

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Cizlová Kateřina

Studijní program: Otevřená informatika
Obor: Softwarové systémy

Název tématu: Využití neuronových sítí v oblasti umělé kreativity v hudbě

Pokyny pro vypracování:

Seznamte se se současnými trendy v umělé inteligenci v oblasti tzv. umělé kreativity, kdy část práce hudebního skladatele je přenesena 'na stroj'. Seznamte se s konkrétními návrhy neuronových sítí, které dle předlohy generují melodie či zvukový signál obsahující hudbu. Vybranou techniku realizujte pomocí běžně dostupných frameworků pro práci s neuronovými sítěmi. Cílem necht' je aplikace, ve které uživatel zadá cílovou délku skladby, popř. další charakteristiky, a aplikace vygeneruje odpovídající zvukový signál obsahující hudbu. Aplikaci podrobte příslušným uživatelským testům.

Seznam odborné literatury:

[1] David Temperley: The Cognition of Basic Musical Structures. MIT Press.

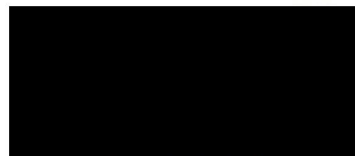
Vedoucí: Ing. Adam Sporka, Ph.D.

Platnost zadání do konce letního semestru 2017/2018



prof. Dr. Michal Pěchouček, MSc.

vedoucí katedry



prof. Ing. Pavel Rípků, CSc.

děkan

V Praze dne 20.2.2017

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Využití neuronových sítí v oblasti umělé kreativity v hudbě

Kateřina Cízlová

Vedoucí: Ing. Adam Sporka Ph.D.
Obor: Otevřená informatika
Studijní program: Softwarové systémy
Květen 2017

Poděkování

Ráda bych poděkovala svému vedoucímu práce Adamu Sporkovi za odborné vedení, cenné rady a vstřícnost při vypracovávání této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně, a že jsem uvedla veškerou použitou literaturu.

V Praze, 21. května 2017

Abstrakt

Tato práce se zabývá tvorbou hudby, kdy je část práce hudebního skladatele přenesena na stroj. Zaměřuje se především na tvorbu hudby pomocí neuronových sítí. Cílem práce je zjistit, jaké jsou současné trendy v této oblasti, seznámit se s návrhy neuronových sítí a nakonec pomocí dostupných frameworků pro práci s neuronovými sítěmi vytvořit webovou aplikaci, která bude po zadání parametrů schopna vytvořit zvukový signál obsahující hudbu. Po vytvoření je aplikace podrobena uživatelskému testování kvůli odhalení případných nedostatků.

Klíčová slova: neuronová síť, zvuk, hudba, harmonie, MIDI, webová aplikace, php, java

Abstract

This work deals with the creation of music, where part of the work of the composer is transferred to the machine. It focuses primarily on creating music using neural networks. The aim of the thesis is to find out the current trends in this field, to get acquainted with neural network designs and finally to use the available frameworks for working with neural networks to create a web application that will be able to create a sound signal containing music, after the parameters are entered. Once created, the application is subjected to user testing to detect any deficiencies.

Keywords: neural network, sound, music, harmony, MIDI, web application, php, java

Obsah

Zadání práce	1	5.4 Uživatelské rozhraní aplikace ...	24
1 Úvod	1	6 Testování	27
2 Rešerše	4	6.1 Úvod	27
3 Neuronové sítě	7	6.2 Podmínky pro testování	28
3.1 Druhy neuronových sítí	7	6.3 Use cases	28
3.2 Použití neuronových sítí	10	6.4 Průběh testování	31
4 Návrh	12	6.5 Závěr	34
4.1 Data	12	7 Úprava aplikace po testování	35
4.2 Neuronová síť	14	8 Závěr	38
4.3 Struktura skladby	14	A Uživatelská příručka	39
4.4 Převod dat do MIDI souborů ...	15	B Obsah CD	40
4.5 Uživatelské rozhraní aplikace ...	15	C Literatura	41
5 Realizace	16		
5.1 Data	17		
5.2 Neuronová síť	21		
5.3 Převod dat do MIDI souborů ...	22		

Obrázky

3.1 Rekurentní neuronová síť se smyčkou [Ola15]	8	5.8 Webová aplikace, záložka SETTINGS	26
3.2 Opakující se modul v RNN [Ola15] 9		5.9 Webová aplikace, záložka SONG 26	
3.3 Opakující se modul v LSTM [Ola15]	9	6.1 Úvodní strana aplikace s označenou položkou SETTINGS v menu.....	29
3.4 Neuron [maRN96]	10	6.2 Úvodní strana aplikace s označenou položkou SETTINGS v textu.....	29
3.5 Sigmoid	11	6.3 Sekce SETTINGS s označeným posuvníkem pro nastavení délky skladby.....	30
3.6 Hyperbolický tangens	11	6.4 Sekce SETTINGS s označeným polem pro název skladby.....	30
5.1 Práce se stopami písně v programu Reaper	20	6.5 Sekce SETTINGS s označeným tlačítkem pro vygenerování skladby 30	
5.2 Práce s notami písně v programu Reaper	20	6.6 Strana pro stáhnutí vygenerované skladby.....	31
5.3 Diagram předzpracování dat určených pro trénování sítě.....	21	7.1 Změna názvu tlačítka EXPORT v sekci SETTINGS na ‚GENERATE‘ 35	
5.4 Diagram vygenerování jedné části skladby	22	7.2 Přidání nápisu „generating...“ pod tlačítko GENERATE v sekci SETTINGS	36
5.5 Diagram převodu jednotlivých částí do MIDI souborů	24	7.3 Tlačítko pro stáhnutí písně v sekci SONG.....	36
5.6 Diagram dokončení skladby spojením jejích částí	24	7.4 Přhrávač bez možnosti stáhnutí písně v sekci SONG	36
5.7 Webová aplikace, záložka INTRODUCTION	25		

7.5 Odkazy v textu v části SETTINGS tučným fontem	36
7.6 Odkaz v textu v části SONG tučným fontem	37

Tabulky

2.1 Markovův řetězec prvního řádu .. 5

5.1 Seznam skladeb 19



Kapitola 1

Úvod

Cílem této práce je zjistit, jaké jsou současné trendy v umělé inteligenci v oblasti takzvané umělé kreativity, kdy část práce hudebního skladatele je přenesena na stroj. Dále pak se seznámím s konkrétními návrhy neuronových sítí, které dle předlohy generují melodie, či zvukový signál obsahující hudbu. Praktická část práce si klade za cíl vytvořit webovou aplikaci, ve které uživatel zadá délku, tempo, strukturu a jméno skladby, a aplikace vygeneruje odpovídající zvukový signál obsahující hudbu.

Toto téma a obsah jsem si vybrala, protože se zajímám o hudbu jako takovou a také o technické trendy s hudbou související. V práci se chystám nejdříve teoreticky popsat některá základní fakta z oboru hudby, poté zmínit metody tvorby hudby pomocí informačních technologií a nakonec vybranou techniku realizovat pomocí dostupných frameworků pro práci s neuronovými sítěmi a vytvořit webovou aplikaci, která bude schopná generovat hudbu na základě kritérií zadaných uživatelem.

V první části popíši některé pojmy nezbytné k pochopení práce a zároveň udělám průzkum současných technik generování hudby pomocí umělé inteligence. V další části popíšu konkrétněji návrhy některých neuronových sítí. Poté vytvořím návrh konečné aplikace. Práci si rozdělím na podproblémy, které detailněji rozeberu a popíši. V následující části bude rozebrána realizace. Stručně popíši, jak jsem problémy z popsané v kapitole Návrh řešila a nastíním, co dělají nejdůležitější třídy a skripty. Až bude práce hotová, podrobím jí uživatelskému testování, které mi pomůže odhalit případné nedostatky aplikace. V další kapitole shrnu případné problémy a popíši, jak jsem postupovala při jejich opravě.

■ Hudební terminologie

Zde jsou vysvětleny klíčové pojmy, důležité k pochopení práce.

Hudba. Hudba je organizované seskupení zvuků, převážně tónů (o určité výšce, délce, síle a barvě). Je ji možné chápat jako posloupnost not nebo signál o určitých frekvencích. V této práci bude hudba chápána jako posloupnost not.

Definice zvuku. Zvuk je mechanické vlnění v látkovém prostředí, které je vnímáno sluchem. Člověk je schopen vnímat frekvence v rozsahu přibližně od 16 Hz do 16 kHz. Mechanické vlnění s frekvencí nižší než 16 Hz se nazývá infrazvuk, který používají k dorozumívání například velryby nebo sloni. Naopak vlnění s frekvencí vyšší než 16 kHz má ultrazvuk, který jsou slyší například delfíni nebo netopýři. [Lep08]

Definice tónu. Zvuky, jejichž grafem závislosti intenzity na čase je periodická funkce, jsou považovány za tóny. V hudbě je tón základním stavebním kamenem. Základní vlastnosti tónu jsou:

- výška – určena především jeho frekvencí,
- délka – jak dlouho tón zní,
- síla – určena amplitudou,
- barva – závisí na hudebním nástroji, který tón vydává. [Lep08]

Grafická značka pro tón je nota. Zapisuje se do notové osnovy a její umístění v notové osnově určuje výšku tónu, její tvar pak označuje délku trvání tónu.

Stupnice. Stupnice je posloupnost tónů, která se řídí určitými pravidly. Každou stupnici tvoří 7 tónů. Dají se rozdělit na hlavní dvě skupiny a to na stupnice durové a mollové. [ka99] Tato práce se bude zabývat pouze stupnicemi durovými. Za výchozí stupnici se považuje stupnice C-dur. Ta obsahuje pouze celé tóny, tedy C, D, E, F, G, A, H.

Harmonie. Harmonie je soulad, rovnováha, či vyrovnanost. [Ola15] V hudbě lze harmonii chápat jako souzvuk tónů. Obvykle je založena na akordech. [SJB]

Akord je souzvuk minimálně třech tónů, které spolu nějakým způsobem souvisí. Tato práce se bude zabývat základními akordy, které se skládají ze třech tónů. Základní notou v akordu je nota ze všech tří not nejhlubší a zároveň nota, po které je akord pojmenován, druhá nota je nota, která je o dva celé tóny vyšší a poslední nota je o čtyři celé tóny vyšší než základní nota. Tedy například akord C-dur, bude tvořit nota C, E a G. [SJe]

Jak již bylo řečeno, každou stupnici tvoří sedm tónů. Akordy používající se v dané stupnici, obvykle obsahují jen noty z té stupnice, a zároveň každá nota ze stupnice je základní notou jednoho akordu, tedy akordy se i jmenují podle not ve stupnici. U každé stupnice tedy lze vytvořit sedm základních durových akordů. Těchto sedm akordů je zvykem pojmenovávat římskými číslicemi. Tedy například pro stupnici C-dur:

- i: C-dur
- ii: D-dur
- iii: E-dur
- iv: F-dur
- v: G-dur
- vi: A-dur
- vii: H-dur

Jednotlivé akordy ve stupnici mají své názvy podle harmonických souvislostí. V této práci se budou využívat jen tři, díky kterým lze doprovodit prakticky jakoukoliv píseň:

- i: tónika
- iv: subdominanta
- v: dominanta [SJa]

Kapitola 2

Rešerše

Tato část se zabývá současnými trendy v umělé inteligenci v oblasti takzvané umělé kreativity, kdy část práce hudebního skladatele je přenesena na stroj.

