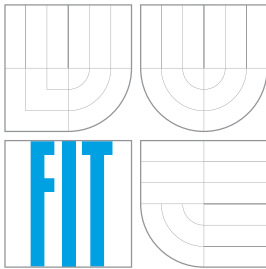


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ALGORITMICKÉ OBCHODOVÁNÍ NA BURZE S VYU- ŽITÍM UMĚLÝCH NEURONOVÝCH SÍTÍ

ALGORITHMIC TRADING USING ARTIFICIAL NEURAL NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

KAREL ČERVÍČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÓKE, Ph.D.

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Červíček Karel**

Obor: Informační technologie

Téma: **Algoritmické obchodování na burze s využitím umělých neuronových sítí**

Algorithmic Trading Using Artificial Neural Networks

Kategorie: Umělá inteligence

Pokyny:

1. Seznamte se s teorií umělých neuronových sítí a intraday obchodování na burze.
2. Prostudujte dostupné techniky technické analýzy.
3. Seznamte se s vybraným nástrojem pro trénování neuronových sítí.
4. Definujte vhodná data pro trénování a testování metod pro algoritmické obchodování.
5. Zvolte vhodné indikátory technické analýzy či jiné parametry pro vstup neuronové sítě. Natrénujte a otestujte základní obchodní algoritmus.
6. Upravte výběr popř. vhodně zkombinujte indikátory a zlepšete Váš základní obchodní algoritmus.
7. Zhodnoťte výsledky, vyberte nejvhodnější metodu (parametry modelu a indikátory) a navrhněte směry dalšího vývoje.
8. Vytvořte A2 plakátek a cca 30 vteřinové video prezentující výsledky vašeho projektu.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 5 ze zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szóke Igor, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

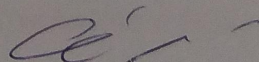
Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Forex je dnes největším trhem na světě. Díky vysoké likviditě je vhodným kandidátem pro intradenní obchodování na základě jisté obchodní strategie založené na technické a fundamentální analýze. Obchodní strategie jdou navrhnout pro automatické algoritmické obchodování. Takováto strategie je navržena s využitím neuronové sítě, která zástává pozici aproximátoru časové řady kurzovních dat na základě, kterého je možné predikovat budoucí vývoj.

Abstract

Forex is the biggest foreign exchange market. Thanks to high liquidity it is a good candidate for intraday trading with certain trading strategies based on technical and fundamental analysis. Trading strategies can be proposed for automatic algorithmic trading. Strategy in this article is designed with a neural network that holds positions as approximator of time series data based on the exchange rate, which can predict the future.

Klíčová slova

strojové učení, algoritmické obchodování, Forex, neuronové sítě, predikce, technická analýza

Keywords

machine learning, algorithmic trading, Forex, Artificial neural network, prediction, technical analysis

Citace

ČERVÍČEK, Karel. *Algoritmické obchodování na burze s využitím umělých neuronových sítí*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Szóke Igor.

Algoritmické obchodování na burze s využitím umělých neuronových sítí

Prohlášení

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

.....
Karel Červíček
17. května 2016

Poděkování

Děkuji za odborné rady svému vedoucímu v oblasti trénování neuronových sítí

© Karel Červíček, 2016.

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	Forex a obchodování	4
2.1	Technická analýza	5
2.2	Klouzavý průměr	6
2.3	MACD	7
2.4	RSI	7
2.5	Obchodní strategie	8
2.6	Rizika a hlubší problémy obchodování	9
3	Využití technologie a jejich odůvodnění	11
3.1	Knihovny a programovací jazyk	11
3.2	Neuronové sítě	11
3.2.1	Umělý neuron	13
3.2.2	Topologie	13
3.2.3	Trénování neuronové sítě	14
3.2.4	Zpětná propagace chyby	15
4	Teoretické poznatky pro úspěšné trénování	16
4.1	Stochastické a dávkové trénování	16
4.2	Náhodné promíchání trénovací sady	16
4.3	Nastavení parametrů	17
4.4	Aktivační funkce	17
4.5	Normalizace	18
4.6	Rozložení pravděpodobnosti kurzovních dat	20
4.7	Redukce náhodného šumu	21
5	Vytvořený obchodní model	22
5.1	Řízení obchodních pozic	23
5.2	Evaluace predikce	24
5.3	Bodový zisk	25
5.4	Využitá topologie	25
5.5	Implementace	27
6	Experimenty	29
6.1	Vypuštění diferencí	29
6.2	Pouze diference	29
6.3	Přidání RSI	30

6.4	Zhodnocení obchodního modelu a experimentů	31
7	Závěr	33
	Literatura	34
	Přílohy	36
	Seznam příloh	37
A	Spuštění a popis funkcí	38
	A.1 Vysvětlení celkového konceptu	39
	A.2 Obsah CD	39

Kapitola 1

Úvod

Tato práce se zabývá využitím prostředků strojového učení 3.2 pro algoritmické obchodování na devizovém trhu. Obchodování je založeno na zpracování dat tvořené trhem 2.1 a následnou predikcí dalších dějů. Záměrně je předchozí věta obecná a nemluví pouze o predikci ceny, neboť přístup k návrhu obchodního modelu není přímočarý a je třeba vyřešit mnoho problémů, které provází návrh v různých částech.

První překážky vyplývají na povrch při bližším seznámení, jak funguje a jak probíhá obchodování na devizovém trhu Forex 2. Díky vysoké likviditě je kolem tohoto trhu velká komunita obchodníků a jiných institucí. Z tohoto celku však vydělává jen malé procento. Obchodník si musí osvojit obchodní strategie a být schopen reagovat na měnící se charakter trhu, z toho plyne, že je také nutné obchodní strategie upravovat, či od základu navrhovat. K návrhu obchodní strategie je pak nutné vhodně využít různých vlastností technických indikátorů, technické analýzy 2.1 nebo analýzy fundamentální. I v případě návrhu automatického obchodního modelu založeného na prostředcích strojového učení jsou využity tyto znalosti a řešení úskalí s tím spojená.

K samotné predikci je využita neuronová síť, kterou je třeba naučit charakter kurzovních dat. K tomu je nutné nejen získat dostatek dat historických, ale také zvolit jejich správné zpracování pro schopnost generalizace jejich charakteru při učení 4. Zvolit zpracování, které by zajistilo co nejlepší generalizaci nebo je také možné říct aproximaci, neboť vývoj kurzovních dat je funkcí nebo přesněji časovou řadou, není také lehké navrhnout. Neuronová síť není schopna správného učení na datech, které nemají normální rozložení, neboť selhávají ověřené postupy normalizace, kterou je nutné nad daty provést pokud s nimi chceme neuronovou síť trénovat.

V poslední fázi návrhu přichází na řadu, jaký zvolit výstup neuronové sítě nebo jak k němu přistupovat. To zahrnuje návrh logiky otevírání a uzavírání obchodních pozic na základě výstupu. Pokud bude zvolena nevhodně, tak znehodnotí předešlé správně provedené kroky. Na druhou stranu pokud do návrhu logiky zahrneme poznatky o Forexu a obchodování na tomto trhu, můžeme zastínit mnoho dalších problémů.

Práce od začátku až do konce tyto problémy řeší. V první řadě popisuje možný návrh obchodních strategií a s tím hlubší zaslíbení do Forexu a možném využití technických indikátorů obecně. Na základě studie jsou porovnány různé využití neuronových sítí především v rámci predikce a popis dopředné vícevrstvé neuronové sítě. S těmito znalostmi je následně postaven automatický obchodní model 5, který využívá technické indikátory, eliminuje problémy s rozložením pravděpodobnosti kurzovních dat a náhodným šumem, také díky vhodnému návrhu výstupu neuronové sítě a jeho zpracováním jsou eliminovány další problémy intradenního obchodování.

Kapitola 2

Forex a obchodování

Kapitola popisující Forex a prezentující stručný přehled jeho fungování je uvedena jako první, neboť má čtenáře zasvětit do oblasti, které je práce věnována. Jejím účelem není ani tak přesné vysvětlení dějů na měnových trzích, ale demonstrace toho, o jak široký a zajímavý obor se jedná. Je zde prostor pro uplatnění technických, ekonomických a mnoha dalších znalostí. Za prvé zde uvádím nezbytné teoretické informace, které vysvětlují dále používané termíny a data. Za druhé zde uvádím vlastní pohled na problematiku, který jsem získal praktickým sledováním Forexu a obchodních strategií.

Název Forex je odvozený z *Foreign Exchange*. Z názvu vyplývá, že jde o směnu měnových párů. Jde o devizový mezinárodní obchodní systém a je zajímavé, že i přes to, že Forex nemá vlastní burzu, jde o největší trh na světě, kde se každý den objem transakcí dostává do bilionů dolarů. Směny jsou prováděny mezibankovními transakcemi.

Z historického hlediska byl zaveden v roce 1875 *Zlatý standard*, díky kterému byly jednotlivé měny v konverzním poměru se zlatem a naopak. S tím došlo ke vzniku kurzů mezi měnami na základě různého množství zlata, kterým je daná měna kryta.

Stojí za to se nad dalším historickým vývojem zamyslet relativně k dnešní světové finanční situaci. Zlatý standard byl zrušen s nástupem první světové války, kde byly kladeny vysoké nároky na financování válečných projektů a s nimi bylo tisknuto více peněz, než bylo zlata.

V roce 1944 však jako náhrada vznikl Brettonwoodský systém, ve kterém americký dolar zastal funkci zlata a stal se primární reverzní měnou, která byla podložena zlatem. Ovšem tento systém začal po čase kolabovat na nedostatku zlata v amerických rezervách a nebyl schopen pokrýt všechny dolary, které měli zahraniční banky ve svých rezervách [7].

V dnešní době je cena měny dána nabídkou a poptávkou, tento systém je nazýván plovoucí měnové kurzy. Pokud se chce jednat stát účastníkem tohoto trhu, není na tom nic náročného. Pokud nemáme relativně velký kapitál a oprávnění vstoupit samostatně na trh, je nutné zajistit si **brokera**. Broker je zprostředkovatel nákupů kontraktů, který si však účtuje jisté poplatky. S tím se dostávám k tom, že u výběru brokera jsou posuzovány podmínky, za kterých nám poskytne přístup na trh. Těmi jsou například:

Spread je rozdíl mezi nákupní a prodejní cenou měny [7]

Poskytovaná páka Finanční páka nám při malém vlastním kapitálu umožňuje obchodovat s využitím výrazně většího půjčeného kapitálu za účelem zvýšení potenciálních zisků a ztrát [7]

Broker poskytuje obchodní platformu s jeho konfigurací. Daná platforma umožňuje

z osobního počítače s připojením k internetu otvírat obchodní pozice, grafickou reprezentaci kurzovních dat nebo také zavedení vlastních obchodních algoritmů. Obchodní pozice jsou děleny na **long** a **short**:

long Jedná se o spekulaci na růst ceny resp. nákupů kontraktů. Pokud poté dojde k samotnému růstu ceny, tak s ní přichází i zisk. Pokud však dojde k poklesu ceny, tak dojde ke ztrátě. [7]

short Jedná se o spekulaci na pokles ceny resp. prodeji kontraktů. Pokud poté dojde k samotnému poklesu ceny, tak s ní přichází i zisk. Pokud však dojde k vzestupu ceny, tak dojde ke ztrátě. [7]

Jak bylo psáno, platforma poskytuje reprezentaci dat v podobě grafu. Nejpoužívanější je graf svíčkový, který se tedy skládá ze svíček, jak je vidět na obrázku 2.1.



