (*Y je D*),

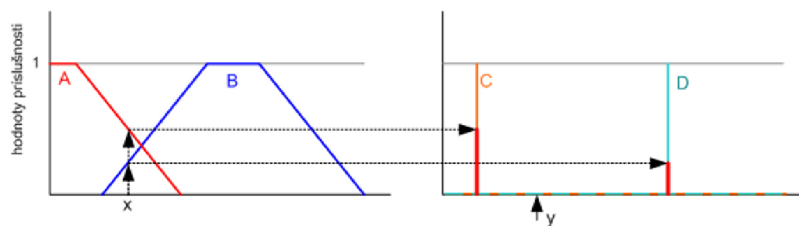
X je A (0,5),
X je B (0,25).

Mamdaniho metóda využíva orezávanie výstupnej funkcie príslušnosti na hodnote príslušnosti vstupu. Defuzzifikovanie hodnoty potom spočíva v nájdení ťažiska plochy tvorenej fuzzy množinami (obr 3.2).

Pri Sugenovej metóde sú fuzzy množiny výstupu zjednodušené takým spôsobom, že majú hodnotu 1 v jedinom bode, všade inde majú hodnotu 0. Tým sa výpočet ťažiska zjednoduší na výpočet váženého priemeru niekoľkých hodnôt (obr. 3.3). [14]



Obrázek 3.2: Mamdaniho inferenčná metóda



Obrázek 3.3: Sugenova inferenčná metóda

3.1.4 Predikcia pomocou fuzzy logiky

V [1](s. 288) je popísaný model programu na odporúčanie vstupu do pozície na burze. Podkladom programu na vyhodnotenie odporúčania je fundamentálna a technická analýza.

Z technickej analýzy využíva trendy časovej rady - intradenný, denný a týždenný. Z fundamentálnej analýzy využíva ukazovatele ako napr. EPS, P/E a ROE². Tieto vstupy sú do programu zadávané ako fuzzifikované hodnoty v lingvistickej forme (nárast, stagnácia a pod.). Fuzzy systém aplikácie obsahuje expertom vytvorenú bázu pravidiel. Pravidlá majú tvar

if (intradenný trend mierne stúpa) **and** (denný trend stagnuje)
and (týždenný trend prudko rastie) **then** (nakupuj).

Inferenciou týchto pravidiel a vstupov program produkuje odporúčanie na vstup do pozície na burze.

3.2 Fuzzy systémy

Podľa [2] pod pojmom *fuzzy systém* rozumieme taký systém, ktorého premenné (jedna alebo niektoré z nich) nadobúdajú hodnôt (stavov), ktoré nie sú definované ostrými číslami (reálnymi číslami v matematickom zmysle slova), ale sú definované slovnými hodnotami, resp. fuzzy množinami.

Každá takáto jazyková premenná je definovaná na odpovedajúcom univerze, ktorým je veľmi často interval reálnych čísel. V tomto špeciálnom prípade sú fuzzy množiny vlastne fuzzy čísla.

Význam týchto systémov sa prejaví už len pri zohľadnení faktu, že každé meranie fyzikálnej veličiny je sprevádzané istou neurčitostou. Jej príčina pochádza z konečnej rozlišovacej schopnosti každého reálneho meriaceho prístroja. Táto nepresnosť však nemusí nutne predstavovať negatívnu vlastnosť. Jedným z dôvodov, prečo sa nadbytočné informácie vylučujú zo systémov úmyselne je redukcia zložitosti systému.

Alternatívny prístup je založený na myšlienke, že kľúčovým elementom v ľudskom myslení nie sú čísla, ale názvy (*labels*) fuzzy množín, tj. tried objektov, v ktorých sa príslušnosť do danej triedy mení postupne, nie skokom. Proces ľudského myslenia a jeho logika uvažovania pracuje skôr s týmito fuzzy množinami ako s klasickou dvojhodnotovou logikou.

Nech x_i , $i = 1, 2, \dots, n$ sú vstupné premenné, definované na univerzách X_i . Každé toto univerzum je pokryté systémom fuzzy množín (slovných hodnôt patričnej slovnej premennej) $L_{x_i} = \{A_i^1, A_i^2, \dots, A_i^{n_i}\}$. Nech je ďalej daný súbor pravidiel podľa vzťahu 3.4,

²EPS je čistý zisk na akcii, P/E je pomer tržnej ceny akcia a zisku na akcii a ROE je podiel zisku a vlastného kapitálu.

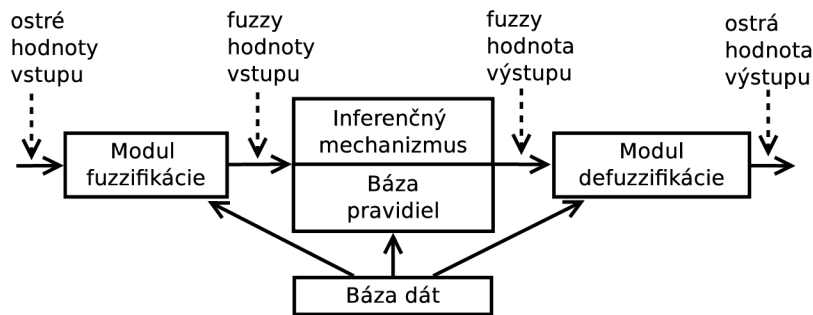
$$R^{(k)} : \text{if } (x_1 = A_1^{(k)}) \text{ and } (x_2 = A_2^{(k)}) \text{ and } \dots \text{ and } (x_n = A_n^{(k)}) \text{ then } (y = B^{(k)}); \quad (3.4)$$

$$k = 1, 2, \dots, r,$$

kde k je číslo pravidla a r je počet pravidiel. Vstupný vektor $\mathbf{x} = x_1, x_2, \dots, x_n$ je teda definovaný na kartézskom súčine $\mathbf{X} = X_1 \times X_2 \times \dots \times X_n$. Označme kartézsky súčin $A^{(k)} = A_1^{(k)} \times A_2^{(k)} \times \dots \times A_n^{(k)}$. Potom každé pravidlo predstavuje fuzzy implikáciu (tj. fuzzy reláciu) $R^{(k)} = A^{(k)} \times B^{(k)}$, ktorá je definovaná na $\mathbf{X} \times Y$ s funkciou príslušnosti $R^{(k)}(\mathbf{x}, y)$, kde $\mathbf{x} \in \mathbf{X}$, $y \in Y$.

Nech A je ľubovoľná fuzzy množina s funkciou príslušnosti $A(\mathbf{x})$ vstupujúca do tejto fuzzy relácie. Táto vstupná množina indukuje vo výstupnom priestore výstupnú fuzzy množinu B_0^k , pre ktorú platí $B_0^k = A \circ R^{(k)}$, kde symbol „ \circ “ značí kompozíciu. Takto formalizovaný systém sa nazýva *fuzzy systém*. [2]

V technických aplikáciách sú vstupné hodnoty najčastejšie vo forme ostrých čísel. Preto, aby sme mohli použiť fuzzy systém, je potrebné doplniť systém o modul *fuzzifikácie* a modul *defuzzifikácie*. Modul fuzzifikácie zaisťuje prevod ostrej hodnoty na fuzzy množinu a modul defuzzifikácie naopak prevod z fuzzy množiny na ostrú hodnotu. Štruktúru takéhoto fuzzy systému vidieť na obr. 3.4.



Obrázek 3.4: Štruktúra fuzzy systému

Báza pravidiel a báza dát je súhrnne označovaná ako *znalostná báza* fuzzy systému. Báza dát obsahuje údaje o fuzzy množinách (ich tvare a polohe v univerze) všetkých premenných systému. Báza pravidiel obsahuje všetky pravidlá fuzzy systému. Úlohou inferenčného mechanizmu je postupne vyhodnocovať všetky pravidlá a agregovať ich výsledky do jednej fuzzy množiny (viď. 3.1.3). Metódy fuzzifikácie a defuzzifikácie budú bližšie popísané v nasledujúcich kapitolách.

3.2.1 Fuzzifikácia

Úlohou fuzzifikácie je transformovať ostrú hodnotu x^* premennej x na fuzzy množinu. Najjednoduchšou formou fuzzifikácie je fuzzifikácia *singletonom*. [2] Výpočet prebieha podľa vzťahu 3.5.

$$\mu_{(x)} = \begin{cases} 1 & x = x^* \\ 0 & \text{inak} \end{cases} \quad (3.5)$$

Hodnota fuzzy množiny je teda počítaná ako funkčná hodnota funkcie príslušnosti v bode ostrej hodnoty x^* premennej x .

3.2.2 Defuzzifikácia

V tejto časti bude popísaná defuzzifikácia metódou priemerov stredov (*Center Average method - CAM*). Táto metóda uvažuje špičkové hodnoty (stredy) každej fuzzy množiny \bar{a}^k , ktoré sú zvážené výškou patričnej fuzzy množiny w^k a to pre každé pravidlo $k = 1, 2, \dots, r$. Ostrá hodnota je teda váženým priemerom špičkových hodnôt

$$y^* = \frac{\sum_{k=1}^r \bar{y}^k w^k}{\sum_{k=1}^r w^k}, \quad (3.6)$$

kde \bar{y}^k je špičková hodnota fuzzy množiny k -teho pravidla a w^k je jej výška daná výstupnou hodnotou pravidla.

O iných metódach, ako napr. metóde stredy plochy, metóde stredy súčtov, či metóde prvého maxima je možné viac sa dočítať v [2].

3.3 Fuzzy neurónové siete

Z hľadiska použiteľnosti má fuzzy logika v oblasti predikcie trhu značné obmedzenia. Tieto obmedzenia sa vzťahujú hlavne na bázu pravidiel (expertný systém), ktorý predstavuje jadro fuzzy systému. Aby bol takýto systém schopný skutočne predikovať vývoj akciového trhu, bolo by nevyhnutné, aby obsahoval obrovskú bázu znalostí a pravidiel, ktoré by museli byť ručne vytvorené odborníkom na základe dlhoročných skúseností. Ak by aj podobný systém vznikol, jeho použiteľnosť by bola časovo limitovaná z dôvodu premenlivosti trhu a pravidlá by bolo nutné často upravovať. Riešenie tohto problému poskytujú neurónové siete, ktoré majú, ako bude popísané ďalej, schopnosť učenia. Takto naučený systém dokáže fungovať ako univerzálny aproximátor, teda môže modelovať s požadovanou presnosťou ľubovoľnú spojitú funkciu, pokiaľ dostane správne vzory na učenie.[3] Ich kombináciou s fuzzy logikou vzniká model, ktorý je schopný učenia na základe vzorky časovej rady a zároveň je schopný pracovať s vágnosťou, ktorá je pre trhy charakteristická.

3.3.1 Umelé neurónové siete

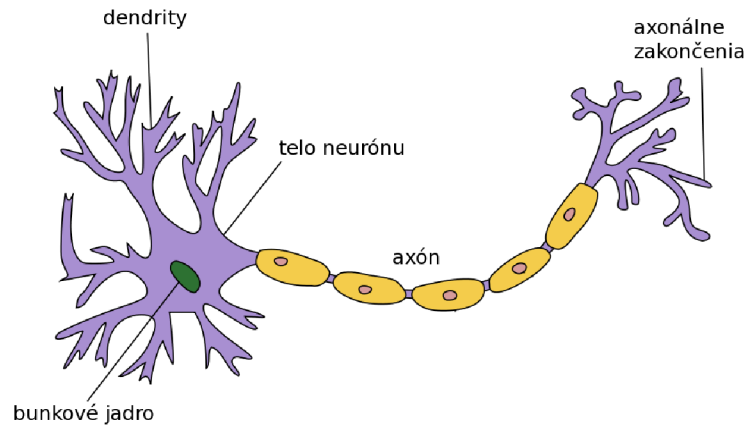
Napriek tomu, že umelé neurónové siete nie sú hlavnou nálpňou tejto práce, budú v tejto kapitole popísané ich základné princípy a ich súvislosť s fuzzy systémami. Východiskom umelých neurónových sietí je biologický neurón a jeho spracovanie signálu a informácií.

