



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Hledání a využití struktury multimediálních dat
Student:	Bc. Petr Liška
Vedoucí:	doc. Ing. RNDr. Martin Holeš, CSc.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2016/17

Pokyny pro vypracování

1. Seznamte se s metodami pro hledání struktur v multimediálních datech, zejména s metodami pro shlukování a klasifikaci, včetně metody hledání struktury dat při zahrnutí sémantického kontextu.
2. Seznamte se s datasetem Million Song Dataset (1 milion záznamů, 280 GB, a více než 50 atributů). Proveďte analýzu potřebnou k pochopení domény.
3. Zvolte způsob ukládání a manipulaci s daty při úkolech při zpracování s ohledem na velikost datasetu. Zvažte zejména paralelizaci a cloudové řešení. Proveďte takové zpracování, aby data byla připravena pro použití metodami z bodu 1.
4. Pokuste se najít struktury v těchto datech za pomoci metod z bodu 1.
5. Na základě poznatků z bodu 4 navrhnete software pro využití těchto znalostí. Diskutujte výběr vhodné implementační technologie a metodologie vývoje, včetně sbírání požadavků, sbírání zpětné vazby od uživatele, způsobů testování a způsobů nasazování. Implementujte první verzi, která bude sloužit jako proof of concept.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
ředitel katedry

V Praze dne 7. září 2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Hledání a využití struktury multimediálních dat

Bc. Petr Liška

Vedoucí práce: doc. Ing. RNDr. Martin Holeňa, CSc.

10. května 2016

Poděkování

Nejprve bych rád poděkoval svému vedoucímu za cenné rady a připomínky. Další poděkování patří výpočetnímu centru MetaCentrum, které mi v rámci programu *CESNET LM2015042* poskytlo výpočetní zdroje nutné k této práci. Na závěr bych rád poděkoval své přítelkyni, rodině a přátelům za podporu nejen v průběhu této práce, ale během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 10. května 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Petr Liška. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Liška, Petr. *Hledání a využití struktury multimediálních dat*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato práce se zabývá hledáním struktury v kolekcích písní populární hudby jednak pomocí dobře známých *self-organizing maps (SOM)*, ale také pomocí *self-organizing maps se zahrnutým sémantickým kontextem (CRSOM)*. V textu jsou popsány všechny provedené kroky v souvislosti s obecným postupem *Dobývání znalostí z databází* a nutné studie a implementace zmíněných map. Výsledkem zkoumání je koncept, který slouží k predikci oblíbenosti nové písně na základě struktury nalezené v kolekci písní reprezentovaných svými akustickými vlastnostmi.

Klíčová slova samoorganizující mapa, samoorganizující mapa se zahrnutým sémantickým kontextem, vytěžování znalostí z dat, shlukování, multimediální data, struktura dat

Abstract

This thesis focuses on searching structures in compilations of popular songs by using well known *self-organizing maps (SOM)* as well as by using *context relevant self-organizing maps (CRSOM)*. This thesis also describes all steps taken in relation to the general process of *Knowledge discovery in databases*

and the required studies and implementations of aforementioned maps. The outcome of the investigation is a concept that is used to predict popularity of a new song based on the structure that is found in a compilation of songs represented by acoustic properties.

Keywords self-organizing map, context-relevant self-organizing map, data-mining, clustering, multimedia data, structure of data

Obsah

Úvod	1
Cíl práce	2
Členění práce	2
1 Použité metody	3
1.1 Dobývání znalostí z databází (KDD)	3
1.2 Metody dataminingu	5
1.3 SOM	7
1.4 CRSOM	11
2 Návrh a implementace	19
2.1 SOM	19
2.2 CRSOM	20
2.3 Implementace experimentů	22
3 Použitá data a předzpracování	27
3.1 Úvod do hudební problematiky	27
3.2 MillionSongDataset a The Echo Nest	27
3.3 Dostupné atributy	28
3.4 Sémantické kontexty	29
3.5 Výběr dat a předzpracování	30
4 Experimenty	33
4.1 Protokol	33
4.2 Učení CRSOM	34
4.3 Testování reprezentace	41
4.4 Hledání struktur	44
5 Predikce úspěšnosti písně	49
5.1 Rešerše	49

5.2	Návrh využití	50
5.3	Proof of concept	50
	Závěr	53
	Literatura	55
	A Seznam použitých zkratk	59
	B Obsah příloženého CD	61
	C Další výsledky se sémantickým kontextem podle úspěšnosti písně	63
	D Datasetsy	65
	D.1 Interpret jako sémantika	65
	D.2 Rok vzniku skladby jako sémantika	66
	D.3 Počet přehrání jako sémantika	66
	D.4 Oblíbenost jako sémantika	66
	E Protokoly experimentů	69
	E.1 Katy Perry a Ramones	69
	E.2 Sex Pistols a The Offspring	69
	E.3 David Bowie, Elvis Presley a AC/DC	70
	E.4 Metallica, Maddona a Louis Armstrong	70
	E.5 Metallica, Maddona a Louis Armstrong náhodně	70
	E.6 Skladby z roku 1960 a 1970	71
	E.7 Skladby z roku 1970 a 2010	71
	E.8 Uživatelská historie 3 třídy	71
	E.9 Uživatelská historie 2 třídy	72
	E.10 Oblíbenost skladby	72
	E.11 Oblíbenost skladby	72
	E.12 Oblíbenost skladby	73

Seznam obrázků

1.1	Kroky KDD podle [1]	4
1.2	Architektura MLP s jednou skrytou vrstvou	7
1.3	Modré značky představují neurony, zatímco červené spoje indikují sousedství.	8
1.4	Zobrazení natrénované SOM s neurony uspořádanými do pravouhlé mřížky 5x5 pro náhodný dvoudimenzionální problém	10
1.5	Příklad sémantické mapy pro tři třídy	11
1.6	Struktura sítě CRSOM	12
1.7	Příklad aktivace neuronů pro různé fáze učení.	14
1.8	Míra aktivace po celé učení pro různé vzdálenosti od BMU	14
1.9	Dataset Iris omezený na dvě dimenze	16
1.10	Mapa CRSOM se formuje s ohledem na sémantický kontext	17
1.11	CRSOM se chová stejně jako SOM pro zafixované $\delta_i^h(t) = 1$	17
2.1	Stahování dat z API a generování datasetů	23
2.2	Zadávání úloh do MetaCentra	25
2.3	Příklad strojově generovaného komentáře	25
2.4	Vyzvedávání výsledků z MetaCentra	26
4.1	Výsledky pro učení bez normalizace	35
4.2		36
4.3	Výsledky pro $max = 1.3$	37
4.4	Výsledky pro $max = 1.0$	37
4.5	Dominantní neuron	37
4.6	Výsledky pro $max = 0.4$	38
4.7	Průměrná chyba sítě (nahore), průměrná topologická restrikce (dole)	39
4.8	Příslušnost neuronů k třídám	40
4.9	Formování sémantické mapy	40
4.10	Pozice neuronů pro první dvě dimenze	41
4.11	Katy Perry a Ramones (exp. E.1)	42

4.12	Sex Pistols a The Offspring (exp. E.2)	43
4.13	David Bowie, Elvis Presley a AC/DC (exp. E.3)	43
4.14	Metallica, Maddona a Louis Armstrong (exp. E.4)	44
4.15	Dataset s náhodnou sémantikou (exp. E.5)	44
4.16	Skladby z 1960 a 1970 (exp. E.6)	45
4.17	Skladby z 1970 a 2010 (exp. E.7)	45
4.18	Posluchačská historie 3 třídy (exp. E.8)	46
4.19	Posluchačská historie s rozdílnou sémantikou (exp. E.9)	47
4.20	Krabicový diagram pro hotttnesss	48
4.21	Oblíbenost skladby (exp. E.10)	48
5.1	Výsledná CRSOM pro experiment (E.11)	52
C.1	Výsledná mapa pro experiment E.12	63

Seznam tabulek

4.1	36
D.1	katy-ram-dataset	65
D.2	off-sexp-dataset	65
D.3	dp-ep-acdc-dataset	65
D.4	met-mad-la-dataset	65
D.5	met-mad-la-random-dataset	65
D.6	60-vs-70-dataset	66
D.7	70-vs-10-dataset	66
D.8	user-3class-dataset	66
D.9	user-2class-dataset	66
D.10	hot-fail-dataset	66
D.11	hit-normal-flop-dataset	66
D.12	hit-fail-dataset2	67

Úvod

„Data – kolekce faktů (čísel, slov, měření, pozorování, . . .), které byly převedeny do formy zpracovatelné počítačem.“

V dnešní době patří data mezi jedno z nejdůležitějších vlastnictví. Obzvláště pro firmy data představují klíčovou konkurenční výhodu, což nutí firmy zaznamenávat a ukládat co největší množství dostupných informací [2]. Nejen z tohoto důvodu se celkový objem dat každé dva roky zdvojnásobí [3]. Nicméně uložená a nezpracovaná data jsou sama o sobě bezcenná. Prává hodnota dat přichází se *znalostmi*, které data obsahují a je možné z nich vytěžit. Základní znalost, kterou můžeme z neznámých dat získat, je objevit jejich *strukturu*.

*„Struktura (z latinského *struere*, skládat, sestavovat, budovat, pořádat) označuje způsob složení, vnitřního uspořádání nějakého objektu, zejména pokud vykazuje nějaké pravidelnosti a zákonitosti.“* [4]

A právě hledání struktury v datech je hlavní zaměření této práce. Předmětem pro objevování struktury poslouží kolekce písní populární hudby, jako zástupci audio, potažmo multimediálních dat [5]. Hudba je doména, která díky své masové oblíbenosti poskytuje obrovské množství stále rostoucích dat a s tím souvisejících možností.

Struktury z dat je možné extrahovat různými způsoby. Jedním z nich je *shluková analýza*. Nejen pro shlukovou analýzu, ale i jiné metody vytěžování znalostí z dat platí, že pro maximální efektivitu je důležité zahrnout do procesu i člověka. Tím se zkombinuje kreativita a obecné znalosti člověka s obrovskými kapacitami úložišť a výpočetní silou dnešních počítačů. [6]

Těžištěm této práce se tak staly *samoorganizující mapy*, které umožňují vizualizovat vysoce dimenzionální data při zachování topologických vlastností. Kromě klasických samoorganizujících map, které vznikly v 80. letech, je také použita jejich novější varianta, která zahrnuje do učení *sémantický kontext dat*.

Cíl práce

Cílem této práce je uskutečnit všechny nutné kroky k získání znalostí z kolekcí písní populární hudby pomocí samoorganizujících map a samoorganizujících map se zahrnutým sémantickým kontextem a tyto znalosti poté využít.

Mezi nutné kroky patří mimo jiné i důkladné nastudování zmíněných metod spolu s jejich implementací, získání dat, příprava dat a vytvoření vhodného prostředí pro provádění experimentů. Využití znalostí spočívá ve vedení diskuze nad výsledky experimentů a implementace prototypu, který bude demonstrovat využitelnost výsledků.

Členění práce

Tento text dokumentuje nejdůležitější činnosti k dosažení výše uvedeného cíle v celkem pěti kapitolách. První kapitola Použité metody (kap. 1) obsahuje zavedení terminologie, obecný popis činností nutných k extrakci znalostí z dat a důkladný popis použitých metod. Kapitola druhá (kap.2) zmiňuje nastalé překážky a jejich řešení při implementaci metod a tvorby experimentálního prostředí. Další kapitola (kap. 3) představuje zdroj použitých dat spolu se základním popisem domény. Dále lze v této kapitole najít popis předzpracování zvolených atributů. Následující kapitola (kap. 4) obsahuje popis a výsledky prováděných experimentů. Závěrečná kapitola (kap. 5) obsahuje popis možného využití získaných znalostí.

Práce kromě hlavních kapitol obsahuje celkem pět příloh v kterých je možné nalézt použité zkratky, obsah přiloženého média, vedlejší výsledky, popisy datasetů a protokoly experimentů.

Použité metody

1.1 Dobývání znalostí z databází (KDD)

Dobývání znalostí z databází (Knowledge discovery in databases, KDD) představuje netriviální extrakci nových a reálně využitelných znalostí z dat [7].

Cílem KDD je získat cenné nové informace, které nebyly z neopracovaných dat explicitně viditelné a vhodným způsobem a formou pak tyto informace zobrazit, popřípadě využít v reálných systémech. [1]

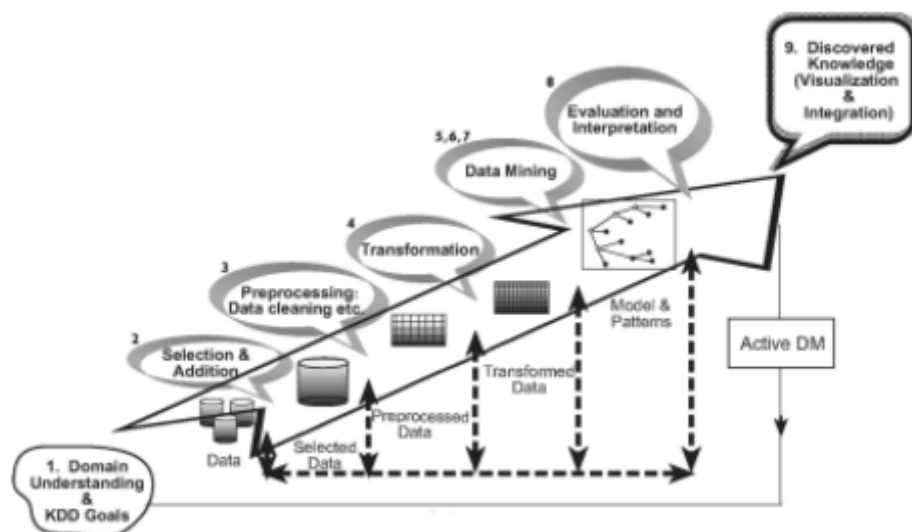
KDD se dá popsat jako proces skládající se z několika kroků. Tyto kroky se zpravidla iterativně opakují a některé vyžadují vysokou interakci uživatele. Kroky KDD, tak jak jsou popsány v [8], jsou následující:

Porozumění aplikační doméně a stanovení cílů

Částí počátečního kroku je důkladně poznat aplikační doménu. Další kritikou částí je stanovení cílů, které by KDD měly přinést koncovému uživateli. Tato rozhodnutí pak významným způsobem ovlivňují jednotlivá rozhodnutí v dalších krocích.

Výběr dat

Potom, co je specifikován cíl, je zapotřebí rozhodnout, z jakých dat se budou znalosti dobývat. Výsledkem tohoto kroku je vstupní množina dat, která velmi ovlivňuje úspěch celého KDD. Pokud například budou chybět některé významné atributy, s ohledem na stanovený cíl, celý proces pravděpodobně nebude úspěšný. Z tohoto pohledu je dobré zahrnout co nejvíce různých atributů. Na druhou stranu s velikostí dat se zvyšuje výpočetní a prostorová náročnost napříč metodami ze všech následujících kroků. Tyto protichůdné požadavky je třeba zvážit a udělat kompromis. Správnému rozhodnutí by měla pomoci iterativní povaha celého procesu.



Obrázek 1.1: Kroky KDD podle [1]

zdroj: [1]

Předzpracování

Tento krok má za cíl zvýšit kvalitu dat. Předzpracování by se mělo například vypořádat s chybějícími a odlehlými hodnotami. Nejen v tomto kroku platí „garbage in garbage out“, koncept běžný pro počítačové vědy a matematiku, který stojí na myšlence, že kvalita výstupu je určena kvalitou vstupu [9].

Transformace

V této části KDD se používají metody na transformaci atributů, jako je například diskretizace. Dále se v tomto kroku může měnit dimenzionalita dat a to buď zmenšovat metodami jako *feature selection*, *feature extraction*, nebo naopak zvětšovat odvozováním atributů nových.

Datamining

Před tímto krokem bychom měli mít k dispozici připravená data. Jak již bylo ovšem zmíněno, celý process KDD je iterativní a tento krok lze provádět s různými datovými sadami a výsledky vzájemně porovnávat.

Ze stanoveného cíle KDD jsou odvozeny kritéria pro zvolení vhodných dataminingových metod. Zvolené metody se poté aplikují na připravená data. Výsledkem tohoto kroku může být natrénovaný model nebo sada extrahovaných znalostí.

Vyhodnocení a interpretace

V této části máme již natrénované modely, popřípadě extrahované znalosti. Tyto výsledky musíme vyhodnotit a interpretovat z hlediska cílů, které byly stanoveny v prvním kroku.

Využití

Využití je posledním krokem celého KDD. Získané znalosti jsou využity v reálných systémech a natrénované modely jsou nasazeny do produčního prostředí. Tento krok se snaží přinést užitek koncovému uživateli. Úspěch tohoto kroku definuje úspěch celého KDD.

1.1.1 Datamining

Jak bylo zmíněno v předchozí kapitole, datamining je jedna z částí dobývání znalostí z databází. V první části této kapitoly jsou popsány některé třídy úloh podle [1], které jsou metody dataminingu schopné řešit. V druhé části jsou představeny základní metody spolu s nastíněním jejich využitelnosti k řešení jednotlivých typů úloh.

Klasifikace je druh problému určení kategorie, do které dané pozorování patří. Proces dataminingu tak spočívá v učení funkce, která zobrazuje datové vektory do jedné z předdefinovaných tříd. Klasifikaci je možné uplatnit například na následující problém: pracovník banky má rozhodnout, zda poskytnout žadateli půjčku. O žadateli má banka spoustu informací, stejně tak jako o dalších klientech, kteří si již v minulosti u banky peníze půjčili a poté je splatili (nebo nesplatili).

Regrese je podobná třída úloh jako klasifikace s rozdílem, že na výstupu se neobjevuje kategorie pozorování, nýbrž reálná proměnná. Jde tak o učení funkce, která mapuje datové vektory do reálné proměnné. Předpovídání atributů (jako např.: teplota, věk, cena, velikost ...) na základě historických dat je typický příklad pro regresi.

Shlukování je úloha, při které je snaha hledat a identifikovat konečný počet shluků, kterým lze popsat data. Jedná se tak o opačný přístup, než u klasifikace, kde máme třídy dat dopředu známé. Shluková analýza může například odpovědět na otázku, jaké typy zákazníků kupují jaké produkty.

1.2 Metody dataminingu

Všechny popisované metody předpokládají, že vstupní objekty jsou popsány sadou atributů a míra podobnosti těchto atributů určuje míru příslušnosti

k nějakému konceptu (například třídě). Takto popsany objekt můžeme chápat jako bod v n -dimenzionálním prostoru. Výstupem metod jsou pak modely, které mapují tento prostor a reprezentují tak hledané znalosti. Zmíněné metody se pak liší v následujících vlastnostech:

- vhodnost pro uvažovaný typ úlohy,
- do jaké míry jsou nalezené znalosti srozumitelné pro uživatele,
- síla metody – jak jsou nalezené znalosti efektivní při klasifikaci nových případů, popřípadě jak složité shluky dokáží výsledky reprezentovat,
- pro jaký typ dat jsou vhodné.