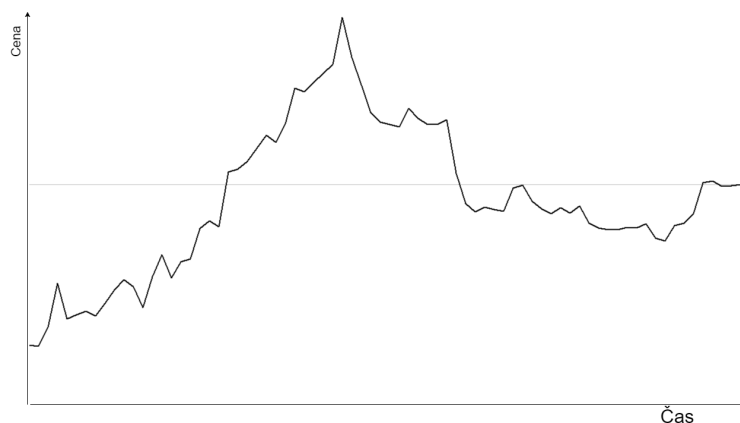
Obrázek 2.1: Svíčkový graf

Každá svíčka se skládá ze čtyř hodnot: high, low, open, close a vyjadřuje jistý časový úsek průběhu daného páru. Open a close vyjadřuje rozdíl otevírací a zavírací cenou. High a low potom zaznamenává nejvyšší a nejnižší cenu v časovém intervalu svíčky. Pro objasnění svíčkový graf je zobrazován v různých časových intervalech, za kterých je svíčka tvořena. Nejčastěji používanými intervaly je 1 minuta, 5 minut, 15 minut, 4 hodiny a 1 den. Dalším používaným grafem, který je využíván v této práci, je graf čárový. Velice dobře zobrazuje cenový průběh na úkor zastínění cenové rozkolísanosti. Tento graf je možné vidět na obrázku 2.2 a zobrazuje hodnoty close.

2.1 Technická analýza

Jde o jeden ze dvou přístupů, jak analyzovat data a následně predikovat jejich další vývoj. Druhým přístupem je analýza fundamentální. Hlavním rozdílem je, že v technické analýze jsou systematicky zkoumána data tvořená samotným trhem, například cena nebo volatilita. Technická analýza se na rozdíl od fundamentální nezabývá jevy, které netvoří trh, ale trh je jimi ovlivňován, například politická situace, zveřejnění ekonomických dat nebo daňová politika.

Na základě tvořené ceny jsou počítány technické indikátory. Technických indikátorů je celá řada. Pomocí matematického vzorce je cena přepočítána a výsledkem je hodnota, která



Obrázek 2.2: Čárový graf

má specifickou informační hodnotu, jaká je aktuální situace na trhu. Například můžeme dostat informaci, že je potvrzený trend, nebo že je trh takzvaně překoupený a je vhodné vystoupit z aktuální pozice. Trend je termín, který zde ještě zazní a relativně k trhu je významově stejný jako v jiných oblastech. Je doporučeno, že obchodník začátečník by měl obchodovat s trendem. Čili jeho snahou je rozpoznat trend, například pomocí technické analýzy a podle toho, zda bude trend rostoucí či klesající vstoupit do pozice. Tímto se budu řídit i v praktické části, kde mojí snahou bude trend predikovat. Technické indikátory je možné kombinovat na základě jejich účelu a vytvářet obchodní strategie 2.6

2.2 Klouzavý průměr



Obrázek 2.3: Ukázka exponenciálního a normálního klouzavého průměru.

Je základním **indikátorem** ¹, kde je počítán průměr svíček za daný interval. Většinou je počítán přes hodnotu close. Pro účely trénování neuronové sítě bude vhodné využít klouzavého průměru a exponenciálního klouzavého průměru, který vhodně kopíruje hodnoty cen nebo případně i další klouzavé průměry. Pro představu jsem vybral jednu ukázkou 2.3

¹Více informací je možné zjistit zde: https://en.wikipedia.org/wiki/Moving_average

a také vzorec, podle kterého je počítán klouzavý průměr MA a exponenciální klouzavý průměr EMA .

$$MA_i(n) = \frac{1}{n} \sum_{i=1}^n P_{n-i} \quad (2.1)$$

$$EMA_i(n) = (P_i - EMA_{i-1}) \frac{2}{n+1} + EMA_{i-1} \quad (2.2)$$

2.3 MACD

Je zkratkou pro **Moving Average Convergence Divergence**². Tento indikátor je tvořen na základě klouzavého průměru. Je tvořen rozdílem 12denního a 26denního exponenciálního klouzavého průměru. Je jasným představitelem indikátorů, které vypovídají o situaci, která se na trhu odehrála. Grafické zobrazení tohoto indikátoru je většinou realizováno v samostatném okně.



Obrázek 2.4: Bílé sloupce značí signální křivku, červeně je značený rozdíl exponenciálních klouzavých průměrů

Dalším důležitým prvkem je signální křivka, tvořená 9denním exponenciálním klouzavým průměrem a nulová přímkou místo, kde 12denní MA je rovno 26dennímu MA. Tento indikátor je vhodný, neboť velice dobře vypovídá a transformuje informace, které jsou pro nás potřebné. Těmi jsou informace o překoupení trhu, potvrzení trhu a také slábnoucí trend.

2.4 RSI

Klasicky používaný index relativní síly, který se řadí do třídy oscilátorů. Jeho hodnota se stále pohybuje mezi 0 – 100. Jde opět o jinou reprezentaci toho, zda je trh překoupený nebo je potvrzený trend, ale díky zcela odlišnému charakteru výpočtu³, který je uvedený níže, jsem usoudil, že bude vhodné ho dále zakomponovat.

²Více informací je možné zjistit zde: <https://cs.wikipedia.org/wiki/MACD>

³Více informací je možné zjistit zde: https://en.wikipedia.org/wiki/Relative_strength_index

$$RSI_t(n) = 100 - \frac{100}{1 + \frac{U(n)}{D(n)}} \quad (2.3)$$

$U(n)$ součet kladných cenových změn za období délky n

$D(n)$ součet záporných cenových změn za období délky n



Obrázek 2.5: Modře znázorněný technický indikátor RSI.

2.5 Obchodní strategie

Aby bylo jasné, na čem je založeno plánování obchodních strategií [8], jsem pro účel ukázky navrhl, jak by mohla obchodní strategie vypadat. Vzal jsem v potaz technické indikátory, o kterých jsem se zmínil a založil strategii na překřížení klouzavých průměrů a detekci otočení trhu na základě hodnot $MACD$ a RSI . Na obrázku 2.6 je možné tuto strategii vidět v nasazení. S překřížením dvou klouzavých průměrů čekám na potvrzení signálu $MACD$, že je trend otočený. Ve chvíli, kdy jsou tyto podmínky splněny, je možné v tomto případě otevřít pozici short, jak ukazují červené šipky.

Také je nutné zajistit správné vystoupení z trhu buď pomocí technických indikátorů nebo pomocí *take profit*, se kterým také blíže souvisí *stop loss*.

take profit Jedná se o jednoduchý výstup z trhu, který je předem nastaven jako určitá cena již v době zadávání pozice. [8]

stop loss Jde o zajištění obchodní pozice proti nepředpokládaným ztrátám v momentu, kdy se trh otočí proti nám. Při nastavování jde o stejný postup, kdy v době nastavování pozice určíme cenovou hladinu. [8]

Na obrázku je červeným křížkem ukázáno, jak je možné indikovat ukončení pozice s využitím technického indikátoru RSI dynamicky s vývojem trhu. Pokud se hodnota RSI dostane pod hodnotu 30 a zpět, tak je pozice ukončena s případným ziskem. Stejný princip strategie je opačně aplikován pro pozici long.



Obrázek 2.6: Příklad kombinace technických indikátorů a možné obchodní strategie.

Obchodní strategie jsou často založené na splnění jistých podmínek, které je možné automatizovat pomocí programu. Odvětví automatického obchodování je také možné vyhledávat pod názvem algoritmické obchodování. V platformě je možné strategie programovat pomocí jazyka MQL, který komunikuje i s jazykem Python, který využívám 3.1.

2.6 Rizika a hlubší problémy obchodování

Přes tři měsíce vlastního sledování a studie trhu jsem zjistil, že je možné využít mnoho obchodních strategií. Jednak zde hrají roli technické indikátory, ale také charakter samotného měnového páru. Díky jednoduchosti a dostupnosti pro mnoho lidí se Forex stále rozrůstá, ovšem jen malé procento obchodníků je ve stabilním zisku. Hlavními problémy jsou neotestované strategie, kdy jde pouze o vizuální rozvahu o tom, zda daná strategie je zisková a dojde k přehlédnutí podstatných problémů. Proto v této práci v praktické části provedu jak vizuální rozvahu, tak i kontrolu výdělku. Ve strategiích založených na technické analýze velice často záleží na nastavení stoploss a takeprofit.

Jako příklad uvedu strategii breakout zones, která je založena na myšlence, že u páru eur/usd je větší volatilita v čase otevření londýnské burzy a pokud po čase otevření je cena páru větší než nejvyšší cena v části asijské seance je možné jít do long pozice a opakem tomu je možnost jít do pozice short. Tato strategie je zisková pouze za podmínky, že jsou správně nastaveny hladiny takeprofitu a stoplossu.

V této práci bude tento problém řešen výstupem z pozice na základě hodnoty, která bude výstupem neuronové sítě a takeprofit tak nebude třeba. Při reálném nasazení je nutné pozici vždy zajistit, kvůli bezpečnosti, ale je možné ji nastavit, tak aby pouze jistila jen velmi

neobvyklé situace a strategii příliš neovlivňovala. Dalším důvodem neúspěchu jsou situace, kdy broker jedná proti nám. I přesto, že řešení těchto situací je nad rámec této práce, je nutné je zde zmínit, neboť z hlediska výběru brokera a nasazení strategie jsou zásadní. Výdělký brokera jsou postaveny na poplatcích za obchod, který Vám zprostředkoval díky jeho velkému kapitálu a obchodu na páku. Při obchodování na demo účtu se u některých brokerů chovají jinak než při reálném obchodu. Často dochází k takzvanému roztahování spreadu. Také se stává, že broker Vaše zadané obchody pouze simuluje na svém serveru a tyto pozice nejsou na reálném trhu, potom může docházet ke zpoždění při vstupu a ukončení pozice. Tyto problémy je nutné řešit vhodným výběrem brokera a smluvních podmínek.

Kapitola 3

Využití technologie a jejich odůvodnění

V této kapitole jsou popsány nástroje, které využívám pro svůj obchodní model. Zvolil jsem cestu použití technologií, které jsou využívány v oblasti strojového učení a zpracování dat obecně. Zde popsané knihovny je možné i přes jejich všestrannost a efektivnost získat zdarma.

3.1 Knihovny a programovací jazyk

Pro implementaci 5.5 jsem zvolil jazyk python. Python je scriptovacím jazykem s velice dobrou dokumentací, kromě toho byl vybrán také díky mnoha knihovnám, které ulehčují práci s velkým množstvím dat a různé operace nad nimi.

Theano Knihovna pro definici matematických výrazů, které je potom možné následně vyhodnocovat v rámci více-rozměrných matic. Při výpočtech je možné efektivní využití GPU. Je založena na jazyku C a výpočty jsou tak velmi rychlé. Theano je možné využít pro stavbu neuronových sítí. Také využívám nadstavbu, knihovnu Keras, zjednodušující definici modelů neuronových sítí.

NumPy tato knihovna umožňuje efektivní práci s vícerozměrnými poli. Také je možné využít mnoha funkcí z lineární algebry, zpracování signálu například Fourierova transformace a generátor náhodných čísel.

matplotlib Využívám pro grafickou reprezentaci dat. Všechny obrázky jsou tvořené právě za pomoci této knihovny.

3.2 Neuronové sítě

Tato část je věnována nejdůležitějšímu nástroji v mé práci. V úvodu k neuronovým sítím a dalších čtyřech Kapitolách popisuji za prvé rešerši na téma využití neuronových sítí pro predikci neboli regresi. Za druhé celkový koncept fungování toho matematického modelu a možných algoritmů, které jsou s neuronovou sítí spjaté. Odůvodňuji jejich využití v hlavním obchodním modelu 5 na základě zde uvedené studie a bližším vysvětlením topologie a principu trénování. Pro ucelený pohled na problematiku jako první uvádím samostatný umělý

neuron, který je základem umělé neuronové sítě, vše je zakončeno částečným vysvětlením algoritmu zpětné propagace chyby.

