Mendelova univerzita v Brně
Provozně ekonomická fakulta

Vlastnosti a použití umělých neuronových sítí založených na teorii adaptivní rezonance

Diplomová práce

Vedoucí práce:
doc. Ing. Jan Žižka, CSc.

Bc. Ondřej Janů

Brno 2015

Chtěl bych poděkovat panu doc. Ing. Janu Žižkovi, CSc. za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat. Dále bych chtěl poděkovat Ing. Lucii Blahové za psychickou podporu po celou dobu studia.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Vlastnosti a použití umělých neuronových sítí založených na teorii adaptivní rezonance** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 18. prosince 2015 _____

Abstract

The recent progress in information and communication technologies has enabled us to obtain and store very large amounts of data. The main problem is how to find and extract useful information contained in data. The main goal of this study is to assess the possibility of using Adaptive resonance theory (ART) neural networks for cluster analysis and information retrieval. Their properties, behavior and success rate for different types of artificial and real data were determined as well as optimal values of their parameters for this purpose. Based on achieved results, the possibility of using ART neural networks for Big data analysis was assessed. Then the application based on ART principles with included graphical user interface was implemented and this process was described.

Keywords

neural networks, adaptive resonance theory, data mining, clustering

Abstrakt

Aktuální pokrok v informačních a komunikačních technologií nám umožňují uchovávat stále větší objemy dat. Získat z těchto dat užitečné informace není jednoduché. Tato práce se zabývá možností získání těchto informací pomocí shlukovací analýzy dat s využitím neuronových sítí založených na teorii adaptivní resonance (ART). Především je kladen důraz na vlastnosti a úspěšnost, které tyto sítě na různých typech dat vykazují, a určení optimálních parametrů pro testovaná data. Na základě získaných výsledků byly ART neuronové sítě zhodnoceny z hlediska možnosti jejich využití pro analýzu Big data. Dále je v této práci popsán postup, na základě kterého je možné implementovat aplikaci realizující principy ART včetně grafického uživatelského rozhraní.

Klíčová slova

neuronové sítě, teorie adaptivní resonance, dolování znalostí z dat, shlukování

Obsah

1 Úvod a cíl práce

1.1 Úvod

Dříve jsme údaje o důležitých událostech či objevech uchovávali především pomocí knih a jiných písemných záznamů. S příchodem informačních technologií se objevil nový způsob uchování těchto informací a bylo možné uchovávat stále vetší množství záznamů. S rostoucím množstvím záznamů rostla i potřeba je dále zpracovávat a vytěžovat z nich co největší množství informací. Tehdejší metody byly pro velké množství dat neefektivní a vznikla potřeba proces zpracování dat automatizovat. Za tímto účelem vzniklo velké množství modelů založených především na statistice a matematice.

V současnosti je sběr dat stále snazší a mnohdy je nutné zpracovávat tak velké objemy dat, na kterých je využití mnoha moderních automatizovaných metod neefektivní. Taková data označujeme často termínem Big data. Vhodným adeptem pro analýzu takto objemných dat jsou neuronové sítě založené na teorii adaptivní resonance (ART).

Pro zpracování Big data jsou ART sítě vhodné hned z několika důvodů. Prvním důvodem je fakt, že záznamy zpracovávají sekvenčně a nepotřebují je v průběhu svého učení udržovat všechny zároveň. Z tohoto důvodu je možné zpracovávat i velmi velké množství záznamů. Jednotlivé prvky každého záznamu jsou pomocí těchto sítí zpracovávány odděleně, a proto není problém analyzovat i mnohorozměrná data. Proces učení je možné libovolně prokládat s procesem kategorizace, díky tomu můžeme již naučenou síť přizpůsobit novým skutečnostem bez nutnosti opakovat celý proces učení.

Koncept ART sítí je znám již od roku 1976, ovšem od té doby byl mnohokrát zdokonalen. Tato práce pojednává o využití moderních modelů ART neuronových sítí v praxi a o jejich vlastnostech.

1.2 Cíl práce

Jedním z cílů této práce je obeznámit čtenáře s metodami a principy dolování znalostí z dat a s využitím neuronových sítí v této oblasti. Důraz je kladen především na neuronové sítě založené na teorii adaptivní rezonance (ART).

Hlavním cílem této práce je otestovat chování vybraných architektur ART neuronových sítí na různých typech dat. Na základě získaných informací budou definovány vlastnosti vybraných modelů, dále bude stanovena jejich úspěšnost pro jednotlivá testovaná data v závislosti na zvolených parametrech sítě a následně budou vyhodnoceny možnosti využití zvolených modelů v praxi.

Pro realizaci vybraných modelů bude nutné vybrat vhodné softwarové řešení a následně tyto modely implementovat. Dále bude vytvořeno grafické uživatelské rozhraní pro usnadnění správy a obsluhy neuronové sítě.

2 Současný stav a vývoj dolování znalostí z dat

Dolování dat (data mining) je součástí počítačové vědy a v současnosti se vyvíjí stále rostoucím tempem. Využívá poznatky a postupy z mnoha různých vědních oblastí. *„Mezi tyto oblasti řadíme především statistiku, strojové učení, umělou inteligenci a databázové systémy [1].“*

S rozvojem informačních technologií je možné získávat stále větší objemy dat a je kladen stále větší důraz na jejich maximální využití. Ovšem získání užitečné znalosti z dat je velmi komplikovaný proces. Dolování dat obecně slouží právě k odhalení na první pohled skrytých struktur a znalostí a jejich převedení do člověku srozumitelné formy [1]. Odhalení užitečných znalostí a struktur výrazně usnadní proces rozhodování a provádění dalších analýz.

2.1 Historický vývoj

Samotný pojem data mining je sice záležitostí 20. a 21. století, ovšem některé teoretické principy, které v této vědní disciplíně využíváme, sahají mnohem dál do minulosti.

V roce 1763 vyšel posmrtně článek anglického duchovního Thomase Bayese (1701–1761), ve kterém byl podrobně popsán vztah mezi podmíněnou pravděpodobností určitého jevu a pravděpodobností opačně podmíněného jevu. Tento vztah je v dnešní době znám jako Bayesova věta a je základem pro Bayesův teorém [2]. Na základě Bayesova teorému fungují současné klasifikátory, jako je například Naivní Bayesovský klasifikátor [3].

O 42 let později v roce 1805 dva němečtí matematici Adrien-Marie Legendre a Carl Friedrich Gauss dali základ regresní analýze, když se pokoušeli přesně určit oběžné dráhy vesmírných těles okolo slunce [2]. K tomuto účelu využili metodu nejmenších čtverců, kterou využíváme dodnes pro stanovení vztahů mezi proměnnými [4].

Dalším velmi důležitým milníkem byl rok 1936, kdy ve svém článku britský matematik a kryptoanalytik Alan Turing popsal univerzální stroj, který je dnes známí jako Turingův stroj [2]. Tím udal základní principy, na kterých fungují moderní počítače, a umožnil sbírat a zpracovávat velké objemy dat.

Warren McCulloch a Walter Pitts v roce 1943 vytvořili model neuronové sítě, ve kterém neuron popisují jako základní jednotku sítě, která může přijímat vstupy, zpracovat je a vygenerovat patřičný výstup [5].

V roce 1965 založil americký doktor Lawrence J. Fogel společnost Decision Science, Inc., která se specializovala na vývoj evolučních algoritmů založených na principech inspirovaných Darwinovou teorií evoluce druhů [2]. Dále v roce 1975 byla vydána revoluční kniha „Adaptation in Natural and Artificial Systems“, která se stala základem právě pro evoluční programování [5].

V roce 1980 firma HNC vytvořila nástroj nazvaný DataBase Mining Workstation (pracovní stanice pro dolování databází), který sloužil k vytváření modelů umělých neuronových sítí, a k ochraně tohoto produktu si nechala zaregistrovat obchodní značku „database mining“ (dolování databází). Další důležitý pojem v oblasti dolování dat je „Knowledge Discovery in Databases – KDD“ (objevování znalostí v databázích) a byl definován Gregorym Piatetsky-Shapirem v roce 1989. Ve stejném roce se také pořádal první KDD workshop. V roce 1990 se poprvé objevuje pojem data mining a je hojně využíván pro ekonomické analýzy ve velkých firmách. Mezi roky 1990 a 2015 byly postupy využívané pro dolování dat neustále vylepšovány, stávaly se populárnějšími a byly využívány stále větším množstvím firem [2].

V současnosti je dolování dat (data mining) nedílnou součástí mnoha oborů, mezi které patří například management, věda, stavebnictví a zdravotnictví. „*Používá se například pro dolování finančních transakcí, měnových trhů, pro účely národní bezpečnosti, mapování genomu, podporu rozhodování a další [2].*“ Protože se data sbírají daleko jednodušeji než dříve, jejich objem stoupá a stále častěji se setkáváme s pojmem Big Data [6].

2.2 Big data

Big data jsou v současnosti velmi diskutovaným tématem. Jedná se o taková data, která jsou rozsahem tak obsáhlá nebo složitá, že použití tradičních metod zpracování dat, je buď velmi komplikované, nebo nemožné. Analýza takových dat je sice složitá, ale má potenciál poskytnout mnohem přesnější výsledky.

Sběr takto rozsáhlých dat umožňuje především rozvoj internetu, elektroniky, informatiky a stálé zvětšování úložného prostoru. Pomocí mobilních zařízení, sociálních sítí, satelitů, kamer, RFID čteček a dalších senzorů je možné taková data získat poměrně levně.

Big data můžeme popsat pomocí následujících šesti charakteristik:

- *Objem* dat je samozřejmě jednou z nejdůležitějších charakteristik. Na základě objemu dat můžeme částečně posoudit, jaký mají data potenciál a zda se skutečně jedná o Big data.
- *Rozmanitost* dat je základní vlastností pro každého datového analytika. Tato znalost usnadňuje samotnou analýzu a umožňuje její jednodušší průběh. To je při práci s Big data jedním z klíčových faktorů.
- *Rychlost*, se kterou data musejí být generována a zpracovávána, aby bylo vyhověno stanoveným požadavkům.
- *Variabilitou* myslíme nestálost dat. K nestálosti často dochází, pokud jsou některé informace uchovávány redundantně a každý zdroj poskytuje jinou informaci o stejném objektu. Nestálost dat má negativní vliv na výsledek analýzy.
- *Kvalita a věrohodnost* dat má přímý vliv na výsledek analýzy. Jsou-li data málo věrohodná, je třeba zvážit, zda jsou vhodná pro analýzu.

- *Složitost* dat má zásadní vliv na proces zpracování. Správa dat může být velmi obtížná, obzvlášť pokud rozsáhlá data pocházejí z více zdrojů. Taková data musí být správně propojena na základě správně stanovených vztahů tak, aby uživatel pochopil informaci, kterou mají data sdělovat [7].

2.3 Základní pojmy strojového učení

Strojové učení patří mezi počítačové vědy a zabývá se tvorbou a využitím algoritmů, které jsou schopné se učit na přijatých datech, a na základě získané znalosti předpovídat vlastnosti nových dat. Využívají se pro dolování dat.

Algoritmy strojového učení se obvykle používají v takových situacích, ve kterých by realizace algoritmu pomocí striktních instrukcí byla buď nemožná, nebo nevhodná. Mezi takové situace patří například rozpoznávání písmen získaných optickým zařízením, realizace vyhledávačů, kategorizace textu a jiné [8].

Hlavním cílem algoritmu strojového učení je generalizovat zkušenosti získané analýzou dat a na základě znalosti získané generalizací kategorizovat další nová data [8]. Analyzovaná data mohou obsahovat na první pohled neznámé struktury a vzory, které je nutné odhalit a popsat pomocí obecného modelu, který následně slouží ke kategorizaci.

2.3.1 Rozpoznávání vzorů

Rozpoznávání vzorů je nedílnou součástí strojového učení. Cílem této disciplíny je rozpoznávání objektů (vzorů) a kategorizace objektů do tříd, které sdružují objekty s podobnými vlastnostmi. Struktura a vlastnosti objektů se u různých typů problémů liší. Objekty mohou reprezentovat obrázky, vlnové délky nebo jakékoliv jiné naměřené veličiny, které chceme kategorizovat [9]. O objektech budeme dále mluvit jako o vzorech.

Rozpoznávání vzorů má dlouhou historii, ale až do roku 1960 se provádělo jen na teoretické a statistické úrovni. Stejně jako u celé řady dalších vědních oborů, rychlý rozvoj výpočetní techniky umožnil využití této vědní disciplíny v praxi i pro objemná data. Stále rostoucí zájem o praktické využití vedl ke zvyšování požadavků a k dalšímu rozvoji na teoretické úrovni. S přechodem naší společnosti z industriální do postindustriální doby nabývá automatizace výroby a sběr a analýza informací stále většího významu [9]. Tento rostoucí trend posouvá rozpoznávání vzorů na hranici současných možností.

Jedna z oblastí, ve které využíváme rozpoznávání vzorů velice často, je strojové vidění. Data bývají obvykle nasnímána pomocí kamery. Poté jsou analyzována a převedena do takové formy, aby co nejlépe vystihovala nasnímaný obraz. Tento postup se využívá například pro výstupní kontrolu výrobků firmy a detekci případných defektů [9].

Další oblastí, kde se hojně využívá rozpoznávání vzorů, je automatické rozpoznávání znaků (OCR – optical character recognotion). OCR systémy ke své činnosti obvykle využívají světelný zdroj, skenovací čočky, zařízení pro manipulaci s dokumentem a detektor. Intenzita odraženého světla je převedena na číselné

hodnoty a ty jsou následně rozděleny na sekvence znaků [9]. S tímto principem se můžeme běžně setkat například u skeneru nebo RFID čtečky.

Počítačem asistované diagnostikování je také velmi důležitou oblastí, kde využíváme rozpoznávání vzorů. Tento způsob diagnostikování je velmi běžný ve zdravotnictví a dá se aplikovat na celou řadu zdravotnických dat jako je např. rentgen, tomografie, ultrazvuk, elektrokardiogramy (ECG), elektroencefalogramy (EEG) [9]. Finální rozhodnutí a způsob léčby jsou samozřejmě vždy provedeny lékařem. Tyto expertní systémy slouží pouze k usnadnění jeho práce.

Rozpoznávání vzorů je nezbytné v oblasti detekce a rozpoznání mluvené řeči. Řeč je sice nejběžnější formou komunikace mezi lidmi, ovšem stroje mají velké potíže lidské řeči porozumět. Do této oblasti vývoje se investuje v současnosti velké množství peněz, protože existuje mnoho potenciálních způsobů využití této technologie. Podobná technologie by mohla výrazně usnadnit mnoho výrobních procesů, řízení strojů v nebezpečném prostředí a zlepšit život handicapovaným [9]. I když se o tuto oblast zajímají přední vědci a bylo učiněno mnoho pokroků, stroje stále nejsou schopné plně rozumět lidské řeči. V současné době stroje umí převést mluvené slovo do ASCII symbolů, ty následně dělit na věty a slova i identifikovat jednotlivé části věty (podmět, přísudek atd.) a dokonce je rozdělovat do skupin (např. pozitivní a negativní) [10].

Výše zmíněné příklady zastupují pouze některé oblasti využití, rozpoznávání vzorů se používá pro řešení mnoha dalších problémů. V závislosti na naší znalosti problému dostupných dat je nutné zvolit správnou formu učení.

2.3.2 Učení s učitelem a bez učitele

Máme-li trénovací data již ohodnocená a použitý algoritmus dokáže tuto předem známou znalost využít při procesu učení, potom se jedná o učení s učitelem. Při využití učení s učitelem dosahujeme obvykle lepší nebo stejné přesnosti kategorizace než při využití učení bez učitele i při trénování na menším objemu dat, avšak ne vždy máme k dispozici trénovací data s označenými výslednými třídami [11].

V situaci, kdy potřebujeme kategorizovat data, ke kterým nemáme vhodná trénovací data s označenými cílovými třídami, musíme zvolit jiný postup, kterému říkáme učení bez učitele. V případě, kdy máme k dispozici pouze množinu vstupních vektorů bez ohodnocení, je naším cílem hledat podobnosti mezi nimi a na základě nalezených podobností je třídit do skupin s požadovaným stupněm podobnosti. Tomuto způsobu kategorizace se říká shlukování (clustering).

2.4 Shlukování

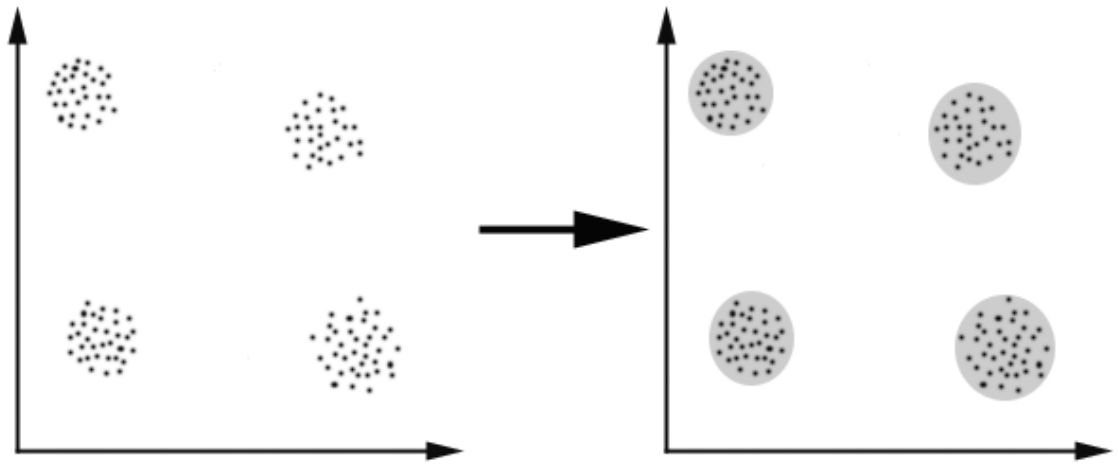
Shlukování je tedy proces, při kterém se snažíme vzory spojit do skupin na základě podobnosti jejich vlastností. Takto vzniklé skupině vzorů říkáme shluk. Protože se při procesu učení nepracuje s předem známým ohodnocením výstupní skupiny, nemusí být na první pohled jasné, jaké vlastnosti jsou typické pro určitý shluk. Interpretaci jednotlivých shluků bývá obvykle zajištěna expertem [12].

2.4.1 Obecný postup shlukování

1. Získání dat
2. Normalizace dat
3. Aplikace shlukovací metody
4. Analýza vytvořených shluků
5. Stanovení přesnosti shlukování [13]

Prvním krokem analýzy shluků je získání dat. Protože hodnoty tvořící vstupní vektor velmi často nepocházejí ze stejných intervalů, je v našem zájmu tyto hodnoty převést na společný interval. Tento proces nazýváme normalizace dat. Po provedení normalizace již hodnoty vstupního vektoru nebudou mít žádnou vazbu na jednotky reálného světa, ovšem výsledek shlukování bude přesnější. Nejčastější formou normalizace je normalizace do intervalu ⟨0, 1⟩. Hodnoty v tomto intervalu jsou vhodné pro rozpoznávání vzorů pomocí většiny dostupných nástrojů. Pro některé nástroje může být vhodná i normalizace na jinak ohraničené intervaly, jiné nástroje mohou být schopné pracovat i na nenormalizovaných datech.

Dalším krokem je aplikace vybrané shlukovací metody. Většina shlukovacích algoritmů funguje na principu porovnávání vzdálenosti v *r*-rozměrném prostoru mezi jednotlivými kategorizovanými vzory (viz Obr. 1). *R*-tá souřadnice vzoru odpovídá *r*-tému prvku vstupního vektoru.



Obr. 1 Shlukování [14]

Po provedení shlukovací metody je nutné otestovat, zda vytvořené shluky odpovídají našim požadavkům a s jakou přesností je nyní algoritmus schopný kategorizovat nové vstupy. To můžeme otestovat například pomocí metody zvané krosvalidace. Při použití této metody jsou data rozdělena na deset stejně velkých částí.

Iterativně vždy vybereme jednu desetinu pro účely testování a zbytek dat se využije jako trénovací data. Iterativně provádíme 10 testů a výsledná úspěšnost je vypočtena jako aritmetický průměr úspěšností dosažených při každé iteraci.

2.4.2 Aplikace shlukování

Shlukování se v praxi využívá pro řešení mnoha různých problémů. V následujícím textu se zaměříme na směry, ve kterých se lze se shlukováním běžně setkat a bude vysvětleno, jakým způsobem je možné shlukování v těchto oblastech použít.

Redukce dat

Velmi často se setkáváme s daty, jejichž objem je obrovský, a zpracování takových dat se může ukázat velmi problematické. Analýzy shluků se může využít pro setřídění dat do určitého počtu skupin, přičemž data uvnitř jedné skupiny mají společné vlastnosti a poté zpracováváme každou skupinu jako samostatnou entitu.

Například při přenosu signálu můžeme využít shlukování tak, že definujeme zástupce pro každý datový shluk. Následně místo přenosu dat samotných, pouze vysíláme předem definovaný kód pro přenášený shluk [15]. Tímto způsobem můžeme docílit výrazné komprese dat.

Generování hypotéz

V tomto případě aplikujeme shlukování na data za účelem získat hypotézu, která vystihuje podstatu dat samotných. Shlukování využíváme jako nástroj pro stanovení hypotéz, ovšem pravdivost těchto hypotéz musí být následně potvrzena využitím jiných dat [16].

Testování hypotéz

Analýzu shluků používáme také pro ověření správnosti určitých hypotéz. Vezměme v úvahu následující hypotézu: Velké společnosti investují v zahraničí. Jedním způsobem, jak ověřit tuto hypotézu, je využití analýzy shluků na dostatečně velkou a reprezentativní množinu společností. Předpokládejme, že každá společnost je reprezentována její velikostí, aktivitami v zahraničí a schopností úspěšně realizovat svoje projekty. Pokud se po aplikování analýzy shluků vytvoří shluk reprezentující velké společnosti, které investují v zahraničí bez ohledu na úspěšnost jejich projektů, potom je tato hypotéza podporována [15].

Předpovědi založené na skupinách

Na základě analýzy shluků je také možné získávat předpovědi. V tomto případě aplikujeme analýzu na dostupnou množinu dat, díky čemuž získáme jeden či více shluků. Tyto shluky jsou charakterizovány vlastnostmi objektů, na základě kterých byly zformovány. Pokud následně získáme další neznámý objekt, je možné určit, do jakého shluku pravděpodobně náleží, a přibližně tak získat jeho vlastnosti na základě vlastností daného shluku [15].

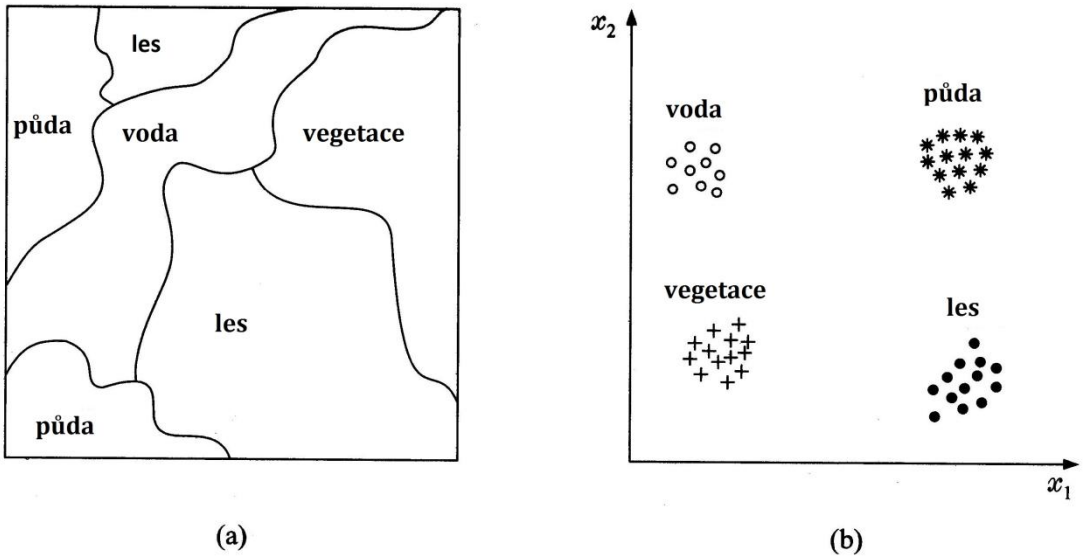
Představme si příklad, kdy je analýza shluků využita na množinu dat, která obsahuje informace o pacientech, kteří absolvovali stejnou nemoc. Data obsahují především informace o reakci pacientů na určité léky a účinnosti léčby [9]. Potom pro nového pacienta můžeme stanovit, do jakého shluku patří, a na základě toho určit způsob léčby, který je pro něj nejvhodnější.