Jedna z možností, jak dělit metody, je podle způsobu učení. První způsob je *učení s učitelem*. V tomto způsobu učení je k dispozici dvojice vstupu a výstupu. Model se snaží aproximovat takto nadefinovanou funkci. Naproti tomu *učení bez učitele* informace o požadovaném výstupu nemá. V následujících odstavcích budou popsáni vybraní představitelé těchto metod. [10]

1.2.1 Metody učení s učitelem

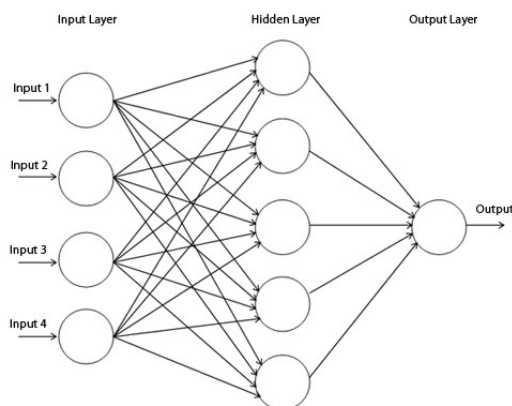
Vícevrstvá neuronová síť

Vícevrstvá neuronová síť, multilayer perceptron (MLP) je druh dopředné umělé neuronové sítě. Stavební prvky takové sítě jsou inspirovány biologickými neurony, tedy základní funkční jednotkou nervové tkáně. Umělé neurony reagují na vstupní signály a podle vnitřních vlastností signál šíří dál. Takové umělé neurony plně pospojované do orientovaného grafu pak tvoří MLP, která je využitelná v úlohách klasifikace a regrese.

Tento orientovaný graf je organizován do vrstev – zpravidla vstupní, jedna skrytá a výstupní vrstva (obr. 1.2). V dopředných sítích, jako je MLP, se signál šíří striktně jedním směrem od vstupu, přes skryté vrstvy až po výstup.

Pro konkrétní nastavení vah a jeden vstup síť reprezentuje analyticky vyjádřitelnou funkci. Protože se MLP učí s učitelem, máme k dispozici požadovaný výstup. Zpravidla je definována funkcí E , která představuje vzdálenost mezi požadovaným a spočteným výstupem. Pomocí gradientu funkce E podle vah je možné zjistit, jak upravit váhy pro zmenšení chyby. To se pak několikrát opakuje pro všechna dostupná vstupní data.

MLP se uplatňují při řešení problémů jako je rozpoznávání obrazu a řeči a další. Jednou z nevýhod takových sítí je obtížná interpretace pro koncového uživatele, obzvláště v případech, kdy jsou sítě MLP využity jako podpora lidského rozhodování.



Obrázek 1.2: Architektura MLP s jednou skrytou vrstvou

zdroj: <https://github.com/cazala/synaptic/wiki/Architect>

1.2.2 Metody učení bez učitele

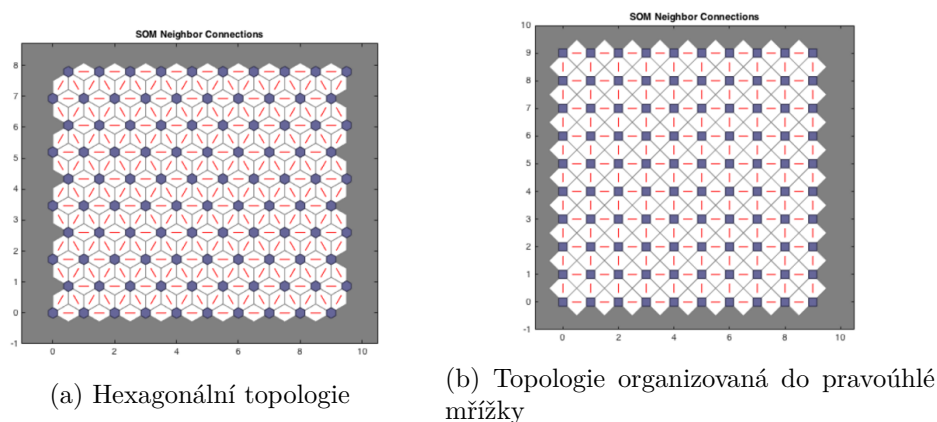
K-means

K-means je oblíbený algoritmus shlukové analýzy, kde počet shluků je předem určen parametrem k . Těchto k shluků je definováno svými *centroidy*, což jsou body ve stejném prostoru, jako vstupní vektory. Vstupní vektor náleží centroidu, kterému je nejbližší podle např. euklidovské vzdálenosti. Nalezení centroidů s optimálním shlukováním (minimální průměry shluků) je NP-těžký problém [11], proto se používá následující heuristika [12]:

1. Náhodně vybereme k centroidů,
2. vstupní vektory přiřadíme nejbližším centroidům, čímž vznikne k shluků,
3. těžiště každého shluku představuje centroid pro novou iteraci,
4. kroky 2-3 se stále opakují, dokud nedojde k ustálení.

1.3 SOM

Self organizing map (SOM), samoorganizující mapa, je umělá neuronová síť navržena Teuvo Kohonenem v 80. letech minulého století. Na rozdíl od MLP se SOM skládá z jedné vrstvy speciálně uspořádaných neuronů. Dalším rozdílem je, že se učí bez učitele a to pomocí tzv. *kompetitivní učení*. [13]



Obrázek 1.3: Modré značky představují neurony, zatímco červené spoje indikují sousedství.

1.3.1 Struktura sítě

SOM se typicky skládá z jedné vrstvy neuronů uspořádaných do pravidelné 2D mřížky¹ a každý neuron je propojen se všemi složkami vstupu. Topologie takové mřížky bývá zpravidla pravidelná a to hexagonální nebo pravoúhlá (obr. 1.3).

Reprezentace sousedství je v mapě určena podle topologie mřížky. Například v topologii organizované do pravoúhlé mřížky (obr. 1.3) má vnitřní neuron s indexem $[a; b]$ čtyři sousedy $[a + 1; b]$, $[a - 1; b]$, $[a; b + 1]$ a $[a; b - 1]$. Krajsní neurony mají logicky sousedů méně.

Stejně jako dopředná síť, SOM má ve spojích uložené váhy. Tuto skutečnost je však názornější zachycovat jako tzv.: *prototypové vektory*, které jsou uloženy u jednotlivých neuronů. Každý prototypový vektor má stejnou dimenzi jako je dimenze vstupních dat.

1.3.2 Učení

Jak již bylo zmíněno, samoorganizující se mapa se učí bez učitele a to kompetitivním učením. Základní myšlenka tohoto typu učení je následující: síť se opakovaně předkládá vstupy a váhové vektory neuronů jsou upravovány tak, aby více odpovídaly rozložení původního prostoru. Konkrétně lze učení popsat pseudokódem:

1. Inicializace vektorů.
2. Jeden vstupní vektor je předložen síti.

¹Je možné použít také 3D mřížky, nicméně v tomto textu budou použity výhradně 2D

3. Je nalezen neuron, který má se vstupem nejpodobnější váhový vektor. Pro tuto podobnost se dá použít např. euklidovská vzdálenost. Takový neuron se nazývá *vítězný neuron*, *best matching unit (BMU)*.
4. Váhový vektor každého neuronu je pak přiblížen vstupnímu vektoru. Velikost tohoto přiblížení je určena *neighborhood function* $\sigma(win, i, t)$. Závisí tedy na vzdálenosti neuronu od BMU, měřené v mřížce – čím dále jsou neurony od sebe, tím je přiblížení slabší. Dále velikost těchto změn slábnou s každou iterací učení. Celý vztah pro úpravu váhového vektoru W neuronu i je určen vztahem 1.1.
5. Znovu body 2 - 5 po N iterací.

$$\mathcal{W}_i(t+1) = \mathcal{W}_i(t) + \alpha \sigma(win, i, t)(\mathcal{X}(t) - \mathcal{W}_i(t)) \quad (1.1)$$

Kde:

- $\mathcal{X}(t)$: vstupní vektor
- $\mathcal{W}_i(t)$: váhový vektor neuronu i
- α : *learning rate* – zvolená konstanta, může být i funkcí epochy

1.3.3 Vlastnosti

SOM, stejně jako mnoho dalších dataminingových metod, předpokládá, že příslušnost tříd je definovaná podobností atributů.

Dále je chování samoorganizujících map ovlivněno počtem neuronů – bylo ukázáno, že samoroorganizující mapy s relativně malým počtem neuronů (oproti velikosti datasetu) se chovají podobně jako metoda K-means [14]. Prototypové vektory pak představují středy shluků v původním prostoru.

Naopak pro sítě s počtem neuronů srovnatelným s počtem vstupních dat, SOM učení vytváří diskrétní reprezentaci vstupního prostoru – *mapu*, která zachovává topologické vlastnosti prostoru, založené na *sousednosti* [15].

1.3.4 Využití

Jelikož se vlastnosti samoorganizujících map různí podle počtu použitých neuronů, využití se liší taktéž podle tohoto parametru.

SOM s malým počtem neuronů je možné využít tam, kde je možné použít K-means. [14]

Naopak samoorganizující mapy s velkým² počtem neuronů se používají převážně jako nástroj pro redukci dimenzionality. Toho lze využít v transformaci dat při procesu KDD (1.1), podobně, jako ostatní metody s tímto účelem [16]. Kterýkoliv vektor ze vstupního prostoru je namapován na pozici v mřížce – ta je určena pozicí neuronu, který se aktivuje pro příslušný vstupní vektor, jinými slovy je pro něj BMU.

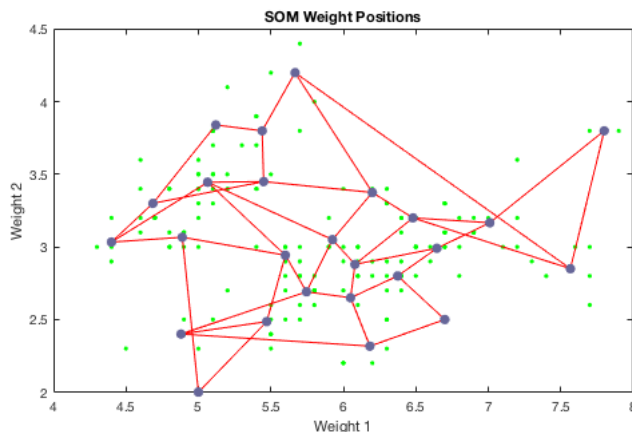
²Velký počet v tomto smyslu představuje řádově srovnatelný s počtem vstupních vektorů.

Dále je možné zkoumat naučenou *mapu*. Tato mapa je určena prototypovými vektory neuronů, které mohou sloužit jako vstupní prostor pro další shlukovou analýzu [17]. Dále je mapa, díky své nízké dimenzionalitě, vhodná pro vizualizace a *explorační analýzu*³, což je také nejrozšířenější použití samoorganizujících map.

Vizualizace map

Vizualizace je mocný nástroj pro extrakci znalostí a hypotéz z dat, převážně protože do procesu zahrnuje lidské vnímání, zkušenosti a kreativitu, které jsou při takové činnosti klíčové [18]. Bohužel tyto lidské schopnosti, stejně tak jako možnosti vizualizace, jsou značně omezené, pokud jsou data vysoce dimenzionální. Jak již ale bylo zmíněno, samoorganizující mapy poskytují dvou-dimenzionální reprezentaci dat, kterou je možné snadno vizualizovat.

Jedním způsobem, jak zobrazovat natrénovanou mapu, je zobrazit vstupní data spolu s pozicemi neuronů. Tímto přístupem ovšem získáme náhled pouze na dvě dimenze. Na obrázku 1.4 je příklad takového vykreslení pro dvou-dimenzionální problém. Zelené body představují vstupní vektory, modré značky jsou neurony a červené spoje znázorňují sousedství mezi neurony. Neurony jsou na mapě vykresleny na místě podle obrazu svého prototypového vektoru ve zobrazení ze vstupního prostoru do dvoudimenzionálního prostoru neuronů.



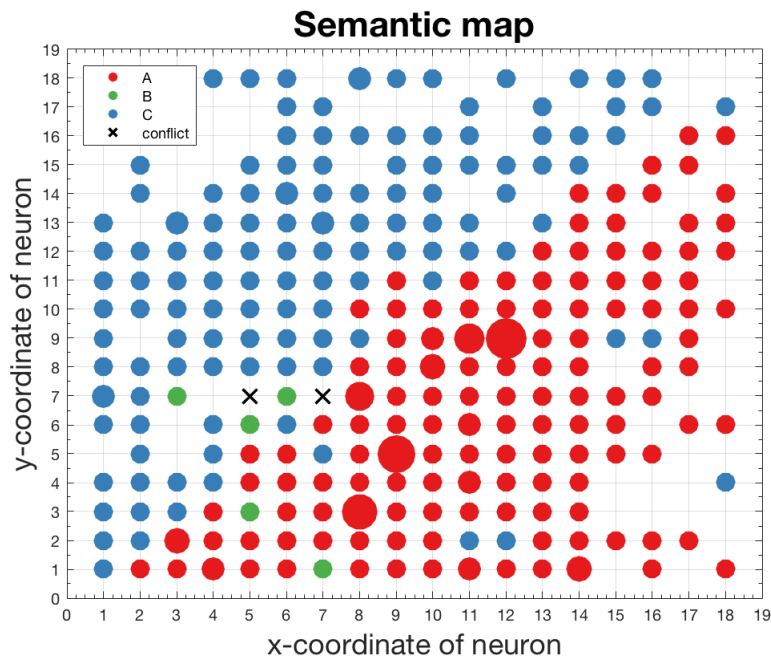
Obrázek 1.4: Zobrazení natrénované SOM s neurony uspořádanými do pravoúhlé mřížky 5x5 pro náhodný dvoudimenzionální problém

³Explorační analýza představuje metody pro průzkum dat a hledání hypotéz.

Další možností je vizualizovat mřížku spolu s dalšími informacemi (třídy), které mohou být již obsaženy ve vstupních vektorech. Pro každý vstupní vektor se spočítá jeho BMU na natrénované SOM. Takovému neuronu se pak přiřadí třída patřící onomu vstupnímu vektoru. Po aplikaci všech vstupních vektorů se provede následující vykreslení:

- neuron nemá nic přiřazeno – nevykreslí se vůbec,
- neuron má přiřazenu třídu k – vykreslí se s barvou přiřazené třídě k s velikostí úměrné počtu přiřazených vstupních vektorů,
- neuron má přiřazené dvě a více tříd – vykreslí se jako černý křížek.

Takové vykreslení se nazývá *sémantická mapa*, jejíž příklad je na obr. 1.5.



Obrázek 1.5: Příklad sémantické mapy pro tři třídy

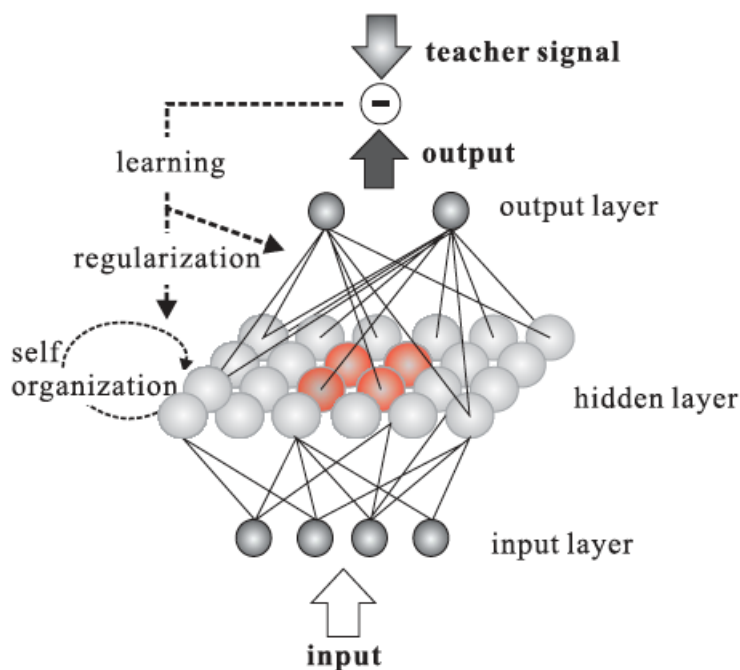
1.4 CRSOM

Context-relevant self organizing map, Samoorganizující mapa se zahrnutým sémantickým kontextem (CRSOM) je umělá neuronová síť navržena Pitoyo Hartonem [19]. Na rozdíl od klasické SOM se CRSOM učí s učitelem, z čehož

vyplývá, že k učení je nutné znát dvojice vstupní a výstupní vektor. Právě výstupní vektory určují tzv. *sémantický kontext učení*, což mohou být například třídy vstupních vektorů.

1.4.1 Struktura sítě

CRSOM se skládá ze tří vrstev neuronů. Prostřední (skrytá) vrstva je organizovaná do 2D mřížky neuronů, tzn. stejně jako u klasické samoorganizující mapy. Oproti SOM má CRSOM ještě jednu vrstvu navíc – výstupní (*kontextovou*). Každý neuron výstupní vrstvy je propojen s každým neuronem ve skryté SOM vrstvě (viz obr. 1.6).



Obrázek 1.6: Struktura sítě CRSOM

zdroj: [19]

Výstup sítě

Výstup celé sítě je složená funkce vstupního vektoru. Jinými slovy první z vrstev aplikuje svou funkci na vstupní vektor a každá další vrstva aplikuje svou funkci na výstup neuronů z předchozí vrstvy. Tento přístup je stejný jako u MLP. V následujícím odstavci budou probrány jednotlivé funkce vrstev a jejich význam.

Skrytá vrstva

První funkce, jejíž vstupem je konkrétní vstupní vektor, je $\mathcal{O}_h^i(t)$. Tato funkce představuje míru aktivace i -tého neuronu ve skryté vrstvě. $\mathcal{O}_h^i(t)$ a skládá se ze dvou multiplikatивních členů (1.2).

$$\mathcal{O}_h^i(t) = e^{-\mathcal{I}_h^i(t)} \sigma(\text{win}, i, t) \quad (1.2)$$

První z členů představuje vztah aktivace neuronu i a podobnosti vstupního a příslušného váhového vektoru – *prostorovou aktivaci*. Tato funkce (1.3) zobrazuje vzdálenost dvou vektorů do intervalu $(0; 1]$.

$$\mathcal{I}_h^i(t) = \|\mathcal{X}(t) - \mathcal{W}_i(t)\|^2 \quad (1.3)$$

Kde:

$\mathcal{X}(t)$: vstupní vektor
 $\mathcal{W}_i(t)$: váhový vektor neuronu i

Druhý člen představuje *topologickou restrikcí* v mapě neuronů (1.4). Jinými slovy jde o aktivování neuronů v závislosti na pozici v mřížce. Čím blíže je neuron k BMU, tím je tato aktivace vyšší, zároveň tato aktivace klesá s časem, díky funkci $s(t)$ popsané vztahem 1.5. V této práci byly parametry s_0 a s_{end} nastaveny na 200 a 0.01, stejně tak, jak je navrženo a empiricky otestováno v originální publikaci [19].

$$\sigma(\text{win}, i, t) = e^{-\frac{\text{dist}(\text{win}, i)}{s(t)}} \quad (1.4)$$

$$s(t) = s_0 \left(\frac{s_{end}}{s_0} \right)^{\frac{t}{t_{end}}} \quad (1.5)$$

Kde:

$\text{dist}(\text{win}, i)$: vzdálenost BMU a neuronu i , měřeno na mřížce
 s_0, s_{end} : 200, 0.01
 t, t_{end} : současná epocha, celkový počet epoch

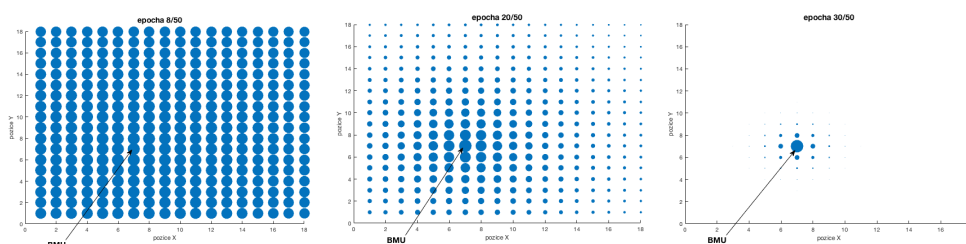
Na obrázku 1.7 jsou zobrazeny tři mřížky neuronů. Každý z obrázků představuje výstup skryté vrstvy pro stejný vstup, ale v různých fázích učení (v různých epochách). Každá diskretní pozice (x, y) tedy představuje jeden neuron a velikost značky na pozici (x, y) představuje relativní míru aktivace s tím, že BMU je v těchto případech na pozici $(7, 7)$.

