



POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

Praca dyplomowa inżynierska

Tworzenie muzyki za pomocą sztucznych sieci neuronowych

autor: Grzegorz Kazana

kierujący pracą: dr inż. Grzegorz Baron

Gliwice, listopad 2019

Oświadczenie

Wyrażam zgodę / Nie wyrażam zgody* na udostępnienie mojej pracy dyplomowej / rozprawy doktorskiej*.

Gliwice, dnia 26 listopada 2019

.....
(podpis)

.....
(poświadczenie wiarygodności
podpisu przez Dziekanat)

* podkreślić właściwe

Oświadczenie promotora

Oświadczam, że praca „Tworzenie muzyki za pomocą sztucznych sieci neuronowych” spełnia wymagania formalne pracy dyplomowej inżynierskiej.

Gliwice, dnia 26 listopada 2019

.....
(podpis promotora)

Spis treści

1	Wstęp	1
1.1	Cel i zastosowania	2
1.2	Zakres pracy	2
2	Analiza tematu	3
2.1	Wprowadzenie do dziedziny	3
2.2	Założenia	4
2.3	Przegląd literatury	5
2.4	Odniesienie do istniejących prac	6
2.5	Opis narzędzi	6
3	Dobór danych	9
3.1	Porównanie formatów danych	10
3.1.1	Pliki audio	10
3.1.2	Notacja ABC	10
3.1.3	Format midi	11
3.1.4	Podsumowanie	11
3.2	Opis wybranego zbioru danych	12
4	Specyfikacja zewnętrzna	13
5	Specyfikacja wewnętrzna	15
6	Weryfikacja i walidacja	17
7	Podsumowanie i wnioski	19

8	Test	21
8.1	test	21
8.1.1	test	22
8.1.2	test	22

Rozdział 1

Wstęp

Współczesne zastosowania metod uczenia maszynowego są bardzo szerokie. Z pośród niezliczonej ilości problemów praktycznie niemożliwych do rozwiązania podejściami klasycznymi, duża część zostaje rozwiązana z wykorzystaniem szerokiej gamy algorytmów oraz modeli uczenia.

Zadania stawiane przed systemami sztucznej inteligencji można podzielić między innymi na problemy:

- klasyfikacji - polegające na przypisaniu przykładu do określonej kategorii, np. rozpoznawanie obiektów na obrazie
- grupowania - polegające na wyodrębnieniu przypadków o podobnych cechach lub współwystępujących, np. systemy sugerujące przedmioty w sklepach internetowych
- regresji - polegające na oszacowaniu wartości ciągłej dla poszczególnych przykładów, np. przewidywanie cen produktów.

Pomimo doniosłości oraz znaczenia zastosowań powyższych zagadnień, nie można w nich dostrzec przejawów kreatywności. Jednym z problemów, w którym można doszukiwać się inwencji jest tworzenie muzyki.

1.1 Cel i zastosowania

Celem pracy jest analiza i porównanie wyników różnych podejść do obróbki plików muzycznych oraz sposobów syntezy próbek z wykorzystaniem metod uczenia maszynowego.

Pomimo że praca ma charakter badawczo-eksperymentatorski, opracowane metody oraz ich wyniki mogą potencjalnie znaleźć zastosowanie w zadaniach ekstrakcji stylu, augmentacji istniejących zbiorów danych, lub w optymistycznym scenariuszu jako narzędzie wspierające proces twórczy muzyków.

1.2 Zakres pracy

Do zadań autora pracy należy

- dobór zbioru danych
- dobór, implementacja i analiza metod przetwarzania plików muzycznych
- dobór i parametryzacja architektury modelu uczenia maszynowego
- wykorzystanie przygotowanych danych w procesie uczenia i syntezy nowych próbek
- ocena uzyskanych wyników

Rozdział 2

Analiza tematu

Celem zadania doboru i analizy metod obróbki plików muzycznych jest przekształcenie danych do postaci użytecznej przez algorytmy uczenia maszynowego. Do istotnych aspektów tego kroku należy między innymi stopień kompresji danych, sposób wyrażania relacji między przykładami, możliwość bezstratnej transformacji odwrotnej.