Další konkrétní příklady využití

Analýzu shluků používáme také pro vyhodnocování dat pořízených pomocí multispektrálního dálkového pozorování země. Jedná se o multispektrální snímky pořízené pomocí speciálních senzorů umístěných na satelitu, letadle nebo vesmírné stanici. Tyto velmi citlivé senzory jsou schopné zachytit veškerou elektromagnetickou energii, která je zemským povrchem pasivně odrážena nebo aktivně generována [17]. Senzory jsou citlivé na různá vlnová pásma elektromagnetického záření. Různé typy zemského povrchu odrážejí elektromagnetickou energii v jednotlivých vlnových pásmech s rozdílnou intenzitou. Například elektromagnetickou energii ve viditelném pásmu odrážejí především minerály, vlhká půda, vodní usazeniny a vegetace. V infračerveném pásmu lze sledovat tepelné vlastnosti povrchu. Jinými slovy: každé pásmo elektromagnetického záření popisuje různé vlastnosti stejného místa na Zemi [9]. Tímto způsobem je možné vytvořit obrazy země, které odpovídají odrazu energie v různých pásmech. Následným úkolem je identifikovat jaké vlastnosti mají různé druhy zemského povrchu v jednotlivých pásmech. Zemský povrch můžeme například dělit do následujících shluků: zástavba, les (listnatý, jehličnatý), zemědělské oblasti, voda a podobně (viz Obr. 2). Každým vstupním vzorem analýzy potom bude jednotlivý pixel obrazu a prvky vstupního vektoru budou tvořit intenzity naměřené v různých vlnových pásmech pro daný pixel. Očekáváme, že stejné typy zemského povrchu budou zařazeny do stejného shluku.

Shlukování se také hojně využívá v sociálních vědách pro zpracování výsledků průzkumů, vytváření statistik a pro usnadnění procesu rozhodování. Chceme-li například zjistit spojitost mezi negramotností, úmrtností dětí a hrubý domácím produktem (HDP) určitého státu. V tomto případě je každá země reprezentována trojrozměrným vektorem, který obsahuje údaje právě o HDP, úmrtnosti dětí a gramotnosti v dané zemi. Po provedení analýzy shluků pravděpodobně nalezneme takový shluk, který bude sdružovat země s malou úrovní HDP, vysokou úmrtností dětí a vysokou úrovní negramotnosti [9].

Dalším možným způsobem využití shlukování je kategorizace textových dat. Přestavme si, že máme množinu dat, která obsahuje textová hodnocení spokojenosti s určitou službou, a naším cílem je rozdělit tato hodnocení na kladná, záporná a popřípadě neutrální [18]. Nejdříve je nutné se zamyslet, co je typické pro pozitivní či negativní komentář. Může jít o počet výskytů určitých slov nebo slovních spojení. Po nalezení takovýchto typických frází zjistíme četnost jejich výskytu v jednotlivých hodnoceních a na základě této četnosti vytvoříme vektor pro každé analyzované hodnocení.



Obr. 2 Shlukování typů zemského povrchu: a) mapová vrstva; b) výsledné shluky [9]

2.4.3 Kategorie shlukovacích algoritmů

Algoritmů pro shlukování existuje velké množství a každý z nich může poskytovat jiné výsledky na stejných datech, proto je vhodné je rozdělit do skupin na základě principu, na kterém fungují.

Sekvenční algoritmy

Sekvenční algoritmy mohou poskytovat více výsledků shlukování. Jedná se o přímočaré a rychlé metody. U většiny z nich prezentujeme vstupní vektor jednou nebo vícekrát, ovšem obvykle ne víc než šestkrát. Finální výsledek je závislý na pořadí vstupních dat [9]. Pro tyto algoritmy je typické generování shluků tvaru hyperkvádru nebo hyperelipsoidu v závislosti na používané vzdálenostní metrice.

Hierarchické shlukovací algoritmy

Hierarchické shlukovací algoritmy můžeme dále rozdělit na spojovací a rozdělovací.

- *Spojovací algoritmy* fungují ve více krocích. Počet shluků vzniklých v prvním kroku je nejvyšší. V každém dalším kroku vznikají shluky spojením dvou shluků z předchozího kroku. Tyto algoritmy jsou vhodné pro detekci shluků s protáhlým tvarem.
- *Rozdělovací algoritmy* také fungují ve více krocích. Počet shluků v prvním kroku je nejmenší. V dalších krocích vznikají shluky rozdělením jednoho shluku z předchozího kroku na dva [9].

Shlukovací algoritmy založené na principu optimalizace cenové funkce

Tyto algoritmy posuzují smysluplnost shluků podle výsledku cenové funkce. Většina algoritmů tohoto typu je založena na diferenciálním počtu. Algoritmus shluky iterativně upravuje do té doby, než dosáhne optimální hodnoty cenové funkce, podle které je stanovena kvalita shlukování. Tuto kategorii je možné rozdělit do několika podkategorií.

- První podkategorii tvoří Hard or Crisp (pevné nebo křehké) shlukovací algoritmy. V tomto typu algoritmů patří vektor pouze do jednoho shluku na základě definovaného kritéria optimality.
- *Pravděpodobnostní shlukovací algoritmy* zařazují vektory do shluků na základě Bayesova teorému.
- *Fuzzy shlukovací algoritmy* nedefinují, do jakého shluku vektor patří, ale určují stupeň, se kterým vektor náleží do každého ze shluků.
- *Algoritmy pro detekci hranice* pracují odlišným způsobem než ostatní. Místo určení shluků samotných se iterativně stanovují a upravují hranice mezi shluky [9].

Ostatní

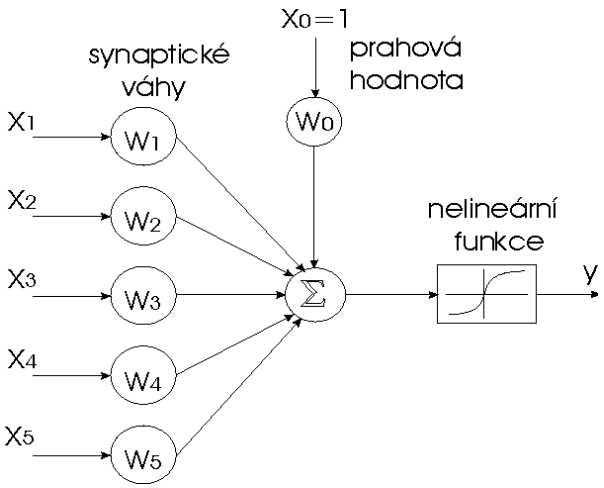
Tato poslední kategorie obsahuje některé speciální druhy shlukovacích technik, které se nedají zařadit do výše zmíněných kategorií.

- *Branch and bound clustering* (shlukování pomocí větvení a ohraničení) je skupina shlukovacích algoritmů, která je schopná generovat velmi přesné výsledky pro předem určený počet shluků a kritérium optimality, aniž by museli iterativně generovat více různých výsledných shlukování. Jejich velkou nevýhodou je vysoká výpočetní složitost.
- *Genetic clustering* (genetické shlukování) algoritmy vychází z Darwinovy teorie o evoluci druhů. Tyto algoritmy se inicializují vytvořením první generace, která je reprezentována možným výsledkem shlukování. Každá další generace by měla reprezentovat lepší výsledek shlukování na základě stanoveného kritéria.
- *Stochastic relaxation* (stochastická relaxace) metody nám za určitých podmínek garantují konvergenci pravděpodobností ke globálně optimálnímu shlukování. Tyto metody mají velkou výpočetní složitost.
- *Valley-seeking clustering* (shlukování pomocí prohledávání údolí) algoritmy reprezentují vstupní vektory jako objekty *r*-rozměrné náhodné proměnné *x*. Vycházejí z běžně uznávaného předpokladu, že oblasti *x*, ve kterých se nachází velké množství vektorů, odpovídají oblastem funkce hustoty pravděpodobnosti (probability density function – pdf). Odhadnutí pdf nám tedy může pomoci zvýraznit oblasti, ve kterých se tvoří shluky.

- *Competitive learning* (soutěživé učení) algoritmy patří mezi iterativní metody, které nepoužívají žádnou cenovou funkci. Obvykle vytvoří více možných výsledných shlukování a následně vyberou jedno takové, které dává největší smysl z hlediska metriky.
- *Density-based clustering* (shlukování založené na hustotě) algoritmy reprezentují shluky jako oblasti *n*-dimenzionálního prostoru s velkou hustotou dat. V tomto ohledu jsou podobné *Valley-seeking* algoritmům, ale k dosažení tohoto cíle využívají jiný postup. Každý algoritmus z této skupiny se snaží různými způsoby kvantifikovat hustotu. Algoritmy z této skupiny je možné využít pro zpracování velmi obsáhlých dat.
- *Subspace clustering* (shlukování podprostoru) algoritmy jsou vhodné pro zpracování mnohorozměrných dat. V některých případech může atributový prostor dosahovat mnoha tisíc rozměrů. Fungují na principu rozdělování mnohorozměrného prostoru na podprostory, ve kterých je výpočet snadnější [9].

2.5 Umělé neuronové sítě

Umělé neuronové sítě patří mezi algoritmy strojového učení. Neuronová síť je tvořena množinou výpočetních jednotek zvaných neurony (uzly), které jsou propojeny jednosměrnými cestami.



Obr. 3 Model umělého neuronu [19]

Pro každou neuronovou síť by mělo obecně platit:

- Obsahuje množinu adaptivních numerických parametrů (váhy).
- Je schopná aproximovat nelineární funkce.

Váhy v podstatě reprezentují sílu spojení mezi neurony a jsou využívány v procesu učení i vyhodnocení. Neurony v jedné vrstvě zpracovávají informaci paralelně a podílejí se tedy na řešení problému společně (viz Obr. 3). Vyobrazený neuron má celkem pět vstupů.

- x_j je j-tý vstup neuronu
- w_j je synaptická váha na j-tém vstupu neuronu
- x_0 je prahová hodnota (bias)
- y je výstupem neuronu

Výstup neuronu je závislý na vstupech neuronu, hodnotách vah, prahové hodnotě i aktivační funkci. Nejdříve je nutné určit sumu S (1), která je tvořena součinem všech vstupů (x) a váhy odpovídající danému vstupu. Pro výpočet výstupu neuronu y je nutné na S aplikovat aktivační funkci $f(S)$(2) [20].

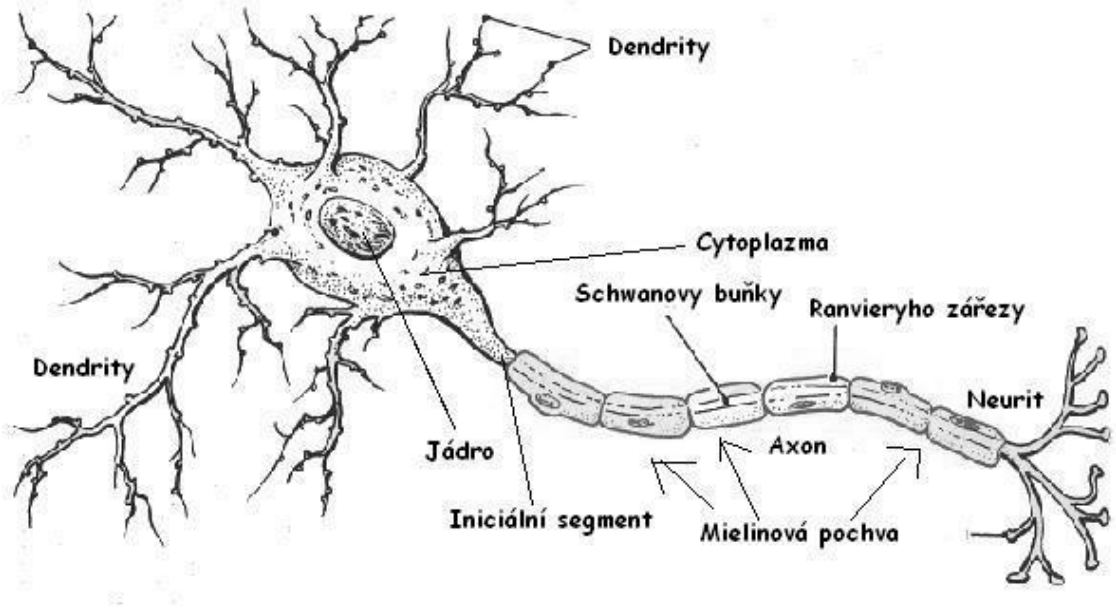
$$S = \sum_{j=0}^{5} w_j \, x_j \tag{1}$$

$$y = f(S) \tag{2}$$

2.5.1 Analogie mezi umělou a biologickou neuronovou sítí

„Umělé neuronové sítě jsou inspirovány biologickými neuronovými sítěmi. Tato vlastnost určitým způsobem předurčuje, že uměle vytvořené neuronové sítě by měly být schopny, z hlediska základních principů, se chovat stejně nebo alespoň podobně jako jejich biologické vzory. Je zřejmé, že vytvoření umělého lidského mozku se všemi jeho schopnostmi je věc jen velmi těžce řešitelná, ať už z hlediska kvantity jeho neuronů či jejich způsobu propojení, chování jednotlivých typů neuronů apod. Nicméně skýtá se tu šance simulovat alespoň některé funkce lidského myšlení a tyto pak implementovat [20].“

Jak již bylo řečeno, umělé neuronové sítě vycházejí z principů, na kterých fungují biologické neuronové sítě (viz Obr. 4). Pro popis biologických neuronových sítí využíváme následující pojmy: jádro (soma), axon, dendrity, synapse.



Obr. 4 Biologická neuronová síť [21]

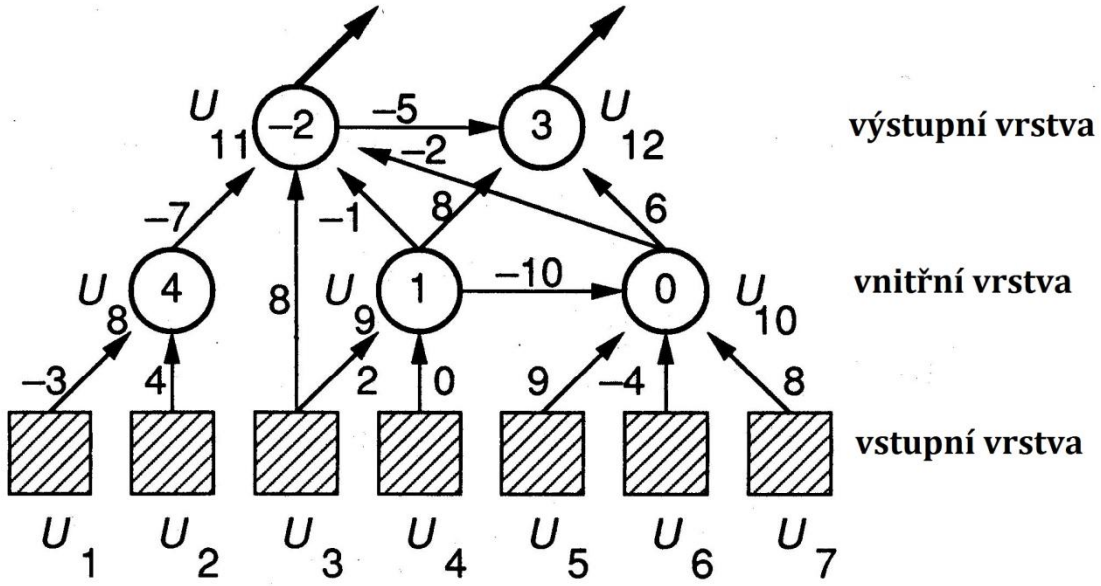
- Jádro neuronu (soma) u biologických neuronových sítí odpovídá umělému neuronu (uzlu).
- Pojem axon u biologických neuronových sítí odpovídá výstupu umělého neuronu.
- Pojem dendrit u biologických neuronových sítí odpovídá vstupu umělého neuronu.
- Pojem synapse u biologických neuronových sítí odpovídá váze vstupu umělého neuronu [22].

2.5.2 Architektura neuronové sítě

Účelem této kapitoly je podrobněji popsat principy, na nichž neuronové sítě fungují a obecné vlastnosti neuronových sítí, vysvětlit funkci jednotlivých neuronů a jejich chování v síti.

Vlastnosti sítě

Neuronová síť je tvořena množinou neuronů spojených jednosměrnými toky. Každý z těchto neuronů pracuje samostatně, ovšem vstup jednoho neuronu může být tvořený výstupem jiného. Každé spojení mezi dvěma neurony je ohodnoceno číselnou hodnotou (váhou), která reprezentuje vliv výstupního neuronu na neuron vstupní. Kladná hodnota váhy značí posílení vazby a naopak záporná hodnota váhy značí blokaci dané vazby. Hodnoty těchto vah udávají vlastnosti neuronové sítě a do jisté míry plní roli jako běžný program. Každý počítačový program může být simulován pomocí neuronové sítě [23].



Obr. 5 Architektura neuronové sítě [23]

Vstupní neurony jsou definovány externě a nemají žádná vstupní spojení [23]. Jejich výstupní spojení se nepočítá, pouze reprezentuje vstup dat do neuronové sítě. Více vstupních neuronů tvoří vstupní vrstvu neuronové sítě (viz Obr. 5).

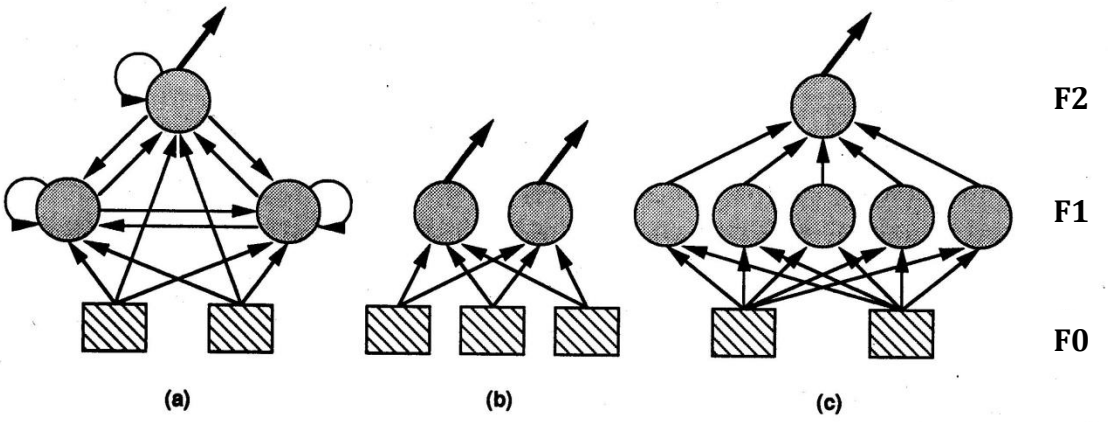
Další speciální skupinou neuronů jsou neurony výstupní (viz Obr. 5). Jejich funkcí je poskytovat výsledné hodnoty a tedy jejich výstupy považujeme za výstupy celé neuronové sítě [23]. Více výstupních neuronů tvoří výstupní vrstvu neuronové sítě.

Takové neurony, které nejsou vstupní ani výstupní, nazýváme neurony vnitřní vrstvy (viz Obr. 5) [23]. Tyto neurony a vazby mezi nimi realizují funkcionalitu neuronové sítě a jsou nezbytné k tomu, aby neuronová síť mohla počítat výsledky složitých funkcí.

Může nastat situace, kdy můžeme definovat vstupy neuronové sítě, a víme, jaké výstupy neuronová síť poskytuje, ale nemáme žádné nebo omezené informace o neuronech a vazbách vnitřní vrstvy [23]. V takové situaci nevíme, jakým způsobem neuronová síť data zpracovává, protože je pro nás vnitřní vrstva skrytá. Sítě, které obsahují podobné skryté vrstvy se skrytými neurony a vazbami, považujeme za černou skříňku (black box).

Podle toho, zda neuronová síť obsahuje zpětnou vazbu a tedy i cykly, rozdělujeme neuronové sítě na dopředné a rekurentní. První skupina se mnohdy používá pro aplikace učení s učitelem [23]. Vnitřní vrstvu neuronové sítě je velmi často vhodné rozdělit do několika podvrstev. Každá vrstva potom reprezentuje oddělenou část procesu a jednotlivé vrstvy mezi sebou mohou sdílet i zpětnou vazbu. U striktně vrstvené neuronové sítě je nutné, aby neurony jedné vrstvy měly vazbu pouze na neurony jiné vrstvy, tedy neurony jedné vrstvy nesmějí sdílet vazbu. Vstupní vrstvu neuronové sítě do počtu vrstev obvykle nezahrnujeme.

Podle počtu vrstev můžeme neuronové sítě rozdělit na jednovrstvé a vícevrstvé. Jednovrstvé neuronové sítě mají striktně vstupní a výstupní vrstvu. Vícevrstvé neuronové sítě mohou mít analogicky vrstev více. Speciálním případem vícevrstvé sítě je plochá síť. Ta obsahuje vrstvu vstupní (F0), jednu skrytou vrstvu (F1) a vrstvu výstupní (F2) (viz Obr. 6) [23].



Obr. 6 Úplná (a), jednovrstvá (b) a plochá (c) neuronová síť [23]

Vlastnosti neuronu

Každý z neuronů uvnitř neuronové sítě je schopen přijímat vstup a generovat výstup. Výstupem může být numerická hodnota nebo indikace aktivace. Například na Obr. 5 můžeme vidět výstup neuronu u_{11}, který využíváme jako jeden z finálních výstupů neuronové sítě a zároveň pro výpočet aktivace neuronu u_{12}.

Vstupy neuronu mohou nabývat diskrétních nebo spojitých hodnot. U diskrétních vstupů jsou to určité množiny hodnot, obvykle se jedná o množiny {0, 1} a {–1, 0, 1}. Spojité vstupy jsou definovány intervalem, obvykle se jedná o interval ⟨0, 1⟩ a ⟨–1, 1⟩.

Dle konvencí každý neuron sítě musí být připojený ke speciálnímu neuronu u_0, jehož výstupem je vždy hodnota 1. Vazba mezi u_0 a jiným neuronem se nazývá bias. Pro usnadnění se neuron u_0 nahrazuje zavedením hodnoty bias uvnitř každého neuronu [23].

Aktivace neuronu je spočtena na základě aktivací všech neuronů do něj přímo připojených. Následně je spočtena vážená suma aktivací vážených silou příslušného spojení. Na Obr. 5 je vidět, že při stanovování aktivace neuronu u_{11} vycházíme z aktivací neuronů u_3, u_8, u_9, u_{10} a vah $w_{11,0}$ (bias), $w_{11,3}$, $w_{11,8}$, $w_{11,10}$ [23]. Nejdříve proběhne výpočet S (1) a následně je aplikována aktivační funkce (2).

Aktivační funkce

Aktivační (přenosová) funkce je nutná pro stanovení výstupu neuronu v závislosti na jeho vstupech, vahách a prahové hodnotě. Nejdříve je na základě těchto hodnot stanovena hodnota S (1), na kterou následně aplikujeme aktivační funkci (2). Protože je výstup neuronu silně závislí právě na aktivační funkci, je nutné tuto funkci zvolit vhodně, aby se neuron choval tak, jak od něj očekáváme. Strmost aktivační funkce určuje rychlost reakce neuronu na změnu nezávislých vstupů. Aktivační funkce se dají rozdělit do tří základních kategorií: prahové, lineární a nelineární. Je nutné mít na paměti, že nelineární funkci nelze aproximovat pouze kombinací lineárních funkcí.

Prahové aktivační funkce jsou schopné generovat pouze určitý počet výstupů, obvykle používáme skokovou funkci, která je schopna generovat obvykle dva nebo tři výstupy (0 a 1 nebo 1, 0 a –1). K aktivaci neuronu dochází při přijetí prahové hodnoty definované funkcí. Tento druh funkce využíváme v situacích, kdy pracujeme se dvěma nebo více dobře oddělenými stavy.