Biologický neurón

Biologický neurón (obr. 3.5) predstavuje základný stavebný prvok centrálného nervového systému človeka. Spracováva informácie a komunikuje s rôznymi časťami tela. Ľudský mozog obsahuje približne 10^{11} neurových buniek rôzneho typu.[3]

Z pohľadu spracovávania a uchovávaní informácií sú dôležité tieto tri časti neurónu:

- synaptické spojenia (dendrity) - miesta príjmu informácií a miesta uchovávaní znalostí od iných neurónov,
- telo bunky (soma) - prijíma informácie zo synaptických spojení a ďalej ich spracováva,
- axón - prenáša spracované informácie z tela bunky do ďalších neurónov.



Obrázek 3.5: Štruktúra biologického neurónu

Elektrický potenciál z dendritov je spracovávaný telom bunky. Ak tento potenciál presiahne určitú úroveň, bunka sa aktivuje a vyšle signál do axónu. Tam sa signál rozvetví a po určitej dobe dosiahne synaptického spojenia iných buniek.

Umelý neurón

Na základe funkčného modelu biologického neurónu navrhli v roku 1943 McCulloch a Pitts jednoduchý model umelého neurónu, ktorý dostal názov *perceptron*. V tomto neuróne sa každý vstupný signál x vynásobí zodpovedajúcim váhovým koeficientom w a následne sa sčíta. Ak táto suma, nazývaná tiež vnútorný potenciál neurónu, prekročí tzv. aktivačnú hodnotu θ , tak sa výstup z neurónu nastaví na logickú jednotku, v opačnom prípade na logickú nulu. Prekročenie tejto prahovej hodnoty má teda za následok vybudenie neurónu, čím sa na jeho výstup privedie hodnota :

$$y = S_{\beta} \left(\sum_{i=1}^n x_i \cdot w_i - \theta \right) \quad (3.7)$$

kde S_{β} je aktivačná funkcia neurónu, najčastejšie definovaná sigmoidou ako

$$S_{\beta}(a) = \frac{1}{1 + e^{-\beta a}}. \quad (3.8)$$

Perceptron má oproti skutočnému neurónu mnoho zjednodušení, napriek tomu je však najdôležitejším modelom neurónových sietí.

Učenie umelej neurónovej siete

Schopnosť učiť sa a klasifikovať vstupné vektory do tried je jednou z najvýznamnejších vlasností neurónových sietí. Počet tried do ktorých sú vstupné vektory triedené je rovný počtu výstupov neurónovej siete. Schopnosť učenia je daná možnosťou nastavovať váhy synaptických spojení na základe nejakého učiaceho algoritmu. V zásade môžu byť tieto algoritmy s tzv. *učiteľom*, alebo *bez učiteľa*.

Pri učení s učiteľom sú na vstup siete privádzané vstupné vektory, pričom vieme, do ktorej kategórie jednotlivé vektory patria. Odozva siete je porovnávaná so správnou klasifikáciou a na základe tohoto rozdielu (chyby klasifikácie) upravuje učiaci algoritmus jednotlivé

váhy spojení. V každom učiacom kroku je na vstup privedený jeden vstupný vektor a zmena váh je upravovaná tak, aby sa chyba zmenšila. Najznámejším algoritmom učenia s učiteľom je algoritmus spätného šírenia chyby (*Backpropagation*).

V prípade učenia bez učiteľa sieť nemá k dispozícii správne výstupy, s ktorými by mohli byť výstupy siete porovnávané. V tomto prípade sieť obsahuje pravidlo, pomocou ktorého môže sama korigovať chybu výstupu.

3.3.2 Neurónové siete a fuzzy logika

Bolo dokázané, že fuzzy systémy, ako aj neurónové siete, sú za určitých podmienok univerzálnymi aproximátormi³. Ponúka sa preto možnosť realizovať fuzzy systémy pomocou neurónovej siete, pričom si táto realizácia zachová schopnosť učenia (vďaka neurónovým sieťam), a tiež schopnosť reprezentovať znalosti (vďaka fuzzy systémom).

Keďže klasické neurónové siete môžu byť využité na realizáciu fuzzy systémov, nachádzame v literatúre pokusy prispôsobiť klasické neurónové siete potrebám fuzzy systémov. Základnou myšlienkou je použitie fuzzy čísel namiesto číselných hodnôt vyskytujúcich sa v štruktúre jedného neurónu, alebo celej siete. Takáto modifikácia vedie na nasledujúce možnosti kombinácií :

1. vstupy sú fuzzy čísla,
2. výstup je fuzzy číslo,
3. váhy sú fuzzy čísla,
4. vážené vstupy každého neurónu nie sú agregované operáciou „suma“, ale je použitá niektorá z t -noriem.

Tento neurón sa nazýva *fuzzy neurón* a siete z neho zostavené *fuzzy neurónové siete* (Fuzzy Neural Network, FNN).[2]

³Môžu modelovať s požadovanou presnosťou ľubovoľnú spojitú funkciu

Kapitola 4

Aplikácia na predikciu akciového trhu

V tejto časti je popísaný návrh a implementácia programu (skriptu), ktorý je jadrom tejto práce.

Táto demonštračná aplikácia má byť schopná na základe zadaných vstupov a parametrov určiť predikciu vývoja časovej rady. Časová rada bude reprezentovať hodnoty určitej akcie v priebehu času. Interval medzi jednotlivými hodnotami bude jeden deň, pričom budú brané do úvahy uzatváracie hodnoty pre daný deň.

Do úvahy budú pri určovaní výslednej predikcie okrem historických hodnôt akcie brané aj dva ďalšie ukazovatele. S ohľadom na to, že sa pri testovaní zameriavam na vývoj akcií popredných amerických firiem, ktoré pôsobia v oblasti informačných technológií (napríklad IBM, Intel, Google), zvolil som po preštudovaní [10, 8, 9] ako ukazovatele index NASDAQ-100 a mieru inflácie v Amerike. Tieto ukazovatele sú bližšie popísané v podkapitolách 4.1.1 a 4.1.2.

Výstupom aplikácie bude určenie ďalšieho vývoja, teda či nastane pokles, mierny pokles, stagnácia, mierny nárast alebo nárast hodnoty akcie počas určitej doby. Je zrejmé, že tieto veličiny nepredstavujú ostré množiny. Ide o fuzzy množiny, ktorých neostrá hranica je určená funkciou príslušnosti, ako bolo popísané v sekcii 3.1.2.

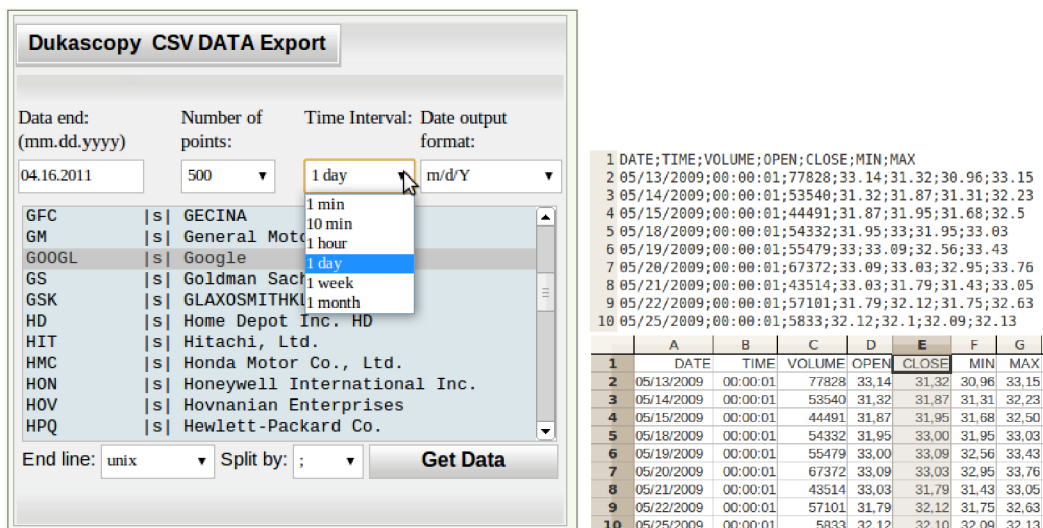
Skript musí byť tiež schopný určiť úspešnosť predikcie, resp. chybu. Úspešnosť bude testovaná na osobitných dátach ku ktorým pred predikciou nebude mať algoritmus prístup.

4.1 Zdrojové dáta

Kvalitné zdrojové dáta su dôležitým predpokladom testovania systému. Ako zdroj som použil internetovú aplikáciu švajčiarskej spoločnosti Dukascopy. Táto firma sa okrem iného zaoberá aj FOREX tradingom¹ a akciami a poskytuje kompletnú obchodovaciu platformu. Na získanie dát však postačil ich online systém pre export historických hodnôt akcií. Jeho použitím je možné stiahnuť historické dáta v podobe časovej rady. Jeho rozhranie ukazuje obr. 4.1.

Rozhranie umožňuje zvoliť si koncový dátum exportu, časový interval medzi hodnotami a počet bodov do minulosti. Pod nimi sa nachádza zoznam symbolov (s označením „s“), komodít (označené „f“ podľa anglického *futures*) a indexov („i“). Po nastavení ukončova-

¹Obchodovanie s cudzími menami



Obrázek 4.1: Dukascopy CSV DATA Export systém

nia riadkov a separátorov je možné stiahnuť tabuľku požadovaných dát vo formáte CSV², pričom ako oddelovače sú použité znaky z poľa *Split by*.

Takto vytvorený CSV súbor je možné načítať priamo tabuľkovým procesorom Open Office. Je len potrebné nastaviť správny separátor pri parsovaní a skontrolovať či sa dáta rozdelia do stĺpcov ako majú. Tabuľka obsahuje v poradí stĺpce udávajúce dátum a čas, počet obchodovaných kontraktov, otváracia cena na začiatku dňa, uzatváracia cena na konci dňa a denné minimum a maximum.

Zo získaných dát je už potom jednoduché extrahovať stĺpec *CLOSE* a uložiť ho do osobitného textového súboru. Rovnaký postup ako na hodnoty akcií som aplikoval aj na časovú radu indexov NASDAQ-100 a infláciu.

4.1.1 NASDAQ-100

NASDAQ-100 je burzový kapitálovo vážený index, ktorý pozostáva zo sto najväčších nefinančných spoločností americkej burzy cenných papierov NASDAQ³. Váha spoločností v tomto indexe je určená ich kapitálovým podielom s určitým horným obmedzením. Nezahrňuje finančné spoločnosti a naopak, zahrňuje pridružené spoločnosti mimo Spojených štátov. Oba tieto faktory odlišujú tento index od indexu *Dow Jones Industrial Average (DOW)* a nezahrnutie finančných spoločností ho odlišuje od indexu *S&P 500*. [10]

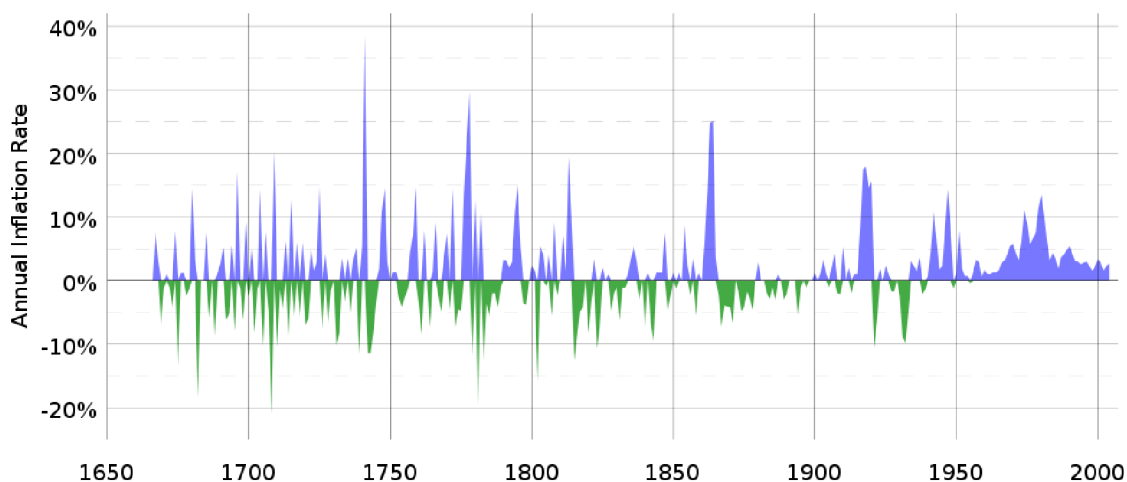
V zozname spoločností figurujú známe korporácie ako napríklad Apple, Cisco Systems, eBay, Google, Intel, Microsoft, Oracle alebo Yahoo!.

