VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE ELLIOTTOVÝCH VLN POMOCÍ NEURONOVÉ SÍTĚ

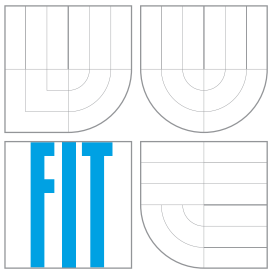
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN GREGA

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE ELLIOTTOVÝCH VLN POMOCÍ NEURONOVÉ SÍTĚ

DETECTION OF ELLIOTT WAVES USING NEURAL NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MARTIN GREGA
AUTHOR

VEDOUCÍ PRÁCE — Ing. MILOŠ MINAŘÍK
SUPERVISOR

BRNO 2012

Abstrakt

Bakalářská práce se zabývá využitím umělých neuronových sítí pro detekci Elliottových vln ve statické časové řadě. Použitá je konkrétně jednoduchá committee machine sestávající z vícevrtsvých perceptronů, trénovaných pomocí algoritmu pružného zpětného šíření chyby. Práce obsahuje návrh a implementaci apikace pro detekci impulzních vln na vstupních datech.

Abstract

This thesis deals with application of artificial neural networks for detection of Elliott waves in static time-series. It is focused mainly on use of simple committee machine consisting of multilayer perceptrons trained by resillient propagation algorithm. Thesis contains design and implementation of application for detection of impulsive waves on input signal.

Klíčová slova

Elliottovy vlny, Vlnová teorie, umělé neuronové sítě, rozpoznávaní vzorů, časové řady

Keywords

Elliott Waves, Wave Principle, Artificial Neural Networks, pattern recognition, time series

Citace

Martin Grega: Detekce Elliottových vln pomocí neuronové sítě, bakalářská práce, Brno, FIT VUT v Brně, 2012

Detekce Elliottových vln pomocí neuronové sítě

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Miloša Minaříka.

.....
Martin Grega
15. května 2012

Poděkování

Za konzultácie a odborné vedenie práce chcem poďakovať Ing. Milošovi Minaříkovi. Taktiež ďakujem svojej rodine a priateľom za podporu, ktorú mi poskytli pri tvorbe tejto práce.

Obsah

Kapitola 1

Úvod

Už od čias vzniku prvých búrz skúsený obchodníci vedeli, že pohyby ceny na trhu nie sú úplne náhodné. Vo veľkej miere sú spôsobené kolektívnou psychológiou. Grafy trhov sú preto odrazom správania jednotlivých účastníkov trhu, ich potrebami a emóciami ako strach a chamtivosť. Zmeny nálady, pomer ponuky a dopytu a emočné výkyvy obchodníkov sa na pohybe ceny prejavujú v určitých vlnách. Práve týmito vzormi sa zaoberá Elliottová vlnová teória, ktorá podľa zákonitostí a pravidiel vytvorených na základe empirických dôkazov a štatistiky, slúži na analýzu tržných pohybov a trendov.

Táto práca sa zaoberá detekciou Elliottových vĺn pomocou umelých neurónových sietí. Približuje problematiku Elliottových vĺn a ich vlastností z pohľadu možností detekcie a klasifikácie. Taktiež obsahuje stručný úvod do oblasti umelých neurónových sietí s prihliadnutím na teoretické východiská pre praktickú časť.

V rámci práce je navrhnutá a implementovaná aplikácia, ktorá umožňuje detekovať jednú z formácií Elliottových vĺn na vstupnej časovej rade pomocou jednoduchej committee machine, ktorá je zložená z viacvrstvových sietí perceptrónov trénovaných prostredníctvom algoritmu pružného spätného šírenia chyby. Výsledkom práce je program, ktorý s určitou presnosťou nachádadza impulzné vlny na zadanej časovej rade.

Na záver je diskutovaná vhodnosť navrhnutého modelu, použiteľnosť v praxi, jeho možné vylepšenia a rozšírenia.

Kapitola 2

Teoretické východiská práce

Táto kapitola prezentuje teoretické základy z oblasti Elliottovej vlnovej teórie a umelých neurónových sietí.

Cieľom teoretickej časti tejto práce je zoznámiť čitateľa s teóriou Elliottových vĺn a poskytnúť mu aspoň hrubú predstavu o tom, čo vlastne Elliottove vlny sú, s prihliadnutím na ich detekciu a klasifikáciu. Práca sa neozaoberá podrobnejším zaradzovaním vĺn do hierarchickej štruktúry realizovaným pomocou súboru pravidiel popísaných v [1] a ich vlastnosťami z hľadiska možnej predpovede budúceho vývoja ceny na trhu. Ďalšie informácie o tejto problematike je možné nájsť v [2].

Ďalej bude čitateľ stručne oboznámený s problematikou umelých neurónových sietí. Táto časť je zameraná na teoretické oblasti, ktorých porozumenie je nevyhnutné pre pochopenie praktickej časti.

Všetky odborné pramene, z ktorých teoretická časť vychádza, sú uvedené na záver práce v časti Literatúra.

2.1 Elliottová vlnová teória

Elliottová vlnová teória je jedným zo spôsobov rozboru trendov na burzových trhoch a patrí medzi najpresnejšie nástroje technickej analýzy.

Jej autorom je americký účtovník Ralph Nelson Elliott. Bola vytvorená v 30. rokoch 20. storočia, avšak k jej rozšíreniu a používaniu vo väčšom merítku prispeli až v 90. rokoch počítače. Vznikla z pozorovania opakujúcich sa pohybov a vzorov na grafoch kapitálového trhu. Vychádza z hypotéz o fraktálnom trhu a zároveň zapadá do neskôr vytvorenej teórie chaosu. [1]

Základom teórie je zistenie, že aj keď sa pohyb ceny môže zdať náhodný a nedpredvídateľný, v skutočnosti dodržiava určité pravidlá a má vnútornú štruktúru, ktorá sa v čase opakuje [1]. Vlnová teória však nie je nástrojom na predikciu trhu, ale slúži ako detailný popis a rozbor fungovania burzových trhov. Ale ak máme poznatky o správaní tržných pohybov, vieme určiť pravdepodobný budúci smer pohybu ceny [2].

Nasledujúce časti podrobnejšie popisujú Elliottove vlny a nevychádzajú len z pôvodných úvah Ralpha Elliotta, ale zahŕňajú aj poznatky jeho nasledovníkov. Táto podkapitola bola spracovaná podľa [1].

2.1.1 Základné pojmy

Vlna je základom Elliottovej teórie. Chápeme ju ako pohyb ceny (akcie alebo iného aktíva) v čase, ktorý má určitú dĺžku a smer. Vyjadruje nerovnováhu ponuky a dopytu na trhu. Ak na trhu prevažuje dopyt, cena rastie, naopak, ak prevažuje ponuka nad dopytom, cena klesá. Tomu zodpovedajú rastúce a klesajúce vlny.

Stupeň vlny je pojem, ktorý vyjadruje skutočnosť, že vlny majú rôznu veľkosť, teda líšia sa vo svojej výške (hodnota rozdielu ceny v počiatočnom a koncovom bode vlny) a dĺžke (čas od začiatku do konca vlny). Stupeň vlny je teda určený jej veľkosťou a pozíciou k susedným vlnám. V závislosti na ich časovej dĺžke Elliott pomenoval deväť stupňov vĺn. Ich pomenovania a symbolika, ktorou sú jednotlivé vlny v grafoch označované, sú uvedené v tabuľke 2.1, prevzatej z [1].

Stupeň vlny	**5 vĺn s trendom**	**3 vlny proti trendu**
Veľký supercyklus	Ⓘ, Ⓘⓘ, Ⓘⓘⓘ, Ⓘⓥ, Ⓥ	ⓐ, ⓑ, ⓒ
Supercyklus	(I), (II), (III), (IV), (V)	(a), (b), (c)
Cyklus	I, II, III, IV, V	a, b, c
Primárny	①, ②, ③, ④, ⑤	Ⓐ, Ⓑ, Ⓒ
Prostredný	(1), (2), (3), (4), (5)	(A), (B), (C)
Malý	1, 2, 3, 4, 5	A, B, C
Menší	ⓘ, ⓘⓘ, ⓘⓘⓘ, ⓘⓥ, ⓥ	ⓐ, ⓑ, ⓒ
Najmenší	(i), (ii), (iii), (iv), (v)	(a), (b), (c)
Minimálny	i, ii, iii, iv, v	a, b, c

Tabuľka 2.1: Názvy stupňov vĺn podľa R. Elliotta a symbolika, ktorou sú jednotlivé vlny označované. Zdroj: [1]

Monovlna popisuje pohyb trhu v jednom smere, až kým nenastane zmena smeru. Je to najjednoduchšia možná vlna, z ktorej sa skladajú vlny vyššieho stupňa a väčšie skupiny vĺn tzv. figúry.

Figúra je skupina vĺn nižšieho stupňa, ktorá spĺňa určité pravidlá. Pravidlá sú podmienky, ktoré musí vlna alebo figúra vo všetkých prípadoch bezpodmienečne spĺňať.

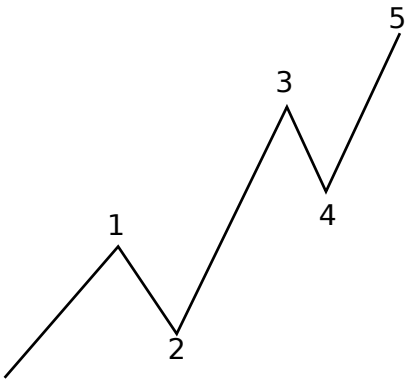
Podrozdelenie znamená rozšírenie alebo rozčlenenie vlny do vĺn nižšieho stupňa.

Štrukturový štítok slúži k zaradeniu cenového pohybu do príslušnej triedy. Označuje už dokončnú formáciu. Zapisujeme ho ako dvojbodku pred číslovkou, ktorá predstavuje počet vĺn vo formácii. Napríklad štítok :5 označuje impulz zložený z piatich vĺn.

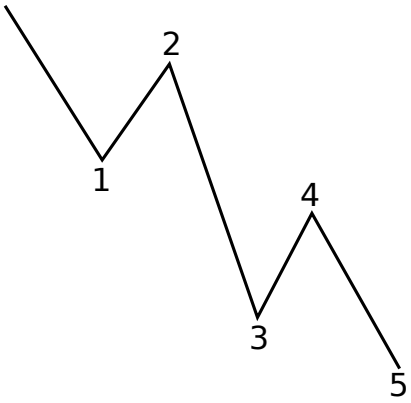
Postupový štítok sa využíva k označeniu každej vlny vo vnútri štandardizovanej figúry. Číslovkami označujeme časti hybných vĺn, časti korekčných vĺn označujeme písmenami.

2.1.2 Impulz

Impulz je považovaný za základnú vlnu Elliottovej teórie a je najtypickejším predstaviteľom hybného režimu, teda pohybu v smere trendu. Impulzná figúra má špecifickú päť vlnovú štruktúru, kde vlny 1, 3, 5 reprezentujú hybný pohyb, vlny 2 a 4 sú korekciami, teda prerušeniami trendu v rámci impulznej vlny. Idealizovaný priebeh impulzu je zobrazený na obrázkoch 2.1 a 2.2.



Obr. 2.1: Rastúci impulz



Obr. 2.2: Klesajúci impulz

Základne stavebné pravidlá

Podľa [1], musí tržný pohyb spĺňať nasledujúce pravidlá, aby mohol byť považovaný za kandidáta na impulsnú figúru, ak jedno z týchto pravidiel nie je dodržané, nejedná sa o impulz, ale korekčný pohyb.

1. Musí existovať päť susediacich vĺn, ktoré vyhovujú štruktúrovým požiadavkam na trendovú alebo zaverečnú figúru.
2. Tri z týchto piatich vĺn musia mať rovnaký smer, zvyšné dve vlny musia mať smer k nim opačný.
3. Ihneď po prvej vlne, nasleduje vlna v opačnom smere, ktorá nesmie byť dlhšia ako prvá vlna.
4. Tretia vlna musí byť dlhšia ako druhá.
5. Ihneď po tretej časti nasleduje vlna v opačnom smere. Táto štvrtá vlna musí byť kratšia ako tretia vlna a zároveň nesmie vstúpiť do cenovej zóny prvej vlny, viď 2.3.
6. Piata vlna musí byť dlhšia ako 38,2 % štvrtej vlny [1]. V prípade, že je piata vlna kratšia ako štvrtá, je táto figúra považoná za impulz so zlyhaním piatej vlny.
7. Tretia vlna nesmie byť najkratšia z vĺn 1, 3 a 5.