Zadanie generowania muzyki polega na ekstrakcji pewnych cech charakterystycznych przykładowych utworów, np. stylu konkretnego artysty, i wykorzystaniu ich przy syntezie tworzonych próbek.

2.1 Wprowadzenie do dziedziny

Jedną z najbardziej rozpowszechnionych metod uczenia maszynowego jest zastosowanie sieci neuronowych. Zaproponowane w latach czterdziestych i rozwijane w drugiej połowie XX wieku, sztuczne sieci neuronowe czerpią inspirację z sposobu funkcjonowania ludzkiego mózgu zbudowanego z komórek nerwowych - neuronów. Połączenia między komórkami są modelowane poprzez wagi, reprezentujące siłę połączenia, a zjawisko aktywacji komórek w sieci poprzez operację ważonej sumy informacji z połączonych neuronów oraz ich wag. W latach siedemdziesiątych opracowano algorytm uczenia sztucznych sieci z powodzeniem wykorzystywany do dziś - propagację wsteczną. Metoda ta polega na minimalizacji błędu predykcji poprzez regulację wag w oparciu o pochodną funkcji błędu. Zastosowania sieci neuronowych

są szerokie, co potwierdza twierdzenie stanowiące o ich możliwości aproksymacji dowolnej funkcji ciągłej w zamkniętym przedziale.

Mimo tego, że ta metoda uczenia maszynowego jest znana od wielu dekad, dopiero w ostatnich latach przeżywa swoisty renesans spowodowany wzrostem dostępnej mocy obliczeniowej oraz możliwością zastosowania w procesie uczenia kart graficznych znacząco przyspieszających równoległe operacje matematyczne.

Ponieważ dane reprezentujące muzykę mają postać sekwencji rozłożonej w czasie, konieczne jest wykorzystanie sieci mających możliwość agregacji stanu tj. posiadające pamięć. Sieciami spełniającymi powyższy warunek są modele należące do grupy rekurencyjnych sieci neuronowych. Najprostszym przykładem rekurencyjnej sieci neuronowej jest sieć na której wejście przekazywany jest również stan wyjść z analizy poprzedniego elementu sekwencji. Niestety, taka architektura jest narażona na wiele problemów, takich jak trudność tworzenia powiązań pomiędzy odległymi elementami sekwencji oraz znikający gradient.

Architekturą rozwiązującą powyższe problemy jest zasugerowana przez XXXXXX architektura Long short-term memory network (LSTM). Kosztem jej większych możliwości jest zwielokrotnienie ilości parametrów, co może przekładać się na dłuższy czas uczenia.

2.2 Założenia

Na potrzeby pracy została przyjęta następująca definicja muzyki:

Muzyką nazywamy ciągi dźwięków tworzące kompozycyjną całość.

Utwory muzyczne można analizować pod wieloma względami, takimi jak:

- rytmiczność - organizacja dźwięków w czasie
- melodyczność - sposób zestawiania następujących dźwięków
- harmonicznosc - spójność i ład występujący między dźwiękami
- dynamika - zróżnicowanie siły dźwięków

W kontekście pracy skupiono się na dwóch aspektach wynikających z powyższej definicji: rytmiczności i tonalności.

2.3 Przegląd literatury

W podobnej tematyce powstało wiele prac, lecz każda wyróżnia się innym podejściem do problemu.

Elementem wspólnym dużej części prac jest wykorzystany format danych wejściowych. W większości studiowanych artykułów, twórcy decydują się na wykorzystanie danych muzycznych w formacie midi [4,5,6,7], aczkolwiek równie popularnym wyjściem jest skorzystanie z notacji ABC [1,2]. Najrzadsze jest podejście opierające się o wykorzystanie surowych plików audio, np. w formacie wav [8].

