

Univerzita Pardubice
Fakulta ekonomicko - správní
Ústav systémového inženýrství a informatiky

Modelování dat charakterizujících virtuální server
pomocí RBF neuronových sítí

Jaroslav Kyncl

Diplomová práce
2012

University of Pardubice
Faculty of Economics and Administration
Institute of System Engineering and Informatics

**Modelling of Data Characterizing a Virtual Server by
RBF Neural Networks**

Jaroslav Kyncl

Thesis
2012

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Bc. Jaroslav Kyncl
Osobní číslo: E100429
Studijní program: N6209 Systémové inženýrství a informatika
Studijní obor: Informatika ve veřejné správě
Název tématu: Modelování dat charakterizujících virtuální server pomocí RBF neuronových sítí
Zadávací katedra: Ústav systémového inženýrství a informatiky

Zásady pro vypracování:

Analyzujte data virtuálního serveru.
Charakterizujte RBF neuronové sítě.
Navrhněte model pro klasifikaci.
Ověřte navržený model.
Uskutečňte analýzu výsledků.

Rozsah grafických prací:

Rozsah pracovní zprávy: cca 60 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

- Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall Inc., New Jersey (1999)
- Kvasnička, V. at al.: Introduction to Neural Networks. Iris, Bratislava (1997) (in Slovak)
- Broomhead, D. S.; Lowe, D.: Multivariate Functional Interpolation and Adaptive Networks. Complex Systems, vol.2, 321-355 (1988)
- Park, J.; Sandberg, I. W.: Universal Approximation using Radial Basis Function Network. Neural Comput., vol.3, 246-257 (1991)
- Park, J.; Sandberg, I. W.: Approximation and Radial Basis Function Network. Neural Comput., vol.5, 305-316 (1993)
- Olej, V.; Hájek, P.: Municipal Creditworthiness Modelling by Radial Basis Function Neural Networks and Sensitive Analysis their Inputs Parameters. 19th International Conference on Artificial Neural Network, ICANN 2009, 14-17 September, Limassol, Cyprus, Alippi, C., Polycarpou, M., Panayiotou, Ch., Ellinas, G., Eds., Springer Berlin Heidelberg New York, 2009, pp.505-514.

Vedoucí diplomové práce:

prof. Ing. Vladimír Olej, CSc.

Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: 3. října 2011

Termín odevzdání diplomové práce: 30. dubna 2012

doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.

doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

dne

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci 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 30. 4. 2012

Bc. Jaroslav Kyncl

PODĚKOVÁNÍ

Tímto bych rád poděkoval především vedoucímu práce panu prof. Ing. Vladimíru Olejovi, CSc. za jeho odborné rady, věcné připomínky, vstřícný přístup a vše, co mi pomohlo při zpracování diplomové práce. Poděkování patří také celé mé rodině za trpělivost a podporu po celou dobu studia.

ANOTACE

Diplomová práce se zabývá návrhem modelu pro klasifikaci dat naměřených na virtuálních serverech Univerzity Pardubice, konkrétně na webovém serveru Portal a databázovém serveru Oracle. Cílem předložené práce je s co nejvyšší přesností klasifikovat data do tříd, které reprezentují možné chybové stavy serverů. Klasifikace je provedena pomocí neuronových sítí typu RBF, metodou učení s učitelem. Na začátku práce jsou charakterizovány pojmy virtualizace, virtualizace serverů a její možné metody. Dále jsou popsány základní pojmy neuronových sítí, metody učení neuronových sítí a neuronové sítě typu RBF. Další kapitola se zabývá návrhem modelu pro klasifikaci. V první části návrhu jsou zanalyzovaná a předzpracovaná poskytnutá data, mezi kterými je také zjišťována možná existence korelační závislosti. Poté následují experimenty, jejichž cílem je zjištění optimální struktury a parametrů navrhované neuronové sítě. K těmto experimentům je využito programového prostředí SPSS Clementine. Na závěr je uskutečněna analýza dosažených výsledků klasifikace a ověření navrženého modelu RBF neuronové sítě.

KLÍČOVÁ SLOVA

Virtualizace, virtuální server, neuronové sítě, radiálně bazické funkce, RBF, klasifikace.

ANNOTATION

This thesis describes the design of model for the classification of measured data on virtual servers of University of Pardubice, specifically on web server Portal and Oracle database server. The aim of this work is to classify data with the highest possible precision into classes representing possible servers' errors. Classification is done using RBF neural networks, by method of learning with the teacher. The first part presents terms virtualization, server virtualization and its methods. The following part describes the basic concepts of neural networks, learning methods of neural networks and RBF neural network type. The next chapter deals with the classification design model. In the first part of the proposal there are analyzed and pre-processed provided data, among which is also detected the possible existence of correlation dependence. The following experiments are aimed on finding the optimal structure and parameters for proposed neural networks. SPSS Clementine software is used for these experiments. Analysis of classification results and verification of designed RBF neural network is presented in the final part.

KEYWORDS

Virtualization, Virtual Server, Neural Networks, Radial Basic Function, RBF, Classification.

OBSAH

ÚVOD.....	10
1 Virtualizace	11
1.1 Základní metody virtualizace.....	13
1.1.1 Hardwarová virtualizace	13
1.1.2 Emulace nebo simulace.....	14
1.1.3 Nativní virtualizace a plná virtualizace.....	14
1.1.4 Částečná virtualizace.....	15
1.1.5 Paravirtualizace	15
1.1.6 Virtualizace na úrovni operačního systému	15
1.1.7 Aplikační virtualizace	16
1.2 Principy virtualizace	16
1.3 Dílčí závěr.....	17
2 Neuronové sítě.....	18
2.1 Neuron	18
2.2 Neuronová síť	19
2.3 Učení neuronové sítě	19
2.4 Struktury neuronové sítě.....	21
2.4.1 Dopředná neuronová síť.....	21
2.4.2 Rekurentní neuronová síť.....	22
2.5 Oblasti využití neuronových sítí	23
2.6 Všeobecný klasifikační problém.....	24
2.7 Dílčí závěr.....	25
3 Neuronové sítě typu RBF.....	26
3.1 Struktura neuronových sítí typu RBF	26
3.2 Učení neuronové sítě typu RBF.....	28

3.2.1	Centra RBF neuronů	29
3.2.2	Počet RBF neuronů ve skryté vrstvě.....	29
3.3	Dílčí závěr.....	32
4	Návrh modelu klasifikace dat pomocí RBF neuronových sítí	33
4.1	Návrh modelu	33
4.2	Analýza dat	34
4.3	Korelace	39
4.4	Předzpracování dat.....	41
4.5	Návrh RBF neuronové sítě	43
4.5.1	Počet neuronů ve skryté vrstvě	48
4.5.2	Počet učicích cyklů	49
4.5.3	Vliv momentu Alpha.....	50
4.5.4	Vliv parametru překrytí.....	51
4.6	Dílčí závěr.....	53
5	Analýza výsledků.....	54
5.1	Navržená RBF neuronová síť'	54
5.2	Výsledky klasifikace.....	56
5.3	Ověření navrženého modelu neuronové sítě.....	58
5.4	Dílčí závěr.....	60
	ZÁVĚR.....	61
	POUŽITÁ LITERATURA	62
	SEZNAM OBRÁZKŮ.....	64
	SEZNAM TABULEK	66
	SEZNAM ZKRATEK	67

ÚVOD

Virtualizace serverů se v dnešní době stává běžnou součástí řešení informačních technologií stále většího počtu organizací. Diplomová práce se zabývá modelováním dat charakterizujících virtuální webový server Portal a databázový server Oracle Univerzity Pardubice. Cílem práce je navrhnout model, který bude na základě naměřených atributů těchto serverů provádět klasifikaci chybových stavů vyskytujících se během jejich provozu. Klasifikace je provedena pomocí neuronových sítí typu RBF, metodou učením s učitelem.

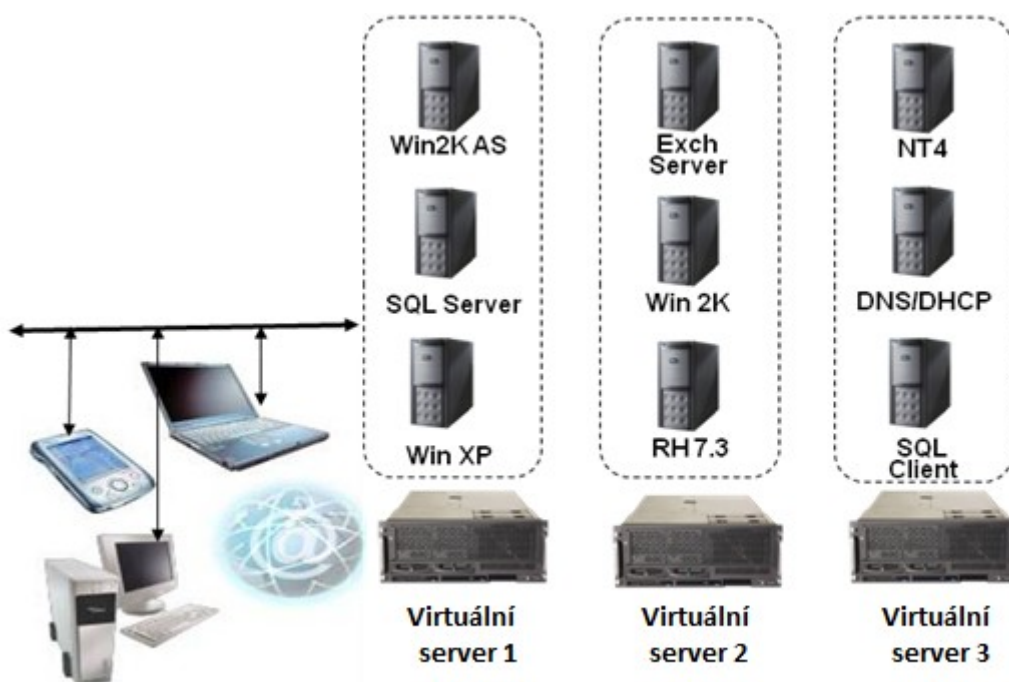
První kapitola vymezuje pojem virtualizace z obecného hlediska. Poté je konkretizována virtualizace serverů a jsou popsány možné druhy této virtualizace. Další kapitola definuje neuronové sítě a její struktury. Je charakterizována základní vlastnost neuronové sítě, kterou je učení, jež lze provést induktivní nebo deduktivní metodou, a to formou jak s učitelem, tak bez učitele. Následně jsou popsány neuronové sítě typu RBF.

Návrh modelu pro klasifikaci je tvořen několika kroky. Nejprve jsou získaná data sloučena do jednoho souboru, analyzována a následně je vytvořena datová matice, která slouží jako vstupní do navrhovaného modelu. Poté je zapotřebí vytvořit korelační matici sloužící k odhalení možné závislosti mezi jednotlivými atributy. Použitá data je ještě nutné předzpracovat do vyhovující podoby pro potřeby modelování.

S připravenými daty je již možné přejít k provádění jednotlivých experimentů, které vedou k získání optimální struktury neuronové sítě a jejich parametrů. Výstupem práce je tedy navržený model neuronové sítě klasifikující naměřená data na virtuálních serverech do tříd, s co nejvyšší přesností. Tyto třídy reprezentují možné chybové stavy serverů. Následně je zapotřebí získané výsledky analyzovat a provést ověření navržené neuronové sítě za pomoci využití validační množiny dat.

1 Virtualizace

Pojem virtualizace je v prostředí informačních technologií běžně chápán jako uspořádání, ve kterém je přístup k systémovým zdrojům možný jako k množině výkonu bez ohledu na jejich fyzické parametry, na základě nichž k nim uživatelé obvykle přistupují. Označení server již tedy není omezeno svou fyzickou podobou (počítač), ale lze jej charakterizovat jako skupinu dostupných zdrojů. Virtualizace poskytuje možnost provozu více virtuálních serverů na jednom fyzickém serveru [1]. Příklad virtualizace serverů je uveden na obr. 1.



Obr. 1 – Ukázka virtualizace serverů, zdroj: [2]

Obr. 1 znázorňuje tři varianty virtualizace serverů. Na každém z těchto serverů jsou provozovány tři virtuální servery, které jsou v obr. 1 označeny konkrétním názvem služby či operačního systému. Jedná se například o operační systém Win XP či službu DNS/DHCP nebo SQL Server.

Kořeny virtualizace lze nalézt již v 60. letech 20. století jako způsob lepšího využití velkého sálového počítače. Jde o tendenci vytváření celých virtuálních strojů založených na kombinaci technik programových a technických prostředků. V ideálním případě by pak jeden fyzický stroj mohl obsluhovat řádově až stovky virtuálních strojů. Vývojem softwarové vrstvy mezi fyzickými prostředky a virtualizovaným operačním systémem

dochází ke kombinaci sdílení systémových prostředků a současně k izolovanosti jednotlivých virtuálních prostředí mezi sebou [3].

Virtualizace nabízí dvě základní řešení:

- virtuální server (Virtual Server) – pod jedním operačním systémem je v provozu více od sebe oddělených serverů;
- virtuální počítač (Virtual Machine) - na jednom počítači je současně provozováno více operačních systémů, program koordinující a spravující tyto systémy nese název virtuální monitor (Virtual Monitor) [4].

K provozu virtuálních strojů je nezbytné vyřešit celou řadu úkolů. Samotná virtualizační soustava musí mít plánovač, který dokáže spravovat požadavky operačních systémů. Zajišťuje, aby se činnosti jednotlivých operačních systémů vzájemně neovlivňovaly. Z hlediska virtualizačního nástroje dochází k přepínání kontextu celých operačních systémů. Algoritmus plánování přístupu k technickým prostředkům zabezpečuje vícenásobný přístup k systémovým mechanismům. Další úkol spočívá v řešení komunikace mezi virtuálními stroji. Jedná se například o bezchybné doručování příchozích paketů z externí sítě [3].

Při hardwarové virtualizaci je vytvořen za použití vhodných programových prostředků virtuální počítač, ten dokáže emulovat fyzický počítač. Dochází tak ke vzniku odděleného prostředí operačního systému izolovaného od hostitelského serveru. Díky souběžné podpoře několika virtuálních počítačů umožní tento přístup provoz několika operačních systémů najednou na jediném fyzickém počítači. Výsledkem této virtualizace je poté úspora za vynakládání finančních prostředků za mnoho nedostatečně využitých serverových počítačů, z nichž je každý vymezen pro konkrétní úlohu. Hardwarová virtualizace serverů nabízí možnost, konsolidovat počet serverových počítačů do menšího počtu lépe vytižených fyzických počítačů [5].

Nejčastěji je virtualizace dosaženo aplikací rozhraní programových prostředků mezi technické prostředky a vlastní operační systém. Tato vrstva programových prostředků komunikuje s fyzickými prostředky, tvoří virtuální prostředí pro provoz operačních systémů a obsluhuje jejich požadavky [3].

Hlavní výhodu lze spatřovat ve vzájemné izolovanosti virtuálních systémů, takže napadení nebo zhroucení jednoho neovlivní druhý a také dochází ke snížení rizika

napadení celého systému [4]. Za další pozitivum je možné považovat konsolidaci serverů, čili sloučení více služeb na méně serverů čímž umožní plné využití technických prostředků výkonu infrastruktury firmy. Rovněž umožní provoz více operačních systémů na jednom fyzickém serveru a zjednoduší zálohování či obnovu záloh, migraci systému nebo Disaster Recovery (DR), což je obnovení provozu kritických systémů. Také nabízí dynamické přidělování výkonu, jak navyšování, tak snižování [1]. Naproti tomu virtualizace přináší vyšší nároky na technické prostředky. Každý virtuální systém potřebuje dané množství fyzické RAM, místa na disku a rovněž vzrůstají požadavky na spolehlivost celého systému. Tím se pochopitelně zvyšuje jeho cena [3].

1.1 Základní metody virtualizace

Virtualizace je nejčastěji dosaženo kombinací technických a programových prostředků, čímž vzniká určitá virtualizační platforma. Existuje řada metod k virtualizaci platformy a každá z nich má samozřejmě své přednosti i nedostatky. K dosažení robustního systému schopného kvalitně plnit svoji funkci většina virtualizačních platform používá propojení více metod. Níže je uveden výčet přístupů k virtualizaci platformy, který je seřazen vzhledem k úplnosti simulace technických prostředků [6].

1.1.1 Hardwarová virtualizace

Hardwarová virtualizace patří mezi nejrozšířenější typ virtualizace. Jedná se o správu a přiřazování technických zdrojů jednotlivým serverům. Mezi technickými prostředky a virtuálními servery je umístěna virtualizační vrstva zahalující technické detaily systému a jejím prostřednictvím je pak k dispozici jen výkon systému. Tato varianta nabízí podporu většímu množství operačních systémů na jednom serveru. Schéma této virtualizace je znázorněno níže na obr. 2 s uvedením konkrétního příkladu operačního systému Windows nebo Linux. Výhody hardwarové virtualizace spočívají v přístupu z hlediska administrátora, vyhrazenosti operační paměti, zcela plné konfigurovatelnosti operačního systému a aplikací, rychlejším obnovení systému, jednoduchosti zálohování a obnově dat i zrychlení restartu nebo reinstalace serveru. Za nevýhody této virtualizace lze označit omezenost výkonu pouze na část technických prostředků či výdaje na operační systém, pořízení nebo pronájem licence. Konkrétní příklady tohoto druhu virtualizace jsou například produkty s názvem VMWARE a XEN [1].



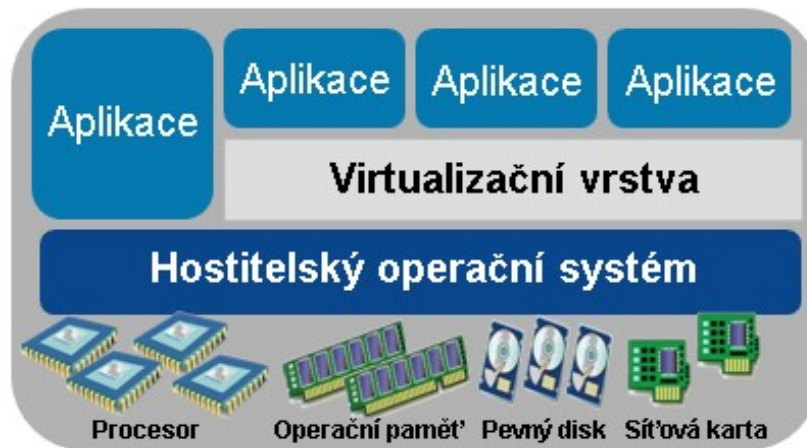
Obr. 2 – Schéma hardwarové virtualizace, zdroj: [1]

1.1.2 Emulace nebo simulace

Virtuální stroj simuluje všechny technické prostředky, čímž umožňuje provoz nemodifikovaného operačního systému řídicího programu na úplně odlišném procesoru. Nástroje tedy dovolují spustit operační systémy, které nejsou prvotně vymezeny pro architekturu fyzické soustavy. Používá se dynamické reinterpetování kódu virtuálního počítače. Tato metoda je využívána k vývoji aplikací pro procesory ještě se nevyskytující na trhu, u nichž je známá jen jejich specifikace. Dále lze tento přístup uplatnit k simulaci prostředí jiných architektur [3]. Emulace je implementována rozsáhlou škálou technik od stavových automatů až po dynamickou rekompilaci na zcela virtualizovaných platformách. Příklady zastupují řešení s názvy Bochs, PaerPC, Microsoft Virtual PC pro PowerPC, QEMU bez akcelerace a emulátor Hercules [6].

1.1.3 Nativní virtualizace a plná virtualizace

Dostačující množství technických prostředků simuluje virtuální stroj tak, aby poskytl oddělený běh nemodifikovaného operačního systému řídicího programu, který je vymezen pro totožný druh CPU [6]. Instrumenty úplné virtualizace tvoří kompletní virtuální technické prostředky. Zprostředkovávají spouštění neupraveného operačního systému se stejnou architekturou jako u fyzického počítače. Nedochozí k přímé komunikaci virtuálního stroje s fyzickými technickými prostředky, ale jeho požadavky odeslané prostřednictvím virtuálního zařízení jsou zpracovány a předány fyzickým zařízením [3]. Většinou je možný souběh více instancí. Jako příklad lze uvést konkrétní produkty s názvy VirtualBox, Virtual Iron, Virtual PC, VMwareWorkstation, VMware Server (dříve známý jako GSX Server), VMware ESX Server, QEMU, Parallels Desktop, Adeos, Mac-On-Linux, Win4BSD, Win4Lin Pro a z/VM [6]. Schéma nativní virtualizace, ze kterého je patrné umístění virtualizační vrstvy mezi jednotlivými aplikacemi a hostitelským operačním systémem, je znázorněno na obr. 3.