Zlyhanie

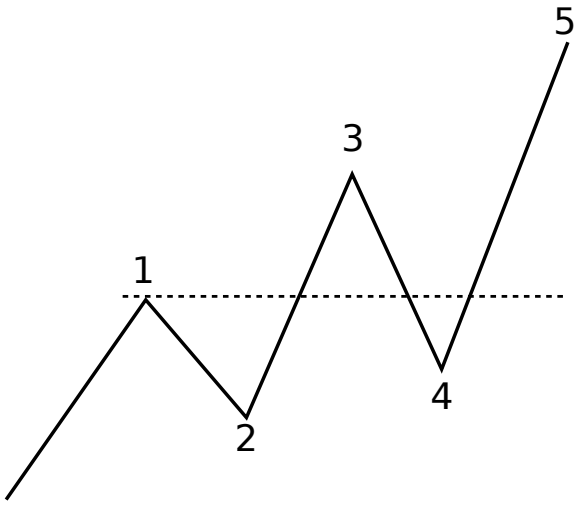
Zlyhanie v impulze nastáva, ak je piata vlna kratšia ako štvrtá, tzn. keď neprekročí koniec tretej vlny. Často sa vyskytuje v prípadoch, kedy tretia vlna vytvorí veľký cenový pohyb. Príklad zlyhania piatej vlny ukazuje obrázok 2.4.

Predĺženie

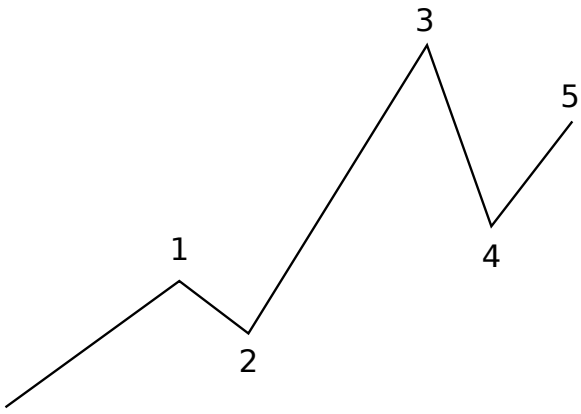
Tento pojem sa vzťahuje iba na hybné vlny v impulze, tzn. prvú, tretiu a piatu vlnu. Predĺženie znamená, že daná vlna je podrozdelená a zároveň je podstatne väčšia ako ostatné vlny vo figúre. Predĺženie je najčastejšie tvorené treťou vlnou, táto situácia je zobrazená na obrázoku 2.5.

Záverečný impulz

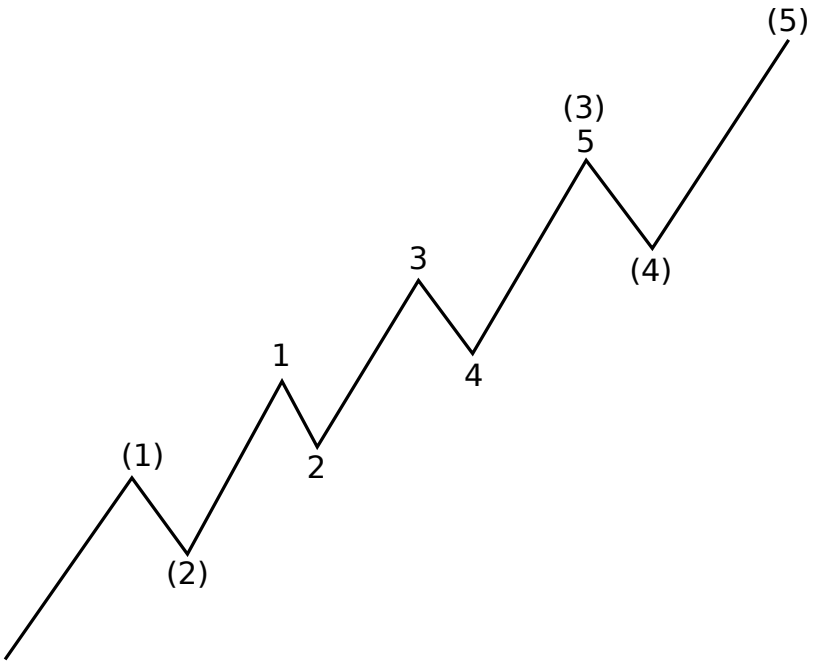
Záverečný impulz alebo tiež diagonálny trojuholník sa vyskytuje v situáciach, kde piata vlna na seba berie podobu trojuholníka, viď 2.6, prebrané z [1]. Tento typ impulzu tvorí čiastočnú výnimku zo stavebného pravidla č. 5, tzn., že na rozdiel od trendového impulzu, štvrtá vlna môže vstúpiť do cennovej zóny prvej vlny. Záverečný impulz sa často vyskytuje na konci figúr vyššieho stupňa a signalizuje nasledujúci dramatický pokles alebo rast ceny na trhu.



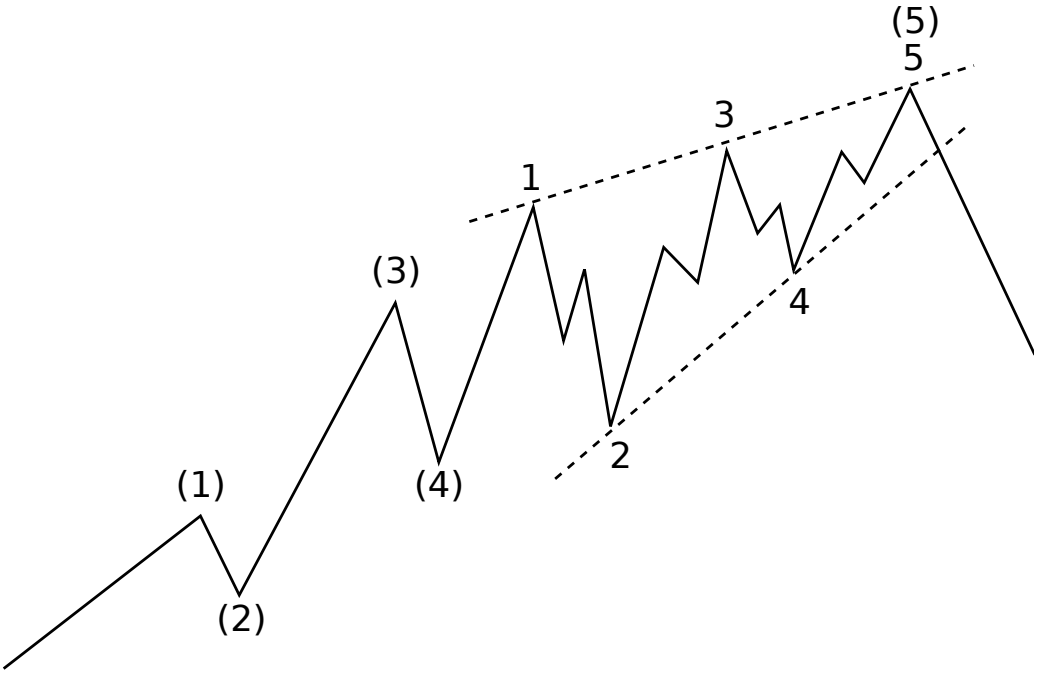
Obr. 2.3: Štvrtá vlna vstupuje do cenovej zóny prvej vlny, čím porušuje stavebné pravidlo č.5. Táto figúra nemože byť považovaná za impulz.



Obr. 2.4: Zlyhanie piatej vlny v impulze



Obr. 2.5: Impulz s predĺženou treťou vlnou



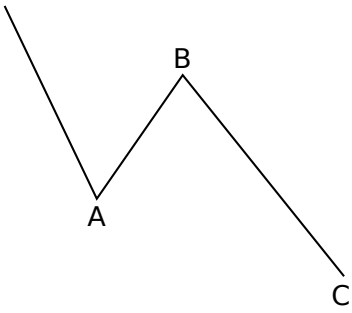
Obr. 2.6: Záverečný impulz. Zdroj: [1]

2.1.3 Korekcia

Korekčné vlny postupujú v opačnom smere ako trend vyššieho stupňa. Prevažne sú tvorené jednou alebo troma monovlnami. V porovnaní s impulzmi sú zložitejšie a menej jednoznačné, napriek tomu je ich identifikácia jednoduchšia, pretože podľa [1], čo nie je impulzom, je korekciou. Rozlišujeme viacero druhov korekčných formácií, ktoré su popísané nižšie.

Cikcak

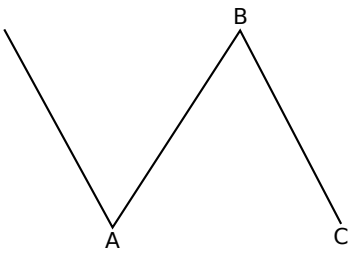
Cikcak formácia má viacero variacií. Najbežnejšie sa vyskytujúcou je normálna, ktorú ukazuje obrázok 2.7, kde vrchol vlny B je nižšie ako začiatok vlny A a vlna C končí nižšie ako vlna A. Ďalšími verziami sú napríklad zkrátená, rozšírená alebo dvojitá cikcak.



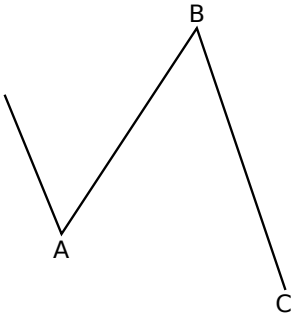
Obr. 2.7: Príklad normálnej cikcak

Rovina

V rovinnej korekcii vlna B končí pri vrchole vlny A a vlna C končí okolo konca vlny A. Ďalším prípadom rovinny je tzv. rozšírená rovina, kde vlna B končí nad začiatkom vlny A a vlna C končí mimo koncovú úroveň vlny A. Príklad roviny a jej rozšírenej formy je ukázaný na obrázkoch 2.8 a 2.9.



Obr. 2.8: Rovina



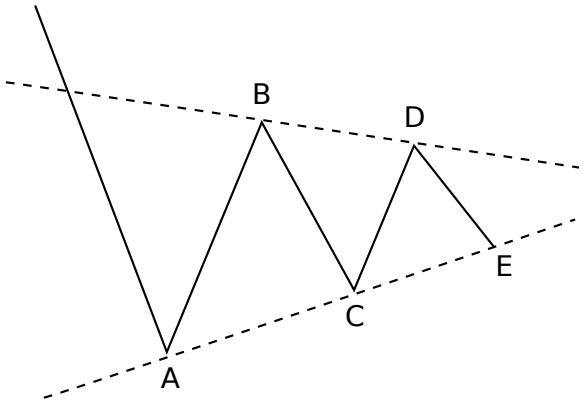
Obr. 2.9: Rozšírena rovina

Trojuholník

Trojuholníková formácia sa skladá z piatich prekrývajúcich sa vĺn, ktoré oscilujú v určitom cenovom pásme. Trojuholník vzniká spojením bodov A-C a bodov B-D vo figúre. Rozlišujeme dva druhy trojuhloníkov: zužujúce a rozširujúce.

Zužujúci sa trojuholník má niekoľko variant: symetrický, zostupný, a vzostupný. Obrázok 2.10, prevzatý z [1], zobrazuje formáciu symetricky sa zužujúceho trojuholníka, kde je línia vrcholov klesajúca, zatiaľ čo línia dna rastie.

Pri rozširujúcom trojuholníku línia vrcholov rastie a línia dna klesá, preto sa nazýva aj obrátene symetrický. Podľa [1], sa takáto figúra vyskytuje len veľmi ojedinele.



