Investiční zlato – IT podpora
Gold investment – IT support

Diplomová práce

Zdeněk Zahoř

Vedoucí diplomové práce: Ing. Jan Jára, Ph.D.
Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky
2011

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Pedagogická fakulta
Akademický rok: **2009/2010**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Zdeněk ZAHOŘ**
Studijní program:	**M7504 Učitelství pro střední školy**
Studijní obory:	**Učitelství matematiky**
	Učitelství výpočetní techniky
Název tématu:	**Investiční zlato - IT podpora**
Zadávající katedra:	**Katedra informatiky**

Zásady pro vypracování:

Investiční zlato je jedna z komodit, kterou lze nakupovat i v malém množství a to i přes internet. Primárně se ceny zlata v České republice v české měně mění podle ceny zlata na světové burze a kurzu koruny vůči dolaru a euru. Cílem práce je tvorba aplikace, která se pokusí předikovat vývoj cen v národním prostředí.
Práce v úvodu stručně seznamuje s historií a specifiky zlata jako investičního prostředku. Dále jsou zpracovány teoretické nástroje pro predikce ceny. Tyto teoretické nástroje jsou použity při vývoji výsledné aplikace. Aplikace získává data pomocí datových pump (textový parsing, ODBC, .NET Web Services), uchovává a dále pak zpracovává pomocí metod umělé inteligence.
Aplikací jsou zkoumány faktory ovlivňující cenu zlata a poté je vyhodnocena úspěšnost predikce. Součástí práce je administrátorská a uživatelská příručka.

Rozsah grafických prací:

Rozsah pracovní zprávy: **60**

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

1. **BOCKER, Hans J. Svoboda jménem zlato : Vzpoura ve světě papírových peněz. Praha : Austria Gold CZ, 2009. 147 s.**
2. **NOVÁK, Mirko, et al. Umělé neuronové sítě : teorie a aplikace. Praha : C.H. Beck, 1998. 382 s.**
3. **PARR-RUD, Olivia. Data Mining : praktický průvodce dolováním dat pro efektivní prodej, cílený marketing a podporu zákazníků (CRM). Praha : Computer Press, 2001. xxvii, 329 s.**
4. **NOVÁK, Vilém. Základy fuzzy modelování. Praha : BEN - technická literatura, 2000. 175 s.**

Vedoucí diplomové práce: **Ing. Jan Jára, Ph.D.**
Katedra informatiky

Datum zadání diplomové práce: **24. listopadu 2009**

Termín odevzdání diplomové práce: **30. dubna 2011**

L.S.

doc. PhDr. Alena Hošpesová, Ph.D.
děkanka

PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 24. listopadu 2009

Prohlášení

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 28.4.2011

….....

Zdeněk Zahoř

Anotace

Diplomová práce se zabývá zkoumáním predikce pomocí umělých neuronových sítí. Na rozdíl od prací zabývajících se podobným tématem, řeší tato práce i samotný sběr dat, na kterých je úspěšnost predikce testována. Konkrétně se jedná o data související s investičním zlatem. Přes internetové obchody lze nakupovat zlato ve formě zlatých slitků. Sami prodejci uvádějí, že jejich ceny se odvíjí od ceny zlata na trhu a kurzu koruny. Cílem práce je posoudit predikční potenciál v těchto datech.

K dosažení cíle byl vytvořen systém pro získávání potřebných dat do datového skladu, jejich předzpracování a vizualizaci. Dále byl vytvořen systém předpovídající ceny, a pomocí něj byla zkoumána úspěšnost predikce.

Klíčová slova

umělé neuronové sítě, predikce, investiční zlato

Abstract

This thesis is about the investigation of the prediction with help of artificial neural networks. It contrasts other papers considering similar topic. This work is focused on the data collected to predict future outcomes. The data specifically deals with gold capital. It is possible to buy gold via the internet in a form of gold bars. The dealers themselves state that the price of gold is derived via the global prize index of gold and from the current value converted to the Czech Koruna rate of exchange. The aim of this work is to consider price forecast correlating input data.

To achieve the desired outcome a system for the data collection, processing, and visualization has been designed. A system dealing with price prediction has been developed to show the accuracy of the results obtained.

Keywords

artificial neural networks, prediction, gold investment

Poděkování

Rád bych poděkoval panu Ing. Janu Járovi, Ph.D. za vedení diplomové práce a čas strávený při konzultacích.

Obsah

1 Úvod

Investovat do zlata je možné dvěma základními způsoby. První možností je investovat virtuálně na burze. To je většinou investice krátkodobá, která je podobná spekulaci s měnami. Druhou možností je investování do fyzického zlata – zlatých mincí a slitků. Zlaté slitky je možné nakupovat přes internet a sami prodejci uvádějí, že jejich ceny se odvíjí o ceny zlata na trhu a kurzu dolaru vůči české koruně. Motivací této práce je vytvořit IT podporu pro tento druh investování. Nejzajímavější podporou pro investora by byla možnost předpovídat vývoj cen zlatých slitků.

Cílem práce je vytvořit softwarový nástroj, kterým je možné prozkoumat predikční potenciál v datech souvisejících s investičním zlatem. Predikování je prováděno pomocí umělých neuronových sítí a na rozdíl od prací na podobné téma je v této práci řešen i samotný sběr dat.

V úvodu práce stručně seznamuje s historií a specifiky zlata jako investičního prostředku. Dále jsou představeny teoretické nástroje, potřebné k realizaci: časové řady a jejich predikce, umělé neuronové sítě a teorie prostředků použitých při sběru a předzpracování dat. V další části je popsáno řešení tvorby výsledného softwaru. Dále je popsána charakteristika nasbíraných dat a zkoumání vytvořeným nástrojem. V závěru je zhodnocení úspěšnosti predikce. Součástí práce je v příloze obsažená administrátorská a uživatelská příručka. Přiloženy jsou také grafy, výsledky výpočtů pomocí vytvořeného nástroje a elektronický nosič (CD). Ten obsahuje text této práce, zdrojové kódy a bázi nasbíraných dat.

2 Investiční zlato

„*Reálná cena zlata je stabilní už 2600 let. Už tisíce let je jedním z oblíbených způsobů investování obchod se zlatem.*“ [1] Hlavními vlastnostmi zlata, díky kterým je tak oblíbené, jsou:

- zlato nepodléhá korozi,
- zlato září jako Slunce,
- zlato se dobře tvaruje,
- zlacení je nejlepší barvení,
- zlata je omezené množství (vlastnit něco, co ostatní mít nemohou, je přitažlivé),
- zlato je jako pojistka – jeho hodnota přetrvá. [2]

Fyzicky lze do zlata investovat nákupem zlatých slitků (cihliček). Vyrábějí se v různých hmotnostech od 1 g až po 1 kg. Tradiční hmotností je troyská unce (Oz), která odpovídá přibližně 31,1 g. Zlaté slitky lze nakupovat ve specializovaných obchodech a to i přes internet. Prodejci stanovují svoje ceny v závislosti na ceně zlata na londýnském trhu a kurzu dolaru vůči koruně. Významnou hodnotou ceny zlata na trhu je dvakrát denně stanovovaný London Gold Fixing (londýnský zlatý fix). Je vyhlašován nejvýznamnějšími obchodníky se zlatem a vztahuje se na obchody o objemu minimálně 1000 Oz. Součástí ceny je také výrobní přirážka („premium“). Dalším specifikem investičního zlata je jeho osvobození od DPH. [1]

O zlato je v poslední době zvýšený zájem z důvodu nedůvěry v globální finanční systém. Zajímavým důkazem tohoto zájmu jsou například automaty na zlaté slitky (tzv. „zlatomaty“).

3 Teoretické nástroje potřebné k realizaci

3.1 Časové řady a jejich predikce

Časovou řadou [3] se rozumí posloupnost hodnot ukazatelů, měřených v určitých časových intervalech a je možno ji formálně zapsat jako posloupnost:

$$y_1, y_2, y_3 \cdots y_n$$

neboli:

$$y_t, t = 1..n$$

kde n je celkový počet pozorování a y_t je hodnota pozorované veličiny v čase t. Obecně jsou časové řady výsledkem empirického sledování nějaké veličiny. Pokud jsou těmito veličinami ceny (např. cena měny, dluhopisu, akcie), označují se jako finanční časové řady [4]. Typickým příkladem finanční časové řady je tedy například vývoj kurzu CZK k USD v nějakém časovém období. Finanční časové řady patří mezi obecnější ekonomické časové řady, které lze různými způsoby třídit:

Podle charakteru ukazatele na **intervalové** a **okamžikové** časové řady. Hodnoty intervalové časové řady závisí na délce pozorovaného intervalu (např. měsíční náklady). Naopak okamžikové časové řady vyjadřují nezávisle na délce pozorování hodnotu v konkrétním okamžiku (např. počet zaměstnanců k určitému dni).

Podle doby sledování na **dlouhodobé** a **krátkodobé**. Dlouhodobé časové řady jsou výsledkem sledování delším než jeden rok, krátkodobé pak odpovídají sledováním půlročním, čtvrtletním, týdenním atd.

Podle druhu měření hodnot na časové řady s **absolutními** nebo **odvozenými ukazateli**. Odvozeným ukazatelem je například produktivita práce, neboť je získána jako poměr produkce a počtu zaměstnanců. [5]

Podle délky intervalu mezi jednotlivými měřeními na časové řady **ekvidistantní** a **neekvidistantní**. Ekvidistantní časové řady mají stejně dlouhý krok mezi pozorováními. Například ČNB stanovuje kurzy každý den ve 14:15. Jelikož ale toto stanovování neprobíhá o víkendech a svátcích, nejsou časové řady těchto kurzů ekvidistantní. Někdy je pak potřeba u takových řad provádět doplněním hodnot převod na řady ekvidistantní.

3.1.1 Principy predikce

Obecně se predikcí myslí předpovídání něčeho, co bude na základě informací z minulosti. Přesněji řečeno je predikce časové řady určování budoucích hodnot na základě již naměřených hodnot. [6]

Formálně tedy jde o hledání funkce, jejíž funkční (predikované) hodnoty jsou závislé na hodnotách minulých. [7]

Existuje mnoho způsobů jak takovou funkci (model) hledat. Patří mezi ně statistické přístupy nebo metody umělé inteligence.

Na predikci lze pohlížet dvěma způsoby:

- predikce „ex-ante“ (extrapolace) je předpověď doposud neznámých hodnot,
- predikce „ex-post“ (interpolace) je předpověď již známých hodnot, sloužící většinou ke kontrole spolehlivosti modelu. [5]

Pro zlepšení výsledků predikce nějaké časové řady je vhodné použít ještě doplňují informace z dalších časových řad, které s předpovídanou řadou souvisejí. Těmto řadám se říká **intervenční řady**. Pro předpovídání ceny zlatých slitků jsou těmito řadami údaje o ceně zlata na trhu, kurzu dolaru a mohly by to být například údaje o poptávce po zlatě, politické situaci atd.

Kritériem, podle kterého se posuzuje úspěšnost predikce, je odchylka skutečných a predikovaných hodnot. Je více způsobů, jak tyto chyby určovat. V této práci se v experimentech používají následující dvě:

- odmocnina ze střední čtvercové chyby (Root Mean Square Error)

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

- střední absolutní odchylka (Mean Absolute Deviation)

$$MAD = \frac{1}{n} \cdot \sum_{t=1}^{n} |y_t - \hat{y}_t|$$

V obou případech je y_t skutečná hodnota, $\hat{y}_t$ předpovídaná a n počet předpovídání.

3.1.2 Korelační koeficient

Dalším kritériem pro posuzování úspěšnosti predikce je korelace mezi řadou reálných (skutečných) hodnot a řadou predikovaných hodnot. Obecně korelační koeficient r vyjadřuje, jak moc jsou na sobě lineárně závislé dvě veličiny. Určí se jako:

$$r = \frac{S_{XY}}{\sqrt{S_X^2 S_Y^2}}$$

kde S_X^2 a S_Y^2 jsou výběrové rozptyly veličin X, Y a S_{XY} je kovariace. Nabývá hodnot od -1 do 1. Pokud je rovný 1, jsou veličiny zcela lineárně závislé a zároveň rostou nebo klesají. Pokud je rovný -1, jsou veličiny také zcela lineárně závislé, ale pokud jedna roste, tak druhá klesá a naopak. Pokud je korelační koeficient roven 0, jsou veličiny nezávislé. [8]

3.1.3 Analýza časových řad

Cílem analýzy časové řady je tvorba modelu, který co nejlépe popisuje danou časovou řadu. Takový model umožňuje porozumět chování a vzájemnému vztahu pozorovaných veličin. Následně podle tohoto modelu je možné předpovídat budoucí vývoj a podle toho se rozhodovat.

Mezi základní metody analýzy časových řad patří:

- dekompozice časové řady,
- Boxova-Jenkinsova metodologie,
- lineární dynamické modely,
- spektrální analýza časových řad. [3]

Dekompozice je rozložení řady na trendovou, cyklickou, sezónní a nepravidelnou složku. Boxova-Jenkinsova metodologie bere v úvahu závislost (korelaci) různých pozorování. Lineární dynamické modely zahrnují další vysvětlující faktory a konečně spektrální analýza znamená pohled na řadu jako na složení sinusových a kosinusových křivek.

3.1.4 Dekompozice časové řady

Tato metoda předpokládá, že každou časovou řadu je možné rozložit na čtyři složky.

Trendová složka T_t vyjadřuje dlouhodobou tendenci vývoje, která je většinou závislá na stálých procesech. Řada může mít trend rostoucí, klesající nebo být bez trendu.

Sezónní složka S_t vyjadřuje pravidelné kolísání okolo trendu v rámci jednoho kalendářního roku. Toto kolísání je způsobeno například střídáním ročních období, pracovním cyklem, plánováním atd.

Cyklická složka C_t vyjadřuje kolísání okolo trendu s periodou větší než je jeden rok. Někdy tato složka není rozeznatelná a je proto zahrnuta v trendu.

Nepravidelná složka e_t (zbytková, reziduální) obsahuje nesystematické náhodné výkyvy, chyby měření atd. Nelze ji popsat funkcí času.

Dekompozici časové řady lze pak provést dvěma způsoby:

- aditivní (hodnoty časové řady jsou součty hodnot jednotlivých složek)

$$y_t = T_t + S_t + C_t + e_t$$

- multiplikativní (hodnoty časové řady jsou součiny hodnot jednotlivých složek)

$$y_t = T_t \cdot S_t \cdot C_t \cdot e_t$$

V obou případech je T_t trendová složka, S_t sezónní složka, C_t cyklická složka a e_t nepravidelná složka.

Pro vyjádření časové řady je potom potřeba hledat vhodné modely jednotlivých složek (například trend může být lineární, polynomický, exponenciální atd.). Jaroslav Teda ve svém seriálu o inteligentních ekonomických systémech [9] provádí toto modelování pomocí umělých neuronových sítí, a tento postup také demonstruje ve svém programu TNeuron. Při použití umělých neuronových sítí sice odpadá nutnost podrobné analýzy řad (stačí dodat dostatečné množství vstupních dat) [6], na druhou stranu je ale obtížnější interpretování výsledků [10]. Tato práce se zabývá právě použitím umělých neuronových sítí pro predikci.

3.2 Data warehousing

Cílem této práce je realizace frameworku, který zahrnuje sběr dat, souvisejících s investičním zlatem, analyzování těchto dat a prozkoumání

predikčního potenciálu v těchto datech. Celý tento rámec připomíná jednu z komponent informačních technologií, která se používá ve firmách k analýze ekonomických či strategických dat a která nese název *data warehousing* (datový sklad).

Význam datových skladů je hezky motivován v knize [11]: „*Kdo objeví hodnotu dat skrytých ve svých vlastních databázích, je to jako by objevil zlatý důl pro svou firmu – navíc důl neustále zásobovaný. Zbývá jen umět z něj těžit.*“

Podle [11] dále *data warhousing* znamená „*využití dat pocházejících z různých podnikových aplikací nebo externích zdrojů, jakými jsou veřejné databáze či informace sesbírané z trhu.*“ V této práci jsou data využívána pro podporu investování do zlatých slitků. Obecně ke sběru dat je dále v [11] uvedeno, že je prováděn „*řízeným, periodickým kopírováním dat z různých zdrojů uvnitř organizace i jejího okolí do prostředí optimalizovaného pro analýzy a zpracování informací, překonávajícího platformové, aplikační, organizační i jiné bariéry.*“ To koresponduje s touto prací, neboť bylo potřeba vyřešit automatizovaný sběr dat a data vhodně předzpracovávat.

Dalším úkolem, který na sběr dat navazuje, bylo hledání souvislostí v datech, kterým se obecně zabývá další oblast informačních technologií – data mining.

3.3 Data mining

Data mining (dolování z dat, vytěžování dat, dobývání znalostí z databází, datokopectví) definoval v roce 1996 Fayyad jako „*netriviální extrakci implicitních, dříve neznámých a potenciálně užitečných informací z dat*“. [12]

Stejný autor také stanovil dva primární cíle data miningu:

- predikce (předpovídání podle nalezených vzorů v datech),
- deskripce (rozhodování na základě nalezených vztahů v datech).

K dosažení cílů se používá několika kroků - od selekce, přes předzpracování, transformaci a dolování až k interpretaci a získání požadovaných znalostí. Jak říká ve své přednášce M. Holeňa [13], byl pojem data mining původně hanlivým označením pro počítačem prováděné stanovování a ověřování velkého počtu hypotéz. Časem se ale data mining (DM) díky svým praktickým výsledkům stal užitečným, a dnes existuje mnoho počítačových nástrojů této metody. Tyto nástroje je možné dělit na:

- univerzální pro DM,
- systémy pro podporu okamžitého rozhodování (predikce),
- programy jiného určení, které lze použít i pro DM. [13]

Autor ve své přednášce [13] dále prezentuje výsledky dotazování lidí zainteresovaných v této oblasti na to, jaké používají programy pro data mining. Nezanedbatelný počet respondentů uvádí jako tento nástroj svoje vlastní kódy (*Your own code*). Do této kategorie patří i výsledný software této práce, neboť jde o program, provádějící predikci, vzniklý úpravou implementace umělé neuronové sítě.

Následující podkapitola seznamuje s umělými neuronovými sítěmi a principem, jak je možné pomocí nich predikovat.

3.4 Umělé neuronové sítě

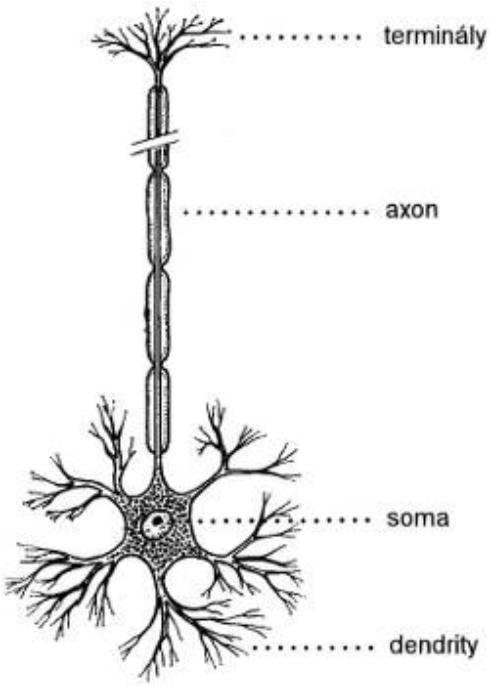
Nejčastější metodou strojového učení v data miningu jsou umělé neuronové sítě (UNS). *„UNS jsou výpočetní modely inspirované biologickými soustavami. Jejich základní jednotka neuron je inspirován biologickým neuronem. Síť se pak skládá z mnoha takových, vzájemně propojených neuronů. UNS se uplatňují v oblastech, jako je predikce, klasifikace, aproximace, dynamické systémy, komprese dat, rozpoznávání obrazu, robotika a mnohé další.“* [14]

Základní principy umělých neuronových sítí vycházejí z jejich biologické předlohy. Nervové buňky jsou v mozku propojeny přes mezineuronové rozhraní, které se nazývá synapse. Různé chemické vlastnosti synapsí způsobují různou synaptickou propustnost. Podle míry této propustnosti, která se označuje jako váha, dochází na synapsi k zesilování (excitaci) nebo zeslabování (inhibici) vzruchu. Klíčovou vlastností neuronu je schopnost při podráždění elektrickým impulsem, který překoná určitou hraniční mez (práh), vygenerovat vlastní elektrický impuls a šířit tak informaci dál v síti. Navíc při každém takovém průchodu signálu se pozměňují synaptické propustnosti (zanechá se paměťová stopa) a tím vlastně dochází k učení.