■ Celulární automat

Celulární automat je soubor „barevných“ buněk v mřížce specifického tvaru, který se vyvíjí prostřednictvím řady diskretních časových kroků podle souboru pravidel na základě stavů sousedních buněk. Hlavní myšlenkou použití celulárního automatu ke tvorbě hudby je udělat pás přes vzor a otočit ho o 90 stupňů. Jakmile je celulární vzor nakloněn na bok, výška každého černého čtverce odpovídá nějaké notě. Speciální mapování odpovídá hudebnímu měřítku, které bylo použito. Každá stupnice vybere některé z dvanácti standardních tónů v oktávě. [WR05]

■ Genetické algoritmy

Genetické algoritmy patří mezi takzvané evoluční algoritmy. Tyto algoritmy používají principy přirozeného výběru, aby vyprodukovaly několik řešení daného problému. Jsou založeny na Darwinově teorii, tedy uvažují, že jedinci s určitými vlastnostmi mají větší šanci na přežití, než jiní. Důležité je vědět, co by mělo výsledné řešení dělat nejlépe. Každý jedinec představuje možné řešení

daného problému a jeho kvalita je vyjádřitelná pomocí takzvané ohodnocovací funkce. Genetický algoritmus tedy hledá taková řešení, pro která bude tato ohodnocovací funkce vycházet nejlépe. [Mat]

V případě generování hudby pomocí genetických algoritmů proces začíná populací jedinců, kteří produkují zvuk, který je buď vytvořen náhodně nebo na základě hudby vygenerované člověkem. Cílem je opakovanou aplikací genetického algoritmu produkovat stále lépe znějící hudbu. Ohodnocovací funkce určuje jak dobře melodie zní. Rozhoduje se například podle toho jestli noty nemají větší rozsah než dvě oktávy nebo jestli mezi notami nejsou příliš dlouhé pauzy. [Cra09]

■ Markovovy řetězce

Markovův řetězec popisuje obvykle diskrétní náhodný (stochastický či pravděpodobnostní) proces, pro který platí, že pravděpodobnost budoucích stavů je plně určena současným stavem, bez ohledu na stavy minulé. Tato vlastnost dovoluje proces znázornit stavovým diagramem, v němž z každého stavu vycházejí hrany možných přechodů očíslované možnou pravděpodobností přechodu do dalšího stavu. [Nav16]

Markovovy řetězce popisují, jak se dostat z jedné události na druhou. Pokud máme tři noty C, D, Eb, které mohou přijít v libovolném pořadí, Markovův řetězec bude určovat pravděpodobnost, jaká bude další nota. Následující tabulka (2.1) ukazuje Markovův řetěz prvního řádu. Existují tři možné stavy, ve kterých může být. Pro každou možnost současného stavu, existují tři další možné noty. Každý řádek v tabulce udává relativní pravděpodobnost, se kterou se přejde na další notu. Například, pokud se v současné době nacházíme na C, existuje 20% pravděpodobnost, že se C bude opakovat, 50% šance, že se přejde do D a 30% šance do Eb. Součet pravděpodobností v každém řádku se rovná 100 %. [Sap03]

Současná nota	Další nota	Další nota	Další nota
	C	D	Eb
C	20%	50%	30%
D	35%	25%	40%
Eb	70%	14%	16%

Tabulka 2.1: Markovův řetězec prvního řádu

■ Deep learning

Deep learning neboli takzvané hluboké učení spadá do oblasti strojového učení. Strojové učení se zabývá algoritmy a technikami, které umožňují počítačovému systému se samostatně učit něco nového bez přímého řízení programem. [Mat13]

Generování hudby pomocí strojového učení se dá rozdělit do dvou hlavních kategorií, na základě vstupu.

Sekvence not. Na vstupu je soubor se sekvencí not, ze které se stroj učí. Výpočetní složitost je poměrně nízká (minuty až pár hodin). Může být realizováno rekurentními neuronovými sítěmi a vrstvami LSTM neuronových sítí. [Bri16]

Raw audio. Na vstupu je raw audio, neboli nekomprimovaný zvuk v syrové formě. Výpočetní složitost je vysoká (několik hodin až dnů). Může být realizováno konvolučními neuronovými sítěmi. [Bri16]

Kapitola 3

Neuronové sítě

3.1 Druhy neuronových sítí

Neuronová síť je struktura určená pro distribuované paralelní zpracování dat, je jedním z výpočetních modelů používaných v umělé inteligenci a je inspirována chováním biologické nervové soustavy.

Skládá se z formálních neuronů, které jsou rozloženy ve vrstvách a spojeny synapsemi. Určitý tvar neuronové sítě, je definován počtem neuronů v jednotlivých vrstvách a jednotlivými spojeními mezi nimi. Neurony si navzájem předávají signály a transformují je pomocí určitých přenosových funkcí. Neuron má libovolný počet vstupů, ale pouze jeden výstup. [aDS]

Hlavní dva charakteristiky určující neuronovou síť jsou architektura (uspořádání jednotlivých neuronů a propojení mezi nimi) a algoritmus učení (způsob, jakým se nastavují váhy ve spojení mezi neurony). [Pan06]

Existuje mnoho různých druhů neuronových sítí. Lze je rozdělit na dvě základní skupiny – dopředné a rekurentní. Pro použití neuronových sítí k účelu vytváření hudby jsou nejvýhodnější Gated recurrent units (GRU), konvoluční neuronové sítě (CNN) a rekurentní neuronové sítě (RNN), do kterých patří LSTM (Long Short-Term Memory) síť. [Bri16, Hla]

■ Dopředné neuronové sítě

Neuronové sítě mohou být propojeny v libovolné struktuře. Pokud se v grafu sítě propojených neuronů nevyskytuje cyklus, jedná se o takzvané dopředné sítě. Tyto sítě vyhodnocují každý vstup zvlášť, bez ohledu na výstupy ostatních. [Pan06]

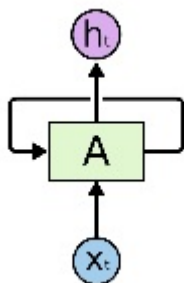
Konvoluční neuronové sítě. Konvoluční neuronové sítě patří do kategorie dopředných neuronových sítí. Tyto sítě bývají hluboké. Propojení neuronů je inspirováno organizací zrakové kůry u zvířat. Tyto sítě se ale spíš hodí například pro klasifikaci obrázků. Trénování je výpočetně náročné, velké sítě se trénují klidně jeden měsíc. [Tea]

■ Rekurentní neuronové sítě

Propojení dopředných neuronových sítí je pro tvoření hudby celkem nevyhovující, ani lidé nezačínají přemýšlet každý okamžik znovu od začátku, naopak chápou nějakou událost, text nebo i hudbu jako celek. Ani hudební skladatel nepřemýšlí nad notami nezávisle, ale vytváří vždy celou skladbu nebo její části.

Opakem dopředné sítě je rekurentní neuronová síť (Recurrent neural network - RNN). Rekurentní neuronové sítě obsahují smyčky, které jim umožňují uchovávat informace.

Na obrázku 3.1 je zobrazena část rekurentní neuronové sítě A , která má vstup x_t a výstup h_t . Smyčka dovoluje, aby byla informace přenesena z jednoho kroku na druhý. [Ola15]

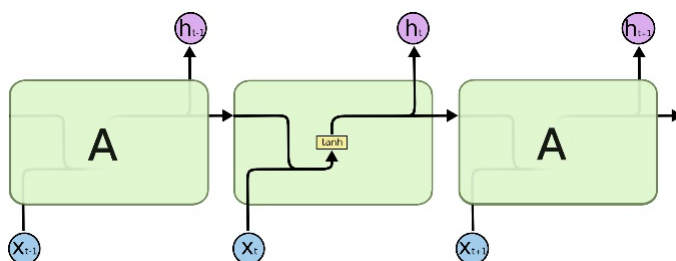


Obrázek 3.1: Rekurentní neuronová síť se smyčkou [Ola15]

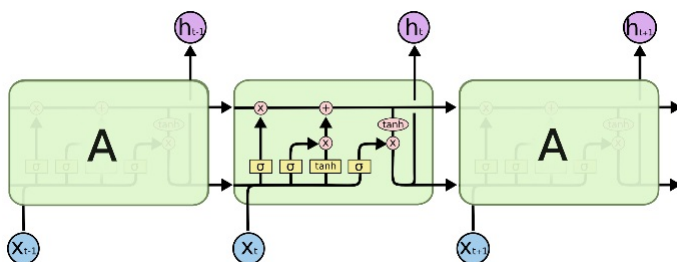
LSTM neuronové síť. V některých případech se k vykonání současného úkolu stačí podívat jen na nedávnou informaci, například u rozpoznávání řeči si stačí zapamatovat slova v jedné větě, aby se dalo předpovědět poslední slovo, není tedy nutné zapamatovávat si předešlé informace. Ale existují případy, kdy je nutné si zapamatovat celý kontext. Teoreticky si standardní rekurentní síť mohou zapamatovat dlouhodobé informace, ale v praxi by toho pravděpodobně nebyly schopné.

Long Short-Term Memory neuronové síť, zkráceně LSTM, jsou speciální druh rekurentních neuronových sítí, schopné zapamatovat si dlouhodobé informace a závislosti.

Všechny rekurentní neuronové síť obsahují řetězec opakujících se modulů neuronových sítí. Ve standardních rekurentních neuronových sítí bude tato struktura velmi jednoduchá (Obrázek 3.2). U LSTM neuronové sítí budou mít tyto modely složitější strukturu (Obrázek 3.3). Každé spojení zde nese celý vektor informací, z výstupu jednoho neuronu do vstupů ostatních.



Obrázek 3.2: Opakující se modul v RNN [Ola15]



Obrázek 3.3: Opakující se modul v LSTM [Ola15]

Klíčem LSTM sítí je propojení cell state, které je vidět v horní části diagramu. Prochází celým řetězcem jen s pár základními lineárními interakcemi. Pro informaci je tedy jednoduché jím projít beze změny. v LSTM sítích lze přidat nebo smazat informaci z tohoto propojení pomocí takzvaných bran. Tyto brány mají možnost nechat informaci projít propojením cell state, jejich výstup je číslo mezi nulou a jedničkou, které určují, jak velké množství informací má projít (0 – žádné informace, 1 – veškeré informace). [Ola15]

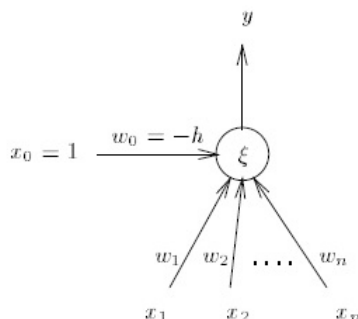
Gated recurrent units. Gated recurrent units jsou bránové mechanismy rekurentních neuronových sítí. Jejich způsob vytváření muziky nebo modelování řeči je podobný jako u LSTM. V různých aspektech jsou ale rozdílné. Například spojují stav buněk se skrytou vrstvou. Výsledný model je jednodušší než standardní modely LSTM. [Ola15]

3.2 Použití neuronových sítí

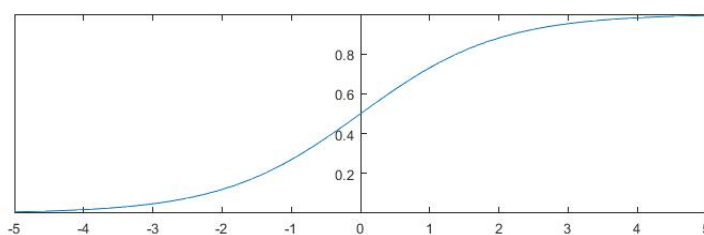
Struktura

Základním prvkem neuronové sítě je formální neuron.