4.1.2 Inflácia

Inflácia je definovaná ako rast všeobecnej cenovej hladiny v čase, alebo aj ako zníženie kúpnej sily peňazí. Jej výpočet sa odvodzuje od indexu spotrebiteľských cien (anglicky *CPI - Consumer price index*).

²Comma-Separated Values[11]

³National Association of Securities Dealers Automated Quotations



Obrázek 4.2: Historická miera inflácie v Spojených štátoch amerických

Vysoká, alebo nepredvídateľná miera inflácie nepriaznivo vplýva na celkovú ekonomiku krajiny. Spôsobuje neefektívnosť trhu a spoločnosti majú problém vytvárať rezervy a dlhodobé plány. Vplyvom inflácie klesá produktivita a firmy sú nútené odobrať zdroje určené na produkty a služby, aby sa mohli sústrediť na zisky a straty spôsobené menovou infláciou.[6] Historický vývoj inflácie v Amerike ukazuje obr. 4.2

4.2 Návrh modelu

Model predikcie pochádzajúci z [1], popísaný v kapitole 3.1.4 má značné nedostatky. Do veľkej miery totiž závisí na tom, že expertý systém je tvorený špecialistom, alebo experimentálne. Úlohou expertného riadiaceho systému je popísať bázu pravidiel, na základe ktorých sú podľa príslušností vstupných parametrov do fuzzy množín vyvedené dôsledky, v našom prípade predikcia. Experimentálne vytvoriť takúto množinu pravidiel by však bolo časovo príliš náročné. Navyše, z dôvodu premenlivosti trhu by bolo potrebné pravidlá často upravovať. Ako výhodnejšie sa javí použiť model založený na fuzzy neurónovej sieti (3.3).

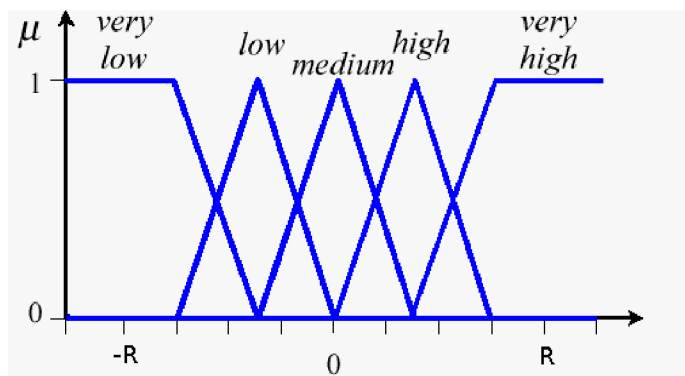
Návrh môjho modelu kombinuje prvky so spomínaných dvoch modelov, pričom využíva jednoduchosť Dostálovho fuzzy modelu a odstraňuje jeho slabé stránky použitím fuzzy neurónových sietí. Navyše je celý proces zadávania hodnôt a výpočtu parametrov automatizovaný, čím sa výrazne skraca práca potrebná na prípravu dát.

4.2.1 Fuzzifikácia

Vstupnými hodnotami aplikácie sú časové rady historických hodnôt určitej akcie a dva doplnujúce faktory. Pre každý bod časovej rady historických hodnôt akcie vypočítam trend vývoja hodnoty s určitým intervalom, napr. pre 5, 10 a 50 dní dozadu. Tieto trendy vyjadrujú charakter zmeny hodnoty v čase.

Na základe charakteru týchto trendov potom vytvorím fuzzy množiny, ktoré reprezentujú spätný vývoj hodnoty v danom čase. Vývoj rozdeľujem do piatich množín - pokles, mierny pokles, stagnácia, mierny nárast a nárast hodnoty akcie.

Fuzzifikujú sa aj dva pomocné indexy, nie však spôsobom ako hodnoty akcie. Fuzzy množiny týchto indexov reprezentujú vzťah k priemernej hodnote indexu - zvýšená, mierne



Obrázek 4.3: Tvar funkcií príslušnosti použitých fuzzy množín

zvýšená, normálna, mierne znížená a znížená hodnota indexu. Tvar funkcie príslušnosti týchto fuzzy množín ukazuje obr. 4.3. Tvary funkcií príslušnosti trendov sú rovnaké, akurát majú iný význam, ako už bolo popísané vyššie.

Z predchádzajúceho je zrejmé, že výsledkom tejto fázy je 25 fuzzy množín. Obsahujú informácie o krátkodobom, strednom a dlhodobom vývoji hodnoty akcie a stave dvoch pomocných indikátorov. K týmto údajom sa pridá ešte hodnota trendu predikcie v danom čase, tiež na určené obdobie. Tento postup je aplikovaný pre každý časový bod všetkých časových rád. Výsledkom je sada dát ktorá slúži ako vstup fuzzy neurónovej siete.

4.2.2 Fuzzy neurónová sieť

Fuzzy neurónová sieť má v tomto modeli úlohu automatizovaného expertného systému. Na základe predaných vstupných parametrov určuje pomocou algoritmu backpropagation hodnoty váh synaptických spojení. Ako predloha na učenie je tomuto algoritmu predaná predikcia. Sieť teda hľadá spojitosť medzi fuzzy množinami (trendami a indikátormi fundamentálnej analýzy) a výslednou predikciou. Naučená sieť funguje potom ako báza pravidiel fuzzy systému, podľa ktorého je na základe množiny vstupov možné určiť predikciu na vopred určený časový interval. Jej výstupom je opäť fuzzy množina, ktorú je potrebné na zistenie predikcie defuzzifikovať.

4.2.3 Defuzzifikácia

Defuzzifikácia fuzzy množín zabezpečí zistenie ostrej hodnoty predikcie. Tú je potom možné použiť napríklad na vykreslenie v grafe. Na defuzzifikáciu je použitá metóda stredu súčtu (COS, Center of Sum)[2]. Výsledok je však prezentovaný aj v lingvistickej forme a to v piatich úrovniach - pokles, mierny pokles, stagnácia, mierny nárast a nárast.

4.3 Technológie

Python

Aplikácia je po dohode s vedúcim bakalárskej práce vytvorená v jazyku Python. Python je moderný objektový programovací jazyk vytvorený Guido van Rossumom, používajú ho státisíce programátorov po celom svete a ich počet sa neustále zvyšuje. Jedným z dôvodov prečo je tomu tak, je, že umožňuje programátorom vytvárať aplikácie omnoho rýchlejšie, než

pomocou bežných programovacích jazykov ako napr. C++ alebo Java. Ide o platformovo nezávislý jazyk, ktorý možno použiť operačnom systéme Linux, Windows, OS/2 ale aj na počítačoch Macintosh. Poskytuje prístup k veľmi výkonnému a užívateľsky jednoduchému grafickému užívateľskému rozhraniu. Jeho vývoj je otvorený a interpret je zdarma. Veľkou výhodou je tiež dobrá dostupnosť knižníc širokého zamerania a celková podpora modularity. Práve použitie knižníc v značnej miere uľahčuje implementáciu často riešených úloh. Pri mojej implementácii mi výrazne pomohli najmä knižnice `SciPy`, `scilab`, `matplotlib` a `PyBrain`.

PyBrain[5]

PyBrain je modulárna knižnica na strojové učenie pre python. Jej cieľom je ponúkať flexibilné a výkonné algoritmy, ktoré sú ľahko použiteľné. Obsahuje množstvo preddefinovaných prostredí na testovanie a porovnávanie algoritmov. PyBrain je skratka pre *Python-Based Reinforcement Learning, Artificial Intelligence and Neural Network Library*. Koncept knižnice PyBrain je zapúzdrovať rôzne algoritmy na spracovanie dát v objekte nazvanom modul. Minimálny modul obsahuje doprednú implementáciu závislú na kolekcii nastavitelých parametrov, najčastejšie pomocou určitých algoritmov strojového učenia.

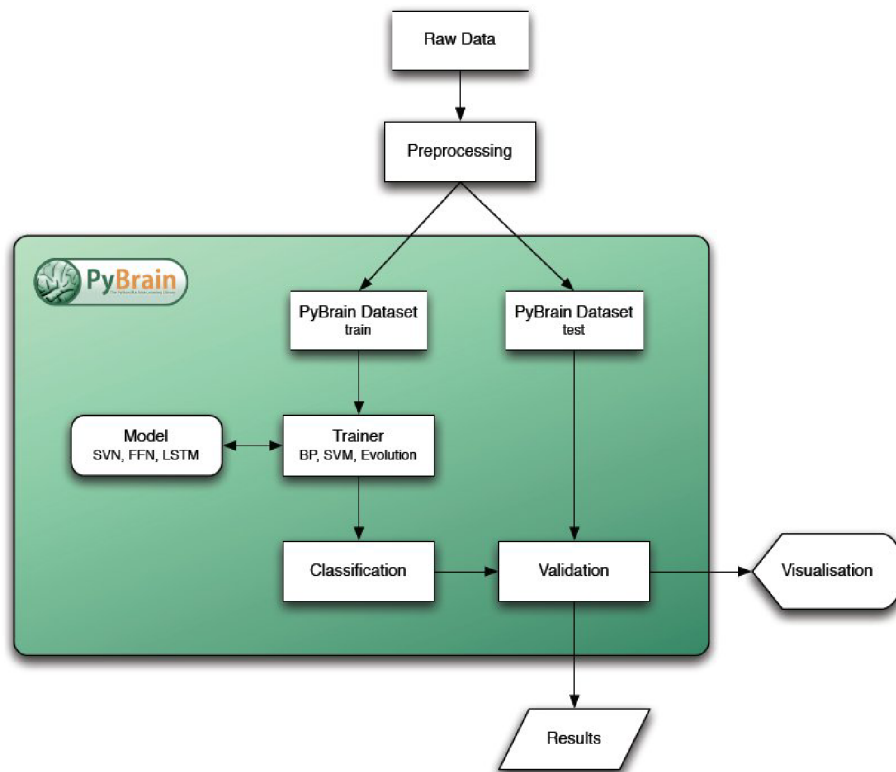
Moduly majú vstupný a výstupný zásobník. Okrem nich majú aj zásobníky obsahujúce chyby, ktoré sa používajú pri určovaní chyby pri algoritme backpropagation. Moduly sú zapuzdrované do objektov triedy *Network* a sú spájané pomocou objektov *Connection*. Môžu obsahovať niekoľko nastaviteľných parametrov, ako napr. váhy. Samotný objekt *Network* je taktiež modulom, preto je možné jednoducho zostaviť hierarchiu viacerých sietí. Na zostavenie často používaných sieťových architektúr existujú v knižnici PyBrain skratky, ktoré urýchľujú modelovanie. Systém však v princípe umožňuje zostaviť akékoľvek spojenie, pokiaľ modeluje orientovaný acyklický graf. Parametre siete (pozn.: váhy synaptických spojení) sú nastavované učiteľom (*Trainer*), ktorý používa zdroj dát (*Dataset*) na naučenie optimálnych parametrov na základe vzorov. Na experimenty s posilovaným učením (*reinforcement learning*) sa miesto zdroju dát používa simulačné prostredie s optimalizáciou. Vnútornú architektúru knižnice a jej vzťahy s okolitým prostredím ukazuje obr. 4.4.