3.4.1 Formální neuron

Neurony *„jsou základní stavební kameny použité přírodou při stavbě mozku“* a jsou *„specializované na přenos, zpracování a uchovávání informací“*. Existuje celá řada různých druhů neuronů (míšní, pyramidový, oční, Purkyňův) a jejich vnitřní struktura může být při hlubším zkoumání složitá jako samotný mozek [15]. Pro formální matematický model neuronu, který ve čtyřicátých letech dvacátého století popsali Warren McCulloch a Walter Pitts, a tím vlastně dali vzniknout celému oboru umělých neuronových

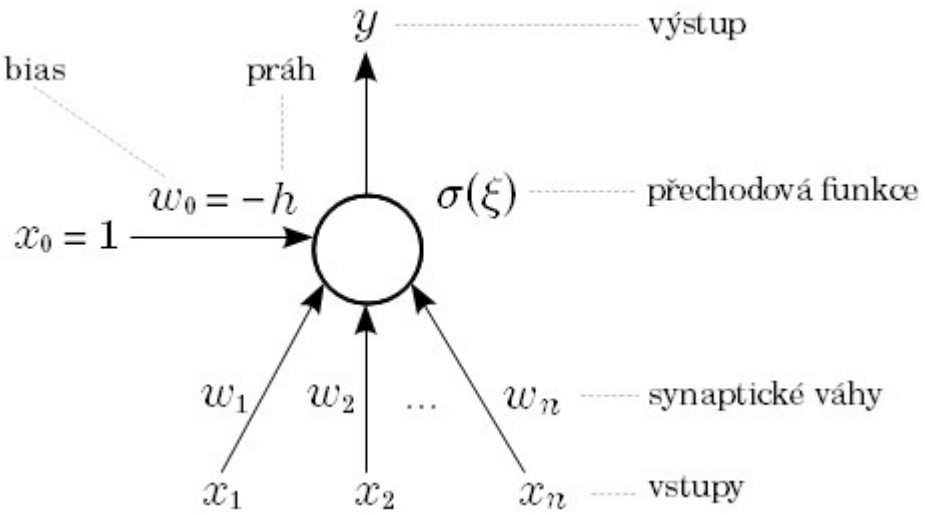
sítí [16], se hodí makroskopický pohled na neuron tak, jak je na ilustraci 3.1. Neuron se skládá ze vstupních kanálů (dendritů), z těla (somatu), výstupního přenosového kanálu (axonu, který může být až 1 m dlouhý [15]) a terminálů axonu, které jsou spojeny s trny dendritů jiných neuronů.



Ilustrace 3.1: Biologický neuron. [16]

Na ilustraci 3.2 je pak znázorněn formální neuron. Má obecně *n* reálných vstupů (x_1,x_2,...x_n), které jsou ohodnoceny *n* reálnými synaptickými vahami (w_1,w_2,...w_n). Váhy reprezentují zesilování nebo zeslabování signálu vstupujícího do neuronu. Pro každý vstup neuronu (n-rozměrný vektor) je vypočítán vnitřní potenciál ξ:

$$\xi = \sum_{i=1}^{n} w_i x_i$$



Ilustrace 3.2: Formální neuron.

Výstupem neuronu je pak hodnota přechodové (aktivační) funkce σ, závislé na potenciálu ξ a prahové hodnotě *h*. Přechodová funkce může být realizována například tak, že její hodnota je 1, pokud je potenciál větší nebo roven prahové hodnotě (neuron je dostatečně „podrážděn“ a sám generuje impuls), a 0, pokud prahu není dosaženo:

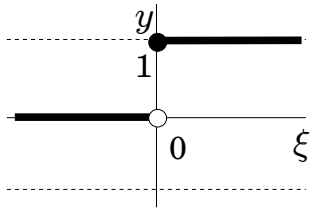
$$y = \sigma(\xi) = \begin{cases} 1 & pokud \quad \xi \geq h \\ 0 & pokud \quad \xi < h \end{cases}$$

V matematickém modelu je ale výhodnější provést následující formální úpravu: zahrnout explicitně vyjádřený práh *h* mezi synaptické váhy – konkrétně přidat k neuronu formální vstup x_0, který je vždy 1 a který je ohodnocen prahem se záporným znaménkem, kterému se říká *bias* a označuje se w_0 (práh je zahrnut v potenciálu, je tedy implicitní). Vnitřní potenciál [16] se potom určí jako:

$$\xi = \sum_{i=0}^{n} w_i x_i$$

Přechodová funkce přejde na tvar:

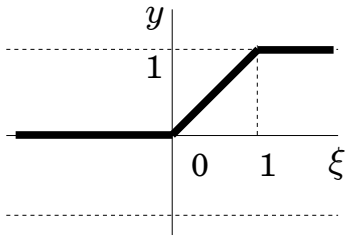
$$y = \sigma(\xi) = \begin{cases} 1 & pokud \quad \xi \geq 0 \\ 0 & pokud \quad \xi < 0 \end{cases}$$



Kromě výše uvedené přechodové funkce, která se nazývá ostrá nelinearita, se používají i jiné aktivační funkce. Jsou to například:

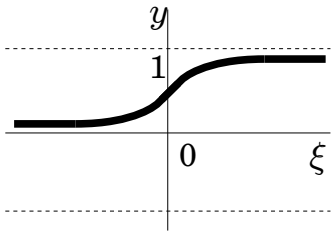
- saturovaná lineární funkce

$$\sigma(\xi) = \begin{cases} 1 & \xi > 1 \\ \xi & 0 \leq \xi \leq 1 \\ 0 & \xi < 0 \end{cases}$$



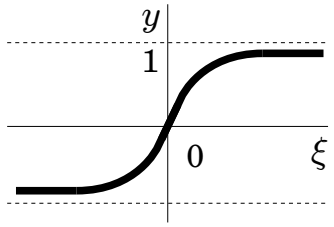
- standardní (logistická) sigmoida

$$\sigma(\xi) = \frac{1}{1 + e^{-\xi}}$$



- hyperbolický tangens

$$\sigma(\xi) = \frac{1 - e^{-\xi}}{1 + e^{-\xi}}$$



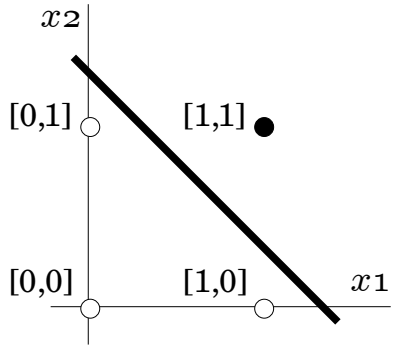
Při práci s neuronovou sítí je potřeba podle zvolených přechodových funkcí také předzpracovat vstupní data. Pokud se v síti využívá standardní logistická sigmoida, měly by vstupy sítě být v intervalu <0,1>. Pokud se používá hyperbolický tangens (který je vlastně bipolárním tvarem standardní sigmoidy), měly by se vstupy sítě nalézat v intervalu <-1,1>.

3.4.2 Lineárně separabilní problém

Pokud se vrátíme k formálnímu neuronu, který má jako aktivační funkci ostrou nelinearitu, můžeme se zamyslet nad tím, jaká je jeho výpočetní schopnost. Neuron s obecně *n* reálnými vstupy a jedním výstupem, který díky aktivační funkci, kterou je ostrá nelinearita, je buď 0 nebo 1, vlastně rozděluje celý prostor R^n na dva poloprostory. V jednom poloprostoru jsou všechny body, které když přivedeme na vstup neuronu, tak neuron jako výstup vypočte 0, a v druhém jsou body, kterým neuron přiřadí 1. Hranicí těchto poloprostorů je obecně nadrovina, jejíž koeficienty jsou synaptické váhy a absolutním členem je práh. Rovnice této hraniční nadroviny je:

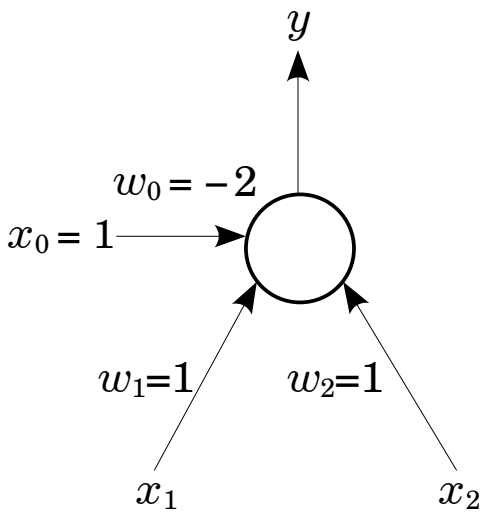
$$w_0 + \sum_{i=1}^{n} w_i x_i = 0$$

Pokud bychom na neuron měli ten požadavek, aby realizoval nějakou logickou funkci dvou proměnných (například logický součin AND), dostaneme speciální případ, kdy neuron musí rozdělit rovinu hraniční přímkou na dvě poloroviny tak, aby body [0,1], [1,0] a [0,0] leželi v polorovině, které neuron přiřazuje 0 a bod [1,1] v polorovině, které je přiřazena 1. Situace, kdy je tento požadavek splněn, může vypadat tak jako na ilustraci 3.3.



Ilustrace 3.3: Lineárně separabilní problém AND.

Příklad neuronu, který by realizoval logickou funkci AND, je na obrázku 3.4. Čísla v obrázku představují hodnoty synaptických vah.



Ilustrace 3.4: Neuron realizující logický součin.

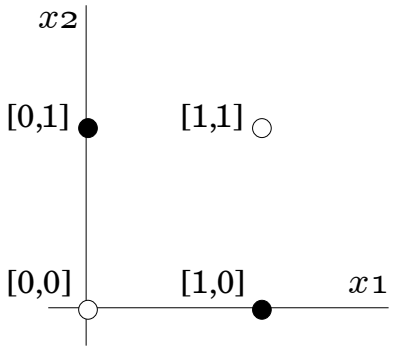
Jenže ne všechny problémy jsou lineárně separovatelné. Obligátním příkladem je logická funkce XOR – ostrá disjunkce (vylučovací „nebo“). Výraz A XOR B je pravdivý, když právě jedna z proměnných A a B je pravda. Pravdivostní tabulka této funkce tedy vypadá takto:

A	B	A XOR B
1	1	0
1	0	1
0	1	1
0	0	0

Tabulka 3.1: Pravdivostní tabulka funkce XOR.

Pokud se podíváme na obrázek 3.5, který přibližuje situaci rovině, vidíme, že není možné najít takovou přímku, která by separovala body, kterým má příslušet výstup 1 a body s výstupem 0. Není tedy ani možné najít takové hodnoty vah, aby neuron realizoval tuto logickou funkci. Hledání vah je

vlastně učení sítě a jeden neuron s ostrou nelinearitou jako aktivační funkcí tedy není schopen se naučit problém, který není lineárně separovatelný.



Ilustrace 3.5: Grafické znázornění logické funkce XOR.

Objevení tohoto problému spolu s přehnanými optimistickými výroky typu „*za několik málo let bude vyvinut umělý mozek*“ [16], posloužilo v šedesátých letech 20. století, kdy ještě nebyl znám učící algoritmus pro vícevrstvé sítě, k diskreditaci výzkumu umělých neuronových sítí. Naštěstí byl objeven algoritmus zpětného šíření chyby (backpropagation), který je nejpoužívanější učící metodou, a který přispěl k oživení zájmu o tento obor umělé inteligence, který přináší nejen konstrukce výpočetních systémů použitelných v praxi, ale také pomáhá pochopit fungování mozku a lidské mysli.

3.4.3 Vlastnosti umělých neuronových sítí

Předchozí podkapitola ukázala, že na složitější problémy potřebujeme více neuronů spojených do sítě. Při zkoumání těchto výpočetních systémů, jak se uvádí v [17], je potřeba se zaměřit zejména na problematiku modelování neuronů a jejich sítí, problematiku učení sítí a na charakteristiky základních druhů neuronových sítí. Podobným způsobem jsou sítě popisovány i v [16] a je u nich tedy možné rozlišovat tyto vlastnosti:

- topologie (architektura) sítě – kolik neuronů je v síti a jak jsou vzájemně propojené,
- konfigurace sítě – jaké jsou hodnoty synaptických vah na spojích neuronů,
- model sítě – jak síť pracuje.

3.4.4 Topologie sítě

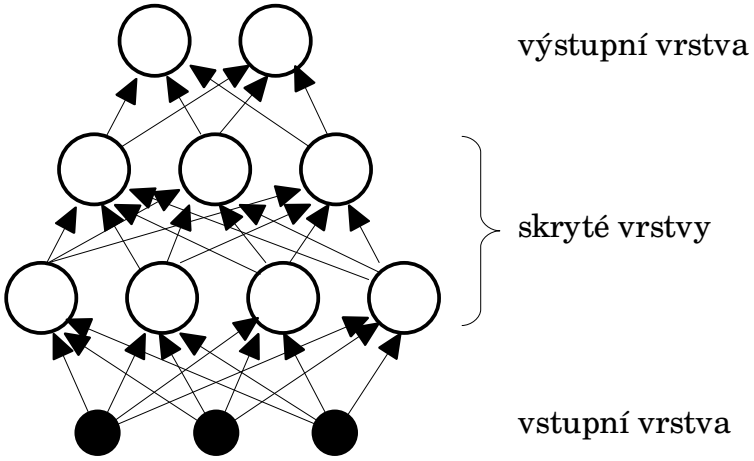
Neuronovou síť si lze představit jako ohodnocený orientovaný graf z teorie grafů. Podle toho, jestli v síti existuje cyklus, se sítě dělí na:

- cyklické (rekurentní) sítě, pokud se v sítí nachází cyklus,
- acyklické (dopředné) sítě, pokud v sítí neexistuje cyklus.

Dopředné sítě mají tu vlastnost, že je možné je „rozplést“ do vrstev a neurony v takové síti pak rozlišujeme na:

- vstupní neurony,
- skryté (pracovní, mezilehlé) neurony,
- výstupní neurony.

Speciální případ je vícevrstvá síť, která má tu vlastnost, že každý neuron je spojen se všemi neurony následující vrstvy. Podle počtu vrstev pak mluvíme o sítích jednovrstvých, dvouvrstvých atd. Do počtu vrstev se ale nezapočítává vstupní vrstva. Jak uvádí [16], architekturu každé dopředné vícevrstvé sítě lze zadat zápisem počtů neuronů v jednotlivých vrstvách, typicky oddělených pomlčkou. Obrázek 3.6 tedy například představuje třívrstvou síť 3-4-3-2 (síť má 3 neurony ve vstupní vrstvě, 4 a 3 v první a druhé skryté vrstvě a 2 ve vrstvě výstupní).



Ilustrace 3.6: Třívrstvá 3-4-3-2 dopředná vícevrstvá síť.

3.4.5 Model sítě

Model sítě se skládá z takzvaných dynamik sítě:

- organizační dynamika sítě určuje, jak se během učení mění topologie sítě (například přidáváním neuronů),
- aktivní dynamika sítě určuje, jak v síti probíhá výpočet pro konkrétní topologii a konfiguraci,
- adaptivní dynamika sítě určuje, jak se síť učí.

Aktivní dynamika sítě určuje aktivační (přenosové) funkce neuronů. Pokud všechny neurony v síti mají stejné aktivační funkce, nazývá se síť homogenní.

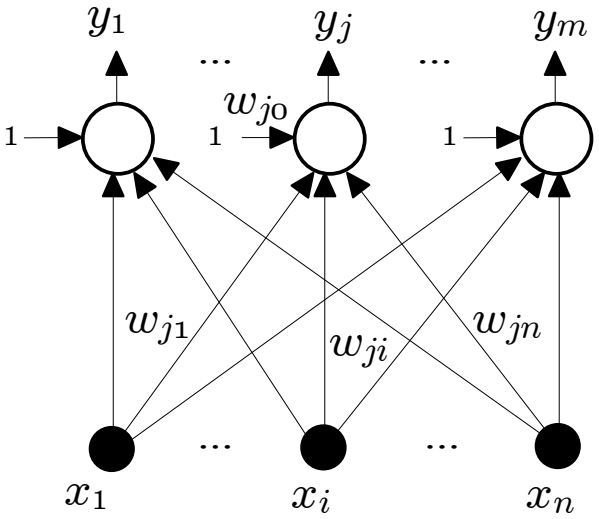
Adaptace (učení sítě) je časově náročný složitý nelineární optimalizační proces [16]. Principem učení je upravit váhy tak (najít takovou konfiguraci), aby síť realizovala co nejpřesněji požadovanou funkci (pro zadaný vstupní vektor vypočítala výstupní vektor). Pro učení se používá takzvaná trénovací množina, což je vlastně posloupnost uspořádaných dvojic [vstupní vektor, požadovaný výstupní vektor]. Rozlišují se dva druhy učení umělých neuronových sítí:

- učení s učitelem – pokud učení probíhá s využitím trénovací množiny,
- učení bez učitele (samoorganizace) – pokud síti nejsou předloženy požadované výstupy, síť například sama hledá „shluky“ a podobnosti.

Pro různé typy sítí existují různé učící algoritmy (učící pravidla).

3.4.6 Síť perceptronů

Jednoduchým model umělé neuronové sítě je síť perceptronů. Byla vynalezena Rosenblattem v roce 1957 a jedná se o jednovrstvou *n*-*m* síť s dopředným šířením (má *n* vstupních a *m* výstupních neuronů). Její architektura je zobrazena na ilustraci 3.7, ve které jsou popsány váhy pouze pro *j*-tý výstupní neuron.



Ilustrace 3.7: Síť perceptronů.

Rosenblatt pro tuto síť navrhl učící algoritmus a dokázal jeho správnost [16]. Jedná se o učení s učitelem a zjednodušeně se dá vyjádřit takto:

1. na začátku se náhodně nastaví váhy
2. opakují se tréninkové cykly

V každém tréninkovém cyklu se pro každý vzor z trénovací množiny upravují všechny váhy podle vztahu:

$$w_{ji}^{(t)} = w_{ji}^{(t-1)} - \varepsilon x_{ki} E_{kj}$$

Tento vztah můžeme číst tak, že novou hodnotu váhy $w_{ji}^{(t)}$ získáme odečtením součinu vstupu vzoru x_{ki}, chyby vzoru E_{kj} a parametru ε od staré hodnoty váhy $w_{ji}^{(t-1)}$.

Parametr $0<\varepsilon \leq 1$ se nazývá rychlost učení a čím je větší, tím síť rychleji zapomíná předchozí naučenou váhu. Rychlost učení se během adaptace může měnit (obvykle zvětšovat).

Učení končí buď dosažením požadované přesnosti (chyby) nebo stanoveného maximálního počtu tréninkových cyklů.

3.4.7 Backpropagation

Backpropagation, neboli algoritmus zpětného šíření chyby, je nejznámější a nejpoužívanější učící algoritmus u vícevrstvých neuronových sítí. Jako mnoho vynálezů prodělal i tento učící algoritmus několik znovuobjevení. Podle [16] byl popsán v jednom z článků sborníku „skupiny PDP“ (Parallel Distributed Processing Group), publikovaném v roce 1986. Podle [18] je autorem z roku 1974 Paul J. Werbos. Ať tak nebo tak, jedná se dodnes o populární metodu s mnoha modifikacemi.

Základním principem toho učícího algoritmu je minimalizace celkové chyby sítě *E*. Tato chyba je závislá na konfiguraci sítě (je funkcí váhového vektoru **w**) a určí se jako součet parciálních chyb E_k (chyb všech *p* vzorů z trénovací množiny):

$$E(\mathbf{w}) = \sum_{k=1}^{p} E_k(\mathbf{w})$$

Jde o gradientní metodu, což znamená, že hledání minima chybové funkce probíhá pomocí parciálních derivací této funkce. Vtipně je tato metoda popsána v [18]: „*Tuto funkci si lze představit jako zakřivenou plochu v hyperprostoru připomínající třídimenzionální pohoří. Okamžitá hodnota vah a prahů je souřadnicí, chyba bodem na této ploše. Z tohoto bodu se snažíme postupně dosáhnout minima. Tuto situaci můžeme připodobnit chůzi po horském masívu za husté mlhy, kdy vidíme jen několik metrů před sebe. Když budeme chtít dojít do údolí (minimální chyba) půjdeme logicky s kopce, tedy proti směru největšího spádu (gradientu). Stejným způsobem tento problém řeší i metoda backpropagation.*“ Úprava vah tedy probíhá podle následujícího vzorce:

$$w_{ji}{}^{(t)} = w_{ji}{}^{(t-1)} + \Delta w_{ji}{}^{(t)}$$

Ten můžeme číst tak, že nová hodnota váhy $w_{ji}^{(t)}$ se získá jako součet staré hodnoty váhy $w_{ji}^{(t-1)}$ a *gradientu* $\Delta w_{ji}^{(t)}$:

$$\Delta w_{ji}{}^{(t)} = -\varepsilon \frac{\partial E}{\partial w_{ji}}(\mathbf{w}^{(\mathbf{t-1})})$$

Gradient je záporná parciální derivace chybové funkce E podle příslušné váhy w_{ji} ve starém váhovém vektoru $\mathbf{w}^{(t-1)}$ vynásobená rychlostí učení ε. Při výpočtu parciální derivace v gradientu zjistíme, že potřebujeme určit jistý člen, který buď:

- lze vypočítat přímo jako chybu vzoru, pokud adaptujeme váhu neuronu, který je ve výstupní vrstvě,
- lze vypočítat pomocí neuronů z následující vyšší vrstvy, pokud adaptujeme váhu, která není vahou neuronu ve výstupní vrstvě (tak se chyba šíří z výstupní vrstvy směrem dolů).

Při učení se tedy nejprve vypočte výstup, z něj chyba vzoru, a tato chyba se začne zpět šířit do nižších vrstev, ve kterých se podle této chyby adaptují váhy. Nutno dodat, že pojmy *dopředná síť* a *zpětné šíření* nejsou v žádném sporu, neboť první popisuje, jak síť vypočítává výstup (aktivní dynamika) a druhý, jak probíhá učení (adaptivní dynamika).

Algoritmus má několik vylepšení:

- rychlost učení se dá měnit během adaptace,
- během učení se upravují tvary přechodových funkcí pomocí *parametru strmosti* (gain),
- ve výpočtu gradientu se zohledňuje předchozí změna váhy násobená *parametrem momentu* α, který obvykle mívá hodnotu 0,9,
- každá váha má svojí speciální rychlost učení.

3.4.8 MADALINE

MADALINE je síť složená z adaptivních lineárních elementů (ADALINE), které jsou podobné perceptronům, ale na rozdíl od nich nemají nelineární přechodové funkce. Jejich výstupem je přímo lineární kombinace vah a vstupů. Tyto sítě mají dnes již pouze historickou hodnotu. Jejich přístup k předpovídání počasí, ve kterém se k vyjádření hodnot využívaly lineárně nezávislé kódy, posloužil jako inspirace pro tuto práci.

