

**UNIVERZITA PARDUBICE
FAKULTA EKONOMICKO-SPRÁVNÍ
ÚSTAV SYSTÉMOVÉHO INŽENÝRSTVÍ A INFORMATIKY**

**MODELOVÁNÍ HRUBÉHO
DOMACÍHO PRODUKTU POMOCÍ
FRONTÁLNÍCH NEURONOVÝCH SÍTÍ**

DIPLOMOVÁ PRÁCE

AUTOR:

Marek SLAVÍK

VEDOUCÍ PRÁCE:

prof. Ing. Vladimír OLEJ, CSc.

KONZULTANT:

Ing. Ivan ŠEDA

2007

**UNIVERSITY OF PARDUBICE
FACULTY OF ECONOMICS AND ADMINISTRATION
INSTITUTE OF SYSTEM ENGINEERING AND INFORMATICS**

**MODELLING OF GROSS DOMESTIC
PRODUCT BY FRONTAL NEURAL
NETWORKS**

THESIS

**AUTHOR:
SUPERVISOR:
ADVISER:**

**Marek SLAVÍK
prof. Ing. Vladimír OLEJ, CSc.
Ing. Ivan ŠEDA**

2007

Vyhrazeno pro zadávací list (strana 1)

Vyhrazeno pro zadávací list (strana 2)

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 20. května 2007

Marek Slavík

PODĚKOVÁNÍ

Touto cestou bych chtěl poděkovat vedoucímu diplomové práce, panu prof. Ing. Vladimírovi Olejovi, CSc. za odbornou pomoc, připomínky k obsahové i formální stránce diplomové práce, vedení při vypracovávání a podporu.

Rád bych dále poděkoval rodičům za jejich celoživotní podporu, pochopení a trpělivost a také za to, že mi umožnili studovat.

Shrnutí

Diplomová práce je zaměřena na návrh modelů frontálních neuronových sítí pro predikci hrubého domácího produktu. Pro učení neuronové sítě je použit index vedoucích ekonomických indikátorů a jeho difúzní index, jimž odpovídá procentuální změna hrubého domácího produktu. Následně je v diplomové práci řešen způsob modelování trendových složek těchto indexů pomocí klouzavých průměrů a exponenciálních vyrovnání. Je zde definována frontální neuronová síť a zmíněny další neuronové sítě vhodné pro predikci. V další části práce jsou analyzovány navržené modely frontálních neuronových sítí.

Summary

This diploma work deals with a suggestion for simulating frontal neural network model for prediction of gross domestic product. For training frontal neural network is applied index of leading economics indicators and diffusion index of leading economics indicators, for which belongs percentage change of gross domestic product. In next part of this diploma work is solved method of modelling trend elements of this indexes by moving averages and exponential adjustments. There is defined frontal neural network and another neural networks that is available for prediction. In last part of this diploma are analyzed models of frontal neural networks.

Obsah

Obsah	- 8 -
Úvod.....	- 10 -
1 Základní pojmy	- 11 -
1.1 Hrubý domácí produkt	- 11 -
1.1.1 Metody vyjádření HDP	- 11 -
1.1.2 Růst HDP	- 14 -
1.2 Hospodářský cyklus	- 14 -
1.2.1 Fáze hospodářského cyklu	- 14 -
1.2.2 Typy hospodářských cyklů	- 15 -
1.3 Ekonomické indikátory	- 15 -
1.3.1 Index vedoucích ekonomických indikátorů	- 16 -
1.3.2 Index koincidence a index zaostávajících indikátorů	- 18 -
1.4 Shrnutí kapitoly	- 18 -
2 Modely na predikci vývoje HDP	- 20 -
2.1 Metody predikce vývoje HDP	- 20 -
2.2 Zjišťování míry přesnosti modelů pro predikce vývoje HDP	- 21 -
2.3 Modely pro predikci vývoje HDP	- 21 -
2.4 Shrnutí kapitoly	- 23 -
3 Časové řady a jejich předzpracování	- 24 -
3.1 Dekompozice časové řady.....	- 24 -
3.1.1 Metoda klouzavých průměrů	- 26 -
3.1.2 Exponenciální vyrovnávání	- 28 -
3.2 Shrnutí kapitoly	- 30 -
4 Neuronové sítě	- 31 -
4.1 Neuronové sítě a umělá inteligence	- 31 -
4.2 Definice neuronové sítě.....	- 32 -
4.3 Shrnutí kapitoly	- 39 -
5 Neuronové sítě vhodné pro predikci	- 40 -
5.1 Frontální neuronová síť	- 40 -
5.2 Dopředná neuronová síť	- 41 -
5.3 Perceptronová neuronová síť	- 43 -
5.4 Učící algoritmus Back-Propagation	- 43 -
5.5 Shrnutí kapitoly	- 45 -
6 Návrh modelu frontální neuronové sítě na predikci HDP	- 46 -
6.1 Návrh frontální neuronové sítě.....	- 46 -

6.2	Výsledky učení neuronové sítě	- 49 -
6.2.1	Protokol o průběhu učení neuronové sítě	- 50 -
6.2.3	Nastavení vah na synapsích a prahy neuronů	- 51 -
6.3	Shrnutí kapitoly	- 52 -
7	Analýza výsledků.....	- 53 -
7.1	Vstupní data	- 53 -
7.2	Modely s hloubkou paměti filtru $b = 4$	- 54 -
7.3	Modely s hloubkou paměti filtru $b = 5$	- 54 -
7.4	Modely s hloubkou paměti filtru $b = 6$	- 55 -
7.5	Modely s hloubkou paměti filtru $b = 7$	- 55 -
7.6	Modely s hloubkou paměti filtru $b = 8$	- 55 -
7.7	Modely s hloubkou paměti filtru $b = 9$	- 56 -
7.8	Přehled výsledků modelování	- 56 -
7.8	Shrnutí kapitoly	- 57 -
8	Závěr	- 58 -
	Seznam použité literatury	- 59 -
	Seznam obrázků.....	- 61 -
	Seznam tabulek.....	- 61 -
	Seznam příloh	- 62 -
	Seznam použitých zkratek	- 63 -
	Příloha 1: Uživatelská příručka k architektuře Time-Delay	- 64 -
	Příloha 2: Protokoly o způsobu naučení vybraných modelů neuronových sítí, jejich nastavení vah synapsí a prahů neuronů.....	- 71 -

Úvod

Hrubý domácí produkt je jedním z komplexních ukazatelů produkční kvality ekonomiky. Jeho možný i skutečný vývoj patří k důležitým faktorům při určování hospodářské politiky státu, při rozhodování o finančních investicích podniků a domácností, při sestavování výrobních programů podniků a dalších důležitých rozhodnutích ekonomických subjektů. Cílem diplomové práce je analýza vstupních parametrů, předzpracování daných časových řad a charakteristika frontálních neuronových sítí z hlediska predikce. Dále je cílem práce návrh modelů predikce hrubého domácího produktu pomocí frontální neuronové sítě a následná verifikace a analýza navržených modelů v prostředí Stuttgarského simulátoru neuronových sítí.

Diplomová práce je členěna celkem do sedmi hlavních kapitol. V první kapitole jsou uvedeny základní pojmy z oblasti hrubého domácího produktu a ekonomických ukazatelů. Ve druhé kapitole jsou popsány modely predikce hrubého domácího produktu od jednodušších až po komplexní modely, které vypracovávají významné organizace a instituce. Třetí kapitola je věnována časovým řadám, jejich členěním a dekompozicím. Jsou zde uvedeny způsoby modelování trendových složek časových řad. Ve čtvrté kapitole jsou definovány základní pojmy z teorie neuronových sítí. Pátá kapitola popisuje neuronové sítě, které jsou vhodné pro predikci. Frontální neuronová síť je navržena v šesté kapitole. V závěrečné kapitole jsou analyzovány navržené modely frontálních neuronových sítí.

1 Základní pojmy

V kapitole jsou definovány základní ekonomické pojmy a je rozdělena do tří částí. V první části je definován hrubý domácí produkt a spolu s ním i jeho druhy, rozdíly mezi nimi, způsoby výpočtu a definice jeho růstu. Následující část se zabývá cyklickými pravidly ve vývoji ekonomiky, délkou jednotlivých cyklů, jejich rozdělením a definováním jednotlivých fází. Poslední část první kapitoly se zabývá ekonomickými ukazateli, jednotlivými typy a možným využitím pro predikci hrubého domácího produktu.

1.1 Hrubý domácí produkt

Hrubý domácí produkt (HDP) je základním národohospodářským agregátem, který představuje veškerý peněžní tok za zboží a služby vyrobené v dané zemi za určité období. Je ukazatelem, podle kterého se posuzuje výkonnost ekonomiky a vypovídá taktéž o ekonomické situaci a životní úrovni v dané zemi. Celkový HDP vyjadřuje ekonomickou sílu dané země.

Reálný a nominální HDP

Ekonomické proměnné [1] představující hodnoty nebo ceny, je možno vyjádřit nominálně (běžná hodnota) nebo reálně (s úpravou zohledňující inflaci). Pomocí nominálního HDP je zjištěna hodnota výkonnosti ekonomiky v cenách za období, ve kterém byla daná hodnota vytvořena a odpovídá běžným cenám. Reálným HDP je zjištěna hodnota výkonnosti ekonomiky za dané období v cenách zvolených v základním roce, tj. stálé ceny. Nominální růst HDP odráží jak růst cen tak i produkce, reálný růst HDP odráží pouze růst produkce. Pomocí tohoto rozdělení je možno sledovat reálné změny objemu produkce, které nejsou ovlivněny cenovými změnami. Převod je prováděn pomocí tzv. deflátoru HDP - vydělením nominálního HDP reálným se získá jeden z nejsouhrnnějších cenových indexů - implicitní deflátor HDP.

1.1.1 Metody vyjádření HDP

K výpočtu HDP se používají [2] následující metody:

- výrobní metoda,
- výdajová metoda,

- důchodová metoda.

Výrobní metoda

Výrobní metoda vyjádření HDP je založena na výpočtu produkce a mezispotřeby odvětví v národní ekonomice. Hrubá přidaná hodnota odvětví je definována jako rozdíl mezi produkcí a mezispotřebou. V kupních cenách je potom HDP vypočítán jako součet hrubé přidané hodnoty všech odvětví, k němuž jsou připočítány daně z produktů a od něhož jsou odečítány dotace na produkty.

- Produkce výrobků a služeb **P** představuje hodnotu zboží a služeb, které jsou výsledkem produkčních činností výrobních jednotek v daném časovém období na určitém území.
- Mezispotřeba **M** představuje hodnotu zboží a služeb spotřebovaných v příslušném období místními producenty v procesu výroby jiného zboží a služeb.
- Daně z produktů **d**.
- Dotace **D** na produkty.

Vztah při použití výrobní metody je možno zapsat tímto způsobem

$$HDP = P - M + d - D . \quad (1.1)$$

Výdajová metoda

Tato metoda vyjádření HDP je založena na měření složek poptávky. Výdaje na konečnou spotřebu a hrubá tvorba kapitálu tvoří domácí poptávku a její součet s vývozy zboží a služeb konečnou poptávku.

- Výdaje na konečnou spotřebu **C** představují výdaje sektoru domácností, vlády a neziskových institucí sloužících domácnostem na konečnou spotřebu.
- Soukromé hrubé domácí investice **I**. Mezi investice se v podnikatelském sektoru počítají nákupy na obnovení a rozšíření zásob kapitálu, na zvýšení zásob surovin a hotových výrobků na skladě. V soukromém sektoru jsou to pak výdaje spojené s výstavbou rodinných domků.

- Výdaje státu na nákup výrobků a služeb **G**. Do státních výdajů se nezapočítávají státní transferové platby.
- Čistý vývoz **N_x** je rozdíl mezi vývozem a dovozem. Musí se odečíst platby za výrobky a služby dovezené ze zahraničí, jelikož tyto produkty nebyly vytvořeny v domácí ekonomice a naproti tomu se musí přičíst platby za vyvezené výrobky a služby, které se sice v ekonomice vyrobily, ale nespotřebovaly.

Vztah pro měření HDP je možno při použití výdajové metody zapsat tímto způsobem

$$HDP = C + I + G + N_x . \quad (1.2)$$

Důchodová metoda

Důchodová metoda vyjadřuje HDP z odhadů složek přidané hodnoty odvětví, tj. sčítá důchody, které vznikají při výrobě, a zachycuje jejich rozdělení mezi výrobní faktory a vládu. Důchody se člení na:

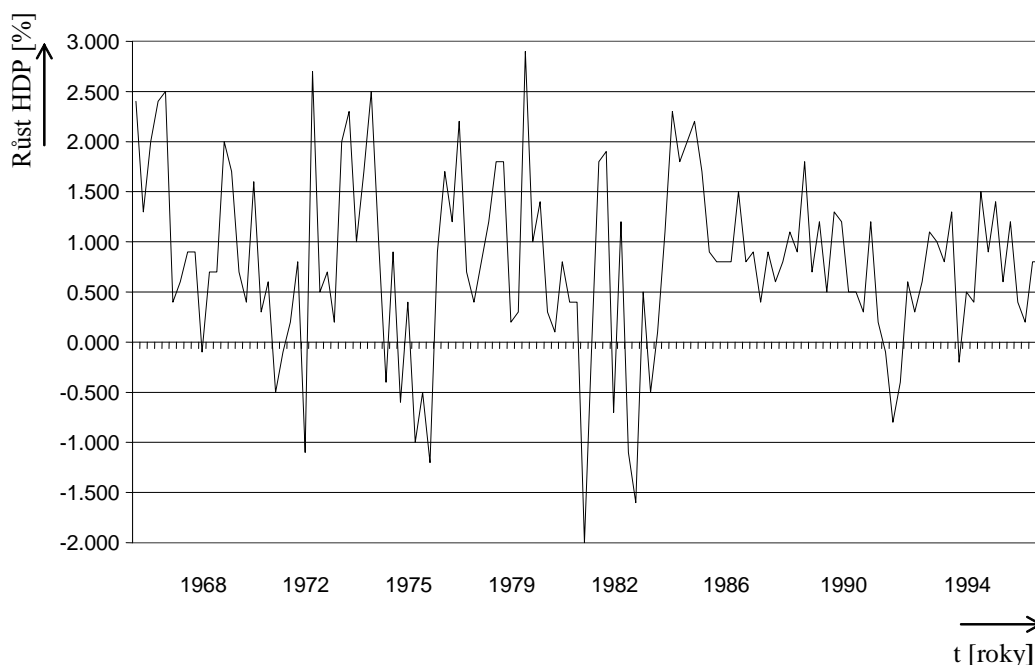
- Náhrady zaměstnancům zahrnují mzdy, platy a sociální příspěvky zaměstnavatelů **w** před zdaněním.
- Důchody samozaměstnavatelů **p**, kde jsou zahrnuty všechny formy plateb za výrobní faktory, jež používají osoby samostatně výdělečně činné, zisky firem v individuálním vlastnictví a zisky právnických osob.
- Renty majitelů půdy **r**.
- Čisté úroky z kapitálu **i**.
- Odpisy a amortizace **a**.
- Nepřímé daně **T**.

Vztah pro měření HDP je možno zapsat při použití důchodové metody tímto způsobem

$$HDP = w + p + r + i + a + T . \quad (1.3)$$

1.1.2 Růst HDP

Růst HDP je definován [2] jako změna HDP mezi dvěma obdobími. Může být kladný, nulový nebo záporný. Velikost růstu HDP nejvíce ovlivňují reálné a cenové změny. Základním předpokladem zvyšování životní úrovně obyvatel je dlouhodobě stabilní růst reálného HDP. Z obr. 1.1 je zřetelný procentuální čtvrtletní růst HDP v USA.



Obr. 1.1 Reálný růst HDP v USA

1.2 Hospodářský cyklus

Tržní ekonomika [1] se nevyvíjí hladce, ale podléhá ekonomickým výkyvům a dochází přitom k strukturálním a cyklickým výkyvům. Ke strukturálním výkyvům dochází vlivem neustále se měnících preferencí spotřebitelů a změnami vzácnosti ekonomických zdrojů, nově získanými technologickými a výrobním poznatky. Cyklické výkyvy jsou naopak charakterizovány všeobecným poklesem a poté zase všeobecným růstem výroby a zaměstnanosti téměř ve všech odvětvích.

1.2.1 Fáze hospodářského cyklu

Cyklické výkyvy [1] v ekonomice se nazývají hospodářské cykly a je pro ně typické střídání fáze expanze a kontrakce. Expanze znamená, že se růst reálného HDP zrychluje a HDP roste nad potenciální produkt. Na obr. 1.2 se nachází mezi dnem a vrcholem cyklu. Kontrakce naopak znamená, že se růst zpomaluje a dochází k poklesu

V časových řadách těchto ukazatelů se projevuje bod zvrátání s určitým časovým předstihem před jeho výskytem v referenční řadě. Vedle předstihových ukazatelů jsou z hlediska načasování (ve vztahu ke skutečnému bodu zvrátání) rozlišovány ještě souběžné a zaostávající ukazatele. Analytické systémy, určené pro signalizaci bodů zvrátání v hospodářském cyklu, spojují vybrané ukazatele rozdělené podle načasování do skupin ve formě souhrnných (kompozitních) ukazatelů (indexů). V USA se vypočítávají a měsíčně publikují kompozitní indexy, jenž jsou konstruované ze tří skupin ukazatelů rozlišených podle načasování bodu zvrátání v jejich časových řadách:

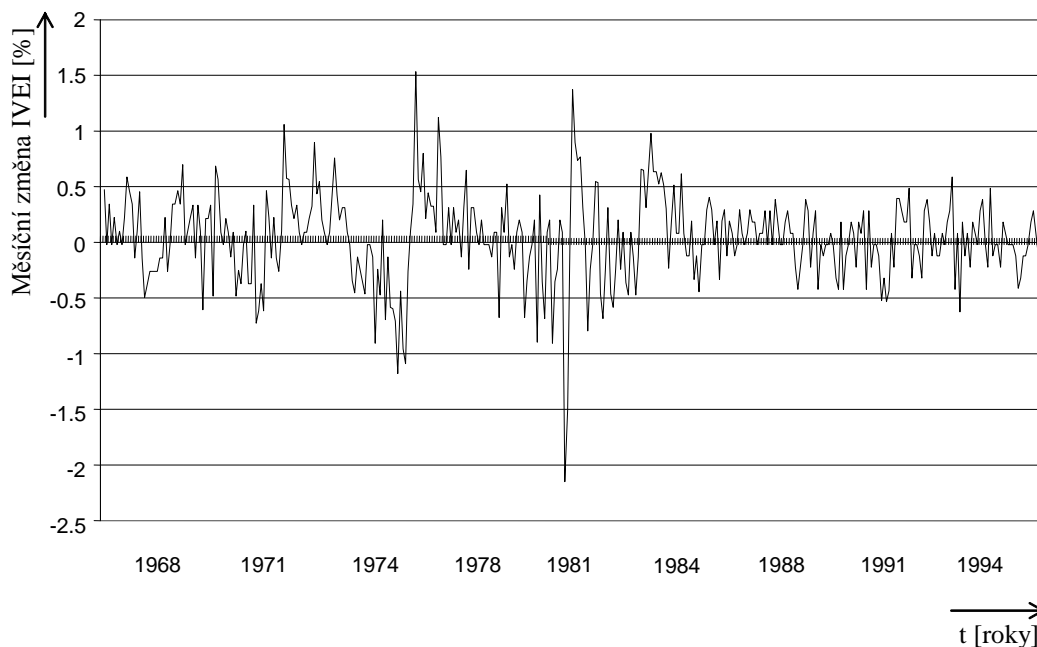
- index vedoucích ekonomických indikátorů,
- index koincidence,
- index zaostávajících indikátorů.