Lineární a nelineární aktivační funkce se liší od prahových především tím, že generují výstupy, které patří do spojitého intervalu. Obvykle se jedná o interval $\langle 0, 1\rangle$ a interval $\langle -1, 1\rangle$. Lineární funkce se obvykle používají pro neurony ve výstupní vrstvě neuronové sítě. Vzorec lineární přenosové funkce (3) je následující:

$$f(S) = S \tag{3}$$

Mezi nejpoužívanější nelineární funkce patří sigmoida, hyperbolická tangenta, radiální bázová funkce a saturační funkce. Sigmoida (4; e – Eulerovo číslo) je schopná generovat výstupy v intervalu $\langle 0, 1\rangle$ a běžně se používá v neuronových sítích pro řešení problému jako celku. Chceme-li raději rozdělit problém na více podproblémů, zvolíme jiný přístup.

$$f(S) = \frac{1}{1 + e^{-S}} \tag{4}$$

Hyperbolická tangenta (5) je speciální formou sigmoidy. I když se v goniometrii běžně používá, její využití v neuronových sítích s goniometrií příliš nesouvisí. Tato funkce má velmi podobné aktivační vlastnosti jako sigmoida, na rozdíl od ní je schopná generovat výstup v intervalu $\langle -1, 1\rangle$ a díky tomuto vyššímu numerickému rozsahu je často využívána místo sigmoidy.

$$f(S) = \frac{\sinh(S)}{\cosh(S)} = \frac{e^{S} - e^{-S}}{e^{S} + e^{-S}} \tag{5}$$

Radiální bázová funkce (RBF) je tvarem podobná Gaussově křivce. Využívá se v situaci, kdy je pro nás výhodnější problém rozdělit na podproblémy, následně jednotlivé instance problému hodnotit z hlediska každého podproblému zvlášť. Tento přístup je z lokálního hlediska přesnější, ale ztrácíme tím globální přehled

o daném problému. Základní verze vzorce pro radiální bázovou funkci (6) obsahuje volitelné parametry a, μ a σ, které ovlivňují tvar křivky [24].

$$f(S) = ae^{-\frac{(S-\mu)^2}{2\sigma^2}} \tag{6}$$

Dynamické vlastnosti

Pro správné fungování neuronové sítě musí být specifikováno, kdy má probíhat aktivace určitých neuronů. V dopředných neuronových sítích bez zpětné vazby obvykle probíhá aktivace neuronů v pevně daném pořadí tak, aby proběhla aktivace každého neuronu předtím, než přejdeme k následujícímu. V tomto případě síť nabyde stabilního stavu po jednom průchodu neuronovou sítí.

U rekurentních modelů neuronových sítí volíme z několika možností. Jednou z možností je projít neurony v předem definovaném pořadí tak, jako tomu bylo u dopředného modelu. Bohužel využitím tohoto způsobu není možné zaručit, že síť nabyde stabilního stavu. Další možností je opakovat průchod sítí v pevně daném pořadí do té doby, dokud síť nenabyde stabilního stavu. Neuronová síť buď nabyde stabilního stavu, nebo se bude proces cyklicky opakovat. Třetí možností je vypočítat aktivaci všech neuronů zároveň a poté provést změnu a následně změnit výstupy všech neuronů zároveň [23]. Tento proces se cyklicky opakuje, stejně jako tomu bylo v předchozím případě. Také můžeme procházet jednotlivé uzly v náhodném pořadí, ovšem v takovém případě nelze zajistit jakoukoliv formu cyklického průběhu a samotný proces se velmi těžko limituje.

Učení

I u neuronových sítí rozlišujeme učení s učitelem a učení bez učitele. Pro učení s učitelem je nutné, aby naše trénovací data obsahovala i správné hodnoty výstupů neuronové sítě, pro učení bez učitele takovéto požadavky na data nemáme.

Problémy, které řešíme při učení s učitelem, se dále rozdělují na lehké a těžké [23]. Řešíme-li lehký problém, máme ke každému záznamu nejen informaci o správných výstupech sítě, ale i informace o správných aktivacích neuronů vnitřní vrstvy. Pro řešení těžkých úkolů máme sice informace o správných výstupech neuronové sítě, ale nevíme, jaké jsou správné aktivace neuronů uvnitř sítě.

Těžké úkoly se dají dále rozdělit do dvou kategorií: problémy volné sítě a problémy pevné sítě. U volných sítí se v průběhu procesu mohou přidávat i další neurony a měnit topologii sítě, naopak u pevných sítí je počet neuronů a topologie sítě pevně daná. V průběhu vlastního procesu učení se mění pouze váhy u spojení [25].

Jednotlivé algoritmy také dělíme podle počtu průchodů, které neuronová síť potřebuje ke správnému stanovení vah. První skupinu tvoří takové algoritmy, kterým ke stanovení vah stačí jen jeden průchod neuronovou sítí, tyto algoritmy patří obecně mezi nejrychlejší.

Naopak iterativní algoritmy musejí obvykle ke správnému stanovení vah průchod stejného záznamu sítí několikrát opakovat. Iterativní algoritmy nám obecně poskytují lepší výsledky, ale jsou výrazně pomalejší [23]. Rychlost učení je velmi

důležitá z hlediska zpracování středních a velkých objemů dat. Pro problémy zahrnující velké objemy dat je použití časově náročných algoritmů nemožné.

V této kapitole byly vysvětleny nejdůležitější vlastnosti neuronů a z nich vytvořených neuronových sítí, můžeme se blíže podívat na jednotlivé algoritmy, které v této oblasti využíváme.

2.5.3 Nejdůležitější modely neuronových sítí

Každý model neuronové sítě vykazuje jiné vlastnosti, a proto je vhodné si shrnout alespoň základní z nich. Tato kapitola pojednává o základních modelech neuronových sítí o jejich vlastnostech a využití.

Neuron první generace byl navržen americkými kybernetiky Warrenem McCullohem a Waltrem Pittsem. Inovace spočívala v zavedení inhibičních vazeb neuronu. První typ vazby je reprezentován synaptickou vahou rovnou hodnotě +1 a druhý pak hodnotou 0. Každý neuron má definovanou svou vnitřní hodnotu prahu, která musí být překonána vnitřním potenciálem neuronu, aby došlo k jeho excitaci. Tento jednoduchý způsob definice neuronu umožňuje modelovat různé procesy, jako například podmíněný reflex. Model se skládá ze tří neuronů s definovanou hodnotou prahů, dvou vstupů (nepodmíněný a podmíněný), jednoho výstupu (podmíněný reflex) a pouze z excitačních vazeb [20]. Vývoj prvního neuronu byl počátkem rozvoje neuronových sítí.

Jeden z nejdůležitějších modelů dodnes používaných je tzv. perceptron, jehož potenciál je definovaný jako vážený součet vstupujících signálů. Pokud tento vnitřní potenciál neuronu překoná jeho prahovou hodnotu, dojde k excitaci neuronu na hodnotu 1. V opačném případě je neuron inhibitován, což je reprezentováno hodnotou 0. V podstatě se jedná o rozdělení vstupního prostoru na dva poloprostory. Jinými slovy, jsme schopni prostřednictvím jednoho perceptronu rozlišit dvě třídy vstupů. Jedné z nich odpovídá excitace rovna hodnotě 1 a druhé inhibice neuronu daná hodnotou 0. Otázkou nyní je, jak stanovit hodnoty vah neuronu, aby byl schopen správně rozpoznávat (přiřazovat do tříd) předložené vstupy. K tomu je potřebné náš perceptron adaptovat na základě trénovací množiny prostřednictvím nějakého algoritmu. Jeden z nejznámějších principů je adaptace (učení) neuronu podle Hebbova pravidla [20].

Dalším hojně využívaným modelem neuronových sítí je backpropagation (zpětné šíření) využívající vícevrstvé neuronové sítě, tedy takové sítě, které mají vstupní, výstupní a alespoň jednu vnitřní vrstvu. Vždy mezi dvěma sousedními vrstvami se pak nachází tzv. úplné propojení neuronů, tedy každý neuron nižší vrstvy je spojen se všemi neurony vrstvy vyšší. Pomocí dopředného šíření signálu je nejprve získána odezva neuronové sítě na vstupní podnět daný excitací neuronů vstupní vrstvy. Takovým způsobem vlastně probíhá šíření signálů i v biologickém systému, kde vstupní vrstva může být tvořena např. zrakovými buňkami a ve výstupní vrstvě mozku jsou pak identifikovány jednotlivé objekty sledování. Otázkou zůstává to nejdůležitější, jakým způsobem jsou stanoveny ony synaptické váhy vedoucí ke korektní odezvě na vstupní signál. Proces stanovení synaptických vah je opět spjat s pojmem učení a adaptace neuronové sítě. Další otázkou je i schopnost

generalizace nad naučeným materiálem, jinými slovy jak je neuronová síť schopna na základě naučeného usuzovat na jevy, které nebyly součástí učení, které však lze nějakým způsobem odvodit [20]. K procesu učení potřebujeme jednak tzv. trénovací množinu a dále také vhodnou metodu, která umožňuje odpovídající adaptaci vah odvodit. Pro tyto účely využíváme právě metodu zpětného šíření. Na rozdíl od dopředného šíření signálu neuronovou sítí, tato metoda adaptace spočívá v opačném šíření informace směrem od vrstev vyšších k vrstvám nižším.

Všechny doposud zmíněné modely neuronových sítí nebraly v úvahu kontext, ve kterém data přicházejí. Výstup těchto sítí je tedy pro stejný vstup vždy stejný. Některé neuronové sítě dokáží vnímat i kontext, ve kterém data přicházejí. Protože u takových neuronových sítí výstup závisí nejen na vstupním vektoru, ale i na kontextu, ve kterém data přicházejí, výstup sítě se pro stejný vstupní vektor může lišit [20]. Chceme-li, aby naše neuronová síť byla schopná zohlednit pro své hodnocení i kontext, ve kterém data přicházejí, využijeme rekurentní neuronovou síť. Rekurentní neuronová síť ke své adaptaci využívá metodu zpětného šíření. Oproti předcházejícím modelům se v tomto případě signál nešíří pouze od vstupní vrstvy směrem k vrstvě výstupní, ale dochází i ke zpětnovazebnému přenosu informace od vrstev vyšších zpět do vrstev nižších [20]. Topologie rekurentní sítě se liší od klasické dalšími rekurentními neurony ve vstupní a vnitřní vrstvě. Ve střední vrstvě je jeden neuron odpovídající jednomu neuronu výstupní vrstvy a ve vstupní vrstvě jsou dva rekurentní neurony odpovídající dvou neuronům vnitřní vrstvy. Každý z těchto neuronů přijímá jediný vstupní signál, patřící jemu příslušejícímu regulérnímu neuronu z následující vrstvy.

Jeden z nejběžnějších modelů neuronových sítí využívaný pro shlukování se nazývá Kohenovy mapy a patří mezi vícevrstvé neuronové sítě. Propojení ve výstupní vrstvě je typické sebeexcitující vazbou a inhibičními vazbami vzhledem k ostatním neuronům. Tento způsob propojení vede k postupnému posilování toho neuronu, který byl na počátku excitován nejvíce. Výsledkem je pak situace, kdy je tento neuron vyexcitován na maximum, zatímco ostatní jsou úplně potlačeny (laterární, postranní inhibice). Tento postup je obdobný postupu excitace a inhibice neuronů ve skutečném mozku. Každý neuron pak reprezentuje nějaký objekt, či třídu objektů ze vstupního prostoru. Takový neuron budeme nazývat jako vítězný uzel (*bm* – best matching) [20]. Každému vektoru je pro správnost shlukování nutné určit odpovídající *bm*, který potom vektor reprezentuje. Takový neuron je poté schopen určit i další vektory, které do dané skupiny patří. Hlavní ideou těchto neuronových sítí je nalézt prostorovou reprezentaci složitých datových struktur. Tedy aby třídy s podobnými vektory byly reprezentovány neurony blízkými si v dané topologii. Tato vlastnost je typická i pro skutečný mozek, kde například jeden konec sluchové části mozkové kůry reaguje na nízké frekvence, zatímco opačný konec reaguje na frekvence vysoké. Tímto způsobem je možné mnohodimenzionální data zobrazit v jednodušším prostoru.

Z Kohenových map vychází další hojně využívaný model, kterému říkáme Counterpropagation (opačné šíření). Dříve, než přistoupíme k popisu tohoto modelu neuronové sítě, pokusme se nejprve o definování tzv. Grossbergovy hvězdy. Tu

si můžeme představit jako vrstvu neuronů obklopující *bm* ve středu. Obdobně jako v případě Kohonenova algoritmu se koeficient učení mění v čase, obvykle je jeho počáteční hodnota 0,2 a postupně při adaptaci dosáhne nulové hodnoty [20]. Takto vytvořené shluky vykazují vysokou míru podobnosti. V Counterpropagation neuronových sítích využíváme jednu skrytou vrstvu tvořenou Kohenovou mapu a druhou pomocí Grossbergovi hvězdy. Nespornou výhodou uvedeného modelu je rychlost jeho adaptace, nevýhodou je menší přesnost odezvy než poskytuje metoda zpětného šíření [20].

Na závěr této kapitoly se seznámíme s Hopfieldovými neuronovými sítěmi, jejichž autorem je John Hopfield, který se zabýval studiem neuronů podobných již dříve uvedeným perceptronům, ale přece jenom s některými podstatnými odlišnostmi. Podstatou problému bylo použití energetické funkce svázané s neuronovou sítí tak, jak je to běžné i u jiných fyzikálních systémů. Hopfieldova síť se skládá z množiny neuronů navzájem úplně v obou směrech propojených. Váhy sítě jsou symetrické ve smyslu rovnosti. Každý neuron má stejně jako perceptron svůj práh a skokovou aktivační funkci. Vstupem neuronu je opět vnitřní potenciál daný váženou sumou výstupů okolních neuronů. Stav neuronů tedy může být buďto standardní {0, 1} nebo bipolární {–1,+1} [20]. S bipolárními stavy neuronů se obvykle u Hopfieldových sítí setkáváme častěji. Hlavní rozdíl Hopfieldova modelu spočívá v tom, že vstup je aplikován na všechny neurony sítě ve formě hodnot –1 a +1, načež následuje cyklus postupných změn excitací neuronů, až do okamžiku dosažení stabilního stavu. Jinými slovy výstupy předchozího kroku se staly novými vstupy současného kroku. Inicializační stav reprezentuje různorodost excitací neuronů, které vzhledem k tomu, že jsou všechny propojeny, se začnou navzájem ovlivňovat. To může znamenat, že jeden neuron se snaží neurony excitovat na rozdíl od jiného, který se snaží o opačné. Výsledkem je nalezení kompromisu – síť relaxovala do stabilního stavu [20]. Hlavním problémem uvedeného algoritmu Hopfieldovy sítě je nebezpečí uvíznutí v některém z nežádoucích lokálních minim energie během relaxace sítě.

2.6 Teorie adaptivní resonance

Teorie adaptivní resonance (ART) byla formulována v roce 1976 matematiky Stephenem Grossbergem a Gailem Carpenterem a je základem řady matematických modelů, které využíváme pro dolování znalostí z dat.

V podstatě se jedná o kognitivní a nervovou teorii popisující způsob, jakým se lidský mozek učí kategorizovat, rozpoznávat a předpovídat události v měnícím se světě. ART podrobně popisuje procesy vědomého formování zkušeností a funguje na principu předpovědí funkčního spojení mezi procesy vědomí, učení, očekávání, odezvy a synchronizace. Tímto spojením je popsán způsob, jak lidský mozek reaguje na okolní prostředí [26]. Tento princip umožňuje adaptaci na základě vjemů okolního světa v reálném čase a nejen díky tomuto faktu se teorie adaptivní resonance využívá pro řadu algoritmů pro účely učení a stabilní kategorizace.

V teorii adaptivní resonance hrají kritickou roli takzvané resonantní stavy. ART předpokládá, že všechny vědomé stavy jsou resonantní [26]. Jednotlivé resonantní stavy jsou mezi sebou propojeny pomocí zpětné vazby, která umožnuje převedení naučených zkušeností do logické reprezentace reálného světa. Algoritmy založené na ART se často využívají pro kategorizaci objemných dat například v lékařských databázích. Tyto algoritmy řadíme mezi neuronové sítě.

U shlukovacích algoritmů probíraných v předchozích kapitolách je často striktně oddělená fáze učení a samotná fáze kategorizace a díky tomu mohou v určitých situacích nastat problémy. Reálný svět se neustále mění a stejně tak data, která z něho pocházejí. V praxi se proto můžeme často ocitnout v situaci, kdy chceme naši neuronovou síť přizpůsobit novým, dříve neznámým skutečnostem. Neuronové sítě, které striktně oddělují proces učení a kategorizace toto obvykle neumožňují a je nutné opakovat celý proces od začátku. Neuronové sítě založené na teorii adaptivní resonance zpracovávají data sekvenčně a při procesu učení nepotřebují všechna trénovací data uchovávat v paměti. Díky tomu je možné prokládat proces učení a proces samotné kategorizace dle potřeby bez rizika znehodnocení již naučených znalostí.

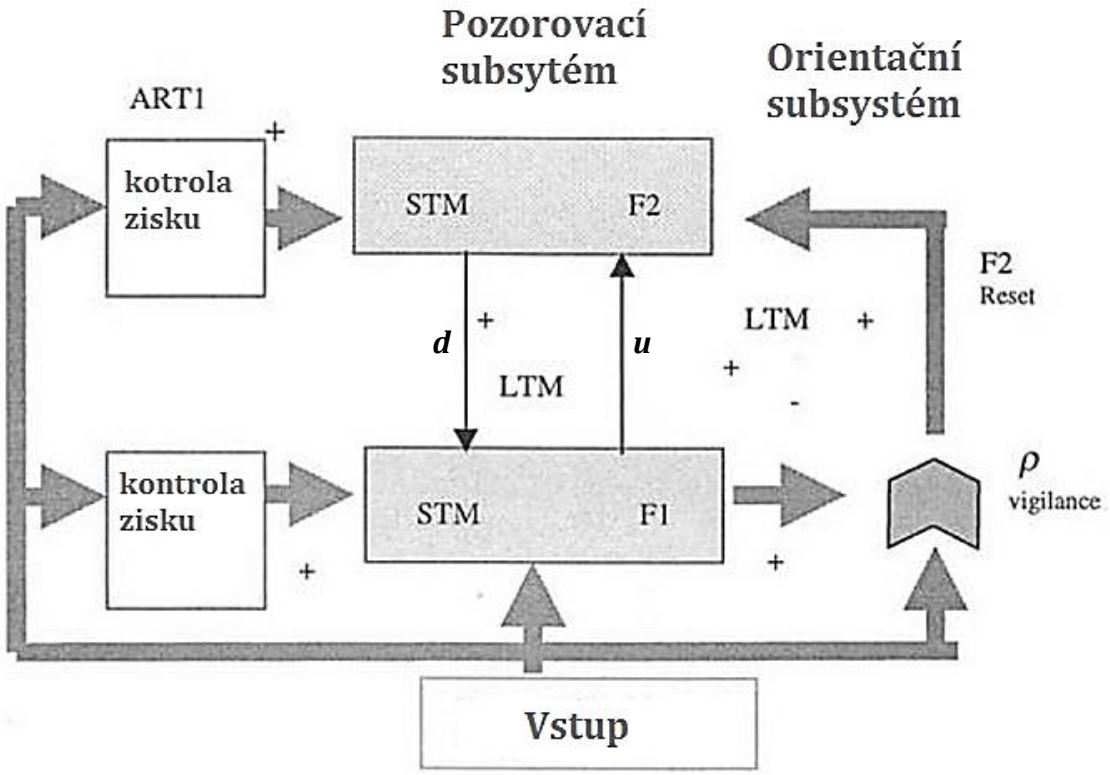
Zatímco pojem neuron je vhodný k vysvětlení základních principů neuronových sítí, v souvislosti s ART neuronovými sítěmi se v odborné literatuře obvykle využívá pojem uzel, který je užíván i v následujícím textu. Oba pojmy jsou vzájemně zaměnitelné.

2.6.1 Princip ART sítí

Neuronové sítě založené na teorii adaptivní resonance zpracovávají vstupní vzory po částech buď změnou vah, nebo pomocí dlouhodobé paměti (LTM – long term memory) adaptivních filtrů (viz Obr. 7). Tyto filtry jsou obsaženy uvnitř vazeb mezi vrstvami F1 a F2. Vrstva F1 slouží k reprezentaci vstupních vektorů ve formě prototypů kategorií. Počet uzlů této vrstvy odpovídá počtu kategorií, na které lze vstupní data rozdělit. Vrstva F2 slouží k reprezentaci jednotlivých shluků a počet uzlů této vrstvy výslednému počtu shluků odpovídá. Mezi těmito dvěma poli dochází ke kooperativním i kompetitivním interakcím realizovaných pomocí filtru (u), který je tvořený vazbami vedoucími z vrstvy F1 do vrstvy F2 [27]. Takovou kombinaci adaptace a kompetitivní interakce často nazýváme kompetitivní učení. S kompetitivním učením v oblasti rozpoznávání vzorů se setkáváme poměrně často, ovšem pro vlastní stabilizaci ART sítě hraje nejdůležitější roli druhý adaptivní filtr vedoucí z vrstvy F2 do F1 (d). Tento typ vazeb reprezentuje naučená očekávání ART sítě a umožňuje síti vykonávat následující procesy: priming (reakce na určitý stimul), porovnávání vzorů a paralelní vyhledávání.

Na Obr. 7 vidíme, že pole F1 a F2 včetně obou zmíněných adaptivních filtrů jsou součástí pozorovacího subsystému. Orientační subsystém slouží jako záloha v situaci, kdy se vstup do vrstvy F1 neshoduje s žádným z naučených očekávání ve vrstvě F2. Při aktivaci orientačního subsystému dojde k resetu reprezentace shluků ve vrstvě F2. Tento reset automaticky vyvolá paralelní prohledávání pozorovacího subsystému. Nalezené alternativy jsou dále testovány a je buď nalezena sho-

da, nebo je vytvořen nový shluk. Rychlost prohledávání je závislá na zvolené rychlosti učení. K významným změnám adaptivních filtrů dochází pouze, když je prohledávání dokončeno a pro vzor rezonující uvnitř systému byla nalezena shoda.



Obr. 7 Architekrura ART1 [27]

Protože učení ART sítě probíhá pouze ve stavech rezonance, je možné manipulovat s poměrem stability a plasticity sítě [27]. Plasticita sítě určuje její potenciál provádět rychlé změny uvnitř dlouhodobé paměti a tedy i schopnost sítě reagovat na potenciální změny v budoucnu.

Adaptivní vyhledávání využívané v ART modelech umožnuje učení vzorů bez rizika uváznutí v nežádoucím stavu nebo lokálním minimu [27]. Kritérium, definující shodu mezi vstupními vzory a uzlem reprezentujícím shluk ve vrstvě F2 můžeme v ART sítích nastavit podle potřeby volbou vigilance parametru, na základě kterého je spuštěna aktivace orientačního subsystému. Vyšší hodnota vigilance parametru představuje striktnější kritérium shody a v takovém případě je obvykle vstupní množina rozdělena do více přesnějších shluků. Nižší hodnota parametru toleruje větší rozdíly a vstupní množina je obvykle rozdělena do menšího množství obsáhlejších shluků. Proces učení je dále regulován pomocí kontroly zisku (viz Obr. 7), která slouží pro regulaci variability vstupních vektorů. Pokusme se nyní o hrubý popis mechanismu modelu ART:

- Předložení vstupního vektoru *x* srovnávací vrstvě. Ten aktivuje vzor v podobě krátkodobé paměti (STM – short term memory).

- Prostřednictvím vazeb vedoucích z F1 do F2 a pomocí laterální inhibice je vybrán uzel s největší excitací, který označíme jako *bm*.
- Vítězný uzel *bm* pak vyšle signál *d* směrem k nižší vrstvě. Tento se nazývá očekávání a je odvozen z předchozí zkušenosti. Tento zpětnovazební signál pak aktivuje ve srovnávací vrstvě nový vzor *X**.
- Původní vektor *x* je porovnán se s novým vzorem *X**.
- Jestliže podobnost mezi oběma vektory (vstupním a zpětnovazebním), tedy mezi realitou a očekáváním, je menší než předdefinovaná hodnota vigilance parametru, pak vybraný uzel *bm* nereprezentuje správnou třídu, do které patří náš vstup, a je tedy odstraněn z množiny možných vítězů.
- Pokud existuje další možný kandidát mezi zbývajícími neurony, otestujeme ho stejným způsobem. Pakliže žádný vhodný *bm* neexistuje, je vtažen do procesu další zatím neangažovaný neuron, který bude garantovat správné rozpoznání našeho vstupu.
- V případě, že podobnost mezi vstupním a zpětnovazebním signálem je větší než hodnota vigilance, tedy *bm* reprezentuje vektor *x*, dochází k jevu rezonance, k souladu mezi vstupem a očekáváním.
- Dojde-li k rezonanci, pak následuje proces adaptace vah sítě [20].

2.6.2 Inovace ART sítí

ART neuronové sítě jsou jedny z nejvíce inovovaných neuronových sítí. Schopnost rozpoznávat, samostatně kategorizovat a sekvenčně zpracovávat binární vzory má mnoho využití, často je ovšem nutné zpracovávat analogové vstupní vzory. Pro tento účel byl Grossbergem a Carpenterem vyvinut model ART2.

Na první pohled se model ART2 příliš neliší od ART1, ale na rozdíl od ART1 můžeme pracovat se spojitými i digitálními daty. Hlavní rozdíl v architektuře je v realizaci vstupní vrstvy F0. U architektury ART1 vstupní vrstvu F0 většinou nezobrazujeme, protože její jediný účel je symbolizovat jednotlivé binární prvky vstupního vektoru. U ART2 každý uzel F0 ve skutečnosti obsahuje šest uzlů, které slouží k převedení jednoho analogového prvku vstupního vektoru do vnitřní reprezentace. Ve většině případů je tato šestice uzlů značena pro zjednodušení pouze jako jeden uzel. Jednotlivé prvky vstupního vektoru jsou mezi sebou spojené pouze pomocí nezávislých proměnných, které jsou pro všechny uzly F0 společné [27].