Tato kapitola logicky souvisí s kapitolou 4, kde je dále rozvedeno, jak celkově přistupovat ke zde uvedeným poznatkům.

S neuronovými sítěmi se setkáme v oboru strojového učení nebo v širším pohledu v oboru *Soft Computing*. Nejsou žádnou novinkou posledního desetiletí, návrh umělého neuronu přišel již přibližně v padesátých letech za účelem simulace biologického neuronu. Neuronové sítě jsou široce využívaným nástrojem pro klasifikaci, rozpoznávání zvuku, obrazu, kompresi dat, predikci a také jsou prvkem simulací.

Z uvedeného širokého pole využití rozvedu jako první predikci. Zde je také mnoho cest, jak neuronovou síť využít, například jako samotný prvek nebo kombinaci více neuronových sítí. Je možné využít optimalizační algoritmy, které zapříčiní lepší konečné výsledky predikce, neboť optimalizují vstupní data (to je možné si představit například optimalizací parametrů technických indikátorů 2.6 nebo optimalizací kombinací různých vstupů na základě složitější ohodnocovací funkce) nebo samotnou topologii 3.2.2. V hlavním obchodním modelu využívám dopřednou vícevrstvou neuronovou síť, které jsou věnovány další čtyři podkapitoly. Rozhodl jsem se na základě studované literatury [5], kde je modelován vývoj burzy v Johannesburgu. Práce se zabývá využitím co nejvíce dat pro modelování, je bráno v potaz 63 různých dat týkajících se Johannesburgské burzy. V projektu jsou zpracované technické indikátory, akciové indexy, ekonomické statistiky a mnoho dalších dat. Tento článek mě velice zaujal především díky jednomu zásadnímu poznatku, který se potvrdil i v mé práci 6. Robert J.Eyden zpracovával co nejvíce dat za účelem co nejširšího pohledu na burzu. Očekával, že pokud bude mít neuronová síť více dat, tak výsledky budou lepší, ovšem po vyloučení 20 datových zdrojů z 63 byly výsledky srovnatelné. V mé práci se domnívám, že tento fakt vysvětlují poznatky v kapitole 4. Robert J.Eyden využívá algoritmus zpětné propagace a vícevrstvou neuronovou síť.

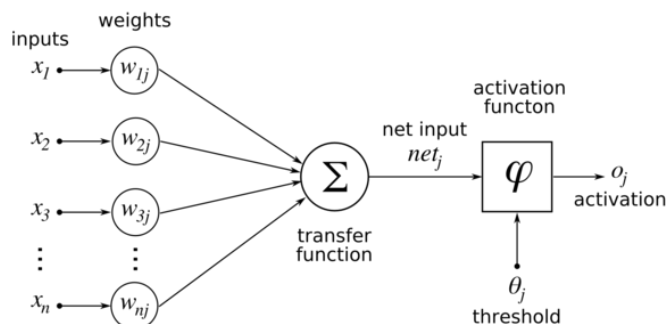
Dopředná neuronová síť je také využita v článku [1], kde je celkově uvedena problematika predikce za podpory neuronových sítí, stejně tak je tomu v článku [6]. Mimo ekonomii také uvedu, že dopředná neuronová síť je využita také například k predikci výkonu solární termální elektrárny[14]. V tomto projektu síť predikuje solární frakci termálního energetického systému s odchylkou 10 procent.

Vedle dopředné neuronové sítě je také mnoho dalších modelů sítí, které jsou často využívány. Například v článku [9], který se zabývá predikcí požadavků na energii u bioklimatické budovy, je využita neuronová síť NARX. Výsledky predikce jsou také uspokojivé, střední chyba činila 11.58%.

Dalším typem, který stojí za zmínění je typ LSTM. Podle [13] dokáže velice dobře generalizovat problém, na kterém například dopředná neuronová síť selhává. V této práci je využívána za účelem překladu vět, kde se jedná o sekvenční data. V tomto ohledu se jedná o adekvátní nástroj, neboť zjednodušeně řečeno, topologie má uměle vytvořenou paměť a je vhodná pro predikci časových řad a sekvenčních dat. Dále jsem ji nevyužil, neboť trénování je výpočetně velmi náročné a v oblasti, které je věnována tato práce, jde především o správný preprocessing dat 4. Ovšem v sekci 5.5 implementuji i možnost využití tohoto modelu.

Také jsem se zmiňoval o využití genetických algoritmů. V [3] jsou využívány k optimálnímu nastavení parametrů dopředné neuronové sítě. Co jsou parametry a jakou hrají roli, popisují dále. Také je zde využita analýza hlavních komponent (PCA), která je zmíněna i v mé práci 4. Za pomoci PCA byla zredukována dimenzionalita dat za účelem lepší predikce. Výsledný model predikoval kvalitu vody v řekách s přesností 91%.

3.2.1 Umělý neuron



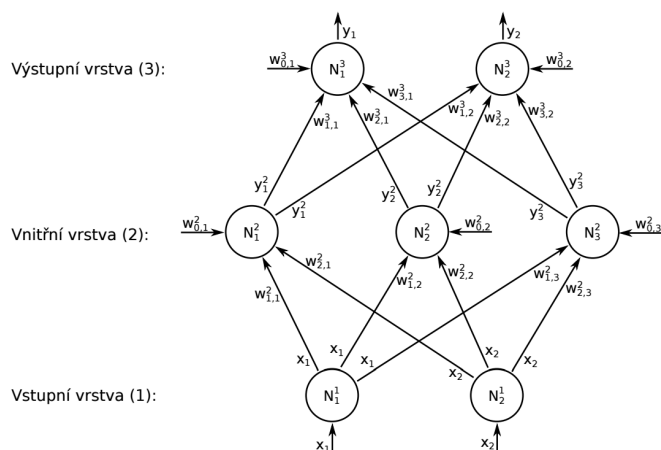
Obrázek 3.1: Umělý neuron, zdroj <http://www.zive.cz>

X je vektorem vstupů do neuronu dále je vektor vah W a také práh Θ . Práh má funkci aktivace neuronu. A to vše je vstupem do přenosové funkce, o aktivačních funkcích je blíže psáno v sekci 4.4. Model neuronové sítě je potom složen z n vrstev těchto umělých neuronů. Vážený součet vstupů udává vnitřní potenciál neuronu, na potenciál neuron reaguje výstupní odezvou. Ještě dodám, že samotný neuron dokáže řešit pouze lineárně separovatelné problémy.

3.2.2 Topologie

Část věnovaná topologii se zabývá její správnou volbou. U dopředné neuronové sítě každý neuron vstupní vrstvy přivádí svůj výstup ke všem neuronům vrstvy vyšší, pokud se nejedná o vrstvu výstupní, v tom případě se jedná o výstupní data samotné neuronové sítě. Z toho tedy vyplývá samotný název modelu této neuronové sítě.

Vícevrstvá neuronová síť je schopna řešit nelineární problémy. Konkrétněji k této práci je schopna aproximovat složité funkce a slouží jako obecný aproximační model. To ovšem neznamená, že počet vrstev by byl úměrný zlepšení výsledků. Pro řešení mnoha problémů je vhodné využít architekturu jedné vstupní, vnořené a výstupní vrstvy [7], pro představu je možné se podívat na ukázkou:



Obrázek 3.2: Příklad topologie

Dále je také nutné zvolit vhodný počet neuronů pro správnou generalizaci problému při trénování. To je netriviální problém a také je nutné specifikovat, o jaká vstupní data pro trénování se jedná. Neexistuje obecný postup, kterým je vhodné se řídit. Je možné využít genetických algoritmů pro nalezení vhodného počtu neuronů, pokud víme, že se bude jednat o relativně velkou síť nebo je možné prakticky začít s malým počtem a ten postupně zvyšovat do objemu, který bude schopný funkci aproximovat. Tím se dostávám k samotné podstatě, že s příliš malým počtem neuronů neuronová síť není schopna správné generalizace, na druhou stranu, pokud bychom zvolili příliš velký počet neuronů, může dojít k přetrénování.

Přetrénování neuronové sítě je časté, jedná se o problém, kdy neuronová síť vykazuje velmi dobré výsledky na trénovacích datech a při reálném nasazení na testovacích datech jsou výsledky nepřijatelné. V kapitole 4 částečně popisuji techniku, díky které je možné tento problém vyřešit.

Popsal jsem, co je možné si představit pod pojmem topologie neuronové sítě relativně k mnou používanému modelu dopředné vícevrstvé neuronové sítě a také několik zásad, kterými se budu řídit a také je prakticky ověřím 6:

1. Není vhodný nadměrný počet skrytých vrstev, využijí jednu nebo dvě.
2. Důležité je vhodně zvolit počet neuronů v každé vrstvě

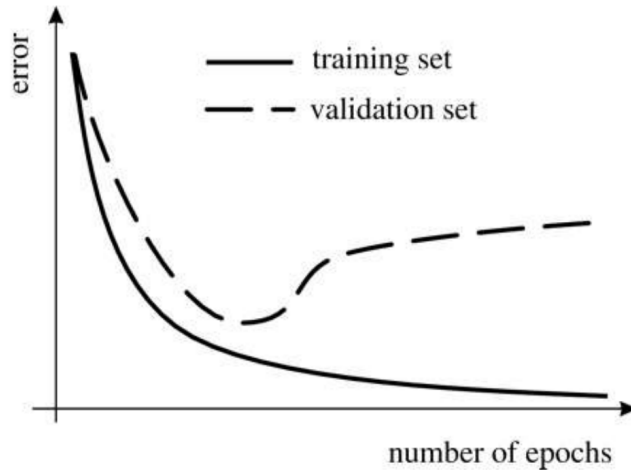
3.2.3 Trénování neuronové sítě

Důležitou vlastností je generalizace, ta je teoreticky dělena na odchylku a na varianci dat. Na začátku trénování je odchylka velká a variance zanedbatelná, ale v průběhu trénování by se měla odchylka zmenšovat a variance by neměla růst. Vhodný konec pro trénování je v době kdy součet variance a odchylky je nejmenší [15].

Trénování je možné chápat jako poslušnost dopředného šíření signálu, zpětné propagace chyby 3.2.4 a adaptace vah. Tento úkon je prováděn v cyklech, kdy jsou na vstup neuronové sítě přiváděny data a na výstupu jsou vyhodnocovány. Trénování neuronové sítě je možné rozdělit na učení s učitelem a bez učitele. První princip je využit, pokud máme k dispozici data k porovnání s výstupními daty neuronové sítě. Další možností je trénování bez učitele, kde přesně není známo co má neuronová síť generalizovat, v rámci této práce se tomuto přístupu nevěnuji.

Při trénování neuronové sítě je vhodné dostupnou datovou sadu rozdělit na část pro trénování a na část validační. Validační sada slouží k dynamickému vyhodnocování trénování. Jak jsem se již zmínil, trénování probíhá v cyklech (často je v literatuře používán termín epocha) a právě díky validační sadě, kterou neuronová síť nevyužívá pro samotné trénování (data validační sady jsou neviděná), je možné zjistit, zda neuronová síť stále generalizuje a také je možné předejít přetrénování na samotné trénovací sadě. Pro představu je možné se podívat na obrázek 3.3, který ukazuje snižování chyby v průběhu trénování pro sadu trénovací a validační, správné ukončení je vhodné provést v momentu, kdy se chyba validační sady nesnižuje.

Princip je založen na funkci $E(D^i, P(Z^i, W))$. Tato funkce symbolicky představuje vyhodnocení dat na výstupu neuronové sítě a dat referenčních, vyjadřuje závislost D^i a $P(Z^i, W)$. Evaluaci této závislosti popisují blíže v sekci 4, v této práci se jedná o vyhodnocení predikce, ale v širším smyslu je možné si za touto funkcí představit jakoukoli vyhodnocovací logiku specifickou k danému řešenému problému. Parametry funkce $P(Z^i, W)$



Obrázek 3.3: Validace, převzato z [10]

jsou vektor vstupních i -tých dat Z^i a vektor vah W . Tyto váhy jsou nastavitelné a v zpětné propagaci jsou ovlivňovány v závislosti na funkci $E(D^i, P(Z^i, W))$.