1.3.1 Index vedoucích ekonomických indikátorů

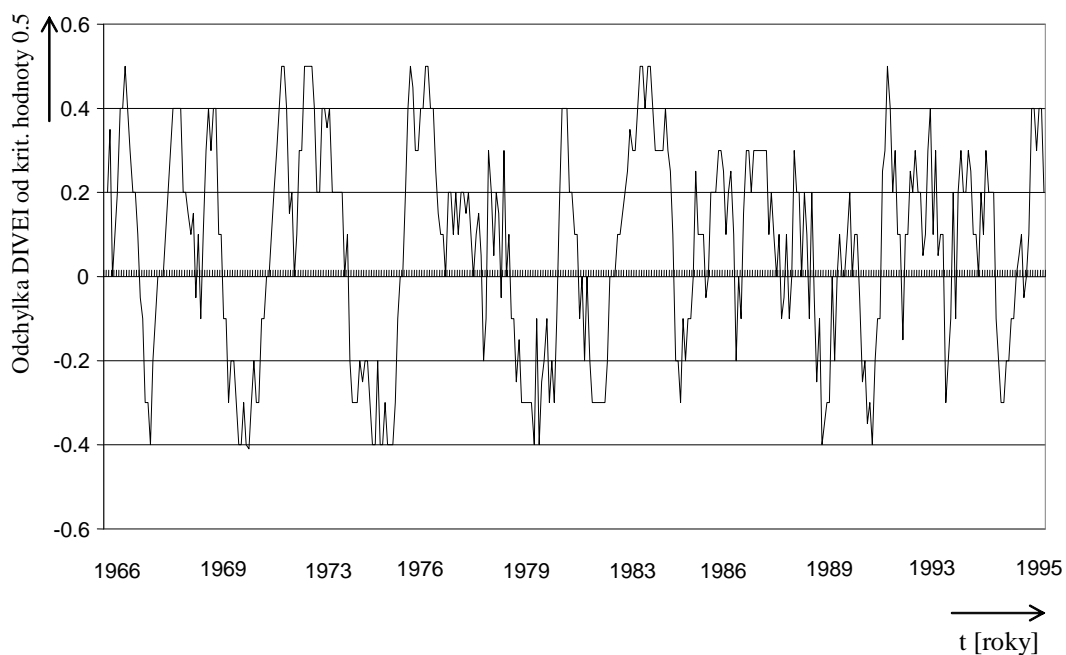
Index vedoucích ekonomických indikátorů (IVEI) je zařazen do skupiny agregátních ukazatelů a je nedefinován [4] jako vážený aritmetický průměr 10 ekonomických proměnných, o kterých je známo, že vedou ekonomický cyklus a že jsou předstihové. Z toho plyne, že se jejich změny v trendu objevují před změnami v trendu agregátní ekonomické aktivity a to ve stejném směru. Těmito ekonomickými proměnnými jsou:

- průměrný týdenní počet odpracovaných hodin ve zpracovatelském průmyslu,
- průměrný týdenní počet nových nároků na čerpání pojištění v nezaměstnanosti,
- hodnota nových objednávek zpracovatelů na spotřební produkty a materiály,
- rychlost dodavatelů (podíl zpracovatelů hlásících zpomalení dodávek),
- nové objednávky zpracovatelů na civilní kapitálové statky,
- počet stavebních povolení na nové soukromé bytové jednotky,
- ceny akcií podle indexu Standard's & Poors 500 Common Stocks,
- peněžní zásoba,
- rozpětí úrokových sazeb (rozdíl mezi krátkodobými a dlouhodobými úrokovými sazbami),
- index spotřebitelských očekávání.

K IVEI je také definován difúzní index vedoucích ekonomických indikátorů (DIVEI), který zjišťuje jaké procento z ekonomických proměnných obsažených v IVEI klesá a jaké roste. Ukazuje tedy, jaká část ekonomických proměnných naznačuje změnu trendu ekonomického růstu. Vývoj uvedených indexů ekonomiky USA v letech 1965 až 1995 je znázorněn na obr. 1.3 a obr. 1.4.



Obr. 1.3 Index vedoucích ekonomických indikátorů (1965 – 1995)



Obr. 1.4 Difúzní index vedoucích ekonomických indikátorů (1965 – 1995)

Oba výše zmíněné indexy je možno použít na predikce vývoje HDP. Při použití těchto dvou indexů vzniká spolehlivý prediktor kontrakce. Předpokládá se, že kontrakce nastane, když průměrná míra růstu IVEI klesne pod 2% a zároveň hodnota DIVEI klesne pod kritickou hodnotu 0.5. Problematická je predikční schopnost těchto indexů v letech 1981, 1982 a 1990, kdy došlo k jasnému signálu recese až na začátku samotné recese, a v roce 1966, kdy došlo k falešnému signálu recese. Do současné doby se tyto indexy nepoužívaly pro predikci vývoje HDP.

1.3.2 Index koincidence a index zaostávajících indikátorů

Index koincidence (Coincident Index) je souběžný indikátor. Skládá se ze čtyř ekonomických proměnných. Jsou jimi:

- počet pracovníků v nezemědělských sektorech – ve fyzických osobách,
- čistý osobní důchod (bez transferových plateb),
- index průmyslové produkce (pokrývá fyzický výstup všech fází produkce), tržby ve zpracovatelském průmyslu a obchodě.

Index zaostávajících indikátorů (Lagging Index) se skládá ze sedmi ekonomických proměnných. Jsou to:

- průměrná délka trvání nezaměstnanosti,
- podíl zásob ve zpracovatelském průmyslu a obchodě na tržbách,
- změna nákladů práce na jednotku výstupu ve zpracovatelském průmyslu,
- průměrná primární bankovní sazba,
- objem půjček poskytnutých bankami a nefinančními institucemi,
- podíl spotřebitelských úvěrů na osobním důchodu,
- změna indexu spotřebitelských cen služeb.

1.4 Shrnutí kapitoly

V kapitole je uvedeno zavedení základních ekonomických pojmů. Byl definován pojem HDP a popsán rozdíl mezi reálným a nominálním HDP. Dále se v kapitole uvádí tři metody jeho výpočtu – výrobní, výdajová a důchodová. Následně byl popsán cyklický vývoj ekonomiky, ekonomický cyklus, jeho jednotlivé fáze a rozdělení cyklů podle délky

trvání. Závěrečná část kapitoly byla věnována ekonomickým ukazatelům s ohledem na jejich souvislost s vývojem a predikcí HDP USA. Detailně byli popsáni předstihoví ukazatelé IVEI a DIVEI.

2 Modely na predikci vývoje HDP

V kapitole jsou v první části definovány metody tvorby modelu predikce vývoje časových řad. V druhé části kapitoly je popsáno určení míry přesnosti modelů predikce vývoje HDP. V poslední části jsou obsaženy funkční modely predikce vývoje HDP USA.

2.1 Metody predikce vývoje HDP

Pomocí dnešní ekonometrie se může modelovat a testovat velké množství modelů reálných vztahů. Nejčastěji se v oblasti predikce ekonomických proměnných používají modely založené na těchto metodách [5].

Metoda extrapolace

Metoda extrapolace vychází z předpokladu, že vývoj v budoucnosti bude analogický s vývojem v minulosti. Do této skupiny patří především statistické metody, jako jsou metody zpracování časových řad, regresní a korelační analýza, analýza a prognóza časových řad.

Ekonometrické modely

Ekonometrické modely jsou založené na soustavě rovnic, které vyjadřují základní vztahy v modelovaném systému. Každý model je však určitým zjednodušením skutečnosti a jeho naplnění potřebnými informacemi a parametry, odrážejícími vzájemné vztahy mezi jednotlivými veličinami, je vždy obtížné.

Metoda scénářů

Spočívá v simulaci možných budoucích situací, kterým by se měl přizpůsobit vývoj modelovaného systému. Scénář by měl uspořádat posloupnost událostí v čase a zachovat jejich logickou vzájemnou návaznost.

Reflexivní a intuitivní metody

Opírají se o subjektivní názory jednotlivců nebo skupin a vycházejí ze zkušeností a poznatků pracovníků dané oblasti. Mívají charakter hypotéz, které se v další predikční činnosti ověřují.

Expertní metody

Jsou to metody založené na využití znalostí a zkušeností odborníků v dané oblasti a jejich schopnosti předvídat možný budoucí vývoj, jeho souvislosti a důsledky.

2.2 Zjišťování míry přesnosti modelů pro predikce vývoje HDP

Existující ekonometrické modely predikce vývoje HDP používají různé formulace vztahů a různé metody odhadů parametrů specifických modelů. Tato skutečnost způsobuje různou míru jejich přesnosti. Za nejpřesnější se všeobecně považují modely vytvořené národními (Centrální banka, Statistické úřady) nebo nadnárodními společnostmi. Přesnost predikce [4] pro všechny typy modelů lze měřit pomocí:

- střední chyby (Average Error, AVE)

$$AVE = \frac{\sum_{n=1}^N (Y_n - Y_{np})}{N}, \quad (2.1)$$

- střední absolutní chyby (Mean Absolute Error, MAE)

$$MAE = \frac{\sum_{n=1}^N |Y_n - Y_{np}|}{N}, \quad (2.2)$$

- střední kvadratické odchyly (Root Mean Squared Error, RMSE)

$$RMSE = \sqrt{\frac{\sum_{n=1}^N (Y_n - Y_{np})^2}{N}}, \quad (2.3)$$

kde: - Y_n je skutečná hodnota růstu,
- Y_{np} je predikovaná hodnota růstu,
- N je počet predikovaných období.

2.3 Modely pro predikci vývoje HDP

V běžné ekonomické praxi se vyskytují modely pro predikci makroekonomických veličin, které vytvářejí významné národní či globální instituce. Prvním ze zmiňovaných je model predikce OECD [4], druhým dále model Federální rezervní banky San Francisco [4].

Model OECD

Příklady ukazatelů přesnosti predikce modelu OECD pro vývoj HDP americké ekonomiky v letech 1980 až 1995 udává Tabulka 2.1. Průměrné chyby jsou definované vztahy (2.1), (2.2), (2.3). Tyto statistiky slouží jako objektivní reference přesnosti nejkvalitnějších existujících modelů predikce HDP a to především z následujících důvodů:

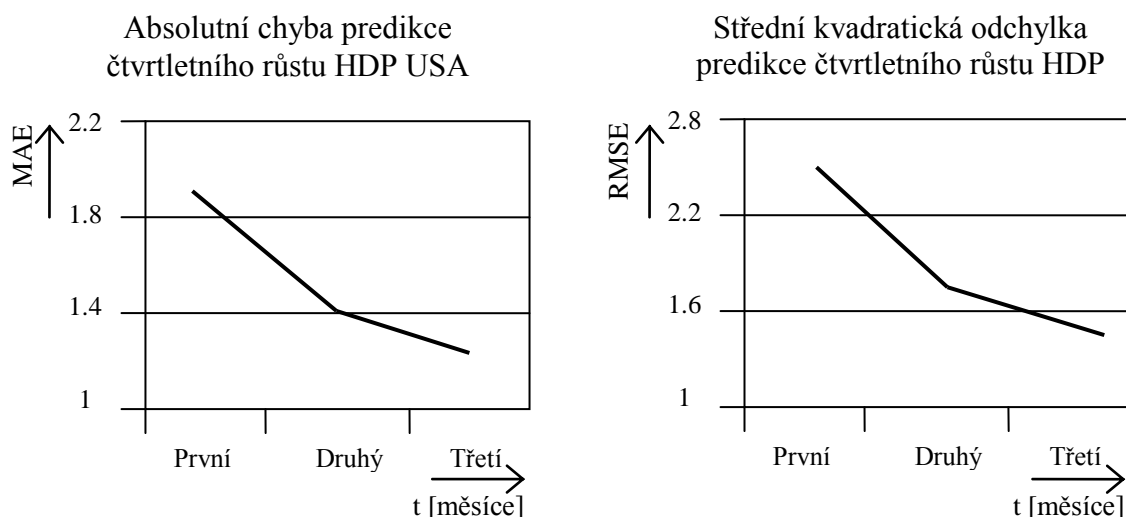
- predikce OECD tvoří konzistentní časovou řadu, umožňující objektivní komparaci přesnosti modelu v čase. Alternativní oficiální statistiky nejsou konzistentní z důvodu změn formy modelu, momentu predikce, apod.,
- komparativní studie uvádějí, že predikované hodnoty modelu OECD jsou co do velikosti i směru porovnatelné s predikcemi podobných institucí.

Tabulka 2.1 Ukazatelé chyby predikce roční míry růstu reálného HDP USA, průměry let 1980 – 95, model OECD

Moment predikce/chyba predikce	6 měsíců před predikovaným obdobím	12 měsíců před predikovaným obdobím
AVE	-0.8	0.23
MAE	0.58	1.05
RMSE	0.72	1.21

Model Federální rezervní banky San Francisco

Výsledný predikovaný vývoj HDP je v modelu Federální rezervní banky San Francisco určován jako řetězec vážených ukazatelů, které mají vliv na budoucí vývoj HDP. Přesnost tohoto modelu predikce čtvrtletního růstu HDP USA uvádí obr. 2.1.



Obr. 2.1 Absolutní chyba a střední kvadratická odchylka predikce analyzovaného čtvrtletního růstu HDP USA, průměry let 1985 – 95, model Federální rezervní banky San Francisco

Jsou to výše zaměstnanosti, průmyslová výroba, skutečné maloobchodní prodeje a skutečné HDP za tři předchozí období. Lze konstatovat, že tento model dosahuje v porovnání s modelem OECD nižší přesnosti. Chyba predikce analyzovaného čtvrtletního růstu, bývá vyšší než chyba predikce ročního růstu, protože predikovaných hodnot je 4krát více. Dalším možným zdrojem výše nepřesnosti je nižší komplexnost struktury modelu. Nadefinování ukazatelů chyb je uvedené v kapitole 2.2.

2.4 Shrnutí kapitoly

V kapitole byly uvedeny základní metody pro tvorbu modelů predikce HDP, popsána byla metoda extrapolace, ekonometrické modely, metoda scénářů a další využívané metody. Dále byly zmíněny způsoby určování míry přesnosti modelů predikce vývoje HDP s detailním popisem jednotlivých typů chyb. V závěru kapitoly byly uvedeny modely pro predikci HDP v USA, konkrétně model OECD a model Federální rezervní banky San Francisco.

3 Časové řady a jejich předzpracování

V kapitole jsou definovány základní pojmy z časových řad, jejich vlastnosti a způsoby dělení. Dále je uveden způsob dekompozice časových řad a jsou zde definovány jejich jednotlivé složky. Ve třetí části se přistupuje k vyrovnání trendové složky časové řady. Jsou zde podrobně uvedeny metody, kterými lze tohoto vyrovnání dosáhnout, způsob jejich použití a jejich zhodnocení.

Dělení časových řad

Při členění časových řad [10] ekonomických ukazatelů nejde pouze o definiční vymezení druhů časových řad, ale především o vyjádření rozdílností v obsahu sledovaných ukazatelů, jež je mnohdy provázeno i specifickými statistickými vlastnostmi. V důsledku toho je pak nutnost volit diferencovaně i prostředky analýzy sloužící k porozumění mechanismu, kterým je vývoj sledovaného jevu utvářen. Základní druhy časových řad ekonomických ukazatelů se rozlišují podle:

- Rozhodného časového hlediska na časové řady **intervalové** (tj. časové řady intervalových ukazatelů) a na časové řady **okamžikové** (tj. časové řady okamžikových ukazatelů).
- Periodicity, s jakou jsou údaje v řadách sledovány, na časové řady **roční** (někdy též dlouhodobé) a na časové řady **krátkodobé**, kde jsou údaje zaznamenávány ve čtvrtletních, měsíčních, týdenních aj. periodách.
- Druhu sledovaných ukazatelů na časové řady **primárních** (prvotních) ukazatelů a na časové řady **sekundárních** (odvozených) charakteristik.
- Způsobu vyjádření údajů na časové řady **naturálních** ukazatelů (hodnoty ukazatele jsou vyjadřovány v naturálních jednotkách) a na časové řady **peněžních** ukazatelů.

3.1 Dekompozice časové řady

Cílem dekompozice časové řady je snaha ji rozložit na několik základních složek. Tento rozklad se provádí za účelem snazší identifikace chování jednotlivých složek. Časovou řadu tedy tvoří

- trendová složka T_t ,
- sezónní složka S_t ,

- cyklická složka C_t ,
- náhodná složka ε_t .

Vlastní rozklad má tvar

$$y_t = T_t + S_t + C_t + \varepsilon_t = Y_t + \varepsilon_t, \quad t = 1, 2, \dots, n, \quad (3.1)$$

kde se Y_t často značí jako teoretická složka ve tvaru $T_t + S_t + C_t$ a náhodná složka ε_t je opomíjena.

Trendová složka

Trendová složka (T_t) odpovídá hlavním tendencím [10] dlouhodobého vývoje statistického ukazatele, který časová řada popisuje. Je možné si představit, že trendová složka vzniká v důsledku působení sil, které systematicky působí ve stejném směru. Trend může být rostoucí, klesající nebo konstantní.

Sezónní složka

Sezónní složka je pravidelně se opakující odchylka od trendové složky, vyskytující se u časových řad s periodicitou kratší než jeden rok. Sezónní změny jsou hlavně způsobeny takovými faktory, jako je střídání ročních dob a lidské zvyky zakotvené institucionálně v ekonomické aktivitě. Nejčastěji lze pozorovat sezónnost u čtvrtletních a měsíčních časových řad.

Cyklická složka

Cyklická složka představuje kolísání okolo trendu v důsledku dlouhodobého cyklického vývoje s délkou vlny delší než jeden rok. Statistika chápe cyklus jako dlouhodobé kolísání s neznámou periodou, která může mít i jiné příčiny než klasický ekonomický cyklus. V této souvislosti se mluví např. o cyklech demografických, inovačních, apod. Někdy nebývá cyklická složka považována za samostatnou složku časové řady, ale je zahrnována pod složku trendovou jako její část (tzv. střednědobý trend), vyjadřující střednědobou tendenci vývoje, která má často oscilační charakter s neznámou, zpravidla proměnlivou periodou.

Náhodná složka

Náhodná složka je taková veličina, kterou nelze popsat žádnou funkcí času. Je to složka, která zbývá po vyloučení trendu, sezónní a cyklické složky. Je tvořena náhodnými

pohyby (fluktuacemi) v průběhu časové řady, které nemají rozpoznatelný systematický charakter. Proto se již také nepočítá mezi předchozí tzv. teoretické složky časové řady. Náhodná složka rovněž pokrývá chyby v měření údajů časové řady a některé chyby (např. zaokrouhlování), kterých se lze dopustit při vlastní analýze řady. Aby byly odůvodněny některé statistické postupy, které se s časovou řadou při klasické dekompozici provádějí, předpokládá se obvykle, že náhodná složka je bílý šum, a někdy dokonce, že se jedná o bílý šum s normálním rozdělením.

3.1.1 Metoda klouzavých průměrů

Podstata vyrovnání [6] pomocí klouzavých průměrů spočívá v tom, že posloupnost empirických pozorování se nahradí řadou průměrů vypočítaných z těchto pozorování. Každý z těchto průměrů reprezentuje určitou skupinu pozorování. Název klouzavý průměr vznikl z toho, že se při postupném výpočtu průměrů postupuje (klouže) vždy o jedno pozorování dopředu, přičemž se poslední pozorování ze skupiny, z níž je průměr počítán, vypouští. Cílem vyrovnání časové řady je potlačení šumových složek. Velmi důležité je stanovení počtu pozorování, ze kterých se jednotlivé klouzavé průměry počítají. Tento počet se nazývá klouzavá část období interpolace a značí se

$$p = 2m + 1, p < n, \quad (3.2)$$

kde n je počet pozorování analyzované řady.

Volba hodnoty p (délky klouzavého průměru) je zásadní a často dost problematická. S rostoucí délkou klouzavého průměru roste stupeň vyhlazení časové řady, ale na druhé straně hrozí nebezpečí, že se potlačí jako šum nebo zdeformují některé důležité systematické složky původního záznamu.

Jednoduché klouzavé průměry

Jednoduché klouzavé průměry jsou vlastně klasické aritmetické průměry z původních hodnot časové řady většinou liché délky $2m+1$. Obecně lze jednoduchý klouzavý průměr zapsat následujícím způsobem

$$y_{t,n} = \frac{1}{2m+1} (y_{t-m} + y_{t-m+1} + \dots + y_t + \dots + y_{t+m-1} + y_{t+m}) \quad (3.3)$$

Pro jednoduchý klouzavý průměr například délky 7 má uvedený vztah tento tvar

$$y_{t,7} = \frac{1}{7} (y_{t-3} + y_{t-2} + y_{t-1} + y_t + y_{t+1} + y_{t+2} + y_{t+3}) \quad (3.4)$$

- kde: - $y_{t,n}$ je hodnota průměru v čase t za časové okno n ,
 - n je délka okna,
 - y_t je získaná hodnota v bodě t .

Vážené klouzavé průměry

Vážené klouzavé průměry [7] zobecňují jednoduché klouzavé průměry tím, že průměrovaným hodnotám $y_{t-m}, y_{t-m+1}, \dots, y_t, \dots, y_{t+m}$ nedávají stejné váhy $1/(2m+1)$, ale váhy vhodným způsobem symetricky klesají od středu k oběma okrajům. Vážené klouzavé průměry vyrovnávají hodnotu časové řady v bodě t váženým průměrem. Vážený klouzavý průměr se vypočítá například pro $n = 7$ podle vztahu

$$y_{t,7} = \frac{1}{21} (2y_{t-3} + 3y_{t-2} + 6y_{t-1} + 7y_t + 6y_{t+1} + 3y_{t+2} + 2y_{t+3}) \quad (3.5)$$

Konstrukce jednotlivých vah je často založena na prokládání pozorování vhodnými křivkami pomocí metody nejmenších čtverců a jako vhodné křivky se nejčastěji volí polynommické křivky. Pro váhy platí, že jsou symetrické kolem střední hodnoty a součet vah je roven jedné.

Trojúhelníkové klouzavé průměry

Trojúhelníkový klouzavý průměr [7] je dvakrát za sebou provedený jednoduchý klouzavý průměr. Největší váhu mají tedy data ležící uprostřed periody délky n . Směrem do minulosti i do přítomnosti váha dat lineárně klesá. Například, trojúhelníkový klouzavý průměr pro $n = 5$ se vypočítá podle vztahu

$$y_{t,5} = \frac{1}{9} (y_{t-2} + 2y_{t-1} + 3y_t + 2y_{t+1} + y_{t+2}) \quad (3.6)$$

V tabulce 3.1 je uvedený přehled vah pro různé délky klouzavých průměrů a různé řády polynomů.