ART1 a ART2 jsou základní používané modely, ale zdaleka ne jediné. Díky nesporným výhodám ART byly tyto dva modely dále rozvíjeny za účelem odstranění některých nedostatků a vylepšení vlastností sítě. Mezi nejznámější modely ART sítí patří následující:

- ART1 je první navržený model a vytvořil základ pro moderní ART sítě. Může pracovat pouze s binárními vstupy.
- ART2 je prvním modelem ART sítí, který je schopný pracovat s analogovými vstupy.
- ART2-A je efektivnější verzí ART2.

- FuzzyART model obohacuje standardní ART2 architekturu o využití fuzzy množin.
- ART3 využívá třetí paměť (MTM – medium term memory), která je inspirována chemickými procesy.
- ARTMAP je jediným probíraným modelem, který funguje na principu učení s učitelem. Skládá se ze dvou jednotek, jedné založené na ART1 a druhé založené na ART2 [27].
- TopoART je vylepšenou verzí FuzzyART. Oproti síti FuzzyART vykazuje nižší citlivost na šum a stabilnější výsledky shlukování. Více informací o této architektuře a jejích modifikacích je možné najít v kapitole 2.6.4.

2.6.3 Architektura FuzzyART

Architektury neuronových sítí testovaných v této práci vycházejí z principů, na kterých funguje neuronová síť FuzzyART, a proto je vhodné se zaměřit nejdříve právě na tuto architekturu. Základní verze architektury Fuzzy ART funguje na principu učení bez učitele, ale existuje i modifikace pro učení s učitelem zvaná Fuzzy ARTMAP [28]. Za návrhem těchto architektur stojí opět Stephen Grossberg a Gail Carpenter, autoři teorie adaptivní resonance. Na tomto projektu dále spolupracovali s Davidem Rosenem, matematikem a odborníkem na fuzzy množiny.

V běžné teorii množin se pouze rozlišuje, zda prvek do dané množiny náleží. Příslušnost prvku do dané množiny lze tedy vyjádřit pomocí binární hodnoty. Prvek do množiny buď náleží (binární hodnota 1), nebo do této množiny nenáleží (binární hodnota 0). Při práci s fuzzy množinami to není tak snadné. Pro každý prvek není pouze striktně stanovena příslušnost do určité množiny, ale je stanoven stupeň této příslušnosti. Velmi zjednodušeně lze říct, že některé prvky do množiny patří více než druhé [29].

FuzzyART zahrnuje všechny základní prvky ART neuronových sítí. Obsahuje tedy porovnávací vrstvu F1 a reprezentační vrstvu F2. Mezi těmito vrstvami probíhá komunikace pomocí adaptivních filtrů u a d a variabilita vstupních vektorů je kontrolována pomocí modulů kontroly zisku. Oproti standardní architektuře ART obsahuje FuzzyART navíc vstupní vrstvu F0. Právě díky této vrstvě jsou novější architektury ART neuronových sítí schopné pracovat nejen s binárními vstupní vektory, ale i analogovými. Pomocí této vrstvy jsou analogové vstupní vektory převedeny na signál reprezentovaný komplementárním kódem, který je pak prezentován porovnávací vrstvě F1 a zaregistrován jako nový vzor.

Z vrstvy F1 se pak tento signál šíří do reprezentační vrstvy F2 pomocí adaptivního filtru u. Z vrstvy F2 je poté pomocí adaptivního filtru d vyslán signál, který reprezentuje očekávání ve formě vzorové kategorie. Výsledkem tohoto procesu je buďto již popsaný stav resonance, nebo adaptivní paralelní prohledávání, jehož výsledkem je vytvoření nového shluku. Tento proces nyní rozebereme podrobněji.

Vstupní vektor (x) je nejprve ve vrstvě F0 transformován pomocí komplementárního kódování a dále šířen do vrstvy F1. Zde je tento vektor registrován jako aktivní vzor (X). Výstupní vektor této vrstvy je šířen skrz několik konvergujících a divergujících cest (filtr u). V průběhu tohoto procesu je vynásoben maticí adap-

tivních vah, které reprezentují dlouhodobou paměť. Tímto způsobem je vygenerován nový vektor, který slouží jako vstup pro vrstvu F2. Ve vrstvě F2 jsou uzly aktivované šířením vstupního vektoru a je vygenerován nový komprimovaný vektor (*c*). V architektuře Fuzzy ART bývá zvolen pouze jediný vítězný uzel mající nejvyšší hodnotu aktivace ve vrstvě F2, a proto je pouze jeden prvek vektoru nenulový. Nenulový prvek je výstupem vítězného uzlu, ten reprezentuje výsledný shluk. Aktivace uzlu F2 může být interpretována jako „tvoření hypotézy" o vstupním vektoru *x* [30].

Po aktivaci uzlů F2 je dále vygenerován signální vektor (*y*), který je odeslán do porovnávací vrstvy F1 pomocí druhého adaptivního filtru *d*. Při průchodu tímto filtrem je vektor vynásoben adaptivní váhovou maticí, která je v tomto filtru obsažena. Vynásobený vektor *y* je možné interpretovat jako „sítí naučená očekávání". Signál reprezentovaný vektorem *y* může být interpretován buď jako „testování hypotézy", nebo jako „vytvoření prototypu *Y*". Nyní je nutné, aby neuronová síť porovnala prototyp *Y* s původním vstupním vektorem *x*. V tomto procesu porovnávání může být změněn vzor *X* potlačením aktivace všech prvků vektoru *x*, které nejsou potvrzené vektorem *y*. Výsledný prototyp *X** symbolizuje takové prvky vstupního vektoru *x*, na základě kterých se neuronová síť rozhoduje. Pokud jsou očekávání *y* dostatečně podobná vstupu *x*, dosáhne neuronová síť stavu resonance. V tomto stavu se neuronová síť přizpůsobuje novým skutečnostem pomocí adaptace vah filtrů *u* a *d*. V této části se neuronová síť Fuzzy ART učí na základě prototypu *Y* místo původního zaregistrovaného vzoru *X* [30].

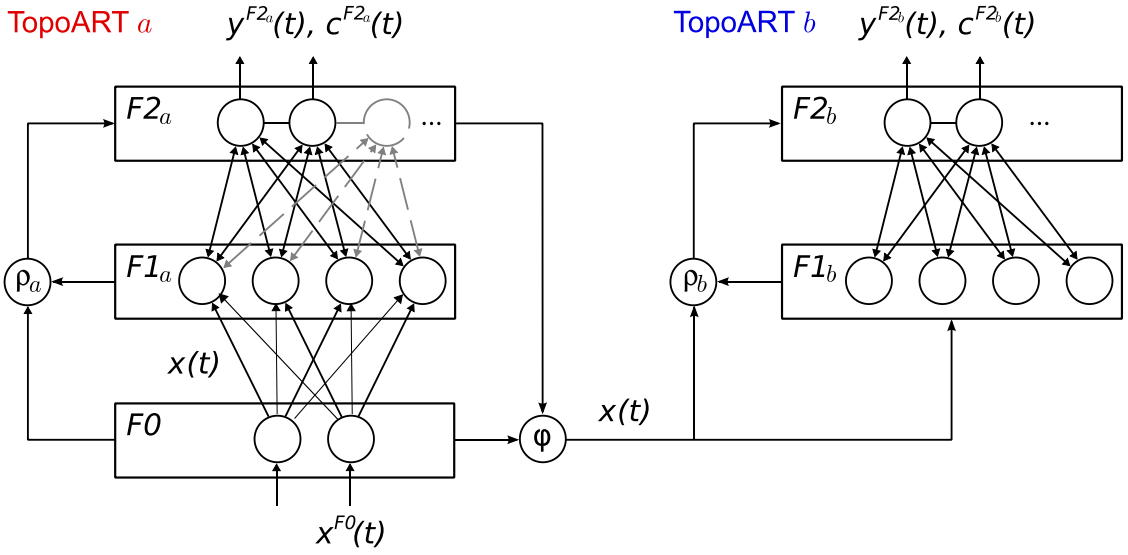
Kritérium akceptovatelné shody vektorů je definováno pomocí bezrozměrného parametru vigilance. Vigilance parametr definuje, jak blízko musí být vzor vstupního vektoru *X* k prototypu *Y*. Nízká hodnota tohoto parametru vede k obecným a méně přesným prototypům. S vyšší hodnotou tohoto parametru se kategorie, kterými jsou definované jednotlivé prototypy, zužují a zpřesňují. Například s nižší hodnotou vigilance parametru je FuzzyART schopná rozeznat obličeje a pokud hodnotu vigilance zvýšíme, může rozeznávat výrazy jednotlivých obličejů. Pokud je rozdíl mezi zaregistrovaným vzorem *X* a prototypem *Y* příliš velký na to, aby uspokojil kritérium shody dané parametrem vigilance, je zahájeno adaptivní prohledávání, jehož cílem je nalézt nový uzel F2, vhodný pro reprezentaci vstupního vektoru *x*. V průběhu tohoto procesu je zabráněno formování asociací mezi prototypy *Y* a *X**. Prohledávací proces resetuje *Y* předtím, než by se podobná asociace mohla zformovat. Po nalezení takového uzlu, který vykazuje dostatečnou shodu se vstupním vektorem *x* a vektorem *y* je zaveden nový uzel ve vrstvě F2, který reprezentuje nový shluk. Nakonec začne proces učení nové kategorie [30].

Proces učení, porovnávání a volby kategorie je u FuzzyART velmi podobný ART2 s tím rozdílem, že proces učení je přizpůsobený práci s fuzzy množinami.

2.6.4 Architektury TopoART a příslušné algoritmy

Architektury neuronových sítí TopoART vycházejí z teoretických pricipů Fuzzy ART. Mají stejný počet síťových vrstev, které mají stejnou úlohu. V zásadě se tyto architektury liší pouze ve dvou věcech. V reprezentační vrstvě F2 mají architektury

založené na TopoART navíc cesty, které vzájemně propojují jednotlivé uzly F2 a definují topologickou strukturu. Zavedením těchto cest je snížena citlivost na šum.



Obr. 8 Architektura TopoART [31]

Uzly F2 jsou nejprve označené jako dočasné a po určitém počtu kroků je vyhodnoceno, zda uzly ponechat (označit jako permanentní) nebo je odstranit.

Další výhodou architektury TopoART (viz Obr. 8) je možnost vytvoření většího množství modulů, které společně sdílejí vrstvu F0. Tyto další moduly se liší od základního modulu hodnotou vigilince parametru (ρ). Do dodatečných modulů jsou propagovány pouze takové uzly F2, které byly označené jako permanentní. Hodnota vigilance parametru (ρ) se automaticky zvyšuje v každém dalším dodatečném modulu. Na Obr. 8 můžeme vidět dva moduly (a, b) [31].

Nyní se podrobněji zaměříme na algoritmus sítě TopoART. V průběhu učení jsou vstupní vektory prezentovány vrstvě F0 v diskrétních časových krocích t. Každý r-rozměrný vstupní vektor $x^{\mathrm{F0}}(t)$ může být reprezentován následujícím způsobem (7):

$$x^{\mathrm{F0}}(t) = [x_1(t), \dots, x_r(t)] \tag{7}$$

Dříve, než je možné tento vstupní vektor prezentovat vrstvě F1, je nutné na něj aplikovat komplementární kódování, a proto každý prvek vstupního vektoru musí ležet v intervalu $\langle 0, 1 \rangle$. Po dokončení procesu kódování vypadá vstupní vektor následovně (8):

$$x(t) = [x_1(t), \dots, x_r(t), 1 - x_1(t), \dots, 1 - x_r(t)] \tag{8}$$

Dále je kódovaný vstupní vektor šířen do vrstvy F1, kde je zaregistrován, a z této vrstvy je vektor šířen do vrstvy F2 pomocí adaptivního filtru u. Každá z cest tvoří-

cích adaptivní filtr *u* obsahuje váhový vektor w_j, kde *j* je označení uzlu, do kterého daná cesta směřuje. Prvky váhového vektoru označují hraniční body jednotlivých kategorií, jejichž prototypy jsou dané uzly vrstvy F1. Aktivace *j*-téhu uzlu F2 (z_j) probíhá na základě následující funkce (9):

$$z_j = \frac{\left|x(t) \wedge w_j(t)\right|}{0{,}001 + \left|w_j(t)\right|} \tag{9}$$

Tato funkce je také nazývána funkcí volby a měří podobnost mezi vstupním vektorem a uzlem *j*. Symbol ∧ značí operátor MIN typický pro fuzzy množiny. Dělení členem $0{,}001 + \left|w_j(t)\right|$ vede k preferenci menších shluků před většími (pokud je to možné).

Dále je na vektor vstupující do této vrstvy aplikována další funkce, kterou nazýváme funkcí shody, na základě které je určeno, zda je daný shluk možné rozšířit o vstupní vektor, aniž by byla překročena maximální velikost *V* (10), která je závislá na počtu rozměrů (*r*) vstupního vektoru a hodnotě parametru vigilance (ρ). Výsledek funkce shody je šířen adaptivním filtrem *d* zpět do porovnávací vrstvy F1.

$$V = r(1 - \rho) \tag{10}$$

Aby bylo možné vstupní vektor do shluku zařadit, musí platit následující vztah (11):

$$\frac{\left|x(t) \wedge w_j(t)\right|}{x(t)} \geq \rho \tag{11}$$

Po aktivaci všech uzlů F2 jsou vybrány dva uzly: uzel s nejvyšší hodnotou aktivace *bm* a uzel s druhou nejvyšší hodnotou aktivace *sbm*. Pro tyto dva uzly *j* jsou upraveny váhy na cestách tvořících adaptivní filtr na základě následujícího vzorce (12):

$$w_j(t+1) = \beta\left(\mathrm{x}(t) \wedge w_j(t)\right) + (1 - \beta) w_j(t) \tag{12}$$

Hodnota β odpovídá parametru learning rates (rychlost učení), kterým je možné stanovit stupeň adaptace vah. Pro uzel *bm* je hodnota β rovna jedné. Pro uzel *sbm* je hodnota β stanovena pomocí volby daného parametru, je vhodné tuto hodnotu zvolit nižší než jedna a samozřejmě vyšší než nula. Při hodnotě nula by nedocházelo k žádné adaptaci vah. Uvedený adaptační vztah odpovídá rozšíření shluku o daný vstupní vektor. V průběhu učení se výsledné shluky nemůžou zmenšovat a díky tomu je dosaženo stabilních výsledků. Navíc je mezi uzly *bm* a *sbm* vytvořena cesta, kterou není možné dále modifikovat.

Za předpokladu, že neexistuje žádný existující uzel F2 reprezentující shluk, do kterého by bylo možné daný vstupní vektor zahrnout, vstoupí neuronová síť do procesu paralelního prohledávání, jehož cílem je najít nový uzel F2, který odpovídá danému vstupnímu vektoru. Pro snížení vlivu šumu obsahuje každý uzel F2 počí-

tadlo, které uchovává počet vstupní vektorů, na kterých daný uzel adaptoval svoje váhy. Po každých 200 celkových výukových krocích je na základě tohoto počítadla rozhodnuto, jestli bude uzel označen jako permanentní a je případně propagován do dalšího modulu. V procesu vyhodnocení dat je vstupní vektor propagován do obou modulů současně. Vigilance parameramетr (ρ) je možné stanovit pouze pro první modul *a*, vigilance parametr druhého modulu *b* je stanoven na základě hodnoty pro modul *a* (13) [31].

$$\rho_b = \frac{1}{2}(\rho_a + 1) \tag{13}$$

Neuronová síť Fast TopoART se od standardní sítě TopoART liší pouze v procesu učení. V situaci, kdy jsou nalezeny odpovídající uzly *bm* a *sbm*, je pro tyto uzly *j* provedena adaptace vah. V případě neuronové sítě Fast TopoART je tato adaptace provedena pomocí jednoduššího vztahu (14).

$$w_j(t+1) = \beta\left(x(t) \wedge w_j(t)\right) \tag{14}$$

Tato úprava vede k vyšší rychlosti učení, má ovšem negativní vliv na přesnost výsledného shlukování a zvyšuje citlivost na šum obsažený v datech [32].

Neuronová síť Hypersphere TopoART se od sítě TopoART liší v reprezentaci kategorií. Prototypy jednotlivých kategorií jsou reprezentovány stejně jako v TopoART pomocí uzlů F1. Tvar těchto kategorií je opět dán hodnotami vah adaptivního filtru. Reprezentace tohoto filtru je mezi těmito architekturami odlišná, což má na tvar kategorie samozřejmě vliv. Zatímco u neuronové sítě TopoART mají kategorie tvar hyperkvádru, u neuronové sítě Hypersphere TopoART mají jednotlivé kategorie tvar hyperelipsoidu a to vede i k rozdílné reprezentaci. Každá kategorie je popsána vektorem vzdáleností hraničního bodu od prototypu v jednotlivých rozměrech (μ) a průměrným poloměrem (R). Těmto kritériím bylo třeba přizpůsobit aktivační funkci (15) uzlů *j* vrstvy F2.

$$z_j(t) = \frac{H - \max\left(R_j,\ \left|x(t) - \mu_j(t)\right|\right)}{H - R_j + 0{,}001} \tag{15}$$

Funkce max vrací nejvyšší hodnotu z hodnot obsažených v závorce a proměnná H je stanovena na základě vzorce (16). Dělení výrazem $H - R_j + 0{,}001$ vede k preferenci menších shluků před většími.

$$H = \frac{1}{2}\sqrt{\sum_{i=1}^{r}(x_i^{\max} - x_i^{\min})^2} \tag{16}$$

Proměnné $x_i^{\max}$ a $x_i^{\min}$ značí očekávané minimum a maximum v rozměru *i*.

Stejně jako v případě TopoART je po stanovení uzlu *bm* nutné ověřit, zda se shluk reprezentovaný tímto uzlem může rozšířit o daný vstupní vektor a zároveň nepřekročit maximální velikost V (17). Funkce shody (18) vypadá v případě Hypersphere TopoART následovně:

$$V = H(1 - \rho) \tag{17}$$

$$1 - \frac{\max\left(R_j,\ \left|x(t) - \mu_j(t)\right|\right)}{H} \geq \rho \tag{18}$$

Na rozdíl od TopoART je v tomto případě velikost V nezávislá na počtu rozměrů (r), což vede k vyšší flexibilitě neuronové sítě.

Posledním rozdílem je způsob, jakým se v procesu učení mění hodnoty μ (19) a R (20) v adaptivních filtrech ve stavu resonance. Opět jsou adaptovány pouze váhy takových uzlů j, které byly označeny jako *bm* a *sbm.* Hodnota β je pro *bm* rovna 1. Pro *sbm* je tato hodnota opět volitelná. Funkce min funguje obdobně jako výše zmíněná funkce max, ale vrací nejnižší hodnotu z hodnot obsažených v závorce [33].

$$\mu_j(t+1) = \mu_j(t) + \frac{\beta_j}{2}\left(1 - \frac{\min\left(R_j(t),\ \left|x(t) - \mu_j(t)\right|\right)}{\left|x(t) - \mu_j(t)\right|}\right)\left|x(t) - \mu_j(t)\right| \tag{19}$$

$$R_j(t+1) = R_j(t) + \frac{\beta_j}{2}\left(\max\left(R_j(t), \left|x(t) - \mu_j(t)\right| - R_j(t)\right)\right) \tag{20}$$

3 Metodika

Hlavním cílem této práce je určit a popsat vlastnosti neuronových sítí založených na teorii adaptivní resonance a dále ohodnotit možnosti jejich praktického využití.

K tomuto účelu je nutné vytvořit vhodný nástroj implementující principy teorie adaptivní resonance a následně jej otestovat na různých typech dat. Tato kapitola podrobně popisuje zdroje dat, na kterých byly provedeny jednotlivé testy, způsoby, jakými byla data upravena, i způsob realizace samotných neuronových sítí a vyhodnocení dosažených výsledků.

3.1 Prostředky využité k realizaci ART neuronových sítí

K otestování byly vybrány tři architektury ART neuronových sítí: TopoART, Fast TopoART a HypersphereTopoART. Nejdříve bylo nutné vytvořit nástroj, který principy těchto sítí vhodně implementuje. Tento nástroj byl vytvořen pomocí programovacího jazyku C# a dále využíval knihovny LibTopoART a knihovnu WindowsForms.

3.1.1 Programovací jazyk C#

C# je objektově orientovaný programovací jazyk podobný jazykům C++ a JAVA. Byl vytvořen společností Microsoft a původně sloužil pro implementaci knihoven, tříd a funkcí pro framework (podpůrná softwarová struktura) .NET. Protože C# patří mezi jazyky kompilované, je nutné zdrojový kód napsaný v tomto jazyce před spuštěním nejprve převést do strojového jazyka (zkompilovat). Zkompilované kódy po spuštění využívají prostředků virtuálního stroje CLR (Common Language Runtime). CLR je tedy nezbytný pro spouštění jakéhokoliv C# kódu a je součástí všech operačních systémů Windows novějších než Windows 98.

3.1.2 Knihovna Windows Forms

Knihovna Windows Forms je součástí frameworku Microsoft .NET a je výhradně určena pro tvorbu grafických uživatelských rozhraní (GUI) pro operační systémy Microsoft Windows. Pro správný chod GUI realizovaných pomocí Windows Forms je nutné mít nainstalovaný systém Windows novější než Windows XP, service pack 3.

3.1.3 Knihovna LibTopoART

Knihovna LibTopoART implementuje principy čtyř architektur neuronových sítí navržených Markem Tscherepanowem. Konkrétně jde o následující ART neuronové sítě: TopoART, Fast TopoART, Episodic TopoART a HypersphereTopoART. Knihovna byla vytvořena promocí nástroje MONO 3.12.0 pro .NET 4.5 a je možné ji využívat na všech systémech Windows novějších než Windows Vista [34].

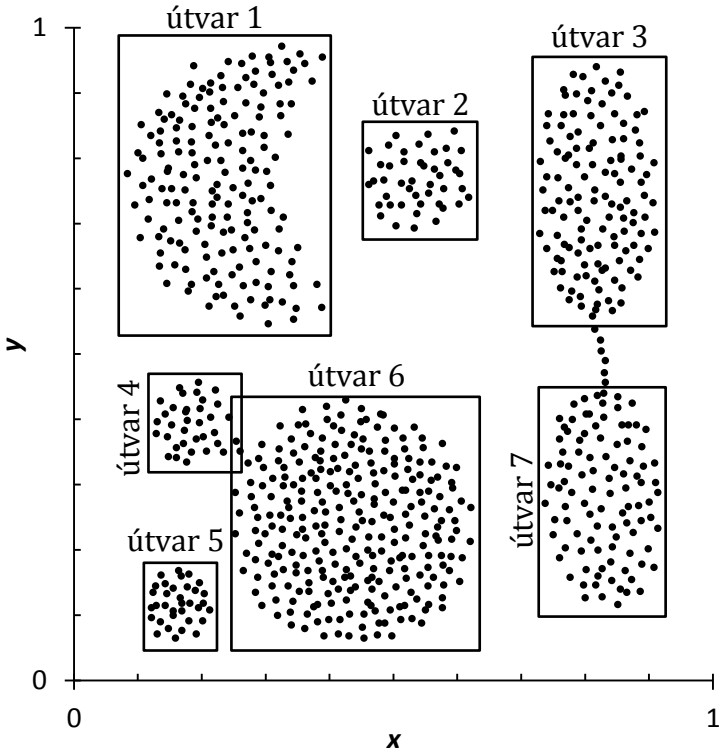
3.2 Data

Pro stanovení vlastností jednotlivých architektur neuronových sítí je nutné je otestovat na různých typech dat, abychom byli schopni určit, které vlastnosti sítí zůstávají neměnné, a naopak, které vlastnosti sítě vykazují pouze pro určitý typ dat.

Testy byly provedeny na dvou umělých datesetech (datových sadách) a jednom reálném datasetu. Umělý dataset Agregation je nejjednodušší z testovaných datasetů a je vhodný pro stanovení základních vlastností testovaných sítí a jejich úspěšnosti na dvourozměrných datech. Dalším testovaným umělým datasetem byl Hyperkvádrový dataset, který obsahuje dva shluky v mnohorozměrném prostoru. Tento dataset je vhodný pro stanovení vlastností a úspěšnosti sítí na mnohorozměrných datech a pro stanovení faktorů, které mají vliv na spotřebu času pro shlukování. Poslední testovaný dataset Hotelová hodnocení byl tvořený reálnou sbírkou textových hodnocení psaných v přirozeném jazyce. Pro taková data je typická reprezentace pomocí řídkých vektorů. Tento dataset byl vhodný pro stanovení reálného využití, vlastností a úspěšnosti na složitých velmi objemných datech.