SciPy[13]

SciPy je open-source softvér pre matematiku, vedecké a inžinierske výpočty. Ide o knižnicu pre python, ktorá je závislá na knižnici NumPy. Tá jej poskytuje prostriedky na rýchu prácu s n-dimenzionálnymi poľami. SciPy používa tieto polia a poskytuje mnoho užívateľsky pohodlných a efektívnych metód na numerickú integráciu a optimalizáciu. Spolu fungujú na všetkých populárnych operačných systémoch, rýchlo sa inštalujú a sú bezplatné. NumPy a SciPy sa ľahko používajú, no napriek tomu sú dostatočne výkonné, pričom sú na nich závislí poprední svetoví vedci a inžinieri.

matplotlib[12]

matplotlib je knižnica na vytváranie dvojrozmerných grafov z polí v jazyku Python. Napriek tomu, že má pôvod v emulovaní grafických príkazov programu MATLAB, je od matlabu nezávislá a môže byť použitá objektovo orientovaným spôsobom jazyka Python. Aj keď je napísaná takmer v čistom Pythone, vo veľkej miere používa modul NumPy a ďalšie rozšírenia kódu, ktoré poskytujú dobrú výkonnosť aj pre veľmi veľké polia. Základnou filozofiou matplotlib-u je poskytnúť užívateľovi metódy na vytvorenie jednoduchých grafov



Obrázek 4.4: Organizácia knižnice PyBrain

použitím len niekoľko málo príkazov, niekedy postačuje aj jeden. Napríklad na vytvorenie histogramu hodnôt nie sú potrebné inštancie objektov, volanie metód, či nastavovanie parametrov. Zdrojový kód modulu `matplotlib` je koncepčne rozdelený na tri časti:

- *rozhranie `pylab`* je skupina funkcií poskytovaných modulom `matplotlib.pylab`, ktorý umožňuje užívateľovi vytvárať grafy pomocou kódu podobnému MATLAB-u.
- *`matplotlib API`* je skupina tried, ktoré zabezpečujú výraznú abstrakciu a vytváranie a ovládanie grafov, obrázkov, textu, atď.
- *`Backend`*⁴ modulu obsahujú vykresľovacie zariadenia závislé na výstupnom zariadení a transformujú reprezentáciu dát z frontendu na výstupné zariadenie.

⁴V softvérovej architektúre sa rozlišujú viaceré vrstvy medzi hardvérom a užívateľom. Front-end je vrstva zabezpečujúca abstrakciu, teda zjednodušenie komponentov, čím vytvára užívateľsky pohodlné rozhranie. Backend je naopak najčastejšie databázou, resp. reprezentáciou dát na najnižšej úrovni.

4.4 Implementácia

V tejto časti budú približené metódy a postupy použité pri implementácii modelu z kapitoly 4.2.

Ako už bolo popísané vyššie vstupné dáta sú uložené v textových súboroch vo formáte CSV. Pre jednoduchšiu manipuláciu som rozdelil jednotlivé vstupy do osobitných súborov. V skripte je potom možné prepísaním parametrov určiť zdroj vstupných dát. Parametre príkazového riadku som kvôli jednoduchosti neimplementoval. Skript totiž obsahuje značné množstvo parametrov ktoré je možné nastaviť, preto by bolo ich určovanie cez príkazový riadok neprehľadné a neefektívne.

4.4.1 Načítanie dát

Python umožňuje svojim objektovým prístupom načítať zdrojové dáta veľmi jednoduchým spôsobom. Načítanie časovej rady do objektu zoznam je možné docieľiť jediným riadkom:

```
cols = open(file, 'r').read().split()
```

Funkcia `read()` načíta celý súbor a funkcia `split()` ho rozdelí na jednotlivé objekty zoznamu. Následne je potrebné pretypovanie hodnôt zoznamu na desatinné čísla

```
cols = [float(i) for i in cols]
```

Všetky takto vytvorené zoznamy uložíam do zapuzdrujúceho zoznamu, čím dostanem jediný objekt obsahujúci všetky zdrojové dáta.

4.4.2 Normalizácia dát

Podľa [3] tieto dáta ďalej normalizujem. Normalizácia je v štatistike proces, ktorý izoluje štatistickú chybu v meraných dátach. Výsledkom tejto transformácie majú dáta nulovú strednú hodnotu a ich rozptyl je rovný jednej. Jej výpočet (4.1) spočíva v odčítaní strednej hodnoty μ od hodnoty X , ktorú chceme normalizovať a výsledok sa podelí štandardnou odchýlkou σ .

$$\frac{X - \mu}{\sigma} \tag{4.1}$$

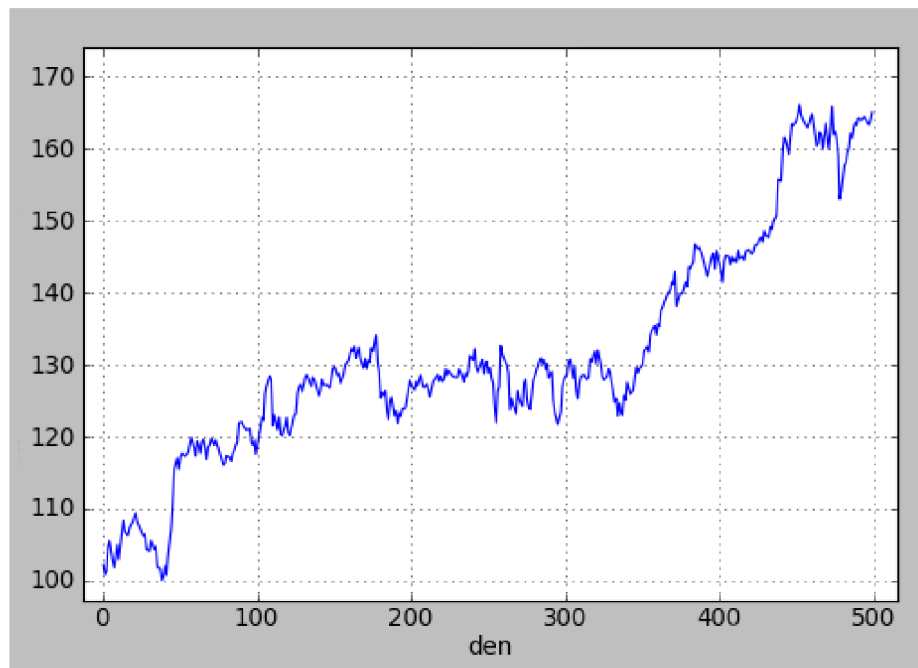
Pre výpočet štandardnej odchýlky platí vzťah 4.2.

$$\sigma = \sqrt{\frac{1}{N}[(x_1 - \mu)^2 + (x_2 - \mu)^2 + \dots + (x_N - \mu)^2]}, \tag{4.2}$$

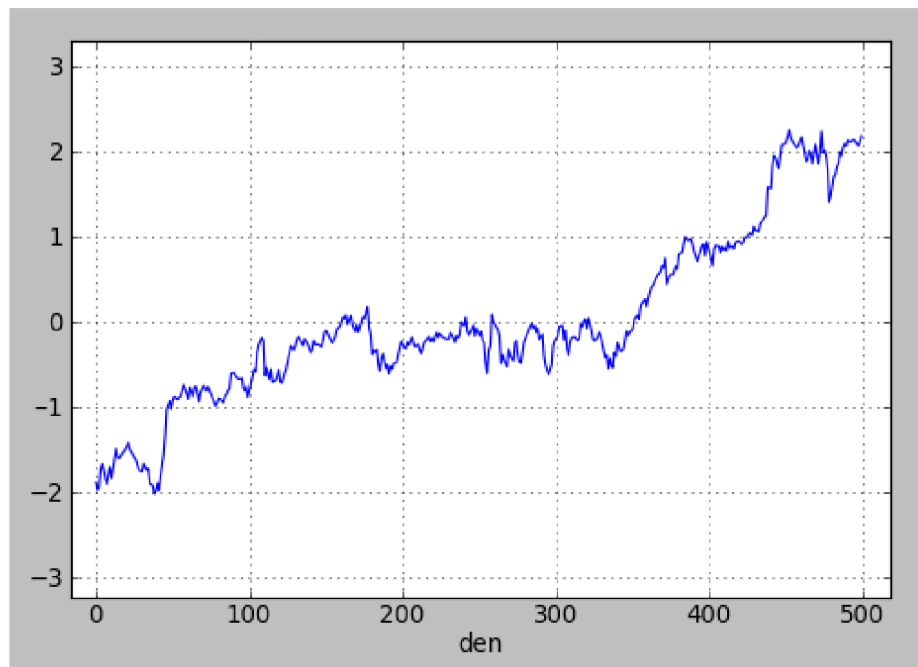
kde

$$\mu = \frac{1}{N}(x_1 + \dots + x_N)$$

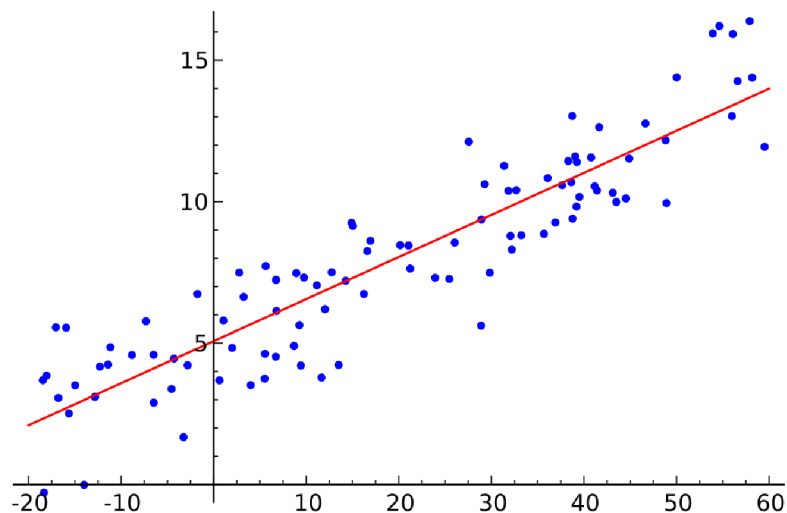
V pythone je možné štandardnú odchýlku poľa získať funkciou `std()`, ktorá je metódou objektu `array`. Tým sa výpočet značne zjednodušuje. Rozdiel medzi pôvodnými a normalizovanými dátami ukazuje obr. 4.5 a 4.6.



Obrázek 4.5: Denormalizované (pôvodné) dáta



Obrázek 4.6: Normalizované dáta



Obrázek 4.7: Lineárna regresia

4.4.3 Lineárna regresia

Na určenie trendu vývoja hodnoty akcie za určitý čas je použitá lineárna regresia. Lineárna regresia predstavuje aproximáciu hodôt polynómom prvého rádu (priamkou), ako vidno na obr. 4.7. Najčastejšie sa na určenie tohoto polynómu používa metóda najmenších štvorcov. Pri tejto metóde sa hľadá taká priamka, aby súčet druhých mocnín odchýlok jednotlivých bodov od priamky bol minimálny. Optimálne parametre sa ďalej určujú pomocou parciálnych derivácií.[16]

Ak uvažujeme funkčnú závislosť $f(x) = ax + b$, súčet štvorcov bude vyzeráť podľa vzťahu 4.3

$$S(a, b) = \sum_{i=1}^n [f(x_i) - y_i]^2 = \sum_{i=1}^n (ax_i + b - y_i)^2 \quad (4.3)$$

Na implementáciu lineárnej regresie možno využiť modul scipy, konkrétne jeho funkciu `scipy.stats.linregress`. Funkcia vracajúca smernicu priamky trendu na základe hodnôt v zozname `y` bude potom vyzeráť takto:

```
def regressionLine(y):
    x = [i for i in range(1, len(y)+1)]
    trend = stats.linregress(x, y) [0]
    return trend
```

4.4.4 Fuzzifikácia

Na fuzzifikáciu dát používam vlastnú univerzálnu funkciu `fuzzify()`, ktorá prijíma ako parametre

- *value* - ostrá hodnota určená na fuzzifikáciu
- *fuzzySetCount* - počet fuzzy množín medzi hodnotami `-fuzzyRange` a `fuzzyRange`
- *fuzzyRange* - citlivosť funkcie príslušnosti, určuje interval, medzi ktoré sa rozdelia fuzzy množiny

Vzniknuté fuzzy množiny majú potom tvar podobný množinám z obr. 4.3. `fuzzyRange` je na obrázku označený ako R. Z ich funkcií príslušnosti vyplýva, že fuzzifikáciou ostrej hodnoty budú nenulové najviac dve fuzzy množiny.