Obr. 2.10: Symetricky zužujúci sa trojuholník. Zdroj: [1]

Kombinácia

Kombinácia je zložená z jednoduchších typov korekcií. Napríklad spojením korekčných vĺn v tvare roviny a trojuholníka dostaneme kombinovanú formáciu tzv. dvojitú trojku.

2.1.4 Matematické vzťahy Elliottových vĺn

Fibonacciho postupnosť

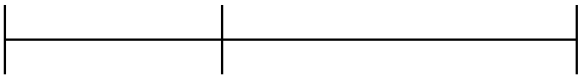
Elliott v svojej knihe Nature's Law označil Fibonacciho postupnosť za matematický základ jeho vlnovej teórie [2]. Pojem Fibonacciho postupnosť v matematike popisuje sekvenciu čísel: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144... Definovať ju môžme ako nekonečnú postupnosť prirodzených čísel, kde prvé dve čísla sú 0 a 1, a každé nasledujúce číslo je súčtom dvoch predchádzajúcich čísel. Rovnica 2.1 udáva jej rekurzívnu definíciu.

$$F(n) = \begin{cases} 0, & \text{pre } n = 0 \\ 1, & \text{pre } n = 1 \\ F(n-1) + F(n-2) & \text{pre } n > 1 \end{cases} \tag{2.1}$$

Zlatý rez

Pojem zlatý rez označuje pomer, ktorý vznikne rozdelením úsečky tak, že pomer väčšej časti k menšej je ekvivalentný pomeru celej úsečky k väčšej časti, viď 2.11. Hodnota tohto pomeru je rovná iracionálnemu číslu:

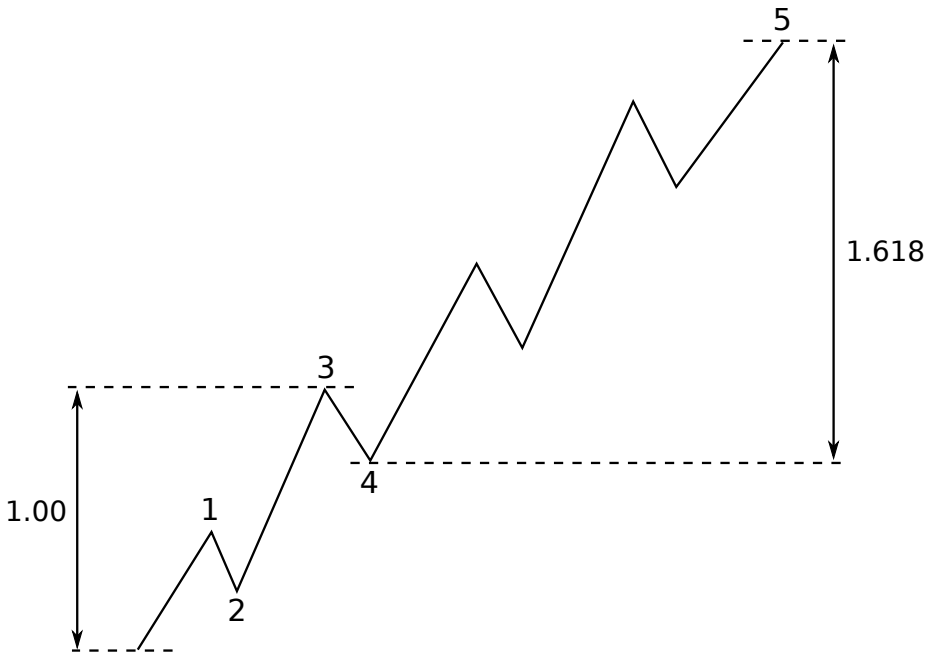
$$\phi = \frac{1+\sqrt{5}}{2} = 1.618033988... \tag{2.2}$$



Obr. 2.11: Zlatý rez

Pomery vĺn v impulze

V impulze majú všetky hybné vlny tendenciu byť voči sebe v pomeroch 1:1, 1:1,618, alebo 1:2,618. Ak je tretia vlna predĺžená, tak výškový pomer prvej a piatej vlny je 1:1 alebo 1:1,618. Pri predĺžení piatej vlny je táto vlna v pomere 1,618:1 k ostatným vlnám. Táto situácia je zobrazená na obrázku 2.12 prebraného z [1]. V prípade, že žiadna z hybných vĺn nie je predĺžená, štvrtá vlna delí výšku impulzu do zlatého rezu.



Obr. 2.12: Predĺžená piata vlna je v pomere 1.618:1 k ostatným vlnám. Zdroj: [1]

2.1.5 Detekcia na základe systému pravidiel

Jeden z možných prístupov k vyhľadávaniu Elliottových vĺn je založený na systéme hierarchických pravidiel, kedy vyhodnocujeme časovú radu postupne od najmenších celkov k väčším. Analýza začína vyhľadaním a vyznačením monovĺn. Potom sú na základe skupiny pravidiel a pomeru medzi monovlnami nájdené figúry vyššieho stupňa a z tých znova ďalšie figúry. Tento postup a všetký pravidlá pre spájanie a konštrukciu vĺn sú bližšie popísané v [1].

Väčšina softvéru určeného pre vyhľadávanie a klasifikáciu Elliottových vĺn funguje práve na základe týchto pravidiel. Jedným z najrozšírenejších programov slúžiacich pre analýzu finančných trhov z pohľadu Elliottovej teórie je Refined Elliott Trader popísaný v [1].

2.2 Umelé neurónové siete

Umelé neurónové siete sú istým nedokonalým modelom myslenia ľudského mozgu [3]. Sú to výpočtové modely, ktoré vznikli na základe abstrakcie funkčných vlastností biologických neurónov a nervového systému. Zvyčajne sa používajú na modelovanie zložitých vzťahov medzi vstupmi a výstupmi a nachádzanie vzorov v dátach.

Ich história siaha do prvej polovice 20. storočia, keď W.S. McCulloch a W. Pits vypracovali model umelého neurónu a F. Rosenblatt vytvoril prvý funkčný perceptrón. Neskôr došlo k objaveniu viacvrstvových sietí a vytvoreniu algoritmu na ich učenie, nazývaného metóda spätného šírenia chyby (*angl.* backpropagation). [3]

V nasledujúcej časti budú stručne predstavené základy umelých neurónových sietí a bližšie priblížené témy, ktoré sú potrebné pre pochopenie praktickej časti.

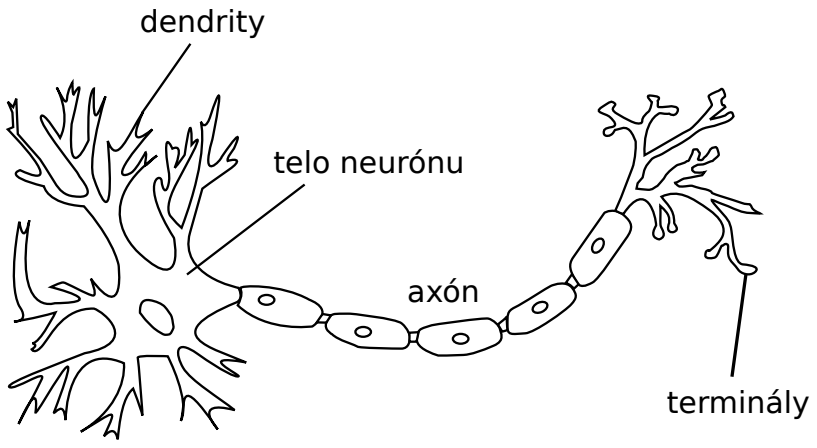
2.2.1 Úvod do neurónových sietí

Táto podkapitpola popisuje teoretické základy neurónových sietí.

Biologický neurón

Biologický neurón, ktorého zjednodušená podoba je zobrazená na obrázku 2.13, je základnou stavebnou jednotkou ľudského centrálneho nervového systému. Ľudský mozog obsahuje približne 10^{10} nervových buniek a 10^{14} synaptických spojení [4]. Štyri základné komponenty nervovej bunky z hľadiska spracovávania, uchovávania a prenosu informácií sú:

- **Dendrity** sú výbežky nervových buniek, ktoré prijímajú a vedú nervové vzruchy smerom do tela neurónu
- **Telo neurónu (soma)** je zodpovedné za spracovanie vstupných informácií (nervových vzruchov) z dendritov
- **Axón** prenáša výstup tela k iným neurónom cez synaptické spojenia
- **Synaptické spojenia** zabezpečujú funkčné spojenie medzi jednotlivými neurónmi



Obr. 2.13: Biologický neurón. Zdroj: [5]

Nervové bunky si predávajú informácie vo forme elektrických a chemických procesov [3]. Vstupný elektrický potenciál je privádzaný dedritmi do tela neurónu, kde sa spracuje.

Ak presiahne určitú hodnotu, bunka sa aktivuje a vysiela signál iným nervovým bunkám prostredníctvom axónu a synaptických spojení.

Na základe biologickej interpretácie funkcie neurónu bola vytvorená jeho jednoduchšia varianta matematickej interpretácie neurónu, nazývaná umelý neurón [3].

Umelý neurón

Umelý neurón je matematický model, ktorý tvorí základný prvok umelých neurónových sietí. Je založený na princípoch biologického neurónu, ale jeho vnútorna štruktúra a funkcia je omnoho jednoduchšia. Podľa [6], najčastejšie používaným modelom neurónu je neurón typu McCulloch a Pitts, ktorý je ilustrovaný na obrázku 2.14. Každý takýto neurón pozostáva z dvoch častí: sieťovej funkcie a aktivačnej funkcie.

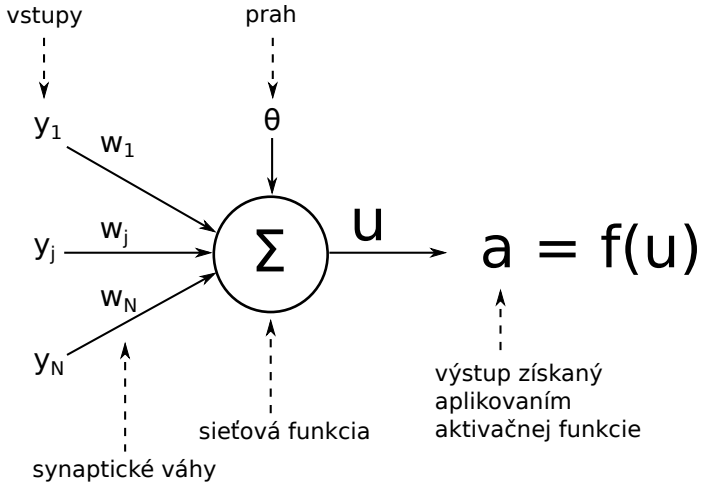
Sieťová funkcia udáva spôsob, akým sú jednotlivé vstupy zkombinované. Najčastejšie používanou je vážená lineárna kombinácia popísaná rovnicou [6]:

$$u = \sum_{j=1}^{N} w_j y_j + \theta, \tag{2.3}$$

kde N je počet vstupov,$\{y_j; 1 <= j <= N\}$ sú vstupy neurónu, $\{w_j; 1 <= j <= N\}$ sú synaptické váhy a parameter θ je prahová hodnota, tiež nazývaná bias.

Výstup neurónu je výsledkom lineárnej alebo nelineárnej transformácie nazývanej aktivačná funkcia, ktorej vstupom je výsledok sieťovej funkcie. Zmyslom použitia aktivačnej funkcie je modifikácia úrovne výstupu na normované hodnoty. Bez použitia takejto transformácie by výstup mohol dosiahnuť vysokých hodnôt, čo by mohlo spôsobiť problémy hlavne u viacvrstvových neurónových sieťach [3]. Existuje niekoľko druhov aktivačných funkcií. Tie najčastejšie používané sú uvedené v tabuľke 2.2 spolu s ich deriváciami, ktoré sa využívaju pri učení siete. Tabuľka je prevzatá z [6].