Zprůměrováním všech hodnot funkce E v rámci dat pro trénování funkcí E_{train} dostaneme hodnotu, která vypovídá o tom, v jakém stavu v rámci trénování náš model je. Čili podstatou je minimalizace funkce E_{train} na základě změn vektoru vah.

3.2.4 Zpětná propagace chyby

Vysvětlení tohoto algoritmu bude pomocí jednoduchého vícevrstvého modelu založeného na učení z gradientu. Každá vrstva je definována funkcí $F_n(W_n, X_{n-1})$. X_n je výstupem každé vrstvy a X_{n-1} je vstupem. W je vektorem vah, který je ovlivnitelný, což bude hrát v tomto algoritmu zásadní roli. Z^i 3.2.3 je vstupním vzorkem dat a je ekvivalentem X_0 . Tím je dáno, že je známa parciální derivace funkce E^p podle proměnné X_n . A pokud je možné spočítat parciální derivaci funkce E^p podle W_n , pomocí diferenčních rovnic je potom možné spočítat X_{n-1} .

$$\frac{\partial E^p}{\partial W_n} = \frac{\partial F_n}{\partial W}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n} \quad (3.1)$$

$$\frac{\partial E^p}{\partial X_{n-1}} = \frac{\partial F_n}{\partial X}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n} \quad (3.2)$$

Kde $\frac{\partial F_n}{\partial W}$ je Jakobian F_n podle proměnné W v bodě daném $((W_n, X_{n-1}))$ a $\frac{\partial F_n}{\partial X}$ je Jakobian podle proměnné X . Jakobian vektorové funkce obsahuje parciální derivace všech výstupů s ohledem ke všem vstupům. Pokud tento výpočet provedeme pro každou vrstvu reverzně pro chybovou funkci, budou spočítány všechny parciální derivace s ohledem na všechny parametry [15]. Speciálním případem tohoto systému je neuronová síť.

Kapitola 4

Teoretické poznatky pro úspěšné trénování

Po úvodu k dopředné neuronové síti 3.2, kterou využívám jako obecný aproximátor časové řady 5, chci dále uvést různé teoretické i praktické poznatky, které se týkají nastavení parametrů neuronové sítě, aktivačních funkcí, normalizace, ucelení jak celkový proces s využitím neuronové sítě funguje a v neposlední řadě ucelení spojitostí těchto poznatků v rámci kurzovních dat a obchodního modelu 5.

V prvních třech kapitolách pouze shrnu, co je myšleno parametry neuronové sítě v mé práci, přístup k trénování a důležitost promíchání trénovací sady. V dalších podkapitolách však uvádím pravděpodobně nejdůležitější informace, díky kterým byla neuronová síť schopna dobré generalizace a vykazovala adekvátní minimalizaci chyby. I přesto, že v rámci této práce přichází v potaz mnoho strategií pro celkový úspěšný výsledek, došel jsem na základě experimentů k závěru, že zásadní roli hraje správná transformace dat. To zahrnuje především správnou normalizaci a převod rozložení pravděpodobnosti kurzovních dat.

4.1 Stochastické a dávkové trénování

Jsou dva používané způsoby trénování, přesněji dva způsoby ustanovení gradientu. V praktické části 5 této práce je využíváno trénování po dávce. Existuje ještě stochastické trénování. To je v mnoha ohledech z hlediska teorie zajímavější [15].

Charakteristika trénování po dávkách ustanovuje daný gradient až po průchodu přes celou dávku dat. V praktické části bude ověřeno, že volba velikosti jedné dávky dat bude hrát roli v tom, jaký výsledek dostaneme.

U stochastického trénování je vybrán vždy jeden prvek a takové trénování je velice nestálé. Hlavní výhodou je především možnost sledování změny při trénování.

V této práci bylo zvoleno trénování po dávkách zejména pro jeho uchopitelné podmínky konvergence. Také je možno aplikovat několik akceleračních technik. Je možné se opřít o více teoretických postupů [15]. Například v praktické části bude otestována metoda, kdy bude neuronová síť trénována s malými dávkami a potom budou zvětšeny.

4.2 Náhodné promíchání trénovací sady

Trénink neuronové sítě je efektivní pokud jsou data na vstupu co nejvíce rozdílná. Neuronová síť by měla co nejvíce generalizovat a pokud bychom bezprostředně za sebou trénovali

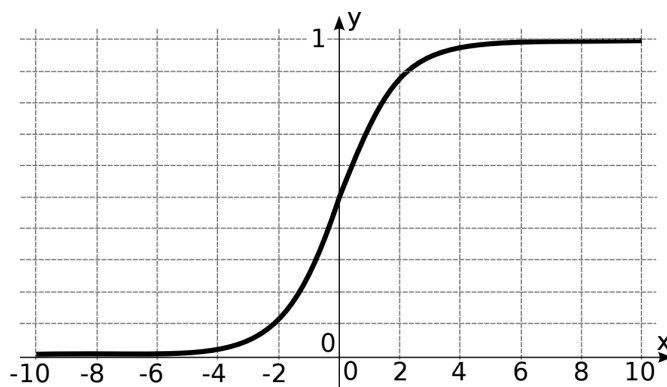
data podobná, tak se gradient ustanoví nepřesně a síť se celkově nenaučí v podstatě nic. Ovšem je těžké stanovit, která data jsou podobná a která ne. Proto je vhodným přístupem data náhodně zamíchat před procesem trénování. Také je možné v průběhu trénování opakovat proces u dat, která vykazovala na výstupu velkou chybu. Tato technika je vhodná, ale v této práci neuplatněná, neboť je riziková z hlediska přetrénování na těchto datech.

4.3 Nastavení parametrů

Zde shrnu důležité parametry a také jejich možné další nastavení. V první řadě učicí faktor určuje rychlost konvergence minimalizace chyby, ovšem z podstaty gradientní metody učení, toto nelze chápat způsobem, že čím vyšší bude hodnota učicího faktoru, tím menší bude výsledná chyba. Pokud si představíme jednoduchý váhový prostor, ve kterém hledáme minimum za pomoci numerické metody a iterační krok bude příliš velký v bodě, který se nachází v blízkosti minima, tak nebudeme schopni dojít ke správnému výsledku a konečná chyba nebude tak malá, jak by mohla být.

Dále jsem se také soustředil na parametry dávky určující ustálení gradientu. Pokud dávka bude příliš malá, tak průběh trénování bude velmi chaotický a nestálý, také výpočetní náročnost bude nepřiměřeně velká.

4.4 Aktivační funkce



Obrázek 4.1: Funkce sigmoidea [7]

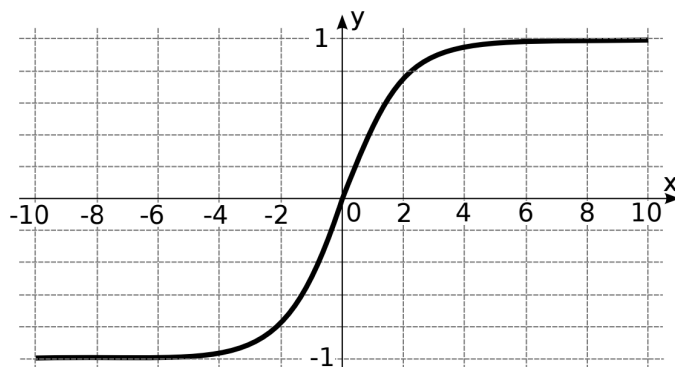
Hlavním prvkem, který dává neuronové síti možnost klasifikovat nelineární problémy, je nelineární aktivační funkce. Nejvíce užívané jsou dvě základní: *sigmoidea*

$$X_i = \frac{1}{1 + e^{-x}} \quad (4.1)$$

a *tanh* :

$$X_i = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (4.2)$$

S odkazem na normalizaci 4.5 je vhodné používat *tanh*, neboť jak je vidět na obrázku 4.2, rozsah oboru hodnot funkce *tanh* je mezi -1,1, na rozdíl od sigmoidy, která je pouze v kladných hodnotách a je využívána v případech, kde na výstupu očekáváme binární výstup



Obrázek 4.2: Funkce tanh [7]

nebo hodnoty pouze v kladných hodnotách. To, jaká aktivační funkce je vhodná, se odvíjí od charakteru dat, se kterými pracujeme. Výhodou těchto funkcí je jejich snadná derivace, což je v zpětné propagaci důležité.

Teorie také poukazuje na fakt, že obě funkce, a to zejména sigmoida, se drží v plochých zónách a učení je tím zpomalené. Proto je možné využít také upravené funkce: $f(x) = 1.7159 \tanh(\frac{2}{3}x)$ [15]. Hlavní výhodou je, že funkční hodnota v 1 je 1. Toto je opět spjato s normalizací dat. Díky tomuto rozsahu je první derivace větší v intervalu $(-1, 1)$. Také je možné zvolit přístup, kdy normalizaci provedeme tak, aby vstupní hodnoty nebyly v plném rozsahu $(-1, 1)$, ale například pouze v rozsahu $-0.8, 0.8$

4.5 Normalizace

Tato část je z hlediska mé práce zásadní. Zde uvedu pohled na problematiku obecných postupů, které je nutné vykonat pro úspěšný trénink. Jako první bych uvedl normalizaci dat. Vstupem pro trénování i pro referenční data by měla být data s nulovou střední hodnotou a jednotkovou variancí. Tento přístup je vhodný, pokud data mají normální rozložení, jak je dále vysvětleno. Pokud by data neměla nulovou střední hodnotu a byla například posunutá do kladných hodnot, pak by proces trénování byl zpomalen. Důvodem je, že váhy by byly ovlivněny pouze stejným znaménkem. Normalizaci je možné provést například podle tohoto vzorce:

$$X_i = \frac{X_i - \min(X)}{\max(X) - \min(X)} \quad (4.3)$$

Nebo:

$$X_i = \frac{X_i - \bar{X}}{\sigma} \quad (4.4)$$

Vzhledem k problémům s rozložením pravděpodobnosti časové řady měnového páru je v praktické části provedeno pouze odečtení průměru a každá hodnota je vynásobena konstantou pro zachování variance

$$X_i = (X_i - \bar{X}) * k \quad (4.5)$$

Konstanta k je zvolena tak, aby data byla v rozmezí $(-1, 1)$.

Vzhledem k času, který byl věnován při testech k tomu, abych našel správnou strategii normalizace a také fakt, že v rámci testování měla právě normalizace největší vliv na celkový výsledek oproti jiným postupům, kdy jsem zkoumal například změny topologie nebo faktoru učení, jsem usoudil, že je vhodné vše dopodrobna rozvést.

Kurzovní data predikovaného páru *EUR/USD* v obchodním modelu 5 tvoří časovou řadu, která není konečná. Je nutné se zamyslet nad uvedenými možnostmi normalizace 4.3, 4.4 a 4.5. V těchto výpočtech participuje například minimum, maximum nebo průměr celkové datové sady a u nekonečné řady tyto hodnoty není možné přesně určit. V práci [11] je časová řada normalizována v rámci jednoho obchodního dne, kde je předpokládáno, že jeden den je z hlediska charakteru trhu ukončená část. Nad využitím stejného postupu jsem uvažoval a zpočátku experimentování jsem předpokládal její výhodu zejména v tom, že neuronová síť bude lépe generalizovat data, kde v každém vektoru na vstupu bude malá odchylka. Toto jsem tedy dovedl do stavu, kdy jsem normalizoval každý trénovací vektor zvlášť.

I přesto, že následující odstavec spíše pojednává o průběhu experimentů, uvádím jej zde, neboť chci, aby odůvodňoval konečný výběr normalizace, kterou využívám v obchodním modelu 5.