Funkcia vracia zoskupenie týchto fuzzy množín v podobe zoznamu. Výstupy z funkcie môžu vyzeráť takto:

```
[0.0, 0.045249897394742389, 0.95475010260525761, 0.0, 0.0]
[0.0, 0.0, 0.046481429632482141, 0.95351857036751786, 0.0]
[0.0, 0.0, 0.50770477069620301, 0.49229522930379699, 0.0]
[0.0, 0.0, 0.0, 0.031316539123626752, 0.96868346087637325]
[0.0, 0.0, 0.0, 0.78691719993014009, 0.21308280006985991]
```

Keďže každá časová rada má iný charakter trendu, nie je možné univerzálne určiť jednu hodnotu parametru `fuzzyRange` pre jednotlivé trendy časovej rady. Preto je potrebné túto hodnotu vypočítať osobitne pre každú časovú radu. Na jej určenie používam funkciu `getTrendRange()`. Tento postup zabezpečí univerzálnosť algoritmu pre akúkoľvek časovú radu.

4.4.5 Fyzyneurónová sieť a jej tréovanie

Po načítaní a normalizácii dát dochádza k vytvoreniu fuzzy neurónovej siete. Tá má na vstupe fuzzy množiny, ktoré boli bližšie popísané v kapitole 4.2.1. Na jej vytvorenie možno použiť zrýchlenú metódu knižnice PyBrain `buildNetwork()` na vytváranie siete. Pomocou jej parametrov je možné jednoduchým spôsobom určiť vnútornú štruktúru, vstupné, výstupné body a iné parametre.

Optimálnu vnútornú štruktúru siete som určil experimentálne. Tieto experimenty sú bližšie popísané v kapitole 5.

Výsledná fuzzy neurónová sieť má tvar podobný obr. 4.8. Vstupnú vrstvu (*Input layer*) tvorí 25 fuzzy hodnôt vytvorených zo zdrojových dát, skrytá vrstva (*Hidden layer*) obsahuje voliteľný počet fuzzy neurónov⁵. Sieť uzatvára skupina výstupných fuzzy neurónov, ktorých počet korešponduje s počtom klasifikačných tried (5).

Tréovanie siete

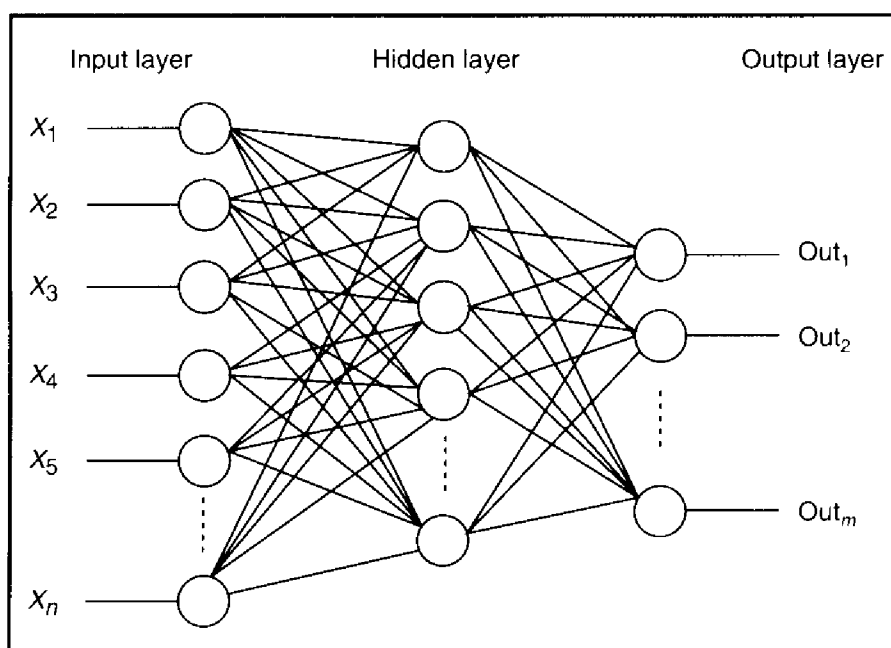
Na tréovanie siete je použitý tréner `BackpropTrainer` z knižnice PyBrain. Ide o triedu, ktorá pomocou algoritmu Backpropagation⁶ natrénuje fuzzy neurónovú sieť na základe vzorových výstupov.

Pred začatím tréovania je potrebné určiť dáta na tréovanie. Na tento účel je zo zdrojových dát vyhradená ich podmnožina. Zvyšná časť slúži ako testovacie dáta, na ktorých sa neskôr určuje chyba predikcie. Testovacie dáta sú vo výslednom grafe zobrazené na modrom pozadí (obr. 4.9).

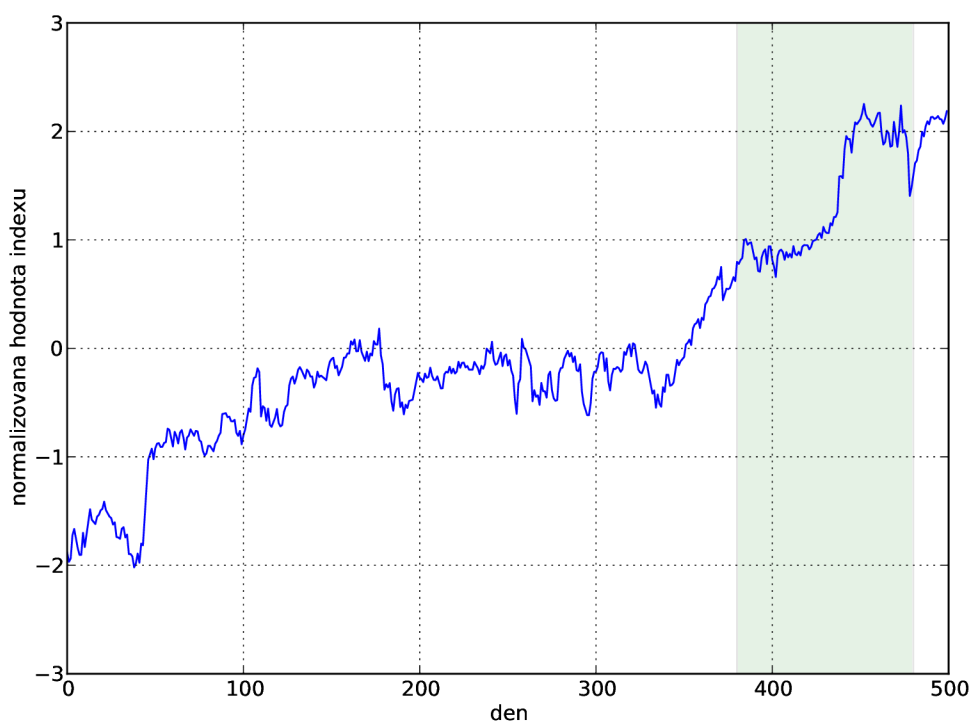
Tréovanie siete zaisťuje funkcia `trainer.trainOnDataset`, ktorej sa ako parametre predávajú `dataset` so zdrojovými dátami a počet tréovacích iterácií. Počas týchto iterácií algoritmus upravuje váhy synaptických spojení a spätne šíri chybu. Absolútna priemerná chyba je počas tohoto procesu vypisovaná na štandardný výstup.

⁵Tento počet je predmetom experimentu v kap. 5

⁶Algoritmus spätného šírenia chyby



Obrázek 4.8: fuzzy neurónová sieť



Obrázek 4.9: Trénovacie a testovacie dáta (v modrom rámečku)

4.4.6 Defuzzifikácia a výstup aplikácie

Po natrénovaní siete, je možné začať s testovaním a určovaním predikcie. Samotná sieť sa správa ako určitá funkcia, ktorá na základe vstupných parametrov vypočíta výstupnú hodnotu. Výstupom fuzzy neurónovej siete v tejto aplikácii je sada piatich fuzzy neurónov, ktoré obsahujú fuzzy hodnoty. Na to aby bolo možné výstup zo siete použiť je potrebné tieto hodnoty defuzzifikovať.

Na defuzzifikáciu používam metódu CAM (*Center Average method*)[2], ktorá bola bližšie popísaná v kapitole 3.2.2. Dôvodom použitia je jej jednoduchosť a tiež symetrickosť použitých fuzzy množín.

Defuzzifikáciou získame nielen charakteristickú kategóriu trendu ale aj jeho ostrú hodnotu v podobe smernice. Tá sa využíva napríklad pri vykresľovaní smeru trendu v podobe oranžovej šípky, ktorá približne určuje jeho ďalší vývoj. Aplikácia pre porovnanie vykresľuje aj modré šípky, ktoré predstavujú skutočný vývoj od bodu, kde má šípka počiatok. Je dôležité poznamenať, že tieto šípky nie sú regresiou hodnôt, ide len o určenie uhla pod akým hodnoty ďalej narastajú. Lineárna regresia by bola k týmto šípkam rovnobežná.

4.4.7 Určenie chyby

Vo všeobecnosti vykonáva aplikácia klasifikačnú úlohu, pri ktorej sa snaží zaradiť vstupné hodnoty do kategórií. Kategórie predstavujú lingvistické hodnoty trendov - pokles, mierny pokles, stagnácia, mierny nárast a nárast. Na určenie chyby je potrebné zistiť, koľko prvkov zo vstupných dát bolo zaradených do správnej (resp. nesprávnej) kategórie.

Aplikácia počíta dva druhy chýb:

- *chybu učenia* - chyba, ktorá vznikne tým, že fuzzy neurónová sieť nie je schopná dostatočne konvergovať a presne aproximovať skutočný výstup na základe zadaných vstupov a
- *testovacia chyba (validačná)* - chyba ktorá skúma schopnosť systému adaptovať sa na dáta, ktoré neboli súčasťou učenia. Táto chyba je v aplikácii podstatná, pretože na základe nej možno odhadnúť presnosť výslednej predikcie.

Z dôvodu príliš jemného rozdelenia fuzzy množín určujúcich charakter trendu (5 stupňov) som sa rozhodol pri určovaní chyby spojiť koncové dve kategórie oboch krajných prípadov. To má za následok, že ak aplikácia predikovala v nejakom bode časovej rady mierny nárast a v skutočnosti išlo o nárast, pokladám túto situáciu za úspešnú predikciu, pretože z praktického hľadiska ide o úspech. Napriek tomu skript vo výpise určí výsledok v 5 stupňovej škále a poskytuje tak presnejšiu informáciu.

Oba druhy chýb sú vo výslednom grafe zobrazené v percentuálnej podobe.

Kapitola 5

Prípadové štúdie

5.1 Vymedzenie problému

Táto kapitola obsahuje niekoľko prípadových štúdií, ktorých výsledkom je určenie predikcie vývoja akcií rôznych spoločností. Obsahuje experimenty, ktoré vedú k určeniu optimálnej vnútornej štruktúry siete a určeniu optimálneho počtu iterácií učenia.

Prípadové štúdie spočívajú v tom, že aplikácii sú ako vstupné údaje predávané časové rady rôznych akcií. Aplikácia ich spracuje a vypíše výsledky predikcie, chybu učenia a testovaciu chybu. Tento proces sa opakuje s rôznym počtom iterácií učenia. Dôvodom na opakovanie testu s rôznymi počtami iterácií je určenie optimálneho počtu iterácií vzhľadom na chyby, ktoré počas predikcie vzniknú.