Obr. 3 – Schéma nativní virtualizace, zdroj: [2]

1.1.4 Částečná virtualizace

Virtuální stroj simuluje větší počet instancí, ale ne všech, prostředí technických prostředků, na kterých je odstartován řídicí program. Jde především o adresní prostor. V takovémto prostředí je podporováno sdílení zdrojů a izolace procesů. Nedokáže však oddělit instance operačních systémů. Z obecného hlediska sice u této verze není možné hovořit o virtuálním stroji, avšak historicky jde o významný přístup. Tato metoda byla použita u systémů CTSS a pokusného IBM M44/44X. Systémy Microsoft Windows i Linux se rovněž zabývají touto technikou [6].

1.1.5 Paravirtualizace

Při paravirtualizaci nejsou tvořeny kompletní virtuální technické prostředky, ale nástroje umožňují pouze komunikaci upraveného jádra virtualizovaného operačního systému s fyzickými technickými prostředky přes speciální API programovací rozhraní upraveného jádra hostitelského systému. Systémová volání se uskutečňují přes hypervisor, který je chápán jako řídicí prvek virtualizace [3]. Jedná se například o produkty s názvem Win4lin 9x, logické domény od Sunu a z/VM od IBM [6].

1.1.6 Virtualizace na úrovni operačního systému

Fyzický server je virtualizován na úrovni operačního systému, čímž je poskytnut současný provoz více izolovaných bezpečných virtuálních serverů na jednom fyzickém serveru [6]. Virtualizační vrstva je v tomto případě umístěna mezi operačním systémem serveru a virtuálními servery, což je zobrazeno na obr. 4. Na jednom fyzickém serveru dochází k podpoře pouze jednoho operačního systému. Mezi hostingovými firmami v České republice patří tento druh virtualizace k nejrozšířenějším formám [1].

Ke konkrétním příkladům náleží systémy Linux VServer, Virtuozzo (pro Linux nebo Windows), OpenVZ, kontejnery Solarisu a FreeBSD Jail [6].



Obr. 4 – Schéma virtualizace na úrovni operačního systému, zdroj: [1]

Výhody virtualizace na úrovni operačního systému lze spatřovat ve vymezeném výkonu serveru, pokročilé konfigurovatelnosti operačního systému a dalších aplikací v rychlejším obnovení činnosti systému, jednoduché zálohovatelnosti a obnově dat, nebo zkrácení doby restartu serveru. Naopak za nevýhody této metody virtualizace lze považovat výkon pouze určité části serveru či určité omezení instalovaných typů programových prostředků [1].

1.1.7 Aplikační virtualizace

Na daném stroji jsou provozovány desktopové nebo serverové aplikace, které používají místní zdroje, ale jejich provoz probíhá na zvláštním virtuálním stroji. V tom spočívá rozdíl oproti tradičnímu lokálnímu použití nativních aplikací, čili programových prostředků nainstalovaných přímo do operačního systému. Taková aplikace pracuje v nevelkém virtuálním prostředí zahrnujícím komponenty nezbytné ke spuštění. Jde například o položky registrů, soubory, proměnné prostředí, prvky uživatelského rozhraní a globální objekty. Virtuální prostředí zde působí jako vrstva mezi aplikacemi a operačním systémem, jež zabraňuje konfliktům mezi aplikací a operačním systémem či mezi aplikacemi navzájem. V příkladech jsou zahrnuty produkty s názvy Java Virtual Machine od Sunu Softricity, Thinstall, Altiris, Portable Apps a Trigence [6].

1.2 Principy virtualizace

Virtuální počítač je prostředí, které je tvořeno pomocí programových prostředků. Fyzické prostředky jsou rozděleny tak, aby na jednom počítači mohlo být v provozu více operačních systémů. Každá operační soustava má pro svůj provoz přidělenou část

prostředků z procesoru počítače, pevného disku, síťové karty a dalšího vybavení. Operační systém neví o tom, že pracuje ve virtuálním prostředí a chová se totožně jako v případě provozu na samostatném počítači. Virtualizace musí být provedena způsobem, který splňuje následující požadavky:

- rozhraní správy – jde o požadavek rozhraní správy, pomocí kterého je možné sledovat a konfigurovat virtuální stroje, musí být schopno programové správy a správy virtuálních serverů na dálku po síti;
- správa paměti – zajištění správného rozdělení paměťových zdrojů, tyto paměťové zdroje přidělené jednotlivým virtuálním serverům jsou od sebe navzájem izolovány;
- plánovač – zabezpečení přístupu k technickým prostředkům jednotlivých virtuálních serverů, plánovač je konfigurovatelným správcem, jednotlivým virtuálním serverům se přiřazují technické prostředky dle potřeby a na základě nastavených priorit;
- sledování stavu – automatické sledování aktuálního stavu všech virtuálních serverů pracujících na jednom počítači, je sledován stav využití CPU, operační paměti, dalšího zařízení a také zda je server v provozu či ne;
- datový prostor a síť – znamená to mít k dispozici úložné a síťové zdroje v potřebném množství ke svému provozu, virtuální server vystupuje navenek jako by měl k dispozici své vlastní pevné disky a síťové rozhraní, musí být zajištěn multiplexní přístup, který je izolovaný a bezpečný [7].

1.3 Dílčí závěr

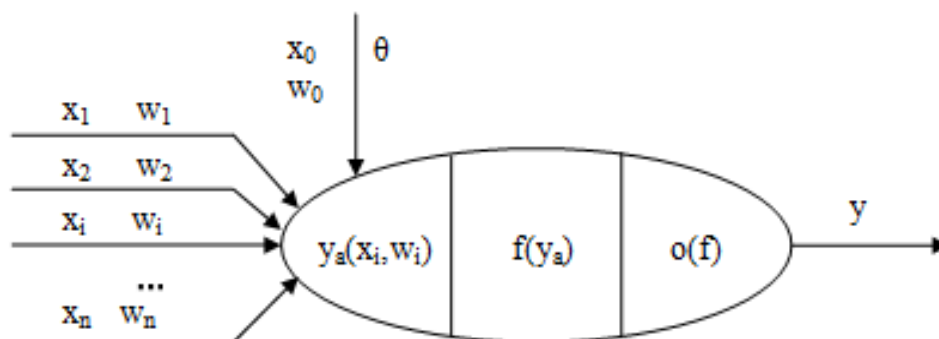
V kapitole je popsán význam pojmu virtualizace a její základní metody, kterými jsou hardwarová virtualizace, emulace nebo simulace, úplná virtualizace, částečná virtualizace, paravirtualizace, virtualizace na úrovni operačního systému a aplikační virtualizace. Dále jsou charakterizovány základní principy virtualizace, mezi které patří rozhraní správy, správa paměti, plánovač, sledování stavu, datový prostor a síť. Virtualizace nabízí zajímavé řešení integrace IT struktury organizací. Mezi její základní přednosti náleží zjednodušení správy, snížení nákladů na provoz, vyšší bezpečnost, vhodné rozdělení zdrojů, konsolidace serverů. Naproti tomu s virtualizací narůstají požadavky na technické prostředky i spolehlivost celého systému, čímž zajisté narůstá cena.

2 Neuronové sítě

V kapitole se uvádí základní pojmy z teorie neuronových sítí, které jsou inspirovány oborem neurofyziologie a odkud je tedy odvozen jejich název. Cílem této kapitoly je popis hlavních pojmů, primární vlastnosti neuronových sítí, kterou je schopnost učit se a základní druhy struktur neuronových sítí. Způsobnost učení je uskutečnitelná dvěma metodami, a to buď s přispěním lidského činitele, nebo bez něho. Rozlišují se v zásadě dvě struktury neuronových sítí – dopředná a rekurentní. Dopředná neuronová síť je složena z vrstev se spoji vedoucími od nejnižších do vyšších a obvykle mohou jednu či více vrstev přeskočit. Naopak v rekurentní neuronové síti je skupina neuronů zapojených v kruhu.

2.1 Neuron

Základní procesní jednotkou neuronové sítě je neuron. Neuronové sítě jsou poté tvořeny množinou modelů neuronů, které se určitým způsobem seskupují a jsou vzájemně mezi sebou propojeny. Struktura formálního modelu neuronu, tj. McCulloch-Pittsova neuronu, je znázorněn na obr. 5 [9].



Obr. 5 – McCulloch-Pittsův neuron, zdroj: [9]

Neuron je tvořen následujícími elementy [9]:

- x_i – vstupy neuronu (výstupy z předcházející vrstvy), $i = 1, 2, \dots, n$;
- n – počet vstupů (počet neuronů v předcházející vrstvě);
- w_i – synaptické váhy;
- y_a – vstupní potenciál neuronu;
- f – aktivační funkce neuronu;

- o – výstupní funkce neuronu;
- θ – práh neuronu;
- y – výstup neuronu.

Každý neuron zahrnuje konečný počet vstupů x_n a pouze jediný výstup y . V kterémkoliv neuronu dochází k přeměně vstupní hodnoty na výstup použitím minimálně dvou výpočetních procedur, výpočtu vstupního potenciálu y_a a aktivační funkce f . Neuronovou sít' je nutné chápat jako dynamický systém, čili systém, jehož vlastnosti se v závislosti na čase mění [9].

2.2 Neuronová sít'

Neuronová sít' je systém složený z výpočetních jednotek – neuronů, které jsou mezi sebou vzájemně propojeny spoji (synapssemi) ohodnocenými váhami. Z hlediska využití se v síti rozlišují tři typy neuronů – vstupní, pracovní (mezilehlé, skryté) a výstupní. Neuronová sít' prochází v průběhu času vývojem, mění se propojení a stav neuronů, adaptují se váhy [8].

Důležitými vlastnostmi neuronových sítí jsou schopnosti:

- učit se – nalézání závislostí v trénovacích datech a jejich následná reprezentace pomocí vah;
- zevšeobecnování (generalizace) získaných poznatků – správná reakce i na neznámé vstupy, na které nebyla neuronová sít' naučena [8];
- univerzální aproximace – dokáže aproximovat jakákoliv nelineární data.

Umělé neuronové sítě jsou inspirovány znalostmi z oboru neurofyzologie a imitují lidský mozek ve dvou aspektech. Jednak v získávání znalostí v průběhu učení neuronové sítě a dále k ukládání znalostí. Nositelem znalostí v neuronových sítích jsou tedy již zmíněné synapse [9].

2.3 Učení neuronové sítě

Učení náleží k základním vlastnostem neuronových sítí, čímž se odlišují od tradičního používání počítačů, kde je zapotřebí vytvořit algoritmus, podle něhož by výpočet proběhl. Pro neuronové sítě je navržen všeobecný algoritmus učení, který neuronová sít' aplikuje ve fázi učení. Rychlost a kvalita učení na předložených datech je určena příhodností tohoto algoritmu [9].

Východiskem všech současných algoritmů učení je tzv. Hebbův zákon učení, který charakterizuje algoritmus modifikace hodnot synaptických vah v nervových systémech živých organismů. Základem tohoto zákona je předpoklad, že jsou-li dva spolu propojené neurony v totožném okamžiku aktivní, pak jsou jejich vazby zesíleny. V obráceném případě jejich vzájemná vazba zeslabí. Pokud je aktivní pouze jeden neuron, nejsou synapse modifikovány. Každý uvažovaný neuron má výhradně dva statusy, a to aktivní nebo neaktivní [9], [10].

Činnost neuronových sítí lze rozdělit podle způsobu nastavování hodnot synaptických vah na dvě základní části:

- fáze učení – v této fázi se znalosti ukládají do synaptických vah neuronové sítě, synaptické váhy se v průběhu učení mění na základě pravidel určených typem učení neuronové sítě, pojmem učení se chápe adaptace neuronové sítě, která po dokončení učení bude nositelkou znalostí nabytých v průběhu učení;
- fáze života – získané znalosti se pouze využívají ve prospěch řešení určitého problému (např. predikce, klasifikace, apod.) a synaptické váhy zůstávají neměnné [9].

Množina hodnot se většinou náhodně člení na trénovací a testovací množinu uspořádaných hodnot. Trénovací množina je používána ve fázi učení. Naproti tomu množina testovací se aplikuje ve fázi života za účelem otestování nabytých znalostí v průběhu učení. Významnost reprezentativnosti trénovacích dat je mimořádná, neboť znalosti se ve fázi učení z těchto dat extrahují do synaptických vah neuronové sítě. Nevhodně vybraná trénovací data, působí negativně na kvalitu učení neuronové sítě. Proces učení tedy probíhá tak, že se na vstupy sítě opakovaně připojuje soubor vzorů požadovaného chování neuronové sítě, tzv. trénovací množina a dochází k vyhledávání optimálního nastavení parametrů sítě, při kterém definovaná funkce rozdílu předkládaného výstupu a výstupu neuronové sítě (chyba E) dosáhne minimální hodnoty [9].

Proces učení neuronové sítě lze uskutečnit těmito dvěma metodami:

- induktivní – jde o syntetický přístup k učení, při kterém jsou všeobecně platné závěry vyvozovány na základě sledování množiny jevů;
- deduktivní – jedná se o analytický přístup, při němž postačí pozorování a analýza jediného jevu [9].

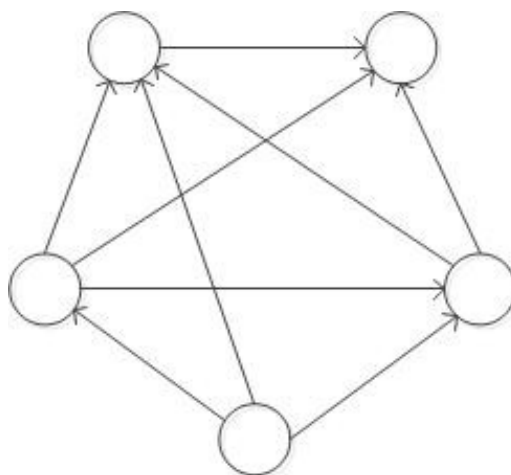
Při učení neuronové sítě je nejvíce uplatňována induktivní metoda, při které je či není zapotřebí lidská podpora. Je-li vyžadována lidská pomoc, jde o učení s učitelem, který dává správnou informaci o požadovaném výstupu neuronové sítě. V opačném případě se jedná o učení bez učitele, kdy učicímu systému není poskytnuta informace o správném výstupu učitelem, ale odvozuje si ji sám od svého výstupu uplatněním zpětné vazby [9].

2.4 Struktury neuronové sítě

Struktura neuronové sítě je určena počtem neuronů a jejich vzájemným propojením v neuronové síti. Z hlediska využití lze rozlišit tři typy neuronů, a to vstupní, skryté a výstupní. Stav neuronové sítě je vymezen stavem všech neuronů v síti. Konfigurace neuronové sítě je prezentována synaptickými vahami všech spojů. Struktura neuronové sítě se dělí na dva základní typy: dopředná a rekurentní [11].

2.4.1 Dopředná neuronová síť

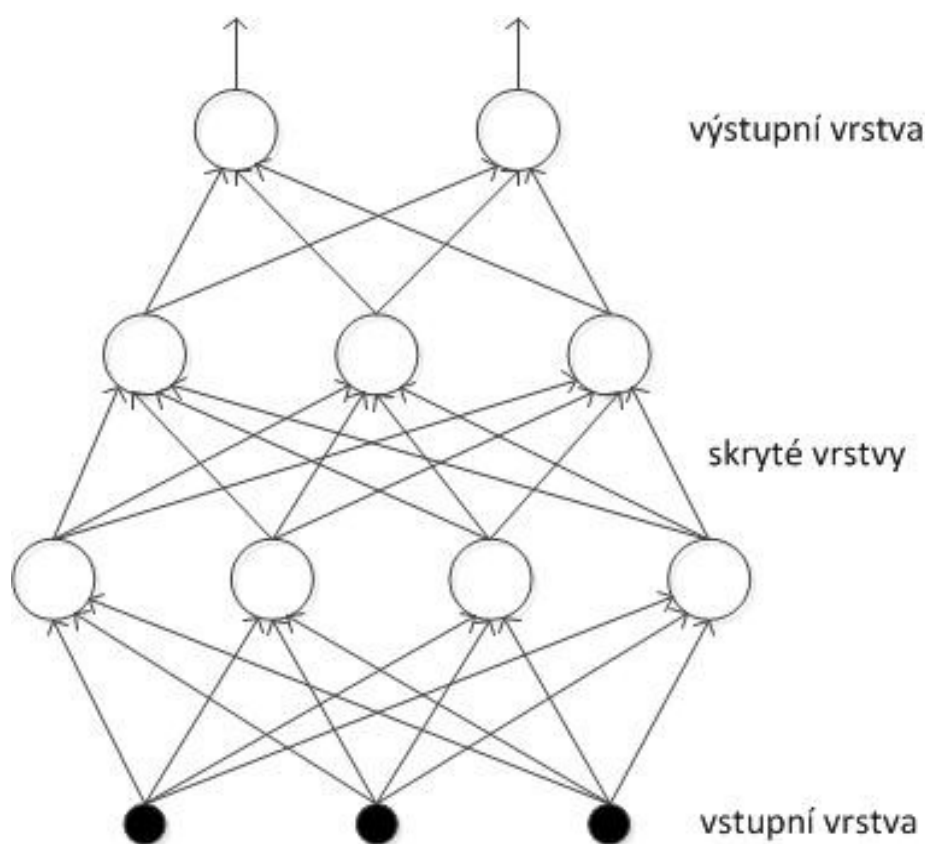
Dopřednými neuronovými sítěmi (resp. acyklickými) je signál šířen po synapsích jedním směrem – dopředu. V těchto sítích neexistuje cyklus, ale všechny cesty vedou jedním směrem, což znázorňuje obr. 6. Neurony jsou propojeny do tzv. vrstev, které jsou uspořádány tak, že spoje mezi jednotlivými neurony směřují jen z nižších vrstev do vyšších a běžně mohou přeskočit jednu či více vrstev [11].



Obr. 6 – Ukázka dopředné jednovrstvé neuronové sítě, zdroj: [11]

Speciálním případem této architektury je tzv. vícevrstvá neuronová síť, jejíž ukázka je vidět na obr. 7. V této síti se nultá (dolní), tzv. vstupní vrstva, skládá ze vstupních neuronů a poslední (horní), tzv. výstupní vrstva je tvořena výstupními neurony. Ostatní

skryté (mezilehlé) vrstvy jsou vytvářeny ze skrytých neuronů. Počty vrstev a neuronů představují parametry neuronové sítě a jsou závislé na povaze řešené úlohy [11].

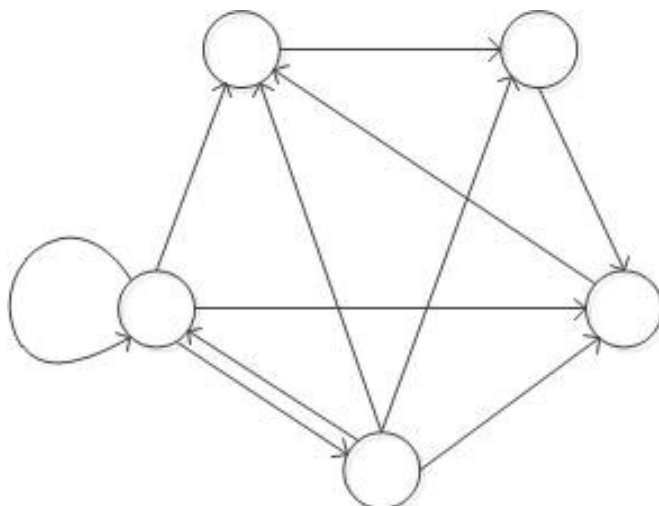


Obr. 7 – Ukázka dopředné vícevrstvé neuronové sítě, zdroj: [11]

Je-li počet neuronů malý, pak neuronová síť nedokáže postihnout veškeré závislosti v trénovacích datech. Pokud je naopak tento počet velký, poté doba učení narůstá a působením nadměrného množství trénovacích dat má síť špatnou schopnost generalizace způsobenou tzv. přeučením. Počet skrytých vrstev je vybírán na základě složitosti modelovaného systému. Vyhovující aproximace chování většiny modelových soustav je možné dosáhnout použitím jedné skryté vrstvy [9].

2.4.2 Rekurentní neuronová síť

V rekurentních neuronových sítích (resp. cyklických) existuje skupina neuronů zapojených v kruhu (tzv. cyklus), což přesně vystihuje obr. 8. Jde o způsob, ve kterém je skupina neuronů a výstup prvního neuronu je vstupem druhého neuronu, jehož výstup je opět vstupem třetího neuronu atd. Až na konci je výstup posledního neuronu v této skupině vstupem prvního neuronu. Nejjednodušším příkladem cyklu je zpětná vazba, ve které je současně výstup, zároveň jeho vstupem. Nejvyššího počtu cyklů je dosaženo, pokud výstup libovolného neuronu je vstupem každého neuronu. [11].