Výsledky trénování, kdy jsem normalizoval každý trénovací vektor, byly nepřijatelné pro další využití. Pokles chyby byl velmi dobrý, ale neuronová síť nebyla schopna aproximovat funkci jako celek. Z toho důvodu jsem uznal za vhodné normalizovat data jako jeden celek podle uvedeného výpočtu 4.5. Důležitost zachování variance je vysvětlena v následující podkapitole.

Objasním, proč si mohu dovolit vypočítat průměr z celé datové sady a proč jsem schopný vypočítat konstantu k . Pokud získaná historická data pro trénování a testování mají celkové rozpětí minima a maxima vhodné vzhledem k posledním hodnotám časové řady, je možné konstantu k vypočítat, neboť můžeme předpokládat, že v blízké době nebude hodnota minima a maxima překročena. Na obrázku 6.2 jsem vytvořil ukázkou dat, která představuje historická data. Pokud jsou trénovací data od začátku grafu až po zelenou vertikální čáru v daném rozpětí, můžeme provést normalizaci podle vyznačeného minima a maxima modrými horizontálními čarami, neboť budoucí data budou v intervalu modrých čar. Dostane-li se hodnota dat při vyhodnocování testovacích dat nebo při reálném nasazení nad nebo pod hodnotu vyznačeného globálního minima resp. maxima, je nutné provést normalizaci a trénování znovu.



Obrázek 4.3: Zobrazení dat pro normalizaci, kde jsou znázorněny dvě rozdílné situace

4.6 Rozložení pravděpodobnosti kurzovních dat

Důležitým poznatkem je, že rozložení pravděpodobnosti signálu **eur/usd** není normální [12]. V kapitole 4.5 je psáno, že normalizace je nutná, ale není možné data transformovat tak, aby měla varianci jedna. V praktických testech jsem testoval různé přístupy normalizace a v testech, kde jsem měnil celkovou varianci dat, byly výsledky nejhorší. Ověřil jsem, že data nemají normální rozložení.

Podle zdroje [12] je možné uvažovat rozložení, které je podobné rozložení exponenciálnímu. Zde není aktivována **centrální limitní věta**¹, která u normálního rozložení splněna je.

Centrální limitní věta 1. *Nechť \bar{X} je průměr náhodného výběru z rozdělení se střední hodnotou μ a rozptylem $\sigma > 0$. Pak*

$$W = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} \quad (4.6)$$

má rozdělení $N(0, 1)$ pro $n \Rightarrow \infty$.

Aplikace: Pokud je n dost velké, pak:

$$\begin{aligned} W &= \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} = \frac{\frac{1}{n} \sum \bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} = \\ &= \frac{\frac{n}{n} \sum \bar{X} - \mu * n}{n * \frac{\sigma}{\sqrt[2]{n}}} = \frac{\sum \bar{X}_i - \mu * n}{\sqrt[2]{n} * \sigma} \end{aligned} \quad (4.7)$$

má přibližně standardní normální rozložení $N(0, 1)$.

Centrální limitní věta říká, že pokud náhodná veličina V z daného výběru může být vyjádřena jako součet náhodných proměnných složených z variancí, tak distribuce V je přibližně normální [12].

Při vyřešení tohoto problému, čili vytvoření dat, které budou v normálním rozložení a budou splňovat centrální limitní větu, budou výsledky predikce znatelně lepší.

Možností, která vedla k nejlepším výsledkům, bylo vytvoření rozdílu dvou hodnot časové řady, čili se jedná o časovou řadu diferencí dvou cen. Tato data následně splňovala podmínky popsané výše a jsou pro trénování vhodná, ale s tímto přístupem je třeba redukovat šum signálu.

$$d = H_i - H_{i-1} \quad (4.8)$$

Také podle článku [4] je možné využít logaritmickou diferenci pro zlepšení generalizace.

$$d = \ln \frac{P_i}{P_{i-1}} \quad (4.9)$$

Kde P_i je cena v čase t a P_{i-1} cena v čase $t - 1$.

¹Více informací je možné zjistit zde: <http://vychodil.inf.upol.cz/kmi/pras/pr09.pdf>

4.7 Redukce náhodného šumu

Při testech obchodního algoritmu hrál náhodný šum zásadní roli a silně ovlivňoval výstupní data neuronové sítě. V tomto případě, kdy využívám neuronovou síť jako obecný aproximátor, velmi často docházelo k problémům, kdy příliš velká variance predikci naprosto znehodnotila. Z tohoto důvodu jsem testoval několik možností, jak redukovat náhodný šum a zvýraznit hlavní trend.

Zde jsem uvažoval nad DCT (discrete cosine transform). Tato metoda je využita pro zpracování signálu. Je podobná jako Fourierova transformace ovšem bez imaginárních složek signálu. Využijeme převodu z časové oblasti do oblasti frekvenční. Tato metoda je široce využívána pro ztrátovou kompresi. Je využita například u JPEG. A v našem případě bude využita pro odstranění náhodného šumu eliminací vysokofrekvenční složky.

Dále také nad vyžitím PCA (Principal Component Analysis). Jde o nalezení lineární transformace ortogonálních bází. Transformace je využívána k dekorelaci dat a snížení dimensionalit se zachováním největší variability.

Tyto dvě metody jsem následně zavrhl, vzhledem k jejich výpočtu. Bázové vektory PCA jsou dány vlastními vektory kovarianční matice, která je počítána takto (převzato z [10]).

$$C = \frac{1}{N} \sum_{i=1}^N (x_i - m)(x_i - m)^T \quad (4.10)$$

Kde C je kovarianční matice získaná z N vstupních vektorů k dispozici pro trénování, x_i je i -tý trénovací vektor a m je odhadovaný střední vektor:

$$m = \frac{1}{N} \sum_{i=1}^N x_i \quad (4.11)$$

Vlastní hodnota spojená s jedním vlastním vektorem je reprezentována množstvím variability získané projekcí vstupního vektoru do daného vlastního vektoru. Proto pouze několik vlastních vektorů odpovídajících největším vlastním hodnotám je použito jako báze pro účely redukce dimenze.

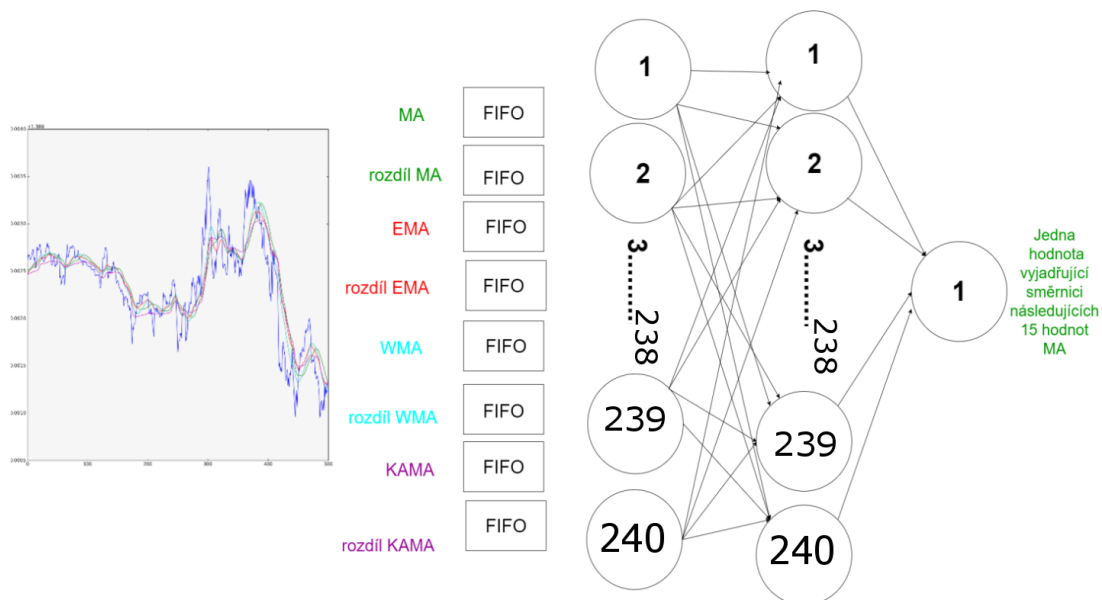
Z tohoto výpočtu je vidět, že pokud má být proveden, je nutné brát jisté informace z budoucnosti, pokud by nebyl aplikován pouze také v jistém časovém okně, ale zde je problém se zachováním celkové variance. Stejně tak je tomu v případě DCT, kde je výpočet takovýto.

$$m = \sum_{n=0}^N x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right) \quad (4.12)$$

Tím jsem chtěl objasnit, proč nakonec využívám klouzavé průměry. Ty jsou počítány pouze v daném okně. Využiji čtyř klouzavých průměrů (MA , EMA , WMA , $KAMA$) pro zachování co nejvíce informací ze samotné časové řady kurzovních dat. Zde je pouze problém jistého spoždění, který je řešen samotným návrhem logiky otevírání obchodních pozic 5.1.

Kapitola 5

Vytvořený obchodní model



Obrázek 5.1: Ukázka návrhu celého systému a uskupení dat na vstupu neuronové sítě.

Po testování byl sestaven konečný obchodní algoritmus, který je ziskový. Po 34 testech, kdy bylo kombinováno několik technických indikátorů, jsem došel k nejlepším výsledkům při modelu, který bude následně popsán.

V první řadě jsem sestavil matici klouzavých průměrů. Základní myšlenkou je predikce první derivace časové řady. Přesněji v tomto případě budu predikovat první derivaci klouzavého průměru. Ještě přesněji je pro každý časový úsek spočítána lineární regrese pro následujících patnáct hodnot. A směrnice této přímky je brána jako první derivace v daném bodě, což znamená, že je možné uvažovat v rámci patnácti minut, neboť v tomto případě bude obchodováno v jednominutovém timeframu.

Dále také aplikuji poznatky popsané dříve pro eliminaci problémů s aktivací centrální limitní věty. Dále jsem zjistil, že je také nutné neuronovou síť stále trénovat s daty, kde je zachována hodnota ceny v tom samém poměru jako v referenčním signálu. Na vstupu neuronové sítě je tedy spojený vektor diferencí klouzavých průměrů a také hodnot samotných. Diference je počítána takto:

$$D_{ma} = MA_n - MA_{n-1} \quad (5.1)$$

$$D_{ema} = EMA_n - EMA_{n-1} \quad (5.2)$$

$$D_{wma} = WMA_n - WMA_{n-1} \quad (5.3)$$

$$D_{kama} = KAMA_n - KAMA_{n-1} \quad (5.4)$$

Dále je spočítána první derivace klasického moving average, která je predikována, čili je na výstupu neuronové sítě. Pro upřesnění představy, jak jsou data na vstupu neuronové sítě uspořádána, je možné se podívat na obrázek 5.1.

Tento přístup je vhodný zejména proto, že na výstupu neuronové sítě je pouze jedna hodnota a trénování neuronové sítě je jednodušší. Také máme jasnou představu o tom, jaký bude trend.