Prílišné zvyšovanie počtu iterácií učenia spôsobuje *preučenosť* siete. Ide o určitý bod kedy už zvyšovanie iterácií učenia nespôsobuje znižovanie testovacej chyby, práve naopak, chyba rastie. Dôvodom je že sieť síce presne aproximuje dáta určené na učenie, avšak pri testovacích dátach vykazuje vysokú chybovosť klasifikácie.

Účelom tohto experimentu je práve určiť optimálny počet iterácií učenia, aby hodnoty chyby učenia boli minimálne (tá s počtom iterácií klesá) a zároveň nespôsobili preučenosť siete, teda aby aj testovacia chyba bola minimálna (od určitého bodu rastie).

Druhé kritérium, ktoré experimenty skúmajú je optimálna vnútorná štruktúra siete. Preto je postup popísaný vyššie opakovaný pre rôzne počty skrytých fuzzy neurónov (nazveme tento parameter *hidden*).

Výsledkom týchto experimentov je teda sieť s optimálnou vnútornou štruktúrou a počtom učiacich iterácií. Funkčná hodnota tejto siete je potom výslednou predikciou aplikácie pre danú časovú radu.

5.2 Prípadová štúdia 1 - Akcie International Business Machines Corporation (IBM)



Spoločnosť IBM, prezývaná aj *Big Blue*, fungujúca od roku 1888 a ako akciová spoločnosť od roku 1911, je poprednou svetovou spoločnosťou v odbore informačných technológií. Medzi hlavné činnosti spoločnosti patrí v súčasnosti výroba a predaj počítačového softvéru a hardvéru a rôzne služby.

Sídlo spoločnosti je v Armonk v New Yorku, USA. V roku 2010 vykázala zisk 23,1 z celkových výnosov vo výške 99,8 miliárd amerických dolárov. S majetkom v hodnote 113,4 mld dolárov a 426 751 zamestnancami patrí medzi najväčšie spoločnosti na svete. Jej akcie sa účastnia obchodovania na NYSE¹ pod symbolom IBM.[15]

Ako zdrojové dáta boli použité uzatváracie hodnoty akcií IBM v horizonte od 13. mája 2009 do 14.4.2011, teda 500 obchodných dní.

Určenie optimálnych parametrov aplikácie

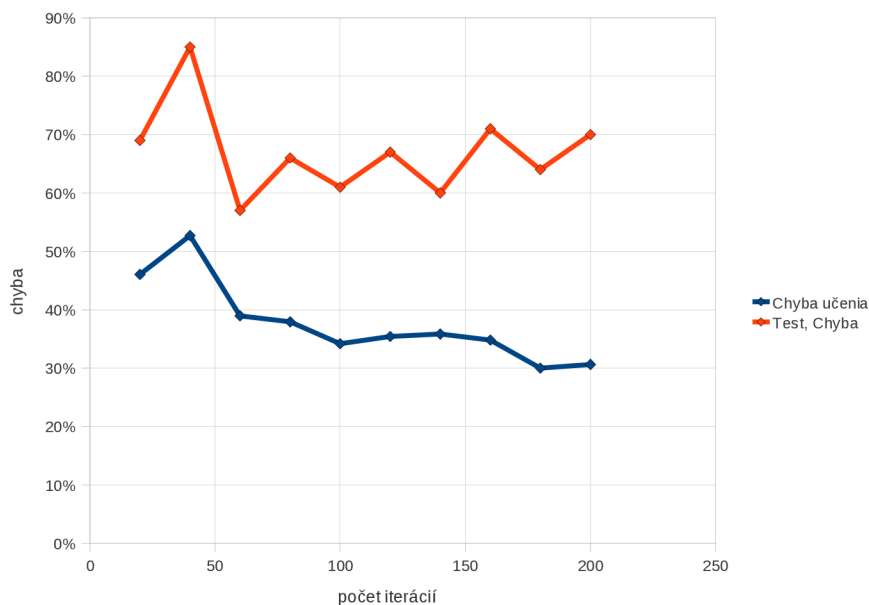
Spomedzi testovaných hodnôt vnútornej štruktúry skrytej vrstvy vykazovala najoptimálnejšie výsledky sieť s 50 fuzzyneurónmi. Testovanie jej chýb pre rôzny počet iterácií zobrazuje tabuľka 5.1. Jej grafická reprezentácia je v grafe 5.1.

Iterácie	Chyba učenia	Test. Chyba
20	46,04%	69,00%
40	52,71%	85,00%
60	38,96%	57,00%
80	37,92%	66,00%
100	34,17%	61,00%
120	35,42%	67,00%
140	35,83%	60,00%
160	34,79%	71,00%
180	30,00%	64,00%
200	30,63%	70,00%

Tabuľka 5.1: Tabuľka chýb pre rôzne hodnoty iterácie učenia (hidden = 50)

Z testov vyplýva, že optimálny počet iterácií sa pohybuje okolo hodnoty 60. Je tiež vidieť, že chyba učenia s narastajúcim počtom iterácií klesá, pričom zároveň chyba testovania v globále rastie. Z grafu testovacej chyby je zrejmé, že sieť je ovplyvňovaná náhodným faktorom, ktorý má základ v náhodnej inicializácii synaptických spojení siete.

¹New York Stock Exchange



Obrázek 5.1: Chyby aplikácie pre rôzne hodnoty iterácií učenia

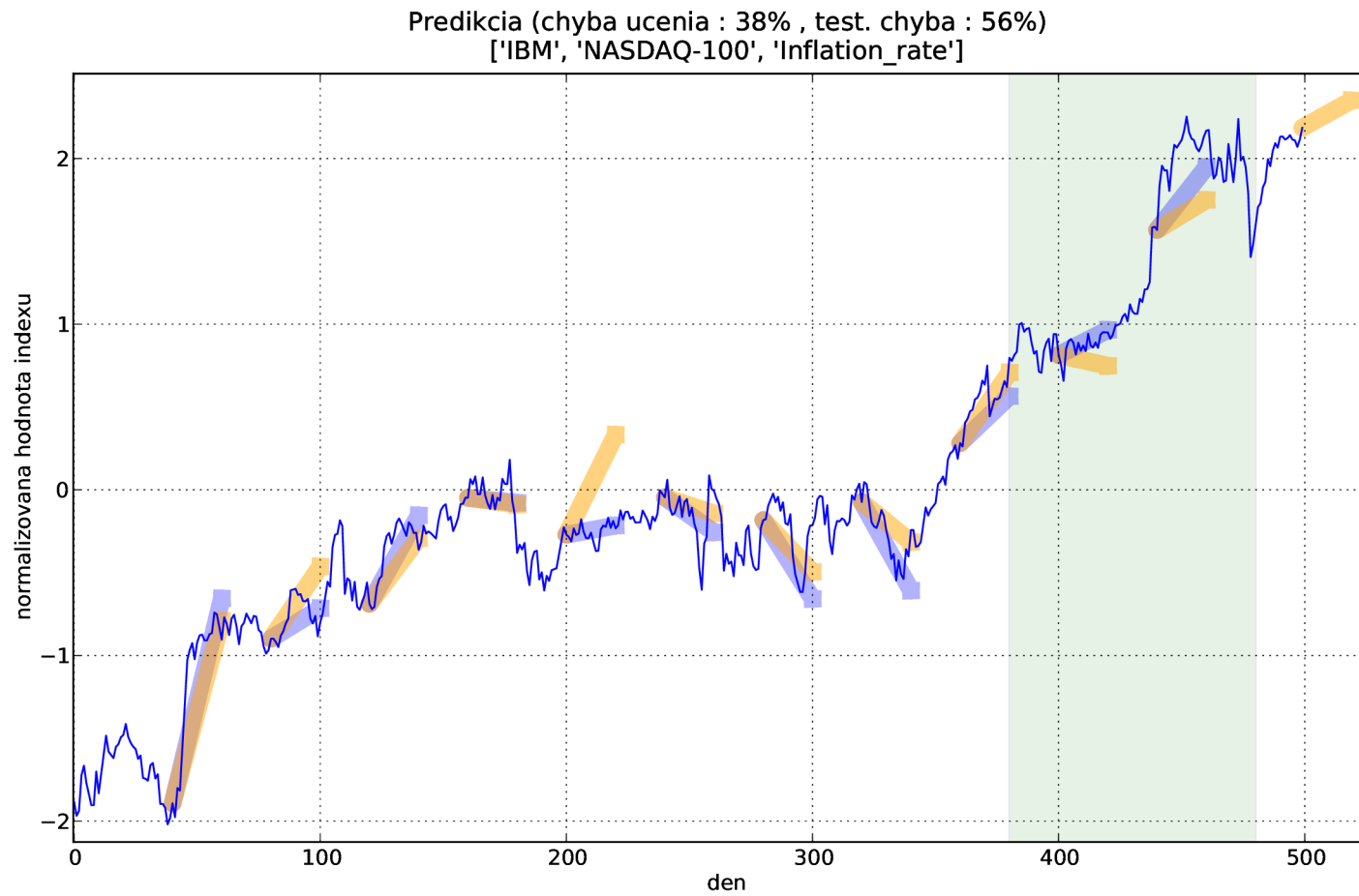
Predikcia

Obr 5.2 ukazuje výslednú predikciu akcie IBM s počtom iterácií 60 a 50 vnútornými fuzzy-neurónami. Biela plocha naľavo od modrého rámčeka predstavuje dáta určené na tréning siete, chyba ich klasifikácie je 38%. V modrom rámčeku sa nachádzajú dáta určené na testovanie, pričom chyba ich klasifikácie je 56%. Modré šípky naznačujú skutočný vývoj 20 obchodných dní dopredu a slúžia na porovnanie s oranžovými šípkami, ktoré sú výsledkom defuzifikácie výstupu fuzzy neurónovej siete. Medzi modrou plochou a koncovou predikciou je medzera, ktorá súvisí s predikovaním poslednej hodnoty z množiny testovacích dát.

Záver

Prvá časť prípadovej štúdie bola zameraná na určenie optimálnych parametrov aplikácie. V druhej časti bola vypočítaná predikcia akcie IBM v horizonte 20 obchodných dní.

Výsledná koncová predikcia bola aplikáciou klasifikovaná ako *mierny nárast*. Táto predikcia sa počas nasledujúcich dní potvrdila, keďže hodnota akcie vzrástla od 14. apríla z 165,16 \$ na 170,53 \$ za kus (hodnota k 5. máju 2011).



Obrázek 5.2: Predikcia vývoja akcie IBM

5.3 Prípadová štúdia 2 - Akcie Google Inc.



Google Inc. je americkou medzinárodnou korporáciou, ktorá sa zaoberá najmä internetovým vyhľadávaním, *cloud computingom*² a reklamnými technológiami. Google poskytuje a vyvíja množstvo internetových služieb a produktov ako napríklad Google Maps, Google Earth, Google Docs, Google Calendar, Google AdWords atď. Svoj profit generuje najmä z reklamy prostredníctvom programu AdWords. Majetok spoločnosti je 57,851 miliardy dolárov a ročný obrat predstavuje 10,381 miliardy dolárov³.

Ako zdrojové dáta boli použité uzatváracie hodnoty akcií Google Inc. v horizonte od 13. mája 2009 do 14.4.2011, teda 500 obchodných dní.

Určenie optimálnych parametrov aplikácie

Pri určovaní vnútornej štruktúry dosahovali najlepšie výsledky siete so 70 a 100 skrytými fuzzyneurónmi. Keďže išlo v týchto prípadoch o porovnateľný výkon siete, vo výslednej predikcii bol použitý model s hidden=70 z dôvodu efektivity. V tabuľke 5.2 je porovnanie chybovosti klasifikácií pre rôzne hodnoty parametru hidden, pričom sú vybrané najlepšie hodnoty z každého experimentu. Testovanie jej chýb pre rôzny počet iterácií zobrazuje tabuľka 5.3. Jej grafická reprezentácia je v grafe 5.3. Počet skrytých neurónov je určený na 70.