Obr. 8 – Ukázka rekurentní neuronové sítě, zdroj: [11]

Rekurentní neuronové sítě mají synapse orientované různými směry a v takovéto struktuře může být neuron současně vstupní i výstupní, tzv. duální neuron. Tento typ neuronové sítě musí vykazovat stabilní vlastnosti při šíření signálu dopředu i při zpětném šíření chyby. Pokud by tomu tak nebylo, mohlo by dojít ke vzniku nekontrolovatelných procesů a celý systém by pozbyl funkčnosti. Všeobecně lze říci, že rekurentní neuronové sítě zastupují dynamicky složitější systém než dopředné neuronové sítě [9].

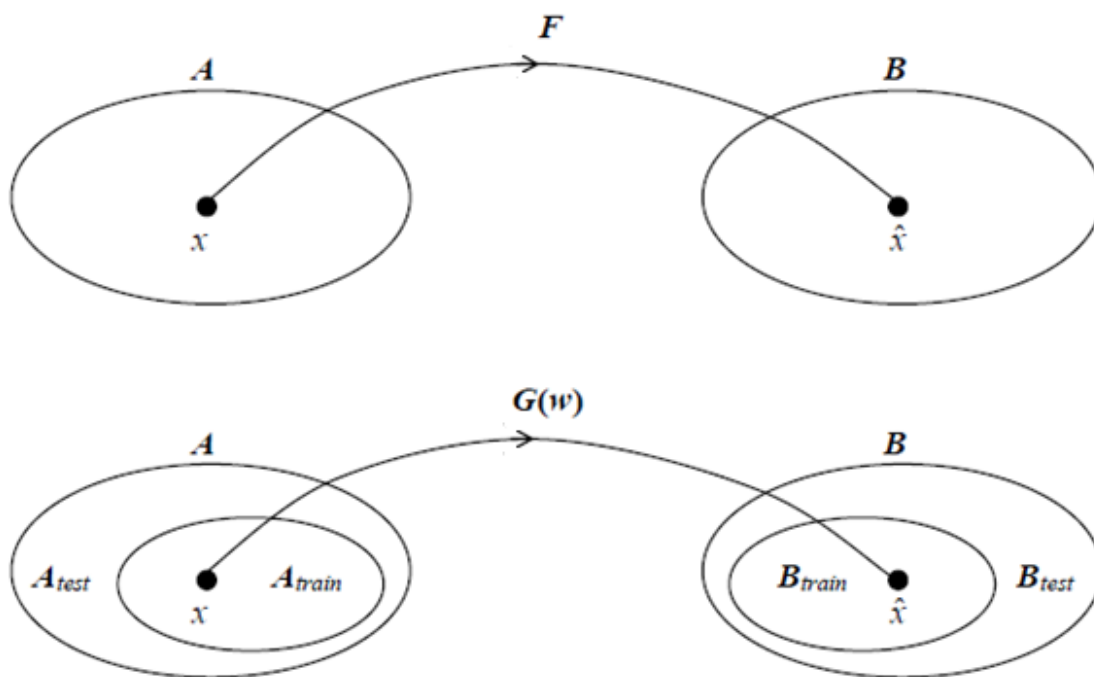
2.5 Oblasti využití neuronových sítí

Neuronové sítě se stále častěji uplatňují při hledání východisek praktických problémů technické i netechnické praxe. Jsou uplatňovány zejména v případech, kdy při řešení konkrétního problému buď není možné matematicky popsat všechny vztahy a souvislosti ovlivňující sledovaný proces nebo v případech, kdy je sice exaktní matematický model sestaven, ale je natolik složitý, že případná algoritmizace úlohy je prakticky nemožná. Nejčastěji jsou neuronové sítě využívány v těchto oblastech:

- otázky aproximace funkcí;
- klasifikace do tříd;
- predikce a následné rozhodování;
- řízení složitých zařízení v dynamicky se měnících podmínkách;
- modelace vývoje trhu;
- transformace a analýza signálu;
- simulace paměti [10].

2.6 Všeobecný klasifikační problém

Nechť je dána všeobecná formulace klasifikačního problému prostřednictvím pojmu zobrazení – funkce definované nad dvěma množinami A a B . Poté bude tento přístup prospěšný pro vysvětlení neuronových sítí jako klasifikátora a prediktora [10].



Obr. 9 – Všeobecný klasifikační problém, zdroj: [10]

Na obr. 9 je zachyceno znázornění $F : A \rightarrow B$, jehož zúžením na podmnožinu A_{train} je získáno nové modelové zobrazení $G(w)$. Funkční tvar tohoto zobrazení je vymezen parametrem w . Požadované funkční hodnoty \hat{x}_i jsou vykládány jako obrazy funkce F , kde $\hat{x}_i = F(x_i)$, ($i = 1, 2, \dots, r$).

Nechť $F(x)$ je funkce definovaná nad množinou A , která přiřadí každému prvku $x \in A$ funkční hodnotu z množiny B , $\hat{x} = F(x) \in B$,

$$F: A \rightarrow B. \quad (2.1)$$

Nechť $G(x, w)$, 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 zobrazení G , potom $\hat{x} = G(x, w) \in B_{train} \subset B$,

$$G_{(w)} : A_{train} \rightarrow B_{train}. \quad (2.2)$$

Formálně je možné konstatovat, 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} (nazvaný testovací množina), $A_{test} = A \setminus A_{train}$. Je pracováno s předpokladem, že pro každé $x_i \in A_{train}$ je získán požadovaný obraz – funkční hodnota \hat{x}_i ,

$$x_1 / \hat{x}_1, x_2 / \hat{x}_2, \dots, x_r / \hat{x}_r. \quad (2.3)$$

Cílem analýzy je nalezení takového parametru w funkce $G(x, w)$, aby funkční hodnoty argumentů z trénovací množiny A_{train} byly co nejbližší obrazům funkce $F(x)$, tj. požadovaným hodnotám. Nechť je dána účelová funkce v následujícím tvaru,

$$E(w) = \frac{1}{2 \sum_{i=1}^r (G(x_i, w) - \hat{x}_i)^2}. \quad (2.4)$$

Tato funkce vyjadřuje sumu čtverců odchylek funkce $G(x, w)$ od požadovaných hodnot \hat{x} braných z trénovací množiny. Předpoklad, aby vypočítané hodnoty $G(x, w)$ byly co nejbližší požadovaným hodnotám \hat{x} je uskutečněna prostřednictvím požadavku minimalizace účelové funkce $E(w)$ vzhledem k parametru w . Funkce $G(x, w)$ je adaptována, když její parametr w je vybrán tak, aby se rovnal své optimální hodnotě, tj. hodnotě ve které má účelová funkce globální minimum [10].

2.7 Dílčí závěr

V této kapitole je proveden popis neuronu a neuronové sítě. Dále je charakterizována základní vlastnost neuronové sítě, kterou je učení. Proces učení lze provést induktivní nebo deduktivní metodou, a to formou jak s učitelem, tak bez učitele. Následně jsou vymezeny dvě základní struktury neuronové sítě určené počtem neuronů a jejich vzájemným propojením v neuronové síti. Těmito typy jsou dopředné a rekurentní neuronové sítě. Kapitulu zakončuje definování všeobecného klasifikačního problému vzhledem k zaměření diplomové práce.

3 Neuronové sítě typu RBF

Radiální bazická funkce (Radial Basis Functions, RBF) náleží k modelům dopředných sítí s jednou skrytou vrstvou a je jedním ze způsobů řešení aproximačních problémů. Prvními, kteří navrhli využití aproximace pomocí radiálních bazických funkcí pro vytvoření nového modelu umělých neuronových sítí, byli Broomhead a Lowe [11].

Termínem radiální bazická funkce se rozumí jakákoliv dopředná neuronová síť používající RBF jako aktivační funkce. Použití RBF pro klasifikaci je vhodné, neboť ve většině případů konkrétní skupina vstupních vektorů p_i náleží k jedné ze tříd $w_{i,j} \in W$, které jsou vyhledávány pomocí neuronové sítě RBF. Je tedy možné zvolit si reprezentativní skupinu zástupce a posoudit soubor v rámci výkonu požadované třídy $w_{i,j} \in \Omega$ jeho okolí. J -tý výstup $f_j(x, H, w)$ je dán váženým součtem výstupů ze skryté vrstvy takto [12]

$$f_j(x, H, w) = \sum_{i=1}^q w_{j,i} \times h_i(x), \quad (3.1)$$

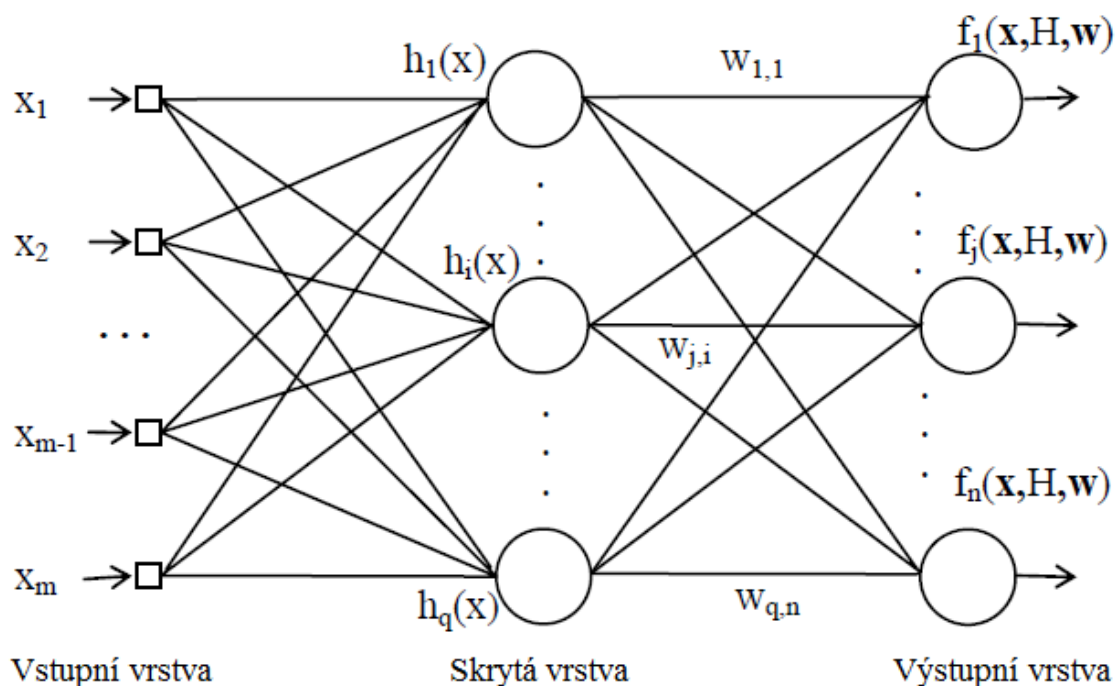
kde:

- $H = \{h_1(x), h_2(x), \dots, h_i(x), \dots, h_q(x)\}$ soubor aktivace funkce RBF neuronu ve skryté vrstvě;
- $w_{j,i}$ synapse jsou váhy;
- $h_i(x)$ je aktivační funkce.

3.1 Struktura neuronových sítí typu RBF

Struktura neuronové sítě typu RBF je tvořena třemi vrstvami plnicími naprosto odlišné úlohy. Jde tedy o třívrstvou neuronovou síť, kde vstupní vrstva neuronů slouží pouze k přenosu vstupních hodnot. Druhou (skrytou) vrstvou tvoří tzv. RBF jednotky, které realizují jednotlivé radiální funkce. Třetí neboli výstupní vrstva je lineární [13]. Struktura neuronové sítě je znázorněna na obr. 10.

Vstupní vrstva poskytuje pouze přenos vstupních hodnot x . Každý z m vstupů vektoru $x = (x_1, x_2, \dots, x_k, \dots, x_m)$ je vstupní hodnotou pro q radiálních bazických funkcí. Dále je obsažena jediná skrytá vrstva tvořená z tzv. RBF neuronů, které uskutečňují jednotlivé radiální bazické funkce. Poslední čili výstupní vrstva je složena z neuronů preceptronového typu a výstup je dán váženým součtem výstupů ze skryté vrstvy [12].



Obr. 10 – Struktura neuronové sítě, zdroj: [12]

Vstupní hodnoty x jsou použity jako parametry aktivačních funkcí v RBF neuronech ve skryté vrstvě a množina těchto funkcí je označena jako $H = \{h_1(x), h_2(x), \dots, h_i(x), \dots, h_q(x)\}$, kde q je počet neuronů ve skryté vrstvě. Aktivační funkce RBF neuronů je charakteristická monotónním stoupáním nebo klesáním se zvyšující se vzdáleností od centra c_i aktivačních funkcí $h_i(x)$ RBF neuronů. Aktivačními funkcemi jsou například Gaussova aktivační funkce, rotační Gaussova aktivační funkce, multikvadratická a inverzní multikvadratická aktivační funkce, Cauchyho funkce apod. [12], [14].

Gaussovu aktivační funkci $h_i(x)$ je možno dle [12] zapsat

$$h(x, C, R) = \sum_{i=1}^q \exp\left(-\frac{\|x-c_i\|^2}{r_i}\right), \quad (3.2)$$

kde:

- $x=(x_1, x_2, \dots, x_k, \dots, x_m)$ vstupní vektor;
- $C=\{c_1, c_2, \dots, c_i, \dots, c_q\}$ centra aktivačních funkcí $h_i(x)$ RBF neuronů;
- $R=\{r_1, r_2, \dots, r_i, \dots, r_q\}$ poloměry (šířky) aktivačních funkcí $h_i(x)$;
- q označuje počet neuronů ve skryté vrstvě.

3.2 Učení neuronové sítě typu RBF

Fáze procesu učení neuronové sítě typu RBF může být definována následovně:

- fáze učení neuronové sítě představuje postup realizace výstupu na základě známých vzorků předkládaných v podobě vstupů do neuronové sítě;
- obecně je fáze učení synonymem pro postup interpolace RBF neuronů na základě jednotlivých datových vzorků [15].

Učení neuronové sítě typu RBF je rozděleno do třech fází a jde jak o učení s učitelem, tak také o učení bez učitele. V průběhu učení je postupováno v síti vzhůru a postupně je pracováno se souřadnicemi středů jednotek, jejich šířkami a nakonec s koeficienty lineární kombinace [11].

Celý algoritmus učení RBF sítě lze popsat dle [11] takto,

vstup:

- trénovací množina $\{(x^{(t)}, d^{(t)}) \mid t = 1, 2, \dots, k\}$,

výstup:

- středy c_j ;
- šířky b_j ;
- váhy w_{sj} .

Trénovací množinu učení RBF sítě tvoří páry vektorů $\{(x^{(t)}, d^{(t)}) \mid t = 1, 2, \dots, k\}$ sestávajících ze vstupů $x^{(t)} \in \mathbb{R}^n$ a požadovaných výstupů $d^{(t)} \in \mathbb{R}^m$.

V první fázi algoritmu je definována pozice středů RBF jednotek reprezentovaných váhami mezi vstupní a skrytou vrstvou. Druhým krokem jsou nastaveny hodnoty případných dalších parametrů u RBF jednotek. Většinou mají jednotky nastavitelný parametr definující šířku oblasti kolem středu, ve které má jednotka relevantní výstup. Ve třetí fázi je běžné učení s učitelem adaptující váhy mezi skrytou a výstupní vrstvou. Vzhledem k tomu, že jde o koeficienty lineární kombinace, je učení lineární a není také nutné šířit chybu do nižších vrstev, neboť jejich váhy jsou už fixovány. To učení je tedy mnohem jednodušší než kupříkladu u standardního zpětného šíření, avšak jeho úspěšnost je závislá na předchozích krocích [11].

3.2.1 Centra RBF neuronů

V procesu učení RBF neuronové sítě je nutné stanovit počet center c_i aktivační funkce $h_i(x)$ a nalézt nejvhodnější pozici RBF center. Dalšími volitelnými parametry jsou poloměry center c_i , rychlost aktivačních funkcí $h_i(x)$ a nastavení vah $w(q, n)$ mezi skrytou a výstupní vrstvou. Jednou z jednoduchých metod je vybrání h náhodných trénovacích vzorků a umístění středů RBF jednotek na jejich vstupní části. Tento přístup předpokládá, že náhodně vybraný počet center c_i bude dostatečně reprezentovat data vstupující do RBF neuronové sítě. Tato metoda je vhodná pouze pro malé soubory vstupních dat. Její použití na větších souborech znamená mnohdy rychlý a zbytečný nárůst počtu RBF neuronů ve skryté vrstvě a neodůvodněnou komplexitu neuronové sítě. Dalším přístupem k umístění centra c_i , aktivačních funkcí $h_i(x)$ RBF neuronů lze realizovat pomocí algoritmu K-means [12], [14], [16].

3.2.2 Počet RBF neuronů ve skryté vrstvě

Stanovení vhodného počtu RBF neuronů ve skryté vrstvě neuronové sítě je podstatným bodem, neboť má zásadní vliv na výsledné vlastnosti neuronové sítě. Je-li zvolen příliš velký počet neuronů skryté vrstvy, může dojít k problému přeučení neuronové sítě. V takovém případě zvládne neuronová síť rozpoznat pouze již naučené vzory. U odlišných vzorů to nedokáže. Je tedy nevhodná pro další použití a proto je problematika správného stanovení počtu neuronů ve skryté vrstvě považována za východisko při sestavování RBF neuronové sítě. Zkoumáním této oblasti se v devadesátých letech dvacátého století zabývali Parth Niyogi a Federico Girosi, kteří definovali matematickou definici optimálního počtu RBF neuronů ve skryté vrstvě neuronové sítě [17].