Grafy se liší pouze fází učení – tedy epochou. V prvním případě je stav aktivace neuronů zobrazen na počátku učení, konkrétně v 8. epoše z celkových 50. Graf dokumentuje skutečnost, že v této fázi jsou do formování mapy zahrnuty velkou měrou všechny neurony na mapě. V dalších dvou případech,

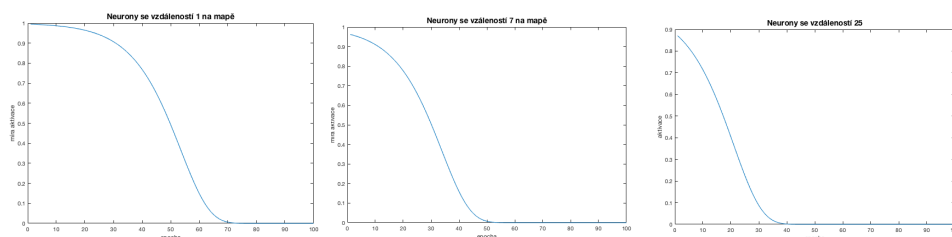
1. POUŽITÉ METODY

kteřé zobrazují stav ve 20. resp. 30. epoše, je rozpoznatelné zmenšující se okolí BMU, které je řízeno funkcí $s(t)$.

Oproti tomu na obr 1.8 je zobrazena závislost míry aktivace pro celý průběh učení pro fixní vzdálenost neuronů. V prvním grafu je vykreslena funkce σ pro neurony vzdálené 1, což se dá interpretovat jako míru topologické aktivace pro nejbližší neuron k BMU. Na zbylých grafech jsou zobrazeny průběhy pro větší vzdálenost od BMU – konkrétně 7 a 25. Z grafů je zřetelné, že vzdálené neurony od BMU se do formování mapy výrazněji zahrnují jen v počátečních fázích učení. Zahnutí neuronů pak strmě klesá a např. neurony vzdálené 25 se do formování mapy po 40. epoše zahrnují jen zanedbatelnou měrou.



Obrázek 1.7: Příklad aktivace neuronů pro různé fáze učení.



Obrázek 1.8: Míra aktivace po celé učení pro různé vzdálenosti od BMU

Výstupní vrstva

Výstup neuronů skryté vrstvy slouží jako vstup do výstupní vrstvy sítě. Výstupní vrstva funguje jako jedna běžná vrstva MLP. Tedy výstup neuronu k ve výstupní vrstvě je dán funkcí $\mathcal{O}_k(t)$ (1.6).

$$\mathcal{I}_k(t) = \sum_i v_{ik}(t) \mathcal{O}_i^h - \theta_k(t)$$

$$\mathcal{O}_k(t) = f(\mathcal{I}_k(t)) \tag{1.6}$$

Kde:

$$\theta_k(t): \text{ bias pro neuron } k$$

$$f(x) : \text{ sigmoida, } \frac{1}{1+e^{-x}}$$

1.4.2 Učení

Protože se CRSOM učí s učitelem, ke každému vstupnímu vektoru máme požadovaný výstupní vektor. Lze zavést funkci $E(t)$, která bude sloužit jako chybová funkce.

$$E(t) = \frac{1}{2} \sum_k (\mathcal{O}_k(t) - \mathcal{T}_k(t))^2 \quad (1.7)$$

Kde:

$$\begin{aligned} \mathcal{O}_k(t) &: \text{výstup neuronu } k \\ \mathcal{T}_k(t) &: \text{požadovaný výstup pro neuron } k \end{aligned}$$

Funkce $E(t)$ představuje chybu pro vstupní vektor a nastavení sítě v podobě vah. Jedním ze způsobů, jak optimalizovat takovou funkci, je *metoda nejstrmějšího sestupu*, tedy pomocí gradientu funkce $E(t)$. Změny ve výstupní vrstvě – tedy váhových vektorů v_{ik} a biasu θ_k – se řídí podle následujících vztahu 1.8, respektive 1.9

$$v_{ik}(t+1) = v_{ik}(t) - \eta_1 \frac{\partial E(t)}{\partial v_{ik}(t)} = v_{ik}(t) - \eta_1 \delta_k(t) \mathcal{O}_h^i(t) \quad (1.8)$$

$$\theta_k(t+1) = \theta_k(t) - \eta_1 \frac{\partial E(t)}{\partial \theta(t)} = \theta(t) + \eta_1 \delta_k(t) \quad (1.9)$$

$$\delta_k(t) = (\mathcal{O}_k(t) - \mathcal{T}_k(t)) \mathcal{O}_k(t) (1 - \mathcal{O}_k(t))$$

Kde:

$$\eta_1: \text{learning rate pro výstupní vrstvu}$$

Stejným způsobem se pak upravují i váhové vektory neuronů ve skryté vrstvě 1.10

$$\mathcal{W}_i(t+1) = \mathcal{W}_i(t) - \eta_2 \frac{\partial E(t)}{\partial \mathcal{W}_i(t)}$$

$$\mathcal{W}_i(t+1) = \mathcal{W}_i(t) - \eta_2 \delta_i^h(t) \sigma(\text{win}, i, t) (\mathcal{X}(t) - \mathcal{W}_i(t)) \quad (1.10)$$

$$\delta_i^h(t) = -e^{-I_i^h(t)} \left(\sum_k \delta_k(t) v_{ik}(t) \right)$$

Kde:

$$\begin{aligned} \eta_2 &: \text{learning rate pro skrytou vrstvu} \\ \mathcal{X}(t) &: \text{vstupní vektor} \\ \mathcal{W}_i(t) &: \text{prototypový vektor neuronu } i \end{aligned}$$

1.4.3 Srovnání se SOM

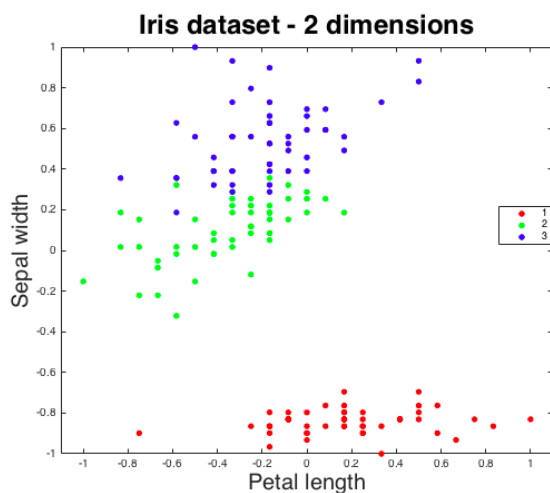
Rozdíl v učení SOM a CRSOM

Na rozdíl od vztahu, který popisuje učení prototypových vektorů v sítích typů SOM (1.1), se v učení prototypových vektorů CRSOM (1.10) navíc vyskytuje člen $\delta_i^h(t)$.

Tento člen slouží jako *regulační signál*, který je určen velikostí chyby na výstupu. Slouží tak jako zpětná vazba z kontextové sítě, která řídí formování skryté vrstvy podle zadaného kontextu.

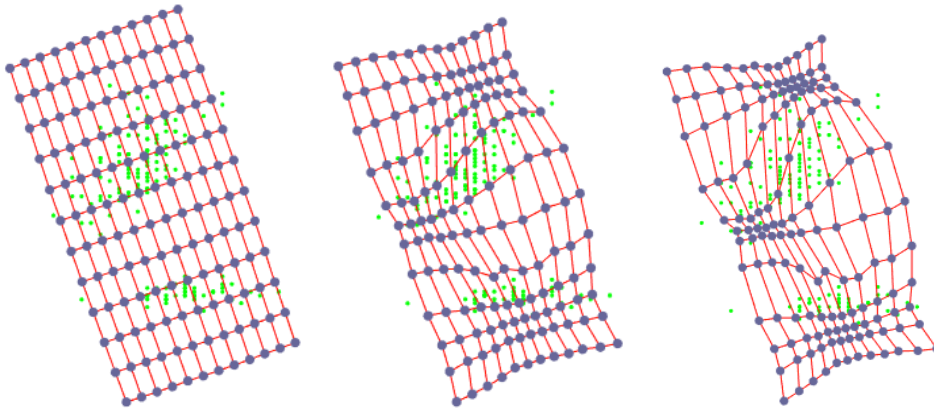
Pouze v případě $\delta_i^h(t) > 0$, je neuron i přiblížen ke vstupnímu vektoru. V případě $\delta_i^h(t) < 0$ je naopak oddálen. Z toho vyplývá, že ke vstupnímu vektoru jsou přiblíženy pouze prototypové vektory neuronů, které přispívají ke snížení chyby, ostatní jsou naopak pokutovány oddálením. To má za důsledek rozdílné chování oproti SOM, kde mapa je formována pouze na základě vstupního prostoru, zatímco u CRSOM je mapa formována na základě vstupního prostoru a dodaného semantického kontextu.

Formování mapy CRSOM je ukázáno na obr. 1.10. Vstupní data tvoří dataset Iris [20] omezený na dvě dimenze 1.9. Toto omezení umožní vizualizovat mapu a vstupní data na jednom grafu bez ztráty informace. Ze série obrázků (1.10) je vidět, že formování respektuje sémantický kontext, kterým jsou v tomto případě jednotlivé třídy datasetu Iris. Mapa vytváří shluky, které odpovídají rozložení tříd ve vstupním prostoru.

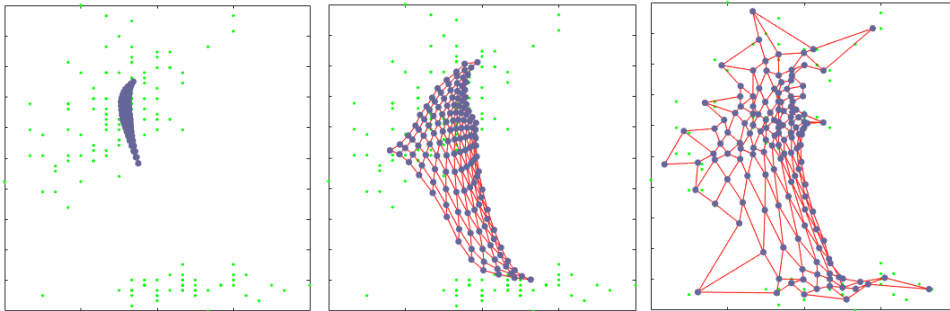


Obrázek 1.9: Dataset Iris omezený na dvě dimenze

V případě, že $\delta_i^h(t)$ je zafixováno na 1, CRSOM simuluje chování SOM: prototypové vektory jsou ke vstupnímu vektoru přiblíženy vždy, bez ohledu na sémantický kontext a výstup sítě. Simulace klasické samoorganizující mapy pomocí CRSOM je doložen na obr. 1.11.



Obrázek 1.10: Mapa CRSOM se formuje s ohledem na sémantický kontext

Obrázek 1.11: CRSOM se chová stejně jako SOM pro zafixované $\delta_i^h(t) = 1$

Rozdíl v interpretaci map SOM a CRSOM

Samoorganizující mapy jsou schopné vysoce dimenzionální prostor mapovat do 2D a zachovat při tom některé topologické vlastnosti vstupního prostoru. Pokud máme k datům nějaký sémantický kontext (třídy) a chceme je nějakým způsobem využít pro formování mapy, není jiná možnost, než využít třídy jako další atributy. Naopak CRSOM s těmito informacemi zachází explicitně a pomocí těchto tříd prostřednictvím regulačního signálu řídí formování mapy podle zadaného kontextu. SOM je schopný vizualizovat strukturu vstupního prostoru, zatímco CRSOM je schopná vizualizovat strukturu problému. [21]

1.4.4 Vlastnosti a využití

CRSOM naučená na konkrétní sémantický kontext může být využita jako klasifikátor tohoto kontextu pro nová data. V takovém případě mapa, která je definovaná skrytou SOM vrstvou, představuje vedlejší produkt takového učení. Tato mapa nám dává intuitivní pochopení výkonnosti takového klasifikátoru.

Kromě toho je možné považovat mapu jako cílový produkt učení. Vzniklá

1. POUŽITÉ METODY

mapa pak může sloužit k vizualizacím klasifikačního problému pro daný sémantický kontext. Dále je možné z mapy extrahovat znalosti a strukturu v podobě vytvořených shluků, stejně jako u SOM s tím rozdílem, že výsledek CR-SOM respektuje při formování sémantický kontext učení a je možné z výsledné sémantické mapy extrahovat rozdílné znalosti [22].

Návrh a implementace

V této kapitole jsou rozebrány nutné kroky pro použití vybraných metod. Zaprvé bylo nutné zajistit implementace oněch metod a zadruhé navrhnout a implementovat platformu pro experimentování s metodami. Pro implementaci metod bylo zvoleno prostředí *Matlab*. Pro práci s experimenty byly zvoleny *Ruby*, *MongoDb*, *Git* a *MetaCentrum*. V této kapitole jsou popsány důvody pro zvolení právě těchto technologií a nástrojů, spolu s popisem jejich využití a vzájemného propojení.

2.1 SOM

Hotová implementace samoorganizujících map je dostupná pro mnoho platform, Matlab byl zvolen zejména kvůli existenci speciálního toolboxu⁴ pro neuronové sítě. Kromě toho Matlab nabízí širokou škálu vykreslování a jednoduchou manipulaci s maticemi. Je třeba také zmínit, že ČVUT vlastní licence na Matlab a příslušné toolboxy pro akademické použití, čímž odpadá finanční stránka rozhodování, která by v neakademickém použití hrála významnou roli. Pro samoorganizující mapy se v Matlabu používá funkce `selforgmap`⁵. Kód pro natrénování samoorganizující mapy o velikosti 10x10 stačí kód zobrazen v ukázce 1.

```
1 som = selforgmap([10 10])  
2 trained_som = train(som, inputs)
```

Ukázka 1: Vytvoření samoorganizující mapy v prostředí Matlab

⁴Toolbox je označení pro rozšíření Matlabu a jeho IDE

⁵Dokumentace je dostupná na <http://www.mathworks.com/help/nnet/ref/selforgmap.html>

2.2 CRSOM

Implementace CRSOM byla mnohem náročnější. Jedním z důvodů bylo to, že v době implementace bylo k dispozici pouze analytické vyjádření fungování sítě, bez jakékoliv vzorové implementace. Správnost sítě tak bylo možné verifikovat pouze na základě výstupů z experimentů provedených Hartonem na známých datových sadách [19].

Další překážkou bylo, že prováděné experimenty byly spouštěny pro 30000 epoch, což by znamenalo několik dní výpočtů pro první naivní implementace.

Postup pro implementaci sítě tak, jak byla navržena Hartonem, byl inspirován přístupem „*Make it work, make it right, make it fast*“ [23].

Tento přístup logicky prioritizuje vývoj software: jako první by měla být zaručena korektní implementace. Až poté je možné implementaci vylepšovat ve směrech designu, robustnosti a výkonu. Pro zachování správnosti software je nutné zavést regresní testování, které zaručí pokračující korektnost i po změnách v rámci „*Make it right*“ a „*make it fast*“.

2.2.1 Make it work

Jak bylo zmíněno, první naivní implementace sítě není možné verifikovat oproti Hartonovým experimentům, na druhou stranu je z definice sítě a z dostupných publikací (např.: [19]) zřejmé, jak se bude mapa formovat pro jednoduché příklady. Takovým jednoduchým příkladem je například dobře známý dataset Iris. Tento dataset byl navíc omezen na dvě dimenze, což poskytuje možnost vizualizovat neurony, určené prototypovými vektory, spolu se vstupními daty v jednom vykreslení. Vizualizace formování se ukázala jako velmi platná pro pochopení a ladění fungování sítě. Kromě zobrazování neuronů na mapě spolu se vstupními daty byly ukládány a zanašeny do grafů další hodnoty významné pro průběh učení. Mezi takové hodnoty patří například hodnota funkce $E(t)$, průměrné výstupy neuronů ze skryté vrstvy, průměrná topologická restrikce a další.

Další překážkou byla vysoká citlivost metody na nastavení parametrů učení [19]. Slabé výsledky tak nemusí vždy znamenat chybu v implementaci, nýbrž například nevhodné nastavení učících parametrů. Tomuto problému bylo čeleno pomocí prohledávání prostoru parametrů a kontrolování všech výsledků, alespoň z počátku verifikace. Pomocí těchto přístupů se podařilo naimplementovat CRSOM, která dávala srovnatelné výsledky na stejných datasetech podle originálních publikací [19].

2.2.2 Make it right and make it fast

První verze sítě byla sice funkční, trpěla ale několika prohřešky jak z výkonového hlediska, tak z designového. Ještě před jakýmkoliv úpravami bylo nutné vytvořit sadu regresních testů, které budou zajišťovat stálou správnost sítě.

Toho bylo docíleno odebráním všech nedeterministických nastavení a uložení stavu sítě po několika epochách učení (*expected net*) pro daný testovací dataset. Regresní testování upravené sítě pak spočívalo ve spuštění učení se stejným nastavením a pro stejný dataset jako pro síť *expected net*. Za předpokladu deterministického učení, naučená síť musí mít stejné hodnoty prototypových vektorů jako *expected net*.

Výkon sítě se měřil pomocí učení na datasetu, který byl výrazně větší, než dataset určený pro testování. K určení problémových míst implementace z hlediska výkonu byl výhradně používán Matlab profiler⁶, který je vestavěný do IDE. Zlepšování sítě bylo provádělo iteračně, nejvýraznější pokroky ve výkonu byly dosaženy následujícími upravami:

Vektorizace

Matlab je optimalizován pro práci s vektory a maticemi, tím pádem kód, který je *vektorizovaný* (ukázka 3) je mnohem rychlejší [24], než ten vyvořený pomocí *smyček* (ukázka 2).

```

1 delta_h = zeros(NEURONS,1);
2 for i = 1:NEURONS
3     s~= sum(delta_k.* cn.IW{1}(i));
4     delta_h(i) = first_part(i) * s;
5 end

```

Ukázka 2: Kód zapsaný pomocí for smyčky

```

1 s~= sum((repmat(delta_k', NEURONS, 1).*(cn.IW{1}'))')';
2 delta_h = first_part .* s;

```

Ukázka 3: Vektorizovaný kód

Prealokace

Pokud je předem známá velikost požadované matice, je vždy rychlejší naalokovat celkový kus paměti dopředu, než požadovanou iterativně navyšovat [25].