3.2.1 Agregation dataset

Agregation (agregace) dataset obsahuje 788 bodů v dvourozměrném prostoru. Každý z těchto bodů je reprezentován dvojicí souřadnic *x* a *y*. Pomocí těchto je vytvořeno celkem sedm rovinných geometrických útvarů.



Obr. 9 Geometrické útvary

Některé z těchto útvarů jsou od sebe odděleny různě velkou prázdnou oblastí, jiné útvary jsou mezi sebou záměrně propojeny různými způsoby. Všechny útvary a jejich značení můžeme vidět na Obr. 9. Tento dataset byl získán ze stránek University of Eastern Finland [35].

3.2.2 Hyperkvádrový dataset

Hyperkvádrový dataset obsahuje dva shluky bodů ohraničené tvarem hyperkvádru v dvoutisícirozměrném prostoru. V každém rozměru je tento prostor ohraničen nulou a jedničkou. Pro každý shluk byl náhodně stanoven jeho střed a kolem tohoto středu bylo náhodně generováno 1 000 bodů, které jsou v každém rozměru od středu vzdáleny maximálně o stanovenou hodnotu. Celkem tedy dataset obsahoval 2 000 bodů. Při použití tohoto postupu je velmi pravděpodobné, že se vzniklé shluky tvaru hyperkvádru budou v mnoha rozměrech překrývat.

3.2.3 Hotelová hodnocení

Tento reálný dataset obsahuje kolekci slovních hodnocení služeb a ubytování, které bylo napsané hotelovými zákazníky v různých zemích. K těmto hodnocením byla dále vždy přidělena hodnota, která vypovídá o spokojenosti hostů. Hodnota 1 odpovídala pozitivnímu hodnocení a hodnota 2 odpovídala hodnocení negativnímu. Tato data byla převzata z portálu Booking.com [36] a pro účely této práce byla vybrána pouze data v angličtině.

3.3 Příprava dat

Většinu datových zdrojů nebylo možné analyzovat v jejich původní formě a bylo nutné je převést do formy k těmto účelům vhodné. Nejběžnějším formátem využívaným pro dolování znalosti z dat je formát CSV.

3.3.1 CSV

Jak již bylo řečeno, CSV je běžný formát v oblasti dolování znalostí z dat. Zkratka CSV pochází z anglického spojení comma-separated values (hodnoty oddělené čárkami). Soubory tohoto typu obsahují pro každý záznam, který chceme zpracovat, jeden řádek. Každý z těchto řádků obsahuje stejný počet hodnot oddělených čárkami, které popisují jednotlivé zkoumané vlastnosti záznamu.

Při využití tohoto formátu může v českém jazyce nastat problém, protože u desetinných čísel využíváme čárku a není jasné, zda čárka odděluje desetinné místo nebo jednotlivé hodnoty. Z tohoto důvodu bylo nezbytné zvolit jednu z běžně používaných modifikací formátu CSV, ve které jsou jednotlivé hodnoty oddělené středníkem místo čárky.

3.3.2 Tabulka výskytů slov

Tabulka počtu výskytů slov je jedním z běžných řešení, kterým je možné reprezentovat textová data ve formě matice a uchovávat je ve formátu CSV. Pro vytvoření podobné matice je nutné zjistit, s jakou frekvencí se vyskytují slova v jednotlivých zkoumaných záznamech a zapsat počet výskytů jednotlivých slov pro každý záznam do tabulky. Každý řádek této tabulky reprezentuje jeden záznam a ve sloupcích je zapsán počet výskytů jednoho slova definovaného pro daný sloupec.

3.3.3 Normalizace Min-Max

Normalizace Min-Max je metoda používaná pro převedení hodnot z určitého intervalu do jiného požadovaného intervalu. K tomuto účelu je využívána technika lineární transformace. Obvykle se metoda Min-Max využívá pro převedení analyzovaných hodnot do intervalu mezi nulou a jedničkou na základě následující rovnice (21). Pro tuto rovnici musejí být správně zvoleny parametry *min* a *max*. Parametr *min* reprezentuje nejnižší hodnotu původního intervalu a parametr *max* reprezentuje naopak nejvyšší hodnotu původního intervalu. Proměnná x_p reprezentuje převáděnou hodnotu v původním intervalu a proměnná x_n reprezentuje stejnou hodnotu převedenou do cílového intervalu.

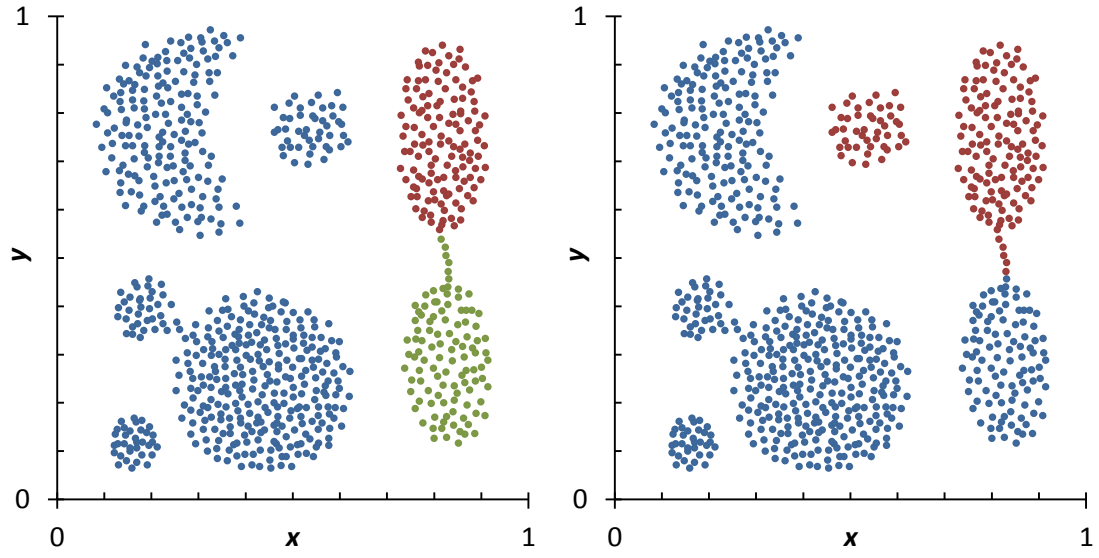
$$x_n = \frac{x_p - min}{max - min} \tag{21}$$

4 Výsledky

4.1 Agregation dataset

Agregation dataset obsahuje celkem sedm souvislých geometrických útvarů v dvourozměrném prostoru. Tyto útvary jsou tvořeny množinami bodů. Body, které náleží do stejného útvaru (množiny), tvoří obrazce s rozdílným tvarem i velikostí. Jednotlivé útvary a jejich značení můžeme vidět na Obr. 9. Výsledky shlukování pro dvourozměrná data je možné velmi dobře vizualizovat. Tento dataset je vhodný pro stanovení vlivu parametru vigilance na výsledek shlukování a určení základních vlastností, které jednotlivé architektury ART neuronových sítí vykazují, a jejich názornou ilustraci.

Nejdříve se zaměříme na vliv parametru vigilance na finální výsledky shlukování. Tento parametr ovlivňuje velikost jednotlivých kategorií, jejichž prototypy jsou reprezentovány uzly vrstvy F1. Velikost těchto kategorií a jejich vzájemná pozice má silný vliv na tvar a počet finálních shluků.



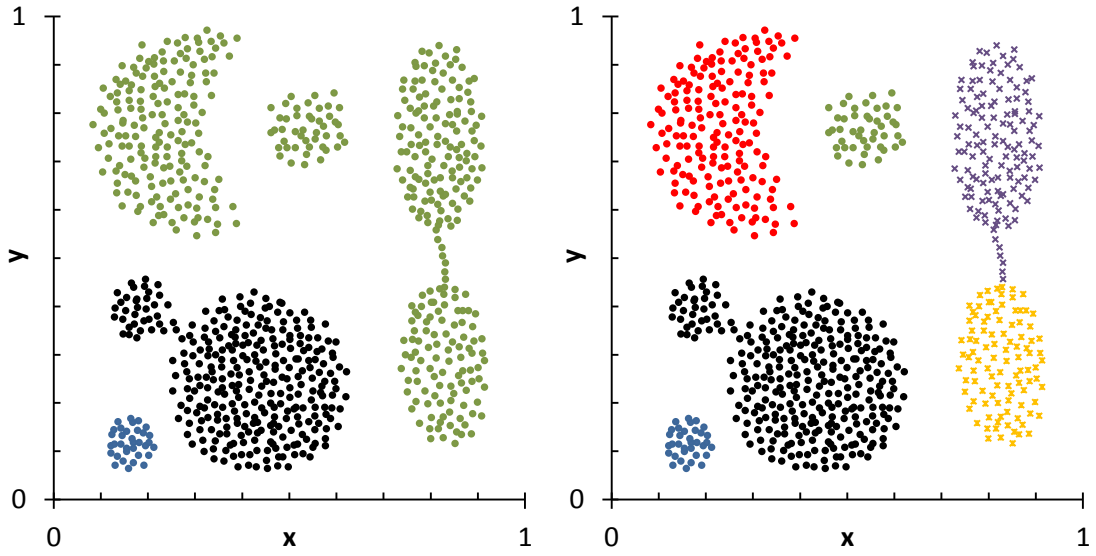
Obr. 10 Shlukování s vigilance parametrem 0,7

Nejdříve byly jednotlivé neuronové sítě testovány při hodnotě vigilance parametru 0,7 a parametru learning rates 0,5. Při nižších hodnotách parametru vigilance se neuronovým sítím data nedařilo rozdělit do více než jednoho shluku. Parametr learning rates měl na výsledek shlukování zanedbatelný vliv. Neuronové sítě TopoART a FastTopoART na těchto datech dosahovaly stejných výsledků shlukování a z tohoto důvodu budou komentovány dohromady. Výsledek shlukování pro tyto dvě neuronové sítě můžeme vidět na Obr. 10 vlevo, zatímco výsledky shlukování dosažené neuronovou sítí Hypersphere TopoART vpravo. Jednotlivé nalezené shluky jsou barevně odlišeny.

Sítím TopoART a FastTopoART se při tomto nastavení podařilo oddělit tři shluky. Velké množství bodů bylo zařazeno do shluku označeného modře i přesto, že jednoznačně tvoří pět odlišných útvarů s různou mírou oddělení. Červený shluk obsahoval pouze body, které tvoří jeden útvar, a proto je možné konstatovat, že byl oddělen přesně. Zelený shluk obsahoval body, které jednak tvoří souvislý tvar, a také body, které tvoří spojnici vedoucí k útvaru 7.

Neuronové síti Hypersphere TopoART se podařilo při stejně nastavených parametrech data rozdělit pouze do dvou shluků. Modře vyznačený shluk obsahoval většinu bodů. Tyto body tvořily pět samostatných oddělených útvarů. Lze konstatovat, že modrý shluk lze dále rozdělit s vyšší přesností. Červený shluk obsahoval body dvou oddělených útvarů (útvar 2 a 3), které by také bylo možné rozdělit přesněji.

Je patrné, že při hodnotě vigilance parametru 0,7 se data nepodařilo rozdělit s dostatečnou přesností. Tento problém úzce souvisí s nastavením hodnoty tohoto parametru. Při této hodnotě měly kategorie (uzly F1) vzniklé v procesu shlukování příliš velkou velikost na to, aby bylo možné jejich pomocí data rozdělit s dostatečnou přesností. V takové situaci je vhodné experiment opakovat s vyšší hodnotou vigilance parametru.



Obr. 11 Shlukování s vigilance parametrem 0,8

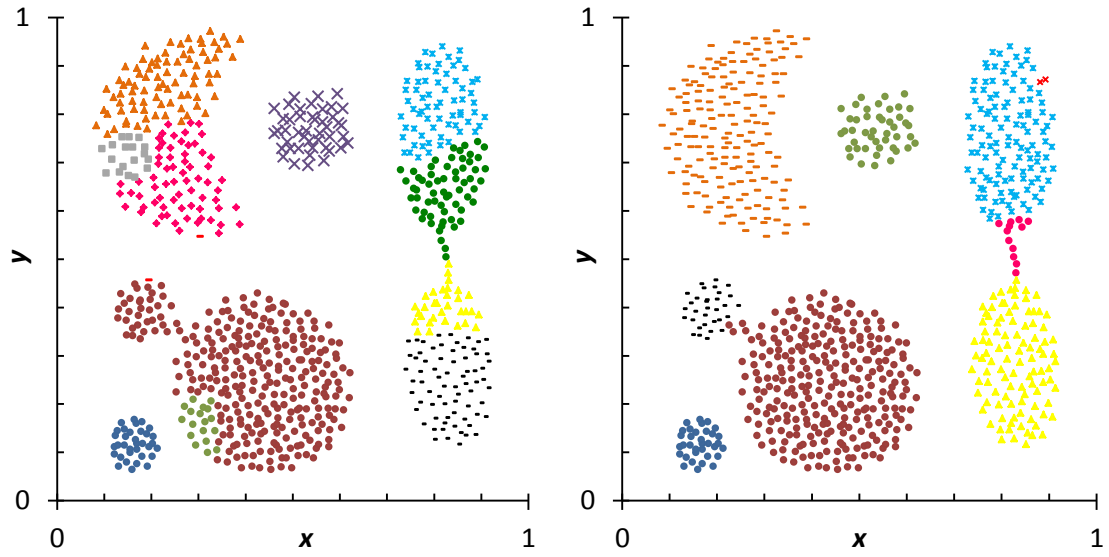
Pro další experimenty byla zvolena hodnota vigilance parametru 0,8. Výsledky poskytnuté neuronovými sítěmi TopoART a FastTopoART je opět možné vidět v levé části Obr. 11 a výsledky poskytnuté neuronovou sítí Hypersphere TopoART jsou v pravé části obrázku.

Sítě TopoART a FastTopoART byly opět schopné data rozdělit do tří shluků. Tyto tři shluky se ovšem výrazně lišily od shluků nalezených s hodnotou vigilance parametru 0,7. Nejvíce bodů bylo obsaženo v zeleném shluku. Zatímco největší shluk v předchozím experimentu obsahoval útvary 1, 2, 4, 5 a 6, při vyšší hodnotě

parametru vigilance byly do největšího shluku zahrnuty útvary 1, 2, 3 a 7. Černý shluk obsahoval body, které náleží do dvou útvarů. Tyto útvary nejsou dobře oddělené, a proto byly zahrnuty do jednoho shluku. Pomocí modrého shluku byl přesně oddělen poslední útvar.

Neuronová síť Hypersphere TopoART při stejných hodnotách parametrů dosáhla výrazně přesnějšího rozdělení do shluků. Shluky vyznačené červeně, zeleně, fialově a žlutě obsahovaly pouze body příslušných útvarů. Fialový shluk obsahoval pouze body útvaru 3 a body tvořící spojnici vedoucí k útvaru 7, ovšem útvary 3 a 7 se podařilo oddělit do samostatných shluků. Černý shluk obsahoval body útvarů 4 a 6. Podobně jako u neuronové sítě TopoART se tyto dva útvary nepodařilo oddělit ani při vyšší hodnotě vigilance parametru.

Zatímco neuronové sítě TopoART a Fast Topo ART stále nebyly schopné data rozdělit dostatečně přesně, výsledky poskytnuté neuronovou sítí Hypersphere TopoART se výrazně vylepšily. Na základě dalšího experimentu bude stanoveno, zda je možné výsledky shlukování vylepšit pomocí volby vyšší hodnoty parametru vigilance.



Obr. 12 Shlukování s vigilance parametrem 0,9

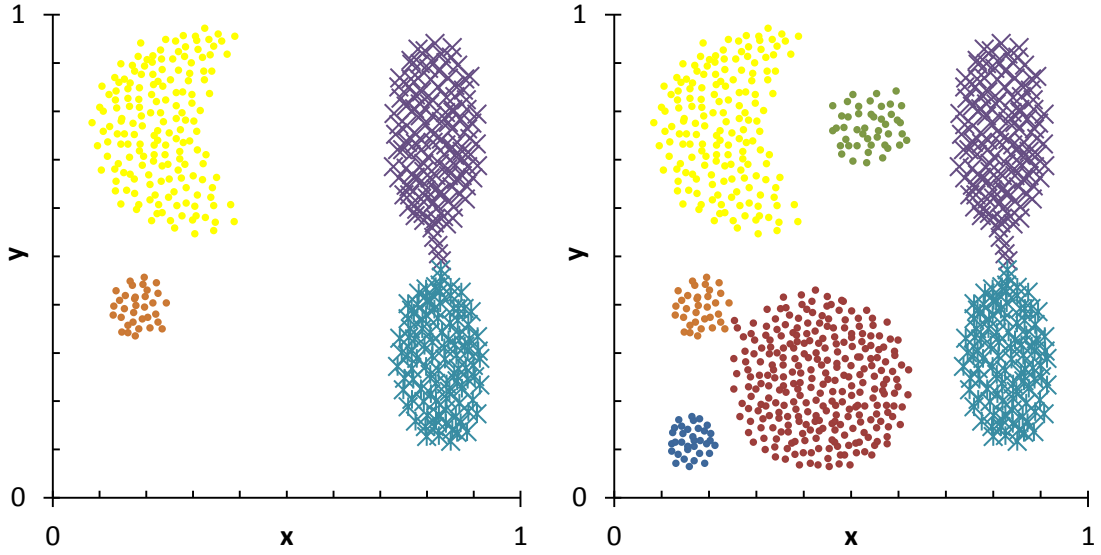
Výsledky poskytnuté neuronovými sítěmi TopoART a FastTopoART při hodnotě parametru vigilance 0,9 vidíme na Obr. 12 v levé části a výsledky poskytnuté neuronovou sítí Hypersphere TopoART se nacházejí v pravé části obrázku.

Neuronové sítě TopoART a FastTopoART rozdělily vstupní data do jedenácti shluků. Vysoký počet shluků svědčí o vyšší přesnosti rozdělení dat, ovšem nelze říci, že se jedná o optimální výsledek. Souvislé útvary 1, 3, 6 a 7 byly rozděleny do dvou a více shluků. Za pozornost stojí především shluk vyznačený červeně. Tento shluk obsahuje body útvarů 4 a 6. Zatímco se od sebe nepodařilo oddělit tyto dva útvary, od útvaru 6 byl oddělen další shluk. Pro další experimenty by bylo vhodné zvolit pro sítě TopoART a Fast TopoART hodnotu vigilance parametru mezi čísly

0,8 a 0,9. Na základě dalších experimentů byla stanovena optimální hodnota vigilance parametru pro tyto sítě 0,86 (viz Obr. 13 – vpravo). Při této hodnotě se podařilo data rozdělit do sedmi shluků.

Neuronové síti Hypersphere TopoART se při hodnotě vigilance parametru 0,9 podařilo vstupní data rozdělit velmi přesně do osmi shluků. (viz Obr. 12 – vpravo). Každý shluk obsahuje pouze body, které náleží pouze jednomu útvaru, a nedochází k rozpadu jednotlivých útvarů na více shluků. Za zmínku stojí fakt, že se podařilo rozdělit útvary 4 a 6 do samostatných shluků a také oddělit shluk označený růžovou barvou, který tvoří spojnici mezi útvary 3 a 7. Na těchto datech se ukázala hodnota vigilance parametru 0,9 jako optimální, při vyšších hodnotách tohoto parametru dochází k nežádoucímu rozpadu jednotlivých útvarů do většího počtu shluků.

Provedené experimenty vhodně ilustrují vliv parametru vigilance na kvalitu výsledného shlukování, ovšem optimální hodnota tohoto parametru se liší v závislosti na povaze testovaných dat a musí být stanovena na základě experimentů, nebo předem získaných znalostí. Při příliš nízkých hodnotách tohoto parametru mají kategorie reprezentované uzly F1 velké rozměry. To má za následek vznik velkých shluků, které by bylo vhodné dále rozdělit. Při příliš vysokých hodnotách tohoto parametru naopak vzniká velké množství malých shluků a dochází k nechtěnému oddělení souvislých útvarů.



Obr. 13 Dodatečná adaptace neuronové sítě TopoART

Velkou výhodou ART neuronových sítí je fakt, že fáze učení a fáze kategorizace dat není striktně oddělena. Díky tomu je možné již naučenou neuronovou síť adaptovat nově vzniklým skutečnostem bez nutnosti začít s celým procesem znovu. Tato schopnost byla ověřena následujícím experimentem.

V prvním kroku experimentu byla neuronová síť TopoART s parametrem vigilance 0,86 a parametrem learning rates 0,5 naučena na bodech útvarů 1, 3, 4 a 7.

Následně byla naučená neuronová síť otestována na celém datasetu. Výsledek shlukování je možné vidět v levé části Obr. 13. V této situaci byla neuronová síť schopna rozeznat pouze ty body, na kterých jsme ji naučily. Následně byl proces učení zopakován. Tentokrát byla neuronovou síť učena pouze na bodech útvarů 2, 4 a 5 a znovu otestována na celém datasetu. Výsledek shlukování je možné vidět v pravé části Obr. 13. I když byla neuronová síť v druhém kroku učena pouze na bodech útvarů 2, 4 a 5, byla schopná rozeznat i shluky, které se naučila v předchozím kroku.

4.2 Hyperkvádrový dataset

Hyperkvádrový dataset obsahuje celkem dva shluky. Každý z těchto shluků reprezentuje množinu 1 000 bodů v 2 000 rozměrném prostoru ohraničenou hyperkvádrem. Celkem dataset obsahuje 2 000 bodů. Každý z těchto bodů je reprezentován neřídkým vstupním vektorem tvořeným 2 000 prvky. Obsažené shluky se v mnoha rozměrech překrývají, což komplikuje proces analýzy. Tento dataset je vhodný pro stanovení úspěšnosti a vlastností, které jednotlivé testované ART neuronové sítě vykazují při práci s daty reprezentovanými vstupními vektory s velkým počtem prvků (mnohorozměrná data). Dále byly na tomto datasetu provedeny testy pro stanovení faktorů, které mají vliv na spotřebu času potřebného k analýze. Vliv jednotlivých faktorů byl definován.

4.2.1 Úspěšnost testovaných ART neuronových sítí

Jednotlivé testované architektury produkovaly velmi odlišné výsledky shlukování v závislosti na pořadí vstupní vektorů v průběhu procesu učení, a proto byly vyhodnoceny zvlášť pro dvě situace. V první situaci byly vstupní vektory rozděleny do skupin podle shluku, do kterého náleží. Následně byly ART neuronové sítě učeny na jednotlivých skupinách vstupních vektorů odděleně. V druhé situaci vstupní vektory přicházely do procesu učení v náhodném pořadí.

Výsledky dosažené v situaci, ve které byla vstupní data rozdělena do skupin podle shluků, byly pro všechny testované architektury neuronových sítí velmi příznivé. Pro nastavení parametru vigilance 0,3 a parametru learning rates 0,5 byly všechny testované neuronové sítě schopné rozdělit vstupní data do dvou shluků. Neuronové sítě TopoART a HypersphereTopoART byly schopné rozeznat prvky jednotlivých shluků s přesností 100 %. Neuronová síť Fast TopoART rozeznala prvky jednotlivých shluků s přesností 99 %. Zbylé jedno procento nesprávně rozeznaných prvků mělo velmi typické znaky. Vždy se jednalo prvky patřící do skupiny, která byla neuronovou sítí zpracovávána až jako druhá. První prvky této skupiny byly nesprávně identifikovány a byly zařazeny do stejného shluku jako prvky skupiny, na kterých se neuronová síť učila nejdříve. Například v situaci, kdy jsme neuronovou síť Fast TopoART učili nejdříve na skupině, která obsahovala prvky prvního shluku, a následně ji začali učit na skupině obsahující prvky druhého shluku, část prvků patřících do druhého shluku byla mylně zařazena do shluku prvního. Tento problém bylo možné částečně vyřešit snížením hodnoty parametru learning

rates. Při hodnotě parametru learning rates 0,3 byl vliv pořadí nepatrně zmírněn, ovšem částečně byla omezena schopnost neuronové sítě se adaptovat. To by mohlo mít negativní vliv v situaci, kdy bychom chtěli neuronovou síť v budoucnu přizpůsobit novým okolnostem.

Ne v každé situaci je možné výuková data rozdělit do podobných skupin. Z tohoto důvodu byly jednotlivé testy provedeny znovu s jediným rozdílem. V této situaci byly jednotlivé vstupní vektory prezentovány jednotlivým ART neuronovým sítím ve zcela náhodném pořadí. Tato změna v procesu učení výrazně ovlivnila kvalitu výsledků poskytnutých jednotlivými sítěmi.