Hidden	Iterácie	Chyba učenia	Test. Chyba
30	100	33,75%	55,00%
50	40	35,63%	54,00%
70	140	27,71%	48,00%
100	200	34,79%	50,00%

Tabuľka 5.2: Tabuľka chýb pre rôzne hodnoty parametru hidden (vybraný najlepší prípad)

Predikcia

Obrázok 5.4 ukazuje výstup aplikácie predikujúci vývoj hodnoty akcií Google. Na prvý pohľad vidieť, že ide o dynamickejší vývoj ako v prípade akcií IBM a hodnota podlieha silnej volatilitě⁴. Chyba učenia dosiahla hodnotu 23% a testovacia chyba 52%, čo je o niečo viac ako pri experimente s parametrom hidden.

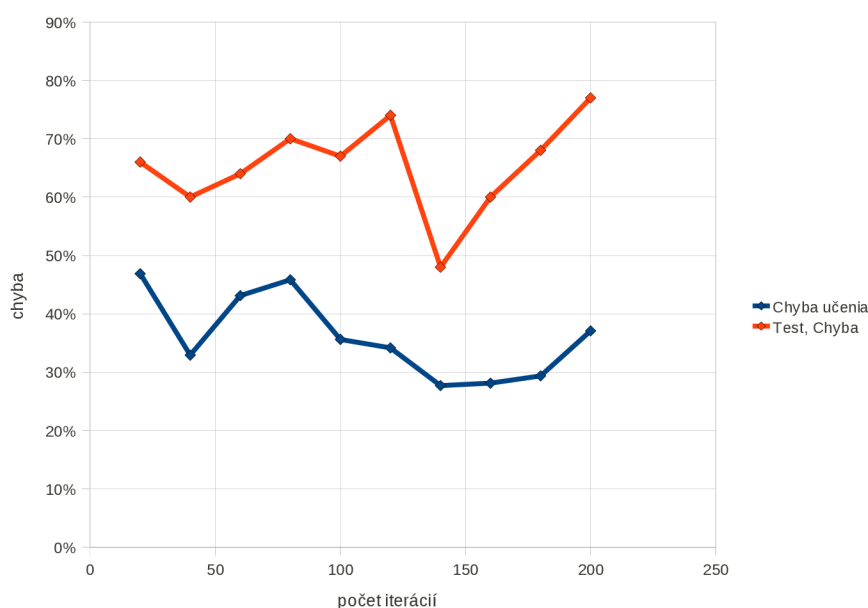
²Cloud computing možno charakterizovať ako poskytovanie služieb či programov uložených na serveroch Internetu s tým, že užívatelia k nim môžu pristupovať napríklad pomocou webového prehliadača alebo klienta danej aplikácie a používať prakticky odkiaľkoľvek.

³U.S. Securities and Exchange Commission (2010).

⁴V ekonómii a štatistike kolísavosť, nestálosť, premenlivosť hodnôt časovej rady.

Iterácie	Chyba učenia	Test. Chyba
20	46,88%	66,00%
40	32,92%	60,00%
60	43,13%	64,00%
80	45,83%	70,00%
100	35,63%	67,00%
120	34,17%	74,00%
140	27,71%	48,00%
160	28,13%	60,00%
180	29,38%	68,00%
200	37,08%	77,00%

Tabulka 5.3: Tabuľka chýb pre rôzne hodnoty parametru hidden (vybraný najlepší prípad)

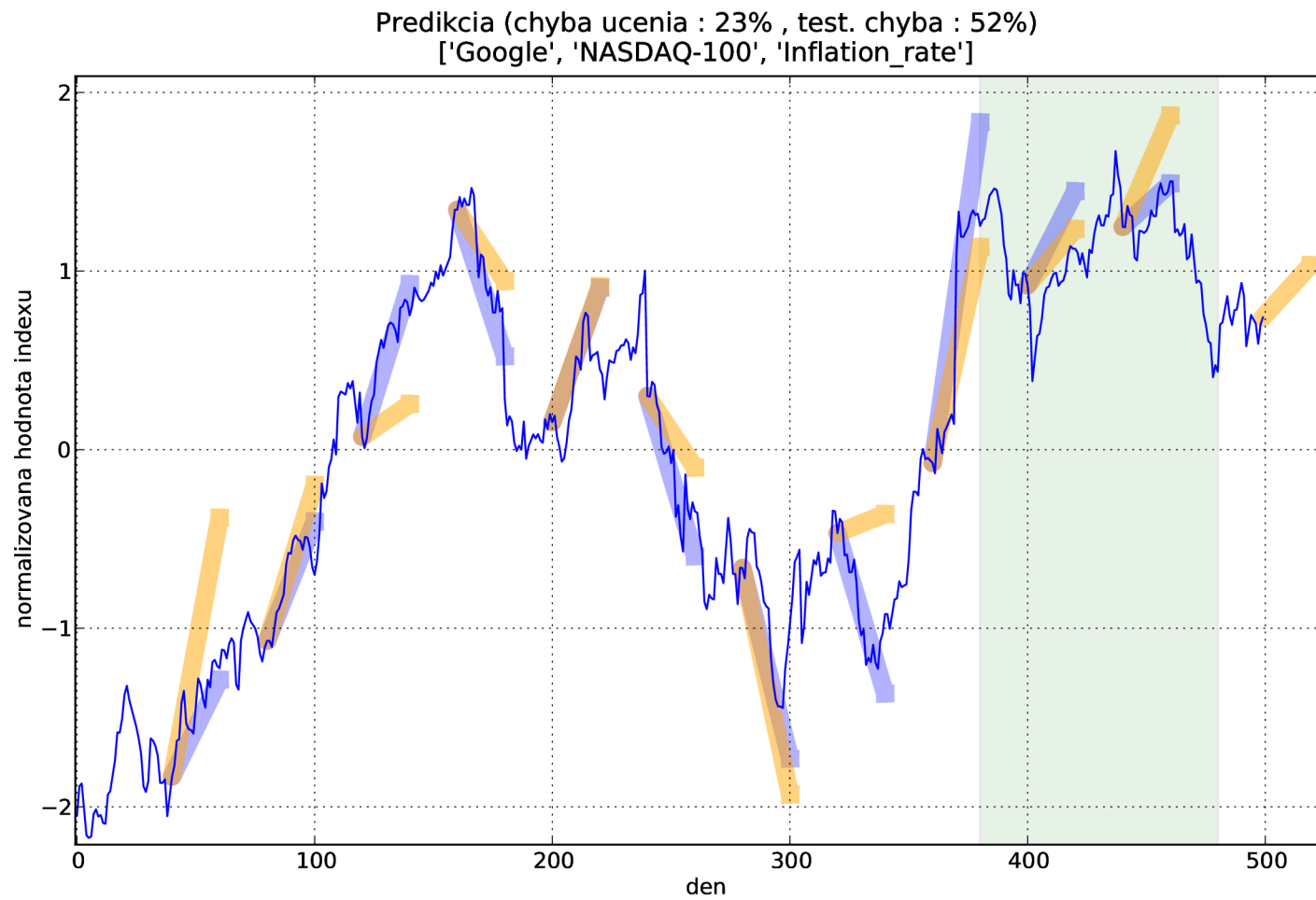


Obrázek 5.3: Chyby aplikácie pre rôzne hodnoty iterácií učenia

Záver

Prvá časť prípadovej štúdie bola zameraná na určenie optimálnych parametrov aplikácie. V druhej časti bola vypočítaná predikcia akcie Google v horizonte 20 obchodných dní.

Výsledok predikcie klasifikovala aplikácia ako *mierny nárast*. Počas nasledujúcich 20 obchodných dní, ktoré boli predmetom predikcie došlo k silnému prepadu hodnoty akcií Google. Zo dňa 14. apríla na 15. apríla klesla hodnota z 579,05\$ na 530,70\$ a táto hodnota pretrvala až do 7. mája (535,63\$). Aplikácia tento výrazný pokles hodnoty akcie nedokázala predpovedať.



Obrázek 5.4: Predikcia vývoja akcií Google

5.4 Prípadová štúdia 3 - Akcie Honeywell International Inc.

Honeywell

Honeywell je konglomerát⁵, ktorého spoločnosti produkujú široký sortiment spotrebných výrobkov, inžinierskych služieb a letových systémov. Cieľovými zákazníkmi sú privátne spoločnosti, korporácie a vláda. Honeywell zamestnáva viac ako 130 000 zamestnancov a jeho ročný obrat je 33,37 miliardy amerických dolárov.⁶

Ako zdrojové dáta boli použité uzatváracie hodnoty akcií Honeywell v horizonte od 13. mája 2009 do 14.4.2011, teda 500 obchodných dní.

Určenie optimálnych parametrov aplikácie

Najlepšie výsledky dosahovala sieť pre parameter hidden zvolený na 50. Tabuľka 5.4 ukazuje chybovosť klasifikácie pre jednotlivé hodnoty iterácií. Z grafu týchto hodnôt (Obr. 5.5) vyplýva, že testovacia chyba v niektorých prípadoch klesla pod úroveň chyby učenia. To je spôsobené dlhodobým stúpajúcim trendom v oblasti testovacích dát (modré pozadie). Sieť takýto trend klasifikovala presnejšie ako premenlivú oblasť dát pri učení (ľavá strana s bielym pozadím).

Iterácie	Chyba učenia	Test. Chyba
20	34,58%	21,00%
40	33,75%	38,00%
60	32,08%	21,00%
80	26,46%	27,00%
100	28,75%	37,00%
120	40,63%	68,00%
140	25,21%	42,00%
160	32,29%	62,00%
180	25,00%	41,00%
200	25,83%	42,00%

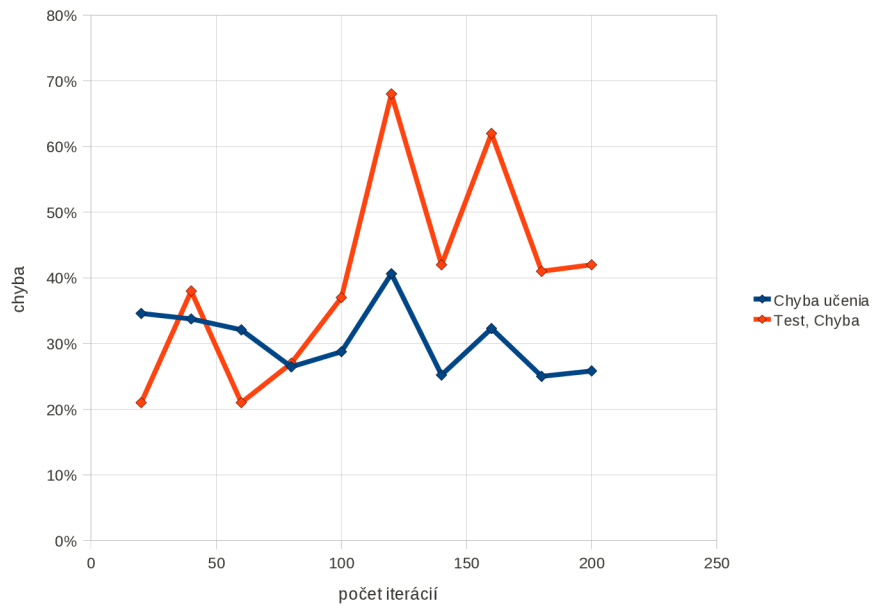
Tabuľka 5.4: Chyby aplikácie pre rôzne hodnoty iterácií učenia

Predikcia

Obrázok 5.6 ukazuje výstupnú predikciu aplikácie pri predikovaní vývoja hodnoty akcií Honeywell. Hodnoty chyby učenia a testovacej chyby sú v porovnaní s predchádzajúcimi experimentami nízke, chyba učenia je 26,5% a testovacia chyba 25%.

⁵Veľký podnik (holding), do ktorého patria podniky rôznych nesúvisiacich odvetví.