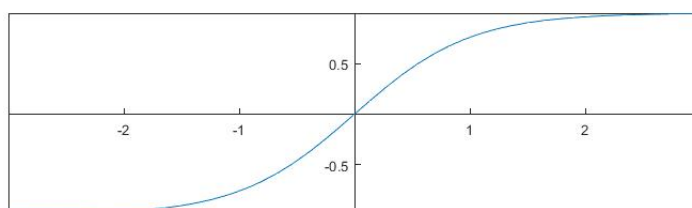
Vstupy neuronu jsou označeny jako x_1, x_2, \dots, x_n . x_0 je speciální vstup, který má vždy hodnotu 0. w_1, w_2, \dots, w_n jsou reálné váhy. ε je vnitřní potenciál, který se vypočítá podle vzorce $\varepsilon = \sum_{i=1}^n w_i \cdot x_i$. y je výstup daný vzorcem $y = \sigma(\varepsilon)$, kde σ je aktivační (přenosová) funkce. Aktivační funkce nemusí být zvolena vždy stejně, měla by však být nelineární. Nejjednodušším typem aktivační funkce je ostrá nelinearita $\left(y = \begin{cases} 1, & \varepsilon > h \\ 0, & \varepsilon < h \end{cases}\right)$, aktivační funkcí ale také může být funkce sigmoid $\left(y = \frac{1}{1+e^{-x}}\right)$, jehož graf je znázorněný na obrázku 3.5 nebo hyperbolický tangens $\left(y = \frac{1-e^{-2x}}{1+e^{-2x}}\right)$ jehož graf je znázorněný na obrázku 3.6. [maRN96]



Obrázek 3.4: Neuron [maRN96]



Obrázek 3.5: Sigmoid



Obrázek 3.6: Hyperbolický tangens

■ Učení neuronové sítě

Cílem učení je nastavit váhy neuronové sítě tak, aby její výstupy byly co nejpřesnější. Existují dva hlavní typy učení.

Učení s učitelem. Při učení s učitelem má algoritmus k dispozici konečnou množinu dvojic, které představují vstupy a odpovídající výstupy dané úlohy. Tato množina tedy představuje správné chování sítě, kterého chceme při učení dosáhnout. Obvykle bývá rozdělena na trénovací a testovací část.

Při samotném trénování se neuronové sítě předkládají vstupy z množiny trénovacích dat a po zjištění výstupu sítě, se postupně upravují váhy nastavené mezi neurony, aby byla odchylka výstupu sítě od korektního výstupu ze zadané množiny dat co nejmenší. Tento postup opakujeme tak dlouho, dokud odchylka nedosáhne požadované malé hodnoty nebo po zastavením jejího poklesu. Poté ověříme síť na testovací množině. [Bla]

Učení bez učitele. Při učení bez učitele neuronová síť nemá k dispozici žádnou množinu vstupů a odpovídajících korektních výstupů jako tomu bylo u učení s učitelem. Hledání správných výsledků probíhá na principu shlukování, kdy se mezi vstupními daty hledají navzájem podobné elementy. [Bla]

Kapitola 4

Návrh

V této kapitole je popsán návrh výsledné aplikace.

K vytvoření aplikace bude použita neuronová síť. Z rešerše vyplývá, že v tomto případě bude nejvýhodnější použít rekurentní neuronovou síť. Dále bude potřeba množina dat, podle kterých se bude síť učit. Ta se získá převedením již existujících skladeb na textové řetězce, které poté budou sloužit jako trénovací množina. Až bude množinu dat kompletní, předloží se síti a ta se začne učit. Poté by měla být síť schopná generovat řetězce podobné těm, které byly v množině dat určené k trénování. Tyto vygenerované textové řetězce budou převedeny do souborů formátu MIDI, obsahující tři stopy: melodii a basovou a harmonickou linku. Potom bude přidána čtvrtá stopa obsahující rytmickou linku. Nakonec bude vytvořeno uživatelské rozhraní webové aplikace, ve kterém si bude možné skladbu přehrát a stáhnout.

4.1 Data

Množina dat pro trénování sítě. Jako data, podle kterých se bude neuronová síť učit, bude použito 50 písní v MIDI formátu. Budou vybírány písně ve čtyřčtvrtovém taktu u kterých je zřetelná souvislá melodie, ta jediná totiž bude potřebná pro trénování sítě. Další kritérium bude výškový rozsah písní, budou hledány takové písně, jejichž melodie po transpozici do tóniny C-dur nebude obsahovat noty nižší než nota C4 a vyšší než nota C6. Pak také bude rozhodovat délka not, je nutné vybírat jen písně, jejichž délky not budou násobky noty šestnáctinové. Skladby splňující tato kritéria budou staženy z dostupných internetových stránek.

Předpracování dat. Každá stažená skladba bude upravena tak, aby obsahovala jen jednu stopu, a to stopu obsahující melodii. Ta je totiž potřebná k další práci. Poté bude píseň transponována do tóniny C-dur. Tento krok je důležitý pro jednodušší přiřazování akordů k melodiím a pro jednodušší a přesnější učení sítě.

Dále je nutné ke každé melodii přiřadit akordy, bude se volit pouze mezi tónikou, dominantou a subdominantou. Jelikož jsou všechny písně v tónině C-dur, jsou to tedy akordy C-dur, G-dur a F-dur. Délka jednoho akordu bude maximálně jedna čtvrtina taktu.

Zakódování dat pro síť . Melodie budou zakódovány do textových řetězců, které budou pro neuronovou síť čitelné, a bude schopná se z nich učit. Každá píseň bude pomyslně rozdělena na části dlouhé 1/16 taktu. Tyto části poté budou převedeny do textového řetězce, přičemž každá část bude reprezentována čtyřmi znaky.

První znak bude určovat výšku noty. Noty se pohybují v rozmezí C4 - C6, každé notě tedy bude přiřazeno jedno velké písmeno podle její výšky. Pro notu C4 to bude písmeno ‚A‘, pro C_{is}4 písmeno ‚B‘, pro D4 písmeno ‚C‘, a tak dále. Pro notu C6 to pak bude písmeno ‚Y‘ a nakonec písmeno ‚Z‘ bude označovat pauzu.

Druhý znak bude určovat pořadí části dlouhé 1/16 taktu v taktu. Každé části tedy bude přiřazeno jedno malé písmeno podle toho, na jaké pozici v taktu se vyskytuje. Pokud se vyskytuje v první šestnáctině, bude přiřazeno písmeno ‚a‘, pokud ve druhé, písmeno ‚b‘, a tak dále, až nakonec části nacházející se v poslední šestnáctině taktu bude přiřazeno písmeno ‚p‘.

Třetí znak bude znázorňovat akord hodící se k této části. Pro znázornění akordů bude použito číslo od jedné do tří. Pro akord F-dur číslo 1, pro C-dur 2 a pro G-dur 3. Protože délka jednoho akordu je maximálně 1/4 taktu, bude toto číslo vždy pro čtyři po sobě jdoucí části stejné.

Poslední znak bude určovat, jestli nota na konci části končí a v navazující části začíná nota nová nebo stále pokračuje do následující části. Použijeme dva znaky: pomlčku (, -) pro noty které v navazující části pokračují a tečku (, .) pro noty které na konci této části končí.

Například pro takt, kde budou za sebou čtyři čtvrtinové noty C4, E4, G4, C4 a akord C-dur trvajícím po celou dobu taktu bude textový řetězec vypadat následovně: „Aa2-Ab2-Ac2-Ad2.Ee2-Ef2-Eg2-Eh2.Hi2-Hj2-Hk2-H2.Am2-An2-Ao2-Ap2.“

4.2 Neuronová síť

Trénování sítě. Všechny padesát textových řetězců představující jednotlivé melodie s akordy budou vloženy za sebe do textového souboru, který bude následně předložen neuronové síti, která se bude snažit pochopit souvislosti v řetězcích a naučit se generovat vlastní podobné řetězce.

Generování dat. Uživatel si bude moci zvolit strukturu písně, respektive uspořádání jednotlivých částí jako je sloka, refrén, přechod a zakončení. Pro každou část písně budou náhodně vybrány první dva znaky (velké písmeno v rozmezí A - Z a malé písmeno v rozmezí a - p), které poté budou předloženy síti. Dále bude také zvolena délka úměrná zadané délce skladby od uživatele. Data, respektive textový řetězec vygenerovaný sítí, budou zkontrolovány a upraveny tak, aby v nich nebyly odchylky (například pokud neobsahují po sobě jdoucí výše popsané čtyři znaky ve správném pořadí). Tato úprava se dělá kvůli snadnému následnému převodu do MIDI souboru. Pak také bude délka řetězce upravena tak, aby po spojení všech částí písně a převodu do MIDI souboru byla skladba tak dlouhá, jak si uživatel zvolil.

4.3 Struktura skladby

Jak již bylo zmíněno, uživatel si bude moci vybrat strukturu písně. Bude si moci vybrat z těchto pěti možností:

1. sloka, zakončení
2. sloka, refrén, sloka, zakončení
3. sloka, refrén, přechod, sloka, zakončení
4. sloka, refrén, sloka, refrén, přechod, sloka, zakončení
5. sloka, refrén, sloka, přechod, sloka, refrén, sloka, zakončení

Pro každou část písně se vygeneruje originální textový řetězec. Délky jednotlivých částí nebudou stejné. Zde jsou vypsány poměry délek částí (kromě zakončující části) jednotlivých struktur, ve smyslu sloka:refrén:zakončení.

1. 1:0:0
2. 3:2:0
3. 3:2:1
4. 3:2:1
5. 3:2:1 pokud je píseň delší než 17 taktů, pokud je kratší, poměr bude 2:2:1

4.4 Převod dat do MIDI souborů

Vygenerované textové řetězce reprezentující části písně budou převedeny do MIDI souborů, které budou obsahovat tři stopy: melodii a basovou a harmonickou linku. Melodie budou vytvořeny podle velkých písmen a teček nebo pomlček v textovém řetězci. Harmonie a basová linka bude vytvořena podle čísel obsažených v řetězci. Přičemž pro harmonii budou použity tóny z oktávy začínající tónem C4 a pro basovou linku bude použita vždy nota podle které je pojmenován akord. Noty budou vybrány z oktávy začínající tónem C3. Nakonec bude přidána čtvrtá stopa, která bude obsahovat rytmickou linku. Takto budou získány soubory ve formátu MIDI pro jednotlivé části písně. Ty poté budou spojeny podle kritérií uživatele do jednoho souboru obsahujícího celou píseň zadané délky.