Funkce adapt

První verze používaly oddělenou kontextovou síť vytvořenou pomocí funkce `network`. Tato síť se pak doučovala pomocí funkce `adapt`. Tento přístup však

⁶http://www.mathworks.com/help/matlab/matlab_prog/profiling-for-improving-performance.html

není vhodný pro cyklické opakování, zejména kvůli znovuvytváření všech obecných struktur s informacemi o učení. Pozdější verze sítě již tento přístup vůbec nevyužívaly.

2.2.3 Výsledná síť

Implementovaná síť je dostupná ve veřejném repozitáři⁷ na GitHubu. V příložené dokumentaci je popis vhodného stažení kódu (v repozitáři jsou také obsaženy datasety a výsledky experimentů). Dále je v dokumentaci popsáno základní použití sítě spolu s možnými vykresleními výsledků.

2.3 Implementace experimentů

Celý proces implementace a experimentování na cílových datech přineslo několik problémů, kterým bylo nutné čelit. Použitá data byla dostupná přes veřejné API a velikost jednoho surového záznamu byl poměrně velký (stovky Kb). Proces implementace si vynutil současnou existenci několika různých verzí sítě s rozdílným nastavením parametrů. Z důvodu dlouhé prodlevy mezi změnou a výsledkem bylo nutné efektivně zaznamenávat důvod konkrétních změn a jejich kontext. Samotné experimenty pak vyžadovaly časté spouštění trénování pro různé datové sady a různé nastavení parametrů učení. Výsledky experimentů pak bylo nutné jednoduše vyhodnocovat, třídít a ukládat.

2.3.1 Tvorba datasetů

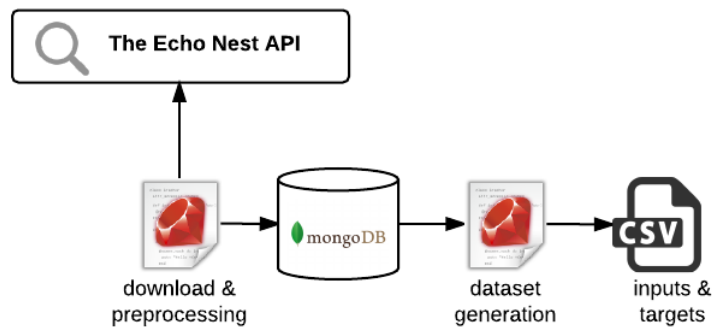
Je třeba zmínit, že všechny vytvořené datové sady pocházely z jednoho zdroje. Z důvodu poměrně objemných záznamů bylo vhodné tato data cachovat do lokální databáze. Pro tento účel byla zvolena MongoDB databáze. Samotné stahování dat z API a generování datasetů bylo docíleno skripty v programovacím jazyce Ruby. V této sekci budou představeny tyto klíčové technologie spolu s odůvodněním jejich výběru. Spolupráci technologií při stahování dat z API a následné generování datasetů je pak zobrazena na obr. 2.4.

MongoDb

MongoDb je dokumentově orientovaná databáze, tedy patří mezi tzv. *NoSQL* databáze. MongoDB umožňuje ukládat data bez dopředu známého schématu, což byl nejvýznamější důvod pro použití právě takové databáze.

Data v MongoDb jsou interně uložena a navenek prezentována ve formátu BSON, což je specifikace a knihovna pro převod JSONu do binární podoby. Data, která poskytuje The Echo Nest prostřednictvím webového API, jsou taktéž ve formátu JSON. Tato skutečnost dělá dokumentovou databázi mnohem lepším kandidátem, než by tomu byly tradiční relační databáze. Navíc

⁷Dostupný na adrese <https://github.com/liskape/crsom>



Obrázek 2.1: Stahování dat z API a generování datasetů

pro využití databáze jako lokální cache jsou nepotřebné další vlastnosti, které nabízejí relační databáze, jako jsou transakční zpracování, integritní omezení a další.

Ruby

Pro ukládání dat do MongoDB a následné generování datasetů ve formátu csv by zvolen jazyk Ruby. Ruby je interpretovaný skriptovací jazyk, který vznikl v roce 1995 v Japonsku. Jazyk Ruby byl zvolen pro svou jednoduchou syntax, možnost krátkého zápisu kódu a širokou dostupnost *gemů*⁸. Příkladem použitého gemu je Mongoid, který slouží jako ODM⁹ framework pro databázi MongoDB. Jednou ze slabých stránek jazyka Ruby je výkon [26], alespoň ve srovnání s kompilovanými jazyky jako je např C++. Tato nevýhoda je ovšem pro toto konkrétní použití nevýznamná, naopak snadný zápis kódu a dostupnost gemů znamená obrovskou časovou úsporu a snadnou modifikaci kódu.

2.3.2 Framework na experimenty

Jak již bylo zmíněno, velké množství experimentů pro různé verze kódu a konfigurace metod přineslo spousty požadavků. Mezi požadavky mimo jiné patřilo například jednoduché nasazování kódu, logování, paralelní pouštění experimentů, uchovávání výsledků a další. Pro naplnění těchto požadavků byly zvoleny dvě hlavní technologie: výpočetní centrum *MetaCentrum* a verzovací nástroj *Git*.

⁸Gem je označení pro balíček kódu v jazyce Ruby

⁹Object-Document-Mapper – mapování mezi dokumenty v databázi a ruby objekty.

MetaCentrum

Metacentrum je projekt Cesnetu, který v České republice zastřešuje většinu aktivit souvisejících se superpočítáním. Má k dispozici řadu výpočetních clusterů, které může mít k dispozici prakticky každý uživatel z akademického prostředí. Na strojích MetaCentra běží unixové operační systémy, které je možné ovládat vzdáleně z příkazového řádku. MetaCentrum obsahuje dva druhy uzlů. Jedním z nich jsou tzv. *čelní uzly*. Na čelní uzly je možné se přihlásit a zadávat úlohy spolu s požadavky na zdroje v podobě úložiště, paměti, CPU a další. Druhým typem jsou *výpočetní uzly*, které jsou vytěžovány plánovacím systémem *Torque*. Torque dohlíží na efektivní a férové využívání zdrojů uživateli, a plánuje, kdy kdo dostane vyžádané stroje k dispozici. [27]

Git

Git je distribuovaný systém správy verzí. Git umožňuje snadným způsobem uchovávat různé verze souboru spolu s jejich kompletní historií. Nová verze se vytváří tzv. *větvemi (branch)*. Ucelené změny v souborech se nazývají *commity*, které obsahují krátký text, který změnu komentuje.

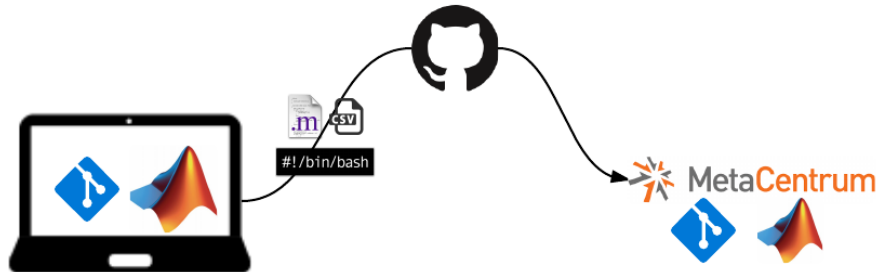
Zadávání úloh

Z důvodu vysoké náročnosti úloh na prostředky byly v drtivé většině případů všechny sítě učeny na výpočetních uzlech v MetaCentru. Pro spuštění učení sítí pro jeden dataset je nutné mít v MetaCentru k dispozici kód v matlabu, vstupní soubory a skript, který zadá úlohy s různými parametry učení. Všechna tato data byla do MetaCentra dopravena přes vzdálený Git repozitář. Spuštění jednoho experimentu tak spočívalo ve vytvoření commitu, který byl dopraven přes vzdálený repozitář do MetaCentra. V MetaCentru pak byl spuštěn skript, který zadal úlohy. Všechnu tuto práci je možné automatizovat jedním bash skriptem (2.2). Jednou z výhod takového přístupu je uchování kompletní historie experimentů spolu se smysluplnými komentáři, které lze generovat strojově. Příklad takového vygenerovaného komentáře (obr. 2.3) je část skriptu, který zadává úlohy s různými parametry učení. Dále jsou automaticky generována jména sítí – jako složenina datasetu a parametrů učení. Toto jméno je pak použito jako identifikátor úlohy pro snadnou inspekci dostupnou přes uživatelské rozhraní MetaCentra. Dále je jméno použito na pojmenování vytvářených souborů – např. logu, kde kterého se ukládají informace o současné epoše a chybě.

Vyzvedávání výsledků

Kromě logů jsou do souborového systému v MetaCentru ukládány také další soubory. Po dokončení učení je celý *workspace*¹⁰ serializován a uložen na disk.

¹⁰Workspace je množina proměnných v paměti.



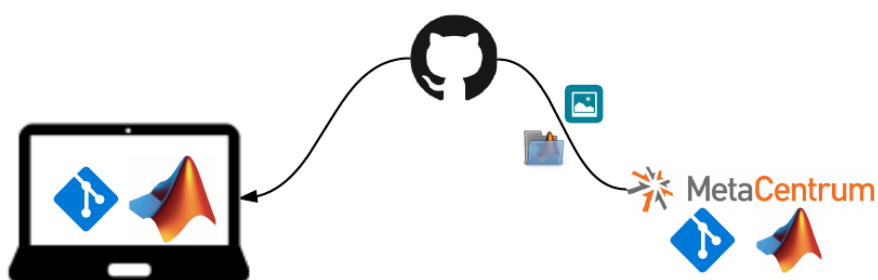
Obrázek 2.2: Zadávání úloh do MetaCentra

```
commit 9c9c6e8f481dde3096e73170d4ee0d1b0890e5f0
Author: pliska <liskape8@gmail.com>
Date: Thu Feb 11 15:51:02 2016 +0100

PROBLEMS="met_mad_la_random1 met_mad_la_random2 met_mad_la_random3"
LEARNING_RATES="0.01 0.05 0.1 0.2"
EPOCHS="500 1000 2000"
NORMALIZE="1 3 4"
MAP_SIZE="1"
```

Obrázek 2.3: Příklad strojově generovaného komentáře

Stejně tak je uložena vizualizace výsledků. Protože jsou tyto výsledky uloženy na disku, je snadné je prostřednictvím Gitu přenést zpět na lokální stroj (2.4). Zpravidla jsou nejdříve přenášeny výsledky v obrázkových podobách – slouží jako rychlé náhledy na výsledky. Po zhodnocení výsledků je možné vybrat konkrétní uložený workspace a přenést ho na lokální stroj. Na lokálním stroji lze workspace načíst a pokračovat v práci stejně, jako kdyby byla síť učena tam.



Obrázek 2.4: Vyzvedávání výsledků z MetaCentra

Použitá data a předzpracování

3.1 Úvod do hudební problematiky

Zvuk je mechanické vlnění v látkovém prostředí, které je schopno vyvolat sluchový vjem. Zvuky můžeme rozdělit na *tóny* a *hluky*. Tóny bývají označovány jako zvuky hudební, hluky jako zvuky nehudební. Tóny vznikají při pravidelném, v čase přibližně periodicky probíhajícím pohybu – kmitání. Při jejich poslechu vzniká v uchu vjem zvuku určité výšky, proto se tónů využívá v hudbě. Tóny se dále dělí na *jednoduché* – určené jednou sinusoidou (základní) a *složené* – obsahují kromě frekvence základní navíc i tzv. vyšší harmonické. Nejdůležitějšími vlastnostmi tónu jsou: *pitch* (výška), *síla* (hlasitost, loudness) a *timbre* (barva tónu).

Výška tónu je dána především frekvencí. Rozeznáváme sedm základních tónů a pět půltónů označené C, C#, D, D#, E, F, F#, G, G#, A, A#, B, kde každý z nich představuje určitou frekvenci.

Hlasitost tónu je subjektivní veličina, která se měří v jednotce **bel** [B].

Barva tónu je vlastnost, která umožňuje sluchem rozlišit dva složené tóny, které mohou mít stejnou absolutní výšku a které jsou vydávány dvěma různými zdroji zvuku. Je určena nejen počtem vyšších harmonických tónů obsažených ve složeném tónu, ale také jejich amplitudami. [28]

3.2 MillionSongDataset a The Echo Nest

MillionSongDataset je volně dostupná kolekce hudebních atributů a metadat pro více než milion skladeb populární hudby. Tento dataset je vytvořený za účelem podpory výzkumu v oblastech algoritmů pro hudební data, *vyhledávání hudebních informací*, *music information retrieval (MIR)* a další. MillionSongDataset byl vytvořen společností The Echo Nest, která se zabývá hudební analýzou a vztahy posluchačů k hudbě. MillionSongDataset slouží jako zkratka k hudebním datům, které jinak The Echo Nest poskytuje prostřednictvím svého webového API.

Při náročnosti metod používaných v této práci je zřejmé, že není možné použít celý tento dataset, který má přibližně 280 GB. Manipulace s takto velkým objemem dat je velmi komplikovaná a tak zmenšování původního datasetu není vhodné řešení. Nejen z těchto důvodů byl využit konkrétní zdroj dat a to již zmíněné API. Tímto přístupem je možné jednoduše vytvářet datasety pro předem zvolené sémantiky. Takové sémantiky mohou být například žánr hudby, období vzniku, oblíbenost, autor a další. Další výhodou, kterou přinesl tento přístup, je vyšší aktuálnost metadat, zejména těch, které jsou odvozeny na základě aktuálních trendů a sociálního vnímání jako je například oblíbenost konkrétní skladby nebo míra známosti hudebního autora.

3.3 Dostupné atributy

3.3.1 Nástroj Analyze

Analyze je nástroj pro analýzu hudebních dat, který používá společnost The Echo Nest. Tento nástroj je volně dostupný prostřednictvím webového API. Vstupem programu je digitální audio soubor (např.: mp3, m4a, wav, mpeg, flv), který je programem analyzován. Výstupní informace programu jsou poskytnuty JSON souborem (4), který popisuje strukturu nahrávky a hudební obsah včetně rytmu, výšek a tónů. Program využívá techniky pro simulování fyzického a kognitivního vnímání hudby člověkem. Konkrétní fungování programu je ale proprietární.

Pro účely této práce je nejdůležitější analýza s názvem "segments". Jeden segment představuje množinu zvukových entit, které jsou relativně neměnné v hudebních attributech. Segmenty jsou charakterizovány začátkem a trváním v sekundách, hlasitostí a popisem výšek a tónů:

loudness je reprezentována celkem třemi údaji: *loudness_start*, *loudness_max* a *loudness_max_time*, které popořadě reprezentují: relativní hlasitost na začátku segmentu, maximální relativní hlasitost v segmentu a čas, který specifikuje nejhlasitější místo v segmentu,

pitches je 12-dimenzionální vektor, kde každá dimenze představuje dominanci hudebního tónu v segmentu. Hudební tóny jsou popořadě: C, C#, D, D#, E, F, F#, G, G#, A, A#, B,

timbre je 12-dimenzionální vektor reprezentující jeden segment z hlediska barev tónů. Samotná společnost The Echo Nest timbre vektor neformálně popisuje jako principiálně stejně jako *Mel-frequency cepstral coefficients (MFCC)*, které se úspěšně používá v úlohách rozpoznávání řeči. Konkrétní význam a způsob získávání těchto vysoce abstraktních hodnot je bohužel proprietární. Porovnání s ostatními metodami z pohledu extrakce popisu zvuku je možné dohledat v publikaci [29].

Pro účely této práce je důležité, že takové vektory jsou vhodné na porovnávání mezi sebou a mají za cíl dobře reprezentovat hudební znaky. Toto tvrzení a další informace o způsobu získávání vektorů jsou k nalezení v oficiální dokumentaci nástroje [30],

confidence je číslo mezi 0 a 1, které symbolizuje jistotu analyzátoru pro konkrétní segment. Z důvodu zjednodušení nebude v této práci vůbec uvažován.

3.3.2 Další atributy

Mimo akustické atributy, které je možné získat z nástroje Analyze, The Echo Nest poskytuje velké množství dalších prostřednictvím svého webového API.

Poskytované atributy je možné rozdělit do tří kategorií. První kategorie jsou metadata. Do této kategorie patří například název skladby, jméno autora nebo rok vzniku. Další kategorií jsou algoritmicky odhadnuté atributy, které popisují skladbu jako celek. Příkladem takových atributů jsou danceability (míra vhodnosti skladby pro tancování) a valence (míra pozitivní energie, kterou skladba vyzařuje).¹¹

Poslední kategorií jsou atributy tzv.: sociální. Do sociálních atributů jsou zařazeny míry oblíbenosti, známosti apod. Jedním z atributů, který patří do této kategorie je atribut *hotttness*.

3.3.2.1 Atribut hotttness

Tato hodnota je udávána u interpretů a písní a odpovídá tomu, jak moc pozornosti se v současné době interpretu nebo písni dostává. Hotttness je desetinné číslo v intervalu $[0; 1]$ a je odvozeno z mnoha tisíců zdrojů jako jsou zmínky na webu, na hudebních blozích a v hudebních recenzích. Dále je hotttness ovlivňován počtem stažení, počtem přehrání a dalšími informacemi [31].

3.4 Sémantické kontexty

Sémantický kontext pro vstupní data — v našem případě rozdělení vstupních dat do tříd — je možné učinit podle několika klíčů.

Jako zdroj sémantik je možné využít metadata poskytnutá spolu s akustickými atributy. Pomocí metadat lze rozdělit skladby do tříd podle interpretů, lokace interpreta, žánrů, oblíbenosti a dalších.

Další možností je využití některých pomocných datasetů, které byli publikovány spolu s datasetem MillionSongDataset.

¹¹Vyčerpávající popis dostupných atributů je k nalezení na adrese: <http://developer.echonest.com/docs/v4/song.html>

Jako první takový je *YearPredictionMSD dataset*¹². Tento dataset obsahuje písně reprezentovány akustickými atributy spolu s cílovou proměnnou, která je rok vzniku skladby.

Další možností je využít *Taste Profile dataset*¹³. Taste Profile dataset obsahuje přes 48 miliónů trojic: (uživatel, id písně, počet přehrání). Tímto způsobem je zaznamenán hudební vkus pro více než jeden milion uživatelů.

3.5 Výběr dat a předzpracování

Jak ilustrují předchozí sekce, možnosti výběru dat a způsobů předzpracování jsou rozsáhlé. Nicméně z důvodu zjednodušení byly zvoleny pouze timbre vektory, které mají ambice nejlépe reprezentovat hudební znaky skladby. Při zvolení čistě akustického atributu se také předejde vytváření něchtěných závislostí mezi daty a zvolenou sémantikou. Každá skladba je reprezentována rozdílným počtem segmentů a tím pádem také rozdílným počtem timbre vektorů. Pro získání konečných atributů byla použita kovarianční matice a průměrný vektor, stejně tak jako v *YearPredictionMSD* datasetu. Tímto přístupem je možné pro každou skladbu získat 90 desetinných čísel, které představují reprezentaci původního objektu.