Tabulka 3.1 Přehled vah pro různé délky klouzavých průměrů a různé řády polynomů

délka\řád	2. a 3.	4. a 5.
3	(0,1,0)	(0,1,0)
5	$\frac{1}{35} (-3,12,17,\dots)$	(0,0,1,...)
7	$\frac{1}{21} (-2,3,6,7,\dots)$	$\frac{1}{231} (5,-30,75,131,\dots)$
9	$\frac{1}{231} (-21,14,39,54,59,\dots)$	$\frac{1}{429} (15,-55,30,135,179,\dots)$
11	$\frac{1}{429} (-36,9,44,69,84,89,\dots)$	$\frac{1}{429} (18,-45,-10,60,120,143,\dots)$

Centrované klouzavé průměry

V ekonomické praxi [7] je často přirozené průměrovat přes sudý počet pozorování. Centrované klouzavé průměry jsou vážené průměry se speciálně zvolenými váhami tak, aby eliminovaly z časové řady sezónní složku. Oproti předcházejícím klouzavým průměrům, které používají liché p , centrované klouzavé průměry používají sudé p tj. $p = 2m$. Centrovaný klouzavý průměr lze vypočítat buď jako průměr ze dvou sousedních klouzavých průměrů, nebo centrované klouzavé průměry pro $2m$ členů jako vážené klouzavé průměry pro $2m+1$ členů, kdy první a poslední člen se počítá s vahou 1, ostatní s vahou 2 a se součtem vah $4m$, tedy

$$y_t = \frac{y_{t-m} + 2y_{t-m+1} + \dots + 2y_t + \dots + 2y_{t+m-1} + y_{t+m}}{4m}. \quad (3.7)$$

3.1.2 Exponenciální vyrovnávání

Exponenciální vyrovnávání [8] je dalším adaptivním přístupem k trendové složce. V případě klouzavých průměru je snaha vyrovnávat v časové řadě polynomickými křivkami krátké úseky, jejichž vhodná délka je často problematická a určená subjektivně. Metoda exponenciálního vyrovnávání [9] je založena na všech předchozích pozorováních, přičemž jejich váha (w) směrem do minulosti klesá podle exponenciální funkce

$$w_t = (1 - \alpha) \cdot \alpha^{n-t}, \quad (3.8)$$

kde n je počet pozorování a α je vyrovnávací konstanta v intervalu (0;1).

K velmi ceněným vlastnostem těchto metod patří jejich rekurentnost. Při dodání nového pozorování lze jeho vyrovnanou hodnotu zkonstruovat snadnou korekcí předchozí vyrovnané hodnoty.

Jednoduché exponenciální vyrovnávání

Při jednoduchém exponenciálním vyrovnávání se předpokládá, že trend se v krátkém období vyvíjí zhruba konstantně. U časových řad, které jsou ve tvaru $y_t = Tr_t + E_t$, lze v případě konstantního trendu nahradit trendovou složku (Tr) konstantou $Tr_t = \beta_0$. Je třeba nalézt odhad parametru β_0 , který se v tomto případě rovná vyrovnané hodnotě y . Vyrovnaná časová řada se vypočítá podle následujícího rekurentního vztahu

$$y_t = (1 - \alpha) \hat{y}_t + \alpha \cdot y_{t-1}. \quad (3.9)$$

Pokud se α blíží k hodnotě 1, tak roste vliv minulých pozorování. Výše uvedený vztah lze přepsat i do následujícího tvaru [9]

$$y_t = y_{t-1} + (1 - \alpha) (\hat{y}_t - y_{t-1}), \quad (3.10)$$

který vysvětluje vytváření nové vyrovnané hodnoty z předchozí vyrovnané hodnoty, opravenou o chybu danou rozdílem mezi skutečnou a předcházející vyrovnanou hodnotou.

Dvojitě exponenciální vyrovnávání

Dvojitě exponenciální vyrovnávání se používá v případě, kdy lze předpokládat, že v krátkém období bude mít trendová složka lineární formu, tj. $Tr_t = \beta_0 + \beta_1 t$. Parametry β_0 a β_1 se odhadují na základě minimalizace vztahu

$$S^2 = \sum_{\tau=1}^{t-1} \alpha^\tau (\hat{y}_{t-\tau} - \beta_0 - \beta_1 \tau)^2. \quad (3.11)$$

Robustní klouzavé průměry

Zvláštní skupinu představují tzv. robustní klouzavé průměry [7], které slouží k potlačení odlehlých pozorování. Jejich přítomnost v časové řadě může značně zkreslit výsledky klasických metod, tj. i výsledky předchozích klouzavých průměrů. Nejznámějším zástupcem této skupiny průměrů je klouzavý medián, který se vypočítá podle vztahu

$$y_t = \text{med}(y_{t-m}, y_{t-m+1}, \dots, y_t, \dots, y_{t+m}). \quad (3.12)$$

Med (medián) vrací prostřední hodnotu z hodnot, které jsou uspořádány do neklesající posloupnosti.

3.2 Shrnutí kapitoly

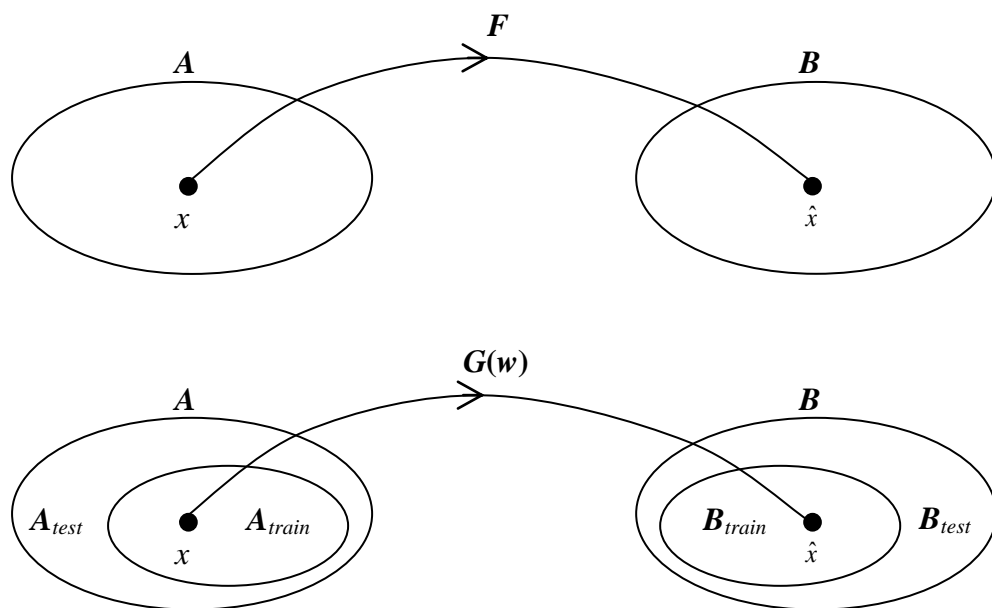
V úvodu kapitoly byla definována časová řada a způsob dělení časových řad ekonomických ukazatelů. Dále byla vysvětlena dekompozice časové řady na jednotlivé její složky. V závěrečné části kapitoly byly uvedeny adaptivní přístupy k modelování trendové složky časové řady. Byly zde podrobně ukázány metody klouzavých průměrů, exponenciálních vyrovnání a klouzavého mediánu.

4 Neuronové sítě

V kapitole jsou uvedeny základní pojmy z teorie neuronových sítí a popsán jejich historický vývoj. Dále jsou uvedeny různé typy vstupů, typy aktivačních funkcí neuronů, struktura neuronové sítě a popsán proces učení.

4.1 Neuronové sítě a umělá inteligence

Neuronová síť [11] je systém sestávající se z výpočetních jednotek – neuronů, které jsou mezi sebou vzájemně propojeny synapsemi ohodnocenými váhami. Schopností adaptovat tyto váhy – učit se – na základě trénovacích vzorů umožňují realizovat kvalitativně novou funkci implicitně obsaženou v trénovacích datech. Důležitou vlastností neuronových sítí je kromě schopnosti učit se, tedy nacházet závislosti v trénovacích datech a ty reprezentovat pomocí vah, také schopnost zevšeobecnování (generalizace) získaných poznatků, tedy schopnost správně reagovat i na neznámé vstupy, na které nebyla neuronová síť naučena. Jinak je toto možno popsat tímto způsobem. Nechť je dána nějaká všeobecná formulace predikčního problému pomocí zobrazení – funkce definované nad dvěma množinami A a B . Potom tento přístup bude užitečný pro interpretaci neuronových sítí jako prediktoru tak i klasifikátoru. Nechť $F(x)$ je funkce definovaná nad množinou A , která přiřadí každému prvku $x \in A$ funkční hodnotu z množiny B , $\hat{x} = F(x) \in B$, jak je uvedeno na obr. 4.1,



Obr. 4.1 Schematické znázornění zobrazení $F : A \rightarrow B$

$$F : A \rightarrow B . \quad (4.1)$$

Dále platí $G(x,w)$ je funkce, jejíž argumenty jsou z konečné podmnožiny $A_{train} = \{x_1, x_2, \dots, x_r\} \subset A$ (nazývané trénovací množina) a w je parametr (nebo parametry) zobrazení G , potom $x = G(x, w) \in B_{train} \subset B$ a

$$G(w) : A_{train} \rightarrow B_{train} . \quad (4.2)$$

Potom je možné říci, že zobrazení $G(w)$ je restrikce zobrazení $F(x)$ nad množinou $A_{train} \subset A$. Komplement A_{train} vzhledem k množině A je označený A_{test} (a nazývá se testovací množina), $A_{test} = A/A_{train} = \{a \in A/a \notin A_{train}\}$. Necht' pro každé $x_i \in A_{train}$ je známa požadovaná funkční hodnota x_i . Zúžením zobrazení $F : A \rightarrow B$ na podmnožinu A_{train} se získá nové modelové zobrazení $G(w)$. Funkční tvar tohoto zobrazení je určený parametrem (parametry) w . Požadované funkční hodnoty x_i jsou interpretované jako obrazy funkce F ve tvaru

$$x_i = F(x_i) \quad (i = 1, 2, \dots, r). \quad (4.3)$$

Cílem úvah je najít takový parametr (nebo parametry) w funkce $G(x,w)$, aby funkční hodnoty argumentů z trénovací množiny A_{train} byly co nejbližším obrazem funkce $F(x)$ (tj. požadovaným hodnotám). Necht' je definována účelová funkce

$$E(w) = \frac{1}{2} \sum_{i=1}^r \|G(x_i, w) - x_i\|^2. \quad (4.4)$$

Tato funkce vyjadřuje sumu čtverců odchylek funkce $G(x,w)$ od požadovaných hodnot x braných z trénovací množiny. Požadavek, aby vypočtené hodnoty $G(x,w)$ byly co nejbližší požadovaným hodnotám \hat{x} je realizován pomocí požadavku minimálnosti účelové funkce $E(w)$ vzhledem k parametru w . Funkce $G(x,w)$ je adaptovaná, když její parametr w je vybrán tak, aby se rovnal své optimální hodnotě (tj. ve kterém má účelová funkce globální minimum).

4.2 Definice neuronové sítě

Paradigma neuronové sítě bude formulováno pomocí grafovo-teoretického přístupu, který vychází z analogie s lidským mozkiem a koncept neuronové sítě bude použitý na konstrukci modelové funkce $G(x,w)$. Formálně je neuronová síť určena jako orientovaný

graf $G = (V, E)$. Výrazy $V = \{v_1, v_2, \dots, v_N\}$ a $E = \{e_1, e_2, \dots, e_M\}$ označují neprázdnou vrcholovou množinu, resp. hranovou množinu grafu G obsahujícího N vrcholů (neuronů) a M hran (spojů). Každý spoj $e \in E$ se interpretuje jako uspořádaná dvojice dvou neuronů z množiny V , $e = (v, v')$. Spoj e začíná v neuronu v a končí v neuronu v' . Množina V je rozložena na disjunktí podmnožiny následujícím způsobem

$$V = V_I \cup V_H \cup V_O, \quad (4.5)$$

kde V_I obsahuje N_I vstupních neuronů, které sousedí jen s vycházejícími hranami V_H , obsahuje N_H skrytých neuronů, které sousedí současně s vycházejícími jako s vcházejícími hranami a V_O obsahuje N_O výstupních neuronů, které sousedí jen s vcházejícími hranami. V následujících úvahách se bude vždy předpokládat, že množiny V_I a V_O jsou neprázdné, tj. neuronová síť obsahuje vždy alespoň jeden vstupní a jeden výstupní neuron.

Vstupy neuronové sítě

Přestože neuronové sítě jsou schopny rozpoznat složité strukturální obrazce, dá se říci, že vstupy neuronů jsou číselné a nejčastěji se lze setkat s následujícími typy vstupů:

- kvalitativní vstupy – nabývají hodnot z diskrétních číselných množin např. $\{-1, +1\}$,
- fuzzy kvalitativní vstupy – jako fuzzy hodnoty jazykových proměnných zavedené na universech systémových proměnných,
- kvantitativní vstupy – obvyklé číselné reprezentace systémových proměnných.

Model neuronu

Neuron je vybaven konečným počtem vstupů x_j , aktivační funkcí a jedním výstupem y_i (ten je množitelný do potřebného počtu kopií). Obecně neuron lze popsat podle tohoto vztahu

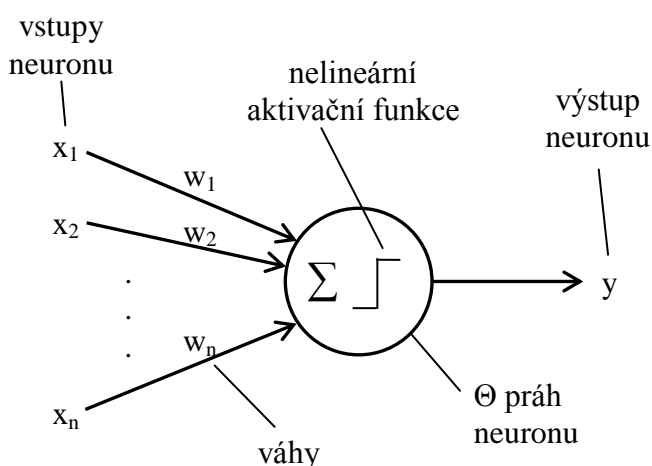
$$\xi_i = \sum_{j \in \Gamma_i^{-1}} w_{ij} x_j + \Theta_i, \quad (4.6)$$

kde: - w_{ij} jsou váhy synapsí,

- x_j jsou vstupy neuronu,

- f je neuronová aktivační funkce,
- Θ je práh neuronu.

Tento vztah lze jednoduše zobrazit pomocí nelineárního modelu neuronu, který je patrný z obr. 4.2. Vztah (4.6) bývá často označován jako vnitřní potenciál neuronu. Váhy w_i jednotlivých neuronů představují jeho lokální paměť a spojením všech neuronů se získá celková paměť neuronové sítě. Učení a jeho optimalizace se provádí změnou těchto vah, tvarováním aktivační funkce, změnou počtu neuronů a strukturálním uspořádáním sítě.



Obr. 4.2 Nelineární model neuronu

Aktivační funkce

Aktivační funkce transformuje hodnotu potenciálu neuronu na číslo z určitého intervalu. Nejčastěji je obor výstupních hodnot omezen intervalem $\langle 0,1 \rangle$, nebo $\langle -1,1 \rangle$. Tato přechodová (aktivační) funkce $t(\xi)$ je z pravé strany je monotónně rostoucí a vyhovuje následujícím asymptotickým podmínkám: $t(\xi) \rightarrow A$, pro $\xi \rightarrow -\infty$ a $t(\xi) \rightarrow B$, pro $\xi \rightarrow \infty$, kde $-\infty < A < B < \infty$. V teorii neuronových sítí se často využívá následující sigmoidální funkce

$$t(\xi) = \frac{B + Ae^{-\xi}}{1 + e^{-\xi}}, \quad (4.7)$$

s první derivací určenou ve tvaru

$$t' \in \left] \frac{A+t}{A+B}, \frac{B-t}{A+B} \right[. \quad (4.8)$$

Tato přechodová funkce zobrazuje celou množinu reálných čísel R na otevřený interval (A,B) , formálně $t: R \rightarrow (A,B)$. Nejčastěji se přechodová funkce využívá pro hodnoty parametrů $A = 0, B = 1$ nebo $A = -1, B = 1$. Aktivity neuronů tvoří vektor $x = (x_1, x_2, \dots, x_N)$. Tento vektor je formálně rozložitelný na tři podvektory obsahující vstupní, skryté a výstupní aktivity

$$x = x_I \oplus x_H \oplus x_O. \quad (4.9)$$

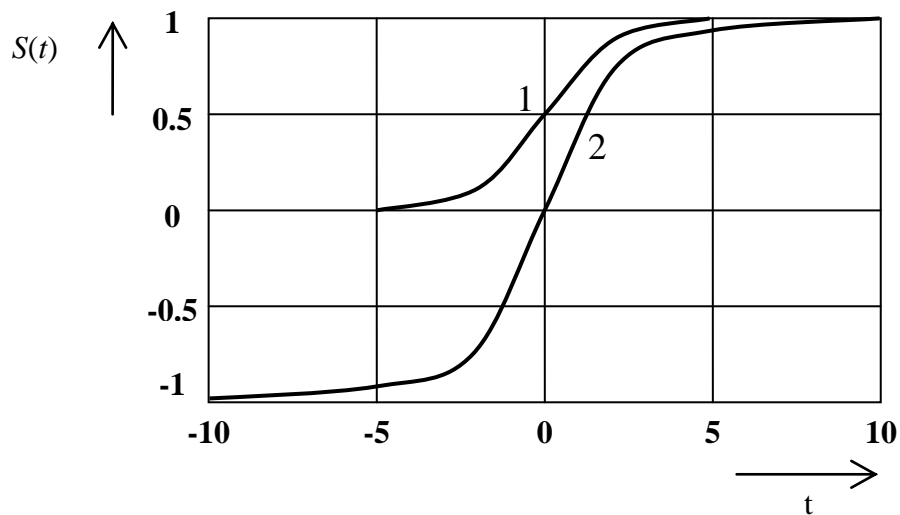
Neuronovou síť s fixovanými váhami a prahovanými koeficienty možno formálně chápat jako funkci

$$G: R^{N_I} \rightarrow (A, B)^{N_O}. \quad (4.10)$$

Tato funkce G přiřadí vstupní aktivitě x_I (deskriptor) výstupní vektor x_O s hodnotami svých složek z otevřeného intervalu (A,B) ve tvaru

$$G(x_I) = x_O. \quad (4.11)$$

Na obr. 4.3 je znázorněn průběh aktivační funkce, kde t je potenciál neuronu, $S(t)$ je transformovaná hodnota. Křivka 1 na obr. 4.3 odpovídá standardní sigmoidální funkci ($A = 0, B = 1$), křivka 2 znázorňuje funkci hyperbolický tangens ($A = -1, B = 1$).

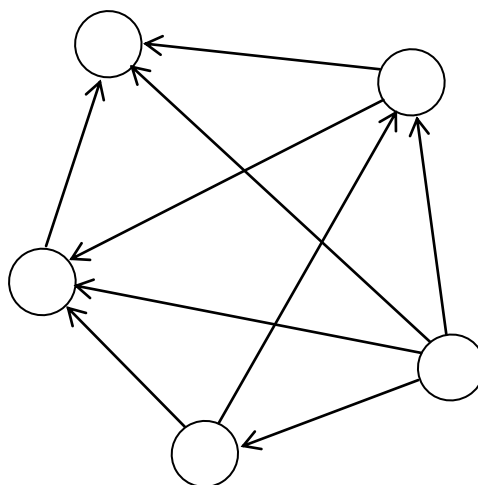


Obr. 4.3 Průběh aktivační funkce

Skryté aktivity nejsou explicitně uvedené, hrají jen úlohu mezivýsledků. Vstupní aktivity jsou určeny deskriptorem, proto jsou pokládány za fixované. Aktivity skrytých neuronů z druhé vrstvy L_2 je možno spočítat jen s použitím vstupních aktivit z vrstvy L_1 . Ve všeobecnosti pro výpočet aktivit vrstvy L_i (kde $i > 1$) je nutné poznat jen aktivity z nižších vrstev L_1, L_2, \dots, L_{i-1} . Tímto rekurentním způsobem se postupně spočítají aktivity všech neuronů. Jako poslední se počítají aktivity výstupních neuronů. Díky tomu se pro neuronové sítě reprezentované acyklickým grafem zažil název *neuronové sítě s dopředným šířením*. Tento jednoduchý postup výpočtu aktivit neuronů je aplikovatelný jen na neuronové sítě reprezentované acyklickým orientovaným grafem. V případě, že graf obsahuje orientované cykly, tento postup není použitelný. Rovnice jsou v tomto případě spřáhnuté a nelineární. Proto jejich řešení (tj. skryté a výstupní aktivity) lze dosáhnout jen použitím iteračního postupu, a to tak, že se startuje z počátečních aktivit, pomocí těchto se spočítají nové aktivity a tyto se v následujícím iteračním kroku použijí jako vstup pro výpočet nových aktivit. Tento iterační postup se opakuje tak dlouho, až rozdíl mezi starými a novými aktivitami je menší než předepsaná přesnost.