4.5 Uživatelské rozhraní aplikace

Jako poslední část práce bude vytvořeno uživatelské rozhraní aplikace. Webová stránka bude mít tři sekce: úvod, nastavení a sekci pro stáhnutí skladby. V úvodu bude popsána aplikace a zároveň bude uživateli vysvětleno, jak by měl při generování skladby postupovat. V sekci nastavení bude možnost nastavit délku skladby v rozmezí 1 až 5 minut, kdy nejmenší rozdíl mezi dvěma různými délkami písně bude jedna sekunda. Dále pak jestli má být píseň smutná nebo veselá. Také uspořádání slok, refrénů a přechodů v skladbě. A nakonec jméno skladby, které bude smět obsahovat jen čísla a malá a velká písmena bez diakritiky. V poslední sekci si uživatel bude moci skladbu přehrát nebo si ji stáhnout v požadovaném formátu. Na výběr budou formáty wav, MIDI a mp3.

Kapitola 5

Realizace

V této části je popsáno, jak byla aplikace realizována, jaké byly použity postupy a techniky.

■ Technické specifikace

- **REAPER:** Pro úpravy stažených skladeb ve formátu MIDI je použit program pro práci s hudebními soubory REAPER¹.
- **Java:** Pro převody textových řetězců do MIDI souborů a opačně jsou použity programy naprogramované v programovacím jazyku Java. Java² je univerzální objektově orientovaný programovací jazyk.
- **Neuronová síť:** V této práci je použita vícevrstvá rekurentní neuronová síť „char-rnn“, naimplementovaná v roce 2015 Andrejem Karpathym skládající se z LSTM, GRU a RNN [Kar15]. Tato síť slouží k trénování a následnému generování nějakého modelu na úrovni znaků. Neboli síť dostane na vstupu jeden textový soubor a postupným trénováním se naučí předvídat další znak v sekvenci. Kód sítě je naprogramován v programovacím jazyku Lua³ a ke spuštění je vyžadován Torch.

¹<http://www.reaper.fm/>

²<https://www.java.com/>

³<https://www.lua.org/>

- **Torch:** Ke spuštění neuronové sítě je vyžadován Torch⁴. Torch je vědecká počítačová platforma s širokou podporou algoritmů pro strojové učení.
- **php:** Pro tvorbu webových stránek je použito php⁵. Jedná se o populární skriptovací jazyk, který je vhodný pro vývoj webových aplikací. Verze php je 7.0.15.
- **JavaScript:** Pro tvorbu webových stránek je použit skriptovací jazyk JavaScript⁶

5.1 Data

V této podkapitole jsou popsány všechny operace týkající tvorby množiny dat, která slouží k trénování neuronové sítě. Na konci této podkapitoly se pak nachází diagram (Obrázek 5.3) shrnující postupy a použité třídy a programy při tvorbě množiny dat.

Množina dat pro trénování sítě. Jak již bylo popsáno v kapitole Návrh, bylo vybráno 50 písní v čtyřčtvrtovém taktu, v MIDI formátu, u kterých je zřetelná souvislá melodie, jejichž melodie po transpozici do tóniny C-dur neobsahuje noty nižší než C4 a vyšší než C6 a jejichž melodie mají délky not, které jsou násobky noty šestnáctinové.

Byl vybrán tento seznam skladeb:

- Lidové písně, u kterých není znám původní autor:
 - Drunken sailor
 - Tancuj, tancuj, vykrúcaj
 - Anička, dušička kde si bola
 - Půjdem spolu do Betléma
- Skladby, u nichž je znám autor:

NÁZEV	INTERPRET	ALBUM	ROK	AUTOŘI
Jingle Bells	-	-	1857	James Pierpont
All my loving	The Beatles	With The Beatles	1963	Lennon, McCartney

⁴<http://torch.ch/>

⁵<http://php.net/>

⁶<https://www.javascript.com/>

5. Realizace

12 days of christmas	-	-	1909	Frederic Austin
Radioactive	Imagine Dragons	Night Vision	2012	Grant, McKee, Mosser, Platzman, Reynolds, Sermon
Holubí dům	Jiří Schelinger	Holubí dům	1972	Uhlíř, Svěrák
Holky z naší školky	Stanislav Hložek, Petr Kotvald	Holky z naší školky	1983	Žák, Vágner
500 Miles	The Journeymen	-	1961	West
Dej mi víc své lásky	Olympic	-	1965	Janda, Chrastina
Boulevard of broken dreams	Green Day	American Idiot	2004	Armstrong, Dirnt, Cool
Honey honey	ABBA	Waterloo	1974	Andersson, Ulvaeus, Anderson
Yesterday	The Beatles	Help!	1965	Lennon, McCartney
When I come around	Green Day	Dookie	1994	Armstrong, Dirnt, Cool
Good riddance	Green Day	Nimrod	1997	Armstrong, Dirnt, Cool
21 guns	Green Day	21st Century Breakdown	2009	Armstrong, Dirnt, Phillips, Cool
Holiday	Green Day	American Idiot	2004	Armstrong, Dirnt, Cool
Don't let me get me	P!nk	Missundaztood	2001	Moore, Austin
All the small thing	Blink 182	Enema of the State	1999	DeLonge, Hoppus
What's my age again	Blink 182	Enema of the State	1999	DeLonge, Hoppus
Jasná zpráva	Olympic	Stejskání	1982	Janda, Vrba
My máma prima rodiče	Michal David	Non stop	1984	Janeček, Řebíček
Hvězdičko blýskavá	Petr Novák	Sladké trápení	1982	Novák, Krečmar
Stín katedrál	Václav Neckář, Helena Vondráčková	Václav Neckář zpívá pro mladé	1966	Svoboda, Fischer
Bob a Bobek	-	-	1979	Petr Skoumal
Pátá	Helena Vondráčková	Růže kvetou dál	1969	Hatch, Štáidl
Vím jedno návrší	Radek Valenta	-	1994	Bárta, Krkoška
Barbie girl	Aqua	Aquarium	1997	Rasted, Norreen, Dif
Levels	Avicii	-	2011	Bergling, Pournouri, Kirkland, Woods, James
Best song ever	One Direction	Midnight Memories	2013	Hector, Ryan, Drewett, Bunetta
Hello	Adele	25	2015	Adkins, Kurstin
What a wonderful world	Louis Armstrong	-	1967	Thiele, Weiss
Život je jen náhoda	Jan Werich, Jožka Srbová, M. Schlesingerová	-	1932	Ježek, Voskovec, Werich
Nebe na zemi	Jan Werich	-	1936	Ježek, Voskovec, Werich

Try	P!nk	The truth about love	2012	Busbee, West
Země vzdálená	BSP	BSP II	1994	Pavlíček, Baláz, Stříhavka
Welcome to paradise	Green Day	Dookie	1994	Armstrong, Dirnt, Cool
She	Green Day	Dookie	1994	Armstrong, Dirnt, Cool
Longview	Green Day	Dookie	1994	Armstrong, Dirnt, Cool
Tears In Heaven	Eric Clapton	Rush	1992	Clapton, Jennings
Let it grow	Eric Clapton	Ocean Boulevard	1974	Clapton
Perfect	P!nk	Greatest Hits... So Far!!!	2010	Moore, Martin, Shellback
Basket case	Green Day	Dookie	1994	Armstrong, Dirnt, Cool
American idiot	Green Day	American Idiot	2004	Armstrong, Dirnt, Hatch, Cool
Minority	Green Day	Warning	2000	Armstrong, Dirnt, Cool
Prosthetic head	Green Day	Nimrod	1997	
Waiting	Green Day	Warning	2000	Armstrong, Dirnt, Hatch, Cool
Hot N Cold	Katy Perry	One of the Boys	2008	Perry, Gottwald, Martin

Tabulka 5.1: Seznam skladeb

Skladby byly stahovány z převážně z těchto webových stránek:

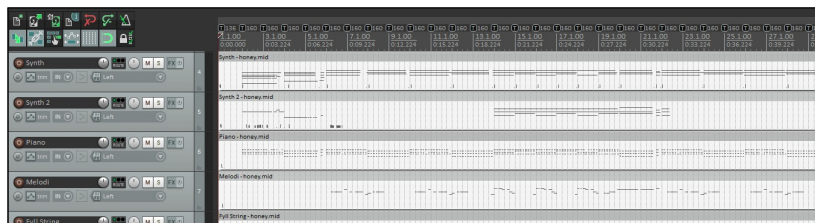
- <http://skladby.sweb.cz/>
- <http://www.midiworld.com/>
- <https://freemidi.org/>
- <http://jarva.wz.cz/hudba/midi.htm>

Předzpracování dat. V této části je nutné předzpracovat data tak, aby obsahovala jen stopu s melodií, poté transponovat píseň do tóniny C-dur a nakonec k melodiím přiřadit správné akordy.

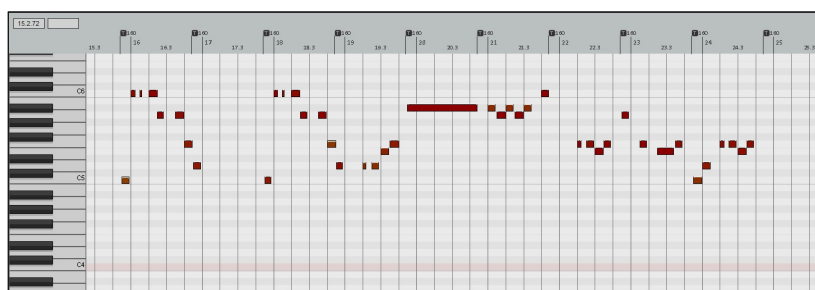
K těmto úpravám byl použit program pro práci s hudebními soubory REAPER. Skladba byla otevřena v programu (viz obrázek 5.1) a byly odstraněny všechny ostatní stopy kromě té obsahující melodii. Poté byla v módu pro úpravu jednotlivých not (viz obrázek 5.2) upravena délka not tak, aby každá nota byla násobkem noty šestnáctinové a melodie byla transponována do harmonie C-dur.

Poté bylo nutné k melodiím přiřadit akordy. Bylo voleno mezi akordy C-dur, F-dur a G-dur. Každý akord může trvat nejméně jednu čtvrtinu taktu.

Posloupnosti akordů byly zapsány do nových textových souborů jako čísla (1 pro F-dur, 2 pro C-dur a 3 pro G-dur) tak, že bylo zapsáno vždy jedno číslo pro jednu čtvrtinu taktu. Tedy pro každý takt jsou v souboru zapsaná čtyři čísla. Bylo tedy vytvořeno 50 nových textových souborů, které obsahují čísla představující akordy. Tato čísla poté budou v další části zakomponována do textových řetězců melodie.