3.4.9 GMDH

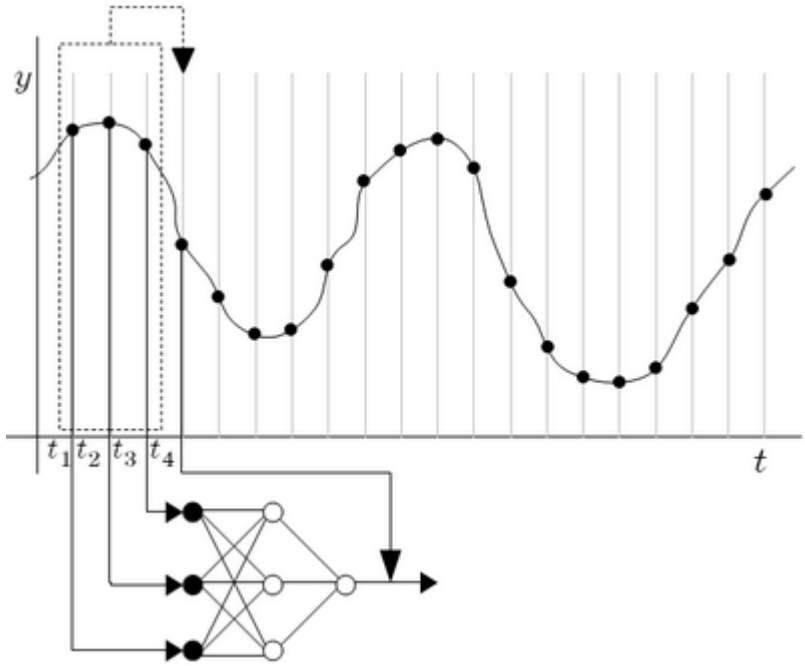
GMDH (Group Method of Data Handling) znamená *skupinová metoda zpracování dat*. Ve výkonových prvcích těchto sítí se podobně jako u ADALINE neuplatňuje žádná nelineární přechodová funkce. Výstupem je kvadratická kombinace dvou vstupních signálů [17]. Tento druh sítě se také často používá pro predikci.

3.4.10 Predikce pomocí umělých neuronových sítí

Umělé neuronové sítě představují jeden z nejpoužívanějších prostředků pro tvorbu automatických nástrojů v oblasti klasifikace nebo predikce [10]. Úlohou predikce je provést analýzu nějaké známé číselné řady (obvykle závislosti nějaké veličiny na čase) tak, abychom nalezli co nejpravděpodobnější průběh pro další časový úsek, který navazuje na uvažovaný známý úsek. Podle [17] je možno k této analýze přistupovat dvěma způsoby:

Prvním je řadu rozdělit do učebních úseků a každý úsek zařadit do určité třídy (například zda v daném úseku došlo k „prudkému stoupání“, „mírnému stoupání“, „mírnému poklesu“, „prudkému poklesu“ nebo „zůstala stejná úroveň“). Síť v tomto případě pak nepredikuje konkrétní hodnotu, ale charakter dalšího průběhu (trend).

Druhým způsobem je určování odhadu konkrétních numerických hodnot. Při tomto způsobu se obvykle vychází z předpokladu, že máme k dispozici časový průběh nějaké veličiny. Z tohoto průběhu navzorkováním vytvoříme ekvidistantní časovou řadu a posouváním časového okna po této řadě vytváříme učební vzory. Hlavními parametry při vytváření vzorů jsou velikost časového okna, počet vzorků v okně a vzdálenost predikované hodnoty. Konkrétní příklad ilustruje obrázek 3.8. Na něm je časové okno vyznačeno šrafovaným obdélníkem. V tomto časové okně jsou tři vzorky. Hodnoty těchto vzorků jsou přivedeny jako vstup sítě a požadovaným výstupem je predikovaná hodnota, vzdálená o jednu jednotku času za posledním vzorkem v okně.



Ilustrace 3.8: Časové okno. [19]

Postupným posouváním časového okna po číselně řadě tak vznikají trénovací vzory, které můžeme vyjádřit takto:

vzor	*vstup*	*výstup*
1	$[y(t_1), y(t_2), y(t_3)]$	$[y(t_4)]$
2	$[y(t_2), y(t_3), y(t_4)]$	$[y(t_5)]$
3	$[y(t_3), y(t_4), y(t_5)]$	$[y(t_6)]$
...	...	...

Pro zpřesnění predikce se někdy také používají informace o průběhu dalších veličin. Řady ze kterých získáváme tyto další vstupní informace se nazývají intervenční řady. Vzory pro učení tedy navíc obsahují další informace o tom, co se v časovém okně dělo s jinými proměnnými.

Na volbě těchto parametrů časového okna a intervenčních proměnných je samozřejmě závislá topologie sítě. Síť musí mít tolik vstupů, kolik je vzorků v okně, a kolik používá intervenčních proměnných. Počet výstupů sítě pak odpovídá počtu předpovídaných hodnot. Volba počtu skrytých vrstev a počtů

neuronů v nich je předmětem experimentování. Neplatí, že čím více, tím lépe. Jedna z heuristik radí použít jednu skrytou vrstvu s dvojnásobným počtem vstupních neuronů. [10]

Dalším důležitým prostředkem jak dosáhnout lepších výsledků predikce a mít možnost zhodnotit její úspěšnost, je rozdělit si dostupná data tak, abychom vytvořili tři časové řady: učící, validační a testovací. Z učící řady sestavujeme trénovací množinu, na validační řadě testujeme, zda je již síť optimálně naučená, a na testovací řadě zkoumáme úspěšnost predikce. Tyto řady se mohou částečně překrývat, ale důležité je, že síť při učení „nevidí“ na celou testovací řadu. [19]

3.4.11 Implementace neuronových sítí

Pro implementaci neuronových sítí, jak uvádí ve své diplomové práci M. Brabec [20], je velmi vhodné objektové programování. Například neuron může mít různé aktivační přechodové funkce, ale má obecně dáno, jak má fungovat. Pro jeho popis je tedy vhodné použít rozhraní (interface). Podobně je tomu pro celou síť, její vrstvy atd.

Cílem této práce ovšem nebylo naprogramovat vlastní neuronovou síť, ale prozkoumat již existující implementace zejména v jazycích PHP, C# a Prolog, a vybrat jednu konkrétní pro realizaci výsledného softwaru. Následující přehled obsahuje dva spíše výukové programy, TNeuron a AIspace Neural Applet, dále projekt FAKE GAME, vyvíjený na ČVUT v Praze, a dále již samotné implementace v PHP a C#. Společným problémem většiny implementací je, že se sice umí naučit jednoduché logické funkce, ale jejich použití na složitější problémy vyžaduje značné úpravy.

TNeuron

Odkaz na projekt: `jaroslav.teda.sweb.cz`

TNeuron je simulátor neuronové sítě s typovými úlohami. Těmi jsou například násobení, predikování kurzu měny nebo spotřeby energie, a dokonce i rozpoznávání obrazů. Program umožňuje vytvořit vlastní síť s prvky typu ADALINE, perceptronů s sigmoidní přenosovou funkcí nebo Kohonenovu síť. Je možné importovat data z textových souborů, a po naučení sítě provádět dotazování. Autorem je Jaroslav Teda.

AIspace Neural Applet

Odkaz na projekt: `www.aispace.org`

AIspace Neural Applet je javovský applet umožňující vizuálně vytvářet neuronovou síť přidáváním a spojováním neuronů na pracovní ploše. Je možné importovat data a automaticky podle nich vytvořit síť. Obsahuje také ukázkové úlohy. Zajímavou funkcí je generátor vytvořené sítě do jazyka Prolog.

FAKE GAME

Odkaz na projekt: `neuron.felk.cvut.cz/game`

FAKE GAME (Fully Automated Knowledge Extraction using Group of Adaptive Models Evolution) je nástroj, používající jednu z verzí sítě GMDH. Je to open source vyvíjený na ČVUT v Praze. Obsahuje různé podpůrné nástroje na předzpracování dat a interpretaci výsledků.

NeuroDotNet

Odkaz na projekt: `neurondotnet.freehostia.com`

NeuroDotNet je projekt v jazyce C#. Obsahuje ukázkové aplikace pro řešení problému XOR, problému obchodního cestujícího nebo rozpoznávání znaků.

AForge.NET

Odkaz na projekt: `www.aforgenet.com`

Jedná se o rozsáhlejší projekt, obsahující nástroje umělé inteligence v C#. Součástí jsou také programy pro aproximaci a predikci funkcí.

C# Neural network library

Odkaz na projekt: `franck.fleurey.free.fr/NeuralNetwork`

Jde o implementaci v C#, která pracuje s umělými neuronovými sítěmi na nižší úrovni – nastavují se samotné neurony a jde tedy spíše o výukovou věc. Zajímavou aplikací v tomto projektu je rozpoznávání obličejů.

NeuralMesh

Odkaz na projekt: `neuralmesh.com`

NeuralMesh je nástroj v PHP na používání umělých neuronových sítí. Instaluje se a vyžaduje databázi. Jde vlastně o webový portál, kam se musí uživatelé registrovat. Uživatel pak může na serveru pracovat s umělými neuronovými sítěmi a svoji práci si ukládat. Na adrese uvedené v odkazu na projekt, je možné si vše vyzkoušet pod demo účtem.

freedelta: Back Propagation Scale

Odkaz na projekt: `freedelta.free.fr`

Tato implementace v PHP je zajímavá tím, že rovnou řeší normalizaci dat. Jde ale spíše o ukázkovou věc. Pro další použití by bylo nutné kód výrazně poupravit.

ANN - Artificial Neural Network for PHP 5.x

Odkaz na projekt: `ann.thwien.de`

ANN (Artificial Neural Network) je projekt realizující neuronovou síť pro PHP ve verzi 5.x. Rozhraní je rovnou připravené pro predikční a klasifikační úlohy. Hezkým příkladem použití tohoto projektu pro predikci je předpovídání množství prodané zmrzliny na základě teploty a vlhkosti vzduchu. Bohužel se ani v ukázkových příkladech nedaří neuronovou síť naučit.

The Tremani Neural Network

Odkaz na projekt: `www.tremani.nl/open-source/neural-network`

Tato implementace byla nakonec vybrána pro realizaci výsledného nástroje. Mimo ukázkový problém XOR se dokázala naučit normalizovaný součin bez nutnosti úpravy původního kódu.

3.5 Postupy předzpracování dat

3.5.1 Doplnění chybějících hodnot

Z různých důvodů mohou v časových řadách chybět hodnoty. Například u kurzu měn ČNB je tímto důvodem prostě to, že ne každý den je nový kurz stanovován. Potřebujeme-li pak mít časové řady úplné, je nutné nějakým způsobem doplnit chybějící hodnoty. Tyto hodnoty samozřejmě nikdy nemohou být plnohodnotné a způsobují menší kvalitu výsledků. Mezi způsoby, jak provést tuto transformaci patří například:

- Nahradit chybějící hodnoty nulami. Tento způsob není příliš vhodný pro predikci.
- Nahradit chybějící hodnoty nějakou střední hodnotou (průměrem, mediánem). K výpočtu se dá použít větší část souboru nebo jen sousední hodnoty.

- Nahradit chybějící hodnoty trendem v celém souboru, získaném regresí.
- Proložit přímkou sousední body. [3]

Posledně zmiňovaný způsob je používán při experimentech. Výpočet této hodnoty vychází z rovnice přímky jdoucí dvěma body $[x_1,y_1]$ a $[x_2,y_2]$:

$$y - y_1 = (x - x_1)\frac{y_2 - y_1}{x_2 - x_1}$$

Pokud má v čase *cas1* veličina hodnotu *hod1*, v čase *cas2* hodnotu *hod2*, tak doplňovanou hodnotu v požadovaném čase *casX* vypočteme (zapsáno v jazyce PHP) takto:

```
$doplnenaHodnota = ($casX-$cas1)*($hod2-$hod1)/
                   ($cas2-$cas1)+$hod1;
```

3.5.2 Normalizace dat

Další transformací, kterou je potřeba s daty provádět, je normalizace. Jde vlastně o vzájemně jednoznačné přiřazení hodnot z jednoho intervalu do druhého. Tato úprava se provádí pro zpřesnění výpočtů a zmenšení výpočetní náročnosti neuronových sítí, které většinou počítají s hodnotami v intervalech <0,1> nebo <-1,1>. Nejpoužívanějšími metodami, jak na tyto intervaly data normalizovat, jsou:

- dekadická normalizace (normalizace posunem desetinné čárky),
- z-score normalizace (normalizace pomocí odchylky od průměru),
- min-max normalizace (lineární transformace),
- soft-max normalizace (nelineární transformace nějakou logistickou funkcí). [21]

Pro experimenty v této práci je používána min-max normalizace. Pro její použití je potřeba určit minimální a maximální hodnotu transformované množiny. Při tom je potřeba dávat pozor, aby se v této množině nevyskytovaly výrazně větší nebo menší hodnoty, než je střední hodnota. To způsobuje u této normalizace využití jen malé části intervalu, na který je prováděn převod.

Pokud jsou $O_{\min}$ a $O_{\max}$ hranice výstupního intervalu, $x_{\min}$ a $x_{\max}$ hranice vstupního intervalu, pak pro normalizovanou hodnotu x' platí:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}(O_{max} - O_{min}) + O_{min}$$

Původní hodnotu x z normalizované hodnoty x' získáme pomocí vztahu:

$$x = \frac{x' - O_{min}}{O_{max} - O_{min}}(x_{max} - x_{min}) + x_{min}$$

Konkrétní zápis funkcí, provádějících tuto normalizaci v jazyce PHP, kde jako výstupní interval je <-1,1>, vypadá takto:

```
define("HI",1);
define("LO",-1);

// Normalizacni vozrce
function norm($x,$xMin,$xMax)
{
   $vysl = ($x-$xMin)/($xMax-$xMin)*(HI-LO)+LO;
   return $vysl;
}
function deNorm($x,$xMin,$xMax)
{
   $vysl = ($x-LO)/(HI-LO)*($xMax-$xMin)+$xMin;
   return $vysl;
}
```

3.5.3 Časový posun

Při zkoumání predikčního potenciálu byl ověřován také předpoklad, že obchody reagují na změnu ceny zlata na trhu a kurzu dolaru s nějakým zpožděním. Další prováděnou transformací s daty je tedy časový posun, který znamená vytvoření časové řady se stejnými hodnotami, ale opožděné nebo předbíhající původní časovou řadu. [3]

3.6 OCR

Při sběru dat bylo potřeba vyřešit čtení cen zapsaných v obrázku. Obecně je strojové čtení (strojové rozpoznávání znaků, Optical Character Recognition) typický problém, který koresponduje s jednou z definic umělé inteligence, kterou v roce 1991 vyslovila Elaine Richová:

„*Umělá inteligence se zabývá tím, jak počítačově řešit úlohy, které dnes zatím zvládají lidé lépe.*“ [22]

Člověk totiž nemá většinou problém přečíst nějaký text psaný ručně, navíc naskenovaný s šumem, rozmazáním atd. Pro počítač to však již může být překážka.

V současné době už ale existují relativně úspěšné metody rozpoznávání písma, ve kterých se uplatňují například umělé neuronové sítě. Užitečné jsou také online OCR nástroje. Při řešení toho dílčího problému při sběru dat nakonec nebylo potřeba používat žádné pokročilé metody nebo externí aplikace, ale postačilo pouze vhodně dekomponovat obrázek. Celý postup je popsán v praktické části práce.

3.7 Prostředky pro parsování textu

3.7.1 Regulární výrazy

Regulární výrazy jsou mocným nástrojem pro zpracování textu. Jsou to jakési „šablony“ pro řetězce znaků, kterými lze popsat hledanou část textu a díky tomu:

- vytáhnout z textu požadovanou informaci,
- přetvářet texty do požadované podoby,
- provádět vyhledávání a nahrazování v textech.

Různé články a seriály o regulárních výrazech na internetu jsou často uváděny hezkým citátem Pavla Satrapy: „*Unix bez regulárních výrazů je jako sex bez partnera/partnerky. Dá se to používat, ale člověk o cosi zásadního přichází.*“ [23] Tento výrok poukazuje na spojitost regulárních výrazů s operačním systémem UNIX. Právě orientace na zpracovávání textů, která přinesla regulární výrazy, je důvodem obliby operačních systémů odvozených od UNIXu. Z toho také vyplývá, že regulární výrazy, které mají základy v teorii formálních jazyků, jsou definovány normou POSIX.

Existuje ovšem více druhů regulárních výrazů. Svoji vlastní syntaxi regulárních výrazů, která se velmi prosadila, přinesl programovací jazyk Perl. Tomuto typu se říká Perl-compatible regulární výrazy.

Následující kód je ukázkou toho, jak vypadá jednoduchý regulární výraz, který je „šablonou“ pro emailové adresy:

```
[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}
```

Tento zápis lze číst tak, že v řetězci, který má být emailovou adresou, musí nejprve být alespoň jeden znak, který je buď malým nebo velkým písmenem, číslicí 0 až 9, tečkou, podtržítkem nebo pomlčkou. Dále musí následovat znak zavináče „@“, pak opět alespoň jeden znak ze stejné skupiny jako před zavináčem. Následovat musí tečka a nakonec 2 až 4 malá nebo velká písmena. Problémem tohoto regulárního výrazu je, že mu vyhovují některé neplatné adresy (např. „--@-.com“) [24]. Regulární výraz, který by pokryl všechny možné platné emailové adresy, by byl mnohem delší a komplikovanější.

3.7.2 Používání regulárních výrazů v PHP

S regulárními výrazy lze pracovat ve většině textových editorů a programovacích jazycích. V jazyce PHP je možné používat oba výše uvedené druhy regulárních výrazů. Funkce pracující s Perl-compatible regulární výrazy

nají předponu „preg“ (např. preg_match_all) a funkce pracující s regulárními výrazy podle normy POSIX mají předponu „ereg“ (např. ereg_replace).

3.8 .NET Web services

Jedním ze zdrojů pro sběr dat o investičním zlatě jsou webové služby. Webová služba (Web service) je jedním z prostředků, kterým se snaží firma Microsoft prosazovat v rámci celé své iniciativy .NET. Jedná se vlastně o aplikaci, která běží na webovém serveru, a která umožňuje klientům používat volatelné funkce API, které se speciálně nazývají *webové metody*. Ovšem nejedná se o technologii, se kterou by přišla přímo společnost Microsoft. Webové služby využívají otevřených standardů jako je HTTP, XML a SOAP (Simple Object Access Protocol), a pro jejich běh není nutný (stejně jako pro běh programů, které webové služby používají) .NET Framework. [25]

Používání webové služby probíhá tak, že klient zformuluje pomocí jazyka SOAP, který je odvozen z XML, dotaz, ten odešle přes HTTP protokol na server, a server vrátí odpověď opět v jazyce SOAP. Velkou výhodou pro vývojáře je, že na většině platforem nemusí řešit parsing HTTP požadavků a s webovými metodami mohou pracovat na stejné úrovni jako s ostatními metodami. Následující kód je ukázka HTTP dotazu, který obsahuje požadavek na informaci o londýnském fixu zformulovaný pomocí SOAP:

```
POST /LondonGoldFix.asmx HTTP/1.1
Host: www.webservicex.net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.webservicex.net/GetLondonGoldAndSilverFix"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetLondonGoldAndSilverFix xmlns="http://www.webservicex.net" />
  </soap:Body>
</soap:Envelope>
```

3.8.1 NuSOAP

Pro práci se standardem SOAP v jazyce PHP existuje projekt NuSOAP. Je to několik PHP tříd, které umožňují formulaci a zpracování dotazů bez nutnosti instalace nějakého rozšíření do PHP. Domovskou stránkou tohoto projektu je `http://www.scottnichol.com/nusoapintro.htm`.

3.9 cron

Jelikož ceny zlatých slitků v obchodech není možné zpětně dohledat, je potřeba provádět pravidelné a automatizované sledování cen. Toto sledování provádí několik PHP skriptů. K pravidelnému spuštění úloh existuje v operačním systému Linux takzvaný cron.

S cronem se pracuje tak, že pomocí příkazu crontab se edituje soubor, ve kterém se specifikuje, kdy a co se má opakovaně spouštět, a samotné spuštění pak provádí softwarový daemon (program, který nepřetržitě běží na pozadí) crond. [26]

Položky záznamu crontab mají následující formát:

```
minute hour day month dayofweek command
```

Minuty jsou určeny číslem v intervalu od 0 do 59, hodiny od 0 do 23, dny v měsíci od 1 do 31, měsíce od 1 do 12 nebo anglickou zkratkou jan, feb atd., dny v týdnu od 0 (neděle) do 6 (sobota) nebo opět zkratkou mon, tue atd., a nakonec jsou uvedeny příkazy. V časových příkazech je možné používat intervaly pomocí pomlčky (například 1-15), je možné specifikovat více údajů oddělených čárkou nebo používat hvězdičku jako zástupce pro všechny možné hodnoty.

Záznam pro cron, který specifikuje, že se každých patnáct minut má spustit skript na kontrolu ceny zlatých slitků, vypadá takto:

```
* 0,15,30,45 * * * /opt/lampp/bin/php check4.php
```

U některých poskytovatelů webhostingu lze cron nastavit vizuálně přes grafickém uživatelské rozhraní.

3.10 HTML 5 a projekt Flot

Nová verze jazyka HTML přinesla několik zajímavých novinek. V HTML 5 je nově snadnější definování layoutu webu díky speciálním tagům pro jednotlivé části stránky. HTML 5 dále zavádí speciální tagy pro přehrávání audio a video souborů a také tag pro kreslení grafiky pomocí JavaScriptu – canvas.

Tento element využívá knihovna Flot [27] k vykreslování 2D grafů, které bylo dříve obtížné a muselo být řešeno jinými prostředky (např. grafickou knihovnou GD v PHP). Flot je implementován v této práci k vizualizaci nasbíraných dat.