Vrstvy neuronové sítě

Neurony jsou umístěny v několika lineárně uspořádaných vrstvách tak, že každý neuron v předchozí vrstvě je propojen s každým neuronem v následující vrstvě, tzv. úplné propojení neuronů, viz obr. 4.4.



Obr. 4.4 Příklad acyklické struktury

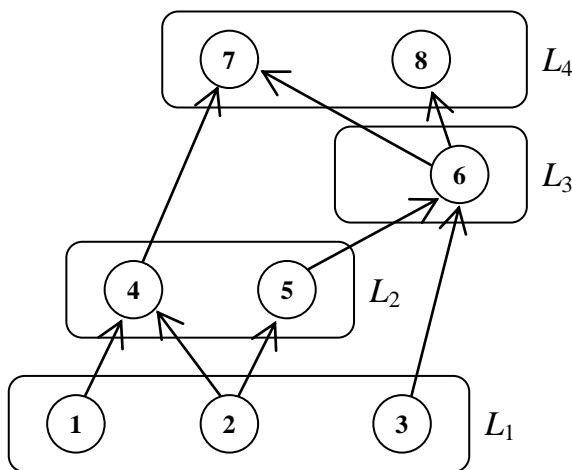
Na obr. 4.5 je znázorněna možnost rozkladu vrcholů (neuronů) acyklického orientovaného grafu na vrstvy L_1, \dots, L_4 .

$$V = L_1 \cup L_2 \cup L_3 \cup \dots \cup L_t, \quad (4.12)$$

kde $L_1 = V_1$ je *vstupní vrstva* (obsahuje pouze vstupní neurony), L_2, L_3, \dots, L_{t-1} jsou *skryté vrstvy* a L_t je *výstupní vrstva*. Vrstva L_i (pro $1 \leq i \leq t$) je určena následujícím jednoduchým způsobem

$$L_i = \{v \in V; d(v) = i-1\}, \quad (4.13)$$

kde vzdálenost $d(v)$ se rovná délce maximální cesty, která spojuje daný neuron se vstupním neuronem. Potom musí platit $d(v) = 0, v \in V_1$. Neuronová síť určená acyklickým grafem je obvykle zvolena tak, že neurony ze dvou sousedících vrstev jsou pospojované všemi možnými spoji. Bohužel, takovýto rozklad množiny neuronů na vrstvy je možný jen pro neuronové sítě reprezentované acyklickými grafy, pro cyklické grafy vzdálenost $d(v)$ může nabývat libovolnou kladnou celočíselnou hodnotu.



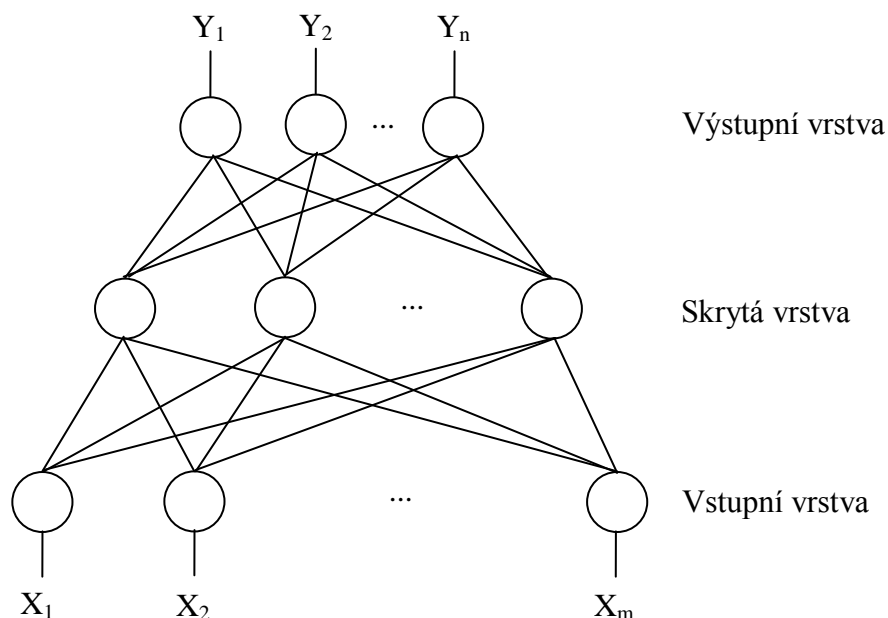
Obr. 4.5 Neuronová síť definovaná jako orientovaný souvislý graf

V první tzv. vstupní vrstvě (Input Layer) jsou vstupní neurony, které identicky přenášejí jednotlivé složky vstupního vektoru. Poslední vrstva se nazývá výstupní vrstva (Output Layer) a vytváří finální výstup. Mezilehlým vrstvám se říká skryté vrstvy (Hidden Layer). Vstupní a výstupní vrstva jsou přesně určeny počtem a typem vstupních a výstupních proměnných modelovaného systému. Počet skrytých vrstev a počet neuronů ve skrytých vrstvách však také není zcela libovolný. Velké počty neuronů ve skrytých

vrstvách sice urychlují dobu učení, ale znesnadňují postup testování sítě (sít' si mnoho pamatuje, ale málo zobecňuje). Je tedy nutné rozhodnout o počtu skrytých vrstev a neuronech ve skrytých vrstvách neuronové sítě. Na určení konkrétního optimálního počtu neuronů ve skryté vrstvě jsou potřebné experimenty. V případě, že je neuronová sít' využívána jako prediktor, doporučuje se ve skryté vrstvě minimálně o jeden neuron méně než ve vstupní. Vrstvy neuronové sítě jsou uvedeny na obr. 4.6.

Učení neuronové sítě

V procesu učení neuronové sítě [4] dochází k úpravám hodnot vah synapsí mezi neurony. Cílem je dosáhnout takový stav neuronové sítě, ve kterém bude schopna klasifikovat všechny vstupní vzorky nebo je správným způsobem zpracovat. Vzhledem ke složitosti struktury neuronových sítí neexistuje v současnosti žádná známá metoda, která by byla schopná jednoznačně určit hodnoty vah synapsí definované úlohy. Často používanými jsou gradientní metody, např. metoda největšího spádu. Úpravou gradientního algoritmu pro hledání lokálního minima funkce vznikl učící algoritmus zpětného šíření chyby – Back-Propagation.



Obr. 4.6 Vrstvy neuronové sítě

Back-Propagation je nejpoužívanější metoda pro učení neuronových sítí. Algoritmus je založený na minimalizaci součtu čtverců chyb s využitím poznatku o

průběhu nelineární aktivační funkce v neuronech. Požadavkem je, aby aktivační funkce byla spojitá, diferencovatelná a monotónně neklesající. Nejčastěji používanou aktivační funkcí je proto sigmoidální funkce a hyperbolický tangens. Učení pomocí algoritmu zpětného šíření chyby vyžaduje přítomnost učitele.

4.3 Shrnutí kapitoly

V první části kapitoly byly vysvětleny základní pojmy z oblasti neuronových sítí, kde byl zmíněn model umělého neuronu, možné typy jeho vstupů a aktivačních funkcí. Na závěr byla popsána struktura neuronových sítí a princip jejich učení pomocí algoritmu Back-Propagation.

5 Neuronové sítě vhodné pro predikci

V kapitole je definováno několik typů neuronových sítí, které jsou vhodné pro predikci časových řad. Jsou zde uvedeny vztahy pro výpočet výstupů těchto neuronových sítí a grafické znázornění. Taktéž je definován učící algoritmus zpětného šíření chyby pro tyto neuronové sítě.

5.1 Frontální neuronová síť

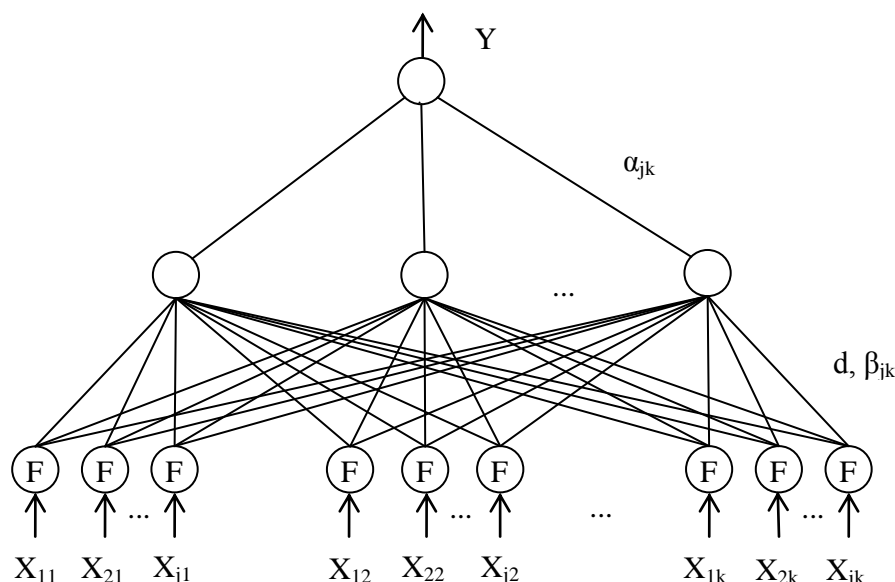
Frontální neuronová síť je speciální dopřednou neuronovou sítí, která se často označuje termínem Time-Delay Neural Net (TDNN). Tato neuronová síť využívá běžné časové prodlevy ke zpracování. Ve frontálních neuronových sítích je celá krátkodobá paměť situovaná na začátku neuronové sítě, jako určitá předvrstva, která zpracovává časový kontext. Tuto neuronovou síť je možné učit klasickým algoritmem zpětného šíření chyby – Back-Propagation. Výstup Delay neuronové sítě [4] se stejnou hloubkou paměti všech filtrů, lineárním filtrem a lineárním výstupním neuronem možno vyjádřit následujícím způsobem

$$Y = \sum_{k=1}^K \alpha_k \cdot d \left(\sum_{j=1}^J \beta_{jk} \cdot \sum_{i=1}^b \chi_{ijk} \cdot X_{ijk} \right), \quad (5.1)$$

kde: - Y výstup neuronové sítě,

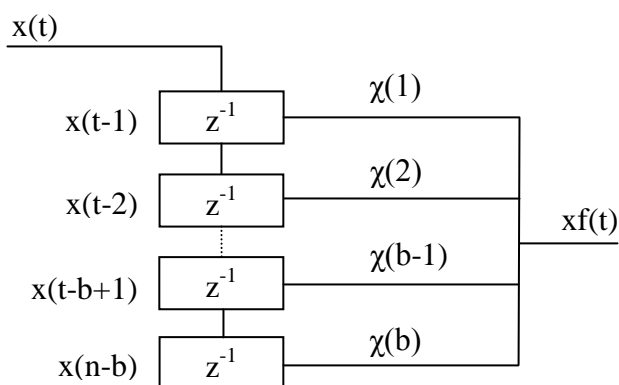
- α vektor vah synapsí mezi neurony ve skryté vrstvě a výstupním neuronem,
- β vektor vah synapsí mezi filtry a neurony ve skryté vrstvě,
- χ vektor vah synapsí uvnitř filtru,
- k index neuronu ve skryté vrstvě,
- d aktivační funkce,
- j index filtru,
- i index vstupu filtru,
- b hloubka krátkodobé paměti filtru,
- X vstupní vektor neuronové sítě.

Frontální neuronová síť je uvedena na obr. 5.1.



Obr. 5.1 Příklad frontální neuronové sítě

Na obr. 5.2 je zobrazen filtr frontální neuronové sítě, kde $x(t)$ je hodnota vstupu filtru v čase t , b je hloubka paměti filtru, χ jsou váhy synapsí v rámci filtru, z^{-1} je operátor jednotkového časového zpoždění a $xf(t)$ je výstup filtru pro hodnotu vstupu v čase t .



Obr. 5.2 Filtr frontální neuronové sítě

5.2 Dopředná neuronová síť

Neuronové sítě s dopředným šířením signálu (Feed-Forward Neural Networks), jsou acyklickým typem sítí (nikdy nevzniká zpětnovazební smyčka). Neurony jsou v nich organizovány do vrstev podle následujícího principu: neurony, z kterých synapse jen vystupují, tvoří vstupní vrstvu neuronové sítě. Na vstupní vrstvě neprobíhají žádné výpočty. Potenciál vstupních neuronů je zadaný uživatelem a neprochází skrz aktivační funkci. Neurony propojené synapsemi, které do nich vstupují a zároveň synapsemi, které z

nich vystupují, tvoří skrytou vrstvu, případně skryté vrstvy neuronové sítě. Na skrytých vrstvách se vypočítávají mezivýsledky řešení problému, jenž nejsou explicitně uvedené, protože jejich interpretace nemá praktický smysl. Neurony, do kterých synapse jen vstupují, tvoří výstupní vrstvu sítě. Na těchto neuronech probíhá výpočet konečného výstupu neuronové sítě. Obecně platí, že neurony ve výstupní vrstvě mezi sebou nejsou propojeny synapsí. Pro případ predikce toto není nutné uvažovat, neboť ve výstupní vrstvě neuronové sítě se nachází právě jeden neuron. Výstup dopředné neuronové sítě lze vyjádřit takto [4]

$$Y = \sum_{k=1}^K \alpha_k \cdot d \left(\sum_{j=1}^J \beta_{jk} \cdot X_{jk} \right), \quad (5.2)$$

kde: - Y je výstup neuronové sítě,

- α vektor vah synapsí mezi neurony ve skryté vrstvě a výstupním neuronem,

- β vektor vah synapsí mezi vstupními neurony a neurony ve skryté vrstvě,

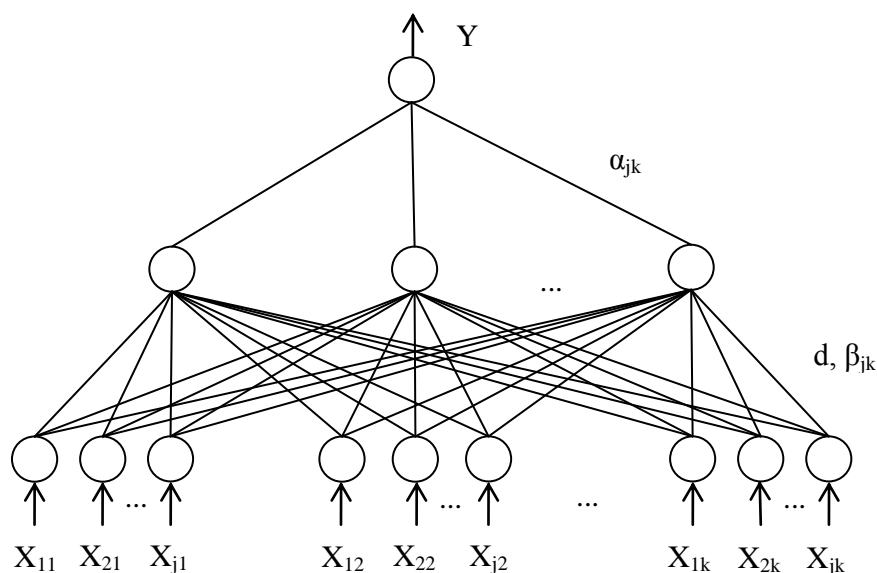
- k index neuronu ve skryté vrstvě,

- d aktivační funkce,

- j index vstupního neuronu,

- X vstupní vektor neuronové sítě.

Struktura dopředné neuronové sítě je znázorněná na obr. 5.3.

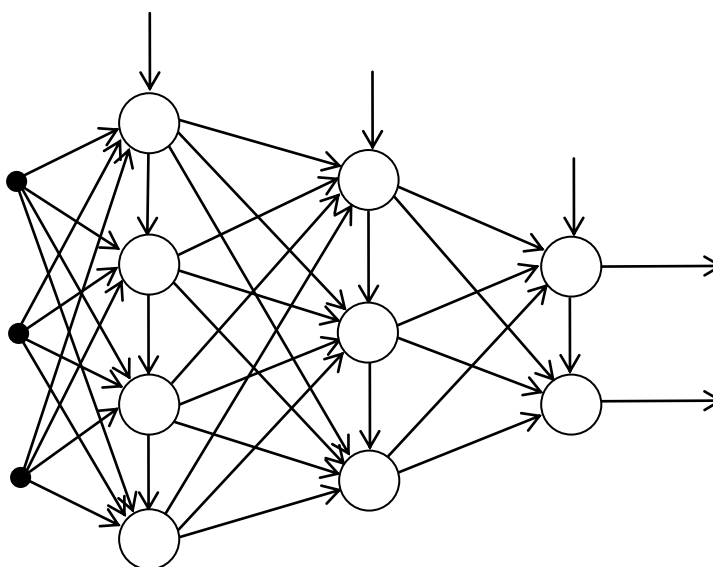


Obr. 5.3 Dopředná neuronová síť

Dopředné neuronové sítě, při jejichž učení je využita metoda zpětného šíření chyby, patří mezi neuronové sítě využívané jako klasifikátory a prediktory (rozdíl je v počtu výstupních neuronů).

5.3 Perceptronová neuronová síť

Perceptronová neuronová síť [11] patří mezi nejznámější a nejpoužívanější neuronové sítě. Její základní prvek, neuron zvaný perceptron, byl popsán F. Rosenblattem v roce 1957 spolu s algoritmem jeho učení, ale teprve v roce 1986 byl popsán algoritmus učení vícevrstvé perceptronové sítě. Díky tomuto algoritmu bylo možné vyřešit pomocí sítě tří neuronů problém logické funkce XOR, který nelze vyřešit pomocí jednoho neuronu. Je to síť učící se s učitelem, tj. trénovací vzory musí kromě vstupních hodnot obsahovat i hodnoty příslušných odpovídajících výstupů. Architektura je zobrazena na obr. 5.4.



Obr. 5.4 Možné uspořádání perceptronové neuronové sítě

5.4 Učící algoritmus Back-Propagation

Algoritmus zpětného šíření chyby Back-Propagation obsahuje tři etapy:

- dopředné šíření vstupního signálu učícího vzoru,
- zpětné šíření chyby,
- aktualizace váhových hodnot na synapsích.

Během dopředného šíření signálu obdrží každý neuron ve vstupní vrstvě vstupní signál a zprostředkuje jeho přenos ke všem neuronům skryté vrstvy. Každý neuron ve skryté vrstvě vypočítá svou hodnotu a pošle tento signál všem neuronům ve výstupní vrstvě. Každý neuron ve výstupní vrstvě vypočítá svou hodnotu, která odpovídá jeho skutečnému výstupu po předložení vstupního vzoru. Po dobu adaptace neuronové sítě jsou srovnávány vypočítané hodnoty se vstupními hodnotami pro každý neuron ve výstupní vrstvě a pro každý učicí vzor. Na základě tohoto srovnání je definována chyba neuronové sítě, pro kterou je vypočítán faktor δ_k , který je částí chyby, která se zpětně šíří z neuronu ve výstupní vrstvě ke všem neuronům předcházející vrstvy, jenž mají s tímto neuronem definovanou synapsi. Podobně lze definovat i faktor δ_j , který je částí chyby šířené zpětně z neuronu ve skryté vrstvě ke všem neuronům vstupní vrstvy, jež mají s tímto neuronem definovanou synapsi. Úprava hodnot vah na synapsích mezi neurony skryté a výstupní vrstvy závisí na faktoru δ_k a hodnotách neuronů ve skryté vrstvě. Obdobně úprava hodnot vah na synapsích mezi neurony vstupní a skryté vrstvy závisí na faktoru δ_j a hodnotách neuronů ve vstupní vrstvě. Aktivační funkce pro neuronové sítě s učícím algoritmem Back-Propagation musí mít následující vlastnosti:

- musí být spojitá,
- diferencovatelná,
- monotónně neklesající.

Nejčastěji používanou aktivační funkcí je proto logická (standardní) sigmoidální funkce (4.7). Chyba sítě $E(\mathbf{w})$ je vzhledem k učicí množině definována jako součet parciálních chyb sítě $E_i(\mathbf{w})$ vzhledem k jednotlivým učícím vzorům

$$E(\mathbf{w}) = \sum_{i=1}^n E_i(\mathbf{w}). \quad (5.7)$$

Parciální chyba $E_i(\mathbf{w})$ sítě pro i -tý učicí vzor je úměrná součtu mocnin odchylek skutečných hodnot výstupu sítě pro vstup i -tého učicího vzoru od požadovaných hodnot u tohoto vzoru

$$E_i = \frac{1}{2} \sum_i (y_i - t_i)^2, \quad (5.8)$$

- kde: - y_i je i -tý výstup neuronové sítě,
 - t_i je i -tý učící vzor.

5.5 Shrnutí kapitoly

V první podkapitole je definována frontální neuronová síť, která bude následně použita k modelování predikce HDP USA. Dále je definována dopředná neuronová síť, v níž nikdy nevzniká zpětná vazba, a vyjádřen její výstup. Následující podkapitola je zaměřena na perceptron, což je neuronová síť, která se často používá pro predikci. V poslední podkapitole je vysvětlen algoritmus zpětného šíření chyby Back-Propagation, který se ve velké míře používá pro učení všech výše popsaných neuronových sítí.