Perceptrón je najjednoduchšia neurónová sieť obsahujúca len jeden umelý neurón s prahovou aktivačnou funkciou. Používa sa ako primitívny lineárny klasifikátor, ktorý je schopný riešiť problémy lineárne separovateľné. [6]



Obr. 2.14: Model umelého neurónu typu McCulloch a Pitts

Názov aktivačnej funkcie	**Predpis** $f(u)$	**Derivácia** $\frac{\mathrm{d}f(u)}{\mathrm{d}u}$
Štandardná sigmoida	$f(u) = \frac{1}{1+e^{-u}}$	$f(u)[1 - f(u)]$
Hyperbolický tangens	$f(u) = \tanh(u)$	$1 - [f(u)]^2$
Lineárna funkcia	$f(u) = au + b$	a
Prahová funckia	$f(u) = \begin{cases} 1, & \text{pre } u > 0 \\ -1, & \text{pre } u < 0 \end{cases}$	neexistuje pre $u = 0$

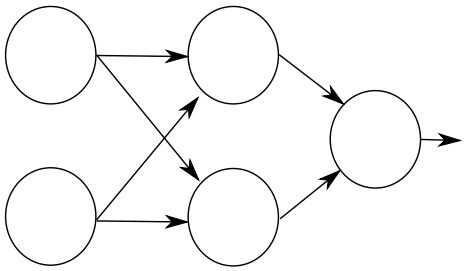
Tabuľka 2.2: Zoznam najpoužívanejších aktivačných funkcií a ich derivácií. Zdroj: [6]

Topológia neurónových sietí

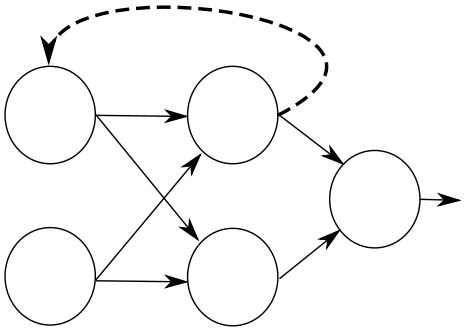
Topológia popisuje spôsob, akým sú neuróny poprepájané, aby zformovali sieť. Spôsob prepojenia jednotlivých neurónov v sieti hraje fundamentálnu rolu vo funkcionalite a výkonne neurónovej siete. Podľa smeru toku dát rozlišujeme siete s dopredným šírením (acyklické) a rekurentné siete (cyklické).

V sieti s acyklickou topológiou sa dáta šíria striktne smerom vpred od vstupných neurónov k výstupným, viď obrázok 2.15. Nenachádzajú sa tu žiadne cykly a spätné väzby, to znamená, že neexistujú žiadne prepojenia výstupov neurónov so vstupmi neurónov v rovnakej alebo predošlej vrstve.

Cyklická neurónová sieť obsahuje aspoň jednu spätnú väzbu, viď obrázok 2.16. Vzhľadom k tejto spätnej väzbe, tieto siete modelujú nelineárny dynamický systém, ktorý obsahuje vnútornú pamäť. Rekurentné siete vykazujú komplexné správanie a sú cieľom aktívneho výzkumu v oblasti umelých neurónových sietí [6].



Obr. 2.15: Sieť s dopredným šírením.



Obr. 2.16: Rekurentná sieť, spätná väzba je vyznačená prerušovanou čiarou.

Učenie neurónovej siete

Schopnosť učiť sa je jedna z najdôležitejších vlastností neurónových sietí. Vo všeobecnosti sa učenie delí na učenie s učiteľom a učenie bez učiteľa.

Pri učení s učiteľom sú sieti poskytnuté dáta, ktoré obsahujú množinu vstupov aj množinu požadovaných cieľových výstupov. Počas tréningu sa sieť snaží upraviť svoje váhy podľa trénovacieho algoritmu tak, aby jej výstupy zodpovedali požadovaným cieľovým hodnotám.

Ak je sieť učená bez učiteľa, tak množina trénovacích dát neobsahuje cieľové hodnoty. Požadovaný výstup siete je teda neznámy. Sieť sa v tomto prípade snaží v dátach nájsť

nejakú skrytú štruktúru, napríklad zhlukovaním jednotlivých vzorov.

Ďalším rozdielom je rozlišovanie tréningovej a prevádzkovej fázy. Pri učení s učiteľom sú tieto fázy oddelené, zatiaľ čo pri učení bez učiteľa prebiehajú súčasne, sieť sa teda učí aj počas riešenia konkrétneho problému [7].

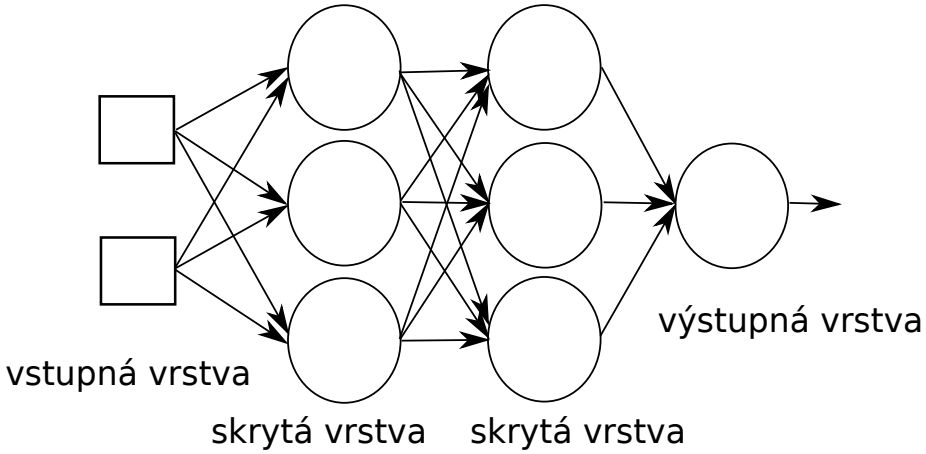
Preučenie (*angl.* overfitting) nastáva, keď sieť pri tréningu začne modelovať náhodný hluk v dátach, namiesto požadovaných vzťahov medzi nimi. V tom prípade si sieť vedie dobre pri tréningu, ale na iných dátach stráca schopnosť zovšeobecňovania. Väčšinou sa vyskytuje v situáciach, keď je vnútorná štruktúra neurónovej siete príliš komplexná vzhľadom na veľkosť množiny trénovacích dát.

2.2.2 Viacvrstvová sieť perceptrónov

Viacvrstvová sieť perceptrónov (angl. multilayer perceptron), je sieť s dopredným šírením. Pozostáva z viacerých vrstiev neurónov a každá vrstva je vo väčšine prípadov plne prepojená s nasledujúcou.

Táto sieť poskytuje možnosť nelineárneho mapovania medzi jej vstupmi a výstupmi. S dostatočným počtom neurónov s nelineárnou aktivačnou funkciou v skrytej vrstve je viacvrstvová sieť perceptrónov použiteľná ako univerzálny prostriedok pre aproximáciu funkcií [6].

Obrázok 2.17 zobrazuje konfiguráciu viacvrstvovej siete perceptrónov s troma vrstvami. Kruhy predstavujú jednotlivé neuróny, ktoré sú organizované vo výstupnej a skrytých vrstvách. Tie sa nazývajú skryté, pretože ich výstup je šírený len do vyššej vrstvy a teda je užívateľovi ukrytý. Vstupná vrstva väčšinou nie je implementovaná neurónmi [6]. Slúži len na prenos vstupných hodnôt do neurónov ďalšej vrstvy.



Obr. 2.17: Trojvrstvová sieť perceptrónov s dvoma skrytými vrstvami obsahujúcimi po tri neuróny a vstupnou vrstvou tvorenou dvoma neurónmi

Pre učenie viacvrstvových sietí existuje množstvo rôznych metód. Najpopulárnejšou a najpoužívanejšou je metóda spätného šírenia chyby (*angl.* error backpropagation). Tá je založená na metóde najväčšieho spádu (*angl.* steepest gradient descent method). Nasleduje jej krátky popis vychádzajúci z [6].

Algoritmus hľadá také nastavenie váh, aby minimalizoval chybu E na K vzoroch:

$$E = \sum_{k=1}^{K} [d(k) - z(k)]^2, \tag{2.4}$$

kde $d(k)$ je požadovaná hodnota a $z(k)$ je výstupom siete.

Nech w_{ij} označuje váhu v sieti od neurónu j k neurónu i a t označuje epochu tréningu. Novú vstupnú váhu v epoche $t+1$ vypočítame ako

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t), \tag{2.5}$$

kde Δw_{ij} je veľkosť zmeny váhy. Keďže ide o gradientovú metódu pre veľkosť zmeny platí:

$$\Delta w_{ij} = -\gamma \frac{\partial E}{\partial w_{ij}}, \tag{2.6}$$

kde γ je rýchlosť učenia (*angl.* learning rate) a $\frac{\partial E}{\partial w_{ij}}$ je derivácia chyby vzhľadom na váhu w_{ij}.

Bližší popis tejto metódy a odvodenia jednotlivých vzorcov sa nachádzajú v [6].

2.2.3 Algoritmus pružného spätného šírenia chyby

Algoritmus pružného spätného šírenia chyby je jednou z variant trénovacieho algoritmu spätného šírenia vo viacvrstvových neurónových sieťach s dopredným šírením. Hlavným rozdielom je, že veľkosť zmeny váhy nie je stanovovená na základe veľkosti parciálnej derivácie $\frac{\partial E}{\partial w_{ij}}$, ale jej znamienka. Výhodou tohto prístupu je podľa [8]:

- je rýchlejši oproti iným metódam určeným pre učenie s učiteľom
- robustný vzhľadom na jeho vnútorné parametre
- je metódou prvého rádu, teda časová a priestorová zložitosť sa stupňuje len lineárne s počtom optimalizovaných parametrov

V nasledujúcej časti, ktorá vychádza z [8], je bližšie popísaný algoritmus pružného spätného šírenia chyby bez spätného sledovania váhy.

Nech w_{ij} označuje váhu v sieti od neurónu j k neurónu i, E je chyba dosiahnutá na množine vzorov a t označuje epochu tréningu. Potom v každej epoche sú nové váhy vypočítané ako

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t). \tag{2.7}$$

Priebeh jednej epochy učenia môže byť rozdelený do dvoch častí. V prvej fáze je veľkosť kroku Δ_{ij} individuálne nastavená pre každú z váh podľa pravidla:

$$\Delta_{ij}(t) \begin{cases} \min(\eta^+ \Delta_{ij}(t-1), \Delta_{max}) & \text{ak } \frac{\partial E}{\partial w_{ij}}(t-1) \cdot \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ \max(\eta^- \Delta_{ij}(t-1), \Delta_{min}) & \text{ak } \frac{\partial E}{\partial w_{ij}}(t-1) \cdot \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ \Delta_{ij}(t-1) & \text{inak,} \end{cases} \tag{2.8}$$

kde $0 < \eta^- < 1 < \eta^+$ a parametre Δ_{min} a Δ_{max} ohraničujú veľkosť kroku.

V druhej fáze je určený smer a veľkosť zmeny váhy $\Delta w_{ij}(t)$ na základe znamienka parcialnej derivácie $\frac{\partial E}{\partial w_{ij}}$ a veľkosti kroku $\Delta_{ij}(t)$ podľa rovnice:

$$\Delta w_{ij}(t) = -\text{sign}\left(\frac{\partial E}{\partial w_{ij}}(t)\right) \Delta_{ij}(t). \tag{2.9}$$

2.2.4 Committee machine

Keďže neurónové siete pracujú s určitou chybou klasifikácie, pri ich aplikácii je bežnou praxou vytvoriť a natrénovať viacero kandidátov. Následne vybrať jedného, ktorý dosahoval najlepšie výsledky na sade testovacích dát, ktorá je nezávislá od množiny dát trénovacích. Avšak to nám nezaručuje, že vybraná sieť bude dosahovať najlepšie výsledky aj na nových testovacích dátach, čo je veľkou slabinou tohto prístupu. Ďalšou nevýhodou je množstvo premárneného času na trénovanie sietí, ktoré v konečnom riešení nepoužijeme. Podľa [9], tieto nedostatky môžu byť prekonané sformovaním množiny sietí do určitej komisie, odborne nazývanej committee machine. Jednotlivým neurónovým sieťam v committee machine je daný rovnaký vstup a ich individuálne výsledky sú skombinované do jedného finálneho výstupu. Význam takéhoto prístupu spočíva v tom, že môže viesť k dosiahnutiu výrazného zlepšenia pri generalizácii, za potreby minimálneho výpočtového výkonu naviac [9]. Nasledujúce časti vychádzajú z [6].