Jedną z różnic występujących między podejściami jest sposób obróbki i tworzenia wewnętrznej reprezentacji danych, gdyż praktycznie wszyscy autorzy proponują własne rozwiązanie problemu. W przypadku danych tekstowych w formacie ABC, częstym elementem jest ograniczenie lub usunięcie meta informacji zawartych w plikach [1,2] oraz pominięcie białych znaków. Konkretnie znaki zapisu są albo przedstawione jako wektory kodu 1 z N [2], lub przekształcane są kolejne wartości liczb naturalnych [1]. W pracach korzystających z plików midi, częstym podejściem do problemu reprezentacji wysokości dźwięku jest zastosowanie kodu 1 z N [4], lecz większym zróżnicowaniem cechuje się podejście do reprezentacji danych w wymiarze czasu. Duża część podejść opiera się na restrykcji możliwych wartości rytmicznych do dyskretnego zbioru [5,6], podczas gdy stosowane są również podejścia korzystające z wewnętrznej częstotliwości próbkowania plików midi [7].

Kluczowym aspektem wszystkich prac jest dobór modelu uczenia maszynowego. Najbardziej popularnym wyborem są rekurencyjne sieci neuronowe [1,2,3,7], ale opisywane są również inne podejścia, na przykład:

- modele kontekstowe, takie jak Continuous Bag-of-Words [1],
- modele generatywno adwersyjne (Generative Adversarial Networks) [1,7],
- modele konwolucyjne [4,8]

Aspektem wartym zwrócenia uwagi, są również poczynione założenia i nałożone ograniczenia. Niektóre metody przytoczone przez autorów prac wliczają:

- pominięcie elementów polifonicznych utworów [2,5],

- ograniczenie zbioru danych do utworów będących w ustalonej tonacji i/lub ustalonym metrum [2,6]
- ograniczenie na występujące wartości rytmiczne [5]

2.4 Odniesienie do istniejących prac

Niniejsza praca w kolejnych rozdziałach prezentuje analizę różnych podejść do kolejnych etapów procesu implementacji systemu, którego celem jest generowanie utworów muzycznych. W rozdziale 3. poruszono problem wyboru formatu danych, oraz przedstawiono opis zbioru na którym były przeprowadzane eksperymenty. Przedstawiono potencjalne trudności wiążące się z wyborem poszczególnych źródeł danych. W kolejnym rozdziale omówiono różne podejścia do zagadnienia reprezentacji wybranego formatu danych w postaci numerycznej. Zawiera również opis wad i zalet każdego z podejść. Treścią piątego rozdziału jest przedstawienie obranego typu modelu uczenia maszynowego oraz wybranej architektury, razem z opisem procesu uczenia. Rozdział poświęcony generacji próbek zawiera opis przyjętego podejścia oraz sposobów wymuszania procesu generacji. Dalsze rozdziały poświęcone są wnioskowi i analizie wyników.

2.5 Opis narzędzi

Podczas implementacji korzystano z narzędzi:

- Google Colab - usługa będąca środowiskiem obliczeniowym przystosowanym do uruchamiania skryptów w języku Python, z dostępem do akceleratorów obliczeniowych w postaci kart graficznych i jednostek tensorowych,
- Tensorflow - biblioteka języka Python służąca do wydajnych obliczeń, z dużym wsparciem dla tworzenia i uczenia głębokich sieci neuronowych. Podczas pracy skorzystano z wysokopoziomowego interfejsu `tf.keras`, zawierającego predefiniowane architektury i warstwy sieci,
- Jupyter notebook - środowisko uruchomieniowe języka Python, umożliwiające przejrzystą ilustrację wykonywanych operacji,

- mido - biblioteka ułatwiająca otwieranie i przetwarzania plików w formacie midi.

Rozdział 3

Dobór danych

Jednym z kluczowych aspektów każdego procesu analizy danych i uczenia maszynowego jest odpowiedni dobór danych. Decyzje podjęte na tym etapie pociągają za sobą konsekwencje w kolejnych etapach procesu. W zależności od postaci danych wejściowych, określany jest konieczny nakład pracy w procesie ich obróbki. Ponadto, odpowiedni dobór danych ma znaczący wpływ na otrzymywane wyniki.

W przypadku danych muzycznych należy zwrócić między innymi na cechy takie jak:

- dostępność danych treningowych,
- sposób reprezentacji upływu czasu,
- sposób wyrażenia wysokości dźwięku,
- jasne przedstawienie zależności między dźwiękami,
- stopień kompresji,
- złożoność samego formatu oraz możliwość istnienia niepoprawnych reprezentacji.