6 Návrh modelu frontální neuronové sítě na predikci HDP

V kapitole je uveden návrh frontální neuronové sítě s filtrem o délce $b = 6$ a se strukturou 10-5-1. Následně je zde popsán postup přípravy vstupních souborů, rozdělení množiny dat na trénovací a testovací množinu, princip nastavování parametrů a funkcí při učení neuronové sítě. Na závěr kapitoly jsou uvedeny výsledky modelování.

6.1 Návrh frontální neuronové sítě

Pokud slouží neuronová síť jako prediktor, výstupní vrstva obsahuje právě jeden neuron. Filtr frontální neuronové sítě b je roven šesti.

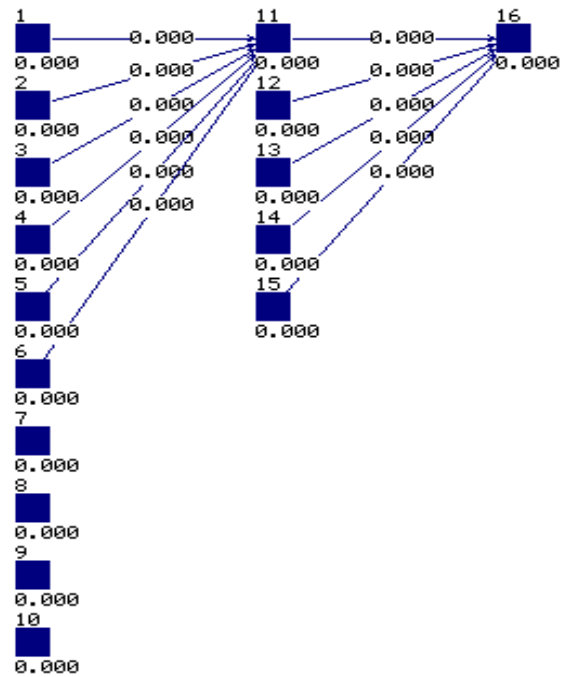
Návrh struktury sítě

Navržená frontální neuronová síť má deset neuronů ve vstupní vrstvě, pět neuronů ve skryté vrstvě a jeden neuron ve výstupní vrstvě. Jako vstupy následující devítiměsíční modely trendových složek:

- jednoduchý klouzavý průměr IVEI,
- vážený klouzavý průměr IVEI,
- trojúhelníkový klouzavý průměr IVEI,
- klouzavý medián IVEI,
- jednoduchý klouzavý průměr DIVEI,
- vážený klouzavý průměr DIVEI,
- trojúhelníkový klouzavý průměr DIVEI,
- klouzavý medián DIVEI,
- jednoduché exponenciální vyrovnání $\alpha = 0.2$ pro IVEI i DIVEI.

Výstupem neuronové sítě je procentuální růst HDP.

Na obr. 6.1 je uvedena navržená frontální neuronová síť s strukturou 10-5-1.



Obr. 6.1 Struktura navržené frontální neuronové sítě 10-5-1

Rozdělení vstupů na trénovací a testovací množinu

Následně byla vstupní data A rozdělena na trénovací množinu A_{train} a testovací množinu A_{test} v poměru 189:177. Je možno zapsat $A = A_{\text{train}} \cup A_{\text{test}}$. Trénovací množina obsahuje údaje z let 1965 – 1981 a je určena pro trénování neuronové sítě. Testovací množina obsahuje údaje z let 1982 – 1995 a je používána k určení přednastavení neuronové sítě pro data, které nebyly předkládány v procesu učení.

Učení neuronové sítě

Při učení frontální neuronové sítě byl použit učící algoritmus zpětného šíření chyby Back-Propagation, který se pro učení těchto typů neuronových sítí používá standardně. Pro tento učící algoritmu jsou nastavovány dva parametry:

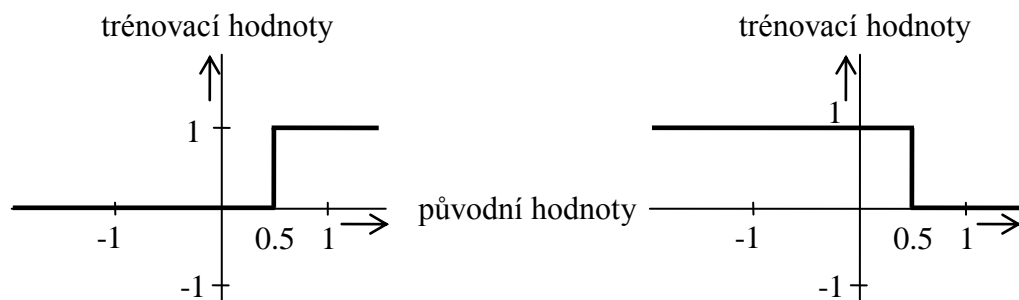
- μ je parametr učení, učící koeficient, který definuje rychlost učení. Čím je délka kroku nastavena na vyšší hodnotu, tím je proces učení rychlejší. Pro tento návrh je zvoleno $\mu = 0.01$.

- d_{\max} je maximální rozdíl mezi učící hodnotou a výstupem výstupního neuronu, který je tolerován, tj. který je možné zpětně šířit. Typické hodnoty d_{\max} jsou 0, 0.1, 0.2. Pro tento návrh je zvoleno $d_{\max} = 0$.

Činnosti s jednotlivými neurony je třeba zpracovávat ve specifickém pořadí, které je závislé na struktuře neuronové sítě. Pro každou síť je nutné vybrat funkci update za účelem požadovaného chování neuronové sítě. Pro frontální neuronové sítě se standardně používá funkce Time-Delay Order, při níž jsou výpočty pro jednotlivé neurony prováděny v pořadí závislém na struktuře neuronové sítě. Touto funkcí se v procesu učení definuje, jakým způsobem má projít přes všechny neurony.

Před učením byla neuronová síť inicializována pomocí inicializační funkce Randomize Wiegths. Pomocí této funkce jsou váhy synapsí náhodně nastaveny na hodnoty z intervalu od -1.0 do 1.0.

Přenosová funkce je prostředek pro rychlou změnu požadovaného výstupu sítě, aniž by musel být změněn soubor vzorů. Tato funkce upravuje výstupní část vzorů, vstupní část zůstává nedotčená. Vstupní hodnoty každého vzoru procházejí touto funkcí před tím, než jsou předloženy síti k učení. V návrzích byla využita funkce Binary, která je uvedena na obr. 6.2. V podstatě lze říci, že je to binární klasifikátor.

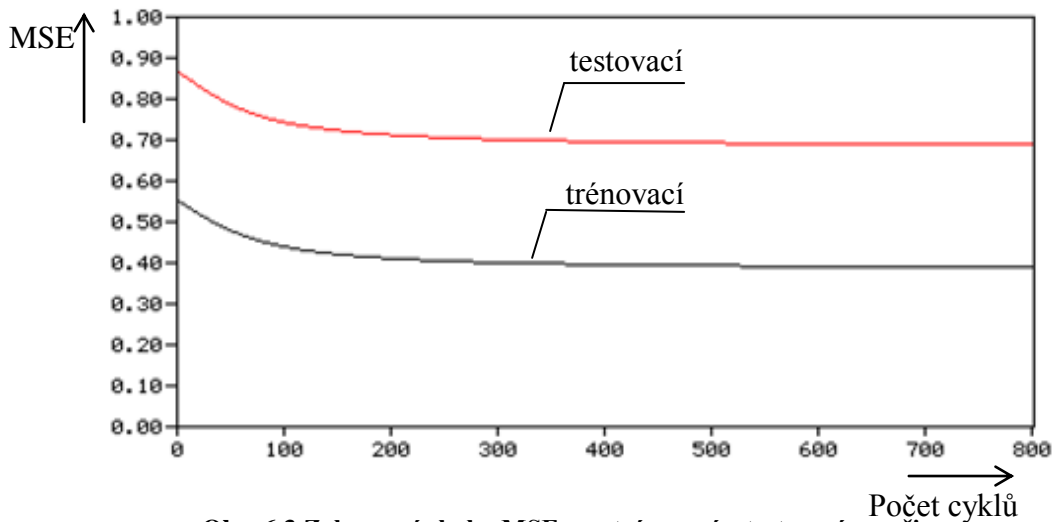


Obr. 6.2 Přenosová funkce Binary

Rozděluje hodnoty výstupních vzorů na hodnoty 0 a 1. Práh je zde implicitně nastaven na hodnotu 0.5, což znamená, že hodnoty vyšší než 0.5 budou učeny jako 1, všechny ostatní, včetně záporných, jako 0. Tato funkce nemá žádné volitelné parametry.

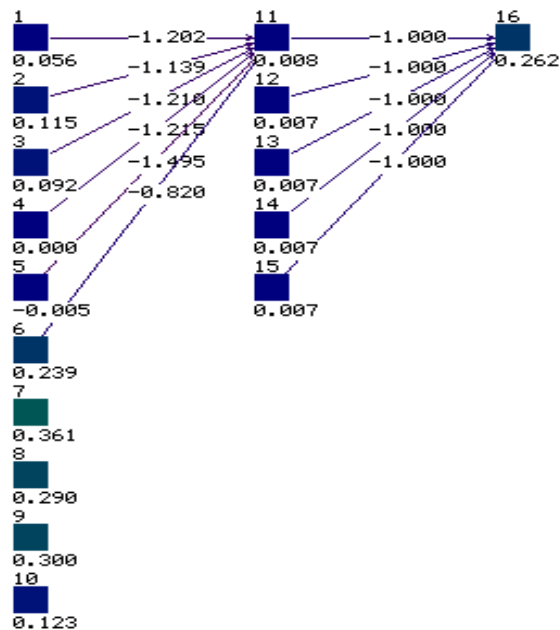
6.2 Výsledky učení neuronové sítě

Neuronová síť se učí s daným počtem cyklů. V každém cyklu jsou všechny trénovací vzory zpracovány neuronovou sítí pouze jednou. Učení probíhalo v tisíci cyklech, v každém druhém cyklu bylo provedeno testování. Při učení sítě je sledována střední kvadratická chyba MSE trénovací množiny v každém učícím cyklu a je při procesu učení sítě minimalizována. Průběh chyby MSE pro navrženou frontální neuronovou síť se strukturou 10-5-1 je zobrazen na obr. 6.3.



Obr. 6.3 Zobrazení chyby MSE pro trénovací a testovací množinu

Na obr. 6.4 je zobrazena frontální neuronová síť, která prošla procesem učení.



Obr. 6.4 Frontální neuronová síť po procesu učení

Tato neuronová síť má již nastaveny váhy jednotlivých synapsí a prahy neuronů.
 Takto vytvořenou neuronovou síť lze použít pro predikci vývoje HDP.

6.2.1 Protokol o průběhu učení neuronové sítě

Learning func: TimeDelayBackprop
 Remap. func: Binary
 Init. func: Randomize_Weights
 Update func: TimeDelay_Order
 Learning func: TimeDelayBackprop

Learning all patterns:
 epochs : 1000
 parameter: 0.01000
 #o-units : 1
 #patterns: 192 (total: 192)

epoch:	SSE	MSE	SSE/o-units
Train 1000:	105.30939	0.54849	105.30939
Test 1000:	148.57376	0.86380	148.57376
Train 900:	83.87697	0.43686	83.87697
Test 900:	127.35912	0.74046	127.35912
Train 800:	78.40769	0.40837	78.40769
Test 800:	122.06600	0.70969	122.06600
Train 700:	76.48042	0.39834	76.48042
Test 700:	120.19688	0.69882	120.19688
Train 600:	75.53819	0.39343	75.53819
Test 600:	119.27588	0.69346	119.27588
Train 500:	74.98549	0.39055	74.98549
Test 500:	118.73239	0.69030	118.73239
Train 400:	74.62319	0.38866	74.62319
Test 400:	118.37495	0.68823	118.37495
Train 300:	74.36720	0.38733	74.36720
Test 300:	118.12227	0.68676	118.12227
Train 200:	74.17637	0.38634	74.17637
Test 200:	117.93414	0.68566	117.93414
Train 100:	74.02836	0.38556	74.02836
Test 100:	117.78877	0.68482	117.78877
Train 1:	73.91069	0.38495	73.91069
Test 1:	117.67377	0.68415	117.67377

----- s t a t i s t i c s -----

Number of Patterns : 192
 Number of parameters (Links+bias): 17
 Sse : 73.9101
 Tss : 42.6667
 Rsq : -0.7323
 Mse : 0.4223
 Rmse : 0.6499
 Adjrsq : -0.8906
 J_p : 0.4597
 gcv : 0.4634
 pc : -1.0688
 S_p : 0.0024
 gmsep : 0.4660
 shibata : 86.9984

```
aic          : -149.2920
sbc          : -93.9145
```

6.2.3 Nastavení vah na synapsích a prahy neuronů

SNNS network definition file V1.4-3D
generated at Mon May 14 14:37:08 2007

```
network name : learn_10_5_1
source files :
no. of units : 16
no. of connections : 11
no. of unit types : 0
no. of site types : 0
```

```
learning function : TimeDelayBackprop
update function  : TimeDelay_Order
```

unit default section :

act	bias	st	subnet	layer	act func	out func
0.00000	0.00000	h	0	1	Act_Logistic	Out_Identity

unit definition section :

no.	typeName	unitName	act	bias	st	position	act func	out func	sites
1			-0.09700	-1.00000	i	2, 2,17632	Act_TD_Logistic		
2			0.07000	-0.99998	i	2, 3,17632	Act_TD_Logistic		
3			-0.00300	-0.99999	i	2, 4,17632	Act_TD_Logistic		
4			-0.22000	-0.99998	i	2, 5,17632	Act_TD_Logistic		
5			0.05900	-0.99998	i	2, 6,17632	Act_TD_Logistic		
6			-0.24400	-0.99999	i	2, 7,17632	Act_TD_Logistic		
7			-0.33000	-0.99999	i	2, 8,17632	Act_TD_Logistic		
8			-0.27600	-0.99997	i	2, 9,17632	Act_TD_Logistic		
9			-0.30000	-0.99997	i	2,10,17632	Act_TD_Logistic		
10			-0.18400	-0.99998	i	2,11,17632	Act_TD_Logistic		
11			0.02024	-4.29950	h	5, 2,17632	Act_TD_Logistic		
12			0.00687	-4.97314	h	5, 3,17632	Act_TD_Logistic		
13			0.00687	-4.97314	h	5, 4,17632	Act_TD_Logistic		
14			0.00687	-4.97313	h	5, 5,17632	Act_TD_Logistic		
15			0.00687	-4.97312	h	5, 6,17632	Act_TD_Logistic		
16			0.25966	-0.99999	o	8, 2,17632	Act_TD_Logistic		

connection definition section :

target	site	source:weight
11		1:-1.20195, 2:-1.13868, 3:-1.21006, 4:-1.21472, 5:-1.49450, 6:-0.82039
16		11:-1.00000, 12:-1.00000, 13:-1.00000, 14:-0.99999, 15:-0.99998

time delay section :

no.	LLN	LUN	Toff	Soff	Ctype
1	3	0	0	0	0
2	3	0	0	0	0
3	3	0	0	0	0
4	3	0	0	0	0
5	3	0	0	0	0
6	3	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	2	0	0	0	0
12	2	0	0	0	0
13	2	0	0	0	0
14	2	0	0	0	0
15	2	0	0	0	0
16	1	0	0	0	0

6.3 Shrnutí kapitoly

Cílem kapitoly byl návrh frontální neuronové sítě se strukturou 10-5-1 a její analýza. Tento model frontální neuronové sítě byl vybrán, protože se jeví jako optimální pro predikci. Má dostatečně velký filtr b a počet neuronů jak ve vstupní tak i ve skryté vrstvě je dostatečný na to, aby byly odhaleny skryté závislosti ve vstupních datech.

V první části je uveden postup návrhu frontální neuronové sítě, popis vstupů a výstupu neuronové sítě, vysvětleno parametrů a funkcí pro učení neuronové sítě. Ve druhé části kapitoly jsou zobrazeny výsledky učení neuronové sítě interpretované grafem zobrazujícím chybu MSE v průběhu učení, protokolem o průběhu učení a zejména zobrazení nastavení vah na synapsích a prahy jednotlivých neuronů.

7 Analýza výsledků

V kapitole jsou popsány modely frontálních neuronových sítí pro predikci HDP lišící se především velikostí filtru b , který definuje velikost klouzavého okna frontální neuronové sítě. Dále se modely liší počtem neuronů ve vstupní a skryté vrstvě. U všech modelů byly použity následující funkce a parametry učení:

- Jako učící funkce byla zvolena Time-Delay Back-Propagation s parametry $\mu = 0.01$ a $d_{\max} = 0$.
- Jako update funkce byla vybrána funkce Time-Delay Order.
- Před učením se neuronová síť inicializovala pomocí funkce Randomize Weights, kdy byly váhy synapsí náhodně nastaveny na hodnoty z intervalu od -1.0 do 1.0.
- Přenosovou funkcí byla zvolena Binary.

7.1 Vstupní data

Prvním krokem při návrhu modelu bylo nutno připravit vstupní data, a to v závislosti na počtu vstupních neuronů neuronové sítě. Pro návrh byly vybrány struktury s 6, 8, 10, 12, 14 vstupními neurony. Jako vstupy do modelů vybrány následující devítiměsíční modely trendových složek:

- jednoduchý klouzavý průměr (JKP) pro IVEI a DIVEI,
- vážený klouzavý průměr (VKP) pro IVEI a DIVEI,
- trojúhelníkový klouzavý průměr (TKP) pro IVEI a DIVEI,
- klouzavý medián (KM) pro IVEI a DIVEI,
- šestiměsíční centrovaný klouzavý průměr (CKP) pro IVEI i DIVEI
- jednoduché exponenciální vyrovnání (EV) pro $\alpha = 0.1$ a $\alpha = 0.2$ pro IVEI a DIVEI.

Rozdělení vstupních dat podle počtu neuronů ve vstupní vrstvě je zobrazeno v tabulce 7.1.

Tabulka 7.1 Vstupní data do modelů

Vstupy do neuronové sítě/počet vstupních neuronů	6	8	10	12	14
JKP pro IVEI a DIVEI	x	x	x	x	x
VKP pro IVEI a DIVEI	x	x	x	x	x
TKP pro IVEI a DIVEI		x	x	x	x

CKP pro IVEI a DIVEI				x	x
KM pro IVEI a DIVEI	x	x	x	x	x
EV pro $\alpha = 0.1$					x
EV pro $\alpha = 0.2$			x	x	x

7.2 Modely s hloubkou paměti filtru $b = 4$

Pro model s velikostí filtru $b=4$ bylo vybráno 5 struktur neuronových sítí lišících se počtem neuronů ve vstupní a skryté vrstvě. Výsledky modelování jsou uvedeny v tabulce 7.2.

Tabulka 7.2 Výsledky chyb pro model s hloubkou filtru $b=4$

Model	MSE	RMSE	Naučenost sítě
6-3-1	0.4067	0.6377	150 cyklů
8-5-1	0.4179	0.6465	250 cyklů
10-7-1	0.4302	0.6559	300 cyklů
12-9-1	0.4434	0.6659	400 cyklů
14-11-1	0.4577	0.6765	450 cyklů

Z výsledků je možno odvodit, že čím větší je počet neuronů ve skryté vrstvě, tím je chyba MSE větší. S větším počtem neuronů ve skryté vrstvě se také snižuje kvalita modelu, jelikož chyba RMSE a k naučení neuronové sítě je třeba větší množství trénovacích cyklů. Jak je možno odvodit z tabulky, model s architekturou 6-3-1 se naučil nejlépe (s nejmenší chybou) nicméně není primárně vhodný na predikci, jelikož má ve skryté vrstvě pouze čtyři neurony a z toho plyne, že není schopen zachytit závislosti ve vstupních datech.

7.3 Modely s hloubkou paměti filtru $b = 5$

Další zkoumané modely měli nastavenou hloubku paměti filtru b rovnu pěti. Byly vybrány 4 architektury neuronových sítí, kdy byl měněn počet neuronů ve vstupní a skryté vrstvě. Oproti předchozím modelům s hloubkou filtru 4 došlo ke zkvalitnění modelů, když došlo k poklesu jak chyby MSE tak RMSE. Klesl také počet cyklů potřebný k naučení neuronové sítě. Výsledky jsou zobrazeny v tabulce 7.3.

Tabulka 7.3 Výsledky chyb pro model s hloubkou filtru $b=5$

Model	MSE	RMSE	Naučenost sítě
8-4-1	0.4144	0.6438	220 cyklů
10-6-1	0.4260	0.6527	290 cyklů
12-8-1	0.4390	0.6626	350 cyklů
14-10-1	0.4528	0.6729	450 cyklů

Při porovnání modelů s hloubkou filtru 4 a 5 s strukturou neuronové sítě 12-9-1 a 12-8-1 lze dojít k závěru, že model s hloubkou filtru 5 dosahuje lepší vlastností ve všech zkoumaných vlastnostech. Nižší je chyba MSE i RMSE a k naučení bylo třeba menší množství učících cyklů. Z porovnání modelů vyplývá, že kvalita modelů se liší jen nepatrně.

7.4 Modely s hloubkou paměti filtru $b = 6$

Následně byly analyzovány modely s hloubkou paměti filtru b rovnu šesti a byly vytvořeny čtyři modelové struktury neuronových sítí. Výsledky modelování jsou zobrazeny v tabulce 7.4.