Obrázek 5.1: Práce se stopami písně v programu Reaper



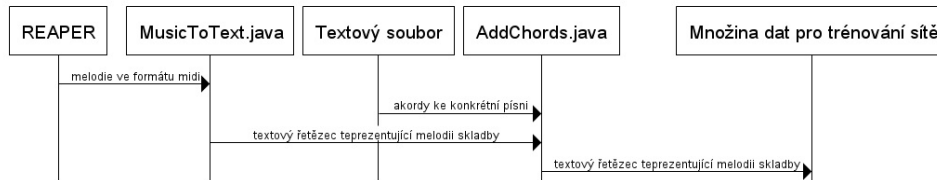
Obrázek 5.2: Práce s notami písně v programu Reaper

Zakódování dat pro síť . Zakódování melodie do textového řetězce, který bude pro neuronovou síť čitelný, zajistil program naprogramovaný v programovacím jazyku Java. Nejdříve byla melodie převedena na textový řetězec, který obsahuje znaky pro výšku noty, pořadí noty v taktu a znak určující jestli nota pokračuje nebo končí. Potom byly textové soubory obsahující čísla od jedné do tří reprezentující akordy spojeny se soubory obsahující textové řetězce reprezentující melodie a to tím způsobem, že každé číslo představující akord byl vždy vložen mezi malé písmeno určující pořadí v textu a znak určující jestli nota pokračuje nebo končí.

Převod melodie z MIDI souboru na textový řetězec umožňuje třída ‚MusicTo-Text.java‘ v balíčku ‚convert‘. Nejdříve byl v metodě ‚convertToArrayOfNotes‘ převeden MIDI soubor na typ ‚Sequence‘, ze kterého byly získány pro práci důležité informace o melodii, ty potom byly v metodě ‚dataToNetwork‘ převedeny na textový řetězec a uloženy do zadaného textového souboru.

Program pro spojení textových souborů je naprogramován ve třídě ‚Add-Chords.java‘ v balíčku ‚convert‘. Textový řetězec je zde vždy rozdělen mezi znakem určujícím pořadí části v taktu a znakem určujícím jestli nota v dané části končí nebo pokračuje v navazující části. Tedy mezi malým písmenem a

tečkou nebo pomlčkou. Do nového řetězce jsou pak postupně umístěny první dva znaky, tedy velké a malé písmeno, poté z druhého souboru číslo reprezentující akord a nakonec poslední znak z prvního souboru, tedy tečka nebo pomlčka. Výsledný textový řetězec je uložen do nového textového souboru.



Obrázek 5.3: Diagram předzpracování dat určených pro trénování sítě

5.2 Neuronová síť

V této podkapitole je popsáno trénování neuronové sítě a generování dat sítě. Na konci se nachází diagram (Obrázek 5.4) shrnující postupy a použité třídy při generování části skladby.

Trénování sítě. Jak bylo zmíněno v kapitole Neuronové sítě, pro použití neuronových sítí k účelu vytváření hudby jsou nejvhodnější Gated recurrent units (GRU), konvoluční neuronové sítě (CNN) a rekurentní neuronové sítě (RNN), do kterých patří LSTM (Long Short-Term Memory) sítě. Jako síť pro generování dat v rámci této aplikace byla použita vícevrstvá rekurentní neuronová síť skládající se z LSTM, GRU a RNN. Tato síť slouží k trénování a následnému generování nějakého modelu na úrovni znaků. Neboli síť dostane na vstupu jeden textový soubor a postupným trénováním se naučí předvídat další znak v sekvenci.

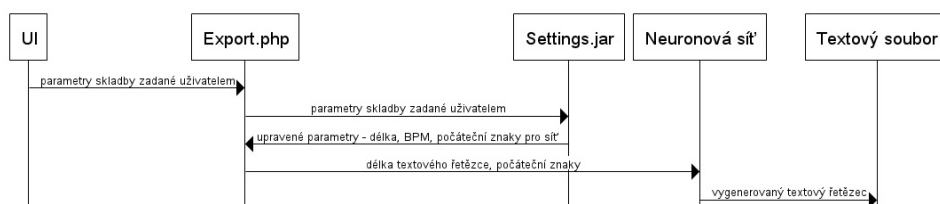
Jako množinu dat pro trénování sítě bylo použito 50 souborů s textovým řetězcem reprezentující melodie s harmoniemi, které byly přepokopírovány do jednoho textového souboru s názvem ‚input.txt‘.

Trénování neuronové sítě bylo spuštěno příkazem ‚th train.lua‘.

Generování dat. Generování dat vychází ze skriptu ‚export.php‘. Uživatel na vstupu zadá mimo jiné délku skladby a tempo, respektive jestli má píseň být smutná nebo veselá, tedy pomalá nebo rychlá. Tato data se poté odešlou do třídy ‚Settings.java‘ v balíčku ‚settings‘, kde se nastaví počet not v písni a tempo v BPM (beats per minute, počet čtvrtových not zahranych za jednu minutu). Přičemž pro volbu „sad“ se zvolí hodnota BPM v rozmezí od 60 do

70 a pro volbu „happy“ od 110 do 170. Jelikož v je v této práci za nejkratší notu považována nota šestnáctinová, počet not a hodnota BPM je zvolena tak, aby bylo toto kritérium splněno, tedy aby násobek délky písně v sekundách, hodnoty BPM a čísla 4 byl dělitelný číslem 60 (počet sekund za jednu minutu) beze zbytku. Po nalezení všech hodnot BPM odpovídajících těmto kritériím je jedno z nich náhodně vybráno a použito. Podle této hodnoty BPM se poté dopočítá počet taktů v písni a počet šestnáctinových not, které budou v posledním taktu. Tento takt není zahrnut v předchozím čísle počtu taktů, noty, které jsou v tomto taktu tedy netvoří celý takt, ale jsou důležité k doplnění skladby, aby měla požadovanou délku. Poté se pro každou část skladby náhodně vygenerují dva znaky, které budou sloužit jako počáteční znaky pro textový řetězec generovaný sítí. První znak bude velké písmeno, označující výšku noty v rozmezí A - Z a druhý znak bude malé písmeno, upřesňující pozici noty v taktu v rozmezí a - p. Žádné dvě části nemohou mít první dva znaky stejné. Pro totožné počáteční znaky totiž neuronová síť generuje stejné řetězce. Mohlo by se tedy stát, že například sloka bude znít úplně stejně jako refrén.

Generování dat neuronovou sítí bylo spuštěno příkazem „th sample.lua“. Byl zvolen soubor, který síť využije k tvorbě nového textu, v tomto případě byl zadán příkaz „cv/lm_lstm_epoch50.00_0.2998.t7“. Pro určení počátečních znaků se použije přepínač „-primetext“, pro nastavení délky textového řetězce příkaz „-length“.



Obrázek 5.4: Diagram vygenerování jedné části skladby

5.3 Převod dat do MIDI souborů

V této podkapitole je popsána úprava textových řetězců vygenerovaných sítí a samotný převod do MIDI souborů. Na konci se poté nachází diagramy shrnující postupy a použité třídy při převodu řetězců do MIDI souborů (Obrázek ??) a spojení všech částí skladby do finální písně, kterou si uživatel bude moci stáhnout (Obrázek 5.6).

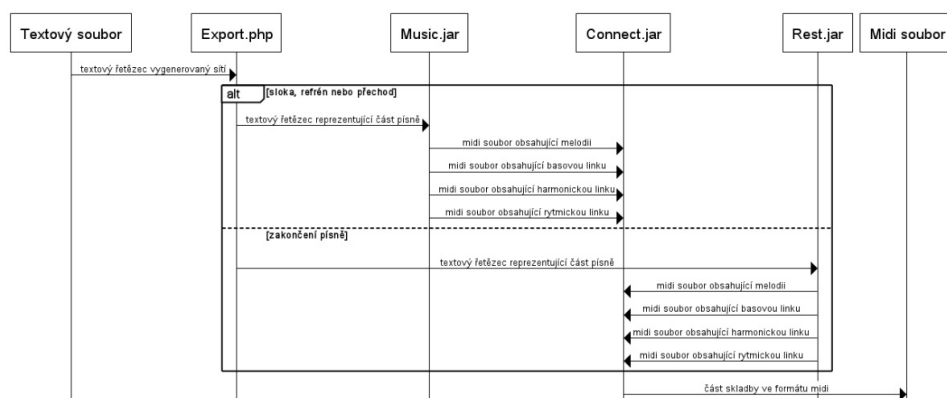
Úprava textových řetězců. Data, respektive textový řetězec vygenerovaný sítí, jsou zkontrolovány a upraveny tak, aby v nich nebyly odchylky (například

pokud neobsahují po sobě jdoucí výše popsané čtyři znaky). Pak také bude délka řetězce upravena tak, aby po spojení všech částí písně a převodu do MIDI souboru byla skladba tak dlouhá, jak si uživatel zvolil. Kód upravující text se nachází ve třídě `EditTextString.java`. Tato třída upravuje textové řetězce tak, že prochází celý vygenerovaný řetězec a hledá vždy jeden souvislý takt, kde všechny znaky a jejich pořadí odpovídá předem určeným konvencím. Pokud nějaký takt předpisům neodpovídá, tuto část přeskočí a hledá dál, dokud řetězec představující správně vygenerované takty není dostatečně dlouhý.

Převod do MIDI souborů. Pro převod textových řetězců byly použity třídy v balíčcích `,music'` a `,rest'`. V balíčku `,music'` se převádí většina skladby, konkrétně části sloka, refrén a přechod. V balíčku `,rest'` se převede zbytek not doplňující píseň, aby byla přesně tak dlouhá, jak uživatel nastavil.

Balíčky obsahují třídy `,Melody.java'`, `,Playback.java'`, `,Bass.java'` a `,Drums.java'`. Přičemž v třídě `,Melody.java'` se podle velkých písmen v řetězci vytvoří stopa melodie, třídy `,Bass.java'` a `,Playback.java'` využívají z textového řetězce čísla reprezentující akordy. Tyto dvě třídy vytvoří stopu basové linky a harmonie. Nakonec se vytvoří stopa obsahující rytmickou linku. Ta se vygeneruje z databáze, která obsahuje čtyři MIDI soubory dlouhé jeden takt, přičemž soubor `,part1.mid'` obsahuje stopu pro první takt písně, části `,part2.mid'` a `,part3.mid'` se postupně střídají až do předposledního taktu, na poslední takt potom bude použit soubor `,part4.mid'`.