Zatímco neuronová síť Hypersphere TopoART dosáhla se stejnými hodnotami parametrů vigilance a learning rates na neseřazených datech stejných výsledků jako na datech seřazených, výsledky poskytnuté neuronovou sítí TopoART a FastTopoART se výrazně lišily. Při stejných hodnotách parametrů neuronové sítě TopoART a FastTopoART rozdělily vstupní data do 48 shluků. Tyto shluky rozdělovaly data především v závislosti na pořadí, ve kterém se na nich neuronové sítě učily. Všechna data obsažená uvnitř jednoho shluku byla zpracována v procesu učení za sebou. Mnohem lepší úrovně kategorizace dosáhly tyto neuronové sítě TopoART a FastTopoART při volbě vigilance parametru 0,24. Při této hodnotě vigilance se podařilo těmto sítím data rozdělit opět do dvou požadovaných shluků (viz Tab. 1).

	Hypersphere TopoART	TopoART	Fast TopoART
vigilance	0,3	0,24	0,24
shluk1 - celkem [%]	50	53	54
shluk2 - celkem [%]	50	47	46
shluk1 - správně [%]	100	52	51
shluk2 - správně [%]	100	51	51
celková úspěšnost [%]	100	52	51

Tab. 1 Výsledky na neseřazených datech

Neuronová síť Hypersphere TopoART dosáhla opět úspěšnosti 100 % při hodnotě vigilance parametru 0,3. Každý z výsledných shluků obsahoval 50 % všech testovaných záznamů a oba shluky byly naprosto čisté.

Výsledky poskytnuté neuronovou sítí TopoART byly výrazně horší než na seřazených datech. Zatímco na seřazených datech tato neuronová síť dosáhla celkové úspěšnosti 100 %, na neseřazených datech dosáhla i s upravenou hodnotou vigilance parametru (0,24) úspěšnosti pouze 52 %. První shluk obsahoval 53 % všech záznamů, ale pouze 52 % těchto záznamů bylo do tohoto shluku zařazeno správně a zbylých 48 % mělo patřit do shluku druhého. Druhý shluk obsahoval 47 % záznamů, z nichž 51 % bylo zařazeno správně. Z těchto výsledků vyplývá, že neuronová síť TopoART selhala v rozdělení vstupních dat do požadovaných shluků.

Neuronová síť Fast TopoART poskytla velmi podobné výsledky. Větší shluk obsahoval 54 % všech testovaných záznamů a zbylých 46 % patřilo do shluku menšího. Celková úspěšnost neuronové sítě Fast TopoART byla 51 %. Vysoká míra

neúspěšnosti sítí TopoART a Fast TopoART mohla být částečně způsobena nižší hodnotou vigilance parametru, ale při vyšších hodnotách parametru vigilance se pouze zvyšoval počet nalezených shluků bez jakékoliv závislosti na shlucích, které data skutečně obsahovala. Hlavním důvodem této nepřesnosti se ovšem zdá být závislost velikosti kategorií na počtu rozměrů, kterou tyto dvě neuronové sítě na rozdíl od Hypersphere TopoART vykazují. Při vysokém počtu rozměrů testovaných dat se u těchto neuronových sítí výrazně zvyšuje vliv pořadí vstupních vektorů na výsledek shlukování.

Z těchto poznatků lze usoudit, že využití neuronových sítí TopoART a Fast TopoART je pro mnohorozměrná data vhodné pouze za předpokladu, že jsme schopni kontrolovat pořadí vstupů ve fázi učení sítí. Neuronová síť Hypersphere TopoART nebyla pořadím vstupů ovlivněna, a proto je pro zpracování podobných dat mnohem vhodnější.

4.2.2 Spotřeba času

Jednotlivé testované architektury ART neuronových sítí byly vyhodnoceny z hlediska času, který potřebují ke zpracování dat. Cílem bylo zjistit, jaké faktory mají na spotřebu času vliv a jakým způsobem ji ovlivňují. Všechny testy byly provedeny na počítači s procesorem Intel Core i3-2350M 2,30 GHz a operační pamětí o velikosti 6 GB.

Protože testované ART neuronové sítě jsou schopné zpracovávat každý jednotlivý vstupní vektor odděleně, celková doba procesu se dá jednoduše určit jako suma časů, potřebných ke zpracování jednotlivých vstupních vektorů. Daleko větší problém je určit právě spotřebu času pro zpracování jednotlivých vstupních vektorů, protože tato spotřeba se mění v průběhu procesu. Nejdříve byl otestován vliv jednotlivých parametrů sítě a dalších faktorů na časovou spotřebu. Mezi tyto faktory patří počet prvků vstupního vektoru (vrstva F0), learning rates parametr, vigilance parametr, počet nalezených shluků (vrstva F2) a počet uzlů v porovnávací vrstvě F1.

Počet prvků vstupního vektoru měl na spotřebu času velmi velký vliv, protože tento parametr ovlivňuje celkový počet uzlů ve vrstvě F0 neuronové sítě. S každým novým prvkem vstupního vektoru se tento počet uzlů F0 zvyšuje o jedna. Na základě měření bylo stanoveno, že s rostoucím počtem uzlů ve vrstvě F0 roste spotřeba času lineárně.

Parametr learning rates neměl na spotřebu času žádný vliv, protože pouze ovlivňuje míru adaptace vah ve stavu resonance. Dalším testovaným faktorem byl parametr vigilance. Tento parametr ovlivňoval spotřebu času nepřímo. Parametr vigilance má vliv na počet uzlů F1 a na počet nalezených shluků, ovšem tyto hodnoty jsou ovlivněné především vstupními daty, které analyzujeme. Právě počet uzlů v porovnávací vrstvě F1 měl zásadní vliv na celkový čas, který potřebujeme k analýze. Na základě měření bylo stanoveno, že s rostoucím počtem uzlů vrstvy F1 roste spotřeba času na zpracování jednoho vstupního vektoru opět lineárně. Výsledný počet shluků neměl sám o sobě na spotřebu času žádný zaznamenatelný vliv.

Celkový čas (t) potřebný ke zpracování jednoho vstupního vektoru je tedy ovlivněn třemi faktory: počtem uzlů ve vrstvě F0 (p_{F0}), současným počtem uzlů ve vrstvě F1 (p_{F1}) a časovou konstantou (k). Čas (t) potřebný ke zpracování se tedy dá určit pomocí následujícího vzorce (22):

$$t = kp_{F0}p_{F1} \tag{22}$$

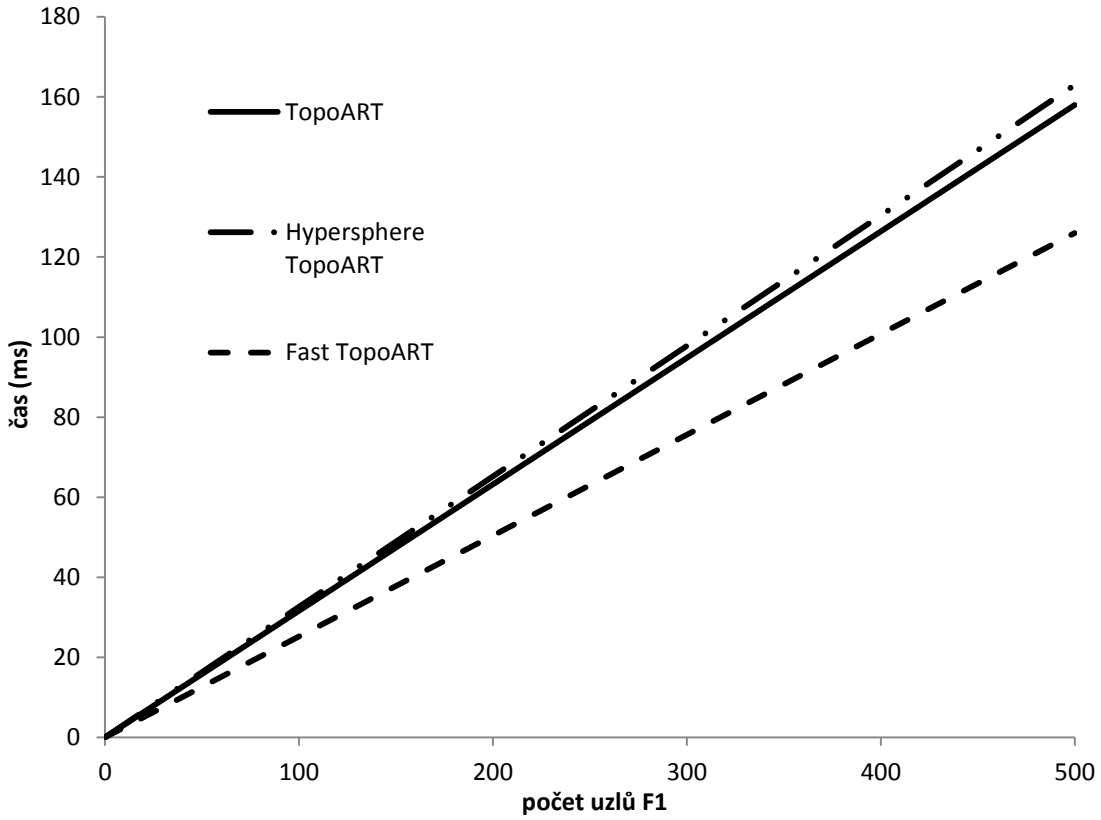
Časová konstanta (k) je rozdílná pro jednotlivé testované architektury neuronových sítí a je samozřejmě závislá i na dostupných výpočetních prostředcích. Hodnoty této konstanty naměřené pro testované architektury ART neuronových sítí můžeme vidět v Tab. 2. Tyto hodnoty byly získané měřením na stroji s výpočetními prostředky uvedenými na začátku kapitoly.

k [ns]	TopoART	Hypersphere TopoART	Fast TopoART
učení	158	163	126
vyhodnocení	111	114	111

Tab. 2 Hodnoty časové konstanty *k* pro jednotlivé testované sítě

Z Tab. 2 vyplývá, že při stejném počtu uzlů ve vrstvě F0 a F1 je nejméně časově náročná architektura Fast TopoART s hodnotou konstanty *k* pro učení 126 ns a pro vyhodnocení 111 ns. Druhá nejméně časově náročná architektura je TopoART, která má hodnotu *k* rovnou 158 ns pro učení a 111 ns pro vyhodnocení. Nejvyšší časovou náročnost má architektura Hypersphere TopoART s hodnotou *k* pro učení 163 ns a pro vyhodnocení 114 ns. To ovšem neznamená, že architektura Hypersphere TopoART bude poskytovat výsledky nejpomaleji. Počet uzlů vrstvy F0 bude sice na stejných datech stejný pro všechny testované sítě, ale počet uzlů vrstvy F1 se může lišit pro různé architektury, i když je využíváme pro analýzu stejných dat.

Na Obr. 14 je patrná lineární závislost spotřeby času na jeden vstupní vektor v závislosti na počtu uzlů F1 při procesu učení. Vrstva F0 obsahovala v této situaci celkem 2000 uzlů. Například v situaci pro 500 uzlů ve vrstvě F1 a 2 000 uzlů v F0 trvá proces učení jednoho vstupního vektoru neuronové síti TopoART 158 ms, Hypersphere TopoART 163 ms a Fast TopoART 126 ms. Proces vyhodnocení by v takovém případě trval neuronové síti TopoART 111 ms, Hypersphere TopoART 114 ms a FastTopoART 111 ms. Je vhodné připomenout, že v průběhu procesu učení se počet uzlů F1 neustále mění. V průběhu vyhodnocení shluků jsou již uzly v F0 i F1 neměnné.



Obr. 14 Spotřeba času v závislosti na počtu uzlů F1

Další faktor, který významně ovlivňuje čas potřebný ke zpracování dat v procesu učení i klasifikace, je čistota dat. Jak již bylo řečeno, testované ART neuronové sítě jsou schopné zpracovávat pouze prvky vstupních vektorů, které jsou v intervalu mezi ⟨0, 1⟩. Například v situaci, kdy přijde na vstup hodnota 1,6 je tato hodnota ART neuronovou sítí automaticky převedena na hodnotu 1. Hodnoty menší než nula jsou naopak převáděny na hodnotu 0. Testované ART neuronové sítě se tedy s tímto problémem jsou schopné poradit automaticky, ale má to výrazný negativní vliv na jejich výkon a to především v situaci, kdy vstupní vektor obsahuje velké množství takto neupravených prvků. V extrémní situaci se proces zpracování dat může z tohoto důvodu až několikanásobně prodloužit. Proto je vhodné data vždy normalizovat například pomocí normalizace Min-Max (viz Kapitola 3.3.3) a případné extrémy očistit.

4.3 Vliv pořadí vstupních dat na proces učení

Velkou výhodou ART neuronových sítí je jejich schopnost zpracovávat v průběhu učení jednotlivé vstupní vektory samostatně. To sebou nese úsporu času i paměťového prostoru. Tento postup má i své nevýhody. Finální výsledek shlukování je ovlivněn pořadím, ve kterém jsou jí vstupní vektory prezentovány, a proto nám

neuronová síť může poskytnout několik různých výsledků shlukování pro naprosto stejná data.

Přijde-li na vstup takový vektor 1, který není možné zařadit do existující kategorie, je pro něj vytvořena nová kategorie, pro kterou tvoří prvky vstupního vektoru souřadnice prototypu. Přijde-li na vstup jiný vektor 2, který lze do této kategorie zařadit, není nutné vytvářet další kategorii. Kdyby ovšem přišel na vstup nejdříve vektor 2, jeho prvky by tvořily prototyp nové kategorie místo prvků vektoru 1. Vektor 1 by byl následně do této kategorie pouze zařazen. Tímto je v rámci procesu učení ovlivněn vznik jednotlivých kategorií a jejich rozložení, což v některých případech má vliv i na jejich počet. Jiný počet a pozice vzniklých kategorií může způsobit rozdílnou míru jejich překrytí a právě překrytí jednotlivých kategorií výrazně ovlivňuje jejich zařazení do finálních shluků.

Pokud jsou shluky dobře ohraničené a navzájem oddělené bude mít pořadí vstupních vektorů na analýzu minimální nebo žádný vliv. V situaci, kdy se hledané shluky překrývají nebo mezi nimi existuje velké množství šumu, může pořadí vstupních vektorů výsledek shlukování výrazně ovlivnit. Tento problém se dá řešit několika způsoby, ale každý z těchto způsobů má své stinné stránky a je vhodné ho využít v jiných situacích.

V první řadě je nutné zkontrolovat, zda není parametr vigilence nastavený na příliš nízkou hodnotu. Tento parametr ovlivňuje velikost kategorií, které jsou reprezentovány uzly vrstvy F1. V takovém případě je možné, že se nepřekrývají samotné shluky, ale překrývají se kategorie, kterými jsou shluky tvořeny. V tomto případě je možné parametr vigilance zvýšit na takovou hodnotu, při které nedojde k překrytí problémových kategorií. Pokud ovšem vigilance parametr zvýšíme příliš, může dojít ke zbytečnému rozdělení shluku do více kategorií než je potřeba a tedy k růstu počtu uzlů ve vrstvě F1 a případně k rozdělení samotného shluku (uzel F2). S vyšším počtem uzlů ve vrstvě F1 se lineárně zvýší doba zpracování každého dalšího záznamu. Při dalším zvyšování hodnoty vigilance může dojít k úplnému rozpadu shluku.

V situacích, kdy změna hodnoty parametru vigilance nepomůže, je možné vliv pořadí vstupních dat částečně zmírnit pomocí nastavení nižší hodnoty parametru learning rates. Tímto způsobem ovšem nelze tento vliv úplně eliminovat a v mnoha případech tento postup nemá žádný efekt. Navíc se zároveň s nižší hodnotou parametru learning rates se snižuje i schopnost neuronové sítě adaptovat.

Máme-li dostatečné informace o analyzovaných datech, můžeme vlivu pořadí vstupů i využít. Pokud je možné vstupní vektor zařadit do obou překrývajících se shluků, je prvek zařazen do toho shluku, do kterého byl zařazen některý z předchozích vektorů naposled. Připravíme si výuková data tak, že je rozdělíme na skupiny. V každé skupině budou vstupní vektory, o kterých si myslíme, že by měly patřit do jednoho shluku a pokud je to možné dáme na začátek každé skupiny takový vstupní vektor, který nelze zařadit do jiného shluku. Tímto způsobem je možné zvýšit přesnost výsledných shluků.

Pořadí vstupních vektorů mělo vliv především při analýze využití neuronových sítí TopoART a Fast TopoART. U těchto architektur je velikost kategorií V (10)

závislá na počtu rozměrů dat (počet uzlů F0). Velikost kategorií (17) neuronové sítě Hypersphere TopoART závislá na počtu rozměrů vstupních dat není. Právě tato vlastnost vede u TopoART a Fast TopoART k vysoké úrovni závislosti na pořadí vstupních vektorů.

4.4 Textová data

4.4.1 Příprava dat

Na surová textová data není možné použít shlukovací analýzu přímo, a proto je nejdříve nutné je pomocí série úprav převést do vhodného formátu. Pro dolování je běžný formát CSV a textová data se tímto způsobem dají reprezentovat tak, že je převedeme do tabulky výskytů jednotlivých slov.

Jednotlivé textové záznamy byly tvořeny hotelovými hodnoceními. Tato hotelová hodnocení byla vyplněna zákazníky a vyjadřují jejich názor na služby, které jim byly poskytnuty. Pro analýzu bylo vybráno zcela náhodně 1 500 pozitivních hodnocení a 1 500 negativních hodnocení. Celkem se tedy jednalo o 3 000 hodnocení. Výběr byl pětkrát opakován pro vytvoření pěti textových korpusů. Protože samotná data mají velmi vysoký objem, budou výsledky jednotlivých provedených kroků prezentovány na malém ilustračním vzorku dat. Tento ilustrační vzorek budou tvořit následující hodnocení:

2 Poor breakfast! Difficult to navigate to the hotel. Not enough signs.
1 Nice clean and the security is good.
2 No quilt on bed and, not a complaint just i would prefer a duvet on bed.
1 Clean modern, efficient and friendly staff who spoke English.

Záznamy obsahují na začátku číselnou hodnotu, která nás informuje o tom, zda bylo hodnocení pozitivní či negativní. Hodnota jedna značí pozitivní hodnocení a hodnota 2 hodnocení negativní. Tento surový dataset bylo nejdříve nutné očistit. Všechna velká písmena byla převedena na malá písmena, aby nedocházelo k oddělení slov jako například *Clean* a *clean*. Dále bylo nutné odstranit slova, která na analýzu nemají žádný pozitivní vliv a jejich přítomnost by způsobila zvýšení šumu. Konkrétně se jedná o zájmena, částice, předložky, spojky, cizí slova a interpunkci. Po provedení těchto úprav by ilustrační vzorek vypadal následovně:

2 poor breakfast difficult navigate hotel not enough signs
1 nice clean security is good
2 no quilt bed not complaint just would prefer duvet bed
1 clean modern efficient friendly staff spoke english

Dále se z upraveného textového korpusu izolovala jednotlivá slova a z nich byl vytvořen slovník. Tento slovník obsahoval průměrně 5 554 slov. V ilustračním případě by obsahoval následující slova: *poor, breakfast, difficult, navigate, hotel, not,*

enough, signs, nice, clean, security, is, good, no, quilt, bed, complaint, just, would prefer, duvet, modern, efficient, friendly, staff, spoke, english. Tedy celkem by ilustrační slovník obsahoval 27 slov. Na základě podobně vzniklého slovníku byla vytvořena tabulka frekvencí výskytu slov. Pro každé hodnocení byl vytvořen řádek v tabulce. Hodnoty ve sloupcích tabulky reprezentovaly počet výskytů jednotlivých slov obsažených ve slovníku pro každé hodnocení. Pro námi specifikovaný ilustrační příklad by tedy tabulka měla 4 řádky a 26 sloupců. Jak již bylo řečeno, skutečné analyzované tabulka výskytů slov obsahovaly 3 000 záznamů s průměrnou velikostí slovníku 5 554 slov (3 000 řádků a 5 554 sloupců).

Dále bylo nutné hodnoty ve vytvořené tabulce výskytů slov normalizovat na interval ⟨0, 1⟩, protože ART neuronové sítě jsou schopné zpracovat pouze hodnoty v tomto intervalu. K tomuto účelu byla využita metoda normalizace Min-Max. Většina slov obsažených ve slovníku se v rámci jednoho záznamu vyskytla jednou, ovšem v některých záznamech se mohla objevit i dvakrát či třikrát. Více než třikrát se stejné slovo v jednom záznamu objevilo pouze ve velmi malém a statisticky zanedbatelném procentu záznamů. Parametr Max metody Min-Max byl nastaven na hodnotu tři, aby zmíněné extrémní případy neměly negativní vliv na výsledek analýzy. Parametr Min byl nastaven na hodnotu nula. Tab. 3 je výslednou tabulkou výskytů slov pro náš ilustrační příklad. Protože je počet slov příliš velký na to, aby se tabulka vešla celá na šířku stránky, byl slovník rozdělen na poloviny a hodnoty jsou uvedeny pro každou polovinu slovníku zvlášť.

		normalizované počty slov - první polovina													
záznam č.	1	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,0
	2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,3	0,3	0,3	0,3	0,0
	3	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3
	4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,0	0,0
		normalizované počty slov - druhá polovina													
záznam č.	1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	-
	2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	-
	3	0,3	0,7	0,3	0,3	0,3	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,0	-
	4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,3	0,3	0,3	0,3	0,3	-

Tab. 3 Tabula výskytů slov

Výsledná tabulka výskytů slov byla následně převedena na formát CSV tak, že jednotlivé řádky tabulky byly odděleny oddělovačem řádků a hodnoty v rámci jednoho řádku byly odděleny středníky.

4.4.2 Výsledky poskytnuté TopoART neuronovou sítí

Umělá neuronová síť TopoART byla aplikována na vytvořenou tabulku frekvencí výskytu slov. Parametr learning rates byl nastaven na hodnotu 0,45 pro snížení vlivu pořadí vstupních dat na výsledek analýzy. Je nutné dávat pozor při dalším

snižování hodnoty tohoto parametru, protože zároveň se snížením vlivu pořadí vstupních dat se také snižuje schopnost sítě se nadále adaptovat. Při nízkých hodnotách parametru vigilance se nepodařilo ART síti data rozdělit do shluků, výsledný počet shluků se rovnal jedné. Při běžných hodnotách vigilance parametru se tento problém nepodařilo vyřešit. Až při velmi vysokých hodnotách vigilance se podařilo oddělit od výsledného shluku další menší shluk. Jako optimální hodnota vigilance parametru se ukázala hodnota 0,99. S touto hodnotou vigilance dosahovala TopoART neuronová síť nejlepších výsledků shlukování. Při zvolení vyšších hodnot vigilance se oddělovaly další shluky, ovšem tyto shluky obsahovali pouze velmi malý počet záznamů (obvykle jeden), a proto pro analýzu neměli žádnou statistickou hodnotu. Pokud došlo k oddělení podobných malých shluků, byly vyřazeny z další analýzy. Konečný počet shluků se tedy rovnal dvěma. Podrobné informace o vytvořených shlucích je možné vidět v Tab. 4.

	všechna hodnocení [%]	pozitivní hodnocení [%]	negativní hodnocení [%]
shluk 1	83	60	40
shluk 2	17	2	98

Tab. 4 Shluky vytvořené pomocí TopoART

První shluk obsahoval většinu záznamů. Průměrně se jednalo o 83 % ze všech testovaných záznamů. 60 % těchto záznamů bylo pozitivních a zbylých 40 % negativní. Jak již bylo řečeno, TopoART neuronová síť měla značné problémy se shlukováním analyzovaných textových dat. První shluk byl pozůstatkem těchto problémů. I když se nakonec podařilo data rozdělit do dvou shluků, většina analyzovaných záznamů byla nadále obsažena v prvním shluku a to bez ohledu na to, zda byly pozitivní či negativní.

Mnohem zajímavější byl druhý nalezený shluk. Tento shluk se podařilo oddělit až při velmi vysokých hodnotách vigilance parametru a obsahoval 17 % všech analyzovaných záznamů. Pro nás zajímavou vlastností tohoto shluku byla jeho čistota. Druhý shluk totiž obsahoval z 98 % negativní záznamy a pouze z 2 % záznamy pozitivní. Právě čistota tohoto shluku by mohla mít praktické využití pro filtrování výrazně odlišných záznamů v textových datech. Konkrétně v tomto případě by bylo možné aplikovat neuronovou síť TopoART pro filtraci negativních hodnocení. V takovém případě by neuronová síť byla schopná filtrovat až 33 % všech negativních hodnocení.

Za předpokladu, že by první shluk reprezentoval pozitivní hodnocení a druhý shluk hodnocení negativní, by celková úspěšnost shlukování záznamů mezi pozitivní a negativní dosáhla 66 %. Tato poměrně nízká úspěšnost je způsobena především prvním shlukem, který obsahoval většinu testovaných záznamů. Přestože většina záznamů obsažených v tomto shluku byla pozitivních (60 %), shluk obsahoval i velké množství negativních záznamů (40 %).