¹²Dostupný na <https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD>.

¹³Dostupný na <http://labrosa.ee.columbia.edu/millionsong/tasteprofile>.

```
1  {
2    "meta":{
3      "analyzer_version":"3.08b", "detailed_status":"OK",
4      "artist":"Michael Jackson", "album":"Thriller",
5      "title":"Billie Jean", "genre":"Rock",
6      "bitrate":192, "sample_rate":44100,
7      "seconds":294, "status_code":0,
8      "timestamp":1279120425, "analysis_time":3.83081,
9      ...
10   },
11   "track":{
12     "duration":294.15293, "loudness":-7.078,
13     "tempo":117.152, "tempo_confidence":0.848,
14     "time_signature":4,
15     "time_signature_confidence":0.42, "key":6,
16     "key_confidence":0.019, "mode":1,
17     "mode_confidence":0.416,
18     ...
19   },
20   "bars": [...],
21   "beats": [...],
22   "tatums": [...],
23   "sections": [...],
24   "segments":
25     [{
26       "start":0.00000, "duration":0.31887,
27       "confidence":1.000, "loudness_start":-60.000,
28       "loudness_max_time":0.10242,
29       "loudness_max":-16.511, "pitches":[0.370,
30       0.067, 0.055, 0.073, 0.108, 0.082, 0.123,
31       0.180, 0.327, 1.000, 0.178, 0.234],
32       "timbre":[24.736, 110.034, 57.822, -171.580,
33       92.572, 230.158, 48.856, 10.804,
34       1.371, 41.446, -66.896, 11.207]
35     },
36     ... ]
37 }
```

Ukázka 4: Výstup nástroje Analyze pro skladbu Billie Jean

Experimenty

4.1 Protokol

V této sekci jsou uvedena společná nastavení, zatímco proměnné parametry jednotlivých experimentů jsou uvedeny v příloze E. Spojení těchto informací postačí k replikaci jakéhokoliv experimentu.

Vstupní data

Pro experimentování jsou použita data, která jsou popsána v kapitole 3. Kromě předzpracování popsaného v kapitole je dále aplikovaná normalizace, jejíž význam a vliv je popsán hned v prvním experimentu 4.2.1. Vstupním vektorům jsou přiřazovány třídy, které představují sémantický kontext učení. Jednotlivé sémantiky budou upřesněny u konkrétních experimentů.

Metody

V experimentech jsou výhradně použity metody SOM a CRSOM. Teoretické základy a analytické vyjádření těchto metod jsou popsány v 1.3 a 1.4. Použitá implementace je pak rozebrána v 2.1 a 2.1.

Společné nastavení metod

Jak bylo popsáno v implementační části práce, pro metodu SOM byla použita samoorganizující síť dodaná v rámci neural network toolboxu. S tímto rozhodnutím bylo převzato velké množství nastavení, které je možné dohledat v dokumentaci¹⁴. Pro učení sítě SOM bylo vždy použito 200 epoch. Mřížka neuronů v těchto sítích byla vždy pravoúhlá (obr. 1.3b).

U sítě CRSOM je proměnlivost nastavení mnohem větší. Všechny sítě tohoto typu mají společnou inicializaci: pro počáteční nastavení prototypových

¹⁴Dokumentace dostupná na: <http://www.mathworks.com/help/nnet/ref/selforgmap.html>

vektorů byla použita metoda `initsompc`, zatímco všechny váhy v kontextové síti byly nastaveny na 0.

Velikosti mřížek neuronů jsou nastavovány v závislosti na velikosti vstupních dat. Pro n vstupních vektorů je zvolena mřížka o stranách $\lfloor\sqrt{n}\rfloor$, $\lfloor\sqrt{n}\rfloor$. Je třeba poznamenat, že takto zvolené nastavení implikuje deterministické výstupy metod.

4.1.1 Semantic relevance index

Semantic relevance index (SRI) je míra, která kvantifikuje schopnost výsledných map zachovávat sémantický význam dat v níže dimenzionální mapě [19]. SRI je poměr mezi I_{class} a O_{class} . O_{class} je míra, která určuje jak moc jsou data s rozdílným významem (třídou) na mapě rozdělena (4.2). Oproti tomu I_{class} udává, jak moc jsou data se stejným významem mapována k sobě (4.3).

$$SRI = \frac{I_{class}}{O_{class}} \quad (4.1)$$

$$I_{class} = \frac{1}{N} \sum_{i=1}^N \|H(X_i) - H(max_{in}(X_i, j))\| \quad (4.2)$$

$$O_{class} = \frac{1}{N(M-1)} \sum_{i=1}^N \sum_{k \neq C(X_i)} \|H(X_i) - H(min_{out}(X_i, j))\| \quad (4.3)$$

Kde:

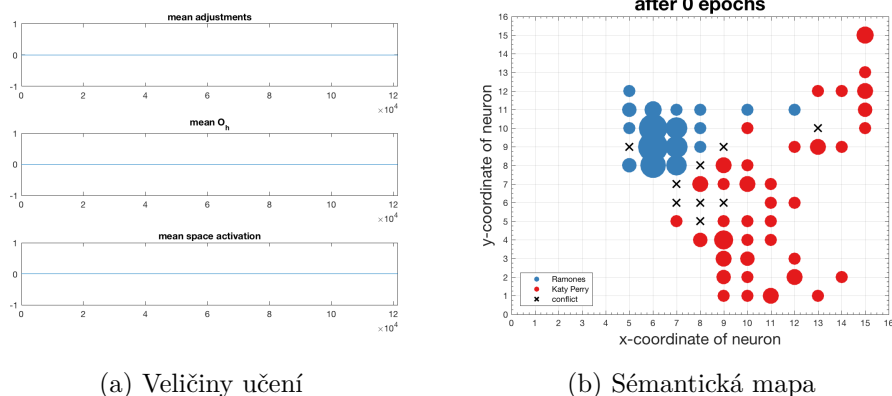
- $H(X_i)$: souřadnice *BMU* na mapě pro vstupní vektor X_i
- $min_{out}(X_i, j)$: nejbližší vstupní vektor k X_i s třídou j
- $C(X_i)$: třída vektoru X_i
- N, M : počet vstupních vektorů, počet tříd
- $max_{in}(X_i)$: vektor s třídou $C(X_i)$ a nejvyšší vzdáleností od X_i

4.2 Učení CRSOM

V této sekci jsou rozebrány některé charakteristiky učení CRSOM. Jako první téma je vliv transformace vstupních dat na učení sítě. Dále je studována průběžná chyba sítě, význam kontextové sítě a průběh učení.

4.2.1 Transformace vstupních dat a vliv na učení

Skutečnost, že učení CRSOM je citlivé na nastavení učících parametrů (např.: learning rate), je experimentálně ukázáno v [19]. Při implementaci CRSOM byla zjištěna další citlivost a to na způsobu normalizace dat. Cílem experimentu je ukázat závislost způsobu normalizace dat a průběhu učení a tyto zá-



Obrázek 4.1: Výsledky pro učení bez normalizace

vislosti interpretovat. Pro účel tohoto experimentu byl zvolen *katy-ram-dataset* (D.1). Všechny sítě v této sekci byly učeny s $LR = 0.1$ po 500 epoch.

Bez normalizace dat

V této sekci je rozebráno chování sítě CRSOM bez použití jakékoliv normalizace. Průběh učení a výsledek je ilustrován obrázky 4.1. Obrázek vlevo (4.1a) se skládá ze tří podobrázků. Na každém je zachycena jedna veličina učení. Na horním podobrázku je vykreslena průměrná úprava prototypových vah. Na prostředním podobrázku je průměrná hodnota výstupu skryté vrstvy O_h . A na dolním podobrázku je vykreslena průměrná prostorová aktivace $e^{-\mathcal{I}_i^h(t)}$. Obrázek vpravo (4.1b) pak zobrazuje výslednou sémantickou mapu.

Vysoce dimenzionální prostor zapříčinil obrovské absolutní vzdálenosti mezi vstupními vektory a nainicializovanými prototypovými vektory. Tím pádem je velikost $e^{-\mathcal{I}_i^h(t)}$ vždy zanedbatelně malá a v reprezentaci matlabovských čísel je rovna 0. To implikuje nulový výstup ze skryté vrstvy respektive nulové úpravy prototypových vektorů. Sémantická mapa (4.1b) tak pouze zobrazuje počáteční stav prototypových vektorů. Hlavním cílem normalizace dat bude zredukovat absolutní velikosti vektorů.

Min-max normalizace

Min-max je přeškálování dat do předem známého intervalu $[min; max]$. Pro Min-max normalizaci byla použita funkce `mapminmax`¹⁵, která je dostupná v matlabu.

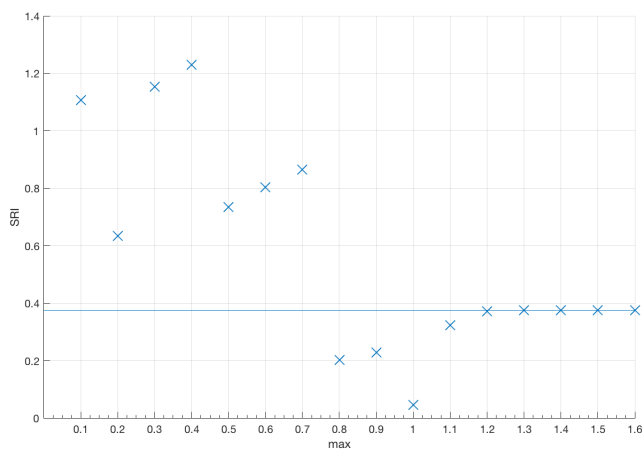
Byly zkoumány průběhy učení a výsledky pro různé meze intervalu. Minimum bylo vždy 0 a pro maxima byly voleny hodnoty $max \in \{0.1, 0.2, \dots, 1.6\}$. Jednotlivé atributy tak byly normalizovány do intervalu $[0; max]$. Výsledné

¹⁵Dokumentace na: <http://www.mathworks.com/help/nnet/ref/mapminmax.html>

4. EXPERIMENTS

SRI pro naučené sítě jsou zaneseny v tabulce 4.1. Na obrázku 4.2 jsou vynešené dvojice hodnot: max a příslušné výsledné SRI odpovídající sémantické mapy naučené CRSOM sítě. Horizontální přímkou v hodnotě 0.3749 představuje hodnotu SRI pro nenatřénovanou (pouze inicializovanou) mapu.

max	SRI
1.6	0.3749
1.5	0.3749
1.4	0.3749
1.3	0.3749
1.2	0.3709
1.1	0.3230
1.0	0.0448
0.9	0.2288
0.8	0.2028
0.7	0.8640
0.6	0.8030
0.5	0.7341
0.4	1.2298
0.3	1.1536
0.2	0.6344
0.1	1.1069



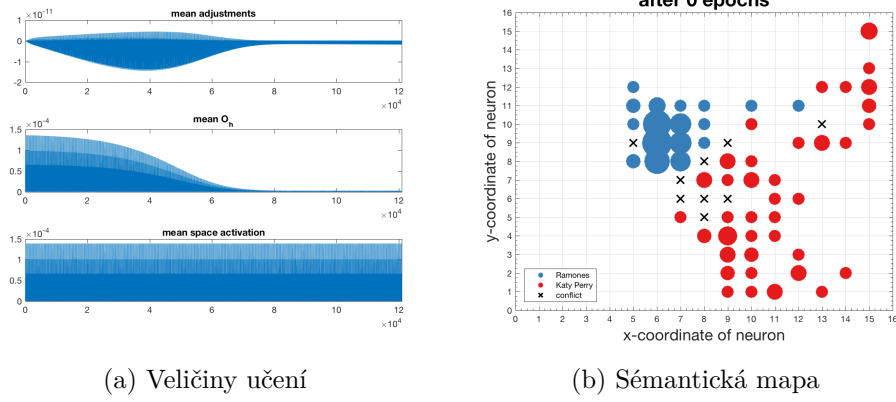
Tabulka 4.1

Obrázek 4.2

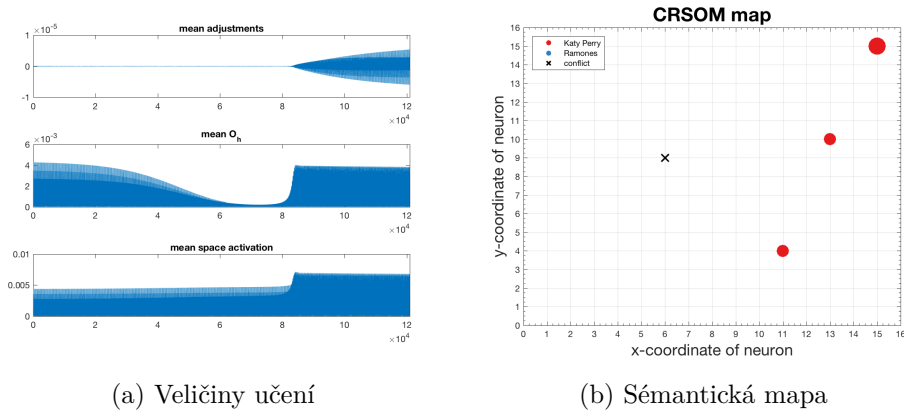
Z výsledků je možné vyzorovat skutečnost, že hodnotám $max \geq 1.3$ odpovídá $SRI = 0.3749$, což je shodná hodnota jako pro CRSOM s nenormalizovanými vstupními daty. Přestože jsou hodnoty prostorové aktivace nenulové (4.3a), pohybují se v řádech 10^{-4} , což zapříčiňuje tak malé úpravy prototypových vektorů (v řádech 10^{-11}), že se po učení se stanoveným počtem epoch mapa vůbec nezmění (4.3b).

Nejnižší hodnota SRI odpovídá $max = 1.0$. Na výsledné mapě (4.4b) je vidět dominantní neuron s pozicí na mřížce [6, 9], který se stal BMU pro většinu vstupních dat. Tuto skutečnost lze vysvětlit tak, že se neuron již od počátku učení nejvíce stával BMU a navíc byl v obklopení neuronů, kterým byla přiřazena stejná sémantika. Tím pádem byl přibližován mnohem větší silou, než ostatní neurony. Počáteční podmínky zmiňovaného neuronu lze pozorovat na předchozích sémantických mapách (4.1b a 4.3b), které jsou shodné s počátečním stavem učení. Dále je toto tvrzení podloženo obrázkem 4.5 který zobrazuje první dvě dimenze prototypových vektorů spolu se vstupními daty.

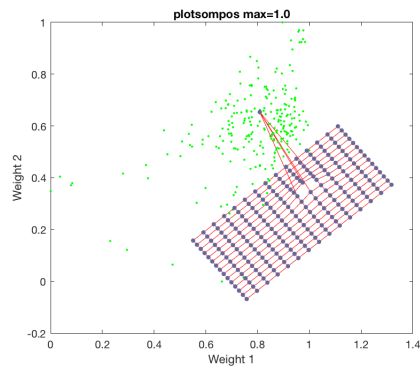
Nejvyšší hodnota SRI odpovídá $max = 0.4$. Výsledná sémantická mapa (4.6b) již odpovídá předpokladům. Samotné výsledky tohoto experimentu jsou interpretovány v sekci 4.3.1.



Obrázek 4.3: Výsledky pro $max = 1.3$

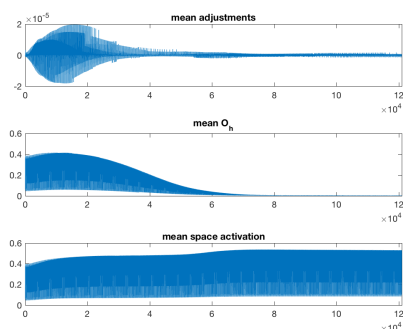


Obrázek 4.4: Výsledky pro $max = 1.0$

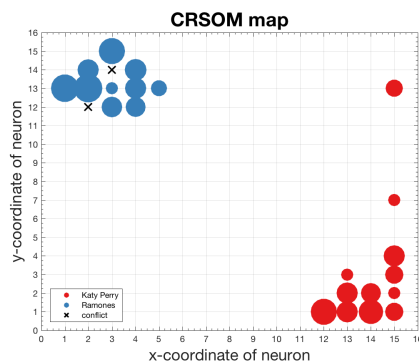


Obrázek 4.5: Dominantní neuron

4. EXPERIMENTY



(a) Veličiny učení



(b) Sémantická mapa

Obrázek 4.6: Výsledky pro $max = 0.4$

Závěr

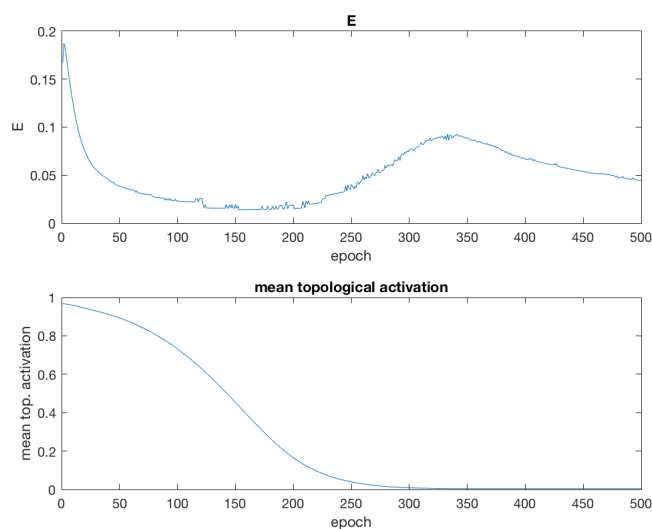
Tyto experimenty prokázaly citlivost učení CRSOM na normalizaci dat. Bez použití normalizace nedocházelo k žádným úpravám prototypových vektorů, proto se tento přístup ukázal jako nevhodný. Pro normalizaci bylo použito mapování atributů do předem zvoleného intervalu, jehož velikost byla empiricky testována. Testy ukázaly nutnost normalizace vstupních dat, nicméně není možné na jejich základě určit přesnou hodnotu max , která by vyhovovala obecně pro všechny problémy. Z toho důvodu bude k hodnotě max přistupováno jako k ostatním proměnným parametrům učení a pro každý problém se bude hledat vhodná hodnota.

4.2.2 Aspekty učení CRSOM

V této sekci bude studováno formování mapy pro síť CRSOM z předchozí sekce s $max = 0.4$, jejíž výsledná sémantická mapa je na obr. 4.6b.

Chyba sítě

Chyba sítě CRSOM již byla nadefinována vztahem 1.7. Na obrázku (4.7 nahore) je zobrazena průměrná chyba v průběhu učení dané epochami. Nemonotónní tvar této křivky je možné vysvětlit následovně: při klasické aplikaci metody nejstrmějšího sestupu se optimalizuje funkce vstupního prostoru a vah. Epochy mají navzájem stejnou množinu vstupů a úpravy vah podle gradientu nám zaručí nerostoucí průměrnou chybu. Oproti tomu při učení sítě CRSOM do funkce E vstupuje ještě další parametr – pořadí epochy. Tím se řídí *topologická aktivace* neuronů, která s rostoucím pořadím epochy klesá – a to exponenciálně. Tím pádem je funkce E definována pro rozdílné epochy různě a není pravda, že celková chyba sítě musí monotónně klesat.