Pro každou část skladby se vytvoří čtyři stopy. Tyto stopy budou spojeny pomocí třídy `,Connect.java'` v balíčku `,connect'`, vzniknou tedy soubory ve formátu MIDI představující části písně. Nakonec tyto části spojíme dohromady pomocí třídy `,Finish.java'` v balíčku `,finish'`. Vzniklý výsledný soubor pojmenujeme podle jména zadaného uživatelem a jeho odkaz přiřadíme do skriptu `,song.php'`, k tlačítku, které odkazuje na stáhnutí písně ve formátu MIDI. Dále pak příkazem „`fluidsynth -F jmenoPisne.wav /usr/share/sounds/sf2/FluidR3_GM.sf2 jmenoPisne.mid`“ převedeme píseň do formátu wav, a odkaz přiřadíme k také do skriptu `,song.php'`, tentokrát k tlačítku odkazující na stáhnutí písně ve formátu wav a nakonec použijeme příkaz „`lame -preset standard jmenoPisne.wav jmenoPisne.mp3`“ k převedení písně do formátu mp3. Odkaz přiřadíme do skriptu `,song.php'` k tlačítku odkazující na stáhnutí písně ve formátu mp3.



Obrázek 5.5: Diagram převodu jednotlivých částí do MIDI souborů



Obrázek 5.6: Diagram dokončení skladby spojením jejích částí

5.4 Uživatelské rozhraní aplikace

Jedná se o webovou aplikaci, uživatelské rozhraní tedy bylo vytvořeno pomocí skriptovacího programovacího jazyka php, kaskádových stylů (CSS) a skriptovacího jazyku JavaScript.

Webová stránka má tři sekce: ‚INTRODUCTION‘, ‚SETTINGS‘ a ‚SONG‘. Tyto sekce se zobrazují v pravé části obrazovky po kliknutí na danou položku v menu, které se nachází v levé části stránky.

V záložce ‚INTRODUCTION‘ (viz obrázek 5.7) je aplikace uživateli uvedena, je popsáno co dělá a jak se správně používá. Zdrojový kód této úvodní stránky se nachází ve skriptu ‚introduction.php‘.

V záložce ‚SETTINGS‘ (viz obrázek 5.8) lze nastavit parametry skladby. V horní části je posuvník, jehož posouváním lze nastavit délku písně v rozmezí mezi 1 až 5 minutami. Zobrazení zvolené délky v závislosti na posouvání posuvníkem, je naprogramováno ve skriptu ‚length_of_song.js‘. Dále lze pomocí přepínačů nastavit jestli má být skladba smutná nebo veselá. Pomocí výběrového pole lze zvolit strukturu písně. Volit lze mezi pěti možnostmi:

- sloka, zakončení

- sloka, refrén, sloka, zakončení

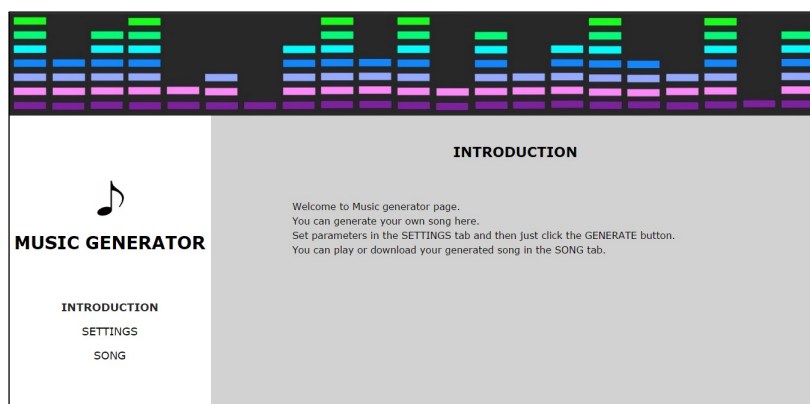
- sloka, refrén, přechod, sloka, zakončení

- sloka, refrén, sloka, refrén, přechod, sloka, zakončení

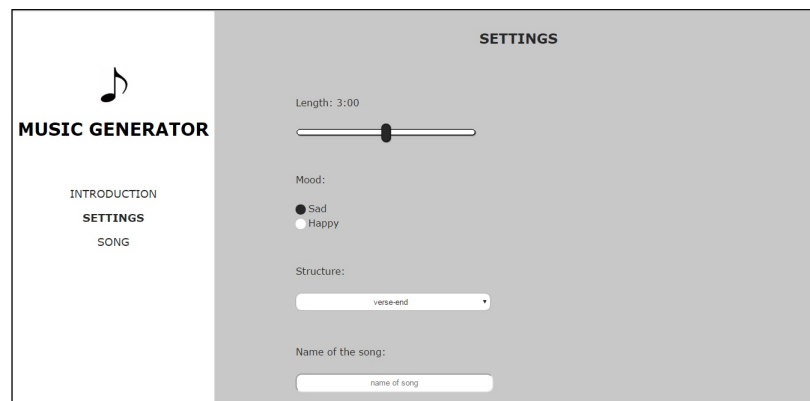
- sloka, refrén, sloka, přechod, sloka, refrén, sloka, zakončení

Nakonec si lze zvolit název písně, který musí obsahovat nejméně 1 a nejvíce 40 znaků (čísla a písmena bez diakritiky) bez mezer. Po kliknutí na tlačítko ‚EXPORT‘ v dolní části stránky se provede přesměrování do skriptu ‚export.php‘ a začne se generovat skladba se zvolenými parametry. Zdrojový kód této stránky s nastavením písně se nachází ve skriptu ‚settings.php‘.

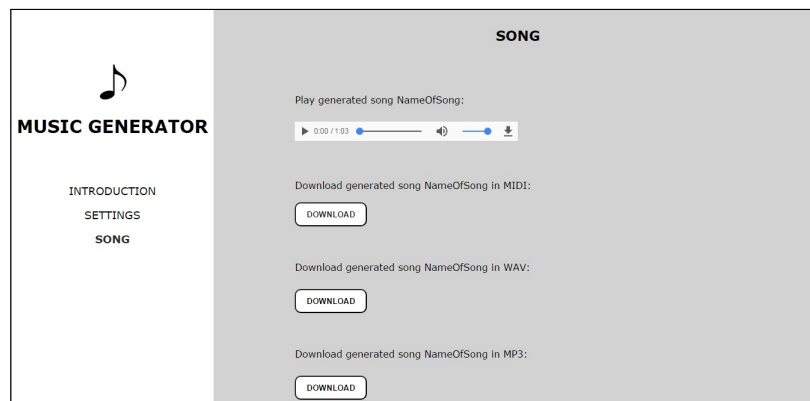
V záložce ‚SONG‘ (viz obrázek 5.9) si lze vygenerovanou píseň přehrát a stáhnout ve formátu MIDI, wav nebo mp3. Po uplynutí jedné hodiny se píseň smaže z databáze a v záložce ‚SONG‘ se objeví nápis informující o tom, že píseň není dostupná a s návodem na vygenerování nové skladby. Mazání písně z databáze je naprogramováno ve skriptu ‚delete_old.php‘. Zdrojový kód této stránky s nastavením písně se nachází ve skriptu ‚song.php‘.



Obrázek 5.7: Webová aplikace, záložka INTRODUCTION



Obrázek 5.8: Webová aplikace, záložka SETTINGS



Obrázek 5.9: Webová aplikace, záložka SONG

Kapitola 6

Testování

6.1 Úvod

Z důvodu odhalení případných nedostatků, je webová aplikace podrobena uživatelskému testování.

Cíle testu. Testování aplikace umožňuje sledovat, jak se uživatelé při jejím používání chovají a jak postupují při plnění jednotlivých, předem zadaných, úkolů. Cílem testování je odhalit případné nedostatky aplikace před jejím finálním dokončením.

Cílová skupina. Aplikace je určena pro uživatele, kteří umějí číst, rozumějí textu psaném v anglickém jazyce a dokáží ovládat počítač na základní úrovni. Pro testování bylo vybráno pět osob splňujících tyto podmínky ve věku od 17 do 47 let.

Popis aplikace. Aplikace je určena ke generování hudby. Předpokládá se, že ji uživatelé budou využívat například pro vygenerování hudebního podkladu ke svému videu. Lze si nastavit různé parametry (délka skladby, tempo, struktura a jméno) a poté hudbu stáhnout ve formátu midi, wav nebo mp3.

6.2 Podmínky pro testování

V této sekci jsou popsány výchozí podmínky, které budou mít všichni účastníci při testování aplikace.

Použitý software a hardware. Testování bude probíhat na vlastních strojích uživatelů se softwarem, na který jsou zvyklí. Důvodem je otestování aplikace na různých strojích s různými softwary a zároveň předejití nepřesností v případě, že by uživatelé dělali problém pracovat na jiném počítači s jiným softwarem nebo v jiném prohlížeči než v tom, na který je zvyklý.

Výchozí stav. Účastníkům testování bude zaslán nebo ukázán odkaz na úvodní stránku aplikace. Předpokládá se, že uživatelé vybraní k testování nebudou mít se zapnutím aplikace problém, proto tento úkon nebude uveden v seznamu úkolů.

Při reálném použití aplikace bude mít nejpravděpodobněji uživatel video o určité délce a bude si chtít vygenerovat stejně dlouhou skladbu, která mu poslouží jako podklad pod své video. Proto každý z účastníků testování dostane zadanou délku skladby, kterou bude mít za úkol vygenerovat.

6.3 Use cases

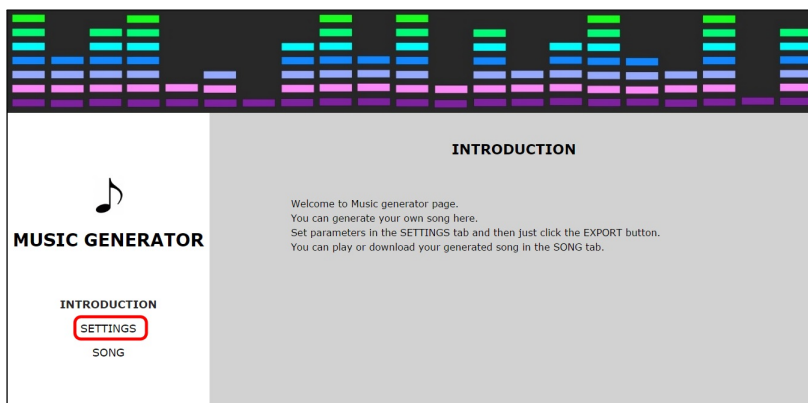
V této části jsou popsány úkoly, které budou mít účastníci testování splnit a zároveň popsáno, jak by měli optimálně postupovat.

Seznam úkolů.

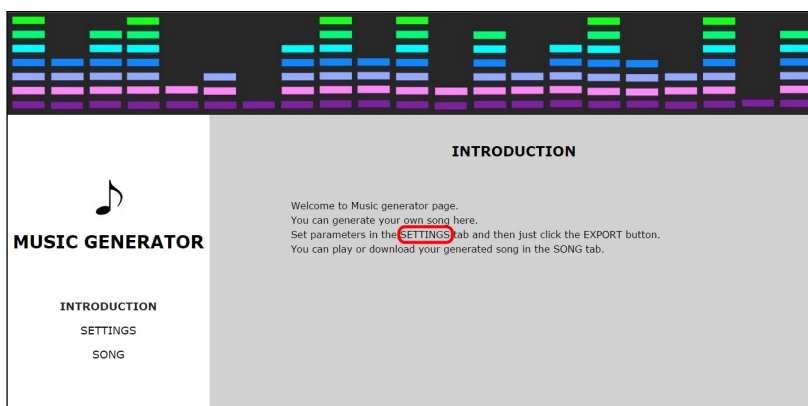
- use case 0: spustit aplikaci
- use case 1: vygenerovat skladbu, která bude splňovat zadané parametry
- use case 2: stáhnou skladbu v jednom z nabízených formátů

Optimální průchod. V této sekci je popsáno, jak by účastníci měli postupovat při plnění jednotlivých úkolů.