Bohužel výsledek byl výrazně závislý na pořadí vstupních dat ve fázi učení neuronové sítě. Protože jsou záznamy reprezentovány řídkými vektory (většina

prvků vektoru je nulových), výsledné shluky se výrazně překrývaly v oblasti nuly. Právě toto překrytí mělo za následek zvýšený vliv pořadí vstupů na úspěšnost shlukování. Vstupní vektory, které bylo možné zařadit do obou shluků, byly zařazeny do posledního aktivního shluku. Tento fakt měl výrazný vliv na kvalitu výsledného shlukování. Nejlepších výsledků kategorizace bylo možné dosáhnout rozdělením podobných záznamů do skupin na základě toho, zda byli pozitivní či negativní a následně neuronovou síť učit zvlášť pro každou z těchto skupin. Při upřednostnění výrazného zástupce skupiny na začátek této skupiny se v některých případech zvýšil počet záznamů obsažených v druhém shluku až na 19 %. V situaci, kdy data seřazená nebyla, neuronová síť TopoART v procesu kategorizace selhala. Takové rozdělení dat ve fázi učení sítě vyžaduje rozsáhlé znalosti o testovaných datech, které v praxi ve většině situací nemáme.

4.4.3 Výsledky poskytnuté Hypersphere TopoART neuronovou sítí

Umělá neuronová síť Hypersphere TopoART byla aplikována na vytvořenou tabulku frekvencí výskytu slov. Parametr learning rates byl opět nastaven na hodnotu 0,45 pro snížení vlivu pořadí vstupních dat na výsledek analýzy, tedy na optimální poměr stability a flexibility sítě. Při nízkých hodnotách parametru vigilance se nepodařilo ART síti data rozdělit do shluků, výsledný počet shluků se rovnal jedné. Až při vyšších hodnotách hodnotách vigilance se podařilo oddělit od výsledného shluku další menší shluk. Optimální hodnota vigilance byla v tomto případě výrazně nižší než tomu bylo u sítě TopoART, konkrétně byla stanovena hodnota 0,9 jako nejvhodnější pro shlukování analyzovaných dat. S touto hodnotou vigilance dosahovala neuronová síť Hypersphere TopoART nejlepších výsledků shlukování. Při zvolení vyšších hodnot vigilance se opět objevovaly malé statisticky zanedbatelné shluky, které byly z analýzy vyřazeny. Konečný počet shluků se rovnal dvěma, ovšem je nutné mít na paměti, že tohoto výsledku neuronová síť Hypersphere TopoART dosáhla s výrazně nižší hodnotou vigelance než neuronová síť TopoART. Podrobné informace o vytvořených shlucích je možné vidět v Tab. 5.

	všechna hodnocení [%]	pozitivní hodnocení [%]	negativní hodnocení [%]
shluk 1	81	61	39
shluk 2	19	4	96

Tab. 5 Shluky vytvořené pomocí Hypersphere TopoART

První shluk byl největší. V tomto případě se průměrně jednalo o 81 % ze všech testovaných záznamů. 61 % těchto záznamů bylo pozitivních a zbylých 39 % negativních. Tento shluk byl nepatrně menší než v případě neuronové sítě TopoART a obsahoval kvalitnější poměr pozitivních a negativních hodnocení.

Druhý shluk byl v případě neuronové sítě Hypersphere TopoART o něco méně čistý než v případě sítě TopoART. 96 % procent záznamů obsažených v tomto shluku bylo pozitivních a 4 % byla negativní. Jinými slovy, shluk dva sice obsahoval větší počet záznamů oproti síti TopoART, ale tyto záznamy navíc měli negativní

vliv na čistotu tohoto shluku. Nejde ovšem o příliš vysoký rozdíl (pouze 2 %), a proto by tento shluk bylo stále možné použít pro filtrování negativních hodnocení. V takovém případě by filtr založený na síti Hypersphere TopoART byl schopen odstranit 36 % negativních hodnocení z jejich celkového počtu, ovšem nelze zanedbat a i počet pozitivních hodnocení, které by tímto způsobem byly odfiltrovány. Konkrétně by šlo o 7 % pozitivních hodnocení z jejich celkového počtu.

Opět za předpokladu, že by první shluk reprezentoval pozitivní hodnocení a druhý shluk hodnocení negativní, celková úspěšnost rozdělení záznamů mezi pozitivní a negativní by dosáhla 68 %. Síť Hypersphere TopoART tedy dosáhla o 2 % vyšší úspěšnosti kategorizace než síť TopoART. Tento rozdíl je způsoben především větším počtem záznamů obsažených v druhém shluku, který se v tomto případě podařilo lépe oddělit.

Výsledek byl opět částečně závislý na pořadí vstupních dat ve fázi učení neuronové sítě. Pořadí mělo negativní vliv především na čistotu druhého shluku, ale srovnání s neuronovou sítí TopoART byl výsledek shlukování výrazně stabilnější. Díky této vyšší úrovni stability je možné použít síť Hypersphere TopoART i na data, o kterých toho příliš mnoho nevíme a s takovými daty se v praxi setkáváme mnohem častěji.

4.4.4 Výsledky poskytnuté Fast TopoART neuronovou sítí

Poslední sítí, kterou jsme testovali na vytvořené textové datasety byla neuronová síť Fast TopoART. Tato neuronová síť byla opět testována s parametrem learning rates 0,45. Bohužel tato architektura neuronové sítě v procesu shlukování textových dat značně selhávala.

Při nižších hodnotách vigilance parametru vznikal jeden velký shluk, který obsahoval všechny testované záznamy. Při zvyšování hodnoty vigilance se situace neměnila, než jsme dosáhli hodnoty 0,99. Při této hodnotě vigilance parametru se nám podařilo oddělit několik shluků. Počet těchto shluků značně kolísal v závislosti na pořadí vstupních dat v průběhu učení.

V situaci, kdy data nebyla žádným způsobem seřazená, se podařilo oddělit průměrně 5 shluků. Jeden z těchto shluků obsahoval většinu analyzovaných záznamů (99 %) a ostatní shluky obsahovali pouze méně než jedno procento záznamů. Při hodnotách vigilance parametru vyšších než 0,99 se pouze zvyšoval počet malých shluků. Po seřazení vstupních dat do skupin na pozitivní a negativní se počet nalezených shluků zmenšil na 2 shluky, ovšem jeden z těchto shluků opět obsahoval více než 99 % analyzovaných záznamů a druhý shluk tedy obsahoval méně než jedno procento analyzovaných záznamů. Takový výsledek shlukování je reálně nepoužitelný.

Z důvodu neúspěšnosti neuronové sítě Fast TopoART, bylo otestováno, zda je možné, že negativní výsledek shlukování může být způsoben volbou parametru learning rates, ovšem i ve všech předchozích testovacích případech byly architektury vždy testované s hodnotou parametru learning rates 0,45, pro snížení vlivu pořadí a zvýšení stability sítě je vhodné parametr learning rates snížit, a proto byl proveden experiment pro hodnotu 0,2. Změna parametru ovšem neměla na výsle-

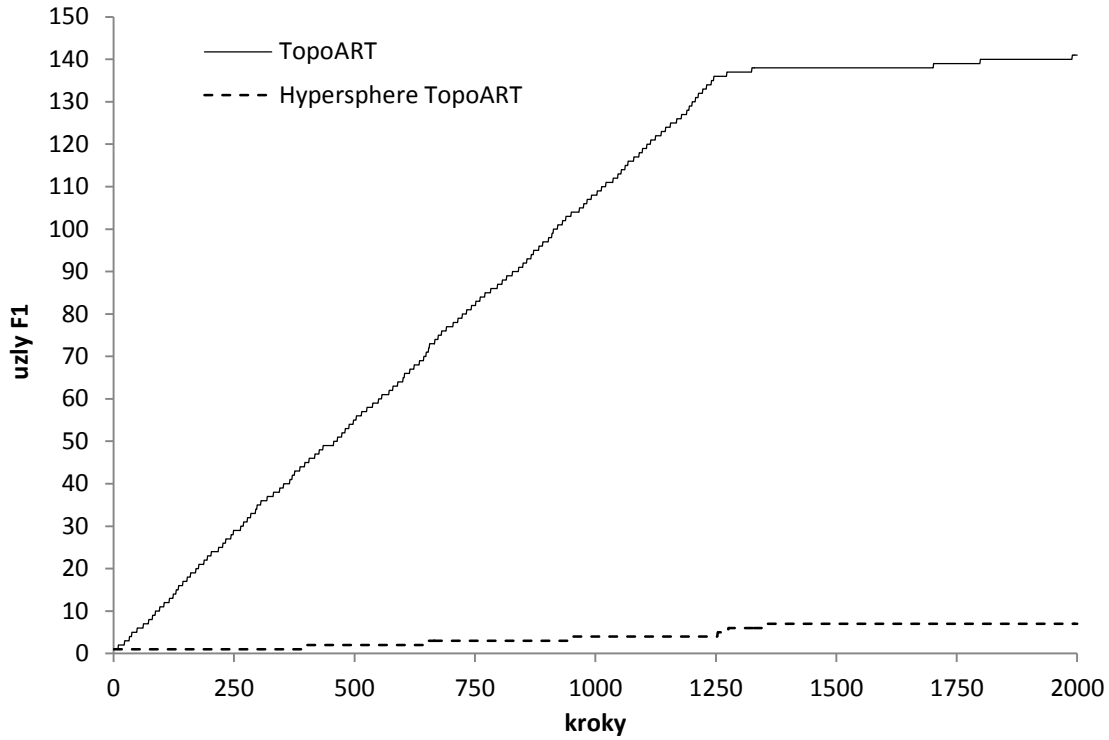
dek shlukování žádný vliv a opět jsme při hodnotě vigilance parametru 0,99 získali jeden velký shluk a několik dalších shluků, které obsahovaly méně než jedno procento záznamů.

Shlukování bylo neúspěšné kvůli povaze textových dat, která jsou reprezentována řídkými vektory. Řídké vektory obsahují malé množství nenulových hodnot, a proto se výsledné shluky často překrývají v oblasti nuly. Ze stejného důvodu se tvořil jeden velký shluk při pokusech s neuronovými sítěmi TopoART a Hypersphere TopoART. TopoART byla navíc z důvodu vysokého počtu rozměrů dat výrazně ovlivněna pořadím, ve kterém tato data přicházela na vstup. V případě Fast TopoART se ovšem nepodařilo oddělit druhý shluk obsahující negativní záznamy, což bylo způsobeno především její vyšší citlivostí na šum a rychlejším a méně přesným procesem adaptace vah v průběhu učení.

4.4.5 Spotřeba času na textových datech

Pouze dvěma testovaným ART neuronovým sítím se podařilo data úspěšně rozdělit do shluků. Konkrétně neuronové síti TopoART se podařilo testovaná data rozdělit do shluků při hodnotě vigilance parametru 0,99 a síti Hypersphere TopoART se data podařilo rozdělit již při hodnotě vigilance parametru 0,9. Pomocí sítě Fast TopoART se data do shluků rozdělit nepodařilo. Rozdílná hodnota vigilance parametru, pro který se data podařilo rozdělit do oddělených shluků, vedla k rozdílné velikosti kategorií, jejichž prototypy jsou reprezentovány pomocí uzlů ve vrstvě F1. Protože síti Hypersphere TopoART se podařilo rozdělit do shluků data již při hodnotě vigilance parametru 0,9, stačil jí k reprezentaci shluků menší počet větších kategorií. Naopak sít TopoART potřebovala větší počet menších kategorií.

Protože počet uzlů ve vrstvě F1 má přímý lineární vliv na spotřebu času, je patrné, že neuronová síť Hypersphere TopoART byla schopná data zpracovat rychleji než síť TopoART. Vývoj počtu uzlů v průběhu procesu učení je možné sledovat na Obr. 15.



Obr. 15 Vývoj uzlů F1 v závislosti na počtu vykonaných kroků učení

Je patrné, že počet uzlů ve vrstvě F1 stoupal v případě neuronové sítě TopoART výrazně rychleji než u neuronové sítě Hypersphere TopoART. Zatímco neuronová síť Hypersphere TopoART využívala na konci procesu učení 7 uzlů F1, neuronová sít TopoART využívala 143 uzlů F1. Počet těchto uzlů se ustálil zhruba po 1500 výukových krocích. Přestože neuronová síť Hypersphere TopoART má vyšší časový koeficient k než síť TopoART, byla schopná data zpracovat výrazně rychleji. Neuronová síť TopoART s časovým koeficientem pro učení k = 158 ns by se na každém dalším záznamu učila zhruba 125 ms a s koeficientem k = 111 ns pro vyhodnocení by každý vstup vyhodnotila za 88 ms. Neuronová síť Hypersphere TopoART má koeficient pro učení k = 163 ns a koeficient pro vyhodnocení k = 114 ns. V tomto případě by se tato síť naučila každý další vstup zhruba za 6 ms a vyhodnocovala by jednotlivé vstupy zhruba za 4 ms.

Tento případ jasně ilustruje situaci, ve které neuronová síť, přestože má vyšší hodnotu koeficientu učení, je schopná data zpracovat výrazně rychleji, protože s nižší hodnotou vigilance parametru je schopná data rozdělit stejně efektivně, ovšem s výrazně nižším počtem uzlů ve vrstvě F1. Tyto výsledky ovšem nelze zobecnit na všechny typy dat. Testovaná textová data bylo snazší rozdělit do kategorií tvaru hyperelipsoidu, a proto fungovala efektivněji neuronová síť Hypersphere TopoART. Na datech, která by bylo snazší rozdělit do kategorií tvaru hyperkvádru, by pravděpodobně lépe fungovala síť TopoART. Pro vyhodnocení velkých objemů

dat podobných testovaným textovým datům bych rozhodně doporučil využít neuronovou síť Hypersphere TopoART.

4.4.6 Shlukování 10 000 textových záznamů

Nejlepších výsledků na textových datech dosáhla neuronová síť Hypersphere TopoART, a proto byla otestována pro vyšší objemy dat. Neuronová síť TopoART byla velmi závislá na pořadí vstupů a měla výrazně vyšší spotřebu času. Neuronové síti Fast TopoART se nepodařilo textová data rozdělit do shluků.

Pro účely tohoto experimentu bylo vytvořeno znovu pět datasetů stejným způsobem, jako je popsán v kapitole 4.4.1. Jediný rozdíl byl v počtu hodnocení, které byly zahrnuty do jednoho textového korpusu. V tomto případě každý textový korpus obsahoval 5 000 negativních hodnocení a 5 000 pozitivních hodnocení. Celkově tedy textový korpus obsahoval 10 000 hodnocení a velikost slovníku dosahovala průměrně 10 703 slov. Výsledné tabulky výskytů slov tedy měly 10 000 řádků a průměrně 10 703 sloupců.

Neuronová síť Hypersphere TopoART byla opět schopná data rozdělit při hodnotě vigilance parametru 0,9 a dosáhla velmi podobného výsledku jako při předchozím experimentu, ovšem v rámci jednotlivých testů byly výsledky stabilnější v tom smyslu, že poměr záznamů v jednotlivých nalezených shlucích pro jednotlivé expermenty mnohem méně kolísal. Dosažené výsledky je možné vidět v Tab. 6.

	všechna hodnocení [%]	pozitivní hodnocení [%]	negativní hodnocení [%]
shluk 1	79	62	38
shluk 2	21	6	94

Tab. 6 Vytvořené shluky

Největší nalezený shluk obsahoval celkem 79 % procent záznamů. Poměr mezi pozitivními a negativními hodnoceními obsaženými v tomto shluku se nepatrně zlepšil. V této situaci obsahoval první shluk 62 % procent pozitivních hodnocení a 38 % negativních hodnocení. Zajímavější byl opět druhý shluk. Celkové procento záznamů obsažených v tomto shluku stouplo na 21 %, ale čistota nalezeného shluku nepatrně poklesla. V této situaci obsahoval shluk dva 94 % negativních záznamů a 6 % záznamů pozitivních. Pokles čistoty tohoto shluku byl pravděpodobně způsoben větším procentem záznamů, které do tohoto shluku byly zařazeny. I přes nepatrně nižší čistotu menšího shluku neuronová síť Hypersphere TopoART dosáhla lepšího celkového výsledku než v předchozím případě. Za předpokladu, že by první shluk reprezentoval pozitivní hodnocení a druhý shluk hodnocení negativní, celková úspěšnost kategorizace záznamů mezi pozitivní a negativní by dosáhla 69 %. Nižší čistota druhého shluku je kompenzována větší velikostí tohoto shluku a lepším poměrem pozitivních a negativních hodnocení v prvním shluku.

Z hlediska spotřeby času při procesu učení a vyhodnocení se situace výrazně změnila. V první řadě měl vytvořený dateset skoro dvakrát více sloupců než da-

taset vytvořený pro 3 000 hodnocení. Jak již bylo řečeno, počet prvků vstupního vektoru (jednotlivé sloupce) musí být rovný počtu uzlů ve vrstvě F0 ART neuronové sítě. Dále se výrazně zvýšil i počet kategorií, které jsou reprezentované pomocí uzlů ve vrstvě F1. Na konci procesu učení bylo těchto uzlů celkem patnáct, tedy o osm uzlů víc než bylo potřebné pro shlukování prvních testovaných datasetů (3 000 záznamů). Růst počtu těchto vzniklých kategorií byl také způsoben větším počtem prvků ve vstupním vektoru. Počet uzlů vrstvy F1 se ustálil přibližně po 5 730 výukových krocích. Neuronová síť Hypersphere TopoART má koeficient pro učení $k = 163$ ns a koeficient pro vyhodnocení $k = 114$ ns. V tomto případě by se tato síť naučila každý další vstup zhruba za 26 ms a vyhodnocovala by jednotlivé vstupy zhruba za 18 ms.

4.4.7 Shrnutí experimentů na textových datech

V této části budou shrnuty výsledky dosažené pomocí jednotlivých architektur neuronových sítí ART. Nejdříve byly provedeny testy na datasetu obsahujícím 3 000 hodnocení. Zatímco sítě TopoART a Hypersphere TopoART byly schopné data rozdělit do dvou shluků, neuronové síti Fast TopoART se to nepodařilo.

Všechny testované sítě se potýkaly s podobným problémem. Textová data byla reprezentována řídkými vektory a v mnoha rozměrech se proto překrývala v oblasti nuly. Šum obsažený v datech měl výrazný negativní vliv na proces kategorizace. Obecně se stále tvořil jeden velký shluk, který obsahoval většinu testovaných dat. Pouze dvěma testovaným architekturám neuronových sítí se podařilo testovaná textová data rozdělit. Shrnutí výsledků dosažených jednotlivými sítěmi pro jednotlivé architektury neuronových sítí můžeme vidět v Tab. 7.

	TopoART	Hypersphere TopoART	Fast TopoART
vigilance	0,99	0,9	0,99
počet shluků	2	2	1
procento záznamů v menším shluku [%]	17	19	-
procento záznamů ve větším shluku [%]	83	81	-
celková úspěšnost [%]	66	68	-
čistota menšího shluku [%]	98	96	-
čistota většího shluku [%]	60	61	-
počet uzlů F1	143	7	-
vliv pořadí vstupů	vysoký	nízký	-

Tab. 7 Shrnutí dosažených výsledků

Neuronová síť Fast TopoART nebyla schopná rozdělit textová data do více než jednoho shluku ani při velmi vysokých hodnotách vigilance parametru. Jakmile jsme překročily hodnotu vigilance parametru 0,99, začaly se od původního shluku odtrhávat malé shluky, které obsahovaly méně než jedno procento záznamů.

Z tohoto důvodu nebylo věcné tuto neuronovou síť dále hodnotit z hlediska dalších kritérií.

Mnohem lepší úrovně poskytnutých výsledků shlukování dosáhly neuronové sítě TopoART a Hypersphere TopoART. Oběma sítím se podařilo rozdělit analyzovaná textová data do dvou výsledných shluků. Větší z těchto shluků obsahoval většinu testovaných záznamů s nepatrnou převahou pozitivních hodnocení. Od tohoto velkého shluku se podařilo oddělit shluk menší s hodnotou vigilance parametru 0,9 pro neuronovou síť Hypersphere TopoART a hodnotou 0,99 pro síť TopoART. Menší shluk vytvořený neuronovou sítí TopoART obsahoval menší počet testovaných záznamů (17 %) než shluk vytvořený sítí Hypersphere TopoART (19 %), ale vykazoval vyšší čistotu. Vyšší celkové úspěšnosti dosahovala síť Hypersphere TopoART (68 %), zatímco síť TopoART dosahovalo úspěšnosti 66 %.

Díky nižší hodnotě vigilance parametru potřebné k rozdělení do shluků potřebovala síť Hypersphere TopoART daleko méně uzlů ve vrstvě F1, a proto vykazovala výrazně vyšší rychlost při procesu učení a hlavně v procesu vyhodnocení. Výsledek shlukování neuronové sítě TopoART byl výrazně ovlivněný pořadím vstupních dat, zatímco výsledky poskytnuté sítí Hypersphere TopoART byli ovlivněné pořadím pouze minimálně.

Na základě těchto informací je možné neuronovou síť TopoART doporučit pro analýzu menších objemů textových dat, o kterých máme dostatek informací. Vyšší čistota druhého shluku umožňuje přesnější identifikaci negativních hodnocení a při menších objemech zpracovávaných dat můžeme lépe kontrolovat pořadí dat při procesu učení. Dalším důvodem pro využití sítě TopoART na menší objemy dat je rychlost. Síť TopoART pracovala výrazně pomaleji než Síť Hypersphere TopoART, což by na vyšších objemech dat mohlo být neefektivní.

Neuronovou síť Hypersphere TopoART bych naopak doporučil pro analýzu vyšších objemů textových dat. Tato síť sice dosáhla menší čistoty druhého shluku, ale pracovala výrazně rychleji s vyšší celkovou úspěšností. Vliv pořadí dat v procesu učení na tuto síť byl minimální, a proto není pro analýzu nutné mít pro analýzu tak velké množství informací o datech jako pro síť TopoART.

Protože se neuronová síť Hypersphere TopoART jevila pro kategorizaci textových dat nejvhodnější ze všech testovaných možností, byla otestována na datesetu obsahujícím 10 000 textových hodnocení. Dosáhla velmi podobných výsledků jako v předchozím experimentu. Opět byla schopná od sebe oddělit dva shluky. Menší shluk obsahoval 21 % všech testovaných textových hodnocení a byl nepatrně méně čistý. Poměr záznamů obsažený v jednotlivých shlucích byl pro jednotlivé vykonané testy stabilnější. Celková úspěšnost se oproti předchozímu experimentu zvýšila na 69 %.

Celkově testovaným neuronovým sítím povedlo oddělit pouze jeden užitečný shluk s velikostí méně než 20 % testovaných záznamů. To je dělá z hlediska reálného využití pro textová data vhodné spíše pro detekci výrazně odlišných záznamů. V praxi by bylo možné využít algoritmus založený na síti TopoART a především Hypersphere TopoART například pro filtraci negativních hodnocení na we-

bovém fóru hotelu. Takový algoritmus by byl schopný vyfiltrovat více než 30 % všech negativních hodnocení.

4.5 Využití pro zpracování Big Data

Na základě experimentů popsaných v kapitolách 4.1, 4.2 a 4.4 byla vyhodnocena možnost využití těchto neuronových sítí pro analýzu velkých objemů dat (Big Data). Tato objemná data obsahující velké množství záznamů je velmi obtížné zpracovávat ověřenými postupy. Většina současných shlukovacích algoritmů potřebuje v procesu učení udržovat všechny vstupní vektory v paměti současně. To má negativní vliv na nejen na velikost paměti potřebné k tomuto procesu, ale i na samotnou spotřebu času potřebnou k vykonání analýzy.

Testované architektury neuronových sítí ART jsou schopné zpracovávat jednotlivé vstupní vektory odděleně. Každý vstupní vektor, který vstoupí do procesu analýzy, je zpracován neuronovou sítí a váhy sítě jsou adaptovány. Po provedení procesu adaptace již tento vstupní vektor není udržován v paměti. Dále bylo na základě experimentu potvrzeno, že tyto neuronové sítě jsou schopné proces učení opakovat, bez nutnosti začít proces učení znovu nebo rizika, že přijdeme o již naučené vzory. Z těchto důvodů se ART neuronové sítě zdají vhodné pro analýzu takto objemných dat.