Jednoduché priemerovanie a hlasovanie

Pri tomto prístupe sú všetci členovia committee machine trénovaní na kompletnej sade trénovacích dát. Dekorelácia je dosiahnutá tým, že siete s rôznymi začiatočnými podmienkami pri trénovaní konvergujú do rôzneho lokálneho minima. Spôsob kombinácie výstupov sa líši podľa riešenej úlohy.

Pri probléme regresie sa používa metóda jednoduchého priemerovania. Výstup committee machine je vypočítaný ako vážený súčet výstupov jednotlivých členov:

$$\widehat{t}(x) = \sum_{i=1}^{M} g_i f_i(x), \tag{2.10}$$

kde M je počet členov komisie, $f_i(x)$ je výstup i-teho člena a g_i sú váhy, ktoré sú nastavené na $g_i = 1/M$, ak chceme jednotlivé výstupy len spriemerovať.

V prípade klasifikácie je kombinácia výstupov realizovaná ako hlasovanie. Vstupný vzor je priradený do triedy, ktorá získa väčšinu hlasov:

$$\widehat{class}(x) = \arg\max_{j} \sum_{i=1}^{M} g_i f_{i,class=j}(x), \tag{2.11}$$

kde $f_{i,class=j}(x)$ je výstup i-teho člena komisie pre triedu j.

Bagging

Základom baggingu je trénovanie jednotlivých neurónových sietí na rôznych dátach. Problémom však je potreba veľkého množstva trénovacích dát, ktoré väčšinou v praxi nie sú k dispozícii. Riešením je, pre každého člena komisie náhodne vybrať množinu dát z originálnej sady, na ktorej bude trénovaný. Výstup takto vytvorenej komisie je potom možné získať jednoduchým priemerovaním alebo hlasovaním v prípade klasifikačného problému. Podľa [6], expirimentálne výsledky nasvedčujú, že bagging vo všeobecnosti prekonáva metódy jednoduchého priemerovania a hlasovania.

Kapitola 3

Aplikácia na detekciu vĺn

Táto kapitola sa zaoberá praktickou časťou tejto práce. Popisuje návrh modelu, použité technológie a implementáciu programu, ktorý je schopný detekovať Elliottove vlny na zadanej časovej rade pomocou umelých neurónových sietí.

Samotná demonštračná aplikácia je rozdelená na dva logické celky. Prvou časťou je tréner, ktorý slúži na vytvorenie a natrénovanie nových neurónových sietí. Tie potom môžu byť pridané k sade sietí používanej pre detekciu vĺn.

Druhú časť predstavuje samotný detekčný systém. Jeho vstupom je statická časová rada reprezentujúca hodnoty niektorého z finančných trhov. Ďalším vstupom je množina natrenovaných neurónových sietí, ktoré sú zapojené do jednoduchej committee machine.

3.1 Návrh trénera

Úlohou trénera je vytvorenie plne prepojenej viacvrstvovej siete perceptrónov s dopredným šírením a jej natrénovanie na sade trénovacích dát.

3.1.1 Štruktúra siete

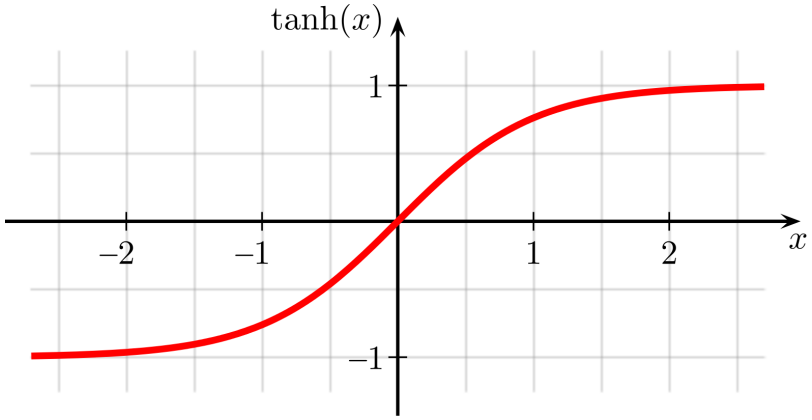
Vstup siete predstavujú pomery dĺžok jednotlivých susediacich vĺn v impulze. Vstupnú vrstvu preto tvoria štyri neuróny. V rámci vnútornej štruktúry siete je dôležitý počet neurónov v skrytej vrstve. Pre dosiahnutie lepšej dekorelácie výstupov jednotlivých sietí v committee machine je tento počet náhodne vybraný z intervalu $< \frac{2}{3}N,\ 2N >$, kde N je počet neurónov vo vstupnej vrstve. Ako nelineárna aktivačná funckia pre neuróny v skrytej vrstve je používaný hyperbolický tangens:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{3.1}$$

vykreslený na obrázku 3.1. Podľa [9], bolo empiricky zistené, že má oproti štandardnej sigmoide malú praktickú výhodu v tom, že pri použití hyperbolického tangensu, trénovacie algoritmy konvergujú rýchlejšie. Výstupná vrstva je tvorená jedným neurónom so štandardnou sigmoidou ako aktivačnou funkciou. To znamená, že výstupy jednotlivých sietí sú v intervale $< 0, 1 >$.

3.1.2 Učenie siete

Učenie prebieha s učiteľom. Použitý je algoritmus pružného spätného šírenia chyby bez spätného sledovania váh, ktorý konverguje omnoho rýchlejšie ako klasický *backpropagation*.



Obr. 3.1: Hyperbolický tangens daný (3.1). Zdroj: [10]

Jednotlivé siete sú v rámci baggingu trénované na rôznych trénovacích sadách, ktoré sú náhodne vybrané zo vstupnej množiny dát.

Ako nástroj proti preučeniu siete slúži podmienka pre zastavenie trénovania v prípade, že začne stúpať validačná chyba. Teda chyba zistená na množine validačných dát, ktorá je nezávislá od množiny dát trénovacích. Vyjadríme ju ako strednú kvadratickú chybu popísanú rovnicou:

$$E = \frac{1}{N} \sum_{i=1}^{N} (d_i - y_i)^2, \tag{3.2}$$

kde N je počet vzorov v množine, d_i je požadovaný výsledok a y_i je výstupom siete pre vzor i.

Zvolené bolo zastavovacie kritérium navrhnuté v [8]:

$$GL(t) = 100 \cdot \left(\frac{E_{va}(t)}{E_{opt}(t)} - 1 \right), \tag{3.3}$$

kde $GL(t)$ označuje stratu generalizácie (*angl.* generalization loss) v epoche t, E_{va} je validačná chyba v epoche t, a E_{opt} je najmenšia validačná chyba počas tréningu do epochy t vypočítaná ako:

$$E_{opt}(t) = \min_{t' \leq t}(E_{va}(t')). \tag{3.4}$$

V prípade, že GL prekročí určitú hranicu α, učenie je zastavené a používa sa konfigurácia siete, pri ktorej bola validačná chyba počas tréningu minimálna.

Trénovacie dáta

Počas tvorby tejto práce nebol k dispozícii zdroj kvalitných dát, ktoré by mohli byť použité na trénovanie sietí. Z tohto dôvodu sa na učenie sietí používa obmedzená sada dát, ktorá bola vytvorená jednoduchým generátorom vzorov. Ten na základe stavebných pravidiel pre impulzné figúry uvedených v podkapitole 2.1.2 vygeneroval vzory impulzov a neimpulzov.

3.2 Návrh detekčného systému

Úlohou detekčného systému je načítať a rozdeliť vstupnú časovú radu na množinu segmentov, na ktorých potom committee machine, pozostávajúca z viacvrstvových neurónových

sietí detekuje impulzné figúry. Detekcia bola obmedzená len na vyhľadávanie impulzov z troch dôvodov:

- impulz je hlavným predstaviľom hybného režimu a je najdôležijšou figúrou
- korekcie sú jednoducho odvoditeľné, ako bolo uvedené v 2.1.3, čo nie je impulzom je korekciou
- nedostatok kvalitných a správne klasifikovaných trénovacích dát

3.2.1 Predspracovanie dát

V rámci analýzy statickej časovej rady je nevyhnutné zamerať sa na hlavné trendové pohyby a vyhnúť sa jemným fluktuáciam ceny. Preto je potrebné zvoliť vhodný spôsob kompresie dát, ktorý tieto nepodstatné výkyvy nebude brať do úvahy. Bola zvolená metóda dôležitých bodov, prezentovaná v [11], ktorá analyzuje časovú radu a vyhľadáva lokálne extrémy, pre ktoré platia určité podmienky. Veľkosť kompresie dát je možné ovládať parametrom R. Čím je jeho hodnota vyššia, tým menej bodov je na časovej rade vyselektovaných.

Bod a_m je dôležitým minimom, ak existujú indexy i a j, kde $i \leq m \leq j$ a platí:

- a_m je minimom na intervale $< a_i, a_j >$
- $a_i/a_m \leq R$ a $a_j/a_m \leq R$

Definícia dôležitého maxima je symetrická. To znamená, že bod a_m je dôležitým maximom, ak existujú indexy i a j, kde $i \leq m \leq j$ a platí:

- a_m je maximom na intervale $< a_i, a_j >$
- $a_m/a_i \leq R$ a $a_m/a_j \leq R$

Normalizácia dát

Cieľom normalizácie je izolovať štatistickú chybu v dátach. Aplikovaním takejto lineárnej transformácie dosiahneme, že jednotlivé vstupy siete budú mať podobné hodnoty. Normalizovanú hodnotu $\widetilde{x_i}$ získame ako

$$\widetilde{x_i} = \frac{x_i - \mu}{\sigma}, \tag{3.5}$$

kde x_i je hodnota, ktorú chceme normalizovať, μ je stredná hodnota daná rovnicou 3.7 a σ je smerodatná odchýlka vypočítaná ako

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}. \tag{3.6}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3.7}$$

Výsledkom takejto lineárnej transformácie je, že dáta majú nulovú strednú hodnotu a rozptyl je rovný jednej.

3.2.2 Committee machine

Pozostáva z množiny sietí vytvorených a natrénovaných trénerom, ktorý je popísaný v podkapitole 3.1. Keďže výstupnú vrstvu sietí tvorí jeden neurón, ktorý ako aktivačnú funckiu používa štandardnú sigmoidu, jeho výstupy ležia v intervale $< 0, 1 >$. Túto výstupnú hodnotu je preto možné chápať ako pravdepodobnosť, že sa na skúmanom segmente nachádza impulzná figúra. Committee machine výstupy jednotlivých sietí spriemeruje. Jej výstup je možné vyjadriť ako

$$p = \frac{1}{N} \sum_{i=1}^{N} f_i(x), \tag{3.8}$$

kde N je počet neurónových sietí tvoriacich committe machine, $f_i(x)$ je výstup i-teho člena a p je výsledná pravdepodobnosť výskytu impulzu. Segmenty u ktorých táto pravdepodobnosť prekročí určitú hranicu sú prehlásené za impulzy.

3.2.3 Výstupný modul

Výstupný modul má za úlohu previesť výsledky z committee machine do zrozumiteľnej podoby. Pri probléme detekcie vzorov na časovej rade je vhodným riešením vykreslenie grafu s nájdenými vzormi. Výstupom detekčného systému je teda graf vstupnej časovej rady, na ktorom sú vykreslené jednotlivé detekované impulzy a vyznačená ich vnútorná štruktúra.

3.3 Použité technológie

Táto podkapitola v krátkosti predstavuje technológie, ktoré boli použité pri implementácii navrhnutého systému.

Python[12]

Python je objektovo orientovaný, interpretovaný programovací jazyk. Je platformovo nezávislý a je štandardným komponentom mnohých operačných systémov ako Windows, Linux/Unix a Mac OS X. Python je taktiež použiteľný ako rozširujúci jazyk pre aplikácie vytvorené v iných programovacích jazykoch, ktoré potrebujú ľahko využiteľné skriptovanie. Výhodou je dobrá dostupnosť knižníc, ktoré značne uľahčujú implementáciu často riešených úloh. Napríklad knižnice ako NumPy, SciPy a matplotlib rozširujú možnosti jazyka Python aj na vedecké použitie.