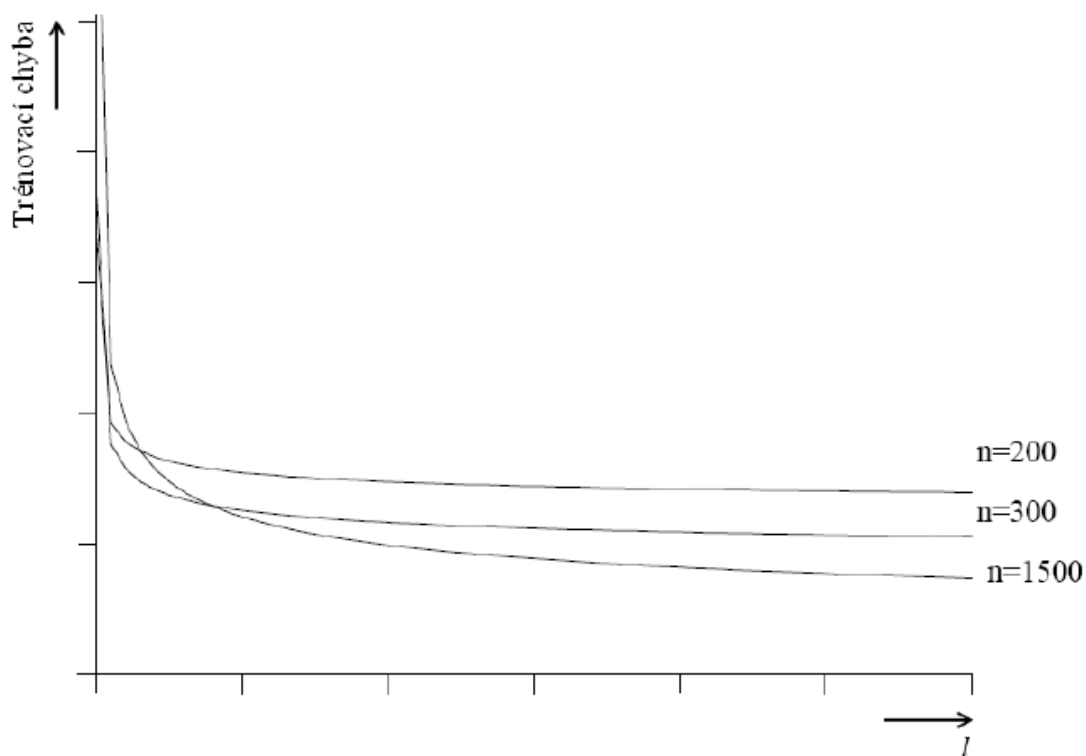
Nechť je dána RBF neuronová síť používající Gaussovu funkci s m_0 vstupními jednotkami a m_1 skrytými jednotkami. Nechť $f(x)$ znázorňuje regresní funkci patřící do Sobolevova prostoru. Potom neuronová síť přebírá trénovací data $\Psi = \{(x_i, d_i)\}_{i=1}^l$ jako náhodný výběr z regresní funkce $f(x)$. Pro jakýkoliv parametr $\delta \in (0,1)$ je trénovací chyba O RBF neuronové sítě ohraničena shora,

$$\|f_0 - \hat{f}_{m_1, l}\|^2 \leq O\left(\frac{1}{m_1}\right) + O\left(\frac{m_0 m_1}{l} \log(m_1 l) + \frac{1}{l} \log\left(\frac{1}{\delta}\right)^{\frac{1}{2}}\right), \quad (3.3)$$

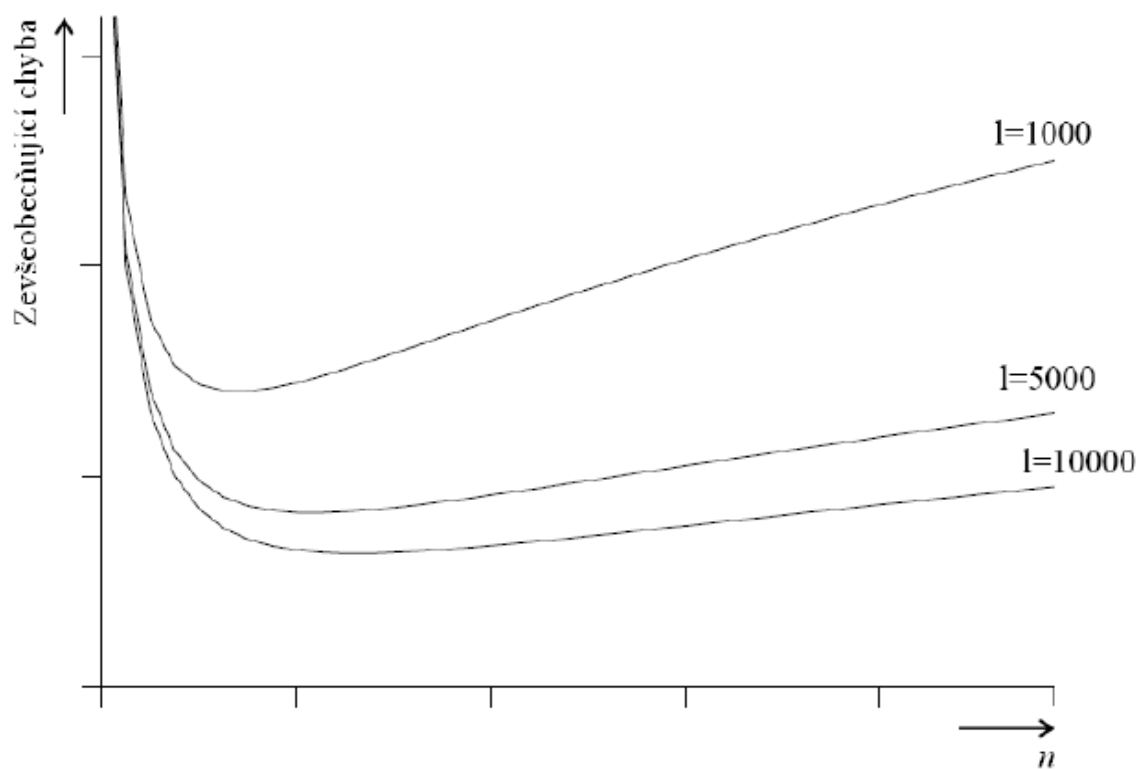
s pravděpodobností vyšší než $1 - \delta$, l je rozsah náhodného výběru (množiny vstupních dat) [13].

Nejdůležitějšími poznatky o počtu RBF neuronů ve skryté vrstvě jsou tyto:

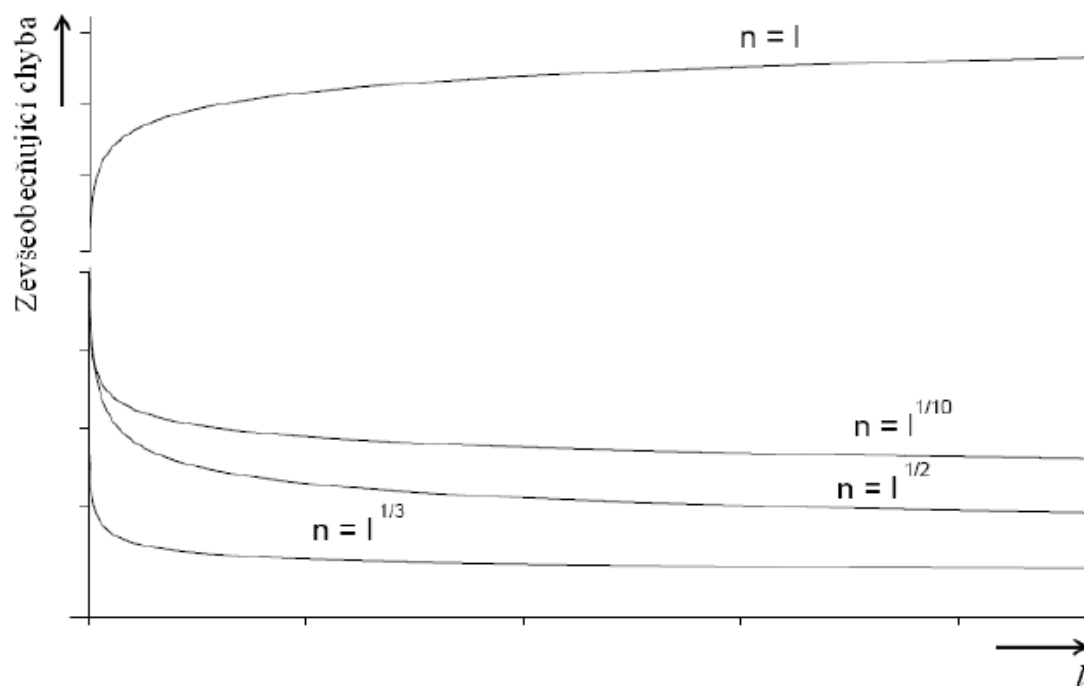
- celková chyba vztahující se k trénovací množině dat konverguje k nule s růstem počtu RBF neuronů m_i ve skryté vrstvě, což dokládá obr. 11 (kde n je počet RBF neuronů, l je množství vstupních dat);
- zevšeobecnující chyba (testovací a validační) klesá pouze při splnění předpokladu, že počet RBF neuronů m_i roste výrazně pomalejším tempem než množství dat l ve vstupní množině, z obr. 12 (kde n je počet RBF neuronů) je patrné, že se s množstvím dat použitých ve vstupní množině posouvá vpravo minimum křivky znázorňující zevšeobecnující chybu;
- průběh trénovací a zevšeobecnující chyby je ovlivněn množstvím dat obsažených v trénovací množině a počtem RBF neuronů ve skryté vrstvě;
- vlivem zvýšení i snížení množství RBF neuronů dochází k nárůstu zevšeobecnující chyby, což představuje obr. 13 (kde n je počet RBF neuronů, l je množství vstupních dat);
- dochází-li k překrývání jednotlivých RBF neuronů, potom se neuronové sítě typu RBF stávají neefektivními pro řešení klasifikačního problému [17].



Obr. 11 – Vliv množství vstupních dat a počtu RBF neuronů na trénovací chybu, zdroj: [17]



Obr. 12 – Vliv počtu RBF neuronů na zevšeobecnující chybu, zdroj: [17]



Obr. 13 – Vliv množství vstupních dat a počtu RBF neuron na zevšeobecnující chybu, zdroj: [17]

3.3 Dílčí závěr

Neuronové sítě typu RBF patří k modelům dopředných neuronových sítí používajících RBF jako aktivační funkce. Strukturu této sítě tvoří tři vrstvy, které plní zcela odlišné úlohy. Úlohou první vrstvy neuronů je přenos vstupních hodnot. Druhá neboli skrytá vrstva je tvořena tzv. RBF jednotkami realizujícími jednotlivé radiální funkce. Třetí neboli výstupní vrstva je lineární. Dále je definován třífázový proces učení, při němž je postupováno v síti směrem vzhůru a postupně je zacházeno se souřadnicemi středů jednotek, jejich šířkami a v závěrečném kroku s koeficienty lineární kombinace. Následně je zmíněna problematika stanovení vhodného počtu RBF neuronů ve skryté vrstvě neuronové sítě, neboť zásadně ovlivňuje výsledné vlastnosti neuronové sítě. Kapitola je ukončena popisem tématu stanovování počtu center RBF.

4 Návrh modelu klasifikace dat pomocí RBF neuronových sítí

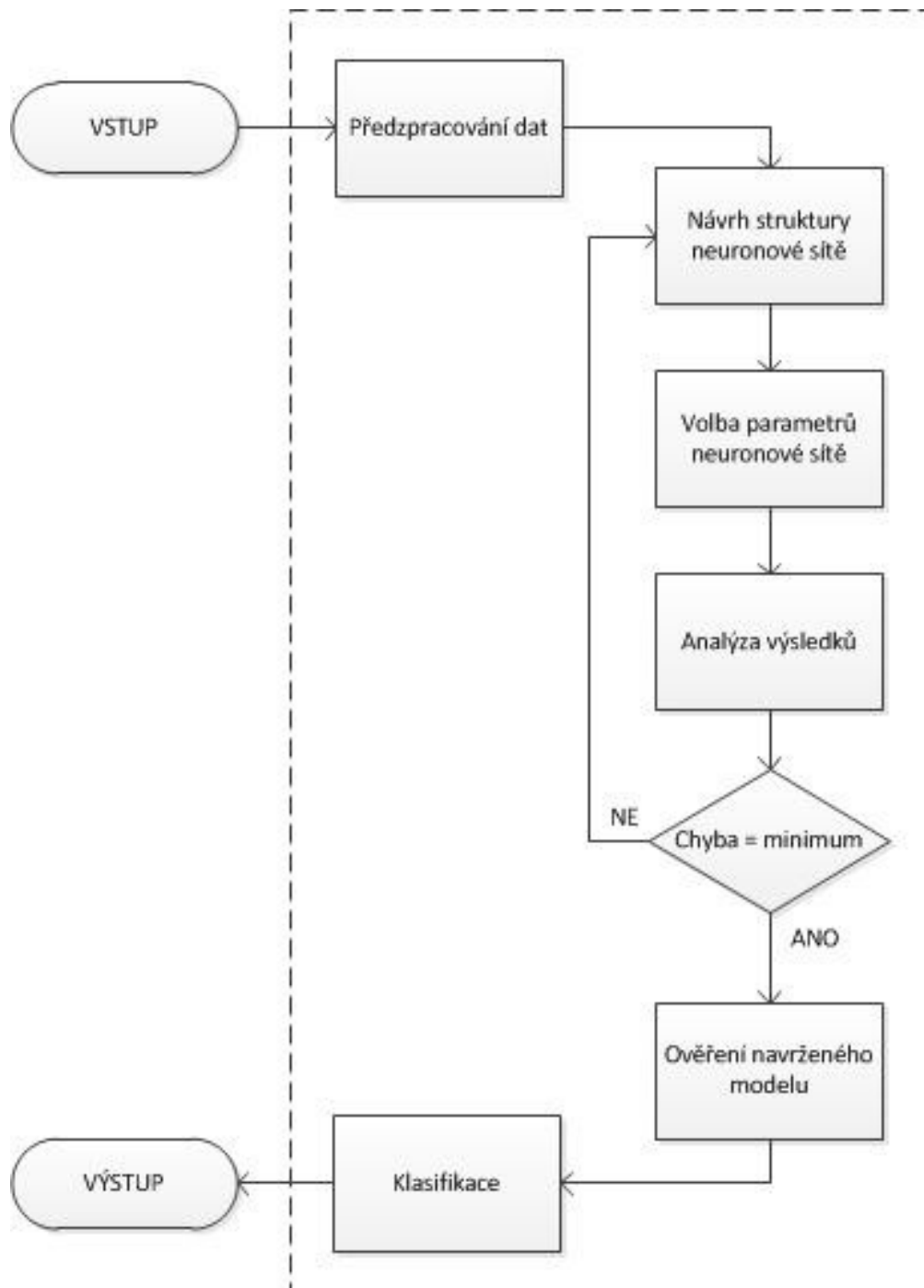
Kapitola obsahuje informace o postupu vytvoření modelu sloužícího ke klasifikaci dat metodou učení s učitelem, pomocí neuronových sítí typu RBF. Nejprve je popsán postup předzpracování dat, dále je uskutečněn návrh struktury a parametrů neuronové sítě. K samotnému provedení klasifikace, rozdělení dat na trénovací a testovací množinu a shlukové analýzy je využito programové prostředí SPSS Clementine. Výstupem je model neuronové sítě s parametry a strukturou, při kterých je dosaženo výsledků klasifikace s co nejvyšší přesností.

4.1 Návrh modelu

Vstupem do modelu je datová matice. Prvním krokem návrhu modelu pro klasifikaci je předzpracování dat, jehož cílem je získat vektor hodnot, který bude vstupem do neuronové sítě. Předzpracování dat znamená provést analýzu dat, tedy zjistit jaké atributy datová matice obsahuje, jaké jsou jejich datové typy, rozsah hodnot a celkové množství záznamů. Z analyzovaných dat je poté vytvořena vstupní matice, prověřena vzájemná korelace mezi atributy, hodnoty jsou standardizovány a rozděleny na trénovací a testovací množinu.

Dalšími kroky modelu jsou samotný návrh struktury a vhodná volba parametrů neuronové sítě. Struktura vstupní a výstupní vrstvy neuronové sítě vychází z počtu vstupních proměnných a počtu tříd, do kterých je prováděna klasifikace. Počet neuronů ve skryté vrstvě je určen na základě experimentů. Zjišťovanými parametry jsou vliv momentu a překrytí RBF neuronů.

Po provedení experimentů následuje analýza výsledků, na základě které je rozhodováno, zda chyba klasifikace dosáhla svého minima. Jestliže není dosaženo minimální chyby, opakuje se návrh struktury, volba parametrů a následná analýza výsledků. V případě, kdy je již dosaženo minimální možné chyby, je provedeno ověření navržené neuronové sítě zavedením validační množiny dat. Je provedena klasifikace. Výstupem z modelu je optimální návrh neuronové sítě RBF pro klasifikaci dat virtuálních serverů. Tento navržený model je znázorněn na obr. 14.



Obr. 14 – Návrh modelu, zdroj: [vlastní]

4.2 Analýza dat

Data charakterizující virtuální servery sloužící pro vytvoření modelu klasifikace, jsou poskytnuta Univerzitou Pardubice. Jedná se o naměřené hodnoty jejich virtuálních serverů, konkrétně webového serveru Portal a databázového serveru Oracle. Portal ke svému provozu využívá databázi Oracle. Jestliže je tedy mimo provoz Oracle databáze, současně nefunguje ani webové rozhraní na serveru Portal.

Získaná data jsou ve formátu csv. Jedná se o 4 soubory, přičemž dva soubory obsahují naměřené hodnoty sledovaných atributů na serveru Portal a Oracle. Ve zbylých dvou souborech je obsažen záznam o jejich aktuálním chybovém stavu. Společným identifikátorem všech souborů je atribut datum a čas. Tyto soubory je zapotřebí pro provádění modelování spojit do jediného souboru. K realizaci spojení souborů obsahujících naměřené hodnoty je využita funkce SVYHLEDAT programového prostředí Microsoft Excel 2010.

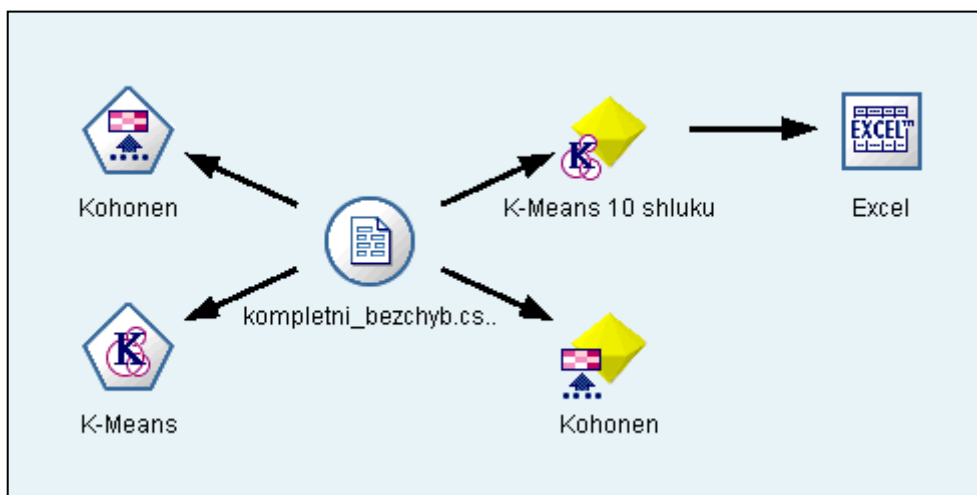
Do souborů obsahujících informaci o chybových stavech je prováděn záznam pouze ve chvíli, kdy chyba nastala a je zapsáno o jaký druh chyby se jedná. Spojení tedy není možné učinit stejným způsobem jako v minulém případě, protože měření hodnot serverů probíhá přesně v pětiminutových intervalech. Nejprve je tedy zapotřebí sjednotit časový záznam, čehož je docíleno pomocí ručního zaokrouhlení na celé pět minuty. Předpokládá se vzhledem k povaze chyb, že se nejedná o momentové chybové stavy, ale chyby delšího trvání. V případě různého chybového stavu na obou serverech současně, je jako nadřazený chápán chybový stav serveru Portal. Teprve nyní je data možné za použití vyhledávacích funkcí připojit k naměřeným hodnotám.

Pomocí funkce FILTR jsou odstraněny měření s chybějícími hodnotami. Výsledný získaný datový soubor obsahuje 41958 provedených měření v pětiminutových intervalech a časovém horizontu od 25. 6. 2010 do 24. 11. 2010. Výsledná datová matice použitá pro klasifikaci dat pomocí neuronových sítí obsahuje 206 měření. Měřené atributy serverů jsou popsány v tab. 1. Z tohoto počtu je 144 záznamů měření, při kterém nastává na serveru chyba a 62 měření při běžném provozním stavu serveru bez chyby.

Výběr záznamů měření, při němž nenastala chyba, je proveden za pomoci shlukové analýzy. Ze záznamů kompletní datové matice jsou vyfiltrována pouze měření, při kterých nenastává chybový stav. Následně je provedena shluková analýza do 10 shluků pomocí metody K-means. Počet 10 shluků je určen nástrojem založeným na Kohonenových samoorganizujících se mapách. Výsledky shlukování jsou exportovány do souboru xls. Strukturu tohoto modelu shlukování zobrazuje obr. 15.