5.1 Řízení obchodních pozic

```

threshold = 0.4
in_position_long = False
in_position_short = False
while derivativei not null do
  if derivativei > threshold then
    if in_position_long == False then
      set long position
      unset short position
      in_position_long = True
      in_position_short = False
    end
  end
  if derivativei < threshold * -1 then
    if in_position_short == False then
      unset long position
      set short position
      in_position_long = False
      in_position_short = True
    end
  end
end

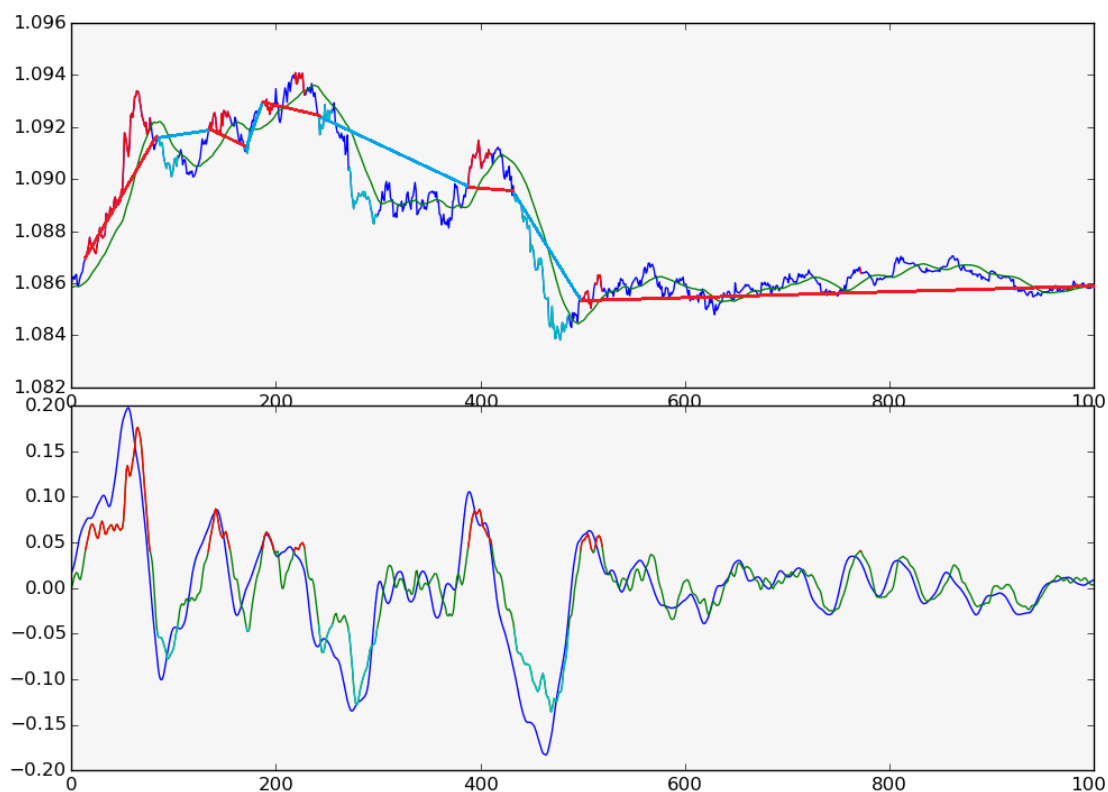
```

Algorithm 1: position set

Vstup a výstup z pozice bude volen na základě velikosti směrnice v daném bodě. Pokud bude směrnice větší než nastavená hodnota, bude nastavena pozice long, pokud již nastavena není. Opakem bude případ nastavení pozice short. Pokud bude hodnota směrnice menší než záporná hodnota prahu, tak bude pozice short nastavena.

Po natrénování neuronové sítě na datech o délce 603 189 (hodnoty close páru **EUR/USD**, které jsem byl schopen získat) jsem provedl test výdělků na 11 000 testovacích datech. Výběr tohoto páru je přímočarý, neboť jde o nejobchodovanější pár na forexu a je velmi stabilní. Jde nám o dlouhodobé obchody bez ukončování, abychom na spreadu zaplatili co nejméně. Proto je pozice ukončena až s nástupem další pozice a pokud je možnost otevřít například pozici long v době, kdy už v long jsme, tak nová pozice není otevřena. Toto je vysvětleno v pseudokódu 1.

5.2 Evaluace predikce



Obrázek 5.2: Ukázka obchodních pozic, kde v horní části grafu jsou kurzovní data. Červené zabarvení udává v jakých místech je překročen práh velikosti predikované směrnice, která je vidět v dolní části jako zeleno-červeno-modrá funkce. Modrá funkce je referenční pravá směrnice. Červené a modré přímky potom udávají pozice long a short.

V oboru strojového učení je rozhodující správné zhodnocení výsledků, které po natrénování obdržíme, v tomto případě zhodnocení predikce. Zde připadá v úvahu vhodná vizualizace dat, funkce pro vyhodnocení chyby predikce a sestavení algoritmu, který bude predikci hodnotit s ohledem na daný problém. Zde je mnoho problémů, například v této práci jsou nejvíce řešeny vlivy vysoké variance, které silně ovlivní výstup, pak také zpoždění predikce. To vše je nutné vhodně zpracovat a vzhledem k tomu, že predikce signálu

burzy je díky chaotickému charakteru dat velmi náročná a podstatou celé práce je ziskový obchodní algoritmus, usoudil jsem, že nevhodnější bude evaluace zisku, tak jak je uvedeno v předchozí kapitole 5.1.

Vše je vidět na obrázku 5.2, kde jsou vyznačené zobchodované pozice v podobě přímků a zabarvení kurzovních dat je pouze v místech, kde je překročen práh. Díky tomu, že jde o zobrazení pouze 100 hodnot, je zde dobře vidět, jak variance vstupních dat ovlivňuje predikci. Také je velmi dobře vidět, kde je překročen práh a proč je nutné otevřít pouze jednu pozici.

Dalším vyhodnocením je jednak samotná chyba na konci poslední epochy, ale v tomto případě hlavně hraje roli poměr chyby a počtu epoch. Zvolil jsem trénování, které je zastaveno dynamicky na základě změny chyby na datech pro validaci a díky tomu je dobře vidět zda neuronová síť má co generalizovat, čili pokud proběhne hodně epoch a chyba se bude stále zmenšovat, tak jdou data dobře generalizovat. Toto je demonstrováno v praktických testech 6.

Jako poslední uvedu ještě jeden způsob vyhodnocení predikce. Pokud hodnota na výstupu neuronové sítě určuje směrnici trendu v dalších 15 minutách, je vhodné zhodnotit, zda za 15 minut dojde ke správné predikci. S každou predikovanou hodnotou je vyhodnocen rozdíl M_t a M_{t+15} , kde M je hodnota klouzavého průměru, ze kterého je počítána první derivace. Dále je také vyhodnocen rozdíl P_t a P_{t+15} , kde P je kurzovní hodnota páru EUR/USD. Tyto rozdíly jsou vyhodnoceny relativně k tomu, zda směrnice naznačuje růst nebo pokles, pokud je rozdíl kladný je přičten k celkovému růstu. Na obrázku 5.3 je možné vidět vyhodnocení predikce růstu, kde modrá funkce vypovídá o kvalitě predikce samotného klouzavého průměru a zelená funkce ukazuje kvalitu predikce růstu samotné ceny P . Na této ukázce je vidět, že predikce neuronové sítě je kvalitní, ovšem klouzavý průměr je zpožděný o proti samotné ceně.

5.3 Bodový zisk

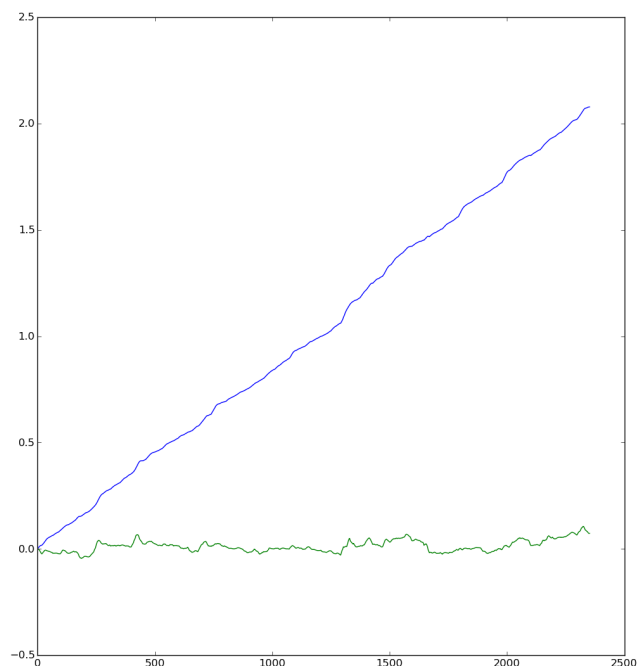
Udávat zisk v bodech jsem usoudil jako nejlepší, neboť konkrétně u páru EUR/USD je jeden pip deset bodů. Uvedený zisk v bodech poté může být vynásoben loty, které jsou základní nákupní jednotkou a zjistit výdělek například v dolarech v poměru k dostupnému kapitálu. Problematika bodů a lotů je blíže vysvětlena v tomto článku [2]

5.4 Využitá topologie

V mnoha případech je topologie sítě zásadní, v počátku testů jsem se zaměřoval právě na topologii a různé nastavení hodnoty `learning rate`¹. Různé změny však nebyly zásadní zejména z důvodů, které jsou popsány výše 4.6. Pro objasnění zde uvedu, že konečný `learning rate` byl nastaven na 0.08 a topologie sítě je tvořena ze vstupní vrstvy, kde je velikost rovna velikosti trénovacího vektoru, dále jedné skryté vrstvy o stejné velikosti jako vstupní a výstupní vrstva má pouze jeden neuron.

Závěrem je přidáno zobrazení 10 000 hodnot jako obrázek 5.4, kde jsou barevně označeny obchodní zóny. Červenou barvou jsou vyznačeny long pozice a světle modrou jsou pozice short. Spodní část grafu je věnována predikované směrnici, která určuje budoucí trend. Pro objasnění tam, kde je směrnice nulová, půjde trend do strany. Jde tedy o myšlenku, kdy vstoupíme do obchodu, pokud očekáváme jasné otočení trhu a pokud není predikován silný

¹Parametr, který určuje, jak moc jsou aktualizovány váhy v každém kroku zpětné propagace



Obrázek 5.3: Evaluace predikce růstu, kde osa x zaznamenává počet vyhodnocených dat a osa y přírůstek růstu v dané hodnotě. Je možné vidět vyhodnocení predikce růstu, kde modrá funkce vypovídá o kvalitě predikce samotného klouzavého průměru a zelená funkce ukazuje kvalitu predikce růstu samotné ceny P . Na této ukázce je vidět, že predikce neuronové sítě je kvalitní, ovšem klouzavý průměr je zpožděný o proti samotné ceně.

obrat, tak vše běží dál. Zisk v testech byl kladný a činil **2762 bodů**. Dále také uvádím chybu po provedení první a poslední epochy 5.1.

Jako poslední zmíním, že konečný výsledek je ovlivněn náhodnou inicializací vah. V příloze je přidán kód A, kterým je možné spustit zpracování dat a predikci s vahami stejnými jako v obchodním modelu. Vliv vah příliš neovlivnil konečnou chybu, ovšem díky mírným odchylkám v aproximaci první derivace jsou změny ve vyhodnocení pozic, tedy v zisku. Na základě daného páru a situace je možné změnit hodnoty prahů, kterými se řídí otevření pozic. Změny však nemají vliv na hlavní poznatky, o kterých je psáno v experimentech a v návrhu.

Tabulka 5.1: Průběh trénování

Epocha	čas trénování	chyba
Epoch 1	10s	loss: 0.0675
Epoch 14	8s	loss: 0.0124



Obrázek 5.4: V horní části jsou kurzovní data, kde červená značí pozice long a modrá pozice short. V podní části je predikovaná směrnice, kde červená a modrá označuje překročení prahu

5.5 Implementace

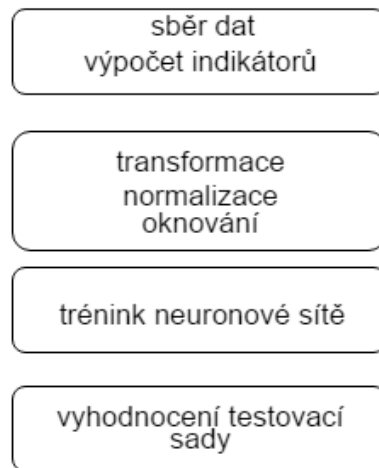
Pro přesnou představu, jak program funguje a jaké jsou jeho možnosti, zde uvedu stručný popis důležité funkcionality. Také bych chtěl v úvodu říci, že zde uvedu funkcionalitu relativně ke konečnému obchodnímu modelu i přesto, že celkové možnosti jsou širší, díky implementaci testů během vytváření konečného nastavení (viz příloha A.1).