4 Řešení

4.1 Architektura

Celý software je řešen jako systém několika PHP skriptů a je rozdělen na tři části:

- modul zajišťující sběr dat (datové pumpy),
- modul provádějící vizualizaci dat,
- modul pro provádění výpočtů umělou neuronovou sítí nad nasbíranými daty.

Z důvodu co nejmenší náročnosti na prostředky, běží vše na freehostingovém účtu serveru Internet Centrum IC.cz společnosti Nodus Technologies spol. s r.o. [28]. Tento freehosting nabízí i automatické spouštění skriptů, které je potřeba pro sběru dat. Na rozdíl od některých i placených hostingů, není potřeba žádné schválení a odůvodnění používání této služby. Stačí jednoduše nakonfigurovat nástroj cron v administrátorské sekci.

Freehosting tedy nabízí v základním bezplatné účtu všechny potřebné prostředky pro realizaci:

- PHP,
- službu FTP pro nahrávání skriptů na server,
- databázi (MySQL i PostgresSQL) s automatickým zálohováním a administračním softwarem (phpMyAdmin),
- cron.

Používání služeb zadarmo přináší samozřejmě i problémy. Není garantována dostupnost a tak se stává, že dochází k výpadkům fungování.

K největšímu výpadku během realizace práce (listopad 2009 až duben 2011) došlo na začátku roku 2011. Od 3. 1. 2011 až téměř do začátku března 2011 nefungovalo z důvodu porušení filesystému na některých diskových polích automatické spouštění skriptů, web běžel pouze ze zálohy a nebylo možné systém aktualizovat.

4.2 Databáze

V řešení je použita databáze MySQL. Pro uchovávání cen zlatých slitků jsou použity tabulky:

Název tabulky	**Použití**
Uchování cen obchodu *ABROS s.r.o. – Investiční zlato*	
gold4you1oz_prodej	Prodejní cena slitku o hmotnosti 1 Oz.
gold4you1oz_vykup	Výkupní cena slitku o hmotnosti 1 Oz.
gold4you100g_prodej	Prodejní cena slitku o hmotnosti 100 g.
gold4you100g_vykup	Výkupní cena slitku o hmotnosti 100 g.
Uchování cen obchodu *Zlaté Mince – Numismatika*	
mince1oz_prodej	Prodejní cena slitku o hmotnosti 1 Oz.
mince1oz_vykup	Výkupní cena slitku o hmotnosti 1 Oz.
mince100g_prodej	Prodejní cena slitku o hmotnosti 100 g.
mince100g_vykup	Výkupní cena slitku o hmotnosti 100 g.

Tabulka 4.1: Přehled tabulek v databázi uchovávajících ceny zlatých slitků.

Všechny tyto tabulky mají následující stejnou strukturu:

Sloupec	**Typ**	**Význam**
cas	timestamp	Čas stanovení nové ceny. Je to automaticky zapsaná hodnota aktuální časové známy.
cena	int(11)	Nová cena produktu.
neplatny	tinyint(1)	Logická hodnota která určuje, zda je cena neplatná nebo platná.

Tabulka 4.2: Struktura tabulek uchovávajících ceny zlatých slitků.

Pro uchovávání aktuální ceny zlata na trhu a aktuálního kurz dolaru se používají dvě tabulky:

Název tabulky	**Použití**
usdphp	Aktuální kurz dolaru (měnového páru)
xauphp	Aktuální cena zlata na trhu

Tabulka 4.3: Přehled tabulek v databázi uchovávajících kurz dolaru a cenu zlata na trhu.

Obě tabulky mají podobnou strukturu:

Sloupec	**Typ**	**Význam**
cas	timestamp	Čas měření. Je to automaticky zapsaná hodnota aktuální časové známky.
hodnota	decimal(6,4)	Hodnota kurzu. Dvouciferné číslo + 4 desetinná místa.
	decimal(7,3)	Hodnota ceny. Čtyřciferné číslo + 3 desetinná místa.

Tabulka 4.4: Struktura tabulek uchovávajících kurz dolaru a cenu zlata na trhu.

4.3 Datové pumpy

Sběr dat probíhá automatizovaně. Každých 10 minut se spouští skripty **check-usd.php** a **check-xau.php**, které ukládají aktuální hodnoty kurzu dolaru a ceny zlata na trhu. Každých 15 minut se spouští skript **check4.php**, který kontroluje ceny zlatých slitků v obchodech.

4.3.1 Ceny zlatých slitků v obchodech

Ukládání cen zlatých slitků probíhá tak, že jsou nejprve získány aktuálně udávané ceny obou produktů v obou obchodech, a pomocí funkce *check* je provedena kontrola, zda se tyto ceny neliší od poslední hodnoty ceny v databázové tabulce příslušného obchodu a produktu. Pokud se tyto hodnoty liší, je do databáze zapsaná nová aktuální cena.

Způsob získání aktuální ceny není ale v obou obchodech stejný. K zisku aktuální ceny v obchodě *ABROS s.r.o. – Investiční zlato* stačí provést textový parsing příslušné internetové stránky pomocí regulárního výrazu. Hodnota ceny je vždy zapsána na stránce za jedinečným tagem. Regulární výraz vypadá takto:

```
!align=\"center\" nowrap=\"nowrap\">[\(]?([\d ]*)!
```

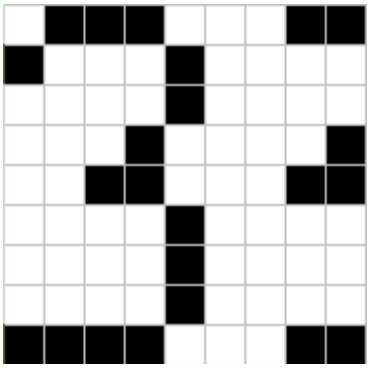
V případě obchodu *Zlaté Mince – Numismatika* je situace složitější. Toto zlatnictví, specializované na prodej investičního zlata na svých internetových stránkách www.zlate-mince.cz, ceny neuvádí ve formě prostého textu, ale každá cena je zapsána ve vygenerovaném obrázku. Zvětšenina jednoho takového obrázku je na ilustraci 4.1. Navíc na celém webu je pomocí javascriptu zakázáno nad všemi obrázky kliknutí pravým tlačítkem myši. Z těchto důvodů není získávání informací o cenách tohoto zlatnictví tak snadné jako v případě druhého obchodu, a bylo potřeba provézt analýzu generovaných obrázků.

Ilustrace 4.1: Zvětšenina obrázku se zapsanou cenou z webu www.zlate-mince.cz.

Přístup, kdy se obchod snaží ztížit extrakci informace o cenách jejich zapsáním do obrázku, připomíná jiný na internetu známý problém – CAPTCHA. Každý uživatel internetu se někdy setkal s tím, že musel před posláním příspěvku do diskuzního fóra, odesláním SMS nebo registrací k nějaké službě, přepsat mnohdy obtížně čitelný text (kontrolní kód) z obrázku do formulářového pole. Tím vlastně musel projít testem, že není počítačový program, ale člověk. Ukázka takového obrázku je na ilustraci 4.2.

Ilustrace 4.2: Ukázka obrázku z projektu reCAPTCHA (www.google.com/recaptcha).

Problém čtení informací z obrázků řeší oblast umělé inteligence OCR (Optical Character Recognition), která se dá česky vyjádřit jako *optické rozpoznávání znaků*. Zahrnuje sofistikované postupy, využívající například umělé neuronové sítě. Jelikož ale obrázky v obchodě nejsou nijak deformovány, podbarvovány nebo jinak grafickými efekty upraveny, jako to bývá obvyklé pro potřeby CAPTCHA, není potřeba používat k přečtení ceny z obrázku žádné speciální postupy. Stačí provézt následující dekompozici obrázku: najít pro každou číslici charakteristické sloupcové vektory kódů barev.



Ilustrace 4.3: Číslice 3 a její charakteristické sloupcové vektory.

Například číslice 3 a její charakteristické sloupcové vektory jsou zobrazeny na ilustraci 4.3. Jelikož ve formátu PNG je kód černé barvy 1 a kód bílé 0, mají tyto vektory (zapsané shora dolů) tvar [1,0,0,0,1,0,0,0,1] a [1,0,0,1,1,0,0,0,1]. V žádné jiné číslici se tyto dva vektory po sobě nevyskytují a čtení tedy můžeme provést postupným procházením všech dvojic po sobě jdoucích vektorů a jejich porovnáváním s předdefinovanými charakteristickými vektory jednotlivých číslic.

Schéma řešení v jazyce PHP vypadá tak, že nejprve jsou definované proměnné obsahující řetězce sloupcových vektorů. Již uváděná číslice 3 je zapsána v proměnné *c3* následujícím způsobem:

```
$c3 = "100010001|100110001";
```

Pomocí funkcí GD knihovny se následně získá odkaz na obrázek a informace o rozměrech obrázku:

```
$obr = imagecreatefrompng($obrAdr);

$obrSize = getimagesize($obrAdr);
```

Podle rozměrů obrázku se v cyklech sestavuje dvojice sloupcových vektorů s kódy barev, získanými pomocí funkce:

```
$sloupec .= imagecolorat($obr,$r,$s);
```

Po sestavení tohoto sloupcového vektoru probíhá jeho porovnávání s proměnnými jednotlivých číslic. Při shodě je do výsledné proměnné připsána nalezená číslice. Celý proces rozpoznání obrázku je zapsán ve funkci s jediným parametrem, kterým je adresa na obrázek s cenou z obchodu.

Od počátku sledování cen v listopadu 2009 fungoval systém spolehlivě až do začátku roku 2011. Díky výstražnému systému mobilních mailů se podařilo odhalit, že obchod provedl na svém webu změnu, která způsobila nefunkčnost čtení ceny z obrázku. Obchod přešel z PHP na technologii ASP a tím se změnil formát generovaného obrázku z PNG na GIF. Zároveň s tím se změnily kódy barev: kód černé barvy se změnil z 1 na 0 a kód bílé barvy z 0 na 251.

Font v obrázku se také změnil a tím i tvary číslic 2, 5 a 7. Bylo potřeba najít a předefinovat charakteristické sloupcové vektory pro tyto číslice.

4.3.2 Kurz dolaru

Kurz dolaru je pro výpočty získáván ze dvou zdrojů. Jedním je devizový kurz ČNB. Česká národní banka na svých webových stránkách poskytuje snadný přístup k datům. Jelikož je možné stáhnout kurzy vybrané měny ve vybraném období přímo v textovém formátu, není potřeba používat regulární výrazy a načtení těchto dat není nijak obtížné. Stačí provést pomocí PHP funkce *explode* rozložení načteného souboru podle konce řádků a oddělovače, kterým je znak „|“. Tyto hodnoty kurzu se do databáze neukládají. V případě potřeby jsou dynamicky načteny.

Druhým zdrojem kurzu dolaru je hodnota aktuálního měnového páru CZK/USD na trhu. Tato hodnota je získávána z webové služby Currency Convertor serveru WebserviceX.NET. Zavoláním metody ConversionRate s parametry FromCurrency („USD“) a ToCurrency („CZK“) je vrácená odpověď ve formátu XML. Původním záměrem bylo načítat tyto hodnoty implementací pro používání webových služeb v PHP NuSoap. Nakonec byl zvolen prostý způsob přečtení ceny regulárním výrazem, ke kterému není potřeba připojovat žádné další skripty. Hodnota ceny je uložena každých 10 minut do databáze.

4.3.3 Cena zlata na trhu

U ceny zlata na trhu je obdobná situace jako u kurzu dolaru. Také zde jsou dva zdroje. Prvním jsou hodnoty londýnského zlatého fixu získávané textovým parsingem webu www.kitco.com. Pro každý rok je potřeba načíst samostatnou stránku.

Druhý zdroj je stejný jako u kurzu dolaru. Jedná se o hodnotu páru USD/XAU (XAU je standardní kód pro zlato) ze stejné webové služby serveru

WebserviceX.NET, a také tato hodnota je ukládána každých 10 minut do databáze.

4.3.4 Výstražný systém

Ke kontrole funkčnosti systému pro ukládání cen z obchodů byl vytvořen jednoduchý kontrolní mechanismus: při každé změně ceny se sestaví řetězec obsahující informace o změnách cen a odešle se jako mail do mobilní schránky. Administrátorovi systému tento mail přijde jako SMS zpráva na mobilní telefon. Jedna konkrétní zpráva vypadá například takto:

```
Od zlato.howto.cz@ic04.ic.cz Ceny GZ:26776>26590
GZv:24499>24329 Gg:85482>84888 Ggv:77192>76655
```

Kvůli zkrácení výsledné zprávy jsou používány zkratky. Každá změna ceny začíná písmenem označujícím obchod (G: *ABROS s.r.o. – Investiční zlato*, Z: *Zlaté Mince – Numismatika*). Pokračuje kódem produktu (Z: slitek o hmotnosti 1 Oz, g: slitek o hmotnosti 100g). Pokud za touto dvojicí znaků následuje písmeno „v“, jedná se o cenu výkupní. Za tímto kódem následuje dvojtečka a za ní již dvojice čísel určujících původní a nově stanovenou cenu.

Díky tomuto kontrolnímu systému se podařilo odhalit několik nedostatků systému. Za prvé se po určité době přestaly aktualizovat ceny jednoho z produktů. V obchodě totiž tento produkt nebyl k dispozici a cena tak byla uvedena v závorkách. S touto závorkou ovšem nepočítal původní regulární výraz, zpracovávající danou stránku.

Dále se podařilo tímto systémem odhalit nekorektní hodnoty. Na začátku roku 2010, pravděpodobně chybou systému, byly všechny ceny v obchodě o jeden řád menší (vypadla poslední cifra). Tyto nově uložené hodnoty byly označeny v databázi jako neplatné a tím nemohly způsobit problémy, například při min-max normalizaci (při této normalizaci by takové hodnoty způsobily, že

většina hodnot by se soustředila okolo jednoho bodu v intervalu, na který se provádí normalizace). Nutno dodat, že pro odběratele těchto SMS zpráv to byla informace velmi pikantní, neboť odběratel mohl hned zareagovat a provést velmi výhodný nákup. Jenže v obchodních podmínkách prodejce je zaneseno, že ceny vzniklé technickou závadou jsou neplatné.

Dále se podařilo odhalit problém s automatickým systémem cron po 3.1.2011, který je popsán v úvodu této kapitoly.

4.4 Realizace výpočtů umělou neuronovou sítí

Každý výpočet se skládá ze dvou kroků zapsaných ve dvou skriptech. Prvním je nastavení parametrů experimentu v souboru se sufixem „_gui.php“, druhým je samotný výpočet v souboru se sufixem „_vypocet.php“.

Grafické rozhraní je tvořeno standardními formulářovými prvky. U některých parametrů je možné zadávat více hodnot najednou. K tomu je používán prvek textarea. Ve skriptech, provádějících výpočet, proběhne parsování hodnot parametrů, pro všechny přípustné variace parametrů jsou vytvořeny trénovací a testovací vzory, proběhne výpočet umělou neuronovou sítí a následně jsou vypsány výsledky.

K výpočtům pomocí umělé neuronové sítě byla vybrána implementace vícevrstvé dopředné sítě s učícím algoritmem backpropagation, jejíž autorem je E. Akerboom [29]. Z nalezených implementací byla vybrána právě tato, neboť byla schopná se naučit jednoduchý součin čísel 1-5 s normalizovanými vstupy.

Pro vypočtení korelačního koeficientu jsou použity statistické funkce v souboru „stat.php“ z oficiální dokumentace PHP [30].

4.4.1 Predikce trendu

Konkrétní vytváření vzorů při predikci trendu je popsáno v kapitole 5.3 . Predikování trendu pak probíhá třemi způsoby, podle toho, do kolika tříd se rozdělí vstupy a výstupy.

Skripty	**Druh experimentu**
bin_bin_bin_gui.php bin_bin_bin_vypocet.php	Vstupy i výstupy jsou klasifikovány bipolárně (1 nebo -1).
tridy_tridy_bin_gui.php tridy_tridy_bin_vypocet.php	Vstupy jsou klasifikovány do zvoleného počtu tříd o zvolených velikostech, výstup je bipolární (1 nebo -1).
tridy_tridy_tridy_gui.php tridy_tridy_tridy_vypocet.php	Vstupy i výstupy jsou klasifikovány do zvoleného počtu tříd o zvolených velikostech.

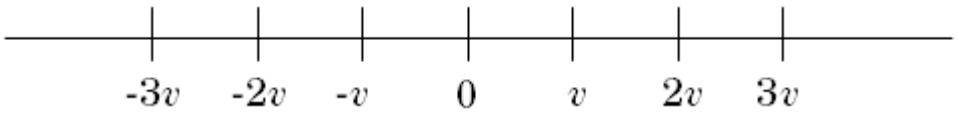
Tabulka 4.5: Tři způsoby predikování trendu.

Bipolární klasifikace probíhá jednoduše: pokud je trend kladný, je oklasifikován 1. Jinak je -1. Do pole *Vzory* je pak přidána dvojice vstupní pole a výstupní pole.

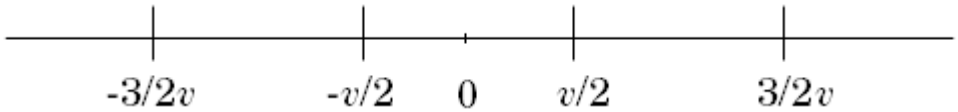
```
$kurzBin = ($kurzTrend>0 ? 1 : -1);
$xauBin = ($xauTrend>0 ? 1 : -1);
$trendBin = ($trend>0 ? 1 : -1);

$Vzory[] = array(array($kurzBin, $xauBin), array($trendBin));
```

Rozdělování do více tříd je složitější. Je závislé na počtu tříd a jejich velikostech. Rozdělování přibližují ilustrace 4.4 a 4.5, kde *v* je velikost třídy.



Ilustrace 4.4: Schéma rozdělení do sudého počtu tříd.



Ilustrace 4.5: Schéma rozdělení do lichého počtu tříd.

Samotné vygenerování vektoru (pole), které vyjadřuje příslušnost do třídy, provádí metoda *getBinVektor* (*HI* a *LO* jsou konstanty krajních hodnot pro vstup neuronové sítě):

```
// Funkce vrací lineární kód podle příslušnosti hodnoty $vstup do
příslušného intervalu.
// Intervaly jsou závislé na velikosti $velikostTridy a jejich počtu
$pocetTrid.
function getBinVektor($vstup, $velikostTridy, $pocetTrid)
{
   $vektor = array();
   for($i=0;$i<$pocetTrid;$i++){ $vektor[]=LO; }

   if($pocetTrid%2==1)
   {
       // lichy pocet trid - okoli nuly
       $hranice = -$velikostTridy/2 - (($pocetTrid-1)/2-
1)*$velikostTridy;
   }
   else
   {
       $hranice=-(($pocetTrid/2)-1)*$velikostTridy;
   }
   for($pokus=0;$pokus<$pocetTrid;$pokus++)
   {
       if($vstup<$hranice)
       {
              $vektor[$pokus]=HI;
              return $vektor;
       }
       $hranice +=$velikostTridy;
   }
   $vektor[$pocetTrid-1]=HI;
   return $vektor;
}
```

Vzory pak vznikají následujícím způsobem:

```
$kurzBin = getBinVektor($kurzTrend,$velikostTridyKurzu,
$pocetTridKurzu);
$xauBin = getBinVektor($xauTrend,$velikostTridyXau,$pocetTridXau);
$trendBin = ($trend>0 ? 1 : -1);

$Vzory[] = array(array_merge($kurzBin, $xauBin), array($trendBin));
```

4.4.2 Predikce hodnoty

Konkrétní vytváření vzorů je popsáno v kapitole 5.4 . Pro predikování se používá dvojice skriptů „predikce_gui.php“ a „predikce_vypocet.php“. Ve skriptu s výpočtem proběhne podle zadaných parametrů navzorkování

časové řady cen v obchodech. Z těchto vzorků jsou metodou *vytvorOkna* vytvořena časová okna, ze kterých jsou síti předány trénovací vzory.

```
// Funkce vytáří ze zadaného pole $zdroj [hodnota, cas] časová okna s
délkou okna $delkaOkna
// a vzdáleností k predikované hodnotě $vzdalPred
// Při vytváření jsou přidány intervenční proměnné:
// - poslední kurz dolaru ČNB před predikovanou hodnotou
// - poslední odpolední londýnský fix před predikovanou hodnotou
function vytvorOkna($zdroj, $delkaOkna, $vzdalPred)
{
   global $MIN_KURZ,$MAX_KURZ,$MIN_FIX,$MAX_FIX;
   $Okna = array();

   $konec = count($zdroj)-$delkaOkna-$vzdalPred;
   for($i=0;$i<=$konec;$i++)
   {
       $Okno = array();

       for($p=$i;$p<$i+$delkaOkna;$p++)
       {
              $Okno["vstup"][] = array($zdroj[$p][1], $zdroj[$p][0]);
       }
       $indexPred = $i+$delkaOkna+$vzdalPred-1;

       $Okno["vystup"] = array($zdroj[$indexPred][1],
$zdroj[$indexPred][0]);

       // Přidání intervenčních proměnných
       $casPred = $zdroj[$indexPred][1];

       $Okno["vstup"][] = array($casPred,
norm(getPosledniKurz($casPred),$MIN_KURZ,$MAX_KURZ));
       $Okno["vstup"][] = array($casPred,
norm(getPosledniFixing($casPred),$MIN_FIX,$MAX_FIX));

       $Okna[] = $Okno;
   }
   return $Okna;
}
```

Jak v predikci trendu, tak hodnoty se pak pracuje s neuronovou sítí následujícím způsobem:

```
// Vytvoření neuronové sítě
$n = new NeuralNetwork($topologie);
$n->setLearningRate(array($rychlostUceni));
$n->setMomentum($moment);
$n->setVerbose($ukecanost);

// Přidání trénovacích vzorů síti
foreach($Vzory as $Vzor)
{
   $n->addTestData($Vzor[0], $Vzor[1]);
}
```

```
// Spuštění učení sítě
$max = 0;
while (!($success = $n->train($pocetKroku, 0.01)) && $max -- > 0)
{
   echo "Nothing found...<hr />";
}
```

Po skončení učení jsou vypočítané predikované hodnoty do budoucnosti a jsou vypsány výsledky výpočtů.