Obrázek 4.7: Průměrná chyba sítě (nahore), průměrná topologická restrikce (dole)

Nárůst chyby, který je možný vidět v rozmezí 200-250 epochy, je možné vysvětlit prudkým poklesem topologické restrikce (4.7 dole). Následná opětovná konvergence E k 0 je způsobená reakcí kontextové sítě na nastalou situaci a učení vah a biasu. V této fázi je již významně upravován pouze BMU pro konkrétní vstupní vektor.

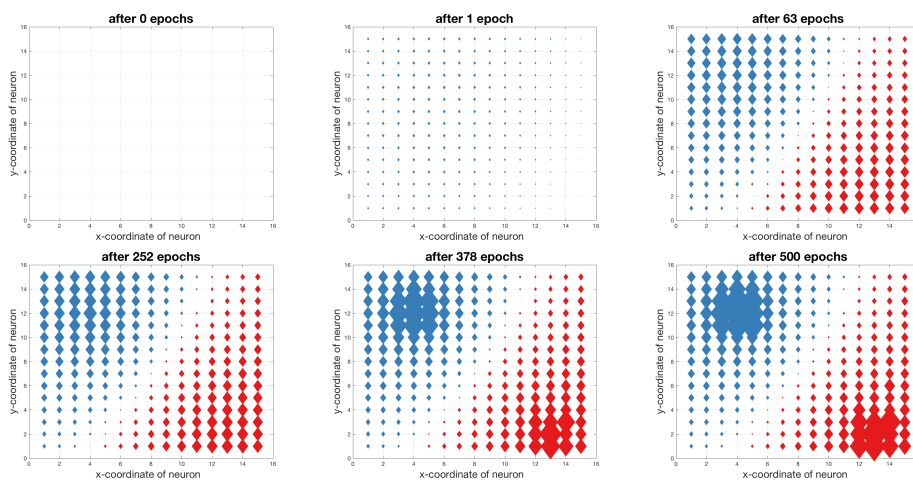
Kontextová síť

V této sekci bude studována kontextová síť. Kontextová síť se mimo jiné skládá z propojení neuronů ze skryté vrstvy a tolika výstupních složek, jako je počet tříd vstupních vektorů. Třídy, které představují sémantický kontext učení, jsou zakódovány v kódu $1 z N$ a proto každá výstupní složka představuje jednu třídu. Váhu v_{ik} tak lze interpretovat jako příslušnost i -tého neuronu k třídě k .

Na obrázku 4.8 jsou vizualizovány tyto příslušnosti neuronů k jednotlivým třídám. Značka na diskrétní pozici $[x; y]$ představuje jeden neuron s pozicí $[x; y]$ v mřížce. Barva značky určuje třídu příslušnosti (červená Katy Perry, modrá Ramones) a velikost určuje míru jistoty této příslušnosti.

Na prvním obrázku jsou zobrazeny příslušnosti před učením sítě. Protože jsou váhy inicializovány na 0, žádný neuron nepřísluší žádné třídě. S přibývajícím epochami je zřetelné rozdělování neuronů do dvou sfér vlivu, kde každá ze sfér odpovídá jedné třídě.

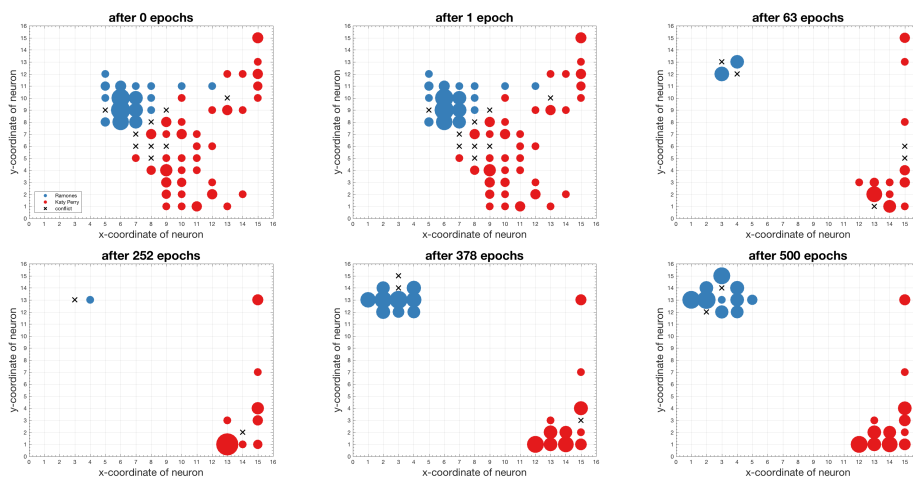
4. EXPERIMENTY



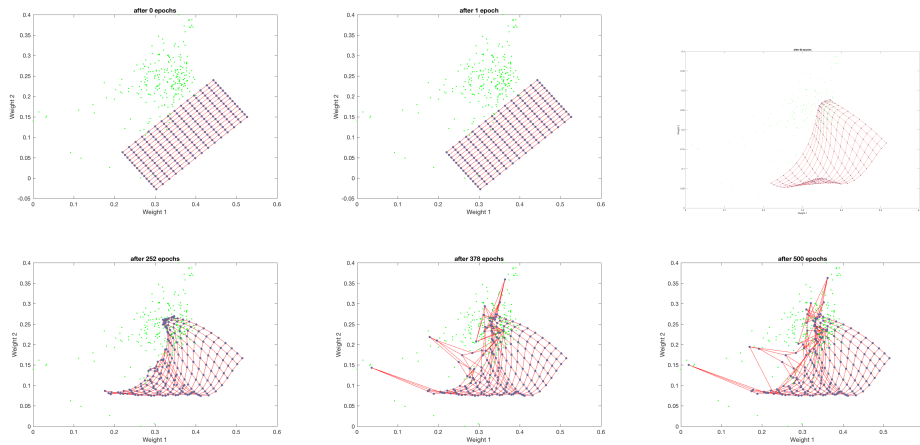
Obrázek 4.8: Příslušnost neuronů k třídám

Průběh učení

Obrázek 4.9 zobrazuje formování sémantické mapy pro tytéž epochy jako na obrázku 4.8. První z obrázku odpovídá nainicializované síti, naopak poslední odpovídá síti po uběhnutí všech 500 epoch. Ve stejných chvílích jsou na obrázku 4.10 zachyceny první dvě dimenze prototypových vektorů spolu se vstupními daty.



Obrázek 4.9: Formování sémantické mapy



Obrázek 4.10: Pozice neuronů pro první dvě dimenze

4.3 Testování reprezentace

Reprezentace písní, tedy příslušných 90 desetinných atributů, vzniklo netriviální extrakcí z akustických signálů. Není jednoznačné, zda je takto zvolený popis původního objektu (písňe) smysluplný a zachovává předpokládané vlastnosti.

Výsledkem této sekce by tak mělo být ujištění, že zvolené metody jsou schopné najít strukturu v kolekcích v takto zvolených reprezentacích hudebních skladeb pro data se sémantickými kontexty u kterých se struktura předpokládá.

V této části proto nepůjde primárně o interpretaci výsledných shluků, nýbrž o nalezení nějaké struktury. Případná validace nalezené struktury je záležitost, kterou má v popisu doménový expert.

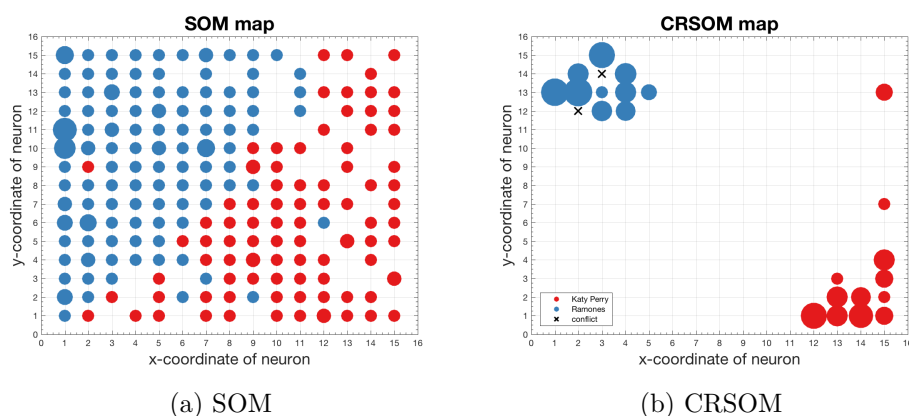
4.3.1 Interpret jako sémantický kontext

Katy Perry a Ramones

Pro počáteční sadu experimentů je sémantický kontext určen podle interpreta skladby. První zkoumaný dataset byl již použitý v předchozích sekcích – `katy-ram-dataset` (D.1), který obsahuje skladby zpěvačky Katy Perry a kapely Ramones. Výsledky tohoto experimentu (exp. E.1), stejně tak jako všech následujících, jsou v podobě dvou sémantických map (obr. 4.11). První mapa (obr. 4.11a) představuje natrénovanou samoorganizující mapu, zatímco druhá mapa (obr. 4.11b) pochází ze skryté vrstvy natrénované samoorganizující mapy se zahrnutým sémantickým kontextem.

SOM při učení nebere v úvahu třídy vstupních vektorů a výsledná sémantická mapa vizualizuje pouze povahu vstupního prostoru. Podle této vizua-

4. EXPERIMENTY



Obrázek 4.11: Katy Perry a Ramones (exp. E.1)

lizace (obr. 4.11a) lze konstatovat celkovou rozdílnost vektorů spadající do různých tříd.

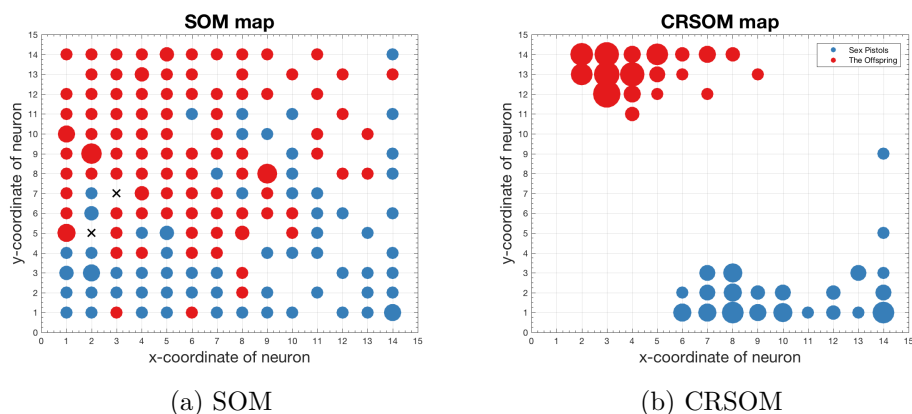
Oproti tomu CRSOM sémantický kontext zahrnuje do procesu učení explicitně a z příslušné sémantické mapy (obr. 4.11b) lze extrahovat rozdílné znalosti. Síť vytvořila dva koncentrované shluky v rozích mapy, což svědčí o dobré oddělitelnosti těchto dvou tříd. Konflikty v horním shluku (např. neuron [3; 14]) implikuje podobné reprezentace pro skladby s rozdílnými sémantikami. Tato podobnost je již zřejmá z mapy SOM – např. neuronem [2; 9]. Neuron [15; 13] představuje shluk odlehlých hodnot. Po prozkoumání objektů, které přísluší tomuto neuronu se ukázalo, že tyto nahrávky neobsahují hudbu, nýbrž se jedná o mluvené slovo Katy Perry a poskytovatel dat ho mylně zařadil mezi písně.

The Offspring a Sex Pistols

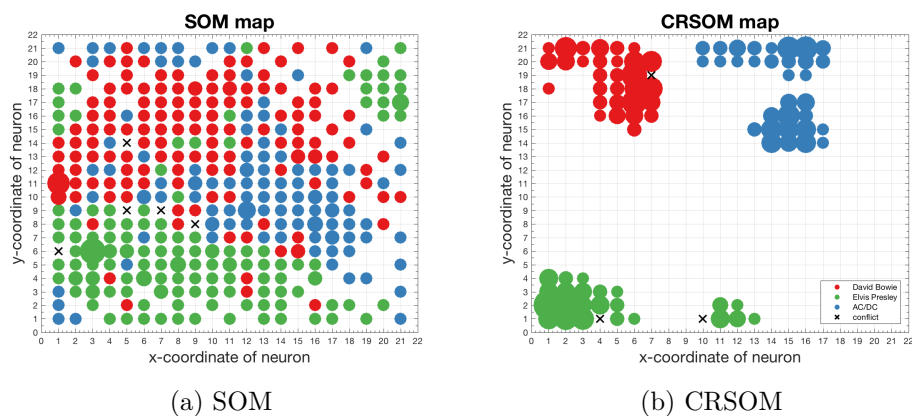
Další dataset se skládá z diskografie kapel Sex Pistols a The Offspring. Mapa SOM (obr. 4.12a) zobrazuje větší promíchanost objektů v prostoru, než se tomu jednalo u předchozího experimentu, což svědčí o větší podobnosti hudebních znaků těchto dvou kapel. I přes tento fakt jsou data dobře oddělitelné podle této sémantiky (obr. 4.12). Výsledná mapa dokonce neobsahuje žádné konflikty, což lze přisoudit delšímu učení (exp. E.2), než tomu bylo v předchozím případě.

David Bowie, Elvis Presley a AC/DC

Další zkoumaný dataset obsahuje skladby následujících interpretů: David Bowie, Elvis Presley a AC/DC. I zde se jedná o dobře rozlišitelné interprety o čemž svědčí výsledky experimentů (obr. 4.13).



Obrázek 4.12: Sex Pistols a The Offspring (exp. E.2)



Obrázek 4.13: David Bowie, Elvis Presley a AC/DC (exp. E.3)

Metallica, Maddona a Louis Armstrong

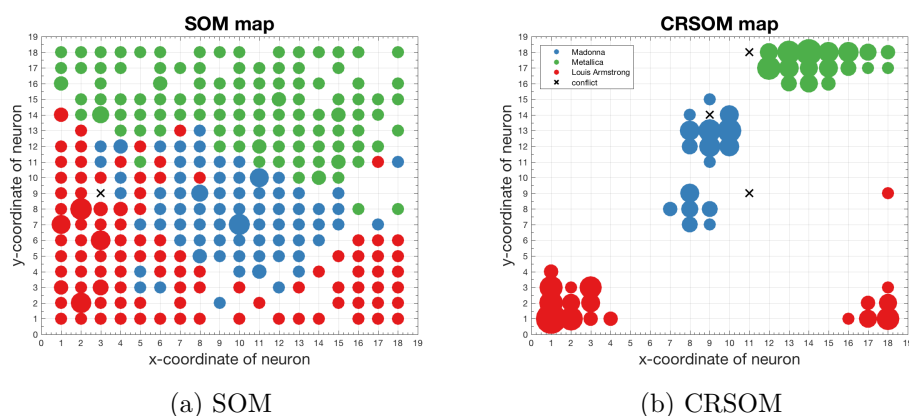
Poslední dataset s daty rozdělenými do tříd podle interpretů je *met-mad-la-dataset* (D.4). Tak jako ve všech předchozích případech, i zde CRSOM síť dokázala sformovat dobře ucelené shluky podle dané sémantiky (obr. 4.14b).

Pro představu, jak vypadá sémantická mapa, ze které nelze extrahovat žádnou strukturu dat, byla vytvořena rozdílná sémantika pro stejný dataset. Sémantika byla vytvořena náhodným prohazením tříd objektů. Síť SOM a CRSOM byly učeny se stejnými parametry jako v předchozím v případě. Výsledky tohoto experimentu jsou na obrázcích 4.15.

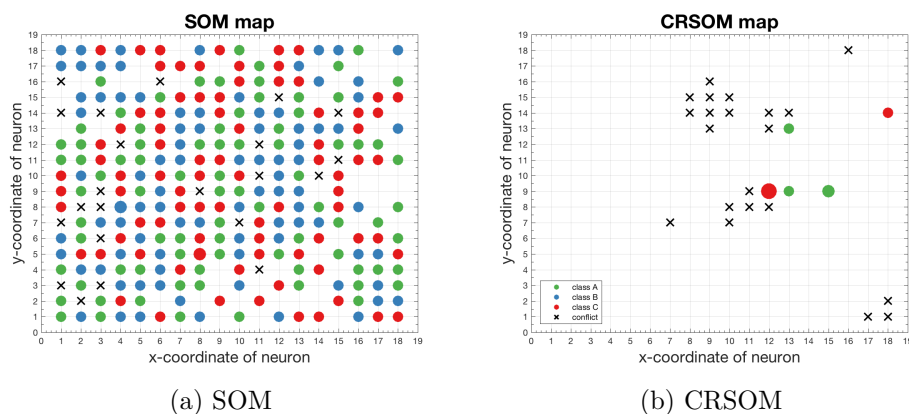
4.3.2 Rok vzniku jako sémantický kontext

Další zkoumaná sémantika je rok vzniku skladby. Pro tento účel byly vytvořeny dva datasety *60-vs-70-dataset* (D.6) a *70-vs-10-dataset* D.7. V tomto

4. EXPERIMENTY



Obrázek 4.14: Metallica, Maddona a Louis Armstrong (exp. E.4)



Obrázek 4.15: Dataset s náhodnou sémantikou (exp. E.5)

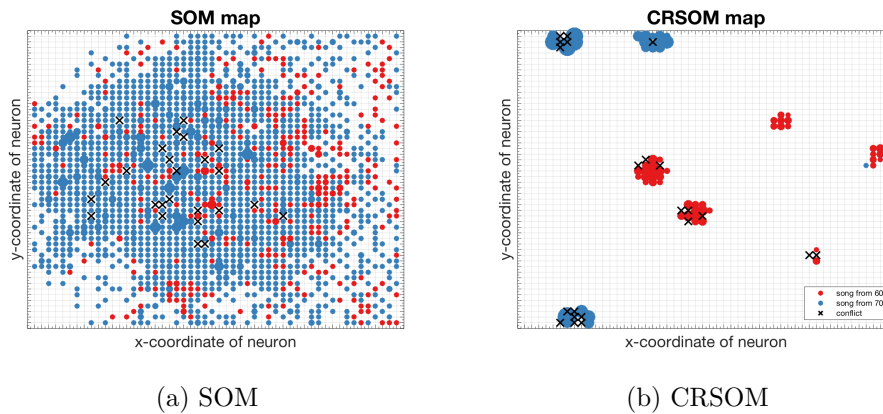
případě se jedná o obrovské datové sady, potažmo sítě, které bylo nutné učit po velkém počtu epoch – v řádech tisíců.

Předpoklad, že pomocí CRSOM map bude možné extrahovat strukturu kolekcí, se naplnil a v obou případech je možné určit několik shluků pro jednotlivé sémantiky. Více shluků v rámci jedné třídy svědčí o rozdílnosti skladeb v jedné sémantice, což je další předpokládaný závěr.