Program pracuje chronologicky s postupem uvedeným na obrázku 5.5, to zahrnuje v první řadě získání historických dat pro trénink a pro testování navržené strategie s natrénovanou neuronovou sítí.

Získat historická data není zcela jednoduché, ale švýcarský broker [Dukascopy](https://www.dukascopy.com/)² některá poskytuje. Získal jsem minutová data páru **EUR/USD**, z kterých jsem vyfiltroval pouze hodnoty **close**, ze kterých jsou vypočítány klouzavé průměry, jak je uvedeno v kapitole 5. Transformace a normalizace zahrnuje postupy uvedené v kapitole 4.5 a 4.6.

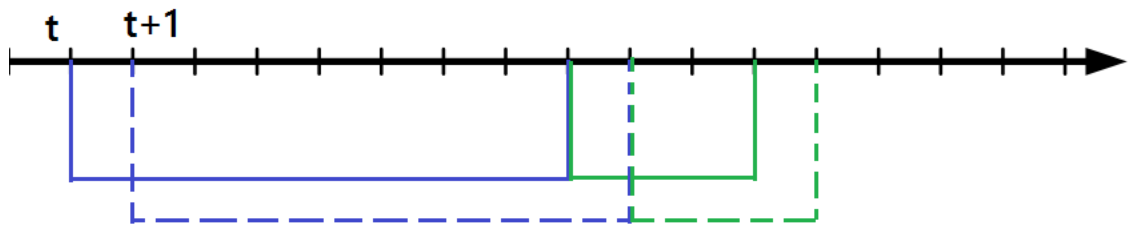
Ovšem princip oknování nebyl přesně popsán. Jednotlivé matematické operace jsou prováděny nad celou časovou řadou z důvodu zachování variance. Z připravených dat je poté třeba vytvořit matici vektorů, které budou v průběhu trénování předávány na vstup neuro-

²Více informací je možné zjistit zde: <https://www.dukascopy.com/>



Obrázek 5.5: Postup kroků prováděných během celého procesu

nové sítě. Vektory jsou získány aplikací klouzavého okna přes původní zpracovanou časovou řadu. Na obrázku 5.6 je vidět modře klouzavé okno pro trénovací vektor a k němu zeleně vytvořený výstupní vektor porovnání na výstupu v čase t a $t + 1$.



Obrázek 5.6: Modré klouzavé okno pro trénovací vektor a k němu zeleně vytvořený výstupní vektor porovnání na výstupu v čase t a $t + 1$

Další postup byl již přesně popsán v předešlých kapitolách. Trénink neuronové sítě využívá sestavenou matici nejen pro trénink, ale i pro následné testování. Z matice je ponechána část dat, která není využita v rámci trénování. V příložených kódech A je soubor s vahami tak, aby byl model stejný jako při implementaci této práce. Na této sadě je provedena simulace obchodů, která je popsána v kapitole 5.1 způsobem, kdy je každý vektor matice další ukončená hodnota close a je možné provést predikci, tedy získání směrnice na výstupu neuronové sítě.

Kapitola 6

Experimenty

Kapitola s experimenty by měla upevnit popsanou důležitost provedených matematických operací nad daty časové řady z kapitoly 4 a 5. U všech experimentů je uveden průběh trénování a zisk v bodech. Uvádění bodového zisku a možný převod na zisk v určité měně podle dostupného kapitálu byl odůvodněn v kapitole 5.4. Sledování průběhu trénování je nutný pro pochopení problému s rozložením pravděpodobnosti a s tím spojenou schopností generalizace neuronové sítě. Jak bylo psáno, ukončení trénování je provedeno dynamicky v případě, že se chyba po vyhodnocení na validační sadě dále nezlepšuje. Z toho vyplývá, že pokud jsou data dobře generalizovatelná pro neuronovou síť, je provedeno více epoch trénování.

6.1 Vypuštění diferencí

Tento test je nejdůležitější, proto bude uveden jako první. Zde jsem vynechal difference klouzavých průměrů, čili data pro trénování rozhodně neměla normální rozložení a zde bude vidět, že neuronová síť nebude schopná správně funkci generalizovat, pokud nesplníme tuto důležitou podmínku. Dalším důležitým bodem je tedy normalizace, ta je však ponechána správně. Potom co na vstupu neuronové sítě byly pouze hodnoty klouzavých průměrů, probíhal trénink takto. Zisk na testovacích datech činil **0 bodů** s průběhem trénování 6.1.

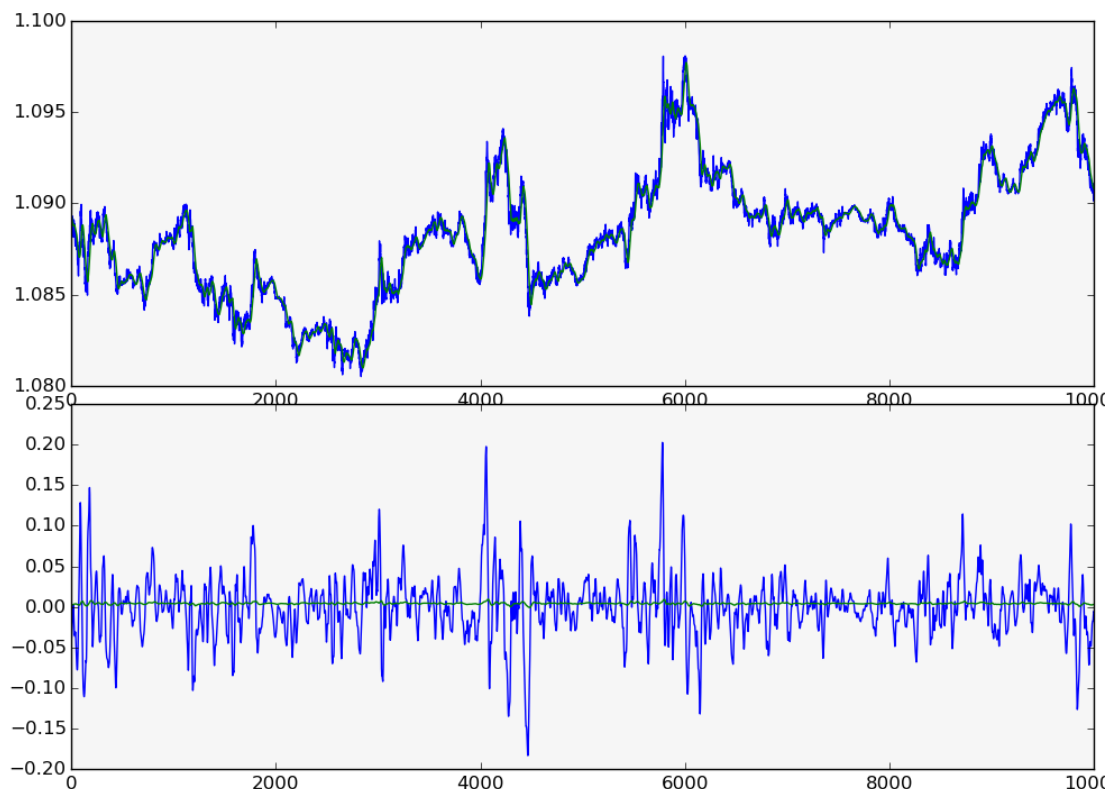
Z tabulky je vidět zřetelně méně provedených epoch proti nasazené strategii a to je přesný případ toho, o čem bylo psáno. Také jsem přidal obrázek 6.2 pro pochopení co se stalo. Z obrázku je jasně vidět, že síť nengeneralizovala skoro žádnou varianci a co se týče odchylky, ta je uspokojivá jen díky tomu, že hodnoty signálu první derivace jsou blízko nuly.

6.2 Pouze difference

Dále bych chtěl ukázat i test, kdy jsem provedl trénování pouze s diferencí klouzavých průměrů a také to, že zde už neuronová síť je schopná funkci generalizovat a je schopna ziskových obchodů. Na počtu epoch a poklesu chyby je vidět, že tento přístup je správný.

Tabulka 6.1: Vyloučení difference

Epocha	čas trénování	chyba
Epoch 1	10s	loss: 0.0733
Epoch 9	8s	loss: 0.0244



Obrázek 6.1: V horní části jsou kurzovní data s klouzavým průměrem. Důležitá je zde dolní část, kde je zobrazena referenční směrnice v porovnání se zelenou predikovanou směrnicí.

Zde také stále zachovávám normalizaci podle 4.5. Zisk na testovacích datech činil **1211 bodů** s průběhem trénování 6.2.

I přes to, že chyba je menší než v hlavní strategii, která je vybrána pro obchod jsou zisky menší a to 1211. Důvodem je, že data jsou pro neuronovou síť zásadní a jejich rozložení pravděpodobnosti je lepší než pouze hodnoty klouzavých průměrů, ale v predikovaných datech byl znatelný šum, který jsem eliminoval tím, že jsem vytvořil již prezentovanou strategii.

6.3 Přidání RSI

Při dalším testu jsem ponechal hlavní ziskovou strategii a k vektoru na vstupu jsem

Tabulka 6.2: Pouze difference

Epocha	čas trénování	chyba
Epoch 1	10s	loss: 0.0183
Epoch 37	8s	loss: 0.0118



Obrázek 6.2: V horní části jsou kurzovní data a v dolní predikovaná směrnice. Červeně jsou označeny pozice long a modře pozice short. Zde je vidět, že neuronová síť byla schopna funkci aproximovat.

přidal technický indikátor *RSI*. Ten je pro trénování vhodný proto, že jeho hodnoty jsou stále ve stejném rozmezí a je možné jej velice dobře normalizovat. Výsledky však nebyly s větším ziskem než u hlavní strategie a to **754 bodů**. Také zde uvedu průběh trénování a konečnou chybu 6.3.

6.4 Zhodnocení obchodního modelu a experimentů

Po provedených experimentech a popsání hlavní obchodní strategie 5 shrnu hlavní úskalí. Experimenty potvrdily, že je nutné vyřešit dva hlavní problémy, kterými jsou transformace dat do přibližně normálního rozložení pravděpodobnosti a redukce náhodného šumu. Pokud byla provedena pouze transformace, čili trénování pouze s diference klouzavých průměrů, tak strategie vykazovala menší zisk oproti hlavnímu návrhu, kde jsou na vstup přivedeny i

Tabulka 6.3: Přidání RSI

Epocha	čas trénování	chyba
Epoch 1	10s	loss: 0.0675
Epoch 14	8s	loss: 0.0124

klouzavé průměry a tím je vstup vyhlazen.

I přes to, že využití klouzavých průměrů přináší popsané výhody, je zde problém se zpožděním klouzavého průměru oproti signálu kurzovních dat. Proto je nutné vstupovat pouze do pozic, kde očekáváme silný dlouhodobý trend, jak je vysvětleno v kapitole 5. Proto v experimentech neuvažuji o jiném návrhu logiky otevírání pozic.

Poslední experiment také potvrzuje nabyté informace z článku [5], v kterém je naznačeno, že zvyšování počtu dat na vstupu neuronové sítě není, tak zásadní, jako spíše jejich vhodný výběr a zpracování.

Kapitola 7

Závěr

Podstatné body, které bylo nutné vyřešit, se týkají rozložení pravděpodobnosti a eliminací náhodného šumu v kurzovních datech. Pokud vezmu v potaz celkový vývoj konečného návrhu, všechny testy a kombinace, tak jsem mnoho času věnoval úpravám topologie, parametrů a samotných modelů neuronových sítí. Však tyto kroky nebyly podstatné oproti návrhu správného výstupu neuronové sítě a především úspěšné transformaci dat. Také normalizace měla vliv na to, zda neuronová síť bude schopna dobré aproximace časové řady. Ta splňovala svůj účel, pouze pokud byla data transformována do přibližně normálního rozložení 4.6. Po vyřešení problému s neschopností generalizace, bylo třeba vyřešit ovlivňování predikce náhodným šumem, zde jsem se snažil využít možnosti často užívané v oblasti strojového učení pro redukci dimensionalit či eliminaci vysokofrekvenčních složek 4.7, avšak nejlépe fungovalo využití klouzavých průměrů. I přes to, že je výsledný obchodní model ziskový, je zde mnoho možností pro další zlepšení.