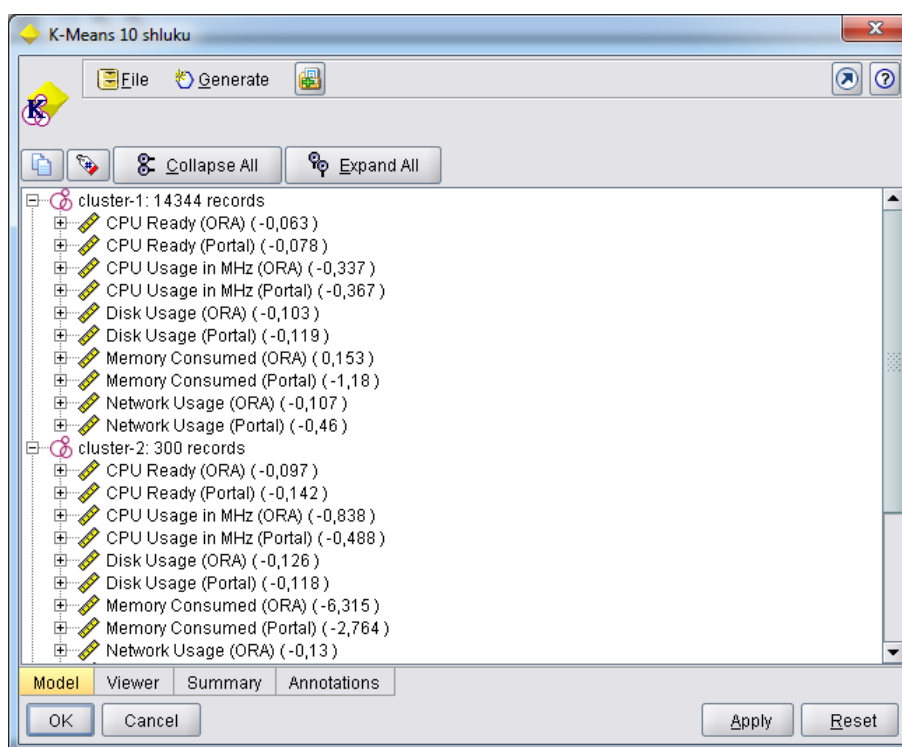
Tab. 1 – Datový slovník, zdroj: [vlastní]

Název atributu	Popis	Datový typ	Rozsah hodnot
Date	Obsahuje záznam data a času měření	Bez typu	-
CPU Usage in MHz (Portal)	Využití CPU v Mhz - Portal	Rozsah	[269 - 5582]
CPU Usage in MHz (ORA)	Využití CPU v Mhz - Oracle	Rozsah	[274 - 9419]
CPU Ready (Portal)	CPU připraven - Portal	Rozsah	[1528 - 191659]
CPU Ready (ORA)	CPU připraven - Oracle	Rozsah	[6 - 98726]
Memory Consumed (Portal)	Množství spotřebované paměti- Portal	Rozsah	[3948918 - 12526848]
Memory Consumed (ORA)	Množství spotřebované paměti - Oracle	Rozsah	[4151665 - 16700301]
Disk Usage (Portal)	Množství spotřebované kapacity datového úložiště - Portal	Rozsah	[11 - 5108]
Disk Usage (ORA)	Množství spotřebované kapacity datového úložiště - Oracle	Rozsah	[1 - 42672]
Network Usage (Portal)	Síťový provoz - Portal	Rozsah	[0 - 2290]
Network Usage (ORA)	Síťový provoz - Oracle	Rozsah	[0 - 43471]
Chyba	Označení chyby	Množina	[0, 1, 2, 3, 4]



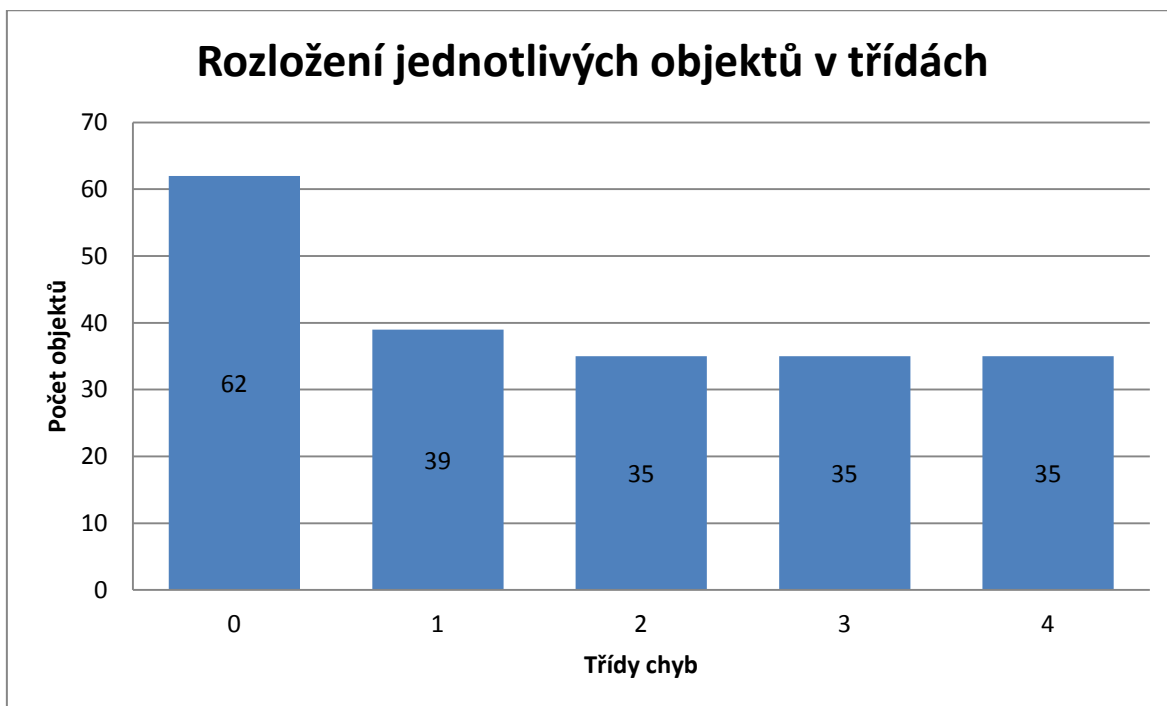
Obr. 15 – Struktura modelu shlukové analýzy, zdroj: [vlastní]

Datová matice obsahuje 144 chybových stavů. Z jednotlivých měření, při kterých chyba nenastala je vybrán pouze omezený počet záznamů, aby došlo k co nejmenšímu rozdílu v četnostech objektů v jednotlivých třídách. Zároveň je ve výběru důležité dostatečné zastoupení reprezentantů všech shluků. Z tohoto důvodu jsou po provedení shlukové analýzy zjištěna centra jednotlivých shluků a z každého shluku je vybráno pomocí postupné filtrace přibližně 6 hodnot položených nejbliže k jeho centru. Náhled zjištěných center shluků je na obr. 16.



Obr. 16 – Centra shluků, zdroj: [vlastní]

Jednotlivé rozložení počtu objektů ze získaného datového souboru v třídách znázorňuje obr. 17. V třídě 0 se nachází 62 objektů. Do třídy 1 náleží 39 objektů. Třídy 2, 3 a 4 obsahují každá 35 záznamů.



Obr. 17 – Rozložení jednotlivých objektů v třídách, zdroj: [vlastní]

Data obsahují atribut s názvem chyba nesoucím informaci o druhu chyby, která se vyskytla v měřený okamžik. Počet kategorizovaných druhů chybových stavů je 5 a jsou uvedeny v tab. 2.

Tab. 2 – Popis chybových stavů serverů, zdroj: [vlastní]

Název chyby	Popis chyby	Označení
Bez chyby	Servery pracují bez chyby	0
Time Out	Žádná odpověď serveru v požadovaném intervalu	1
Service Error	Chyba sledované služby	2
Load Warning	Malé přetížení systému	3
Load Critical	Kritické přetížení systému	4

Monitorovací systém virtuálních serverů Portal a Oracle zaznamenávající jednotlivé chybové stavy je programový prostředek Nagios.

Vysvětlení chybových stavů:

- Bez chyby – sledovaná služba je dostupná, jedná se o běžný provoz serveru bez komplikací;
- Time Out - server neodpověděl v požadovaném časovém intervalu na kontrolní dotaz monitorovacímu systému, tato chyba se může vyskytnout například při velkém množství současných přístupů uživatelů na server;
- Service Error - může nastat v případě interní chyby serveru;
- Load Warning - jedná se o malé přetížení systému, při kterém je systém ještě schopen zachovat dostupnost poskytovaných služeb, tato chyba se zaznamenává v případě překročení nižší nastavené hranice počtu procesů čekajících na zpracování, při stále zvyšující se zátěži serveru hrozí chybový stav Load Critical;
- Load Critical - v případě této chyby již zpravidla dochází k nedostupnosti služby, počet procesů čekajících na zpracování se dostává nad nejvyšší stanovenou hranici, systém je přetížený, možným východiskem pro obnovení provozu služby je restart serveru.

4.3 Korelace

Na vybraných 206 záznamech je v prvním kroku nutno zjistit, zda neexistuje korelační závislost mezi hodnotami jednotlivých atributů. Pokud je mezi atributy závislost zjištěna, jako vstupní do neuronové sítě je použit pouze jeden z nich. Korelační koeficient je číslo nabývající hodnoty mezi -1 a +1, které měří míru asociace dvou proměnných. Kladná hodnota korelačního koeficientu naznačuje pozitivní asociaci a naopak záporná hodnota určuje negativní nebo inverzní vztah. V případě, že nabývá hodnoty 0, neexistuje mezi proměnnými žádný vztah.

Ke zjištění koeficientů korelace je využita funkce analýza dat z programového prostředí Microsoft Excel 2010. Pro vybraná data je sestavena korelační matice s hodnotami uvedenými v tab. 3. Z korelační matice není zjištěna žádná významná závislost mezi jednotlivými atributy, a proto jsou všechny vybrané atributy použity k návrhu neuronové sítě.

Tab. 3 – Korelační matice, zdroj: [vlastní]

	1	2	3	4	5	6	7	8	9	10
1	1									
2	0,337446085	1								
3	0,230774137	0,110014637	1							
4	0,374629817	-0,15859741	0,032494428	1						
5	0,781147591	0,478824535	0,245722389	-0,094648901	1					
6	0,685012655	0,318373004	0,513445466	-0,019115083	0,709919982	1				
7	-0,649454488	-0,312271083	-0,345183358	0,108321278	-0,759846767	-0,798376996	1			
8	0,047253114	0,019269774	0,002051068	0,028178305	0,009486379	0,051088542	0,061145223	1		
9	0,216671846	0,142804763	0,070413001	-0,049675617	0,283356381	0,342120585	-0,277169413	-0,076713237	1	
10	0,342913605	0,355275454	0,116871052	-0,106859383	0,608170158	0,365739071	-0,281972882	-0,032673907	0,485412798	1

Legenda

- 1 CPU Usage in MHz (Portal)
- 2 CPU Ready (Portal)
- 3 Memory Consumed (Portal)
- 4 Disk Usage (Portal)
- 5 Network Usage (Portal)
- 6 CPU Usage in MHz (ORA)
- 7 CPU Ready (ORA)
- 8 Memory Consumed (ORA)
- 9 Disk Usage (ORA)
- 10 Network Usage (ORA)

4.4 Předzpracování dat

Mezi hodnotami jednotlivých atributů obsažených v původní datové matici jsou velké rozdíly v měřených jednotkách. Pro potřeby vytvoření modelu klasifikace je zapotřebí, aby byly naměřené hodnoty mezi sebou vzájemně porovnatelné. Z tohoto důvodu je provedena standardizace datové matice podle vztahu 4.1

$$x_{ij} = \frac{z_{ij} - \bar{z}_j}{s_j}, \quad (4.1)$$

kde:

- z_{ij} - původní hodnota;
- \bar{z}_j - aritmetický průměr;
- s_j - směrodatná odchylka.

Aritmetický průměr z hodnot jednotlivých atributů byl vypočítán dle vztahu 4.2, směrodatná odchylka na základě vztahu 4.3

$$\bar{z}_j = \frac{1}{n} \sum_1^n z_{ij}, \quad (4.2)$$

$$s_j = \sqrt{\frac{1}{n} \sum_1^n (z_{ij} - \bar{z}_j)^2}, \quad (4.3)$$

kde:

- n – počet prvků.

Dalším krokem přípravy dat pro potřeby vytvoření modelu klasifikace je rozklad na trénovací a testovací množinu. Na základě dat z trénovací množiny je neuronová síť naučena a data z množiny testovací slouží k testování s jakou chybou je schopna takto naučená síť klasifikovat data do odpovídajících tříd.

Rozklad množiny dat na trénovací a testovací množinu, tedy $A = A_{train} \cup A_{test}$, lze provést například za pomoci metod shlukové analýzy. Potom je získáno rozdělení prvků množiny A do jednotlivých shluků, které obsahují podobné objekty, viz vztah 4.4 a vztah 4.5 [9]

$$A = c_1 \cup c_2 \cup \dots \cup c_p, \quad (4.4)$$

kde:

- c_i – jednotlivé shluky, shluk obsahuje n prvků z množiny A ;
- p – počet shluků,

$$c_i = \{o_1^{(i)}, o_2^{(i)}, \dots, o_{n_i}^{(i)}\}, \quad (4.5)$$

kde:

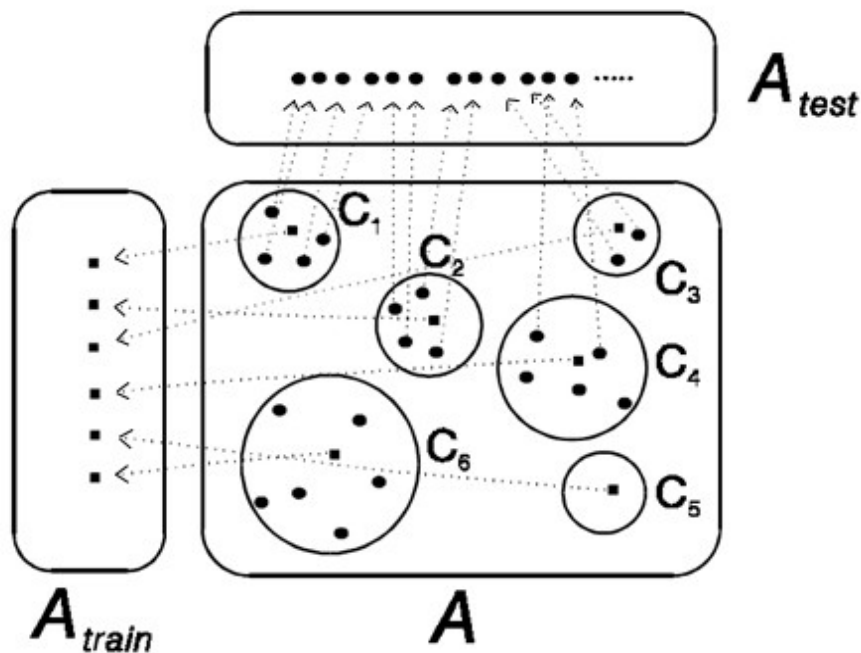
- $o_j^{(i)}$ – objekty patřící do shluku i .

Dále je pracováno s předpokladem, že objekt $o_1^{(i)} \in C_i$ je objekt ze shluku C_i a je nejbližší jeho centru. Tento objekt je chápán jako reprezentant objektů shluku C_i . Potom je možné trénovací a testovací množinu objektů vyjádřit vztahy, vztah 4.6 a 4.7 [10]

$$A_{train} = \{o_1^{(1)}, o_1^{(2)}, \dots, o_1^{(p)}\}, \quad (4.6)$$

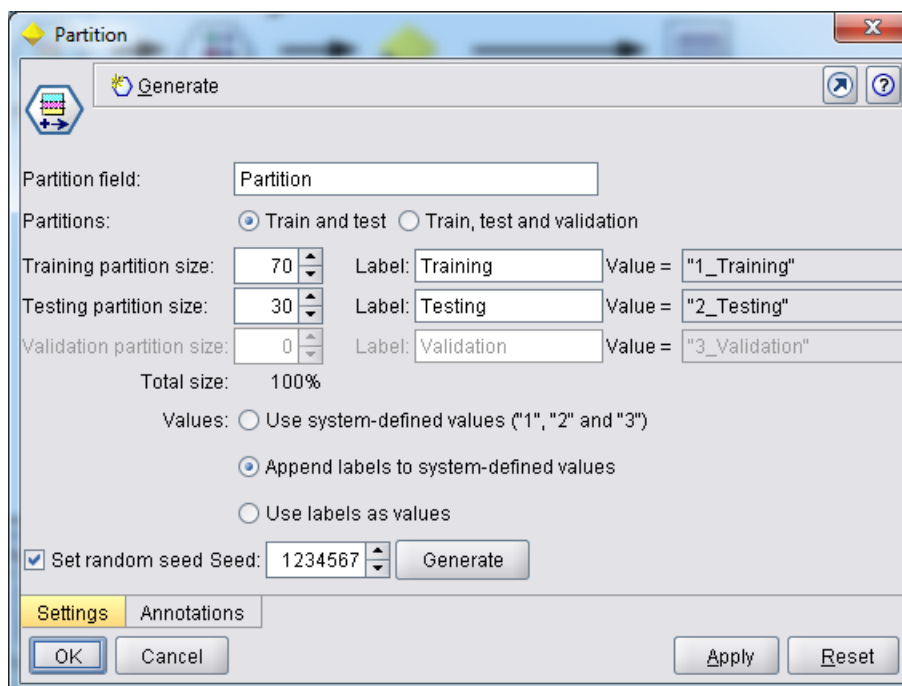
$$A_{test} = (c_1 - \{o_1^{(1)}\}) \cup (c_2 - \{o_1^{(2)}\}) \cup \dots \cup (c_p - \{o_1^{(p)}\}). \quad (4.7)$$

Trénovací množina je tedy složena ze všech objektů reprezentujících jednotlivé shluky, jak je vidět na obr. 18.

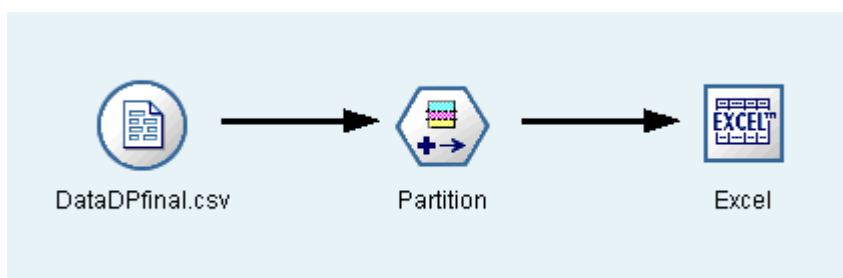


Obr. 18 – Rozdělení množiny na trénovací a testovací, zdroj: [10]

Pro potřeby modelu klasifikace dat je provedeno rozdělení množiny připravených dat v poměru 70 % trénovací a 30 % testovací množina dat. Uvedené rozdělení je realizováno za pomoci nástroje s názvem Partition. Jeho nastavení zachycuje obr. 19. Model pro rozdělení dat je znázorněn na obr. 20. Datová matice s informací o rozdělení množiny na trénovací a testovací je vyexportována a uložena do souboru xls. V této fázi jsou data připravena ke klasifikaci.



Obr. 19 – Nastavení Partition, zdroj: [vlastní]

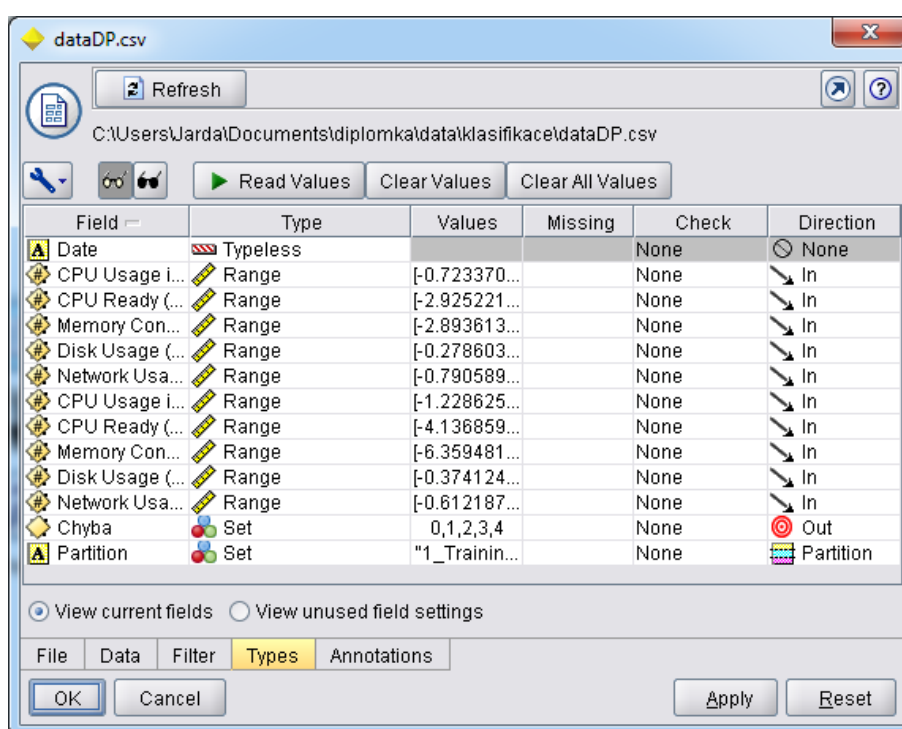


Obr. 20 – Struktura modelu rozdělení dat, zdroj: [vlastní]

4.5 Návrh RBF neuronové sítě

Neuronová síť typu RBF obsahuje vstupní vrstvu, skrytou vrstvu a výstupní vrstvu. Počet neuronů ve vstupní vrstvě je určen na základě délky vstupního vektoru dat. V tomto případě tvoří vstupní vektor 10 objektů. Ve vstupní vrstvě neuronové sítě je tedy 10 neuronů. Výstupní vrstva neuronové sítě závisí na počtu tříd, do kterých jsou jednotlivé objekty klasifikovány. Počet chybových tříd jednotlivých měření je 5. Výstupní vrstva je

tedy složena z 5 výstupních neuronů. Ve skryté vrstvě je počet skrytých neuronů zjištěn na základě minimální chyby při klasifikaci. Ke zjištění této minimální chyby a následně pro návrh optimální neuronové sítě jsou provedeny jednotlivé experimenty. Nejprve jsou načtena vstupní data včetně určení jejich datových typů a rozsahu hodnot. Dále je zapotřebí nastavit charakter jednotlivých proměnných viz obr. 21. Konkrétně se jedná o vstupy do neuronové sítě, což jsou jednotlivé měřené atributy. Dále výstupní proměnná, v tomto případě atribut s názvem chyba nesoucí informaci o druhu třídy. Jedná se tedy o učitele neuronové sítě. Poslední je atribut partition, který nese informaci o rozdělení dat na trénovací a testovací množinu.