3.1 Porównanie formatów danych

3.1.1 Pliki audio

Najmniej restrykcyjnymi formatami plików audio są formaty będące zapisem amplitudy fali akustycznej. Przykładami plików tego rodzaju są pliki WAV oraz AIFF. Kluczowym parametrem plików audio jest częstotliwość próbkowania. Od niej zależy zakres możliwych do wyrażenia częstotliwości. Typową wartością dla plików WAV jest 44100Hz, co według twierdzenia Nyquista-Shannona przekłada się na zakres częstotliwości ograniczony wartością 22050Hz, będącą bliską granicą słyszalności ucha ludzkiego.

Ponieważ wysokości dźwięków wyrażone są jedynie poprzez sekwencje zmian amplitudy sygnału, ich ekstrakcja wymagałaby wykorzystania transformaty Fouriera.

Największą zaletą i wadą tego formatu jednocześnie jest jego swoboda. Postać fali akustycznej pozwala na przedstawienie każdego dźwięku będącego w paśmie przenoszenia, za równo złożonych wieloinstrumentowych kompozycji muzycznych, prostych melodii, jak i dźwięków nie podlegających pod definicję muzyki. Prowadzi to do niskiego stopnia kompresji informacji, wymuszającego złożone metody obróbki i skomplikowane modele uczenia.

3.1.2 Notacja ABC

Notacja ABC jest jedną z cyfrowych postaci klasycznego zapisu nutowego. Dane w plikach ABC przedstawione są za pomocą znaków ASCII.

Ponieważ jest to odpowiednik zapisu nutowego, oznacza to że za równo wysokości dźwięków, jak i wartości rytmiczne należą do dyskretnego zbioru. Fakt ten nakłada spore ograniczenie na treść plików, lecz w kontekście muzyki jest to ograniczenia bardzo pomocne.

Poza informacją o samym dźwięku i jego wartości rytmicznej, format zawiera również metadane utworu takie jak autor, sygnatura i tempo.

Dużą zaletą tego formatu jest jego wysoki stopień kompresji, za pomocą relatywnie krótkich ciągów znaków jesteśmy w stanie opisać znaczną część utworu muzycznego. Głównym mankamentem notacji ABC w kontekście niniejszej pracy

jest fakt, że nie wszystkie ciągi znaków są poprawnymi zapisami w notacji ABC. Oznacza to, że pojedyncze błędy w generowanych sekwencjach mogą powodować niepoprawność całego utworu. W przypadku niemożliwości wyuczenia modelu tworzenia całkowicie poprawnych w ciągów wymagałoby walidacji i obsługi poprawiania błędów w zapisie.

3.1.3 Format midi

Pliki w formacie midi są zapisem wiadomości przesyłanych protokołem komunikacyjnym o ten samej nazwie. Każdy plik midi składa się z kanałów oraz ścieżek. Poszczególne kanały reprezentują poszczególne instrumenty biorące udział w nagraniu. Na poszczególnych ścieżkach znajdują się wiadomości reprezentujące zdarzenia w utworze. Do najczęstszych z nich należą: `note_on`, `note_off`, `set_tempo`.

W przypadku wiadomości związanych z rozpoczęciem lub zakończeniem dźwięku dołączony jest również numer dźwięku z zakresu 0-127. Wartości tego parametru odpowiadają kolejnym dźwiękom skali dwunastotonowej poczynając od C-1 i kończąc na G9. Dostępne dźwięki są nadzbiorem dźwięków dostępnych na klawiaturze klasycznego fortepianu.

Dodatkowym parametrem każdej wiadomości jest wartość `time`, będąca ilością ticków które upłynęły od poprzedniej wiadomości. Oznacza to, że długość trwania dźwięku można wyznaczyć poprzez zsumowanie parametrów `time` wiadomości występujących pomiędzy odpowiadającymi sobie zdarzeniami `note_on` i `note_off`.

Możliwa jest konwersja wartości za równo na czas w milisekundach jak i wartości rytmiczne. Stałe potrzebne do tych przekształceń są najczęściej zawarte w meta wiadomościach znajdujących się na początku utworu.