4.5 Realizace vykreslování grafů pomocí knihovny Flot

K vykreslování grafů se používá javascriptová knihovna Flot. Soubor „index.php“ v adresáři „grafy“ obsahuje, jak grafické rozhraní pro zadání požadovaného časového období a obchodu a produktu, tak i prvky volající funkce knihovny Flot. Vykreslení grafů probíhá po zavolání funkce *plot*. Jedním z parametrů této funkce je identifikátor tagu *div*, ve kterém se má graf zobrazit.

```
<div id="fix-kurz" style="width:1000px;height:300px"></div>

$.plot($("#fix-kurz"),
  [
       { data: fixingPM, label: "Fix PM [USD/Oz]" },
       { data: dolarCNB, label: "Kurz dolaru ČNB [Kč/USD]", yaxis: 2}
   ],
  {
       colors: ["#FFFF00","#0000FF"],
       xaxis: {min: FixMINX, max: FixMAXX, mode: "time", tickSize: [1,
"month"], timeformat:"%d.%m.%y"},
       legend: { position: 'nw' },
   });
```

V ukázce je definován tag *div* s identifikátorem „fix-kurz“, ve kterém se zobrazí graf průběhu londýnského fixu a kurzu dolaru ČNB. Funkce *plot* jako první parametr udává právě tento identifikátor. Druhým parametrem jsou specifikovány zdroje dat. Těmi jsou pole s názvy „fixingPM“ a „dolarCNB“. Tato pole jsou vygenerována v PHP skriptech v souborech „fixing.php“ a „kurz_cnb.php“. Třetím parametrem jsou formátovací prvky.

5 Zkoumání vytvořeným softwarem

V následující části práce je popsána povaha jednotlivých dat, na kterých je prováděno zkoumání.

Následně jsou vysloveny a vyhodnoceny předpoklady o predikčním potenciálu v nasbíraných datech. Predikční potenciál je zkoumán ze dvou pohledů: zda je možné predikovat trend na základě předchozího průběhu měnového páru CZK/USD a ceny zlata na trhu, a zda je možné predikovat konkrétní hodnoty pomocí umělé neuronové sítě tak, jak je to popsáno v teoretické části práce.

V obou případech je popsán způsob tvorby vzorů pro neuronovou síť a jsou okomentovány výsledky výpočtů.

5.1 Charakteristika dat

5.1.1 Cena zlatých slitků v obchodech

Údaje o cenách v obchodech jsou pro zkoumání klíčové, neboť právě chování a reakce obchodů je předmětem předpovídání. Byly vytipovány dva internetové obchody a dva produkty (zlaté slitky o hmotnostech 1 Oz a 100 gramů), jejichž ceny byly kontrolovány a při každé změně ceny došlo k uložení nové ceny do databáze. Tento sběr dat byl zahájen na konci listopadu 2009. Týdenní průběh ceny slitku o hmotnosti 1 Oz v obchodě *ABROS s.r.o. – Investiční zlato* zobrazuje tabulka 5.1.

den v týdnu	datum	čas	nová cena [Kč]
pátek	13.8.2010	09:45:14	25667
pátek	13.8.2010	10:30:12	25717
pátek	13.8.2010	13:15:11	25870
pátek	13.8.2010	23:15:13	26006
pondělí	16.8.2010	08:30:20	26047
pondělí	16.8.2010	08:45:21	25945
středa	18.8.2010	10:00:32	25955
středa	18.8.2010	14:15:11	25904
středa	18.8.2010	23:15:32	25961
čtvrtek	19.8.2010	08:15:31	26015
čtvrtek	19.8.2010	10:15:32	26096

Tabulka 5.1: Týdenní průběh stanovování ceny zlatého slitku o hmotnosti 1 Oz v obchodě.

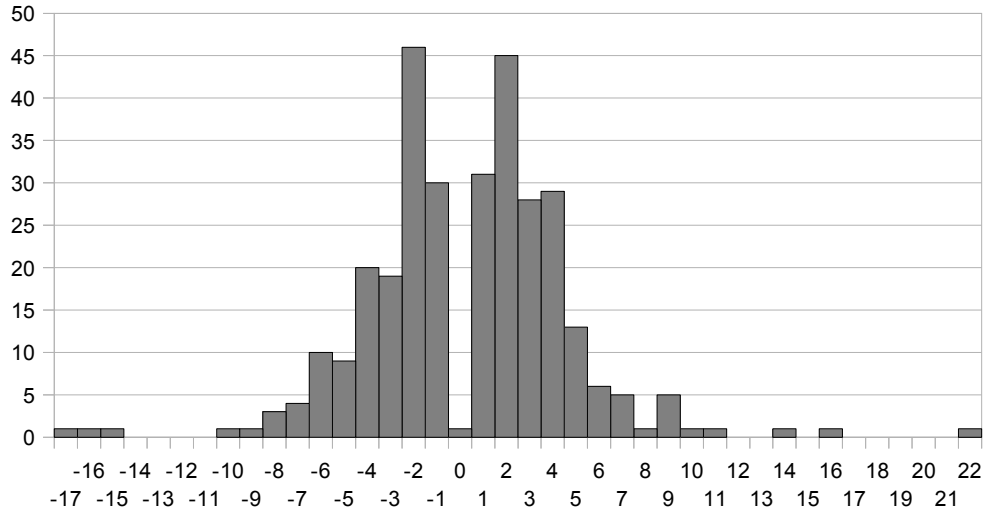
Z tabulky je vidět, že časová řada ceny není ekvidistantní (časové intervaly nejsou stejně dlouhé). O víkendu nebo v úterý 17. 8. obchod cenu nezměnil, ale v některých všedních dnech proběhlo hned několik změn.

Podobným způsobem je neekvidistantní i časová řada cen v obchodě *Zlaté Mince – Numismatika*. Tento obchod nestanovuje nové ceny tak často. Poměr počtu změn obchodu *Zlaté Mince – Numismatika* a obchodu *ABROS s.r.o. – Investiční zlato* je přibližně 2:5. Byla zjištěna zajímavá vlastnost chování obchodu *Zlaté Mince – Numismatika*. Pokud si v tabulce vypíšeme u každé nové ceny, o kolik se vůči předchozí ceně změnila, zjistíme, že tyto změny jsou většinou násobky určitého minimálního kroku. Konkrétně u ceny 100 gramového slitku je tímto krokem 177 Kč.

datum	čas	nová cen [Kč]	změna [Kč]	násobek 177 Kč
12.8.2010	08:00:36	81774	-354	-2
12.8.2010	12:15:23	82482	708	4
12.8.2010	15:15:54	83544	1062	6
13.8.2010	07:45:13	84252	708	4
16.8.2010	08:15:20	84783	531	3
18.8.2010	07:45:34	84606	-177	-1
18.8.2010	16:00:34	84252	-354	-2
19.8.2010	08:00:31	84429	177	1
19.8.2010	15:45:34	84783	354	2
20.8.2010	15:15:13	85314	531	3

Tabulka 5.2: Změny cen slitku o hmotnosti 100 g v obchodě.

Závislost počtu změn - ve zkoumaném období listopad 2009 až listopad 2010 - na násobku minimálního kroku zobrazuje graf 5.1. Graf ukazuje, že tato závislost připomíná normální rozdělení: nejčastěji dochází v obchodě ke změnám ceny o ± 354 Kč, o něco méně ke změnám o ± 177 Kč.



Graf 5.1: Závislost počtu změn na velikosti min. kroku.

5.1.2 Cena zlata

Jak je již popsáno v úvodu, důležitým údajem pro obchody při stanovování cen je London Gold Fixing. Jelikož je jeho hodnota stanovována jen dvakrát denně, ale ceny v obchodech (jak je popsáno v předchozí podkapitole) se někdy mění několikrát za den, je dalším zdrojem „on-line“ kurz USD/XAU z webové služby serveru WebservicesX.net. Tento kurz se mění v krátkých intervalech (pokud je otevřen trh, tak cca každých 15 sekund) a je vzorkován každých 10 minut. Taková řada je tedy ekvidistantní až na případy, kdy je server příliš zatížen a není možné získat odpověď. Tím v časové řadě vznikají prázdná místa.

Londýnský fix se hodí jako intervenční proměnná pro predikci z delších časových oken (několika dnů), „on-line“ kurz se zas hodí pro předpovídání na úrovni jednotlivých změn, kterých, jak už bylo uvedeno výše, je i několik za den a mezi těmito změnami se ani londýnský fix nemusí měnit.

5.1.3 Kurz amerického dolaru

U dat kurzu amerického dolaru vůči české měně je situace obdobná. Jedním zdrojem je kurz ČNB, který je stanovován jednou denně a hodí se tedy díky této periodě jako intervenční proměnná pro predikci na delší časová okna. Druhým je „on-line“ kurz CZK/USD, který je získáván ze stejného serveru jako „on-line“ kurz USD/XAU. Má tedy podobné vlastnosti a hodí se pro předpovídání v intervalech jednotlivých změn.

5.2 Grafická analýza vztahu mezi daty

Inspirací pro vytvoření předpokladů bylo zkoumání grafů dat popsaných výše. Pokud se podíváme na průběhy londýnského fixu, kurzu dolaru ČNB a ceny libovolného ze zkoumaných produktů za rok 2010, vidíme, že obchody reagují očekávaným způsobem.

Svislé čáry v grafech v příloze C - *Vývoj cen během roku 2010* oddělují 6 období roku 2010:

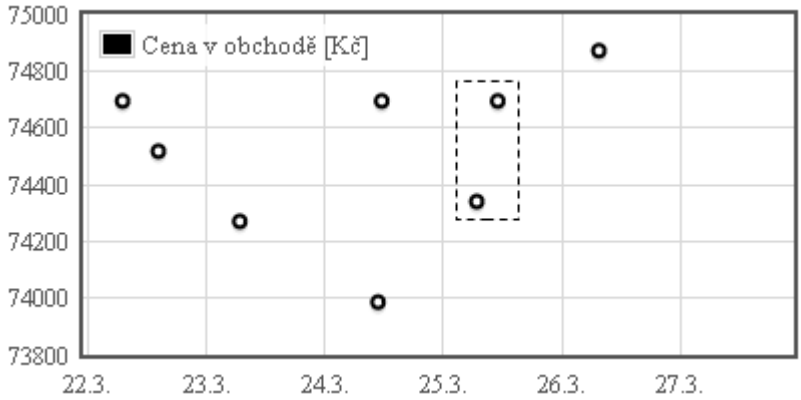
- od začátku roku do poloviny dubna se kurz dolaru a londýnský fix nijak výrazně nemění a cena v obchodě tedy zůstává také bez výraznějších změn,
- v dalším období, zhruba do první čtvrtiny měsíce června, roste jak kurz dolaru, tak i londýnský fix a obchod tím reaguje prudkým růstem ceny,
- v dalším období během června a července je situace opačná – kurz dolaru a londýnský fix jdou dolů a tedy i cena v obchodě klesá,
- v dalším období, které tvoří celý měsíc září, roste jak kurz dolaru, tak londýnský fix, a cena jde opět nahoru,
- v dalším období, tvořeném měsíci září a říjen, londýnský fix roste, ale protože současně jde kurz dolaru dolů, tak se cena v obchodě výrazně nemění,
- v posledním období, od začátku listopadu do konce roku 2010, opět roste jak dolar, tak londýnský fix a cena v obchodě roste.

5.3 Bipolární klasifikace trendu

Tato podkapitola popisuje zkoumání predikce trendu cen v obchodech. Konkrétně je ověřován následující předpoklad:

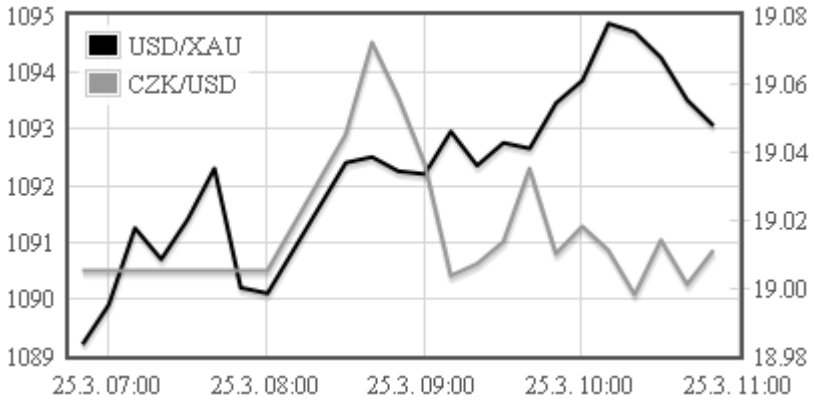
Předpoklad 1: Obchody s určitým zpožděním (řádově desítkách minut, maximálně několika hodin) reagují na změnu kurzu dolaru a ceny zlata na burze změnou ceny produktu. Cenu zvyšují, pokud vzroste kurz dolaru a cena zlata na burze a naopak cenu snižují, pokud kurz dolaru a cena zlata na burze klesá.

Pro ověření tohoto předpokladu je potřeba zkoumat intervaly mezi jednotlivými změnami cen. V grafu 5.2 jsou zobrazeny změny ceny v obchodě *Zlaté Mince - Numismatika* během jednoho týdne. Bod v grafu značí, že v uvedeném čase byla stanovena nová cena produktu.



Graf 5.2: Stanovování nových cen během jednoho týdne.

Jedna konkrétní změna, kdy obchod zvýšil cenu o 354 Kč, proběhla 25.3. a je v grafu 5.2 vyznačena čárkovaným obdélníkem. Jak se v tomto časovém okně měnil kurz dolaru a cena zlata na burze zobrazuje graf 5.3.



Graf 5.3: Vývoj kurzu dolaru a ceny zlata na burze v intervalu jedné konkrétní změny ceny produktu.

Na počátku tohoto intervalu byl kurz dolaru 19,005 Kč/USD, na konci 19,011 Kč/USD. Kurz tedy vzrostl o 0,006 Kč za dolar. Zlato na burze vzrostlo z 1089,2 USD/Oz na 1093,05 USD/Oz, tedy o 3,85 dolaru za troyskou unci.

Z každého takového intervalu ve zvoleném zkoumaném období je vytvořen tréninkový vzor pro neuronovou síť tak, že vstupním vektorem je dvojice [*trend kurzu dolaru*, *trend zlata na burze*], kde trend je roven 1 pokud je změna větší než 0, jinak je trend -1 (trend je tak bipolárně oklasifikován). Výstupem je vektor pouze s jednou složkou, která vyjadřuje změnu ceny v obchodě a opět je 1, pokud cena v obchodě vzrostla, jinak je -1.

Protože chceme také ověřit, že obchody reagují s nějakým zpožděním, je potřeba počítat s tím, že časové okno, ve kterém zkoumáme trend kurzu dolaru a ceny zlata na burze, může být posunuto a tréninkové vzory jsou na tomto posunu závislé. Pokud ale tento posun bude v řádech desítek minut, měla by síť realizovat výpočet tak, jak je uvedeno a vysvětleno v tabulce 5.3.

vstupní vektor	výstupní vektor	vysvětlení
[1,1]	[*hodnota blízká* 1]	Růst kurzu dolaru a ceny zlata na trhu způsobí růst ceny v obchodě.
[-1,-1]	[*hodnota blízká* -1]	Pokles kurzu dolaru a ceny zlata na trhu způsobí pokles ceny v obchodě.
[-1,1]	[?]	Pokud dolar a zlato na burze v časovém okně mají opačný trend, není odhadována výstupní hodnota. V obou případech by ale výsledek měl ležet mezi výstupy předchozích dvou případů.
[1,-1]	[?]	

Tabulka 5.3: Očekávané výstupy sítě.

Pokud by se ve větším počtu výpočtů ukazovalo, že výsledek pro vstup [-1,1] je obvykle větší než výsledek pro vstup [1,-1], dá se z toho usuzovat, že trend zlata má větší váhu na výsledek než trend dolaru a opačně.

Předpoklad byl ověřován výpočtem na období 1.1.2010 – 1.1.2011 sítí s topologií 2-5-1. Počet učících kroků byl 1000. Výsledky výpočtu shrnuje tabulka 5.4. Z ní je vidět, že pokud jsou jako zpoždění reakce obchodu nastaveny hodnoty 0, 15, 30, 45, 60, 75 minut, tak síť odpovídá podle

očekávání, jak je vysvětleno v tabulce 5.3. U každého tohoto zpoždění platí, že odpověď na vstup [-1,1] je větší než na vstup [1,-1]. Větší vliv na výsledek má tedy trend změny ceny zlata na burze.

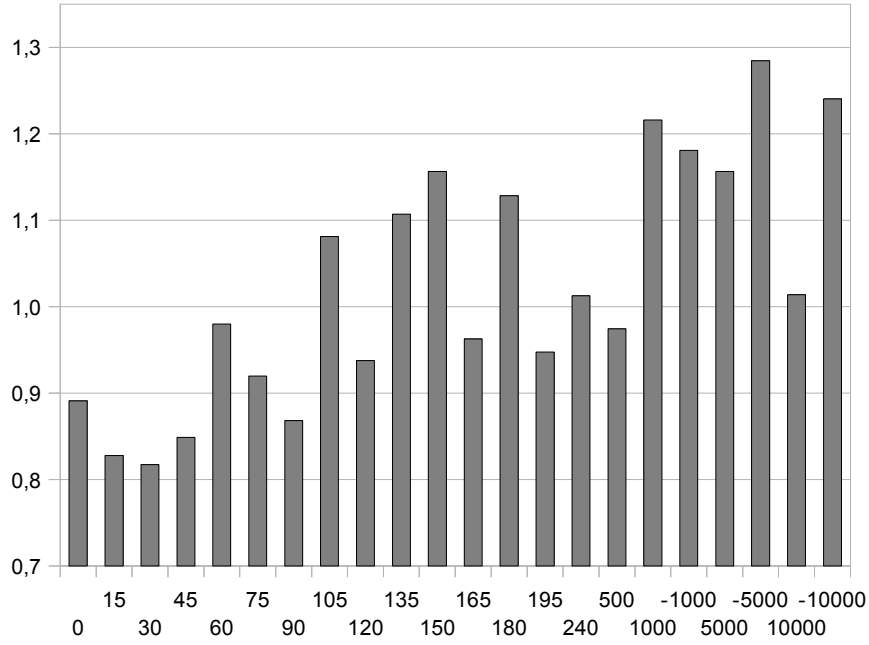
Pokud jsou zpoždění 90, 105, 120, 135, 150, 165, 180, 195 minut, tak již výsledky tak výrazně nekorespondují s očekáváním.

Pokud jsou zpoždění 240, 500, 1000, -1000, 5000, -5000, 10000, -10000 minut, tak výsledky jsou velmi nečitelné a v některých případech i porušují předpoklad, že odpovědi na vstup [-1,1] a [1,-1] mají ležet mezi výsledky výpočtů pro vstupy [1,1], [-1,-1].

Zpoždění (min)	Odpovědi sítě				RMS
	[1,1]	[-1,1]	[1,-1]	[-1,-1]	
0	0,768489	-0,021988	-0,851661	-0,899904	0,891051
15	0,940372	0,414197	-0,659443	-0,778806	0,827736
30	0,933863	0,205652	-0,798990	-0,856549	0,817165
45	0,899871	0,697043	-0,649998	-0,850565	0,848682
60	0,756918	-0,480578	-0,725503	-0,975834	0,979803
75	0,951620	0,773701	-0,038786	-0,205652	0,919576
90	0,873305	0,168369	-0,465265	-0,465265	0,868011
105	0,934638	-0,725956	-0,725956	-0,725956	1,081174
120	0,925268	0,157253	-0,725831	-0,725831	0,937638
135	0,861903	-0,764604	-0,764604	-0,764604	1,106953
150	0,672210	-0,861023	-0,861023	-0,861023	1,156547
165	0,991102	0,653455	-0,616018	-0,616018	0,962676
180	0,827879	0,638931	0,610473	0,610473	1,128436
195	0,995483	0,594860	-0,756655	-0,756655	0,947431
240	0,550523	-0,682055	-0,506340	-0,506340	1,012628
500	0,621820	-0,390721	-0,390721	-0,390721	0,974194
1000	-0,064540	0,876489	0,876489	0,483379	1,216187
-1000	-0,397970	0,671335	0,671335	0,859095	1,180742
5000	0,597067	0,672536	0,672536	0,672536	1,156451
-5000	0,939359	-0,737982	-0,737982	-0,737982	1,284576
10000	0,364043	0,294708	0,294708	0,031449	1,013804
-10000	0,776471	-0,809014	-0,809014	-0,404691	1,240462

Tabulka 5.4: Výsledky výpočtů predikce trendu.

Jaký vliv má zpoždění na schopnost sítě naučit odpovídat podle očekávaného chování obchodů, ukazuje graf 5.4. S rostoucím zpožděním se zvětšuje RMS chyba.



Graf 5.4: Závislost RMS chyby na zpoždění obchodu.

Pro potvrzení předpokladu 1 by bylo potřeba provést významně větší počet výpočtů na různých časových obdobích a s různými topologiemi sítě. Tato podkapitola ukázala, jakým způsobem tyto výpočty provádět, a že pro konkrétní nastavení experimentu je umělá neuronová síť schopná odhalit predikční potenciál pro předpovídání trendu v nasbíraných datech.