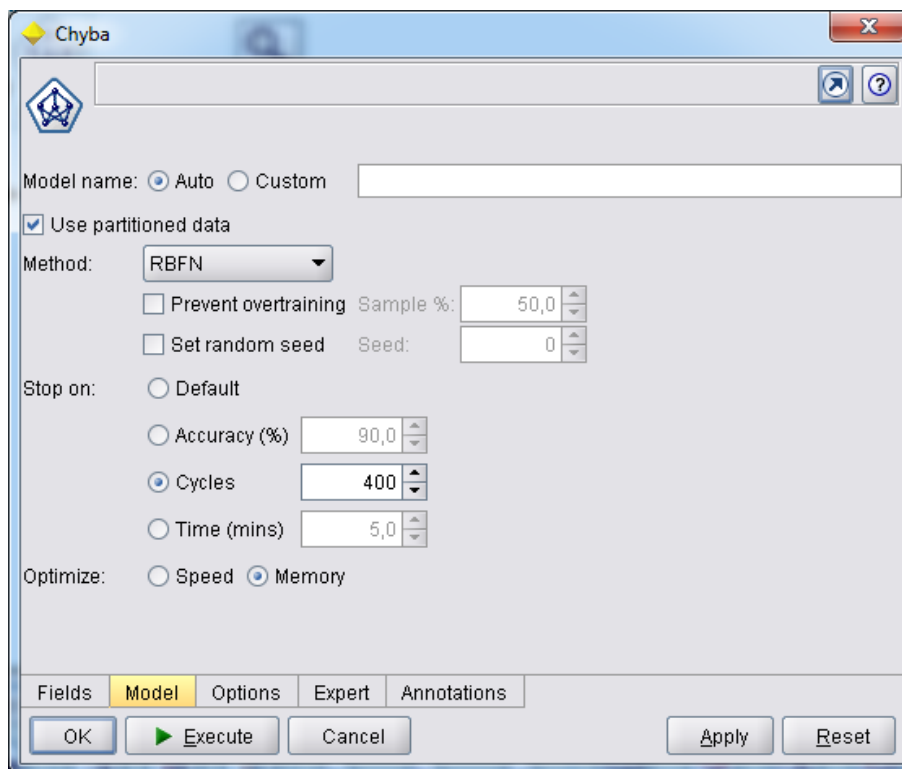


Obr. 21 – Nastavení charakteru proměnných datové matice, zdroj: [vlastní]

Dalším využitím při návrhu neuronové sítě, je nástroj s názvem Neural Net. Na základě jeho nastavení vznikne model neuronové sítě. Využity jsou tyto nastavitelné parametry:

- Method - slouží k nastavení typu neuronové sítě, tedy RBF;
- Cycles - nastavení počtu cyklů učení neuronové sítě.

Nastavení Method a Cycles znázorňuje obr. 22.

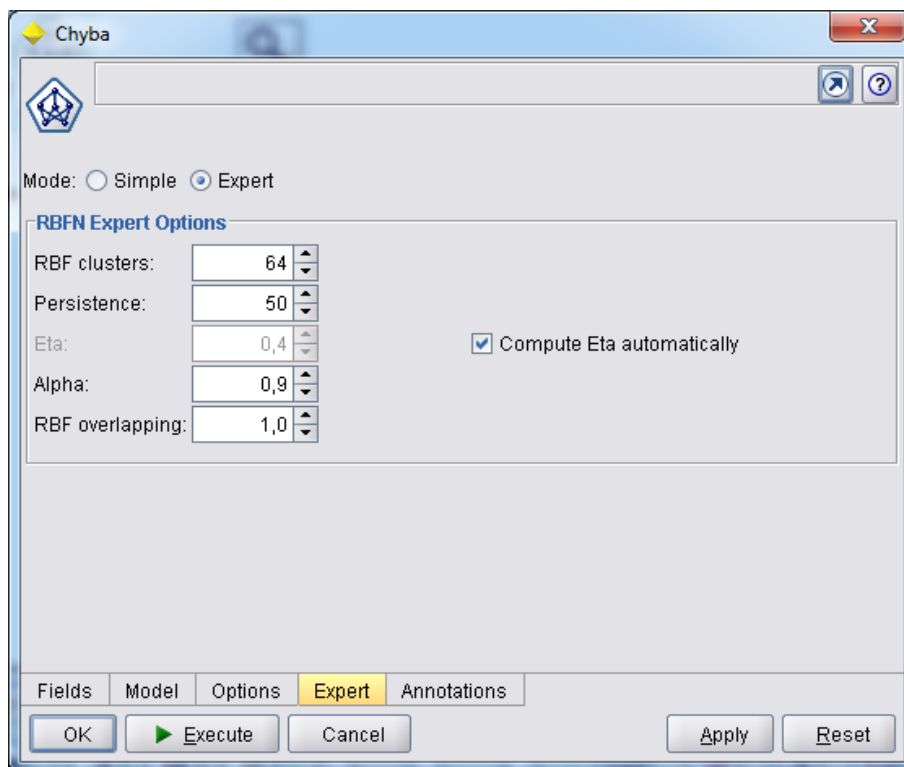


Obr. 22 – Nastavení RBF neuronové sítě, zdroj: [vlastní]

Další nastavitelné parametry jsou:

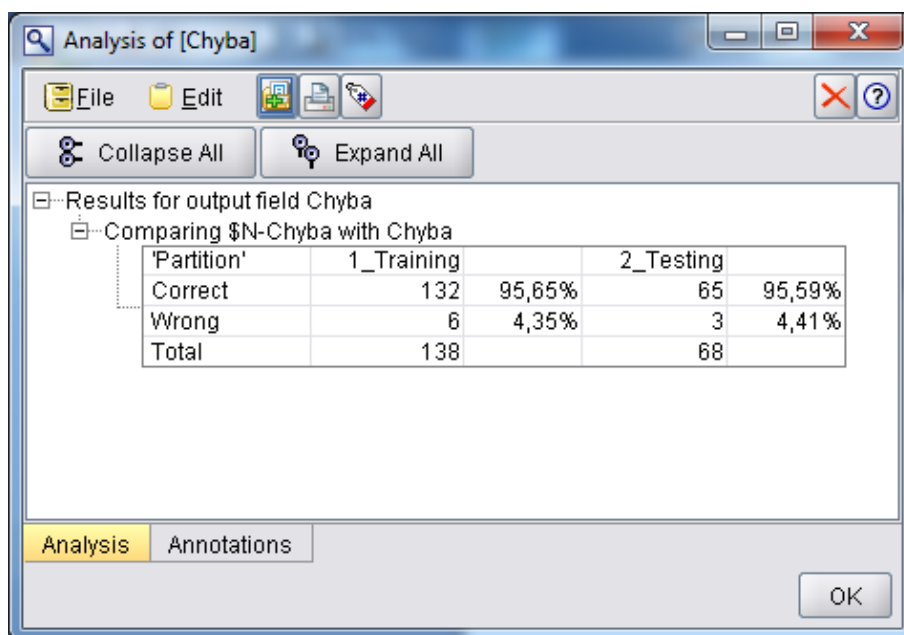
- RBF Clusters – nastavení počtu RBF neuronů ve skryté vrstvě;
- Persistence – nastavení počtu cyklů, po kterých pokračuje proces učení v případě, kdy již nedochází ke změně chyby klasifikace;
- Eta – nastavení rychlosti učení neuronové sítě;
- Alpha – nastavení parametru neuronové sítě vliv momentu;
- RBF overlapping – nastavení překrytí RBF neuronů.

Nastavení výše uvedených parametrů neuronové sítě zachycuje obr. 23. Parametr Persistence určuje počet učicích cyklů, po který bude síť nadále pracovat i v případě, kdy již nedochází k žádné změně vývoje chyby. Persistence je nastaven na hodnotu 50 cyklů. Parametr Eta je konstanta, která stanovuje rychlost učení neuronové sítě. V tomto případě je zvolena možnost nastavení této konstanty automaticky, což způsobí rychlejší učení neuronové sítě. Požadovaného výsledku je dosaženo s menším počtem učicích cyklů, protože parametr Eta je v průběhu učení automaticky dopočítáván dle aktuálního stavu.



Obr. 23 – Nastavení parametrů neuronové sítě, zdroj: [vlastní]

K prezentaci výsledků klasifikace navržené neuronové sítě je využit prostředek s názvem Analysis, který zobrazuje přesnost klasifikace na trénovací a testovací množině dat. Je ilustrován na obr. 24.



Obr. 24 – Výstup Analysis, zdroj: [vlastní]

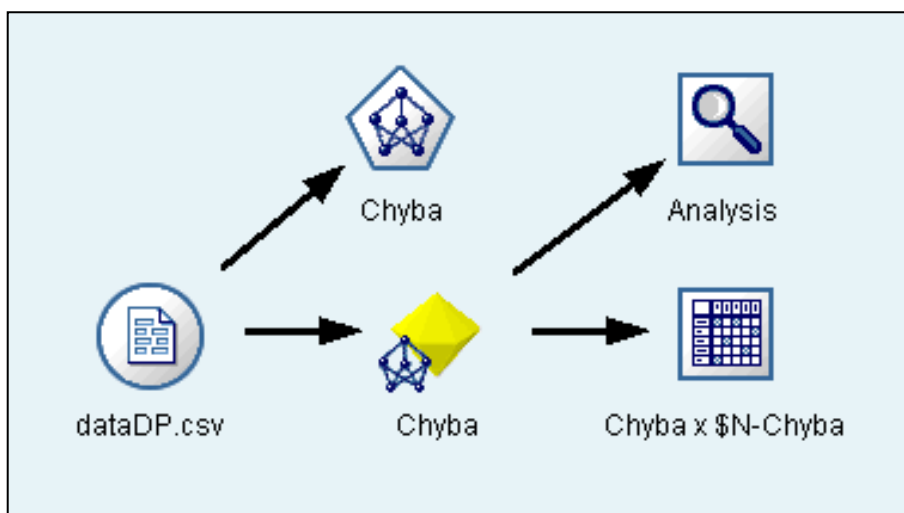
Ke zjištění počtu správně klasifikovaných prvků do jednotlivých tříd je použit nástroj Matrix. Výstupem tohoto nástroje je matice hodnot, kde v řádcích jsou znázorněny hodnoty původního atributu s názvem Chyba a ve sloupcích výstup klasifikace do těchto chybových tříd pomocí neuronové sítě. Na hlavní diagonále matice je počet shody v jednotlivých třídách. Ukázka výstupu je ilustrována na obr. 25.

Chyba	0	1	2	3	4
0	60	0	2	0	0
1	0	32	2	3	2
2	0	0	35	0	0
3	0	0	0	35	0
4	0	0	0	0	35

Cells contain: cross-tabulation of fields (including missing values)
Chi-square = 734,56, df = 16, probability = 0

Obr. 25 – Výstup Matrix, zdroj: [vlastní]

Celkový pohled na výše popsanou výslednou strukturu modelu sloužícího k návrhu RBF neuronových sítí, tedy provádění jednotlivých experimentů, je vyobrazen na obr. 26.



Obr. 26 – Struktura návrhu v SPSS Clementine, zdroj: [vlastní]

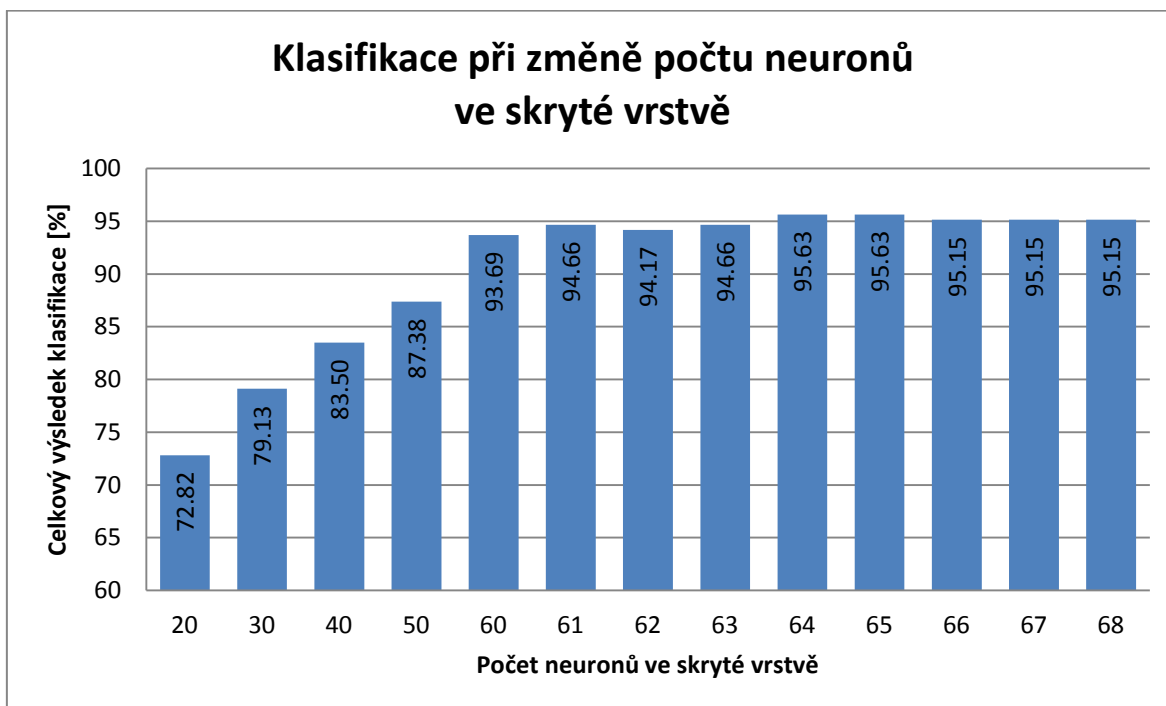
4.5.1 Počet neuronů ve skryté vrstvě

Výsledný počet neuronů ve skryté vrstvě, který významně ovlivňuje výsledek klasifikace, je dán na základě provedení experimentů. Je tedy vykonáno několik experimentů, jejichž výsledky znázorňuje tab. 4.

Tab. 4 – Výsledky klasifikace při změně počtu neuronů ve skryté vrstvě, zdroj: [vlastní]

Počet cyklů	Počet neuronů ve skryté vrstvě	Parametr Alpha	Překrytí	Klasifikace do tříd					Výsledky klasifikace [%]		Celkový výsledek klasifikace [%]
				0	1	2	3	4	Trénovací množina dat	Testovací množina dat	
400	20	0.9	1	47	15	35	25	28	74.64	69.12	72.82
400	30	0.9	1	49	16	35	35	28	81.16	75.00	79.13
400	40	0.9	1	50	24	35	35	28	84.78	80.88	83.50
400	50	0.9	1	55	27	35	35	28	88.41	85.29	87.38
400	60	0.9	1	60	28	35	35	35	94.20	92.65	93.69
400	61	0.9	1	59	31	35	35	35	94.93	94.12	94.66
400	62	0.9	1	59	30	35	35	35	94.20	94.12	94.17
400	63	0.9	1	60	30	35	35	35	94.93	94.12	94.66
400	64	0.9	1	60	32	35	35	35	95.65	95.59	95.63
400	65	0.9	1	60	32	35	35	35	95.65	95.59	95.63
400	66	0.9	1	59	32	35	35	35	95.65	94.12	95.15
400	67	0.9	1	59	32	35	35	35	95.65	94.12	95.15
400	68	0.9	1	59	32	35	35	35	95.65	94.12	95.15

Z tab. 4 je patrné, že jednotlivé experimenty jsou prováděny při konstantních hodnotách parametrů počet učicích cyklů, parametru Alpha a překrytí. Při vzrůstajícím počtu neuronů ve skryté vrstvě se zvyšuje přesnost klasifikace až k hodnotě počtu neuronů 64. Stejného výsledku je dosaženo i při nastavení počtu neuronů 65. Při dalším zvyšování tohoto počtu již přesnost klasifikace mírně klesá. Tento průběh je patrný z obr. 27, který uvádí přesnost klasifikace při změně počtu neuronů ve skryté vrstvě. Optimální počet je na základě těchto výsledků 64.



Obr. 27 – Klasifikace při změně počtu neuronů ve skryté vrstvě, zdroj: [vlastní]

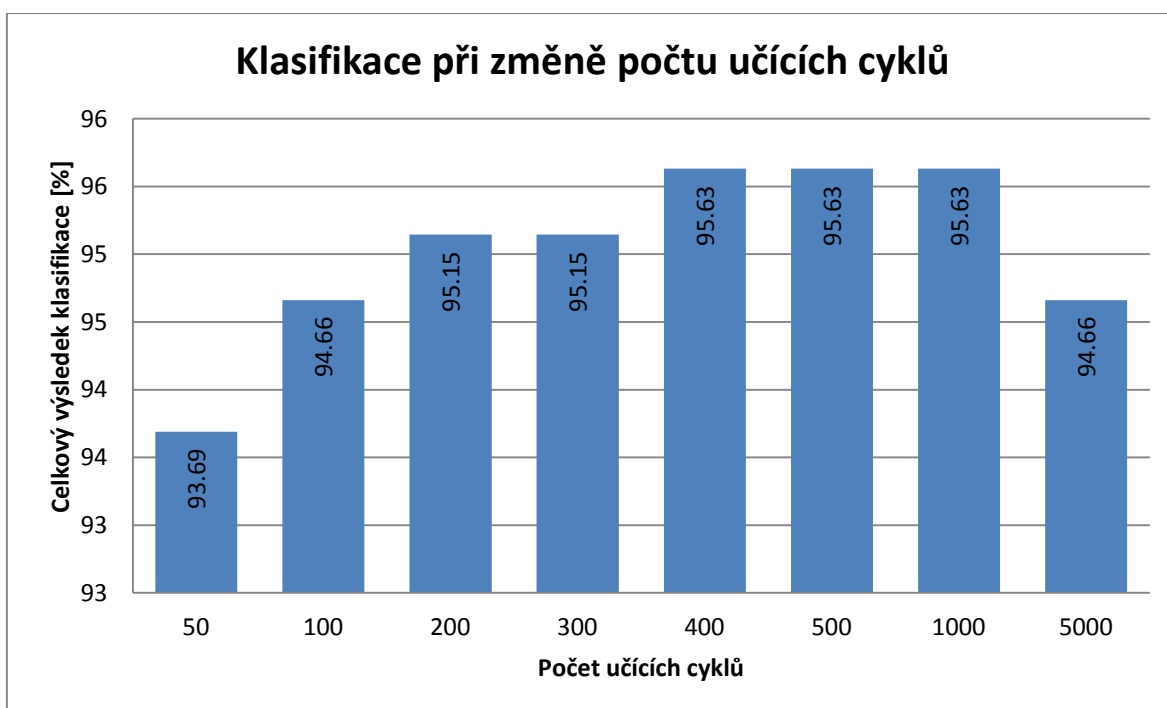
4.5.2 Počet učících cyklů

Výsledek klasifikace závisí na počtu cyklů učení neuronové sítě. Během učení dochází k postupnému snižování chyby klasifikace. Při dosažení minimální chyby je neuronová síť naučena a další učení sítě vede k tzv. přeučení neuronové sítě. V tomto případě se začne chyba opět zvyšovat.