Tabulka 7.4 Výsledky chyb pro model s hloubkou filtru $b=6$

Model	MSE	RMSE	Naučenost sítě
8-3-1	0.4110	0.6411	200 cyklů
10-5-1	0.4223	0.6499	270 cyklů
12-7-1	0.4346	0.6593	370 cyklů
14-9-1	0.4480	0.6693	420 cyklů

U všech testovaných modelů opět došlo ke snížení chyb MSE a RMSE a také poklesl počet cyklů nutný k naučení neuronové sítě.

7.5 Modely s hloubkou paměti filtru $b = 7$

Dalšími modely jsou ty, jenž mají hloubky paměti filtru b nastaveny na sedm. Z tabulky 7.5 vyplývá, že se se zvětšováním filtru b kvalita analyzovaných modelů stále zvyšuje. Chyby MSE a RMSE jsou v porovnání s předešlými modely se stejným počtem vstupních neuronů nižší a rychlost naučení sítě také poklesla.

Tabulka 7.5 Výsledky chyb pro model s hloubkou filtru $b=7$

Model	MSE	RMSE	Naučenost sítě
10-4-1	0.4188	0.6471	240 cyklů
12-6-1	0.4308	0.6564	290 cyklů
14-8-1	0.4434	0.6659	390 cyklů

7.6 Modely s hloubkou paměti filtru $b = 8$

Předposledními analyzovanými modely jsou ty, u nichž je hloubka paměti filtru b rovna osmi. Výsledky jsou zobrazeny v tabulce 7.6. Opět se zmenšila jak chyba učení a poklesl počet cyklů nutných k naučení neuronové sítě.

Tabulka 7.6 Výsledky chyb pro model s hloubkou filtru b=8

Model	MSE	RMSE	Naučenost sítě
10-3-1	0.4153	0.6444	200 cyklů
12-5-1	0.4271	0.6535	270 cyklů
14-7-1	0.4394	0.6629	300 cyklů

Z porovnání těchto modelů s předešlými vyplývá, že kvalita naučení modelů se zvyšuje úměrně se zvětšujícím se filtrem b frontální neuronové sítě.

7.7 Modely s hloubkou paměti filtru b = 9

Posledními modely jsou ty, jenž mají hloubky paměti filtru b nastavenou na devět. Z tabulky 7.7 plyne, že kvalita analyzovaných modelů se opět zvýšila a to ve všech parametrech.

Tabulka 7.7 Výsledky chyb pro model s hloubkou filtru b=9

Model	MSE	RMSE	Naučenost sítě
12-4-1	0.4235	0.6508	250 cyklů
14-6-1	0.4356	0.66	340 cyklů

Chyby MSE a RMSE jsou v porovnání s předešlými modely se stejným počtem vstupních neuronů nejnižší a rychlost naučení sítě také ještě poklesla na nejnižší hodnotu.

7.8 Přehled výsledků modelování

V tabulce 7.8 jsou přehledně zobrazeny výsledky učení při různých počtech vstupních neuronů a různé velikosti filtru b.

Tabulka 7.8 Chyba MSE v závislosti na počtu vstupních neuronů a velikosti filtru

Počet neuronů ve vstupní vrstvě	b = 4	b = 5	b = 6	b = 7	b = 8	b = 9
6	0.4067					
8	0.4179	0.4144	0.4110			
10		0.4260	0.4223	0.4188	0.4153	
12		0.4390	0.4346	0.4308	0.4271	0.4235
14		0.4528	0.4480	0.4434	0.4394	0.4356

Z dosažených výsledků modelování byl zvolen jako nejlepší model 10-5-1 s filtrem b=6, který se jeví jako nejlepší pro predikci. Velikost hloubky filtru zajišťuje dostatečný rozsah klouzavého okna na zachycení skrytých závislostí v datech a co nejpřesnější naučení neuronové sítě. Tato struktura neuronové sítě má také dostatečný počet neuronů ve

skryté vrstvě, které jsou nezbytné pro odhalení skrytých závislostí v datech. Chyba učení $MSE = 0.4223$ je také přijatelná.

7.8 Shrnutí kapitoly

Cílem kapitoly byla analýza navržených modelů. Celkem bylo navrženo 21 modelů frontálních neuronových sítí, které se navzájem lišily velikostí filtru, počty neuronů ve vstupních a skrytých vrstvách. Nejvhodnějším modelem pro predikci HDP byl zvolen model s architekturou 10-5-1, který má dostatečně velký filtr b a počet neuronů jak ve vstupní tak i ve skryté vrstvě je dostatečný na to, aby byly odhaleny skryté závislosti ve vstupních datech. Tento model je podrobně popsán v kapitole 6. Při zvětšování velikosti filtru chyba MSE i RMSE klesá a taktéž se snižuje počet cyklů nutných k natrénování neuronové sítě. Dále je možno konstatovat, že větší počet neuronů jak ve vstupní tak i skryté vrstvě zvyšuje chybu MSE, RMSE i počet cyklů, které jsou potřebné k naučení sítě.

8 Závěr

Cílem diplomové práce byl návrh frontální neuronové sítě na predikci HDP. K dispozici byli předstihoví ukazatelé, kteří jsou vhodní pro modelování vývoje HDP. Jednalo se o IVEI a DIVEI. Oba indexy byly pomocí adaptivních přístupů k vyrovnání trendové složky modelovány a posloužily jako vstupní data do neuronových sítí, které byly konstruovány a analyzovány v prostředí SNNS.

U vytvořených modelů frontálních neuronových sítí byla měněna hloubka paměti filtru a pro každý z nich vytvořeno několik modelů neuronových sítí. V nich se měnil počet neuronů ve vstupní a skryté vrstvě. Zjišťována byla změna MSE, RMSE a počet cyklů potřebných k naučení neuronové sítě při zachování všech funkcí a jejich parametrů.

Z dosažených výsledků je možné říci, že frontální neuronové sítě jsou vhodným prostředkem na vytváření predikčních modelů. Nejlepším analyzovaným modelem pro predikci HDP je model s filtrem šest, který obsahoval deset neuronů ve vstupní vrstvě a pět neuronů ve skryté vrstvě.

Seznam použité literatury

- [1] HOLMAN, R. *Ekonomie*. 1. vydání. Praha: C. H. Beck, 1999. ISBN 80-7179-255-1. 726 s.
- [2] KADEŘÁBKOVÁ, A. *Základy makroekonomické analýzy*. 1. vydání. Praha: Linde, 2003. ISBN 80-86131-36-X. 175 s.
- [3] RUSMICOVÁ, L., SOUKUP, J. *Makroekonomie základní kurs*. Dotisk 5. vydání. Slaný: MELANDRIUM, 2002. ISBN 80-86175-24-3. 167 s.
- [4] OLEJ, V. *Modelovanie ekonomických procesov na báze výpočtovej inteligencie*. Vědecká monografie. Hradec Králové: Miloš Vognar – M&V, 2003. ISBN 80-903024-9-1. 160s.
- [5] SPĚVÁČEK, V. *Makroekonomická analýza a prognózování*. Praha: Vysoká škola ekonomická, Fakulta národohospodářská, 1999. ISBN 80-7079-922-6. 119 s.
- [6] ZAPLETAL, J. *Úvod do analýzy ekonomických časových řad*. 1. vydání. Brno: PC-DIR Real, 2000. ISBN 80-214-1719-6. 112 s.
- [7] CIPRA, T. *Praktický průvodce finanční a pojistnou matematikou*. 1. vydání. Praha: HZ Praha, 1995. ISBN 80-901918-0-0. 320 s.
- [8] CIPRA, T. *Analýza časových řad s aplikacemi v ekonomii*. 1. vydání. Praha: SNTL/Alfa, 1986. ISBN 04-012-86. 248 s.
- [9] HANČLOVÁ, J., TUHÝ, L. *Úvod do analýzy časových řad* [online].2003. [cit. 2006-04-03]. Dostupné na <http://gis.vsb.cz/pan/Skoleni_Texty/TextySkoleni/AnalyzaCasRad.pdf>.
- [10] HINDLS R., HRONOVÁ, S., SEGER, J. *Statistika pro ekonomy*. 4. vydání. Praha: Professional Publishing, 2003. ISBN 80-86419-52-5. 415 s.
- [11] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol. *Umělá inteligence (4)*. 1. vydání. Praha: Academia, Akademie věd České republiky, 2003. ISBN 80-200-1044-0. 475 s.
- [12] KVASNIČKA, V. a kol. *Úvod do teórie neurónových sietí*. Bratislava: IRIS, 1997. ISBN 80-88778-30-1. 262 s.
- [13] HAYKIN, S. *Neural Networks – A Comprehensive Foundation*. 2nd edition. New Jersey: Prentice Hall. 1999. ISBN 0-13-273350-1. 842 s.
- [14] OLEJ, V. Prediction of Gross Domestic Product Development on the Basis of Frontal Neural Networks, Genetic and Eugenic Algorithms. 2nd Euro-International Symposium on Computational Intelligence, E-ISCI 2002, Intelligent Technologies-Theory and Applications, New Trend in Intelligent Technologies, Netherlands, 2002, pp. 309-314, IOS Press Ohmsha, ISSN 0922-6389, ISBN 1 58603 256 9 IOS Press, ISBN 4 274 90512 8 Ohmsha.
- [15] OLEJ, V. Prediction of Gross Domestic Product Development by Frontal Neural Networks with Learning Process on the Basis Genetic and Eugenic Algorithms. Neural Network World, International Journal on Non-Standard Computing and Artificial Intelligence, editor: M. Novák, Czech Republic, Vol.12, No.3, 2002, pp.279-291, ISSN 1210-0552.

- [16] OLEJ, V. Design of the Models of Neural Networks and the Takagi-Sugeno Fuzzy Inference System for Prediction of the Gross Domestic Product Development., WSEAS Transactions on Systems, WSEAS Press, Issue 4, Vol.4, April 2005, pp.314-319, ISSN 1109-2777.
- [17] OLEJ, V. Design of the Models of Neural Networks and the Takagi-Sugeno Fuzzy Inference System for Prediction of the Gross Domestic Product Development. Proc. of the 6th International Conference on Neural Networks, Lisbon, Portugal, June 16-18, 2005, pp.120-125, ISBN 960-8457-24-6.
- [18] OLEJ, V. Modelovanie ekonomických procesov na báze vypočítavej inteligencie. [Vedecká monografia], Miloš Vognar-M&V, Hradec Králové, Česká republika, 2003, 160s., ISBN 80-903024-9-1.

Seznam obrázků

Obr. 1.1 Reálný růst HDP v USA	- 14 -
Obr. 1.2 Hospodářský cyklus	- 15 -
Obr. 1.3 Index vedoucích ekonomických indikátorů (1965 – 1995).....	- 17 -
Obr. 1.4 Difúzní index vedoucích ekonomických indikátorů (1965 – 1995).....	- 17 -
Obr. 2.1 Absolutní chyba a střední kvadratická odchylka predikce analyzovaného čtvrtletního růstu HDP USA, průměry let 1985 – 95, model Federální rezervní banky San Francisco	- 22 -
Obr. 4.1 Schematické znázornění zobrazení $F : A \rightarrow B$	- 31 -
Obr. 4.2 Nelineární model neuronu	- 34 -
Obr. 4.3 Průběh aktivační funkce	- 35 -
Obr. 4.4 Příklad acyklické struktury	- 36 -
Obr. 4.5 Neuronová síť definovaná jako orientovaný souvislý graf.....	- 37 -
Obr. 4.6 Vrstvy neuronové sítě	- 38 -
Obr. 5.1 Příklad frontální neuronové sítě	- 41 -
Obr. 5.2 Filtr frontální neuronové sítě	- 41 -
Obr. 5.3 Dopředná neuronová síť	- 42 -
Obr. 5.4 Možné uspořádání perceptrónové neuronové sítě	- 43 -
Obr. 6.1 Struktura navržené frontální neuronové sítě 10-5-1	- 47 -
Obr. 6.2 Přenosová funkce Binary	- 48 -
Obr. 6.3 Zobrazení chyby MSE pro trénovací a testovací množinu.....	- 49 -
Obr. 6.4 Frontální neuronová síť po procesu učení	- 49 -

Seznam tabulek

Tabulka 2.1 Ukazatelé chyby predikce roční míry růstu reálného HDP USA, průměry let 1980 – 95, model OECD.....	- 22 -
Tabulka 3.1 Přehled vah pro různé délky klouzavých průměrů a různé řády polynomů	- 28 -
Tabulka 7.1 Vstupní data do modelů	- 53 -
Tabulka 7.2 Výsledky chyb pro model s hloubkou filtru $b=4$	- 54 -
Tabulka 7.3 Výsledky chyb pro model s hloubkou filtru $b=5$	- 54 -
Tabulka 7.4 Výsledky chyb pro model s hloubkou filtru $b=6$	- 55 -
Tabulka 7.5 Výsledky chyb pro model s hloubkou filtru $b=7$	- 55 -
Tabulka 7.6 Výsledky chyb pro model s hloubkou filtru $b=8$	- 56 -
Tabulka 7.7 Výsledky chyb pro model s hloubkou filtru $b=9$	- 56 -
Tabulka 7.8 Chyba MSE v závislosti na počtu vstupních neuronů a velikosti filtru.....	- 56 -

Seznam příloh

Příloha 1: Uživatelská příručka k architektuře Time-Delay

Příloha 2: Protokoly o způsobu naučení vybraných modelů neuronových sítí, jejich nastavení vah synapsí a prahů neuronů

Seznam použitých zkratk

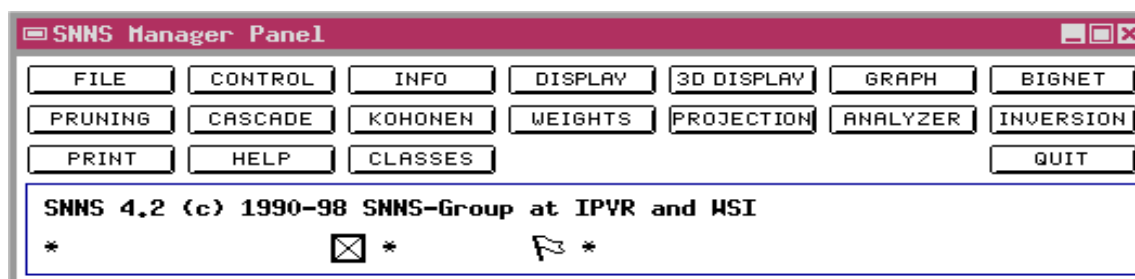
HDP	hrubý domácí produkt
IVEI	index vedoucích ekonomických indikátorů
DIVEI	difúzní index vedoucích ekonomických indikátorů
AVE	střední chyba
MAE	střední absolutní chyba
MSE	střední kvadratická chyba
RMSE	střední kvadratická odchylka
OECD	Organization for Economic Cooperation and Development
FRBSF	Federální rezervní banka San Francisco
SNNS	Stuttgart Neural Network Simulator
JKP	jednoduchý klouzavý průměr
VKP	vážený klouzavý průměr
TKP	trojúhelníkový klouzavý průměr
CKP	centrovaný klouzavý průměr
KM	klouzavý medián
EV1	jednoduché exponenciální vyrovnání
EV2	dvojitě exponenciální vyrovnání
TDNN	Time-Delay Neural Net

Příloha 1: Uživatelská příručka k architektuře Time-Delay

Stuttgartský neuronový simulátor (SNNS) je programový produkt pro simulaci neuronových sítí. Tento systém je významný svou rozsáhlostí, univerzálností, množstvím implementovaných architektur a učících algoritmů.

SNNS Manager

Po startu SNNS simulátoru se na obrazovce objeví základní panel – SNNS Manager. Tento panel je znázorněn na obr. 1. Skládá se ze tří řádků ikon. První řádek obsahuje funkce pro návrh NS, ve druhém řádku jsou uvedeny funkce pro úpravu parametrů jiných než dopředných NS (např. Kohonenovy mapy) a ve třetím řádku jsou pomocné funkce.



Obr. 1 Manager panel SNNS

Ze SNNS Managera jsou aktivována ostatní okna, která jsou užívána pro návrh NS. Kvůli této jeho vlastnosti se doporučuje, aby byl tento panel zobrazen po celou dobu práce se SNNS simulátorem.

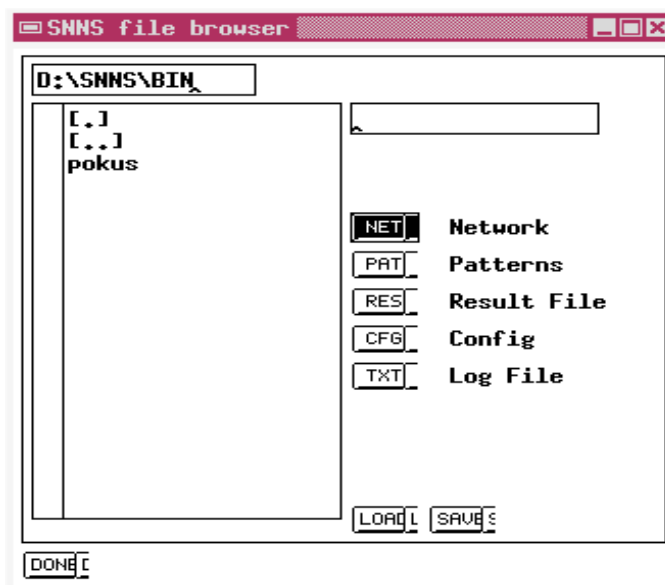
Otevírání a ukládání souborů

Pro práci se SNNS simulátorem je využíváno pět druhů souborů. Nejdůležitějšími jsou:

- NET** - soubory s příponou **‘.net’** obsahují informace o struktuře NS a učících algoritmech,
- PAT** - v souborech s příponou **‘.pat’** jsou zapsána trénovací a kontrolní (validační) data,
- RES** - výsledky navržených NS jsou ukládány do souborů s příponou **‘.res’**.

Soubory pro práci se SNNS simulátorem jsou načteny pomocí nabídky **FILE** na panelu **SNNS Manager**. Pomocí této funkce jsou vybrány soubory definující strukturu NS (**‘.net’**)

a soubory s trénovacími a kontrolními daty. Okno, které se zobrazí po aktivování nabídky FILE je na obr. 2.

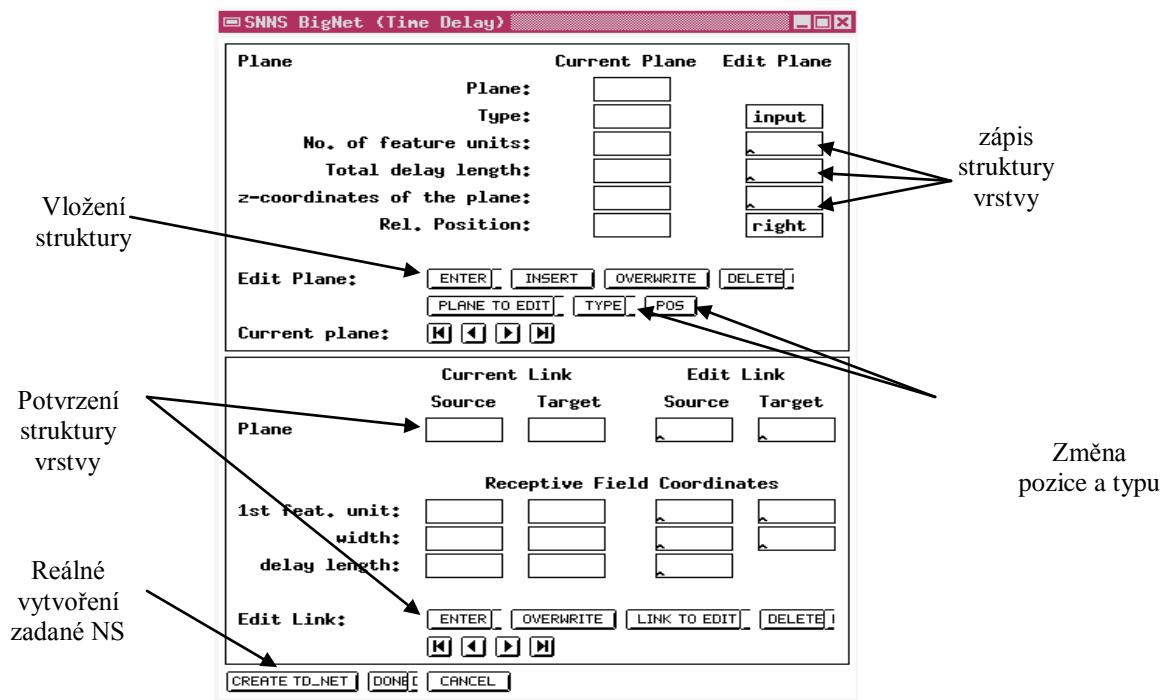


Obr. 2 Okno práce se soubory

Ve vrchním textovém poli je zobrazen aktuální adresář. V levé části okna jsou zobrazeny soubory vybraného typu, se kterými SNNS simulátor pracuje a v hranatých závorkách jsou zobrazeny adresáře. Typ souboru je vybrán v pravé části okna aktivováním typu souboru (např. **PAT**). Po aktivování souboru, který poslouží k práci se SNNS simulátorem, se jeho název objeví vpravo nahoře v poli, kde se zobrazují jména souborů. Vybraný soubor je načten pomocí tlačítka **LOAD** a práce s oknem je ukončena aktivováním **DONE** v levém dolním rohu okna.