5.4 Predikce hodnoty

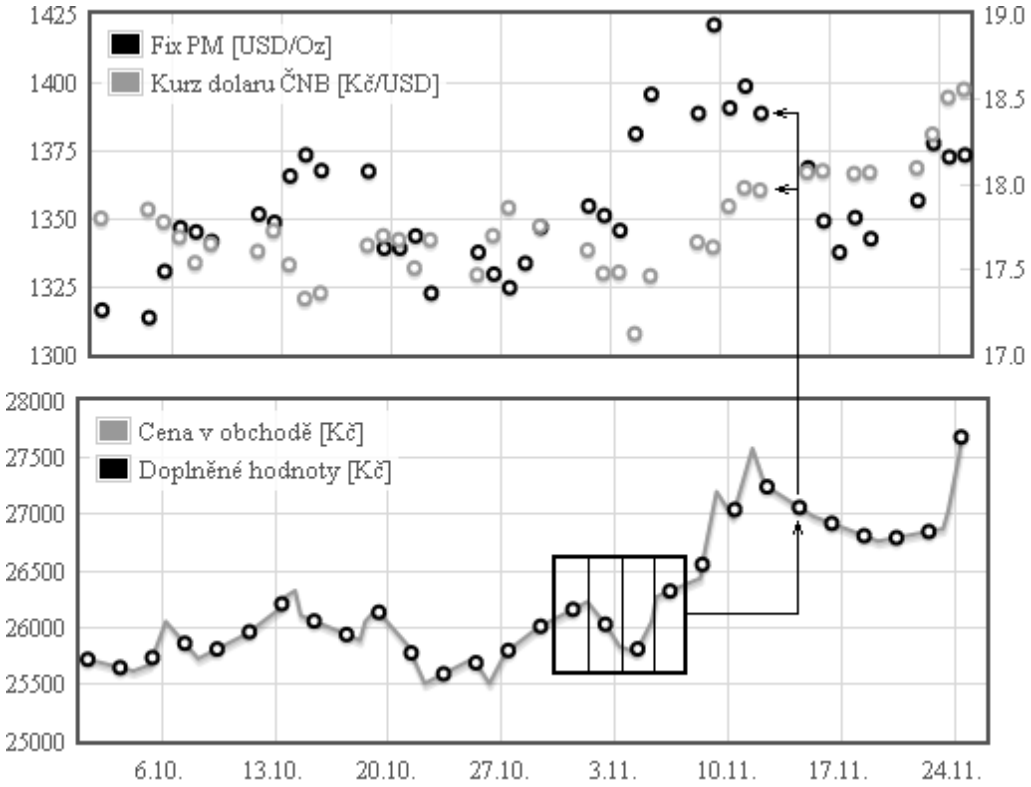
Druhým pohledem na predikční potenciál v datech je zkoumání, s jakou úspěšností je možné predikovat konkrétní hodnoty.

Předpoklad 2: Data mají predikční potenciál pro předpovídání konkrétní hodnoty pomocí UNS.

Jaký je obecný princip predikce hodnoty pomocí umělé neuronové sítě, je popsáno v teoretické části. V experimentech jsou vzory vytvářeny následujícím postupem:

Zvolí se časové období pro trénovací a testovací množinu. Například 1. 1. 2010 až 1. 1. 2011 pro trénovací období, a 1. 1. 2011 až 1. 4. 2011 pro testovací období. Postup vytváření vzorů je pro obě množiny stejný. V prvním dnu časového období se ve zvoleném čase v hodinách a minutách (například 12:00) lineární interpolací určí první hodnota ceny zlatého slitku. Další hodnoty se zjišťují vždy v čase vzdáleném o zadaný počet sekund („časová jednotka“, například 172800 sekund jsou přesně dva dny a další hodnoty v našem případě jsou tedy určeny 3. 1. 2010 ve 12:00, 5. 1. 2010 ve 12:00 atd., dokud tyto časy patří do zvoleného období). Tím je v celém zvoleném časovém období navzorkována ekvidistantní časová řada. Z hodnot této řady se vytváří vzory tak, že se vždy vezme zadaný počet po sobě jdoucích vzorků jako vstupní vektor (např. 6). Výstupem je predikovaná hodnota vzdálená o zadaný počet časových jednotek (např. 2, to v našem příkladě znamená vzdálenost 4 dny) od poslední hodnoty vstupního vektoru a navíc se ke vstupnímu vektoru přiřadí ještě dvě hodnoty, a to poslední kurz dolaru ČNB a poslední odpolední londýnský zlatý fix před časem predikované hodnoty. Všechny hodnoty jsou normalizovány min-max normalizací.

Tvorbu vzorů přibližuje ilustrace 5.1. V ní na dolním grafu je šedou čarou vyznačen průběh ceny zlatého slitku o hmotnosti 1 Oz v období od 1. 10. 2010 do 25. 11. 2010. Body na této křivce zobrazují lineární interpolací doplněné hodnoty vzdálené od sebe 2 dny („časová jednotka“). Horní graf zobrazuje průběh zlatého londýnského fixu a kurzu dolaru ČNB ve stejném období. Obdélníkem na dolním grafu je vyznačeno jedno časové okno, ze kterého je vytvořen vzor se čtyřmi vzorky. Predikovaná hodnota je vzdálena o 4 časové jednotky (8 dní) od poslední hodnoty v okně. Šipkami v horním grafu jsou pak označeny hodnoty posledního kurzu dolaru ČNB a londýnského zlatého fixu před predikovanou hodnotou, které budou přidány do vstupního vektoru vzoru.

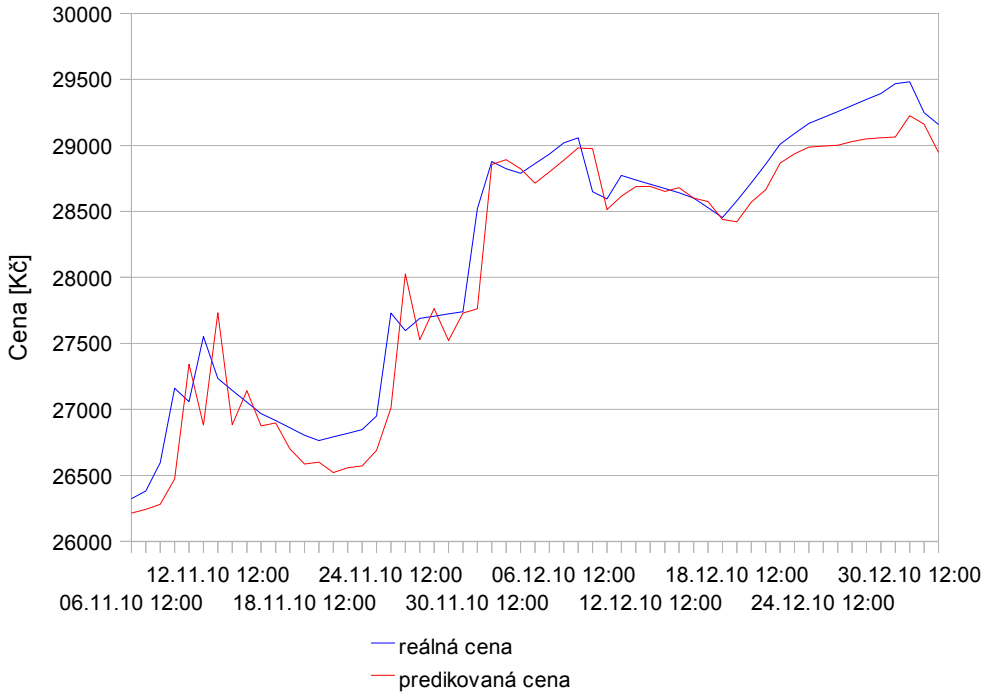


Ilustrace 5.1: Tvorba vzoru pro predikci hodnoty.

Pro zkoumání umožňuje software zvolit obchod a produkt, na kterém se bude predikce provádět, časový interval, ze kterého se vytvoří trénovací vzory, časový interval, na kterém se otestuje výpočet, délku časového okna a vzdálenost predikované hodnoty. Aby bylo porovnávání výsledků objektivní, je možné také omezit maximální počet trénovacích a testovacích vzorů. Dále je možné volit topologie a další parametry sítě. Po skončení učení sítě je skriptem vrácena RMS chyba a průměrná absolutní chyba v korunách na trénovací a testovací množině, a koeficienty korelace mezi řadami skutečných a predikovaných hodnot v trénovacím a testovacím období. Všechny skutečné i predikované hodnoty jsou vytisknuty do textových polí ve formátu CSV tak, aby bylo snadné je přenést do tabulkového procesoru a vizualizovat si výsledky predikce. Také je sestaven dotaz pro predikci do budoucnosti. Hodnota predikované ceny (odpověď na tento dotaz) spolu s časem, na kdy se predikuje, je také vytištěna.

5.4.1 Systematický posun

Během experimentů byl zjištěn nápadný vzájemný posun predikovaných a reálných hodnot. Tento problém lze vidět na grafu 5.5. Tento graf zobrazuje reálné a predikované hodnoty na testovací množině konkrétního výpočtu z kapitoly 8.4.1 - *Výpočet demonstrující systematický posun* z přílohy, kde předpovídané hodnoty jakoby předbíhají hodnoty skutečné. Protože se nepodařilo nalézt implementační chybu a posun nebyl zřejmý u všech výpočtů, je prováděna po skončení učení korekce výsledků. Řady reálných a predikovaných hodnot jsou vůči sobě posunuty, a jsou přepočítány hodnoty průměrných absolutních chyb a korelačních koeficientů. Tyto hodnoty jsou vypsány s poznámkou „posun“. Posunuté řady jsou také vytištěny s nadpisem „Po korekci posunu“.

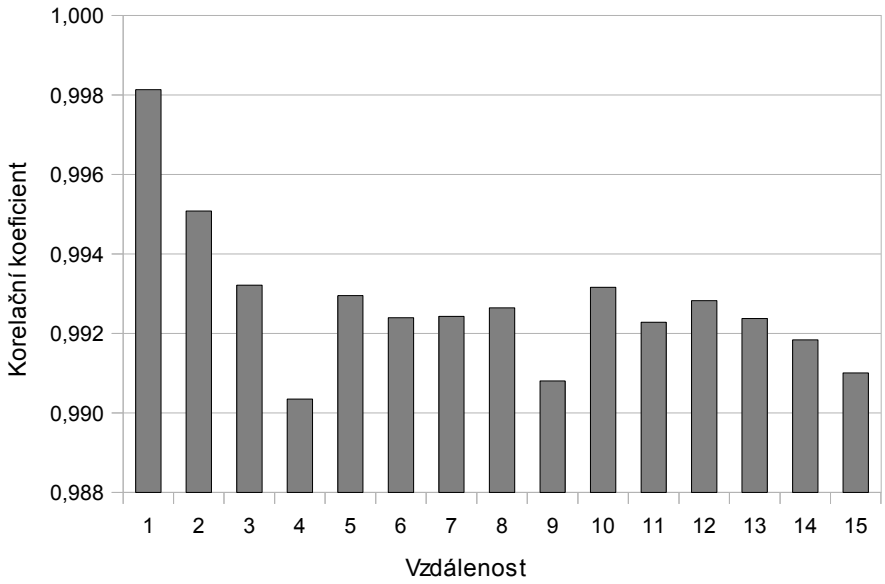


Graf 5.5: Graf s nápadným vzájemným posunem řad reálných a predikovaných cen.

Graf 5.5 zobrazuje výsledek na testovací množině. Před posunem je průměrná absolutní chyba 206,94 Kč a po posunu 134,45 Kč. Z výsledků výpočtu z přílohy je vidět, že korekce posunu zlepšuje i hodnoty korelačních koeficientů. V dalších výpočtech se ukazuje, že chyby u posunutých řad jsou obvykle menší.

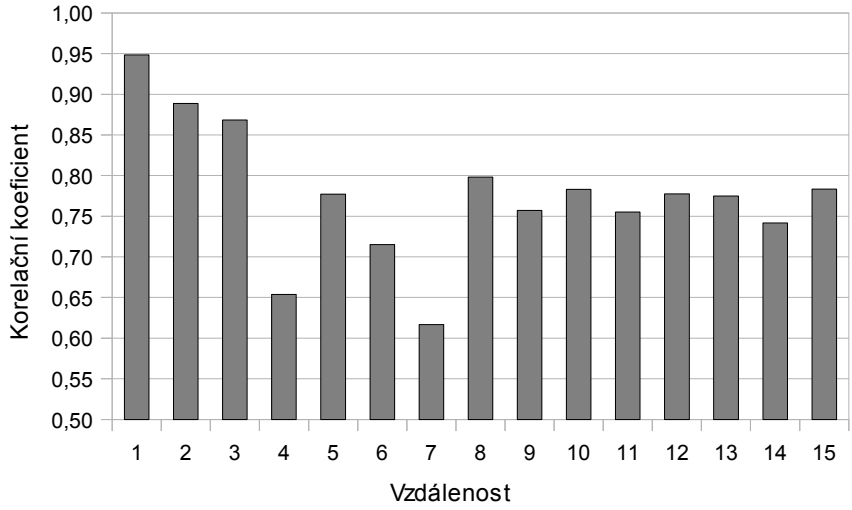
5.4.2 Vliv vzdálenosti předpovídaných hodnot

Předmětem dalšího zkoumání bylo zjišťování vlivu vzdálenosti predikované hodnoty na přesnost predikce. Lze předpokládat, že čím vzdálenější bude predikovaná hodnota, tím horší budou výsledky predikce. Ověřování předpokladu probíhalo pro vzdálenosti od 1 až do 15 časových jednotek. Podrobné nastavení experimentu a výsledky výpočtů jsou uvedeny v příloze kapitoly 8.4.2 - *Výpočet vlivu vzdálenosti na predikci*.



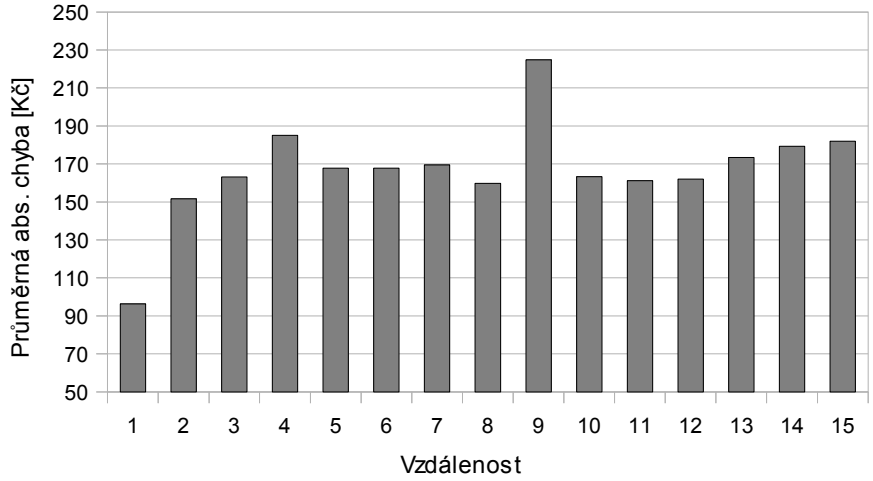
Graf 5.6: Závislost korelačního koeficientu na vzdálenosti předpovídané hodnoty na trénovací množině.

Pokud zobrazíme závislost korelačního koeficientu (po korekci posunu) na vzdálenosti na trénovací i testovací množině, vidíme, že hodnoty korelačního koeficientu s rostoucí vzdáleností mírně klesají. To odpovídá předpokladu.

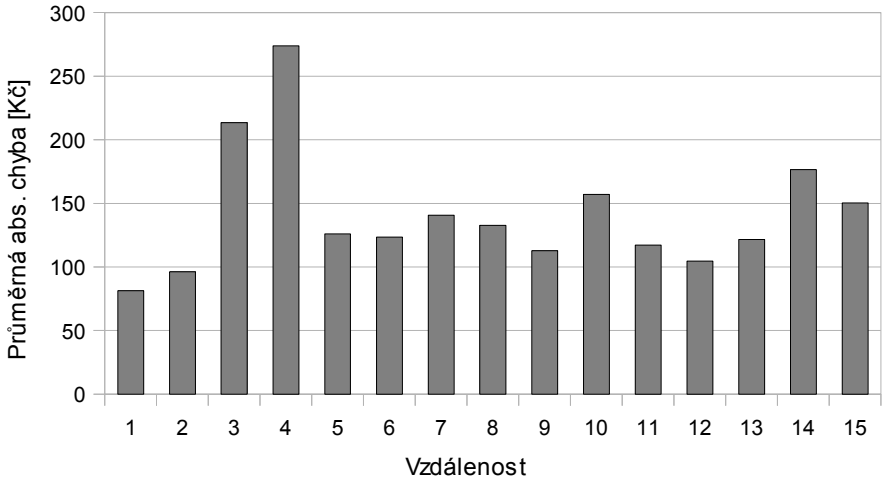


Graf 5.7: Závislost korelačního koeficientu na vzdálenosti předpovídané hodnoty na testovací množině.

Průměrné absolutní chyby (po korekci posunu) s předpokladem příliš nekorespondují. První hodnota na trénovací množině je sice výrazně nižší (nejbližší hodnota je predikovaná nejlépe), ale s rostoucí vzdáleností chyba neroste. Podobně je tomu na testovací množině. Zde ale dokonce ani první hodnota není nijak výrazně nižší.



Graf 5.8: Závislost průměrné absolutní chyby na vzdálenosti predikované hodnoty na trénovací množině.

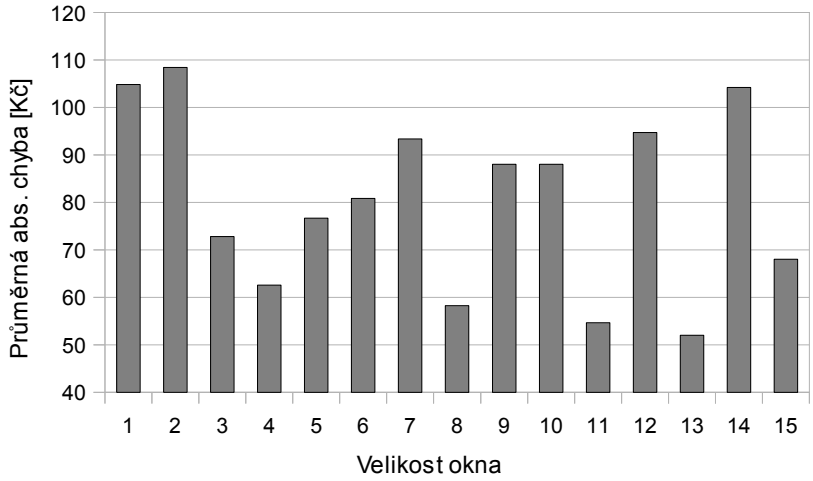


Graf 5.9: Závislost průměrné absolutní chyby na vzdálenosti predikované hodnoty na testovací množině.

Tyto výsledky mohou ukazovat na nevhodný způsob použití intervenčních proměnných. Jestliže jsou ke vstupům přidány hodnoty poslední ceny zlata na trhu a kurzu dolaru před predikovanou hodnotou, tak je možné, že tyto proměnné mají dominantní vliv na výsledek. Na vzdálenosti od poslední hodnoty v časovém okně pak příliš nezáleží.

5.4.3 Vliv velikosti časového okna

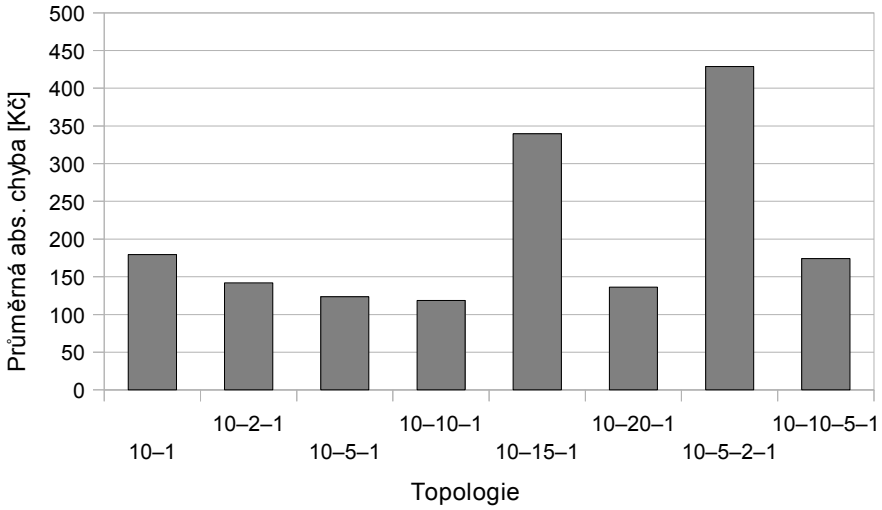
Dalším zkoumaným problémem byl vliv velikosti časového okna na přesnost predikce. Pro toto zkoumání nebyl vysloven žádný předpoklad. Není známa heuristika, která by radila, jak velké má být časové okno při predikování neuronovými sítěmi. Přesnost predikce byla měřena pro velikosti oken od 1 do 15. Nastavení experimentu a výsledky výpočtů jsou zaznamenány v kapitole 8.4.3 - *Výpočet vlivu velikosti okna na predikci*. Z výsledků těchto výpočtů nelze vyčíst žádný prokazatelný závěr. Pouze průměrná absolutní chyba (po korekci posunu) na testovací množině ukazuje, že nižších chyb je dosahováno při velikostech oken 4, 8, 11 a 13. Použití pouze jedné nebo dvou předchozích hodnot způsobuje velké chyby ve výsledku.



Graf 5.10: Závislost průměrné absolutní chyby na velikosti časového okna na testovací množině.