PyBrain[13]

PyBrain je skratka pre *Python-Based Reinforcement Learning, Artificial Intelligence and Neural Network Library*. Je to modulárna knižnica jazyka Python, ktorá ponúka ľahko použiteľné, flexibilné a výkonné algoritmy určené pre strojové učenie. Vo veľkej miere sa zameriava na prácu s umelými neurónovými sieťami a implementuje množstvo metód pre učenie s učiteľom alebo bez.

matplotlib [14]

Matplotlib je knižnica v jazyku Python, ktorá slúži na vykresľovanie dvojrozmerných grafov z polí. Pôvod má v emulovaní grafických príkazov MATLABU, napriek tomu je od neho

nezávislá a môže byť používaná objektovo orientovaným spôsobom. Pre dobrú výkonnosť aj pri práci s nadmerne rozmernými poliami vo veľkej miere využíva modul NumPy. Kód knižnice je konceptuálne rozdelený do troch častí:

- **rozhranie pylab** umožňuje vytváranie grafov, pomocou kódu pomerne podobného MATLABu
- **matplotlib frontend** predstavuje skupinu tried, ktoré zabezpečujú vytváranie a ovládanie grafov.
- **backend** transformuje dáta z frontendu na výstupné zariadenie

3.4 Implementácia trénera

Trénovacie dáta sú načítané zo súboru pomocou modulu `pickle` a majú takúto štruktúru:

```
[0.61511512348, 0.520648596567, -1.7309766132, 0.59521289317, 0]
[-0.769503694367, 1.62716433459, -0.871780320501, 0.0141196802760, 1]
[-0.756609791416, -0.333387687274, -0.621694669766, 1.71169214845, 1]
[-1.07974598621, 0.880127249923, -0.91004230642, 1.10966104271, 0]
[-1.01055029204, 1.32639306369, -0.922640687148, 0.60679791549, 1]
```

Každý riadok reprezentuje jeden trénovací vzor. Prvé štyri hodnoty predstavujú normalizované pomery vĺn, piata hodnota je požadovaný výstup siete. Teda jednotka znamená, že na danom vzore sa nachádza impulz a nula, že daný vzor nespĺňa podmienky impuzlnej figúry.

Z načítanej sady dát sú náhodne vybrané vzory a rozdelené do dvoch množín. Jedna z nich je určená na trénovanie, druhá na určovanie validačnej chyby pri tréningu. Množiny dát sú reprezentované objektom typu `SupervisedDataSet`.

K vytvoreniu neurónovej siete je použitá metóda `BuildNetwork()`. Pomocou jej parametrov je možné jednoducho určiť štruktúru siete, počet neurónov v jednotlivých vrstvách a ich aktivačné funckie.

Na učenie neurónových sietí je použítá trieda `RPropMinusTrainer` z knižnice PyBrain. Ide o triedu, ktorá pomocou algoritmu pružného spätného šírenia chyby, viď pseudokód 3.1, trénuje sieť na zvolenej množine trénovacích dát.

Samotný tréning vykonáva funkcia `trainer.train()`, ktorá upraví váhy a posunie sieť do ďalšej epochy. Po každej iterácii je vypočítaná nová validačná chyba pomocou funkcie `trainer.testOnData()`, ktorá vracia strednú kvadratickú chybu popísanú rovnicou 3.2. Na obrázku 3.2 je možné vidieť príklad vývoja trénovacej a validačnej chyby počas tréningu.

Celý proces učenia siete riadi jednoduchý konečný automat, ktorý tréning ukončí v prípade, ak bol vykonaný vopred stanovený počet iterácií, bola dosiahnutá cieľová chybovosť siete alebo v prípade, že je zaznamenaná strata generalizácie, teda ak GL prekročí hraničnú hodnotu α ako bolo uvedené v 3.1.2.

Keďže v knižnici PyBrain nie je možnosť sa spätne vrátiť k predošlým váham, neurónová sieť je pri dosiahnutí novej minimálnej validačnej chyby E_{opt}, uložená do súboru pomocou modulu `pickle`. Pri ukončení tréningu potom používame vždy konfiguráciu siete, ktorá dosiahla minimálnu chybovosť na validačnej množine dát.

Pseudokód 3.1 Algoritmus pružného spätného šírenia chyby bez spätného sledovania váh. Zdroj: [15]

for all w_{ij} **do**

if $\frac{\partial E}{\partial w_{ij}}(t-1) \cdot \frac{\partial E}{\partial w_{ij}}(t) > 0$ **then**

$\Delta_{ij}(t) = \min(\eta^+ \Delta_{ij}(t-1), \Delta_{max})$

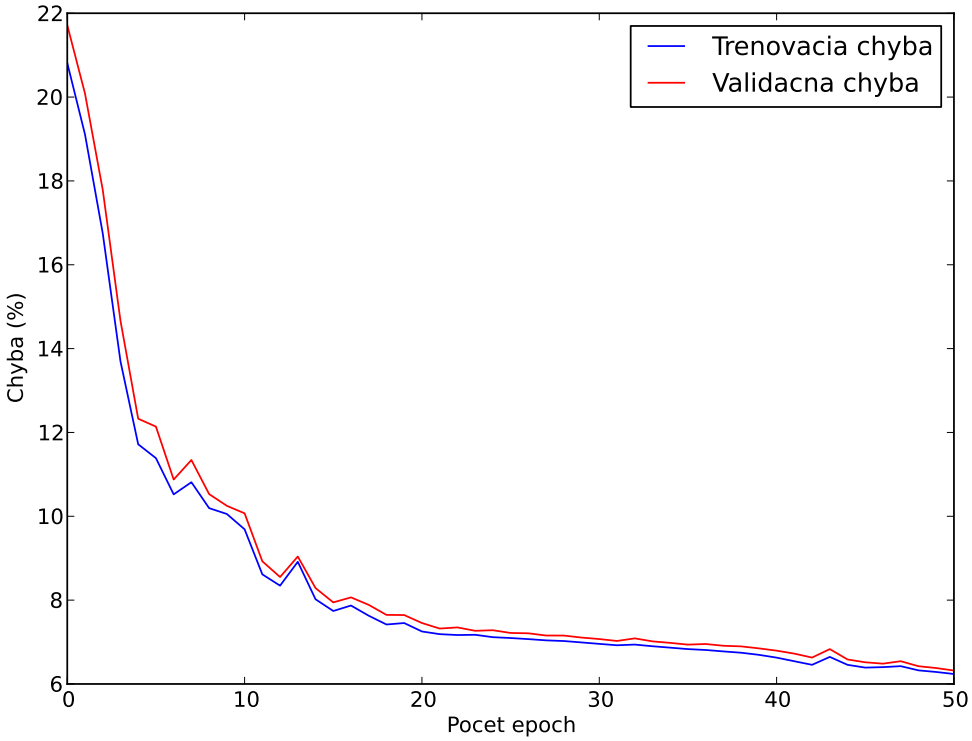
else if $\frac{\partial E}{\partial w_{ij}}(t-1) \cdot \frac{\partial E}{\partial w_{ij}}(t) < 0$ **then**

$\Delta_{ij}(t) = \max(\eta^- \Delta_{ij}(t-1), \Delta_{min})$

end if

$w_{ij}(t+1) = w_{ij}(t) - \text{sign}\left(\frac{\partial E}{\partial w_{ij}}(t)\right) \cdot \Delta_{ij}(t)$

end for



Obr. 3.2: Ukážka vývoja trénovacej a validačnej chyby pri tréningu siete

3.5 Implementácia detekčného systému

Na začiatku program zo súboru načíta zadanú časovú radu. Množina natrénovaných neurónových sietí je zo súboru načítaná modulom `pickle`.

Pre predspracovanie dát slúži trieda `Preprocessor`. Jej metóda `createSegments()` vyhľadá lokálne extrémy na základe algoritmu popísaného pseudokódom 3.2. Na obrázkoch 3.3 a 3.4 je možné vidieť nájdené lokálne minimá a maximá, ktoré sú na grafe vyznačené červenými bodmi. Z týchto bodov sú následne vytvorené jednotlivé segmenty, ktoré sú uložené ako objekty triedy `Segment`.

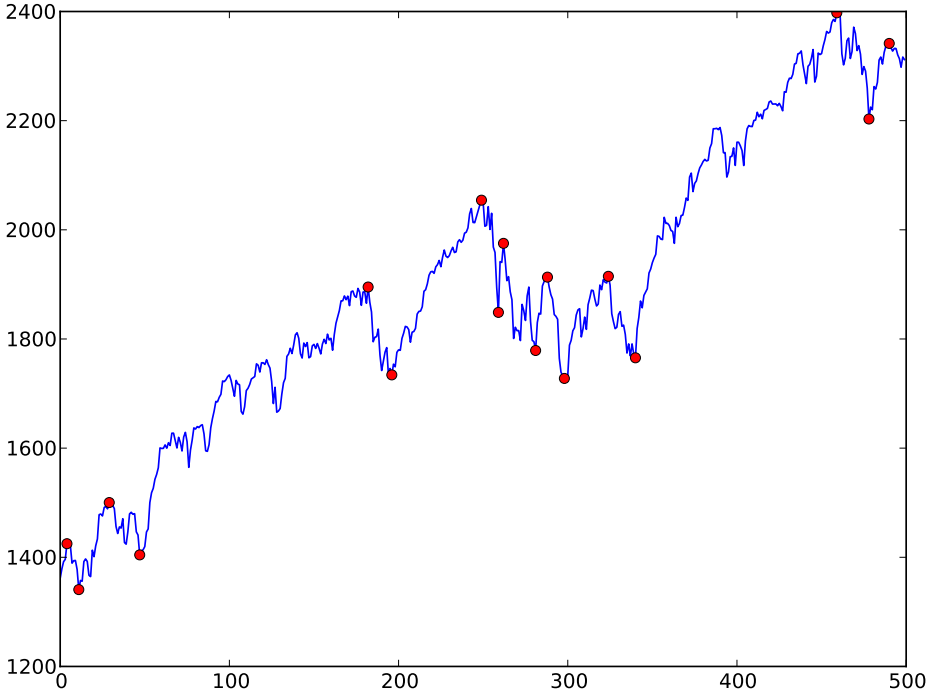
Tieto objekty majú uložený údaj o svojej polohe v rámci časovej rady. Taktiež obsahujú informáciu o svojej vnútornej štruktúre, z ktorej sú vytvorené vstupy pre neurónové siete v committee machine. Vstupy sú navzorkované na požadovanú veľkosť, teda na počet neurónov vstupnej vrstvy a normalizované pomocou metódy `normalize()`. Tá je súčasťou triedy `Preprocessor`.

Committee machine je reprezentovaná triedou `CommitteeMachine`. Obsahuje zoznam načítaných neurónových sietí, ktoré sú do objektu pridané metódou `addMember()`. Na detekciu slúži metóda `checkSegments()`, ktorá ako vstupný parameter očakáva pole vytvorených segmentov. Všetky segmenty sú prejdené množinou sietí a ich výstupy sú sprieme-rované. U segmentov, na ktorých výsledok prekročil určitú hranicu je nastavený príznak detekovaného impulzu metódou `setDetected()`.