Text považuji za důležitý pro čtenáře, který by chtěl systém dále rozvíjet nebo pracovat na systému podobném. Popis průběhu práce popsany v předchozím odstavci jasně popisuje, které části by bylo vhodné dále rozvíjet. Dále na základě této práce se domnívám, že není vhodné vynakládat snahu do predikce přesné ceny. Ale je vhodné navrhnout strategii, tak aby na výstupu neuronové sítě byla jedna hodnota, která bude predikovat přesnou událost například silný obrat trendu, tak je navrženo v obchodním modelu 5.1 nebo vytvoření strategie, která bude trénována pouze na datech po nějaké fundamentální úvaze například trénování a predikce pouze na datech v jisté časové zóně otevírání obchodních center nebo vyhlásování ekonomických zpráv.

Dalším možným rozšířením, který byl nad rámec této práce, je využití optimalizačních algoritmu pro úpravu technických indikátorů nebo jiných dat na vstupu neuronové sítě. Dále využití numerických metod pro přesnější modelování rozložení pravděpodobnosti kurzovních dat a následnou transformaci.

Literatura

- [1] nad B. Eddy Patuwo, G. Z.; Hu, M. Y.: *Forecasting with artificial neural networks:: The state of the art [online]*. Graduate School of Management, Kent State University, Kent, Ohio 44242-0001, USA, 1998, doi:10.1016/S0169-2070(97)00044-7, [cit. 2016-02-30]
<http://www.sciencedirect.com/science/article/pii/S0169207097000447>.
- [2] BOSSA: Bod, spread, lot [online]. 2012, [cit. 2016-03-30].
URL <http://bossa.cz/cs/vzdelani/forex-uvod/bod-spread-lot>
- [3] Ding, Y.; Cai, Y.; Sun, P.; aj.: *The Use of Combined Neural Networks and Genetic Algorithms for Prediction of River Water Quality*. Journal of Applied Research and Technology [online]., 2014, doi:10.1016/S1665-6423(14)71629-3, [cit. 2016-02-15]
<http://linkinghub.elsevier.com/retrieve/pii/S1665642314716293>.
- [4] EVANS; Cain; PAPPAS, K.; aj.: Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation [online]. Mathematical and Computer Modelling, [cit. 2016-03-21]
<http://www.sciencedirect.com/science/article/pii/S0895717713000290>.
- [5] EYDEN: *The Application of Neural Networks in the Forecasting of Share Prices*. Finance and Technology Publishing, 1996, ISBN 978-0965133203.
- [6] Iebeling Kaastraa, M. B.: *Designing a neural network for forecasting financial and economic time series [online]*. Elsevier B.V., 23 March 1995, doi:10.1016/0925-2312(95)00039-9, [cit. 2016-02-30]
<http://www.sciencedirect.com/science/article/pii/S0169207097000447>.
- [7] KAČER, P.: *FOREXOVÝ AUTOMATICKÝ OBCHODNÍ SYSTÉM ZALOŽENÝ NA NEURONOVÝCH SÍTÍCH*. Diplomová práce, VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2015, vedoucí: doc. Ing. VÁCLAV JIRSÍK, CSc. https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=99820.
- [8] Kathy, L.: *FOREX – Ziskové intradenní a swingové obchodní strategie*. FXstreet.cz s.r.o., 2011, iISBN: 978-80-904418-2-8.
- [9] Mena, R.; Rodrígueza, F.; Castillaa, M.; aj.: *A prediction model based on neural networks for the energy consumption of a bioclimatic building*. Energy and Buildings [online]. Addison-Wesley UK, 2014, doi:10.1016/j.enbuild.2014.06.052, [cit. 2016-03-1]
<http://www.sciencedirect.com/science/article/pii/S0378778814005349>.

- [10] MIKULENČÁK, R.: PREDIKCE KURSŮ PRO OBCHODOVÁNÍ NA AKCIOVÝCH TRZÍCH. 2011, vedoucí: Doc. Dr. Ing. Jan Černocký <http://www.fit.vutbr.cz/study/DP/DP.php?id=17801&file=t> [cit. 2016-03-26].
- [11] PUTNA, L.: *PREDIKCE VÝVOJE KURZU POMOCÍ UMĚLÝCH NEURONOVÝCH SÍTÍ*. Diplomová práce, VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2015, vedoucí: Ing. Igor Szóke, Ph.D. https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=117974.
- [12] SHARPE, M. J.: LOGNORMAL MODEL FOR STOCK PRICES [online]. MATHEMATICS DEPARTMENT, UCSD, [cit. 2016-01-18] <http://math.ucsd.edu/~msharpe/stockgrowth.pdf>.
- [13] Sutskever; Ilya; Vinyals; aj.: Sequence to Sequence Learning with Neural Networks [online]. In *Advances in Neural Information Processing Systems 27*, editace Z. Ghahramani; M. Welling; C. Cortes; N. D. Lawrence; K. Q. Weinberger, Curran Associates, Inc., 2014, s. 3104–3112, [cit. 2016-02-30]. URL <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- [14] Wahiba, Y.; Evgueniy, E.: *Performance prediction of a solar thermal energy system using artificial neural networks*. Applied Thermal Engineering [online], 2014, doi:10.1016/j.applthermaleng.2014.07.040., [cit. 2016-02-3] <http://www.sciencedirect.com/science/article/pii/S1359431114006164>.
- [15] Yann, L.; Leon, B.; Genevieve, B. O.; aj.: Efficient BackProp [online]. Dec 1998, [cit. 2016-01-5] <http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf>.

Přílohy

Seznam příloh

A Spuštění a popis funkcí	38
A.1 Vysvětlení celkového konceptu	39
A.2 Obsah CD	39

Příloha A

Spuštění a popis funkcí

Spuštění zpracování a trénování je možné pokud spustíme **main.py**. Zde jsou různé proměnné pro nastavení zpracování (ty jsou dále komentovány v samotném kódu) a načtení získaných historických dat. Z nich je vybrána hodnota *close*, ze které jsou dále počítány klouzavé průměry pomocí funkce **getIndicators**.

```
getIndicators(self, cena, matice, tech_list,
              set_plot, delka_zobrazeni, kn, area)
```

Parametr *cena* je numpy pole, ze kterého je technický indikátor vypočítán. *Matice* slouží pro předání proměnné, do které je uložen výsledek. Nejdůležitějším je však *tech_list*, jde o seznam s volbou technických indikátorů, které chceme spočítat. Další parametry *set_plot* a *delka_zobrazeni* slouží pro grafický výstup při debugování. Předposlední parametr *kn* předává pomocnou knihovnu s důležitými funkcemi a poslední určuje délku klouzavého okna, ve kterém je indikátor počítán. Tedy pomocí funkce *getIndicators* vytvořím potřebné klouzavé průměry.

```
ma_price = self.getIndicators(price_mat, tech_mat,
                              ['ma'], False, view_len, pc, 30)
ema_price = self.getIndicators(price_mat, tech_mat,
                              ['ema'], False, view_len, pc, 30)
wma_price = self.getIndicators(price_mat, tech_mat,
                              ['wma'], False, view_len, pc, 30)
kama_price = self.klouzaveokno(price_mat, tech_mat,
                              ['kama'], False, view_len, pc, 30)
```

Klouzavé průměry jsou následně zpracovány pomocí funkce **priceProces**.

```
priceProces(self, price, metod, metod_param,
            kn, cook)
```

Parametr *price* opět reprezentuje numpy pole určené ke zpracování. Další dva parametry *metod* a *metod_param* jsou seznamy. První určuje v jakém rozmezí budou data zpracována, druhý potom nastavuje jaké operace budou provedeny a v jakém pořadí. Parametr *kn* opět předává pomocnou knihovnu, která je popsána dále a parametr *cook* je využíván pouze pokud nezpracováváme data jako celek, čili pro konečný návrh není podstatný.

Funkcí *priceProces* tedy spočítáme klouzavé průměry *MA*, *EMA*, *WMA*, *KAMA* a jejich diference. Zde uvedu pouze využití funkce pro jeden vektor klouzavého průměru, neboť ostatní tři jsou počítány stejně.

```

ma_price_tr_mat = self.priceProces (ma_price , [ 'all_data ' ] ,
                                     [ 'cut_mean ' , 'pull ' ] , pc , True)
ma_price_tr_mat_b = self.priceProces (ma_price , [ 'all_data ' ] ,
                                       [ 'price_diff ' , 'cut_mean ' , 'pull ' ] , pc , True)

```

Jak je vidět pomocí *all_data* nastavíme zpracování na celé datové sadě. Nejdříve je provedeno odečtení odchylky a úprava podle 4.5. Také je vidět, že v druhém případě je počítána diference. Funkce vrací matici trénovacích vektorů.

V poslední fázi jsou vytvořeny data pro porovnání na výstupu neuronové sítě, k tomu je určena funkce **targetProces**.

```

targetProces (self , data , metod , gap ,
              cut , kn ):

```

Parametr *data* opět předává data pro zpracování. Pomocí *metod* je nastaveno, jakým způsobem chceme výstup neuronové sítě na výstupu porovnávat. Za zmínění také stojí parametr *gap*, který nastavuje, zda chceme rozstup mezi vstupním trénovacím vektorem a vektorem na výstupu. Parametr *cut* je také nutné zmínit. Aplikace funkce *getIndicators* způsobí jisté zkrácení, neboť pokud v tomto konkrétním případě počítáme klouzavé průměry v intervalu 30 hodnot, tak výsledný vektor je o tuto hodnotu zkrácen, hodnotu je nutné předat jako parametr. Ovšem díky tomu, že v konečném modelu je na výstupu přibližná první derivace, tak je tato hodnota nastavena na 0.

```

targetProces (ma_price , 'smernice ' , 0 , 0 , pc)

```

Také je nutné říci, že stejně jako je vektor zkrácen výpočtem klouzavých průměrů, tak je zkrácen výpočtem diferencí o jednu hodnotu.

V konečné fázi je inicializován **KerasSystem**, který zajistí predikci s použitím modelu. Vyhodnocení je nutné sputit zvlášť pomocí *EvalClass.py*, princip je popsán v kapitole 5.1.

A.1 Vysvětlení celkového konceptu

V první části jsem popsal, jak jsou funkce využívány v konečné části. Důvodem proč jsou zde i jiné parametry a možnosti volby, je celkový proces experimentování. Vše je blíže okomentováno v samotném kódu. Širší využití uvedu jen velmi stručně, neboť by tato práce zbytečně nabyla na textu, který není podstatný vzhledem ke konečnému použití, ale chtěl jsem zveřejnit celistvý kód, kde je mnoho dalších užitečných možností.

Například kromě celého signálu je možné provést normalizaci pouzera každý trénovací vektor zvlášť, nebo kombinovat a implementovat nové zpracování. V souboru **ProcessClass** jsou implementovány funkce pro vhodné zobrazení dat nebo také výpočty samotných matematických operací je zde implementována možnost aplikace *DCT*, různé přístupy normalizace, prahování hodnot pro realizaci klasifikátoru a různé další možnosti.

Pro spuštění **main.py** a **EvalClass.py** je nutné nainstalovat popsané knihovny 3.1 a další nástroje využívané těmito knihovnami.

A.2 Obsah CD

Zdrojový text latex soubory pro vysázení technické zprávy.

Technická zpráva konečná technická zpráva.

Readme soubor s popisem spuštění a potřebných knihoven.

Zdrojové kódy kódy pro spuštění testu a potřebné váhy.

Plakát pro prezentaci.

Video prezentuje spuštění zdrojových kódů.

Licence pro možnosti použití.