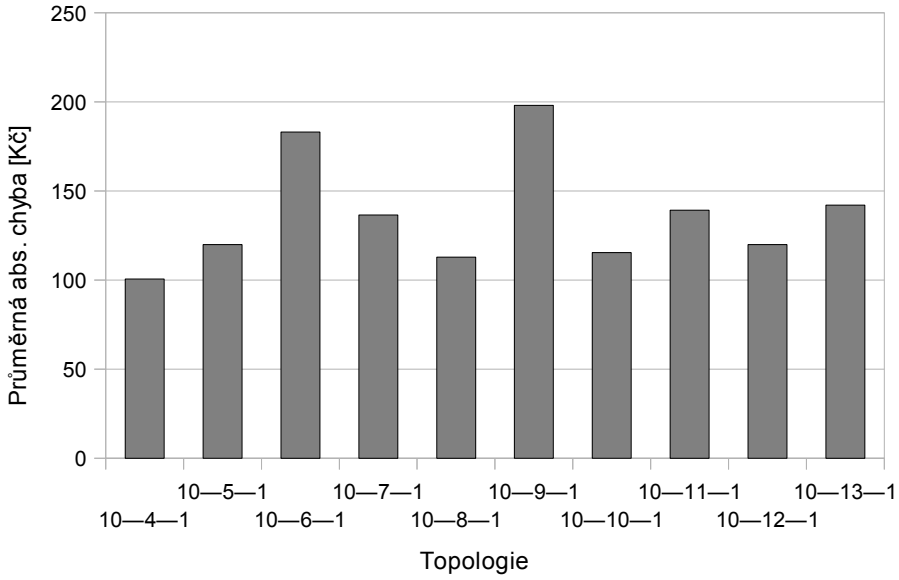
5.4.4 Hledání optimální topologie

Jedním z největších problémů při práci umělými neuronovými sítěmi je volba parametrů sítě. Jedním z nejdůležitějších je topologie. Další experiment, jehož podrobné nastavení a výsledky jdou v kapitole 8.4.4 - *Výpočet hledání optimální topologie* v příloze, se snaží pro konkrétní období hledat nejlepší topologii. Výsledky prvního pokusu přibližuje graf 5.11.



Graf 5.11: Závislost průměrné abs. chyby na topologii.

Z něj je vidět, že nejmenší chyba sítě je pro topologii 10-10-1. Proto v druhém pokusu proběhly výpočty pro topologie blízké této topologii.



Graf 5.12: Závislost průměrné abs. chyby na topologii na testovací množině.

Z výsledků druhého pokusu není průkazné, že by některá z topologií byla výrazně lepší. Nejlepší jsou topologie 10-4-1, 10-8-1, 10-10-1 a 10-12-1.

Z výsledků v příloze lze ještě odhalit, že použití bohatší topologie 10-5-5-5-1 přináší špatné výsledky výpočtů.

6 Závěr

V práci se podařilo realizovat, a díky výstražnému mechanismu také vyladit, systém sbírající data, související s investičním zlatem. Sběr dat byl zahájen v listopadu 2009. Dále vznikla aplikace vizualizující nasbíraná data a aplikace provádějící předpovídání cen zlatých slitků v obchodech pomocí umělých neuronových sítí.

Experimenty byl zkoumán predikční potenciál v nasbíraných datech. Z jejich výsledků vyplývá, že je možné naučit síť predikovat trend ceny. Při predikování konkrétní hodnoty ceny sice křivky reálných a predikovaných cen mívají podobné tvary (což je potvrzeno vysokými hodnotami korelačních koeficientů), ale průměrné absolutní chyby přepočtené na ceny v korunách, jsou relativně vysoké. Důvodem může být velký počet transformací s daty před vytvořením trénovacích vzorů. Praktická využitelnost pro rozhodování není tedy zatím příliš velká. Ukázala se také relativní nezávislost přesnosti predikce na vzdálenosti předpovídané hodnoty, což může znamenat nevhodné použití intervenčních proměnných.

Obecnou známkou umělých neuronových sítí je neexistence nějakého osvědčeného postupu pro získání kvalitních výsledků. Je tedy možné, že větší přesnosti predikce by se dalo dosáhnout, kdyby se do vzorů zahrnuly další faktory, mající vliv na cenu zlata, a kdyby bylo k dispozici větší množství dat.

Jiný přístup k vytváření trénovacích vzorů spolu s použitím dalších predikčních metod může být předmětem pokračování této práce.

7 Seznam použité literatury

[1] POTŮČEK, Jan. *ZlatýPortál.cz* [online]. [cit. 2011-04-13]. Dostupné z WWW: <http://www.zlatyportal.cz>.

[2] BERNSTEIN, Peter L. *Dějiny zlata*. Praha : Grada Publishing, 2003. 384 s.

[3] HANČLOVÁ, Jana; TVRDÝ, Lubor. *Úvod do analýzy časových řad*. 2003. 34 s. Dostupné z WWW: <http://gis.vsb.cz/pan-old/Skoleni_Texty/TextySkoleni/_vti_cnf/AnalyzaCasRad.pdf>.

[4] ARLT, Josef; ARLTOVÁ, Markéta. *Finanční časové řady : vlastnosti, metody modelování, příklady a aplikace*. Praha : Grada Publishing, 2003. 220 s.

[5] ARLT, Josef; ARLTOVÁ, Markéta; RUBLÍKOVÁ, Eva. *Analýza ekonomických časových řad s příklady*. Praha : Vysoká škola ekonomická, Fakulta informatiky a statistiky, 2002. 147 s.

[6] BOUŠKA, Josef. *Neuronové sítě pro predikci časových řad*. Praha, 2008. 84 s. Diplomová práce. ČVUT v Praze, Fakulta elektrotechnická, Katedra počítačů.

[7] FOKA, Amalia. *Time Series Prediction Using Evolving Polynomial Neural Networks*. 1999. 120 s. Dizertační práce. University of Manchester Institute of Science and Technology.

[8] ZVÁROVÁ, Jana. *Základy statistiky pro biomedicínské obory* [online]. 1999 [cit. 2011-04-13]. Dostupné z WWW: <http://new.euromise.org/czech/tajne/ucebnice/html/html/statist.html>.

[9] TEDA, Jaroslav. *Programujte.com : IT portál o programování, grafice a webdesignu* [online]. 2006 [cit. 2011-04-13]. Inteligentní ekonomické systémy. Dostupné z WWW: <http://programujte.com/?akce=clanek&cl=2006010101-inteligentni-ekonomicke-systemy-iii->.

[10] BERKA, Petr. *Neuronové sítě* [online]. [cit. 2011-04-13]. Dostupné z WWW: <http://sorry.vse.cz/~berka/docs/izi456/kap_5.4.pdf>.

[11] HUMPHRIES, Mark; HAWKIND, Michael W.; DY, Michelle C. *Data warehousing : návrh a implementace*. Praha : Computer Press, 2002. 257 s.

[12] FAYYAD, Usama M., et al. *Advances in Knowledge Discovery and Data Mining*. Mit Press, 1996. 625 s.

[13] HOLEŇA, Martin. *Algoritmy data miningu*. (přednáška) Praha : FIT ČVUT. [cit. 2011-04-13]. Dostupné z WWW: <http://www.avc-cvut.cz/avc.php?id=9665>.

[14] HORKÝ, Ladislav; BŘINDA, Karel. *Neuronové sítě*. Praha : ČVUT v Praze, 2009 [cit. 2010-06-25]. Dostupné z WWW: <http://fyzsem.fjfi.cvut.cz/2008-2009/Leto09/proc/neurony.pdf>.

[15] NOVÁK, Mirko; FABER, Josef; KUFUDAKI, Olga. *Neuronové sítě a informační systémy živých organismů*. Praha : Grada, 1993. 272 s.

[16] ŠÍMA, Jiří; NERUDA, Roman. *Teoretické otázky neuronových sítí*. Praha : MATFYZPRESS, 1996. 390 s.

[17] NOVÁK, Mirko. *Neuronové sítě a neuropočítače*. Praha : Senzo, 1992. 192 s.

[18] KORDÍK, Pavel. *Vytěžování dat*. (přednáška) Praha : FEL ČVUT. [cit. 2011-04-13]. Dostupné z WWW: <http://cw.felk.cvut.cz/lib/exe/fetch.php/courses/y336vd/prednasky/p10-doprednens.pdf>.

[19] OBITKO, Marek. *Předpovídání pomocí neuronových sítí* [online]. 1999 [cit. 2011-04-13]. Dostupné z WWW: <http://www.obitko.com/tutorials/predpovidani-neuronovou-siti/>.

[20] BRABEC, Michal. *Návrh struktury a trénovacího algoritmu perceptronu pro účely predikce hodnot binárních časových řad*. České Budějovice, 1996. 44 s. Diplomová práce. Jihočeská univerzita, Pedagogická fakulta, Katedra informatiky.

[21] MEŠKO, Dezider. *Normalizace dat pro neuronovou síť GAME*. Praha, 2008. 33 s. Bakalářská práce. ČVUT v Praze, Fakulta elektrotechnická, Katedra počítačů.

[22] MAŘÍK, Vladimír; ŠTĚPÁNKOVÁ, Olga; LAŽANSKÝ, Jiří. *Umělá inteligence (1)*. Praha : Academia, 1993. 264 s.

[23] SATRAPA, Pavel. *Regulární výrazy* [online]. c2000 [cit. 2011-04-13]. Dostupné z WWW: <http://kobrd.zbytky.net/C%20jazyk/regularni_vyr.satrapa/>.

[24] PECKA, Miroslav. *Regulární výrazy - Regexp | tutoriály, testery | PHP, Perl, Javascript, .NET* [online]. c2008 [cit. 2011-04-13]. Regulární výraz pro e-mail. Dostupné z WWW: <http://www.regularnivyrazy.info/email.html>.

[25] PROSISE, Jeff. *Programování v Microsoft.NET*. Brno : Computer Press, 2003. 712 s.

[26] WELSH, Matt, et al. *Používáme Linux : podrobný průvodce Linuxem*. Praha : Computer Press, 2003. 659 s.

[27] *Google Code* [online]. c2011 [cit. 2011-04-13]. Flot - Attractive Javascript plotting for jQuery. Dostupné z WWW: <http://code.google.com/p/flot/>.

[28] *IC.cz - webhosting zdarma, freehosting, PHP4, PHP5, MySql, PostgreSql, stránky zdarma, blog, fotogalerie* [online]. c2002 [cit. 2011-04-13]. Dostupné z WWW: <http://www.ic.cz/>.

[29] AKERBOOM, *E. Tremani | An advanced neural network in PHP* [online]. c2011 [cit. 2011-04-13]. Dostupné z WWW: <http://www.tremani.nl/open-source/neural-network/>.

[30] *PHP: Hypertext Preprocessor* [online]. c2001 [cit. 2011-04-13]. PHP: stats_stat_correlation - Manual. Dostupné z WWW: <http://php.net/manual/en/function.stats-stat-correlation.php>.

8 Přílohy

8.1 PŘÍLOHA A – Administrátorská příručka

Aplikace je umístěná na freehostingovém účtu serveru Internet Centrum IC.cz. Tento účet se administruje přihlášením na http://user.ic.cz.

Administrace databáze se provádí na http://mysql.ic.cz/phpmyadmin/.

8.1.1 Výpis kořenového adresáře webové aplikace

```
|   .htaccess
|   check-usd.php
|   check-xau.php
|   check4.php
|   index.php
|   parsing_funkce2.php
|
├──cfg
|       config.php
|       db2.php
|
├──grafy
|   |   index.php
|   |
|   ├──data-script
|   |       fixing.php
|   |       kurz_cnb.php
|   |       obchody.php
|   |
|   └──lib-flot
|           excanvas.min.js
|           jquery.flot.js
|           jquery.js
|
└──vyzkum
        bin_bin_bin_gui.php
        bin_bin_bin_vypocet.php
        class_neuralnetwork.php
        predikce_gui.php
        predikce_vypocet.php
        stat.php
        tridy_tridy_bin_gui.php
        tridy_tridy_bin_vypocet.php
        tridy_tridy_tridy_gui.php
        tridy_tridy_tridy_vypocet.php
```

8.1.2 Popis jednotlivých částí

Adresář **cfg** obsahuje konfigurační soubory základního nastavení.

V souboru **config.php** se nastavují přihlašovací údaje pro spojení s databází, jsou zde nastaveny názvy tabulek v databázi a také mobilní mail pro zasílání informací o nových cenách

Soubor **db2.php** je skript zajišťující spojení s databází. Dále obsahuje dvě funkce pro provádění dotazů do databáze:

Funkce *dotaz* provádí dotaz (typu SELECT) do databáze a výsledek vrací v asociativním poli.

Funkce *dotazNo* provádí dotaz bez navracení výsledku (pro dotazy typu INSERT).

Datové pumpy ceny zlata na trhu a aktuálního kurzu dolaru jsou realizovány ve skriptech **check-usd.php** a **check-xau.php**. Oba dva získávají odpověď webové služby.

Skript **check4.php** provádí kontrolu cen v obchodech. Funkce *check* porovnává aktuální cenu produktu a poslední cenu z databáze. Pokud je aktuální cena jiná, provede zápis této nové ceny do databáze. Při změně ceny alespoň jednoho produktu je pomocí funkce *mail* odeslána informace o pohybu ceny. Ke zjišťování aktuálních cen se využívá skript **parsing_funkce2.php**, který obsahuje konkrétní funkce provádějící textový parsing a dekompozici obrázku.

Spouštění předchozích skriptů probíhá automatizovaně, pomocí systému cron. Konfigurace se provádí v Klientské sekci v části IC Tools → Cron. Skripty **check-usd.php** a **check-xau.php** jsou spouštěny každých deset minut. Skript **chceck4.php** je spouštěn každých 15 minut. Podle toho je nastavení v klientské sekci následující:

Hodina	Minuta	Den v měsíci	Měsíc	Den v týdnu	Url
*	0,15,30,45	*	*	*	http://gold.howto.cz/check4.php
*	0,10,20,30,40,50	*	*	*	http://gold.howto.cz/check-xau.php
*	0,10,20,30,40,50	*	*	*	http://gold.howto.cz/check-usd.php

Hlavní skript **index.php** v kořenovém adresáři slouží ke kontrole funkčnosti datových pump. Spuštěním tohoto skriptu se stejným způsobem jako ve skriptu **check4.php** provede načtení všech aktuálních cen. Navíc se do tabulky zobrazí i odkazy na zdrojové stránky, ze kterých jsou ceny zjišťovány.

V adresáři **grafy** se nalézají skripty, sloužící k vizualizaci dat z databáze. Samotné vykreslování grafů provádí javascriptová knihovna Flot v podadreáři **lib-flot**. Skript **index.php** obsahuje grafické rozhraní pro výběr vykreslovaných dat a nastavení grafů projektu Flot. Zpracování dat pro vykreslení zajišťují skripty v podadresáři **data-script**:

Skript **fixing.php** zajišťuje načtení zlatého londýnského fixu ve zvoleném období.

Skript **kurz_cnb.php** zajišťuje načtení kurzu dolaru ČNB ve zvoleném období.

Skript **obchody.php** zajišťuje načtení cen vybraného obchodu a produktu ve zvoleném období.

V adresáři **vyzkum** se nalézají skripty pro provádění výpočtů neuronovou sítí nad daty z databáze. Soubory obsahující v názvu **„_gui“** obsahují grafické rozhraní pro nastavení experimentů. Soubory obsahující v názvu **„_vypocet“** provádějí výpočet. Tyto skripty využívají implementaci neuronové sítě ze souboru **class_neuralnetwork.php**. Pro vypočítání výsledků výpočtů se využívá soubor **stat.php**, který obsahuje statistické funkce.

Protože doba výpočtu neuronovou sítí může být dlouhá, je v kořenovém adresáři ještě soubor **.htaccess**, ve kterém je nastavena maximální doba provádění php skriptů. Tuto dobu ale umožňuje poskytovatel hostingu nastavit maximálně na 6 minut.

8.2 PŘÍLOHA B – Uživatelská příručka

Webová stránka gold.howto.cz nabízí jako podporu pro investování do zlatých slitků pro uživatele tři služby:

8.2.1 Rychlé zobrazení cen produktů na jednom místě

Zobrazením stránky na adrese http://gold.howto.cz se na jednom místě v tabulce vypíše aktuální prodejní a výkupní cena zlatých slitků o hmotnosti 1 Oz a 100 g ze dvou vybraných obchodů.

Záhlaví tabulky obsahuje odkazy na stránky produktů v elektronických obchodech vybraných obchodů. Tím se lze rychle dostat na místo, odkud lze provést nákup.

8.2.2 Prohlížení grafů cen souvisejících s investičním zlatem

Z hlavní stránky se lze odkazem „Grafy“ dostat na adresu http://gold.howto.cz/grafy, na které je možné si prohlížet grafy cen souvisejících s investičním zlatem.

Vykreslení grafů proběhne po zadání časového období a obchodu a produktu. Časové období se zadává intervalem ohraničeným dny „od“ a „do“. Oba dva dny se zadají zápisem do textových polí v pořadí číslo dne, měsíce a roku. Jako počáteční den je přednastaven 1.1.2009 a koncový je aktuální den.

Obchod a produkt se vybere v rozbalovacím seznamu.

Vykreslení se provede po stisknutí tlačítka „Zobrazit grafy“.

Horní graf zobrazuje průběh odpoledního londýnského zlatého fixu v dolarech za 1 Oz žlutou čarou s legendou „Fix PM [USD/Oz]“ a měřítkem vlevo ve vybraném časovém období. Dále zobrazuje průběh kurz dolaru ČNB

v českých korunách modrou barvou s legendou „Kurz dolaru ČNB [Kč/USD]“ a měřítkem vpravo ve stejném časovém období.

Spodní graf zobrazuje průběh ceny ve vybraném obchodě v českých korunách za vybraný produkt černou čarou s legendou „Cena v obchodě [Kč]“ ve stejném časovém období.

8.2.3 Predikování cen pomocí umělé neuronové sítě

Z hlavní stránky se lze odkazem „Predikce“ dostat na adresu, na které je možné experimentovat s predikcí cen pomocí umělé neuronové sítě. Soubory končící na „gui.php“ jsou skripty s grafickým uživatelským rozhraním, pomocí nichž se nastavují a spouští experimenty. Tato část aplikace je určena pro uživatele seznámeného s principy umělých neuronových sítí.

Skript	**Druh experimentu**
bin_bin_bin_gui.php	Predikce trendu – vstupy i výstupy jsou klasifikovány bipolárně (1 nebo -1).
tridy_tridy_bin_gui.php	Predikce trendu – vstupy je možné klasifikovat do zvoleného počtu tříd o zvolených velikostech, výstup je bipolární (1 nebo -1).
tridy_tridy_tridy_gui.php	Predikce trendu – vstupy i výstupy je možné klasifikovat do zvoleného počtu tříd o zvolených velikostech.
predikce_gui.php	Predikování hodnoty.

Tabulka 8.1: Přehled skriptů s grafickým uživatelským rozhraním pro nastavení experimentů.

Rozhraní pro spouštění jednotlivých experimentů je podobné. Nastavení je vždy rozděleno do několika očíslovaných kroků.

1. **Výběr obchodu**

Zde se z rozbalovacího seznamu vybere obchod a produkt, na kterém bude probíhat výpočet.

2. Nastavení vstupních dat

Predikce trendu		
název	**příklad zápisu**	**význam**
Časová období	1.2.2010-1.8.2010	Období, ze kterého se budou vytvářet vzory pro neuronovou síť. Data jsou oddělena pomlčkou bez mezer. Je možné zadat více období – každé na samostatný řádek v textovém poli.
Zpoždění obchodu	30	Celé číslo udává zpoždění v minutách. Podle tohoto údaje se posouvá interval, ve kterém se zjišťuje průběh ceny zlata na trhu a kurzu dolaru. Lze zadat více zpoždění – každé na samostatný řádek.
Třídy změny kurzu dolaru	0.05/3	Udává rozdělení změny kurzu dolaru do více tříd. První údaj před lomítkem je desetinné číslo s desetinnou tečkou, udávající velikost třídy v korunách za dolar. Za lomítkem je počet tříd, do kterých je kurz rozdělen.
Třídy změny ceny zlata na burze	6/3	Stejné jako u *Třídy změny kurzu dolaru*. Před lomítkem je velikost třídy v dolarech za 1 Oz. Za lomítkem je počet tříd.
Třídy ceny v obchodě	55/4	Stejné jako u *Třídy změny kurzu dolaru*. Před lomítkem je velikost třídy v korunách za zvolený produkt. Za lomítkem je počet tříd.
Predikce hodnoty		
název	**příklad zápisu**	**význam**
Časová období (trénovací)	1.1.2009-1.1.2010	Období, ze kterého se vytvářejí trénovací vzory. Stejný způsob zápisu jako u predikce trendu.
Počet trén. vzorů	300	Přirozené číslo udávající maximální počet trénovacích vzorů.

Časová období (testovací)	1.1.2009-1.1.2010	Období, ze kterého se vytvářejí testovací vzory. Stejný způsob zápisu jako u predikce trendu.
Počet test. vzorů	50	Přirozené číslo udávající maximální počet testovacích vzorů.
Čas	12:00	Čas ve kterém se provádí první navzorkování ceny v obchodě.
Časová jednotka	86400	Časový interval mezi jednotlivými vzorky v sekundách (86400 s = 1 den).
Počet vzorků v okně	10	Počet vzorků v okně, na základě kterých se predikuje. Lze zadat více počtů – každý na samostatný řádek.
Vzdálenost předpovídaného	2	Vzdálenost predikované hodnoty v předem zadaných časových jednotkách. Lze zadat více vzdáleností – každou na samostatný řádek.