Tab. 5 – Výsledky klasifikace při změně počtu učících cyklů, zdroj: [vlastní]

Počet cyklů	Počet neuronů ve skryté vrstvě	Parametr Alpha	Překrytí	Klasifikace do tříd					Výsledky klasifikace [%]		Celkový výsledek klasifikace [%]
				0	1	2	3	4	Trénovací množina dat	Testovací množina dat	
50	64	0.9	1	59	29	35	35	35	94.2	92.65	93.69
100	64	0.9	1	59	31	35	35	35	95.65	92.65	94.66
200	64	0.9	1	59	32	35	35	35	95.65	94.12	95.15
300	64	0.9	1	59	32	35	35	35	95.65	94.12	95.15
400	64	0.9	1	60	32	35	35	35	95.65	95.59	95.63
500	64	0.9	1	60	32	35	35	35	95.65	95.59	95.63
1000	64	0.9	1	60	32	35	35	35	95.65	95.59	95.63
5000	64	0.9	1	59	31	35	35	35	95.65	92.65	94.66

Zjištění optimálního počtu učících cyklů a průběh přesnosti klasifikace znázorňuje tab. 5. Jednotlivé experimenty jsou prováděny při konstantní hodnotě počtu neuronů ve skryté vrstvě, parametru Alpha a parametru překrytí. Optimální počet učících cyklů zjištěný na základě těchto experimentů je 400. Při této hodnotě je dosaženo klasifikace s nejvyšší možnou přesností. Při dalším zvyšování počtu cyklů je chyba pro hodnoty 500 a 1000 cyklů neměnná, pro 5000 cyklů se již přesnost klasifikace snižuje. Grafická ilustrace průběhu přesnosti klasifikace je zaznamenána na obr. 28.



Obr. 28 – Klasifikace při změně počtu učících cyklů, zdroj: [vlastní]

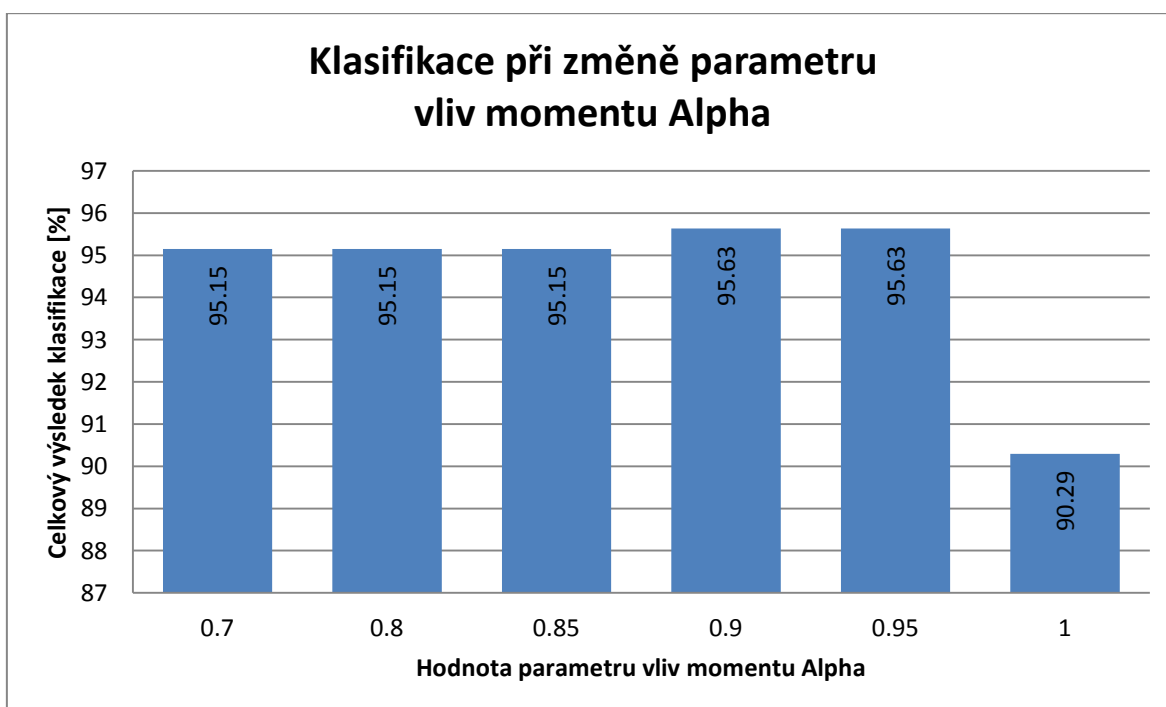
4.5.3 Vliv momentu Alpha

Dalším nastavovaným parametrem neuronové sítě je vliv momentu Alpha. Při učení sítě slouží tento parametr k pokračování učení i v případě, kdy již nastal lokální extrém vývoje chyby. Nalezení optimální hodnoty tohoto parametru, je prováděno za pomoci experimentů.

Tab. 6 – Výsledky klasifikace při změně parametru vliv momentu Alpha, zdroj: [vlastní]

Počet cyklů	Počet neuronů ve skryté vrstvě	Parametr Alpha	Překrytí	Klasifikace do tříd					Výsledky klasifikace [%]		Celkový výsledek klasifikace [%]
				0	1	2	3	4	Trénovací množina dat	Testovací množina dat	
400	64	0.7	1	59	32	35	35	35	95.65	94.12	95.15
400	64	0.8	1	59	32	35	35	35	95.65	94.12	95.15
400	64	0.85	1	59	32	35	35	35	95.65	94.12	95.15
400	64	0.9	1	60	32	35	35	35	95.65	95.59	95.63
400	64	0.95	1	60	32	35	35	35	95.65	95.59	95.63
400	64	1	1	51	30	35	35	35	94.93	80.88	90.29

Tab. 6 obsahuje výsledky klasifikace při změnách hodnoty parametru vlivu momentu. Ostatní nastavitelné hodnoty jsou konstantní. Nejlepší výsledek klasifikace je dosahován při nastavení Alpha na hodnotu 0.9 a 0.95. Jako optimální hodnota je zvolena hodnota 0.9. Přesnost klasifikace při změně parametru Alpha znázorňuje obr. 29.



Obr. 29 – Klasifikace při změně parametru vliv momentu Alpha, zdroj: [vlastní]

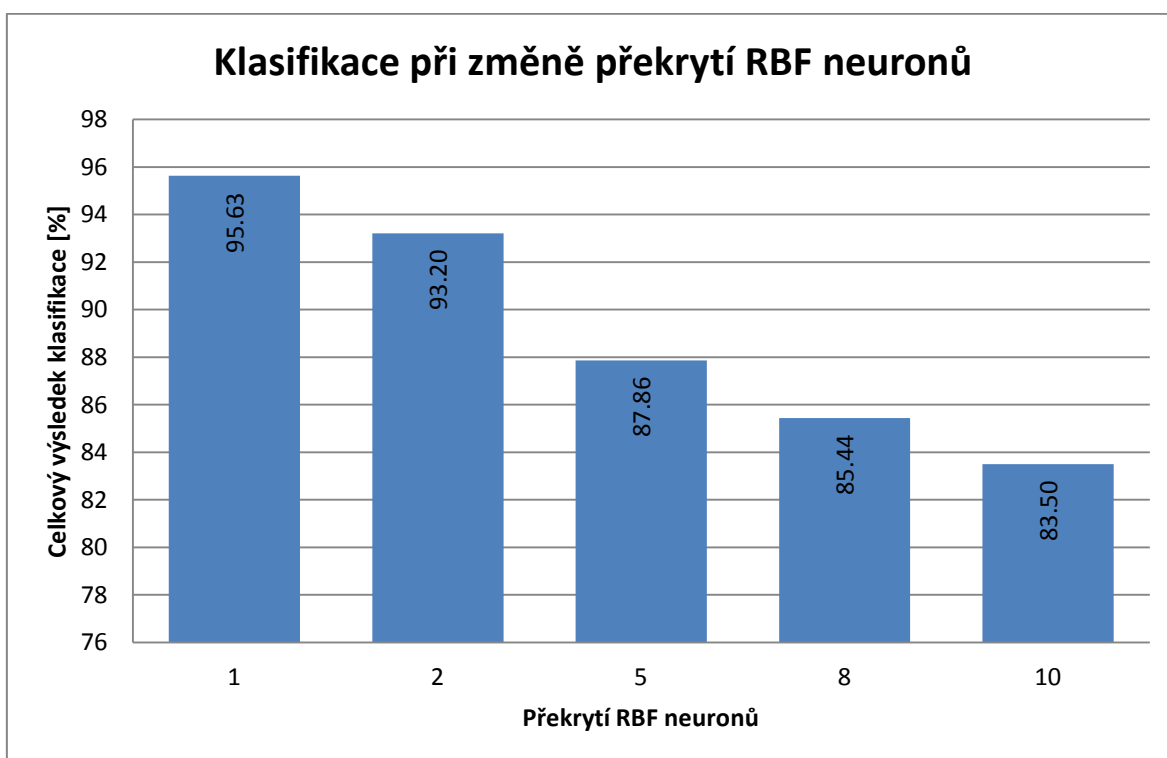
4.5.4 Vliv parametru překrytí

Další experimenty jsou prováděny při změně parametru překrytí RBF neuronů ve skryté vrstvě. Tyto neurony jsou chápány jako shluky dat. V případě nastavení parametru na hodnotu 1 není povoleno žádné překrytí shluků.

Tab. 7 – Výsledky klasifikace při změně parametru překrytí, zdroj: [vlastní]

Počet cyklů	Počet neuronů ve skryté vrstvě	Parametr Alpha	Překrytí	Klasifikace do tříd					Výsledky klasifikace [%]		Celkový výsledek klasifikace [%]
				0	1	2	3	4	Trénovací množina dat	Testovací množina dat	
400	64	0.9	1	60	32	35	35	35	95.65	95.59	95.63
400	64	0.9	2	58	29	35	35	35	93.48	92.65	93.20
400	64	0.9	5	54	27	35	30	35	90.58	82.35	87.86
400	64	0.9	8	55	23	35	35	28	88.41	79.41	85.44
400	64	0.9	10	50	24	35	35	28	83.33	83.82	83.50

Vliv změny nastavení parametru překrytí neuronů je patrný z výsledků obsažených v tab. 7. Optimální hodnotou nastavení je 1, tedy neuronová síť bez překrytí RBF neuronů. Při nastavení překrytí neuronů 2 a více přesnost klasifikace postupně klesá, což je graficky znázorněno v obr. 30.



Obr. 30 – Klasifikace při změně parametru překrytí, zdroj: [vlastní]

4.6 Dílčí závěr

Kapitola se zabývá návrhem neuronových sítí pro potřeby klasifikace. Nejprve je popsán postup vzniku vstupní datové matice a následně jsou tato data, která jsou předmětem klasifikace, analyzována. Dalším krokem je standardizace hodnot datové matice z důvodu odstranění velkých rozdílů a korelace. Mezi hodnotami jednotlivých atributů není zjištěna nijak významná korelační závislost. Pro návrh optimální neuronové sítě jsou data rozdělena na trénovací a testovací množinu v poměru 70:30. V dalším kroku jsou objasněny jednotlivé experimenty včetně jejich výsledků, které slouží k zjištění optimální struktury a parametrů neuronové sítě typu RBF.

5 Analýza výsledků

Cílem této kapitoly je analyzovat výsledky klasifikace dat do chybových tříd provedené pomocí RBF neuronových sítí, metodou učení s učitelem. Navrženou neuronovou sítí je zapotřebí ověřit provedením experimentu s validační množinou dat. Kapitola obsahuje výsledky klasifikace navrženým modelem včetně výsledků s přidáním validační množiny dat. Dále konkrétní parametry optimální navržené neuronové sítě RBF pro klasifikaci naměřených hodnot atributů virtuálních serverů Univerzity Pardubice Portal a Oracle.

5.1 Navržená RBF neuronová síť

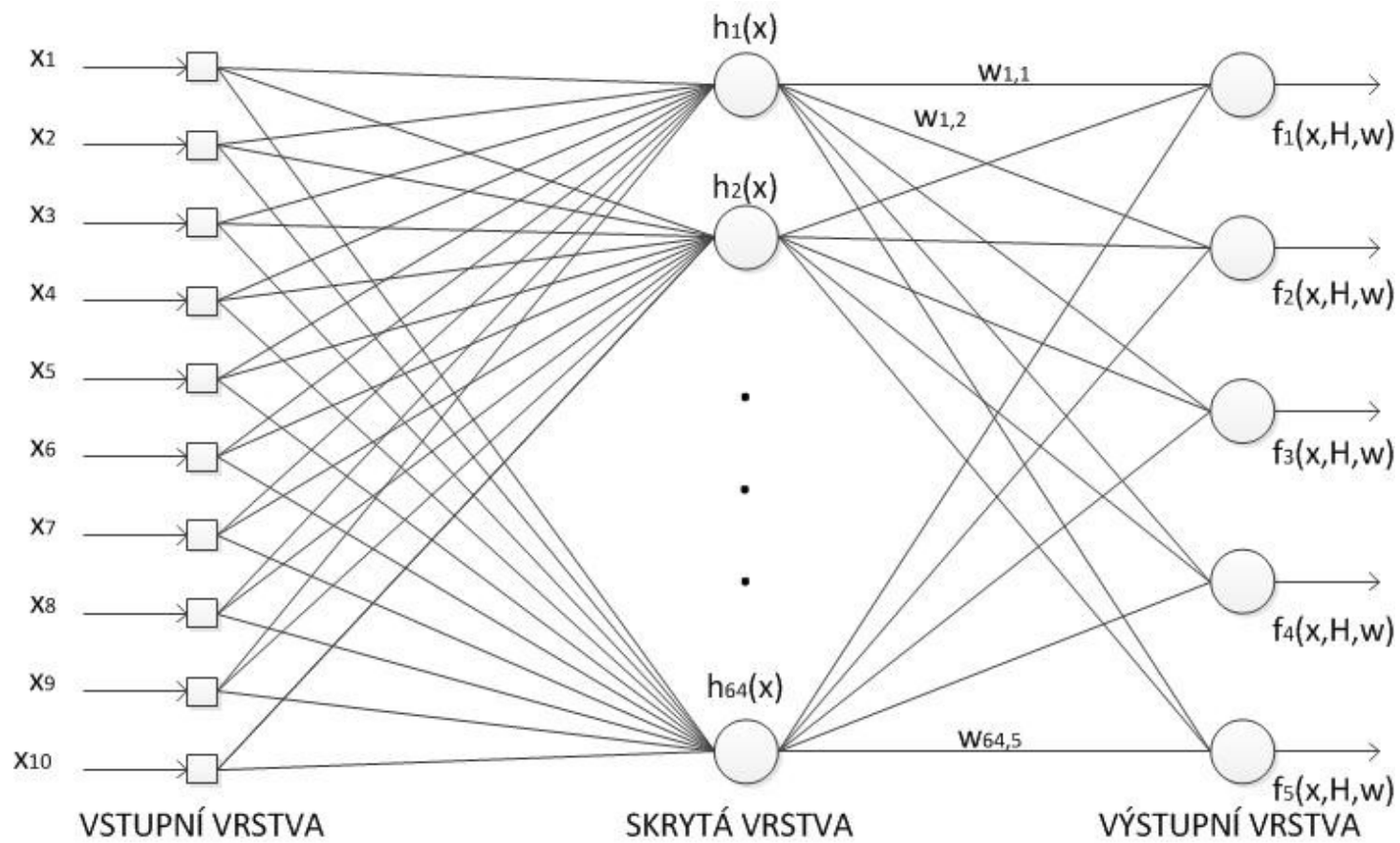
Na základě provedených experimentů je navržena neuronová síť s následující strukturou a parametry:

- vstupní vrstva – počet vstupních neuronů je 10, podle počtu atributů vstupní matice;
- skrytá vrstva – na základě experimentů voleno 64 RBF neuronů ve skryté vrstvě;
- výstupní vrstva – cílem je klasifikace do 5 tříd, proto obsahuje výstupní vrstva 5 výstupních neuronů;
- počet učících cyklů – dle výsledků experimentů zvoleno 400 učících cyklů;
- vliv momentu – z experimentů plyne volená hodnota 0.9;
- překrytí RBF neuronů ve skryté vrstvě – nejlepších výsledků je dosaženo při překrytí 1, tedy žádné překrývání RBF neuronů.

Výsledná struktura navržené neuronové sítě, která zachycuje její vstupní vrstvu, skrytou vrstvu a výstupní vrstvu, je znázorněna na obr. 31,

kde:

- $x = (x_1, x_2, \dots, x_{10})$ vstupní vektor;
- $H = \{h_1(x), h_2(x), \dots, h_{64}(x)\}$ množina RBF aktivačních funkcí;
- $w = \{w_{1,1}, w_{1,2}, \dots, w_{64,5}\}$ množina synapsí;
- $F = \{f_1(x, H, w), f_2(x, H, w), \dots, f_5(x, H, w)\}$ výstup, který je funkcí x , H a w .



Obr. 31 – Struktura navržené RBF neuronové sítě, zdroj: [vlastní]

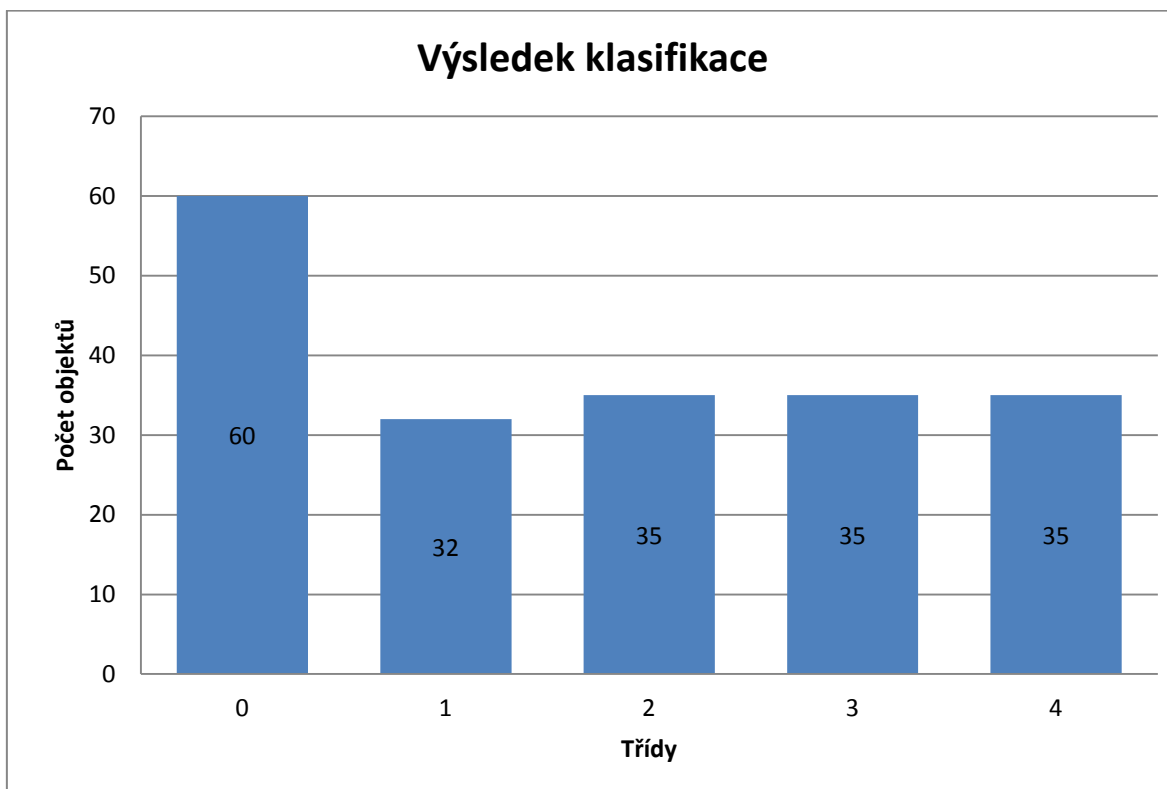
5.2 Výsledky klasifikace

Nejlepší výsledky klasifikace, tedy výsledky, kdy je dosaženo nejnižší hodnoty chyby, jsou zaznamenány v tab. 8. Tyto výsledky jsou získány na základě provedení experimentů s různým množstvím neuronů ve skryté vrstvě, počtem učicích cyklů, nastavením překrytí neuronů a vlivem momentu Alpha, jak je blíže popsáno v předchozí kapitole 4.5.

Tab. 8 – Výsledek klasifikace, zdroj: [vlastní]

Počet cyklů	Počet neuronů ve skryté vrstvě	Parametr Alpha	Překrytí	Klasifikace do tříd					Výsledky klasifikace [%]		Celkový výsledek klasifikace [%]
				0	1	2	3	4	Trénovací množina dat	Testovací množina dat	
400	64	0.9	1	60	32	35	35	35	95.65	95.59	95.63

Z tab. 8 je tedy patrný nejlepší celkový výsledek klasifikace pomocí RBF neuronových sítí s úspěšností 95.63 %, což se blíží hodnotě bezchybné klasifikace 100 %. Správné určení třídy jednotlivých objektů patřících do množiny trénovacích dat je dosaženo v 95.65 % všech případů. V množině dat testovacích je dosaženo správné klasifikace u 95.59 % objektů. Rozdíl v chybě mezi množinou dat trénovací a testovací je zanedbatelný, vypovídá to o spolehlivosti navrženého modelu. Tyto procentuální výsledky jsou vypočítány na základě počtu správně klasifikovaných objektů do patřících tříd, což je ilustrováno na grafu obr. 32. Do třídy 0 je správně klasifikováno 60 objektů, do třídy 1 je to 32 objektů. Vstupní datová matice obsahuje v třídě 0 objektů 62, v třídě 1 je 39 objektů. Třídy 2, 3 a 4 zahrnují každá 35 objektů, což je stejný počet jako ve vstupní matici. Neuronová síť tedy všechny objekty patřící do těchto tříd klasifikovala do jim odpovídajících tříd.