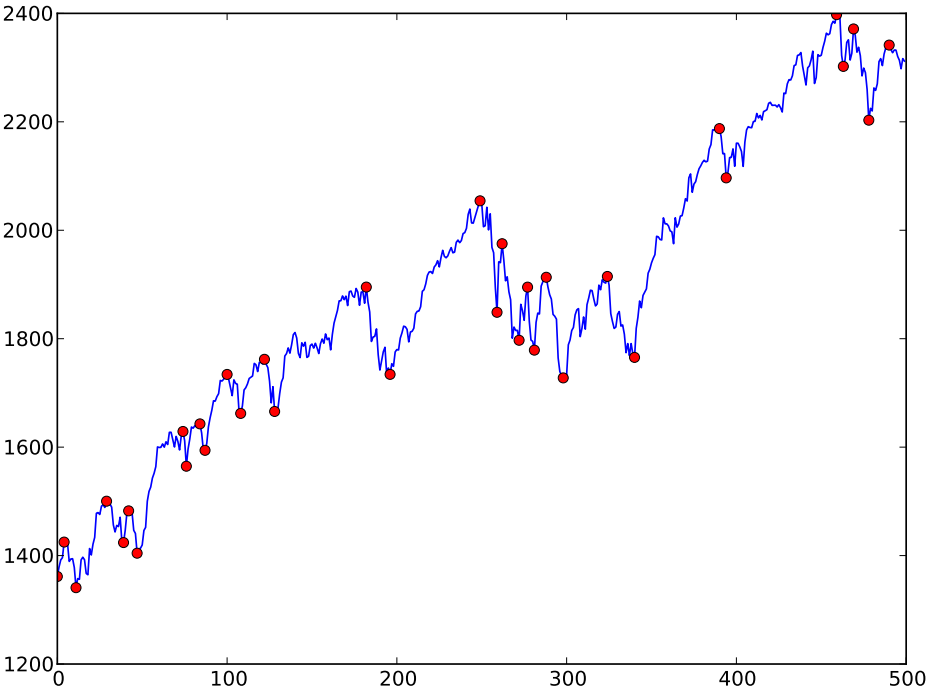
Po ukončení analýzy je zoznam segmentov predaný výstupnému objektu typu `Output`. Ten obsahuje metódu `showGraph()`, ktorá vykreslí zadanú časovú radu spolu s detekovanými impulzmi pomocou knižnice matplotlib a funckie `plot()`.

Pseudokód 3.2 Dôležité body. Zdroj: [11]

```
function IMPORTANT_POINTS
    i = FIND_FIRST_TWO
    if i < N and a_i > a_1 then
        i = FIND_MINIMUM(i)
    end if
    while i < N do
        i = FIND_MAXIMUM(i)
        if i < N then
            i = FIND_MINIMUM(i)
        end if
    end while
end function
function FIND_FIRST_TWO
    imin = 0
    imax = 0
    while i < N and a_imax/a_i < R and a_i/a_imin < R do
        if a_i > A_imax then imax = i
        end if
        if a_i < A_imin then imin = i
        end if
        i = i + 1
    end while
    uloz imin, imax
    return i
end function
function FIND_MINIMUM(i)
    imin = i
    while i < N and a_i/a_imin < R do
        if a_i < A_imin then imin = i
        end if
        i = i + 1
    end while
    uloz imin
    return i
end function
function FIND_MAXIMUM(i)
    imax = i
    while i < N and a_imax/a_i < R do
        if a_i > A_imax then imax = i
        end if
        i = i + 1
    end while
    uloz imax
    return i
end function
```



Obr. 3.3: Nájdené lokálne extrémy pre R = 1.06



Obr. 3.4: Nájdené lokálne extrémy pre R = 1.03

Kapitola 4

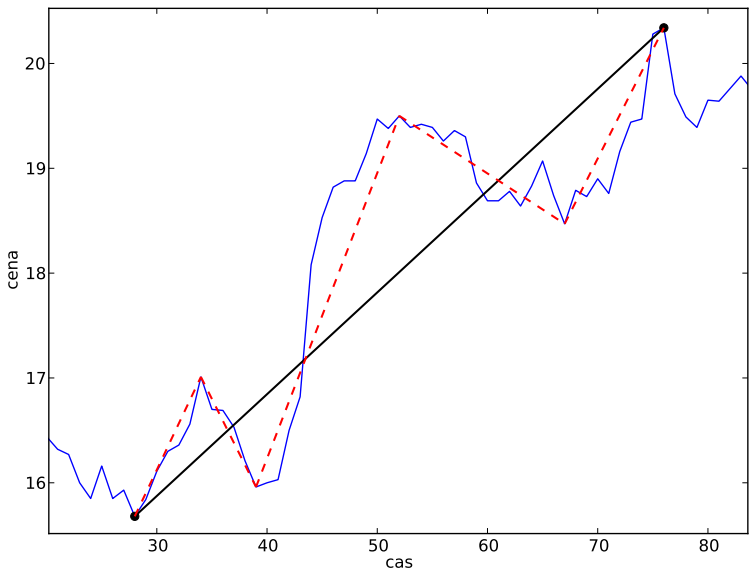
Výsledky

Táto kapitola prezentuje výsledky dosiahnuté navrhnutým systémom. Použité konštanty boli určené na základe testov a experimentovania s programom.

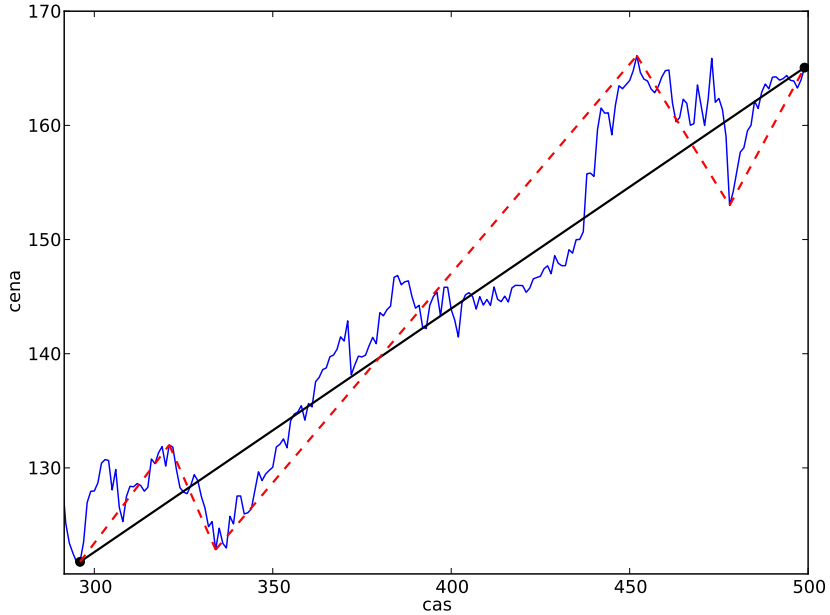
Bola vytvorená ukážková sada dvadsiatich funkčných neurónových sietí, ktoré sú schopné s určitou presnosťou nájsť impulzy na testovacích dátach. Pre demonštráciu sú použité skutočné dáta finančného trhu. Presnejšie akciových trhov Google, IBM, Intel a akciového indexu NASDAQ-100.

Obrázky 4.1, 4.2 zobrazujú výrez dát so správne detekovaným impulzom. Príklad nesprávne určeného impulzu je na obrázku 4.3. Detekované impulzy na kompletnej sade dát s rôzne nastaveným parametrom R sú zobrazené na obrázkoch 4.4, 4.5, 4.6 a 4.7. Na obrázku 4.6 je možné vidieť, že nie všetky figúry spĺňajú požiadavky na impulz a teda sú detekované nesprávne.

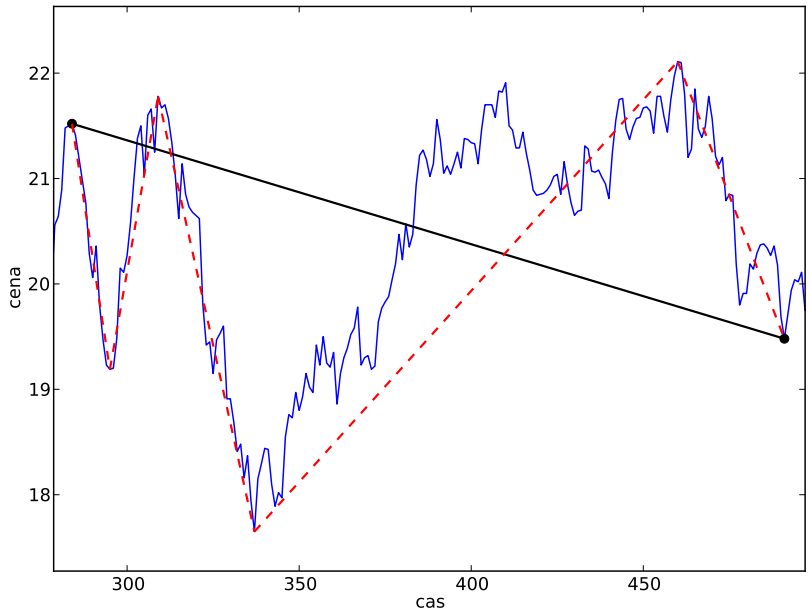
Nájdené impulzy sú na grafoch vyznačené čiernymi úsečkami. Červená prerušovaná čiara znázorňuje vnútornú štruktúru impulzu. Korekcie nie sú vyznačené kvôli lepšej prehľadnosti výsledkov, avšak čo nie je impulzom je korekciou. Pri grafoch nie sú uvedené jednotky veličín na osiach x (čas) a y (cena), pretože program pracuje s bezrozmernými dátami.



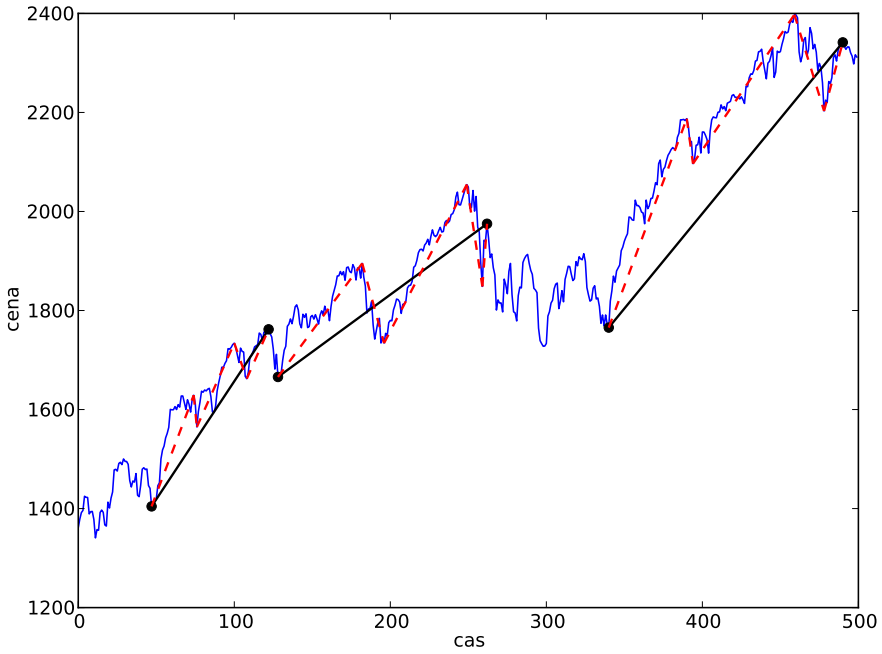
Obr. 4.1: Správne detekovaný impulz na výreze dát akciového trhu Intel.



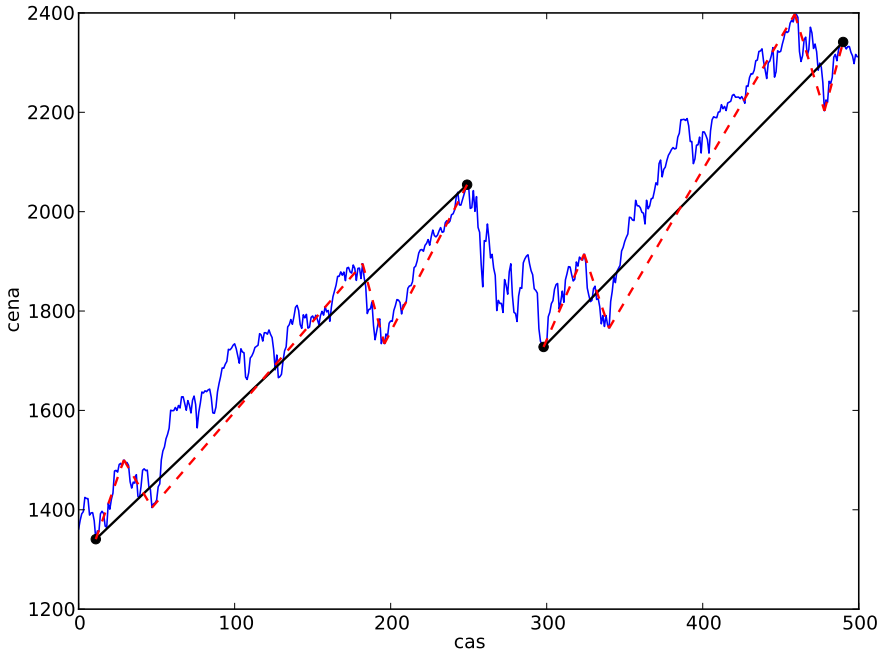
Obr. 4.2: Správne detekovaný impulz na výreze dát akciového trhu IBM.