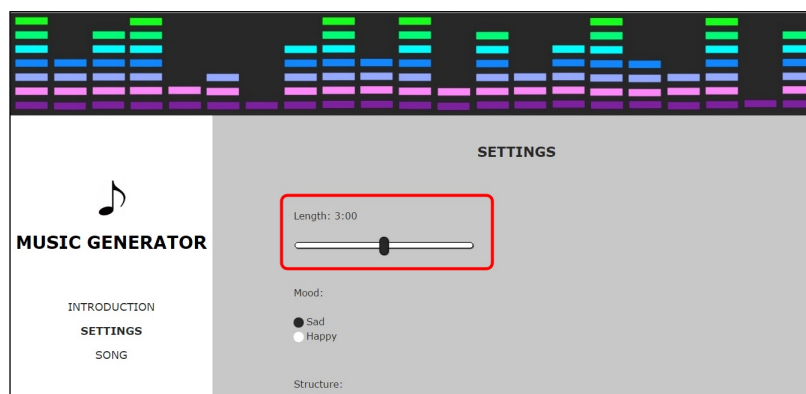
1. **use case 1** Uživatel se přesune do záložky nastavení kliknutím na položku **SETTINGS** v menu (Obrázek 6.1) nebo na odkaz v textu (Obrázek 6.2). Poté zvolí délku skladby pomocí posuvníku v horní části stránky (Obrázek 6.3), popřípadě další parametry skladby jako je tempo a struktura. Dále musí vyplnit název skladby do pole, které je umístěno pod textem „Name of the song“ (Obrázek 6.4). Toto pole smí obsahovat pouze čísla a písmena bez diakritiky, a musí být dlouhé nejméně jeden a nejvíce čtyřicet znaků. Nakonec stiskne tlačítko **EXPORT** (Obrázek 6.5) a tím se skladba vygeneruje.



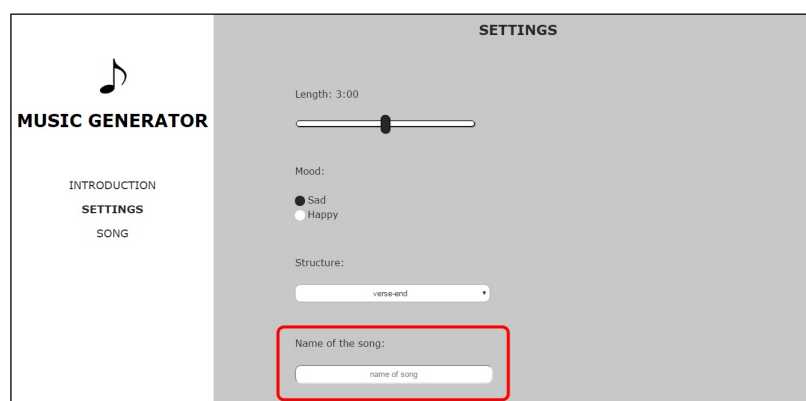
Obrázek 6.1: Úvodní strana aplikace s označenou položkou **SETTINGS** v menu



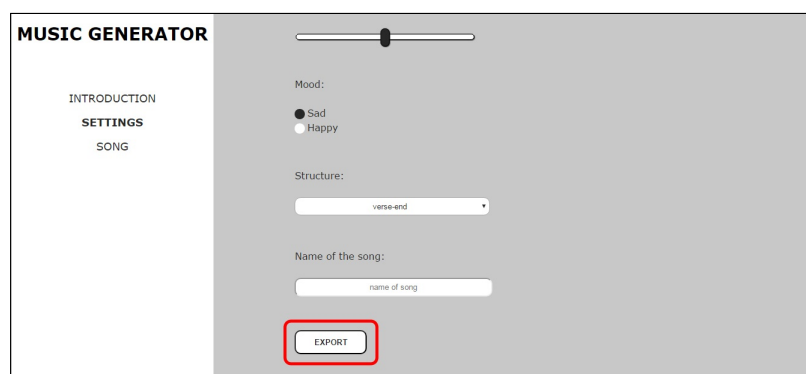
Obrázek 6.2: Úvodní strana aplikace s označenou položkou **SETTINGS** v textu



Obrázek 6.3: Sekce SETTINGS s označeným posuvníkem pro nastavení délky skladby

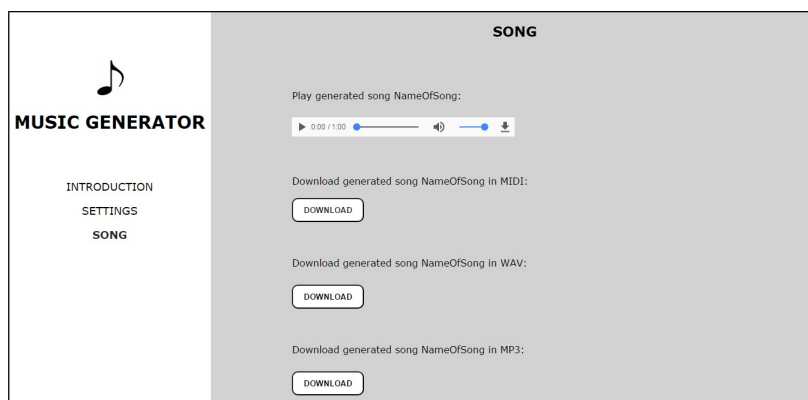


Obrázek 6.4: Sekce SETTINGS s označeným polem pro název skladby



Obrázek 6.5: Sekce SETTINGS s označeným tlačítkem pro vygenerování skladby

2. **use case 2** Uživatel je po vygenerování skladby automaticky přeměrován do záložky SONG (Obrázek 6.6). Zde si skladbu přehraje a poté jí pomocí tlačítka DOWNLOAD stáhne do svého počítače ve formátu midi, wav nebo mp3.



Obrázek 6.6: Strana pro stáhnutí vygenerované skladby

6.4 Průběh testování

Zde jsou uvedeny informace o účastnících testování a zároveň popsáno jak postupovali při jednotlivých úkolech.

1. účastník.

- věk: 22 let
- software: Windows 10
- prohlížeč: Edge
- zadaná délka skladby: 3:00

1. **use case 1** Účastník našel odkaz na záložku **SETTINGS** v textu na úvodní stránce, každopádně si ale myslí, že to byla spíše náhoda a odkaz není moc dobře viditelný. Přesunul se tedy na stranu s nastavením. Bez problémů nastavil požadovanou délku skladby, zvolil tempo a strukturu a skladbu pojmenoval. Tlačítko **EXPORT** sloužící k vygenerování skladby mu přijde zavádějící, zvolil by tedy raději jiné pojmenování. Při kliknutí na tlačítko se záložka s vygenerovanou skladbou načítá až několik sekund, ale na první pohled není vidět žádný progres, uživatel si tedy může myslet, že udělal chybu.
2. **use case 2** Stáhnutí skladby účastníkovi nedělalo problém. Skladbu stáhl ve formátu mp3.

2. účastník.

- věk: 47 let
- software: Windows 7
- prohlížeč: Mozilla Firefox
- zadaná délka skladby: 2:45

1. **use case 1** Účastník odkaz na záložku SETTINGS v textu na úvodní stránce nenašel, na stránku s nastavením se tedy dostal přes položku v menu. Bez problémů nastavil požadovanou délku skladby a zvolil tempo a název. U výběru struktury chvíli váhal nad významem slova „bridge“, nakonec ale zvolil jednu z možností. Při kliknutí na tlačítko vygenerování skladby má stejný problém jako předchozí účastník. Neví, jestli neudělal chybu, protože stránka s odkazy na stáhnutí vygenerované skladby se dlouho načítá.
2. **use case 2** Druhý úkol účastníkovi nedělal větší problém. Skladbu stáhl ve formátu mp3.

3. účastník.

- věk: 44 let
- software: Windows 7
- prohlížeč: Chrome
- zadaná délka skladby: 3:20

1. **use case 1** Účastník odkaz na záložku SETTINGS v textu na úvodní stránce přehlédl, na stránku s nastavením se tedy dostal přes položku v menu. Bez problémů nastavil požadovanou délku skladby, zvolil tempo a strukturu. Při vyplňování názvu skladby chvíli váhá. Při kliknutí na tlačítko vygenerování skladby má stejný problém jako předchozí účastníci, stránka s vygenerovanou skladbou se dlouho načítá a není si jistý, jestli je vše v pořádku a jak dlouho má ještě čekat.
2. **use case 2** Stáhnutí skladby účastníkovi nedělalo problém, stáhl ji ve formátu mp3.

4. účastník.

- věk: 47 let
- software: Windows 7
- prohlížeč: Chrome
- zadaná délka skladby: 4:10

1. **use case 1** Účastník neviděl odkaz na záložku SETTINGS v textu na úvodní stránce, na stránku s nastavením se tedy dostal přes položku v menu. Nastavil požadovanou délku skladby, zvolil tempo, strukturu a název. Při kliknutí na tlačítko vygenerování skladby má stejný problém jako předchozí účastníci, stránka s vygenerovanou se dlouho načítá a není si jistý, jestli je vše v pořádku.
2. **use case 2** Účastník si chce skladbu přehrát, což se mu daří. Poté ale klikne na tlačítko stáhnutí v přehrávači. Skladba se stáhne ve formátu mp3, ale účastník to nemůže vědět, protože informace o formátu souboru, který lze stáhnout pomocí přehrávače není nikde napsaná. Přehrávač byl totiž původně určen jen k přehrávání.

5. účastník.

- věk: 17 let
- software: Windows 10
- prohlížeč: Mozilla Firefox
- zadaná délka skladby: 1:05

1. **use case 1** Účastník nenašel odkaz na záložku SETTINGS v textu na úvodní stránce, na stránku s nastavením se tedy dostal přes položku v menu. Nastavil požadovanou délku skladby, zvolil tempo, strukturu a název. Při kliknutí na tlačítko vygenerování skladby si stejně jako předchozí účastníci není jistý, jestli je vše v pořádku, protože se stránka načítá dlouhou dobu.
2. **use case 2** S druhým úkolem účastník neměl problém. Skladbu stáhl ve formátu mp3.

6.5 Závěr

Všichni účastníci testování nakonec oba úkoly zvládly bez větších potíží. Nejzávažnějším odhaleným problémem bylo neinformování uživatelů o dlouhotrvajícím generování skladby. Další problém byl například matoucí název tlačítka pro generování skladby nebo tlačítka pro stáhnutí skladby zobrazující se v prohlížeči Chrome v liště přehrávače.