Začněme nejdříve neuronovou sítí Fast TopoART, protože vykazuje nejnižší hodnotu koeficientu *k* pro učení i vyhodnocení. Tato síť vykazovala velkou přesnost při experimentech na málo rozměrných datech. Na hyperkvádrovém datasetu, který měl 2 000 rozměrů, začala být tato neuronová síť velmi ovlivněná pořadím vstupních vektorů v procesu učení, což mělo fatální vliv na výsledky shlukování. Použití této neuronové sítě pro vytvoření kvalitních shluků bylo možné pouze za předpokladu, že jsme měli dostatečné informace o testovaných datech a byli jsme schopni rozdělit testovaná data do skupin, na kterých byla neuronová síť učena odděleně. Získat tyto znalosti o velké objemu dat může být v mnoha případech nemožné. Při pokusech s textovými daty se ovšem ukázalo, že vysoké množství šumu obsaženého v těchto datech znemožňuje analýzu pomocí této neuronové sítě. Z hlediska využití pro analýzu Big data bych tuto neuronovou síť doporučil pouze pro analýzu velkého množství dat s malým počtem rozměrů, která neobsahují velké množství šumu. Pro taková data by byla tato neuronová síť schopná poskytovat kvalitní výsledky nejrychleji ze všech testovaných architektur ART.

Neuronová síť TopoART měla hodnotu koeficientu *k* vyšší než Fast TopoART a nižší než Hypersphere TopoART. Při experimentech s dvourozměrnými daty byla schopná pracovat se stejnou přesností jako síť Fast TopoART, ovšem nevýrazně nižší rychlostí. Při experimentech s Hyperkvádrovým datesetem dosáhla vyšší přesnosti shlukování než síť Fast TopoART, ovšem potýkala se stejnými problémy týkajících se pořadí vstupních vektorů. Při experimentech s textovými daty byla na rozdíl od sítě Fast TopoART schopná rozdělit je do shluků, i přes velké množství šumu obsaženého v těchto datech. Na rozdíl od sítě Fast TopoART je TopoART schopný oddělit shluky i v datech obsahujících šum, ale potýká se stejnými pro-

blémy s pořadím vstupních vektorů jako síť Fast TopoART. Z tohoto důvodu bych tuto neuronovou síť doporučil opět především pro analýzu velkého množství dat s malým počtem rozměrů. Tato data mohou obsahovat i malé množství šumu.

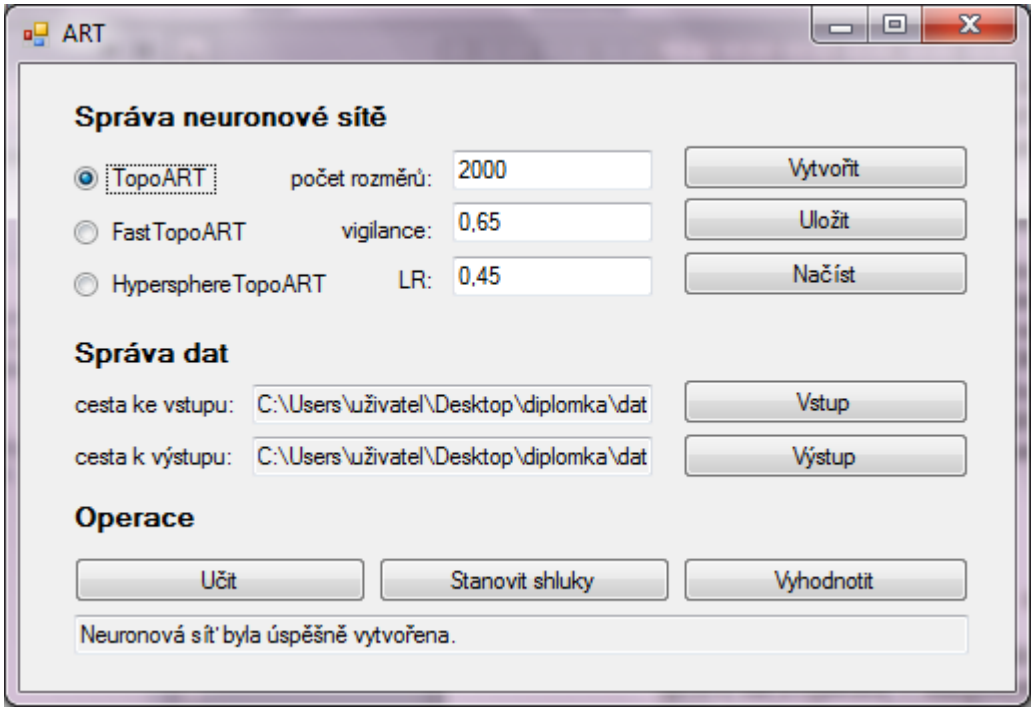
Neuronová síť Hypersphere TopoART má nejvyšší hodnotu koeficientu *k* pro učení i pro vyhodnocení. Na málo rozměrných datech vykazovala podobnou přesnost jako sítě TopoART a Fast TopoART. Na hyperkvádrovém datasetu dosáhla výrazně lepších výsledků. Byla schopná data rozdělit do kvalitních shluků i bez nutnosti data nejdříve rozdělit do skupin a dosahovala zhruba stejných výsledků shlukování bez ohledu na pořadí dat. Tyto vlastnosti se potvrdily i při analýze textových dat, kdy byla schopná rozdělit tato data do shluků při nižší hodnotě parametru vigilance, než neuronová síť TopoART, což vedlo k nižšímu počtu uzlů vrstvy F1 a nižší spotřebě času. Tato neuronová síť je vhodná pro analýzu mnohorozměrných dat. Přesnějších výpočet vzdálenosti hranic jednotlivých kategorií od jejich prototypu, který je v této neuronové síti využit, má sice negativní vliv na koeficient *k,* ovšem tato přesnost s sebou nese výrazně vyšší stabilitu při analýze vícerozměrných dat. Z těchto důvodu bych tuto neuronovou síť doporučil pro zpracování velkého počtu vstupních vektorů, které mnohou obsahovat i vysoký počet prvků.

4.6 Aplikace a grafické uživatelské rozhraní

Součástí této práce byla i implementace aplikace, která usnadňuje využití testovaných ART neuronových sítí, včetně jednoduchého grafického uživatelského rozhraní (GUI). Pro implementaci veškeré funkcionality poskytované touto aplikací byl využit programovací jazyk C#. Dále byly využity dvě externí knihovny: LibTopoART a WindowsFORMS. Knihovna LibTopoART usnadňuje implementaci ART neuronových sítí a jejich využití. Knihovna WindowsFORMS slouží k návrhu a implementaci grafických elementů, ze kterých se GUI skládá. Vytvořené GUI je možné vidět na Obr. 16.

Okno aplikace se dá rozdělit do tří základních částí: Správa neuronové sítě, Správa dat a Operace. V nejspodnější části okna se nachází uzamčený textový panel INFO, který nám poskytuje důležité informace o průběhu a výsledcích jednotlivých vykonaných uživatelských akcí.

První oblastí okna výsledné aplikace je Správa neuronové sítě. V této části jsou dostupné celkem tři uživatelské akce: vytvořit novou neuronovou síť, uložit aktivní neuronovou síť a načíst uloženou neuronovou síť ze souboru. Chceme-li vytvořit novou neuronovou síť, musíme nejdříve zaškrtnout pole, které odpovídá požadované architektuře (TopoART, Fast TopoART nebo Hypersphere TopoART).



Obr. 16 Grafické uživatelské rozhraní

Po zvolení požadované architektury je dále nutné specifikovat základní vlastnosti vytvářené neuronové sítě. První a nejdůležitější vlastností je počet rozměrů vstupních dat. Jinými slovy jde o počet prvků, které obsahuje každý zpracovávaný vstupní vektor. Na základě toho počtu bude vytvořen odpovídající počet uzlů ve vrstvě F0 neuronové sítě. Déle je nutné nastavit hodnotu parametru vigilance a parametru learning rates, který je v rámci GUI označen jako LR. Hodnoty těchto parametrů musí náležet do intervalu ⟨0, 1⟩. Je-li zvolena požadovaná architektura a parametry ART neuronové sítě, je možné síť vytvořit kliknutím na tlačítko Vytvořit. Tím je aktivován následující zdrojový kód využívající myšlenek polymorfismu:

```
//zavedení proměnné t reprezentující ART neuronovu síť
//objekt třídy, ze které dědí jednotlivé architektury
ITopoART t = null;
//volání příslušných konstruktorů
if (basic.Checked)
     t = new TopoART(rozmery, 1, vigilance);
else if(fast.Checked)
     t = new Fast_TopoART(rozmery, 1, vigilance);
else if (hypersphere.Checked)
     t = new Hypersphere_TopoART(rozmery, 1, vigilance);
//nastavení parametru learning rates
t.Beta_sbm = finalLR;
```

Polymorfismus spočívá v tom, že nejdříve vytvoříme proměnnou s datovým typem reprezentujícím objekt třídy, ze které dědí všechny třídy, mezi kterými se rozhodujeme. Na základě podmínek (volba architektury) je této proměnné přiřazen objekt požadované třídy pomocí zavolání vhodného konstruktoru. V konstruktoru je nutné specifikovat počet rozměrů neuronové sítě a hodnotu parametru vigilance. Vytvořené neuronové síti je možné definovat hodnotu parametru learning rates. V rámci GUI je tato hodnota nastavena na základě obsahu odpovídajícího pole při vytvoření neuronové sítě.

Vytvořená aplikace umožňuje nejen vytvořit novou ART neuronovou síť, ale také uložit aktivní neuronovou síť pro pozdější využití. Pro tyto účely slouží tlačítko Uložit. Po kliknutí na toto tlačítko se zobrazí dialogové okno typické pro operační systém Windows, ve kterém je nutné zvolit název a umístění souboru, do kterého se aktivní neuronová síť uloží.

Pro načtení podobně uložené neuronové sítě slouží tlačítko Načíst. Po stisknutí tohoto tlačítka se opět otevře dialogové okno typické pro operační systémy Windows. V tomto okně zvolíme soubor, ve kterém je neuronová síť uložena. Před načtením neuronové sítě je nutné definovat, o jakou architekturu se jedná a kolik má načítaná síť rozměrů, protože tyto informace jsou nutné pro volání konstruktoru, pomocí kterého neuronovou síť načítáme. Hodnoty ostatních parametrů se doplní automaticky po načtení sítě.

Další oblast okna slouží ke správě dat. Tato oblast je tvořena dvěma dvojicemi grafických elementů. První dvojice odpovídá vstupnímu souboru s daty, která chceme analyzovat. V textovém poli je možné vidět cestu k danému souboru a pomocí tlačítka Vstup je otevřeno dialogové okno, ve kterém tento soubor vybereme. Vstupní soubor musí být ve formátu CSV odděleným středníky (viz Kapitola 3.3.1). Stejným způsobem je pomocí druhé dvojice reprezentován soubor výstupní.

Poslední oblast grafického okna slouží k obsluze jednotlivých procesů shlukovací analýzy. Je-li v aplikaci zvolena aktivní neuronová síť a vstupní soubor, je možné využít tlačítka Učit a Stanovit shluky. Po kliknutí na tlačítko Učit se aktivní neuronová síť začne učit postupně na jednotlivých řádcích vstupního souboru (CSV). Na každém z těchto řádků (vstupních vektorů) se aktivní neuronová síť iterativně učí pomocí následujícího příkazu:

```
t.Learn(vstupní_vektor);
```

Po dokončení procesu učení je vhodné přepočítat výsledné shluky a stanovit jejich počet. To provedeme pomocí tlačítka Stanovit shluky, které zavolá následující příkazy:

```
//výpočet shluků
t.ComputeClusterIDs();
//Výpis počtu shluků do textového panelu INFO
info.Text = Convert.ToString(t.ClusterNum[0]);
```

Po přepočtení shluků je jejich finální počet vypsán v textovém panelu INFO. Po provedení procesu učení a stanovení shluků je možné začít vstupní data vyhodnocovat pomocí tlačítka Vyhodnotit, které je provedeno iterativním opakováním následujících příkazů:

```
//vyhodnocení vstupního vektoru
F2_output[] result = t.GetBMOutput(InputArray.ToArray());
//zápis výsledku na výstup
temp = Convert.ToString(result[0].bm_cluster_ID);
output.AppendLine(temp);
```

Číselný identifikátor reprezentující výsledný shluk je přiřazen každému vstupnímu vektoru a následně je tento identifikátor zapsán na samostatný řádek výstupního souboru v takovém pořadí, v jakém vstupní vektory přišly na vstup.

5 Závěr

V současné době máme k dispozici stále rostoucí objemy dat. Pro zpracování dat s vysokým objemem může být využití nástrojů běžně využívaných pro dolování znalostí z dat velmi časově a paměťově náročné. Tento problém je možné zmírnit využitím ART neuronových sítí, které se využívají pro sekvenční shlukování. Právě ART neuronové sítě byly testovány v této práci.

V teoretické části práce byly rozebrány základní pojmy v oblasti dolování znalostí z dat a byl vysvětlen základní rozdíl mezi učením s učitelem a bez učitele. Dále tato část pojednávala především o algoritmech fungujících na principech učení bez učitele a jejich využití v oblasti analýzy shluků. Tyto algoritmy byly rozděleny do několika skupin podle jejich vlastností a byly popsány základní principy, na kterých jsou tyto algoritmy založeny. V této oblasti jsou často využívány algoritmy založené na umělých neuronových sítích a konkrétně využití umělých neuronových sítí založených na teorii adaptivní resonance pro generování shluků bylo hlavním tématem této práce. Z tohoto důvodu pojednával závěr teoretické části právě o umělých neuronových sítích a teorii adaptivní resonance.

V metodické části je možné se dočíst o způsobu realizace testovaných ART neuronových sítí a prostředcích, které k tomuto účelu byly využity. Dále tato část pojednávala o postupu testování vybraných neuronových sítí. Byly popsány vlastnosti všech datasetů, na kterých jsme sítě testovali, způsob jejich další přípravy pro zpracování neuronovou sítí a případně i jejich původ.

Další část této práce popisovala průběh jednotlivých experimentů, dosažené výsledky a proces implementace grafického uživatelského rozhraní. Nejdříve byly provedeny experimenty, ve kterých jsme testovali vlastnosti vybraných architektur neuronových sítí na umělých dvourozměrných datech. Výsledky těchto experimentů bylo snadné prezentovat ve formě rovinných bodových grafů a na základě těchto výsledků byl definován vliv parametru vigilance na přesnost výsledných shluků a byly stanoveny základní vlastnosti testovaných neuronových sítí. Na těchto datech testované neuronové sítě dosahovaly velmi podobných výsledků. Optimálního výsledku shlukování dosáhli neuronové sítě TopoART a Fast TopoART při hodnotě vigilance parametru 0,86 a síť Hypersphere TopoART při hodnotě vigilance parametru 0,9.

V druhém experimentu byly neuronové sítě testované na datasetu, který obsahoval dva shluky ve tvaru hyperkvádru v prostoru s dvěma tisíci rozměry. Tento dataset byl určený k otestování schopnosti vybraných neuronových sítí analyzovat data, která jsou popsána mnoha atributy, a dále k přesnému definování času potřebného pro vedení procesu analýzy a faktorů, které na tento čas mají přímý i nepřímý vliv. Na tomto datasetu nejlépe obstála neuronová síť Hypersphere TopoART, která byla schopná tyto shluky oddělit naprosto přesně, a nebyla při tom ovlivněna pořadím vstupních vektorů. Neuronové sítě TopoART a Fast TopoART byly silně ovlivněné pořadím vstupních vektorů a byly schopné dosáhnout optimálních výsledků shlukování pouze v případě, že byly učeny na vstupních vektorech v uvedeném pořadí. Na základě provedených experimentů byl okomentován

vliv pořadí vstupních vektorů na výsledek shlukování a byly doporučeny způsoby, kterými lze tento vliv zmírnit nebo dokonce využít.

Přímý vliv na spotřebu času měl počet uzlů ve vrstvách F0 a F1 neuronové sítě. Obě hodnoty ovlivňovaly spotřebu času potřebnou k analýze lineárně. Za předpokladu stejného počtu uzlů v obou vrstvách byla data schopna analyzovat nejrychleji neuronová síť Fast TopoART. Za stejného předpokladu byla nejpomalejší síť Hypersphere TopoART, síť TopoART by byla jen nepatrně rychlejší.

Poslední experiment otestoval vybrané architektury ART neuronových sítí na reálných datech. Dataset byl vytvořen ze souboru hotelových hodnocení a obsahoval stejný počet hodnocení pozitivních a negativních. Na tomto datasetu nejlépe obstála neuronová síť Hypersphere TopoART, která byla schopná oddělit velmi čistý shluk obsahující pouze negativní hodnocení. Tato neuronová síť byla nejméně ovlivněna pořadím vstupních vektorů a byla schopná oddělit zmíněný čistý shluk při nižší hodnotě vigilance parametru než síť TopoART, což vedlo k menšímu počtu uzlů ve vrstvě F1. Protože síť Hypersphere TopoART obstála nejlépe, byla otestována i na textovém datasetu s výrazně vyšším objemem. Neuronová síť TopoART byla schopná data rozdělit až při hodnotě vigilance parametru 0,99. Výsledky byly nestabilní a silně ovlivněné pořadím vstupů. Neuronová síť Fast TopoART nebyla schopná data rozdělit do shluků z důvodu velkého množství šumu, který byl v datech obsažen. Dále byly na základě provedených experimentů testované architektury ART neuronových sítí vyhodnoceny z hlediska možnosti jejich využití pro analýzu Big data a byla uvedena doporučení o využití jednotlivých architektur k tomuto účelu.

Nakonec byla popsána implementace grafického uživatelského rozhraní, okomentován jeho vzhled a funkcionalita a vysvětlen způsob jeho použití.

6 Literatura

[1] HASTIE, T., R. TIBSHIRANI and J. FRIEDMAN. *The elements of statistical learning: data mining, inference, and prediction*. 2nd ed. New York, NY: Springer, 2009, 745 p. ISBN 9780387848587

[2] RAYLI. *History of datamining*. [Online] 2015. [cit. 17. 8. 2015] Dostupné z: http://rayli.net/blog/data/history-of-data-mining/

[3] MURTY, Narasimha and Susheela DEVI. *Pattern recognition: an algorithmic approach*. 1st. ed. London: Springer, 2011. ISBN 0857294946.

[4] STIGLER, Stephen M. Gauss and the Invention of Least Squares. *The Annals of Statistics* [online]. 1981, **9**(3): pp. 465-474 [cit. 20. 8. 2015]. DOI: 10.1214/aos/1176345451. Dostupné z: http://projecteuclid.org/euclid.aos/1176345451

[5] HOLLAND, John H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. 1st. ed. Mass.: MIT Press, 1992, 211 p. ISBN 0262581116.

[6] BOYD, Danah and Kate CRAWFORD. CRITICAL QUESTIONS FOR BIG DATA. *Information, Communication & Society* [online]. 2012, **15**(5): 662-679 [cit. 20. 8. 2015]. DOI: 10.1080/1369118X.2012.678878. Dostupné z: http://www. tandfonline.com/doi/abs/10.1080/1369118X.2012.67887

[7] HILBERT, M. *Digital Technology and Social Change*. [Online] 2015. [cit. 21. 8. 2015] Dostupné z: https://canvas.instructure.com/courses/949415

[8] BISHOP, Christopher M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006, 738 p. ISBN 03-873-1073-8.

[9] THEODORIDIS, Sergios and Konstantinos KOUTROUMBAS. *Pattern recognition*. 3rd ed. San Diego, Academic Press, 2006, 837 p. ISBN 0123695317.

[10] BENESTY, J., M. SONDHI and Y. HUANG. *Springer handbook of speech processing*. London: Springer, 2008, 1176 p. ISBN 3540491252.

[11] MOHRI, M., A. ROSTAMIZADEH and A.TALWALKAR. *Foundations of machine learning*. Cambridge: MIT Press, 2012, 414 p. ISBN 978-0-262-01825-8.

[12] EVERITT, Brian. *Cluster Analysis*. 5th ed. Chichester: Wiley, 2011, 330 p. ISBN 9780470749913.

[13] ROMESBURG, H. *Cluster analysis for researchers*. Morrisville: Lulu Press, 2004, 334 p. ISBN 1411606175.

[14] DEIB. *Tutorial*. [Online] 2015. [cit. 22. 8. 2015] Dostupné z: http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/

[15] THEODORIDIS, Sergios and Konstantinos KOUTROUMBAS. *Pattern recognition*. 4th. ed. Burlington: Elsevier, 2009, 961 p. ISBN 978-1-59749-272-0.

[16] LUGINAAH, I., K. GOREY, T. OIAMO, K. TANG, E. HOLOWATY, C. HAMM and F. WRIGHT. A geographical analysis of breast cancer clustering in southern On-

tario: generating hypotheses on environmental influences. *International Journal of Environmental Health Research* [online]. 2012, **22**(3): 232-248 [cit. 21. 8. 2015]. DOI: 10.1080/09603123.2011.634386. Dostupné z: http://www.tandfonline. com/doi/abs/10.1080/09603123.2011.634386

[17] [LEE, Ickjai. Geospatial Clustering in Data-Rich Environments. In: *Knowledge-Based Intelligent Information and Engineering Systems* [online]. Berlin: Springer, 2005, pp. 336-342 [cit. 10. 9. 2015]. DOI: 10.1007/11554028_47. ISBN 978-3-540-28897-8. Dostupné také z: http://link.springer.com/10.1007/11554028_47

[18] SRIVASTAVA, Ashok and Mehran SAHAMI. *Text mining: classification, clustering, and applications*. Boca Raton,: CRC Press, 2009, 290 p. ISBN 9781420059403.

[19] Karol Molnár. Úvod do problematiky umělých neuronových sítí. *Elektorevue.* [online]. 2000 [cit. 10. 9. 2015]. Dostupné z: http://www.elektrorevue.cz/clanky/00013/index.html

[20] VONDRÁK, Ivo. *Umělá inteligence a neuronové sítě*. 3. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2009, 139 s. ISBN 978-80-248-1981-5.

[21] Nervová soustava. *Golgihoaparat.* [online]. 2009 [cit. 10. 9. 2015]. Dostupné z: *http://golgihoaparat.blog.cz/0903*

[22] TRENZ, Oldřich. *Integrované informační systémy*. 2014, 187 s. Dostupné také z: https://akela.mendelu.cz/~xtrenz/IIS/IIS%20-%20prednasky%20-%202014_v10.pdf

[23] GALLANT, Stephen. *Neural network learning and expert systems*. Cambridge: MIT Press, 1993, 365 p. ISBN 0262071452.

[24] HAYKIN, Simon S. *Neural networks: a comprehensive foundation*. 2nd. ed. Upper Saddle River: Prentice Hall, 1999, 842 p. ISBN 0132733501

[25] MARTIN, Anthony and Peter BARTLETT. *Neural network learning: theoretical foundations.* Cambridge: Cambridge University Press, 2009. ISBN 052111862x.

[26] CARPENTER, Gail and Stephen GROSSBERG. *Adaptive resonance theory* [online]. 2009 [cit. 15. 9. 2015]. Dostupné z: http://digilib.bu.edu/ojs/index.php/trs/article/viewFile/92/91

[27] JAIN, L., B. LAZZERINI and H. UGUR. *Innovations in ART neural networks*. New York: Physica-Verlag, 2000, 258 p. ISBN 37-908-1270-6.

[28] CARPENTER, G., S. GROSSBERG, N. MARKUZON, J. REYNOLDS and D. ROSEN. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks* [online]. 1992, **3**(5): 698-713 [cit. 10. 10. 2015]. DOI: 10.1109/72.159059. Dostupné z: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=159059

[29] ZADEH, L., G. KLIR and B. YUAN. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers*. River Edge: World Scientific, 1996, 826 p. ISBN 9810224222.

[30] CARPENTER, G., S. GROSSBERG and D. ROSEN. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*. 1991, **4**(6): 759-771. DOI: 10.1016/0893-6080(91)90056-B.

[31] TSCHEREPANOW, Marko. Incremental On-line Clustering with a Topology-learning Hierarchical ART Neural Network Using Hyperspherical Categories. In: *Poster and Industry Proceedings of the Industrial Conference on Data Mining*. Fockendorf: ibai-publishing, 2012, pp. 22-34.

[32] TSCHEREPANOW, Marko. TopoART. A topology learning hierarchical ART network. In: *Proceedings of the International Conference on Artificial Neural Networks*. Berlin: Springer, 2010, pp. 157–167.

[33] TSCHEREPANOW, M., M. KORTKAMP a M. KAMMER. A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data. *Neural Networks*. 2011, **24**(8): 906-916. DOI: 10.1016/j.neunet.2011.05.009.

[34] Downloads. *LibTopoART*. [online]. 2015 [cit. 10. 5. 2015]. Dostupné z: http://www.libtopoart.eu/

[35] Clustering datasets. *Joensuu*. [online]. 2015 [cit. 17. 5. 2015]. Dostupné z: https://cs.joensuu.fi/sipu/datasets/

[36] Hodnocení. *Booking*. [online]. 2015 [cit. 5. 6. 2015]. Dostupné z: http://www.booking.com/

Přílohy

Přílohy je možné najít na DVD přiloženém v obálce na zadní straně desek.