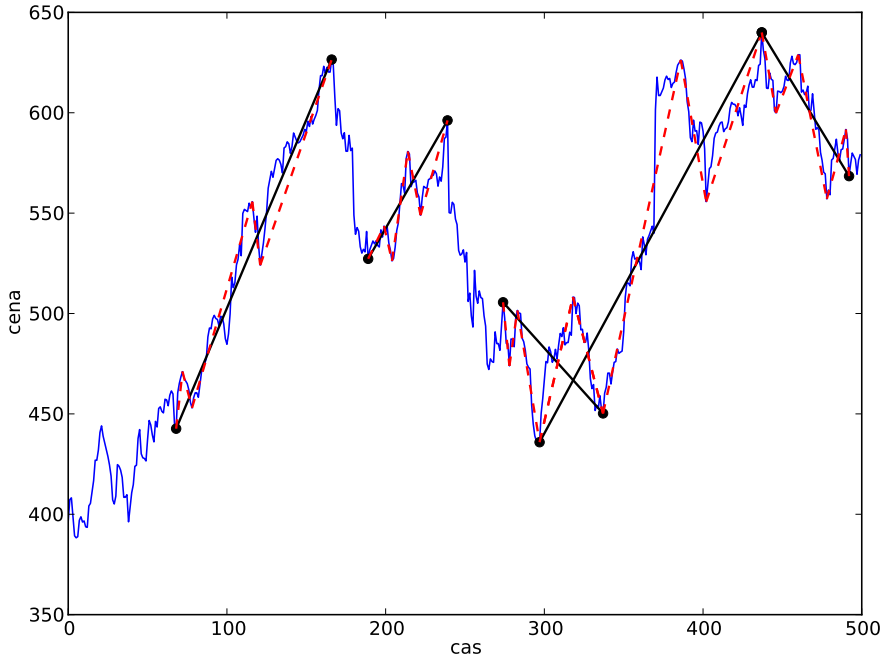
Obr. 4.3: Nesprávne detekovaný impulz na výreze dát akciového trhu Intel.



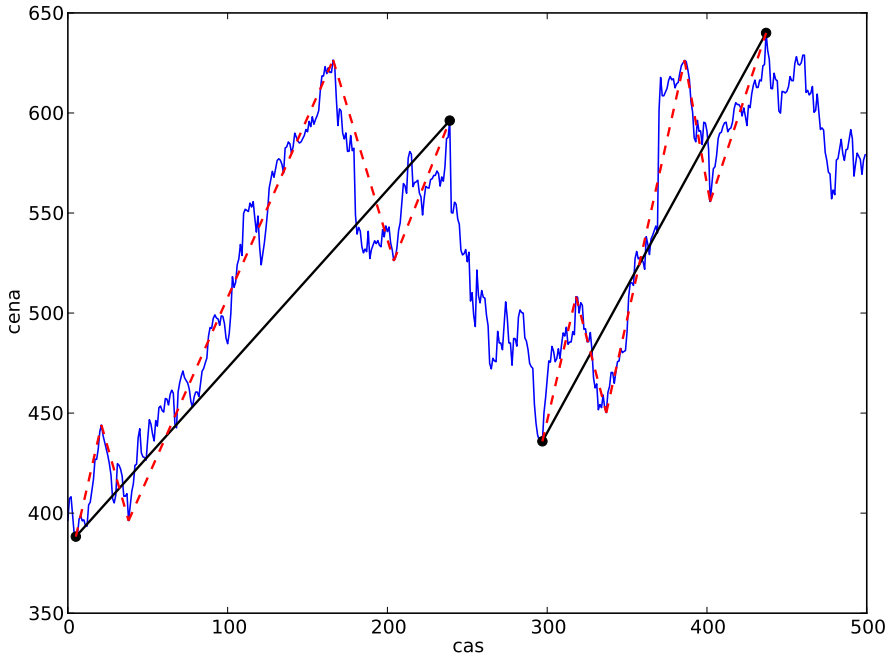
Obr. 4.4: Detekované impulzy na dátach akciového trhu NASDAQ-100. Použítá celá sada sietí a parameter $R = 1,04$.



Obr. 4.5: Detekované impulzy na dátach akciového trhu NASDAQ-100. Použítá celá sada sietí a parameter $R = 1,06$.



Obr. 4.6: Detekované impulzy na dátach akciového trhu Google. Použítá celá sada sietí a parameter $R = 1,03$.



Obr. 4.7: Detekované impulzy na dátach akciového trhu Google. Použítá celá sada sietí a parameter $R = 1,07$.

Kapitola 5

Záver

V rámci práce boli predstavené základy Elliottovej vlnovej teórie a úvod do problematiky neurónových sietí. Na základe získaných teoretických poznatkov bol navrhnutý a implementovaný detekčný systém, ktorý je schopný detekovať impulzné figúry na statickej časovej rade pomocou algoritmu pre vyhľadávanie lokálnych extrémov a umelých neurónových sietí.

Výhodou tohto prístupu je priama detekcia vĺn vyššieho stupňa. Teda v prípade záujmu o rozsiahlejšie trendy, nie je potrebné zostavovať kompletnú štruktúru na základe systému hierarchických pravidiel od najmenších monovĺn, ale je možná bezprostredná detekcia impulzov vyšších stupňov, ktoré sú predstaviteľmi týchto dlhodobých trendov.

Nevýhodou navrhnutého postupu je výskyt nesprávne určených impulzov z dôvodu, že neurónové siete vždy pracujú s určitou chybou klasifikácie. Taktiež nám systém neposkytuje presnejšie informácie o vnútornej štruktúre jednotlivých figúr a vlnách nižších stupňov, ktoré ich tvoria.

Aj keď systém nedosahuje vždy úplne správne výsledky, pri vyššej kompresii dát si vedie relatívne dobre. Možným vylepšením je zmeniť alebo prepracovať metódu predspracovania dát a tvorby segmentov, ktorá momentálne značne obmedzuje výsledky programu.

V budúcnosti je tiež možné zamerať sa na použitie iných typov umelých neurónových sietí, napríklad Support Vector Machine. V prípade detekcie a klasifikácie korekcií by bolo vhodné aplikovanie sietí učiacich sa bez učiteľa. Tie by vzhľadom na nedostatok správne klasifikovaných trénovacích dát mali byť schopné odhaliť vnútornú štruktúru korekčných figúr. Jedno z možných rozšírení predstavuje aj tvorba hierarchickej štruktúry vĺn na základe detekovaných impulzov, čo by mohlo značne zlepšiť použiteľnosť tohto prístupu v praxi.

Samotné výstupy programu na základe Elliottovej teórie poskytujú len obmedzenú predstavu o dianí na trhu, avšak môžu byť použité v rámci robustnejšej technickej analýzy.

Literatúra

[1] SOJKA, Z. a DOSTÁL, P. *Elliottovy vlny.* Brno: Tribun EU, 2008. 272 s. ISBN 978-80-7399-630-7.

[2] PRECHTER, R. *The elliott wave principle: key to market behavior.* Gainesville, GA: Elliott Wave International/New Classic Library, 2005. ISBN 09-327-5075-3.

[3] DOSTÁL, P. *Pokročilé metody analýz a modelování v podnikatelství a veřejné správě.* Brno: Akademické nakladatelství CERM, 2008. ISBN 978-80-7204-605-8.

[4] ZBOŘIL, F. V. *Soft Computing* [online]. 2010 [cit. 2012-04-26]. Dostupné na: http://www.fit.vutbr.cz/study/courses/SFC.

[5] *Neuron. Wikimedia Commons* [online]. 2004- [cit. 2012-04-24]. Dostupné na: http://commons.wikimedia.org/wiki/Neuron.

[6] HU, Y. H. a HWANG, J.-N. *Handbook of neural network signal processing.* Boca Raton: CRC Press, 2002. ISBN 08-493-2359-2.

[7] STERGIOU, C. a SIGANOS, D. *Neural Networks* [online]. 1996 [cit. 2012-04-29]. Dostupné na: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html.

[8] PRECHELT, L. *Proben1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules.* Fakultat fur Informatik Universitat Karlsruhe, 1994. Technická správa 21/94.

[9] BISHOP, C. M. *Neural Networks for pattern recognition.* Oxford: Oxford University Press, 1995. ISBN 01-985-3864-2.

[10] *Hyperbolic Tangent. Wikimedia Commons* [online]. 2004- [cit. 2012-04-24]. Dostupné na: https://commons.wikimedia.org/wiki/File:Hyperbolic_Tangent.svg.

[11] PRATT, K. B. *Locating patterns in discrete time-series.* University of South Florida, 2001. Dizertačná práca.

[12] *About. Python Programming Language* [online]. 1990-2012 [cit. 2012-05-03]. Dostupné na: http://www.python.org/about/.

[13] *PyBrain* [online]. 2010- [cit. 2012-05-03]. Dostupné na: http://pybrain.org/pages/home.

[14] *Introduction. Matplotlib: python plotting* [online]. 2008- [cit. 2012-05-03]. Dostupné na: http://matplotlib.sourceforge.net/users/intro.html.

[15] IGEL, C. a HÜSKEN, M. Empirical evaluation of the improved Rprop learning algorithms. *Neurocomputing*. 2003, roč. 50. S. 105–123.

Dodatok A

Manuál

Pre detekciu vĺn je potrebné zadať názov súboru, obsahujúceho vstupnú časovú radu, viď A.2. Potom spustiť skript `detektor.py`.

Pre natrénovanie novej siete stačí spustiť skript `trainer.py`. Jeho výstupom je nová sieť uložená do priečinka /networks s názvom `mlp.net`.

A.1 Minimálne požiadavky programu

Pre spustenie skriptov je potrebný interpret Python 2.7 spolu s nainštalovanými knižnicami PyBrain, matplotlib verzia 0.98+, NumPy verzia 1.1+ a SciPy verzia 0.6+.

Na súbor, ktorý obsahuje vstupnú časovú radu, je len jedna požiadavka a to, že každá z hodnôt je na samostatnom riadku.

A.2 Možnosti ovplyvnenia programu

Pre zmenu sady používaných neurónových sietí stačí, pridať alebo odobrať súbor s uloženou sieťou do/z priečinka /networks. Skript `detektor.py` načítava všetky súbory, ktoré tento priečinok obsahuje.

Jednotlivé časti systému je možné ovplyvniť zmenou parametrou v kóde skriptou. Parametre príkazového riadku nie sú implementované.

Parametre vo východzom stave boli stanovené počas testovania aplikácie. To však nezaručuje, že pri nich program dosahuje najlepšie výsledky.

detector.py

- *DATA_FILE* názov súboru s časovou radou, musí sa nachádzať v priečinku /data
- *R* určuje veľkosť kompresie dát, odporúčané hodnoty sú v intervale $< 1,03; 1,07 >$
- *PROB_BORDER* hraničná hodnota pravdepodobnosti výskytu impulzu v %, odporúčaná hodnota je viac ako 70 %

trainer.py

- *EPOCH_NUM* počet epôch tréningu
- *ALPHA* hraničná hodnota pre stratu generalizácie v %

- *NUM_OF_HIDDEN_NEURONS* počet neurónov v skrytej vrstve, ak je rovný 0, tak bude stanovený náhodne
- *TARGET_ERROR* požadovaná chybovosť siete v %
- *NUM_OF_TRAIN_SAMPLES* počet vzorov v trénovacej množine