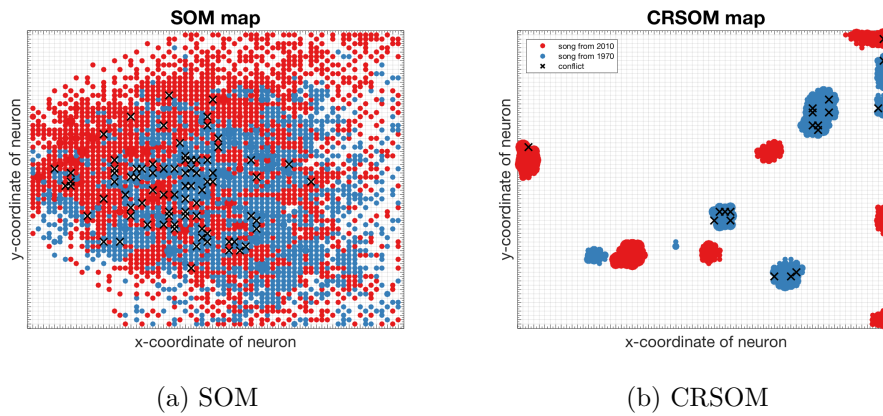
4.4 Hledání struktur

4.4.1 Posluchačská historie jako sémantický kontext

V této sekci je použita historie přehrávání náhodného uživatele. Jsou k dispozici přehrané skladby a k nim počet přehrávání (*play count*) z datasetu Taste Profile, který byl představen v sekci 3.4. Z počtu přehrávání byly následujícím způsobem vytvořeny tři třídy:



Obrázek 4.16: Skladby z 1960 a 1970 (exp. E.6)



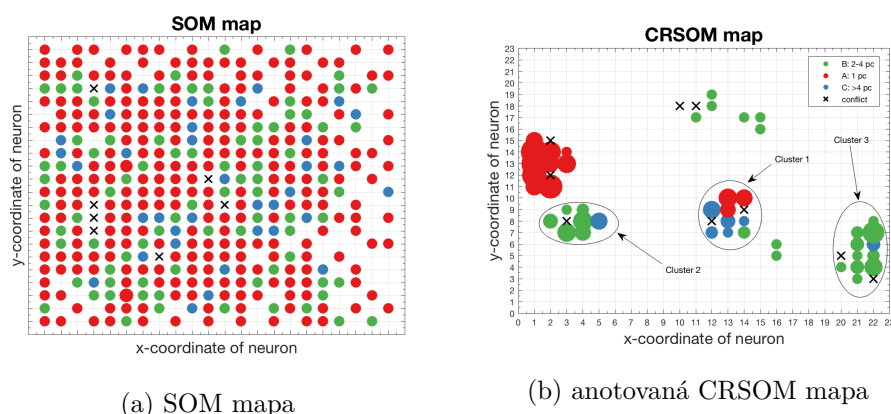
Obrázek 4.17: Skladby z 1970 a 2010 (exp. E.7)

Rozdělení dat do třech tříd

- **třída A:** 1 přehrání,
- **třída B:** 2 – 4 přehrání,
- **třída C:** ≥ 5 přehrání.

Předpokladem je, že počty přehrání odrážejí uživatelskou oblíbenost skladby a tato oblíbenost souvisí s hudebními znaky, které jsou zachycené v reprezentaci písně. Výsledky učení v kontextu takto zvolené sémantiky jsou na obr. 4.18. Na rozdíl od předchozích experimentů, kde mapy obsahovaly z pohledu sémantiky homogenní shluky, v tomto případě (obr. 4.18a) síť vytvořila několik heterogenních shluků s několika konflikty. To je možné dát za vinu špatnému rozdělení do tříd, které neodráží žádné zákonitosti, které by bylo možné zachytit sémantickou mapou. Byly identifikovány dva problémy se současným

4. EXPERIMENTY



Obrázek 4.18: Posluchačská historie 3 třídy (exp. E.8)

rozdělením do tříd, které souvisí se skutečností, že počty přehrání představují zachycení stavu v jednom okamžiku.

První problém je se skladbami, které mají jediné přehrání. U takových písní není jasné, jestli se jedná o konečný počet – jen pak bylo možné píseň interpretovat jako pro uživatele neoblíbenou.

Další problém je uměle stanovená hranice pěti přehrání, která představuje vysoce oblíbenou skladbu. Shluky 2 a 3 obsahují pouze hudbu třídy B a C a je možné je označit jako oblíbené s potenciálem dalších přehrávání.

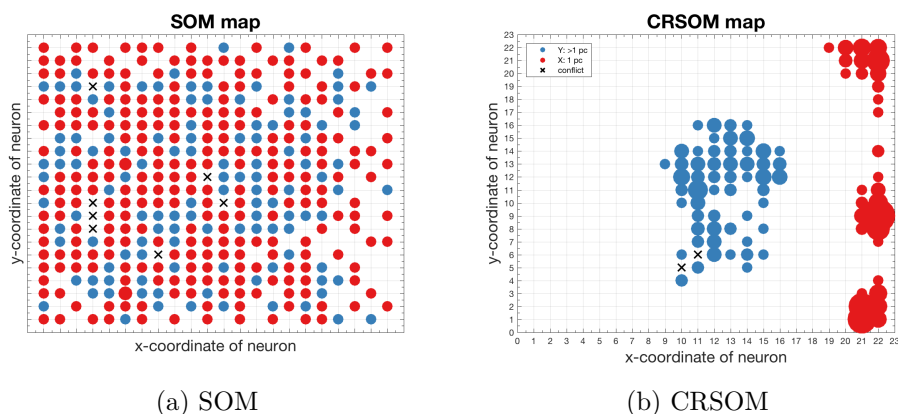
Ve shluku 1 jsou koncentrované skladby všech tříd, což znamená, že i přes své rozdílné sémantiky nebyly učením odděleny, což implikuje velmi podobné reprezentace. Za stanovených předpokladů je možné skladby s jedním přehráním ve shluku 1 interpretovat jako nově objevené a jelikož sdílejí hudební znaky s objekty z ostatních tříd, toto jediné přehrání interpretovat jako dočasné.

V dalším experimentu byla prozkoumána následující sémantika pro stejná data:

Rozdělení dat do dvou tříd

- **třída X:** 1 přehrání,
- **třída Y:** ≥ 2 přehrání.

Výsledky jsou k vidění na obr. 4.19. Na rozdíl od předchozí sémantiky takto zvolené třídy vytvářejí vzájemně dobře oddělitelné shluky (obr. 4.19b). Shluky spadající do třídy Y vytváří nepřiliš koncentrovaný shluk, což svědčí o hudební rozmanitosti těchto skladeb. Dva konflikty (neurony [10; 5] a [11; 6]) byly BMU pro některé skladby s jedním přehráním. Pro podobnost reprezentací se skladbami z třídy Y lze předpokládat, že uživatel tyto skladby přehraje v budoucnu.



Obrázek 4.19: Posluchačská historie s rozdílnou sémantikou (exp. E.9)

Kategorie X vytvořila několik shluků, které učení CRSOM nepodařilo sloučit, což implikuje vysoké rozdílnosti v reprezentacích v rámci jedné třídy. Tento fakt je možné vysvětlit již zmíněným nejasným významem jednoho přehrání, který se dá interpretovat jako neoblíbená hudba nebo jako nová hudba.

4.4.2 Oblíbenost jako sémantický kontext

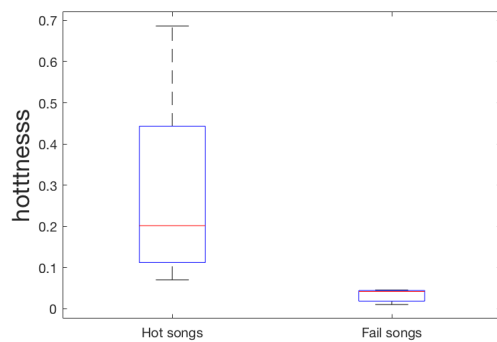
V této sekci budou zkoumány struktury hudebních kolekcí se sémantikou podle současné oblíbenosti. V představení dostupných dat (konkrétně sekce 3.3.2.1) byl představen atribut hottness, který tuto informaci obsahuje.

Protože je atribut hottness desetinné číslo, bylo nutné ho denormalizovat do dvou tříd (Fail a Hot). Konkrétní rozdělení je názorně předvedeno v krabicovém diagramu na obrázku 4.20.

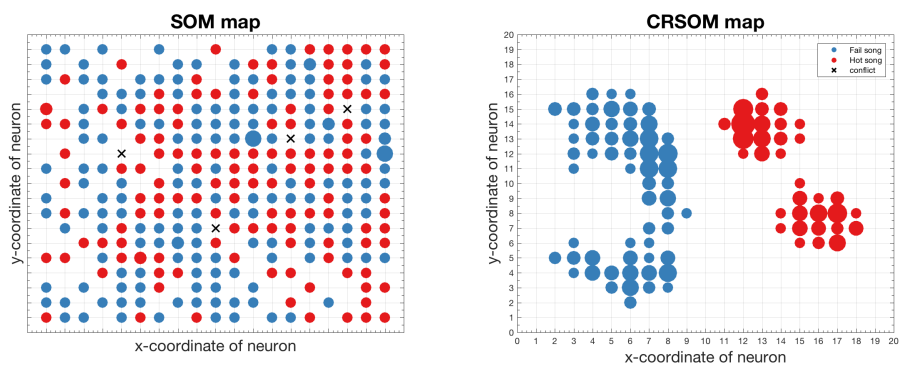
Z výsledků (obr. 4.21) lze konstatovat, že třídy písní rozdělené podle oblíbenosti jsou dobře oddělitelné, dokonce se žádný neuron nestal BMU pro vstupní vektory s rozdílnými třídami.

Přesto mapa neobsahuje koncentrované shluky, což svědčí o tom, že v obou třídách se vyskytují skladby z různých hudebních odvětví.

4. EXPERIMENTY



Obrázek 4.20: Krabicový diagram pro hotttness



(a) SOM

(b) CRSOM

Obrázek 4.21: Oblíbenost skladby (exp. E.10)

Predikce úspěšnosti písně

Posledním krokem KDD je využití objevených znalostí v reálných systémech. Využití CRSOM na predikci oblíbenosti skladby bylo inspirováno posledním experimentem (4.4.2). Následná rešerše ukázala, že toto téma je velmi oblíbené a zároveň z vědeckého hlediska rozporuplné.

5.1 Rešerše

Dokladem oblíbenosti je termín *Hit Song Science (HSS)*, který představuje snahu automatizovaně předpovídat úspěch písně metodami jako je strojové učení.

Rozporuplnost tohoto tématu ukazují publikace s názvy jako: *Automatic prediction of hit songs* [32], *Hit song science is not yet a science* [33] a *Hit song science once again a science* [34].

Nepředvídatelnost kulturního trhu, proměnlivost trendů a předpokládaná neschopnost odvození popularity z akustických vlastností písní patří mezi hlavní kritiky HSS a podle [33] nebylo možné předpovídat popularitu písně pomocí tehdy nejmodernějších metod strojového učení.

Navzdory tomu byl sestrojen model, který úspěšně předpověděl populární taneční písně pro rok 2015 podle žebříčku Hot Dance/Electronic Songs¹⁶, což lze považovat za úspěch v odvětví HSS. [35]

Na rozdíl od všech předchozích publikací tento úspěšný model používá The Echo Nest data jako zdroj atributů — tzn. stejný jako v této práci. Nejúspěšnějším modelem se stala *Logistická regrese*, další zkoumané modely a výsledky jsou k nalezení v originální publikaci [36]. Tento model se dočkal nasazení v podobě webové služby Dance hit prediction app¹⁷, která po nahrání libovolné písně odpoví desetinným číslem, které představuje pravděpodobnost, že se na-

¹⁶Dostupný na adrese: <http://www.billboard.com/charts/dance-electronic-songs>

¹⁷Dostupná na adrese: <http://dorienherremans.com/dance/uploader.php>

hraná píseň stane populární, v tomto případě, že se dostane do nejlepších 10 písní ve zmíněném žebříčku.

5.2 Návrh využití

Z rešerše vyplývá, že predikce úspěšnosti hudební skladby z akustických vlastností je nesnadný úkol. V této práci nepůjde o překonání výsledků v přesnosti predikce, nýbrž bude využit jiný přístup k tomuto problému, který nabízí samoorganizující mapa se zahrnutým sémantickým kontextem.

S velkým množstvím prediktivních modelů (např. MLP) je nutné zacházet jako s tzv. *black box*. To znamená, že nelze jednoduše vysvětlit rozhodnutí modelu pro konkrétní vstupní data, to se týká i zmíněné implementace Dance hit prediction app.

Oproti tomu natrénovaná CRSOM nabízí pohled na strukturu dat a dokáže umístit nová data na mapu na základě pozice neuronu, který se aktivuje (stane se BMU) pro novou píseň.

Pokud natrénovaná mapa obsahuje dobře oddělené shluky, uživatel sám dokáže intuitivně určit pravděpodobnost úspěchu písně podle pozice vítězného neuronu. K podpoření jeho závěru mu dále dopomůže interaktivní prozkoumávání mapy spolu s ukázkami písní.

5.3 Proof of concept

V této sekci budou identifikovány klíčové části návrhu a popis technických záležitostí a případné implementace.

1. Transformace hudebního souboru do atributů

Echo nest API umožňuje přístup k nástroji Analyze a získat tak všechny potřebné informace (stejně jako v ukázce 4) z audio souboru. Tato data pak projdou běžným předzpracováním stejným způsobem jako ostatní vstupní data použitá v této práci.

2. BMU pro nová data

Tato funkčnost je již používána při tvorbě sémantických map.

3. Zpětné přiřazení písní k neuronu

Tato funkčnost byla implementována již v rámci experimentů a to pro účely interpretace map.

4. Hudební ukázky k datům

Pro implementaci této funkcionality lze použít 30 sekundové hudební ukázky nabízené společností Spotify¹⁸ prostřednictvím webového API¹⁹.

¹⁸Spotify je služba, která nabízí streamování či podcasting hudby.

¹⁹Dokumentace na adrese <https://developer.spotify.com/web-api/>

5.3.1 Natrénování mapy

Pro zmíněné účely bylo staženo celkem 5140 písní populární hudby. Interpreti všech písní se řadí mezi neznámé kapely, zpěváky a zpěvačky — existující maximálně 4 roky — což by mělo omezit vliv známosti interpreta na úspěch skladby. Dále se tím bojuje proti proměnlivosti trendů a hudebních vkusů napříč časovými epochami.

Z důvodu velké náročnosti učení sítě CRSOM byly písně rozděleny na tři datasety podle velikosti. Zde budou probrány pouze nejlepší výsledky, které vzešly z nejmenšího datasetu `hit-normal-flop-dataset` (D.11), který obsahuje celkem 1541 písní. Všechny písně jsou reprezentovány a transformovány stejným stylem jako v kapitole 4.

Sémantický kontext pro skladby byl volen následujícím způsobem:

Rozdělení dat do tří tříd podle hotttnesss

- **třída flop:** ≤ 0.08 ,
- **třída normal:** $0.09 - 0.25$,
- **třída hit:** ≥ 0.26 .

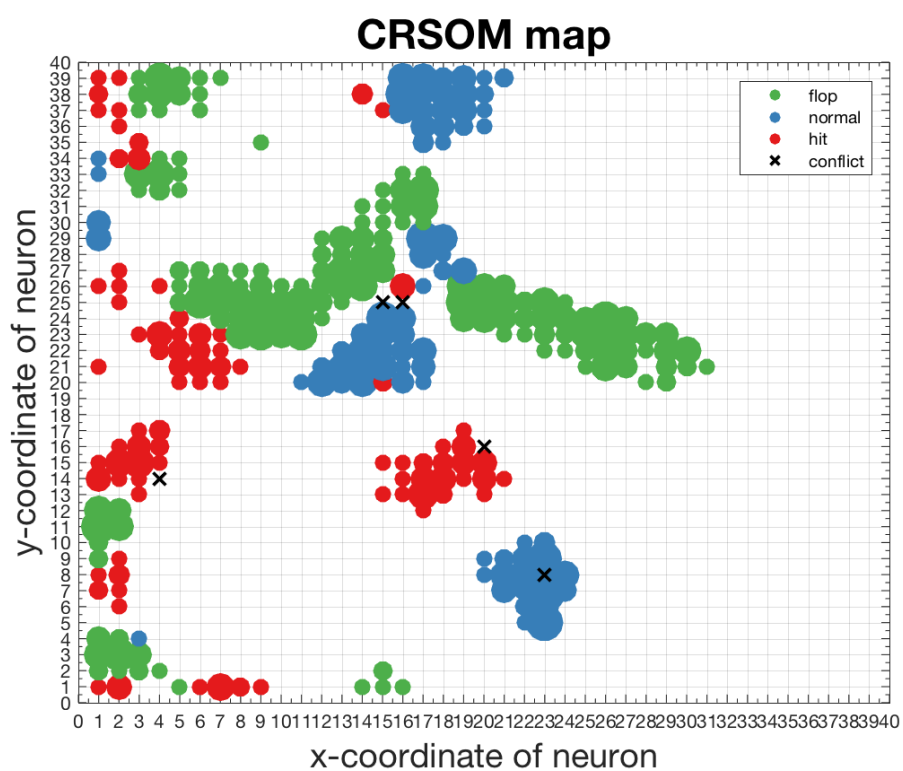
Z výsledné sémantické mapy (obr. 5.1) lze snadno odvodit vnitřní strukturu těchto dat, která může sloužit jako podklad pro rozhodování o úspěšnosti nové písně.

Dále byla vyzkoušena rozdílná diskretizace atributu hotttnesss – výsledky tohoto pokusu jsou k nahlédnutí v příloze C.

5.3.2 Diskuze o využitelnosti

Pro využití mapy jako plnohodnotného klasifikátoru by bylo zapotřebí použít všechna data, což je při současné implementaci sítě nezvládnutelné. Zároveň by bylo nutné ověřit generalizační schopnosti CRSOM. Tento koncept je využitelný spíše jako pomocník pro vytváření vlastních závěrů o potencionální oblíbenosti písně na základě objevené struktury v datech.

Celý tento koncept je možné implementovat v podobě webové služby, stejně jako Dance hit prediction app.



Obrázek 5.1: Výsledná CRSOM pro experiment (E.11)

Závěr

Samoorganizující mapy (SOM) a samoorganizující mapy se zahrnutým sémantickým kontextem (CRSOM) umožňují vizualizovat vysoce dimenzionální prostor v podobě sémantických map a objevovat jeho skrytou strukturu. Z výsledků pro jednotlivé sítě lze získat různé znalosti. SOM při mapování prostoru zachovává topologické vlastnosti a z výsledné sémantické mapy lze odvodit povahu vstupního prostoru v podobě rozmístění vstupních dat. CRSOM při učení explicitně využívá sémantický kontext dat (třídy) a výslednou sémantickou mapu lze interpretovat jako vizualizaci klasifikačního problému.

V rámci této práce byly úspěšně použity zmíněné metody na kolekce populární hudby jako zástupce multimediálních (audio) dat. Písně byly výhradně reprezentovány svými akustickými vlastnostmi, konkrétně timbre vektory poskytnuté službou Analyze, a tento popis se ukázal jako smysluplný například pro zachycení rozdílů mezi interprety a roky vzniků skladeb. Možný přístup při hledání a interpretaci struktury v podobě vytvořených shluků v sémantické mapě byl demonstrován pro různé sémantické kontexty a různá data.

Experiment se sémantikou podle oblíbenosti písně inspiroval možné využití CRSOM mapy pro odhad úspěšnosti písně, které bylo dále rozpracováno v podobě konceptu. Tento koncept přináší nový pohled na Hit Song Science (HSS), v kterém není snahou vytvořit model, který predikuje úspěšnost jako black box, nýbrž využívá nalezenou strukturu a nechá uživatele o výsledku rozhodnout na základě umístění písně do sémantické mapy a explorační okolků písní s hudebními ukázkami.