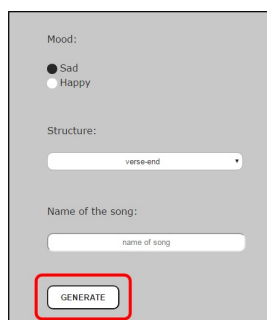
Za zmínku také stojí, že všichni účastníci testování si skladbu nakonec stáhli ve formátu mp3. Jelikož je ale tlačítka sloužící ke stáhnutí skladby v tomto formátu až v dolní části stránky, nelze předpokládat, že je to způsobeno chybou uživatelského rozhraní. Formát mp3 je nejspíše ze všech třech nabízených formátů nejrozšířenější a každý z účastníků testování ho znal. Problémy odhalené testováním aplikace budou opraveny a řešení popsáno v kapitole „Úprava aplikace po testování“.

Kapitola 7

Úprava aplikace po testování

V předchozí kapitole (Kapitola 6) bylo zjištěno, že se v aplikaci vyskytují drobné chyby. Vesměs ale byli uživatelé s aplikací spokojeni. V následujících podkapitolách jsou rozebrány nedostatky, které testování uživatelského rozhraní odhalilo a zároveň je popsáno, jak byly tyto chyby odstraněny.

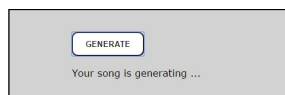
Tlačítko EXPORT. Jedním z nedostatků bylo neúplné pochopení významu tlačítka EXPORT, které sloužilo ke generování muziky v sekci SETTINGS. Název tlačítka byl tedy změněn na GENERATE (viz obrázek 7.1).



Obrázek 7.1: Změna názvu tlačítka EXPORT v sekci SETTINGS na ‚GENERATE‘

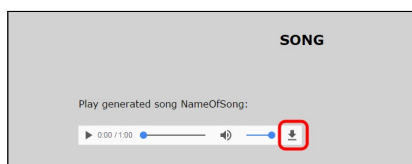
Neinformovanost při dlouhotrvajícím generování skladby. Pravděpodobně největším nedostatkem bylo podle všeho neinformování uživatele při dlouhém

čekání na vygenerování skladby po stisknutí tlačítka GENERATE v sekci SETTINGS. Tento nedostatek byl opraven přidáním nápisu „generating...“ pod tlačítko GENERATE (viz obrázek 7.2). Po této úpravě by tedy mělo být uživatelům zřejmé, že se nikde nestala chyba, jen generování skladby trvá delší dobu.

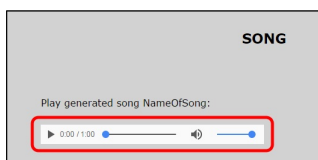


Obrázek 7.2: Přidání nápisu „generating...“ pod tlačítko GENERATE v sekci SETTINGS

Možnost stáhnutí písně v přehrávači. Další výtkou byla možnost stáhnout píseň rovnou v přehrávači pod nápisem „Play generated song“ (viz obrázek 7.3). Uživatel totiž neví, v jakém formátu se píseň stáhne. Tento nedostatek byl vyřešen pomocí kaskádových stylů tak, že tlačítko pro stáhnutí písně bylo odstraněno. Upravený přehrávač je k vidění na obrázku 7.4.

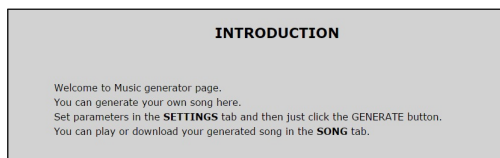


Obrázek 7.3: Tlačítko pro stáhnutí písně v sekci SONG

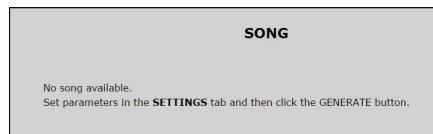


Obrázek 7.4: Přehrávač bez možnosti stáhnutí písně v sekci SONG

Viditelný odkaz textu. Jen jeden účastník z pěti si všiml odkazu na záložku SETTINGS v textu na úvodní straně. Je tedy potřeba udělat odkaz jasně viditelný. Všechna slova v textu, která slouží jako odkaz, tedy byla napsána tučným fontem (viz obrázky 7.5 a 7.6).



Obrázek 7.5: Odkazy v textu v části SETTINGS tučným fontem



Obrázek 7.6: Odkaz v textu v části SONG tučným fontem



Kapitola 8

Závěr

Pro svou práci jsem si vybrala velmi současné téma, tedy problematiku generování hudby počítačem. Je to směr, o který se zajímá jak mnoho amatérských nadšenců, tak i komerční sféra. Výsledky jejich práce jsou pak dobře uplatnitelné v praxi, což bylo jedno z kritérií, proč jsem si zvolila právě toto téma. Úvodem jsem popsala současné techniky využívané ke generování hudby pomocí umělé inteligence. Podrobněji jsem pak popsala neuronové sítě, jejich strukturu, chování, schopnost učení se. Pro tvorbu aplikace, která generuje melodii, jsem si vybrala vícevrstvou rekurentní neuronovou síť. Nejdříve jsem si vytvořila množinu dat, která představovala již existující písně ve formátu midi. Poté jsem písně upravila a z každé použila jen její melodii. Tu jsem převedla do textových řetězců, které jsem následně předložila neuronové síti jako množinu dat, podle kterých se bude učit. Dále jsem vytvořila program, ve kterém se textové řetězce převádějí zpátky do midi souborů, a taky program pro skládání jednotlivých částí písně dohromady. Nakonec jsem vytvořila uživatelské rozhraní, ve kterém si uživatel zvolí délku skladby, tempo, strukturu a jméno a aplikace mu vygeneruje skladbu splňující jeho požadavky.

Případná vylepšení aplikace. V tuto chvíli je vytvořena fungující aplikace pro generování hudby. Lze ji ale do budoucna ještě vylepšit. Například bych mohla rozšířit množinu dat pro trénování sítě, přidat možnost nastavení více parametrů skladby uživatelem, nebo rozdělit množinu dat k trénování sítě podle žánrů, a následně pak generovat skladby různých žánrů.

Příloha A

Uživatelská příručka

URL webové aplikace: <http://147.32.81.78/web/>

Úvod. Na úvodní straně se zobrazí úvod popisující aplikaci a zároveň obecný návod pro použití aplikace. V textu lze kliknout na slova ‚SETTINGS‘ a ‚SONG‘, které jsou odkazy, které uživatele přesměrují na stejnojmennou záložku.

Nastavení. V sekci ‚SETTINGS‘ lze nastavit parametry skladby. V horní části je posuvník, kterým lze nastavit délku písně (od 1:00 do 5:00), pod ním je přepínač kterým lze nastavit, jestli bude píseň smutná nebo veselá, dále je na stránce výběrové pole pro výběr struktury a nakonec textové pole pro název skladby. Název písně musí být nejméně jeden a nejvíce čtyřicet znaků dlouhý a zároveň obsahovat jen čísla a písmena bez diakritiky. Kliknutím na tlačítko ‚GENERATE‘ se začne generovat skladba s požadovanými vlastnostmi.

Stáhnutí. Pokud byla v sekci ‚SETTINGS‘ vygenerována píseň, v sekci ‚SONG‘ si ji lze přehrát kliknutím na přehrávač pod nápisem „Play generated song“ nebo stáhnout kliknutím na jedno z tlačítek ‚DOWNLOAD‘ ve formátu midi, wav nebo mp3.



Příloha B

Obsah CD

- Adresář **Data** obsahuje textový soubor s daty, která byla použita k trénování sítě
- Adresář **Java** obsahuje zdrojové kódy programů vytvořených v jazyce Java
- Adresář **Text** obsahuje zdrojové tex soubory k závěrečné zprávě
- Adresář **Vygenerovane__skladby** obsahuje skladby vygenerované aplikací
- Adresář **Web** obsahuje zdrojové kódy v php, JavaScript a CSS.
- Soubor **cizlokat_zaverecna_zprava_BP.pdf** je elektronická vaerze dokumentace k bakalářské práci
- Soubor **url_aplikace.txt** obsahuje URL webové aplikace

Příloha C

Literatura

- [aDS] Christos Stergiou a Dimitrios Siganos, *Neural networks - introduction to neural networks*, https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html, [Online].
- [Bla] Milan Blaha, *Adaptační dynamika neuronu*, <http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--adaptacni-dynamika-neuronu--uceni-s-ucitelem>, [Online].
- [Bri16] Frank Brinkkemper, *Analyzing six deep learning tools for music generation*, <http://www.asimovinstitute.org/analyzing-deep-learning-tools-music>, Říjen 2016, [Online].
- [Cra09] Jamie Craane, *Beginning harmonic analysis*, <http://jcraane.blogspot.cz/2009/06/melody-composition-using-genetic.html>, Červen 2009, [Online].
- [Hla] Václav Hlaváč, *Umělé neuronové sítě z pohledu rozpoznávání*, people.ciirc.cvut.cz/~hlavac/TeachPresCz/31Rozp/72UmeleNN.ppt, [Online].
- [ka99] kolektiv autorů, *Všeobecná encyklopedie v osmi svazcích*, 1999.
- [Kar15] Andrej Karpathy, *The unreasonable effectiveness of recurrent neural networks*, <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>, Květen 2015, [Online].
- [Lep08] Oldřich Lepil, *Fyzika pro gymnázia, mechanické kmitání a vlnění*, 2008.

- [maRN96] Jiří Šíma a Roman Neruda, *Teoretické otázky neuronových sítí*, 1996.
- [Mat] MathWorks, *Genetic algorithm*, <https://www.mathworks.com/discovery/genetic-algorithm.html>, [Online].
- [Mat13] Jiří Materna, *Deep learning: budoucnost strojového učení?*, <https://vyhledavani.sblog.cz/2013/01/09/deep-learning-budoucnost-strojoveho-uceni>, Leden 2013, [Online].
- [Nav16] Mirko Navara, *Markovovy řetězce*, cmp.felk.cvut.cz/~navara/psi/Markov_print.pdf, Prosinec 2016, [Online].
- [Ola15] Christopher Olah, *Understanding lstm networks*, <http://colah.github.io/posts/2015-08-Understanding-LSTMs>, Srpen 2015, [Online].
- [Pan06] Jimmy Pang, *Neural networks - network architecture*, <http://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/feedforward.html>, Prosinec 2006, [Online].
- [Sap03] Craig Stuart Sapp, *Digital music programming ii: Markov chains*, <http://peabody.sapp.org/class/dmp2/lab/markov1>, 2003, [Online].
- [SJa] Catherine Schmidt-Jones, *Beginning harmonic analysis*, <https://www.earmaster.com/music-theory-online/ch05/chapter-5-5.html>, [Online].
- [SJB] _____, *Harmony*, <https://www.earmaster.com/music-theory-online/ch02/chapter-2-5.html#m11654>, [Online].
- [SJC] _____, *Triads*, <https://www.earmaster.com/music-theory-online/ch05/chapter-5-1.html>, [Online].
- [Tea] Theano Development Team, *Convolutional neural networks (lenet)*, <http://www.deeplearning.net/tutorial/lenet.html>, [Online].
- [WR05] Inc. Wolfram Research, *Wolframtones - how it works*, <http://tones.wolfram.com/about/how-it-works>, 2005, [Online].