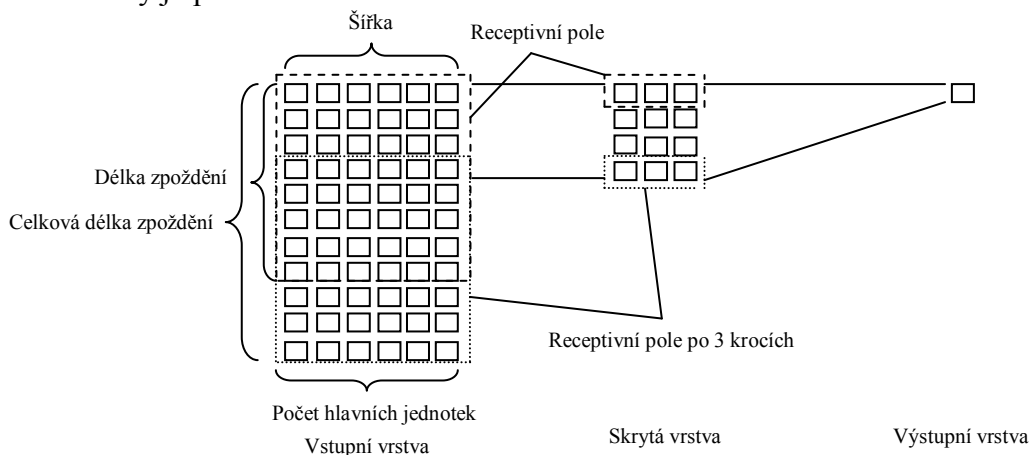
Návrh neuronové sítě

Po aktivaci funkce **BIGNET** z panelu **SNNS Manager** a vybráním nabídky **TIME-DELAY** je zadávána struktura NS. V horní části okna na obr. 3 je definována struktura NS. Do volných míst jsou zadávány parametry NS a jak mají být zobrazeny na obrazovce. Ve spodní části okna se propojují jednotlivé vrstvy NS. Nejprve je definována vstupní vrstva vyplněním volných míst vpravo nahoře v části **Edit Plane** (struktura vrstvy). Vstupní vrstva je definována skupinou parametrů NS uspořádaných způsobem značeným na obr. 4.



Obr. 3 Okno pro návrh time-delay neuronové sítě

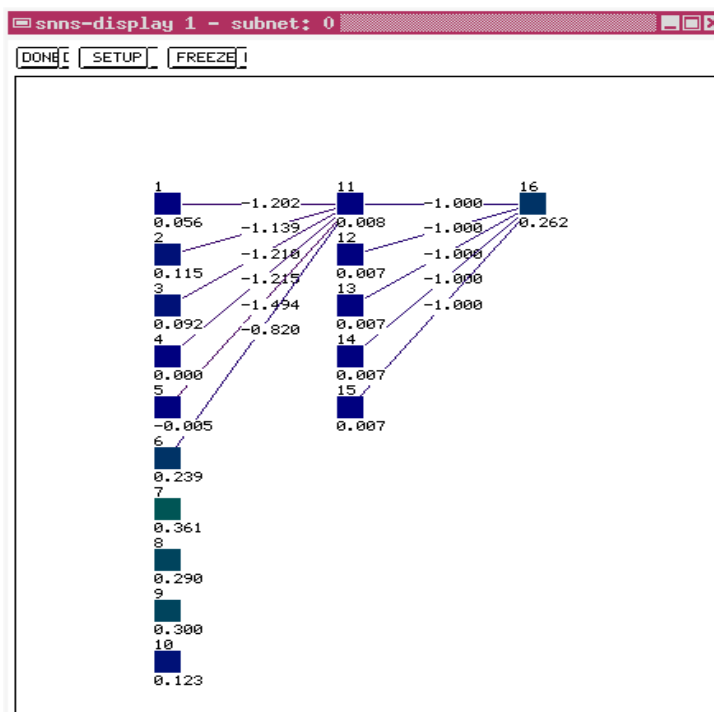
Má-li mít např. vstupní vrstva 66 neuronů, do kolonky *No. of feature units*: je zapsána 6, do kolonky *Total delay length*: 11 a do kolonky *Z-coordinates of the plane*: 0. Struktura vrstvy je potvrzena tlačítkem **ENTER**.



Obr. 4 Názvosloví TDDN v SNNS

Stejně jako vstupní vrstva jsou editovány také skrytá vrstva (*No. of feature units*: 3, *Total delay length*: 4, *Z-coordinates of the plane*: 0) a výstupní vrstva (*No. of feature units*: 1, *Total delay length*: 1, *Z-coordinates of the plane*: 0). Typ vrstvy se mění pomocí tlačítka **TYPE**, struktura vrstvy je uložena aktivováním tlačítka **ENTER**. Tlačítkem **POS** je nastaveno uspořádání NS v grafickém vyjádření. Neuronová síť může být uspořádána doprava (*right*), doleva (*left*) nebo dolů (*below*).

Po definování struktury jednotlivých vrstev se ve spodní části – Link editoru musí ještě jednotlivé vrstvy NS propojit. V prázdných pozicích pro *Plane* v *Edit Linku* se do *Source* zadává číslo výchozí vrstvy (pro vstupní=1, skrytá=2, výstupní=3) a do *Target* číslo skryté vrstvy. Do kolonky *Ist feat. unit:* se zadává souřadnice [1,1], do kolonky *width* se zadává šířka NS, z obrázku počet *Feature Unit* a do *Delay Length* se zadává délka zpoždění znázorněná na obr 4. Pro příklad NS při propojování vstupní a skryté vrstvy se vloží do *Source* 1, do *Target* 2, do dvojice políček *Ist feat. unit:* 1 a 1, *width* 6 a 3, *Delay Length* je 8 a následně se potvrdí **ENTER**. Poté je nutné opakovat stejný postup ještě pro skrytou a výstupní vrstvu. Do políčka *Source* 2, do *Target* 3, do dvojice políček *Ist feat. unit:* 1 a 1, *width* 3 a 1, *Delay Length* se zadá 4 a potvrdí se **ENTER**. Pro konečné vytvoření sítě je nutné stisknout tlačítko **CREATE TD_NET**. Tím je vytvořeno propojení pro všechny vrstvy. Struktura NS je zobrazena pomocí nabídky **DISPLAY** z panelu **SNNS Manager**. Ukázka tohoto okna je na obr. 5.



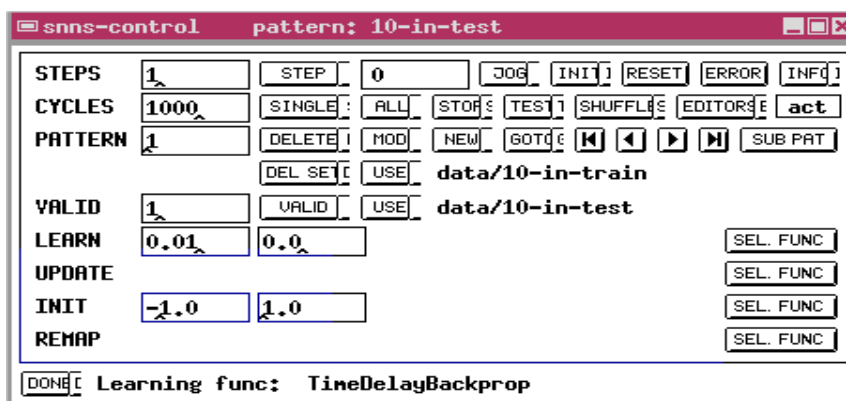
Obr.5 Struktura neuronové sítě

Neurony jsou propojeny stiskem tlačítka **SETUP** v okně **DISPLAY**, kde je nabídka **links** aktivována tlačítkem **ON**. Práce s oknem je ukončena pomocí **DONE**. Je-li

vytvořen návrh struktury NS a načteny soubory obsahující trénovací a kontrolní data, je možné přejít k učení NS.

Učení neuronové sítě

Učení NS je uskutečněno pomocí nabídky **CONTROL**, která je aktivována z panelu **SNNS Manager**. Okno pro učení NS se skládá ze dvou částí. V horní části jsou definovány parametry pro učení NS, ve spodní části jsou čtyři řádky s volnými místy pro vyplnění parametrů učení a intervalu, v kterém budou váhy náhodně rozloženy při inicializaci NS. Přednastavené hodnoty parametrů učení jsou (0.2 0) a přednastavená hodnota váhy je v intervalu (1.0 -1.0). Ukázka okna pro učení NS je na obr. 6.



Obr. 6 Okno pro trénování a testování neuronové sítě

Většina NS je před učením inicializována. Inicializace je aktivována pomocí tlačítka **INIT** v horním řádku. Interval, ve kterém budou náhodně rozloženy hodnoty při inicializaci, je možné změnit ve spodní části okna ve volných místech vedle názvu **INIT**. Pro učení frontální NS je většinou používán algoritmus *TimeDelayBackprop*. Algoritmus učení je možné vybrat aktivováním horního tlačítka **SEL FUNC**. Druhé tlačítko **SEL. FUNC**. slouží k výběru aktualizací funkce (*Update Function*), pomocí které jsou neurony NS používány postupně. Pořadí neuronů závisí na vybrané aktualizací funkci. Aktualizací funkcí pro Time-Delay neuronové sítě je *TimeDelay_Order*. Pomocí třetího tlačítka **SEL. FUNC**. je vybrána inicializací funkce, která slouží k nastavení počátečních hodnot (vah). Z inicializací funkcí je nejčastěji používána funkce *Randomize_Weights*. Čtvrtým tlačítkem **SEL. FUNC**. je vybrána funkce, pomocí které je změněn požadovaný výstup bez zásahu do trénovacího a kontrolního souboru (tzn. do souborů obsahujících vzory). Je možné použít funkci *Threshold*, popřípadě i jiné.

Učení NS je uskutečněno pro řadu cyklů, množství cyklů se nastaví v horní části vedle názvu **CYCLES**. Všechny vstupní vzory (spojení vstup – výstup) jsou použity v každém cyklu. Někdy je dáována přednost náhodnému použití vzorů, tento náhodný výběr je proveden aktivováním tlačítka **SHUFFLE**. Před spuštěním učícího procesu je možné aktivovat panel **GRAPH**, v kterém je zobrazena chyba procesu učení. V panelu **GRAPH** musíme ještě změnit v kolonce **Display: SSE** na **MSE**. Proces učení je aktivován tlačítkem **ALL** a ukončen pomocí tlačítka **STOP**.

Tlačítko **USE** slouží k výběru trénovacích (**horní tlačítko USE**) a kontrolních (validačních) dat (**dolní tlačítko USE**). Pokud je napsána nenulová hodnota vedle názvu **VALID**, budou použita kontrolní data a chyba použití těchto dat bude v grafu znázorněna červenou barvou, pokud bude napsána nula, budou kontrolní data vyřazena. Pomocí tlačítka **TEST** je možné si postupně prohlédnout jednotlivé vzory (vstupní a výstupní hodnoty). Parametry **LEARN** slouží k nastavení učení. Čím vyšší číslo je u prvního parametru tím lépe se NS učí. Druhý parametr udává s jakou chybou se má NS učit. Hodnoty určitého vzoru je možno si prohlédnout dosazením čísla vzoru do volného místa vedle názvu **PATTERN** a aktivováním tlačítka **GOTO** nebo použitím šipek pro posun vedle tlačítka **GOTO**. Hodnoty vzorů se zobrazí na panelu **DISPLAY**. Hodnotami na výstupu jsou v tomto programovém systému *targets* (empirické výstupy).

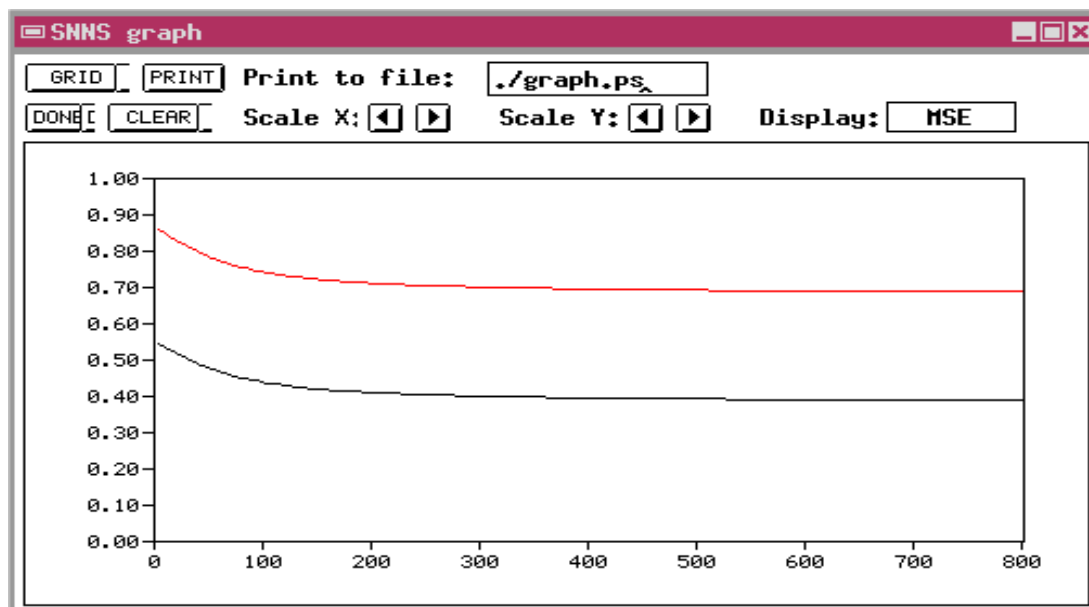
Zobrazení chyby učení neuronové sítě

Pomocí nabídky **GRAPH**, která je aktivována z panelu **SNNS Manager**, je možné zobrazit průběh chyby při učení NS na trénovacích i kontrolních datech. Ukázka okna s průběhem chyby je na obr. 7. Na x-ové ose je počet cyklů, na y-ové pak velikost chyby. Aktivováním nabídky **CLEAR** je graf vyčištěn pro další návrh NS. Šípkami je možné měnit měřítko x-ové a y-ové souřadnice. Pomocí nabídky na pravé straně (kde je na obr. napsáno **MSE**) se mění způsob zobrazení chyby.

System nabízí tyto možnosti:

- **SSE (celková čtvercová chyba),**
- **MSE (střední čtvercová chyba),**
- **SSE/out (celková čtvercová chyba / počet výstupů).**

Práce s oknem je ukončena aktivováním nabídky **DONE**.



Obr. 7 Průběh chyby při učení neuronové sítě

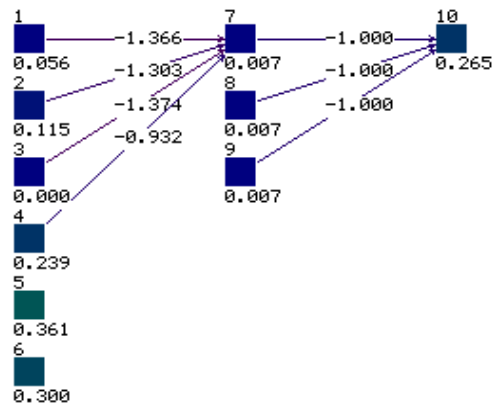
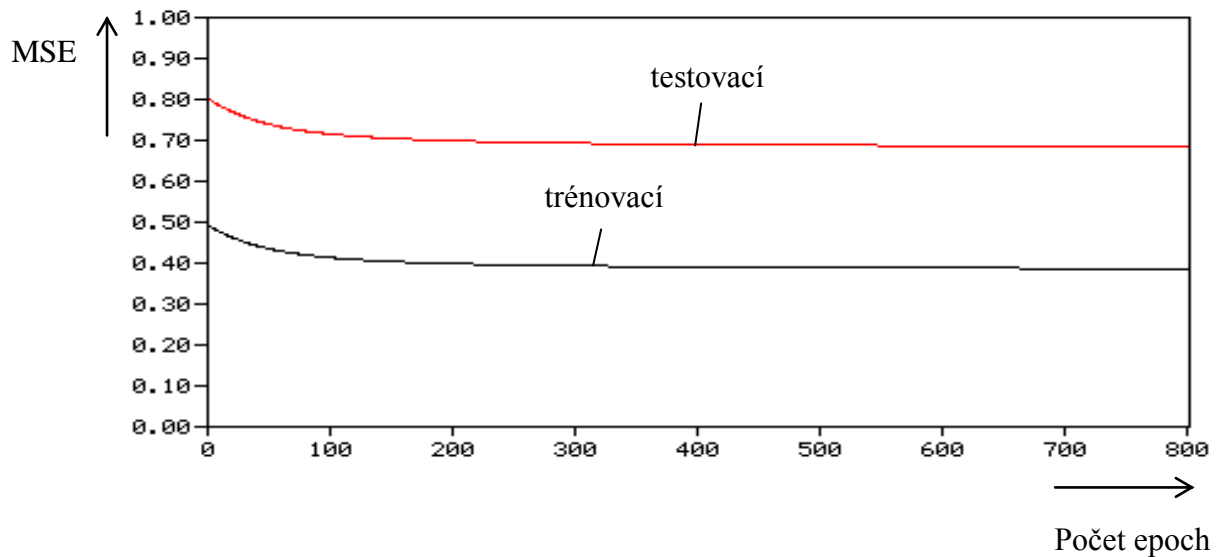
Uložení výsledků

Výsledky navrženého modelu lze uložit pomocí nabídky **FILE** na panelu **SNNS Manager** a aktivováním tlačítka **RES** a pravé části okna **FILE**. Název souboru, do kterého mají být výsledky uloženy, je zapsán do volného místa v pravé horní části okna **FILE**. Výběr adresáře pro uložení souboru je proveden v levé části tohoto okna. Aktivováním tlačítka **SAVE** se objeví okno, které je znázorněné na obr. 8. Stiskem tlačítka **DONE** v tomto okně se výsledky uloží. (Výsledky v jiném formátu se ukládají aktivováním nabídky **NET** v okně **FILE**).

Obr. 8 Uložení výsledků

Příloha 2: Protokoly o způsobu naučení vybraných modelů neuronových sítí, jejich nastavení vah synapsí a prahů neuronů

Model s filtrem 4 a strukturou neuronové sítě 6-3-1



Remap. func: Binary
 Init. func: Randomize_Weights
 Update func: TimeDelay_Order
 Learning func: TimeDelayBackprop

Learning all patterns:
 epochs : 1000
 parameter: 0.01000
 #o-units : 1
 #patterns: 192 (total: 192)

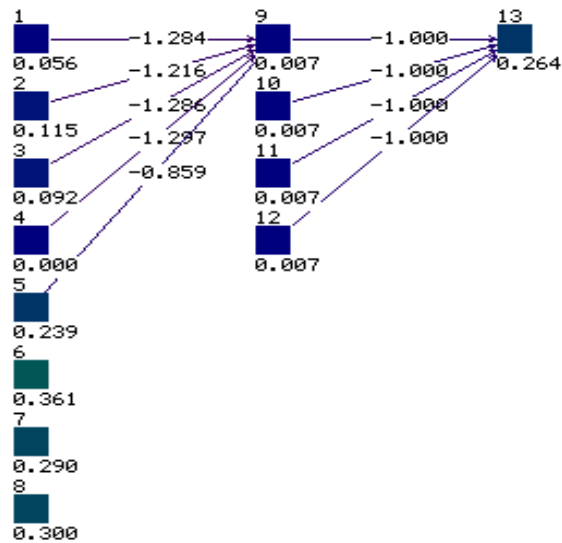
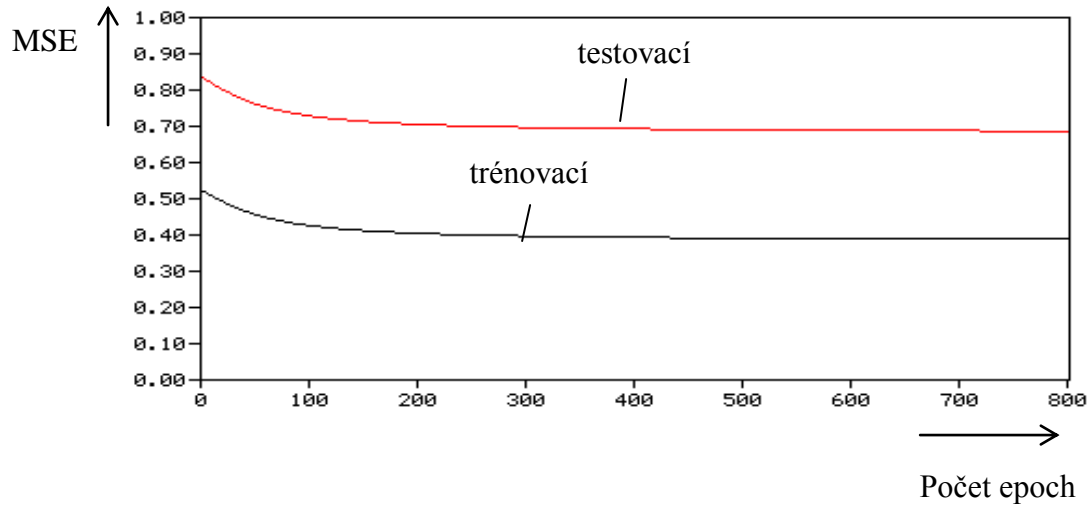
epoch:	SSE	MSE	SSE/o-units
Train 1000:	93.82813	0.48869	93.82813
Test 1000:	137.42377	0.79898	137.42377
Train 900:	78.90276	0.41095	78.90276