⁶U.S. Securities and Exchange Commission, 2010

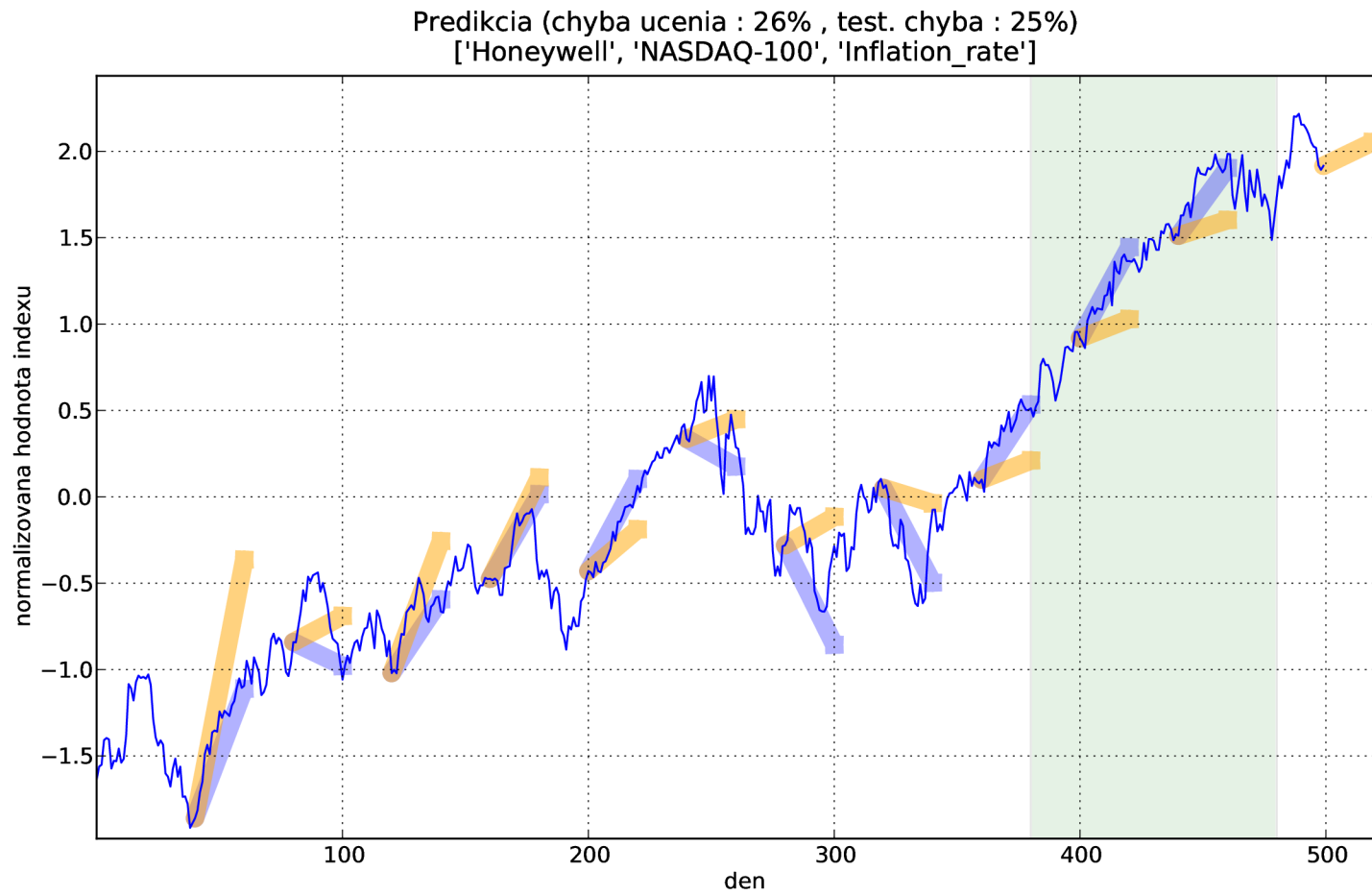


Obrázek 5.5: Chyby aplikácie pre rôzne hodnoty iterácií učenia

Záver

Prvá časť prípadovej štúdie bola zameraná na určenie optimálnych parametrov aplikácie. V druhej časti bola vypočítaná predikcia akcie Honeywell v horizonte 20 obchodných dní.

Aplikácia klasifikovala výsledok predikcie ako *nárast*. Hodnota akcie stúpala v priebehu 20 dní od 14.4.2011 z 57,49\$ na 60,32\$ (4.5.2011). Predikciu preto možno považovať za úspešnú.



Obrázek 5.6: Predikcia vývoja akcií Honeywell

5.5 Zhodnotenie prípadových štúdií

V tejto časti práce bol vytvorený model testovaný na reálnych dátach troch spoločností - IBM, Google a Honeywell. Tieto časové rady tvorené uzatváracími cenami boli vybrané zámerne, aby bolo možné pozorovať výstupy aplikácie rôznych charakterov trendu. Úlohou aplikácie bolo predikovať vývoj hodnoty akcie v horizonte 20 obchodných dní. Po uplynutí tejto doby bol výsledok aplikácie porovnaný s reálnym vývojom.

Z prípadových štúdií je zrejmé, že použitý model má schopnosť lepšie aproximovať a následne predikovať časové rady, ktoré majú plynulý priebeh, ako bolo možné pozorovať pri predikovaní vývoja akcií spoločnosti Honeywell a IBM. Naopak, keď dochádzalo k prudkým zmenám hodnoty (prudký prepád akcií Google), model nebol schopný tento vývoj predikovať.

Zhodnotenie využitia fuzzy logiky a fuzzy neurónových sietí pri predikcii akciového trhu

Po porovnaní výsledkov aplikácie s reálnym vývojom možno skonštatovať úspech dvoch z troch vykonaných predikcií. Napriek tomu však model často vykazoval rozporuplné výsledky pri rovnakých parametroch. Dôvodom tohto chovania je zrejme náhodná inicializácia siete malými hodnotami.

Najlepšie výsledky boli pozorované pri plynulých časových radách bez prudkých zmien (validačná chyba okolo 25%). Pri niektorých časových radách s extrémnou volatilitou mal však model problém konvergovať a validačná chyba klasifikácie presahovala 70%. Dobré výsledky nastali tiež pri opakovaní určitého vzoru v zdrojových dátach, čo súvisí s bázou pravidiel fuzzy systému, ktorý sa takýto vzor naučí a pri validácii je ho schopný rozpoznať.

Z globálneho hľadiska ale nepovažujem použitie tohoto modelu za spoľahlivé. Dôvodom sú hlavne vysoké hodnoty testovacej (validačnej) chyby, ktoré majú zrejme súvis s jednoduchosťou modelu.

5.5.1 Možnosti rozšírenia

Predikcia trhu je v súčasnosti aktívne rozvíjajúcou sa disciplínou a existuje mnoho metód a postupov na jej docielenie. Niektoré z nich sú popísané v nasledujúcom zozname. Skombinovanie týchto metód s postupmi popísanými v tejto práci dáva reálny predpoklad na zlepšenie výkonnosti modelu a môže viesť k úspešnejšej predikcii.

- **Elliotove vlny** - vychádzajú z teórie fraktálnych trhov. Fungujú na princípe odhaľovania vzorov v časových radách, ktoré sa v nich opakovane vyskytujú v rôznych veľkostiach.
- **Support a rezistencia** (metóda dvojitého vrcholu alebo dvojitého dna) - support je cena, ktorá je predávajúcimi vnímaná ako spodná hranica, pod ktorú už nie sú ochotní predávať. Naopak, rezistencia je vnímaná kupujúcimi ako strop, maximum, ktorý sú za jednotku ochotní zaplatiť. Určenie týchto bodov môže napomôcť k predvídaní ďalšieho vývoja.[7]
- **Hurstov exponent** - určuje mieru chaotickosti časovej rady. Dokáže rozlíšiť chaotickú (fraktálnu) časovú radu od náhodnej a najšť dlhodobý pamäťový cyklus u chaotickej časovej rady.

Kapitola 6

Záver

V tejto práci bol čitateľ najprv zoznámený s teoretickými základmi fuzzy logiky a fuzzy systémov. Táto teória bola ďalej rozšírená o fuzzy neurónové siete, čím dostal dostatočnú teoretickú výbavu na porozumenie praktickej časti.

V praktickej časti bol na základe modelov popísaných v teoretickej časti vytvorený model využívajúci fuzzy neurónovú sieť ako dynamicky učiacu sa bazu pravidiel fuzzy systému. Tento model bol následne implementovaný v jazyku python. Výsledná aplikácia rieši klasifikačnú úlohu, pri ktorej sa snaží predikovať vývoj hodnoty akcie v horizonte niekoľkých obchodných dní. Vstup aplikácie tvorí sada historických trendov s rôznymi dĺžkami, ku ktorým sú pridané dva pomocné indexy.

Posledná časť sa venovala prípadovým štúdiám, v ktorých bol predikovaný vývoj akcií spoločností IBM, Google a Honeywell. Následne boli výsledky predikcií a výkonnosť modelu zhodnotené. Prípadové štúdie ukázali, že výkonnosť modelu je závislá na dvoch faktoroch - volatilita vstupných dát a opakujúcich sa vzoroch. V globále však model nedosahoval výsledky zlučiteľné s použitím v praxi.

Východisko je možné hľadať v spojení tohoto modelu s inými, často používanými technikami technickej analýzy ako napríklad Elliotove vlny, metóda dvojitého vrcholu alebo dvojitého dna (support a rezistencia), či použitie Hurstovho exponentu.

Na záver možno povedať, že fuzzy logika má pre svoju schopnosť vyjadrovania nepresnosti spojením s umelou inteligenciou v oblasti predikcie akciového trhu veľký potenciál a stále ponúka široký priestor na nové prístupy k riešeniam problémov, ktoré možno považovať za nadčasové.

Literatura

- [1] Dostál, P.: *Pokročilé metody analýz a modelování v podnikatelství a veřejné správě*. CERM, 2009, iISBN 9788072046058.
- [2] Jura, P.: *Základy fuzzy logiky pro řízení a modelování*. VUTIUM, 2003, iISBN 80-214-2261-0.
- [3] Marček, D.: *Neuronové sítě a fuzzy časové řady s aplikacemi v ekonomice*. Slezská univerzita, 2002, iISBN 80-7248-157-6.
- [4] Navara, M.: *Základy fuzzy množin*. ČVUT, 2007, iISBN 978-80-01-03668-6.
- [5] Schaul, T.; Bayer, J.; Wierstra, D.; aj.: PyBrain. *Journal of Machine Learning Research*, 2010.
- [6] Taylor, T.: *Principles of Economics*. Freeload Press, 2008, iISBN 193078905.
- [7] WWW: Financnik.cz - komodity, akcie, burza, forex. <http://www.financnik.cz>.
- [8] WWW: Consumer price index. http://en.wikipedia.org/wiki/Consumer_price_index, [cit. 2011-04-17].
- [9] WWW: Dow Jones Stock Market Index - DJIA. <http://forecastchart.com/dow-jones-unemployment.html>, [cit. 2011-04-17].
- [10] WWW: NASDAQ-100 stock market index. <http://en.wikipedia.org/wiki/NASDAQ-100>, [cit. 2011-04-17].
- [11] WWW: RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files. <http://tools.ietf.org/html/rfc4180>, [cit. 2011-04-17].
- [12] WWW: matplotlib library. <http://matplotlib.sourceforge.net/>, [cit. 2011-04-18].
- [13] WWW: SciPy - Scientific Tools for Python. <http://http://www.scipy.org/>, [cit. 2011-04-18].
- [14] WWW: Fuzzy logika a fuzzy znalostné systémy. <http://www2.fiit.stuba.sk/~kapustik/ZS/Clanky0708/pok/index.html#Sekcia41>, [cit. 2011-05-01].
- [15] WWW: International Business Machines Corporation (IBM). <http://en.wikipedia.org/wiki/IBM>, [cit. 2011-05-01].

- [16] WWW: Lineární regrese.
http://cs.wikipedia.org/wiki/Line%C3%A1rn%C3%AD_regrese, [cit. 2011-05-01].
- [17] Zadeh, L.: Fuzzy Sets. *Information Control*, ročník 8, 1965: s. 338–353,
<http://www-bisc.cs.berkeley.edu/Zadeh-1965.pdf>.