Obr. 32 – Výsledek klasifikace, zdroj: [vlastní]

Pro zjištění chyb klasifikace v konkrétních třídách je využita k reprezentaci výsledků matice, která znázorňuje počet objektů v jednotlivých třídách, viz tab. 9. Na hlavní diagonále se nachází počet objektů správně klasifikovaných do jednotlivých tříd. Proti tomu počet chybně klasifikovaných objektů leží mimo tuto diagonálu.

Tab. 9 – Matice výsledků klasifikace, zdroj: [vlastní]

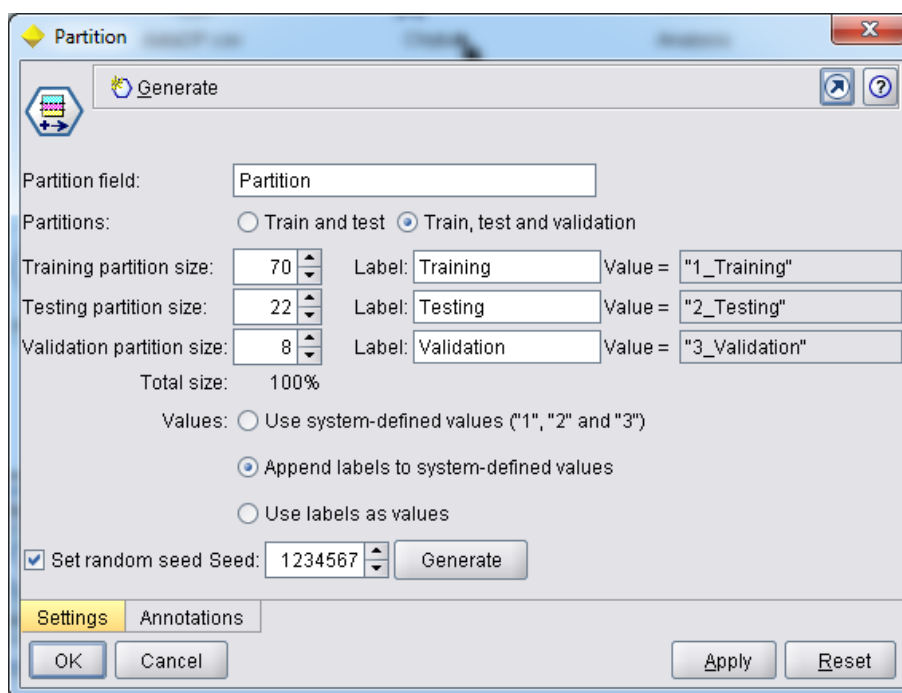
Skutečné třídy chyb	Třídy chyb RBF					
	Třída	0	1	2	3	4
0	60	0	2	0	0	
1	0	32	2	3	2	
2	0	0	35	0	0	
3	0	0	0	35	0	
4	0	0	0	0	35	

Z matice výsledků vyplývá, že výskyt chyby při klasifikaci nastal konkrétně u třídy 0 a třídy 1. Objekty tříd 2, 3, 4 jsou klasifikovány bez chyby. Chyby, které nastaly při klasifikaci objektů patřících do třídy 0, jsou dvě. Oba tyto objekty jsou chybně klasifikovány do třídy 2. Počet chybně klasifikovaných objektů patřících do třídy 1 je sedm. Dále klasifikovala neuronová síť chybně dva objekty do třídy 2, tři objekty do

třídy 3 a dva objekty do třídy 4. Z celkového počtu 206 objektů v datové matici je za pomoci navržené neuronové sítě RBF korektně zařazeno do odpovídajících tříd 195 objektů a 9 nekorektně. Ze zjištěných výsledků tedy vyplývá schopnost klasifikace dat měřených atributů virtuálních serverů pomocí navržené neuronové sítě typu RBF s poměrně vysokou přesností.

5.3 Ověření navrženého modelu neuronové sítě

K ověření navrženého modelu neuronové sítě je zapotřebí zavést validační množinu dat. Data patřící do trénovací a testovací množiny jsou totiž data, které má neuronová síť k dispozici během procesu učení. Validací množina dat však v procesu učení nevystupuje a z toho důvodu je možné jejím zavedením do neuronové sítě návrh ověřit. Vstupní data jsou znovu rozdělena pomocí nástroje Partition na trénovací, testovací a validační množinu dat, v poměru 70 %, 22 % a 8 %, což znázorňuje obr. 33.



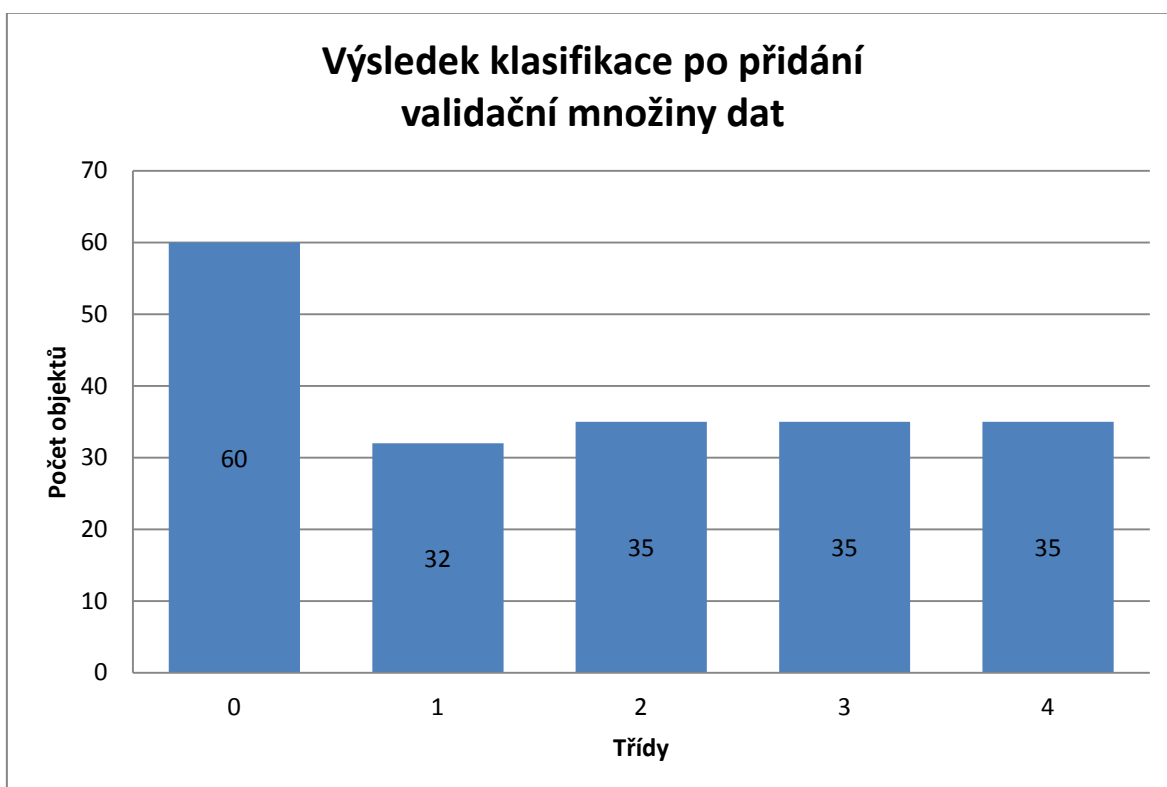
Obr. 33 – Nastavení Partition s validační množinou dat, zdroj: [vlastní]

Struktura a parametry neuronové sítě jsou nastaveny dle navrženého optimálního modelu. Ostatní využití nástroje a jejich nastavení se nemění a jsou již popsány v kapitole 5.1. Provedením klasifikace s validační množinou dat jsou získány výsledky uvedené v tab. 10 a vyobrazené na obr. 34.

Tab. 10 – Výsledek klasifikace po přidání validační množiny dat, zdroj: [vlastní]

Počet cyklů	Počet neuronů ve skryté vrstvě	Parametr Alpha	Překrytí	Klasifikace do tříd					Výsledky klasifikace [%]			Celkový výsledek klasifikace [%]
				0	1	2	3	4	Trénovací množina dat	Testovací množina dat	Validační množina dat	
400	64	0,9	1	60	32	35	35	35	95.65	94.74	96.67	95.63

Z tabulky je patrné, že po přidání validační množiny dat dosahuje navržená neuronová síť stejných výsledků jako v předchozím případě při jejím návrhu. Dá se tedy rozhodnout o správnosti navrženého modelu. Na základě těchto dosažených výsledků je možné tvrzení, že navržený model je dostatečně spolehlivým a přesným klasifikátorem chybových stavů virtuálních serverů Portal a Oracle.



Obr. 34 – Výsledek klasifikace po přidání validační množiny dat, zdroj: [vlastní]

5.4 Dílčí závěr

Zaměření této kapitoly se vztahuje k výsledkům klasifikace získaných navrženým modelem neuronové sítě RBF. Nejprve je popsána struktura a parametry navržené neuronové sítě. Dosažené výsledky jsou zanalyzovány. Z analýzy plyne dostatečná přesnost klasifikace. Dále je popsáno ověření správnosti navrženého modelu za pomoci validační množiny dat, z jehož výsledků vyplývá, že navržený model je spolehlivý a dostatečně přesný pro potřeby klasifikace dat naměřených na virtuálních serverech.

ZÁVĚR

Práce je zaměřená na klasifikaci dat virtuálních serverů pomocí neuronových sítí typu RBF. Nejprve pojednává o virtualizaci serverů a možnostech jejího využití. Dále jsou popsány základní pojmy z oblasti neuronových sítí jako například učení neuronových sítí, jejich struktura, všeobecný klasifikační problém a neuronové sítě typu RBF.

Cílem práce bylo vytvořit model neuronové sítě typu RBF, který je schopen s co nejvyšší přesností klasifikovat předložená data naměřená na virtuálních serverech Portal a Oracle do tříd. Tyto třídy reprezentují možné chybové stavy, které se mohou na serverech vyskytnout. Velké množství dat je nejprve sjednoceno na základě identifikátoru datum do jedné datové matice a následně jsou odstraněny záznamy s chybějícími hodnotami. Celková matice dat obsahuje 41958 platných záznamů měření. Objemná datová matice je nejdříve za pomoci metod shlukové analýzy zredukována na matici o 206 záznamech, obsahující 10 měřených atributů. Tato data jsou dále standardizována. Vstupní vektor neuronové sítě obsahuje všech 10 měřených atributů, a to na základě korelační analýzy, pomocí které není mezi těmito atributy zjištěna významná korelační závislost. Učitelem neuronové sítě je atribut chyba nesoucí informaci o skutečném přiřazení záznamu chybového stavu serverů.

Na základě provedení experimentů je navržena neuronová síť s počtem 64 RBF neuronů ve skryté vrstvě bez povolení jejich vzájemného překrývání. Je naučena během 400 učících cyklů, s nastaveným parametrem vliv momentu na hodnotu 0.9. Tato navržená neuronová síť je ověřena pomocí validační množiny dat a je schopna požadovanou klasifikaci provést s přesností 95.63 %. Neuronová síť typu RBF je tedy vhodným prostředkem pro klasifikaci těchto dat.

POUŽITÁ LITERATURA

- [1] HÁJEK, J. *Virtualizace snadno a rychle*. IT Systems, 2007, 7 - 8, s. 52 – 53. ISSN 1802-002X
- [2] KREMSER, L. *Virtualizace serverů, cesta k vyšší efektivitě IT*. Fujitsu - Siemens Computers [online]. 2005 [cit. 2011-12-13]. Dostupné z WWW: <www.64bit.cz/soubory/3661/prezentace-o-virtualizaci-it.ppt?ms=3>
- [3] TOMEČEK, J. *Virtualizace na úrovni jádra operačního systému*. abclinuxu.cz. [online]. 2007-07-31 [cit. 2011-12-03]. Dostupné z WWW: <<http://www.abclinuxu.cz/clanky/system/virtualizace-na-urovni-jadra-operacniho-systemu>>
- [4] ANDREJKOVIČ, J. *Klonovanie tučniaka alebo virtualizácia v Linuxe: úvod*. Root.cz. [online]. 2005-10-04 [cit. 2011-12-02]. Dostupné z WWW: <<http://www.root.cz/clanky/klonovanie-tucniaka-alebo-virtualizacia-v-linuxe-uvod/>>
- [5] *Virtualizace serverů*. Microsoft Private Cloud. [online]. 2011 [cit. 2011-11-20]. Dostupné z WWW: <<http://www.microsoft.com/cze/privatnicloud/produkty/server.aspx>>
- [6] *Virtualizace*. MANAG a.s., Divize 52 - ICT. [online]. 2008-09 [cit. 2011-12-05]. Dostupné z WWW: <http://www.manag.com/public/data/data/Virtualizace_v1.3.pdf>
- [7] TULLOCH, M. a kol. *Understanding Microsoft Virtualization Solutions*. Microsoft Press. [online]. 2010 [cit. 2012-02-01]. Dostupné z WWW: <<http://download.microsoft.com/DOWNLOAD/5/B/4/5B46A838-67BB-4F7C-92CB-EABCA285DFDD/693821EBOOK.pdf>>
- [8] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol. *Umělá inteligence (4)*. Praha: Akademie věd České republiky, 2003. 471 s. ISBN 80-200-1044-0
- [9] OLEJ, V., HÁJEK, P. *Úvod do umělé inteligence. Moderní přístupy. Distanční opora*. Pardubice: Univerzita Pardubice, 2010. 98 s. ISBN 978-80-7395-307-2
- [10] KVASNIČKA, V. a kol. *Úvod do teórie neuronových sietí*. Bratislava: IRIS, 1997. 264 s. ISBN 80-88778-30-1
- [11] ŠÍMA, J., NERUDA, R. *Teoretické otázky neuronových sítí*. Praha: MATFYZPRESS, UK, 1996. 390 s. ISBN 80-85863-18-9

- [12] OLEJ, V., HÁJEK, P. *Municipal Creditworthiness Modelling by Radial Basis Function Neural Networks and Sensitive Analysis their Inputs Parameters*. 19th International Conference on Artificial Neural Network, ICANN 2009, 14-17 September, Limassol, Cyprus, Alippi, C., Polycarpou, M., Panayiotou, Ch., Ellinas, G., Eds., Springer Berlin Heidelberg New York, 2009, pp.505-514, ISSN 0302-9743, ISBN 3-642-04276-7
- [13] HAYKIN, S. *NEURAL NETWORKS : A Comprehensive Foundation*. New Jersey: Prentice Hall, 1999. 2nd edition 823 p. ISBN 81-7808-300-0
- [14] PARK, J., SANDBERG, I. W. *Universal Approximation using Radial Basis Function Network*. Neural Computation, 1991. vol.3, pp. 246-257
- [15] BROOMHEAD, D. S., LOWE, D. *Multivariate Functional Interpolation and Adaptive Networks*. Complex Systems, 1988. vol.2, pp. 321-355
- [16] PARK, J., SANDBERG, I. W. *Approximation and Radial Basis Function Network*. Neural Comput, 1993. vol.5, pp. 305-316
- [17] NIYOGI, P., GIROSI, F. *On the Relationship Between Generalization Error, Hypothesis Complexity, and Sample Complexity for Radial Basis Functions*. [online]. 1994-02 [cit. 2012-03-11]. Dostupné z WWW: < <http://18.7.29.232/bitstream/handle/1721.1/6624/AIM-1467.pdf?sequence=2> >

SEZNAM OBRÁZKŮ

Obr. 1 – Ukázka virtualizace serverů, zdroj: [2]	11
Obr. 2 – Schéma hardwarové virtualizace, zdroj: [1].....	14
Obr. 3 – Schéma nativní virtualizace, zdroj: [2].....	15
Obr. 4 – Schéma virtualizace na úrovni operačního systému, zdroj: [1].....	16
Obr. 5 – McCulloch-Pittsův neuron, zdroj: [9]	18
Obr. 6 – Ukázka dopředné jednovrstvé neuronové sítě, zdroj: [11].....	21
Obr. 7 – Ukázka dopředné vícevrstvé neuronové sítě, zdroj: [11].....	22
Obr. 8 – Ukázka rekurentní neuronové sítě, zdroj: [11].....	23
Obr. 9 – Všeobecný klasifikační problém, zdroj: [10]	24
Obr. 10 – Struktura neuronové sítě, zdroj: [12].....	27
Obr. 11 – Vliv množství vstupních dat a počtu RBF neuronů na trénovací chybu, zdroj: [17]	30
Obr. 12 – Vliv počtu RBF neuronů na zevšeobecňující chybu, zdroj: [17]	31
Obr. 13 – Vliv množství vstupních dat a počtu RBF neuronů na zevšeobecňující chybu, zdroj: [17]	31
Obr. 14 – Návrh modelu, zdroj: [vlastní]	34
Obr. 15 – Struktura modelu shlukové analýzy, zdroj: [vlastní]	37
Obr. 16 – Centra shluků, zdroj: [vlastní]	37
Obr. 17 – Rozložení jednotlivých objektů v třídách, zdroj: [vlastní]	38
Obr. 18 – Rozdělení množiny na trénovací a testovací, zdroj: [10]	42
Obr. 19 – Nastavení Partition, zdroj: [vlastní].....	43
Obr. 20 – Struktura modelu rozdělení dat, zdroj: [vlastní].....	43
Obr. 21 – Nastavení charakteru proměnných datové matice, zdroj: [vlastní]	44
Obr. 22 – Nastavení RBF neuronové sítě, zdroj: [vlastní]	45
Obr. 23 – Nastavení parametrů neuronové sítě, zdroj: [vlastní].....	46

Obr. 24 – Výstup Analysis, zdroj: [vlastní].....	46
Obr. 25 – Výstup Matrix, zdroj: [vlastní].....	47
Obr. 26 – Struktura návrhu v SPSS Clementine, zdroj: [vlastní].....	47
Obr. 27 – Klasifikace při změně počtu neuronů ve skryté vrstvě, zdroj: [vlastní].....	49
Obr. 28 – Klasifikace při změně počtu učících cyklů, zdroj: [vlastní].....	50
Obr. 29 – Klasifikace při změně parametru vliv momentu Alpha, zdroj: [vlastní].....	51
Obr. 30 – Klasifikace při změně parametru překrytí, zdroj: [vlastní].....	52
Obr. 31 – Struktura navržené RBF neuronové sítě, zdroj: [vlastní].....	55
Obr. 32 – Výsledek klasifikace, zdroj: [vlastní].....	57
Obr. 33 – Nastavení Partition s validační množinou dat, zdroj: [vlastní].....	58
Obr. 34 – Výsledek klasifikace po přidání validační množiny dat, zdroj: [vlastní].....	59

SEZNAM TABULEK

Tab. 1 – Datový slovník, zdroj: [vlastní].....	36
Tab. 2 – Popis chybových stavů serverů, zdroj: [vlastní].....	38
Tab. 3 – Korelační matice, zdroj: [vlastní].....	40
Tab. 4 – Výsledky klasifikace při změně počtu neuronů ve skryté vrstvě, zdroj: [vlastní]	48
Tab. 5 – Výsledky klasifikace při změně počtu učících cyklů, zdroj: [vlastní]	49
Tab. 6 – Výsledky klasifikace při změně parametru vliv momentu Alpha, zdroj: [vlastní]	51
Tab. 7 – Výsledky klasifikace při změně parametru překrytí, zdroj: [vlastní].....	52
Tab. 8 – Výsledek klasifikace, zdroj: [vlastní].....	56
Tab. 9 – Matice výsledků klasifikace, zdroj: [vlastní]	57
Tab. 10 – Výsledek klasifikace po přidání validační množiny dat, zdroj: [vlastní]	59

SEZNAM ZKRATEK

DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
DR	Disaster Recovery
RAM	Random Access Memory
PC	Personal Computer
CPU	Central Processing Unit
API	Application Programming Interface
IT	Informační technologie
RBF	Radial Basis Function
MHz	Mega Hertz