Test 900:	122.61192	0.71286	122.61192
Train 800:	76.07226	0.39621	76.07226
Test 800:	119.79638	0.69649	119.79638
Train 700:	75.04602	0.39086	75.04602
Test 700:	118.76523	0.69050	118.76523
Train 600:	74.52888	0.38817	74.52888
Test 600:	118.24168	0.68745	118.24168
Train 500:	74.21975	0.38656	74.21975
Test 500:	117.92697	0.68562	117.92697
Train 400:	74.01467	0.38549	74.01467
Test 400:	117.71741	0.68440	117.71741
Train 300:	73.86893	0.38473	73.86893
Test 300:	117.56807	0.68354	117.56807
Train 200:	73.75993	0.38417	73.75993
Test 200:	117.45612	0.68288	117.45612
Train 100:	73.67555	0.38373	73.67555
Test 100:	117.36944	0.68238	117.36944
Train 1:	73.60884	0.38338	73.60884
Test 1:	117.30075	0.68198	117.30075

----- s t a t i s t i c s -----

Number of Patterns :	192	
Number of parameters (Links+bias) :	11	
sse :	73.6085	
tss :	42.6667	
rsq :	-0.7252	
mse :	0.4067	
rmse :	0.6377	
adjrsq :	-0.8205	
J_p :	0.4300	
gcv :	0.4314	
pc :	-0.9349	
S_p :	0.0023	
gmsep :	0.4338	
shibata :	82.0428	
aic :	-162.0770	
sbc :	-126.2446	

Model s filtrem 5 a strukturou neuronové sítě 8-4-1



Remap. func: Binary
 Init. func: Randomize_Weights
 Update func: TimeDelay_Order
 Learning func: TimeDelayBackprop

Learning all patterns:
 epochs : 1000
 parameter: 0.01000
 #o-units : 1
 #patterns: 192 (total: 192)

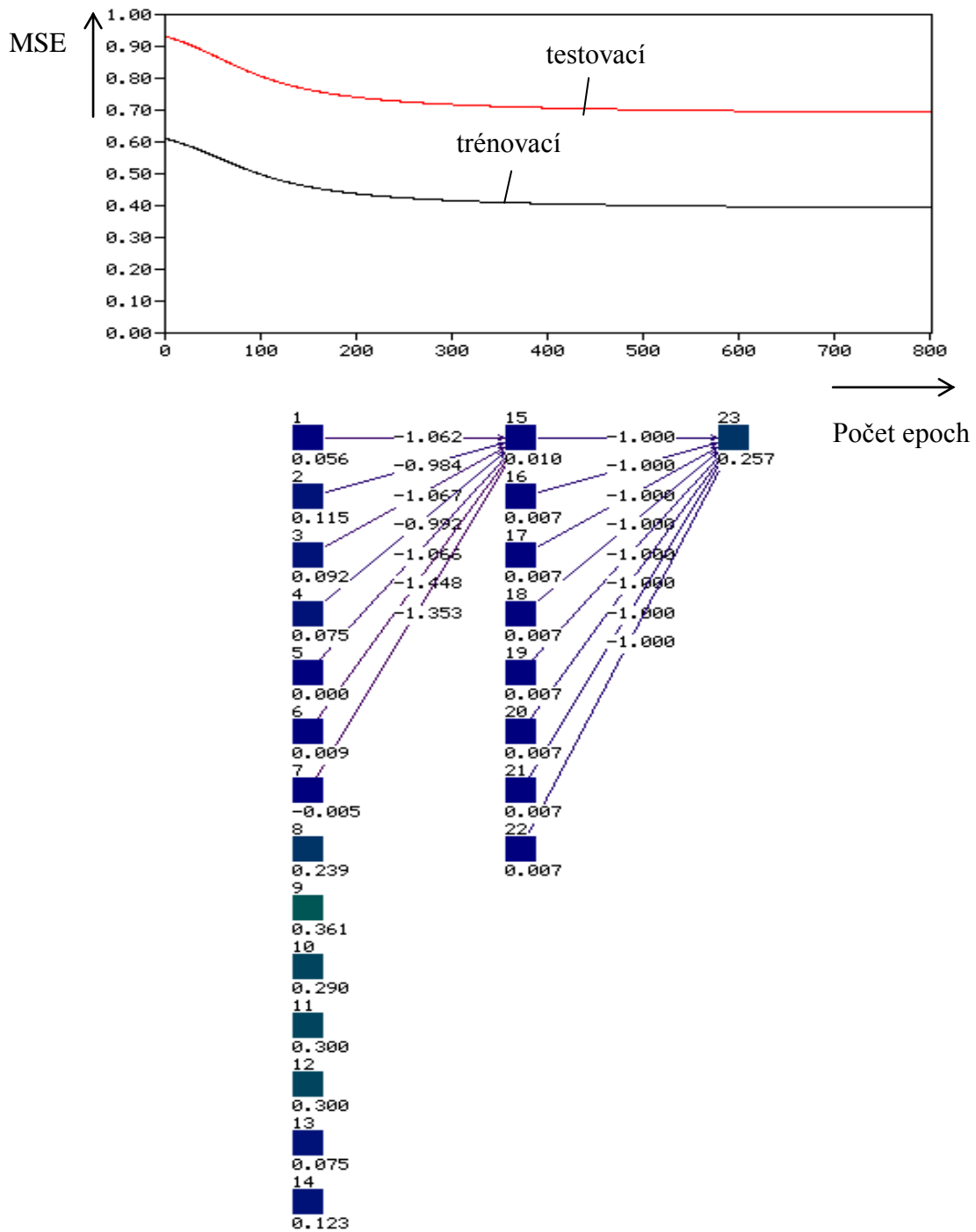
epoch:	SSE	MSE	SSE/o-units
Train 1000:	100.02796	0.52098	100.02796
Test 1000:	143.42998	0.83390	143.42998
Train 900:	81.27164	0.42329	81.27164
Test 900:	124.87063	0.72599	124.87063

Train 800:	77.20428	0.40211	77.20428
Test 800:	120.89767	0.70289	120.89767
Train 700:	75.75413	0.39455	75.75413
Test 700:	119.47054	0.69460	119.47054
Train 600:	75.03442	0.39080	75.03442
Test 600:	118.75633	0.69044	118.75633
Train 500:	74.60809	0.38858	74.60809
Test 500:	118.33096	0.68797	118.33096
Train 400:	74.32693	0.38712	74.32693
Test 400:	118.04931	0.68633	118.04931
Train 300:	74.12766	0.38608	74.12766
Test 300:	117.84939	0.68517	117.84939
Train 200:	73.97884	0.38531	73.97884
Test 200:	117.70010	0.68430	117.70010
Train 100:	73.86351	0.38471	73.86351
Test 100:	117.58466	0.68363	117.58466
Train 1:	73.77205	0.38423	73.77205
Test 1:	117.49330	0.68310	117.49330

----- s t a t i s t i c s -----

Number of Patterns :	192	
Number of parameters (Links+bias)	:	14
sse	:	73.7716
tss	:	42.6667
rsq	:	-0.7290
mse	:	0.4144
rmse	:	0.6438
adjrsq	:	-0.8553
J_p	:	0.4447
gcv	:	0.4470
pc	:	-1.0010
S_p	:	0.0023
gmsep	:	0.4496
shibata	:	84.5300
aic	:	-155.6521
sbc	:	-110.0472

Model s filtrem 7 a strukturou neuronové sítě 14-8-1



Learning func: TimeDelayBackprop
 Update func: TimeDelay_Order
 Init. func: Randomize_Weights
 Remap. func: Binary

Learning all patterns:
 epochs : 1000
 parameter: 0.01000

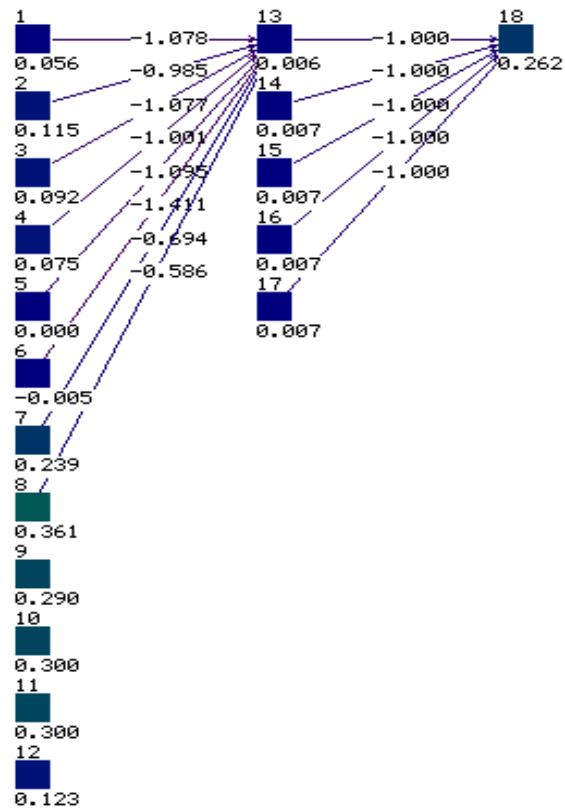
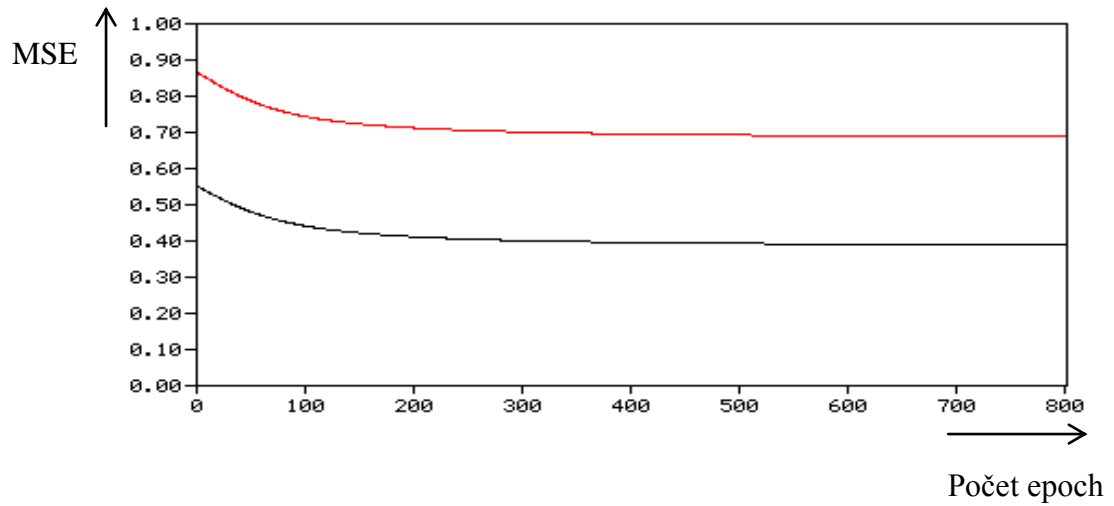
#o-units : 1
 #patterns: 192 (total: 192)

epoch:	SSE	MSE	SSE/o-units
Train 1000:	116.75684	0.60811	116.75684
Test 1000:	159.85307	0.92938	159.85307
Train 900:	95.07242	0.49517	95.07242
Test 900:	138.20157	0.80350	138.20157
Train 800:	83.49172	0.43485	83.49172
Test 800:	126.90978	0.73785	126.90978
Train 700:	79.39062	0.41349	79.39062
Test 700:	122.98412	0.71502	122.98412
Train 600:	77.49867	0.40364	77.49867
Test 600:	121.17383	0.70450	121.17383
Train 500:	76.43773	0.39811	76.43773
Test 500:	120.15403	0.69857	120.15403
Train 400:	75.76406	0.39460	75.76406
Test 400:	119.50374	0.69479	119.50374
Train 300:	75.29947	0.39218	75.29947
Test 300:	119.05399	0.69217	119.05399
Train 200:	74.95964	0.39041	74.95964
Test 200:	118.72462	0.69026	118.72462
Train 100:	74.69987	0.38906	74.69987
Test 100:	118.47293	0.68880	118.47293
Train 1:	74.49705	0.38801	74.49705
Test 1:	118.27661	0.68765	118.27661

----- s t a t i s t i c s -----

Number of Patterns : 192
 Number of parameters (Links+bias) : 24
 sse : 74.4961
 tss : 42.6667
 rsq : -0.7460
 mse : 0.4434
 rmse : 0.6659
 adjrsq : -0.9850
 J_p : 0.4989
 gcv : 0.5068
 pc : -1.2449
 S_p : 0.0027
 gmsep : 0.5098
 shibata : 93.1201
 aic : -133.7758
 sbc : -55.5959

Model s filtrem 8 a strukturou neuronové sítě 12-5-1



Learning func: TimeDelayBackprop
 Update func: TimeDelay_Order
 Init. func: Randomize_Weights
 Remap. func: Binary

Learning all patterns:
 epochs : 1000
 parameter: 0.01000

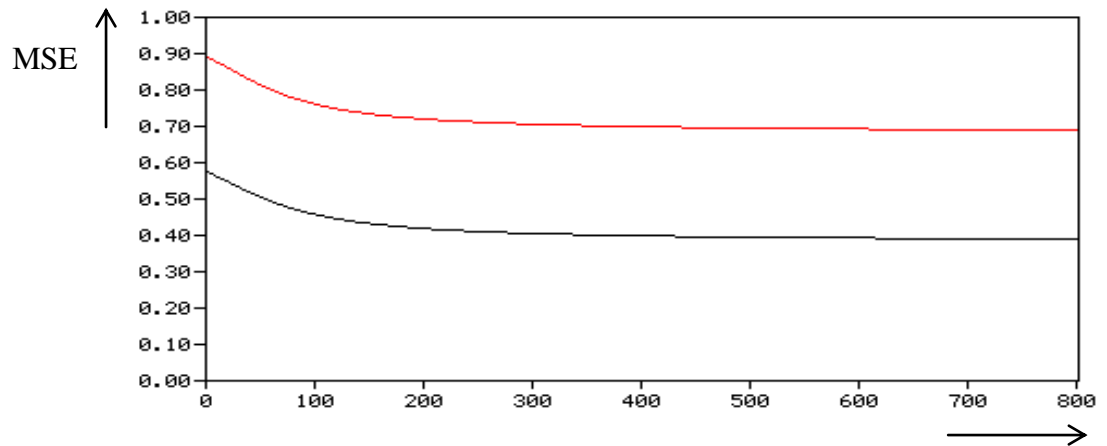
#o-units : 1
 #patterns: 192 (total: 192)

epoch:	SSE	MSE	SSE/o-units
Train 1000:	105.30968	0.54849	105.30968
Test 1000:	148.44171	0.86303	148.44171
Train 900:	84.24190	0.43876	84.24190
Test 900:	127.48769	0.74121	127.48769
Train 800:	78.53706	0.40905	78.53706
Test 800:	122.09148	0.70983	122.09148
Train 700:	76.51503	0.39852	76.51503
Test 700:	120.19319	0.69880	120.19319
Train 600:	75.54113	0.39344	75.54113
Test 600:	119.26583	0.69341	119.26583
Train 500:	74.97677	0.39050	74.97677
Test 500:	118.72163	0.69024	118.72163
Train 400:	74.60973	0.38859	74.60973
Test 400:	118.36488	0.68817	118.36488
Train 300:	74.35181	0.38725	74.35181
Test 300:	118.11324	0.68670	118.11324
Train 200:	74.16009	0.38625	74.16009
Test 200:	117.92609	0.68562	117.92609
Train 100:	74.01163	0.38548	74.01163
Test 100:	117.78155	0.68478	117.78155
Train 1:	73.89388	0.38486	73.89388
Test 1:	117.66737	0.68411	117.66737

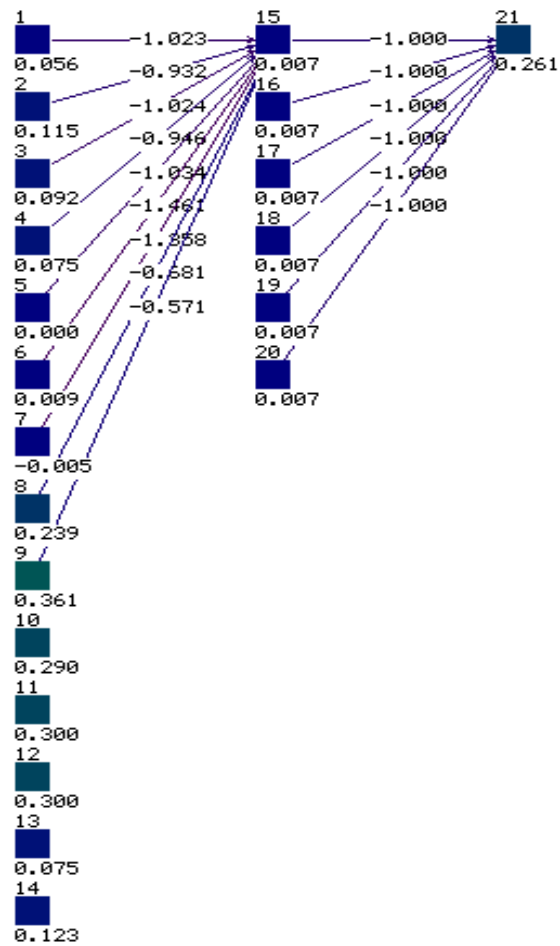
----- s t a t i s t i c s -----

Number of Patterns : 192
 Number of parameters (Links+bias) : 19
 sse : 73.8933
 tss : 42.6667
 rsq : -0.7319
 mse : 0.4271
 rmse : 0.6536
 adjrsq : -0.9121
 J_p : 0.4694
 gcv : 0.4740
 pc : -1.1123
 S_p : 0.0025
 gmsep : 0.4768
 shibata : 88.5180
 aic : -145.3356
 sbc : -83.4432

Model s filtrem 9 a strukturou neuronové sítě 14-6-1



Počet epoch



Learning func: TimeDelayBackprop
 Update func: TimeDelay_Order
 Init. func: Randomize_Weights
 Remap. func: Binary

Learning all patterns:
 epochs : 1000

parameter: 0.01000
 #o-units : 1
 #patterns: 192 (total: 192)

epoch:	SSE	MSE	SSE/o-units
Train 1000:	109.81858	0.57197	109.81858
Test 1000:	152.86711	0.88876	152.86711
Train 900:	87.28709	0.45462	87.28709
Test 900:	130.40454	0.75817	130.40454
Train 800:	79.93624	0.41633	79.93624
Test 800:	123.42355	0.71758	123.42355
Train 700:	77.33407	0.40278	77.33407
Test 700:	120.99432	0.70346	120.99432
Train 600:	76.09761	0.39634	76.09761
Test 600:	119.82705	0.69667	119.82705
Train 500:	75.38908	0.39265	75.38908
Test 500:	119.14973	0.69273	119.14973
Train 400:	74.93190	0.39027	74.93190
Test 400:	118.70883	0.69017	118.70883
Train 300:	74.61240	0.38861	74.61240
Test 300:	118.39943	0.68837	118.39943
Train 200:	74.37606	0.38738	74.37606
Test 200:	118.17039	0.68704	118.17039
Train 100:	74.19363	0.38643	74.19363
Test 100:	117.99371	0.68601	117.99371
Train 1:	74.04917	0.38567	74.04917
Test 1:	117.85426	0.68520	117.85426

----- s t a t i s t i c s -----

Number of Patterns : 192
 Number of parameters (Links+bias) : 22
 sse : 74.0485
 tss : 42.6667
 rsq : -0.7355
 mse : 0.4356
 rmse : 0.6600
 adjrsq : -0.9499
 J_p : 0.4855
 gcv : 0.4919
 pc : -1.1847
 S_p : 0.0026
 gmsep : 0.4948
 shibata : 91.0179
 aic : -138.9329
 sbc : -67.2680

Údaje pro knihovnickou databázi

Název práce	Modelování hrubého domácího produktu pomocí frontálních neuronových sítí
Autor práce	Marek Slavík
Obor	Informatika ve veřejné správě
Rok obhajoby	2007
Vedoucí práce	prof. Ing. Vladimír Olej, CSc.
Anotace	<p>Diplomová práce je zaměřena na návrh modelu frontální neuronové sítě pro predikci hrubého domácího produktu. Pro učení neuronové sítě je použit index vedoucích ekonomických indikátorů a jeho difúzní index, kterým odpovídá procentuální změna HDP.</p> <p>V první části diplomové práce je řešen způsob modelování trendových složek těchto indexů pomocí klouzavých průměrů a exponenciálních vyrovnání. Je zde definována frontální neuronová síť a uvedeny další neuronové sítě vhodné pro predikci. V druhé části jsou analyzovány navržené modely frontálních neuronových sítí pomocí Stuttgartského simulátoru neuronových sítí.</p>
Klíčová slova	predikce HDP, IVEI, DIVEI, časová řada, neuronové sítě vhodné pro predikci, frontální neuronové sítě, Back-Propagation, SNNS