3. Nastavení neuronové sítě

název	**příklad zápisu**	**význam**
Topologie	7-14-1	Topologie sítě. První číslo je počet neuronů ve vstupní vrstvě. Počet vstupních neuronů musí být u predikce trendu součtem počtu tříd pro kurz dolaru a ceny zlata na trhu. U predikce hodnoty musí být stejný jako je počet vzorků v okně zvětšený o dvě (intervenční proměnné kurzu dolaru a ceny zlata na trhu). Následují počty neuronů ve skrytých vrstvách oddělené pomlčkami. Poslední číslo, udávající počet výstupních neuronů, musí být u predikce trendu stejné jako je počet tříd cen v obchodě. U predikce hodnoty musí být 1. Lze zadat více topologií – každou na samostatný řádek.

Počet učících kroků	1000	Počet učících kroků. Lze zadat více počtů - každý na samostatný řádek.
Rychlost učení	0.1	Desetinné číslo s desetinnou tečkou udávající rychlost učení sítě. Lze zadat více rychlostí – každou na samostatný řádek.
Moment	0.9	Desetinné číslo s desetinnou tečkou, udávající parametr momentu pro učení sítě. Lze zadat více parametrů – každý na samostatný řádek.
Průběžný výpis	Ano/Ne	Zaškrtávací tlačítko. Pokud je zaškrtnuto, bude během výpočtu probíhat průběžný výpis velikostí chyb.

4. Dotazy

U predikování trendu je možné zadat síti normalizované dotazy, na které po naučení síť odpoví. Každý dotaz je ve formě vektoru hodnot 1 a -1 oddělených čárkou. Velikost vektoru musí odpovídat součtu zadaných počtů tříd pro kurz dolaru a cenu zlata na trhu. Je možné zadat více dotazů – každý na samostatný řádek.

Po stisknutí tlačítka „Spustit experiment“ dojde k výpočtům pro všechny přípustné variace zadaných parametrů (např. musí odpovídat topologie).

Výpis výsledků u predikce trendu

Po výpočtech predikcí trendu je nejprve vypsána tabulka posledních změn kurzu dolaru a ceny zlata na trhu, jak v absolutních číslech, tak i normalizovaně. Dále je vypsána tabulka shrnující rozdělení kurzu dolaru a ceny zlata na trhu do tříd. Výsledky výpočtů jsou poté vypsány následujícím způsobem:

Pro každé časové období a zpoždění obchodu je vytisknuta jedna tabulka, která na prvních dvou řádcích zobrazuje právě tyto parametry. Dále jsou

v textových polích vypsána vstupní data podle období a zpoždění, a z nich vygenerované vzory pro neuronovou síť podle nastavení tříd. Tyto údaje lze vykopírovat a použít pro experimenty v jiných programech. Na dalším řádku je informace o počtu vygenerovaných vzorů, rychlosti učení a parametru momentu. Uvnitř posledního řádku jsou již tabulky s konkrétními výsledky výpočtů. V každé tabulce je v levém sloupci číslo, udávající počet učících kroků a v pravém sloupci je postupně vypsána topologie, RMS chyba, odpověď na poslední změnu (což je vlastně predikce nejbližší reakce obchodu) a nakonec odpovědi na zadané dotazy (spojený vstupní i výstupní vektor s hodnotami oddělenými čárkami).

Výpis výsledků u predikce hodnoty

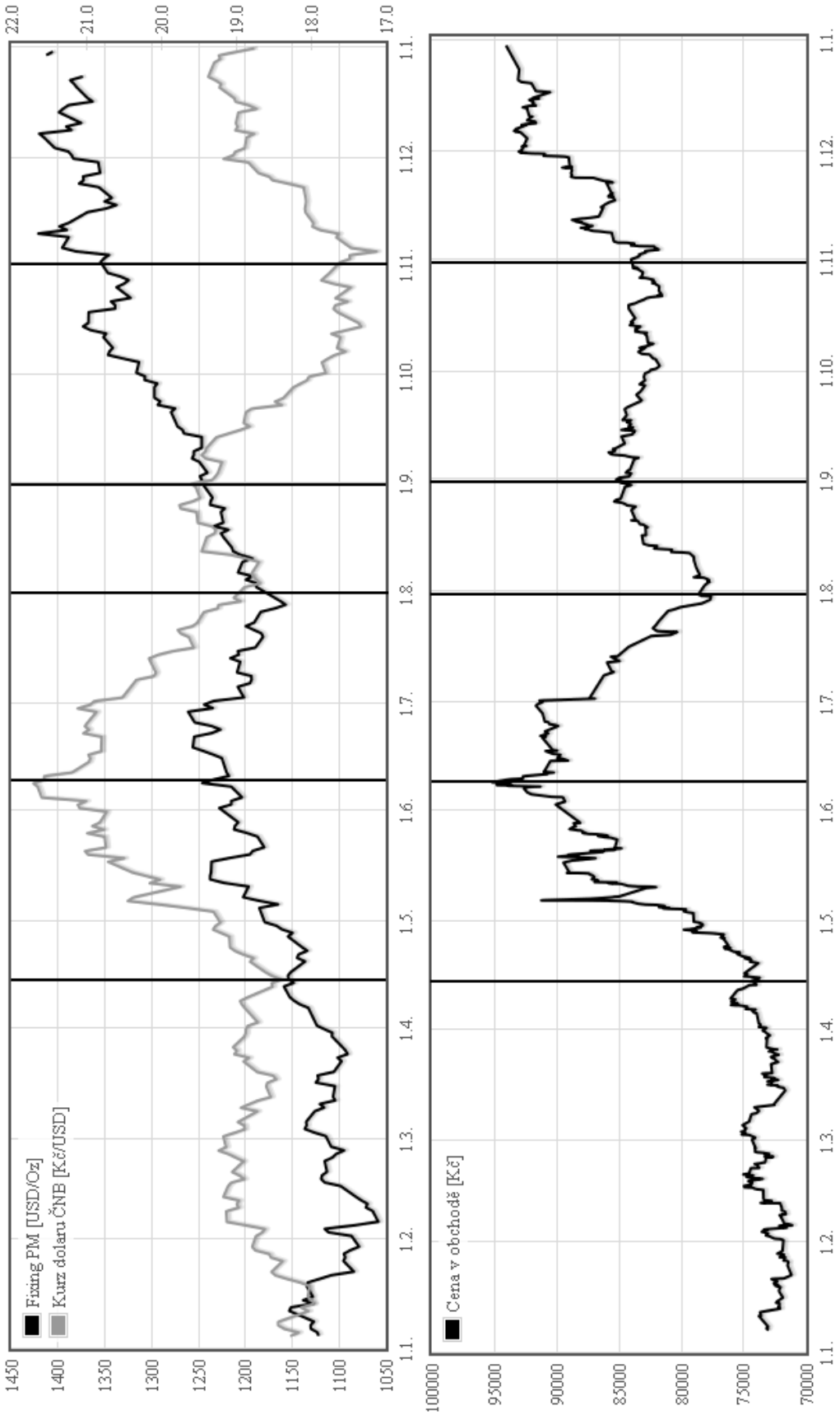
Po výpočtech je nejprve vypsána časová jednotka (interval mezi vzorky) v sekundách. Dále jsou vypsána trénovací a testovací období, velikost okna, vzdálenost předpovídaného, rychlost učení, parametr momentu, počet učících kroků a všechny topologie. Pokud topologie vyhovuje předešlým parametrům, jsou v textových polích vypsány reálné a predikované hodnoty na trénovací a testovací množině před i po korekci posunu. Formát těchto hodnot je:

```
datum a čas|reálná hodnota|predikovaná hodnota
```

Tyto údaje lze vykopírovat a např. v programu Excel rozdělit pomocí oddělovače „|“ a vizualizovat.

V tabulce jsou dále vypsány výsledky výpočtu: RMS chyby na trénovací a testovací množině, průměrné absolutní chyby na trénovací a testovací množině, korelační koeficienty mezi reálnými a predikovanými řadami na trénovací a testovací množině (vše před i po korekci posunu) a predikovaná hodnota do budoucnosti spolu s časem, udávajícím, na kdy se predikuje.

8.3 PŘÍLOHA C – Graf vývoje cen během roku 2010



8.4 PŘÍLOHA D – Výsledky výpočtů

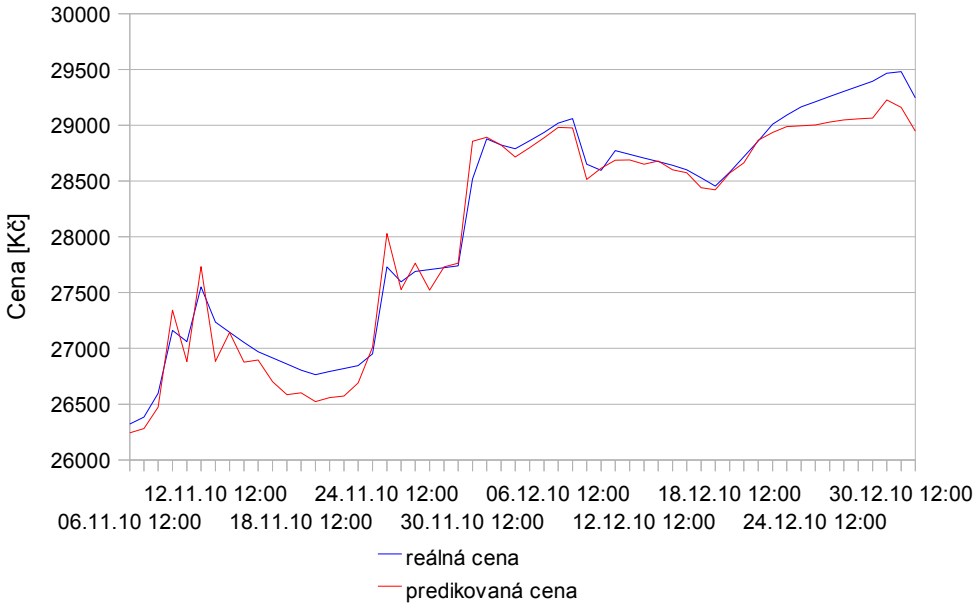
8.4.1 Výpočet demonstrující systematický posun

Nastavení experimentu	
Obchod a produkt	Zlaté Mince – Numismatika, 1 Oz
Časová jednotka	86400 s (1 den)
Trénovací období	1.1.2010-1.11.2010
Testovací období	1.11.2010-1.1.2011
Velikost okna	5
Vzdálenost předpovídaného	1
Rychlost učení	0,1
Parametr momentu	0,9
Počet učících kroků	500
Topologie	7-5-1

Tabulka 8.2: Nastavení experimentu pro demonstraci systematického posunu.

Výsledky výpočtu	
RMS chyba trénovací množiny	0,06242355
RMS chyba testovací množiny	0,06523133
Průměrná absolutní chyba na trénovací množině	187,96 Kč
Průměrná absolutní chyba na trénovací množině (posun)	143,46 Kč
Průměrná absolutní chyba na testovací množině	206,94 Kč
Průměrná absolutní chyba na testovací množině (posun)	134,45 Kč
Koeficient korelace na trénovací množině	0,99413242
Koeficient korelace na trénovací množině (posun)	0,99805074
Koeficient korelace na testovací množině	0,97199315
Koeficient korelace na testovací množině (posun)	0,98905221

Tabulka 8.3: Výsledky výpočtu pro demonstraci systematického posunu.



Graf 8.1: Hodnoty na testovacím období po korekci posunu.

8.4.2 Výpočet vlivu vzdálenosti na predikci

Nastavení experimentu	
Obchod a produkt	Zlaté Mince – Numismatika, 1 Oz
Časová jednotka	86400 s
Trénovací období	1.2.2010-1.11.2010
Testovací období	1.12.2009-1.2.2010
Velikost okna	5
Vzdálenost předpovídaného	1 až 15 *(předmět zkoumání)*
Max. počet trénovacích vzorů	255
Max. počet testovacích vzorů	44
Rychlost učení	0.1
Parametr momentu	0.9
Počet učících kroků	500
Topologie	7-5-1

Tabulka 8.4: Nastavení experimentu pro zkoumání vlivu vzdálenosti.

Trénovací množina					
Vzdálenost	RMS	MAD	MAD posun	r	r posun
1	0,05157792	144,72	96,33	0,99393399	0,99812840
2	0,06134426	175,83	151,59	0,99168923	0,99508014
3	0,06040990	173,03	163,2	0,99195911	0,99321224
4	0,06964176	197,52	185,09	0,98813298	0,99034604
5	0,06247713	178,17	167,84	0,99105088	0,99295285
6	0,06042994	175,15	167,78	0,99130556	0,99239027
7	0,06375844	182,82	169,39	0,99081544	0,99242748
8	0,05919494	165,24	159,76	0,99136344	0,99264455
9	0,07350469	229,52	224,8	0,98959811	0,99080507
10	0,06679605	191,01	163,28	0,98977099	0,99315985
11	0,06076818	170,27	161,16	0,99091505	0,99227865
12	0,06102229	170,72	162	0,99087461	0,99282234
13	0,06438832	183,76	173,36	0,99087890	0,99237176
14	0,06231167	175,84	179,24	0,99102884	0,99183893
15	0,06360449	178,02	181,88	0,99030635	0,99100754

Testovací množina					
Vzdálenost	RMS	MAD	MAD posun	r	r posun
1	0,04747113	104,09	81,35	0,87443153	0,94834466
2	0,03719381	109,23	96,17	0,77507329	0,88870835
3	0,05103221	218,64	213,55	0,77363805	0,86834049
4	0,07709544	276,54	273,74	0,57397653	0,65384801
5	0,04292769	136,28	125,93	0,70287457	0,77713546
6	0,04133806	133,72	123,42	0,65594734	0,71518361
7	0,04364480	143,67	140,67	0,61618546	0,61662745
8	0,04559820	132,23	132,78	0,79424684	0,79823106
9	0,04337432	115,39	112,75	0,72681303	0,75713515
10	0,05138662	160,4	157,14	0,75901948	0,78312928
11	0,04420068	119,49	117,22	0,72310622	0,75518237
12	0,03722669	108,18	104,67	0,75172259	0,77737321
13	0,04330733	125,53	121,66	0,74444063	0,77471654
14	0,05040929	179,01	176,57	0,70909956	0,74170695
15	0,04561853	150,4	150,34	0,74031219	0,78339323

Tabulka 8.5: Výsledky experimentu pro zkoumání vlivu vzdálenosti.

8.4.3 Výpočet vlivu velikosti okna na predikci

Nastavení experimentu	
Obchod a produkt	Zlaté Mince – Numismatika, 1 Oz
Časová jednotka	86400 s (1 den)
Trénovací období	1.2.2010-1.11.2010
Testovací období	1.12.2009-1.2.2010
Max. počet trénovacích vzorů	250
Max. počet testovacích vzorů	40
Topologie	vždy jedna skrytá vrstva se 4 neurony
Velikost okna	1 až 15 *(předmět zkoumání)*
Vzdálenost předpovídaného	1
Rychlost učení	0.1
Parametr momentu	0.9
Počet učících kroků	500

Tabulka 8.6: Nastavení experimentu pro zkoumání vlivu velikosti okna.

Trénovací množina					
Velikost okna	RMS	MAD	MAD posun	r	r posun
1	0,05036008	138,14	65,99	0,99355569	0,99863805
2	0,05679299	165,94	113,3	0,99344836	0,99789538
3	0,04767818	128,33	79,98	0,99423251	0,99804081
4	0,05230615	148,05	106,95	0,99362894	0,99735745
5	0,04754127	132,68	86,82	0,99426369	0,99772773
6	0,05190804	150,01	111,15	0,99421421	0,99769380
7	0,04854247	133,51	87,56	0,99400206	0,99765092
8	0,04792617	133,18	100,28	0,99432063	0,99724907
9	0,05209964	150,1	110,99	0,99357194	0,99686182
10	0,05018785	146,08	119,09	0,99433538	0,99689895
11	0,05195653	149,32	111,26	0,99361586	0,99712055
12	0,04833797	138,48	92,08	0,99451028	0,99775192
13	0,04875807	137,35	104,39	0,99419842	0,99689940
14	0,05507176	155,7	122,25	0,99347556	0,99668367
15	0,04926153	140,63	125,33	0,99454305	0,99619504

Testovací množina					
Velikost okna	RMS	MAD	MAD posun	r	r posun
1	0,02780199	124,16	104,81	0,86813141	0,96005839
2	0,02569079	117,86	108,38	0,89300747	0,97241600
3	0,02544513	95,68	72,81	0,88698485	0,96418358
4	0,02856029	85,4	62,58	0,87895826	0,95767078
5	0,03081810	98,94	76,69	0,86115636	0,96524690
6	0,03149095	108,41	80,8	0,86940318	0,97485789
7	0,05172463	120,18	93,37	0,84569400	0,93478872
8	0,03165017	81,9	58,24	0,87133085	0,96636255
9	0,02613687	109,25	88,04	0,87326218	0,96096359
10	0,03431584	103,36	88	0,84130188	0,93759145
11	0,02830640	79,71	54,63	0,88287155	0,94713529
12	0,02888025	114,88	94,71	0,85779968	0,94977785
13	0,03455584	78,25	51,99	0,86951000	0,96253245
14	0,03256318	116,93	104,18	0,86508352	0,96475559
15	0,03808996	86,93	68	0,87143903	0,95921215

Tabulka 8.7: Výsledky experimentu pro zkoumání vlivu velikosti okna.

8.4.4 Výpočet hledání optimální topologie

Nastavení experimentu	
Obchod a produkt	Zlaté Mince – Numismatika, 1 Oz
Časová jednotka	86400 s (1 den)
Trénovací období	1.2.2010-1.11.2010
Testovací období	1.12.2009-1.2.2010
Velikost okna	8
Vzdálenost předpovídaného	1
Topologie	různé *(předmět zkoumání)*
Rychlost učení	0.1
Parametr momentu	0.9
Počet učících kroků	500

Tabulka 8.8: Nastavení experimentu pro hledání optimální topologie.

Trénovací množina					
Topologie	RMS	MAD	MAD posun	r	r posun
10–1	0,05635123	154,61	121,06	0,99252387	0,99617361
10–2–1	0,05071355	138,49	96,88	0,99403842	0,99743924
10–5–1	0,05697302	165,76	125,53	0,99324350	0,99682192
10–10–1	0,05320447	154,18	124,82	0,99396101	0,99666477
10–15–1	0,13987245	478,02	477,93	0,97882980	0,98131286
10–20–1	0,11221854	362,78	341,07	0,97849906	0,98219940
10–5–2–1	0,09407704	255,82	240,14	0,98111994	0,98411847
10–10–5–1	0,05662167	159,31	132,15	0,99316064	0,99612035
10–5–5–5–1	0,34905836	1118,73	1111,96	0,77525789	0,78342101

Tabulka 8.9: Výsledky experimentu na trénovací množině pro hledání optimální topologie.

Testovací množina					
Topologie	RMS	MAD	MAD posun	r	r posun
10–1	0,04314811	177,79	179,54	0,91218862	0,95978557
10–2–1	0,05385224	148,41	141,78	0,88752487	0,95110956
10–5–1	0,03961745	141,54	123,65	0,89460637	0,94529082
10–10–1	0,04219086	130,18	118,61	0,89865641	0,95014726
10–15–1	0,04677006	336,5	339,55	0,90025610	0,96994483
10–20–1	0,06007663	149,69	136,42	0,90198186	0,96212975
10–5–2–1	0,15359112	429,41	428,64	0,47264810	0,50140756
10–10–5–1	0,06205780	203,67	174,16	0,87244787	0,93622474
10–5–5–5–1	0,30513962	2137,22	2135,96	-2,92E-014	-1,80E-014

Tabulka 8.10: Výsledky experimentu na testovací množině pro hledání optimální topologie.

Trénovací množina					
Topologie	RMS	MAD	MAD posun	r	r posun
10—4—1	0,05464430	157,53	118,68	0,99322736	0,99668603
10—5—1	0,05307982	151,79	114,07	0,99461149	0,99805609
10—6—1	0,05501179	152,98	125,6	0,99400439	0,99727811
10—7—1	0,05004335	135,56	91,21	0,99419137	0,99759701
10—8—1	0,05215207	145,17	118,79	0,99446780	0,99750731
10—9—1	0,05099908	141,85	120,03	0,99490912	0,99705949
10—10—1	0,04873032	135,02	107,37	0,99446185	0,99652085
10—11—1	0,05067186	138,66	118,62	0,99432383	0,99604263
10—12—1	0,05262838	152,01	128,03	0,99406335	0,99626811
10—13—1	0,05587289	162,35	124,28	0,99381859	0,99703736

Tabulka 8.11: Výsledky experimentu na trénovací množině pro hledání optimální topologie.

Testovací množina					
Topologie	RMS	MAD	MAD posun	r	r posun
10—4—1	0,05448899	122,16	100,6	0,84979008	0,88720429
10—5—1	0,04698576	131,05	119,91	0,89975264	0,96651467
10—6—1	0,04048109	181,01	183,11	0,88450388	0,93510098
10—7—1	0,05667950	145,35	136,57	0,87682960	0,94464380
10—8—1	0,03549771	121,93	112,81	0,91402762	0,97368943
10—9—1	0,06981153	200,21	198,05	0,90287543	0,96060732
10—10—1	0,05494833	131,35	115,41	0,90212977	0,96069648
10—11—1	0,04747135	142,88	139,24	0,91221324	0,97083550
10—12—1	0,04596996	132,43	119,86	0,90423280	0,96284270
10—13—1	0,07899259	147,93	142,01	0,84500687	0,86483045

Tabulka 8.12: Výsledky experimentu na testovací množině pro hledání optimální topologie.

8.5 PŘÍLOHA E – Obsah přiloženého CD

```
|   Diplomová práce, Zdeněk Zahoř (P06500).pdf
|
├───db
|       ht_gold.sql
|
└───src
```

Přiložené CD obsahuje text této práce ve formátu PDF, exportovanou databázi s nasbíranými daty v souboru **ht_gold.sql** v adresáři **db** a zdrojové kódy webové aplikace v adresáři **src**.