Z důvodu výpočetní náročnosti CRSOM nebyly v této práci ověřeny generalizační schopnosti této sítě pro potřeby konceptu. Další práce by tak měla primárně směřovat ke zrychlení sítě, ať už v podobě vyladění implementace, použití paralelizace nebo dalších přístupů, které mohou zrychlit učení. Poté by bylo možné formálně definovat koncept jako klasifikační úlohu a běžnými přístupy znalostního inženýrství tento cíl naplnit.

Literatura

- [1] Fayyad, U.; Piatetsky-shapiro, G.; Smyth, P.: From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, ročník 17, 1996: s. 37–54.
- [2] Warne, T.: Data: your most valuable asset. A business case for data governance. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <https://www.linkedin.com/pulse/20141119183044-243338813-data-your-most-valuable-asset-a-business-case-for-data-governance>
- [3] Johnston, S. J.: World's Data Doubling Every Two Years. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <http://www.infostor.com/storage-management/worlds-data-doubling-every-two-years-.html>
- [4] [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <http://slovník-cizich-slov.abz.cz/web.php/slovo/struktura>
- [5] Marshall, D.: Multimedia data. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <https://www.cs.cf.ac.uk/Dave/Multimedia/node141.html>
- [6] Zhang, Q.; Segall, R.; Cao, M.: *Visual Analytics and Interactive Technologies: Data, Text, and Web Mining Applications*. Premier reference source, Information Science Reference, 2011, ISBN 9781609601027. Dostupné z: <https://books.google.cz/books?id=52dQuAAACAAJ>
- [7] Piatetski, G.; Frawley, W.: *Knowledge Discovery in Databases*. Cambridge, MA, USA: MIT Press, 1991, ISBN 0262660709.
- [8] Maimon, O.; Rokach, L.: *Data Mining and Knowledge Discovery Handbook*, kapitola Introduction to Knowledge Discovery in Databases. Boston, MA: Springer US, 2005, ISBN 978-0-387-25465-4, s. 1–17, doi: 10.1007/0-387-25465-X_1. Dostupné z: http://dx.doi.org/10.1007/0-387-25465-X_1

- [9] Rouse, M.: garbage in, garbage out (GIGO). [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <http://searchsoftwarequality.techtarget.com/definition/garbage-in-garbage-out>
- [10] Berka, P.: Proces dobývání znalostí). [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <http://euromise.vse.cz/kdd/index.php>
- [11] Mahajan, M.; Nimbhorkar, P.; Varadarajan, K.: The Planar K-means Problem is NP-hard. *Theor. Comput. Sci.*, ročník 442, Červenec 2012: s. 13–21, ISSN 0304-3975, doi:10.1016/j.tcs.2010.05.034. Dostupné z: <http://dx.doi.org/10.1016/j.tcs.2010.05.034>
- [12] Coursera: Proces dobývání znalostí. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <https://www.coursera.org/learn/machine-learning/lecture/93VPG/k-means-algorithm>
- [13] Kohonen, T.: Essentials of the Self-organizing Map. *Neural Netw.*, ročník 37, Leden 2013: s. 52–65, ISSN 0893-6080, doi:10.1016/j.neunet.2012.09.018. Dostupné z: <http://dx.doi.org/10.1016/j.neunet.2012.09.018>
- [14] Bação, F.; Lobo, V.; Painho, M.: *Computational Science – ICCS 2005: 5th International Conference, Atlanta, GA, USA, May 22-25, 2005, Proceedings, Part III*, kapitola Self-organizing Maps as Substitutes for K-Means Clustering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, ISBN 978-3-540-32118-7, s. 476–483, doi:10.1007/11428862_65. Dostupné z: http://dx.doi.org/10.1007/11428862_65
- [15] Sayad, D. S.: An Introduction to Data Mining. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: http://www.saedsayad.com/clustering_som.htm
- [16] Campoy, P.: Dimensionality reduction by self organizing maps that preserve distances in output space. In *2009 International Joint Conference on Neural Networks*, June 2009, ISSN 2161-4393, s. 432–438, doi:10.1109/IJCNN.2009.5179009.
- [17] Vesanto, J.; Alhoniemi, E.: Clustering of the Self-Organizing Map. 2000.
- [18] Keim, D. A.: Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, ročník 8, č. 1, Leden 2002: s. 1–8, ISSN 1077-2626, doi:10.1109/2945.981847. Dostupné z: <http://dx.doi.org/10.1109/2945.981847>
- [19] Hartono, P.; Hollensen, P.; Trappenberg, T.: Learning-Regulated Context Relevant Topographical Map. *IEEE Transactions on Neural Networks*

- and Learning Systems*, ročník 26, č. 10, Oct 2015: s. 2323–2335, ISSN 2162-237X, doi:10.1109/TNNLS.2014.2379275.
- [20] FISHER, R. A.: THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. *Annals of Eugenics*, ročník 7, č. 2, 1936: s. 179–188, ISSN 2050-1439, doi:10.1111/j.1469-1809.1936.tb02137.x. Dostupné z: <http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x>
- [21] Hartono, P.; Trappenberg, T.: Classificability-regulated self-organizing map using restricted RBF. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, ISSN 2161-4393, s. 1–5, doi: 10.1109/IJCNN.2013.6706732.
- [22] Hartono, P.; Ogawa, K.: Visualizing Learning Management System data using Context-Relevant Self-Organizing Map. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2014, ISSN 1062-922X, s. 3487–3491, doi:10.1109/SMC.2014.6974469.
- [23] Make It Work Make It Right Make It Fast. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <http://c2.com/cgi/wiki?MakeItWorkMakeItRightMakeItFast>
- [24] Vectorization. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: http://www.mathworks.com/help/matlab/matlab_prog/vectorization.html
- [25] Vectorization. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: http://www.mathworks.com/help/matlab/matlab_prog/preallocating-arrays.html
- [26] Benchmark language implementations. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <https://attractivechaos.github.io/plb/>
- [27] MetaCentrum wiki. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: https://wiki.metacentrum.cz/wiki/Main_Page1
- [28] Encyklopedie fyziky. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <http://fyzika.jreichl.com/main.article/view/152-zvukove-vlneni>
- [29] Schindler, A.; Rauber, A.: *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation: 10th International Workshop, AMR 2012, Copenhagen, Denmark, October 24-25, 2012, Revised Selected Papers*, kapitola Capturing the Temporal Domain in Echonest Features for Improved Classification Effectiveness. Cham: Springer International Publishing, 2014, ISBN 978-3-319-12093-5, s. 214–227, doi:10.1007/978-3-319-12093-5_13. Dostupné z: http://dx.doi.org/10.1007/978-3-319-12093-5_13

- [30] Jehan, T.: Analyzer Documentation. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf
- [31] Lamere, P.: Posts Tagged hotttnesss. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <https://musicmachinery.com/tag/hotttnesss/>
- [32] Dhanaraj, R.; Logan, B.: Automatic prediction of hit songs. In *In Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2005, s. 488–491.
- [33] Pachet, F.; Roy, P.: Hit song science is not yet a science. In *In Proc. of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, 2008, s. 355–360.
- [34] Ni, Y.; Santos-Rodriguez, R.; Mcvicar, M.; aj.: Hit song science once again a science. *4th International Workshop on Machine Learning and Music*, 2011.
- [35] Neal, M.: A Machine Successfully Predicted the Hit Dance Songs of 2015. [online], Naposledy navštíveno 3. 5. 2016. Dostupné z: <http://motherboard.vice.com/read/a-machine-successfully-predicted-the-hit-dance-songs-of-2015>
- [36] Herremans, D.; Martens, D.; Sörensen, K.: Dance hit song prediction. Working papers, University of Antwerp, Faculty of Applied Economics, 2014. Dostupné z: <http://EconPapers.repec.org/RePEc:ant:wpaper:2014003>

Seznam použitých zkratek

- API** Application programming interface
- BMU** Best matching unit
- BSON** Binary Structured Object Notation
- CRSOM** Context-relevant self-organizing map
- HSS** Hit song science
- JSON** JavaScript Object Notation
- KDD** Knowledge discovery in databases
- MLP** Multilayer perceptron
- SOM** Self-organizing map

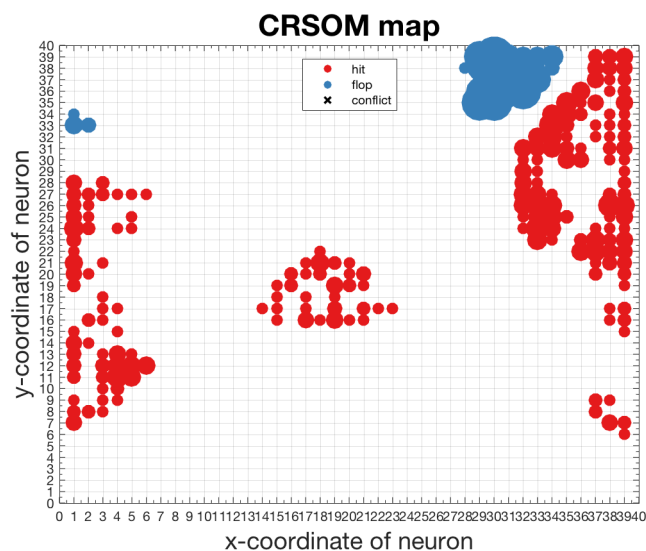
Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF

Další výsledky se sémantickým kontextem podle úspěšnosti písňe

Rozdělení dat do tříd podle hotttnesss

- třída flop: ≤ 0.12 ,
- třída hit: > 0.12 .



Obrázek C.1: Výsledná mapa pro experiment E.12

Datasets

D.1 Interpret jako sémantika

Tabulka D.1: katy-ram-dataset

celkový počet záznamů	Katy Perry	Ramones
242	63%	37%

Tabulka D.2: off-sexp-dataset

celkový počet záznamů	The Offspring	Sex Pistols
211	63%	37%

Tabulka D.3: dp-ep-acdc-dataset

celkový počet záznamů	David Bowie	Elvis Presley	AC/DC
465	36%	35%	29%

Tabulka D.4: met-mad-la-dataset

celkový počet záznamů	Metallica	Louis Armstrong	Maddona
343	39%	26%	35%

Tabulka D.5: met-mad-la-random-dataset

celkový počet záznamů	–	–	–
343	39%	26%	35%

D.2 Rok vzniku skladby jako sémantika

Tabulka D.6: 60-vs-70-dataset

celkový počet záznamů	Skladba z roku 1960	Skladba z roku 1970
2773	16%	84%

Tabulka D.7: 70-vs-10-dataset

celkový počet záznamů	Skladba z roku 1970	Skladba z roku 2010
4698	50%	50%

D.3 Počet přehrání jako sémantika

Tabulka D.8: user-3class-dataset

celkový počet záznamů	1 přehrání	2–4 přehrání	více než 4 přehrání
521	67%	23%	10%

Tabulka D.9: user-2class-dataset

celkový počet záznamů	1 přehrání	více než 1 přehrání
521	67%	33%

D.4 Oblíbenost jako sémantika

Tabulka D.10: hot-fail-dataset

celkový počet záznamů	Hot	Fail
363	43%	57%

Tabulka D.11: hit-normal-flop-dataset

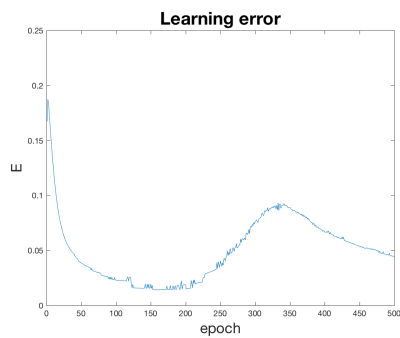
celkový počet záznamů	flop	normal	hit
1542	50%	36%	14%

Tabulka D.12: hit-fail-dataset2

celkový počet záznamů	Hot	Fail
1542	37%	63%

Protokoly experimentů

E.1 Katy Perry a Ramones



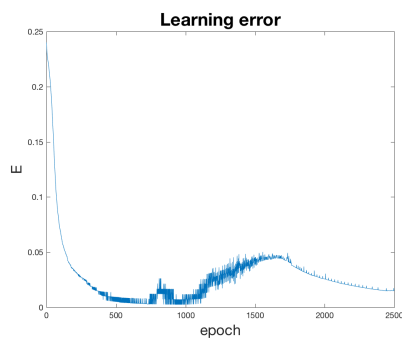
dataset: katy-ram-dataset(D.1)

normalizace: interval [0;0.5]

počet epoch: 500

learning rate: 0.1

E.2 Sex Pistols a The Offspring



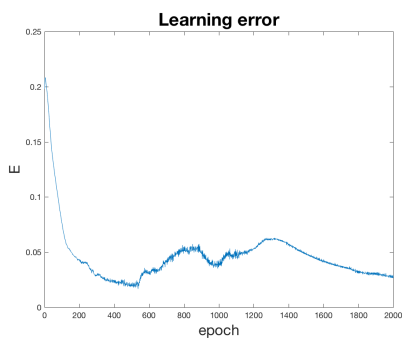
dataset: off-sexp-dataset(D.2)

normalizace: interval [0;0.4]

počet epoch: 2500

learning rate: 0.1

E.3 David Bowie, Elvis Presley a AC/DC



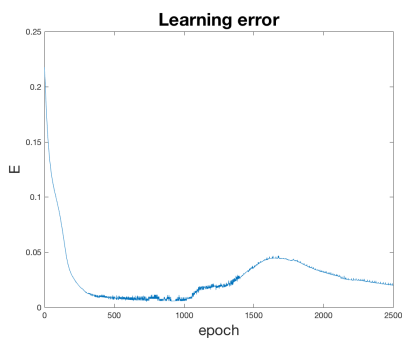
dataset: off-sexp-dataset(D.3)

normalizace: interval [0; 0.5]

počet epoch: 2000

learning rate: 0.1

E.4 Metallica, Maddona a Louis Armstrong



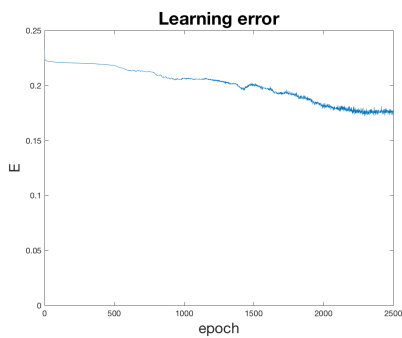
dataset: met-mad-la-dataset(D.4)

normalizace: interval [0; 0.5]

počet epoch: 2500

learning rate: 0.05

E.5 Metallica, Maddona a Louis Armstrong náhodně



dataset:

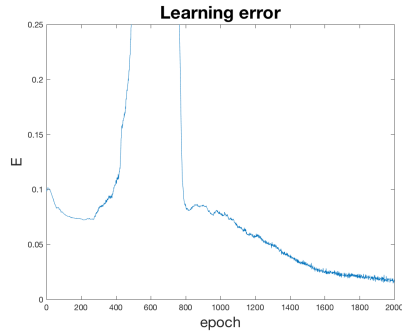
met-mad-la-random-dataset(D.5)

normalizace: interval [0; 0.5]

počet epoch: 2500

learning rate: 0.05

E.6 Skladby z roku 1960 a 1970



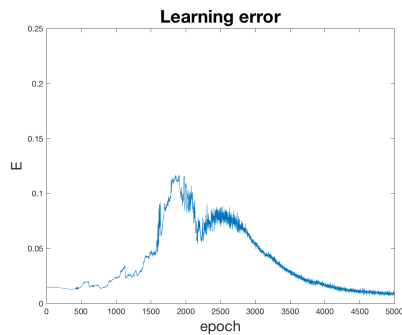
dataset: 60-vs-70-dataset(D.6)

normalizace: interval [0;0.5]

počet epoch: 2000

learning rate: 0.05

E.7 Skladby z roku 1970 a 2010



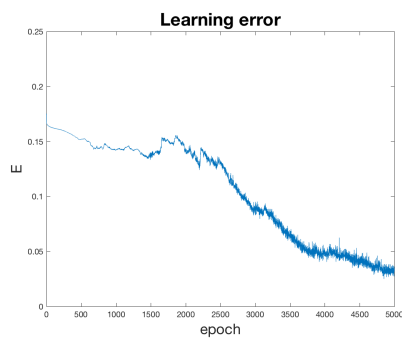
dataset: 70-vs-10-dataset(D.7)

normalizace: interval [0;0.5]

počet epoch: 5000

learning rate: 0.05

E.8 Uživatelská historie 3 třídy



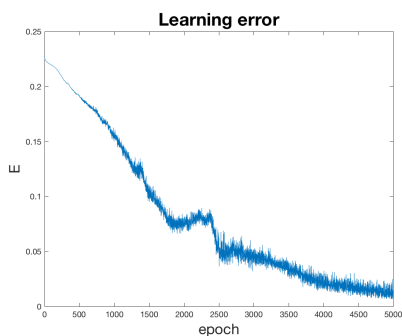
dataset: user-3class-dataset(D.8)

normalizace: interval [0;0.5]

počet epoch: 5000

learning rate: 0.1

E.9 Uživatelská historie 2 třídy



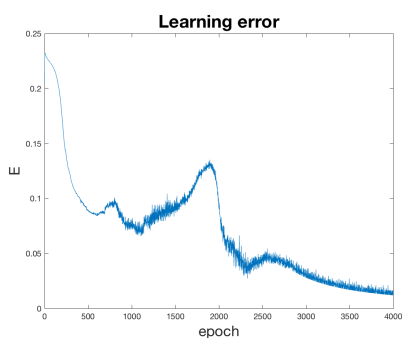
dataset: user-2class-dataset(D.9)

normalizace: interval [0;0.5]

počet epoch: 5000

learning rate: 0.1

E.10 Oblíbenost skladby



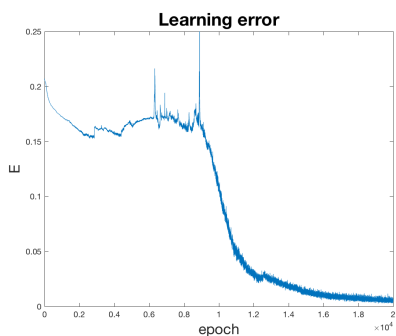
dataset: hot-fail-dataset(D.10)

normalizace: interval [0;0.5]

počet epoch: 4000

learning rate: 0.1

E.11 Oblíbenost skladby



dataset:

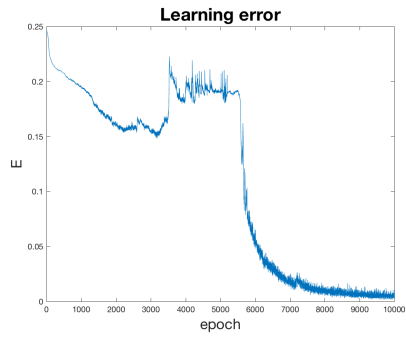
hit-normal-flop-dataset(D.11)

normalizace: interval [0;0.5]

počet epoch: 10000

learning rate: 0.1

E.12 Oblíbenost skladby



dataset: hit-fail-dataset2(D.12)

normalizace: interval [0; 0.5]

počet epoch: 10000

learning rate: 0.1