Zaletą formatu midi jest dyskretna reprezentacja wysokości dźwięku oraz pozwalająca na elastyczną interpretację postać upływu czasu.

3.1.4 Podsumowanie

Wszystkie omówione formaty danych mają wady i zalety. Cechą opisującą wszystkie z nich, jest bogata dostępność danych, przez co wybór dokonano na podstawie pozostałych cech. W dalszej części pracy opisywane podejścia będą dotyczyć jedynie plików w formacie midi, na którego wybór zdecydowano się drogą eliminacji. Pliki

audio odrzucono przez wymaganą złożoność wstępnej obróbki, a notację ABC z powodu obaw związanych z trudnościami zagwarantowania syntaktycznej poprawności generowanych ciągów znaków.

3.2 Opis wybranego zbioru danych

Rozdział 4

Specyfikacja zewnętrzna

- wymagania sprzętowe i programowe
- sposób instalacji
- sposób aktywacji
- kategorie użytkowników
- sposób obsługi
- administracja systemem
- kwestie bezpieczeństwa
- przykład działania
- scenariusze korzystania z systemu

Rozdział 5

Specyfikacja wewnętrzna

- przedstawienie idei
- architektura systemu
- opis struktur danych
- komponenty, moduły, biblioteki
- przegląd ważniejszych klas
- przegląd ważniejszych algorytmów (własnych)
- szczegóły implementacji ważniejszych fragmentów
- diagramy uml
- zastosowane wzorce projektowe

Rozdział 6

Weryfikacja i walidacja

- sposób testowania w ramach pracy
- organizacja eksperymentów
- przypadki testowe
- zakres testowania (pełny / niepełny)
- wykryte i usunięte błędy
- opcjonalne wyniki badań eksperymentalnych

Rozdział 7

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych dalszych prac (rozbudowa funkcjonalna)
- problemy napotkane w trakcie pracy

Rozdział 8

Test

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed nunc odio, vulputate at sollicitudin sit amet, facilisis nec mi. Nam tincidunt, urna non facilisis imperdiet, tortor felis vehicula arcu, mattis fringilla elit quam ac enim. Aenean eleifend elit a consequat aliquet. Ut mattis eros eu euismod feugiat. Pellentesque bibendum porta nulla non hendrerit. Mauris tincidunt lectus eget dui placerat semper. Vestibulum ac semper turpis. Aenean vulputate egestas nunc sit amet consequat.

1. Nam egestas, libero eget sodales sodales, enim massa fringilla nulla, vel pharetra leo dui in tellus. Vestibulum sed justo eu quam porta tincidunt.
2. Nam egestas, libero eget sodales sodales, enim massa fringilla nulla, vel pharetra leo dui in tellus. Vestibulum sed justo eu quam porta tincidunt.
3. Nam egestas, libero eget sodales sodales, enim massa fringilla nulla, vel pharetra leo dui in tellus. Vestibulum sed justo eu quam porta tincidunt.

8.1 test

Nam egestas, libero eget sodales sodales, enim massa fringilla nulla, vel pharetra leo dui in tellus. Vestibulum sed justo eu quam porta tincidunt. Proin eleifend nulla sem, quis ultricies elit laoreet sed. Nulla venenatis fringilla luctus. Nullam porta turpis sit amet diam volutpat fermentum. Maecenas vehicula fringilla felis at varius.

8.1.1 test

Nam egestas, libero eget sodales sodales, enim massa fringilla nulla, vel pharetra leo dui in tellus. Vestibulum sed justo eu quam porta tincidunt. Proin eleifend nulla sem, quis ultricies elit laoreet sed. Nulla venenatis fringilla luctus. Nullam porta turpis sit amet diam volutpat fermentum.

Maecenas vehicula fringilla felis at varius.

8.1.2 test

Nam egestas, libero eget sodales sodales, enim massa fringilla nulla, vel pharetra leo dui in tellus. Vestibulum sed justo eu quam porta tincidunt. Proin eleifend nulla sem, quis ultricies elit laoreet sed. Nulla venenatis fringilla luctus. Nullam porta turpis sit amet diam volutpat fermentum. Maecenas vehicula fringilla felis at varius.

Bibliografia

Dodatki

